Archive of Maple User Group

MUG

Questions and Answers

Nasser M. Abbasi

September 7, 2024

Contents

CHAPTER 1

Introduction

This is my manual conversion of old and gone now Maple user groups answers.

These was orginally collected from MUG thanks to Dr. U. Klein `U.Klein@Math.RWTH-Aachen.DE`.

The original MUG and original collection is gone, but I had old copy saved. I was not able to contact Dr. U. Klein in order to check on its status.

So I am in the process of manually moving all MUG questions and answer to Latex to generate PDF and new HTML pages, which is taking long time and will take many more months to complete.

This is done by using only the public content of the questions and answers in the orginal MUG messages and starting brand new way of formating it into Latex. I tried to keep the same layout as the above collection and all information as it was originally.

This is work in progress.

Used LuaLatex for pdf and Tex4ht for conversion of Latex to HTML.

I kept all the content of questions and answer exactly the same, but improved the code formating so it is easier to read and copy.

CHAPTER 2

&

2.1 &(ampersand) operators, infixform / complain (9.10.01)

2.1.1 Bertfried Fauser

Below you will find an example of insensible design of mathematical structures in Maple. This is a pity, since Maple should take care especially about notation, since a wrong notation gives wrong prejudices and may cause even serious errors and prevents Maple from being a valuable tool in education.

Maple offers &(ampersand) operators working as a type of products. They do possess a prefix form `&op(a,...,b)` and an infix form (hence 'product') `a &op .. &op b`.

While the prefix form is pretty good in distinguishing its arguments, the infix form is very insensible.

The below given example is from Maple 5.1, but it can be reproduced in Maple 7 also.

We define operators &f and &s (the later was intended to be a tensor product, hence being flat, multilinear, but this does not conflict with our claim).

This operator (function) can take a sequence of arguments, which reads in the prefix form `&f(a1,..,an)`. Arguments are clearly separates in such a type of sequence.

BUT! in the infix form &f has to *decide* what it takes as arguments. However, this depends on the definition of the operator &f, and its assumed behaviour. The only pretty save way to implement this is to take any expression until the end of the expression or a second occurrence of &f. However, &f decides *wrongly* to take only the next *factor* !

1. Let &f be any operator:

   `&f(a+b,c+d) = (a+b) &f (c+d)` and no simplification should take place, but Maple (5 and 7) decide to produce

```
>a+b &f c+d;
                                  a + ( b &f c) + d
```

   which is nonsense since the operator is not allowed to do anything with its arguments which could be sums also.

2. Let &s be a linear associative (flat) operator, it ill-behaves in the same way with respect to the product.

```
> restart:
> define(`&s`,flat,multilinear):
```

   we consider infix form expressions of &s

```
> a &s b &s c;
                                     &s(a, b, c)
```

OK, that was easy, since nothing was to do,

BUT:

```
> a*b &s c*d;
                                  a (b &s c) d
```

NOTE the difference to

```
> (a*b) &s (c*d);
                                   a b &s c d
```

Moreover, &s is not aware, what it should treat as a second factor

```
> a &s b+c;
                                  (a &s b) + c
```

while it should come up with

```
a &s b + a &s c
```

since we assumed multi-linearity.

From the help-pages one learns that `'&'` binds stronger than anything but . `(respectively ||),::,%` BUT that belongs only to the name of the operator, i.e. one has not to write (&s) for the operator itself. (See online help to precedences).

The laconic comment of the Maple online help to `'&'` that 'Maple imposes no semantics (i.e. sense) on the neutral (i.e. &) operators' should be accompanied by the statement that it puts no syntax either.

This behaviour is much more annoying in the case of a 'defined' operator, which is supposed to behave e.g. flat and multilinear. Beside further problems with define, this makes this device very peculiar if not useless (for products, i.e. &(ampersand) operators).

E.g. a tensor product defined in this way needs a user who is aware of its properties. A student will fail to use this function or to learn from a robust behaviour of such a function what a tensor product can do and what not.

BUGFIX: The *intelligent* user sets parentheses where Maple does not know about the fact, or uses *exclusively* the stable prefix form.

2.1.2 Willard, Daniel Dr (15.10.01)

A partial observation: Maple has a priority sequence for the exercise of operators: the following execution of operators indicates that Maple ignores spaces in operator performance and treats `$f` as higher on the priority list than `+`:

```
>a+b &f c+d;
                                a + ( b &f c) + d
```

This explanation seems to fit some other anomalies quoted.

2.1.3 Bertfried Fauser (23.10.01)

Yes, I agree with your observation. However, the *define* utility of Maple suggests to use &(ampersand) operators (see online help-page). These operators are assumed to have no preassigned assertions like being associative, commutative, linear (in the first argument) or multi-linear. Regarding the above given example, the infix form does not make any sense with the define facility. This was the meaning of our posting.

Indeed, we stated already there that using parentheses is a 'bug fix' for this problem. However, it is annoying from a mathematical point of view and makes this device unusable in teaching (because you have to teach how the function works and students cannot test its behaviour since its not robust)

`&(ampersand)anything` should in its infix form take everything up to the end of input or the next occurrence of an `&(ampersand)`, this is possible, see e.g. the new 'error' in Maple 6 and 7.

2.2 &t operators (19.12.97)

2.2.1 Rafal Ablamowicz (23.10.01)

Thanks to all who have replied to my questions regarding the '&'- type operators in Maple. Your comments and suggestions have been very helpful.

In return, let me show you my code of an multilinear operator '&t' which is also left-associative. It gives the basic structure of a tensor product of orthogonal Clifford algebras. It is part of a small extension package G[raded] T[tensor] P[roduct] to my 'CLIFFORD' package to enable computations in graded tensor products of orthogonal Clifford algebras. These algebras are Z2-graded and their graded tensor products are Z2-isomorphic to orthogonal Clifford algebras in higher dimensions. These strange types tensorprod, clipolynom, cliscalar are defined in the two packages.

If anyone is interested in these packages, please e-mail me.

```
GTP[`&t`]:=
proc(a1::{`*`,`+`,tensorprod,clipolynom,cliscalar},a2::{cliscalar,clipolynom})
local co,a11,a22,p,i;
if nargs<2 then ERROR(`at least two arguments are needed`) fi;
if nargs>2 then RETURN(`&t`(`&t`(args[1..2]),args[3..nargs])) fi;
if type(a1,`+`) then RETURN(map(`&t`,a1,a2))
elif type(a2,`+`) then RETURN(map2(`&t`,a1,a2)) fi;
if type(a1,`*`) and hastype(a1,tensorprod) then
                    a11:=select(type,a1,tensorprod):
                    co:=remove(type,a1,tensorprod):
                    RETURN(co*(`&t`(a11,a2)))
fi;
if type(a1,`*`) and not hastype(a1,tensorprod) then a11:=select(type,a1,clibasmon):
                     co:=remove(type,a1,clibasmon):
                     RETURN(co*(`&t`(a11,a2)))
elif type(a2,`*`) and not hastype(a2,tensorprod) then a22:=select(type,a2,clibasmon):
                       co:=remove(type,a2,clibasmon):
                       RETURN(co*(`&t`(a1,a22)))
fi;
p:=args[1];for i from 2 to nargs do p:='`&t`'(p,args[i]) od;
RETURN(p);
end:
```

CHAPTER 3

0

3.1 0.i in maple 6 (17.10.00)

3.1.1 Theodore Kolokolnikov (23.10.01)

```
> evalf(0.*I);
                                  0. I
```

It would be better if Maple would produce "0." instead.

I mean, I know that Maple now treats complex numbers differently than reals; but still...

```
> evalb(0.*I=0.);
                                  true
```

3.1.2 Andrzej Pindor (19.10.00)

This is another bizzare consequence of the new way chosen to represent complex numbers. Note that:

```
    |\^/|     Maple 6 (SUN SPARC SOLARIS)
._|\|   |/|_. Copyright (c) 2000 by Waterloo Maple Inc.
 \  MAPLE  /  All rights reserved. Maple is a registered trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
```

```
> op(0.*I);
                                    0.

> whattype(0.*I);
                                  complex
```

To retain the complex type Maple insist on keeping "I" in the expression. Interestingly

```
> op(2+3*I);
                                  2, 3

> op(0+3*I);
                                    3
```

Would it be not more consistent if the result of `op(0+3*I)` was `0,3` ? In particular, that

```
> op(0. + 0.*I);
                                 0., 0.

> op(0.+0*I);
                                  0, 0
```

Why then:

```
> op(0+0*I);
                                    0

> op(0+0.*I);
                                    0.
```

3.1.3 David Harrington (19.10.00)

The help under fnormal suggests that this is deliberate, and gives a remedy:

"fnormal preserves numeric type and sign information as much as possible. Thus, for example, `fnormal(1e-20*I, 10) = 0.*I` not `0.` . This ensures that branching behaviour is generally not affected by fnormal. Use simplify(expr, zero) to remove 0 real or imaginary parts of complex floating point numbers."

CHAPTER 4

2

4.1 2d tangents (27.5.02)

4.1.1 Dr. U. Kasper (23.10.01)

Does anyone outside there know of a package helping to plot the tangent to a 2d curve (not being a circle) at a given point?

4.1.2 Robert Israel (30.5.02)

In Maple 8 you could compute the tangent line using the procedure TangentLine in the new VectorCalculus package.

Otherwise, just compute it yourself: if it's a parametric curve `x = X(t), y = Y(t)`, the tangent line at `t=t0` is parametrically

```
x = X(t0) + D(X)(t0)*s, y = Y(t0) + D(Y)(t0)*s
```

Or if the curve is given implicitly by `F(x,y) = c` and `[x0,y0]` is a point on the curve where the gradient of F is nonzero, the tangent line is given implicitly by

```
(x-x0)*D[1](F)(x0,y0) + (y-y0)*D[2](F)(x0,y0) = 0
```

or parametrically by

```
x = x0 + s*D[2](F)(x0,y0), y = y0 - s*D[1](F)(x0,y0)
```

CHAPTER 5

3

5.1 3-d plot of a tetrahedron (6.9.00)

5.1.1 John J. Reinmann (23.10.01)

Can anyone show me an efficient way to get a 3-D plot of the tetrahedron formed by the following four points in 3-space:

```
(1,4,-1), (2,0,0), (-1,-4,3), (4,5,7) ?
```

I would like each surface to have a different color.

5.1.2 Robert Israel (7.9.00)

Sure.

```
pts:= [[1,4,-1], [2,0,0], [-1,-4,3], [4,5,7]];
triples:= combinat[choose](pts, 3);
colours:= [seq(colour=COLOUR(RGB,op(j)), j=_COLORRGB)];
plots[display](zip(plots[polygonplot3d], triples, colours),
  scaling=constrained);
```

5.1.3 Carl DeVore (7.9.00)

Here's a proc to draw any tetrahedron with any four colors.

```
DrawTetrahedron:= proc(Pts,Colors)
   local it,k;
   it:= combstruct[iterstructs](Combination(Pts), size= 3);
   RETURN
      (plots[display]
         ([seq
            (plots[display]
               (PLOT3D(POLYGONS(combstruct[nextstruct](it)))
               ,color= Colors[k]
               )
            ,k= 1..4
            )
          ], args[3..nargs]
         )
      )
end:

DrawTetrahedron
   ([[1,4,-1],[2,0,0],[-1,-4,3],[4,5,7]]
   ,[red,green,blue,yellow]
   ,axes= boxed
   );
```

5.1.4 John S Robinson (8.9.00)

This works:

```
> with(combinat):
> vertices:=[[1,4,-1], [2,0,0], [-1,-4,3], [4,5,7]];

     vertices := [[1, 4, -1], [2, 0, 0], [-1, -4, 3], [4, 5, 7]]

> faces:=choose(vertices,3);

  faces := [[[1, 4, -1], [2, 0, 0], [-1, -4, 3]],
```

```
        [[1, 4, -1], [2, 0, 0], [4, 5, 7]],

        [[1, 4, -1], [-1, -4, 3], [4, 5, 7]],

        [[2, 0, 0], [-1, -4, 3], [4, 5, 7]]]

> PLOT3D(POLYGONS(op(faces)));
```

... but it was lucky you wanted a tetrahedron, since all triples of vertices are the faces.

I wish there were some way to plug our own co-ordinates into plotpolyhedra().

5.1.5 Klaus Volpert (12.9.00)

```
with(plots):with(plottools):
 a:=[1,4,-1]:
 b:=[2,0,0]:
 c:=[-1,-4,3]:
 d:=[4,5,7]:
 side1:=polygon([a,b,c],color=green):
 side2:=polygon([a,b,d],color=red):
 side3:=polygon([a,c,d],color=blue):
 side4:=polygon([b,c,d],color=gold):
 display([side1,side2,side3,side4]);
```

5.1.6 Helmut Kahovec (14.9.00)

You may use gtetrahedron() of the geom3d package:

```
restart;
with(geom3d):
Warning, the name polar has been redefined

point(P1,1,4,-1),point(P2,2,0,0),point(P3,-1,-4,3),point(P4,4,5,7);

                              P1, P2, P3, P4

gtetrahedron(T,[P1,P2,P3,P4]);
```

```
                                    T

draw(T,axes=BOXED,scaling=UNCONSTRAINED,labels=[x,y,z]);
```

5.2 3d calculation from v5 takes 60times more on v7 (15.10.01)

5.2.1 Martyn Caron (23.10.01)

We had buy a v7 version of Maple. All my calculation that was working good on v5 take up to 60 times more on v7. My PC is an PIII/866, on Windows 98 SE. Ex. Takes 3 second on v5 takes 180 seconds on v7. Here is an exemple of the calculation:

```
restart;with(plots):with(plottools):
`De forme avec des cylindres`;
 volume:=cuboid([0,0,0],[4,4,4],color=green):
 f1:=cylinder([2,2,3.9],0.5,0.15,color=pink):
 f3:=seq(rotate(cylinder([3.9,i,i],0.4,0.15,color=pink),Pi/2,[[3.9,0,i],
         [3.9,4,i]]),i=1..3):
 f6:=seq(seq(cylinder([i,j,-0.05],0.4,0.15,color=pink),i=1..3),j=[1,3]):
 display(volume,f1,f3,f6,scaling=constrained,axes=none,orientation=[-40,70]);
 c1:=cylinder([0,0,0],1,1,color=red):
 rotate(c1,Pi/2,[[1,1,1],[2,2,2]]);
```

5.2.2 Frank Gerdes (15.10.01)

Because I am thinking about upgrading from Maple 6 to Maple 7, I did the test with Martyn's code.

My system is a PC with WIN 98 SE, 366MHz, 128MB.

Maple V Release 5.1: 85 seconds

Maple 6.02: 193 seconds

The perfomance difference is not as dramatic as the one reported by Martyn, but certainly significant.

I made sure to quit Maple V version before I started Maple 6 and vise versa. I also repeated the test after rebooting, etc.

Now, I am wondering whether Martyn's result with Maple 7 tell me that I should stay away from Maple 7.

5.2.3 Robert Israel (16.10.01)

Examination of the code of `plottools[rotate]` shows an obvious inefficiency: the transformation involves a function that is hellishly complicated, but should be very simple. One possible patch is to make `a1, a2, a3` local variables and change line 41:

```
f := proc (a1, a2, a3) ...etc... end proc
```

to

```
f:= unapply(...etc..., a1,a2,a3)
```

5.2.4 Thomas Richard (26.10.01)

Frank Gerdes wrote:

`| I made sure to quit Maple V version before I started Maple 6 ...`

This should be unnecessary for such comparisons (except in cases with extreme memory consumption).

`| Now, I am wondering whether Martyn's result with Maple 7 tell me ...`

Wrapping the last rotate statement in display() (probably intended so), I obtain these timings (P-III 700MHz, NT4 SP6, Matrox G200 AGP):

```
Maple 7 : 53.326s

Maple 6.02a: 51.363s

Release 5.1: 32.305s
```

In Release 5.1, the graphics are not as accurate as in newer versions: the dice marks are not completely filled, and the grid lines are less regularly distributed on the cylinder. Maybe a matter of taste...

5.2.5 Frank Gerdes (25.10.01)

Meanwhile I got Maple 7 and found that it is as slow as Maple 6 when executing Martyn's code. I mean slow compared to Maple V R5.1 (see previous postings below).

I had a look at the code of `plottools[rotate]` and I agree that is is very complicated and that it might be inefficient. But I am fairly sure that Maple V R5.1, Maple 6 and Maple 7 use the same code. I did a visual inspection and could not find any signficant differences.

So, that leaves me still puzzled why Maple 6 and Maple 7 are so much slower when executing `plottools/rotate'-code` than Maple V R5.1.

P.S.: I suppose most of the time one doesn't notice these things, because PCs seem to get faster at a slightly larger rate than software gets slower.

I got the following timings: (U. Klein)

```
MapleV.5.1: 42.899s

Maple6.01: 77.361s

Maple7: 85.040s

Maple8: 4.210s
```

5.3 3d graph, rotate after saving (3.1.02)

5.3.1 Erik Leunissen

Is it possible to save a 3d picture in a more or less standard format such that interactive rotation (either by maple or another application) remains possible?

5.3.2 Gerald A. Edgar (4.1.02)

When you have a 3D plot window, the Export menu lists some 2D output formats, but also 3D formats DXF and POV. A POV file can then be rendered by the free programs POV-Ray or MegaPOV. But these programs do not seem to export to any other 3D formats. A quick web search did not find any conversion programs from POV to other 3D formats.

Looking at a POV file shows that it is a text file, and the syntax is readable, so conversion programs should be possible.

What about DXF? Unlike POV, most 3D programs *can* import and export DXF. So I guess if you want Maple to output 3D information (and you cannot use POV files directly), then DXF format is the one to use.

The drawback is that DXF has no colors, just triangles.

Perhaps some enterprising person will write a parser for POV into a more common color 3D format (3DMF, OBJ, 3DS). Or maybe it would be easier to parse Maple's own plot3D objects?

5.3.3 Frank Gerdes (4.1.02)

Perhaps the JavaView library for Maple is what you want. See `www.javaview.de`.

It works with Maple 6 and Maple 7. I think there are different installation files for Win98 and Win2000/NT.

5.3.4 Thomas Richard (7.1.02)

DXF is a CAD format, and thus has its limitations, but rotating, zooming etc. remains possible. It was introduced in Maple 6 (and broken in Maple 6.01 (but not in 6 / 6.02 / 7)).

We use Autodesk VoloView Express for viewing.

5.3.5 Dr Francis J. Wright (7.1.02)

You can save in VRML (Virtual Reality Modeling Language) format using `plottools[vrml]`. The result can be viewed and rotated in real time by a VRML browser, such as Cosmo Player which is free. See section 4.6 of my book "Computing with Maple"; the VRML example from the book is on the book web site at `http://centaur.maths.qmw.ac.uk/CwM/`.

5.3.6 John Richardson (17.1.02)

Dr Francis J. Wright wrote: ...

COSMO PLAYER IS DEAD.....DEAD.....DEAD.....DEAD.....

The current VRML 97 commercial viewer is Cortona for the Macintosh and either Contact or Cortona for the PC.

`http://www.blaxxun.com`

`http://www.parallelgraphics.com`

There are open source viewers for the PC, Macintosh and UNIX.

`http://www.web3d.org` for more info.

I just returned from Macworld where I showed off a speech enabled multimedia UI for the OpenVRML open source VRML 97 viewer. It is really cool. It is a modification of the MacLookat OpenVRML viewer called SimVRMLLookat. The Maclookat application and source can be found at Source Forge. The SimVRMLLookat source will be up on the site later this month.

OpenVRML is available for the PC and UNIX also. Speech is only available on the Mac. Eat your hearts out PC people.....:)

`http://sourceforge.net/projects/openvrml/`

Note: YES, Cosmo Player was realy great and cool in it's time.

Maclookat is for OS 9. You can download the double clickable application from the sourceforge site. You can also build it in Codewarrior 5 under OS 9. Nobody has been successful in building it with Codewarrior 6 or Codewarrior 7. The build requires some JPEG libraries for Version 0.10.x.

Note: MacLookat does not have a user guide. So, using the option key is the walk mode and the shift key is the Pan mode. "a" is slide up. "z" is slide down. "." and "," is rotate. There are also arrow key modes. Just plain click and drag is "rotate/tilt" mode.

The Mac OS X version is being developed by Michael Louka. He originally developed MacLookat.

However, if you want to amuse yourself with a modified Maclookat, call SimVRML-Lookat, which is speech enabled and is ultra cool I can send it to you in double clickable form. SimVRMLLookat has an API that is currently tailored towards Apple Events, SuperCard and Labview. Also, RoboLab and Lego Mindstormes.

Note that Apple Events are isolated in Classic. So the apps run under classic but the Apple Events fall into a black hole.

Also, any other Maple users who crave 3-D viewers, feel free to request SimVRMLLookat. My goal is to post SimVRMLLookat on the source forge web site by Jan 25.

My solution for VRML 1.0 is to try and load it into Spazz3D 2.4 running under Virtual PC 4 or 5. Note that VPC 5 requires 9.1 or 9.2 or OS X. Then if it loads, I just re-export it to VRML 97 from Spazz3D.

Of course, VPC with Win 98 is $175. Spazz3D is free for 2 weeks then disables itself. Stay away from VPC with NT. `http://www.spazz3d.com`

If it does not load into Spazz3D there are several options. Spazz3D may tell you why it does not load. If so, then edit with Microsoft word and correct the illegal field which is most likely the case.

Spotlight is a GLARING example of VRML 1 nodes with problems in VRML 97.

Another option is to load it into Amapi 3D 6 or higher. The company that developed Amapi, TGS, makes commercial Open Inventor libraries and VRML was developed

from Inventor. Amapi will import VRML 1 and VRML 97. Amapi is in the $300 range. Amapi is maintained by TGS's subsidiary, Eovia.

The advantage of this solution is that you can export to Shockwave 3D and VET (Adobe Atmosphere). `http://www.eovia.com`

Another Option is Meshworks. Look on the Macweb3d site. It is free and may inport VRML 1.

Mac OS 8.

The main problem is OpenGL. You have to download it from Apple and load it yourself. I would be interested to hear your results on using MacLookat under OS 8. Just curious, why do you use OS 8?

Note: for the Waterloo Maple programmers that troll this list you should upgrade the export to handle VRML 97. While you are at it, make sure you pay attention to the Universal Media solution for texture export or any textures you export will view differently in different viewers.

Note: Nexternet may have licensed the Cosmo Player source code and re-released it. De-install Cosmo befor installing Pivoron, their player. They are partnered with Cybelius which is PC only so may not be available on the Mac. Their site is difficult.

5.4 3d graphs and html (8.6.01)

5.4.1 Peter Lindsay

I've got a "3D" graph using surfdata which rotates about etc in Maple6.

Is there any way I can put this on the web so that users can get this to "rotate" as viewed through a browser?

5.4.2 Gerald A. Edgar (11.6.01)

There is "LiveMath"... `http://www.livemath.com/` but you would have to put your graph into LiveMath Maker, and your users would have to get the free LiveMath plug-in. That may be more effort than it is worth?

Here is an attempt of mine to do something like this:

`http://www.math.ohio-state.edu/~edgar/H263/final.html`

see Problem 9 near the bottom. The graph was exported from Maple as DXF, then rendered into a QTVR object. The advantage is that the user needs only the Quicktime plug-in, which many already have.

5.4.3 Maerivoet, Roland (12.6.01)

If you mean by surfdata: saved ("get") an animation that was made with Maple 6 (and exported as html), then you "put" it in the webdirectory next to the webpage where it is used. Otherwise, having generated the animation in a Maple-worksheet: export to html etc...

5.4.4 Norbert Van den Bergh (12.6.01)

Export your worksheet as html document and manipulate the resulting gif file with an animation package (for example Paint Shop) to get (1) a decent compression rate and (2) any other extra's you would prefer.

5.4.5 Nadarajah, Thirugnanasothy (12.6.01)

What about VRML?

VRML models can be viewed with plug-ins and without plug-ins (there are restrictions)

If you are interested, I can give more details later. It may not be the best option. But I think it should work.

5.4.6 Robert Israel (12.6.01)

I believe this can be done using JavaView. See `http://www.cecm.sfu.ca/projects/webDemo/htm/featur.htm`

According to that web page:

JavaView provides a superior viewing environment to augment and enhance the plot of geometrical objects in Maple. It provides several features that are non-existent in the Maple plotter, such as mouse controlled scaling, translation, and auto view modes. JavaView implements arc-ball rotation, making object viewing smoother and less directionally constrained than in Maple.

Furthermore, JavaView offers a point modeling feature that allows plots to be manually manipulated.

The predominant feature of the JavaviewLib is the capacity to export Maple generated models into one of two applet based viewers - one optimized for speed, the other for customizability. This greatly enhances the current state of plot object export in Maple - no longer do dynamic plots need to be converted to static images when creating html pages from Maple worksheets. One can also export plot data to a variety of other formats such as VMRL or JavaViews own XML format where data can be viewed as a markup tree or further developed upon.

Efforts were made to maintain the aesthetic presentation of Maple geometries and their corresponding axes upon export. With JavaviewLib, models created in other modeling applications such as Maya and Mathematica, can easily be imported into Maple's viewing environment.

5.4.7 Dr. TANAKA, Kazuo (13.6.01)

Though there may not be the way how to manage it with Maple, there is a non-commercial Java applet to do with Mathematica.

That is named LiveGraphics3D.

Please refer to

`http://wwwvis.informatik.uni-stuttgart.de/~kraus/LiveGraphics3D/`

5.5 3d plot as a standalone object (17.2.00)

5.5.1 Bernard Marcheterre

Does anyone know how to transform a 3Dplot created from Maple into a standalone object that could be opened without Maple but in which you could still move the plot in all directions. I heard this could be done via VRML but have no idea of how or where to start...

5.5.2 Herman Jaramillo (22.2.00)

Here is a piece of program that I took from the Gallery section on the Maple web page site, without permission.

Reference: Gilbert Labelle, LACIM-UQAM, Janvier 1991

```
setop

pic :=plot3d([u*cos(v),u*sin(v),u+v],u=-Pi..Pi,v=-2*Pi..2*Pi,
   grid=[60,60],orientation=[0,57],style=patchnogrid,
   color=COLOR(RGB,.9000000000, .7490196100, .09803921500),
   lightmodel=light4,shading=XY,scaling=constrained):
display(pic)

# plottools[vrml](pic,`22.wrl`);
```

That example might be useful.

5.6 3d plot help (5.10.00)

5.6.1 Jason C. Leach

Can anyone give me a few tips on plotting in 3d? I'm doing a class on Calc with 3 variables and would like to use maple to better visualize some of the problems.

For example, if I have 3 points (3,0.0), (0,2,0), and (0,0,4) that make up a triangle, I'd like to plot the 3 vectors between them. Then, I might do something like find the distance from a 4th point to that triangle; or perhaps from a point to a line.

Perhaps I should be asking, if I have several vectors, how can I plot them in 3d.

While I'm at it, can maple deal with error values that one typically finds in physics. For example I'd like to add: 3.0 (plus or minus) 1.0, and 4.1 (plus or minus) 2.1.

I think I'm just going to have to pick up a maple book.

5.6.2 Mike May (9.10.00)

Let me give a number of fundamentally different paths depending on how you want to use Maple for teaching and how much Maple you already know.

The best package of Maple macros for teaching Multi-variable calculus that I know o is the mvcal2 package by Cheung, et. al. It can be obtained from

`http://fmwww.bc.edu/MT/gopher/maple/mvcal2/dir.html`

Among other commands, the package has a `drawvector3d` command. I would recommend the package route if you are planning to use Maple almost entirely for demonstrations. (Maple is then something that the students do not learn.) You should also be fairly good with Maple yourself. You may also want to use the package approach if your program assumes that the students will learn how to do serious programming with Maple.

A second route is to use worksheets both for demonstrations and for student activities. I have been using Maple that way for multi-variable calculus for a number of years now. My worksheets are available via anonymous ftp from `euler.slu.edu`. I have about 30 worksheets for multi-variable calculus, organized according to the McCallum book. You sound like you would be interested in the preliminary worksheets that deal with introducing the students to Maple and to plotting.

Chapter 15 has worksheets on plotting in alternate coordinate systems and chapter 12 has worksheets on drawing vectors. The worksheets can also be found at

`http://euler.slu.edu/Dept/Courses/Coursewaredevelopment.html`

I like the worksheet model because it allows me to use Maple for fairly complicated graphics without teaching the students a lot of programming. I have also found it to be a good way to make material accessible to other faculty here.

Finally, since you mentioned picking up a Maple book, I can't resist giving a plug for the one I co-authored. Getting Started With Maple by Cheung, Keough, and May is sold by Wiley press. It is aimed at an introductory level.

5.7 3d plots to dxf (28.2.00)

5.7.1 Jochen Mauch (9.10.00)

I have a funny idea: I want to use Maple 3D plots within Kinetix 3D Studio. The stupid thing is that Maple has no support for DXF output, and 3D Studio cannot import VRML files!

5.7.2 Peter Hajek (29.2.00)

Please visit: `www.maplesoft.com/sales/freetrialcd.html`

to request a free trial of Maple 6, now with DXF export.

If you need something in the meantime, please send me an email directly and lets see what we can do for you.

5.7.3 John F. Richardson (1.3.00)

Try `http://www.Web3d.org`. This site can lead you to all the major software systems, some free, that can translate between 3D file formats such as VRML, DXF and 3DS (studio max). You can also try `http://web3d.about.com`.

Since I believe that the Maple VRML exporter is homegrown you might want to check the sites for applications that check VRML files for correct syntax. If you don't have any sophisticated textures the translation process should be consistent from VRML to DXF/3DS.

5.7.4 Mohamed Bendame (1.3.00)

With Maple6 you can now export Maple 3D plots as DXF files which can be opened in CAD packages such as AutoCAD. This is one of the new features of the new release, and I am sure lot of people out there will find it very exciting. For further info on what's new in Maple6 check the URL below:

`http://www.maplesoft.com/products/Maple6/whatsnew.html`

5.8 3d plots of data from files (28.2.00)

5.8.1 H. Mark Hubey (1.3.00)

How does one create a 3D plot in maple of points generated for a 3D surface using another programming language such as Java or C++. IOW, I want to use Maple to read a file of data for a 3D plot.

5.8.2 Ben Bolker (29.2.00)

readdata and listplot3d

5.8.3 Norbert Roth (1.3.00)

In Maple the `'plot[pointplot]'` command is used to generate 3D point plots. The provided data should be a set or a list.

E.g.

If your data-file look like this:

```
 2.7 0 1
 1 2 1
 2 0 1
-1 0 0
 3 1.5 -2
```

then this creates a 3D-point plot:

```
restart;
readlib(readdata):
l:=readdata(`ram:plot1.data`,3):
with(plots):
pointplot(l,axes=frame,title=`testplot`);
```

Additional options (axes, title,labels, orientation, projection, view, color,...) are available.

For a surface plot have a look at this:

```
restart;
```

```
cosdata:=[seq([ seq([i,j,evalf(cos((i+j)/5))], i=-3..3)],j=3..3)];
```

cosdata is a list of lists

```
with(plots):
surfdata(cosdata,axes=frame,style=patch,title=`testplot`);
```

several other options are available - see plot/plot3d

So if you provide the surface data in an appropriate format, it should be easy to generate an impressive plot.

5.9 3d plots with patch (10.5.00)

5.9.1 Sandy Yates

I am plotting a function where I need an 100x100 grid for the 3D plot to look reasonable. My problem is that I would like the see the plot with PATCH.

When I plot it with PATCH the patching is so dense that the ZHUE cannot be seen as the plt is very black. Is it possible to plot every second or 3rd PATCH line? Or is there another method to getting a lower PATCH grid compared the to plot grid? (I realize why maple may not want to do that as the line of te patch may cut the surface.)

5.9.2 Robert Israel (11.5.00)

My suggestion would be to combine (using display) a plot using patchnogrid and plots of grid lines. It would be simplest to do the latter using spacecurve, but if you want, you can extract them from the plot structure, e.g. as follows:

```
p1:= plot3d(f(x,y), x=a..b, y=c..d, grid=[100,100], style=patchnogrid):
A:= op([1,3],p1): # this should be the hfarray of z values
dx:= (b-a)/99: dy:= (d-c)/99:
lines:= seq([seq([a+(i-1)*dx, c+3*(j-1)*dy, A[i,3*j-2]], i=1..100)],
            j=1..34),
        seq([seq([a+3*(i-1)*dx, c+(j-1)*dy, A[3*i-2,j]], j=1..100)],
            i=1..34):
plots[display]({p1, PLOT3D(CURVES(lines),COLOR(RGB,0,0,0))});
```

5.9.3 John S Robinson (12.5.00)

If I understand correctly, you can get round the problem by combining two plots, as in:

```
p1:=plot3d((x*y)+2*x+3+y,x=-1..2,y=-2..1,grid=[100,100]):
display(p1);
```

this is all black

```
p2:=plot3d((x*y)+2*x+3+y,x=-1..2,y=-2..1,grid=[100,100],style=PATCHNOGRID):
display(p2);

p3:=plot3d((x*y)+2*x+3+y,x=-1..2,y=-2..1,grid=[10,10],style=WIREFRAME,colour=BLACK):
```

```
display(p3);
display({p2,p3});
```

this seems OK

5.9.4 Jurgen Barsuhn (15.5.00)

As far as I know there is no possibility to omit selected grid lines. However, you may omit all grid lines by choosing patchnogrid from the style menu (or by including the option style=patchnogrid into the plot3d-statement).

To improve the impression you may include contour lines that may be seen under style-patch and contour. Here you can either choose the number of contour lines or you may specify a list of function values, for which you wish to see contour lines.

An example of a summit and two saddles:

```
f:=16-2*x^2-4*y^2+x^2*y;
plot3d(f,x=-5..5,y=-4..4,view=-16..16,grid=[50,50],shading=ZHUE,
       style=patchnogrid);
plot3d(f,x=-5..5,y=-4..4,view=-16..16,grid=[50,50],shading=ZHUE,
       style=patchcontour,contours=6);
plot3d(f,x=-5..5,y=-4..4,view=-16..16,grid=[50,50],shading=ZHUE,
       style=patchcontour,contours=[-2,0,2]);
```

5.10 3d surface plotting from data points (14.5.02)

5.10.1 Darrell Barabash (15.5.00)

After many attempts with Maple 6 over the last couple days, I give up. I have some numeric data which I want to plot as a 3D surface. Ideally, I would like to plot it as contours. The problem is that I can only get 3D point plots to work and have been unable to figure out how to get it to display as a surface. The data is a list of `[x,y,z]` points where x and y are the independent variables.

```
data := [[50.7385, .8424, 17.76], [60.1174, 7.2830, 17.96],
         [61.4359, -2.9078, 17.65], [54.5203, -9.5282, 16.17],
         [46.5648, -8.8578, 17.68], [41.7216, -3.8881, 17.46],
         [40.8820, 2.5928, 16.98], [44.3202, 8.6542, 17.04],
         [51.9302, 11.2453, 17.61], [59.8513, 6.7143, 16.29],
         [70.7850, 14.2986, 17.74], [70.6260, 14.5001, 17.42],
         [62.0037, 20.7985, 17.23], [52.7317, 22.0789, 16.38],
         [45.2121, 19.9142, 17.05], [39.5056, 15.7564, 16.74],
         [35.9793, 10.8419, 16.69], [34.0844, 5.6868, 16.95],
         [33.5315, .5582, 17.19], [34.2978, -4.4322, 14.59],
         [36.2701, -9.2649, 13.52], [39.5079, -13.6210, 17.81],
         [44.2834, -17.3221, 17.53], [50.7196, -19.7107, 17.29],
         [58.8775, -19.7284, 17.37], [67.6327, -15.8741, 17.35],
         [74.6338, -7.3103, 17.44], [76.3174, 4.6175, 17.49],
         [80.6759, -25.1588, 17.17], [29.2350, 11.9718, 16.26],
         [31.0006, 16.4310, 15.96], [71.2992, 32.7692, 16.88]]
```

I have tried the pointplot3d function as well as "surfdata" (which I am, no doubt, using incorrectly). In all cases the best that I can do is get a scattergram of 3D points.

5.10.2 Robert Israel (15.5.02)

"surfdata" would be the function to use if your data were organized in rows and columns. This doesn't seem to be the case. However, looking at the projection on the xy plane it seems that you have points 2 to 10 in a closed curve surrounding point 1, points 11 to 28 in a closed curve around that in the opposite direction, and then points 29 to 32 outside that, with point 29 near 26, 30 and 31 near 17, and 32 near 13. So I might do it this way:

```
L1:= [data[1]$19]:L2:= [seq(data[12-i]$2,i=2..10),data[10]]:
L3:= [seq(data[i],i=11..28),data[11]]:
u3:= [FAIL$3]:
L4:= [u3$2,data[32],u3$2,data[31],data[30],u3$8,data[29],u3$3]:
```

5.10.3 Dr. TANAKA, Kazuo (16.5.02)

or polygonplot3d with style=patch and after drawing the figure select "Patch and Contour" from task bar "style".

```
with(plots):
polygonplot3d([[50.7385, .8424, 17.76], [60.1174, 7.2830, 17.96],
[61.4359, -2.9078, 17.65], [54.5203, -9.5282, 16.17], [46.5648, -8.8578,
17.68], [41.7216, -3.8881, 17.46], [40.8820, 2.5928, 16.98], [44.3202,
8.6542, 17.04], [51.9302, 11.2453, 17.61], [59.8513, 6.7143, 16.29],
[70.7850, 14.2986, 17.74], [70.6260, 14.5001, 17.42], [62.0037, 20.7985,
17.23], [52.7317, 22.0789, 16.38], [45.2121, 19.9142, 17.05], [39.5056,
15.7564, 16.74], [35.9793, 10.8419, 16.69], [34.0844, 5.6868, 16.95],
[33.5315, .5582, 17.19], [34.2978, -4.4322, 14.59], [36.2701, -9.2649,
13.52], [39.5079, -13.6210, 17.81], [44.2834, -17.3221, 17.53], [50.7196,
-19.7107, 17.29], [58.8775, -19.7284, 17.37], [67.6327, -15.8741, 17.35],
[74.6338, -7.3103, 17.44], [76.3174, 4.6175, 17.49], [80.6759, -25.1588,
17.17], [29.2350, 11.9718, 16.26], [31.0006, 16.4310, 15.96], [71.2992,
32.7692, 16.88]],axes=boxed,style=contour);
```

CHAPTER 6

A

6.1 a book of release 4 worksheets (8.8.96)

6.1.1 Carl Eberhart

I have prepared a 'book of worksheets' for my Calc 2 students this fall and decided to make it available to anyone who wants to use it. A tar file is available at `http://www.ms.uky.edu/~carl` Below is the readme that comes with it.

These MapleVr4 worksheets contain the manual Problem Solving with Maple: A handbook for Calculus Students. by Carl Eberhart, Dept of Mathematics, University of Kentucky. This 'book of worksheets' is meant to help calculus students learn how to use the language Maple to solve problems.

Maple, like any language, is most easily learned on a 'need to know' basis by observing how others with proficiency in the language use it and interacting with or modifying samples of their usage.

This handbook provides lots of samples of problems and their solutions for you to inspect and modify.

The worksheet work2.mws contains a 'table of contents' with hyperlinks to the other worksheets. Just click on the link to go to the topic you want. Each worksheet has several bookmarks which can be accessed from the View menu at the top of the worksheet.

In addition, there is an index worksheet which contains an alphabetized list of all the bookmarks in the handbook, with links to the worksheet containing the bookmark. Just click on the topic you want.

I have referenced lots of uses of Maple words like plot, solve, diff, and others together

with lots of specific problems and their solutions, concepts that come up in calculus, etc.

I have included a version of a perl script I wrote to construct the index worksheet for the 'book of worksheets'. It has been tested in Windows 95 and also unix, although I cannot say whether the worksheets the unix version of MapleVr4 since it is not yet available.

Earlier versions of this handbook were written with the support of the NSF Calculus Reform Program grant (DUE-9252494).

If you don't have netscape, but do have ftp, you can get a zipfile of the worksheets by anonymous ftp from `ftp.ms.uky.edu`

cd to `ATE/ma503` and look for `handf96.zip`

6.2 $\frac{a}{b}$ and ab^{-1} sometimes differ (26.9.97)

6.2.1 Harold P. Boas

I was intrigued to discover that Maple evaluates

```
5/6.0    and    5*6.0^(-1)
```

to floating-point numbers that differ by 2 units in the last decimal place. The on-line help for arithmetic operations states that

`a/b` is of type '*' with operands a and `b^(-1)`

but evidently this does not tell the whole story.

I am guessing that Maple evaluates `5/6.0` by invoking some built-in algorithm for floating-point division, while Maple evaluates `5*6.0^(-1)` by first finding a floating-point approximation to `1/6`, and then multiplying by 5. Is this a correct conjecture?

An amusing corollary of this phenomenon is that Maple's floating-point evaluation is "non-commutative": for example, Maple evaluates

```
5/6.0 and 5.0/6
```

to different results.

What are the precise circumstances under which Maple does treat `a/b` and `a*b^(-1)` as equal?

6.2.2 Robert Israel (6.10.97)

Looks right to me. Actually this is happening during the preliminary "simplification" which is performed on all Maple input before any evaluation takes place. Thus if you do this with variables, there is no difference:

```
a:= 5; b:= 6.0;
a/b = a*b^(-1);

                .8333333333 = .8333333333
```

In general floating-point computation is non-commutative. The simplest example is that A + B - A returns 0 when A is much larger than B. Mind you, that _won't_

necessarily be true if the automatic simplifier is allowed to do its magic:

```
> 1.0e17 + 1 - 1.0e17;
          0
```

but

```
a:= 1.0e17: b:= 1:
a + b - a;  # this will be simplified to b before the arithmetic is performed.
          1
```

6.3 abc-conjecture (12.9.96)

6.3.1 John B. Cosgrave

In the coming academic year I want to try to explain to my students the so-called 'abc-conjecture', a conjecture (initially by Oesterle', and refined by Masser) of the past decade which is rightly considered to be a truly profound one, with many deep consequences (see - for example - Dorian Goldfeld's article "Beyond the Last Theorem" in the March/April '96 issue of "The Sciences" (New York), or the article by Fields medallist Alan Baker in the recent centennial issue of the 'Mathematical Gazette').

Because of the help that I got in connection with my square-free question, and the related 'product' question, I am now in a position to experiment with the following programme, whose 'meaning' is simply this: one is trying to find relatively prime values of 'a' and 'b' (i.e. igcd(a, b)=1) such that their sum 'c' has the PROPERTY that the square-free part of a*b*c DIVIDED by 'c' has a 'small' value (in the procedure below, 'small' means 'less than 1'):

```
with(numtheory): # this is needed for 'factorset' in the following

small_Masser:=proc(n)  # the 'Masser' is explained below
local a, b;  # one doesn't need an extra local 'c'
for a to n do
for b to (a-1) do
if igcd(a,b)=1 and
convert(factorset(a*b*(a+b)),`*`)/(a+b) < 1
then print(a, b, convert(factorset(a*b*(a+b)),`*`)/(a+b))
fi od od end:
```

Comment One: My computing facilities don't allow me to let 'n' get too large. n=500 took 1452 seconds on my 486. The last of the outputs - for n=500 - was:

```
                    343, 169, 91/256
```

the SIGNIFICANCE of which you should sense from:

```
                    343 + 169 = 512,
namely:             7^3 + 13^2 = 2^9.
```

They are not, of course, all as structured as that one. Here is another:

```
                        343, 32, 14/25

which is related to:    343 + 32 = 375,

namely:                 7^3 + 2^5 = 3*5^3
```

Comment Two: the 'c' of the `'abc-conjecture'` is the `'a+b'` in the above.

Comment Three: It is a wonderful theorem of David Masser's that IF the '1'in line six of the above programme is replaced by ANY small positive quantity, THEN there will always be output PROVIDED one chooses 'n' to be SUFFICIENTLY LARGE.

(If I had had '.1' in place of '1' when I executed `'small_Masser(500)'`, then there would have been no output.)

Comment Four: The 'abc-conjecture' is ... (the margin is too small to contain it). By the way, one of the very many consequences of the conjecture is Fermat's famous 'Last Theorem' (and also K.F.Roth's legendary theorem on rational approximations to algebraic numbers - for which Roth won the Fields medal).

If any of you are able to suggest ways of improving the speed of the above procedure, I would appreciate hearing from you.

Also, I would appreciate hearing from those of you who have fast computing facilities about the following: HOW BIG did you have to make 'n' before you got output with the '1' in line six replaced by .1, or, better still, replaced by .001 (you probably won't get an 'n', though - by Masser's Theorem - there has to be such an 'n'), or whatever limit you are able to push it to.

6.3.2 Robert Israel

This can be speeded up quite a bit, I think. Note that the test will fail if a+b is squarefree. Since most integers are squarefree, you could do the outer loop over a+b instead of a, checking first for squarefreeness and at the same time getting the factorset of a+b. Instead of factoring a*b*(a+b), take the union of the factors of a, b and a+b. This way, no actual factoring ends up being done in the inner loop, because factorset remembers its previous results. Then instead of dividing by a+b and checking if the resulting fraction is less than 1, avoid division and stay with integers by checking if the product of the factors is less than a+b. Here, then, is my version.

```
with(numtheory):
```

```
new_Masser:=proc(n)
local a, b, s, f;
for s from 3 to 2*n-1 do
   f:= factorset(s);
   if convert(f, `*`) < s then
     for a from floor(s/2)+1 to min(s-1,n) do
        b:= s-a;
        if igcd(a,b) = 1 and
           convert(`union`(f, factorset(a), factorset(b)),`*`) < s
        then print(a, b, convert(`union`(f, factorset(a), factorset(b)),`*`)/s)
        fi
     od
   fi
od
end:
```

`new_Masser(500)` in Release 4 ran for me in 42.208 CPU seconds (admittedly, on a faster machine than a 486).

6.3.3 Andrei Broder

On my machine, the code below for $n = 500$, runs about 12 times faster than the original code. I guess that further optimizations are possible. In particular, on the math side, can b really get as large as a-1?

If you plan to show the students your code for various problems, may I suggest to consult a colleague in Computer Science that specializes in algorithms or algorithms engineering? They might help you write faster code and in the process teach the students some "tricks of the trade", such as pre-computation of values, cheap weak tests to avoid expensive exact tests, etc.

```
small_Masser_1:=proc(n)
local a, b, i, fset, sqf, t;
fset := array(1..2*n);
for i to 2*n do fset[i] := factorset(i) od;  #  Precompute factorsets
sqf := map(x -> convert(x,`*`), fset); # Precompute square free values
    for a to n do
        for b to (a-1) do
            if igcd(a,b) = 1 and sqf[a]*sqf[b]/(a+b) < 1 then # Second tests avoids e
              t := convert(fset[a] union fset[b] union fset[a+b], `*`)/(a+b);
```

```
            if t < 1 then print(a, b, t) fi;
          fi
        od
    od
end;
```

6.3.4 Andrei Broder

Dr. Israel's code (`new_Masser`) is about 10% faster than mine (`small_Masser_1`). Fortunately our ideas can be combined as in `comb_Masser` (all codes below) which runs more than twice as fast.

BTW, changing the test `sqf[a]*sqf[b]/(a+b) < 1` to `sqf[a]*sqf[b] < (a+b)` in `small_Masser_1 (= small_Masser_2)` reduced its running time by about 10%.

I never realized that integer comparison is so much cheaper than rationals comparison.

```
Approximate times (n=500) on my machine under Maple V.3:
comb_Masser ~ 8.5
small_Masser_2 ~ 18.1
new_Masser ~ 18.6
small_Masser_1 ~ 20.5
small_Masser (original code) ~ 256.7
```

```
comb_Masser :=
proc(n)
local a,b,s,i,fset,sqf;
    fset := array(1 .. 2*n);
    for i to 2*n do  fset[i] := factorset(i) od;
    sqf := map(x -> convert(x,`*`),fset);
    for s from 3 to 2*n-1 do
        if sqf[s] < s then
            for a from floor(1/2*s)+1 to min(s-1,n) do
                b := s-a;
                if igcd(a,b) = 1 and sqf[a]*sqf[b] < s and
                    convert(
                    (fset[a] union fset[b]) union fset[s],`*`)
                     < s then
                    print(a,b,convert(
```

```
                        (fset[a] union fset[b]) union fset[s],`*`
                        )/s)
                fi
            od
        fi
    od
end;
```

```
small_Masser_2 :=
proc(n)
local a,b,i,fset,sqf,t;
    fset := array(1 .. 2*n);
    for i to 2*n do  fset[i] := factorset(i) od;
    sqf := map(x -> convert(x,`*`),fset);
    for a to n do
        for b to a-1 do
            if igcd(a,b) = 1 and sqf[a]*sqf[b] < a+b then
                t := convert(
                    (fset[a] union fset[b]) union fset[a+b],`*`)
                    /(a+b);
                if t < 1 then print(a,b,t) fi
            fi
        od
    od
end;
```

```
new_Masser :=
proc(n)
local a,b,s,f;
    for s from 3 to 2*n-1 do
        f := factorset(s);
        if convert(f,`*`) < s then
            for a from floor(1/2*s)+1 to min(s-1,n) do
                b := s-a;
                if igcd(a,b) = 1 and convert(
                    `union`(f,factorset(a),factorset(b)),`*`) <
                    s then
                    print(a,b,convert(
```

```
                        `union`(f,factorset(a),factorset(b)),`*`
                        )/s)
                fi
            od
        fi
    od
end;
```

```
small_Masser_1:=proc(n)
local a, b, i, fset, sqf, t;
fset := array(1..2*n);
for i to 2*n do fset[i] := factorset(i) od;  #  Precompute factorsets
sqf := map(x -> convert(x,`*`), fset); # Precompute square free values
    for a to n do
        for b to (a-1) do
            if igcd(a,b) = 1 and sqf[a]*sqf[b]/(a+b) < 1 then # Second tests avoids e
              t := convert(fset[a] union fset[b] union fset[a+b], `*`)/(a+b);
              if t < 1 then print(a, b, t) fi;
           fi
        od
    od
end;
```

```
small_Masser := proc(n)
local a,b;
    for a to n do
        for b to a-1 do
            if igcd(a,b) = 1 and
                convert(factorset(a*b*(a+b)),`*`)/(a+b) < 1 then
                print(
                    a,b,convert(factorset(a*b*(a+b)),`*`)/(a+b))
            fi
        od
    od
end;
```

6.3.5 Olaf Becken

> Also, I would appreciate hearing from those of you who have fast computing ...

I found this thread very interesting, especially, how the "collective intellect" works. Now I wish to add my own 5-cent-ideas (sp?).

From `sqf[a]*sqf[b]*sqf[s] < s, a+b=s, a>b` and `igcd(a,b) = 1` one can prove that (in order of importance)

- `sqf[a] < a  ( from sqf[a]*1*2 <= sqf[a]*sqf[b]*sqf[s] < s <= 2*a )`
- `2*sqf[s] < s` - `igcd(sqf[a],sqf[s]) = 1` and `igcd(sqf[b],sqf[s]) = 1`

Using these conditions one can speed up the procedure. I tried

```
Masser_6th_version :=proc(n)
local sqf,a,b,s;
   sqf := array(1 .. 2*n-1,[1,2]);
   for s from 3 to 2*n-1 do
       sqf[s] := convert(factorset(s),`*`);
       if 2*% < s then
          for a from floor(s/2)+1 to min(s-1,n) do
             if sqf[a] < a and sqf[a]*sqf[s] < s then
                 b := s-a;
                 if igcd(a,b) = 1 and sqf[a]*sqf[b]*sqf[s] < s then
                        print(a,b,sqf[a]*sqf[b]*sqf[s]/s)
                 fi
             fi
          od
       fi
   od
end;
```

which needs (with $n = 500$) 9.2 seconds on my computer whereas `comb_Masser` needs 15.7 seconds.

BTW, the lowest number I got for `sqf[a]*sqf[b]*sqf[s]/s` is `21/1024 ~ 0.0205` with `3^10+7^3=29*2^11`.

6.3.6 Dave Reynolds

Okay, I've had a chance to analyze and I see what's happening. Let $sqf(x)$ represent the product of the factorset of x.

Hence we are looking for small values of

```
   sqf(a)*sqf(b)*sqf(a+b)/(a+b)


                    n
let b = 1, let a = 3  - 1


                  n                       n           n
let f(n)  = sqf(3  - 1) * sqf(1) * sqf(3  - 1 + 1)/(3  - 1 + 1)


                  n                 n
          = sqf(3  - 1) * 1 * 3 / 3


                  n         (n - 1)
          = sqf(3  - 1) / 3
```

Now we analyze f(2n) from above.

```
                  2n        (2n - 1)
    f(2n) = sqf(3  - 1) / 3


        2n         n       n
since  3  - 1  = (3  - 1)(3  + 1)


                   n       n          (2n - 1)
    f(2n) = sqf((3  - 1)(3  + 1)) / 3
       n         n
Now,  3 - 1 and 3 + 1 may share common factors.
```

Since both expressions are even, we know they at least share 2 as a common factor.

Hence

```
          n       n                 n              n
    sqf((3  - 1)(3  + 1))  <=  sqf(3  - 1) * sqf((3  + 1) / 2)
```

Therefore

```
                   n         (n - 1)              n              n
```

```
f(2n) <= sqf(3  - 1) / 3          *   sqf(((3  + 1) / 2) / 3

                           n              n
f(2n) <= f(n)  *  sqf(((3  + 1) / 2) / 3
```

Since

```
      n                                n
sqf((3 + 1) / 2) is at most  1/2 *  3 + 1,

      n
     3 + 1
and  -----  is very close to 1 for large n,
       n
      3
```

we are basically saying that $f(2n)$ is less than or equal to roughly `1/2 * f(n)`. We can always get lucky and obtain even more common factors and make a leap, but halving each time we double n is good enough.

We have a way of calculating arbitrarily small values for $f(n)$. This demonstrates:

```
                    n, f(n), evalf(f(n))


                        22
                   5, ----, .2716049383
                        81

                       1342
                  10, -----, .06818066352
                      19683

                      7924510
                20, ----------, .006818181816
                    1162261467

                 13815528930746510
            40, ------------------, .003409090909
                4052555153018976267
```

$$80,\ \frac{8398228943996927461141091488999051 0}{4926960980478197443869440340212776586 7},\ .001704545455$$

6.3.7 Dave Reynolds

I started following this thread very late, so I don't know if these comments apply.

Factoring is very time consuming. A simple "sieve" technique can generate the sqf array much more efficiently. This is a times savings for moderate 'n'.

However, for fairly large 'n', the `O(n^2)` behavior of the secondary loop overshadows all else.

Compare with `Masser_6th_version`:

```
daver1 := proc(n)
local sqf,a,b,s,m;
   m := 2*n-1;
   sqf := array(1 .. m);
   for s to m do
       sqf[s] := 1;
   od;
   for s from 2 to m do
       if sqf[s] = 1 then
           a := s;
           while a <= m do
               sqf[a] := sqf[a] * s;
               a := a + s;
           od
       fi
   od;
   for s from 3 to 2*n-1 do
       if 2*sqf[s] < s then
          for a from floor(s/2)+1 to min(s-1,n) do
             if sqf[a] < a and sqf[a]*sqf[s] < s then
                 b := s-a;
                 if igcd(a,b) = 1 and sqf[a]*sqf[b]*sqf[s] < s then
                        print(a,b,sqf[a]*sqf[b]*sqf[s]/s)
```

```
                fi
            fi
        od
    fi
  od
end;
```

It is also easier to convert to C code.

```
#include <stdio.h>

unsigned long gcd(m, n)
unsigned long m, n;
{
    unsigned long r;

    do {
    r = m % n;
    m = n;
    n = r;
    } while (r != 0);

    return(m);
}

#define n 5160
#define m (2 * (n) - 1)

unsigned long sqf[m + 1];  /* C arrays are zero based; ignore 1st element */

nt main()
{
    unsigned long a, b, s, t, u, v;
    double best, d;

    for (s = 1; s <= m; s++)
    sqf[s] = 1;

    for (s = 2; s <= m; s++)
    if (sqf[s] == 1)
        for (a = s; a <= m; a += s)
```

```
        sqf[a] *= s;

    best = 1.0;
    for (s = 3; s <= m; s++) {
    if ((2 * sqf[s]) < s) {
        t = s - 1;
        if (t > n)
        t = n;
        for (a = s / 2 + 1;  a <= t; a++) {
            if ((sqf[a] < a) && ((u = sqf[a] * sqf[s]) < s)) {
                b = s - a;
                if ((gcd(a, b) == 1) && ((v = (u * sqf[b])) < s)) {
                d = (double) v / (double) s;
                if (d < best) {
                    printf("%ld, %ld, %ld/%ld = %f\n", a, b, v, s, d);
                    best = d;
                   }
                }
           }
        }
      }
    }
}
```

Output of above program:

```
8, 1, 6/9 = 0.666667
63, 1, 42/64 = 0.656250
49, 32, 42/81 = 0.518519
80, 1, 30/81 = 0.370370
125, 3, 30/128 = 0.234375
512, 1, 114/513 = 0.222222
1024, 5, 210/1029 = 0.204082
2187, 10, 390/2197 = 0.177515
2400, 1, 210/2401 = 0.087464
4374, 1, 210/4375 = 0.048000
32768, 37, 1110/32805 = 0.033836
59049, 343, 1218/59392 = 0.020508
255879, 121, 4290/256000 = 0.016758
(The next line is invalid. The result of 32 bit overflow!) 234375, 165941, 2546/400316
```

Anyone with 64 bit integers wish to take it further?

Instead of exhaustive search, I decided to quickly check special cases.

```
            x        y
       if  a  + b = c   and gcd(a, b) = 1 then

                                    (y-1)
          the result is   a*sqf(b)/c
```

If y is sufficiently large, this could be very small.

I decided to simplify this further to:

```
                    x                    x
        t + 1 = base     Hence t = (base  - 1)

                                (x-1)
   This results in  sqf(t)/base
```

Here is some simple Maple code to play with.

```
with(numtheory):
sqf:=x->convert(factorset(x),`*`);
try:=proc(base,max_power)
  local a,s,t,best;
  best:=1:
  s:=base:
  for i from 2 to max_power do
    olds:=s:
    s:=s*base;
    a:=s-1;
    t:=sqf(a)/olds:
    if (t < best) then
      print(i, t, evalf(t)):
      best:=t:
    fi
  od
end;
```

```
> try(3,50);

                              2, 2/3, .6666666667

                                  10
                              4, ----, .3703703704
                                  27

                                  22
                              5, ----, .2716049383
                                  81

                                  410
                              8, ----, .1874714220
                                 2187

                                  1342
                            10, -----, .06818066352
                                19683

                                 7924510
                         20, ----------, .006818181816
                             1162261467

                          13815528930746510
                    40, ------------------, .003409090909
                        4052555153018976267

bytes used=1000096, alloc=655240, time=1.17
```

```
> ifactor(3^10-1);

                                   3     2
                                 (2)  (11)  (61)

> ifactor(3^20-1);

                               4    2     2
```

```
                          (2)  (5)  (11)  (61) (1181)

> ifactor(3^40-1);

                      5    2     2
                 (2)  (5)  (11)  (41) (61) (1181) (42521761)

#  Hmmmm... this is peculiar!

> ifactor(3^80-1);

    6    2     2
 (2)  (5)  (11)  (17) (41) (61) (193) (1181) (128653413121) (42521761) (14401)


#  Is this a coincidence?

> sqf(3^80-1)/3^79;

                      839822894399692746114109148899905l0
                     ------------------------------------
                     492696098047819744386944034021277658 67

> evalf(%);

                                  .001704545455
```

6.4 about maple functions dll and dde with maple (6.1.99)

6.4.1 LEE C.Y.

I am trying to use Maple as a 'helper' program to perform some calculations for another main program. Is dynamic data exchange (DDE) possible with Maple? On the other hand, are there any DLL files with Maple functions available?

If the above are possible, it would be very useful for me to know where information on the commands/parameters/functions...etc for the DDE or in the DLLs could be found.

6.4.2 Michael McCabe (15.1.99)

What you want is Mathedge, i.e. the Maple kernel. A couple of years ago I pioneered its use in DLL mode by linking it to interactive multimedia material developed using Asymetrix Toolbook. Others have used the more cumbersome DDE approach. Although I developed some extremely exciting and innovative courseware for vector calculus within the UK Mathematics Courseware Consortium, it remains still to be released since agreement over Mathedge licencing has still not been agreed.

I used Mathedge corresponding to Maple release 3 - others may be able to advise on whether there is a later version.

6.4.3 Claus Gotfred Rasmussen (15.1.99)

I'm also using Maple to perform some calculations for a another main-program (i.e. SAS), so I've also tried to use DDE in Maple - without success however. Instead I was able to make my main-program export the relevant data in Maple-format. For instance it exports a simple txt-file containing the following lines:

```
a:=2.4;
b:=3.7;
```

Then I've written Maple-code that reads (`read ‘c:/.../filename.txt‘;`) these txt-files and executes the contents. Then there are some Maple-calculations and in the end of my Maple-file I've exported (`writedata(‘c:/.../filename.txt‘, variable);`) the results to a txt-file, which the main-program can read.

Be sure to notice the syntax of the read and write statements, especially the characters `‘` and `/`.

Please feel free to email me, if you have any comments or questions.

6.5 abstract algebra impossible in maple? (20.12.02)

6.5.1 Charles James Leonardo Quarra Cappiello

The dot product on Maple ('.' operator) has no effect upon undefined objects, even if the effect of such operation may depend only on the conmutative properties.

a desired capability would be the following:

```
>(A + t*B).(A + t*B);

           (A + tB)(A + tB)

>expand(%);

                          2              2  2
                         A  + 2 A t B + t  B

>collect(%,t) assuming commute(t,[A,B]); (t commutes with A and B)

                          2                 2  2
                         A  + t(BA + AB) + t (B )
```

currently this doesnt seem to be possible

Other thing that 'should' be easily computable but seem to not be implemented yet is bra-ket formalism

```
ie:

 > <psi|:=a*<e1| + b*<e2| + c*<e3|:            //(a basis)

 > expand( <psi|rho> );

          a*<e1|rho> + b*<e2|rho> + c*<e3|rho>

 > Projector:= |psi><psi|:

 > Projector.|psi>;
```

```
        -            -            -            -             -             -
       ( a*|e1> + b*|e2> + c*|e3> )*(aa<e1|e1> + ab<e1|e2> + ac<e1|e3> +

-            -            -            -            -            -
ba<e2|e1> + bb<e2|e2> + bc<e2|e3> + ca<e3|e1> + cb<e3|e2> + cc<e3|e3>)

 >% assuming(real,a,b,c), additionally( orthonormal, |e1> , |e2> , |e3> );

                                          2    2    2
     ( a*|e1> + b*|e2> + c*|e3> )*(a   + b   + c )
```

But again, this doesnt seem to be possible, and im not aware what language resources are available to Maple to implement it. Any hints about this?

6.5.2 Robert Israel (20.12.02)

| a desired capability would be the following: ...

Huh? This is only valid if A commutes with t*B and t commutes with B. You should get `A.A + A.(t*B) + (t*B).A + (t*B).(t*B)`.

It is true that expand does not do anything with ".". Here is at least a partial remedy:

```
`expand/.` := proc ()
  local i, n, A;
  n:= nargs;
  A := op(map(expand,[args]));
  for i to n do
    if type(A[i],`+`) then
      return map(t -> `expand/.`(A[1 .. i-1],t,A[i+1 .. n]),A[i])
    end if
  end do;
  `.`(A)
end proc;
```

| `>collect(%,t)` assuming `commute(t,[A,B]); (t commutes with A and B)`

It would be "commute".

6.5.3 Dr Francis J. Wright (2.1.03)

> The dot product on Maple ('.' operator) has no effect ...

Dot is specifically defined to be non-commutative unless one of its operands is explicitly numerical. However, one way to make . commutative in Maple 8 is as follows (which should work in all versions since Maple 6):

```
> use `.`=`*` in
>    (A + t*B).(A + t*B)
> end use;

                                        2
                                (A + t B)
```

> Other thing that 'should' be easily computable ...

Since Maple 6, the read syntax <. . . | . . .> is used to represent a row vector, so it could be used for bra-ket notation only if there were some mechanism for redefining read syntax and I'm not aware of any such mechanism.

Customization of the read and print facilities is an area in which Maple is weak compared with other CASs (such as REDUCE). If you are prepared to use a different read syntax, such as bra and ket as functions, then it should not be too hard to implement the formalism. You could overload . in a module to implement the various non-commutative products involved.

6.6 access maple via tcl (29.5.00)

6.6.1 Rafal Ablamowicz

I am looking for a tutorial or an example how to use Tcl (for example, Tcl /Tk 8.3.1) for Windows in order to provide a web access to my own procedures written in Maple, e.g., to my package CLIFFORD, and then output results back to the web page.

At this point, however, as an example, I would like to write a web page with an appropriate Tcl code that would use my procedure which takes as input two positive integers n1 and n2 and outputs a list of length n1 of positive integers whose sum equals n2. I have written a procedure in Maple V 5.1 that does that; now I would like to make it possible to run it from the web.

6.6.2 Dr Francis J. Wright (30.5.00)

Does it have to be Tcl? I could give you access to some perl code that I use. Very briefly, the basic technique that I use, and it would probably need to be similar whatever the CGI processor, is this. Take the form input and write it to a file in Maple syntax. Include code to load any Maple libraries etc. Start a command-line version of Maple, using the "system" function or equivalent, that reads the input file and writes to an output file.

(You could use pipes instead of intermediate files, but that is harder to debug and less portable.) Generate appropriate http headers etc, convert the Maple output to whatever format you want to send the user (HTML, MathML, TeX, ...) and output it to the standard output of the CGI program.

There are several perl facilities for generating dynamic HTML, among which I currently use the CGI.pm module. I expect that there are Tcl analogues.

6.6.3 Claus Gotfred Rasmussen (31.5.00)

I've no knowledge of Tcl, but I've had a similar problem with interactions between SAS and Maple. I used the command line interface of Maple, because I could execute dos commands from within SAS.

I used simple .txt-files to transfer data from SAS to Maple, which can be read in to Maple via the READ statement. I made an .txt-file containg all Maple statements, and executed a dos-command like `c:\maplev\bin.win\cmaple.exe c:\maplefile.txt` from within SAS. In the end of the maple-program, i had some WRITEDATA statements, that put the results into txt-files, and I could then finally read these from SAS.

I know it's a long way to go, but I found no better solutions.

6.6.4 Theodore Kolokolnikov (2.6.00)

I have a related question: I've seen how to use perl to fork processes under unix so that perl can load & execute maple code when a web surfer requests. I'm curious however, whether it's possible with perl to fork a process just once, and then use the same process everytime a webber surfs in?

6.6.5 Dr Francis J. Wright (9.6.00)

I think the answer is yes, but I haven't yet found the need to worry about this so I have no actual experience of it. I believe that apache provides a facility for locking a perl and/or any CGI program in memory. Also, I suggest you take a look at `http://www.ActiveState.com/`

6.7 access to maple within matlab (30.9.99)

6.7.1 Mohamad Saad

Is it possible to access and execute a MAPLE program within MATLAB. I already have some symbolic programs written in MAPLE to model some mechanical systems. I use the numerical results to simulate these systems in Matlab.

I would like to execute the Maple program within Matlab, i.e from a MATLAB session and than to collect the data and use it in Matlab. I don't want to rewrite these programs using Matlab symbolic toolbox.

Is there any way to do this directly within Matlab ?

6.7.2 Nestor E. Sanchez (5.10.99)

I had that problem last year. I had derived a set of ODE in Maple, but the set was too large to be solved using Maple. My equations were thousands of lines. I wanted to use Matlab to solve the system. I got in touch with them. They told me that it was simple. However, it never work.

I wrote them in the correct format but the restrictions of Matlab to read the equations were just impossible. You had to end the lines that continue with ... and then you can only read so many like that. In other words I just gave up.

There is not a direct way. You have to make a file with Maple and read it in Matlab with all the constraints. If you are able to do it, please let me know.

If the number of equations is low it must be OK.

6.7.3 Claus Gotfred Rasmussen (11.10.99)

You can use the command line version of maple (cmaple.exe) which can be run from a dos-prompt - that is if it is possible to execute a call to a dos-prompt from within matlab.

I've tried it from within SAS, and it was possible, but not easy.

Feel free to write me for more details if needed.

6.8 ace 3.0, new environment (19.6.98)

6.8.1 Sebastien Veigneau

I am very pleased to inform you that ACE 3.0 is now available.

ACE is an Algebraic Combinatorics Environment for the computer algebra system MAPLE. As each version of ACE, the new version 3.0 has been released on the 2nd of June. This third version has been totally reorganized in order to obtain an easy-to-use and efficient tool for computations in algebraic combinatorics.

It includes 18 packages (+4 packages of tools) to handle some combinatorial structures such as partitions, compositions, permutations, codes, symmetric functions, Schubert polynomials, non commutative symmetric functions, words, tableaux, polynomials in several variables, etc.:

```
ACE : Algebraic Combinatorics Environment

SFA : Symmetric Functions on different Alphabets
FM  : Free Module over symmetric polynomials
NCSF: NonCommutative Symmetric  Functions
CLG : Characters of Linear Groups
SP  : Schubert Polynomials
SYMF: SYMmetric Functions
TAB : TABleaux
FREE: FREE algebra
BNA : hyperoctahedral Algebras
SGA : Symmetric Group Algebra
NCA : NilCoxeter Algebra
IDCA: IDCoxeter Algebra
HEKA: HEcKe Algebra

BN  : hyperoctahedral groups
SG  : Symmetric Groups
CG  : Classical Groups

PART: PARTitions
COMP: COMPositions

ERR : ERRors
```

```
TEX : TEX output
TYP : TYPe checking
```

which cluster about 400 functions.

It can either be obtained via the HTML page of ACE 3.0 that corresponds to the URL:

`http://phalanstere.univ-mlv.fr/~ace`

or directly by anonymous ftp on:

Name: `weyl.univ-mlv.fr` Address: `192.134.103.58`

To connect to the remote weyl computer, run the UNIX command:

ftp `weyl.univ-mlv.fr` (or ftp `192.134.103.58`)

It will contact the weyl machine and ask you for a login name and a password. Give anonymous as your login name and then your electronic mail address as the password. You are then allowed to get any expressly permitted file from the server.

Maple programs are located in the `/pub/lang/maple` directory, among which there is the ACE directory containing the distribution of ACE. In particular, to get the 3.0 release of ACE:

`cd /pub/lang/maple/ACE/3.0`

ACE 3.0 is available for any UNIX, MS-DOS or MACINTOSH system, for any release 1, 2, 3 and 4 of Maple V. It provides also some pre-installed versions of ACE 3.0 which give you a way to use ACE 3.0 without running the installation procedure. This easy installation is explained in:

`http://phalanstere.univ-mlv.fr/~ace/ACE/3.0/install.html`

The ACE system will also be available soon on a Cd-Rom:

`http://phalanstere.univ-mlv.fr/~ace/ACE/3.0/cdrom.html`

The Cd-Rom includes all sources of ACE 3.0, both as tar files and also as extracted archives. The tar files give you a way to get ACE sources and move them to your system in a efficient manner, while extracted archives give you the possibility to install ACE on your system without copying sources of ACE on your disk. It can be for instance useful if you don't have as much free space on your disk as needed (about 4Mb) to copy source files.

The ACE Cd-Rom also includes pre-installed versions of ACE 3.0. This allows you to install ACE on your system by simply copying the "compiled" Maple files on your disk.

These files are available as tar archive files, but also as extracted archive files. This has the advantage to give you a way to test or use ACE without installing it physically on your system.

The Cd-Rom also includes some related papers and contains the Html, postscript and DVI documentations.

Some data have been stored on the Cd-Rom. For instance, there are TeX files and Maple files that present some tables, of polynomials for example, that have been computed with ACE during many hours and even many days long.

It also contains all sources of previous versions of ACE, 1.0 and 2.0, together with their corresponding documentations.

If you decide to install ACE on your system, please send me an email for my own statistics...and thank you for sending this email to those who can be interested in using ACE.

If you have any problem or question, don't hesitate to send an email. Thank you for your interest in ACE, I hope this system can be useful for you and others...

6.9 adding null character to a name (24.4.01)

6.9.1 Greg Nash

Adding a null character to a name, shouldn't affect it's interpretation:

```
x := 'x';
evalb(x=x||``);
```

```
x := x
 true
```

However, when doing the same thing in a procedure, this same behavior doesn't seem to apply:

```
test := proc()
local x;
print(evalb(eval(x)=eval(x||``)));
end:
test();
```

```
false
```

What am I missing?

6.9.2 Helmut Kahovec (26.4.01)

Well, you are missing the following (cf. online help page of cat):

"Note that if the result of cat is a name, then it is in fact a global name, even though there might be a local variable of the same name currently active."

and

"This allows you to make global assignments from within procedures. It illustrates the fact that only global names are returned by cat."

(the latter being the comment on the last example of that online help page). Thus we have in turn:

```
x:='x':
```

```
test1:=proc()
local x;
  print(evalb(x=cat(x,``)));
  print(x-cat(x,``))
end:

test1();
```

```
                              false
                              x - x
```

```
test2:=proc()
global x;
  print(evalb(x=cat(x,``)));
  print(x-cat(x,``))
end:

test2();
```

```
                               true
                                0
```

6.9.3 Robert Israel (29.4.01)

Actually you're not adding anything to the name. But you can't construct a local variable inside a procedure. The constructed name refers to the global variable x, not the local variable of the procedure. See my Maple Advisor Database page `?Constructing_names_of_local_variables_and_formal_parameters` (`http://www.math.ubc.ca/~israel/advisor`):

The reason for this behaviour is connected to the way the Maple kernel processes procedure definitions. Internally, names are used for reference to global variables, but not local variables or formal parameters. For example:

```
> test3:= proc(x) local v; v:= x+g end:
> pointto(disassemble(addressof(eval(test3)))[6]);

                        LOC[1] := args[1] + g
```

Thus in the internal version of the procedure definition, the local variable name v is replaced by `LOC[1]`, and the formal parameter name x is replaced by `args[1]` (more precisely, these are special constructions which Maple interprets as references to the first local variable and the first formal parameter). If a name is only produced when the procedure is executed, these replacements do not occur and the name is interpreted as a global variable.

6.10 advisor database for maple 6 (24.5.00)

6.10.1 Robert Israel

I have now made a new release of my Maple Advisor Database available at `http://www.math.ubc.ca/~israel/advisor`.

The new release is so far only for Maple 6: the Release 5 and Release 4 databases are unchanged so far (although some additions to them may be coming in the next few weeks).

Notable additions to this release include the following functions:

```
    csum - check convergence of a series
    expands - symbolic version of expand
    greatestroot and leastroot - approximate greatest or least
    root of an equation or expression on a real interval.
    PVInt - numerical principal value integral
    quickplot and quickplot3d - fast 2-d and 3-d curve or
    point plotting of a list or hfarray
```

I appreciate hearing any comments on this project.

You can access the Maple Advisor Database on-line, or download and install it on your own computer. The database consists of a library of Maple procedures, plus a database of Maple help pages in the following categories:

```
- advice on how to use and program Maple, containing answers to
  many of the common questions that users, especially students and
  other new users, have about Maple.
- explanations of common error messages
- help pages for the procedures in the library
- work-arounds and fixes for bugs in Maple.
```

6.10.2 Tom Casselman (1.6.00)

I have downloaded Robert Israel's Maple advisor database for both M5.1 and M6. How do I incorporate Bob's databases into my Maple 5 and Maple 6 programs.

Bob was not able to help since he is not familiar with Macintosh systems. I am running using OS 9.0.4.

6.10.3 Wolfgang Ziller (2.6.00)

I tried to install Robert Israel's advisor database on my laptop which runs Windows98SE both in Maple V5.1 and V6 with only partial success.

I downloaded the database in a separate advisor directory.

The only way I can get it to work, if I issue

```
libname:=`C:\\Program Files\\Maple 6/advisor`,libname;
in a running Maple session ( / instead of // does not work at all)
```

If I put the same command in the maple6.ini file, Maple does not recognize the change (libname gives the old libname only).

Did anyone get this to work in Windows 98 ?

6.10.4 Robert Israel (5.6.00)

Don't put it in maple6.ini. That is a Windows initialization file, not the Maple initialization file. You want to put it in a file named "maple.ini". You probably don't have one yet if you haven't made it yourself.

It should normally go into the directory that is current when you start Maple, or else in Maple's "bin.wnt" directory. You could produce it with any text editor, e.g. Notepad or Wordpad. If you wish, you can produce the file in Maple itself, as follows (this will put the file in the current directory, appending to any existing maple.ini file there):

```
fopen("maple.ini",APPEND):
fprintf("maple.ini",
  "libname:= `c:/Program Files/Maple 6/advisor`,libname:\n"):
fclose("maple.ini"):
```

Note that the change won't take effect until you exit this Maple session.

6.10.5 Carl Eberhart (5.6.00)

Yes. Following the instructions given in the link `http://www.math.ubc.ca/~israel/advisor/`

Make a directory `c:/Program Files/Maple 6/advisor` and download the maple 6 database into it and unzip the database. then in the maple 6 bin.wnt directory create a file maple.ini containing the line

```
libname := "c:\\Program Files\\Maple 6/advisor",libname:
```

When you fire up maple6 again, the advisor database is available via help. It works very nicely.

6.11 afactors (17.3.99)

6.11.1 Werner Heiss

The command

```
evala(AFactors(-t^7-t^3-2*t^6-2*t^2+2))
```

didn't terminate after several hours. Is it normal that it takes so long or might it be a bug?

6.11.2 Michael Monagan (18.3.99)

Yes, you are asking it to compute the splitting field which may be degree `7! = 5040` – it will have to factor a polynomial of degree 5040 over Z.

The computation blows up. Even degree 5 is non trivial in the general case.

6.11.3 Nils Bruin (19.3.99)

This polynomial is the product of a linear factor and an irreducible degree 6 factor. The degree 6 factor has galois group S6, so you're asking maple to construct a degree `6!` extension (as a degree 2 ext. of a degree 3ext etc.) This is a huge task, so I'd consider it normal that this takes so long. I can imagine that the answer wouldn't even fit in your computer.

6.11.4 Helmut Kahovec (19.3.99)

IMO there is not a bug here: your polynomial is just too complicated for being decomposed into linear factors by Maple. Look at the following (simple) example:

```
> restart;
> alias(alpha=RootOf(Z^6+Z^5+Z^4+Z^3+Z^2+Z+1));

> expr1:=t^7-1;

> evala(AFactor(expr1));

                                  4
  (t - 1) (t - alpha) (t - alpha )
```

```
                          2        3        4        5
      (t + 1 + alpha + alpha  + alpha  + alpha  + alpha )

                 2            3            5
      (t - alpha ) (t - alpha ) (t - alpha )
```

Anyway, your polynomial may in fact be partially decomposed into linear factors:

```
> -t^7-2*t^6-t^3-2*t^2+2;

                       7      6    3      2
                     -t  - 2 t  - t  - 2 t  + 2

> factor(%);

                           6    5    4    3
                -(1 + t) (t  + t  - t  + t  + 2 t - 2)

> expr2:=remove(type,%,{linear,constant});

                           6    5    4    3
                 expr2 := t  + t  - t  + t  + 2 t - 2

> alias(beta=RootOf(Z^6+Z^5-Z^4+Z^3+2*Z-2));

> map(collect,factor(expr2,beta),[t]);

      4           2       3       5
  (beta  + 2 + beta  - beta  + beta

                4              2       3
        + (beta  + beta - beta  + beta ) t

                3                   2   2               4
        + (beta  + 1 - beta + beta ) t  + (1 + beta) t

                           2   3    5
        + (beta - 1 + beta ) t  + t ) (t - beta)
```

Apparently, the nonlinear part of this is very complicated. Maple might not be able to decompose it any further.

6.12 aim: automatic assessment of mathematics using web/maple (30.11.00)

6.12.1 Theodore Kolokolnikov

We present AIM (Alice Interactive Mathematics), a web based system for automatically administering graded tests and homeworks with mathematical content.

The two distinguishing features of AIM are its use of Maple for performing computations and visualization of mathematical formulas; and its facilities for giving partial credit and feedback.

AIM also offers randomisation of quizzes and questions; versatility in question and quiz design; extensive grade reporting and monitoring capabilities; ability to collect surveys; web interface for both teacher and student. AIM can be used to administer graded tests, homeworks or ungraded self-assessment exercises.

AIM has been tested extensively during the last year at several Belgian and UK universities. A complete list of courses and sites using AIM is available at AIM homepage.

AIM is available free of charge. It can be installed on a Windows or Unix computer that has Maple and an internet connection. For more information and to try it, go to AIM homepage:

`http://allserv.rug.ac.be/~nvdbergh/aim/docs`

6.13 airy's equation (7.12.00)

6.13.1 Chuck Baker

Is it possible to develop a power series solution, in powers of x, for Airy's equations using Maple?

$$y'' - xy = 0$$

6.13.2 Chris Eilbeck (8.12.00)

Use the expansions given in Abramowitz and Stegun, Handbook of Mathematical Functions, Section 10.4.

6.13.3 Bill Bauldry (8.12.00)

A naive approach will work here to get one solution:

```
>  de := (D@@2)(y)(x) - x*y(x) =0;

                              (2)
                    de := (D   )(y)(x) - x y(x) = 0

>  Order := 10: # if you want more than 5 terms
>  dsolve(de, y(x), series);

                                          3                  4
   y(x) = y(0) + D(y)(0) x + 1/6 y(0) x  + 1/12 D(y)(0) x  +

                     6                   7                  9      10
         1/180 y(0) x  + 1/504 D(y)(0) x  + 1/12960 y(0) x  + O(x  )
```

But it may be more fun to work with the predefined functions:

```
>  Order := 6:
>  taylor(AiryAi(x), x);
>  taylor(AiryBi(x), x);

          (1/3)          (1/6)                          (1/3)
```

```
       3              3      GAMMA(2/3)            3         3
  1/3 ---------- - 1/2 ---------------- x + 1/18 ---------- x  -
      GAMMA(2/3)             Pi                  GAMMA(2/3)

             (1/6)
            3      GAMMA(2/3)  4      6
       1/24 ---------------- x  + O(x )
                     Pi

        (5/6)          (2/3)                        (5/6)
       3              3      GAMMA(2/3)            3         3
  1/3 ---------- + 1/2 ---------------- x + 1/18 ---------- x  +
      GAMMA(2/3)             Pi                  GAMMA(2/3)

             (2/3)
            3      GAMMA(2/3)  4      6
       1/24 ---------------- x  + O(x )
                     Pi
```

6.13.4 Dr Francis J. Wright (8.12.00)

From the online help for AiryAi in maple6:

```
series(AiryAi(x),x,4);

        1/3           1/6                          1/3
       3             3    GAMMA(2/3)              3        3      4
 1/3 ---------- - 1/2 --------------- x + 1/18 ---------- x  + O(x )
     GAMMA(2/3)             Pi                  GAMMA(2/3)
```

There are probably also harder ways to do it!

6.13.5 Craig B. Watkins (8.12.00)

It's quite easy. One example I have is:

```
Order:=20;
f1:=dsolve({diff(y(x),x)=exp(x*y),y(0)=0},y(x),'type=series');
f2:=subs(f1,y(x));
f3:=convert(f2,polynom);
plot(f3,x=-1..1);
g1:=dsolve({diff(y(x),x$2)+x*y=0,y(0)=0,D(y)(0)=1},y(x),'type=series');
g2:=subs(g1,y(x));
g3:=convert(g2,polynom);
plot(g3,x=-1..5,scaling=constrained);
```

This may be more than you wanted (and I use $y'' + xy = 0$ for this example), but it's what I like to show undergraduates.

An explanation of what this is used for (somewhat MIT-dependent in terms of subject matter is at

`http://web.mit.edu/18.03-esg/www/cws00/maple/seriessol.html`

6.13.6 Willard, Daniel Dr (8.12.00)

Try this (Release 5.1):

```
restart;#?dsolve[series]
with(ODEtools);
ode:=diff(y(x),x$2)-x*y(x)=0;
Order:=20;
dsolve(ode,y(x),'type=series');
 # or dsolve({ode, y(0)=A,D(y)(0)=B},y(x),'type=series');
```

6.13.7 Robert Israel (8.12.00)

There were lots of replies (including mine) that used the "series" option for dsolve, but Chuck later confirmed to me that he was interested in a symbolic expression for the coefficients. In Maple 6, this can be obtained using the new "formal_series" option:

```
dsolve((D@@2)(y)(x)-x*y(x)=0,y(x),'formal_series',coeffs=mhypergeom);

                      4
 {_C[1] x + 1/12 _C[1] x

                            /infinity                                  \
                            | -----              n  (3 n + 1)         |
                            |  \                                       |
                            |   )          (1/3)  x                    |
          2/9 _C[1] Pi sqrt(3) |   )     -------------------------------|
                            |  /        n                              |
                            | -----   3  GAMMA(n + 4/3) GAMMA(n + 1)|
                            \ n = 2                                    /
       + ------------------------------------------------------------------,
                                  GAMMA(2/3)

                       3
      _C[1] + 1/6 _C[1] x

                          /infinity                                  \
                          | -----                n  (3 n)           |
                          |  \                                       |
                          |   )            (1/3)  x                  |
       + _C[1] GAMMA(2/3) |   )     -------------------------------|}
                          |  /                    n                  |
                          | -----   GAMMA(n + 1) 3  GAMMA(n + 2/3)|
                          \ n = 2                                    /
```

6.13.8 Francois Boucher (8.12.00)

Two ideas (with mapleV4) :

```
> with(powseries):
> edo:=diff(y(x),x,x)-x*y(x);

                              / 2      \
                              |d       |
                        edo := |--- y(x)| - x y(x)
                              |  2     |
                              \dx      /

> sol:=powsolve({edo,y(0)=1,D(y)(0)=0});

                      sol := proc(powparm)  ...  end


 sol(_k);

                               a(_k - 3)
                              -----------
                              _k (_k - 1)

> ak:=rsolve({a(k)=a(k-1)/3/k/(3*k-1),a(0)=1},a(k));

                                (-k)
                               9     GAMMA(2/3)
                    ak := --------------------------
                          GAMMA(k + 1) GAMMA(k + 2/3)

> sum(ak*x^(3 *k),k=0..infinity);

                                                 3
                       hypergeom([], [2/3], 1/9 x )

> restart:
> with(share):
#See ?share and ?share,contents for information about the share library
```

```
> readshare(gfun,analysis):
> with(gfun):
> edo:=diff(y(x),x,x)-x*y(x);

                              / 2      \
                              |d       |
                        edo := |--- y(x)| - x y(x)
                              |  2     |
                              \dx      /

> rec:=diffeqtorec({edo,y(0)=1,D(y)(0)=0},y(x),u(m));

rec :=
                 2
    {-u(m) + (m  + 5 m + 6) u(m + 3), u(2) = 0, u(0) = 1, u(1) = 0}
rsolve don't work on this but :

> rec2:=op(1,rec);

                                  2
                rec2 := -u(m) + (m  + 5 m + 6) u(m + 3)

> rec2:=subs([u(m)=v(n),u(m+3)=v(n+1),m=3*n],rec2);

                                   2
               rec2 := -v(n) + (9 n  + 15 n + 6) v(n + 1)

> vn:=rsolve({rec2=0,v(0)=1},v(n));

                                (-n)
                               9     GAMMA(2/3)
                   vn := ---------------------------
                         GAMMA(n + 1) GAMMA(n + 2/3)

> sol:=sum(vn*x^(3*n),n=0..infinity);

                                                 3
                  sol := hypergeom([], [2/3], 1/9 x )
```

6.14 algebra (4.4.97)

6.14.1 Jean-Pierre Cherdieu

I would like to use Maple in a course of General Algebra for beginners, but I am sure that I am not the first. If you have some good example could you please ssend it to me ?

6.14.2 joyner.david (11.4.97)

There's an elementary example on my web page

`http://web.usna.navy.mil/~wdj/group.htm`

More maple stuff can be found on my web page

`http://web.usna.navy.mil/~wdj/mplstuff.htm`

6.15 algebraic manipulations with radicals (9.8.01)

6.15.1 Les Wright

I am a mathematical hobbyist puttering around with MVR4 (I know, dated software....) exploring some elementary number theory. In particular, I am trying to get a better grasp of the classic Dirichlet proofs for FLT with n = 5, which is basically high school algebra very ingeniously applied.

I am looking at numbers of the form $a + b\sqrt{5}$. On raising this to the fifth power and expanding things out I get something that is correct but not compact.

The result is an expression with three terms in a and b and three terms in a, b, and sqrt(5).

How do I tell Maple to collect the terms so that I get the result in this form:

(sum of terms without `sqrt(5)) + (sum of "coefficients" of terms with sqrt(5)) * sqrt(5),`

i.e. a number of the form `A + B*sqrt(5)`, where A and B are sums of terms in a and b ?

6.15.2 Robert Israel (10.8.01)

Well, one thing you can do is substitute a symbol for sqrt(5), collect with respect to that, and substitute back. For example:

```
> R:= (your expression);
> subs(sq5=sqrt(5),collect(subs(sqrt(5)=sq5,R),sq5));
```

6.15.3 Helmut Kahovec (12.8.01)

This is not at all a trivial task. Below you will find two possible solutions for MapleV/Release4. Each of them works with algebraic numbers of type `a+b*sqrt(D)` or `a+b*sqrt(-D)`. The first approach does not work within procedures.

Solution 1

```
> restart;
> alias(alpha=sqrt(5));
> x1:=a+b*alpha;
```

```
> x1e:=sort(frontend(collect,[expand(x1^5),alpha]));

         5       3  2          4       4         2  3       5
 x1e := a  + 50 a  b  + 125 a b  + (5 a  b + 50 a  b  + 25 b ) alpha

> x2:=a+b*I*alpha;

                          x2 := a + I b alpha

> x2e:=sort(frontend(collect,[expand(x2^5),alpha]));

        5       3  2          4
x2e := a  - 50 a  b  + 125 a b

             4           2  3         5
    + (5 I a  b - 50 I a  b  + 25 I b ) alpha

> alias(alpha=alpha);
> rules:={I='i',sqrt(5)=sqrt5};
> invrules:=map(u->op(2,u)=op(1,u),rules);
> subs(invrules,applyop(factor,-1,subs(rules,x1e)));

       5       3  2          4         4       2  2      4   1/2
      a  + 50 a  b  + 125 a b  + 5 b (a  + 10 a  b  + 5 b ) 5

> subs(invrules,applyop(factor,-1,subs(rules,x2e)));

      5       3  2          4           4       2  2      4   1/2
     a  - 50 a  b  + 125 a b  + 5 I b (a  - 10 a  b  + 5 b ) 5
```

Solution 2

```
> restart;

> collecta:=proc(e,K)
    if has(K,I) then
      evalc(expand(e));
      %-op(-1,%)+map(u->frontend(factor,[u]),op(-1,%)/(K/I))*(K/I)
    else
```

```
      sort(frontend(collect,[expand(e),K]));
      map(u->frontend(factor,[u]),%)
    fi
  end:

> x1:=a+b*sqrt(5);

> collecta(x1^5,sqrt(5));

       5       3  2          4         4       2  2      4   1/2
      a  + 50 a  b  + 125 a b  + 5 b (a  + 10 a  b  + 5 b ) 5

> x2:=a+b*sqrt(-5);
> collecta(x2^5,sqrt(-5));

      5       3  2          4           4       2  2      4   1/2
     a  - 50 a  b  + 125 a b  + 5 I b (a  - 10 a  b  + 5 b ) 5
```

6.15.4 Thomas Richard (13.8.01)

One approach is to use "remove" for the first sum and "select" for the second sum (note the trick to extract sqrt(5)):

```
> restart:
> w:=sqrt(5);
> f:=a+b*w;
> e:=expand(f^5);

      5      4    1/2       3  2       2  3  1/2          4
e := a  + 5 a  b 5    + 50 a  b  + 50 a  b  5    + 125 a b

           5  1/2
     + 25 b  5

> remove(has,e,w)+w*expand(select(has,e,w)/w);

      5       3  2          4    1/2     4         2  3       5
     a  + 50 a  b  + 125 a b  + 5    (5 a  b + 50 a  b  + 25 b )
```

6.16 algebraic relation (11.9.97)

6.16.1 Malte Henkel

Is there a way in Maple to perform systematic algebraic simplifications in complex calculations ? I would like to diagonalize a matrix which depends on a parameter r. It is known that this parameter satisfies an algebraic relation, `r^2 = r+1`, say.

How could I teach Maple to use this relation in combination with the eigenvals routine ? Is there a way through the assume facility ?

6.16.2 Stanley J Houghton (21.9.97)

You need to look at the help pages for simplify (use `?simplify,siderels`). This will show you how to simplify with respect to side relationships specified as a set of equations. Thus a simple example based on your query would be

```
> simplify(r^2+r^3+4*r^4,{r^2=r+1});

                              15 r + 10
```

6.17 algorithm of ei (9.2.96)

6.17.1 Xiaoqing Liu

I would like to calculate Ei (Exponential Integral) function with complex argument in my programming. Neither IMSL subroutine nor Numerical Recipes subroutines can do the job. It seems only Maple can do the job. But I don't know the algorithm of calculating Ei in Maple.

6.17.2 Dave Hare (20.2.96)

Maple calculates the n'th exponential integral (which it denotes `Ei(n,x)`) using one of several algorithms, depending on the value of n, the location of x in the complex plane and the precision required (the value of Digits).

The algorithms used (listed in the order in which they are considered) are:

```
- asymptotic series (A&S 5.1.51)
- Taylor series expansion centred at Re(x)
- continued fraction expansion (A&S 5.1.22)
- Taylor series expansion centred at 0 (A&S 5.1.11/12)
```

("A&S" = Abramowitz and Stegun). The Cauchy Principal Value function, usually denoted by Ei, and obtained in Maple as Ei(x), is defined (in Maple) only for real arguments x.

For more detailed information, you can look at the code directly, by readlib'ing and then print'ing any or all of the following routines:

```
evalf/Ei
evalf/Ei/{real,complex,asympt,taylory,confrac,taylor}
```

6.17.3 Peter Montgomery (21.2.96)

You can list many Maple procedures (though not their comments) using `interface(verboseproc = 2)`. Under Maple version 3, the work is done in `evalf/Ei/complex`. This program prints that procedure, which I'll let you decipher. The code is copyrighted.

```
interface(verboseproc = 2);
readlib(Ei);              # See Ei procedure
printlevel := 25;
Ei(5, 10 + 8.9*I);        # Calls evalf/Ei/complex
print(`evalf/Ei/complex`);
```

6.17.4 Robert Israel (21.2.96)

Well, since most of Maple's code is available, you can find out for yourself. The following should do it:

```
> interface(verboseproc=2);
> readlib(`evalf/Ei/complex`);
```

(and then look at the various procedures that this one calls). Good luck in trying to sort it out (there are no comments).

6.18 algsubs problem (12.10.00)

6.18.1 Paula R. Klink

I am a relatively new user of maple. I am trying to solve an equation of the form `y=(a*c)/(a+b)`. I would like to solve for another variable x which is equal to `a/(a+b)`. When I use

```
algsubs(a/(a+b)=x, y);
```

I receive an error stating: Error, (in algsubs) cannot compute degree of a pattern in a.

6.18.2 Tom Cassleman (18.10.00)

I think algsubs cannot recognize a "buried" pattern. I tried to rewrite your eqation as:

```
y:=a/(a+b)*c
```

and then

```
algsubs(a/(a+b)=x,y);
```

But that failed to! I think what happens here is that when you enter your equation, maple stores it as

`y=a*c/(a+b)` and cannot find the unique "pattern" `a/(a+b)` in that expression. This seems to be a problem with algsubs. Perhaps someone else can find a solution.

6.18.3 E. Elbraechter (30.10.00)

Note the expression `y=(a*c)/(a+b)` cannot be handled by algsubs but the substitution expression `a/(a+b)=x` has not the appropriate form.

First you can change the substitution expression `a/(a+b)=x` and then use algsubs or subs:

```
eq_y :=  y = (a*c)/(a+b);
a/(a+b)=x;
isolate(%, (a+b));
algsubs(%, eq_y, b);
subs(%%, eq_y);

                                  a c
```

```
                    eq_y := y = -----
                                a + b


                                 a
                               ----- = x
                               a + b


                              a + b = a/x


                                y = c x


                                y = c x
```

Second you use powsubs of the student package to substitute a factor:

```
student[powsubs](a/(a+b)=x, eq_y);

                                y = c x
```

6.19 algsubs, bug (25.5.01)

6.19.1 Luis Goddyn

For lack of knowlege of a better place to report these things, here is a bug appearing in Maple versions 3, 4, 5, 5.1 and 6.

```
> algsubs(x=y, 1/x );
                                    1/x
The correct output should be 1/y.
```

The bug affects expressions containing operands of the form `x^k` for negative integers k.

I have traced it to the following line in 'algsubs/expanded', which causes it to falsely assert that `x^k (k<0)` is already in "expanded form":

```
> `algsubs/expanded`(1/x,[x]);
                                  true

> print(`algsubs/expanded`);
proc(f, vars)
...
elif type(f, `^`) then
        evalb(not (has(op(1, f), vars) and type(op(1, f), `+`)))
...
```

It is easy but annoying to fix this problem. Does WaterlooMaple make available a patch file of elementary bug fixes such as this? ("elementary" = bugs not associated with the kernel). One could then simply tack it into the .mapleinit file and, VOILA, no need to wait until the next release version... I found not even a report of known bugs at `www.maplesoft.com`! Could someone please advise me of existing repositories?

6.19.2 Bill Page (28.5.01)

But the online documentation for algsubs (in Maple 5 at least) states specifically:

Note that the requirement for monomials in a to divide monomials in f means that the negative powers of u in the following example are not substituted, and must be handled separately as shown.

```
> f := a/u^4+b/u^2+c+d*u^2+e*u^4;

                          a      b           2      4
                 f := ---- + ---- + c + d u  + e u
                        4      2
                       u      u

> algsubs(u^2=v,f);
                                  2     b      a
                      d v + c + e v  + ---- + ----
                                       2      4
                                      u      u

> algsubs(1/u^2=1/v,f);
                                                2
                        4      2   a + v b + c v
                     e u  + d u  + -------------
                                         2
                                        v
```

Hence, to substitute for both positive and negative powers

```
> algsubs(u^2=v,algsubs(1/u^2=1/v,f));

                        4      3                2
                     e v  + d v  + a + v b + c v
                     ---------------------------
                                  2
                                 v
```

6.19.3 Helmut Kahovec (31.5.01)

Well, you may do the following:

```
> restart;
> seq(algsubs(1/x=1/y,1/x^k),k=1..10);

                1     1     1     1     1     1     1     1     1
          1/y, ----, ----, ----, ----, ----, ----, ----, ----, ---
                 2     3     4     5     6     7     8     9    10
                y     y     y     y     y     y     y     y    y

> seq(algsubs(1/x=1/y,x^(-k)),k=1..10);

                1     1     1     1     1     1     1     1     1
          1/y, ----, ----, ----, ----, ----, ----, ----, ----, ---
                 2     3     4     5     6     7     8     9    10
                y     y     y     y     y     y     y     y    y
```

6.20 alias for 2nd deriv (25.4.00)

6.20.1 Matthias Kawski

This probably is something old – but I can't find any help. Is there a way to define aliases for second derivatives? I tried all sorts of variations, of the following...

```
> restart;                              working w/ MAPLE V r.5
> alias(wt=diff(w(t),t)):
> diff(w(t),t);
                                  I, wt
                                    wt

> alias(utt=diff(u(t),t,t)):
> diff(u(t),t,t);
                                   2
                                  d
                                  --- u(t)
                                    2
                                  dt
```

I can work around this in simple examples as the above, but where I really need it is in some huge formulas involving derivatives of Christoffel symbols (working with my own programs, similar to the tensor-package functionality).

Thus, simple workarounds like using

```
      map(q->subs(diff(f(x,y,y),x,y,y)=fxyy,q),mylist);
```

are rather impractical.

6.20.2 Kayll Lake (26.4.00)

This feature is available in GRTensorII via "autoAlias".

Please see

`http://grtensor.phy.queensu.ca/`

6.20.3 Nathan Sokalski (26.4.00)

I believe there is a way to make an alias for the 2nd derivative. The following is an example I made. Hopefully you don't mind using D instead of diff.

```
> alias(second=(D@@2));
                              I, second
> f:=x->x^4;
                             f := x -> x^4
> second(f)(x);
                                  12x^2
```

I didn't really test this with any other functions, but I expect it would work the same as D (other than the fact that it's the 2nd derivative). And in case you can't figure it out yourself, second is just the variable name I chose. You could also modify this alias to do the 3rd, 4th, 1000th, or whatever derivative by changing it to `alias(second=(D@@3));` , `alias(second=(D@@4));` , `alias(second=(D@@1000));`.

6.20.4 Edgardo S. Cheb-Terrab (26.4.00)

I don't know how to change the behavior of Maple's 'alias' command, but I know how to somehow solve your problem.

First, concerning the display on the screen, in the PDEtools package distributed with Maple 6 (if you use Maple 5 or 4 you could download the versions ODEtools + PDEtools available on the web at `http://lie.uwaterloo.ca/PDEtools.htm`), there is a command called 'declare', which is used to declare "functionality" and change the display of derivatives. For example:

```
> with(PDEtools):
```

Now declare u as a function of t, and 't' as the "prime derivative":

```
    > declare( u(t), prime=t);

      derivatives with respect to:, t,
      of functions of one variable will now be displayed with '

                    u(t), will now be displayed as, u
```

Let's see the effect:

```
> diff(u(t),t,t) + u(t);

                              u'' + u
> lprint(%);
diff(diff(u(t),t),t)+u(t)
```

So:

```
* you still input diff(u(t),t,t);

* you 'see' derivatives displayed with a prime and u(t) as u

* the output however is really diff(u(t),t,t)
```

and hence your input and output hasn't changed but you read things in the screen with ease, as when you write formulas with your hand. For instance, compare this enhanced display for

```
> diff(y(t),`$`(t,2))*diff(y(t),t)*y(t)*f(t)-2*diff(y(t),t)^3*t^6
> +2*diff(y(t),t)^2*y(t)*diff(g(t),t)+y(t)^5;

                             3  6       2         5
          y'' y' y f - 2 y'  t  + 2 y'  y g' + y
```

with its standard output on the screen:

```
  / 2      \
  |d       | /d      \               /d      \3  6
  |--- y(t)| |-- y(t)| y(t) f(t) - 2 |-- y(t)|  t
  |  2     | \dt     /               \dt     /
  \dt      /

            /d      \2      /d      \       5
        + 2 |-- y(t)|  y(t) |-- g(t)| + y(t)
            \dt     /       \dt     /
```

..after seeing the enhanced one, the redundancy in above is close to intolerable for me.

This 'declare' command is potentially more helpful with partial derivatives of many-variable functions(for instance derivatives of Christoffel symbols), since it presents the derivatives as "indexed", which is what we use to do by hand. For instance:

```
> macro( G = Gamma(x,y,z) , df = diff);  # macros to simplify input
> declare(G);

            Gamma(x, y, z), will now be displayed as, Gamma

> df(G,x) + df(G,y)*G;

                        Gamma[x] + Gamma[y] Gamma
```

Finally, concerning not what you receive as output but what you type as input, if you download this PDEtools and ODEtools libraries from the web address on top, there is also a new command called `'diff_table'` which could facilitate your life too, and used in conjunction with 'declare' give the whole functionality you seem to be looking for. For example, let's assign to 'T' the `'diff_table'` of Gamma(x,y,z):

```
> with(ODEtools):

> T := diff_table( Gamma(x,y,z) ):
```

and now try

```
> T[];        # derivative of "order zero"

                                Gamma
```

The display above shows just "Gamma" because of the 'declare' statement, but what it is behind what you see is actually

```
> lprint(%);
Gamma(x,y,z)
```

The thing is that now you enter *any* partial derivative as an indexed object, you receive *the full standard Maple diff expression* and you see the thing with *compact

display* on the screen:

```
> T[x];
                                Gamma[x]

> lprint(%);
diff(Gamma(x,y,z),x)

> T[x,x,y,y,z];

                          Gamma[x, x, y, y, z]

> lprint(%);
diff(diff(diff(diff(diff(Gamma(x,y,z),x),x),y),y),z)

> T[x,y]*T[z] - T[y,z]*T[x] - T[z,x]*T[y];

  Gamma[x, y] Gamma[z] - Gamma[y, z] Gamma[x] - Gamma[x, z] Gamma[y]
```

and so on. This `'diff_table'` command also frees you of having to state an alias for each partial derivative you want to use and avoids potential collisions between your aliases and other Maple routines.

Hope this combination of 'declare' and `'diff_table'` helps. I use these commands all the time during interactive work and got so used with them that I can't even imagine going back to the standard Maple input/output.

6.20.5 Roberto Sussman (1.5.00)

I think that the solution to aliasing derivatives provided by Edgardo Chen-Terrab is too cumbersome. Here is an alternative.

I did a small procedure "difalias" that completely solves this problem. It aliases derivatives as subindices.

I work in General Relativity and I need to manipulate large tensor components full of partial derivatives. I find it very useful to be able to declare all functions and their dependencies (as in REDUCE) and not to have to carry them along.

Kayll Lake implemented the difalias procedure into grtensor but I do not like the way they did it: derivatives are aliased by functions are not !!

Then, from R5 onward (including R6) there is a display bug preventing aliasing of second (or higher) order derivatives. This bug carries on in grtensor implementation of difalias.

Joe Riel showed me how to correct this display bug: you need to erase the 'print/diff' library. He also wrote the code in terms of permutations of indices and up to arbitrary order of derivatives (difalias goes only up to 3rd order derivatives).

If you use R3 or R4 you can download difalias (the code and binaries) from my maple website

`http://www.nuclecu.unam.mx/~unamaple/difalias_.html`

If you are using R5 or R6 you can generate your own .m file by running the code below:

I hope it works !!

```
########## code follows #############
#########################################################
###The difalias procedure for R5
###Roberto A Sussman, December 1994
###Revised July 1999 (to correct aliasing bug in R5).
#########################################################
###Usage:
###for S=S(t) type difalias(t,S)
###for S=S(t,x,y,z) type difalias([t,x,y,z], S)
###for S,U,V depending on t type difalias(t, [S,U,V])
###for S,U,V depending on t,x,y,z type difalias([t,x,y,z], ###[S,U,V])
#########################################################
### Works up to 3rd order derivatives
#########################################################################
###If used in R5, the instruction `print/diff` := NULL: erases
###the `print/diff` library and unaliased derivatives look like
###nested derivatives.
###If used with R3 or R4, omit all reference to `print/diff`
###See http://www.nuclecu.unam.mx/~unamaple/difalias_.html
#########################################################################

  difalias :=

  proc(vars,funs)
  local i,j,k,l,nvar,nfun;
```

```
##the following instruction is only necessary in R5 and R6
global `print/diff`;
option `Copyright (c) 1994 by Roberto A Sussman. All rights reserved.`;
##the following instruction is only necessary in R5 and R6
`print/diff` := NULL:
    if type(vars,name) and type(funs,name)
     then
       RETURN(alias(funs = funs(vars),seq(
       funs[vars $ i] = diff(funs(vars),vars $ i),i = 1 .. 3)))
     elif type(vars,name) and type(funs,list)
      then
       RETURN(alias(seq(funs[i] = funs[i](vars),i = 1 .. nops(funs)),seq(seq(

       funs[i][vars $ j] = diff(funs[i](vars),vars $ j),
       i = 1 .. nops(funs)),j = 1 .. 3)))
     elif type(vars,list) and type(funs,name)
      then
       RETURN(alias(
       seq(funs = funs(op(vars)),i = 1 .. nops(vars)),
       seq(funs[vars[i]] = diff(funs(op(vars)),vars[i]),
       i = 1 .. nops(vars)),seq(seq(
       funs[vars[i],vars[j]] =        diff(funs(op(vars)),vars[i],vars[j]),

       i = 1 .. nops(vars)),j = 1 .. nops(vars)),
       seq(seq(seq(funs[vars[i],vars[j],vars[k]] =
       diff(funs(op(vars)),vars[i],vars[j],vars[k]),
       i = 1 .. nops(vars)),j = 1 .. nops(vars)),
       k=1..nops(vars)) ))
     elif type(vars,list) and type(funs,list)
      then
       RETURN(alias(seq(seq(
       funs[k] = funs[k](op(vars)),i = 1 .. nops(vars)),
       k = 1 .. nops(funs)),seq(seq(
       funs[k][vars[i]] = diff(funs[k](op(vars)),vars[i]),
       i = 1 .. nops(vars)),k = 1 .. nops(funs)),seq(seq(
       seq(funs[k][vars[i],vars[j]] =
       diff(funs[k](op(vars)),vars[i],vars[j]),
       i = 1 .. nops(vars)),j = 1 .. nops(vars)),
       k = 1 .. nops(funs)), seq(seq(seq(
```

```
        seq(funs[k][vars[i],vars[j], vars[l]] =
        diff(funs[k](op(vars)),vars[i],vars[j], vars[l]),
        i = 1 .. nops(vars)),j = 1 .. nops(vars)),
        k = 1 .. nops(funs)),l=1..nops(vars) )))
      else
        RETURN('difalias'(vars,funs))
      fi;
end;
```

6.20.6 Denis Pollney (8.5.00)

Roberto Sussman <sussky@mail.giga.com> wrote:

| I did a small procedure "difalias" that completely ...

The GRTensor 'autoAlias()' function is quite different from difalias() in a number of respects. It operates on an expression and automatically picks out terms which involve a derivative and applies an alias to these terms.

These derivatives can be of any order. The user does not have to specify the functions to be aliased or the independent variables. Thus

```
> autoAlias ( diff(p(x,y),x,y,y,y) + diff(q(r,t),r) + q(r,t) );

   p[x,y,y,y] + q[r] + q
```

In general, I'm not a big fan of aliasing for the standard reason: you invite a lot of confusion by, for example, aliasing something like f(r) by f. If you later refer to f(r) again, Maple thinks you are talking about f(r)(r), though it is not always apparent that this is what is happening from the output.

There is also a GRTensor-specific reason for not wanting to alias the function itself. Consider, for example, the Riemann tensor 'R' for some metric for which one of the components happens to be given by the function R(t). Once 'R(t)' has been aliased to 'R', you can not distinguish between it and the R which you would like to use for the tensor name.

The GRTensor autoAlias() only shies away from aliasing the function itself when such an ambiguity might exist, ie. when the alias corresponds to an existing GRTensor-defined tensor. In the above example, q(r,t) was aliased.

```
< Then, from R5 onward (including R6) there is a display bug ...
```

autoAlias() also makes use of Joe Riel's 'print/diff' trick for R5 (though in a somewhat different way than is done in difalias()) and can handle any number of derivatives.

For anyone interested, I've attached the most recent source code below. There's not too much GRTensor-dependent stuff there, so you should be able to read it into Maple and it should work for you (try the example above, for instance). Apologies for the lack of documentation and general state of the code, it's somewhat less than self-explanatory, I know.

```
# /*@@
#   @file   autoAlias.mpl
#   @date   9 Feb 1999
#   @author Denis Pollney
#   @desc
#           Alias functions so that their derivatives appear as
#           subscripts. Based on an idea by Roberto Sussman and
#           modified by Peter Musgrave for GRTensorII. Rewritten
#           for MapleV.5 by Denis Pollney based on a suggestion
#           by Joe Riel [MUG].
#   @enddesc
#   @hist
#            9 Feb 1999 Rewritten from scratch. Fixes problems with R5
#                       use of alias by modifying `print/diff` [dp]
#           27 Jul 1999 Added check for whether the aliased function
#                       also corresponds to a grtensor object, in which
#                       case it is not aliased [dp]
#   @endhist
# @@*/
#=============================================================================
grG_usedNameSet := {}:

autoAlias := proc (expr)
global dlist_, grG_usedNameSet:
local a, fn, f, x, df, aset, warnedSet:

  aset := {alias()}:
  dlist_ := NULL:

  # the following sets the global dlist_ variable:
```

```
  eval (subs(diff = `autoAlias/dlist`, expr)):

  warnedSet := {}:

  for a in {dlist_} do

    fn := op(1,a):
    f := `autoAlias/extractname` (fn):
    x := `autoAlias/extractvars` (fn), op(op(2,a)):

    if not member (op(0,f)[x], aset) then
      aset := {alias (eval (op(0,f)[x] = subs (diff=''diff'',`$`=''`$`'',
        readlib (`print/diff`)(f, x))))}:
    fi:

    if not member (op(0,f), aset) then

      if not member (op(0,f), grG_usedNameSet) then

        aset := {alias (op(0,f) = f)}:

      else

        if not member (op(0,f), warnedSet) then

          printf (`Warning: %a corresponds to a previously defined GRTensor object --
                   to avoid ambiguities, only its derivatives have been aliased.\n`, c
          warnedSet := warnedSet union {op(0,f)}:

        fi:
      fi:
    fi:

  od:

  dlist_ := 'dlist_':
  expr;
end:
```

```
`autoAlias/dlist` := proc (expr, d)
global dlist_:

  dlist_ := dlist_, [expr,{d}]:

  RETURN (diff(expr, d)):
end:


`autoAlias/diff` := proc (expr, diffby)
local varSeq, f, df, fname:

  varSeq := diffby:

  if type (expr, {function,procedure}) then

    f := expr:

    while type (f,{function,procedure}) and op (0,f) = ''diff'' do
      varSeq := op (2,f), varSeq:
      f := op (1,f):
    od:

    fname := `autoAlias/extractname` (f):

    print (expr, {varSeq}, {diffby});

    df := subs (diff=''diff'', `$`=''`$`'',
      readlib (`print/diff`)(f,diffby)):
    alias (fname[varSeq] = subs (diff=''diff'', `$`=''`$`'',
      readlib (`print/diff`)(f,diffby))):
    RETURN (fname[varSeq]):

  fi:

  RETURN (diff(fname,diffby)):
end:
```

```
`autoAlias/extractname` := proc (f)

  if op(0,f)=diff then
    RETURN (`autoAlias/extractname` (op(1,f))):
  fi:

  RETURN (f):
end:


`autoAlias/extractvars` := proc (f)
local a, v:

  if op(0,f)='diff' then

    v := NULL:

    for a from 2 to nops(f) do
      v := v, op(a,f):
    od:

    RETURN (`autoAlias/extractvars`(op(1,f)), v):

  else

    RETURN(NULL):

  fi:
end:
```

6.20.7 Ray Vickson ((15.5.00)

An *excellent* source is the book by Heck. I don't have it with me now, but its (somewhat misleading) title is "Introduction to Maple". He does things similar to what you want in some very complex examples.

6.21 alias of alias (16.12.99)

6.21.1 Francis Sergeraert

I don't understand how to manage *nested* extension fields. Typical example:

```
>alias(alpha = RootOf(X^2-2)) ;
                                  I   alpha

> alias(beta = RootOf(X^2-alpha)) ;
Error, (in RootOf) expression independent of, _Z
```

Why this error ? Of course I try to replace X by the system _Z:

```
> alias(beta = RootOf(_Z^2-alpha)) ;

                               I   alpha   beta

> evala(beta^4) ;
                                  alpha^2
```

Obviously the answer should be 2 ; this looks like if only the beta-definition is considered. Still more strange, this works :

```
> evala(alpha^2) ;
                                     2
```

But neither this:

```
> evala(evala(beta^4)) ;
                                  alpha^2
nor this:

> evala(beta^4) ; evala(%) ;
                                  alpha^2
                                  alpha^2
```

The explanations in the documentation about Primfield, evala, ... are not concerned by examples of this type. Of course in my example it would be sufficient to give

`alias(beta = RootOf(X^4-2))`, but I'm in fact interested by much more complicated situations.

6.21.2 Stanley J Houghton (17.12.99)

Your problem stems from an alias defined in terms of another alias.

Maple states in the help page for alias as follows:

"The arguments to alias are equations. When alias is called, the equations are evaluated from left to right, but are not subjected to any existing aliases. Therefore, you cannot define one alias in terms of another. Next, the aliases are defined. Finally, a sequence of all existing aliases is returned as the function value."

Thus it is not allowed.

However, I think it would be better if Maple returned an error message to make this clear.

6.21.3 Michael Monagan (17.12.99)

The problem is the way alias works.

A fix (not explained) is this

```
> nestalias := proc() alias(args) end;

                    nestalias := proc() alias(args) end

> nestalias(alpha = RootOf(X^2-2)) ;

                                  I, alpha

> nestalias(beta = RootOf(X^2-alpha)) ;

                               I, alpha, beta

> evala(beta^4);
                                      2
```

Note, you can't use nestalias to turn off or change an alias.

6.21.4 Helmut Kahovec (20.12.99)

The following may solve your problem:

```
> restart;
> alias(alpha=RootOf(X^2-2));

                                I, alpha

> alias(beta=RootOf(X^2-RootOf(X^2-2)));

                             I, alpha, beta

> map(evala,[alpha^2-2,beta^4-2]);

                                [0, 0]
```

6.21.5 Helmut Kahovec (10.1.00)

Michael Monagan wrote: ...

That's a very nice trick, really. If I correctly understand it, then one has to delay executing alias():

```
> restart;
> alias(alpha=RootOf(X^2-2));

                                I, alpha

> eval('alias'(beta=RootOf(X^2-alpha)));

                             I, alpha, beta

> evala(alpha^2);

                                   2

> evala(beta^4);

                                   2
```

Thus you don't have to spend another procedure name.

6.22 alias problem (25.1.99)

6.22.1 Roberto A Sussman

The command alias in Release 5 and 5.1 fails to alias second (or greater) order derivatives. Consider the following instructions

```
> alias(F=F(x));
> alias(Fx=diff(F(x),x));
```

These I copied from the examples in the help file of alias. Everything is OK up to this point, if you key in

```
> diff(F,x);
```

you get "Fx", as stipulated by the alias. However, as you key in the following alias instruction

```
> alias(Fxx=diff(F(x),x,x));
```

you get the right output `(I,F,Fx,Fxx)`, but it does not work: the second derivative is not aliased since the call

```
> diff(F,x,x);
```

does not produce "Fxx" as in the first order derivative. This does not happen in Release 4 and 3 (the aliasing works fine), but it happens in R5. I am using R5 and R5.1, mac and windows versions (the hardware is irrelevant).

Is this a bug? or am I missing some very special Maple-related subtlety?

For those interested, I extended the alias command to a "difalias" procedure that includes aliasing of partial derivatives up to 3rd order. The derivatives appear as subindices and can be differentiated with diff and manipulated as indexed names.

This utility is very helpful for manipulating large expressions containing partial derivatives (or ordinary ones). The difalias code has been already incorporated into the tensor manipulator grtensor, though, they removed the zero order aliasing, which in my opinion defeats the purpose of this utility. I also did an unaliasing procedure.

However, as I was porting these simple procedures from R4 to R5, I noticed the (possible)

bug I am reporting, and so difalias only aliases first derivatives in R5.

Those interested in getting the code or binaries of difalias and undifalias, please send me a note to

sussman@nuclecu.unam.mx

I will also place the material soon in my web page http://www.nuclecu.unam.mx/~sussman for downloading

6.22.2 Willard, Daniel, Dr.(3.2.99)

From the Help file on alias I quote: "This means that you cannot define one alias in terms of another. " I suspect this is your problem.

Try:

```
alias(Fx=diff(F(x),x));
alias(Fxx=diff(F(x),x$2));
alias(Fxxx=diff(F(x),x$3));
```

this seems to work ad lib:

```
diff(Fxxxx,x$3)=diff(F(x),x$7): etc.
```

6.22.3 Joe Riel (4.2.99

Roberto Sussman asks why alias doesn't work properly in R5 with second or higher order derivatives. The problem lies in the 'print/diff' procedure, which is new for R5. In order to avoid an ugly display, 'print/diff' combines repeated terms in a derivative. To see this, erase the 'print/diff' procedure and see what happens.

```
> restart;
> interface(prettyprint=0);
> diff(f(x,y),x,x);
diff(f(x,y),`$`(x,2))
> `print/diff` := NULL:
> diff(f(x,y),x,x);
diff(diff(f(x,y),x),x)
```

Note that with 'print/diff' erased, aliasing works correctly. The output of second and higher order differentials, however, is very ugly.

Because the print alias mechanism operates on the output of a print procedure, you can get aliasing to work by properly aliasing to the output of print.

The following procedure simplifies the process

```
aliasdiff := proc(eq)
local fx,dfx;
option `Copyright (c) 1999 by Joseph Riel. All rights reserved.`;
    if nargs > 1 then RETURN(map(procname,[args])[]) fi;
    fx,dfx := op(eq);
    fx = subs(diff=''diff'',`$`=''`$`'',readlib(`print/diff`)(op(dfx)));
    alias(%);
    fx
end:
```

For example,

```
alias(f = f(x,y)):
aliasdiff(fx=diff(f,x), fxx=diff(f,x,x),
          fy=diff(f,y), fyy=diff(f,y,y), fxy=diff(f,x,y)):
\begin{MAPLEinline}

Note that aliasdiff, unlike alias, allows you to define aliases
in terms of previously defined aliases.

\begin{MAPLEinline}
> f = f(x,y);

                              f = f(x, y)
> diff(%,x);
                                  d
                            fx = -- f(x, y)
                                  dx
> diff(%,y);
                                   2
                                  d
                          fxy = ----- f(x, y)
```

```
dy dx
```

6.23 alias within a procedure (25.4.01)

6.23.1 Juansi Dominguez

I'm writing in Maple V an algorithm that calculates a polynomial (error-locator polynomial). I'm working in the finite field F_16, so a primitive element satisfy `a^4+a+1`. So I want to have my output (that polynomial) in that field. I mean that he must simplify eg `a^15` in 1. To get that I make an alias: `alias(al=RootOf(x^4+x+1));` and then I simplify my polynomial.

But the problem is that he does not simplify it in the algorithm. The first time I run the algorithm with eg

```
sugiyama(al^9+al^12*x+x^2+al^14*x^3+al^13*x^4+x^5);
```

He gives me a polynomial that is not simplified (thus he gives me a very large polynomial with `al^130` in it)

If I run the program for the second time he does simplify it. Why is it that he won't simplify the polynomial the first time. The problem is now that I can't use this algorithm in another algorithm because the polynomial are to big (I want to have those polynomials (the primitive root alpha reduced).

The algorithm is as follows:

```
> sugiyama:=proc(E)
> local a, b, r, A, t;
> alias(al=RootOf(x^4+x+1));  # the alias!
> t:=(degree(E,x)+1)/2:
> a[0]:=x^(2*t):
> b[0]:=E:
> A[0]:=array(1..2,1..2,[[1,0],[0,1]]):
> r:=1:
> while degree(b[r-1],x) >= t do
>   b[r]:=simplify(a[r-1]-(Quo(a[r-1],b[r-1],x) mod 2)*b[r-1]) mod 2;
>   a[r]:=b[r-1];
>   A[r]:=simplify(evalm(evalm((array(1..2,1..2,[[0,1],
                  [1,-Quo(a[r-1],b[r-1],x)mod 2]])))&*evalm(A[r-1])));
>   r:=r+1;
> od;
> collect(simplify(A[r-1][2,2]) mod 2, x);  # and to simplify....
> end:
```

6.23.2 Rafal Ablamowicz (25.4.01)

The reason your program needs two executions in order to get the desired result is as follows:

During the first execution of your program, Maple distinguishes between the formal parameter 'al' that you use in your input and the alias 'al' that you define inside of your procedure. In short, the alias definition does not affect the formal parameter 'al' and that's why your program returns long, unsimplified result.

If you go to the help page ?alias, you will find there this sentence:

"Parameters and local variables are not affected by aliases."

To see this more clearly, I have added one extra line to your program to show what happens. [Note: To see Maple output, please copy and paste the text below into maple V worksheet and execute.]

```
> sugiyama:=proc(E) local a,b,r,A,t;
> alias(al=RootOf(x^4+x+1));
> print(evalb(al=RootOf(x^4+x+1))); #my extra line
> t:=(degree(E,x)+1)/2:
> a[0]:=x^(2*t):
> b[0]:=E:
> A[0]:=array(1..2,1..2,[[1,0],[0,1]]):
> r:=1:
> while degree(b[r-1],x) >= t do
> b[r]:=simplify(a[r-1]-(Quo(a[r-1],b[r-1],x) mod 2)*b[r-1]) mod 2;
> a[r]:=b[r-1];
> A[r]:=simplify(evalm(evalm((array(1..2,1..2,[[0,1],
                [1,-Quo(a[r-1],b[r-1],x)mod 2]])))&*evalm(A[r-1])));
> r:=r+1;
> od;
> collect(simplify(A[r-1][2,2]) mod 2, x);  # and to simplify....
> end:
> sugiyama(al^9+al^12*x+x^2+al^14*x^3+al^13*x^4+x^5); #first run
```

If you execute the above code, you will notice output 'false' from the extra line "`print(evalb(al=RootOf(x^4+x+1)));`" This means that Maple doesn't recognize

that alias yet even though it is recognizable outside of the procedure now:

```
> print(evalb(al=RootOf(x^4+x+1)));
```

However, when you execute your program one more time, this time 'al' in the input is recognized as one of the existing aliases and correct answer is returned:

```
> sugiyama(al^9+al^12*x+x^2+al^14*x^3+al^13*x^4+x^5); #second run
```

Notice 'false' in the above even though separately issued command below yields the expected result, like before:

```
> print(evalb(al=RootOf(x^4+x+1)));
```

This is Maple's peculiarity!

The only solution to your question of avoiding executing twice the same procedure so that alias 'al' is recognized, the way I see it, is to define that alias before you execute your program like this:

```
> alias(al=RootOf(x^4+x+1)); #alias defined before the procedure

> sugiyama2:=proc(E) local a,b,r,A,t;
> #alias(al=RootOf(x^4+x+1)); #no alias definition in the procedure
> print(evalb(al=RootOf(x^4+x+1))); #my extra line
> t:=(degree(E,x)+1)/2:
> a[0]:=x^(2*t):
> b[0]:=E:
> A[0]:=array(1..2,1..2,[[1,0],[0,1]]):
> r:=1:
> while degree(b[r-1],x) >= t do
> b[r]:=simplify(a[r-1]-(Quo(a[r-1],b[r-1],x) mod 2)*b[r-1]) mod 2;
> a[r]:=b[r-1];
> A[r]:=simplify(evalm(evalm((array(1..2,1..2,[[0,1],
                [1,-Quo(a[r-1],b[r-1],x)mod 2]])))&*evalm(A[r-1])));
> r:=r+1;
> od;
> collect(simplify(A[r-1][2,2]) mod 2, x);  # and to simplify....
> end:
> sugiyama2(al^9+al^12*x+x^2+al^14*x^3+al^13*x^4+x^5); #first run
```

The above gives the expected result in one run.

I have not found a way to fool Maple into replacing with the alias, that you want to define inside of your procedure, the formal parameter used by you.

My suggestion is just to define alias 'al' in your worksheet before you ever use your new procedure 'sugiyama2'.

6.23.3 Carl DeVore (26.4.01)

Your problem is that the alias command needs to be at the global level. I also simplified your algorithm. There is no need for all that indexing and those array commands. I think that this is much easier to understand:

```
> sugiyama:=proc(E)
>    local a, b, x, A, t, new_a, Q;
>    x:= indets(E)[];
>    t:= (degree(E,x)+1)/2;
>    a:= x^(2*t);
>    b:= E;
>    A:= [[1,0],[0,1]];
>    while degree(b,x) >= t do
>       Q:= Quo(a,b,x) mod 2;
>       new_a:= b;
>       b:= simplify(a-Q*b) mod 2;
>       a:= new_a;
>       A:= evalm([[0,1],[1,-Q]] &* A)
>    od;
>    collect(simplify(A[2,2]) mod 2, x)  # and to simplify....
>  end:
>
> alias(al=RootOf(x^4+x+1));  # the alias!
```

al

```
> sugiyama(al^9+al^12*x+x^2+al^14*x^3+al^13*x^4+x^5);

                2     3   3           3   2                2     3
  (1 + al + al  + al ) x  + (al + al ) x  + (1 + al + al  + al ) x
```

```
              2     3
       + 1 + al  + al
```

Note that the way that I set it up, you do not need to use "x" as the variable in the polynomial.

6.24 aliased i (13.8.99)

6.24.1 RossAllan McKenna

I have a number of expressions that should evaluate to zero, but remain as, e.g. abIc-Iacb. Even if I `algsubs abc=1`, it remains I-I for calls to eval, evala, or evalc. What am I missing?

6.24.2 Jurgen Barsuhn(17.8.99)

Could it be that you just forgot the multiplication operator * which is necessary for the Maple input but does not appear in the prettyprinted output?

```
> a*b*I*c-I*a*c*b;

                                  0
```

6.24.3 Helmut Kahovec (20.8.99)

It looks like you have returned a local variable from a procedure:

```
> restart;
> f:=proc(a,b,c) local I; a*b*I*c-sqrt(-1)*a*c*b end:
> f(a,b,c);

                          a b I c - I a c b

> eval(%),evala(%),evalc(%);

        a b I c - I a c b, a b c (I - I), a b I c - I a c b

> subs({a=1,b=1,c=1},%%);

                               I - I
```

6.24.4 RossAllan McKenna (31.8.99)

The manipulations involved pages of equations, but I did finally track down the problem. One of the elements involved an alias which contained an expression defined in terms of what was intended to be the imaginary number 'I'. A greatly simplified equivalent would be:

```
restart;
alias(y=I*x);
z:=I*x;
f:=diff(y,x)-diff(z,x);
```

Beware the aliased alias!

6.24.5 Heike Koch-Beuttenmueller (8.9.99)

To understand the problem, try the following:

```
> alias(y=I*x);
> z:=I*x;
> z;
> diff(y,x)-diff(z,x);
                                  I - I
> simplify(%);
                                  I - I
> a1:=diff(y,x);
                                a1 := I
> whattype(%);
                                 symbol
```

Here I is only a symbol , not an alias !!!!

Maple help writes:

When alias is called, the equations are evaluated from left to right but are not subjected to any existing aliases. This means that you cannot define one alias in terms of another.

But using I in the above equation you defined one alias in terms of another.

```
> a2:=diff (z,x);
                                a2 := I
```

```
> whattype(%);
                                  ^
> a1-a2;;
                                I - I
> whattype(%);
                                  +
> alias(v=sqrt(-1)*x);
> a3:=diff(v,x);
                               a3 := I
> whattype(a3);
                                  ^
> a2-a3;
                                  0
```

6.25 allvalues (31.1.97)

6.25.1 Liangfu Zhou

I am having a trouble with allvalues.

I have a result from solve containing RootOf. The Argument to the RootOf is a fourth order polynomial containing symbolic coefficients.

In my program, I need to run allvalues four times at a loop. Sometimes I can get the answers at first three, then get the error message at last one. Sometimes I only can finish first or second loop. I never get four complete answers.The error message are always the same:

Cannot evaluate Allvalues root sequence with symbolic coefficients.

The manual indicates that this error message happens with a RootOf with a fifth order (or higher) equation.

I recall reading that only sometimes is there a symbolic solution for a fourth-order equation. However, it appears that we can run allvalues one time on a fourth order equation and get a result.

Then, we try allvalues again on the exact same equation and get the above error message.

We are trying to characterize the problem–it's happening inside a large program.

Is the allvalues routine using a random algorithm somewhere so that it will sometimes work and sometimes not work?

6.25.2 Bruce Hartley

Maple release 3 will do this if you set the environmental variable _EnvExplicit to true eg

`_EnvExplicit:=true` The general quartic will then solve for symbolic coefficients. I don't know whether this works when you have a RootsOf expression for the solution of an equation but perhaps it will not give such expressions for quartic equations if the `_EnvExplicit` is set true.

6.25.3 Marc A. Murison

`_EnvExplicit` works in release 4 as well. Also, an alternativee to the `_EnvExplicit` approach is to use the function, allvalues(). That is,

```
eq := a*x^4 + b*x^3 + c*x^2 +d*x + e:
solve(eq,x):    #produces RootOf(a*_Z^4+b*_Z^3+c*_Z^2+d*_Z+e)
allvalues(%);   #turns the RootOf into the full, explicit set of solutions
```

6.25.4 Andrzej Pindor

I have found that, at least in Maple V R4, you have to be very carefull when using symbolic solutions to the quartic equation. For instance, try the following:

```
1. get the roots of a polynomial x^4+a*x^3+b*x^2+c*x+d .

2. put a=b=c=0 and simplify.

3. put d=1 - after some massaging (using combine twice) you will get correct roots.

4. put d=-1 - immediately you get all roots equal to zero. Now try this:

1. get the roots of a plynomial x^4+a*x^3+b*x^2+c*x+d .

2. put a=b=c=0, d=-1 and simplify - Maple comes out with the following message:

   Error (in simplify/recurse) division by zero.
3. put a=0=c=0, d=1 and simplify - you will get complex expressions with radicals.
```

If you now use combine followed by simplify you will get roots as `1,-I,I,-1` which is incorrect. If you instead try evaluate the said expressions in floating point you will get correct answers.

6.26 allvalues, number of solutions (3.4.00)

6.26.1 Eugenio Roanes Lozano

Can anyone give an explanation for the following strange behaviour of Maple V.5.1 (it is clear that the system represents the intersection of a circle and a parabola):

```
allvalues( solve( {x^2+y^2-1,x^2-y} , {x,y} ) ):
```

Nevertheless, Maple returns 8 (!) points (in fact there are two real solutions and two complex -non real- ones). The 4 extra points that Maple returns do not satisfy the original equations.

6.26.2 Helmut Kahovec (8.4.00)

RootOf() has been remarkably extended in maple6. One gets the following four solutions in maple6:

```
> restart;
> x^2+y^2-1,x^2-y;

> solve({%},{x,y});

                              2                       2
  {x = RootOf(-RootOf(_Z + _Z  - 1, label = _L1) + _Z , label = _L2),

                           2
       y = RootOf(_Z + _Z  - 1, label = _L1)}

> allvalues(%);

  {y = - 1/2 + 1/2 sqrt(5), x = 1/2 sqrt(-2 + 2 sqrt(5))},

       {y = - 1/2 + 1/2 sqrt(5), x = - 1/2 sqrt(-2 + 2 sqrt(5))},

       {x = 1/2 sqrt(-2 - 2 sqrt(5)), y = - 1/2 - 1/2 sqrt(5)},

       {x = - 1/2 sqrt(-2 - 2 sqrt(5)), y = - 1/2 - 1/2 sqrt(5)}
```

6.26.3 Jurgen Barsuhn (11.4.00)

As far as I know, you cannot avoid those many solutions, of which only a part is appropriate. As in many cases, in your case "solve" returns RootOf-expressions, telling that the solutions are among the roots of certain polynomials. allvalues returns all roots of these polynomials, even those that are no solutions to the original problem. The only way I know is to insert the "solutions" found into the original equations, as is demonstrated in the following piece of Maple code. You may rewrite the loop for automatically discarding the "non-solutions".

```
> solve( {x^2+y^2-1,x^2-y} , {x,y} );
> restart;
> equat:=x^2+y^2-1,x^2-y;
> solve({equat},{x,y});
> Solution:=allvalues(%);
> for i to nops([Solution]) do
> i,simplify(subs(Solution[i],[equat])),Solution[i],evalf(Solution[i])
> od;
```

On the other hand, fsolve returns only one solution at a time. You can look for the second solution by giving a search interval. Hence you need to have an idea in advance, how many solutions exist and where they approximately are situated (e.g. by a graph).

```
> fsolve({equat},{x,y});
> fsolve({equat},{x,y},{x=0..1});# if the first statement found the
                                 # "negative x solution"
```

6.27 alternating series, sum and add (22.11.99)

6.27.1 John Kurtzke

I have found Maple adds terms in a series very quickly, but I have had problems with the alternating harmonic series, `sum( (-1)^(n+1)/n )` It will sum this up to 1000 quickly, but with 10,000 or more, Maple gobbles up memory to the limit and locks things up.

Specifically:

`evalf( sum( ((-1)^(n+1))/n, n=1..N) );`

works well when `N= 1000`, but not at all for `N = 10000`.

I then broke things up to even & odd and Maple computed the sums in a flash:

```
odds := evalf( sum( 1/(2*n+1), n=1..N) );
evens := evalf( sum(1/(2*n), n=1..N) );
```

I used 10000 for N and Maple computed these two sums in the blink of an eye.

I suspect Maple is having trouble with the alternating nature of the original series, but I don't know why.

This has happened using Maple V Release 5.0 on both my (old) `Power Mac 8100/100` and on a `Pentium III` PC in one of our computer classrooms.

6.27.2 Bill Bauldry (23.11.99)

Consider using an inert Sum instead of sum. The sum attempts an antidifference, then goes to the evalf. The Sum just dumps to evalf. Here's what I get on a Mac G3:

```
> restart;
> S := N -> evalf( sum( ((-1)^(n+1))/n, n=1..N) ):
> SI := N -> evalf( Sum( ((-1)^(n+1))/n, n=1..N) ):
> time():
> evaln(S)[1000] = S(1000);
> (time() - %%)*seconds;

> time():
> evaln(SI)[1000] = SI(1000);
> (time() - %%)*seconds;
```

```
                          S[1000] = .6926474306
                                .317 seconds

                          SI[1000] = .6926474306
                                .067 seconds
> time():
> evaln(S)[10000] = S(10000);
> (time() - %%)*seconds;

> time():
> evaln(SI)[10000] = SI(10000);
> (time() - %%)*seconds;
                          S[10000] = .6930971831
                              12.333 seconds

                         SI[10000] = .6930971831
                                .783 seconds
```

6.27.3 Robert Israel (23.11.99)

I don't think that's it at all. The "sum" function is for symbolic summation. It finds `sum((-1)^(n+1)/n, n=1..N)` (with N left unassigned) as

```
                            (N + 1)
          ln(2) + 1/2 (-1)          (Psi(1 + 1/2 N) - Psi(1/2 N + 1/2))
and it finds sum(1/n, n=1..N) as

          Psi(N + 1) + gamma
It finds sum(1/n, n=1..10000) as

          Psi(10001) + gamma
```

(which "evalf" can then handle, giving a numerical answer 9.787606026). However, for some reason (I don't know exactly why) it doesn't calculate the symbolic solution for `sum((-1)^n/n, n=1..N)` when N has a numerical value; instead it simply adds the terms of the sum. This is done with rational arithmetic, so the result will be a rational number with huge numerator and denominator: for N=10000 the numerator and denominator each have 4346 digits, and I expect that for N=10000 they will have

about 4320 digits.

If you want a numerical answer obtained from adding up the individual terms, you can use

```
> evalf(Sum((-1)^(n+1)/n, n=1..10000));
```

Or if you want to evaluate the symbolic answer numerically, you can use

```
> evalf(eval(sum((-1)^(n+1)/n, n=1..N), N=10000));
```

6.27.4 Douglas B. Meade (24.11.99)

There are a number of issues involved with your observation. I am working with Release 5.1 under Solaris (UNIX) so some of the memory issues are a little different.

First, you need to be aware of the differences between the sum and add commands. The sum (and Sum) commands attempt to find explicit formulae for definite and indefinite sums; only add actually adds each term in the sum. (Well, sum will add the terms if all attempts to find an explicit formula fail.)

The following session illustrates some of these differences. Note how Maple automatically simplifies the sum when sum is used. This makes all evaluations extremely fast. The next section, which uses the intert summation, Sum, is similar.

It is somewhat disturbing that Maple could not make up its mind whether to return a difference of Psi functions or the LerchPhi function, but that is an issue for the developers (hint, hint). In the third section, the sum is computed using add.

Here we see a slowdown as N increases. In this demonstration the timings are slight superlinear; in other tests they were much more linear. The specifics of these timings are likely to be very dependent on the amount of memory, etc.

I hope you will find this response informative, even if it only indirectly addresses your question about even and odd sums.

```
> restart;
> s := sum( (-1)^(n+1)/n, n=1..N );

                          (N + 1)
   s := ln(2) + 1/2 (-1)         (Psi(1 + 1/2 N) - Psi(1/2 N + 1/2))
```

```
> st := time(): N := 10^3: evalf( s ); time()-st;

                              .692647431    .010

> st := time(): N := 10^4: evalf( s ); time()-st;

                              .693097184     0

> st := time(): N := 10^5: evalf( s ); time()-st;

                              .693142181     0

> st := time(): N := 10^3-1: evalf( s ); time()-st;

                              .693647431     0

> st := time(): N := 10^4-1: evalf( s ); time()-st;

                              .693197184     0

> st := time(): N := 10^5-1: evalf( s ); time()-st;

                              .693152181     0


> restart;
> S := Sum( (-1)^(n+1)/n, n=1..N );

> st := time(): value( S ); time()-st;

                         (N + 1)
           ln(2) + (-1)          (-LerchPhi(-1, 1, N) + 1/N)

                                .210

> st := time(): N := 10^3: evalf( S ); time()-st;

                             .6926474306     .080
```

```
> st := time(): N := 10^4: evalf( S ); time()-st;

                              .6930971831    1.000

> st := time(): N := 10^5: evalf( S ); time()-st;

                              .6931421806   11.150

> st := time(): N := 10^3-1: evalf( S ); time()-st;

                              .6936474306     .080

> st := time(): N := 10^4-1: evalf( S ); time()-st;

                              .6931971831    1.020

> st := time(): N := 10^5-1: evalf( S ); time()-st;

                              .6931521806   11.150

> restart;
> N := 10^3: st := time(): evalf( add( (-1)^(n+1)/n, n=1..N )); time()-st;

                              .6926474306     .220

> N := 10^4: st := time(): evalf( add( (-1)^(n+1)/n, n=1..N ) ); time()-st;

                              .6930971831   16.410

> N := 10^4-1: st := time(): evalf( add( (-1)^(n+1)/n, n=1..N ) ); time()-st;

                              .6931971831   16.540
```

6.28 an easy integral. not for maple 6. (31.10.00)

6.28.1 Don Hartig

The following calculation was made in Maple 5.1 (courtesy of George Luna):

```
> restart:
> Int(Int(sqrt(x^3+1),x=sqrt(y)..1),y=0..1);

                      1    1
                     /    /
                    |    |              3
                    |    |        sqrt(x  + 1) dx dy
                    |    |
                   /    /
                     0    sqrt(y)

> evalf(%);

                                .4063171388
Try it in Maple 6.
```

It is corrected with Maple 7. (U. Klein)

6.28.2 HARALD PLEYM (2.11.00)

Here is what I get:

```
>A:=Int(Int(sqrt(x^3+1),x=sqrt(y)..1),y=0..1);
> evalf(A);
Error, (in depends) too many levels of recursion
```

But if you try

```
A:=Int(Int(sqrt(x^3+1),x=sqrt(y)..1),y=0..1);
in R5 you get some of the same error

evalf(%);
Error, (in LinearProp) too many levels of recursion
```

6.28.3 TANGUY Christian (6.11.00)

Funny indeed, all the more so when you try `int(int(...))` instead of `Int(Int(...))`. The first integration is performed, but an `evalf(%)` then freezes Maple 6 on my `Pentium II. 2/9*(sqrt(8)-1)` might be reached after SOME time..

6.29 analytic linear system solving (27.11.97)

6.29.1 Charles LANTA

Forwarded from Olivier Froment SNCF - Ecole Centrale de Paris

I would like to solve symbolically a linear system of the (classical) form : `Ax=b`

where A is "almost "full `6*6` square matrix

Up to 5, that seems possible but with 6 the objects manipulated seem too large...

Is there something I can do for simplifying , before solving the system, the shape of matrix A to make it work?

6.29.2 Tom Holly (28.11.97)

I once had a verb|16x16| matrix to deal with, and I found using the LU factorization to be very helpful.

6.29.3 Colin Birch (3.12.97)

By coincidence I also recently wanted a symbolic solution for an "almost" full `6*6` square matrix. Mine solved, but took a long time to produce a very large and awkward solution. In the end I found the LU decomposition much more useful (command `linalg[LUdecomp](...)`).

In fact it was fairly easy to apply an LU decomposition such as Crout's algorithm (Numerical Recipes in C, W.H. Press et al. (1992)) manually to the matrix.

6.30 andprop and orprop evaluating (29.7.97)

6.30.1 Jaroslav Hajtmar

Is there a way to evaluate AndProp and OrProp functions? These functions works very good, but output form is not mathematical. I would like to get result in RealRange(s) form(s).

For example:

```
> rr1:=RealRange(0,10);
> rr2:=RealRange(5,20);
> result1:=AndProp(rr1,rr2);
> result2:=OrProp(rr1,rr2);

                        rr1 := RealRange(0, 10)

                        rr2 := RealRange(5, 20)

        result1 := AndProp(RealRange(0, 10), RealRange(5, 20))

        result2 := OrProp(RealRange(0, 10), RealRange(5, 20))
```

I want to get result1 in form `RealRange(5,10)` and result2 in form `RealRange(0,20)`.

I think that is it a big problem, especially when borders of real intervals are open.

I want to write any procedures, which do set operations on RealRanges by analogy set operations "union", "intersect" and "minus", because AndProp and OrProp output forms are very unintelligible. What do you think about?

6.30.2 Robert Israel (30.7.97)

Since "about" knows how to simplify range expressions, the easiest way may be to use it. The properties of an object can be recovered from the table `property/object`.

```
> assume(zz, result1);
> `property/object`[zz];

          RealRange(5,10)

> assume(zz, result2);
> `property/object`[zz];

          RealRange(0,20)
```

6.30.3 Gaston Gonnet (6.8.97)

If you load the assume function, then the functions are defined and you will have the desired behaviour. E.g.

```
> assume(x>0);
> rr1:=RealRange(0,10);
                          rr1 := RealRange(0, 10)

> rr2:=RealRange(5,20);
                          rr2 := RealRange(5, 20)

> result1:=AndProp(rr1,rr2);
                        result1 := RealRange(5, 10)

> result2:=OrProp(rr1,rr2);
                        result2 := RealRange(0, 20)
```

6.31 animate & stats distribution (23.9.98)

6.31.1 Richard Gonzalez

I'm trying to prepare a demo for class. I want to animate some distributions as a function of degrees of freedom and am having problem (most likely I'm misunderstanding something about evaluation).

Here is a static plot of an example of what I want to do:

```
with(stats);
display({plot(statevalf[pdf, studentst[25]](x),
x=-6..6,colour=red),
plot(statevalf[pdf, studentst[5]](x), x=-6..6,colour=blue),
plot(statevalf[pdf, studentst[2]](x),
x=-6..6,colour=black),plot(statevalf[pdf, studentst[1000]](x),
x=-6..6,colour=green)});
```

I thought the following would be a way to animate the plot as a function of the parameter of "studentst" but all I get are axes

```
with(plots);
animate(statevalf[pdf, studentst[t]](x), x=-3..3,t=5..1000);
```

this example works fine

```
animate( sin(x*t),x=-10..10,t=1..2);
```

I don't understand why the latter animate works and the former doesn't.

6.31.2 Robert Israel (24.9.98)

Premature evaluation strikes again! The animate function, like most Maple functions, evaluates its arguments first. The result is that `statevalf[pdf, studentst[t]](x)` gets called with a symbolic t, and statevalf doesn't like that: it insists on the parameter of studentst being numerical. The simplest way to get around it, I think, is to use `display(..., insequence=true)` instead of animate. Something like this:

```
> display([seq(plot(statevalf[pdf, studentst[5+100*j]](x), x=-3..3),
                        j=0..10)], insequence=true);
```

This works because seq is not like most Maple functions: it only evaluates the first argument after substituting a numerical value for the loop variable j.

6.31.3 Bill Bauldry (24.9.98)

It's an evaluation problem. My favorite trick for evaluating animate works here, too. Try:

```
with(stats):
> with(plots):
> clr := vector([blue,black,red,green]):
> P := t ->
  plot(statevalf[pdf, studentst[t]](x), x=-6..6, colour=clr[1+(t mod 4)]);
> display([seq(P(5*t), t=1..10)], insequence=true);
```

6.31.4 Preben Alsholm (25.9.98)

The problem is that in studentst[t] the degree of freedom t has to be a positive integer. However, animate picks floats in the range you give for t. You could replace t with round(t), or you could instead create the frames first and then display them in sequence:

The second idea first:

```
L:=[seq(plot(statevalf[pdf, studentst[t]](x),
             x=-3..3),t=[2,5,25,100,500,1000])]:
display(L,insequence=true);
```

The first remedy:

```
animate(statevalf[pdf, studentst[round(t)]](x), x=-3..3,t=1..1000);
```

6.32 animate and evaluation of variables (9.7.96)

6.32.1 Mike May

I have started to play with Maple Version V4. I have been having some trouble with the animate command. There are some obvious nice animations for calculus where you string together approximations, either with sums, like rightbox, or with series, like taylor.

My problem is that when I try to use these with animate, I get an error indicating a parameter is a wrong type. (Taylor wants a natural number for degree. The parameters in animate seem to be automatically real.)

```
************sample code*****************

> with(plots):convert(taylor(sin(x),x=0, 8), polynom);
> plot({sin(x), convert(taylor(sin(x),x=0, 8), polynom)}, x=-3..6,
      y=-1..1);
> animate({sin(x), convert(taylor(sin(x),x=0, n), polynom)}, n=2..10,
         frames=9, view=[-3..5, -1..1]);
Error, wrong number (or type) of parameters in function taylor
```

6.32.2 Robert Israel (16.7.96)

Actually, the main problem is one of premature evaluation. The arguments to "animate", like those of nearly all functions, are evaluated before "animate" is called. Thus "taylor" is called with third argument the symbolic variable n. Unfortunately, you can't use the usual remedy of quoting the offending expression:

```
> animate({sin(x), 'convert(taylor(sin(x),x=0, n), polynom)'}, x = -3 .. 5,
          n = 2 .. 10, frames = 9);
```

because this will have the opposite problem: when "taylor" is called, n will be assigned a numerical value, but so will x. If you're determined to use "animate", you might do it as follows (note the use of "round" to make the n values into integers):

```
> animate({sin(x), 'subs(t=x,convert(taylor(sin(t),t=0, round(n)), polynom))'},
      x = -3 .. 5, n = 2 .. 10, frames = 9);
```

A simpler approach is to use "display(..., insequence=true)". Note that "seq" assigns values to the index variable before evaluating its first argument.

```
> display([ seq( plot({sin(x), convert(taylor(sin(x), x = 0, n), polynom)},
          x = -3 .. 5), n = 2 .. 10) ], insequence=true);
```

The second approach should also be more efficient, since you only call "taylor" once for each n.

6.32.3 Jan-Moritz Franosch (17.7.96)

I just found a solution but I still don't know why it must be so complicated:

```
with(plots);
p:=proc(x,n)
  local i,y,expr;
  i:=trunc(n);
  expr:=convert( taylor( sin(y),y=0,i ), polynom );
  RETURN(expr);
end;
animate( 'p(x,n)', x=-3..5, n=2..10, frames=9, view=[-3..5,-1..1] );
```

The quotes (`'p(x,n)'`) are necessary because animate seems to try to evaluate `p(x,n)` and this leads to the error you described because n is even no number.

Replacing x by y in the procedure is necessary because something like `taylor( sin(1.7),1.7=0,5 )` is not possible of course.

I think plot should be improved so it can be used more easy in such situations.

6.32.4 Dan Dubois (17.7.96)

The taylor() routine does expect a non-negative integer as a third argument. You may wish to plot the integer values between 2 and 8 and animate them using the display() routine:

```
    |\^/|     Maple V Release 4 (WM - Internal Use Only)
._|\|   |/|_. Copyright (c) 1981-1996 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
```

```
> with(plots):convert(taylor(sin(x),x=0, 8), polynom);
                                3          5          7
                      x - 1/6 x  + 1/120 x  - 1/5040 x


> myproc:=proc(n) convert(taylor(sin(x),x=0, n), polynom); end;
       myproc := proc(n) convert(taylor(sin(x), x = 0, n), polynom) end


> for i from 2 to 8 do
> p[i]:=plot({sin(x),myproc(i)},x=-3..5,y=-1..1):
> od:

> display([seq(p[j],j=2..8)],insequence=true);
[...]
```

6.33 animate: vertical lines and multiple colors (14.3.00)

6.33.1 Nathan Sokalski

I have been playing with the animate command, and have recently encountered a situation in which I need a vertical line (it will move from side to side, but remain vertical). Because there is no function of x that can give me this, I am forced to use a command. I have found the command line([a,b],[a,b]), which is used to draw a line segment, in the package plottools, but this did not appear to work with animations. How do I make a vertical line in an animation?

Most of my animations also contain more than 1 function. I do know about the option color=, but unlike the plot command, animate will only let you put one color there. I find it hard to believe that you must make everything the same color. Is there a way to use multiple colors like you do with the plot command?

Note: I use Maple V Release 5 Student Version

6.33.2 John Little (15.3.00)

About your questions concerning animations. 1) You want to plot your vertical lines as parametric curves. 2) You can generate several animations with the curves plotted in different colors, then show them together using the display command from the plots package. Here's a simple example showing both things:

```
with(plots):
L1:=animate([t,u,u=0..1],t=-2..2,color=blue):
L2:=animate([2-t,u,u=0..1],t=-2..2,color=red):
display({L1,L2});
```

6.33.3 Carl DeVore (15.3.00)

Using the "display" command with the argument "insequence= true" is a much more flexible way to produce animations. The "animate" command only works with functions. Any plot whatsoever can be animated with the "insequence" option, including plots containing text.

The vertical line x=a can be expressed as a function in polar coordinates as `r = a*sec(theta)`. You can then use the range of theta to control the height of the line segment. You can

include the argument "coords= polar" to the animate command.

There are many other ways to draw a line segment. You could use a parametric plot:

```
plot([a, t,  t= c..d])
```

will plot a vertical line segment from [a,c] to [a,d]. But this is using an elephant gun to swat a fly.

Other ways to draw the line segment from [a,b] to [c,d]:

```
plot( [[a,b],[c,d]] )
pointplot ( [[a,b],[c,d]], connect= true)
```

and my favorite (because it will avoid some function call overhead if you are putting many line segments in one display):

```
display(CURVES([[a,b],[c,d]]))
```

Note the capital letters.

Putting it all together, the following command will produce an animation of n+1 vertical line segments moving through the color spectrum:

```
display([seq(CURVES( [[k/n,0],[k/n,1]], COLOR(HUE,k/n)), k= 0..n)]
       ,thickness= 9, view= [0..1,0..1], insequence= true)
```

Using n=100, this command executes in 0.41 seconds on my SPARCstation 4, not a particularly fast computer; whereas doing the equivalent thing with the plot command takes over 5 secs; with line, 0.7 secs; and with pointplot, 1.27 secs.

There are many good examples of using the "insequence" option in _Calculus the Maple Way_ by Robert B Israel. In particular, note that each frame of the animation can itself be a complex structure created with the display command.

If anyone wants them, I will email them worksheets with animations of:

1. the osculating circle rolling along a 2D or 3D curve

2. Newton's method for functions of a single real variable. This one uses text in the animation to print

a. the equation of the tangent line at each iteration

b. the x-intercept at each iteration

c. context-sensitive advice on how improve the approximation.

6.33.4 Preben Alsholm (16.3.00)

To get a vertical line x=a, you could use a parametric representation, e.g. `[a, t, t=0..1]`.

Try the following:

```
f:= t -> 1-t^2:
p1:=plot(f,-1..1,color=blue):
p2:=animate([a,t*f(a),t=0..1],a=-1..1,color=red):
display(p1,p2);
```

6.33.5 Norbert Roth (16.3.00)

Here's a little 'dirty' procedure which does the trick.

```
restart;
with(plots):
pl:=proc(xa,ya,xe,ye,xstep,ystep,xytimes)
          local i,j;
          PLOT(ANIMATE(seq([CURVES([seq([evalf(xa+(i-1)*(xe-xa)+(j-1)*xstep),
                                         evalf(ya+(i-1)*(ye-ya)+(j-1)*ystep)],i=1..2)
                                    ],j=1..xytimes)))
        end:
```

Short description:

```
The procedure pl produces a plot structure.
The parameters are: xa,ya   - (coordinates of the beginning point of the line)
                    xe,xe   - (coordinates of the ending point of the line)
                    xstep   - difference between the x coordinates
                              of the actual point and next point to be drawn
                    ystep   - difference between the y coordinates
                              of the actual point and next point to be drawn
                    xytimes - how many animation frames
Example:
```

```
> gp:=pl(-.25,-1.5,.25,-1,Pi/6,5/12,6):
> display({gp},view=[-.25..3.55,-3..3]);
```

Displaying several animation in one window:

E.g. create normal animation via:

```
> fp:=animate(sin(x)*t*cos(x),x=0..Pi,t=1..5,frames=6,color=blue):
> display({gp,fp},color=green);
| your color question
```

Look at the above examples how to manage it, should be obvious.

Add. Info: The outer color definition used in 'display' is overwritten temporary in 'fp'.

Further and more detailed info in the manuals and help pages, e.g. `?plot[structure], ?plot[options]`
...

6.33.6 Stanley J Houghton (16.3.00)

I use the plot command with a list of coordinates (two in this example). Try as an example:

```
with(plots):
display([
   seq(
       plot([[x,0],[x,1]])
    ,x=1..100)
  ],insequence=true);
```

It plots a unit verticle line at various x positions.

6.33.7 Symancyk, Daniel (16.3.00)

You can use display of the plots package with `insequence=true`. Create a plot structure for each frame and specify the color you want in each frame.

The following will produce an animation with four frames. In the first frame we have the line x=1 in blue. In the second one x=1.5 is green, etc.

```
a[1]:=plot([[1,0],[1,1]],x=0..2,color=blue):
a[2]:=plot([[1.5,0],[1.5,1]],x=0..2,color=green):
```

```
a[3]:=plot([[1.75,0],[1.75,1]],x=0..2,color=red):
a[4]:=plot([[0.1,0],[0.1,1]],x=0..2,color=yellow):
s:=seq(a[i],i=1..4):
plots[display](s,insequence=true);
```

In the work above all of the vertical lines have the same height. If you want to vary the height using a function and you'd like the function to be displayed as well, then the following will do it.

h will give the height over interval [a,b]

```
> h:=x->x^2+1;
> a:=-1;b:=2;
```

there will be 26 frames

```
> n:=25;
```

the color of each vertical line is randomly determined

```
g:=(h,a,b,i)->plot([[a+i*((b-a)/n),0],[a+i*((b-a)/n),h(a+i*((b-a)/n))]],
     a..b,color=COLOR(RGB,rand()/10^12, rand()/10^12, rand()/10^12)):
c:=plot(h,a..b,color=black):
for j from 0 to 25 do
p[ j ]:=plots[display]({g(h,a,b, j),c}):
od:
s0:=seq(p[ j ], j=0..25):
plots[display](s0,insequence=true);
```

6.33.8 Douglas B. Meade (16.3.00)

If I understand your request, it is possible to create what you want using the display command from the plots package. The following sequence of commands illustrates the general idea by drawing the vertical height of a function at 21 uniformly spaced points in the interval [-1,1].

```
> restart;
> with( plots ):
>
```

```
> f := x -> x^2+1;

                                        2
                              f := x -> x  + 1

> P := plot( f, -1..1, color=BLUE ):
> P2 := seq(
         display( P,
                  plot( [[x,0],[x,f(x)]] ) ),
         x=[i/10 $ i=-10..10] ):
> display( P2, insequence=true );
```

I hope you can modify this for your needs,

6.34 animatecurve problem (24.10.98)

6.34.1 John Kurtzke

I was thrilled to see animatecurve in Release 5 of Maple V, but lately, I have had problems. I am using a Power Mac with System 7.5.3 Revision 2.

I have been trying to animate the path of a projectile (baseball) and what I thought was working before no longer works:

```
x := t -> t*v*cos(u);
y := t -> -16*t^2 + t*v*sin(u) + 3;
v := 105;
u := Pi/4;
plot([x(t), y(t), t=0..4.68]);
```

gives the correct plot.

(Why 4.68? That's when the ball hits the ground).

`animatecurve([x(t), y(t), t=0..4.68]);` also used to give the right animation until this past week. Defing `x & y` as expressions instead of functions also used to work. Now animatecurve gives a graph for `x & y` values far beyond the correct ones.

Moreover, `animatecurve( [ cos(t), sin(t), t=0..2*Pi]);` works once, at most. After that, it gives various erroneous plots (my favorite is one with a set of points on the unit circle and lines between point on the other side of the circle – it's a really neast picture that I could never do myself).

Writing down the `x & y` expressions directly in animatecurve sometimes works (but only once) and sometimes not.

This is a problem with my own personal copy of Maple (whether I am connected to the University network or not), with the network versions – both `Mac & PC`, and with a UNIX version installed on a Sun Sparc Station.

Yesterday, I stumbled across something that has worked thus far: With above notation,

```
animatecurve( [x, y, 0..4.68]);       # or, for the circle,
animatecurve([cos, sin, 0..2*Pi]);
```

I am glad I found sometihng that did work, but what I was doing before should have continued to work – certainly that's what the help page for animate curve suggests.

I have given Maple plenty of memory – 30 MB, so I do not think it is a memory issue.

6.34.2 Robert Israel (2.11.98)

The bug in animatecurve seems to be that it uses the global variables u and v for its own purposes. These should have been declared as local variables. If they have been assigned values, animatecurve may not work properly. So the easy work-around is to use some other variable names instead of u and v.

6.34.3 Preben Alsholm (2.11.98)

Here is another bug, or, if you will, inconsistency, in animatecurve. Compare the following:

```
with(plots):
animatecurve(cos(t),t=0..Pi,coords=polar);
animatecurve(arccos(r),r=-1..1,coords=polar);
animate(a*cos(t),t=0..Pi,a=0..1,coords=polar);
plot(cos(t),t=0..Pi,coords=polar);
```

Whereas plot and animate understand the expression (here `cos(t)`) as referring to r expressed in terms of theta, animatecurve takes the opposite view. The last three commands give circles, where the first does not.

There is no difference in interpretation when the parametric form is used: `[cos(t),t,t=0..Pi]`.

6.35 animated graphics (19.3.00)

6.35.1 Sandy Yates

I am using Maple 6 (SIX). I am trying to develop a visual example where an object by animating the rotation of a surface around a sphere that can be run below.

I cannot get the animation (through a sequence) to work.

Does anybody have any ideas how to get this to work? I am sure it worked one time!

```
restart: with(plots):with(plottools):
f:=(x,y)->(x^2-y^2)/5 -5;
n:=40:
a := plot3d( f(x,y), x=-5..5,y=-5..5 ):
c := sphere([0,0,0], 2):
display3d( [c,a], axes=NONE, light=[0,0,1,1,1], scaling=CONSTRAINED,
            orientation=[70,80], style=PATCHNOGRID);
display3d( seq( plot3d ( f(x,y), x=-5..5, y=-5..5, scaling=CONSTRAINED,
            axes=NONE, orientation=[60+i*720/n,70], shading=ZHUE ),
            i=0..n-1 ), insequence=true );
display3d( seq( plot3d( {f(x,y),c }, x=-5..5,y=-5..5, style=PATCHNOGRID,
            axes=NONE, light=[0,0,1,1,1], orientation=[60+i*720/n,70],
            shading=ZHUE ), i=0..n-1  ), insequence=true );
```

6.35.2 Helmut Kahovec(21.3.00)

You may try the following:

```
> restart;
> with(plots):
> with(plottools):
> f:=(x,y)->(x^2-y^2)/5-5:
> n:=40:
> a:=plot3d(
>   f(x,y),
>   x=-5..5,y=-5..5,
>   style=PATCHNOGRID,
>   shading=ZHUE
> ):
```

```
> c:=sphere([0,0,0],2):
> display3d(
>   [a,c],
>   orientation=[70,80],
>   axes=NONE,
>   scaling=CONSTRAINED,
>   light=[45,45,1,1,1]
> );

> s:=NULL:
> for i to n do
>   s:=s,display3d(
>     [a,c],
>     orientation=[70+(i-1)*720/n,80]
>   )
> od:
> display3d(
>   [s],
>   axes=NONE,
>   scaling=CONSTRAINED,
>   light=[45,45,1,1,1],
>   insequence=true
> );
```

6.35.3 Bill Bauldry(21.3.00)

I believe the problem is that plot3d doesn't understand sphere. You'll need to use display to combine the plot3d of f and the sphere; possibly by:

```
display(seq(display(c, plot3d(f(x, y), x=-5..5, y=-5..5);
```

Then you can use the animation on the combined sequence:

```
PF := (i,n)-> plot3d( f(x,y), x=-5..5,y=-5..5, style=PATCHNOGRID,
axes=NONE, light=[0,0,1,1,1], orientation=[60+i*720/n,70], shading=ZHUE):

display(seq(display(c, PF(i, n)), i=0..n-1), insequence=true);
```

6.35.4 Clinton Wolfe (21.3.00)

Try this as the last line:

```
>display3d( seq( display3d( [c,a], axes=NONE, light=[0,0,1,1,1],
            scaling=CONSTRAINED, orientation=[60+i*720/n,70],
            style=PATCHNOGRID),i=0..n-1), insequence=true );
```

(display returns a plot object, just like plot3d.)

6.35.5 Robert Israel (21.3.00)

Your c is a plot structure, not an algebraic expression in x and y, so you don't want to put it into the plot3d command. What you want is something like

```
> display(seq(display({c,plot3d(...)}), i=0..n-1), insequence=true);
```

6.36 animated phase portraits (27.8.99)

6.36.1 RJ Briggs

The code below creates animated phaseportraits for arbitrary, autonomous system of ODEs. Using the Lorenz equations as an example, I take the user through the creation of the animation so that he or she knows what they are really looking at when the code compiles and runs. There are two significant problems with the code that reduce the power of the program, but in the main it works.

Please see my statements of these problems, inserted in the code below. Note that code lines begin with '>' and comment lines do not.

```
> restart:
> with(plots):
> with(DEtools):
> 1+1:
> sys:= [diff(x(t),t) = -s*x(t)+s*y(t),
>        diff(y(t),t) = r*x(t)-y(t)-x(t)*z(t),
>        diff(z(t),t) = -b*z(t)+x(t)*y(t)];
> s:=10:r:=28:b:=8/3:
> inits:=[[x(0)=2,y(0)=3,z(0)=5]];
```

Programming Problem: Phaseportrait, the function in Maple that this worksheet will make the most use of, creates a two-dimensional phaseportrait for a given system of ODEs for a given set of initial conditions. You can give the program a list of initial conditions to create phaseportraits for and the program will draw the trajectories all on the same graph, but it is difficult and perhaps impossible to create a small procedure that generates a set of initial conditions for this function (maybe you can get it to work: see initialconditions or maybe DEtools. You might also try DEtools[phaseportrait]).

I would love to see a small procedure that could specify arbitrarily many initial conditions in an arbitrarily small neighborhood (or ball/cell)–it would greatly aid in the analysis of sensitive dependence on initial conditions for the system.

```
> Lxz:= proc(n);
>   phaseportrait(sys,[x(t),y(t),z(t)],
>   t=0..n,
>   inits,
>   stepsize=.005,
```

```
>   scene=[x(t),z(t)],
>   linecolor=t);
> end:
> Lxy:= proc(n);
>   phaseportrait(sys,[x(t),y(t),z(t)],
>   t=0..n,
>   inits,
>   stepsize=.005,
>   scene=[x(t),y(t)],
>   linecolor=t);
> end:
> lzlistxz:= [Lxz(1), Lxz(2), Lxz(3), Lxz(4), Lxz(5), Lxz(6), Lxz(7),
  Lxz(8), Lxz(9), Lxz(10),  Lxz(11), Lxz(12), Lxz(13), Lxz(14), Lxz(15),
  Lxz(16), Lxz(17), Lxz(18), Lxz(19), Lxz(20),  Lxz(21), Lxz(22), Lxz(23),
  Lxz(24), Lxz(25), Lxz(26), Lxz(27), Lxz(28), Lxz(29), Lxz(30)]:
> lzlistxy:= [Lxy(1), Lxy(2), Lxy(3), Lxy(4), Lxy(5), Lxy(6), Lxy(7),
  Lxy(8), Lxy(9), Lxy(10), Lxy(11), Lxy(12), Lxy(13), Lxy(14), Lxy(15),
  Lxy(16), Lxy(17), Lxy(18), Lxy(19), Lxy(20), Lxy(21), Lxy(22), Lxy(23),
  Lxy(24), Lxy(25), Lxy(26), Lxy(27), Lxy(28), Lxy(29), Lxy(30)]:
> display(lzlistxz, insequence=true);
> display(lzlistxy, insequence=true);
```

Programming Problem: As hard as I have tried, I can't get the display command to take a generated list (as opposed to a written list, such as lzlistxy above) of phaseportraits for input.

I'd like to be able to generate a list of phaseportraits to display based on a starting time t1 and an ending time t2 and a stepsize (e.g. if t1 = 1 and t2 =2 and the stepsize is 0.25, I'd like Maple to generate the phaseportraits Lxz(1), Lxz(1.25), Lxz(1.5), Lxz(1.75), and Lxz(2)).

The problem I have encountered is not getting the computer to generate such a list–the problem mainly seems to lie in the format of the elements of the list.

Instead of holding the elements in the list as actual plots, Maple holds them as instructions for plots. Unfortunately, the display command does not accept a list of instructions for plots as an input, so I'm at an impasse.

A small procedure that gets around this problem would greatly increase the usability of this program since it would give the user the ability to create arbitrarily smooth animation for an arbitrary length of time.

6.36.2 Joe Riel (27.8.99)

One solution, applicable elsewhere, is to have a procedure to generate an arbitrary linear sequence. Thus

```
`for` := proc(f,t,b)
local x,sq;
description `Compute a sequence of numbers from F to T by [B]|1.`;
    sq := NULL;
    for x from f to t by `if`(2 < nargs, b, 1) do
        sq := sq,x
    od;
    sq
end: # `for`
```

You can then apply it to your problem in the following manner

```
lzlistxz := [seq(Lxz(xx), xx=`for`(1,2,0.2))]:
display(lzlistxz, insequence=true);
```

6.36.3 Robert Israel (29.8.99)

I really can't see what you find difficult here. Both problems seem quite straightforward to me.

First, here's a procedure to randomly generate n initial conditions within epsilon (in each coordinate) of `[x0,y0,z0]`:

```
makeinits:= proc(x0,y0,z0,n,epsilon)
   local i, rnd;
   rnd:= random(-100 .. 100);
   [ seq([x(0)=x0+rnd()*epsilon/100, y(0)=y0+rnd()*epsilon/100,
          z(0)=z0+rnd()*epsilon/100], i=1..n)];
   end;
```

So to produce such a set of initial conditions, you can say e.g.

```
inits:= makeinits(3,4,5, 10, 0.1);
```

And to generate a list of phaseportraits, all you need is something like this:

```
lzlistxz:= [seq(Lxz(i), i=1..30)]:
display(lzlistxz, insequence=true);
```

6.36.4 Preben Alsholm (1.9.99)

You can avoid having to type that long list by using a construction like this

```
lzlistxz:=eval( [ seq( 'Lxz'(k), k=1..30) ] );
```

It seems to me that a lot of unnecessary computation is going on, though. Here is a quick attempt at handling the problem differently. The idea is to compute only Lxz(30), and then make an animation out of that plot structure.

The procedure animatePLOT below is made to handle a 2-dimensional PLOT structure containing only one curve. It calls one of two subprocedures.

One of these (animatePLOT1) handles a PLOT structure containing one list of points (the typical output from plotting one curve), the other (animatePLOT2) handles a PLOT structure containing a sequence of lists of points, which is the output from phaseportrait, when linecolor has been specified.

The PLOT structure that can be input to animatePLOT2 must be of the form `PLOT(CURVES(...),..)`. The syntax is

`animatePLOT( p, n );`

where p is the PLOT structure and the optional n is the number of frames (default=number of points in the list (or members of the sequence)).

```
> restart;
> animatePLOT:=proc(p::PLOT)
> if nops(select(has,[op(p)],CURVES))>1 then ERROR("More than one curve") fi;
> if nops(select(type,{op(op(1,p))},list))>1 then animatePLOT2(args) else
    animatePLOT1(args) fi
> end:
>
> animatePLOT2:=proc(p) local n,p1,k,L,N,m,S,S1,S2,LC;
> L:=select(type,[op(op(1,p))],listlist);
> S:=remove(type,[op(op(1,p))],listlist);
> S1:=remove(has,S,{COLOR,COLOUR});
> S2:=op(select(has,S,{COLOR,COLOUR}));
```

```
> LC:=[op(S2)];
> N:=nops(L);
> if nargs>1 then n:=args[-1] else n:=N fi;
> m:=floor(N/n);
> for k from 1 to n do
> p1[k]:=subsop(1=CURVES(op(L[1..k*m]),COLOR(op(LC[1..k*m+1])),op(S1)),p);
> od;
> plots[display]([seq(p1[k],k=1..n)],insequence=true)
> end:
>
> animatePLOT1:=proc(p) local n,p1,k,L,N,m;
> L:=op([1,1],p);
> N:=nops(L);
> if nargs>1 then n:=args[-1] else n:=N fi;
> m:=floor(N/n);
> for k from 1 to n do
> p1[k]:=subsop([1,1]=L[1..k*m],p);
> od;
> plots[display]([seq(p1[k],k=1..n)],insequence=true)
> end:

> p:=plot(sin,0..2*Pi):animatePLOT(p);
> p:=plot([sin,cos,sin*cos,sin+cos],0..2*Pi):
> animatePLOT(p);
> p:=plot(sin,0..2*Pi,style=point):
> animatePLOT(p);
> with(DEtools):
> sys:=[diff(x(t),t)=-s*x(t)+s*y(t),diff(y(t),t)=r*x(t)-y(t)-x(t)*z(t),
        diff(z(t),t)=-b*z(t)+x(t)*y(t)];
> s:=10:r:=28:b:=8/3:
> inits:=[[x(0)=2,y(0)=3,z(0)=5]];
> p:=phaseportrait(sys,[x(t),y(t),z(t)],t=0..30,inits,stepsize=.005,
                   scene=[x(t),z(t)],linecolor=t):
> animatePLOT(p,30);
```

Needless to say, this procedure is rather raw and needs improvement. I'm aware of animatecurve, but that seems to use the same number of points for each frame, which doesn't produce good pictures at the end unless numpoints is set to a high value.

6.37 animating [display3d] structures (26.7.95)

6.37.1 Kevin Dowling

I am having trouble animating a series of plots[display3d] results and wonder if someone has a solution to this. I believe the problem is may be that animates is unable to animate a list of lists of display3d structures, but am not sure. Here's the sequence of things that should illustrate the problem:

This plots a sphere and translates it by 10:

```
>sphereplot := a -> plot3d(rot(a,
  [sin(p)*cos(t), sin(p)*sin(t), 10 - cos(p)]),
  p = 0..Pi, t = 0..2*Pi, style = patch, scaling = constrained);

       [Rot is a simple function I wrote that takes an input angle,
       in degrees and a vector and rotates the object about the
       origin by a.]
```

This plots a 'necklace' of spheres:

```
>plots[display3d]({seq(sphereplot(Pi*30*post/180), post = 1..12)});
```

Now, I want to animate this to produce a spinning 'necklace' to make it look like a set of orbiting spheres. I have tried many things, here are some related successes:

```
       This displays, each sphere in the necklace, one at a time, but
       in a very strange order using the insequence option to
       display3d:

>plots[display3d]({seq(sphereplot(Pi*30*pos/180), pos = 1..12)},
insequence = true);
```

```
       This animates a single sphere orbiting the periphery of a
       circle:

>plots[animate3d]((rot(Pi*30*star/180,
  [sin(p)*cos(t), sin(p)*sin(te), 10 - cos(p)]), p = 0..Pi, t = 0..2*Pi),
  star = 1..12, frames = 20, scaling = constrained);
```

```
        But I cannot animate the full necklace that is produced by the
        plots[display3d] command that generates the necklace. For
        example, this doesn't work:

>plots[display3d](rollerplot(Pi*30*posi/180), posi = 1..12, insequence = true);
Error, (in plot3d/options3d) unknown or bad optional argument, posi = 1 .. 12
```

```
        Another approach I tried was to individually generate each
        intermediate position (five will suffice) and then provide a
        list to plots[animate3d], like so:

>plotframe1 := plots[display3d]({seq(sphereplot(Pi*30*pos/180), pos = 1..12)}):

        ...and so on for each postion with a different argument to
        sphereplot, this gives five 3d structures. Then I tried this
        but this didn't work either:

>plots[animate3d][plotframe1, plotframe2, plotframe3, plotframe4, plotframe5];
```

If somemone can see the fundamental flaw in trying to animate [display3d] outputs, please let me know - or better, if you know of a solution!

6.37.2 Robert Israel

You didn't include your definition of "rot", so I don't know how you want to rotate it. By the way, you shouldn't use the name "sphereplot" which is a command in the "plots" package.

```
|>      This displays, each sphere in the necklace, one at a time, but
|>      in a very strange order using the insequence option to
|>      display3d:

|> >plots[display3d]({seq(sphereplot(Pi*30*pos/180), pos = 1..12)},
|> insequence = true);
It's in a strange order because you use a set rather than a list, and Maple can put se

>plots[display3d]([seq(sphereplot(Pi*30*pos/180), pos = 1..12)],
```

```
insequence = true);

|> But I cannot animate the full necklace that is produced by the
|> plots[display3d] command that generates the necklace. For
|> example, this doesn't work:

|> >plots[display3d](rollerplot(Pi*30*posi/180), posi = 1..12, insequence = true);
|> Error, (in plot3d/options3d) unknown or bad optional argument, posi = 1 .. 12
```

You didn't say what "rollerplot" is. Is "posi=1..12" supposed to be an argument to "rollerplot"? Or should there be another "seq" in there? As it is, you have "posi=1..12" as an argument to "display3d", which of course doesn't know what it's supposed to do with that.

```
|> Another approach I tried was to individually generate each
|> intermediate position (five will suffice) and then provide a
|>      list to plots[animate3d], like so:

|>plotframe1 := plots[display3d]({seq(sphereplot(Pi*30*pos/180), pos = 1..12)}):

|>      ...and so on for each postion with a different argument to
|>      sphereplot, this gives five 3d structures. Then I tried this
|>      but this didn't work either:

|>plots[animate3d][plotframe1, plotframe2, plotframe3, plotframe4, plotframe5];
```

Now I'm really confused! You're not even putting the list in parentheses? And why did you switch to "animate3d"? That must be called as `animate3d(F, x=x1..x2, y=y1..y2,t=t1..t2)` where F is an expression (or set of expressions) in variables x,y and t. What would work is

```
> plots[display3d]([plotframe1, plotframe2, plotframe3, plotframe4, plotframe5],
    insequence=true);
```

Of course you could define a command to generate the frames, and do something like

```
> plots[display3d]([ seq(plotframe(k), k=1..5) ], insequence=true);

|> If somemone can see the fundamental flaw in trying to animate
```

```
|> [display3d] outputs, please let me know - or better, if you know of a
|> solution! Thank you.
```

There's no fundamental flaw, it's just a matter of being careful with your ()'s, []'s and {}'s.

6.37.3 MR BOB G COLLINGS

For this statement to animate in sequence you need a list rather than a set. i.e.

```
>plots[display3d]([seq(sphereplot(Pi*30*pos/180), pos = 1..12)],
insequence = true);
```

Or on this one try

```
>plots[display3d]([plotframe1, plotframe2, plotframe3, plotframe4,
plotframe5],insequence=true);
```

6.38 animating a zoom process (21.9.99)

6.38.1 Richard Pulskamp

We cannot achieve a desired effect in an animation. Does anyone have a workaround? We are running Release 5.1. An example of the code used follows.

The particular difficulties are described later.

Here is an animation we want to use in Calculus I to mimic repeated zooming in on the graph of a function together with its tangent line:

```
f:=x->x^3+2*x-1:
with(plots):
A:=array(1..8):
for i from 1 to 8 do
plot1[i]:=plot(f(x),x=1-.5^i..1+.5^i,axes=box,color=red):
plot2[i]:=plot(5*x-3,x=1-.5^i..1+.5^i,axes=box,color=blue):
plot3[i]:=pointplot([[1,2]],style=point,symbol=circle):
A[i]:=display([plot1[i],plot2[i],plot3[i]]):
od:
display([seq(A[k],k=1..8)],insequence=true);
```

The above animation does show the frames which create the "zooming- in-effect" we want, but we do not get the desired effect because Maple does not enlarge the individual frames. In other words, it would appear that the graph dimensions cannot be dynamically resized.

DIFFICULTY I :We would like to produce an animation of the following pictures (exactly as shown) in an animation where the scale on the axes changes (i.e. new scales on both axes from frame to frame).

```
> display(%);
```

The `Plot[Structure] option "FRAMESCALING"` seems to have been designed to do this job. Experimentation with this option showed that this option is not functioning properly, as the following example should demonstrates:

```
for i from 1 to 5 do
plot1[i]:=plot(sin(x),x=0..i*Pi,axes=box):
od:
```

```
display(seq(plot1[i],i=1..5),insequence=true,framescaling=nonuniform);
```

DIFFICULTY 2: Since we would like to post these animations for our students, we want to preset the frame-rate, i.e. the number of FRAMES PER SECOND in the animation command rather than this having to be done by the viewer. How can this be done?

This does not seem to be an option.

6.38.2 Rexford Cordill (29.10.99)

Zooming in on a 2D or 3D plot in MapleV is a feature that should be well documented, but to my knowledge it isn't. I found a very good solution in the DEtools[zoom] command. The MapleV help page on DEtools[zoom] is very good and you shouldn't have any trouble implementing it. Just implement this command in your 'for' loop followed by

```
plots[display]([seq(plot1[j],j=1..5)],insequence=true);
```

You can fake varying the frame rate by merely duplicating plot structures in the list of plot structures passed to plots[display]. For example:

Instead of

```
plots[display]([plot1[1],plot1[2],plot1[3],plot1[4]],insequence=true);
```

You could write

```
plots[display]([plot1[1],plot1[1],plot1[2],plot1[2],plot1[3],plot1[3],plot1[4],plot1[4
                insequence=true);
```

Duplicating the plots in the list would appear to halve the frame rate. Note that this basic technique could be used to display part of the animation at a different rate than other parts.

6.38.3 Bernd Rossa (29.10.99)

A few days ago you were kind to respond to a message which my colleague Dick Pulskamp had forwarded to MUG concerning the animation of a zooming process in Maple V release 5.1.

You pointed out that DEtools contains the zoom-command, which is, of course, true. This command seems to have been designed for enlarging a segment of a plot structure without having Maple recalculate the structure (from scratch).

This command speeds up the process of seeting up the frames needed to produce the desired "zooming movie".

However, as before (when I constructed the frames without the zoom command) my problem occurs when I want to display the frames `"insequence=true"`.

In the animation, Maple refuses to adjust the scale on the window! Instead, the portion of the graph shrinks inside of the largest (first) window, rather than the scales on the window changing which would create the desired zooming (i.e. cranking up a microscope) effect.

Please look at the following example (which uses "zoom"):

```
> with(plots):with(DEtools):
```

Define a function:

```
> f:=x->x^3+2*x-1:
```

Define a plot-structure to zoom in on:

```
> A:=plot(f(x),x=0.5..1.5,y=0..5,axes=box,color=red):
> display(A);
```

Define the frames which zoom in on (1,2):

```
> for i from 1 to 8 do
> plot1[i]:=zoom(A,1=1-.5^i..1+0.5^i,2=2-5*.5^i..2+5*.5^i):
> od:
```

Finally display the frames "insequence=true":

```
> display([A,seq(plot1[i],i=1..8)],insequence=true);
```

Notice that the portion of the graph shrinks. No zooming effect! But if you look at each frame separately, the zooming occurred. Maple does not change the scale on the individual windows. That's my problem. Any further ideas? (Am I being stupid?) :-).

6.38.4 Robert Israel (3.11.99)

```
| DIFFICULTY I :We would like to produce an animation ...
```

You might try something like this:

```
> with(plots): with(plottools):
> maketicks:= proc(a,b,s,c)
  # get list of tickmarks for plot scaled by scale factor s
  # about centre c, with unscaled bounds a .. b
  local xt, r,j;
  r:= b-a;
  xt:= 10^ilog10(r);
  if r < 1.5*xt then xt:= xt/5
  elif r < 4*xt then xt:= xt/2
  fi;
  [seq(c+j*s*xt = c+j*xt, j = floor((a-c)/xt) .. ceil((b-c)/xt))];
  end;

> for i from 1 to 30 do
    p1:= scale(plot(f(x),x= 1 - 1.1^(-i) .. 1 + 1.1^(-i), axes=box,
      colour=red), 1.1^(i-1), 1.1^(i-1),[1,2]):
    p2[i]:= display(p1, labels=[x,y],
      xtickmarks = maketicks(1-1.1^(-i),1+1.1^(-i),1.1^(i-1),1),
      ytickmarks = maketicks(2-6*1.1^(-i),2+6*1.1^(-i),1.1^(i-1),2));
    od:
> display([seq(p2[i],i=1..30)],insequence=true);
```

6.39 animating plots of taylor polynomials (28.2.02)

6.39.1 John Losse

In attempting to show students Taylor polynomials of successive degrees, we tried defining the first few via the loop

```
for k from 1 to 8 by 1 do
 tp[k]:=convert(taylor(f,x=0,k+1),polynom)
end do;
```

This worked.

Next we hoped to plot the polynomials one after the other in the same window, but the animate command

```
animate(tp[s],x=-10..10,s=0..8, view=[-10..10,-3..3],frames = 9);
```

produces an empty plot.

We've found a very elaborate workaround but it seems the basic idea - step from one member of an indexed list to another - ought to work.

6.39.2 Carl Devore (1.3.02)

Animate requires s to be a real parameter, not integer. Use instead

```
plots[display]
   ([seq(plot(tp[s], x= -10..10, -3..3), s= 0..8)]
   ,insequence= true
   );
```

or, in Maple 7, the more elegant syntax

```
plots[display]
   ([seq](plot(tp[s], x= -10..10, -3..3), s= 0..8)
   ,insequence= true
   );
```

While I applaud your efforts to show your students the mechanics of creating these animations, you may be interested in also showing them my worksheet "Animation of Taylor series conv available at the Maple Applications Center http://www.mapleapps.com.

This produces labelled animations several orders of magnitude faster than your example above, handles discontinuities, and most of the full range of plot options.

For example

```
f:= x-> exp(cos(x))*tan(x)+cos(2*x)+sin(Pi*x);
TaylorAnim(f, Pi, 0..2*Pi, 50, -10..10, discont= true);
```

animates the complicated, discontinuous function f and its first 50 Taylor polynomials expanded at x=Pi with view window [0..2*Pi, -10..10] using 1.3 seconds and neglible memory on a Pentium II 400 MHz.

The comments in my code give several pointers on how to produce these high-speed animations.

6.39.3 Douglas B. Meade (1.3.01)

Animate works best when the animation variable is continuous. It is possible to achieve this for your situation but it's not intuitive or simple.

I prefer to create a list of plots and then animate these with the display command. For example,

```
> f := 2*cos(x);
>
> for k from 1 to 8 by 1 do
>   tp[k]:=convert(taylor(f,x=0,k+1),polynom)
> end do;
>
> P := [seq( plot([f,tp[k]],x=-10..10,y=-3..3), k=1..8 )]:
> display( P, insequence=true );
```

6.39.4 Robert Israel (3.3.02)

```
plots[display](
   [seq(
      plot(tp[k], x=-10..10, title=sprintf("degree %d",k)),
      k=1..8)],
   insequence=true, view=[-10..10,-3..3]);
```

6.39.5 Alan Gold (4.3.02)

The animate command is not very versatile. I almost never use it. Instead, create plot structures for each of your functions:

```
for k from 1 by 1 to 8 do
 pl[k]:=plot(tp[k],x=-10..10,view=[-10..10,-3..3]):
od:
```

Then use display with the option insequence=true

```
with(plots):
display([seq(pl[k],k=1..8)],insequence=true);
```

You might want to look at "Animations for the Calculus Classroom" in the Maple Applications Centre for some further examples.

6.40 animation of drawing a function point-by-point (31.10.02)

6.40.1 Sandy Yates

I am not very experienced with Maple (v7). I am trying to ANIMATE drawing an (x, f(x)) function where for each iteration I would like to draw more of the function, i.e, incrementing x for each step animation step.

For example, I would like to plot

```
f:= t -> exp(-t/100)*cos(t);
plot(f(t),t=0..600,numpoints=400);
```

so that each stage of the animation more points are included. I would hope that the plot gives the impression that data points are being acquired and plotted with time. This is for some illustrations I have in mind.

6.40.2 Robert Israel (1.11.02)

```
with(plots):
display([seq(plot(f(t),t=0..20*j, numpoints=3*j), j=1..30)],
        insequence=true);
```

6.40.3 Dr Francis J. Wright (1.11.02)

I discuss this type of animation in my book "Computing with Maple"; see `http://centaur.maths.qmul.ac.uk/CwM/` for details.

I suggest you take a look at the cardioid animation under "Graphics examples" on that web page. If that does essentially what you want, then you might like to download the solutions to the exercises for Chapter 3, which are available as a Maple 6 worksheet under "Exercise solutions".

(Of course, I would also encourage you to buy the book!)

This worksheet also runs in Maple 7. (In fact, I have just fixed a problem in it caused by the fact that Maple 7 will not accept a zero-length parametric curve, whereas Maple 6 would.)

6.40.4 Allan Wittkopf (1.11.02)

The following is quite short and will give you what you need:

```
f:= t -> exp(-t/100)*cos(t);
nf := 11:
plots[display]([
   seq(
      plot(f(t),t=0..(i-1)/(nf-1)*600+1e-15,
           numpoints=max(2,trunc(1+(i-1)/(nf-1)*400))),
       i=1..nf)
              ],insequence=true);
```

In the above 'nf' is the number of frames, and can be adjusted to suit - the 600 is the last time value, and the 400 is the number of points in the last frame - notice, they are defined as ratios of the final values in terms of `'(i-1)/(nf-1)'`.

This can be made more efficient, but it is much less elegant.

6.40.5 Carl Devore (2.11.02)

```
plots[animate](
   `if`(t<x, f(t), undefined)
  ,t= 0..600
  ,x= 0..600
  ,numpoints= 400
  ,frames= 100
);
```

Of course, you can change the ranges and the number of frames to whatever you want.

Here is another solution. This one works much, much faster and gives a smoother animation, but it is more difficult to understand for a new user.

```
TracePlot:= proc(P::specfunc(anything, PLOT))
   local A,C,C0,Cn,Pn,np,chunk,fr,f;
   fr:= 16;  # Defaukt number of frames
   hasoption([args[2..-1]], 'frames', 'fr');
   C:= op(1,P);
   C0:= op(0,C);
```

```
   A:= op(1,C);
   Cn:= op(2..-1,C);
   Pn:= op(2..-1,P);
   np:= nops(A);
   chunk:= np/fr;
   PLOT(ANIMATE(seq([CO(A[1..trunc(chunk*f)], Cn), Pn], f= 0..fr)))
end proc:
And call it as:

TracePlot(
   plot(exp(-t/100)*cos(t), t= 0..100, numpoints= 400)
  ,frames= 100
);
```

6.40.6 Koch-Beuttenmueller (4.11.02)

May-be animatecurve(....) is the function you need ? (after with(plots))

6.41 animation orbit structure (11.4.97)

6.41.1 Aldrovando Azeredo Araujo

I am very new to Maple and maybe my problem is easily solved. Anyway, I am working with differential equations with a geometrical approach. So I am more interested in analysing the orbit structure of differential equations.

Is there a plot procedure that gives me the animated orbit of some initial condition in the space, like following some orbit of Lorentz's equation into the attractor? For example like the "comet" function in MatLab?

6.41.2 Robert Israel (16.4.97)

I don't know about "comet", but here's one thing you can do. This requires Release 4 (but there are ways to get similar effects in Release 3).

```
> with(plots): with(linalg):
> lorenz:=  D(x)(t) = sigma * (y - x), D(y)(t) = rho * x - y - x*z,
                  D(z)(t) = - beta*z + x*y;
> sigma:= 10; rho:= 28; beta:= 8/3;
> ics:= x(0) = 1, y(0) = 1, z(0) = 1 ;
> vals:= dsolve({lorenz, ics}, [x(t), y(t), z(t)], numeric,
      value=array([seq(.01*n, n=0 .. 2999)]), method=classical[rk4]):
# this calculates 3000 points on the trajectory: may take a while
> vmatrix:= delcols(vals[2,1],1..1):
> vlist:= convert(vmatrix, listlist):
# vlist is a list of 3000 points
> traj:= spacecurve(vlist):
# this is the trajectory, plotted as a curve
> for j from 1 to 20 do
      frame[j]:= pointplot3d([ seq(vlist[20*k+j], k = 0 .. 149)]) od:
# this produces 20 frames, each consisting of a plot of 150 points on the
# trajectory
> display({ traj,
                display([seq(frame[j],j=1..20)], insequence=true,
                        scaling=constrained)},orientation=[80,80]);
# this produces the animation.  Click on the plot to get "VCR controls" on
# the Context Bar.  Click the Loop button (the one that looks like an oval with
# an arrowhead), and Play (the triangle pointing right).
```

6.41.3 faia michael a (17.4.97)

There is a section of Nicolaides-Walkington that does this. Incidentally, I believe Lorenz is spelled without a t. N-W make the same error.

6.42 animation problem in maple 6 and maple 7 for unix/linux (18.11.02)

6.42.1 Bertfried Fauser

While in mapleV the following works and produces the expected 'movie' – no wizardry heer :)

```
> N:=10:
> anim:=array(1..N):
> for i from 1 to N do
> k:=i/N:
> P1[i]:=PLOT(POINTS([k,1-k]));
> P2[i]:=PLOT(POINTS(1/2,1/2]));
> anim[i]:=display([P1[i],P2[i]]);
> od:
> display([seq(anim[i],i=1..N)],insequence=true,view=[-2..2,-2..2]);
```

it does not in maple 6! (Only the last frame occure, all others are empty)

6.42.2 Thomas Richard(20.11.02)

This was a known in Maple 6/7 for UNIX/Linux. A workaround is to issue

`plotsetup(gdi):`

before calling `display(...)`.

To switch back to the OpenGL driver, use `plotsetup(default)`:

In Maple 8 this bug has been fixed.

6.43 animation problem (15.5.00)

6.43.1 Aldrovando Azeredo Araujo

I've listed the code of a simple program I've been dealing with. I am trying to show my students simple models like the one below that treats chasing situations. Here we have a dog that chases a jogger. The first problem deal with the right time I should stop computations since when the dog catches up the jogger we have a singularity in the differential equation.

So I would like to know the time the dog reach the jogger to plot the two trajectories using insequence=true. I want to put all this comands in a procedure. How do I do that?

As anyone can see I had to make a sequence of plots that take a lot of time to be generated. Is it possible to get this animation faster? I can do that in matlab quite fast.

```
> restart:
> with(plots):
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
> x0:=2: y0:=1:
>
>
> K:=sqrt((X(t)-x(t))^2+(Y(t)-y(t))^2);

                                    2                2
             K := sqrt((X(t) - x(t))  + (Y(t) - y(t)) )

> eq1:=diff(x(t),t)=(v/K)*(X(t)-x(t));

                d                      v (X(t) - x(t))
       eq1 := -- x(t) = ------------------------------------
                dt                            2                2
                          sqrt((X(t) - x(t))  + (Y(t) - y(t)) )

> eq2:=diff(y(t),t)=(v/K)*(Y(t)-y(t));
```

```
                d                        v (Y(t) - y(t))
       eq2 := -- y(t) = ------------------------------------
                dt                              2                2
                          sqrt((X(t) - x(t))  + (Y(t) - y(t)) )

> X:=t->(t+1)*sin(2.5*t);

                        X := t -> (t + 1) sin(2.5 t)

> Y:=t->(t+1)*cos(2.5*t);

                        Y := t -> (t + 1) cos(2.5 t)

> v:=4.6*t:
> var:={x(t),y(t)}:
> init:={x(0)=x0,y(0)=y0}:
> sys:={eq1,eq2}:
> Sol:=dsolve(sys union init, var, numeric, output=listprocedure);

  Sol := [t = (proc(t)  ...  end), x(t) = (proc(t)  ...  end),

         y(t) = (proc(t)  ...  end)]

> xt:=subs(Sol,x(t));

                          xt := proc(t)  ...  end

> yt:=subs(Sol,y(t)):
> Tdog[0]:=[x0,y0]:Tman[0]:=[X(0),Y(0)]:
> xt(1.98);solve(r*0.05=1.98);

                             -2.897729591544956


                               39.60000000

> for i from 1 by 1 to 39 do
    Tdog[i]:=Tdog[i-1],[xt(i*0.05),yt(i*0.05)]:
    Tman_dog[i]:=plots[display]({plot([Tdog[i]],color=blue),
```

```
                                    plot([X(t),Y(t),t=0..i*0.05],color=red)}):
  od:
> display([seq(Tman_dog[i],i=1..39)],insequence=true);
```

6.43.2 Willard, Daniel Dr (17.5.00)

There is a book on differential gaming that you might wish to read: "Advances in Missile Guidance" purchased from AIAA Publications, 9 Jay Gould Court, Waldorf, MD 20602. It treats such problems and also ones where the dog tries to escape, and gives some Maple programs to go along with it.

6.43.3 John Reinmann (19.5.00)

In answer to your question "how can I find the time `(T_catchup)` when the dog catches up with the jogger?", I propose the following approach:

When you obtain the numerical solutions for xt and yt, you find that when the time exceeds the `T_catchup`, Maple gives the error messages `"sqrt of a negative number"` or "division by zero".

Thus, you can use this error message to signal when you have overestimated the value of `T_catchup`. You use Maple's "traperror" command.

Taking both error messages into account and improving the way I change "del" in the "for" loop, I now recommend the following Maple code to determine `q = T_catchup` (where, q denotes the time):

```
>q:=-1:
  del:=1:
  A:=`sqrt of a negative number`:
  B:=`division by zero`:
  test:=false:
  for i from 1 to n do
     while test=false do
        q:=q+del:
        result:=traperror(yt(q)):  # xt(q) could be used instead of yt(q)
        test:=evalb((result=A) or (result=B)):
     od:
     q:=q-del:
     del:=del/10:
```

```
    test:=false:
  od:
  q;
```

The result depends on the number of Digits used. The following results are for Digits=12.

```
for n=4, q=1989331/1000000

for n=5, q=19833319891/10000000000
```

In either case, if you increase the last digit in the numerator by 1, you get the error message above; so these results are very good.

The test I used to see if the value of `T_catchup` was good, was to evaluate the expression for the distance between the jogger and the dog, which is

```
distance = sqrt((X(q)-xt(q))^2+(Y(q)-yt(q))^2)

for n=4, distance = 0.16594...e-5

for n=5, distance = 0.63480...e-7

For n=11, the above code gives q=19893319961/100000000000 = 1.9893319961
```

When I plotted distance versus time, I got the interesting result that at first the distance closed and then opened up for a while and finally the distance closed again and the dog caught up with the jogger.

For an animation that can be generated quickly, I suggest using "pointplot", which can show the position of the dog and the jogger simultaneously as symbols (say a blue circle for the jogger and a red diamond for the dog).

Actually, this is a lot of fun to observe. Here's the code:

let q be the time at which the dog catches up with the jogger, and xt(t), yt(t), X(t), Y(t) be the same as in your code.

```
>display(seq(display({pointplot([X(q*j/40.0001),Y(q*j/40.0001)],color=blue,
         symbol=circle),pointplot([xt(q*j/40.0001),yt(q*j/40.0001)],color=red,
         symbol=diamond)}),j=0..40),insequence=true,title=`red-dog blue-jogger`,
         scaling=constrained);
```

I used `q*j/40.0001` rather than `q*j/40` to avoid the possibility of roundoff errors causing `q*40/40` to slightly exceed q, in which case you would get an error message.

6.44 animation to web-postable format, to convert (6.4.99)

6.44.1 Joel A. Shapiro

I have a Unix (Sun) machine with Maple V Rel 4.00h. I am interested in making animations similar in nature to "Filling the Area" in `http://www.maplesoft.com/www/apps/sinec1.html`, which is a gif. I can generate what I want in Maple with instructions just like those in that web page, but I don't know how to save an animation or how to convert it into a gif. `?devices` says gif is obsolete and no longer supplied with Maple V.

The objects I want to show are line drawings and flat-filled surface faces, so gif would appear a suitable format.

6.44.2 Philip Yasskin (9.4.99)

Try Release 5. gif's are working again. You can do a `plotsetup(gif, plotoutput="myfile.gif");` before the plot and a `plotsetup(default);` after the plot.

This works for static plots and animations. This works on Win and Unix but not on Mac.

On Mac you need to export the worksheet as HTML and find the gif file in the output. (This also works for Win and Unix.)

6.44.3 Robert Israel (9.4.99)

It's simple in Release 5, where "Export to HTML" will do the trick (the animations in a worksheet become animated GIF's). In Release 4, what you can do is save the individual frames as JPEG files, then combine these into an animated GIF using some other software. There are shareware programs on various platforms that will do this, e.g. you might try ImageMagick at `http://www.wizards.dupont.com/cristy/ImageMagick.html`.

6.45 animation, side by side, maple 6 (27.6.01)

6.45.1 Bernd Rossa

A couple years ago, when we were using release 5.1 I wrote a few side by side animations like the following:

```
> for i from 0 to N do
> plott[i]:=plot(...
> plottt[i]:=plot(...
> plotttt[i]:=pointplot(...
> od:

> for i from 0 to N do
> movie1[i]:=display({plott[i],plottt[i],plotttt[i]}):
> od:
```

And then the same again, with slightly changed content.

```
> for i from 0 to N do
> plott[i]:=plot(...
> plottt[i]:=plot(...
> plotttt[i]:=pointplot(...
> od:

> for i from 1 to N do
> movie2[i]:=display({plott[i],plottt[i],plotttt[i]}):
> od:

> A:=display([seq(cart[l],l=1..N)],insequence=true):
> B:=display([seq(polarcord[l],l=1..N)],insequence=true):
> display(array([A,B]));
```

This code worked fine with release 5.1, and in release 6.1 I get the error message:

```
Plotting error, object options take the form CLASS(VALUE)
```

Can anyone figure aut what's going on, and how to get around the new problem?

Note: It is importent that each frame is actually a display of several plots. Otherwise a straight

```
display(array([plot(...),plot(...)]));
```

works fine (in both releases).

I am desparate for help, since I have used Maple to construct little "videos" of what's in my head for my students, and it has worked well.

6.45.2 Robert Israel (28.6.01)

This bug seems to be fixed in Maple 7.

A workaround seems to be to substitute ALIGNLEFT for DEFAULT before the final display. Thus:

```
> display(subs(DEFAULT=ALIGNLEFT,array([A,B])));
```

6.46 animations (12.5.99)

6.46.1 Peter Weigand

I've tried an animation, that illuminates an gray tubeplot with several userlights. The following command lines works fine in Release 4.0, but it doesn't work in Release 5.0. What's the reason for this?

```
> with(plots):
> knot:=[ -10*cos(t) - 2*cos(5*t) + 15*sin(2*t),  -15*cos(2*t) + 10*sin(t)
          - sin(5*t), 10*cos(3*t) ];
> ori:=[175,55];rad:=4;
> setoptions3d(orientation=ori, scaling=constrained, style=patch, axes=none);
> l1:=[seq([90,45,0,evalf(i/20,2),1], i=0..20)]:
> l2:=[seq([90,45,0,evalf(1-i/20,2),1], i=0..20)]:
> lighters:=[op(l1),op(l2)]:
> L:=[seq(tubeplot( knot, t= 0..2*Pi,radius=rad,shading=zgrayscale,light=l,
                    ambientlight=[0.1,0.5,0.1]) ,l=lighters)]:
> display(L, insequence=true);
```

6.46.2 CORNIL Jack Michel (30.5.99)

I tested this example

with release 5 / windows 3.1 : NO problem

with release 5 / windows 95 : I found the problem you did

with release 5.1/ windows 95 : I found the problem you did

I think that the problem is with the graphix interface for windows 95

6.47 animations, to control the speed (30.11.98)

6.47.1 Leif Abrahamsson

Does anyone know if there is any way to control the speed of an animation (created either with animate or `display([seq(...)],insequence=true))` via some option (i e without slowing it down in the animation menu-line)? The problem is that when one exports a Maple-file to html, it is not possible to regulate the speed in e g Netscape.

6.47.2 Nobukazu Shimeno(2.12.98)

An animation is exported as a GIF animation. You can edit a GIF animation by softwares such as GIF Construction Set for Windows

`http://www.mindworkshop.com/alchemy/gifcon.html`.

I use GifBuilder on PowerMacintosh to change the speed of an animation in

`http://150.55.84.33/maple/gallery-e.html`.

6.47.3 John Pais (2.12.98)

I have had the same problem and here are two approaches:

1. Edit the animated gif created when you export the maple worksheet to html, inserting control statements before each image that sets the frame rate (delay between frames). For this I use the Gif Construction Set which is shareware that can be downloaded at: `http://www.mindworkshop.com/alchemy/gifcon.html`

Unfortunately, there is no way to put maple-like animation controls in animated gifs.

2. I have written a Java animation player that has all the needed controls (stop/start, faster/slower, 1 frame back/forward). Instead, of exporting the maple animation, I export only the gifs for each individual frame and then read them into my Java player.

You can find examples of both approaches at:

`http://interactive-mathvision.com/CKM/CKM.html`

6.48 anomalous true-false test (2.8.99)

6.48.1 Willard, Daniel Dr

The following program gives an erroneous answer to the "is" test half-way through, but a correct one at the end. The only difference is in the "assume" statement just ahead of the test: whether to assume `v = or <= 1/(P*q^n)`.

The answer should be "true" regardless. What gives? I am using Maple V 5.1 on a PC.

```
y:=n->v*(1-q^n);
m:=n->y(n)-y(n-1);
t:=n->(m(n)-m(n+1))/(1+m(n)*m(n+1));
P:=2/3;
q:=1-P;
t(n);
expand(%);
assume(n>=0,v<=1/(P*q^n));
about(v);
is(t(n)>=t(n+1));

restart;
y:=n->v*(1-q^n);
m:=n->y(n)-y(n-1);
t:=n->(m(n)-m(n+1))/(1+m(n)*m(n+1));
P:=2/3;
q:=1-P;
t(n);
expand(%);
assume(n>=0,v=1/(P*q^n));
about(v);
is(t(n)>=t(n+1));
```

6.48.2 Robert Israel (4.8.99)

Actually Maple's answer is quite correct in this case: it is not true in general that `t(n) >= t(n+1)` under the assumptions `n>=0,v<=1/(P*q^n)`.

For example:

```
> normal(t(1) - t(2));
                                                    2
                                          v~ (-81 + 4 v~ )
                                  -24 ------------------------
                                                 2             2
                                     (27 + 4 v~ ) (243 + 4 v~ )
```

The sign depends on the sign of v. But you didn't tell Maple anything that determines that sign. You probably meant to assume in addition that `v >= 0`.

```
> additionally(v >= 0);
> is(t(n) >= t(n+1));
                              FAIL
```

The "assume" facility is rather limited in capabilities, so it's not really all that surprising when a result is "FAIL". In general it's a very difficult problem to decide when an expression is nonnegative for all values of the variables in some region defined by inequalities. In this particular case, I think I can trace some of Maple's difficulties to some rather simple deficiencies.

```
> is(3^n > 0);
FAIL
```

It seems that Maple doesn't know that a positive power of 3 is positive. In fact it doesn't even know that this is real. Well, we can tell Maple this:

```
> additionally(3^n > 0);
```

Unfortunately, this still doesn't help. Even though Maple now "knows" that `3^n > 0`, it doesn't seem able to use that information in other expressions.

```
> is ((3^n)^2 > 0);
                              FAIL
```

```
> assume(x > 0); is(x + 3^n > 0);

                          FAIL
```

It may actually be better to use "exp", which Maple seems to know more about, rather than powers of a specific number. So I'll change q to `exp(lnq)`, where `lnq < 0 (which implies 0 < q <`

```
> restart; y:=n->v*(1-exp(n*lnq));
> m:=n->y(n)-y(n-1);
> t:=n->(m(n)-m(n+1))/(1+m(n)*m(n+1));

> assume(n>=0, v<= 1/((1-exp(lnq))*exp(n*lnq)), v>=0, lnq<0);

> is(t(n)>=t(n+1));
```

```
                          true
```

6.48.3 Helmut Kahovec (6.8.99)

>The answer should be "true" regardless.

I disagree. In the first case n=1 and v=-1 do not contradict your assumptions. If you substitute them into t(n) and t(n+1) then you get:

```
> restart;
> y:=n->v*(1-q^n):
> m:=n->y(n)-y(n-1):
> t:=n->(m(n)-m(n+1))/(1+m(n)*m(n+1)):
> p:=2/3: q:=1-p:
> assume(n>=0,v<=1/(p*q^n));
> is(t(n)>=t(n+1));

                                false

> simplify(subs({n=1,v=-1},t(n)-t(n+1)));

                                -1848
                                -----
```

```
                                  7657

> is(%>=0);

                                  false
```

In the second case assuming `v=1/(p*q^n)` is virtually the same as substituting it into `t(n)-t(n+1)`. The latter is zero for all n and thus greater than or equal to zero, anyway:

```
> restart;
> y:=n->v*(1-q^n):
> m:=n->y(n)-y(n-1):
> t:=n->(m(n)-m(n+1))/(1+m(n)*m(n+1)):
> p:=2/3: q:=1-p:
> assume(n>=0,v=1/(p*q^n));
> is(t(n)>=t(n+1));

                                  true

> simplify(subs({v=1/(p*q^n)},t(n)-t(n+1)));

                                   0

> is(%>=0);

                                  true
```

If you make more specific assumptions about n and v then Maple fails even if `_EnvTry` is set to 'hard':

```
> restart;
> y:=n->v*(1-q^n):
> m:=n->y(n)-y(n-1):
> t:=n->(m(n)-m(n+1))/(1+m(n)*m(n+1)):
> p:=2/3: q:=1-p:
> e:=simplify(t(n)-t(n+1));

                            n      (n + 1)      2
                           3  v (9         - 4 v )
```

```
                e := 8 ---------------------------
                                n      2       n      2
                              (3 9  + 4 v ) (27 9  + 4 v )

> assume(n,nonnegint);
> assume(v,positive);
> additionally(v<=1/(p*q^n));
> _EnvTry:=hard:
> is(e,nonneg);

                                  FAIL
```

The reason is that Maple cannot determine whether `3^n (or 9^n)` is greater than or equal to zero:

```
> [op(e)];

        n~       (n~ + 1)       2        1              1
    [8, 3  , v~, 9         - 4 v~ , -------------, --------------]
                                       n~       2      n~       2
                                    3 9   + 4 v~   27 9   + 4 v~

> map(is,%,nonneg);

                [true, FAIL, true, FAIL, FAIL, FAIL]

> is(3^n,nonneg);

                                  FAIL
```

This is a weak point of the assume facility, of course.

6.49 appendto or write to a file, maplevr3 and windows (27.6.00)

6.49.1 Charles Saunders

I am using Maple V Release 3. What is the entire process in order to writeto or appendto a file (myfile.txt). Is there a command which is required before the writeto or appendto command? The commands do not appear to work, also where is "myfile.txt" placed on my computer.

6.49.2 Robert Israel (28.6.00)

It appears that this does not work as expected in the worksheet interface of Release 3 under Windows. It does work in the DOS version of Release 3, and in later releases. For example:

```
> writeto(`myfile.txt`); a;
> b;
> printf(`c\n`);
> lprint(d);
> writeto(terminal);
```

In Release 3 under Windows 95, this leaves the following in the file myfile.txt:

```
b;
printf(`c\n`);
c
lprint(d);
d
writeto(terminal);
```

Thus: input lines after the one containing the "writeto" and up to and including the one with the next "writeto" (but without the prompts) are sent to the file, as are the results of printf and lprint commands, but not ordinary output.

The file will be placed in the directory that is current when Maple is started: by default this will be Maple's "bin" directory, so you want to be very careful not to overwrite any of Maple's own files.

In later releases, or the DOS ordinary output would also be written to the file, as would

the ">" prompt from input lines, so it would contain

```
                                        a

> b;
                                        b

> printf(`c\n`);
c
> lprint(d);
d
> writeto(terminal);
```

As for the statements in the help page ?writeto

```
... all future commands will have their results immediately stored in filename and
will not be displayed on the screen. - While in this mode, no prompt will appear
at the terminal.
```

this may once have been true, but now does not seem to be the case at all.

6.50 application center (15.3.99)

6.50.1 Michelle Gauthier

Announcing Waterloo Maple's NEW Application Center! This collection of Maple worksheets and sample solutions profiles how Maple V has been used around the world and across many disciplines. Visit `http://www.maplesoft.com/application_center.html` to see this dynamic collection of solutions to real world problems.

6.50.2 Scott Rabuka (28.6.99)

Check out the Waterloo Maple Inc. website at `http://www.maplesoft.com` and explore the Application Center with its application oriented Maple V content.

Visitors to the Application Center,

`http://www.maplesoft.com/application_center.html`

have numbered over 16,000 in just three months!

If you have a Maple V application you would like to share, send your proposal to `applications@maplesoft.com`. Share your achievements with the Maple world.

6.51 applications of maple to physics and differential equations (24.10.97)

6.51.1 J. David Wright

I would like to know about any resources for using maple to solve classical physics problems in mechanics, electro-statics/dynamics, and quantum mechanics, etc. I am especially interested in those applications which would relate to the methods of solving differential equations and boundary value problems in these areas.

I am thinking about seperation of variables, method of Frobenius, power series solutions, differentiation of power series, recursion relations, polynomial manipulation, etc.

Any information about books, ftp sites, web sites, etc. would be appreciated.

6.51.2 Michael Komma (2.10.97)

Since you are interested in *any* resources, I can offer you:

Moderne Physik mit Maple - Von Newton to Feynman (ITP) and

`http://userwst1.fh-reutlingen.de/~komma` -> Physik

(both in German `;-(`

6.51.3 Volker Schmidt (28.10.97)

I know only a german book: `"Jvrn Borgert: Maple in der Physik"` from Addison-Wesley (Germany).

In this book you can find examples for maintaine of energy and impulse, electrostatics/dynamics, differantial equations, optic and quantum mechanics.

6.51.4 Willard, Daniel, Dr., DUSA-OR (28.10.97)

Look in the journal Computers in Physics, published monthly by the Americal Institute of Physics.

6.51.5 Ron Greene (29.10.97)

My book, Classical Mechanics with Maple does some of this at a sophomore/junior level.

6.51.6 Tom Johnston (30.10.97)

The `www.amazon.com` book site has quite a few MAPLE books listed for various versions of MAPLE. Many are dedicated to solving diff. E. problems and applications of MAPLE in various areas of science and engineering.

6.51.7 Philip Yasskin (4.11.97)

Try our vector calculus package called `vec_calc` which can be obtained from

`http://calclab.math.tamu.edu/maple/vec_calc/`

It will do curve analysis, differential operators in function notation, multiple integrals and line and surface integrals.

6.51.8 Jean GARRIGUES (6.11.97)

I have written a package named tens3d. It is available at

`http://esm2.imt-mrs.fr/gar/maple/tens.html`

It is designed for tensorial calculus in classical physics in ANY 3D coordinates system. I contains algebric and differential operators on vectors and tensors.

NB: Actually, help files and name of procs are in french.

6.51.9 Nestor E. Sanchez (12.11.97)

I would recommend `Enns&McGuire`, Nonlinear Physics with Maple. Printed by Birkhauser. It has a lab book to apply the theory with Maple.

6.52 arc length integration and singularities (27.4.00)

6.52.1 Riddle, Larry

I would like to compute the arc length of the function f computed in the code below.

```
> g := sin(Pi*x)^(1/6);
> A := evalf(Int(g,x=0..1));
> f := g/A;
> fp := diff(f,x);
> arclength := evalf(Int(sqrt(1+fp^2),x=0..1));
```

I realize that the integral for the arclength is improper at both endpoints, so Maple must deal with the singularities. However, Maple just sits and sits and sits working on this problem. I have tried adding the option to reduce the number of digits of precision with no effect.

I have tried adding the flag to specify the method. I was able to get an answer at one point, though I am now unable to replicate that success. Either I give up waiting for Maple, or Maple reports that it cannot handle the singularity.

Mathematica gives the numerical value in a few seconds. The problem occurs with both release 5.0 and 6.0. I'm using Maple on a pentium machine.

So is there anything I can do to get Maple to do this integration? This is just one example from several that came up when my students worked on a project to find arc length of functions that have area 1 on interval `[0,1]` (hence the calculation of A above) and are 0 at both x=0 and x=1.

6.52.2 Robert Israel (28.4.00)

Since Maple seems to be having trouble dealing with the singularity, you might try a do-it-yourself approach. First of all, use symmetry to reduce the problem to integrating on `0..1/2`. Now the integrand has the following series near x = 0:

```
> S:= series(sqrt(1+fp^2),x,15);
```

Approximate the integral on, say, `0 .. 0.05` by the integral of this series (by examining the terms, it appears that the convergence of the series on this interval is sufficiently rapid).

```
> J1:= int(convert(S,polynom),x=0..0.05);

                J1 := .8205287812
```

On the rest of the interval, numerical integration has no problem.

```
 > J2:= evalf(Int(sqrt(1+fp^2),x=0.05 .. 0.5));
```

The result is:

```
> J:= 2*(J1+J2);

              J := 2.789958073
```

6.52.3 Peter Lindsay (29.4.00)

My run of mma does give an answer but it gives warnings about the accuracy. I used mma version 3.0

and by taking limits `{x,0.00000000000000001,0.9999999999999999}]` got warnings and result as follows:

```
In[44]:=
N[%]
NIntegrate::"slwcon":
    "Numerical integration converging too slowly; suspect one of the
following: singularity, value of the integration being 0, oscillatory
integrand, or insufficient WorkingPrecision. If your integrand is
oscillatory try using the option Method->Oscillatory in NIntegrate."
NIntegrate::"slwcon":
    "Numerical integration converging too slowly; suspect one of the
following: singularity, value of the integration being 0, oscillatory
integrand, or insufficient WorkingPrecision. If your integrand is
oscillatory try using the option Method->Oscillatory in NIntegrate."
NIntegrate::"ncvb":
    "NIntegrate failed to converge to prescribed accuracy after \!\(7\)
recursive bisections in \!\(x\) near \!\(x\) = \!\(0.999999999999999822`\)."
Out[44]=
2.78505
```

Using Maple 6, I did the arc length from `x=0.2 to x=0.8`, then added twice the arc length of the inverse function from the limits `0 .. to 1.016907...` and got

`2.789958072`.

I have the files here: if you are interested then I can easily send you them as email attachments.

6.52.4 Willard, Daniel Dr (8.5.00)

Can you change the variable so that there is no singularity at one end?

6.53 arcsin integral (18.3.97)

6.53.1 Bernard Marcheterre

This question refers to the integration of the square of arcsin(x). If you give Maple these commands:

```
>assume(x>0,x<1):
>int(arcsin(x),x);
```

It will respond with the correct result, at least according to Schaum's Mathematical Handbook of Formulas and Tables, equation 14.471. If, instead you go with:

```
>assume(x>0,x<1):
>int((arcsin(x))^2,x);
```

The unevalueted integral is returned. I found a way to reach the result (equation 14.476, Schaum) trough intparts from the student package but I wonder, is there a special command that I need to know? Have I reached the limit of the integration kernel with this special case?

6.53.2 Robert Mc Dougall (27.3.97)

We both reached the same limit :o) A package written with the all the asumptions made for all trigonometric functions would be awesome .

6.53.3 David Holmgren (4.4.97)

When I saw this posting, I thought I might try MuPAD 1.3 on the problem (I realise that this is rather naughty, but not to worry). In the following, asin(x) is the same as arcsin(x):

```
>> int(asin(x)^2,x)
```

which after only a short time (30 sec) produces:

```
                2                2    1/2
-2x + x asin(x) + 2 asin(x) (-x + 1)
```

I have no idea of what assumptions MuPAD makes with this problem.

6.53.4 Giorgio Taricco (11.4.97)

There is a simple way to tell Maple how to calculate

`int(arcsin(x)^n,x)`.

The following code does the job:

```
> restart:with(student):          # Restart and load student package

> assume(cos(u)>0):               # used later on

> Int(arcsin(x)^2,x):             # arcsin integral (inert form)

> changevar(arcsin(x)=u,%,u):     # change variables x->u

> value(simplify(%)):             # calculate integral

> subs(u=arcsin(x),%):            # change u back to x

> simplify(%);                    # final simplification
```

and works for any positive integer power of arcsin(x).

6.54 area problem (9.11.98)

6.54.1 Powerhouse Museum

How do I calculate the area under a curve in Maple when I haven't got a formula but experimental data?

Please send response to newsgroup or directly to myself: `michaelvdk@hotmail.nospam.com` (omit the nospam for the true address).

6.54.2 John Little (11.11.98)

There are any number of standard techniques from numerical analysis for this kind of calculation – they are called numerical quadrature rules. A simple, commonly used example of these methods is Simpson's Rule. If you have an odd number `n = 2k+1` of data points

```
         (x_1,f(x_1)), (x_2,f(x_2)), ... , (x_n, f(x_n))
with x_0 < x_1 < ... < x_n and with x_i - x_{i-1} = h for all i,
```

(i.e. the independent variable values are *equally spaced*) then the Simpson's Rule approximation to the integral of f over the interval $[x_1, x_n]$ (the "area under the curve" that you want) is

```
  (f(x_1) + 4f(x_2) + 2f(x_3) + 4f(x_4) + ... + 4f(x_{n-1}) + f(x_n))*h/3
```

(the 4's and 2's alternate in the summation). If you have the dependent variable values stored in a 1-dimensional array (or list) you can compute this in Maple by the following procedure (assuming $n = 2k + 1$ is odd):

```
ListSimpson := proc(fvals,n,h)

   local Simpson,j;

   Simpson := fvals[1] + fvals[n];
   for j from 2 to n-1 do
     if type(j,even)
       Simpson := Simpson + 4*fvals[j]
     else
       Simpson := Simpson + 2*fvals[j]
```

```
      fi
    od;
    RETURN(Simpson*h/3)
    end:
```

Here are two examples:

```
> read `/home/fac/little/Classes/Numer-98/ListSimpson.map`;
> fvals:=[1.4, 2.5, 2.3, 3.4, 5.6];

                  fvals := [1.4, 2.5, 2.3, 3.4, 5.6]
With equally spaced x-values, h = .9:

> ListSimpson(fvals,5,.9);

                              10.56000000
```

Another example showing close agreement with the actual value of an integral – data points sampled from the sin function at $x = .1$ to .9 at intervals of .1:

```
> fvals:=[seq(sin(.1+(i-1)*(.1)),i=1..9)];

  fvals := [.09983341665, .1986693308, .2955202067, .3894183423,

        .4794255386, .5646424734, .6442176872, .7173560909,

        .7833269096]

> ListSimpson(fvals,9,.1);

                              .3733944047

> int(sin(x),x=0.1..0.9);

                              .3733941970
```

For information on other similar methods, you can consult almost any textbook on numerical analysis. Hope this information is useful.

6.54.3 Michael Brennan (12.11.98)

All you need do is define a function, say $f(x)$ using the experimental data. Then you can apply either the command simpson or trapezoid from the student package to find approximates to the area. Of course I am assuming the data you have is in the form of a table where the readings are taken repeatedly over a period of time.##e.g.

Suppose you are trying to find the area of a swimming pool where you are given the following measurements for the width of the pool, where the readings are taken every two metres.

Area of a Swimming Pool. First enter the data as a list.

```
> with(student);
> L:=[0,6.2,7.2,6.8,5.6,5.0,4.8,4.8,0]:

> for k from 0 to 8 do w(2*k):=L[k+1] od;

> trapezoid(w(x),x=0..16,8);

> evalf(%);

> simpson(w(x),x=0..16,8);

> evalf(%);
```

We would require about 85 m^2 of material to cover pool.

6.54.4 Helmut Kahovec (13.11.98)

If you have a mathematical model then fit it to the data. If the regression is linear then see the online help page for stats[fit,leastsquare]. If the regression is nonlinear then see the article

```
    Fit for Anything?
    The Maple Reporter, Summer 1998 Edition
    pp. 6-7
```

You may download The Maple Reporter from http://www.maplesoft.com. Afterwards transform the fit into a procedure using unapply() and (numerically) integrate this procedure using evalf/Int.

If you don't have a mathematical model then I would suggest using splines to convert the data points into a (piecewise defined) procedure. See the online help page for spline(). If you have many points then let the degree of the segment polynomials be 1 (i.e., use lines), otherwise use cubic splines. However, if there are 30+ data points, the calculation of the splines may take some time, anyway. Afterwards (numerically) integrate the piecewise defined procedure within each interval separately using **evalf/Int** and add these areas using add(). Below I give you two examples of this.

```
> restart;
First example:

> f1:=x->exp(-x);

                          f1 := x -> exp(-x)

> evalf(Int( f1(x), x=0..2 ));

                             .8646647168

> l1:=[seq( [i/5,f1(i/5)], i=0..10 )]:
> plot(l1,style=POINT);

> readlib(spline):
> f1s:=spline(
>   map2(op,1,l1),
>   map2(op,2,l1),
>   x,
>   3
> ):
> add(
>   evalf(Int( f1s, x=l1[i][1]..l1[i+1][1] )),
>   i=1..(nops(l1)-1)
> );

                             .8648805898
```

Second example:

```
> f2:=x->ln(1+x);
```

```
                          f2 := x -> ln(1 + x)

> evalf(Int( f2(x), x=0..3 ));

                               2.545177444

> l2:=[seq([i/10,f2(i/10)],i=0..30)]:
> plot(l2,style=POINT);
```

If you don't need the splines explicitly you may define a procedure that calculates the area from this list of data points at once:

```
> AreaSpline:=proc(l::list(list),degr::integer)
> local x,i;
>   add(
>     evalf(
>       Int(
>         readlib(spline)(
>           map2(op,1,l),
>           map2(op,2,l),
>           x,
>           degr
>         ),
>         x=l[i][1]..l[i+1][1]
>       )
>     ),
>     i=1..(nops(l)-1)
>   )
> end:
> AreaSpline(l2,1);

                               2.544552717
```

6.54.5 Willard, Daniel (16.11.98)

Elementary calculus texts define integrals as the limit of a process of summing adjacent rectangles whose heights are the location of the left-hand ordinate (a data point), or the right-hand ordinate, or some average of the two. Simpson's Rule. Trapezoidal interpolation.

Splines (cubic fits to successive sets of data points). See the math section of the CRC Handbook of Physics and Chemistry, or Abramowitz and Stegun Handbook of Mathematical Functions.

6.55 args and &-operators problem (2.12.97)

6.55.1 Rafal Ablamowicz

I have discovered a few annoying problems (bugs?) with 'args' when defining &-operators. Below I can reproduce two problems, while the third one is elusive and appears randomly. I will address it too although I don't have a verbatim Maple output to show. I would appreciate any hints how to avoid them, or to understand them.

Let's define function `&t` as follows:

```
> `&t`:=proc() local L: L:=[args]; RETURN(L) end:
>
> K1 := `&t`(e1,e2,e3);
                          K1 := [e1, e2, e3]
> type(%,list(name));
                                  true
```

The above two answers are correct: `[e1, e2, e3]` is a list of names.

Here is the first problem:

```
> K := e1 &t e2 &t e3;
                         K := [[e1, e2], e3]
```

Notice, that the output form has changed even though `&t` has not: unlike before, now the output is a list consisting of a list and a name (I suspect that this is a result of any &-operator being implicitly left-associative in Maple).

Next, Maple confirms that K is of the type [list,name] and that K is not of the type list(name) like K1 was:

```
> type(K,[list,name]);
                                  true
> type(K,list(name));
                                 false
```

Given that the outputs K1 and K are different to me is illogical and for sure causes programming problems.

Here comes the second problem. I believe it is even worse than the first one: it has to

do with the same type checking except now it will be done inside the procedure, not outside. For that purpose, I will slightly re-define `&t` and call it `&T` for the sake of discussion:

```
> `&T`:=proc() local L: L:=[args];
>       if type(L,list(name)) then RETURN(list_of_names) elif
>          type(L,[list,name]) then RETURN(list_and_name) else
>          ERROR(`wrong types`) fi end:
>
> `&T`(e1,e2,e3);
                              list_of_names
```

This answer is correct and it has been expected since `L=[args]` is a list of name. Now please take a note of the following surprising result:

```
> e1 &T e2 &T e3;
                              list_of_names
```

That is unexpected and different than before: K above was of the type [list,name] so why does Maple think now that [args] inside `&T` is of type list(name) when before clearly [args] was explicitly not of that type? What is going on? Why two different results from the type checking, one done inside and one done outside?

The third problem has to do with Maple returning sometimes `[e1 &t e2, e3]` instead of `[[e1,e2],e3]` in the first example. I have no clue why. Has anyone encountered this problem before?

PS I run Maple V R. 4 patch 4.00b April 16, 1996, under Windows NT and 95.

6.55.2 Stanley J Houghton(3.12.97)

As I understand it, when you apply `&t` as an applicative expression such as

```
   `&t`(e1,e2,e3)
```

you are applying the procedure to an argument list of three variables but when you do

```
   e1 &t e2 &t e3
```

you are applying the procedure to an argument list of two arguments (e1,e2) and then

repeating the application of the procedure to (again to two arguments) the result of this operation and e3).

I comment further below and hope it helps

| Here is the first problem: ...

Your input is a equivalent to

```
> K:= `&t`(`&t`(e1,e2),e3);
```

Hence the resulting nested list

| Here comes the second problem. ...

This is like requesting (again)

```
>  `&t`(`&t`(e1,e2),e3);
```

which results in evaluation of `` `&t`(list_of_names,e3) `` which leads to the result you got.

6.55.3 Robert Israel(3.12.97)

| Here is the first problem: ...

The output has changed because the input has changed. In other words, Maple parses `e1 &t e2 &t e3` as `(e1 &t e2) &t e3` and does not automatically simplify it to `` `&t`(e1, e2, e3) ``. Seems reasonable enough to me.

| Given that the outputs K1 and K are different to me is illogical ...

I don't know what's illogical about it. `e1 &t e2 &t e3` and `` `&t`(e1, e2, e3) `` are different, that's all. When used in infix form, a `"&"` operator is a binary operator, not a ternary operator.

If `` `&t`(e1,e2,e3) `` is what you want, then `` `&t`(e1,e2,e3) `` is what you have to say.

| Here comes the second problem. ...

Not at all surprising, and nothing to do with type checking. Remember that Maple parses the input as `(e1 &T e2) &T e3`. So first it calls `&T` with arguments e1 and e2. Since [e1,e2] is a list of names, the result of that is a name, namely `"list_of_names"`. Then it calls `&T` with arguments `list_of_names` and e3, and since `[list_of_names, e3]` is a list of names it returns `list_of_names`.

| The third problem has to do with Maple returning sometimes ...

I have no idea why this would happen. It would mean that somehow the `e1 &t e2` was not being evaluated. Are you sure that it happens in this example exactly as you gave it to us? I suspect that it is not "random", but that you have done something in your Maple session that caused `e1 &t e2` to return unevaluated.

P.S. One thing that `_is_ a bug`, in my opinion, is that (if `&q` is unassigned) Maple prints both `(a &q b) &q c and a &q (b &q c) as a &q b &q c` (although the internal representations are different - you can use lprint to see that). For a non-associative operator this can lead to confusion.

6.55.4 Joseph Riel (3.12.97)

The following procedure does what you want (though I suspect your example was a simple model of the real problem).

```
`&t` := proc() map(proc(a) if a::list then a[] else a fi end, [args]) end:
```

| Here comes the second problem....

The reason is that you are returning `list_of_names`, not an actual list. What you should have done [to demonstrate the problem] is,

```
`&T`:=proc()
local L:
  L:=[args];
  if type(L,list(name)) then
       print(list_of_names)
  else print(list_and_name) fi;
  L
end:
```

now

```
a &T b &T c;
                      list_of_names
                      list_and_name
                       [[a, b], c]
```

which is as expected.

6.55.5 Burkhard Wald(4.12.97)

| Here is the first problem: ...

A procedure with an `&-name` is not implicitly assocative. First take an unnassigned operator name `&A` . You can write

```
> x &A y;
           x &A y
instead of

> `&A`(x,y);
           x &A y
You also can write

> x &A y &A z;
          x &A y &A z
```

which is interpreted as `‘&A‘(‘&A(x,y),z)` which is different from `‘&A‘(x,y,z)`.

Compare the following.

```
> eq:=(x &A y) &A z = x &A (y &A z);
           x &A y &A z = x &A y &A z
```

The echo on the screen let us thing that the equation is true. Not so.

```
> evalb(eq);
           false

> op(lhs(eq));
        x &A y, z

> op(rhs(eq));
        x, y &A z
```

Now you can understand what happends in your second example.

`&T` is called twice. First with the arguments e1,e2 and second with the arguments `list_of_names,e3` . Of cource `&T returns again list_of_names` .

| The third problem has to do with Maple returning sometimes ...

For example

```
> 'e1 &t e2' &t e3;
```

gives

```
   [e1 &t e2, e3]
```

I think your above operators are only pre-examples and what you really want to do is more complex. May be, I can give you some hints, if you say a bit more about your goals.

6.55.6 Jean GARRIGUES (4.12.97)

| Here is the first problem: ...

The result seems to me correct. It prooves that Maple interpret `"e1 &t e2 &t e3" as "(e1 &t e2) &t` Expressions are interpreted from left to right. The second `"&t" receives [e1,e2]` as first argument and e3 as second argument.

I assume you want a "concatenation". The following `&t` returns a sequence instead of a list

```
> restart:
> `&t`:=proc() local L: L:=args; RETURN(L) end:
> K1:=[e1 &t e2];
      K1 := [e1, e2]

> K2:=[e1 &t e2 &t e3];
      K2 := [e1, e2, e3]
```

| Here comes the second problem. ...

Why surprising ? Maple interpret `"e1 &T e2 &T e3" as "(e1 &T e2) &T e3"`. The result of `(e1 &T e2) is the string "list_of_names"` ...

6.55.7 Philip Yasskin (8.12.97)

Problem 1:

The difference between `&t(e1,e2,e2) and e1 &t e2 &t e3` is that the first calls `&t` once while the second calls `&t` twice.

Problem 2:

Try putting a trace into the procedure:

```
`&T`:=proc() option trace: local L: L:=[args];
      if type(L,list(name)) then RETURN(list_of_names) elif
      type(L,[list,name]) then RETURN(list_and_name) else
      ERROR(`wrong types`) fi end:
```

Then you get:

```
> &T(e1,e2,e3);
{--> enter &T, args = e1, e2, e3

                          L := [e1, e2, e3]

<-- exit &T (now at top level) = list_of_names}

                            list_of_names
```

and

```
> e1 &T e2 &T e3;
{--> enter &T, args = e1, e2

                            L := [e1, e2]

<-- exit &T (now at top level) = list_of_names}
{--> enter &T, args = list_of_names, e3

                       L := [list_of_names, e3]

<-- exit &T (now at top level) = list_of_names}
```

```
list_of_names
```

Clearly in `e1 &T e2 &T e3` there were two calls to `&T`. The first time it returned `"list_of names"` as it should; the second time `"list_of names"` was the first argument which is itself a name, so the result `"list_of names"` is correct.

6.56 arguments of a function, op function (19.9.00)

6.56.1 Tom Lovie

I wanted to know if there is a function that will return the argument to a RootOf function. Like an inverse RootOf function.

Example:

```
G := Q=RootOf(_Z^2+4*_Z+5);

        G := Q=RootOf(_Z^2+4*_Z+5)

rhs(G);
        RootOf(_Z^2+4*_Z+5)
```

now, I'd like to get at the quantity `_Z^2+4*_Z+5` so I could pass it to a discrim function.

6.56.2 Paul Goossens (20.9.00)

Try

```
op(1,RootOf(_Z^2+4*_Z+5));
```

6.56.3 Kevin Ulmes (20.9.00)

You could try the op function, to break down the root of function. The op function breaks down larger complex expressions into components, so used in this case, it would look something like this:

```
G := Q=RootOf(_Z^2+4*_Z+5);

                                      2
                  G := Q = RootOf(_Z  + 4 _Z + 5)

op(rhs(G));

                                    2
                                  _Z  + 4 _Z + 5
```

You can assign that last result to a variable and pass it to the discrim function, or to any other function that you want.

6.56.4 Robert Israel (20.9.00)

```
op(%);
```

or, for other forms of RootOf,

```
op(1,%);
```

6.56.5 Douglas Wilhelm Harder (20.9.00)

In this case, `op( * )` will work, but if something has multiple arguments, you can use `op( N, * )` to select the Nth operand:

```
> op( f( a, b, c ) );
                              a, b, c

> op( 1, f( a, b, c ) );
                                 a

> op( 2, f( a, b, c ) );
                                 b

> A := RootOf( _Z^2+4*_Z+5 ):
> op( 1, A );
                               2
                            _Z  + 4 _Z + 5
```

although it is probably a good idea to change the `_Z` to some other variable as `_Z` is recognized as being special in a number of cases in Maple:

```
> eval( op( 1, A ), _Z = x );
                                2
                               x  + 4 x + 5
```

You could always just enter the routine:

```
> ExtractRootOf := proc(A::specfunc(anything,RootOf), y::algebraic)
       eval( op( 1, A ), _Z = y );
```

```
end:

> ExtractRootOf( RootOf( _Z^2+4*_Z+5 ), x );

                                  2
                                 x  + 4 x + 5

> ExtractRootOf( RootOf( _Z^2+4*_Z+5 ), x^2 );

                                  4      2
                                 x  + 4 x  + 5
```

to do this for you automatically.

6.57 arrays and copies of arrays (14.12.00)

6.57.1 Jan van den Heuvel

I've been doing some programming in Maple 6, involving manipulation of arrays of sets. (If you want to know, the sets represent out- or in-neighbours of the vertices of directed graphs.) While trying to figure out why I couldn't get something to work, I encouraged some strange behaviour. Here is an example :

The following simple procedure takes as input an array, and just replaces its first element by an empty set.

```
clearset := proc(arrayset)
    local a;
    a := arrayset;
    a[1] := {};
    eval(a);
  end:
```

So lets take a small example.

```
> aset := array(1..2,[{1},{2}]);

                          aset := [{1}, {2}]
```

And now run the procedure, with the expected outcome.

```
> clearset(aset);
                                [{}, {2}]
```

But now the weird thing. Also the original array has changed. In other words, the procedure "clearset" has changed one of its input variables !!??

```
> eval(aset);
                                [{}, {2}]
```

Now it may have something to do with my clumsy programming skills, but a similar example using arrays with numbers behaves different :

```
clearset2 := proc(arraynumbers)
```

```
    local a;
    a := arraynumbers;
    a[1] := 0;
    eval(a);
  end:
```

```
> cset := [1,2];
                              cset := [1, 2]

> clearset2(cset);
                                   [0, 2]

> eval(cset);
                                   [1, 2]
```

Any explanations ?

A second, related question. Suppose I have an array, and I want to make a copy of it which doesn't change.

With variables I would do

```
> a := 1;
                                   a := 1

> b := eval(a);
                                   b := 1

> a := 0;
                                   a := 0

> b;
```

1 But with array this doesn't work !?

```
> aa := array([1,2,3]);
                             aa := [1, 2, 3]

> bb := eval(aa);
                             bb := [1, 2, 3]
```

```
> aa[1] := 0;
                                aa[1] := 0

> eval(bb);
                                [0, 2, 3]
```

How to do this instead ?

6.57.2 Robert Israel (15.12.00)

| But now the weird thing. Also the original array has changed. ...

Yes. The point is that := doesn't make a copy. After a := arrayset, both a and arrayset have the same value, which is an array. Changes to the array will affect both.

| Now it may have something to do with my clumsy programming ...

It makes no difference whether the entries are sets or numbers.

```
| > cset := [1,2];
```

That's not an array, it's a list. Lists and arrays are quite different. An assignment to a list element, e.g. `a[1]:= 0`, makes a new copy of the list and assigns it to a.

| A second, related question. Suppose I have an array, ...

Carl DeVore and I have just had an extensive discussion in this list about copying arrays, rtables, procedures and modules. For an array, the simplest way is `B:= copy(A)`.

see: Dynamic code with lexical variables (U. Klein)

6.57.3 Douglas Wilhelm Harder (15.12.00)

Use the copy command to make a copy of an array/matrix/table/vector or a Array/-Matrix/Vector. This is not new to Maple6.

```
clearset := proc(arrayset)
    local a;
    a := copy( arrayset );   # Change
    a[1] := {};
    eval(a);
  end:
```

In your second example, you are using a list, not an array. If you were to use:

```
> cset := array( [1, 2] );
```

then you would get the same behaviour as with your first example.

For your last example, copy also works.

```
> aa := array([1,2,3]);

            aa := [1, 2, 3]

> bb := copy(aa);

            bb := [1, 2, 3]

> aa[1] := 0;

              aa[1] := 0

> print(bb);
                [1, 2, 3]
> print(aa);
                [0, 2, 3]
```

6.57.4 Carl DeVore(18.12.00)

Have you considered using the networks package? Type ?networks for more info.

```
| > clearset := proc(arrayset)
```

You could just as well assign the element directly (no local variable needed):

```
clearset:= proc(a) a[1]:= 1; a end;
```

| But now the weird thing. Also the original array has changed. ...

It has only changed the contents on the variable, not the variable itself. It would be

different if you tried assigning directly to a.

```
a:= array(...whatever...);
```

would give you an error message inside the procedure.

The array is just a box. Changing the box is different from changing the contents of the box.

Here's an analogy: Ernic and Christina are children each with a box of toys. Ernic's box is red and has a loosely attached sign that says "Ernic's toys". Christina's is blue and has a loosely attached sign that says "Christrina's toys". Ernic acquires a new toy and puts it in his box.

Is there any need for him to change the sign on his box in any way? Of course not. Christina decides that she would like a red box for her toys. She discusses it with Ernic and he decides that he would like a blue box, and they decide to trade boxes. (Whether they decide to trade the toys also really doesn't matter to this analogy.) Should Ernic's sign now be put on the blue box and Christina's sign on the red box? Of course.

Think of the array name "a" as the sign on the box.

| Now it may have something to do with my clumsy programming ...

cset in not an array, it is a list. If you merely put something in square brackets, it is a list. For reasons that I don't fully understand, the designers of Maple decided that a list is not "just a box". Even though Maple will allow it in this simple case, you probably shouldn't be assigning to the elements of a list anyway. Weird things can happen that I won't go into right now.

| A second, related question. Suppose I have an array, ...

That "eval" would only be needed in exceptional circumstances.

| But with array this doesn't work !?

That command has only put a new, additional, sign on the box. Ernic and Christina decide to keep their toys in the same box and both signs are put on the one box. It has not created a new box. To create a new box, use the copy command:

```
bb:= copy(aa);
```

Also, over the past week or so, Robert Israel, Helmut Kahovec, and myself have been discussing on this mailing list more efficient ways of copying various objects, including

arrays.

6.58 arrays in group/transgroup (14.4.00)

6.58.1 Mike May, S.J.

Using Maple 6 I have parallel constructions that are being treated differently. Can anyone tell me why?

Motivational background:

I am updating worksheets I use in teaching abstract algebra so that they will run under Maple 6. The worksheets on computing Galois groups need the most revision since the galois function was rewritten between versions V4 and V5 and again with version 6.

The worksheet walks the students throught he computation focusing on the technique rather than just on the answer. (By the way, I think it is a nice feature of Maple that you can gain access to the code used for a computation.)

In the revision to version 6, information that was stored in the table `‘group/transgrp‘` is now scattered across a collection of tables `‘group/transgroup/InfoDeg‘` where Info is one of `{name, order, parity, generators, SnConjugates}` and Deg is an integer from 1 to 11.

At the same time the table `‘galois/groups‘` no longer contains group names that my students will recognize. I was able to rewrite the worksheets, but ran into the following interesting behavior:

Maple question:

Why is Maple treating the list of SnConjugates different from the other 4 lists in the code below?

```
> deg := 5:
> numb := 3:
> [`group/transgroup/names`||deg[numb],
>  `group/transgroup/order`||deg[numb],
>  `group/transgroup/parity`||deg[numb],
>  `group/transgroup/generators`||deg[numb],
>  `group/transgroup/SnConjugates`||deg[numb]];

   [{"F(5)", "5:4"}, 20, -1, {[[1, 2, 3, 4, 5]], [[1, 2, 4, 3]]},
        group/transgroup/SnConjugates5[3]]

> [eval(`group/transgroup/names`||deg[numb]),
```

```
>  eval(`group/transgroup/order`||deg[numb]),
>  eval(`group/transgroup/parity`||deg[numb]),
>  eval(`group/transgroup/generators`||deg[numb]),
>  eval(`group/transgroup/SnConjugates`||deg[numb])];

   [{"F(5)", "5:4"}, 20, -1, {[[1, 2, 3, 4, 5]], [[1, 2, 4, 3]]},
         [1, 0, 5, 0, 0, 10, 4]]
```

As the second block of code shows I can make Maple behave in the desired fashion. I am curious as to why parallel constructions seem to be treated differently.

6.58.2 Robert Israel (20.4.00)

‘group/transgroup/names5‘ and ‘group/transgroup/generators5‘ are arrays of sets, while ‘group/transgroup/order5‘ and ‘group/transgroup/parity5‘ are arrays of integers.

So e.g. ‘group/transgroup/names5‘[3] is a set, and this is subject to normal evaluation rules. However ‘group/transgroup/SnConjugates5‘ is an array of arrays, and arrays have last-name evaluation. Therefore

```
> ‘group/transgroup/SnConjugates5‘[3];
```

does not fully evaluate the array ‘group/transgroup/SnConjugates5‘[3] unless you use "eval".

6.59 assign the result of dsolve (27.10.95)

6.59.1 Burkhard Wald

May be you have a result of dsolve and want to plot it. For example:

```
> res:={a(t)=exp(t),b(t)=exp(-t)}:
> assign(res);
> plot({a(t),b(t)},t=-5..5);
```

Nice! It works!

But! Look at `a(0), a(t[1]-t[2])` ! What is "a" ? "a" ist a procedure with `a(t)=exp(t)` in its remember table. Nothing more! This is not what I want. What I want, can I get by:

```
> map(x->map(unapply,x,t),res);

                        { a = exp, b = t -> exp(- t) }

> assign(%);
```

And now think you are in a physics course and want to explain how easy it is with maple to solve problems and come to a visualization.

Is there anybody who know what I can do instead of this `map(x->map(unapply,x,t),res)` ?

6.59.2 Bhairav Joshi

I am also interested in the question you raise. An awkward method that I have used in such situations is this:

```
af := unapply( rhs(res[1]), t);
bf := unapply( rhs(res[2]), t);
```

$af(t)$ and $bf(t)$ can now be used in the usual manner.

6.59.3 Neil E. Berger 312-413-2139

well you could always do...

```
> plot({seq(rhs(res[i]),i=1..nops(res))},t=0..1);
```

or even set up a little proc...

```
> plot_sol_set := proc(x,range) plot({seq(rhs(x[i]),i=1..nops(x))},t=range)end;
> plot_sol_set(res,0..2);
```

this has the advantage of having no side effects, but of course the solutions must all depend on t.

6.60 assigned argument in a procedure (6.1.99)

6.60.1 Eddie Saudrais

I would like to test if an argument of a procedure is or not assigned. The result should be like (assuming test is the name of the procedure) :

```
>un:=1:
>test(un) ;
                          true
>test(deux);
                          false
```

The number of arguments is not supposed to be known.

6.60.2 Joe Riel (14.1.99)

The procedure assigned will indicate whether its argument is assigned, however, in order to use this in a procedure (I assume that you want to do something else) the formal parameter must be typed with uneval, i.e.

```
test := proc(x::uneval) assigned(x) end:
```

It isn't possible, I believe, to handle an indefinite number of arguments.

6.60.3 Dr. B. Wald (15.1.99)

This function is part of maple and called assigned . Inside a procedure it works with global and local names. But you have to be carfull. Take

```
f:=proc(x) assigned(x) end;

a:=1;
f(a);
Error, wrong number (or type) of parameters in function assigned
```

This is because the evaluation of a to 1 take place before the call of f . If you want to pass the variable a (instead of the value of a) to f, you have to use a uneval-quotation in the function call

```
f('a');
```

A better way is to declare the parameter of f as an "eval to name" parameter

```
f:=proc(x::evaln) assigned(x) end;
```

Now f(a) gives no problem. I hope this helps.

6.60.4 Stanley J Houghton (15.1.99)

In rel 5 I make use of the uneval formal parameter type. This passes the name through, eg as 'x' in the example proc below that does what you want I believe, and can be tested with the built in function 'assigned':

```
> f:=proc(x::uneval)if assigned(x) then ("OK") else ("not OK") fi end:
> f(un);f(deux);
                                    "OK"
                                "not OK"
```

However, if you want the value of x you do need to use eval(x) for the value (or eval(x,n), where n is 2 or greater).

6.61 assigning to a function a(1) (1.10.01)

6.61.1 Arieh Tal

The following behavior puzzles me: apparently Maple doesn't consider a(1) as a different name than a but is also giving an error when I try to reassign a(1) and not when I reassign a?!

```
> a(1):=1;

                              a(1) := 1

> a:=12;

                               a := 12

> a(1);

                                  12

> a(1):=1;
Error, invalid left hand side in assignment

> a:=13;

                               a := 13

> a(5);

                                  13

> a(12):=1;
Error, invalid left hand side in assignment

> a(12);
                            13
```

6.61.2 Adri van der Meer (2.10.01)

```
> a(1):=1;
                                  a(1) := 1
```

Here you make an undefined procedure a and place the value a(1) in its remember table:

```
> whattype(eval(a));
                                  procedure

> op(4,eval(a));
                                table([1 = 1])


> a:=12;
                                   a := 12
```

Now you assign to a the value 12:

```
> whattype(a);
                                   integer
```

```
... which may act as the constant function x -> 12:
```

```
> a(1);
                                       12

> a(1):=1;
Error, invalid left hand side in assignment
```

But, of course, a constant function doesn't have a remember table, so you can not place a value in it.

6.61.3 Dr Francis J. Wright(2.10.01)

My best advice is not to use functional forms (i.e. anything of the general form y(x)) as variables. Maple allows this, but it causes considerable confusion. a(1) IS different from a. I will try to explain below what is happening in your example.

```
> a(1):=1;

                                a(1) := 1
```

a is unbound and so a(1) is a valid functional form that can be used as a variable.

```
> a:=12;

                                a := 12

> a(1);

                                   12
```

Now a has the value 12, and Maple accepts 12 as a constant function, so that 12 applied to any argument(s) returns the value 12.

```
> a(1):=1;
Error, invalid left hand side in assignment
```

But 12(1) is not a valid functional form for assignment. The above also explains the rest of your example.

If you want to define functions then use either the arrow syntax or the full procedure syntax.

[In fact, an assignment of the form `y(x) := z` implicitly assigns a procedure to y (if necessary) and then inserts an entry in its remember table, which can be very useful for defining "point" mappings, base cases for recursion, etc. The error above arises because the integer 12 cannot be used as an identifier and so cannot be assigned a procedure.]

6.61.4 Preben Alsholm (3.10.01)

a(1) is not a valid name in Maple. But if a is unassigned or the name of a procedure, then a(1) means a applied to 1 and you can assign to that, like `a(1):=37;` which has the effect of putting that information in the remember table for a.

The command `a:=12;` makes a the integer 12. Even the command `a:= x->12;` is by Maple simplified to the integer 12, thus also in this case making it impossible to assign to a remember table. That in fact the command a(1); gives 12 is due to the fact that 12(1); by Maple is simplified to 12. You could do like this:

```
a:=proc()  12  end proc;
```

This way a is not simplified to the integer 12 and so we can assign to its remember table:

```
a(1):=37;
```

To see the remember table, do this:

```
op(4,eval(a));
```

6.61.5 Robert Israel (9.10.01)

a(1) isn't a name, it's the value of the function "a" on the argument 1. At the start of your session, "a" was unassigned. When you entered

```
> a(1):= 1;
```

Maple actually assigned a value to "a", namely a function that returns unevaluated, but has a remember table containing the value you assigned:

```
> interface(verboseproc=3):
  eval(a);

              proc() option remember; 'procname(args)' end proc # (1) = 1
```

When you then said

```
> a:= 12;
```

you erased this definition, and gave "a" the number 12 as a value. A number can be used as a function with a constant value:

```
> 12(3);

                12
```

But a number doesn't have a remember table, so

```
> a(1):= 1;
```

is not allowed.

6.62 assignment in do loop (27.4.00)

6.62.1 Sherwin J. Singer

I am a newbie at Maple. I have lots of experience with Mma, and now I'm trying to master Maple 6. I'm having trouble with a calculation – strongly suspect its an "evaluation" thing – but so far it is resistant to whatever documentation I turn to. (Also I'm a little fustrated because Maple 6 on my NT computer crashes when attempting to read the Maple file from a MAC...guess I have to import as Maple text).

The actual workboot is complicated but I can reproduce my error with a simple calculation shown below. No doubt there are simpler ways to program this simple example, but in the actual complicated case using a do loop and appending to a list seems best, so I'd appreciate if someone could help me find my error.

* I'm trying to make c, a list of Arrays of Vectors, from within a do loop over the variable i.

* I calculate a Vector R for each value of i, and I want to append the current value of R to a list nmaed c. Evidently I'm appending the symbol R, because every way I do it, all members of c are equal to the latest value of R. I've tried eval, evalm(out of desperation...evalm is really for linalg, right?), value,...no success

```
with(LinearAlgebra):
 R:=Array(1..6);
 R[1]:=Vector([1,1,1]); R[2]:=Vector([2,2,2]);
 c:=[];
 for i from 1 by 1 to 4 do
     for j from 3 by 1 to 6 do
         R[j]:=R[j-1]+Vector([i,i,i]);
     end do;
 c:=[op(c),eval(R)];
 end do;
```

p.s. In the windows version is there a way to set a preference for making only the enter key start evaluation, not the return key. The is a way on the MAC version, but I haven't found it on the windows versin.

6.62.2 Herman Jaramillo (28.4.00)

I am using Maple release 5 and have no problems. I see that the syntax in Maple 6 is different. I am surprise to see Capital letters at the starting of some instructions (like Array or Vector) that is the typical style of Mathematica.

Here is the example you just wrote down:

```
c := [];
                                  c := []

for i from 1 by 1 to 4 do
 for j from 3 by 1 to 6 do
    R[j]:= R[j-1]+vector([i,i,i]):
 od:
c := [op(c),evalm(R[i])]:
od:

c;
           [[1, 1, 1], [2, 2, 2], [5, 5, 5], [10, 10, 10]]
```

I believe this is what you should get, shouldn't it?.

6.63 assignment of variables in functions and loops (5.2.01)

6.63.1 Richard Patterson

Compare the two commands

a) sum(k,k=1..3); and

b) for k from 1 to 3 do print(k) od;

For (a), coming into the command, k must be unassigned but leaving the command, k has no assigned value.

For (b), coming into the command it doesn't matter if k is assigned or not, but on leaving the command, k does have an assigned value.

So my question is, can someone give me some way to think about these two types of commands, to categorize them logically with others, so that I won't continually make mistakes with them. It would be so much easier if these were consistent.

6.63.2 Carl DeVore(7.2.01)

Here's is a possible mnemonic for it:

In the first command, in the syntax, variable k is *accessed first, then assigned*.

In the second command, k is *assigned first, then accessed*.

It's not a perfect statement of what's really going on, but perhaps you can work with it.

6.63.3 Dr Francis J. Wright (7.2.01)

sum is a Maple function implemented as a procedure, which evaluates its arguments before doing anything with them, as do all procedures by default. Hence, if k has a value before sum is executed, sum sees that value internally. `sum(k,k=1..3)` is a Maple expression consisting of a single function call, and in Maple an expression is a valid instance of a statement.

(For this kind of explicit summation you should use the function add, which does not evaluate its arguments and behaves much more like the for loop.)

for k do ... is a Maple statement, which has completely different semantics from a function call. A statement in Maple is not an expression. There is no reason to expect it to behave similarly to sum.

It says use the variable k (not its value) as the loop counter in the following loop.

I suggest you stop thinking in terms of "commands", and instead think of expressions that evaluate to values and statements that do something and do not evaluate to anything.

6.63.4 Adri van der Meer (7.2.01)

(a) "sum" is a procedure and applying a procedure

(1) implies full evaluation of the arguments;

(2) doesn't affect the value of the arguments.

So

```
k := 10:
sum(k,k=1..3);
evaluates to

sum(10,10=1..3);
```

There are some exceptions

(1) the arguments of eval, evalf, seq and traperror are not evaluated

(2) arguments with type array or table only evaluate to the last name, i.e. if you have

```
A := : B := A:
then foo(B); evaluates to foo(A) and not to foo( array(1..2,[p,q]) )
```

(3) Some procedures can have names as arguments that can get a value by the procedure.

(b) In a loop like

```
for k from 1 to 3 do print(k) od;
```

the first thing done is the assignment `k := 1` and this works even if k has already a value. When the loop ends the last thing done is the assignment `k := 4` (followed by something like: `if k>3` then leave)

6.63.5 Keith Geddes (7.2.01)

I happen to have posted a couple weeks ago, to our internal group of Maple developers, a response related to what you are mentioning above.

The main point: Consistency holds among "add", "mul", "seq", ...

The "sum" and "product" commands are in a different category – consistent with "int" as alluded to in my message below.

Most often, users really want to be using "add" not "sum". Use "add" for explicit summation of values. Use "sum" only for symbolic summation.

It looks to me that sum needs special evaluation rules. Consider:

Well ... this has been long-debated in the past and I believe the correct conclusion was made.

While your one type of example here would "work well" with the special evaluation rules you propose, other examples would "not work well".

Consider the analogy between sum and int. Should int(f, x) *not* evaluate f? It has to, and it does.

I believe that both sum and int must evaluate their arguments as at present.

What the user needs to be pointed to is: Use add (not sum) for the type of situations in your example below.

```
> f:=proc(n) if (n mod 2)=0 then 1  else -1 fi end;

          f := proc(n) if n mod 2 = 0 then 1 else -1 end if end proc

> sum(f(n),n=0..50);
                                       -51

> sum('f(n)',n=0..50);
                                        1
```

I think the average user is not expected to delay evaluation - the first argument to sum should be of type uneval.

Here is how it works with add (to the user's delight!) –

```
> f:=proc(n) if (n mod 2)=0 then 1  else -1 fi end;
```

```
          f := proc(n) if n mod 2 = 0 then 1 else -1 end if end proc

> add(f(n),n=0..50);
                                        1
```

6.63.6 Joe Riel (home) (7.2.01)

A possible reason that you are having problems is that you are using sum when you should be using add. While the syntax of add is similar to sum, the summation variable in add is always local to the function, that is, it doesn't matter whether a variable of the same name has been assigned.

sum is intended to be used for symbolic summations. As such, it must be able to work with existing expressions. add, on the other hand, must have the summation variable explicitly included within the expression. To see this, consider the following

```
proc() local y;
 y := i^2;
 [sum(y,i=1..3),add(y,i=1..3)]
end();

          [14, 3*i^2]
```

The add function gave a different result because its summation variable i is different from the variable i in y. From this it is apparent (I believe) that the summation variable for a sum "must" be unassigned. Otherwise the expression to be summed is meaningless.

6.64 Assignment (4.6.02)

6.64.1 Edgar G. Goodaire

After many years, I have still kept caught up on Maple's assignment and unassignment rules.

What follows makes perfectly good sense to me.

```
> a := 7;

                                a := 7

> x := a;

                                x := 7

> x := x;

                                x := 7

> a :=5;

                                a := 5

> x;

                                  7
```

What follows does not!

```
A := matrix(2,2,[0,0,0,0]);
                                    [0    0]
                              A := [       ]
                                    [0    0]

> X := A;

                                X := A
```

```
> X := X;

                                X := A

> A := matrix(2,2,[1,1,1,1]);

                                  [1    1]
                           A := [      ]
                                  [1    1]

> evalm(X);

                                [1    1]
                                [      ]
                                [1    1]
```

6.64.2 Robert Israel(5.6.02)

It's "last name evaluation". See "Last name evaluation" in my Maple Advisor Database, `http://www.math.ubc.ca/~israel/advisor`:

"Normally, when you enter an expression it is evaluated completely. If your expression contains a variable that has a value, the variable is replaced by that value; if it contains a function call such as f(x), where f is a procedure or function, then this is replaced by the value returned by f with argument x. And if those values in turn contain variables or function calls, they are also evaluated.

Eventually (we hope) Maple obtains a form that does not require any further evaluation, and this is what it returns to us. However, certain types of object have special evaluation rules. In particular, tables, arrays (including vectors and matrices, but not Vectors or Matrices) and procedures use last name evaluation. This means that if the result of one level of evaluation of a name would be a table, array or procedure, then the chain of evaluation stops with that name."

This occurs when you say `"X:= A;"`: the right side is only evaluated as far as the name A, because another level of evaluation would yield an array. So X is assigned the name A, rather than the actual matrix. The result is that when you change A, X is affected.

If by getting around this problem you mean that you want to assign X the matrix rather than the name, you could say

```
> X:= eval(A);
```

This evaluates A fully (to a matrix) before assigning the value to X. Note that X and A now point to the same structure. If you reassign A by `"A:= ..."`, X will not be affected. However, any changes to the matrix will affect both A and X, e.g. if you say `"A[1,2]:= 3;"` it will make `X[1,2]` be 3 also. On the other hand, you could say

```
> X:= copy(A);
```

which will assign X a new matrix that has the same entries as A, but changes in one will not affect the other.

6.64.3 Edwin Clark(5.6.02)

Use copyinto:

```
> with(linalg):
> A := matrix(2,2,[0,0,0,0]);
> X:=matrix(2,2);
> copyinto(A,X,1,1);

> A:=matrix(2,2,[1,1,1,1]);

> evalm(X);

                                [0    0]
                                [      ]
                                [0    0]

> evalm(A);

                                [1    1]
                                [      ]
                                [1    1]
```

6.64.4 Dr Francis J. Wright (6.6.02)

By using rtable-based matrices instead of table-based matrices, i.e. use Matrix instead of matrix (and then don't use evalm). This only applies to Maple 6 and later.

Alternatively, use eval (or evalm) to force full evaluation, e.g.

```
X := eval(X);
```

would give what I assume you wanted.

The reason is to do with Maple's evaluation rules rather than how it handles assignment. A variable whose value is a table-based data structure or a procedure evaluates by default to itself rather than its value, as you see above. This does not apply to rtable-based values, which behave more like algebraic values.

6.64.5 Thomas Richard (6.6.02)

This way, only a pointer is copied (sometimes called a "shallow copy"). For copying the actual data structure (a "deep copy"), use

```
X:=evalm(X);

or (more general, works for arrays, tables and rtables):

\begin{MAPLEinline}
X:=copy(X);
```

6.65 Associated Legendre functions (24.9.99)

6.65.1 TANGUY Christian

There seem to be something curious about the associated Legendre function of the first kind, LegendreP(v,u,x) (I am using version 5.1 on a PC under NT 4.0):

```
> g1 := sum('binomial(2*n+1,2*k+1)*2^(3*k)','k'=0..n);

                                          (3/4)     n
        g1 := 1/8 (2 n + 1) sqrt(Pi) 8      (-7)  LegendreP(n, -1/2, -9/7)
```

let us see g1 for n=0 and 1

```
> for n from 0 to 1 do g1 od;

                                    (3/4)
                           -2 (-1)

                          (3/4)
          - 21/8 sqrt(Pi) 8      LegendreP(1, -1/2, -9/7)
```

first surprise, we would have expected 1 and 11, respectively, and certainly not a complex number in the case n=0. I found no way to obtain the answers expressed as integers, so I decided to force the floating point evaluation of g1

```
> for n from 0 to 3 do evalf(g1) od;

                    1.414213562 - 1.414213562 I

                            22.00000000

                            298.0000001

                            4286.000004
```

Hmmm, truncation errors... AND, apart from the spurious complex value for n=0, what seems to be TWICE the correct result.

Let us try something else, using the inert operator Sum.

```
> g1b := Sum(binomial(2*n+1,2*k+1)*2^(3*k),k=0..n):
> for n from 0 to 3 do value(g1b) od;

                                  1

                                 11

                                149

                               2143
```

correct results at last!

In fact, the result is the sum of the nth powers of `9 +- 4sqrt(2)` with appropriate coefficients. It can also be calculated using

```
h := proc(n) option remember;
if n<2 then 10*n+1
else 18*h(n-1)-49*h(n-2)
fi
end;
```

I have two questions:

- why are the LegendreP functions used in the simplification of sums when even the Help about them acknowledges that their definition is a bit tricky, with `_EnvLegendreCut being -1..1 or 1..infinity`?

- is the numerical evaluation of these functions correctly implemented? `LegendreP(0,-1/2,cos(phi))` differs from `2*sqrt(tan(phi/2)/Pi)`, as given in the `Gradshteyn-Ryzhik (eq. 8.753.1)`.

6.65.2 Helmut Kahovec (30.9.99)

Apparently there is a bug in Release 5 concerning Legendre functions. Release 4 uses hypergeometric functions for the sum:

```
> restart;
> g1a:=sum(binomial(2*n+1,2*k+1)*2^(3*k),k=0..n);

        g1a := (2 n + 1) hypergeom([-n + 1/2, -n], [3/2], 8)

> for n from 0 to 3 do simplify(g1a) od;

                                1
                                11
                               149
                               2143
```

This is the correct result.

6.66 Assume a constant (3.2.00)

6.66.1 Mohamad Saad

I would like to assume a parameter as a constant using MAPLE V R5 I used the following:

```
> assume(L, constant);
> type(L, constant);

                                false

> whattype(L);

                                symbol
```

how can I assume a constant

6.66.2 Dr Francis J. Wright (4.2.00)

Here is one possible solution (in Maple V R5.1, but I think this has worked in most previous releases of Maple V):

```
> constants := L, constants;

   constants := L, false, gamma, infinity, true, Catalan, FAIL, Pi

> type(L, constant);

                                 true
```

6.66.3 Dr Francis J. Wright (4.2.00)

Here is one possible solution (in Maple V R5.1, but I think this has worked in most previous releases of Maple V):

```
> constants := L, constants;

   constants := L, false, gamma, infinity, true, Catalan, FAIL, Pi

> type(L, constant);

                                true
```

6.66.4 Robert Israel (4.2.00)

There is a global variable "constants" that contains all names that Maple knows as symbolic constants. To make L a constant, you just append it to this variable's value:

```
> constants:= constants, L;

> type(L, constant);

                           true
```

If you want "evalf" to give L a numerical value, you can assign that to `` `evalf/constant/L` ``. For example:

```
> `evalf/constant/L`:= 1.2345;
> evalf(2+L);

                           3.2345
```

I get as result: 2. + L. As stated in the help page of evalf you should define `` `evalf/constant/L` `` as a procedure. If I do so: `` `evalf/constant/L` := proc() 1.2345 end: `` I get the above mentioned result for `evalf(2+L)`. (U. Klein)

6.66.5 Colin Birch (7.2.00)

The error is that assume fixes properties of variables and relationships between them. It cannot affect types. The correct syntax for checking properties is:

```
> is(L,constant);

                                true
```

The confusion was caused because, as well as the property constant, there is also a type constant, which seems to include the type numeric and any names held within the global variable constants.

Thus it is possible to give a name or symbol the type constant, using a command like:

```
> constants := constants,myconstant;
```

I don't know what the benefit of doing this is.

There are other properties and types that share names, such as integer.

I hope this helps, but I am not sure what significance the property constant has in Maple. Perhaps the question should be "Why do you want to assume that L is constant?"

6.67 assume and is (4.2.01)

6.67.1 Arif Zaman

Here is a simple problem which assume seems to FAIL on:

```
> assume(x<1);assume(y<1);
> is ((1-x)>0);
                        true

> is ((1-y)>0);
                        true

> is((1-x)*(1-y)>0);
                        FAIL
```

6.67.2 Colin Birch (13.2.01)

In fact the behaviour is even stranger than you report:

```
> restart;
> assume(x<1);assume(y<1);
> is((1-x)*(1-y)<0);

                                false

> is((1-x)*(1-y)>=0);

                                 FAIL
```

6.67.3 Willard, Daniel Dr (13.2.01)

Ask Maple why many of their programs don't accept "assume" as mandatory directive.

6.68 assume and protect (11.9.98)

6.68.1 Brillet Jean

What do you think about that ? (MAPLE r5 for Mac)

```
> protect(Z):
> assume(Z>0);
Error, (in t1) attempting to assign to `Z` which is protected
```

OK, now

```
> unprotect(Z):
> assume(Z>0):
> protect(Z):

> about(Z);
Originally Z, renamed Z~:
  is assumed to be: RealRange(Open(0),infinity)
```

OK, but

```
> assume(Z<0);
```

Despite the protect() statement, there is no error message and

```
> about(Z);

Originally Z, renamed Z~:
  is assumed to be: RealRange(-infinity,Open(0))
```

Is it a bug or a normal behavior?

6.68.2 Robert Israel (17.9.98)

```
| > protect(Z):
| > assume(Z>0);
| Error, (in t1) attempting to assign to `Z` which is protected
```

The point is that "assume" produces a new variable Z , and assigns this to Z. Protected variables cannot be assigned new values.

```
| > unprotect(Z):
| > assume(Z>0):
| > protect(Z):
| > assume(Z<0);
| Despite the protect() statement, there is no error  ...
```

Like most functions, "protect" has its arguments evaluated before it begins. So if you try to protect a variable that has been assigned a value, "protect" attempts to protect that value. In many cases this can't be done, and produces an error message:

```
> a:= 3: protect(a);
Error, (in protect) wrong number (or type) of parameters in function setattribute
```

The way to protect a variable that has been assigned a value is to delay evaluation with quotes:

```
> protect('a');
> a:= 4;
Error, attempting to assign to `a` which is protected
```

So when you enter "protect(Z)", what is really protected is the value of Z, which is the new variable Z . The next "assume" doesn't affect that Z , instead it produces a different Z and assigns it to Z. So the protection has no effect. On the other hand, "protect('Z')" would work.

```
| Is it a bug or a normal behavior?
```

It's slightly surprising behavior, but I wouldn't call it a bug. Compared to all the other complaints about the "assume" facility, which really could use some major rethinking, I'd say this is a very minor point.

6.68.3 Willard, Daniel, Dr. (18.9.98)

It is not inconsistent. The second use of "assume" over-writes the first. If you had used "additionally" instead of "assume" the second time, You should have got an error message.

6.69 Assume facility with functions, Assume and functional forms (11.2.97)

6.69.1 Mark S. Broski

Is it possible for the assume facility to recognize functional forms?

For instance,

`assume (f(t)>0)`

`is(f(t)>0)`

gives the result TRUE

but if I enter

`is(f(t)+f(t)>0)`

an error code is returned.

6.69.2 Robert Israel

Not an error code, but FAIL (indicating Maple doesn't know) in Release 3. This particular problem is fixed in Release 4:

```
> is(f(t)+f(t)>0);
                              true
```

However, assuming `f(t) > 0` doesn't say anything about f(x), or any other value of the function except literally f(t).

```
> is(f(x)>0);
                             FAIL
```

Ideally you'd like to be able to say

```
> assume(f>0);
```

But this doesn't work:

```
> is(f(x)>0);
                             FAIL
```

6.69.3 Brian Blank

I note that Robert Israel has already given a response to the question as asked. Perhaps the following constructive workaround might be of some use to you:

```
> f := t -> exp(Re(f1(t))):  # Use abs(f1(t)) if you want nonnegative.

> is( f(x) > 0 );

                                true

> is(f(x)+f(t) > 0);

                                true

> is( 1/f(Pi) > 0 );

                                true

> is(sqrt(f(t)) , real);

                                true

> is( ln(f(t)) , real );

                                true

> assume( b < c );   is( int( f(t) , t = b .. c ) > 0);

                                true
```

6.70 Assume not handled (24.10.96)

6.70.1 Stephen M. Carr

I'm forwarding this to the MUG from a local professor, for comments. The following shows some annoying behaviour in both R.3 and R.4., at least from a pedagogical perspective. While `exp(-infinity)` is nicely evaluated to zero by Maple, multiply the argument by an assumed postive constant and the exp is no longer evaluated. Any reasonable work-arounds or other comments about how to make Maple a little "smarter" about this?

```
Symbolic Evaluation of Limits at Infinity
G.D. Stubley
Mechanical Engineering Department
University of  Waterloo
Waterloo, ON, N2L 3G1
email: stubley@sunwise.uwaterloo.ca
```

First example works as expected. In this short sample file we consider the application of boundary conditions at large values (infinite) of the independent variable in solutions to differential equations. As an example problem, let's say that the general solution to the differential equation is:

```
> restart:
> f := x -> c1*exp(-x) + c2*exp(+x);

                  f := x -> c1 exp(-x) + c2 exp(x)
```

where c1 and c2 are constants to be determined. For this example assume that the boundary condtions are:

```
> bc1 := f(0) = 1;

                          bc1 := c1 + c2 = 1
and

> bc2 := f(infinity) = 0;

                     bc2 := c2 exp(infinity) = 0
```

Notice how the limit of **exp(-infinity)** is correctly set to 0 so that bc2 can be readily solved for c2.

```
> c2:=solve(bc2,c2);

                                c2 := 0
```

This value for c2 is now substituted into bc1 which is solved for c1 which gives the expected final result for f(x).

```
> bc1 := subs(c2=c2,bc1);

                           bc1 := c1 = 1

> c1 := solve(bc1,c1);

                              c1 := 1

> f(x);

                              exp(-x)
```

Second example illustrating the issue at hand. Now let's consider a second example that differs only in that the argument of the exponential functions is

```
> Omega * x
```

where Omega is a positive real constant. Notice how we assume and check that Omega is a positive real number.

```
> restart:
> assume(Omega,real);
> additionally(Omega > 0);
> additionally(Omega < 1.0e+50);
> about(Omega);
Originally Omega, renamed Omega~:
  is assumed to be: RealRange(Open(0),Open(.10e51))
> f := x -> c1*exp(-Omega*x) + c2*exp(+Omega*x);
```

```
f := x -> c1 exp(-Omega x) + c2 exp(Omega x)
```

where c1 and c2 are constants to be determined. It is curious that Omega is not renamed `Omega ~ ~` in this expression. Again assume that the boundary conditions are:

```
> bc1 := f(0) = 1;

                          bc1 := c1 + c2 = 1
and

> bc2 := f(infinity) = 0;

    bc2 := c1 exp(-Omega~ infinity) + c2 exp(Omega~ infinity) = 0
```

Notice how the limit of `exp(-Omega*infinity)` is not set to 0 so that bc2 is not readily solved for c2.

```
> c2:=solve(bc2,c2);

                            c1 exp(-Omega~ infinity)
                  c2 := - -------------------------
                              exp(Omega~ infinity)
```

This gets uglier when c2 is substituted into bc1 which is then solved for c1 to give the unnecessarily complicated final result for f(x).

```
> bc1 := subs(c2=c2,bc1);

                            c1 exp(-Omega~ infinity)
              bc1 := c1 - ------------------------- = 1
                              exp(Omega~ infinity)

> c1 := solve(bc1,c1);

                           exp(Omega~ infinity)
       c1 := - ------------------------------------------
               -exp(Omega~ infinity) + exp(-Omega~ infinity)

> f(x);
```

```
       exp(Omega~ infinity) exp(-Omega~ x)
- ------------------------------------------
  -exp(Omega~ infinity) + exp(-Omega~ infinity)

            exp(-Omega~ infinity) exp(Omega~ x)
    + ------------------------------------------
      -exp(Omega~ infinity) + exp(-Omega~ infinity)
```

Does this simplify when we Omega an actual value?

```
> f := subs(Omega = 2,f(x));

f :=

          exp(infinity) exp(-2 x)
    - -------------------------------
      -exp(infinity) + exp(-infinity)

           exp(-infinity) exp(2 x)
    + -------------------------------
      -exp(infinity) + exp(-infinity)

> f := unapply(simplify(%),x);

                        f := x -> exp(-2 x)
```

While it is reassuring that everything works out OK in the end when actual values are substituted, it does seem that some of the symbolic nature of the solution has been lost.

To my simple way of thinking I would have expected the second example to unfold just as the first example did. What has gone wrong?

6.70.2 Robert Israel

From a pedagogical perspective, I think it's better to make the user a little smarter. First, instead of evaluating a function `"at" x = -infinity` you should be taking a `limit as x -> -infinity`, which is really what is going on.

Unfortunately, this doesn't go completely smoothly either:

```
> assume(Omega > 0);
> f:= x -> c1*exp(-Omega*x) + c2*exp(+Omega*x);
> bc1:= f(0) = 1;
> bc2:= limit(f(x), x=infinity) = 0;

                        bc1 := c1 + c2 = 1

                  bc2 := signum(c2) infinity = 0
```

And Maple can't solve this for c2.

```
> solve(bc2, c2);

                          RootOf(signum(_Z))
```

So it's up to the user to recognize that this means $c2 = 0$. The Maple user still requires some mathematical knowledge. This is not necessarily a bad thing...

6.71 Assume problem (15.4.97)

6.71.1 Jaroslav Hajtmar

I am a beginner to Maple V rel. 4 and a novice on Maple User Group. I have two "assume" questions.

1) I think, that "`assume(...)`" command has no influence on solving equations. Is it true?

for example :

```
> assume(x,real);assume(m,real);
> eq:=x^2-m*x+1=0; # quadratic equation with reals coefs.

                                2
                     eq := x~  - m~ x~ + 1 = 0

> ineqdiscr:=discrim(lhs(eq),x)>0; # for 2 real roots

                                            2
                     ineqdiscr := 0 < -4 + m~

> solve(ineqdiscr,m);

     RealRange(-infinity, Open(-2)), RealRange(Open(2), infinity)

> additionally(m,RealRange(Open(-2),Open(2))):
#discrim < 0  => roots are complex, but it was assume(x,real) ...
> solve(eq,x);

                              2 1/2                          2 1/2
       1/2 m~ + 1/2 (-4 + m~ )    , 1/2 m~ - 1/2 (-4 + m~ )
```

Why does Maple compute real roots? Assume(x,real) has no effect ? Or where is the mistake?

2) I want to use "assume" command for variable x from `RealRange(-infinity, Open(-2))` or from `RealRange(Open(2), infinity)` (union intervals)

Is it possible?

6.71.2 Robert Israel(21.4.97)

```
| 1) I think, that "assume(...)" command has no influence on ...
```

It doesn't restrict the values that "solve" will find. I wouldn't quite say it "has no influence", because it may in fact influence transformations of the equations that occur in the process of solving them. For example:

```
> assume(x > 0); assume(y < 0);
> solve(sqrt(x^2)=1, x);

                        1

> solve(sqrt(y^2)=1, y);

                       -1
```

This is because `sqrt(x^2)` is simplified to x while `sqrt(y^2)` is simplified to -y (actually, before "solve" even starts).

```
| Why does Maple compute real roots? Assume(x,real) has no effect ?
```

It doesn't cause Maple to check whether the results satisfy the assumptions.

Actually, Maple doesn't even know whether or not `sqrt(-4+m^2)` is real.

```
> assume(m, RealRange(Open(-2),Open(2)));
> is(sqrt(-4+m^2),real);
                                      FAIL
```

although it does know that `-4+m^2 < 0`:

```
> is(-4+m^2 < 0);
                                    true
> is(-4+m^2 >= 0);
                                    false
```

But even if it did know, it wouldn't restrict the values obtained for x.

You can sometimes restrict the output of "solve" by including inequalities. For example:

```
> solve(x^4=1, x);
                                1, -1, I, -I

> solve({x^4=1, x < infinity}, x);

                                 {x = 1}, {x = -1}
```

However, this usually doesn't work if there are symbolic parameters.

```
> assume(m > 0);
> solve({x^4 = m, x < infinity}, x);
```

(No result is returned)

And, strangely enough (still with the same assumption on m):

```
> solve({x^4 = m^4, x < infinity}, x);

                                 {x = -m~}
```

(Note to Maple developers: this is a bug)

(No result in Maple V Release 5, U. Klein)

```
| 2) I want to use "assume" command for variable x from ...
```

Yes.

```
> assume(x, OrProp(RealRange(-infinity, Open(-2)),
                                  RealRange(Open(2),infinity)));
```

6.71.3 Willard, Daniel (DUSA) (23.4.97)

Be aware that "assume" carries no weight with "type". In `BesselI(x,y)`, for example, the Maple routine requires y to be of type integer, and preceding its use with the command `"assume(y,integer)"` does not get past the typing check early in the program.

Stupid!. I have complained to Maple about it without any observable improvement.

6.71.4 Robert Israel (24.4.97)

`Willard, Daniel (DUSA) <WillaD@hqda.army.mil> wrote:`

```
| Be aware that "assume" carries no weight with "type". ...
```

Please clarify your complaint. I would insist that "assume" should `_not_` carry any weight with "type": "type" must distinguish between, for example, variables and numbers. A variable is still a variable, not a number, even when you assume its values are integers. If you want to know whether the value of a quantity is an integer, you can use "is", not "type".

As for BesselI, I presume you're talking about the first argument, not the second, where integers are not particularly special. AFAIK it does not "insist" that the first argument is an integer either: Maple is perfectly happy to calculate, e.g.,

```
> BesselI(Pi, Pi);
                              BesselI(Pi, Pi)
> evalf(%);
                               1.011423336
```

I guess your complaint may be with some of the automatic simplifications, e.g.

```
> BesselI(3, 0);
                                  0

> assume(x, posint); BesselI(x,0);

                              BesselI(x~, 0)
> simplify(%);

                              BesselI(x~, 0)
```

In that case, you do have a point.

6.72 Assume(16.2.01)

6.72.1 John Trapp

I am sure that this has been reported before.

```
> assume(-3<t);additionally(t<3);
> about(t);
Originally t, renamed t~:
  is assumed to be: RealRange(Open(-3),Open(3))

> int(2/(t^2-9),t);

> int(-2/(9-t^2),t);

> -int(2/(9-t^2),t);
```

In each case the solution is `1/3ln(t~-3)-1/3ln(t~+3)`. Maple does not recognize the domain of the integrand at all and we have the log of a negative number. Ugly.

6.72.2 Carl DeVore (19.2.01)

If you allow for the constant of integration to be complex, then the result makes sense. If you evaluate this anitiderivative between real limits, you will get a real answer.

I do realize, however, that this is difficult to explain to a beginning calculus student when you're trying to teach them Maple.

6.72.3 E. Elbraechter (20.2.01)

From the online help of 'int' the statement:

Note that no constant of integration appears in the result.

Therefore the integral of an real valued integrand can have a constant imaginary part;

```
> restart;
> assume(-3<t, t<3);
>   int(2/(t^2-9),t):
>   evalc(%):
> J  := %;
```

```
            J := 1/3 ln(3 - t) - 1/3 ln(t + 3) + 1/3 I Pi

> Jr := Re(J);
               Jr := 1/3 ln(3 - t) - 1/3 ln(t + 3)
```

For `-3 < t < 3` this is a real valued expression.

6.72.4 Adri van der Meer (20.2.01)

Try:

```
> f := int(2/(x^2-9),x=0..t);
```

and Maple will recognize the assumption made on t.

6.73 assume, help needed (22.2.99)

6.73.1 Kiat Huang

I'm using xmaple-VR5 on an Alpha DEC and I have the following recurring problem.

Once I do

```
> assume( A > 1/4 , L > 0 );
```

say which makes a particular ode possess complex eigenvectors,

```
> ode:= A*diff(u(x),x$4) + diff(u(x),x$2) + u(x) - 1 = 0;
```

then once I solve for the boundary conditions for the constants `_Ci` (produced by dsolve) and substitute back into the solution, using unapply, `u(A,L,x)` Maple is unable (or at least unwilling) to allow me to substitue for A or L. i.e.

```
> u(B,T,x);
```

produces the output `u(A,L,x) for any B,T.`

I've tried unassign and `A:='A'` but to no avail.

6.73.2 Helmut Kahovec (5.3.99)

At each assumption Maple generates a local name usually printed with a tilde. The original variable is assigned that local name and must not be assigned any other value explicitly.

If you want to assign that local variable a specific value then you have to use assign() with the original variable, though. assign() evaluates its arguments and thus assigns in fact the local variable that specific value. For more see online help or [1].

Anyway, I'd solve your problem as follows:

```
> restart;
> assume(A>1/4,L>0);
```

IMO it's not necessary to make assumptions about L.

```
> ode:=A*diff(u(x),x$4)+diff(u(x),x$2)+u(x)-1=0;
```

```
                    / 4      \   / 2      \
                    |d       |   |d       |
          ode := A~ |--- u(x)| + |--- u(x)| + u(x) - 1 = 0
                    |  4     |   |  2     |
                    \dx      /   \dx      /

> sol:=dsolve({ode},{u(x)});
```

You didn't tell us your specific boundary conditions, so let's assume the following equations eq1, eq2, ... for example. Note the order of substituting into $u(x)$:

```
> eq1:=eval(subs(sol,x=0,u(x)))=0;

                 eq1 := 1 + _C1 + _C2 + _C3 + _C4 = 0

> eq2:=eval(subs(sol,x=L,u(x)))=0;

> eq3:=_C3=0;

                            eq3 := _C3 = 0

> eq4:=_C4=0;

                            eq4 := _C4 = 0
We don't want to have RootOf expressions in the solution:

> _EnvExplicit:=true;

                        _EnvExplicit := true

> s:=solve({eq.(1..4)},{_C.(1..4)});
```

Let's take the first set of s, substitute sol and `s[1] into u(x)` (in that order), simplify the expression, and establish $u1()$ as a function of A, L, x. Note that we do not call the solution function u since $u(x)$ should remain an unevaluated function call:

```
> u1:=unapply(simplify(subs(sol,s[1],u(x)),symbolic),A,L,x);
```

Note that the formal parameters A, L, x of the solution function $u1$ are different from `A~,L~,x` in earlier expressions. Now you get:

```
> u1(B,T,x);
```

You may repeat the last two steps for the remaining three sets `s[2], s[3], and s[3] of s.`

Reference:

`[1] Andre Heck, "Introduction to Maple" 2nd ed., Springer Verlag 1996, pp. 87-91`

6.74 assume, multiple assumptions (16.3.98)

6.74.1 Lucy Schloesser

The function `p(k,n) with k,n` Integer and `n>=k>=0` is defined as follows.

```
> assume(n, posint);
> assume(k, posint);
> assume(n>=k);
> assume(k>=0);
> p:=x^k*(1-x)^(n-k)*binomial(n, k);
```

I want maple to solve `int(x^k*(1-x)^(n-k)*binomial(n,k), x=0..1);` . It is `= 1/(n+1);` (with integration by parts)

```
> int(p, x=0..1);
```

Definite integration: Can't determine if the integral is convergent. Need to know the sign of `--> -n+k`

Will now try indefinite integration and then take limits. Has someone written a program to solve this problem?

6.74.2 Douglas B. Meade (17.3.98)

To obtain the expected result of this calculation it is necessary to use a little caution in the way assumptions are declared. In particular, the assume command overrides previous assumptions; to add assumptions the additionally command should be used. To illustrate the difference, use the about command to query the current assumptions about a specific Maple name. All of this, together with the simplifications necessary to obtain the simplified form for your integral are contained in the Maple session attached to the end of this message.

```
> restart;
> p:=x^k*(1-x)^(n-k)*binomial(n, k):
> about( n ); about( k );
n:
  nothing known about this object

k:
```

```
  nothing known about this object

>
> assume(n, posint);
> assume(k, posint);
> assume(n>=k);              # overrides both previous assumptions!
> assume(k>=0);
> about( n ); about( k );
Originally n, renamed n~:
  nothing known about this object

Originally k, renamed k~:
  is assumed to be: RealRange(0,infinity)

>
> assume(n, posint);
> assume(k, posint);
> additionally(n>=k);     # adds this condition for both  n  and  k
> #assume(k>=0);          # unneeded since k is already positive
> about( n ); about( k );
Originally n, renamed n~:
  Involved in the following expressions with properties
    -n+k assumed RealRange(-infinity,0)
  is assumed to be: AndProp(RealRange(1,infinity),integer)
  also used in the following assumed objects
  [-n+k] assumed RealRange(-infinity,0)

Originally k, renamed k~:
  Involved in the following expressions with properties
    -n+k assumed RealRange(-infinity,0)
  is assumed to be: AndProp(RealRange(1,infinity),integer)
  also used in the following assumed objects
  [-n+k] assumed RealRange(-infinity,0)

> A := int( p, x=0..1 );

                A := binomial(n, k) Beta(k + 1, 1 + n - k)

>
```

```
> convert( A, GAMMA );

                              GAMMA(n + 1)
                              ------------
                              GAMMA(2 + n)

>  simplify( % );

                                   1
                                 -----
                                 n + 1
```

6.75 asympt ignores order on first use (10.1.00)

6.75.1 John S Robinson

Here is a quirk discovered by a colleague:

asympt does not seem to honour the Order setting the first time it is called for a particular expression:

```
> restart: R:=sqrt(r^2-a^2);

                                      2    2
                           R := sqrt(r  - a )
```

Series expansion for large r converted to an algebraic expression. Note that Order is ignored.

I think this is well described as a 'quirk'; Maple seems to need to do one series expansion of the expression before it starts to honour the value of Order. If you skip/execute the a0 line below then the a1 line ignores/honours the Order option. If you cange the - in the a0 line to +, it works differently, so it seems to be something to do with remembering the series for that particular expression.

```
> a0:=asympt(1/sqrt(x^2-a^2),x);

                          2          4          6
                         a          a          a        1
       a0 := 1/x + 1/2 ---- + 3/8 ---- + 5/16 ---- + O(----)
                          3          5          7        9
                         x          x          x        x

> a1:=asympt(1/R,r,4);

                                      2
                                     a        1
                     a1 := 1/r + 1/2 ---- + O(----)
                                      3        5
                                     r        r
```

```
> a2:=asympt(1/R,r,4);

                                    2
                                   a        1
                  a2 := 1/r + 1/2 ---- + O(----)
                                    3       5
                                   r       r

> a3:=asympt(1/R,r);

                                2          4
                               a          a        1
             a3 := 1/r + 1/2 ---- + 3/8 ---- + O(----)
                                3          5       7
                               r          r       r
```

Without executing the a0 line we get:

```
> restart: R:=sqrt(r^2-a^2);
> a1:=asympt(1/R,r,4);

                                2          4
                               a          a        1
             a1 := 1/r + 1/2 ---- + 3/8 ---- + O(----)
                                3          5       7
                               r          r       r

> a2:=asympt(1/R,r,4);

                                    2
                                   a        1
                  a2 := 1/r + 1/2 ---- + O(----)
                                    3       5
                                   r       r

> a3:=asympt(1/R,r);

                           2          4          6
                          a          a          a       1
```

```
a3 := 1/r + 1/2 ---- + 3/8 ---- + 5/16 ---- + O(----)
                    3          5           7        9
                   r          r           r        r
```

This is not a great problem, but I am intrigued as to why it happens, and if there is a better anser than 'do it twice'

6.75.2 Robert Israel(18.1.00)

This is actually a bug in "series" (which "asympt" calls). I don't have any advice other than to call it twice. The reason it works the second time, I think, is that "series" consults its remember table to see if it has computed the same series before to this or higher order; if it has done so, it truncates the series appropriately.

6.76 asympt() with multiple variables? (16.2.01)

6.76.1 Daniel Krofchick

I am trying to simplify several long expressions using the assumption that some constants in the expression are much larger than others. The expressions are obtained by solving a cubic. Here is the code

```
lambda:=[solve(l^3+A*l^2+B*l+C,l)]:
A:=K32+K34+K43+K21+K23+K12:
B:=K21*K43+K23*K34+K12*K43+K12*K34+K12*K32+K21*K34+K32*K43+K23*K43+K12*K23+K21*K32:
C:=K43*K21*K32+K12*K23*K34+K12*K32*K43+K12*K23*K43:
```

where all $Kij > 0$ and presumably Real (since they represent rate constants). The assumption I want to use to simplify the expression is `K12,K21,K34,K43>>K23,K32`).

Is there any way to do this in Maple 6? I have been playing with asympt which works well when I want to make this kind of assumption for one variable (in a simpler system derived from a cubic), but the function isn't defined for multiple variables.

6.76.2 Chris Eilbeck (19.2.01)

One way would be to write `K23= t*k23, K32 = t*k32`, and expand in a Taylor series in t. But the answer is still very messy in this case, even with t=0!

6.76.3 Boris Alexeev (20.2.01)

1st answer:

```
restart;
f:=exp(1/x+1/y): n:=2: m:=3:
asympt(f,x,n): remove(has,%,0):
asympt(%,y,m): remove(has,%,0);
```

Towards solution:

Is this from chemical kinetics? YES!

chemical reaction

```
x1=x2
x2=x3
x3=x4
```

differential equations

```
x1' = -k12*x1+k21*x2
x2' = k12*x1-k21*x2-k23*x2+k32*x3
x3' = k23*x2-k32*x3-k34*x3+k43*x4
x4' = k34*x3-k43*x4
```

Jacobi matrix

```
array([[l+K12,-K21,0,0],[-K12,l+K21+K23,-K32,0],[0,-K23,l+K32+K34,-K43],
       [0,0,-K34,l+K43]]);
```

2nd answer:

```
restart;
```

get Jacobi matrix

```
array([[l+K12,-K21,0,0],[-K12,l+K21+K23,-K32,0],[0,-K23,l+K32+K34,-K43],
      [0,0,-K34,l+K43]]);
ll:=simplify(linalg[det](%)/l):
```

switch asymptotics on

```
s:={K12=x*k12, K21=x*k21, K34=x*k34, K43=x*k43};
```

switch asymptotics off

```
si:={k12=K12,k21=K21,k34=K34,k43=K43,x=1}; sll:=subs(s,ll/x^2);
```

get crude solution (one eigenvalue)

```
asympt(sll,x,1): remove(has,%,0);
collect(subs(si,%),l,factor);
`crude eigenvalue=`,solve(%,l);
```

get next (and last) approximation (two values)

```
asympt(sll,x,2): remove(has,%,0);
collect(subs(si,%),l,factor);
`eigenvalue revisited (even twins)=`,solve(%,l);
```

6.76.4 Helmut Kahovec (21.2.01)

Instead of `K12,K21,K34,K43 >> K23,K32` I'd like to prefer `K23,K32 << K12,K21,K34,K43`. Then we may use mtaylor() as shown in the Maple6 session below.

```
> restart;

> A:=K32+K34+K43+K21+K23+K12:
> B:=K21*K43+K23*K34+K12*K43+K12*K34+K12*K32+
>    K21*K34+K32*K43+K23*K43+K12*K23+K21*K32:
> C:=K43*K21*K32+K12*K23*K34+K12*K32*K43+K12*K23*K43:

> equ:=l^3+A*l^2+B*l+C:

> lambda:=[solve(equ,l)]: nops(lambda);

                                3

> mtaylor(lambda[1],[K23,K32],2,[1,1]):
> map(factor,%):
> subs(
>   {signum(K43-K21+K34-K12)=1,signum(-K43+K21-K34+K12)=-1},
>   map(simplify,%,assume=positive)
> ):
> map(factor,%):
> subs(
>   {signum(K43-K21+K34-K12)=1,signum(-K43+K21-K34+K12)=-1},
>   map(simplify,%,assume=positive)
> ):
> map(factor,%):
> subs(
>   {signum(K43-K21+K34-K12)=1,signum(-K43+K21-K34+K12)=-1},
>   map(simplify,%,assume=positive)
```

```
> ):
> map(factor,%):
```

Unfortunately, I could not find another way to force Maple to choose the appropriate branch cut:

```
> subs(J=I,map(simplify,subs(I=J,%),symbolic)):
> lambda1:=map(factor,%);

                                          K32 K34
                lambda1 := -K34 - K43 - ---------
                                         K34 + K43

> mtaylor(lambda[2],[K23,K32],2,[1,1]):
> map(factor,%):
> subs(
>   {signum(K43-K21+K34-K12)=1,signum(-K43+K21-K34+K12)=-1},
>   map(simplify,%,assume=positive)
> ):
> map(factor,%):
> subs(
>   {signum(K43-K21+K34-K12)=1,signum(-K43+K21-K34+K12)=-1},
>   map(simplify,%,assume=positive)
> ):
> map(factor,%):
> subs(
>   {signum(K43-K21+K34-K12)=1,signum(-K43+K21-K34+K12)=-1},
>   map(simplify,%,assume=positive)
> ):
> map(factor,%):
> subs(J=I,map(simplify,subs(I=J,%),symbolic)):
> lambda2:=map(factor,%);

                                          K21 K23
                lambda2 := -K12 - K21 - ---------
                                         K21 + K12

> mtaylor(lambda[3],[K23,K32],2,[1,1]):
> map(factor,%):
```

```
> subs(
>   {signum(K43-K21+K34-K12)=1,signum(-K43+K21-K34+K12)=-1},
>   map(simplify,%,assume=positive)
> ):
> map(factor,%):
> subs(
>   {signum(K43-K21+K34-K12)=1,signum(-K43+K21-K34+K12)=-1},
>   map(simplify,%,assume=positive)
> ):
> map(factor,%):
> subs(
>   {signum(K43-K21+K34-K12)=1,signum(-K43+K21-K34+K12)=-1},
>   map(simplify,%,assume=positive)
> ):
> map(factor,%):
> subs(J=I,map(simplify,subs(I=J,%),symbolic)):
> lambda3:=map(factor,%);

                                  K23 K12     K32 K43
                lambda3 := - --------- - ---------
                                K21 + K12   K34 + K43
```

Finally, let us check the solutions:

```
> testeq(
>   A=simplify(-(lambda1+lambda2+lambda3))
> );

                                  true

> testeq(
>   B=factor(
>     mtaylor(
>       expand(
>         lambda1*lambda2+lambda1*lambda3+lambda2*lambda3
>       ),
>       [K23,K32],2,[1,1]
>     )
>   )
```

```
> );

                                  true

> testeq(
>   C=factor(
>     mtaylor(
>       expand(-lambda1*lambda2*lambda3),
>       [K23,K32],2,[1,1]
>     )
>   )
> );

                                  true
```

6.77 asymptotics of ln(ln(x)+a) (15.4.96)

6.77.1 Sjoerd W. Rienstra

```
asympt(ln(x+a),x);
```

gives a correct asymptotic series for large x. However,

```
asympt(ln(ln(x)+a),x);
```

gives only the trivial answer, i.e. the same. How do I convince Maple that $\ln(x)$ also becomes large if x is large?

6.77.2 Robert Israel (25.4.96)

As far as I know, `asympt(f(x), x, n)`, which is essentially the same as `subs(t=1/x, series(f(1/t), t` is supposed to do its calculations using all terms that are `O(1/x^(n-epsilon))` for `epsilon > 0` (i.e. it won't include terms like `ln(x)/x^n`), but will include `x^(-n+1/10))`. The asymptotic series you want is

```
                              2           3           4
                 a           a           a           a
ln(ln(x)) + ----- - 1/2 ------ + 1/3 ------ - 1/4 ------ + ...
            ln(x)            2           3           4
                          ln(x)       ln(x)       ln(x)
```

which would have infinitely many terms that are not `O(1/x^p)` for any `p > 0`. So that does not really fit the job description for "asympt". Anyway, you can get your series by

```
subs(u = ln(x), asympt(ln(u+a), u));
```

6.77.3 Gerald A. Edgar (26.4.96)

What would you like the answer to be? Presumably you do not object if `asympt(x^2,x)` returns `x^2` ... Similarly, `ln(ln(x)+a)` is one of the basic types that Maple uses. Examples:

```
> asympt(ln(ln(x)+a),x);
                          ln(ln(x) + a)


> asympt(ln(ln(2*x)),x);
                        ln(ln(x) + ln(2))
```

```
> asympt(ln(ln(x^2)),x);
                          ln(2 ln(x))


> asympt(ln(ln(3*x^2+2*x-2)),x,2);

                                    2                 1
     ln(ln(3) + 2 ln(x)) + ---------------------- + O(----)
                           3 (ln(3) + 2 ln(x)) x       2
                                                      x
```

6.77.4 Sjoerd W. Rienstra (14.5.96)

I would like to thank the MUG contributors Robert Israel and Gerald Edgar for their responses on my question. They explain what happens inside Maple. Nevertheless, I'm not fully convinced that Maple's results are consistent, and are what they should be.

The argument that Maple only expands in powers of x is probably true in most cases, but not in general. For example,

```
asympt(ln(exp(x)+a),x);
```

gives an asymptotic series in powers of `exp(x)`.

Also:

```
asympt(ln(ln(x)+x+exp(x)),x);
```

gives powers of exp(x), an asymptotic expansion in x in the coefficients of each exp(x), and a seemingly random "order" of the error `O([x/exp(x)]^p)`.

On the other hand, the similar command

```
asympt(1/(exp(x)+a),x);
```

does not do a thing, apart from the error message

Error, (in asympt) unable to compute series At the same time,

```
asympt(1/(x+ln(x)+a),x);
```

gives a series in powers of x with ln(x) considered as a constant.

Apparently: in general the asymptotic gauge functions are `x^p`, and sometimes `exp(p*x)`. Mixed expressions of x and exp(x) can be handled under a logarithm, but not under another power, and `ln(x)` is ignored as a function of x, and treated as a constant.

I think a more consistent approach would be to consider

```
(negative powers of ...),..1, ...,ln(ln(x))^p, ln(x)^p, x^p,
exp(x)^p, exp(x^q)^p, exp(exp(x)^q)^p, ....
```

as gauge functions. Maybe this is a bit too ambitious, but then at least

```
exp(-px), x^(-p), ln(x)^(-p), 1, ln(x)^p, x^p, exp(px).
```

Otherwise, the name "asymptotic expansion" is a bit misleading, and should be something like "Laurent series expansion around infinity".

6.77.5 Gerald A. Edgar (17.5.96)

There is a package (gdev) written by a third party that does certain expansions in Maple. Get info from Bruno Salvy, `Bruno.Salvy@inria.fr`, or perhaps even on the web at

`ftp://ftp.inria.fr/INRIA/Projects/algo/www/intro.html`

Here is your example...

```
> readlib(gdev);
proc(fct:algebraic) ... end

> readlib(glimit);
proc(fct:algebraic) ... end

> gdev(log(log(x)+a),x=infinity,4);

                                2           3
                  a           a           a           1
```

```
    ln(ln(x)) + ----- - 1/2 ------ + 1/3 ------ + O(------)
                ln(x)            2            3          4
                              ln(x)        ln(x)      ln(x)
```

6.78 auto correct button and 1 to infinity (4.12.02)

6.78.1 A. Prashanth

I wish to bring the following two points to the notice of the maple user community. please elucidate what these observations of mine imply (from a maple viewpoint and also from the theory viewpoint).

first, maple outputs unity when i launch the command:

```
> 1^infinity;
```

while analysis tells us this entity is actually an indeterminate form.

second, when i use the 'auto correct the syntax of the expression' button, the resulting syntax produces an error as its not an acceptable maple syntax as the following commands list shows:

```
> series(BesselJ(3,x),x);
```

produces a correct answer; but when the command at the prompt is incorrect:

```
> series(BesselJ(3,x,x);
```

and i use the auto correct button, maple 7 produces the following which is incorrect syntax as far as the series and BesselJ functions of the package are concerned:

```
> series(BesselJ(3,x,x));
Error, (in BesselJ) expecting 2 arguments, but received 3
```

please address these observations.

6.78.2 Carl Devore

```
 [ first, maple outputs unity ...
```

Given a function `f(x)` defined on the real numbers, it is reasonable to define `f(infinity) = limit(f(x)`, if this limit exists (or is +/- infinity). In this case, `f(x) = 1^x`. It is reasonable because it gives the *unique* (not indeterminate) continuous extension of the function over the compactification.

When a math book says that `1^infinity` is an indeterminate form, they mean that if f(x) or g(x) are *unknown* functions, except that we know `limit(f(x), x=a) = 1`

and `limit(g(x), x=a) = infinity`, then we still don't have enough information to compute `limit(f(x)^g(x), x= a)`.

If both f(x) and g(x) are known, then to say that the answer is indeterminate is simply a failure to give an answer.

`[ second, when i use the 'auto correct' ...`

It seems reasonable to me to expect an auto-correct feature to attempt to balance your parentheses, which is what is happening here. I think it is asking too much for it to check the number of arguments in each function call. Consider all that could happen:

```
- Procedures can have variable number of arguments.

- You could have redefined BesselJ to take three arguments.

- The number of arguments needed could be a function of the first argument.
```

You need to make a distinction between syntax and semantics. `series(BesselJ(3,x,x))` is correct Maple syntax.

6.78.3 Stephen Forrest (4.12.02)

`[ first, maple outputs unity ...`

In what sense is this an indeterminate form? If you consider a sequence of complex numbers $\{a_n\}$ going to infinity along any path, then $1^(a_n) = 1$ for any n. So the limit is defined and equal to 1.

As an example of something that `_is_` an indeterminate form, take $(-1)^\infty$, for which Maple returns `undefined + undefined*I`.

`[second, when i use the 'auto correct' ...`

In the first case, with `"series(BesselJ(3,x,x)"`, the error was a missing parenthesis. In the second case, the error was an omitted argument.

It's too hard for Maple to check that the number of arguments to a function is correct, so it doesn't try. It just makes sure that all the parentheses match, that you have an "end if" for every "if", etc.

6.78.4 Laurent Bernardin (5.12.02)

While Maple returns 1 by default for 1^{∞}

```
> 1^infinity;
                                        1
```

However, the fact that this is an invalid operation has been detected:

```
> NumericStatus();
invalid_operation = true, division_by_zero = false, overflow = false,

    underflow = false, inexact = false, real_to_complex = false
```

As with all numeric events, the default behaviour can be changed by installing a different event handler:

```
> NumericEventHandler(invalid_operation=(t->undefined));
                          invalid_operation = default

> 1^infinity;
                                    undefined

> NumericEventHandler(invalid_operation=exception);
                     invalid_operation = (t -> undefined)

> 1^infinity;
Error, numeric exception: invalid operation
```

(This is in Maple 6 and higher)

6.78.5 Robert Israel (5.12.02)

[first, maple outputs unity ...

Not a bug. In general, Maple is literal-minded, and you should not use 1 or infinity (or, for that matter, any other constant) when what you mean is some quantity that is approaching that constant. If what you want is a limit, then use the "limit" function.

[second, when i use the 'auto correct' ...

In fact the syntax of the command

```
series(BesselJ(3,x,x))
```

is correct, as far as Maple's parser is concerned; it just happens that the `BesselJ` function in Maple 7 does not allow 3 arguments. I would call this a problem of semantics rather than of syntax, since in Maple it is in general impossible to tell how many arguments a function will accept without looking at the code.

6.79 autocorrelation and crosscorrelation (3.12.96)

6.79.1 Sreejith Sukumaran

Are there any routines in Maple V which allows one to autocorrelation and crosscorrelation , given two sets of data ? Given a set of data points , velocity versus time , and there is a periodicity - is there a simple procedure to know the period ?

6.79.2 David Holmgren

I think that you will have to start from the ground-up with Maple as far as auto- and cross-correlation of data are concerned. It would not be too difficult to use the built-in FFT and iFFT functions to implement the correlation theorem (ie., `CCF = iFFT( FFT(f(t))*conj(FFT(`

6.80 automated grading of maple worksheet (25.5.01)

6.80.1 Guy Gendron

I am using Maple in a course and I am looking for an automated grading program of Maple worksheets.

Anything available on the market that would read the worksheet and look for specific answers or commands?

6.80.2 Theodore Kolokolnikov (28.5.01)

Although not exactly what you want, you might be interested in having a look at `http://allserv.rug.ac.be/~nvdbergh/aim`

It's a program that uses the Maple engine in combination with the web to grade math homework questions. It allows you to design math homework/test questions that the students can take over the web. The program is free – see website for more details.

6.80.3 Paul Eakin (28.5.01)

I'm not certain what your exact needs are.

We have a system developed at the University of Kentucky which may be of interest. We don't use the worksheet directly but rather export it to html. Problem sets are created as worksheets with embedded tags delineating problems and describing answers and answer-format. The exported html is processed into a document and posted to the system website `www.mathclass.com` with a browser interface. The system handles all of the bookkeeping one would expect an automated system to do.

We are developing a problem-solving course for teachers which employs the system as part of an NSF project. The materials for the preliminary version of the course offered in spring 2000 can be found at `http://www.msc.uky.edu/paul` - look for the materials on ma375/310

There are tutorials for students and teachers - there as well as some sets of slides from some recent talks on the system. The slides from the MAA meeting on March 6, 2001 at the bottom of the page are probably the best description. They unfortunately seem to crash my Netscape browser but work well with Internet Explorer.

The web site `mathclass.com` is due for a major "face lift" this summer but is functional.

Anyone is welcome to experiment with it as a student - there is a "getting started" tutorial there. Access to the course management and authoring tools require a higher level of access which we are happy to provide to colleagues.

6.80.4 Dirk F M Hermans (4.6.01)

It's not quite what you're looking for, but you may be able to do something with it. Look at AIM, which is a Maple based system for testing, build around maple code for assessing answers to questions. Yo'll find more information on

`http://allserv.rug.ac.be:8081/aim/`

6.81 automatic evaluation (27.6.01)

6.81.1 Guenter Fambach

I want to write the following expression to maple

```
g:=1/16*(5*x^4-28*x^3+36*x^2)
```

unfortunately, the output from maple is

```
g := 5/16*x^4-7/4*x^3+9/4*x^2
```

so maple has the factor 1/16 multiplicated with the others.

But now, I don't want maple to do that step automatically, I just want

```
g:=1/16*(5*x^4-28*x^3+36*x^2).
```

Question: How can I avoid this?

6.81.2 Chris Eilbeck (28.6.01)

This is a "feature" in Maple, any pure numerical factors are merged into the expression and cannot be extracted using the factor command. However in your case your expression has algebraic factors and can be written (and saved) as

```
1/16*x^2*(x2)*(5*x18);
```

A more general trick is to multiply by a dummy algebraic variable, say K, then put K=1 at some later stage.

6.81.3 Douglas Wilhelm Harder (28.6.01)

If you're looking for actual functionality, you cannot change this behaviour.

If you're looking for display only:

```
> 1/16*``(5*x^428*x^3+36*x^2);
                    4       3       2
          1/16 (5 x   28 x  + 36 x )
```

or

```
> 1/16 &* (5*x^428*x^3+36*x^2);
                     4       3       2
       1/16 &* (5 x   - 28 x  + 36 x )
```

work. In the first case you're multiplying 1/16 by a function, the name of which is ‘‘ (which is not printed when displayed) and whose argument is `5*x^4 ... .`

Otherwise, you can also use 'Standard Math' or 'Standard Math Input'. For example, highlight your input and select `Format>Convert to>Standard Math` . though this is only useful for input, not output.

6.81.4 Robert Israel (28.6.01)

Try this:

```
> g:=``/16*(5*x^428*x^3+36*x^2);
                           4       3       2
          g := 1/16 (5 x   28 x  + 36 x )
```

To get rid of the invisible name ‘‘, you can substitute ‘‘=1.

6.81.5 Helmut Kahovec (29.6.01)

Automatic distributing ‘*‘ over expressions of type ‘+‘ is a feature of Maple. The only way out is using the ‘‘() procedure:

```
> g:=``(1/16)*(5*x^428*x^3+36*x^2);
                                 4       3       2
               g := (1/16) (5 x   28 x  + 36 x )

> expand(%);
                           4        3        2
                     5/16 x   7/4 x  + 9/4 x
```

However, note that only ‘expand/‘() is defined. Thus other procedures may not know how to handle ‘‘(). You have to first expand expressions containing ‘‘()

6.81.6 Stanley J Houghton (11.7.01)

I have used the empty name. Thus in your example input

```
> g:=1/16*``(5*x^428*x^3+36*x^2);
```

this prints as you want since it contains a function call to `` in effect. However, to evaluate it you need to define `` as the identity function

```
> ``:=x>x;
```

Another way is to use

```
>g:=``/16*(5*x^428*x^3+36*x^2);
```

which doesn't evaluate but prints correctly. Evaluation is performed by using `subs(``=1,g) or by ``:=1`

However, both I admit are a "cludge".

6.82 automatic simplification, to switch off (6.4.99)

6.82.1 Eno Tonisson

Maple's automatic simplification is usually very useful. It seems to be comfortable that like terms in a sum or product are collected or greatest common divisors are removed or etc.

But ...

Is it possible to "switch off" the automatic simplification?

After entering

```
> eq1:=4*x+5*(x-4)=0;
```

Maple gives

```
eq1 := 9 x - 20 = 0
```

Is it possible to get `eq1 := 4*x+5*(x-4)=0`?

6.82.2 Helmut Kahovec(9.4.99)

Well, not directly. But look at the following trick:

```
> restart;
> eq1:=4*x+5*``(x-4)=0;

                    eq1 := 4 x + 5  (x - 4) = 0
```

‘‘() is the null string function as used by Maple when printing the result of ifactor():

```
> ifactor(10);

                                (2)  (5)


> lprint(%);
``(2)*``(5)
```

expand() simply returns the arguments of the null string function:

```
> showstat(`expand/`);

`expand/` := proc()
   1    args
end

> expand(eq1);

                                9 x - 20 = 0
```

6.82.3 Robert Israel (9.4.99)

What you can do is use the very useful
verb|“| function (that’s a name with no letters!). Enter the equation as

```
> eq1:= 4*x + 5* ``(x-4) = 0;

              eq1 := 4 x + 5  (x - 4) = 0
```

Before solving the equation or doing anything else that would require breaking up the (x-4), you should use "expand":

```
> expand(eq1);
              9 x - 20 = 0
```

6.83 avoid using brute force numerical methods (26.4.99)

6.83.1 Kevin Ulmes

I have a function that describes temperature throughout space, and a heat seeking particle that starts at a point and flies along the gradient to this function. I had to solve for the coordinates of the point when it was 5 units away from the origin. This problem was one that I had in my Math class, and I ended up solving it as shown below using brute force, which was what we were supposed to do. The function is `T=1/2*x^2*y-x*z+x+y`, and the initial starting point is `(1,0,2)`. I was wondering if anyone could help me out, in solving this problem exactly, or if there is anyway that I could optimize the current while loop that I have so that it would run faster, and therefore calculate the resultant point more accurately. I am using Maple V r4 to do this:

```
> with(linalg):
> with(plots):
> mag:=proc(vect); RETURN(sqrt(dotprod(vect,vect))): end:
> T:=1/2*x^2*y-x*z+x+y;

                                  2
                      T := 1/2 x  y - x z + x + y

> t_grad:=convert(grad(T,[x,y,z]),list):
> initial_point:=[1,0,2]:
> initial_direc:=subs({x=1,y=5,z=2},t_grad):
```

The following execuation loops determines the location of the particle when it is 5 units away from the origin. The accuracy of this result is determined by setting the err variable in the initial line. Digits are set to 20 to minimize roundoff error during the calculation.

Initilize variables so no errors returned

```
> Digits:=20:
> point_current:=initial_point:
> direc_current:=initial_direc:
> dist:=mag(point_current):
```

```
> er:=.001:
> while dist < 5 do
>   incre:=direc_current*fsolve(mag(scalarmul(convert(direc_current,vector),x))
    =er,x=0..infinity):
>   point_current:=evalm(point_current+incre);
>   direc_current:=subs({x=point_current[1],y=point_current[2],
                         z=point_current[3]},t_grad):
>   dist:=mag(point_current);
> od:
> current_point:=convert(point_current,list);
> distance:=dist;
> direction:=direc_current;
```

I am look at the current point, and taking the gradient vector, and taking a very small step in that direction. and evaluating the new point to check if it is more than 5 units away from the origin. I believe that I am doing this right, but it takes a very long time to solve, when `err < .001 (like > 10 min)`. And as `err >` the time increases very rapidly. I'm running the Windows version of MapleVr4, on a PII 400, running Win98 (just in case that is important).

6.83.2 Robert Israel (28.4.99)

Consider the system of differential equations

```
dp/dt = grad(T(p))
where p = [x,y,z]. This describes a particle moving so that its velocity is
equal to the gradient vector. It's more convenient for your purposes, however,
to write p as a function of r (the distance from 0). By the Chain Rule,

dp/dr = (dp/dt)/(dr/dt) = (grad(T(p)))/(p . grad(T(p))/r)
      = r grad(T(p))/(p . grad(T(p)))
```

You have the initial condition `p(sqrt(5)) = [1,0,2]`, and you want to find p(5). It's unlikely that there would be an exact "closed-form" solution, but Maple can do it numerically.

```
> with(linalg):
  T:=  1/2*x^2*y-x*z+x+y;
```

```
  G:= grad(T,[x,y,z]);
  Rs:=map(normal,evalm( sqrt(x^2+y^2+z^2) * G/dotprod([x,y,z],G)));
  des:= {diff(x(r),r) = subs(x=x(r),y=y(r),z=z(r),Rs[1]),
         diff(y(r),r) = subs(x=x(r),y=y(r),z=z(r),Rs[2]),
         diff(z(r),r) = subs(x=x(r),y=y(r),z=z(r),Rs[3])};
  ics:= {x(sqrt(5))=1, y(sqrt(5))=0, z(sqrt(5))=2};
  F:=dsolve(des union ics, {x(r),y(r),z(r)},numeric);
  F(5);

[r = 5, y(r) = -1.952448370119240, x(r) = 3.252309163629305,

       z(r) = 3.257365618028734]
```

6.84 axis labels (18.2.99)

6.84.1 John F. Putz

The following produces a plot with the axes labeled correctly:

```
> with(plots):
> spacecurve([t*cos(t), 0, t], t=0..4*Pi, axes=boxed, labels=["x","y","z"]);
```

However, changing the axes to normal produces a plot with the labels reversed on the x- and y-axes:

```
> spacecurve([t*cos(t), 0, t], t=0..4*Pi, axes=normal, labels=["x","y","z"]);
```

Am I missing something?

6.84.2 Dr Francis J. Wright (22.2.99)

Having taken a look (using R5), I think it is just bad positioning that makes the labels appear to be reversed. By dragging the plot around I can make the labels jump all over the place. I think you have just hit on an example where the automatic positioning algorithm does not work too well.

6.84.3 Robert Israel (22.2.99)

Yes, to some extent. The "y" really is labelling the y axis and the "x" the x axis, although the "y" may appear to be closer to the x axis. The placement of axis labels is rather tricky, and unfortunately Maple still hasn't got it right. In particular, in "normal" axes with the bounds symmetrical about 0 or nearly so it's a bad idea to have the label in the middle of the axis.

CHAPTER 7

B

7.1 back substitution (2.7.96)

7.1.1 Carl David

I wonder if someone has a good strategy for backsubstituting?

For instance, define `f(x,y,z) = sqrt(x**2+y**2+z**2)` and then take a derivative of a function such as `diff(f(x,y,z),x)`. At this point I would like to back substitute for f, i.e., remove `sqrt(x**2+y**2+z**2)` where ever it occurs and replace it with f.

I find that I need contortions involving `subs(%1=f,...);` and that this trial and error procedure is not pedagogically reasonable.

Is there a scheme for doing this kind of back substitition so that physically meaningful variables can be employed consistently in the manner which pencil and paper work is done in physics and chemistry?

7.1.2 Jan-Moritz Franosch (8.7.96)

I think this is a very general problem in Maple. You may try using the alias function, but that is certainly not a final solution:

```
alias( a= 1/sqrt(x^2+y^2) );
diff( sqrt(x^2+y^2) , x );
                                  a*x
```

Maple automatically substitutes the subexpression `1/sqrt(...)` by a. Note that unfortunately `alias( a=sqrt(...) )` does not work because of the internal structure of

Maple expressions. But subs does not work also:

```
1/sqrt(...);
subs( sqrt(...)=b );
  1/sqrt(...)
```

I and certainly many Maple users have the problem of really "intelligent" backsubstitution in a way like "Find the most simple expressions using these substitutions or functions ...". E.g. if should be possible to find out that `x^4+y^2` should be expressed by `f(x^2,y)` if `f(x,y)=x^2+y^2` has been previously defined or given as a possible substitution.

7.1.3 DANIEL WILLARD, WILL (11.7.96)

comment and a caveat. ASSUMING the range/characteristics of a variable does not make it so in the 'eyes' of "whattype". This is, I believe, an error on Maple's part. It screws up the performance of such programs as `BesselI(n,x)` when n is only ASSUMEd integer, not TYPEd integer.

7.2 background colour of graph (3.3.00)

7.2.1 Aldrovando Azeredo Araujo

When exporting a worksheet to HTML i would like to have control of the background color of the graphs generated as .gifs..Especially i would like to generate a transparent background gif for the graphs. Anyone knows how to do this or where to find an answer?

7.2.2 Craig B. Watkins (6.3.00)

You need to insert the command

```
plotsetup(gif, plotoutput=`figurename.gif`,plotoptions="transparent=true");
```

7.2.3 jed wallace (6.3.00)

i think the easiest way to place a transparent color would be to open the image.gif file in a photoshop style program. "magic wand" the white space and delete.

if you do not change the folder or file name when re-saving then there should not be any problem in the maple generated html coding.

7.3 backgroundcolor, foregroundcolor (23.1.03)

7.3.1 Kristian Jantz

Is it possible to set up a backgroundcolor for a plot in maple? i cant find a suitable option, i would also like to know whether there is a command that allows to set the foregroundcolor for all coming plots, so that i dont need to type `color=XXX` all the way

7.3.2 Robert Israel (31.1.03)

Under Windows, it is possible to set the background colour for plots in "window", but not "inline" mode. In Maple's "Users" directory (or in the directory for a specific user, if you have installed Maple in multi-user mode) there should be a file named maple8.ini (or maple7.ini or maple6.ini if you have Maple 7 or Maple 6).

This is a Windows initialization file for Maple. It is a text file that can be edited with Notepad or any text editor. Look for a line that starts

`PlotBGColor=`

If what comes next is default or 255 255 255, the plot backgrounds will be white. If it is 0 0 0, they will be black. You can specify any three numbers from 0 to 255 for the red, green and blue components. This only takes effect when you start a new Maple session; you can't change the colour in the middle of a session.

The default foreground colours for most plots are kept (in RGB form) in the variable `_COLORRGB`. By default this is set to

```
[1.0, 0., 0.], [0., 1.0, 0.], [1.0, 1.0, 0.], [0., 0., 1.0],
[1.0, 0., 1.0], [0., 1.0, 1.0]
```

i.e. the sequence is red, green, yellow, blue, magenta, cyan (if there's just one curve to plot, it will be red, but if there are several the first will be red, the second green, etc., repeating if necessary). You can change this. The RGB values for the named colours can be found in `` `plot/colortable` ``.

For example, to make the sequence gold, gray, maroon, wheat, plum, brown, aquamarine you could say

```
> mycolours:= [gold, gray, maroon, wheat, plum, brown, aquamarine];
> _COLORRGB:= seq(`plot/colortable`[c], c = mycolours);
```

7.3.3 James Frye (11.2.03)

I am having a great deal of trouble getting Maple set up so that I can work with it. The problem is that I need large, high-contrast text, in white (or other easily-visible color) on a BLACK background. I can't figure out how to get this: there are no options on the menus, I can find nothing in the help that seems relevant, and none of the other users or the system people here know.

Is it just so obvious that no one ever bothered to document it? I mean most programs seem to be smart enough to automatically pick up the user's settings from the OS, but Maple just presents me with a glaring white screen and tiny little text.

I'm using Maple 8 as a remote application hosted on the university's system (using Citrix), and displaying on a Win 2K machine. Is the remote display the problem? Would it work if I simply went and bought a copy for myself? I don't mind spending the money if it will work, but not just to discover that it doesn't, you know?

7.3.4 Dr Francis J. Wright (16.2.03)

On a Windows XP Home stand-alone system the following works for Maple 8. It's mainly a question of configuring Windows rather than Maple.

Open the Windows Display Properties dialogue, e.g. right click on the desktop. Select the Appearance tab and then the Advanced button. From the Item drop-down menu select Window and then select the colour to be Black. Then select the Font colour to be White.

The main Maple 8 window respects this colour scheme, but the Maple Format/Styles... dialogue box doesn't and so is unusable. But, in principle and when using the default colour scheme, one can change most of the attributes of the text styles used by Maple and/or create new styles. So this should provide further control over the text used by Maple.

Whether this is possible on Win 2K and/or Citrix I can't say.

7.4 bar-chart - histogram (14.1.02)

7.4.1 jrc

I can't find an example or topic in 'Help' for Maple 7 (student version) which will directly produce a common bar-chart. I conclude reluctantly that Maple does not have this functionality. Is this correct?

7.4.2 Robert Israel (15.1.02)

You might want to try BarChart from my Maple Advisor Database, `http://www.math.ubc.ca/~israel/advisor`

7.4.3 Dr. TANAKA, Kazuo (16.1.02)

In the following book, a procedure to produce a bar chart is described.

T.Oguni, Maple V and its actual usages (Science,Tokyo, 1997) p.137 (in Japanese)

```
>with(plots):
>with(plottools):
```

A procedure to make bar chart

```
> barchart:=proc(data::list(integer))
> local n,p,i;
> n:=nops(data);
> p:=i->[[i-1,0],[i-1,data[i]],[i,data[i]],[i,0]];
> PLOT(seq(POLYGONS(evalf(p(i))),i=1..n));
> end;
```

An example

```
> data:=[10,40,20,30];
> barchart(data);
```

7.4.4 Thomas Richard (16.1.02)

No, they're just called histograms. :-) When Help / Full Text Search is fed with "bar chart", it finds statplots,histogram (in the middle of the hit list). Look up that help page.

7.4.5 Michael Faia (24.1.02)

Here's a minimalist procedure that works, although it's a little labor-intensive:

```
> R := n -> (((3^n - 2^(n+1)) + 1) / 2) + 1;
> with(stats[statplots]):
> data2 := [Weight(2.5..3.5, R(3)), Weight(3.5..4.5, R(4)),
            Weight(4.5..5.5, R(5))]:
> histogram(data2, xtickmarks=3);
```

7.5 bell command (26.7.01)

7.5.1 Edward Collett (26.7.01)

More shocking is that there is no bell command! Something I had in BASIC 20 years ago. I think that this is because the folks at MAPLE don't do calculations that last more than 1 minute.....

7.5.2 Thomas Richard (1.8.01)

In the command line versions (preferable for long running calculations), you should use

```
printf("\a");
```

which emits the bell character at least on Unix and Windows. A more general approach might be to call an external beeper tool with system() or ssystem().

7.5.3 Carl DeVore (3.8.01)

To get a bell on Unix, do

```
system("echo \a");
```

I don't know what to do for Windows. You could probably write a trivial program in C and call it with system.

7.5.4 Helmut Kahovec (9.8.01)

Carl DeVore wrote: ...

Well, on a PC running MS Windows NT 4.0 and Maple6 the following works, i.e. beeps, too:

```
> system("echo \a");
                    0
```

7.5.5 Edward Collett (11.8.01)

Helmut Kahovec wrote: ...

Your Bell Command worked! Congratulations and many thanks! I used Windows 98 and MAPLE 6. Now, for an equally challenging problem. Do you know how to program a BELL command so that the bell sound continues with a time delay between bell sounds, e.g., 10 seconds?

7.5.6 Carl DeVore (13.8.01)

Helmut Kahovec wrote: ...

Works on Windows Me also, but (apparently, for me, at least) not on Window 95. I haven't tried, Windows 98 yet.

7.5.7 Helmut Kahovec (16.8.01)

Edward Collett wrote: ...

Well, I would do it that way:

```
> delay,maxcount:=2.0,10:

> while maxcount>0 do
    system("echo \a");
    t0:=time();
    while time()-t0<delay do end do;
    maxcount:=maxcount-1
  end do:
```

Of course you may change the maximum number of beeps ('maxcount') and the delay ('delay').

7.5.8 Carl DeVore (21.8.01)

Helmut Kahovec wrote: ...

The following way will beep until you kill it, and also uses real time rather than computation time.

```
> RepeatBeep:= proc(delay)
>   local t0;
>   do
>      t0:= iolib(25);
>      system("echo \a");
>      while abs(iolib(25)-t0) < delay do od
>   od
> end proc:
```

However, I did not propose this solution before because my experience is that if you overuse the system command, Windows, not just Maple, will crash, or become unstable. This, of course, applies to both of our solutions.

The "experience" I refer to is with Windows ME. Perhaps other Windows versions hold up better.

So I think that the best idea is to use a Windows "alarm" utility and use ONE system command to call it. Surely such an alarm utility is available for free somewhere.

7.6 besirk (27.6.02)

7.6.1 Gabriel Garcia

We are going to use the BESIRK code to solve this particular problem. BESIRK is an implementation of a semi-implicit Runge-Kutta method (see Schwalbe et al, 1996) and is much faster than the methods built in to dsolve/numeric that we used in another example. We read the code into Maple.

Does anyone know where I can get this BESIRK code?

7.6.2 Ross Taylor (18.7.02)

From

`http://www.clarkson.edu/~chengweb/faculty/taylor/maple/numerics/besirk/`

You need the files besirk and besirkhelp.mws, the latter shows how to set it up.

FYI: I am co-author of BESIRK.

7.7 beta distribution (2.11.95)

7.7.1 Vadim Kutsyy

I tried to let Maple to calculate moments of beta distributions:

```
> assume(r1>0);
> assume(r2>0);
> int(x^k*x^(r1-1)*(1-x)^(r2-1)/Beta(r1,r2),x=0..1);
                        1
                        /  k  (r1~ - 1)        (r2~ - 1)
                       |  x  x          (1 - x)
                       |  ------------------------------ dx
                       |           Beta(r1~, r2~)
                      /
                      0
> simplify(%);
                        1
                        /  (r1~ - 1 + k)        (r2~ - 1)
                       |  x              (1 - x)
                       |  ------------------------------- dx
                       |            Beta(r1~, r2~)
                      /
                      0
```

Maple even can not compute simple Beta integral:

```
> int(x^(r1-1)*(1-x)^(r2-1),x=0..1);
                          1
                          /
                         |   (r1~ - 1)        (r2~ - 1)
                         |  x          (1 - x)          dx
                         |
                        /
                        0
```

which is equal to `Beta(r1,r2)`.

If I assign that integral to be equal to `Beta(r1,r2)`, Maple will give this answer to the integral, but not if I change r1 by `(r1+1)` (which is nessary to compute moments).

7.7.2 Neil E. Berger 312-413-2139

And what is even worse is that

```
int(exp(-t)*t^(z-1),t=0..infinity)
```

is not even recognized as `GAMMA(z)`

The Beta function is defined in terms of the GAMMA function.

The real point is that Maple does not know about Beta except how to compute it if given floats, and also if given specific exact numbers in the form `Beta(x,y)` for which the GAMMA function definition makes sense and has an exact value. For an inside look do:

```
interface(verboseproc=2);
print(GAMMA);
print(`evalf/GAMMA`);
print(Beta);
```

This does not mean that Maple can recognize a given integral to be Beta or for that matter to be GAMMA. If the integration techniques were based on pattern matching this might be a simple fix, but my understanding is that they are not, and it is not.

7.7.3 Michael Monagan

Neil, you need to tell Maple that $z > 0$ to do this integral. I'm sorry that Maple doesn't give you a clue, it should, and it does in Release 4. In Release 3, if you do `infolevel[int] := 4;` then Maple will then tell you that it could not determine the sign of z and you will then know to `assume(z>0);` Here is the integral computed

```
    |\^/|     Maple V Release 3 (Simon Fraser University)
._|\|   |/|_. Copyright (c) 1981-1994 by Waterloo Maple Software and the
 \  MAPLE  /  University of Waterloo. All rights reserved. Maple and Maple V
 <____ ____>  are registered trademarks of Waterloo Maple Software.
      |       Type ? for help.
> assume(z>0);
> int(exp(-t)*t^(z-1),t=0..infinity);

                                    GAMMA(z~)
```

```
>The Beta function is defined in terms of the GAMMA function.
>
>The real point is that Maple does not know about Beta except how to
>compute it if given floats, and also if given specific exact ...
```

This is not true. There are integrals where Maple will return the result as a formula involving the Beta function. I don't have one handy, perhaps someone else can show an example. Unfortunately, Maple is has hidden this capability from you and I can't blame you for comming to this conclusion.

7.7.4 Robert Israel

It is (at least in Release 3) if you `"assume(z > 0)"`, which ensures convergence. It doesn't work to `"assume(Re(z) > 0)"`, nor does `int(exp(-t)*t^(x+I*y-1), t=0..infinity)` work with assumptions $x > 0$ and y real.

7.8 bi-polar coordinates (15.9.1995)

7.8.1 Robert Gragg

Does anyone have experience with bi-polar codinates in Maple?

7.8.2 Douglas B. Meade

I would like to mention that I recently submitted a Maple package for use with orthogonal curvilinear coordinate systems (including bi-polar coordinates) to the Share Library. This package, curvcoord, should be available in the near future. In the meantime, if anyone is interested, I'll be glad to share additional details.

I don't know what you want to be able to do with bi-polar coordinates, but it's possible my package might be of some use.

7.9 bibliography link (23.9.96)

7.9.1 Pierre-Nicolas Ratier

The following link contains a very large (468 entries) bibliography of publications about the Maple symbolic algebra system.

If you want to connect, you have to do :

First : connect to :

`http://www.netlib.org/bibnet/subjects/`

Second : At the following line, click on

```
file    bibnet/subjects/maple-extract.bib    <---
for     Maple symbolic algebra system
```

7.10 bigebra package ver. 0.01 (31.10.00)

7.10.1 Rafal Ablamowicz

On page `http://www.math.tntech.edu/rafal/` I have recently added a link to a Maple V Rel. 5.1 worksheet with "Bigebra" package ver. 0.01 (beta). This package requires two other packages "Cliff4" and "Cli4plus" (for computations with Clifford algebras) also available for downloading from the same page.

"Bigebra" package is a result of joint work between Bertfried Fauser, Universitat Konstanz, `http://kaluza.physik.uni-konstanz.de/~fauser/` and myself.

The package consisting of 32 procedures and two patches for `‘define/skeleton‘` and `‘define/multilinear‘` is still under development and expansion. More information on the package will be posted on the web page.

7.11 binary tree data structure (3.2.03)

7.11.1 Bruno Guerrieri

I am trying to use Maple to set up binary trees and work with them. Is there a person in the community who has already written the standard procedures for insertion, deletion of nodes etc.. and would not mind sharing?

In addition, although I do not want to go the route of C, C++, etc, the concept of pointer is especially useful when dealing with trees. Since speed is not paramount to what I am doing, is there a "construct", (be it artificial and to be refined in future versions), a definition-like "thing" that someone could set up at the beginning of the worksheet that would allow one to almost translate word for word certain C procedures such as node deletion in a tree, exploiting the recursive nature of the situation.

The construct would, of course, be the equivalent of the pointer concept `p->left or p->right` as one can see in the following example:

Here is an example of a C procedure

```
void printtree(node *p)
{ if (p !=3D NULL)
    { printtree(p->left);
      printf(........,p->count, p->pword);
      printtree(p->right);
    }
}
```

I tried the network package to plot some binary trees and that was like to trying to ride a wild bronco.

7.11.2 Carl Devore (15.2.03)

See `?examples,binarytree`

| Since speed is not paramount to what I am doing, ...

Yes, they are called modules (see `?module`), and as far as I can tell, there is no performance penalty for using them. The Maple syntax would be `p:-left rather than C's p->left`.

A skeleton would be

```
BinaryTree:= proc(...)
```

```
   module()
      export left, right, ...;
      left:= ...;
      right:= ...;
      ...
   end module
end proc;

p:= BinaryTree(...);
p:-left;
```

Note that a procedure that returns a module (as above) is far more powerful than just a module.

7.11.3 Dr Francis J. Wright (16.2.03)

You might like to take a look at `C:\Program Files\Maple8\ examples\binarytree.mws` (although the full pathname of this file may be different on your installation), which illustrates insertion, deletion, etc. There is also a similar implementation of linked lists described in the Maple 8 Advanced Programming Guide. These both use inert functions to represent nodes.

Alternatively, if you represented a node as a module (or record) then you could use the object-oriented syntax `node:-left` etc, which is as close as I think you will get to C pointer syntax.

7.12 binomial coefficients (11.1.03)

7.12.1 Brendan McKay

The following problem has no important application that I know of. It just occurred to me as something that would interest a few people in this group.

Say that a binomial coefficient `binomial(n,k)` is "non-trivial" if n and k are integers such that `2 <= k <= n-2`. The problem is to determine if (and how) a given number is a non-trivial binomial coefficient.

For example, if we are given

11159690566590580740354583612991667619792058478202676400

we would like to quickly recognise that it equals `binomial(187,92)`.

I suspect that considering the theory of binomial coefficients modulo a prime might be productive.

7.12.2 Ronald Bruck (14.1.03)

I think you will essentially have to factor the integer, or at least find the biggest prime factor of the integer. Your example factors as

```
2^4 * 3 * 5^2 * 7 * 11^2 * 13 * 17 * 31 * 37 * 53 * 59 * 61 * 97
    * 101 * 103 * 107 *109 * 113 * 127 * 131 * 137 * 139 * 149
    * 151 * 157 * 163 * 167 * 173 * 179 * 181
```

and since the next prime after 181 is 191, this means the n must be between 181 and 190, inclusive. Not a big range, and once you know n you can find k with a few trials.

Conversely, if you CAN write the integer as a binomial coefficient, you will have reduced the range (and if k is large, quite substantially reduced the range) that n can lie in. So this is essentially a factorization problem.

Note that the number of times a prime p divides a binomial coefficient is $N(n,p) - N(k,p) - N(k,n-k)$, where `N(n,p) = sum of floor(n/p^i), i = 1 to infinity` (really a finite sum).

Assuming `k <= n/2` (which we can surely arrange) this means that all primes p between k and n must appear in the factorization. The longest such list is `{97,101,103,107,...,181}` for your example, so I would start at k = 97 and work down.

7.12.3 Robert Israel (15.1.03)

Given a number like that, you might first try to factor it, using the "easy" option (because if it is binomial(n,k) with n not very large, its prime factors should all be fairly small). In your example, if your number is B, we get

```
> ifactor(B,easy);
      4           2           2
   (2)   (3)  (5)   (7)  (11)   (13)  (17)  (31)  (37)  (53)  (59)

         (61)  (97)  (101)  (103)  (107)  (109)  (113)  (127)  (131)

         (137)  (139)  (149)  (151)  (157)  (163)  (167)  (173)

         (179)  (181)
```

The largest prime factor is 181.

If `B = binomial(n,k)` with `(wlog) k <= n/2`, any prime between `n-k+1` and n inclusive will divide B. Assuming there are such primes, the most n could be is 190 (because 191 is the next prime after 181).

So there aren't too many possibilities for n. For any given n, we could use Stirling's approximation to determine k approximately and then quickly refine this to find which k will make the magnitude of `binomial(n,k)` close to that of B. In this case we must have `n >= 187`, else `binomial(n,floor(n/2))` is too small. And checking $n = 187$, we find indeed that $k = 92$ works.

So here's my proposed code. It will return the pair `[n,k]` if successful, an error if it can't factor B, or FAIL if it can factor B and determines there is no possible `[n,k]` (or no prime between `n-k and n`).

```
binomialfinder:= proc(B)
  local F,p,q,Q,Q1,n,t,k,B0;
  F:= ifactors(B,easy)[2];
  if hastype(F,name) then error "couldn't easily factor %1",B fi;
  p:= F[-1,1];
  q:= nextprime(p)-1;
  for n from p to q do
    Q:= ln(B) + 1/2*ln(2*Pi*n*t*(1-t)) + n*(t*ln(t)+(1-t)*ln(1-t)):
    Q1:= evalf(subs(t=1/2,Q));
```

```
    if Q1 > 0.01 then next # B too big
    elif Q1 > -0.01 then k:= floor(n/2)
    else k:= round(n*fsolve(Q=0,t=0..1/2))
    fi;
    B0:= binomial(n,k);
    if B0 >= B then
      while B0 > B do
        k:= k - 1;
        B0:= B0*(k+1)/(n-k);
      od
    else
      while B0 < B do
        k:= k + 1;
        if k > n/2 then B0:= infinity
        else B0:= B0*(k+1)/(n-k);
        fi
      od
    fi;
    if B0 = B then return [n,k] fi;
  od;
FAIL
end;
```

7.13 binomial (21.6.97)

7.13.1 Metha Kamminga

One of my students tried to solve a very simple equation with Maple(release 4).

```
binomial(8,2)=binomial(x,6)
```

Maple gave the solution $x = -3$ but by heart you can see it must be $x = 6$.

Can anybody explain me the structure of the binomialfunction in case of negative integers?

```
> binomial(8,2);
                                  28
> binomial(8,6);
                                  28
> GAMMA(9)/GAMMA(7)/GAMMA(3);
                                  28
> binomial(-3,6);
                                  28
> GAMMA(-2)/GAMMA(7)/GAMMA(-8);
Error, (in GAMMA) singularity encountered
> eq:=binomial(8,2)=binomial(x,6);
                    eq := 28 = binomial(x, 6)
> solve(eq,x);
                    RootOf(28 - binomial(_Z, 6))
> allvalues(%);
                           -3.000000000
> fsolve(eq,x);
                           -3.000000000
> fsolve(eq,x,1..10);
                            8.000000000
> plot(binomial(x,6),x=-10..10);
```

7.13.2 Petr Lisonek (23.6.97)

By the very definition of binomial coefficients, this is a polynomial equation and you must bring it to that form first.

```
> eq:=binomial(8,2)=binomial(x,6);

                          eq := 28 = binomial(x, 6)

> eq:=expand(eq);

           eq := 28 = 1/720 (x - 5) (x - 4) (x - 3) (x - 2) (x - 1) x

> solve(eq,x);

                                  1/2 1/2                          1/2 1/2
  8, -3, 5/2 + 1/2 (- 43 + 4 I 551   )   , 5/2 - 1/2 (- 43 + 4 I 551   )   ,

                               1/2 1/2                          1/2 1/2
     5/2 + 1/2 (- 43 - 4 I 551   )   , 5/2 - 1/2 (- 43 - 4 I 551   )
```

7.13.3 Jan-Moritz Franosch (23.6.97)

To find out the definition of the binomial-function in Maple type this

```
> interface(verboseproc=2);
> print(binomial);
```

and you will see that

```
binomial(n,k):=(-1)**k*binomial(k-n-1,k) for n<0, k>0
```

This is simply the formular

```
binomial(n,k)=n*(n-1)*...*(n-k+1) / k*(k-1)*...*1
for n<0.
```

The help-function `?binomial` does not tell the whole truth in this case.

A workaround would be to let fsolve only search for positive solutions:

```
> fsolve( binomial(x,6)=28, x=0..infinity );
  8.000000
```

7.13.4 Robert Israel (23.6.97)

The general definition (which works for any complex n, and nonnegative integer m) is `binomial(n,m) = n (n-1) ... (n-m+1)/m!`

Thus `binomial(-n,m) = (-1)^m binomial(n+m,m)`.

You might look at Graham, Knuth and Patashnik, "Concrete Mathematics", for more discussion and applications of this.

7.14 bizarre convert to piecewise (8.11.99)

7.14.1 Ron Grimmer

The command piecewise has come up in different ways recently and I would like to ask about my experience with this command.

I am attemping to teach a class on Maple and am perplexed about using the command convert with piecewise.

I have run into several problems when converting max to piecewise. In particular,

```
> f:=max(x^5,x^2);

                                  2   5
                        f := max(x , x )

> g:=convert(f,piecewise);

                        {     2
                        {    x             x <= 1
                   g := {
                        {    2    5
                        { 2 x  - x          1 < x
```

Several examples of this type were discovered. Even more interesting is that sometimes a convert could be done with the correct answer being returned but after a restart an incorrect answer would be returned. One can even oscillate between the correct and an incorrect answer.

This happens on the student version release 4 in our lab and also on my release 5.0 Professional version.

7.14.2 Robert Israel (10.11.99)

At first I thought this bug had been corrected in Release 5.1, but some experimentation revealed that it still happens (sometimes). The factor that it only happens sometimes would tend to indicate that it depends on the arbitrarily-chosen ordering of some set or expression. I've traced the bug into `'unitstep/simpout'`, which consistently does the following:

```
> `unitstep/simpout`(unitstep(x^2 - x^5), x);

     unitstep(x) + unitstep(-x) + unitstep(x - 1)
```

Here it seems "unitstep(x)" is supposed to be 1 if x > 0 and 0 if x <= 0. So unitstep(x^2 - x^5) should be 1 if x < 0 or 0 < x < 1, 0 otherwise, which could be

```
       unitstep(x) + unitstep(-x) + unitstep(1 - x) - 1
```

But Maple's answer is incorrect for x > 1.

On the other hand,

```
> `unitstep/simpout`(unitstep(x^5 - x^2), x);

     unitstep(x-1)
```

which is correct.

7.14.3 Herman Jaramillo (10.11.99)

I believe that this bug was fixed in version 5.1. Here is what I get:

```
f := max(x^5, x^2);

> g := convert(f,piecewise);

                          {  2
                          { x          x <= 1
                     g := {
                          {  5
                          { x          1 < x
```

7.15 black in color structures (24.10.01)

7.15.1 Dr. U. Kasper

I have some PLOT (not plot) structures with different COLORs and I am unable to give one of them the color black. What are the values (or the value) for RGB, HUE or HSV? Or is there an other option for this?

7.15.2 Theodore Kolokolnikov (25.10.01)

The trick I use is the following. Suppose that 'pic' is your plot structure, for example:

```
pic := plot(sin(x),x=0..10):
```

Then you can do this:

```
eval(subs(COLOUR=(()->NULL), pic));
```

This will turn everything in your picture black.

7.15.3 Robert Israel (25.10.01)

Black is`COLOUR(RGB,0,0,0)`. It doesn't have a HUE. In HSV it would be `COLOUR(HSV,h,s,0)` where h and s can be anything from 0 to 1.

You can get the RGB values for the named colours from the table `` `plot/colortable` ``. So, for example:

```
> `plot/colortable`[sienna];

    [.55686275, .41960784, .13725490]
```

7.16 block matrix operation functions (12.3.97)

7.16.1 Stephen M. Carr

Has anyone developed some procedures for handling block matrix operations? I.e., blocks may be names of matrix stuctures (assigned or not) and operations on block matrices result in matrix multiplication and addition of the block rows and columns? This would require manipulating matrices of matrices (arrays of arrays) in the correct fashion.

The `linalg[blockmatrix]` function just helps build up regular matrices from smaller blocks but doesn't preserve the block nature of the bigger matrix.

7.16.2 Jim Gunson (24.3.97)

As a result of a request received by MUG I wrote some routines last summer. They can be found on my web page:

`http://www.kwantlen.bc.ca/~sci/math/gunson/maple`

For those without net access I can email them. Comments and suggestions would be appreciated.

7.17 book on maple (22.2.00)

7.17.1 Jose M Lasso

I need more material to learn Maple, can someone advice about what book to buy??

PS: Maple: An introduction and reference/Michael Kofler is a good book?, shoud I buy it?

7.17.2 Renato Portugal (23.2.00)

My preference goes for

A. Heck, Introduction to Maple (2nd ed.)

and also for

R. Nicolaides, N. Walkington, Maple a comprehensive introduction

not forgeting the ‘‘Learning and Programming Guides’’.

7.17.3 Bill Bauldry (23.2.00)

In my opinion, Andre Heck's "Intro to Maple" from Springer is the best choice

7.17.4 Ricardo Mansilla (23.2.00)

There are a lot of book. I suggest you one of my prefered:

"Nonlinear Phisics with MAPLE for Scientist and Engineers"

7.17.5 Helmut Kahovec (23.2.00)

I'd definitely recommend [1], [2], and [3].

```
References:

[1] K.M. Heal et al., "The Maple Learning Guide" for Release 5, Springer Verlag New Yo

[2] M.B. Monagan et al., "Maple V Programming Guide" for Release 5, Springer Verlag Ne

[3] Andre' Heck, "Introduction to Maple", 2nd Ed., Springer Verlag New York 1996, ISBN
[down] [up] Antonio Molina   (24.2.00)
```

I suggest you:

```
CALCULOS MATEMATICOS POR ORDENADOR CON MAPLE V RELEASE 5
Autores: Eugenio Roanes Garcma y Eugenio Roanes Lozano
Editorial: Rubiqos-1860, S.A.
ISBN: 84-8041-112-0
Fecha: Abril 1999
```

Have a look at: `http://www.addlink.es/go.asp?page=/product/waterloo/book.htm`

7.17.6 Wilhelm Werner (24.2.00)

Koflers book indeed is a very good book! If you have left the introductory level you should also take Andre Heck's book (Springer Verlag) into serious consideration.

7.17.7 Paul Eakin (25.2.00)

Carl Eberhart has a very nice and free "maple handbook" on his home page `http://www.ms.uky.edu/~carl`

7.17.8 Paul H. Ware (3.3.00)

The book by Abell and Braselton `"Maple by Example"` is an excellent resource. It is certainly worth the price. If I didn't have it, I would buy it.

7.18 boolean evaluation (23.11.01)

7.18.1 PierLuigi Zezza

I am not able to understand the following behavior If I write

```
> h:= proc(x):
>     if type(x, realcons) then
>          if x<=Pi then x^2
>          else x-1
>          fi;
>     else
>          'h(x)';
>     fi;
> end:
```

I get the error message

```
 > h(2);
Error, (in h) cannot evaluate boolean: 2-Pi <= 0
```

While if I write

```
> h:= proc(x):
>     if type(x, realcons) then
>          if is(x<=Pi) then x^2
>          else x-1
>          fi;
>     else
>          'h(x)';
>     fi;
> end:
```

I get

```
> h(2);
                                  4
```

I thought that writing `if x<=Pi or if is(x<=Pi)` was exactly the same, clearly I was

wrong can someone explain me the difference

7.18.2 Robert Israel (26.11.01)

No, `"if x <= Pi" and "if is(x <= Pi)"` are not the same, and you have found a perfect example of this.

"if" by itself only does a very low-level Boolean evaluation of the condition. It can use `< and <=` to compare integers, fractions and floats but that's about it. It can't even handle algebraic constants, e.g. `if 2 < sqrt(10) then` ... will produce an error. Moreover, `if a=b then ...` with symbolic expressions only succeeds if a is literally the same as b, e.g. `if (x+y)^2 = x^2 + 2*x*y + y^2 then A else B fi` will return B.

"is" tries much harder to evaluate the condition, including the use of any assumptions that have been made.

7.18.3 Dr Francis J. Wright (28.11.01)

`x<=Pi` is just an expression that builds a data structure for some other Maple facility to interpret. If you use it in a boolean context (such as an "if" statement) then Maple implicitly applies `evalb` to it, but can only evaluate it to a boolean if both sides of the inequality are numerical. In your example, x has a numerical value (namely 2) but Pi is just a symbol; Pi only acquires a numerical value if a numerical evaluation function such as evalf is applied to it.

By contrast, the function "is" takes a more mathematical view and tries to perform the necessary simplification and evaluation. Here are some examples:

```
> x := 2:
> x<=Pi;

                              2 <= Pi

> evalb(x<=Pi);

                            2 - Pi <= 0

> evalf(x<=Pi);

                          2. <= 3.141592654
```

```
> evalb(evalf(x<=Pi));

                                  true

> is(x<=Pi);

                                  true
```

However, beware that comparing floating-point approximations is generally unreliable.

See pages 189-190 of my book "Computing with Maple" (http://centaur.maths.qmw.ac.uk/CwM/) for a bit more detail on this general topic.

7.19 booleans and plot statements (6.3.01)

7.19.1 Bruce M Hartley

I have a problem with a procedure which I want to plot.

Consider the following.

```
restart:with(plots):
PrPDPD:=proc(x)
local out, critical_value;
critical_value:=evalf(Pi/4.0);
if x > critical_value then
out:=0.0;
else
out:=1.0;
fi;
out;
end;

PrPDPD(0.5);
```

```
gives 1.0 OK

PrPDPD(0.8);
gives 0.0 OK.

but

plot(PrPDPD(x),x=0..1.0);
gives an error message

Error, (in PrPDPD) cannot evaluate boolean.
```

Can anyone tell me what is going wrong and how I can use a procedure containing boolean statements inside a plot command?

This is Maple release 5.5

7.19.2 Carl DeVore (7.3.01)

`plot('PrPDPD(x)', x= 0..1.0);`

Note that it is impossible to determine the truth of the expression `x < Pi/4` if x does not have a numerical value.

Thus, we need to delay the evaluation of the expression `PrPDPD(x)` until the numerical values are assigned. That's what the quotes do.

Your expression `PrPDPD(x)` already has a well-defined name in Maple, indeed in math in general. It is `Heaviside(Pi/4 - x)`.

| It is curious that if I put an equal sign instead of a less ...

Yes, that can be very annoying feature. I wish that Maple would fail with an error message for that one too. Perhaps there is a way to make it fail that I don't know about. When it doesn't fail, I think this behavior leads to erroneous computations.

Anyway, here's why it happens: Consider the expression `"x = 3"`, where x is an undefined variable. Is an undefined variable the same thing as a number? Of course not. So are they equal? No. Then they must be not equal.

Here's another way of looking at it: I give you two sentences, A and B:

```
A: "The symbol x is the same thing as the number three."

B: "The symbol x is less than the number three."
```

The first sentence makes sense, and is clearly false. The second sentence is nonsense.

7.19.3 Adri van der Meer (8.3.01)

If you call `plot(something,options)` then the first thing that happens is that Maple evaluates the arguments `"something"` and `"options"`.

Thus `PrPDPD(x)` is evaluated before the variable x is assigned a (numerical) value, and Maple cannot dicide if `x > critical_value`.

Now you can do two things:

(1) prevent Maple to evaluate PrPDPD(x) before x has a numerical value. Possibilities:

```
    plot('PrPDPD(x)',x=0..1.0);    or
```

```
plot(PrPDPD, 0..1.0);
```

(2) Alter the procedure so that it can handle the case that x doesn't have a numerical value:

```
PrPDPD:=proc(x)
  local out, critical_value;
  critical_value:=evalf(Pi/4.0);
  if not type(x,numeric) then RETURN('procname(args)')
  elif x > critical_value then out:=0.0
  else out:=1.0
  fi;
  out;
end;
```

Now `plot(PrPDPD(x),x=0..1.0);` results in the disired plot.

7.19.4 Symancyk, Daniel (8.3.01)

There are several things you can do to get what you want.

You can use single quotes to delay the evaluation of the if in your procedure.

```
plot('PrPDPD(x)',x=0..1.0);
```

You can use the name of your function along with the domain without the variable.

```
plot(PrPDPD,0..1.0);
```

You can avoid having to use the single quotes in another way if you use the piecewise command in your proc.

```
PrPDPDnew:=proc(x)
piecewise(x<Pi/4,1,x>=Pi/4,0)
end:
Then

plot(PrPDPDnew(x),x=0..1.0);
```

will work.

7.20 boundary conditions in pdsolve (23.11.98)

7.20.1 Tom Casselman

Well I thought I had a simple problem to solve using maple: a first order linear pde of a function of two variables: `n(x,t)`. This pde has the form:

```
dn/dt + v*dn/dx+n/tau=G0*Heaviside(t)
```

(the derivitives are in fact partial derivitives)

with boundary conditions: `n(x,0)=0 and n(0,t)=0`. Note v and G0 are positive constants.

When I invoke `pdsolve in R5 (Mac)` I get a solution with an unknown function of x and t:

```
_F1(t-x/v)
```

The form is not useful to me as I do not know `_F1`. How do I insert the above boundry conditions into pdsolve or is pdsolve strictly for symbolic solutions?

Should I avoid using pdsolve and use Laplace transforms instead?

7.20.2 Robert Israel (24.11.98)

pdsolve doesn't always find enough solutions for arbitrary boundary values, but in this case it does work. I assume you're interested in the solution for `t > 0 and x > 0`.

```
> alias(N=n(x,t)); assume(v>0, G0>0, t>0, x>0);
> de:= diff(N,t)+v*diff(N,x)+N/tau=G0*Heaviside(t);
> sol:=pdsolve(de,N);
                                                  x        t v - x
                sol := n(x, t) = G0 tau + exp(- -----) _F1(-------)
                                                tau v         v
```

For the boundary condition at `t=0`:

```
> eval(sol,{t=0,N=0});
                                          x
                      0 = G0 tau + exp(- -----) _F1(- x/v)
                                        tau v
```

```
> isolate(subs(x=-s*v,%), _F1(s));
                                                  G0 tau
                                      _F1(s) = - --------
                                                     s
                                                exp(---)
                                                    tau
```

Note that this determines `_F1(s) for s <= 0`.

For the boundary condition at `x=0`:

```
> eval(sol,{x=0,N=0});
                                      0 = G0 tau + _F1(t)
```

And this tells you `_F1(t) for t >= 0`.

7.20.3 Willard, Daniel, Dr. (24.11.98)

Try:

```
>with(PDEtools);
>DE:=diff(n(t,x),t)+v*diff(n(t,x),x)+n(t,x)/tau=G0*Heaviside(t);
> pdsolve(DE);
```

PDEtools is in shareware if you can't find it elsewhere (eg, in R4).

Yes, substitute your BC in the solution and solve for `_F1`. Basically you have two equations for it and both must be satisfied.

7.20.4 TANGUY Christian (24.11.98)

The general solution to your differential equation is indeed

```
    sol(x,t) := _F1(x-v*t) + G0*tau*(exp(t/tau)-1)*Heaviside(t)
```

Using the first boundary condition `n(x,0) = 0 implies _F1(x) = 0` for arbitrary x, so that `_F1=0`.

This gives the solution `G0*tau*(exp(t/tau)-1)*Heaviside(t)`.

Then you add another boundary condition `n(0,t) = 0` , which would lead to an impossibility, unless you assume $t < 0$.

I am wondering about your boundary conditions: aren't they too many for a first-order differential equation?

7.21 boundary problem (15.5.00)

7.21.1 Romek Kaminski

In Mathematica

```
DSolve[{y''''[x]==q,y[0],y[1]==0,y''[0]=0,y''[1]=0},y[x],x]
```

gives the answer

```
y[x]->l^3*q*x/24-l*q*x^3/12+q*x^4/24
```

I don't know how to write this boundary problem in maple.

7.21.2 Jan Zijlstra (16.5.00)

I think the following command will accomplish what you are looking for:

```
> eq:=(D@@4)(y)(x)=q;
  dsolve({eq,y(0)=0,y(1)=0,(D@@2)(y)(0)=0,(D@@2)(y)(1)=0},y(x));


                             (4)
                      eq := (D   )(y)(x) = q

                                4            3          3
                  y(x) = 1/24 q x  - 1/12 q l x  + 1/24 q l  x
```

7.21.3 Brian Kelly (16.5.00)

Try the following syntax:

```
dsolve( {(D@@4)(y)(x)=q,y(0)=0,y(1)=0,(D@@2)(y)(0)=0,(D@@2)(y)(1)=0},y(x));
```

Note the use of `()` **`versus`** `[]` and the use of `(D@@k)(y)(x)` to represent the k th derivatitive of y evaluated at x

7.21.4 Stanley J Houghton (22.5.00)

Just an observation but I am not sure if it helps.

Your earlier example had four arbitrary constants in the general solution and four initial conditions as a basis for their resolution.

The general solution to the two differential equations above has 8 arbitrary constants. Your initial conditions only provide 6 simultaneous equations.

Are two initial conditions missing?

7.21.5 Romek Kaminski (22.5.00)

I think the name boundary is better because it is a space unknow x, not time t. In another way $x = l$ is on another end of x axe, then $x = 0$ then name initial is not good.

I think the initial condition problems are more easy to solve then more complicated boundary problems. I think the problem is proper naturaly and has a solution in mathematica 3.0 In earlier version of mathematica we have only solutions for initial problems.

Sorry for my unwise answers but I am only a civil engineer and never studied mathematics.

The answers from mathematica are

```
y1[x]=(9/16*l^3*q*x-3/2*l*q*x^3+q*x^4)/24
y2[x]=(-l^4*q/64+17/64*l^3*q*x-3/8*l^2*q*x^2+l*q*x^3/8)/6
```

7.21.6 Stanley J Houghton (22.5.00)

I will refer this to MUG to seek advice on the number of initial conditions required to provide a solution without arbitrary constants. Your general solution still has eight arbitrary constants and yet there are only 6 initial conditions.

I would be interested to see the input to Mathematica for comparison.

7.21.7 Romek Kaminski (22.5.00)

For mathematica:

```
DSolve[{y1''''[x]==q,y2''''[x]==0,y1[0]==0,y2[l]==0,y1''[0]==0,
y2''[l]==0,y1[l/2]==y2[l/2],y1'[l/2]==y2'[l/2], y1''[l/2]==y2''[l/2],
y1'''[l/2]==y2'''[l/2]},{y1[x],y2[x]},x]
```

7.21.8 Stanley J Houghton (22.5.00)

Your translation from Mathematica omitted two initial conditions .. $Y1(0) = 0$ and $Y2(l) = 0$.

Unfortunately, when I added them, function dsolve failed to produce an answer, ie from

```
> dsolve({(D@@4)(y1)(x)=q,(D@@4)(y2)(x)=0,  # 2 Diff Equations
>        y1(0)=0,y2(l)=0,       # IC 1 and 2
>        (D@@2)(y1)(0)=0,(D@@2)(y2)(l)=0,   # IC 3 and 4
>        y1(l/2)=y2(l/2),                        # IC 5
>        (D@@1)(y1)(l/2)=(D@@1)(y2)(l/2),    # IC 6
>        (D@@2)(y1)(l/2)=(D@@2)(y2)(l/2),    # IC 7
>        (D@@3)(y1)(l/2)=(D@@3)(y2)(l/2)},   # IC 8
>        {y1(x),y2(x)});
```

I think it may be a fault in the way Maple tackles the problem but others may be able to advise on a work-round..

If I produce the general result and solve for the constants using the initial conditions, however, we do end up with the same solution as Mathematica. Thus

```
> dsol:=dsolve({(D@@4)(y1)(x)=q,(D@@4)(y2)(x)=0},
>        {y1(x),y2(x)});
```

now set up the functions y1 and y2, used to form the equations from the initial conditions

```
> y1,y2:=map(unapply,subs(dsol,[y1(x),y2(x)]),x)[];
```

now form the initial condition based equations

```
> IC:={y1(0)=0,y2(l)=0,
       (D@@2)(y1)(0)=0,(D@@2)(y2)(l)=0,
```

```
        y1(1/2)=y2(1/2),
        (D@@1)(y1)(1/2)=(D@@1)(y2)(1/2),
        (D@@2)(y1)(1/2)=(D@@2)(y2)(1/2),
        (D@@3)(y1)(1/2)=(D@@3)(y2)(1/2)};
```

now solve these for the constants

```
> sols:=solve(IC,{_C1,_C2,_C3,_C4,_C5,_C6,_C7,_C8});
```

now reform the functions y1 and y2

```
> y1,y2:=map(unapply,subs(sols,[y1(x),y2(x)]),x)[];
```

The resulting functions are exactly the same as you had from Mathematica.

7.21.9 Edgardo S. Cheb-Terrab (30.5.00)

The reason for why Maple failed is because some of the initial conditions (ICS) - those you numbered 6, 7 and 8 - are more like constraints than ICs, in that you are not saying what the value of the `{y1,y2}` or their derivatives are. The logical interpretation of giving an IC like

```
   > (D@@1)(y1)(1/2)=(D@@1)(y2)(1/2)
```

then would be

```
   > (D@@1)(y1)(1/2) = k
   > (D@@1)(y2)(1/2) = k
```

plus "adjust k to what would be necessary" in order to solve the problem. It is good that these emails are here so that next Maple can implement this.

This functionality is also available in Maple for releases 4, 5 and 6, if you install ODEtools (`http://lie.uwaterloo.ca/odetools.htm`); the ODE problem posted will then be solved as follows:

In Maple 4 use 'odsolve' instead of 'dsolve'

```
> dsolve({(D@@4)(y1)(x)=q,(D@@4)(y2)(x)=0,    # 2 Diff Equations
          y1(0)=0,y2(1)=0,                    # IC 1 and 2
          (D@@2)(y1)(0)=0,(D@@2)(y2)(1)=0,    # IC 3 and 4
```

```
        y1(l/2)=y2(l/2),                    # IC 5
        (D@@1)(y1)(l/2)=(D@@1)(y2)(l/2),    # IC 6
        (D@@2)(y1)(l/2)=(D@@2)(y2)(l/2),    # IC 7
        (D@@3)(y1)(l/2)=(D@@3)(y2)(l/2)},   # IC 8
        {y1(x),y2(x)});

                  3           2  2   17     3              4
{y2(x) = 1/48 q l x  - 1/16 q l  x  + --- q l  x - 1/384 q l ,
                                      384

                    4             3            3
     y1(x) = 1/24 q x  - 1/16 q l x  + 3/128 q l  x}
```

7.22 boundary value problem, to solve (13.10.95)

7.22.1 Ulrich Klein

We have a problem in solving a differential equation. We have a differential equation, say de and an initial value condition: $y(0) = 0$ and a boundary value condition: `D(y)(20)=0`, eg.

If we try to solve this equation using the dsolve command, Maple complains about the second condition. Besides, we use the numeric option of the dsolve command.

Do you know a way to solve such an equation with Maple?

7.22.2 Douglas B. Meade (16.10.95)

Maple's numeric dsolve is, as the on-line help describes, of use only for initial value problems. You are asking for the numerical solution of a two-point boundary value problem.

While you do not provide enough information to know exactly what type of ODE you are attempting to solve, I might have something that will be of interest.

I have recently developed a Maple implementation of the simple shooting method.

This procedure works very much like `dsolve[numeric]`. It must be used with caution, as the numerical method is not always stable. A paper introducing this procedure (shoot) and demonstrating its usage on three problems will appear in the upcoming Special Issue of MapleTech. (These examples are nonlinear and some involve systems.)

The on-line help for shoot, which includes two (simple) examples, is attached to the end of this message.

Anyone interested in this paper and/or procedure should send mail to: Douglas B. Meade

```
    |\^/|     Maple V Release 3 (University of South Carolina)
._|\|   |/|_. Copyright (c) 1981-1994 by Waterloo Maple Software and the
 \  MAPLE  /  University of Waterloo. All rights reserved. Maple and Maple V
 <____ ____>  are registered trademarks of Waterloo Maple Software.
      |       Type ? for help.
> read shoot;
> ?shoot
FUNCTION: shoot - shooting method for two-point boundary value
```

```
                problems

CALLING SEQUENCE: shoot(deqns, vars, bc, init, options)

PARAMETERS:
          deqns   - ordinary differential equations in vars
          vars    - variables to be solved for
          bc      - boundary conditions
          init    - initial values for shooting method
          options - optional arguments

SYNOPSIS:
- shoot uses the shooting method to compute approximate solutions
  to two-point boundary value problems for a first-order system
  of ordinary differential equations using the shooting method.
  The default output of this procedure is a list of procedures,
  as generated by numeric dsolve (see ?dsolve,numeric)

- Each of the first four arguments must be specified as a single
  equation or a list or set of equations (very much like the
  arguments to dsolve).

- The first two arguments are exactly the same as for dsolve.
  That is, the first argument contains the differential equations
  and initial conditions. The second argument contains the
  dependent variable(s) in the problem.

- The initial conditions will include both the boundary values
  at one endpoint and the auxiliary initial conditions needed
  for the shooting method (but not the initial conditions for
  the sensitivity equations). The initial values for the auxiliary
  conditions will be given in terms of the shooting method
  parameters. There should be an equal number of differential
  equations and initial conditions.

- The the third argument contains the boundary condition(s) -- as
  a single equation or as a list or set of equations -- at the
  second boundary point. These are the control equations for the
  iterations.
```

```
- The final required argument is a single equation or list or
  set of equations containing the parameter values for the
  first iteration of the shooting method.

- Any optional arguments must be in the form of keyword=value.
  These can be any valid optional argument to dsolve, or one of
  the following (with default settings listed in parentheses):
            'bcerr'    = bctol          (Float(1,2-Digits))
            'iterr'    = ittol          (Float(1,2-Digits))
            'maxiter'  = maxit          (10)
            'itmethod' = itmeth         ('newton')
  where:
   - bctol is the maximum acceptable L2-error between the computed
     and exact boundary conditions;
   - ittol is the maximum acceptable L2-error between successive
     parameter estimates;
   - maxiter is the maximum number of iterations
   - itmeth is the root-finding method used to compute new
     parameter estimates (only Newton-Raphson is available)

- The only known option to dsolve that is not supported is
  output=listprocedure. However, THERE IS NO GUARANTEE THAT ALL
  COMBINATIONS OF OPTIONS WILL WORK AS EXPECTED.

- Each iteration of the shooting method begins by solving (by
  numeric dsolve) the initial value problem using the most recent
  set of shooting method parameters. Each boundary condition is
  then evaluated. If these conditions are not satisfied to the
  requested tolerance, one iteration of the Newton-Raphson method
  is used to generate a new set of shooting method parameters.

- Additional information about each iteration can be obtained
  via the infolevel command. To see each set of parameters,
  use infolevel[shoot]:=1:. To include the error at the
  boundary, use infolevel[shoot]:=2:. Use infolevel[shoot]:=3:
  to see the full system of differential equations involved in
  the iterations. To reset to normal usage, use infolevle[shoot]:=0:.
```

```
- For more information on the shooting method, see Stoer and
  Bulirsch (p. 469) or other standard numerical analysis texts.

EXAMPLES:
> restart;
> with(plots):
> read shoot;
--------------------------------------------------------------------------------
# Example 1: x'' + x = cos(t), x(1)=3, x'(3)=1
> ODE := { diff( x(t), t ) = y(t), diff( y(t), t ) = -x(t)+cos(t) };

                        d                   d
            ODE := {---- x(t) = y(t), ---- y(t) = - x(t) + cos(t)}
                       dt                  dt
> FNS := { x(t), y(t) };

                                FNS := {x(t), y(t)}
> IC := { x(1)=3, y(1)=alpha };

                           IC := {x(1) = 3, y(1) = alpha}
> BC := y(3)=1;

                                  BC := y(3) = 1
> S := shoot( ODE union IC, FNS, BC, alpha=0 );

  S := [t = proc(t) ... end, x(t) = proc(t) ... end, y(t) = proc(t) ... end]

> odeplot( S, [t,x(t)], 1..3 );
--------------------------------------------------------------------------------
# Example 2: x''+x = cos(t), x(1)=3, x(3)+x'(3)=1
> infolevel[shoot]:=1:
> S := shoot( ODE union IC, FNS, x(3)+y(3)=1, alpha=0, bcerr=Float(5,-7) ):
shoot:   New parameter values:    alpha = 0
shoot:   New parameter values:    alpha = 12.08987391
shoot:   New parameter values:    alpha = 12.08987493

> infolevel[shoot]:=0:
> odeplot( S, [ [t,x(t)], [t, x(t)+y(t)] ], 1..3, 0..20, labels=[x,t] );
# The exact solution is not hard to find for this problem:
```

```
> X := rhs( dsolve( { diff(x(t),t$2)+x(t)=cos(t), x(1)=3 }, x(t), method=laplace ) ):
> Xp := diff(X,t):
> c := solve( subs(t=3,X+Xp)=1, {D(x)(1)} ):
# Note that the final value of the shooting parameter is correct
# to 5 digits to the right of the decimal point
> evalf(c);

                              {D(x)(1) = 12.08987954}
> X := subs(c,X): evalf(X);

       .4207354924 (t - 1.) cos(t - 1.) + .2701511530 (t - 1.) sin(t - 1.)

            + 11.66914405 sin(t - 1.) + 3. cos(t - 1.)
# The results are very comparable when viewed graphically
> p1 := odeplot( S, [t,x(t)], 1..3, style=point ):
> p2 := plot( X, t=1..3, color=red ):
> display( {p1,p2} );
```

SEE ALSO: `dsolve[numeric]`, `userinfo` PS: It worked very well for our problem. U. Klein

7.23 bracket as execution group delimiter (3.12.02)

7.23.1 A. Prashanth

Got this basic question about how the bracket on the maple worksheet could be made longer to accommodate more lines of the code. each command should end with a semi-colon and the moment the 'enter' button is pressed the command is implemented with the prompt shifting to the next bracket.

i wish to stay on inside a single bracket and build all my code there and not go on to the next new prompt (on a new bracket).

i have used quite a few commands of maple 7 but as i need now that all my nested for loops be run in one go, i'll need to fix the problem i just described.

7.23.2 Stanley J Houghton (4.12.02)

You can use shift-enter to move down a line without execution .. eventually executing by pressing enter when pointing somewhere within the expression sequence you typed.

You can also join an execution group to the next group by pressing f4 on a windows implementation

7.23.3 Joe Riel (4.12.02)

If you use shift-enter, you get a new line without a prompt (the previous line is not sent to the Maple engine). Hitting enter after a series of shift-enters sends all the lines to the Maple engine.

If you do much programming in Maple you will want to do so in a separate text editor, one in which you are comfortable, and then read the resulting file into Maple (see ?read). Emacs is a powerful programming text editor, freely available on almost any platform. There is a maplev-mode available for Emacs at my home page `www.k-online.com/~joer`, it understands Maple syntax and has many useful features.

7.23.4 Dr Francis J. Wright (6.12.02)

The Enter key executes the code in the current execution group. You need to use Shift-Enter, which just gives a new line; it extends the execution group delimiter and does not give a new prompt. It is particularly important to include loop initialisation within the same execution group as the loop code, since re-executing a loop without also re-executing its initialisation usually gives the wrong result and can be quite confusing!

7.24 braille for mathematics (28.1.00)

7.24.1 Rafal Ablamowicz

I apologize for not asking a typical MUG-like question, but this group may know an answer to the following problem.

At Tennessee Tech we have a blind student who attends mathematics classes. A student volunteer takes written notes for that blind student which are later scanned and converted into a Word document. My special education specialists are looking for a converter of a mathematics text to Braille: they have a Braille printer but apparently it cannot read and print mathematics symbols in Braille.

Short of using TeX, mathematics can be very nicely typed up in Maple. Does anyone have any experience with printing mathematics text, possibly created with Maple, in Braille?

I would appreciate any information you might have on an appropriate software that would enable a Braille printer recognize and print mathematics symbols.

7.24.2 Chris Augeri (3.2.00)

This is a start, I just used

www.google.com

terms:braille mathematics notation

`http://www.rdcbraille.com/nemeth.html`

`http://www.tsbvi.edu/math/math-resources.htm`

`http://www.braille.org/papers/unive/unive.html`

`http://www.w3.org/TR/1998/REC-MathML-19980407/chapter1.html`

7.24.3 Bob Gotwals (7.2.00)

More braille sites:

Online instruction program, INCLUDING Nemeth Mathematics:

`http://www.brl.org`

7.24.4 Steven Sahyun (15.2.00)

In regards to Braille and mathematics:

The Science Access Project at Oregon State University has developed a program to aid Blind students with math and science. The program's name is TRIANGLE and it is a mathematics scratchpad designed for blind users.

It has editor buffers to type and edit math and science equations, a built in graphing calculator (the graphs are visual as well auditory), a table viewer, and a figure viewer. It is a DOS program and supports voice screen readers, Braille displays, as well as prints to standard Braille printers. The program is free.

We are also working on a self-voicing calculator program for Win95/98 that offers more capabilities than that of the DOS version.

For more information about these or other projects we are working on, please visit our web site at: `http://dots.physics.orst.edu`

or contact me

`http://dots.physics.orst.edu/triangle.html`

7.25 bspline in maplev.6 (20.2.01)

7.25.1 Oliver Schwarz

Is there a way to access the old bspline-package from Release 4 in Release 6?

7.25.2 Jason Schattman(21.2.01)

The old bspline-package from Release 4 is available for download from the Maple Application Center (`www.http://www.mapleapps.com/`) under the category "Numerical Analysis".

7.25.3 Robert K Wright (21.2.01)

Although I don't have release 6, I understand that the bspline package is gone in that release. Earlier in the year a request like yours appeared, and I made publicly available a bspline routine that I wrote for my own research and teaching. You can get it at:

`http://www.emba.uvm.edu/~wright/mapleprgs/makebsp`

7.26 bspline (13.6.00)

7.26.1 Don Ramirez

What has happen to bspline in Maple 6?

```
readlib(bspline);
Error, could not find `bspline` in the library
```

7.26.2 Robert Israel (23.6.00)

Good question. It seems to have disappeared without a trace (not even a mention on the `?updates,Maple6,compatibility` help page).

One thing I noticed by looking at the code for bspline in Release 5.1 is that it uses the totorder package, which has been discontinued (but I don't think it would be too hard to modify the code to use assume rather than totorder).

7.26.3 Robert K Wright (11.7.00)

Below is a link to a b-spline generator I wrote that doesn't depend on Maple's built-in routine. It also uses Carl de Boor's recursive formulas rather than the definition in terms of divided differences of a truncated power function, and thus doesn't require a-priori ordering of the knots.

`http://www.emba.uvm.edu/~wright/mapleprgs/makebsp`

Hope this is useful to others out there:

7.27 bug affecting piecewise in maple v.5 (22.2.99)

7.27.1 Robert Michael Sinclair (PHj)

Can anyone explain the following? The first result is fine, but the second is blatantly wrong (the only difference is that the second function has "sin(x)/x" instead of "sin(x)" for x>0):

```
    |\^/|     Maple V Release 5 (WMI Campus Wide License)
._|\|   |/|_. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.

> piecewise(x>0,sin(x), x=0,1, x<0,sin(x)/x);
                              { sin(x)        0 < x
                              {
                              {   1           x = 0
                              {
                              { sin(x)
                              { ------        x < 0
                              {   x

> diff(%,x);
                         { cos(x)   sin(x)
                         { ------ - ------        x < 0
                         {   x         2
                         {            x
                         {
                         {    undefined           x = 0
                         {
                         {     cos(x)             0 < x

> piecewise(x>0,sin(x)/x, x=0,1, x<0,sin(x)/x);
                              { sin(x)
                              { ------        0 < x
                              {   x
                              {
                              {   1           x = 0
```

```
                              {
                              { sin(x)
                              { ------        x < 0
                              {   x

> diff(%,x);
                            { undefined          x = 0
                            {
                            {    0            otherwise
```

It is corrected with Maple 6. (U. Klein)

7.27.2 Preben Alsholm (24.2.99)

The bug appears to be in `'convert/piecewise'`. The procedure `'diff/piecewise'` converts the expression to Heaviside before it (correctly) differentiates the expression.

The result is then converted to piecewise, the result of which is wrong (in your last example).

Here is an imitation of what is going on:

```
> restart;
> p:=piecewise(x>0,sin(x)/x, x=0,1, x<0,sin(x)/x):
> pH:=convert(p,Heaviside);

                                sin(x)   sin(x) Dirac(x)
              pH := Dirac(x) + ------ - ---------------
                                 x             x

> pHd:=diff(pH,x);

                         cos(x)   sin(x)   cos(x) Dirac(x)
  pHd := Dirac(1, x) + ------ - ------ - ---------------
                         x         2            x
                                  x

            sin(x) Dirac(x)   sin(x) Dirac(1, x)
          + --------------- - ------------------
                   2                  x
```

```
                x

> convert(pHd,piecewise);

                    { undefined          x = 0
                    {
                    {     0            otherwise

> simplify(pH); #OK

                                 sin(x)
                                 ------
                                   x

> simplify(pHd); #OK

                            cos(x) x - sin(x)
                            -----------------
                                    2
                                   x
```

7.28 bug in ellipticf(z, k), maple v.5 to maple 8 (30.6.02)

7.28.1 Vladimir Bondarenko

According to Maple Help

The incomplete elliptic integral EllipticF is defined by

```
EllipticF(z,k) = int(1/sqrt(1-t^2)/sqrt(1-k^2*t^2),t=0..z)
```

Translation into Mathematica yields

```
Integrate[1/Sqrt[1-t^2]/Sqrt[1-k^2*t^2],{t,0,z}]

EllipticF[ArcSin[z], k^2]
```

Therefore, we can make the following comparison.

Maple 8	Mathematica 4.2
EllipticF(z,k)	EllipticF[ArcSin[z], k^2]

Let us substitute concrete values.

These outputs coinside:

evalf(EllipticF(1/2,1/2),20);	N[EllipticF[ArcSin[1/2], 1/4], 20]
.52942862705190581774	0.52942862705190581774

Another perfect agreement:

evalf(EllipticF(I,1/2),20);	N[EllipticF[ArcSin[I], 1/4], 20]
0.+.85122374907118540906*I	0.85122374907118540906 I

Yet another perfect agreement:

evalf(EllipticF(1,I),20);	N[EllipticF[ArcSin[1], -1], 20]

```
1.3110287771460599052                1.3110287771460599052
```

However, these outputs differ

```
evalf(EllipticF(I,I),20);            N[EllipticF[ArcSin[I], -1], 20]

0.+1.3085903338656260177*I           1.3110287771460599052 I
```

Question 1) Which of them is correct? (my idea is, the Mathematica's one)

Question 2) How to prove your answer to 1) in an easy and elegant way? Naturally, one can use the AGM idea but this looks somewhat tedious (me, lazy...)

7.28.2 Robert Israel (4.7.02)

Yes, Mathematica is correct. Note this:

```
> restart;
  evalf(EllipticF(I,I),20);
  evalf(EllipticF(I,I),30);

          0.+1.3006012664935504810 I
          0.+1.303488620041800288360308325 86 I
```

So Maple isn't even consistent with itself. It seems that whatever numerical method Maple is using for `EllipticF` works very poorly in this case, which is right on the circle of convergence of the power series. Note that `EllipticF(I*t,I)` should be `I*EllipticF(t,I)` for `|t| <= 1`: I think they are different for `|t| > 1` because different branches are chosen.

7.28.3 Christian Hoffmann (5.7.02)

Just a quick hint:

Have look at the arithmetic-geometric mean (AGM) and its connection to elliptic integrals. Certain elliptic integrals obey an invariance relation involving the AGM. A good book on this is

```
Borwein, J.M., Borwein, P.B., 1987. Pi and the AGM: A study in analytic number theory
```

```
Borwein, J.M., Borwein, P.B., 1984. The arithmetric-geometric mean and fast computatio
```

7.28.4 Carl Devore (9.7.02)

I trust Maple's numerical integration; I think that is the most well-written part of Maple. We can define the `EllipticF` function so that it is numerically evaluated via numerical integration:

```
> EllF:= (z,k)-> evalf(Int(1/sqrt(1-t^2)/sqrt(1-k^2*t^2), t= 0..z);
```

Now `EllF(I,I)` will correspond with the Mathematica answer, so I think the Mathematica is correct.

7.28.5 Jim FitzSimons (23.7.02)

This is my correct answer. Using Bill Carlson's methods.

```
LOAD("carlson")

ELLIPTICF(#i,#i)

;Approx(#2)
1.31102877714605990523241979494555970684137747571581158140841085 1*#i

1.3110287771460599052 I
```

I think Maple is wrong.

7.29 bug in 1/(u!) function, maple 7 (7.3.02)

7.29.1 Adam Millican-Slater

This message supercedes previous message with a much more complicated function.

On maple 7 after defining the function; `c:=(u)->evalf(1/(u!)); with Digits:=40;` I have managed to get two different answers when evaluating this function at `u=4999`.

If I define `m:=10000 and n:=9999` and then evaluate, first `c(m/2) then c((n-1)/2)` I get the same answer.

If I then evaluate `c(m/2) and c((n-1)/2)` again I get results which are again equal, but different from the intial result. This can be repeated to obtain further different results.

This problem can be avoided by evaluating `c(evalf(m/2)) and c(evalf((n-1)/2))`.

This problem does not seem to exist in Maple 6.

It is corrected with Maple 8 (U. Klein)

7.29.2 Stephen Forrest (8.3.02)

Yes, there's a known problem with factorials in Maple 7 that has nothing to do with evalf or Digits; I think your problem is a manifestation of this. The bug was reported on `comp.soft-sys.math.maple` a while back, and you can see with this example from Maple 7:

```
> 5000!/4999!;
                        1
```

versus, in Maple 6 and previous, the correct

```
> 5000!/4999!
                      5000
```

7.29.3 Dr Francis J. Wright (11.3.02)

Here are a few general comments that might be relevant.

If I remember correctly, `n!` behaves asymptotically something like `n^n`, so `1000!` has about 3000 digits and you are likely to see different numerical errors in you use different floating-point approximations with less than about 3000 significant digits.

If you use a floating-point operand with `!` then presumably Maple evaluates it as a gamma function (since factorial proper is defined only for non-negative integers) and so runs completely different code.

The factorial implementation in Maple 7 is much more sophisticated than that in Maple 6, one consequence of which is that Factorial is used as a function name in Maple 7 and so is protected, which breaks (trivially) an example in my book!

7.30 bug in [detools,dchangevar] in maple v.4 (8.9.97)

7.30.1 Georges Thomas

```
> with(DEtools,Dchangevar):
> de := 'diff'(sin(x)*diff(f(x),x),x)/sin(x) ;
> collect(simplify(Dchangevar({x = arccos(y), f(x)=F(y)}, de, x,y)),diff) ;
```

Maple R4 answers:

```
        -y*diff(F(y),y)  + (1-y^2)*diff(F(y),y$2) ;
```

I expected:

```
         -2*y*diff(F(y),y)+ (1-y^2)*diff(F(y),y$2) ;
```

In the online help about Dchangevar, the first example in "Examples/Additional examples" gives:

```
> de := diff(y(x),x$2) = y(x)*diff(y(x),x)/x ;
> Dchangevar({x=exp(t),y(x)=Y(t)},de,x,t) ;
```

Maple answers, after some simplication:

```
            diff(Y(t),t$2) = Y(t)*diff(Y(t),t) ;
```

Answer expected:

```
            diff(Y(t),t$2) = (Y(t)+1)*diff(Y(t),t) ;
```

It seems that Dchangevar mistakes if the order of the differential equation is `>=2` and if the change of variable is not linear.

Is there a bug in Maple or in my mind?

The bug is removed in Maple V Release 5. (U. Klein)

7.30.2 Edgardo S. Cheb-Terrab (9.9.97)

As an alternative, you can use `PDEtools[dchange]`, which is able to perform non-linear changes of variables and also in PDEs. In your example,

```
> de := 'diff'(sin(x)*diff(f(x),x),x)/sin(x) ;
> tr := {x = arccos(y), f(x)=F(y)};

                tr := {x = arccos(y), f(x) = F(y)}

> normal(dchange(tr,de));

             / 2      \                      / 2      \
             |d       |  2     /d      \     |d       |
            -|--- F(y)| y  - 2 |-- F(y)| y + |--- F(y)|
             |  2     |        \dy     /     |  2     |
             \dy      /                      \dy      /
```

PDEtools is a package distributed in the Share Library of Maple R4. The last version of PDEtools is found at

`http://lie.uwaterloo.ca/pdetools.html` (Canada)

or

`http://dft.if.uerj.br/pdetools.html` (Brazil)

7.30.3 Preben Alsholm (10.9.97)

I get the same as you do! There must be a bug in Dchangevar.

7.30.4 Juergen Boehm (11.9.97)

I checked the variable transformation with a procedure of my own (vsubs) and could confirm the supposed results of G. Thomas and also reproduce the wrong answers that Dchangevar gave :

```
> vsubs([y],[x=arccos(y)],[f(x)=F(y)],diff(sin(x)*diff(f(x),x),x)/sin(x));

/
|
|                     2 1/2 /d      \
```

```
|-cos(arccos(y)) (1 - y )    |-- F(y)| -
|                            \dy     /
\

                             //d      \                          \
                             ||-- F(y)| y              / 2      \|
                         2 1/2 |\dy     /          2 1/2 |d       ||
    sin(arccos(y)) (1 - y )    |----------- - (1 - y )    |--- F(y)||
                             |      2 1/2               |  2     ||
                             \(1 - y )                  \dy      //

    \
    |
    |
    |/sin(arccos(y))
    |
    /

> simplify(%);

              / 2      \                      / 2      \
              |d       |  2     /d      \     |d       |
             -|--- F(y)| y  - 2 |-- F(y)| y + |--- F(y)|
              |  2     |        \dy     /     |  2     |
              \dy      /                      \dy      /

> with(DEtools,Dchangevar);

                             [Dchangevar]

> Dchangevar({x=arccos(y),f(x)=F(y)},diff(sin(x)*diff(f(x),x),x)/sin(x),x,y);

/
|                      2 1/2 /d      \
|-cos(arccos(y)) (1 - y )    |-- F(y)|
|                            \dy     /
\

                             / 2      \\
```

```
                              2  |d       ||
    + sin(arccos(y)) (1 - y ) |--- F(y)||/sin(arccos(y))
                              |  2     ||
                              \dy      //

> simplify(%);

                             / 2      \   / 2      \
             /d      \       |d       |   |d       |  2
            -|-- F(y)| y + |--- F(y)| - |--- F(y)| y
             \dy     /       |  2     |   |  2     |
                             \dy      /   \dy      /

> vsubs([t],[x=exp(t)],[y(x)=Y(t)],diff(y(x),x,x)-y(x)*diff(y(x),x)/x);

       /                              / 2      \\
       |         /d      \            |d       ||
exp(-t) |-exp(-t) |-- Y(t)| + exp(-t) |--- Y(t)||
       |         \dt     /            |  2     ||
       \                              \dt      //

                   /d      \
      Y(t) exp(-t) |-- Y(t)|
                   \dt     /
    - ----------------------
              exp(t)

> simplify(%);

                                  / 2      \
          /d      \               |d       |
-exp(-2 t) |-- Y(t)| + exp(-2 t) |--- Y(t)|
          \dt     /               |  2     |
                                  \dt      /

           /d      \
    - Y(t) |-- Y(t)| exp(-2 t)
           \dt     /
```

```
> Dchangevar({x=exp(t),y(x)=Y(t)},diff(y(x),x,x)-y(x)*diff(y(x),x)/x,x,t);
>
```

$$\frac{\frac{d^2}{dt^2}Y(t)}{e^{(2t)}} - \frac{Y(t)\left(\frac{d}{dt}Y(t)\right)}{e^{(2t)}}$$

7.30.5 Yuri Muzychka (12.9.97)

I have noticed the bug in DEtools a long time ago. I had sent Maple a worksheet summarizing where the bug occurs, but never received any follow up.

If I recall, what I had observed was that DEtools did not know how to apply the chain rule properly.

I discovered this while trying to get maple to perform the same set of calculations I had done on paper. Had the change of variable worked properly, I would have reduced my ODE to one which has the hypergeometric function as a solution. Otherwise, Maple could not solve it. I was trying to reproduce some work I had found in a paper I was using for my research.

7.31 bug in ‘evalf/ellipticf‘, maplevr5, maple 6 and maple 7 (21.8.00)

7.31.1 Vlad Dyadichev

I think there is a bug in ‘evalf/EllipticF‘ function over the complex numbers (at least in MapleVR5). Consider the following input:

```
>Z:=int(sqrt(1+x^4)-x^2,x=a..b):
```

(The result is rather long expression with elliptic integral of the first kind functions; the second term was added to produce O(1) result for large (a-b)). Now consider, for example

```
>Digits:=30:
>evalf(eval(Z,{a=0,b=100}));

  1.230884231340394930190059037 31 -

        .000089255864456015125306340931936 6 I
```

The error is obvious — the integral of real valued function along the real axis should be real. Moreover, when the ‘int‘ is invoked with real limits or symbols ‘assume‘d to be real it gives manifestly real answer (probably correct).

7.31.2 Preben Alsholm (30.8.00)

Well, it gets more and more curious:

```
> restart;
> assume(a<1,b>1);
> int(sqrt(1+x^4)-x^2,x=a..b);
>
          3         3                      4                       4
  - 1/3 b~  + 1/3 a~  + 1/3 b~ sqrt(1 + b~ ) - 1/3 a~ sqrt(1 + a~ )
```

a nice (but unfortunately wrong) result. Changing the assume-line to

```
>assume(a<1,b>1,a>-1);
```

seems to produce the correct result, as does also the explicit command

```
> int(sqrt(1+x^4)-x^2, x=0..100);
```

7.32 bug in algebraic symbolic integration in maple v.4 and maple v.5 (26.9.97)

7.32.1 luca ciotti

I found a very strange bug in the symbolic integration routine in MapleV4, working with a very simple algebraic function:

```
f:=  A/( (1+x^2)^(3/2) * (1 + sqrt(1 + x^2)))

 int(f, x=0..infinity) = 0 <===========
```

a result obviously wrong.

Three important facts must be noticed :

1)The previous result is obtained when the exponent 3/2 is replaced by 5/2, 7/2, 9/2 etc, but NOT for EVEN exponents

2)Most strange, CORRECT result is obtained when the symbolic constant A (or any other symbolic constant) is REPLACED in the definition of f with a NUMBER, i.e., this strange behavior is present ONLY when the numerator of f is SYMBOLIC.

3)I found the IDENTICAL behavior in MapleV.3

Both the versions of Maple I'm using are under Unix.

It is corrected with Maple 6. (U. Klein)

7.32.2 Douglas B. Meade (29.9.97)

I found your posting interesting. In this response I first present my findings concerning the computations you report, then present a different approach to the problem that appears to be more promising. You should be able to copy either Maple session directly into a Maple worksheet (R4).

I do not obtain the result that you report. I, too, am working with Release 4 on UNIX (Solaris). While my results are different, they are not what you might expect. Here's what I found:

```
> restart;
> interface( version );
```

```
Maple Worksheet Interface, Release 4, SUN SPARC SOLARIS, Apr 19, 1997

> interface( patchlevel );

                                  2

> f:=  A/( (1+x^2)^(3/2) * (1 + sqrt(1 + x^2)));

                                    A
                 f := ------------------------------
                                2 3/2            2 1/2
                      (1 + x^2)    (1 + (1 + x^2)   )

> int( f, x=0..infinity );

                   infinity - signum(A) infinity

> int( subs(A=1,f), x=0..infinity );

                             - 1/2 Pi + 2

> evalf( Int( subs(A=1,f), x=0..infinity ) );

                             .4292036732
```

For additional information on Maple's evaluation of this integral you might insert the command:

```
> infolevel[int]:=5;
```

before the int command. The extra output will provide some insight into Maple's processing. In this case it appears that the problem is divided into two elliptic integrals that converge only under certain conditions. While the conditions are displayed in the userinfo, they do not appear to be carried forward with the result.

The "even" exponents are not as much of a problem because of the elimination of one of the square roots from the denominator. These roots are likely to be the ultimate source of the difficulties for this problem.

In fact, the repeated occurrence of `sqrt(1+x^2)` suggests a substitution that might

yield a more tractable formulation of the problem. The **changevar** command from Maple's student package can be used (WITH CAUTION - it's not a problem here, but I seem to remember that changevar can become confused, e.g., limits of integration for non-monotone substitutions) to complete the change of variables.

```
> restart;
> with( student ):
> f := A/((1+x^2)^p*(1+(1+x^2)^(1/2)));

                                    A
                  f := --------------------------
                                2 p            2 1/2
                       (1 + x^2)   (1 + (1 + x^2)   )

> F := Int( f, x=0..infinity );

                   infinity
                  /
                 |                        A
          F := |          -------------------------- dx
                 |                2 p            2 1/2
                /           (1 + x^2)   (1 + (1 + x^2)   )
                  0

> G := changevar( (1+x^2)=u^2, F );

                   infinity
                  /
                 |                      A u
           G := |          ------------------------- du
                 |              2 p                2 1/2
                /           (u^2)   (u + 1) (-1 + u^2)
                  1

> value( subs(p=3/2,G) );

                          2 A - 1/2 Pi A

> value( subs(p=4/2,G) );
```

```
                          -2 A + 3/4 Pi A

> value( subs(p=5/2,G) );

                          8/3 A - 3/4 Pi A

> value( subs(p=6/2,G) );

                                    15
                          - 8/3 A + -- Pi A
                                    16

> value( subs(p=7/2,G) );

                                    15
                          16/5 A - -- Pi A
                                    16
```

I hope you have found this useful and informative.

7.32.3 Ferdinand Gleisberg (30.9.97)

I add to Luca's observations:

With

```
> assume(A>0);
```

the correct result is obtained !

Ferdinand Gleisberg.

7.32.4 Willard, Daniel, Dr., DUSA-OR (30.9.97)

Would it work better if you defined `f` as `f:=x->` etc?

7.32.5 luca ciotti (3.10.97)

Thank you very much for all your interesting comments!

I would like to add the observation that assuming `A<0` the previous integral is returned infinity!

Obviously, my problem is not the specific evaluation of the previous integral (it can be solved very easily by hand), but a more general problem. In fact, suppose you have a VERY LONGE symbolic problem that you cannot control step by step, a problem like the previous can be quite dangerous.

Moreover, I think that a "normal" user cannot know ALL what Maple is doing solving a problem (suppose maple use the previous integral in doing a much more complex integration as an intermediate step...) Certainly, many evident errors (like the previous one) can certainly be handled with some "interactive" work, but unfortunately not always such work is doable.....

7.33 Bug in allvalues (26.4.96)

7.33.1 ANDREAS JUNG

There is a bug in allvalues in Release 3, patchlevel 3. Everything works well on polynomials that have at least two different roots:

```
allvalues(RootOf(x^5*(x-2)))          ==> 0,0,0,0,0,2
allvalues(RootOf((x-7)^4*(x-3)))      ==> 3,7,7,7,7
allvalues(RootOf((x-1)*(x-2)*(x-3))) ==> 1,2,3
```

But on polynomials $(x-k)^n$ which are not expanded and on x^n, Maple fails and only returns the root one time instead of n times:

```
allvalues(RootOf(x^2))              ==> 0
allvalues(RootOf(x^3))              ==> 0
allvalues(RootOf(x^4))              ==> 0
allvalues(RootOf((x-7)^4))          ==> 7
allvalues(RootOf(expand((x-7)^4))) ==> 7,7,7,7
```

So you can work around the bug by simply expanding the polynomial. But there still remains the faulty case x^n.

Is this bug corrected in Release 4 or Release 5?

The same behaviour up to Maple 8 (U. Klein)

7.34 bug in CMaple generating JPEGs, Maple 7 and 8 (31.3.02)

7.34.1 Moore, Chuck

I've been using CMaple to crunch data and create text output for some time now. It does a really good job at that. However, I've run into a problem in trying to have CMaple generate a plot as a "JPEG".

I'm sending a series of commands to CMaple. CMaple seems to process the commands OK (i.e. no error messages arise), however no "JPEG" is created. Yet, I can run (i.e. cut/paste) the same commands in interactive Maple and get a "JPEG".

7.34.2 Robert Israel (1.4.02)

Maybe the plot file is being put in a directory where you didn't think of looking for it. It seems there's a bug in command-line Maple 7 (at least under Unix): the "currentdir" command does not influence the placement of jpeg files. For example:

```
> currentdir();
                              "/home/israel"

> currentdir(tmp);
> plotsetup(jpeg,plotoutput="foo.jpg");
> plot(x^2,x=0..1);
```

... and the plot file is found in `/home/israel`, not in `/home/israel/tmp`. This bug doesn't seem to affect interactive Maple, or Maple 6.

7.34.3 Moore, Chuck (1.4.02)

Thanks for the advice.

The currentdir() command does not seem to work under (Windows 2000 - Maple7) CMaple. Further, the plotsetup and plot commands, which you gave as samples (below), work fine on my interactive Maple system. However, the exact same commands DO NOT work under CMaple.

I would ask of anyone who is currently running Maple7 to try and run (just) the plotsetup and plot commands of your sample and let me know if they work (i.e. produce a plot).

7.34.4 Robert Israel (1.4.02)

The behaviour in command-line Maple 7 under Windows 95 is similar to under Linux. The plot is produced, in my case in the directory `"F:\\Program Files\\Maple 7"`, which is the current one (as shown by currentdir()) when Maple is started. It appears to be identical to the plot produced by the same commands in worksheet Maple.

7.34.5 Moore, Chuck (1.4.02)

Hmmmmm.

I did the following on command-line Maple7 (Windows 2000) :

```
> with(plots);
> plotsetup(jpeg,plotoutput="test.jpeg");
> plot(x^2,x=0..1);
> done;
```

... and got nothing.

On top of that, the currentdir command seems to be non-functional:

```
> currentdir();
                currentdir()

> currentdir("C:/Workarea");
                currentdir("C:/Workarea");

> currentdir();
                currentdir()
```

Again, the interactive Maple works fine.

7.34.6 Moore, Chuck (12.4.02)

Typing the command `"libname;" returned "c:/lib"` .

On my workstation, this directory doesn't exist. Maple 7 is installed in `C:/Program Files/Maple 7`.

I dug up the information on the 'libname' function through GUI Maple : my MAPLE environment variable was not set. The `"c:/program files/maple 7/bin.wnt"` directory was identified in the PATH variable though, which was why CMAPLE would start but the libname function returned only `"c:/lib"`.

I defined (i.e. permanently set) the MAPLE environment variable to `"c:/program files/maple 7/"`. Then, I verified that this fixed the problem by creating a plot using the test commands (below).

7.35 Bug in combine in Maple V.4 (27.11.96)

7.35.1 Preben Alsholm

I ran into this bug today (Maple V, Release 4.00b):

```
combine( 2^(1/3)+I*2^(1/3) );

                    2*2^(1/3)
```

The same bug appears with combine(..., radical). It also appears for other expressions of the same form:

```
n^(p/q)+I*n^(p/q)
```

with n, p, and q being actual integers.

Addendum to my first post.

```
combine( a^(1/3)+b*I*a^(1/3) );

                  a^(1/3)+b*a^(1/3)
```

The imaginary unit I is simply gone!

The bug is removed in Maple V Release 5. (U. Klein)

7.35.2 Uwe Klemt

Using the option 'conjugate' in combine avoids at least the disappearing of the imaginary unit, but does no combining at all. I guess this is not what you really want.

7.35.3 Sjoerd W. Rienstra

Indeed a mystery:

in anything like:

```
   combine(a^(2/7)+I*b^(5/13));
```

with powers, other than `(1/2)`, the "I" dispappears.

However, if we add the option "power"

```
combine(a^(2/7)+I*b^(5/13),power);
```

the answer is again allright.

7.36 bug in condition numbers in Maple V.3 to Maple V.5 (31.1.98)

7.36.1 Dale Alspach

Here is a MAPLE BUG for the benefit of the MUG. It's still there in MAPLE V Release 4, although I ran this under UNIX maple. I read the output into the program (or the other way around).

Notice the MAPLE 2-norm is actually the _SMALLEST_ singular value, not the largest. The 2-condition is the reciprocal of what it should be, which is the ratio of the largest over the smallest singular value.

```
#  This MAPLE program illustrates a bug in MAPLE that occurs
##               when computing 2-norms of matrices.
 restart;
 with(linalg):
Warning: new definition for   norm
Warning: new definition for   trace
 A:=randmatrix(5,5);
                              [ -85  -55  -37  -35   97 ]
                              [                        ]
                              [  50   79   56   49   63 ]
                              [                        ]
                         A := [  57  -59   45   -8  -93 ]
                              [                        ]
                              [  92   43  -62   77   66 ]
                              [                        ]
                              [  54   -5   99  -61  -50 ]

 sgm:=evalf(Svd(A));
   sgm := [ 215.3159815, 192.8527895, 110.7852504, 66.73636029, 33.05625793 ]

 norm2A_1:=sgm[1];
                             norm2A_1 := 215.3159815

 norm2A_2:=evalf(norm(A,2));
                             norm2A_2 := 33.05625794
```

```
cond2_1:=sgm[1]/sgm[5];
                              cond2_1 := 6.513622381

cond2_2:=evalf(cond(A,2));
                              cond2_2 := .1535244049
```

It is corrected with Maple 6. (U. Klein)

7.36.2 Prof. Dr. Peter Thieler

One of my students drew my attention to your remark on Maple's behaviour with respect to the 2-norm of matrices.

Let me refer to the nice book GOLUB/ORTEGA, Scientific Computing - An Introduction with Parallel Computing, Academic Press, 1993. There I find a short and very well usable chapter on basic facts that are of interest in the field of numerical linear algebra. The subchapter 2.3 is dealing with norms.

The authors define the EUCLIDean matrix norm - similar to the way Maple does. See then subchapter 6.2 for norm based condition numbers. Again, GOLUB and ORTEGA give the definition Maple is also using.

Trying `?linalg,norm` , you will get the information:

```
"The '2'-norm of a matrix is the square root of the maximum eigenvalue of the matrix
```

You may use your example to reproduce Maple's definition:

```
> restart:with(linalg):
Warning, new definition for norm
Warning, new definition for trace
> A:=randmatrix(5,5);

                    [-85    -55    -37    -35     97]
                    [                               ]
                    [ 50     79     56     49     63]
                    [                               ]
               A := [ 57    -59     45     -8    -93]
                    [                               ]
                    [ 92     43    -62     77     66]
                    [                               ]
                    [ 54     -5     99    -61    -50]
```

```
> Automatic:=norm(A,2);

  Automatic := sqrt(max(| RootOf(-102991333311217647241

                                                      2
        + 130759387784582485 _Z - 36880268261270 _Z

                      3            4     5
        + 3286110146 _Z  - 101373 _Z  + _Z ) |))

> ByHand:=sqrt(max(eigenvals(evalm(A &* transpose(A)))));

  ByHand := sqrt(max(RootOf(-102991333311217647241

                                                      2
        + 130759387784582485 _Z - 36880268261270 _Z

                      3            4     5
        + 3286110146 _Z  - 101373 _Z  + _Z )))

> evalf(Automatic)=evalf(ByHand);

                    33.05625794 = 33.05625794
```

Trying `?linalg,condition` , you get the text:

`"cond(A, normname) computes norm(A, normname) * norm(inverse(A), normname)."`

Again, you may use your example to test Maple:

```
> evalf(cond(A,2))=evalf(norm(A,2)*norm(A^(-1),2));

                    .1535244049 = .1535244049
```

I use the same book, subchapter 2.2, to recall that `B:=A*transpose(A) and C:=transpose(A)*A` both are positive definite (which means that they cannot have negative eigenvalues). Moreover, both of them are symmetric, which means that `det(B-lambda*I)=det(C-lambda*I)` holds and, thus, B and C do have the same eigenvalues (not: eigenvectors!). Hence, the GOLUB/ORTEGA definition (using the spectral radius of C to determine

the EUCLID norm of A) and Maple's definition (using the spectral radius of B to do the same) coincide and reflect what can be found elsewhere as well.

```
> ByHandTranspose:=sqrt(max(eigenvals(evalm(transpose(A)&*A))));

  ByHandTranspose := sqrt(max(RootOf(-102991333311217647241

                                                    2
        + 130759387784582485 _Z - 36880268261270 _Z

                        3             4     5
        + 3286110146 _Z  - 101373 _Z  + _Z )))

> evalf(Automatic),evalf(ByHand),evalf(ByHandTranspose);

               33.05625794, 33.05625794, 33.05625794
```

Taking all this into account, I am sure that you will agree, that Maple is quite good in calculating the EUCLIDean norm or the EUCLIDean condition number of a matrix.

I think there is no bug.

Nevertheless, it is also very common to use condition numbers different to that seen above. The definition you mention reminds me of the TODD number (R. TODD, The Condition of Certain Matrices, Quart. J. Mech. and Appl. Math. 2 (1949) 469-472; Math. Rev. 11, 619) which is given by the quotient of the largest of the absolute vaues of the eigenvalues of A over the smallest of the absolute vaues of the eigenvalues of A. This is, in other words, the spectral radius of A, multiplied by the spectral radius of the inverse of A.

In this case, your example yields:

```
> num:=max(op(map(abs,[allvalues(eigenvalues(A))])));

                          num := 125.5772260

> den:=min(op(map(abs,[allvalues(eigenvalues(A))])));

                          den := 78.82169898

> T:=num/den;
```

```
                              T := 1.593180909

> fac:=max(op(map(abs,[allvalues(eigenvalues(A^(-1)))])));

                             fac := .01268686178

> T=num*fac;

                          1.593180909 = 1.593180909
```

7.37 bug in contourplot in Maple 6 (15.4.00)

7.37.1 Marko Horbatsch

It seems that maple6 (trial version) cannot calculate contours correctly.

Try the phase space plot for the simple pendulum (m=l=1, g=10):

```
Tkin:=omega^2/2;
Upot:=10*(1-cos(phi));
plots[contourplot](Tkin+Upot,phi=-3*Pi/2..3*Pi/2,omega=-10..10,
                   contours=[1,5,9,13,17,21,25,29]);
```

In version 5.01 we observe correctly that beyond the 5th contour we have rotations (rather than oscillations) as solutions. In Maple6 all 8 contours are (wrongly) shown as closed.

It is corrected with Maple 6.01. (U. Klein)

7.37.2 Robert Israel (25.4.00)

For a more dramatic example of this bug, try

```
plots[contourplot](y/x,x=0.1 .. 1, y = 0.1 .. 1);
```

This bug affects the contourplot of any expression f where denom(f) is not 1. What is returned is a contourplot of numer(f), not f.

The culprit is in line 133 of ‘plot/iplot2d‘:

```
    f := `plot3d/makefunc`(numer(F),[x, y])
```

It can be fixed by replacing numer(F) by F in that line.

7.38 Bug in contourplot in Maple6 (17.10.00)

7.38.1 Carl Eberhart

Avanish Sathaye, who is using Maple in his calc 3 class, has observed that when

```
plots[contourplot]((x+y^2)/(x^2+y),x=-3..3,y=-3..3);
```

is issued to maple6, a contourplot of the numerator is returned. when the rational expression is made into a function

```
f := (x,y)->(x+y^2)/(x^2+y);
```

and

```
plots[contourplot](f, -3..3, -3..3);
```

is issued a different, even more bizarre, behavior occurs.

7.38.2 Denis Sevee (19.10.00)

Even simpler expressions give similar results. For example,

```
contourplot(x/y,x=-3..3,y=-3..3);
```

For more odd variations, try

```
contourplot( [x,y,x/y],x=-3..3,y=-3..3);
```

It seems that contourplot has problems with fractions(?).

You can always resort to 'plot3d' with the plotting style set to contour. If you rotate the resulting plot so that you're looking down from above you get a pretty good result.

7.38.3 Robert Israel (19.10.00)

This is a known bug in Maple 6, which has been corrected in Maple 6.01. A patch is also available in my Maple Advisor Database,

`http://www.math.ubc.ca/~israel/advisor`.

Since f is unbounded on this interval, it's not surprising that contourplot has problems here (it tries to choose contour values that are equally spaced over the range of observed values of f, but those values will be rather large and will only occur for a few sampled points).

More surprising is the behaviour when you specify the contour values. For example:

```
> plots[contourplot]((x,y) -> (x+y^2)/(x^2+y),-3..3,-3..3,
                     contours=[0]);
```

A work-around for this is to modify the function so that the denominator won't be exactly 0 at any of the grid points. For example:

```
> plots[contourplot]((x,y) -> (x+y^2)/(x^2+y+0.00000001),-3..3,-3..3,
                     contours=[-3,-2,-1,0,1,2,3]);
```

It still looks a bit strange because linear interpolation of the function values doesn't work well across the curve where the denominator is 0. That problem will be alleviated somewhat by increasing the "grid" values: try it with e.g. `grid=[100,100]`.

7.39 bug in contourplot options, Maple 7 and Maple 8 (27.5.02)

7.39.1 Dale Alspach

It appears that there is some bug in contourplot (Maple 7 linux). The combination of variable range for y, filled=true and explicit contours produces a large red triangle `(side 10^14) for (x^2+y^2)^(1/4)`. Changing the exponent from `1/4 to 1/2 or 1/3` yields the expected plot. `1/5 and 1/6` also produce triangles.

This plots nonsense:

```
contourplot((x^2+y^2)^(1/4),x=-2..2,y=-(4-x^2)^(1/2)..(4-x^2)^(1/2),
            filled=true,contours=[0,.2,.4,.6,.8,1,1.1,1.2,1.3]);
```

These plot something reasonable:

```
contourplot((x^2+y^2)^(1/4),x=-2..2,y=-2..2,filled=true,
            contours=[0,.2,.4,.6,.8,1,1.1,1.2,1.3]);

contourplot((x^2+y^2)^(1/2),x=-2..2,y=-(4-x^2)^(1/2)..(4-x^2)^(1/2),
            filled=true,contours=[0,.2,.4,.6,.8,1,1.1,1.2,1.3]);

contourplot((x^2+y^2)^(1/4),x=-2..2,y=-(4-x^2)^(1/2)..(4-x^2)^(1/2),
            filled=true,contours=10);
```

7.39.2 Robert Israel (4.6.02)

The bug seems to be fixed in Maple 8.

I get the same bug in Maple 8 (U. Klein)

7.40 bug in convert/expsincos in Maple 6 (8.2.01)

7.40.1 Pavel Honzatko

Does anybody know, why the following occurs in Maple 6.01 and how to fix this.

```
>convert(I*cos(x),expsincos);
Error, (in convert/expsincos) too many levels of recursion
```

This problem in slightly modified form appears in integration of

```
>int(exp(-I*w*t)/cosh(t),t);
Error, (in convert/expsincos) too many levels of recursion
```

It is corrected with Maple 7. (U. Klein)

7.40.2 Preben Alsholm (12.2.01)

This is no explanation, but just an addition. I tried the following

```
trace(`convert/expsincos`);
convert(I*cos(x),expsincos);
```

and I got quite a number of identical statements of the form

enter `convert/expsincos, args = I`

before my Windows 95 decided to shut down the program.

If we look at the procedure `‘convert/expsincos‘` with

```
> showstat(`convert/expsincos`);
```

then we see in line 24:

```
map(`convert/expsincos`,f)
```

When `‘convert/expsincos‘` is applied to `I*cos(x)` the first that happens is that `‘convert/expsincos‘` is mapped onto `’I’ and cos(x)`. It is `’I’` that is the problematic input to `‘convert/expsincos‘` since it is not handled by any other line in the

procedure. Thus 'I' will by line 24 be sent to `convert/expsincos`(I) which by its line 24 will send 'I' to etc.....

The 'I' should have been handled by the first line but

```
type(I,ratpoly(numeric));
```

returns false in Release 6, but not in Release 5.1.

I suspect that the bug is caused by the change in the handling of complex numbers from Release 5.1 to 6.

7.40.3 Robert Israel (12.2.01)

This is a bad bug, which makes convert(..., expsincos) practically useless on anything containing I. The problem shows up in an even simpler form:

```
> convert(I, expsincos);

Error, (in convert/expsincos) too many levels of recursion
```

The problem, I think, is that `convert/expsincos` starts with

```
      if nargs = 1 and type(f,ratpoly(numeric)) then
        ...
      elif nargs = 2 and not has(f,x) then
        ...
      elif type(f,function) and nops(f) = 1 then
        ...
      elif type(f,name) then
        ...
      else
        map(`convert/expsincos`,f)
      end if
```

Since none of the other conditions is true in convert(I,expsincos), the last "else" clause will run, and `convert/expsincos` will be called recursively. It used to be that I was an alias for (-1)^(1/2), so there would be two recursive calls of `convert/expsincos` with -1 and with 1/2. However, as of Maple 6, I is of type "complex", and is considered to be atomic. Thus now `convert/expsincos` is called recursively with the same argument

I, and we get an infinite recursion.

At first I thought that a fix was to use the remember table of ‘convert/expsincos‘, with

```
> `convert/expsincos`(I):= I;
```

However, there would still be trouble with every other object of type "complex". Instead, I'll write a "wrapper" for ‘convert/expsincos‘:

```
> oldconv:= eval(`convert/expsincos`);
  `convert/expsincos`:= proc(f)
      if type(f,And(complex,atomic)) then f
      else oldconv(args)
      fi
   end;
```

This ought to work.

7.40.4 Werner Knoben (16.2.01)

The reason for the mistake of the function convert/expsincos is that the complex unit I is not any more of type numeric. In the fourth line of the source code of the command ‘convert/expsincos‘ you find

```
if nargs = 1 and type(f,ratpoly(numeric)) then f
```

Change this to

```
if nargs = 1 and type(f,ratpoly({numeric,complex})) then f
```

and the function works correct.

7.41 bug in convert[ratpoly], Maple 6 and Maple 7 (26.3.01)

7.41.1 Florian Dufey

I wanted to calculate the pade approximant to a Taylor series for given value of x.

I specified `Digits:=31`.

I used for this `convert(series, ratpoly,i,i)` (or `convert(series, ratpoly,i-1,i)`) , `pade(series,x,[i,i])` and `convert(series,confrac,‘subdiagonal‘)` where series is a series of order `2i+1 (2i)`. Finally, I substituted $x = 1$.

Independently, I calculated the corresponding approximants using Wynn's epsilon algorithm.

For $i < 10$ all three methods lead to nearly the same results (up to some 25 digits).

For $i > 10$, `convert(ratpoly)` and pade suddenly returned results which where only correct to 4 digits.

I did these calculations under MapleV rel. 2 and under Maple6.

Under Maple 6 however, the `convert(confrac)` gave an error message `"division by zero series"` in the case that the series were of order `2*i`.

The help page doesn't mention the algorithm used in the `convert(confrac)` routine.

7.41.2 Helmut Kahovec (28.3.01)

After I had looked at the source code of ‘`convert/confrac`‘ I tried the undocumented third parameter ‘`superdiagonal`‘ -- and it worked. Here is what I have got in Maple6:

```
> restart;
> with(numapprox):

> f:=proc(expr,N,x0,d)
> local x,a1,a2,a3,a4;
>   x:=op(indets(expr,name));
>   a1:=normal(
>     pade(
>       convert(series(expr,x=0,2*(N+1)),polynom),
>       x,
```

```
>       [N,N]
>     )
>   );
>   a2:=normal(
>     convert(series(expr,x=0,2*(N+1)),ratpoly,N,N));
>   a3:=normal(
>     convert(series(expr,x=0,2*N),confrac));
>   a4:=normal(
>     convert(
>       series(expr,x=0,2*N),
>       confrac,
>       `superdiagonal`      # <===!!!===
>     )
>   );
>   if nops({a1,a2,a3,a4})=1 then
>     map(
>       print@(u->evalf(eval(u,x=x0),d)),
>       [a1,expr]
>     )
>   end if;
>   NULL
> end proc:

> f(sin(x),10,1,20);

                        .84147098480789650667

> f(sin(x),20,2,40);

              .90929742682568169539601986591174484268 75

> f(sin(x),30,3,60);

    .14112000805986722210074480280811027984693326425226558416 1912
```

All four approximations `a1,a2,a3,a4` were the same if the order parameters were equal to those used in f().

7.41.3 Florian Dufey (28.3.01)

As the error with the `convert[ratpoly]` routine doesn't seem to appear for every input, I will give my input. The results of the `convert[confrac]` routine seem to converge to the correct result `(4.931704...)` which I also know from calculations which don't use a series expansion.

The two methods differ first for the `[10,10]` approximant: `confrac: 4.9330598771621... ratpoly: 4` the `[9,9]` approximants coincide to 26 digits.

file perout:

```
  4.90661431619462896731500323866 3
 0.00000000000000000000000000000000E+00
 0.10027506324793592648746731895 23
-4.28251291615037778797765619180 45E-04
-0.95522391792325032959339887788 126
 8.57616229714549927209301117386 72E-03
  17.45154716638185567939630131163
-0.23506197408776038896497448542 91
 -398.45479423658136259855220112 13
  7.15624213535457379638521503501 5
  10188.89027936986829144576634036
 -228.74491003490420819574996294 55
 -279146.80381689989449934270641 61
  7520.44786443533622187880655473 2
  8011969.52134534049912162617838 7
 -251826.21746045488400964646674 42
 -237795243.605846253700868145765 4
  8542006.31266927634108607184793 6
  7238714408.232864480377075988925
 -292532053.558144500701994481458 1
 -224760078091.07118862486160926 89
  10092326226.252192804862204295 84
  7090687290387.637279509929437663
 -350231863205.15728507770627445 21
 -226638741608644.0867600979233863
  12212162115859.32689138894896721
  7323623163746757.774836679568589
 -427512752999902.9314599007038804
```

```
 -238860835495485678.6412699643812
  15016037765877188.75356709241297
  7852792004077195796.414068303233
 -528932737787606479.5042379263840
 -259961139015313134301.4401844604
  18677396449998980331.82972087396
  8658192149601240597963.646201997
 -660947566734026317829.4553646339
 -289917464226794648572825.0770237
  23433697291895503144416.31715501
  97542584733198754010031O9.610984
 -832233107565159476893114.9681807
 -32958781687553228680998550 9.9060
  29600674200295888158483002.85808
  11179473545033574608724817973.04
 -10542523362664374298911119 19.944
 -380528467764493572539940998621.0
  37593898674137432962073722880.83
 1.29936789804732973390058922944 66E+31
 -13420579056816196531738957004 94.
-4.44976941087612001958969643124 33E+32
 4.79583167834379609333909894090 83E+31
 1.52791501724739721135422397302 10E+34
-1.71537248003551886792391001624 62E+33
-5.25927064321226044396887975002 60E+35
 6.14076240049155827201959486922 40E+34
 1.81440931850270827587995895021 20E+37
-2.20002088382990845621913597608 71E+36
-6.27272415319796080155893969154 61E+38
 7.88762927094921618873863910253 67E+37
 2.17281982090234608656241450014 18E+40
-2.82981882455493032872187521671 17E+39
-7.54014849744817267765596226609 49E+41
 1.01588108952949099211675170661 16E+41
 2.62103338168990456210206377305 78E+43
-3.64905933795664476347190178356 07E+42
-9.12545434452231084372052714721 62E+44
 1.31146367585187572653592392807 92E+44
 3.18187377429676052711789616126 33E+46
```

```
  -4.71578443999692005886594210591 72E+45
  -1.11101259243817265886370146644 41E+48
   1.69652544988164665253867482716 77E+47
   3.88442811386228317029160667293 95E+49
  -6.10608986934590814794633694913 58E+48
  -1.35979920368790973691676958530 04E+51
   2.19862456910232274003782810996 25E+50
   4.76576108031875852363580929206 46E+52
  -7.91979016487667117815238918557 11E+51
  -1.67213810797562621876122896227 93E+54
   2.85391789110234456115779522351 47E+53
   5.87312401941051156278603494792 70E+55
  -1.02878759013820045770510721303 92E+55
  -2.06490479431595361096479935579 15E+57
```

maple program:

```
> a:=[];
> Digits:=31;
> printlevel:=0;
> for i from 1 to 82 do
> S:=readline(`perout`):
> a:=[op(a),op(1,sscanf(S,`%a`))];
> od;
> #don't care about the error message from sscanf
> print(a);
> printlevel:=1;
> l:='l';
> f:='f';
> f:=x->sum(op(l+2,a)*x^(l-1),l=1..79);
> for i from 1 to 38 do
> print(i-1,i,subs(x=1,(convert(series(f(x),x,2*i),confrac,'subdiagonal')))
              +op(1,a));
> print(i,i,subs(x=1,(convert(series(f(x),x,2*i+1),confrac,'subdiagonal')))
            +op(1,a));
> print(i-1,i,subs(x=1,(convert(series(f(x),x,2*i),ratpoly,i-1,i)))+op(1,a));
> print(i,i,subs(x=1,(convert(series(f(x),x,2*i+1),ratpoly,i,i)))+op(1,a));
> od;
```

7.41.4 Helmut Kahovec (2.4.01)

You certainly mean

ratpoly: 4.9330730107748...

don't you? Reading in your file perout. I can exactly reproduce your results as follows (note that I am using ‘`superdiagonal`‘ again):

```
> restart;
> with(numapprox):

> readText:=proc(fname)
> local fd,S,line;
>   fd:=open(fname,READ);
>   S:=NULL;
>   line:=readline(fname);
>   while line<>0 do
>     S:=S,line;
>     line:=readline(fname)
>   end do;
>   close(fd);
>   S
> end proc:

> Digits:=31:
> L:=map(op@sscanf,[readText("perout.")],"%a"):
> P:=sort(add(L[i+2]*x^(i-1),i=1..nops(L)-2)):

> f:=proc(expr,N,x0)
> local x,a1,a2;
> global L;
>   x:=op(indets(expr,name));
>   a1:=normal(
      convert(series(expr,x=0,2*N+1),confrac,`superdiagonal`)
    );
>   a2:=normal(
      convert(series(expr,x=0,2*N+1),ratpoly,N,N)
    );
>   evalf(eval([a1,a2],x=x0))
```

```
> end proc:

> map(print@(u->L[1]+u),f(P,9,1)):

                  4.934381129522362692062112024892

> map(print@(u->L[1]+u),f(P,10,1)):

                  4.933059877162190764768477498212
                  4.933073010774838552651467870788

> map(print@(u->L[1]+u),f(P,11,1)):

                  4.933059894927514843532036653480
                  4.933069963374504461817256514371

> map(print@(u->L[1]+u),f(P,38,1)):

                  4.931704232773919949343610941915
                  4.933073010774838552651467870788
```

In order to get a comparable result with ‘`convert/ratpoly`‘ you have to use many more digits than 31. If you use 1000 digits then you get, for example:

```
> Digits:=1000:
> map(print@(u->L[1]+u),f(P,38,1)):

  4.931704232882756563816776321859584868340431246936300189687750\
        46\
        123452996337354096358461984608306640112305368008509960794\
        2\
        032871596129101065947175668179988449542207147314625107058\
        2\
        799042866206681067175238331017321040355778499430711835076\
        7\
        813983382770438159945836599717325970806932250787199767056\
        9\
        314907186272117470180677100924892302631749578534188062618\
        8\
        852242078335299507780568217019028845447976493476900323872\
        3\
        226093175537385106640756388750530545743008800226962819388\
        3\
        284608131969880211193581920002707297777588760302876731473\
        4\
        795838711029872071693953380860915326145753073749834209406\
        1\
        685567052232437944444992763387844377641564061555720586096\
        1\
        008448966151947944052637280604262079882566582713599070222\
        0\
```

```
      22454959248980861098897107497281549639972481822522208352 64\
      81255771020026205061284178198893743250416112820032491391 08\
      53902445090382178631681341745536744836862890681202397904 01\
      71326761931168062504026779887006319260765314699127530546 21\
      47295980642634263280503971540672633906490167708816174469 65\
      770737846

4.93170423288275656381677632185958486834043124693630018968775046\
      12345299633735409635846198460830664011230536800850996079 42\
      03287159612910106594717566817998844954220714731462510705 82\
      79904286620668106717523833101732104035577849943071183507 67\
      81398338277043815994583659971732597080693225078719976705 69\
      31490718627211747018067710092489230263174957853418806261 88\
      85224207833529950778056821701902884544797649347690032387 23\
      22609317553738510664075638875053054574300880022696281938 83\
      28460813196988021119358192000270729777758876030287673147 34\
      79583871102987207169395338086091532614575307374983420940 61\
      68556705223243794444499276338784437764156406155572058609 61\
      00844896615194794405263728060426207988256658271359907022 20\
      22454959248980861098897107497281549639972481822522208352 64\
      81255771020026205061284178198893743250416112820032491391 08\
      53902445090382178631681341745536744836862890681202397904 01\
      71326761931168062504026779887006319260765314699127530546 21\
      47295980642634263280503971540672633906490167705068965267 38\
      283094823                                                  ^^^^^^^^^^^^
      ^^^^^^^^^
```

Thus, keeping the number of digits fixed, ‘`convert/confrac`‘ gives much more precise results than ‘`convert/ratpoly`‘ if the coefficients in the polynomial to be approximated vary as much as in your case.

7.42 bug in D, Maple V to Maple 8 (6.8.02)

7.42.1 Paul E.S. Wormer

Here follows a transcript of a Maple 6 session on a PC:

```
> D[1](f); D[2](f);

                                D[1](f)

                                D[2](f)

> D[1](exp@f);

                            (exp@f) D[1](f)

> D[2](exp@f);
Error, (in D/exp) invalid arguments
```

My question is: what is wrong with the last statement?

7.42.2 Robert Israel(7.8.02)

It's a bug, and still present in Maple 8. A work-around is

```
> g:= (s,t) -> exp(f(s,t));
  D[2](g);
```

7.42.3 Bill Whiten (8.8.02)

`D[n]` refers to nth argument, not nth derivative e.g.:

```
> D[1](exp);

                                  exp

> D[1](x->x^2);

                                x -> 2 x
```

```
> D[2]](exp);
Error, (in D/exp) invalid arguments
> D[2](x->x^2);
Error, (in D/procedure) index out of range: function takes only 1 arguments
> D[1]((x,y)->x^2+y^2);

                              (x, y) -> 2 x

> D[2]((x,y)->x^2+y^2);

                              (x, y) -> 2 y

>
> D[1](D[1](exp));

                                   exp

> D[1](D[2]((x,y)->x^2*y^2));

                             (x, y) -> 4 x y
```

7.42.4 Stanley J Houghton (9.8.02)

It looks to me like a bug arising from

```
> D[2](f@g);
      (D[2](f)@g)*D[2](g)
```

Surely, by definition, it should deliver the result `(D[1](f)@g)*D[2](g)` when f is a function of one variable (as with exp in your case).

In addition, I see that the same error arises when the variables are explicitly shown in

```
> D[2]((x,y)->(exp@f)(x,y));
```

7.43 BUG in densityplot in Maple 6.01 (2.11.00)

7.43.1 Dr. TANAKA, Kazuo

In densityplot, we can specify `grid[m,n]`. Even though m is not requied to be equal to n, we have birrare results when m does not equal n.

For example,

we have a correct plotting by evaluating the followings;

```
>f:=(x,y)->1+cos(x);
>densityplot(f(x,y),x=-6..3,y=-4..2,axes=none,grid=[100,100],
                                                  style=patchnogrid);
```

The result is somewhat strange when we do the following:

```
>densityplot(f(x,y),x=-6..3,y=-4..2,axes=none,grid=[100,50],
                                                  style=patchnogrid);
```

7.43.2 Robert Israel (6.11.00)

Bugs in densityplot have been a problem since Release 5. Although the worst one, which caused x and y directions to be interchanged, was fixed in Maple 6, this one is still with us. A workaround is to prevent `evalhf` from being used, e.g. instead of `densityplot(f(x,y),...)` use `densityplot(Re(f(x,y)),...)`. Of course this slows things down quite a bit.

It is corrected with Maple 7. (U. Klein)

7.44 Bug in densityplot in Maple V.5 (13.8.99)

7.44.1 Robert Michael Sinclair

Densityplot (in the plots package) fails to produce correctly scaled RGB values when given a procedure in which complex numbers appear - even if they have no influence on the return value. In the following, I define two procedures, f and g, which differ only in the value of an irrelevant constant. In f it is 2001, in g it is `I (i.e. sqrt(-1))`. Note the difference in the output of densityplot.

This bug is annoying because it prevents one from plotting the Mandelbrot set in an obvious manner (using a procedure which iterates a complex map but returns the integer number of iterations).

```
    |\^/|     Maple V Release 5.1 (WMI Campus Wide License)
._|\|   |/|_. Copyright (c) 1981-1998 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> with(plots):
> f := proc(X,Y)
>        local Z;
>        if X>0 then Z := 2001*Y
>               else Z := 0
>        fi;
>        X+100
>      end:
> g := proc(X,Y)
>        local Z;
>        if X>0 then Z := I*Y
>               else Z := 0
>        fi;
>        X+100
>      end:
> densityplot(f,-1.5..1.5,-1.5..1.5);

  GGGGGGGGGGGGGGGGG1.5*GGGGGGGGGGGGGGGGGGGGG
  GGGGGGGGGGGGGGGGGGGG*GGGGGGGGGGGGGGGGGGGGG
  GGGGGGGGGGGGGGGGGGGG*GGGGGGGGGGGGGGGGGGGGG
  GGGGGGGGGGGGGGGGGGGG*GGGGGGGGGGGGGGGGGGGGG
```

```
  GGGGGGGGGGGGGGGGGG1*GGGGGGGGGGGGGGGGGGGG
  GGGGGGGGGGGGGGGGGGG*GGGGGGGGGGGGGGGGGGGG
  GGGGGGGGGGGGGGGGGGG*GGGGGGGGGGGGGGGGGGGG
  GGGGGGGGGGGGGGGGGG0.5*GGGGGGGGGGGGGGGGGGGG
  GGGGGGGGGGGGGGGGGGG*GGGGGGGGGGGGGGGGGGGG
  GGGGGGGGGGGGGGGGGGG*GGGGGGGGGGGGGGGGGGGG
 -****************************************-
 -1.5HHHH-1HHHH-0.5HHH0*HHHHH0.5HHHHH1HHHHH1.5
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHH-0.5*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHH-1*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHH-1.5*HHHHHHHHHHHHHHHHHHHH

> densityplot(g,-1.5..1.5,-1.5..1.5);

  HHHHHHHHHHHHHHHHH1.5*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHH1*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHH0.5*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
 -****************************************-
 -1.5HHHH-1HHHH-0.5HHH0*HHHHH0.5HHHHH1HHHHH1.5
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHH-0.5*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHH-1*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHH
```

```
HHHHHHHHHHHHHHHHHHHHH*HHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHH-1.5*HHHHHHHHHHHHHHHHHHHHH
```

It is corrected with Maple 6. (U. Klein)

7.44.2 Robert Israel (20.8.99)

There are actually two bugs to note here. The first is, I think, the worst because it interchanges the x and y variables (note that the shading seems to depend on y rather than x). This occurs when Maple uses `evalhf` (i.e. hardware floating-point) to evaluate the function to be plotted. The occurrence of I prevents evalhf from working, so Maple uses ordinary evalf (software floating-point), and here the second bug occurs: the RGB values are not properly scaled. Of course you could scale the function to be plotted yourself. But here's a fix:

```
> fixgray:= proc(p)
   local newp, cspecs, csp, cseq, cmax, cmin, i;
   cspecs:= indets(p,specfunc(anything,COLOR));
   newp:= p;
   for csp in cspecs do
     cseq:= op(2..-1,csp);
     cmax:= max(cseq);
     cmin:= min(cseq);
     cseq:= seq((cseq[i]-cmin)/(cmax-cmin),i=1..nops(csp)-1);
     newp:= subs(csp=COLOR(op(1,csp),cseq), newp);
     od;
   newp;
   end;

> fixgray(densityplot(g,-1.5..1.5,-1.5..1.5));
```

This should work OK. Note that this does not fix the interchange of x and y.

7.44.3 Helmut Kahovec (20.8.99)

In Release 4 both density plots are correct.

In Release 5 the density plot of f is also incorrect: there should be a horizontal density gradient instead of a vertical one. Since the body of f can be evaluated by evalhf() the procedure `plots[densityplot]()` calls `‘plot3d/density_hf‘()` plus `‘plot3d/densfunc_hf‘(), ‘plot3d/colorsur2‘()` plus `‘plot3d/cgridhf2‘()`, and `‘plots/densityplot/RGB_hf‘()`, respectively.

`‘plot3d/density_hf‘(), ‘plot3d/colorsur2‘(), and ‘plots/densityplot/RGB_hf‘()` define certain hfarrays A, B, and C. `‘plot3d/densfunc_hf‘(), ‘plot3d/cgridhf2‘()`, and `‘plots/densityplot/RGB_hf‘()` process A, B, and C. The processing is done by nested loops.

The inner index corresponds to x and the outer index to y. However, in the procedure `‘plot3d/cgridhf2‘()` the inner index corresponds to y and the the outer index to x.

This inconsistency exchanges the coordinate axes of all density plots.

Since the first line of `g (i.e., Z:=I*Y;)` cannot be evaluated using evalhf() the procedure `plots[densityplot]()` calls `‘plot3d/colorsur‘()` which in turn calls `‘plot3d/cgridf‘()`. The latter also has that inconsistency w.r.t. the loop indices. Additionally, it incorrectly computes the COLOR() part of the plot structure in this case and the density plot appears to be black.

`Z:=5 mod 2;` cannot be evaluated by evalhf() either. Hence the following procedure h gives an incorrect (black) density plot, too:

```
> restart;

> h:=proc(X,Y)
> local Z;
>   Z:=5 mod 2;
>   X+100
> end:

> densityplot(h,-1.5..1.5,-1.5..1.5,axes=FRAME);
```

7.45 bug in densityplot in MapleV.5 and patch (6.10.98)

7.45.1 Dave Damiano

I noticed in the Unix version R5 that density plot interchanges the roles of the independent variables. For example,

```
densityplot(x^2 -y,x=-3..3,y=-3..3);
```

yields a density plot of

y^2-x

and

```
densityplot(x^2-y,y=-3..3,x=-3..3);
```

yields the correct density plot but with the axes labeled incorrectly.

Is there a patch for this error?

It is corrected with Maple 6.01. (U. Klein)

7.45.2 Eric Johnstone (10.10.98)

Just got MAPLEVR5 (win95). I thought I would run an example from A. Heck's book (Introduction to MAPLE, Springer Verlag, 2ed page 428):

```
> restart;
> with(plots):
> phi:=ln(sqrt((x+ 1)^2 + y^2)) - ln(sqrt((x-1)^2 + y^2));
> densityplot(phi,x=-2..2,y=-1..1,grid=[80,80],axes=box,style=patchnogrid,
  colorstyle=HUE);
```

Release 4 does the plot correctly (like in Heck's book). Release 5 confuses the axes!!!!!!

This problem is not seen for contourplot or gradplot however. The problem appears to be endemic to densityplot

Bummer.

It is as if the arguments of the function are permuted or something like that within densityplot. Anybody else notice this problem? Is there a work-around?

The quality of the output from densityplot is better though, if one is prepared to ignore this minor detail.

Also, it appears that R5 renders the plot about 2-3 times faster in 1/2 memory than R4.

7.45.3 Robert Israel (24.10.98)

As several people have noticed, densityplot in Release 5 has a bug which causes x and y to be "interchanged". Actually the result of

```
densityplot(f(x,y), x = a .. b, y = c .. d)
```

is what should be obtained from

```
densityplot(f(a + (b-a)*(y-c)/(d-c), c + (d-c)*(x-a)/(b-a)), x = a .. b, y = c .. d
```

So a work-around is to use

```
> dplot:= proc(f, r1, r2)
   local x, y, a, b, c, d, g;
   if not typematch(r1, x::name = a::anything .. b::anything) then
      ERROR("incorrect second argument") fi;
   if not typematch(r2, y::name = c::anything .. d::anything) then
     ERROR("incorrect third argument") fi;
   g:= unapply(f, x, y);
   densityplot(g(a+(b-a)*(y-c)/(d-c), c + (d-c)*(x-a)/(b-a)), args[2..-1]);
   end;
```

The bug seems to be related to the use of hfarrays in densityplot.

7.46 Bug in derivative of JacobiDN in Maple 5 and Maple 6 (27.4.98)

7.46.1 Robert Michael Sinclair

Maple V R5's expression for the partial derivative of JacobiDN with respect to its second argument is incorrect (the sign of the first summand is wrong, see below). A workaround is to substitute all occurrences of `JacobiDN(u,k)` in an expression which is to be differentiated with `sqrt(1-k^2*JacobiSN(u,k)^2)`, and then differentiate.

In the following,

`dj(u,k)` returns Maple's incorrect expression for the derivative,

`cdj(u,k)` returns Maple's expression but with a minus sign before the first summand inserted by hand,

`ndj(u,k,h)` approximates the derivative by the standard difference quotient with step h, and

`sdj(u,k)` returns an expression computed using the workaround suggested above:

```
    |\^/|     Maple V Release 5 (WMI Campus Wide License)
._|\|   |/|_. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.

> dj:=unapply(diff(JacobiDN(u,k),k),u,k);

                                      2
                    k JacobiSN(u, k)  JacobiDN(u, k)
  dj := (u, k) -> --------------------------------
                                    2
                                1 - k

          - k JacobiCN(u, k) JacobiSN(u, k) u +

        k JacobiCN(u, k) JacobiSN(u, k) EllipticE(JacobiSN(u, k), k)/

              2
```

```
        (1 - k )

> lprint(dj(u,k));

   k*JacobiSN(u,k)^2*JacobiDN(u,k)/(1-k^2)-k*JacobiCN(u,k)*JacobiSN(u,k)*u
  +k*JacobiCN(u,k)*JacobiSN(u,k)/(1-k^2)*EllipticE(JacobiSN(u,k),k)

> cdj:=(u,k)->
> -k*JacobiSN(u,k)^2*JacobiDN(u,k)/(1-k^2)-k*JacobiCN(u,k)*JacobiSN(u,k)*u
> +k*JacobiCN(u,k)*JacobiSN(u,k)/(1-k^2)*EllipticE(JacobiSN(u,k),k);

                                      2
                    k JacobiSN(u, k)  JacobiDN(u, k)
  cdj := (u, k) -> - --------------------------------
                                       2
                                 1 - k

         - k JacobiCN(u, k) JacobiSN(u, k) u +

        k JacobiCN(u, k) JacobiSN(u, k) EllipticE(JacobiSN(u, k), k)/

              2
        (1 - k )

> ndj:=(u,k,h)->(JacobiDN(u,k+h)-JacobiDN(u,k))/h;

                           JacobiDN(u, k + h) - JacobiDN(u, k)
       ndj := (u, k, h) -> -----------------------------------
                                          h

> sdj:=unapply(diff(sqrt(1-k^2*JacobiSN(u,k)^2),k),u,k);

                       /                                          /
                       |                  2      2               |
  sdj := (u, k) -> 1/2 |-2 k JacobiSN(u, k)  - 2 k  JacobiSN(u, k) |
                       |                                          |
                       \                                          \

                                      3
```

```
        k JacobiSN(u, k)   k JacobiSN(u, k)
        ---------------- - -----------------
                  2                  2
              1 - k              1 - k

          u JacobiCN(u, k) JacobiDN(u, k)
         + -------------------------------
                         k

          JacobiCN(u, k) JacobiDN(u, k) EllipticE(JacobiSN(u, k), k)
         - ------------------------------------------------------------
                                          2
                                 k (1 - k )

        \\
        ||   /           2               2
        ||  /  sqrt(1 - k  JacobiSN(u, k) )
        || /
        //
```

```
> Digits:=40;
```

```
                          Digits := 40
```

```
> evalf(dj(0.5,0.99));
```

```
             18.697448894844288444423426841972475757 34
```

```
> evalf(cdj(0.5,0.99));
```

```
             -.220767304895483330592093370822369837823
```

```
> evalf(ndj(0.5,0.99,0.001)); evalf(ndj(0.5,0.99,0.0001));
```

```
             -.220871893342378065237365283705969733 8000
             -.220777764373708992586748405746685460000
```

```
> evalf(sdj(0.5,0.99));
```

```
-.220767304895483330592093370822369837832 6
```

See also: Bug in Jacobi elliptic function

It is corrected with Maple 7 (U. Klein)

7.47 bug in display in simplex package, Maple 4 and Maple 5 (3.5.96)

7.47.1 PierLuigi Zezza

The command "display" of the simplex package does not work properly as it is shown by the example in the help page

7.47.2 Jiri Hrebicek (10.5.96)

I have got troubles with using command `display(C)` in package simplex.

The description of command says that the set C of inequalities will be displayed in the matrix form Ax rel B but the matrix A is without the last column and B is the last column of A.

This is also in Examples of command display in help.

I use Maple V Release 4.

It is corrected with Maple 6. (U. Klein)

7.48 Bug in dsolve in Maple V.4 to Maple 6 (19.2.97)

7.48.1 W.M.Anderson

I've just discovered a bug that Release 4 seems to have introduced to dsolve. As an example to my students I offered the following homogeneous equation

```
> de:= diff(y(x),x)= y(x)/x + tan(y(x)/x);
> cond:=y(1)=Pi/3;
```

dsolve in Release V.3 gave the answer

```
> y(x) = x*arcsin(x*sqrt(3)/2)
```

but in V.4 two (!!) solutions are returned, one of which is the previous - in the less friendly form

```
    y(x) = x*arctan(x/sqrt(4/3 - x^2))
```

The other solution is the negative of this (and so is wrong as it does not satisfy the condition at x=1).

If one of the solutions is extracted for examination, it changes form yet again to

```
        x*arctan(x*sqrt(-9x^2 - 12)/(3*x^2 - 4))
```

[I know this bug all has to do with branches in the complex plane, but my students have no idea about that.]

It is corrected with Maple 7. (U. Klein)

7.49 BUG in dsolve in Maple V.4 (13.12.96)

7.49.1 Stanley J Houghton

This is a very much simplified version of a problem that indicates a bug in dsolve. I am running vn 4 on a Pentium PC under Windows 3.x

The simplified version makes it easier to understand.

```
> dsolve(diff(z(t),t) = int(y(u),u = 0 .. t), z(t));

   z(t) = 1/2*y(u)*t^2+_C1

> diff(%,t);

   diff(z(t),t) = y(u)*t
```

Certainly the integration variable 'u' should not appear anywhere in the result and, as shown above, if I differentiate the error is even more obvious.

The bug is removed with Maple V Release 5. (U. Klein)

7.49.2 Dan Dubois

Is any more information about $y(u)$ known? Maple is treating $y(u)$ as a constant when `dsolve()` is applied.

```
    |\^/|     Maple V Release 4
._|\|   |/|_. Copyright (c) 1981-1996 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.

> dsolve(diff(z(t),t) = int(y(u),u = 0 .. t), z(t));

                                        2
                          z(t) = 1/2 y(u) t  + _C1
```

If instead we were to use a variable a:

```
> dsolve(diff(z(t),t) = int(a,u = 0 .. t), z(t));
```

```
                                        2
                              z(t) = 1/2 a t  + _C1
```

If however y(u) were defined:

```
> y(u):=u^3;
                                         3
                                   y(u) := u

> dsolve(diff(z(t),t) = int(y(u),u = 0 .. t), z(t));

                                        5
                               z(t) = 1/20 t  + _C1
```

I will however report this problem to our Math developers to investigate further.

7.49.3 Stanley J Houghton

I feel y(u) is clearly a funtion of a variable u and int(y(u),u=0..t) is clearly a function of t, although Maple certainly appears to be evaluating

```
int(y(u),u=0..t)
```

incorrectly to

```
y(u) * t
```

I think this is unreasonable and an error. Maple doesn't evaluated the integral above wrongly in any other context. If the form of y is unknown, the integral should surely remain unevaluated. There is something wrong with Maple's interpretation of `int(y(u),u=0..t)` in this context but I am not sure what.

7.49.4 Douglas B. Meade

While Stanley Houghton admits that his problem is a "very much simplified version" of the real problem of interest, I hope the following comments are of use.

Here is the original problem

```
> ODE1 := diff( z(t), t ) = Int( y(u), u=0..t );

                                        t
                                       /
                              d        |
                   ODE1 := -- z(t) =   |   y(u) du
                              dt       |
                                       /
                                        0
```

and the solution obtained using Release 4

```
> dsolve( ODE1, z(t) );

                               t
                              /
                              |                      2
                z(t) = y(u) t |   1 du - 1/2 y(u) t  + _C1
                              |
                             /
                              0
```

Note that the original problem can be written in a more traditional form by differentiating the equation

```
> ODE2 := diff( ODE1, t );

                                  2
                                 d
                      ODE2 := --- z(t) = y(t)
                                   2
                                 dt
```

Using the fact that `D(y)(0)=0`, Maple reports the solution:

```
> dsolve( { ODE2, D(z)(0)=0 }, z(t) );

                      t                t
                     /                /
                    |                |
          z(t) = - |   u y(u) du +  |   y(u) du t + _C1
                    |                |
                   /                /
                    0                0
```

This is, presumably, closer to what Stan would like to see.

It would seem to me that dsolve could reasonably apply transformations that convert the input argument into a form that can be handled. Of course, I can see situations where these transformation would not be appropriate, or other problems might arise.

7.49.5 Joe Riel

As a workaround you can convert the integro-differential equation to a pure differential equation,

```
deq := diff(z(t),t) = int(y(u),u=0..t):
initcond := eval(subs(t=0, convert(deq,D))):
dsolve({initcond,diff(deq,t)},{z(t)});

                      t                t
                     /                /
                    |                |
          z(t) = - |   u y(u) du +  |   y(u) du t + _C1
                    |                |
                   /                /
                    0                0

subs(%,deq): %;

                       t              t
                      /              /
                     |              |
```

```
                    |   y(u) du =  |   y(u) du
                    |              |
                   /              /
                     0              0
```

7.49.6 Stanley J Houghton

Many thanks for the suggestion. I can indeed apply your workround to the original problem and get the correct result.

I have also discovered that I get the same result if I use Laplace mode in dsolve and, again, the result is correct if I solve

```
diff(z(t),t)=f(t)
```

and then evaluate the result after assigning

```
f:=t->int(y(u),u=0..t);
```

However, the important issue is the way 'frontend' handles functions of the form `int(y(u),u=0..t)` for it is in frontend (used within `'dsolve'`) that the misinterpretation occurs. It tries to freeze $y(u)$ independent of the second argument to int. As a result, the problem extends to other activities involving such functions.

This has been passed back to Maple who are considering it carefully. I am sure there are other ramifications.

7.49.7 Robert Israel

Stanley J Houghton wrote:

```
> dsolve(diff(z(t),t) = int(y(u),u = 0 .. t), z(t));

   z(t) = 1/2*y(u)*t^2+_C1
```

Yes, certainly it's a bug. I guess "dsolve" doesn't expect unevaluated integrals in the equations.

As a workaround, I would use something like this:

```
> dsolve(diff(z(t),t) = Y(t), z(t));
```

```
                                  /
                                  |
                        z(t) =    |  Y(t) dt + _C1
                                  |
                                 /
```

```
> subs(Y(t)=int(y(u),u=0..t), %);
```

```
                                    t
                                  /   /
                                 |   |
                       z(t) =    |   |  y(u) du dt + _C1
                                 |   |
                                /   /
                                    0
```

```
> diff(%,t);
```

```
                                      t
                                      /
                                     |
                          d          |
                       ---- z(t) =   |  y(u) du
                          dt         |
                                     |
                                    /
                                    0
```

7.50 bug in dsolve in Maple V.5 (20.4.98)

7.50.1 Dr Francis J. Wright

There appears to be a new bug in Release 5 that prevents Maple from solving odes with conditions if the unknown function is an indexed name rather than a symbol.

Release 4 could do this. I think that the problem is in the library procedure `‘dsolve/IC/format‘`. The worksheet

`http://www.maths.qmw.ac.uk/~fjw/public_maple/dsolve_IC_format.mws`

proposes a fix and gives a test example, although I have not investigated this very deeply nor tested it very thoroughly. I would be interested to hear of examples where my fix also fails. The problem is essentially that this procedure tries to treat functions, e.g. `y(0)`, and derivatives, e.g. `D(y)(0)`, the same, in a way that happens to work if y is a symbol but not if it is indexed, e.g. `z[1]`.

(The indexed notation `z[i]` is pleasanter than the symbol notation `z.i` when dealing with a system of odes.)

7.50.2 Edgardo S. Cheb-Terrab (26.6.98)

Yes, it was in `‘dsolve/IC/format‘`. This problem was detected in January, and posted in `sci.math.symbolic`. A fix to this is also provided inside the `ODEtools` package available at `http://lie.uwaterloo.ca/odetools.html`.

7.51 bug in dsolve with initial condition in Maple 6 (29.6.00)

7.51.1 Federico Rocchi

I have a problem with dsolve. I'm using version 6 under Windows 98. I'd like to solve the following simple equation:

```
> restart;
> eq:=diff(f1(T),T)+diff(f0(T),T)=-k*f1(T);
> dsolve({eq},f1(T));

        f1(T) = (Int(-diff(f0(T),T)*exp(k*T),T)+_C1)*exp(-k*T)
```

This solution seems indeed correct. I'd like now to impose the initial condition f1(T0)=0; the command

```
> dsolve({eq,f1(T0)=0},f1(T));
```

fails to give the answer; I waited about 20 minutes and then interrupted the calculation. Even if I understand that Maple doesn't know the form of `f0(T)` I'd like at least to have the symbolic result `f1(T)` cast in the following form:

```
exp(-k*T)*(intat(exp(k*x)*diff(f0(x),x),x=T0)
           -intat(exp(k*x)*diff(f0(x),x),x=T)
```

or other equivalent forms.

Am I missing anything, making errors in Maple or in the math? Or is there a bug?

7.51.2 Edgardo S. Cheb-Terrab (15.7.00)

The ODE problem you are showing is indeed easy and Maple's failure is unexpected. Both the root of the failure and a workaround are as follows.

```
    > eq:=diff(f1(T),T)+diff(f0(T),T)=-k*f1(T);

                        eq := f1' + f0' = -k f1
```

So: your unknown is `f1(x)`. You also have the "derivative" of `f0` - an unknown function somehow parameterizing your problem. The workaround is: represent `f0'(x) as g0(x)` (that is: not as a derivative):

```
> diff(f1(T),T) + g0(T) = -k*f1(T);

                      f1' + g0(T) = -k f1

> dsolve( { %, f1(T0)=0 }, f1(T) );

                      T
                     /
                    |
              f1 =  |   -f0(u) exp(k u) du exp(-k T)
                    |
                   /
                    T0
```

The answer above is correct. Now, where is the root of Maple's failure? Try debug(limit) and you will see it. To compute the solution above, dsolve calls limit (this is reasonable), and when doing that with your original "eq", limit receives:

```
> exp(-k*T)*(Int(-diff(f0(u),u)*exp(k*u),u = T0 .. T)+_C[1]);

       /   T                                  \
       |  /                                   |
       | |    /d       \                      |
       | |   -|-- f0(u)| exp(k u) du + _C[1]| exp(-k T)
       | |    \du      /                      |
       |/                                     |
       \  T0                                  /

> limit( %, T = T0);
```

In above, because of a bug, limit enters an infinite loop (routines below limit get into confusion due to the presence of a derivative in the integrand).

Of course this confusion below limit should not happen - a derivative in the integrand should not be a problem. Anyway, that is why the workaround consists of "avoid that

derivative in your original eq, for instance representing `diff(f0(T),T) by g0(T)`".

7.52 bug in dsolve, Maple 8, and workaround (16.1.03)

7.52.1 Roberto Sussman

I found a bug when applying dsolve in the case when one of the ODE's is defined by means of piecewise. Using piecewise is necessary when you need to provide initial conditions at $x = 0$ when you have terms of the form `1/x` (and you know there is no singularity). The following instructions illustrate the problem:

```
> e1:=diff(mu(x),x)=x^2*exp(psi(x));
                              d                          2
                      e1 := -- mu(x) = x  exp(psi(x))
                              dx

> e2:=diff(psi(x),x)=piecewise(x>0, -mu(x)/x^2, x=0,0);
                                      {   mu(x)
                           d          { - -----        0 < x
                  e2 := -- psi(x) = {     2
                           dx        {    x
                                      {
                                      {    0           x = 0

> S:=dsolve({e1,e2,mu(0)=0,psi(0)=0},{mu(x),psi(x)}, type=numeric);
Error, (in f) unable to store 'piecewise()' when datatype=float[8]
```

These are the equilibrium equations for an isothermal sphere (a standard text book problem in newtonian astrophysics) and initial conditions should be given as `mu(0)=0,psi(0)=0` for physical reasons. However, since there is a term `mu(x)/x^2` in the second equation, I normaly used piecewise to be able to set these initial conditions.

This works very well in all previous Maple releases (from V5.1 to 7). I am using command line version of Maple8 for the Mac OSX but the same problem occurs in Windows and linux versions of Maple 8.

Is there a new feature of dsolve that allows you to set this type of initial conditions without using piecewise or is this a real bug?

It is corrected with Maple 9. (U. Klein)

7.52.2 Allan Wittkopf (30.1.03)

Yes, this is indeed a bug.

There is, however, a workaround for this bug.

`dsolve/numeric` allows a procedure-based specification of a system. This means that you can define the rhs of the ODE as a computation procedure to be used by `dsolve/numeric`, and allows much more flexibility than just piecewise.

Information on this can be found in `?numeric,IVP`.

The relevant options are `'procedure','procvars','initial', and 'start'`.

For the same problem in Maple 8, we make the mapping:

```
Y[1] <-> mu(x), YP[1] <-> diff(mu(x),x)

Y[1] <-> eta(x), YP[1] <-> diff(eta(x),x)
```

We can then define:

```
> pvars := [mu(x),psi(x)];
                            pvars := [mu(x), psi(x)]

> dproc := proc(N,x,Y,YP)
>    YP[1] := x^2*exp(Y[2]);
>    if x>0 then
>       YP[2] := -Y[1]/x^2;
>    else
>       YP[2] := 0;
>    end if;
> end proc;

dproc := proc(N, x, Y, YP)
    YP[1] := x^2*exp(Y[2]);
    if 0 < x then YP[2] := -Y[1]/x^2 else YP[2] := 0 end if
end proc

> init := array([0,0]);
                                  init := [0, 0]
```

Where I note that I used an 'if' condition to avoid $x = 0$ for the equation for YP[2] (i.e. diff(eta(x),x)).

Now call dsolve/numeric with this information:

```
> S := dsolve(numeric,procedure=dproc,procvars=pvars,initial=init,start=0);
                        S := proc(x_rkf45)  ...  end proc
```

And things are identical to the prior versions:

Maple 8:

```
> S(1);
    [x = 1., mu(x) = 0.302901326325882625, psi(x) = -0.158827757643356910]

> S(10);
     [x = 10., mu(x) = 25.1061282275108368, psi(x) = -3.73655959664163140]
```

Maple 7:

```
> e1:=diff(mu(x),x)=x^2*exp(psi(x));
                               d           2
                        e1 := -- mu(x) = x  exp(psi(x))
                               dx

> e2:=diff(psi(x),x)=piecewise(x>0, -mu(x)/x^2, x=0,0);
                                       {   mu(x)
                            d          { - -----        0 < x
                   e2 := -- psi(x) = {     2
                            dx        {    x
                                       {
                                       {    0           x = 0

> S:=dsolve({e1,e2,mu(0)=0,psi(0)=0},{mu(x),psi(x)}, type=numeric);
                        S := proc(rkf45_x)  ...  end proc

> S(1);
     [x = 1., mu(x) = .302901367581909331, psi(x) = -.158827753673755290]

> S(10);
```

```
     [x = 10., mu(x) = 25.1061294255031591, psi(x) = -3.73655980695852863]
```

7.52.3 Preben Alsholm (30.1.03)

Removing the piecewise part seems to do it:

```
> e1:=diff(mu(x),x)=x^2*exp(psi(x)):

> e2:=diff(psi(x),x)=-mu(x)/x^2:
> S:=dsolve({e1,e2,mu(0)=0,psi(0)=0},{mu(x),psi(x)}, type=numeric):
> S(.3);

 [x = 0.3, mu(x) = 0.00891969113809897886,

       psi(x) = -0.0149329128389725352]

> plots[odeplot](S,[[x,mu(x)],[x,psi(x)]],x=0..5);
```

7.52.4 Robert Israel (30.1.03)

Apart from the Maple problem, there's a mathematical problem here. Putting a discontinuous term into a differential equation is likely to result in having no solutions, and you can't expect Maple's numerics to handle the delicate issues of analysis.

What I would do is find a series solution near $x = 0$, evaluate it at some positive x (hopefully well within the radius of convergence), and use that as the starting point for a numerical solution.

In this case, after noting that according to the first equation, in any solution that is analytic at 0 we must have `mu(x) = O(x^3)`:

```
> Order:= 17;
  e1:=diff(mu(x),x)=x^2*exp(psi(x));
  e2:=diff(psi(x),x) = -mu(x)/x^2;
  S:= {mu(x) = add(a[i]*x^i, i=3..17), psi(x) = add(b[i]*x^i, i=1..15)};
  se1:= series(eval(lhs(e1)-rhs(e1), S), x);
  se2:= series(eval(lhs(e2)-rhs(e2), S), x);
  eqs:= {coeffs(convert(se1,polynom),x), coeffs(convert(se2,polynom),x)};
  solve(eqs);
```

```
             2869                -629                168947
  {b[12] = -----------, b[10] = ---------, a[15] = ------------,
           13135122000          224532000          689593905000

                61                -2869               629
        b[8] = -------, a[13] = ----------, a[11] = --------,
               1632960          1094593500          22453200

                -61                           -1          -1
        a[9] = ------, a[7] = 1/315, a[5] = --, b[6] = ----,
               204120                        30          1890

                                -90151991
        b[4] = 1/120, a[17] = ----------------, b[1] = 0,
                              3938960385360000

                   -168947
        b[14] = -------------, a[4] = 0, b[3] = 0, b[2] = -1/6,
                 9654314670000

        a[6] = 0, b[5] = 0, a[8] = 0, b[7] = 0, a[10] = 0, b[9] = 0,

        a[12] = 0, b[11] = 0, a[14] = 0, b[13] = 0, a[16] = 0,

        b[15] = 0, a[3] = 1/3}

> Ssol:= subs(%, S);

                            2          4           6     61     8
  Ssol := {psi(x) = -1/6 x  + 1/120 x  - 1/1890 x  + ------- x
                                                     1632960

              629     10      2869      12      168947      14
         - --------- x   + ---------- x   - ------------ x   , mu(x)
           224532000       13135122000       9654314670000

                3         5          7     61    9     629     11
         = 1/3 x  - 1/30 x  + 1/315 x  - ------ x  + -------- x
```

```
                                          204120       22453200

            2869      13       168947      15        90151991       17
        - ---------- x    + ------------ x    - ---------------- x  }
          1094593500        689593905000        3938960385360000
```

The radius of convergence seems to be well over 1, since the coefficients are decreasing in size. This series solution should be very accurate at, say,`x = 1/2`.

```
> IC:= eval(Ssol,x=0.5);

      IC := {psi(0.5) = -0.04115395731, mu(0.5) = 0.04064923128}

> Nsol:= dsolve({e1,e2, op(IC)},{psi(x),mu(x)}, numeric);
```

This numerical solution should be good for `x >= 1/2` .

7.52.5 Roberto Sussman (3.2.03)

Thanks for your replies

Preben Alsholm's remark (problem goes out by removing piecewise) is correct `**BUT ONLY**` in Maple 8. If you try this system of ODE's in previous releases you get error messages

With Maple 7:

```
"Error, (in S) cannot evaluate the solution further right of 0., probably a singularit
```

With Maple V5.1 and Maple 6:

```
"Error, (in s) division by zero"
```

That is why I solved this problem in earlier releases with piecewise and it worked very well.

Robert Israel's remarks are absolutely relevant: in general you have to test the behavior of the involved terms near x=0 (or near any possible pole). However, in the case of the ODE system I mentioned (the isothermal sphere) I already knew that regularity at $x = 0$ is fulfiled.

The key issue is that dsolve numerical should be able to handle piecewise functions as it did in previous releases. There should be (I guess) some Maple related problem in

the example I mentioned. Am I reporting a Maple bug or (accidentaly) finding a Maple feature???

7.53 Bug in dsolve, Maple V.4 (24.5.96)

7.53.1 Alain Leroux

I found a bug in Maple V-4: when you use dsolve with initial conditions, and when there are two or more values of the constants that can be found, Maple gives just one solution. Example

```
> A:=A0*(1-ksi(t)):Equ:=diff(A,t)=-k*sqrt(A):

> dsolve({Equ,ksi(0)=0},ksi(t));

                                      2  2             1/2
                                     t  k  + 4 t k A0
                    ksi(t) = - 1/4 ------------------
                                           A0
```

But there are two solutions.

The workaround for this bug is to patch the end of ‘dsolve/consts‘:

```
...
    csol := solve(eqns, consts);
    if csol = NULL then NULL
    else
        csol:={csol};
        if nops(csol)=1 then
             Solns := subs(op(csol), Solns);
        else Solns := map(subs,csol,op(Solns));
        fi;

                if (type(solns, 'set') or type(Solns,`set`)) then
RETURN({Solns});
                else RETURN(Solns);
                fi
    fi end:
```

But to use this patch, you must unprotect dsolve.

```
>unprotect(dsolve);with(libphys):
```

```
> dsolve({Equ,ksi(0)=0},ksi(t));

                  2  2            1/2                  2  2            1/2
                 t  k  + 4 t k A0                    t  k  - 4 t k A0
 {ksi(t) = - 1/4 -------------------, ksi(t) = - 1/4 -------------------}
                          A0                                  A0
```

The bug is removed with Maple V Release 5. (U. Klein)

7.53.2 Preben Alsholm (29.5.96)

There is indeed a bug in dsolve (R4), but the problem is not that only one solution is reported for the initial value problem given by Alain Leroux. The problem is that the wrong solution is reported!

The initial value problem has a unique solution (locally, - by the general theorem on uniqueness).

Consider the following two simplified versions:

```
equ1:=diff(x(t),t)=sqrt(1-x(t)):

dsolve( {equ1, x(0)=0}, x(t));

                    x(t)= -1/4*t^2+t

# This solution is correct

equ2:= -diff(x(t),t)= -sqrt(1-x(t)):

dsolve( {equ2, x(0)=0}, x(t));

                    x(t)= -1/4*t^2-t
```

The last solution is obviously incorrect, as equ2 is equivalent to equ1. That the correct solution can be joined with the constant solution $x = 1$ at $t = 2$ is irrelevant in this context.

7.53.3 Yuri Muzychka (4.6.96)

Try this one

$$y(x)'' + 2/xy(x)' - m^2 y(x) = 0$$

the solution in R4 is mixed exponentials and hyperbolics. Go figure! Why not just return hyperbolics or exponentials as R2 and R3 did.

7.54 bug in dsolve, Maple V.5 (27.4.98)

7.54.1 Dr. Winfried Auzinger

Similarly as Francis J. Wright, I encountered problems with the new version of dsolve in Rel.5: Consider the following example:

```
restart;
f1:=a11*y1(x)+a12*y2(x)+b1(x);
f2:=a21*y1(x)+a22*y2(x)+b2(x);
odesys:={diff(y1(x),x)=f1,diff(y2(x),x)=f2,y1(0)=y1ini,y2(0)=y2ini};
dsolve(odesys,{y1(x),y2(x)});
```

Under Rel.4 this yields a correct representation of the solution of such a system. Evidently it does not work under Rel.5; the output contains strange things like integrals where the lower limit is the null string and the upper limit is 0. (?)

This bug is removed with the new ODEtools package of Maple V.5 (U. Klein)

7.54.2 Willard, Daniel, Dr., DUSA-OR (29.4.98)

I had no problem with R5. I did write `f1:=x->etc and diff(y1(x),x)=f1(x)`. Used Windows 95.

7.54.3 Heike Koch-Beuttenmueller (4.5.98)

Some days ago, there was a question to the MUG about the representation changes of differential equations, but the expression was very great. At our university there occurred a similar problem with a much smaller expression:

The problem:

```
eq1:=(D@@2)(x)(t)+k^2*x(t)=f(t);

                      (2)           2
             eq1 := (D   )(x)(t) + k  x(t) = f(t)

> i1:=x(0)=0;

                         i1 := x(0) = 0
> i2:=D(x)(0)=0;
```

```
                    i2 := D(x)(0) = 0
```

The representation of the result in Maple V Release 4:

```
dsolve({eq1,i1,i2},x(t));

       /    t
       |   /
       |  |
x(t) = |- |   sin(k u) f(u) du cos(k t)
       |  |
       | /
       \   0

          t                          \
         /                           |
        |                            |
    +   |   cos(k u) f(u) du sin(k t)|/k
        |                            |
       /                             |
         0                           /
```

but in Release 5 we get:

WARNING: ‘dsolve‘ has been extensively rewritten, many new result forms can occur and options are slightly different, see help page for details

```
> dsolve({eq1,i1,i2},x(t));

           /
          |  cos(k t) f(t)
  x(t) =  |  ------------- dt sin(k t)
          |        k
         /

             /
            |    sin(k t) f(t)
        +   |  - ------------- dt cos(k t)
```

```
          |          k
        /

          0
         /
          |   cos(k _a) f(_a)
      -   |   --------------- d_a sin(k t)
          |          k
        /


          0
         /
          |   sin(k _a) f(_a)
      +   |   --------------- d_a cos(k t)
          |          k
        /
```

The question is now , how the different integrals have to be interpreted and why the representation was changed?

7.54.4 Edgardo S. Cheb-Terrab (30.6.98)

The answer by Maple R5 is correct. You can verify this using the new command odetest:

```
> dsolve(odesys,{y1(x),y2(x)});      # large output here...
> odetest(%, remove(has,odesys,0));

                                {0}
```

Note that in Maple R5 there is a new command, intat, for representing integrals `"evaluated at a point" (see ?intat)`. Intats are typically useful to represent integrals evaluated at a point which cannot be an integration variable.

The visual representation for intats (with an upper limit of integration representing the evaluation point) is not so unusual - see for instance: W.E. Boyce and R.C. Diprima, "Elementary Differential Equations and Boundary Value Problems", or at a more advanced level: G.W. Bluman and S. Kumei, Symmetries and Differential Equations, Applied Mathematical Sciences 81, Springer-Verlag (1989).

I agree however in that for boundary value problems, it would more interesting to have - by default - definite integrals instead of intats.

Possible solutions to your problem are:

1) by installing the ODEtools package, dsolve automatically avoids using intats in boundary value problems. For example, for your odesys, you receive an answer without intats, as in Maple R4. ODEtools is available at `http://lie.uwaterloo.ca/odetools.html`.

2) ODEtools also provides a small conversion routine, `'intat_to_int'`, which I'm pasting below. This routine converts intats found in a given expression into "equivalent" (up to an integration constant) definite integrals as follows:

```
> intat(f(x),x=a);
                                 a
                                /
                               |
                               |   f(x) dx
                               |
                              /

> intat_to_int(%) = intat_to_int(%, definite);

                                       a
                          /            /
                         |            |
                         |  f(a) da =  |   f(x) dx
                         |            |
                        /            /
                                      0
```

Note that when converting an intat into a definite integral, the resulting expression will be different from the original one by an integration constant (the integral evaluated at "0").

Here are the two subroutines conforming the `intat_to_int` command.

```
intat_to_int := proc(expr)
local ans, extra, ii, zz;
```

```
if has(expr,['intat,Intat,Int']) then
        zz := sort([op(indets(expr,{'intat,Intat,Int'}))],
                (a,b) -> not has(a,b));
        if zz = [] then
                expr
        else
                ans := expr;
                if nargs > 1 then
                    if nargs > 2 then
                        ERROR(`Too many arguments; expected only two arguments`);
                    elif args[2] <> 'definite' then
                        ERROR(`Expected second argument to be 'definite',
                                received`,args[2])
                    else
                        extra := 1
                    fi;
                else
                    extra := NULL;
                fi;
                to nops(zz) do
                        ii := zz[1]=`intat_to_int/do`(zz[1],extra);
                    zz := subs(ii,subsop(1=NULL,zz));
                    ans := subs(ii,ans);
                od;
        fi;
else
        expr
fi;
end:


`intat_to_int/do` := proc(II)
local dx_at_t, integrand, t, x;

if op(0,II)='Int' then
    int(op(II))
else               # it is an intat or Intat

# determine the integrand and the new integration variable dx
```

```
    integrand := op(1,II);
    dx_at_t := op(2,II);
    x := lhs(dx_at_t);
    t := rhs(dx_at_t);
    if nargs = 1
    and x <> '_Z'
    and type(t,'name')
    and t <> '_Z'
    and not has(integrand,t) then
        int(subs(dx_at_t,integrand), t);
        elif t = 0 then
        int(integrand, x=`PDEtools/_C`(II)..t);
    else
        int(integrand, x=0..t);
    fi;
fi;
end:
```

7.54.5 Harald Pleym (1.5.98)

First some general comments on R5.

I have spent some time during the last month to convert my worksheets from R4 to R5. During that work I have discovered a lot of new errors in R5 (all these are reported to the Support), which was not present in R4. Fortunately the opposite is also true. Maplesoft should have spent much more time testing R5 before they released it. Despite the new fancy things in R5, it seems to me that R5 is much of a RUSH WORK and that is not to Maples credit. I have been told that Maple will correct some of the most serious errors in R5 in a new patch in the near future.

We intend to install R5 in our network before the next semester (term) and introduce R5 for all our students. But in lack of an updated R5, I think I prefer to continue using R4 in classes.

So to the unnessecary complicated solution of the following differential equation in R5 compared to R4. Please let me know if someone has a shorter way to end up with the R4 solution than I have made below.

R5 solution:

```
> ode:= r*diff(R(r),r)+r^2*diff(R(r),r$2) = lambda^2*R(r);
                                    / 2      \
                   /d      \    2 |d       |         2
          ode := r |-- R(r)| + r  |--- R(r)| = lambda  R(r)
                   \dr     /      |  2     |
                                  \dr      /
>sol:=dsolve(ode1,R(r));

             sol := R(r) = _C1 cosh(lambda ln(r)) + _C2 sinh(lambda ln(r))
```

This equation is uneccessary complicated for further use. To compare, the solution in R4 is:

```
                    solR4 := R(r) = _C1*r^lambda+_C2*r^(-lambda);
```

You have to manipulate the R5 solution sol to get the R4 solution, which I prefer for further use, for instance in this way:

```
> combine(convert(sol,exp));

           R(r) = _C1 (1/2 exp(lambda ln(r)) + 1/2 exp(-lambda ln(r)))
                  + _C2 (1/2 exp(lambda ln(r)) - 1/2 exp(-lambda ln(r)))

> simplify(%);
                   lambda             (-lambda)             lambda
   R(r) = 1/2 _C1 r       + 1/2 _C1 r           + 1/2 _C2 r

                                      (-lambda)
                         - 1/2 _C2 r

>subs(r^lambda=a,r^(-lambda)=b,sol2);

                   R(r) = 1/2 _C1 a + 1/2 _C1 b + 1/2 _C2 a - 1/2 _C2 b

> collect(%,[a,b]);

                   R(r) = (1/2 _C1 + 1/2 _C2) a + (1/2 _C1 - 1/2 _C2) b
```

```
>subs(a=r^lambda,b=r^(-lambda),op([2,1,1],%)=C1, op([2,2,1],%)=C2,%);

                                  lambda       (-lambda)
                      R(r) = C1 r       + C2 r
```

7.54.6 Edgardo S. Cheb-Terrab (26.6.98)

Although the problem you are mentioning is not directly related to the ODEtools routines (dsolve R5 is based on ODEtools '97), by installing the present version of ODEtools (`http://lie.uwaterloo.ca/odetools.html`), the answer to your linear ODE (actually an Euler ODE) is as straightforward as in R4:

```
> ode:= r*diff(R(r),r)+r^2*diff(R(r),r$2) = lambda^2*R(r);

                                    2             2
                   ode := r R' + r  R'' = lambda  R

> dsolve(ode);  # Maple R5 with ODEtools

                             lambda        (-lambda)
                   R = _C1 r       + _C2 r
```

7.54.7 Willard, Daniel, Dr., DUSA-OR ((29.6.98))

Use:

```
>simplify(convert(%,exp)); #This works.
```

7.55 Bug in eigenvectors with LinearAlgebra in Maple 6 (27.11.00)

7.55.1 Jean Brillet

Is there a bug in "LinearAlgebra[Eingenvectors]" function (MAPLE 6.01) ?

```
> restart;
> A:=<<sqrt(2),1>|<1,0>>;

                        [sqrt(2)    1]
                   A := [           ]
                        [   1       0]
```

linalg gives the correct answer

```
> linalg[eigenvectors](A);

[1/2 sqrt(2) + 1/2 sqrt(6), 1, {[1/2 sqrt(2) + 1/2 sqrt(6), 1]}],

[1/2 sqrt(2) - 1/2 sqrt(6), 1, {[1/2 sqrt(2) - 1/2 sqrt(6), 1]}]
```

but LinearAlgebra gives

```
> LinearAlgebra[Eigenvectors](A);

                  [1/2 sqrt(2) + 1/2 sqrt(6)]  [0    0]
                  [                         ], [     ]
                  [1/2 sqrt(2) - 1/2 sqrt(6)]  [0    0]
```

Floating result (with `sqrt(2.)` in A) is correct.

It is corrected with Maple 7. (U. Klein)

7.56 Bug in Elliptic Integral, Maple 6 to Maple 8 (24.9.01)

7.56.1 James R. FitzSimons

Maple is giving the wrong answer using elliptic integrals.

```
> evalf(Int(sqrt((x^2+1)/(x+2)), x=-1..5));

                              7.277500982

> evalf(int(sqrt((x^2+1)/(x+2)), x=-1..5));

                      14.18222461 + 4.701625434 I
```

This is the numeric result using DERIVE. 7.2775009819508996444

7.56.2 Koch-Beuttenmueller (26.9.01)

In MapleVr5.1 the results were still identical, but since Maple6 they are different:

```
> i1:=evalf(Int(sqrt((x^2+1)/(x+2)), x=-1..5));

                           i1 := 7.277500982

> i2:=int(sqrt((x^2+1)/(x+2)), x=-1..5);

  i2 := 2/3 sqrt(182) - 4/1365 I sqrt(182) sqrt(2 - 10 I)

        sqrt(70 + 35 I) sqrt(2) sqrt(1 + 5 I)

        EllipticE(1/2 sqrt(2 - 10 I), %1) + 1/273 I sqrt(182)

        sqrt(2 - 10 I) sqrt(70 + 35 I) sqrt(2) sqrt(1 + 5 I)

        EllipticF(1/2 sqrt(2 - 10 I), %1) - 2/1365 sqrt(182)

        sqrt(2 - 10 I) sqrt(70 + 35 I) sqrt(2) sqrt(1 + 5 I)
```

```
        EllipticE(1/2 sqrt(2 - 10 I), %1) - 2/3 sqrt(2) + 8/15 I

        sqrt(2 + 2 I) sqrt(10 + 5 I) sqrt(1 - I)

        EllipticE(1/2 sqrt(2 + 2 I), %1) - 2/3 I sqrt(2 + 2 I)

        sqrt(10 + 5 I) sqrt(1 - I) EllipticF(1/2 sqrt(2 + 2 I), %1)

         + 4/15 sqrt(2 + 2 I) sqrt(10 + 5 I) sqrt(1 - I)

        EllipticE(1/2 sqrt(2 + 2 I), %1)

  %1 := 1/5 sqrt(10 - 20 I)

> evalf(%);

                                                -8
                      7.277500973 + .3 10    I

> interface(version);

  Maple Worksheet Interface, Release 5.1, SUN SPARC SOLARIS, Jan 7\
         1999

> i2:=int(sqrt((x^2+1)/(x+2)), x=-1..5);
  i2 := -2/6825 I sqrt(-45 - 35 I) sqrt(-45 + 35 I) sqrt(70 + 35 I)

        sqrt(182) EllipticF(1/5 sqrt(70 + 35 I), %1) + 2/3 sqrt(182)

         + 4/6825 sqrt(-45 - 35 I) sqrt(-45 + 35 I) sqrt(70 + 35 I)

        sqrt(182) EllipticE(1/5 sqrt(70 + 35 I), %1) - 2/3 sqrt(2) -

        4/75 sqrt(15 - 5 I) sqrt(15 + 5 I) sqrt(10 + 5 I) sqrt(2)

        EllipticE(1/5 sqrt(10 + 5 I), %1) + 2/75 I sqrt(15 - 5 I)

        sqrt(15 + 5 I) sqrt(10 + 5 I) sqrt(2)
```

```
        EllipticF(1/5 sqrt(10 + 5 I), %1)

  %1 := 2/5 sqrt(5) - 1/5 I sqrt(5)

> evalf(%);

                    14.18222461 + 4.701625434 I

> interface(version);

  Maple Worksheet Interface, Maple 6.01, SUN SPARC SOLARIS, June 9  2000 Build ID 7951
```

Until now I told my students it is good to use different CAS to test if an integral is right. Now I can even tell them use different Maple versions to see if the results can be correct ?

7.56.3 Gerald A. Edgar (26.9.01)

It may be related to choice of square-roots...

```
> evalf(int(sqrt((x^2+1)/(x+2)),x=-1..5));

                    14.18222460 + 4.701625434 I

> evalf(int(sqrt(x^2+1)/sqrt(x+2),x=-1..5));

                             7.277500972
```

7.57 bug in elliptic integral, Maple 6 to Maple 8 (28.8.01)

7.57.1 James R. FitzSimons

What did I do wrong? Maple 6 did not give me an answer.

```
> evalf(int(327082769*(689*t-sqrt(689*sqrt(6005)/2+23377)-373)/(sqrt(474721*t^2
> +t*(sqrt(654165538*sqrt(6005)+44390211268)-513994)-sqrt(191787284*sqrt(6005)
> +12735814988)+689*sqrt(6005)+139178)*(474721*t^2-t*(sqrt(654165538*sqrt(6005)
> +44390211268)+513994)+sqrt(191787284*sqrt(6005)+12735814988)+689*sqrt(6005)
> +139178)^(3/2))
> +I*(sqrt(7371138192815137l729*sqrt(6005)/2
> -2500944811806087978097)-2289579383)/(sqrt(474721*t^2+t*(sqrt(654165538
> *sqrt(6005)+44390211268)-513994)-sqrt(191787284*sqrt(6005)+12735814988)
> +689*sqrt(6005)+139178)*(474721*t^2-t*(sqrt(654165538*sqrt(6005)+44390211268)
> +513994)+sqrt(191787284*sqrt(6005)+12735814988)+689*sqrt(6005)+139178)^(3/2))
> ,t=0..1));

Error, (in int/ellalg/trxstandard/4) int/ellalg/trxstandard/4 uses a 7th
argument, L, which is missing
```

7.57.2 Robert Israel (29.8.01)

You've run into a bug in Maple's evaluation of elliptic integrals. Note, however, that if you wanted a numerical answer, you should use "Int" instead of "int". The answer I get is `-20.69727045+41.02175888*I`.

The way it is now, "int" is called first to do the integration in closed form, and then (if that didn't run into the bug) "evalf" would evaluate the result. With `evalf(Int(...))`, numerical integration would be used with no attempt to integrate symbolically.

Maple 7 didn't return the error message, but it also didn't return any answer in a reasonable time, so I'm not sure if the bug has been fixed.

Maple 8 returns: Error, (in gcdex) invalid arguments (U. Klein)

7.57.3 Jim FitzSimons (29.8.01)

Robert, thank you, I did want the closed form solution first. Maple usually runs faster that way when the results are 64 digits. George's algorithms are very good. I have my own algorithms for symbolic and numerical evaluation using symmetric ellpitic integrals.

`http://www.getnet.net/~cherry/derive/`

I did not program them using Maple yet. Maybe somebody better at using Maple will do it.

See also: bug in integration 3 (U. Klein)

7.58 Bug in Elliptic Integral, Maple V to Maple 8 (25.7.01)

7.58.1 Tim McLarnan

I'm not a big expert on special functions, but it appears to me that Maple has been incorrectly calculating the values of incomplete elliptic integrals at least since Maple V. The problem I'll describe here exists on my Intel box under both Maple 6 and Maple 7. The values computed by Maple V release 5 on a Mac are slightly different from Maple 7's answers, but also wrong.

The errors are not small - in one example, Maple computes a value of `1.80+0.62i` when the correct answer is `0.27+0.97i`.

Here's an example with the EllipticE function. Maple reports the value

```
> evalf(EllipticE(1/5*sqrt(70+35*I),2/5*sqrt(5)-1/5*I*sqrt(5)));

                    1.801868551 + .6195716925 I
```

On the other hand, the EllipticE function is defined by an integral which Maple can evaluate explicitly,

```
> z := evalf(1/5*sqrt(70+35*I)):
  k := evalf(2/5*sqrt(5)-1/5*I*sqrt(5)):
  evalf(Int(sqrt(1-k^2*t^2)/sqrt(1-t^2),t=0..z));

                    .2742693842 + .9709289578 I
```

(This second calculation takes Maple close to 10 minutes on my computer; MuPAD confirms the result in a couple of seconds.)

I think the results of these calculations should be equal, and they are not.

Similarly, with the EllipticF function, Maple gives the value

```
> evalf(EllipticF(1/5*sqrt(70+35*I),2/5*sqrt(5)-1/5*I*sqrt(5)));

                   -.008504510023 + 1.374884128 I
```

Working out numerically the defining integral for this function yields (again confirmed

by MuPAD)

```
> z := evalf(1/5*sqrt(70+35*I)):
  k := evalf(2/5*sqrt(5)-1/5*I*sqrt(5)):
  evalf(Int(1/sqrt(1-t^2)/sqrt(1-k^2*t^2),t=0..z));

                    2.193595226 + .5178375879 I
```

Again, I think these two calculations should have given the same result, and they do not.

Now, it would be easy to dismiss these observations as

(1) Of no consequence, because when was the last time anybody used an incomplete elliptic integral, and

(2) Probably a misunderstanding on the part of someone who doesn't know that much about special functions. (Guilty.)

Let me argue that instead, this is a problem that could bite any user, and that my values for these functions are right (and Maple's wrong).

Consider the function `sqrt((x^2+1)/(x+2))`. This is a positive real function for $x > -2$. A plot suggests that `int(sqrt((x^2+1)/(x+2)), x=-1..5)` should be about 8.

Maple computes this integral using incomplete elliptic integrals, including the two discussed above.

```
> int(sqrt((x^2+1)/(x+2)), x=-1..5);


     1/2
2 182                  1/2            1/2             1/2             1/2
-------- - 2/6825 I 182    (70 + 35 I)    (-45 + 35 I)    (-45 - 35 I)
   3

                         1/2     1/2
             (70 + 35 I)     2 5             1/2              1/2
    EllipticF(--------------, ------ - 1/5 I 5   ) + 4/6825 182
                   5           5

               1/2             1/2             1/2
```

```
(70 + 35 I)    (-45 + 35 I)    (-45 - 35 I)

                      1/2     1/2                    1/2
          (70 + 35 I)     2 5             1/2    2 2              1/2
EllipticE(--------------, ------ - 1/5 I 5   ) - ------ + 2/75 I 2
               5           5                       3

          1/2           1/2           1/2
(10 + 5 I)    (15 + 5 I)    (15 - 5 I)

                     1/2     1/2
          (10 + 5 I)     2 5             1/2          1/2           1/2
EllipticF(-------------, ------ - 1/5 I 5   ) - 4/75 2    (10 + 5 I)
               5           5

                                              1/2     1/2
          1/2           1/2           (10 + 5 I)     2 5             1/2
(15 + 5 I)    (15 - 5 I)    EllipticE(-------------, ------ - 1/5 I 5   )
                                             5           5
```

Maple then asserts that the numerical value of the integral is

```
> evalf(%);

                        14.18222460 + 4.701625434 I
```

an obviously absurd answer. A correct answer can be obtained by replacing a limit of integration by a float:

```
> int(sqrt((x^2+1)/(x+2)), x=-1.0..5);

                              7.277500982
```

The first explanation that would occur to one here is that Maple's integral is wrong, but I think this is not so. One can differentiate Maple's value for the integral `int(sqrt((t^2+1)/(t+2)), t=-1..x)` by hand, and one seems to get `sqrt((x^2+1)/(x+2))` back, just as one ought. The real clincher, though, is that if one takes Maple's expression for the definite integral and plugs in the values I computed above for the EllipticE and EllipticF functions, the result is `7.277500984`, which agrees exactly with the result of computing the definite

integral numerically. (There are 4 elliptic integrals in Maple's definite integral; Maple seems to compute 2 of them right, and 2 of them wrong.)

A Maple worksheet showing a bit more detail on all this is at

`http://www.cs.earlham.edu/~timm/ellipticIntBug.mws`

I've checked the calculations here with Maple 6 under Windows and Linux, Maple 7 under Linux, and Maple V release 5 under MacOS. The exact wrong answers differ slightly between Maple V and Maple 7, but all these versions show the same problem.

I hope I'm just doing something wrong here, but if not, it's extremely alarming to me that one seems not to be able to rely on Maple correctly to compute the values of fundamental functions, and that bugs of this sort can persist for so many years.

7.58.2 Frederic PASCAL (14.9.01)

for computing the numerical value of the integral

`Int(sqrt((x^2+1)/(x+2)), x=-1..5)`

you first compute an algebraic expression of the integral then you ask for a numerical evaluation of the later expression.

You are increasing all the chances to get a wrong result and it is the case. Indeed it is a problem of rounding errors.

Let's take a more simple integral :

`Int(x^n/(x+a),x=0..1) with n=10 and a=10`

The integral should be less than 1.

The algebraic expression is the following :

```
>a:=10: n:=10: int(x^n/(x+a),x=0..1);

        10000000000*ln(11)-10000000000*ln(10)-300227066381/315
```

Now when Maple (MapleV5) computes

```
> a:=10; n:=10; evalf(int(x^n/(x+a),x=0..1));

                    2.0
```

Maple first computes the algebraic expression, then evaluates the floating expression with Digits:=10. During this step, it is doing a rounding error since you are computing a small value as the difference of large numbers.

```
>evalf(10000000000*ln(11));evalf(10000000000*ln(10));evalf(300227066381/315);
                                          11
                              .2397895273 10

                                          11
                              .2302585093 10

                                          9
                              .9531017980 10
```

The computation of ln is correct (well the 10 first digits, the next ones 0 are not) but you don't have enough digits to compute an exact difference.

Here the exact value is (now Maple doesn't compute an algebraic expression)

```
> evalf(Int(x^n/(x+a),x=0..1));

                               .008327965519
```

Remarks with `Digits:=25` you get an almost correct value

```
> Digits:=25:a:=10:n:=10:evalf(int(x^n/(x+a),x=0..1));

                            .0083279655188951
```

In your case, the problem is of the same order (i also don't know much about special functions) :

```
> r:=int(sqrt((x^2+1)/(x+2)), x=-1..5):
> Digits:=10: evalf(r);
> Digits:=25: evalf(r);
  evalf(Int(sqrt((x^2+1)/(x+2)), x=-1..5));
                                          -8
                      7.277500973 + .3 10   I
```

```
                                                        -22
          7.27750098195089964449116 8 - .19 10     I


                   7.27750098195089964449121 2
```

I could not reproduce the first two results (U. Klein)

Using numerics with Maple must be done with precaution.

7.58.3 James R. FitzSimons (24.9.01)

Maple has a bug.

Maple is giving the wrong answer using elliptic integrals.

```
> evalf(Int(sqrt((x^2+1)/(x+2)), x=-1..5));

                          7.277500982

> evalf(int(sqrt((x^2+1)/(x+2)), x=-1..5));

                  14.18222461 + 4.701625434 I
```

This is the numeric result using DERIVE.

```
7.2775009819508996444
```

7.59 bug in elliptic integrals in Maple V.5 (23.7.98)

7.59.1 Robert Michael Sinclair

In preparing a demonstration of the Arithmetic Geometric Mean for students, I came across the following behaviour, in which the second symbolic integral (or its numerical evaluation) are wrong. All numerical values should have been the same:

```
    |\^/|     Maple V Release 5 (WMI Campus Wide License)
._|\|   |/|_. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.

> f:=(a,b,theta)->1/sqrt(a^2*cos(theta)^2+b^2*sin(theta)^2);

                                              1
    f := (a, b, theta) -> ------------------------------------
                                 2           2    2           2
                          sqrt(a  cos(theta)  + b  sin(theta) )

> int(f(1,1/2,theta),theta=0..Pi/2);

                        EllipticK(1/2 sqrt(3))

> evalf(%);
                             2.156515648

> int(f(3/4,1/sqrt(2),theta),theta=0..Pi/2);        # This is the problem

  8/3 sqrt(3) EllipticK(1/9 sqrt(30))
         - 4/3 sqrt(3) EllipticF(1/2 sqrt(3), 1/9 sqrt(30))

> evalf(%);
                             5.546323855

> evalf(Int(f(1,1/2,theta),theta=0..Pi/2));
```

```
                                2.156515648

> evalf(Int(f(3/4,1/sqrt(2),theta),theta=0..Pi/2)); # It works here

                                2.156515648
```

It is corrected with Maple 6. (U. Klein)

7.59.2 Heike Koch-Beuttenmueller (24.7.98)

You can avoid this bug in the following way:

It works for both problems:

```
p3:=int(f(1,1/2,theta),theta=0..t);
                /
  p3 := - 1/2 |(1 + cos(t)) sqrt(4) sqrt(2)
                \

              3 cos(t) - I sqrt(3) + I sqrt(3) cos(t) + 1
        sqrt(-----------------------------------------)
                             1 + cos(t)

               -6 cos(t) - 2 I sqrt(3) + 2 I sqrt(3) cos(t) - 2
        sqrt(- ----------------------------------------------)
                                 1 + cos(t)

                       (-1 + cos(t)) sqrt(2 + 2 I sqrt(3))
        EllipticF(1/2 ----------------------------------,
                                   sin(t)

                                    \   /
        1/2 sqrt(-2 - 2 I sqrt(3)))|  /  (
                                    / /

                     2
        sqrt(3 cos(t)  + 1) sqrt(2 + 2 I sqrt(3)))

> p4:=int(f(3/4,1/sqrt(2),theta),theta=0..t);
```

```
                /
  p4 := - 4/3 |(1 + cos(t)) sqrt(2)
                \

                    cos(t) - 2 I sqrt(2) + 2 I sqrt(2) cos(t) + 8
              sqrt(-----------------------------------------------)
                                    1 + cos(t)

                     -2 cos(t) - 4 I sqrt(2) + 4 I sqrt(2) cos(t) - 16
              sqrt(- ------------------------------------------------)
                                       1 + cos(t)

                           (-1 + cos(t)) sqrt(-7 + 4 I sqrt(2))
              EllipticF(1/3 -----------------------------------,
                                        sin(t)

                                       \   /
              1/9 sqrt(17 + 56 I sqrt(2)))|  /  (
                                       / /

                         2
              sqrt(cos(t)  + 8) sqrt(-7 + 4 I sqrt(2)))

> f3:=unapply(p3,t);

                         /
  f3 := t -> - 1/2 |(1 + cos(t)) sqrt(4) sqrt(2)
                         \

                    3 cos(t) - I sqrt(3) + I sqrt(3) cos(t) + 1
              sqrt(-----------------------------------------)
                                  1 + cos(t)

                     -6 cos(t) - 2 I sqrt(3) + 2 I sqrt(3) cos(t) - 2
              sqrt(- ------------------------------------------------)
                                       1 + cos(t)

                           (-1 + cos(t)) sqrt(2 + 2 I sqrt(3))
```

```
       EllipticF(1/2 ----------------------------------,
                                    sin(t)

                                   \   /
       1/2 sqrt(-2 - 2 I sqrt(3)))|  /  (
                                   / /

                     2
       sqrt(3 cos(t)  + 1) sqrt(2 + 2 I sqrt(3)))

> f4:=unapply(p4,t);

                    /
  f4 := t -> - 4/3 |(1 + cos(t)) sqrt(2)
                    \

            cos(t) - 2 I sqrt(2) + 2 I sqrt(2) cos(t) + 8
       sqrt(-----------------------------------------------)
                             1 + cos(t)

              -2 cos(t) - 4 I sqrt(2) + 4 I sqrt(2) cos(t) - 16
       sqrt(- ------------------------------------------------)
                                1 + cos(t)

                    (-1 + cos(t)) sqrt(-7 + 4 I sqrt(2))
       EllipticF(1/3 -----------------------------------,
                                    sin(t)

                                   \   /
       1/9 sqrt(17 + 56 I sqrt(2)))|  /  (
                                   / /

                     2
       sqrt(cos(t)  + 8) sqrt(-7 + 4 I sqrt(2)))
```

This looks more complicated.

```
> f3(Pi/2);
```

```
  1/2 sqrt(4) sqrt(2) sqrt(1 - I sqrt(3)) EllipticF(

        1/2 sqrt(2 + 2 I sqrt(3)), 1/2 sqrt(-2 - 2 I sqrt(3)))

> evalf(%);

                                                  -9
                  2.156515647 - .3744778224 10    I

> f4(Pi/2);

  1/6 sqrt(2) sqrt(8 - 2 I sqrt(2)) sqrt(16 + 4 I sqrt(2)) EllipticF(

        1/3 sqrt(-7 + 4 I sqrt(2)), 1/9 sqrt(17 + 56 I sqrt(2)))

        sqrt(8)/sqrt(-7 + 4 I sqrt(2))

> evalf(%);

                                                  -9
                  2.156515649 - .1619270887 10    I

> evalf(%%,20);

                                                        -19
        2.1565156474996432357 - .20833956398182576545 10    I
```

You get a small imaginary value,which must be due to rounding errors, so you can neglect it.

7.59.3 Herb Brown (24.7.98)

It is interesting that this works correctly in release 4:

```
> f:=(a,b,theta)->1/sqrt(a^2*cos(theta)^2+b^2*sin(theta)^2);

                                          1
    f := (a, b, theta) -> ------------------------------------
                                2           2    2           2
```

```
                          sqrt(a  cos(theta)  + b  sin(theta) )

> int(f(3/4,1/sqrt(2),theta),theta=0..Pi/2);

                                    1/2  1/2       1/2
                 2 EllipticF(1/3 3    2   , 1/2 3   )

> evalf(%);

                                2.156515648
```

7.60 bug in elliptic integration in Maple 5.1 (2.5.01)

7.60.1 Nathan Sokalski

When using Maple V and Maple 6 I am recieving two different answers when using the int() function with a diff() inside of it. The entered lines are exactly the same. A book of mine gives the answer shown by Maple 6, but I have never recieved any different any between the 2 versions when using int() or diff() before.

Where is the problem, or is it just a special situation?

Maple V:

```
>f:=x->x^3;

                                    3
                          f := x -> x

>int(sqrt(1+(diff(f(x),x))^2),x=0..4); evalf(%);

  4/3 sqrt(2305) + 2/3 EllipticK(1/2 sqrt(2))

                        24
        - 1/3 EllipticF(---, 1/2 sqrt(2))
                        145

                           65.19438187
```

Maple 6:

```
>f:=x->x^3;

                                    3
                          f := x -> x

>int(sqrt(1+(diff(f(x),x))^2),x=0..4); evalf(%);

  4/3 sqrt(2305) + 2/9 sqrt(3) EllipticK(1/2 sqrt(2))
```

```
        - 1/9 sqrt(3) EllipticF(8/49 sqrt(3), 1/2 sqrt(2))

                                64.67196791
```

7.60.2 Robert Israel (3.5.01)

Maple 6 is correct here. It's a bug in previous releases, which has now been corrected. To explore it further:

```
> assume(P > 4, P < 6);
> JP:= int(sqrt(1+(diff(f(x),x))^2),x=0..P);
```

Release 5.1 has

```
  JP := 1/3 P sqrt(1 + 9 P )

        + 2/3 EllipticK(1/2 sqrt(2)) - 1/3

                       P
       EllipticF(6 --------, 1/2 sqrt(2))
                       2
                  9 P  + 1
```

But:

```
> simplify(diff(JP, P) - sqrt(1+diff(f(x),x))^2));

                   4                       4
       -sqrt(81 P  + 1) + 3 sqrt(1 + 9 P )
   2/3 ---------------------------------
                     4           4
         sqrt(1 + 9 P ) sqrt(81 P  + 1)
```

This should be 0.

Maple 6 has

```
                          4
  JP := 1/3 P sqrt(1 + 9 P )
```

```
        + 2/9 sqrt(3) EllipticK(1/2 sqrt(2))

                                  sqrt(3) P
        - 1/9 sqrt(3) EllipticF(2 ---------, 1/2 sqrt(2))
                                     2
                                  3 P  + 1

> simplify(diff(JP, P) - sqrt(1+diff(f(x),x))^2));

                0
```

7.61 bug in evalc by functions with indexed names in Maple V.4 to Maple 6 (29.7.97)

7.61.1 Eike Elbraechter

I believe, there is a bug in 'evalc'. The routine don't work, if the expression contains unevaluated function calls with indexed names.

```
> restart;
  kernelopts(version);
  interface (version);
  interface (patchlevel);
  libname;

  Maple V, Release 4, IBM RISC UNIX, Nov 20, 1996, RS2-54CD-328813-7
  Maple Worksheet Interface, Release 4, IBM RISC UNIX, Nov 25, 1996
                                  2
          /usr/local/maple4/update, /usr/local/maple4/lib

> `evalc/b` := proc()      # to test the user interface of evalc
      RETURN( u(t)+I*v(t) );
  end:

> for w in [ I*( a      +I*b       )
           , I*( a(t)   +I*b(t)    )
           , I*( a[1]   +I*b[1]    )
           , I*( a[1](t)+I*b[1](t) )
           ]
  do
    'evalc'(w) = evalc( w);
  od;

evalc(I*(a+I*b))              = I*a-b
evalc(I*(a(t)+I*b(t)))        = -u(t)+I*(a(t)-v(t))
evalc(I*(a[1]+I*b[1]))        = I*a[1]-b[1]
evalc(I*(a[1](t)+I*b[1](t))) = I*(a[1](t)+I*b[1](t))
```

The last expression with an indexed name of an unevaluated function call is not ex-

panded.

If the source of the routine ‘evalc/evalc‘ near the line 13 (get with list of the debugger) is changed from

```
     elif type(z,'function') then
..... to

    elif type(z, 'function') then
        g := op(0, z);
        if type(g, name ) then
            f := `if`( type(g, 'indexed'), op(0,g), g );
            f := `evalc/`.f;
            if not type(f, 'procedure') then traperror(readlib(f)) fi;
            if type(f, 'procedure') then w := (`evalc/`.g)(op(z))
            else
          ......
```

than the result is full evaluated:

```
> for w in [ I*( a      +I*b       )
           , I*( a(t)   +I*b(t)    )
           , I*( a[1]   +I*b[1]    )
           , I*( a[1](t)+I*b[1](t) )
           ]
  do
    'evalc'(w) = evalc( w);
  od;

evalc(I*(a+I*b))              = I*a-b
evalc(I*(a(t)+I*b(t)))        = -u(t)+I*(a(t)-v(t))
evalc(I*(a[1]+I*b[1]))        = I*a[1]-b[1]
evalc(I*(a[1](t)+I*b[1](t))) = -u(t)+I*(a[1](t)-v(t))
```

but I am not sure, that this is a correct correction.

It is corrected with Maple 7. (U. Klein)

7.62 BUG in evalc in Maple V (16.12.97)

7.62.1 Rafael Gallego

I have noted a bug in 'evalc':

```
> restart:
> evalc(Re(A)+I*Im(A));

                                A

> assume(A,complex);
> evalc(Re(A)+I*Im(A));

                            Re(A) + I
```

It is corrected with Maple 6. (U. Klein)

7.63 Bug in evalc in Maple V.4 (21.2.97)

7.63.1 luca ciotti

I think I found a bug in MapleV.4(00f).

evaluating

```
a:=int(exp(-x)/(1+x^2),x=0..infinity);
```

I get

```
a:=(I/2) (exp(I) Ei(1,I) - exp(-I) Ei(1,-I))
```

with

```
evalf(a) = 0.6214496240....
```

On the contrary,

```
evalf(evalc(a))= -1.0759... + I 2.643559
```

which is obviously wrong.

I suspect the bug is in the evalc() command.

In fact,

MapleV.3

```
evalc(Ei(1,I))    = - Ci(1) + I (Si(1) - 1/2 Pi)

evalc(Ei(1,-I))   = - Ci(1) - I (Si(1) - 1/2 Pi)
```

MapleV.4

```
evalc(Ei(1,I))    = - Ci(1) + I (Si(1) - 1/2 Pi)

evalc(Ei(1,-I))   = - Ci(1) - I (Si(1) + 3/2 Pi)
                                       ^^^^^^^^
```

The bug is removed with Maple V Release 5. (U. Klein)

7.63.2 Sjoerd W. Rienstra

Continuing on this theme of Lucca Ciotti (sorry to say it, but Maple is a goldmine if it comes down to errors in branch cuts of complex functions):

The integral of a real function like: `log(sin(x))` is of course determined up to a constant. If the integral happens to be explicitly available, this constant may be appear to be complex, and I can understand that if one of the occurring functions is multivalued (and has branches, branch cuts, etc) this constant may jump from one value to another along the `x-interval`. (This is why it is always tricky to accept a result from `int(f(x),x=a..b)` straightaway.)

However, the imaginary part of the integral of `log(sin(x))` is not piecewise constant on `0..Pi`!

Between `0 and Pi/2` it is constant, but it decays linearly between `Pi/2 and Pi`. (Then along `Pi..2 Pi` it also grows linearly, but this is how it should, because `sin(x)<0, and log(sin(x))` an imaginary constant.)

So there is something more fishy going on ....

7.64 bug in export to html in Maple 8 (10.6.02)

7.64.1 Carl Eberhart

When I export a worksheet to html in Maple 8, it opens all the sections before the export is made. Is there is fix to this? I am using Windows 2000.

7.64.2 Thomas Schramm (12.6.02)

This is a known bug reported during the beta testing. I hope there will be a patch very soon!

7.64.3 HARALD PLEYM (12.6.02)

I have reported the problem you mention. But for the moment I don't know about any fix.

There is also another "serious" bug (also reported). If you hide input before export is made, the input is still visible in the exported html worksheet.

7.65 bug in factor with right mouse button, Maple6 (17.8.00)

7.65.1 Metha Kamminga

After several times trying out, we must conclude that there is a bug in factor using the right mouse button. I am using Maple6 on Windows 98. Cklicking in the output and using expand an factor there will be the following session.

```
> restart;(x^3+3*y^2)*(5*x^2-4*y^3);

                      3      2      2      3
                     (x  + 3 y ) (5 x  - 4 y )

> 5*x^5-4*x^3*y^3+15*y^2*x^2-12*y^5;

                    5      3  3       2  2       5
                  5 x  - 4 x  y  + 15 y  x  - 12 y

> x^5-(x^3+3*y^2)*(5*x^2-4*y^3);

                     5     3      2      2      3
                    x  - (x  + 3 y ) (5 x  - 4 y )
```

The commands factor and expand will work without problems but with right mouse button, it is wrong in most of the situations with factor.

7.66 Bug in geom3d-package in Maple V.5 (15.3.99)

7.66.1 Arnold Zitterbart

Bug in the geom3d-package ! intersection(s,p1,p2) fails, if s is parallet to `x1_x2_plane`

```
> restart;
> with(geom3d):
> point(A,[4,9,3]):A=coordinates(A);

                              A = [4, 9, 3]

> point(B,[-2,5,3]):B=coordinates(B);

                              B = [-2, 5, 3]

> line(g,[A,B]);
                                       g
```

g is obviously parallel to the `x1_x2_plane`

```
p1:= any plane containing g:

> point(C1,[2,6,7]):plane(p1,[A,B,C1]);

                                       p1
p2:= any other plane containing g too:

> point(C2,[-3,6,9]):plane(p2,[A,B,C2]);

                                       p2

> intersection(s,p1,p2);
                                       s
```

but s is not the intersection-line:

```
> F:=FixedPoint(s):'F'=coordinates(F);
```

```
                          F = [0, 19/3, -3]

> IsOnObject(F,g);

                                  false

> IsOnObject(F,p1);

                                  false

> IsOnObject(F,p2);

                                  false
```

The sign of the x3-coord has to be changed:

```
> point(F_new,[xcoord(F),ycoord(F),-zcoord(F)]);

                                  F_new

> IsOnObject(F_new,g);

                                   true

> IsOnObject(F_new,p1);

                                   true

> IsOnObject(F_new,p2);

                                   true
```

therefore: a new intersection-procedure has to be defined:

```
> intersection_pp:=proc(g::symbol,p1::symbol,p2::symbol)
>    local P;
>    intersection(g,p1,p2);
>
```

```
point(P,[xcoord(FixedPoint(g)),ycoord(FixedPoint(g)),-zcoord(FixedPoint(g))]):

>    line(g,[P,ParallelVector(g)]);
> end;
```

Are there any other solutions to this problem ?

It is corrected with Maple 6. (U. Klein)

7.67 Bug in geometry[ellipse] in Maple V.4 (9.2.98)

7.67.1 Sergei Zuyev

Here is geometry-bug.mws file consisting just of a few lines and visualizing the following bug in `geometry[ellipse]` procedure: ellipse is computed incorrectly if in its definition one uses floats.

Maple Release 4 Version 4.00f of 2 Dec. 1996

`geometry[ellipse]` bug report

```
> restart:
> with(geometry):
```

Define two ellipses given by the same equation but written in integers and floats:

```
> _EnvHorizontalName := 'x': _EnvVerticalName := 'y':
> ellipse(e1,x^2/2+y^2=1  ):
> ellipse(e2,x^2/2+y^2=1.0):
```

Now observe the result (the erroneous one is in red):

```
> draw({e1(color=green),e2(color=red)},scaling=CONSTRAINED,axes=NORMAL);
```

Let's verify that it is NOT a bug of draw:

```
> ?geometry[ellipse];
> coordinates(center(e1));

                                [0, 0]

> coordinates(center(e2));

                    [-1.000000000, -1.000000000]
```

The help topic on geometry[ellipse] also contains misprints in the command syntax: "] " is lost:

```
ellipse(p, ['directrix'=dir,'focus'=fou, 'eccentricity'=ecc, n )
ellipse(p, ['foci'=foi,'MajorAxis'=lma,n )
ellipse(p, ['foci'=foi,'MinorAxis'=lmi,n )
ellipse(p, ['foci'=foi,'distance'=dis,n )
ellipse(p, ['MajorAxis'=ep1, 'MinorAxis'=ep2,n )
```

should definitely be

```
ellipse(p, ['directrix'=dir,'focus'=fou, 'eccentricity'=ecc], n )
ellipse(p, ['foci'=foi,'MajorAxis'=lma],n )
ellipse(p, ['foci'=foi,'MinorAxis'=lmi],n )
ellipse(p, ['foci'=foi,'distance'=dis],n )
ellipse(p, ['MajorAxis'=ep1, 'MinorAxis'=ep2],n )
```

7.67.2 Ha Quang Le (11.2.98)

The bug is fixed in release 5:

```
> with(geometry):


Define two ellipses given by the same equation but written in integers and floats:

\begin{MAPLEinline}
> _EnvHorizontalName := 'x': _EnvVerticalName := 'y':
> ellipse(e1,x^2/2+y^2=1  ):
> ellipse(e2,x^2/2+y^2=1.0):
> coordinates(center(e1));
                                    [0, 0]

> coordinates(center(e2));
                                    [0, 0]
```

7.68 Bug in Help on packages in Maple 7 with linux (28.2.02)

7.68.1 Ian Anderson

Click on Help.

Click on Topic Search.

Type packages.

Click on packages, programming.

This look like a great page for info on packages! but

NONE OF THE LINKS ON THIS PAGE WORK.

7.69 Bug in hypergeom in Maple V.4 (23.9.96)

7.69.1 Tony Blackett

Beware of a bug in Maple V R4 hypergeom function. The following examples illustrate the problem:

```
> readlib(hypergeom):
> hypergeom([2],[6],2*I*x);

                          2          3
       I (1 + 3/2 I x - x  - 1/3 I x  - exp(2 I x) (1 - 1/2 I x))
   -15 ------------------------------------------------------------
                                   5
                                  x
```

This first computation causes no problems, but when k is used in place of x, we get a serious error:

```
> hypergeom([2],[6],2*I*k);
Error, (in sum) 0^0 is undefined
 executing statement: convert([`$`(e,x)],`+`)
 locals defined as: r = r, sums = sums, k = k, naming = naming, unnaming =
unnaming, a = 0, ans = ans, b = 4, dab = 4, dxx = FAIL, i = i, tp = tp, xx =
k, np = np, oe = oe, ii = ii, mm = mm, rootslist = rootslist, dummy = dummy,
has_diffs = false, tmp = tmp, variable = variable
 sum called with arguments:
pochhammer(-3,k)*(2*(-1)^(1/2)*k)^k/k!/pochhammer(-4,k), k = 0 .. 3
```

The hypergeom function uses k as a summation index, however, this variable has not been declared as a local variable. Consequently, the index variable k is assumed to be global and gets confused with the formal parameter variable that evaluates to `2*(-1)^(1/2)*k`.

This one should be of interest to Technical Support and the Maple developers to fix.

The bug is removed with Maple V Release 5. (U. Klein)

7.70 bug in inequal in Maple 6 (5.2.01)

7.70.1 Christian und Sabine Koch

In working towards linear programming with my students I found maple 6.01 doing weird things:

```
with (plots);
inequal(x+y<2,x=-1..2, y=-1..2);
```

shows a triangle!!!!

```
inequal(x+y<2,x=-1..3, y=-1..2);
```

shows an angle with vertex at (-1,1) !!

```
inequal(x+y<2,x=-1..3, y=-1..3);
```

shows the region left and under the 2nd diagonal ???

It is corrected with Maple 7. (U. Klein)

7.70.2 Robert Israel (6.2.01)

In fact "inequal" always plots a strict inequality `(a*x + b*y < c or a*x + b*y > c)` as if `c = 0`. This affects Maple 6 and 6.01 but not Maple V.

This particular bug can be worked around by using "inequalities" from my Maple Advisor Database, `http://www.math.ubc.ca/~israel/advisor`, but it is only a symptom of a more serious problem.

Consider:

```
> x - 5 < 0;

      x < 5
```

Maple 5.1 would leave this as-is.

It seems that Maple now automatically simplifies a strict inequality with terms of type "numeric" on both sides so that the numeric term is only on the right. This means you'll run into errors if (as occurs in the code of "inequal") you produce an inequality of the

form `expr < 0` and then proceed on the assumption that all the terms of expr stay on the left side. I would not be surprised if this causes bugs in other places besides "inequal".

I have not seen any mention of this in the ?updates help pages.

7.70.3 Jay Treiman (7.2.01)

It appears that inequal also has problems with <=. In Maple 6.01

```
>constraints := {5*x[1]+5*x[2]=300,
>                  .6*x[1]+1.5*x[2]<=63,
>                 x[1]<=50, x[2]<=35,
>                 x[1]>=0,x[2]>=0};
>inequal(constraints,x[1]=-5..100,x[2]=-5..100,
>      optionsexcluded=(color=white),
>      optionsfeasible=(color=grey));
```

gave me the wrong region.

7.70.4 Symancyk, Daniel (9.2.01)

I tried adding optionsfeasible and optionsexcluded to see if they would help. There must be a bug in 6.01 when strict inequalities are involved. In R5.1 the following works fine.

```
with (plots);
inequal(x+y<2,x=-1..2,y=-1..2,optionsfeasible=(color=yellow),
        optionsexcluded=(color=white));
```

However, in 6.01 there is the error that you described. When the < is changed to <= in 6.01, the command works as it should.

7.71 bug in int and signum in Maple V.4 and Maple V.5 (4.9.97)

7.71.1 Andreas Jung

There is a bug in the integration of expressions involving the signum function (Release 4, Solaris):

```
> int(signum(x-s)*cos(s), s=0..Pi);
                                    0
```

The result is only correct for `x<=0` or `x>=Pi`. For `0<x<Pi`, it should be `2*sin(x)`. Converting all signums to Heaviside gives the correct result:

```
> int(convert(signum(x-s)*cos(s), Heaviside), s=0..Pi);

        (signum(-x + Pi) + 1) sin(x) + (signum(x) - 1) sin(x)
```

So always converting to Heaviside before integrating is a possible workaround. However, it's ugly, and the results given by int contain signum rather than Heaviside.

Is there a patch available for this integration bug?

It is corrected with Maple 6. (U. Klein)

7.71.2 Douglas B. Meade (8.9.97)

The piecewise package in Release 4 has had a number of problems. Some of the problems have been fixed in the patches, but there are still problems at patchlevel 2 (the current patch).

As examples, check out the examples in the online help for dsolve,piecewise. Compare the displayed output with the output obtained by executing the same command under the current patchlevel. I'd love to see how many of my ODE students would see that the solution in the online help CANNOT be the solution to this ODE.

The second example is more relevant to Andreas' posting. Conversion to/from abs, signum, piecewise, and Heaviside is not perfect.

```
> f := abs(x);
```

```
                              f := | x |

> df := diff( f, x );

                            df := abs(1, x)

> convert( df, piecewise );

                                    1

> convert( df, Heaviside );

                                    1
```

Note that `abs(1,x)` is another name for `signum(x)`. The problem here is that '`piecewise/abs`' effectively simplifies all two argument calls to abs to 1 (via the mapping `f -> Heaviside(-f) + Heaviside(f)`). The fix is obvious: replace this mapping with `f -> -1 + 2*Heaviside(f)`. This bug (together with the fix) were reported to Maple's Tech Support in November 1996.

While this does not fix Andreas' problem, I expect it sheds some light on a probably source of the problem.

7.71.3 Preben Alsholm (15.9.97)

I tried Douglas Meade's examples of problems with the piecewise package. If you differentiate one more time than he did, then you find

```
f:=abs(x):

df2:=diff(f,x,x);
                                  df2:=signum(1,x)

convert(df2,piecewise); # OK

              PIECEWISE([undefined, x = 0],[0,otherwise])

convert(df2,Heaviside); # NOT OK
                                        1
```

How is that explained?

7.71.4 Douglas B. Meade (21.9.97)

A quick glance at `‘Heaviside/signum‘` shows that it is written to expect only one argument `( f -> -1 + 2*Heaviside(f) )`. Thus, (essentially) all calls to `‘Heaviside/signum‘` with two arguments return the value 1.

This oversight should account for the behavior you report.

7.72 bug in int if integrand is a derivative (13.12.01)

7.72.1 E. Elbraechter

In the following special case the integral of a derivative is wrong.

```
> restart;
> interface(version,patchlevel);


Maple Worksheet Interface, Maple 7.00, IBM INTEL LINUX, May 28 2   001Build ID 96223,

> D[1](f)(xi,g(xi));
> Intat(%, xi=x): % = value(%);

                          D[1](f)(xi, g(xi))

                  x
                 /
                |
                |   D[1](f)(xi, g(xi)) dxi = f(x, g(x))
                |
               /
```

In case of a multi variate function in `'int/D'` it is not regarded, that more than one parameter could depend on the integration variable.

The same result with Maple 8. (U. Klein)

7.73 Bug in int in Maple 6 (28.6.00)

7.73.1 Martin Hirschmanner

There seems to be a bug in Maple 6. When I do the following computation I get different results for the numerical Intergration. In the first case I took as limits `q=0..2*Pi` and in the second `q=0..2*evalf(Pi)`. In the first case the result is wrong. In Maple V both computations give the same result.

Is this a bug ?

```
> restart:
> with(linalg):
> h:=matrix(9,1,[a1*cos(q),a1*sin(q),q,r*cos(q)+a2*sqrt(1-lambda^2*(sin(q))^2),
                 (r-lambda*a2)*sin(q),-arcsin(lambda*sin(q)),
                 r*cos(q)+l*sqrt(1-lambda^2*(sin(q))^2),0,0]);
> hst:=map(diff,h,q);
> M:=diag(m1,m1,I1,m2,m2,I2,m3,m3,I3);
> Theta:=evalm(transpose(hst)&* M &* hst);
> l:=0.25;
> r:=0.03;
> lambda:=r/l;
> a1:=21.877e-3;
> m1:=50.32;
> I1:=1.0267;
> a2:=0.034345;
> m2:=16.645;
> I2:=0.23251;
> m3:=44.55;

> Theta_m:=evalf(int(1/2/Pi*Theta[1,1],q=0..2*Pi));

                        Theta_m := 1.058110253

> Theta_m:=evalf(int(1/2/evalf(Pi)*Theta[1,1],q=0..2*evalf(Pi)));

                        Theta_m := 1.085648004
```

7.73.2 Robert Israel (29.6.00)

es, this is a bug. But it's not numerical integration, it's symbolic integration. If you want numerical integration, use `evalf(Int(...))` not `evalf(int(...))`.

Here is this bug in a simpler form:

```
> h:= (1-cos(t)/sqrt(1-a*sin(t)^2))^2;
> int(expand(h),t) - int(h,t);

                          t
```

The integral of the expanded form of h would be correct (at least when a is small and positive). Somehow it's as if `int(h,t)` expands h but forgets about the 1. In your example, if you use `expand(Theta[1,1])` instead of `Theta[1,1]` you should get the correct answer `1.085648004` both times.

7.73.3 Helmut Kahovec (18.7.00)

All I found out is that if the (trivial) changes described below are made in `` `int/indefinite`() `` and `` `int/definite`() `` then the example input will give the correct results.

Unfortunately, that statement is not correct, it is a bug, really. I am sorry and congratulate Robert Israel!

Well, I did not at all expect the symbolic integration to be wrong. Note that releases 4 and 5 of Maple V cannot evaluate that integral symbolically and thus numerically integrate definite integrals like

```
> lambda:=1/2:
> j:=(1+lambda^2*cos(q)/(sqrt(1-lambda^2*sin(q)^2)))^2;
> evalf(int(j,q=0..2*Pi));
```

maple6, however, can indeed integrate

```
> int(j,q);
```

and

```
> int(j,q=0..2*Pi);
```

symbolically but, as Robert has shown, will give the wrong result. `int/indefinite`() and `int/definite`() should expand their first argument in this case. I cannot pinpoint the bug more closely than that. If we make the following trivial changes to `int/indefinite`() and `int/definite`()

```
`int/indefinite` := proc(FF, zz)
local ff, opts, answer;
global StandardFunctions;
   1   ff := expand(FF);
   2   if nargs = 2 then
   ...
  19   if answer = FAIL then
  20     answer := ('int')(ff,zz)
       end if;
  21   answer
end proc

`int/definite` := proc(FF, zz, aa, bb)
local ff, answer, opts;
   1   ff := expand(FF);
   2   if nargs = 5 then
   ...
  33   if answer = FAIL then
  34     answer := ('int')(
           ff,zz = aa .. bb,op(subs(('formula') = NULL,opts))
         )
       end if;
  35   RETURN(answer)
end proc
```

then the following integrals will give the correct results:

```
> int(j,q);

  q + arcsin(1/2 sin(q))

        - 1/8 sqrt(3) arctan(1/3 (2 tan(1/2 q) - 1) sqrt(3))

        - 1/8 sqrt(3) arctan(1/3 (2 tan(1/2 q) + 1) sqrt(3))
```

```
        + 1/2 arctan(tan(1/2 q))

> int(j,q=0..2*Pi);

                        5/2 Pi - 1/4 sqrt(3) Pi
```

Note, too, that maple6 integrates

```
> int(j,q=0..2*evalf(Pi));
```

numerically right from the beginning, cf. lines `1-4` of the procedure `int()`.

7.73.4 Willard, Daniel Dr (21.7.00)

Please note that V 5.1 will integrate it symbolically with the answer given in terms of elliptic integrals.

7.74 bug in int Maple V.51 to Maple 8 (24.9.01)

7.74.1 Guy Gendron

It sounds like int is having problem with integrating the following functions.

```
> z_modele_1:=x*cos(x)^3;
> z_modele_2:=sin(x)+x*cos(x)-sin(x/2);
> evalf(int(abs(z_modele_2-z_modele_1),x=0..5));
> evalf(Int(abs(z_modele_2-z_modele_1),x=0..5));

                                          3
                      z_modele_1 := x cos(x)

            z_modele_2 := sin(x) + x cos(x) - sin(1/2 x)

                           2.041317817

                           5.959032441
```

The correct answer is `5.95...` .

These differences have been noticed both in Maple 6 `(build=16401)` and Maple 7 `(build=96223)` on Wintel.

7.74.2 Adri van der Meer (30.10.01)

This is a bug in int indeed.

Maple 6:

```
> K := int(abs(z_modele_2-z_modele_1),x=0..X):
> eval(K, X=0 );

                                  -11/9
```

If you (only) need a numerical approximation of an integral, it is always better to use the inert form Int.

Because:

`evalf(Int...)` makes a numerical approximation of the integral;

`evalf(int...)` first integrates symbolically and calculates a numerical approximation of the result.

7.75 bug in int() routine in Maple V.4 and Maple V.5 (1.3.97)

7.75.1 luca ciotti

I think I found another bug in MapleV.4 (00f, unix version)

```
a:= int(x/(sin(x)^3+cos(x)^3),x=0..Pi/2);
```

Maple is apparently able to do a symbolically using dilogaritms.

But,

```
evalf(a)= -.16989 -1.3791 I
```

which is wrong, being the integrand strictly positive.

Moreover, a direct numerical computation gives

```
a:=   1.4751.....
```

In Maple V Release 5 the first calculation results in .3784842693 + 2.758230170 I, while an evalf(Int(...)); leads to a := 1.4751....

It is corrected with Maple 6. (U. Klein)

7.75.2 Brian Blank (15.3.97)

The antiderivative found by MAPLE has some singularities; one is in the interval of integration. The rest of the worksheet attached below seems to indicate that the antiderivative is correct in [0,Pi/4).

```
> A := int( x/(sin(x)^3+cos(x)^3),x ):

> readlib(discont):  evalf( discont(A, x ) );
                                  -9
{-2.356194490 + .2638180391 10   I, 2.356194490 - .6584789486 I,

    -.7853981634 + .6584789479 I, .7853981634 + .6584789479 I,

                                                              -9
```

```
    2.356194490 + .6584789479 I, -.7853981634 + .2638180391 10    I,

                                                                  -9
    .7853981634 - .6584789486 I, .7853981634 + .2638180391 10    I,

    -.7853981634 - .6584789486 I, -2.356194490 + .6584789479 I,

                                                                   -9
    -2.356194490 - .6584789486 I, 2.356194490 + .2638180391 10    I}


> B := diff(A,x):  C := abs(B - x/(sin(x)^3+cos(x)^3) ):
> for  i  from  1  to 100  do
> if
>    evalf(subs(x=i*Pi/400,C)) > 10.^(-7)
>    then  print( [ i*Pi/400, evalf(subs(x=i*Pi/400,C))])
> fi
> od;

                      49                     -6
                     [--- Pi, .6861817978 10   ]
                      200


                      99                     -5
                     [--- Pi, .1288606775 10   ]
                      400


                                             10
                     [1/4 Pi, .1447924238 10   ]
```

Note the positive exponent of the last evaluation.

```
> subs(x=97*Pi/400,A) - subs(x=0,A): evalf(% );

                                        -8
                      .3687071756 + .22 10    I
```

```
> (evalf@Int)( x/(sin(x)^3+cos(x)^3),x = 0 .. 97*Pi/400 );

                                .3687071760
```

7.75.3 Christian TANGUY (26.3.97)

I think this is yet another example of trouble one can get with some integrals calculated by Maple V. As already noticed by other MUGers, the pedestrian procedure seems to be the following:

- use Int(...) instead of int(...), and THEN use evalf. Here, it gives

```
                1.475106979
```

- if more than the numerical value is needed, and when Maple V suggests

an obviously wrong result, all hope is not lost. A solution DOES exists, that can be obtained (1) with paper and pencil (2) by tweaking the integral.

In the above case, one can play with the integrand and replace the x factor in the numerator by `Pi/4`, because of the symmetry of the denominator. Then, one can express the denominator with respect to `t=x-Pi/4`. Surprise (?), the expression simplifies greatly as

```
Pi/sqrt(2)*int(1/(1-u^2)/(1+2*u^2),u=0..sqrt(1/2))
```

Maple V 3.0 (and 4.00b, U. Klein) for Windows can take it from there and gives the correct result, even though in a not so simple form (in my opinion). Anyway, the result is:

```
> evalf(Pi^2/12+Pi*sqrt(2)/6*ln(sqrt(2)+1));

                1.475106979
```

7.76 BUG in int-sum-dirac in Maple V.4 (15.11.96)

7.76.1 Renato Portugal

It seems to have a bug in Maple integrator with Dirac and probably with Heaviside functions.

```
> s1 := Sum(Dirac(x-i),i=0..5);

                                5
                               -----
                                \
                      s1 :=      )   Dirac(x - i)
                                /
                               -----
                               i = 0

> int(s1,x);

                /-----  \   /-----  \                    /-----  \
                | \     |   | \     |                    | \     |
Heaviside(x - i) |  )   1| + |  )   0| x - Heaviside(x - i) |  )   0|
                | /     |   | /     |                    | /     |
                \-----  /   \-----  /                    \-----  /
```

The correct result should be:

```
> sum(int(Dirac(x-i),x),i=0..5);

Heaviside(x) + Heaviside(x - 1) + Heaviside(x - 2) + Heaviside(x - 3)

     + Heaviside(x - 4) + Heaviside(x - 5)
```

An active unevaluated sum cause an error in Maple integrator:

```
> s2 := sum(Dirac(x-i),i=0..N);

                                N
```

```
                                 -----
                                  \
                          s2 :=    )   Dirac(x - i)
                                  /
                                 -----
                                 i = 0

> eval(int(s2,x));
Error, (in int) expecting two arguments
```

The bug seems to be in function ‘piecewise/extsimp‘.

The bug is removed with Maple V Release 5. (U. Klein)

7.76.2 David Holmgren

With regard to Dr. Portugal's problem concerning Dirac functions, first of all I cannot reproduce the problem (using R3 on either a PC or Sun workstation). (You will get the result of Dr. Portugal using Release 4, U. Klein). Also, the problem is very simply fixed if one defines s1, s2 as functions. Here's some output:

The original equation...

```
> s1:=Sum(Dirac(x-i),i=0..5);

                                   5
                                 -----
                                  \
                          s1 :=    )   Dirac(x - i)
                                  /
                                 -----
                                 i = 0
```

Try integrating it..

```
> int(s1,x);

                                   5
                                / -----
                               |   \
```

```
                              |    )   Dirac(x - i) dx
                              |   /
                             /   -----
                                 i = 0
```

Now do it as a function...

```
> readlib(unassign):
> unassign('s1');
> s1:=x->sum(Dirac(x-i),i=0..5);

                                      5
                                    -----
                                     \
                          s1 := x ->  )   Dirac(x - i)
                                     /
                                    -----
                                    i = 0

> s1(0);

                                   Dirac(0)

> s1(1);

                                   Dirac(0)
```

Seems OK. Now integrate...

```
> int(s1(x),x);

    Heaviside(x) + Heaviside(x - 1) + Heaviside(x - 2) + Heaviside(x - 3)

         + Heaviside(x - 4) + Heaviside(x - 5)
```

...which is the required answer. (The same result with Release 4, U. Klein) Do the same with s2...

```
> s2:=x->sum(Dirac(x-i),i=0..N);
```

```
                                    N
                                  -----
                                   \
                     s2 := x ->     )   Dirac(x - i)
                                   /
                                  -----
                                  i = 0

> s2(0);

                                    N
                                  -----
                                   \
                                    )   Dirac(- i)
                                   /
                                  -----
                                  i = 0

> s2(1);

                                    N
                                  -----
                                   \
                                    )   Dirac(1 - i)
                                   /
                                  -----
                                  i = 0

> int(s2(x),x);

                                    N
                                / -----
                               |   \
                               |    )   Dirac(x - i) dx
                               |   /
                              /   -----
                                  i = 0
```

(With Release 4 you will get the error message: Error, (in int) expecting two arguments, U. Klein)

```
> eval(%);
```

$$\sum_{i=0}^{N} \mathrm{Dirac}(x - i)\, dx$$

the problem is that N is not defined in any way...

```
> N:=5;
```

$$N := 5$$

```
> int(s2(x),x);
```

$$\mathrm{Heaviside}(x) + \mathrm{Heaviside}(x - 1) + \mathrm{Heaviside}(x - 2) + \mathrm{Heaviside}(x - 3) + \mathrm{Heaviside}(x - 4) + \mathrm{Heaviside}(x - 5)$$

```
> N:=10;
```

$$N := 10$$

```
> int(s2(x),x);
```

$$\mathrm{Heaviside}(x) + \mathrm{Heaviside}(x - 1) + \mathrm{Heaviside}(x - 2) + \mathrm{Heaviside}(x - 3) + \mathrm{Heaviside}(x - 4) + \mathrm{Heaviside}(x - 5) + \mathrm{Heaviside}(x - 6) + \mathrm{Heaviside}(x - 7) + \mathrm{Heaviside}(x - 8) + \mathrm{Heaviside}(x - 9) + \mathrm{Heaviside}(x - 10)$$

Anyway, this doesn't appear to be a bug, at least with Maple R3. The problem lies in the definitions of s1, s2, and the use of the inert summation function.

7.77 bug in int/defDirac (5.5.96)

7.77.1 Gerhard Hejc

I found the following bug in the code of the procedure "int/defDirac":

```
...
if type(f,`*`) then
    for z in f do
        if type(z,function) and op(0,z) = Dirac then
            argument := op(1,z);
            if nops(z) = 1 and type(argument,linear(x)) then
                argument := expand(argument);
                c0 := -coeff(argument,x,0)/coeff(argument,x,1);
                s1 := signum(c0-a);
                s2 := signum(b-c0);
                F := subs(z = 1,f);
                if b = infinity and a = -infinity or
                    member(s1,{0,1}) and b = infinity or
                    member(s2,{0,1}) and a = -infinity or
                    member(s2,{0,1}) and member(s1,{0,1}) then
                    ans := traperror(eval(subs(x = c0,F)));
                    if lasterror = ans then
                        ans := traperror(limit(F,x = c0));
                        if lasterror = ans then RETURN('FAIL')
                        else RETURN(ans/abs(coeff(argument,x,1)))
                        fi
                    else RETURN(ans/abs(coeff(argument,x,1)))
                    fi
                elif s1 = -1 and b = infinity or
                    s2 = -1 and a = -infinity or s2 = -1 or s1 = -1
                     then
                    RETURN(0)
                else RETURN('FAIL')
                fi
            elif nops(z) = 2 and type(argument,integer) and
                1 <= argument and type(op(2,z),linear(x)) then
                argument := expand(op(2,z));
                c0 := -coeff(argument,x,0)/coeff(argument,x,1);
```

```
                        s1 := signum(c0-a);
                        s2 := signum(b-c0);
#
# old code:
# F := (-1)^op(1,z)*diff(subs(z = 1,f),x $ op(1,z));
#
# new code:
#
                        F := (-1/coeff(argument,x,1))^op(1,z)*
                                        diff(subs(z = 1,f),x $ op(1,z));
#
                       if b = infinity and a = -infinity or
                            member(s1,{0,1}) and b = infinity or
                            member(s2,{0,1}) and a = -infinity or
                            member(s2,{0,1}) and member(s1,{0,1}) then
                            ans := traperror(eval(subs(x = c0,F)));
                            if lasterror = ans then
                                ans := traperror(limit(F,x = c0));
                                if lasterror = ans then RETURN('FAIL')
                                else RETURN(ans/abs(coeff(argument,x,1)))
                                fi
                            else RETURN(ans/abs(coeff(argument,x,1)))
                            fi
                        elif s1 = -1 and b = infinity or
                            s2 = -1 and a = -infinity or s2 = -1 or s1 = -1
                             then
                            RETURN(0)
                        else RETURN('FAIL')
                        fi
                    else RETURN('FAIL')
                    fi
                fi
            od
        ...
```

This bug causes a lot of strange results when integrating over Dirac functions.

The code hasn't changed up to Maple 7. For you sent no example of the bug, I can't decide if the bug is still present. (U. Klein)

7.78 bug in integer parsing in Maple 6 (17.1.01)

7.78.1 Renato Portugal

What's going on in Maple 6?

```
> restart;

> 1/21474836480;
Error, division by zero

> 21474836480;

                                  0

> 214748364810;

                                  10

> 2147483648;

                             -2147483648

> kernelopts(version);

          Maple 6, IBM INTEL NT, Jan 31 2000 Build ID 16401
```

It is corrected with Maple 7. (U. Klein)

7.78.2 Robert Israel (18.1.01)

That's a strange bug! Apparently it exists in the code that recognizes numbers in input. You also have e.g.

```
> 21474836481.234;

    1.234
```

It seems only to affect numbers that start `214748364x` where x is 8 or 9. The fact that

2147483648 = 2^31 is a clue: perhaps there's some 32-bit field overflowing here.

This bug was not present in Release 5.1.

7.78.3 Luis Goddyn (18.1.01)

Yes indeed, something is very wrong, even in 6.01. I clearly has to do with the sign bits of the new internal representation of integers (a hybrid between base 2 and base 10). Question: When can integer calculations be trusted in version 6?

```
> 2^31;
                                  2147483648

> 2147483648;
                                  -2147483648

> 2147483648041364781964379 4;
            413647819643790)+'--.(--.(

> %+1;
            413647819643790)+'--.(--.)

> kernelopts(version);

   Maple 6.01, SUN SPARC SOLARIS, June 9 2000 Build ID 79514
```

7.78.4 Heike Koch-Beuttenmueller (19.1.01)

This is a terrible bug in Maple 6.0. It did not exist in MapleV R5.1 You can find it on other platforms as well. I tried it on Sun and on Dec alpha:

Sun 32- bit machine:

```
> restart:
```

Here it works correct!!!

```
> a1:=2^31;

                          a1 := 2147483648
```

```
> a3:=1/a1;

                              a3 := 1/2147483648
```

Here all is wrong and the result depends upon assigning the starting value to a variable or calculating directly!!!!!!!!!

```
> b1:=2147483648;

                              b1 := -2147483648

> c2:=21474836480;

                                   c2 := 0

> kernelopts(version);

      Maple 6.01, SUN SPARC SOLARIS, June 9 2000 Build ID 79514
```

On DEC this part works , but because it is a 64-bit version you find the same behaviour near 2^64!!!!!!!!!!!

```
> restart:
```

Here it works correct!!!

```
> a1:=2^63;

                         a1 := 9223372036854775808

> a3:=1/a1;

                        a3 := 1/9223372036854775808
```

Here all is wrong and the result depends upon assigning the starting value to a variable or calculating directly!!!!!!!!!

```
> b1:=9223372036854775808;
```

```
                    b1 := -9223372036854775808

> c2:=92233720368547758080;

                              c2 := 0

> b3:=1/b1;

                                  -1
                    b3 := -------------------
                          9223372036854775808

> kernelopts(version);

        Maple 6.01, DEC ALPHA UNIX, Jun 9 2000 Build ID 79514
```

I hope its only a question of the wrong conversion of written integers near `2^31 or 2^63` to internal form and not a problem of wrong calculations?

7.78.5 Carlos Ruiz de Velasco y Bellas (19.1.01)

I dont know, but I have more "evident" (and infinite?) bugs:

```
> s:=[seq(2^(31+i)*5^i,i=0..300)]:
> map(x->parse(convert(x,string)),s);

         [-2147483648, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

> kernelopts(version);

     Maple 6.01, APPLE_PPC_MAC, Sun, Jun 11, 2000 Build ID 79514
```

But I worry, what about the not evident bugs?

7.78.6 Stanley J Houghton (5.2.01)

Much as I agree with the feelings expressed, I was under the impression that the bug only occurred on *input* of such integers, i.e. something I type in, not something that appears in mid calculation. I am concerned with what is being said here.

I though that `2147483646+2` would not cause a fault but `2147483648` will cause an error (sign incorrect in this case).

It needs correcting but can someone confirm or otherwise that it is an input only problem, ie not a calculation result error too?

7.78.7 Eithne Murray

I would like to take this opportunity to address the concerns raised about the integer parsing bug. I am including more details from our developers on exactly when this bug does and does not occur, and why. This way, each user may judge for himself whether or not this issue will affect his work. A workaround is provided so that the problem situation can be avoided entirely.

Because of the exact nature of this bug, the number of users who will run into this problem is very small. However, we still recognize that it is a serious problem that must be corrected. As Sultan Saini stated in a previous posting, this problem has already been corrected in our development version. We are currently determining the best method for getting this correction to our users.

When the Bug Occurs

This bug occurs when Maple is parsing integers whose first 10 digits are either `(+/-)2147483648 or (+/-)2147483649` (on 32-bit machines) or when the first 19 digits are either `(+/-)9223372036854775808 or (+/-)9223372036854775809` (on 64-bit machines). Only integers with those exact prefixes are affected.

```
> 2147483648;

> -2147483648;

> 21474836480;

> 21474836498;
```

It will occur when the number is explicitly typed into the session, cut and paste into

the session, or read in from a file.

When the Bug Does Not Occur

This problem is caused by a bug in the parser. It will not occur when the number is generated as a result of a calculation.

```
> 2147483647 + 1;
                      2147483648

> % + 1000000000;
                      3147483648
```

It does not occur in any types of numbers other than integers.

```
> 2147483648.0;
```

It does not occur for any integers other than those starting with the two 10-digit prefixes above.

Workaround

Since calculated results are not affected, this problem can be avoided by replacing any number starting with one of the two 10-digit problem strings by an equivalent expression that does not use the prefix. For example, if your worksheet contains the input line

```
> f(21474836489);
```

replace it with

```
> f(214748364 * 10^2 + 89);
```

What Caused the Bug

When Maple reads an integer inputted by a user it reads one digit at a time, building up the result. For small integers this scan can be done using the standard hardware '`long`' format. For large integers Maple's internal INTPOS or INTNEG data structure must be used. Both scans start by using the hardware format and switch to Maple's format only when needed. The bug was in the check for when to switch scan models. It was checking if the current `integer * 10` would still fit in a long and then proceeding to continue with a hardware scan if so. It should have been check-

ing if the current `integer * 10 + next digit` would fit. Thus integers of the form `2^(WORDSIZE-1)... and 2^(WORDSIZE-1)+1...` written out explicitly were subject to error.

This bug was introduced in Maple 6 due to a significant change in the way Maple stores small vs. large integers. Unfortunately, despite thousands of regression tests and millions of lines of library code this bug was not noticed before shipping the product.

Please be assured that we are taking this issue seriously, and we will inform you once decisions regarding the availability of the correction are made.

7.79 bug in integrating abs(..), Maple 6 (21.8.00)

7.79.1 Olivier Klein

I obtain a bad answer when I type the following integral:

```
> restart; assume(T>0); int(2*Pi/T*abs(sin(2*Pi/T*t)),t=0..T);
                                    2
```

(it is wrong, the answer should be 4)

It is not clear with me where the mistake is?

It is corrected with Maple 7. (U. Klein)

7.80 bug in integrating abs(...) (26.3.96)

7.80.1 A. van der Meer

In release 4 signum() still causes trouble:

```
> Int(Int(abs(y-x^2),x=-1..1),y=0..2);

                          2    1
                         /    /
                        |    |          2
                        |    |   | y - x  | dx dy
                        |    |
                       /    /
                       0    -1

> value(%);

                                  2


> Int(Int(abs(y-x^2),y=0..2),x=-1..1);

                          1    2
                         /    /
                        |    |          2
                        |    |   | y - x  | dy dx
                        |    |
                       /    /
                       -1   0

> value(%);

                                 46
                                 --
                                 15
```

The last answer is correct. R3 gives no answer for the first integral. The best policy is to avoid `abs() or signum()` and always use `piecewise()`.

The same result up to Maple 6.

With Maple 7 and Maple 8 you get with value(%); evalf(%); 3.216988933 instead of 3.066666835, what you get with evalf(Int(Int(...)); (U. Klein)

7.81 bug in integration 1 in Maple V.4 and Maple V.5 (13.5.98)

7.81.1 Marko Horbatsch

Can anyone explain the following integration error in Maple (Mma 3.0 has no problem with these integrals)

It is a class of integrals over the entire real axis over a bell-shaped function, where the location of the hump is displaced (which should not affect the calculation of the area under the curve). When the hump is centered at $x = 0$, the calculation is fine, but when the hump is displaced along the real axis by an undetermined (real) amount, the integration answer is wrong (negative infinity).

Users beware, Developers fix this up please (happens in V4 and V5!)

```
restart; assume(v,real); assume(w>0);

int(1/(x^2+w^2)^2,x=-infinity..infinity);
1/2*Pi/w^3

int(1/((x-v)^2+w^2)^2,x=-infinity..infinity);
-infinity

int(1/((x-v)^2+2)^2,x=-infinity..infinity);
-infinity

int(1/((x+3)^2+w^2)^2,x=-infinity..infinity);
-infinity

int(1/((x+3)^2+4)^2,x=-infinity..infinity);
1/16*Pi
```

It is corrected with Maple 6. (U. Klein)

7.81.2 Willard, Daniel, Dr., DUSA-OR (28.5.98)

Try:

```
assume(w>0); limit(int(1/((x-v)^2+w^2)^2, x=-R..R), R=infinity);
```

7.82 bug in integration 2 in Maple V.4 and Maple V.5 (18.11.98)

7.82.1 Robert Michael Sinclair (PHj)

There is a bug in the definite integration algorithm which only appears when the limits of integration are floating point numbers. Expressions of the form "limit/X" appear uninvited in the result. I had thought that this problem had been fixed in the transition from V4 to V5, but it is still around. In the first example below, the limits of integration are rational numbers, so the problem does not appear. In the second, the same limits have been written as floating point numbers, and the result is suddenly useless.

```
    |\^/|     Maple V Release 5 (WMI Campus Wide License)
._|\|   |/|_. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.

> int( x/sin(1-x^2), x=1/10..9/10 );

                 19       19                 19
- 1/2 ln(1 - cot(---) sin(---)) + 1/2 ln(sin(---))
                 100      100                100

                     99       99                 99
    + 1/2 ln(1 - cot(---) sin(---)) - 1/2 ln(sin(---))
                     100      100                100

> int( x/sin(1-x^2), x=0.1..0.9 );
                                                                    2
.8671793960 + 1.570796327 I signum(csc(-.19 - 1.8 limit/X + 1. limit/X )

                                          2
    - 1. cot(-.19 - 1.8 limit/X + 1. limit/X )) - 1.570796327 I signum(

                                  2                                    2
   csc(-.99 + .2 limit/X + limit/X ) - 1. cot(-.99 + .2 limit/X + limit/X ))
```

It is corrected with Maple 6. (U. Klein)

7.82.2 David L. Johnson (19.11.98)

Yech. It does clear up, however, if you add abs() around the argument of sin():

```
> int( x/sin(abs(1-x^2)), x=0.1..0.9 );

                                  .8671793961
```

which agrees with the rational version.

7.82.3 Helmut Kahovec (21.11.98)

This bug occurs in Release 4 and Release 5 because in your example

- int calculates the antiderivative of the integrand and then calls int/DefInt.

- int/Defint calls limit to calculate the value of the antiderivative at the limits of integration.

- limit substitutes `L-‘limit/X‘` for the indeterminate in the antiderivative. L is either limit of integration.

- limit does a series expansion of the antiderivative w.r.t. `‘limit/X‘` and calls `series/leadterm` since L is a float and there isn't any float in the antiderivative.

- series/leadterm does not work correctly if the series expansion of the antiderivative doesn't have an explicit leading term. It always takes the first term of the series expansion.

Thus the real bug is in series/leadterm. Look at the following:

```
> restart;
> j:=x/sin(1-x^2);

                                    x
                        j := - ------------
                                       2
                               sin(-1 + x )

> Int(j,x): %=value(%);
```

```
      /
     |          x                                  2             2
     |  - ------------ dx = - 1/2 ln(csc(-1 + x ) - cot(-1 + x ))
     |              2
    /     sin(-1 + x )

> J:=rhs(%);

                                        2             2
             J := - 1/2 ln(csc(-1 + x ) - cot(-1 + x ))

> limit(J,x=0.1,right);

.3082524570 - 1.570796327 I csgn(I (

                                    2
    -csc(-.99 + .2 limit/X + limit/X )

                                       2
     + 1. cot(-.99 + .2 limit/X + limit/X )))

> limit(J,x=0.9,left);

1.175431853 + 1.570796327 I csgn(I (

                                     2
    csc(-.19 - 1.8 limit/X + 1. limit/X )

                                          2
     - 1. cot(-.19 - 1.8 limit/X + 1. limit/X )))
```

Compare with the following:

```
> limit(J,x=1/10,right);

                    99       99                 99
    - 1/2 ln(1 - cot(---) sin(---)) + 1/2 ln(sin(---)) - 1/2 I Pi
                    100      100                100
```

```
> limit(J,x=9/10,left);

                    19       19                  19
   - 1/2 ln(1 - cot(---) sin(---)) + 1/2 ln(sin(---)) - 1/2 I Pi
                    100      100                 100
```

Actually, the order of the series expansion is 6. We use 4 to save space:

```
> series(J,x,4);

/          1
|- 1/2 ln(------ - cot(1))
\        sin(1)

                                2             2      \
    + 1/2 I csgn(I (csc(-1 + x ) - cot(-1 + x ))) Pi| -
                                                     /

       cos(1)             2
       ------- - 1 - cot(1)
            2
       sin(1)                2      4
   1/2 --------------------- x  + O(x )
             1
          ------ - cot(1)
          sin(1)

> series(leadterm(J),x,4);

/          1
|- 1/2 ln(------ - cot(1))
\        sin(1)

                                 2             2      \
    - 1/2 I csgn(I (-csc(-1 + x ) + cot(-1 + x ))) Pi|
                                                      /
```

Note that if L is a float, then series/leadterm seems to be called if there also is a float in the integrand. But if you manually introduce a float in the antiderivative then

series/leadterm isn't called by limit and the antiderivative is calculated correctly at the limits of integration:

```
> subs(-1+x^2=-1.+x^2,J);

                                    2                2
                - 1/2 ln(csc(-1. + x ) - cot(-1. + x ))

> limit(%,x=0.1,right);

                      .3082524570 - 1.570796327 I

> limit(%%,x=0.9,left);

                      1.175431854 - 1.570796327 I
```

Now we may easily construct another example of this bug:

```
> -ln(sin(-1+x^2));

                                         2
                           -ln(sin(-1 + x ))

> series(%,x,6);

                                    2          cos(1)  2
(-ln(sin(1)) + I csgn(I sin(-1 + x )) Pi) + ------ x  +
                                           sin(1)

    /                 2\
    |          cos(1) |  4      6
    |1/2 + 1/2 -------| x  + O(x )
    |                 2|
    \          sin(1) /

> series(leadterm(%%),x,6);

                                                 2
              (-ln(sin(1)) + I csgn(I sin(-1 + x )) Pi)
```

```
> diff(%%%,x);

                                        2
                                   cos(-1 + x ) x
                              -2 --------------
                                          2
                                   sin(-1 + x )

> int(%,x);

                                         2
                              -ln(sin(-1 + x ))

> int(%%,x=0.1..0.9);

1.487659539

                                                                    2
     + 3.141592654 I csgn(I sin(-.19 - 1.8 limit/X + 1. limit/X ))

                                                  2
     - 1. I csgn(I sin(-.99 + .2 limit/X + limit/X )) Pi

> int(%%%,x=1/10..9/10);

                              19             99
                     -ln(sin(---)) + ln(sin(---))
                              100            100
```

7.83 bug in integration 3, Maple 5 to Maple 6 (7.12.98)

7.83.1 Douglas B. Meade

NOTE: This is, essentially, a bug report. I welcome help proposing a complete patch but expect that Maple's Technical Support will take responsibility for seeing that similar mistakes have not been made elsewhere in the integration system and for seeing that my suggestions do not cause additional problems.

I have discovered a bug deep within Maple's integration procedures. This problem was discovered during a conversation with Jon Johnson at last month's ICTCM.

Jon's question was: why does Maple (R5) fail to evaluate the following iterated integral

```
> f := sqrt( (cos(x)-cos(y))^2 + (sin(x)-sin(y))^2 );
> A := Int( Int( f, y=0..2*Pi ), x=0..2*Pi );
> value ( A );
```

The error message that is produced by the last command is:

```
Error, (in int/ellalg/trxstandard/4) int/ellalg/trxstandard/4 uses a
7th argument, L, which is missing
```

A little exploration (with tracelast, showstat, and debug) led me to discover that `int/ellalg/trxstandard` calls `int/ellalg/trxstandard/4` with only 6 arguments.

Arguments 7 and 8 (L and U) are required.

```
> showstat( `int/ellalg/trxstandard` );
```

The ninth argument is used at the end of the procedure

```
> showstat( `int/ellalg/trxstandard/4`, 1..20 );
```

My conjecture is that the default values for L and U are -infinity and infinity, respectively. I do not know what z represents so cannot determine a suitable default value. (I tried 0, but this was not successful.)

Here is the code that I used to

```
> `int/ellalg/trxstandard/4orig` := op( `int/ellalg/trxstandard/4` );

> `int/ellalg/trxstandard/4` := proc(lc, a, b, c, d, x, L, U, z)
>     local LL, UU, zz;
>     if nargs<7 then LL:=-infinity fi;
>     if nargs<8 then UU:= infinity fi;
> #    if nargs<9 then zz:= 0        fi;
>     `int/ellalg/trxstandard/4`(lc, a, b, c, d, x, LL, UU, z);
> end;
```

With this "fix" the evaluation of the integrals progresses a little further but halts with a message similar to the original message.

If someone knows what value to assign to z, I'll be glad to continue my testing. Otherwise, I hope Maple's Technical Support will be able to come up with a comprehensive solution to this problem.

With Maple 6 I got no error message, but also no result. It is corrected with Maple 7 (U. Klein)

7.83.2 Joe Riel (15.12.98)

Note that the call to ‘int/ellalg/trxstandard/4‘ in ‘int/ellalg/trxstandard‘ is

```
  `int/ellalg/trxstandard/4`(lc, op(srL), x, L, U, z)
```

This suggests that the problem is in op(srL), it should expand to 4 elements. Looking at the assignment to srL, it appears that the culprit is the procedure ‘int/ellalg/trxstandard/roots‘.

An obvious error in that procedure is the assignment to r after the if then else statement. It is

```
r := traperror(sort(f, `int/ellalg/trxstandard/compare`));
```

This is clearly wrong because f is an unassigned local variable. Sorting it makes no sense. Either a previous assignment to f is missing or this line should probably be

```
r := traperror(sort(rts, `int/ellalg/trxstandard/compare`));
```

Making this change returned an answer for the inner integral, 0, which is obviously wrong (and the same wrong answer that R4 returns).

So I cannot give you a solution, but I'm confident that the problem lies in `'int/ellalg/trxstandard/roo`

If you restrict x to the open range 0 to π, i.e.

```
assume(x, RealRange(Open(0),Open(Pi)):
```

then the `'.../compare'` routine used in `'.../roots'` is able to sort the solutions. However, because the expression has a double root at `sin(x)/(1+cos(x))`, only three values are in the list. The `'.../trxstandard'` procedure generates an error because it attempts to access a `[nonexistent]` fourth element in the list returned by `'.../roots'`. This is with the substitution of rts for f in `'.../roots'`.

When `'.../roots' [and possibly '.../trxstandard]` is eventually correctly patched, I suspect that if you are ever to evaluate the double integral you will have to split the outer integral into two pieces, `0..Pi and Pi..2*Pi`, and integrate these separately, using assumptions on x.

7.83.3 Graham P. McCauley (18.1.99)

On 07 Dec 1998 Douglas Meade submitted a report on what he thought was a bug in the R5 integration procedures. I have tried to follow this up, and suspect that the bug may really be in the routine `'solve'`.

Douglas found that

```
> f:=sqrt((cos(x)-cos(y))^2+(sin(x)-sin(y))^2);
> A:=Int(Int(f,x=0..2*Pi),y=0..2*Pi);
> value(A);
```

gave rise to the message

```
Error, (in int/ellalg/trxstandard/4) int/ellalg/trxstandard/4 uses a 7th
argument, L, which is missing
```

He used `'tracelast'` to locate the offending call

`#('int/ellalg/trxstandard',26): 'int/ellalg/trxstandard/4'(lc,op(srL),x,L,U,z)`

and tried to add extra arguments at the end of the list.

It seems to me that ‘op(srL)‘ should provide four values (the roots of a quartic polynomial in ‘x‘ whose coefficicents are trigonometric expressions in ‘y‘). Use of the debugger shows that ‘srL‘ in the present case is an unassigned variable ‘f‘.

This variable is in fact local to ‘int/ellalg/trxstandard/roots‘. In the present case it remains unassigned in a nest of conditional statements. These appear to be an attempt to deal with the fact that ‘solve‘ might fail to report some roots of a quartic ‘sr‘ in ‘x‘ whose coefficents are trigonometric expressions in ‘y‘. Adding a final clause

```
else f := [op(rts), simplify(eval(tcoeff(sr,x)/lc/convert(rts,`*`)))]
```

to the nest ensures that if only one root is missing (and none of them are zero!) it is added to the list ‘rts‘. [I have checked that it is indeed a slightly disguised repeat of ‘rts[3]‘]. The new list is assigned to ‘f‘ which is sorted in some way and RETURNed to become ‘srL‘ above. Now MapleV soldiers on happily to produce the value zero.

Two mysteries remain:

(1) why does ‘solve‘ miss the repeated root?

(2) why does the call of ‘int/ellalg/trxstandard‘ have a final argument ‘z‘ that is given the value ‘droot‘ that seems to suggest a double root was already suspected? [I haven’t found where this argument is used. It is the one that was troubling Douglas Meade in his proposed "fix".] A final word on the final result. The value zero is presumably not what the originator of the integral expected. We can see how it comes about if we tidy up the integrand and divide the square region of integration diagonally by writing

```
> g:=combine(simplify(f,trig),trig);

                g := (2-2*cos(x-y))^(1/2)

> A1:=Int(Int(g,x=0..y),y=0..2*Pi):
> A2:=Int(Int(g,x=y..2*Pi),y=0..2*Pi):
```

Both A1 and A2 give the value `4*Pi*4^(1/2)`. [This appears to come out without invoking ‘int/ellalg/trxstandard‘.] One naturally reads these values as `8*Pi` since ‘sqrt‘ of a real positive quantity is taken to be positive. However when we ask MapleV to integrate over the whole square at once, it effectively insists on an analytic interpretation of the integrand that makes the two triangles cancel!

7.84 Bug in integration 4, Maple V to Maple 7 (20.7.00)

7.84.1 Joao C A Barata

I believe I have found a bug in Maple's integration package (I use Maple V.5).

Computing

```
> int( 1/((A - cos(p + K))*(A - cos(p - K))), p=-Pi..Pi );
```

Maple returns zero, what is certainly wrong. Even being more specific about the parameters A, B and K produces errors. For example

```
> assume(A > 1);
> assume(B > 1);
> assume(K, real);

> int(1/((A - cos(p + K))*(A - cos(p - K))), p=-Pi..Pi);
```

also returns zero. Note that for `A > 1 and B > 1` the integrand is positive and, hence, the integral cannot be zero.

If you specify numerical values for A, B and K, as below

```
> evalf(int(1/((2 - cos(p + 3))*(2 - cos(p - 3))), p=-Pi..Pi));
```

Maple returns the correct (non zero) value, something like 2.40245 .

I would like to know if there is a way to avoid error like this. The integrals I need to compute are actually more complicated than this one, but also involve integrals of ratios of trigonometrical polynomials.

7.84.2 Robert Israel (21.7.00)

This is basically a branch cut problem. Unfortunately, this sort of thing is not unexpected. Maple does this integral by first computing an antiderivative, a rather complicated expression involving logarithms and arctans, and then uses the Fundamental Theorem of Calculus. It tries to identify discontinuities of the antiderivative, but apparently is unsuccessful (depending on the values of the parameters). Maple 6 may be trying harder, but seems to run into a bug:

```
Error, (in limit) invalid limiting point
```

I get the same error message with Maple 7. Including the assumptions I get a piecewise solution. Maple 8 produces a piecewise solution in both cases. (U. Klein)

Including the assumptions I get a set of solutions with Maple 6. (U. Klein)

7.84.3 Helmut Kahovec (24.7.00)

The current version of Maple, maple6, seems to get caught in an endless loop while computing your definite integral:

```
> restart;
> j:=1/((A-cos(p+K))*(A-cos(p-K)));

                                    1
                j := ---------------------------------
                     (A - cos(p + K)) (A - cos(p - K))

> int(j,p=-Pi..Pi);
Warning, computation interrupted
```

Note, however, that `evalf(int(...));` means that Maple tries to compute the definite integral symbolically before calculating a numerical approximation:

```
> int(1/((2 - cos(p + 3))*(2 - cos(p - 3))), p=-Pi..Pi);

                                      2                 2
  - 4/3 (4 sqrt(3) Pi - 3 %2 %4 cos(6)  - 3 %2 %4 sin(6)

                  ... lines skipped ...
```

```
                    2         2
  %1 := -4 + cos(6)  + sin(6)

                      2         2
  %2 := sqrt(4 - cos(6)  - sin(6) )

               %2 (2 cot(3/2) + cot(3/2) cos(6) + sin(6))
  %3 := arctan(-----------------------------------------)
                                 %1

               %2 (-2 cot(3/2) - cot(3/2) cos(6) - sin(6))
  %4 := arctan(-----------------------------------------)
                                 %1

> evalf(%);

                              2.402450983
```

If you want to avoid any symbolic integration you should use `evalf(Int(...));` .

7.85 bug in integration dependent on N, Maple 6 to Maple 8 (5.4.02)

7.85.1 Glenn Sowell

I have seen the following integration bug in Maple 6 & 7, but not in 5.1.

The bug seems to be N dependent, where N is the power of the denominator of the integrand.

```
> interface(version);
                      Maple Worksheet Interface, Maple 7.00,
                      APPLE_PPC_MAC, Tue, May 29, 2001 Build ID 96223
> assume(a>0);
> f := 1/(x^2+a^2)^N;
> # N=6 This does not work.
> N:=6:
> Df := diff(f,x,x);
> fDf := simplify( f * Df );
> Int1 := int( fDf, x=0..infinity );

                                    Int1 := 0
```

A surprising result!

Try expanding fDf before integrating:

```
> Int2 := int( expand(fDf), x=0..infinity );

                     468027 Pi
          Int2 := - ------------
                              25
                    1048576 a~
```

A different result?!

he same dichotomy occurs for N>=6. For N<=5, both results are the same.

It gets stranger...

Do the indefinite integral first

```
> Int3 := int( fDf, x);
```

Now apply the limit by hand:

```
> Int2 := limit( Int3, x=infinity ) - limit( Int3, x=0 );

                                  468027 Pi
                      Int2 := - ------------
                                        25
                                1048576 a~
```

Another surprising result!

Evidently, the definite integral algorithm is doing something wrong!

Why does it occur for `N>=6` and not for `N<=5`?

It is corrected with Maple 9 (U. Klein)

7.85.2 Helmut Kahovec (12.4.02)

Actually, the procedure '`int/definite/contour/residue`', which computes residues during definite integration, is flawed. Additionally, the procedure '`residue`' cannot compute a residue if the order of the pole is sufficiently high. Following is a Maple7 session on a PC running MS Windows NT 4.0.

```
> assume(a>0);
> num,den,alpha,m:=(13*_X^2-a^2)*ln(-_X),(_X^2+a^2)^14,I*a,14:

> residue(eval(num/den,_X=x),x=alpha);

                                 2    2
                          (13 x  - a ) ln(-x)
                    residue(------------------, x = I a)
                                  2    2 14
                                 (x  + a )
```

'residue' returns unevaluated. We now modify 'residue' as indicated:

```
> unprotect(residue);
```

```
> residue:=proc(f,a)
  local g,i,t,x,t1;
  options remember;
    if nargs<>2 or not type(a,equation) or not type(op(1,a),name) then
      error "invalid arguments"
    end if;
    x:=op(1,a);
    if has(op(2,a),x) then
      error "invalid point"
    elif type(op(2,a),infinity) then
      g:=-1/x^2*subs(x=1/x,f)
    else
      g:=normal(subs(x=x+op(2,a),f),expanded)
    end if;
#    for i from 0 to 5 do   <=== replace the number 5 by 6 ===
    for i from 0 to 6 do
      t:=traperror(series(g,x,i^2+2));
      if t=0 then return 0 end if;
      if t=lasterror or not type(t,series) then next end if;
      t1:=order(t);
      if t1=infinity or -1<t1 then return coeff(t,x,-1) end if
    end do;
    'residue(args)'
  end proc:

> protect(residue);

> residue(eval(num/den,_X=x),x=alpha): R11:=simplify(expand(%));

  R11 :=

                        669278610 I ln(a) - 2269536821 I + 334639305 Pi
       1/17993564160 ------------------------------------------------
                                          25
                                         a
```

This seems to be the correct value. During definite integration the procedure ‘`int/definite/contour/re` gets called with the appropriate arguments. However, since ‘`int/definite/contour/residue`‘

is flawed, it will return an incorrect residue if the order of the pole is high enough.

```
> `int/definite/contour/residue`(num,den,alpha,m);

                                  -79
                                -------- I
                                14057472
                                ----------
                                    25
                                   a
```

We fix the incorrect lines in ‘int/definite/contour/residue‘ as indicated:

```
> `int/definite/contour/residue`:=proc(n,d,a,m)
  local i,r,v1;
    if nargs=4 and m=1 then
      i:=n/diff(d,_X);
      i:=traperror(eval(subs(_X=a,i)));
      if i<>lasterror then return i end if
    end if;
    for i from 0 by 3 to 12 do
      r:=traperror(`int/definite/contour/seriesc`(n/d,_X=a,i));
      if r=FAIL then
        return FAIL
      elif r=lasterror or
           not type(r,series) or
           op(nops(r),r)<0 and op(nops(r)-1,r)=O(1)
      then
        next
      else
        break
      end if
    end do;
    if i<=12 then
      if hastype(n/d,'nonreal') or hastype(a,'nonreal') then
        return coeff(r,evalc(_X-a),-1)
      else
        return coeff(r,_X-a,-1)
      end if
```

```
    end if;
    if type(d,polynom(anything,_X)) and nargs=4 and 1<m then
      v1:=quo(d,(_X-a)^m,_X,'r');
      if r<>0 then return FAIL end if;
#
#     One has to differentiate n/v1 instead of differentiating n
#     alone and dividing by v1 afterwards!
#
#                                          ///
#                                          vvv
      i:=traperror(eval(subs(_X=a,diff(n/v1,`$`(_X,m-1))/(m-1)!)));
      if i<>lasterror then
        return i
      else
        return FAIL
      end if
    elif nargs=4 and 1<m then
#
#     One has to differentiate (_X-a)^m*(n/d) instead of (_X-a)^m
#     alone!
#                              ////
#                              vvvv
      i:=limit(diff((_X-a)^m*n/d,`$`(_X,m-1))/(m-1)!,_X=a);
      if type(i,function) and op(0,i)=limit or
         type(i,undefined) or
         type(i,range)
      then
        FAIL
      else
        i
      end if
    else
      FAIL
    end if
  end proc:
```

Now, ‘int/definite/contour/residue‘ returns the same result as ’residue’:

```
> `int/definite/contour/residue`(num,den,alpha,m):
```

```
> R12:=simplify(expand(%));

  R12 :=

                        669278610 I ln(a) - 2269536821 I + 334639305 Pi
       1/17993564160 ------------------------------------------------
                                           25
                                          a
```

The other pole of your integrand gives the following residue:

```
> num,den,alpha,m:=(13*_X^2-a^2)*ln(-_X),(_X^2+a^2)^14,-I*a,14:
> residue(eval(num/den,_X=x),x=alpha): R21:=simplify(expand(%));

  R21 := - 1/17993564160

        669278610 I ln(a) - 334639305 Pi - 2269536821 I
        ----------------------------------------------
                              25
                             a

> `int/definite/contour/residue`(num,den,alpha,m):
> R22:=simplify(expand(%));

  R22 := - 1/17993564160

        669278610 I ln(a) - 334639305 Pi - 2269536821 I
        ----------------------------------------------
                              25
                             a
```

Finally, Maple will correctly compute your definite integral, too:

```
> f:=1/(x^2+a^2)^N:
> N:=6:
> Df:=diff(f,x,x):
> fDf:=simplify(f*Df);

                                      2    2
```

```
                                        13 x  - a
                              fDf := 12 -----------
                                          2    2 14
                                        (x  + a )

> int(fDf,x=0..infinity): normal(%,expanded);

                                   468027  Pi
                                 - ------- ---
                                   1048576  25
                                          a
```

If we expand the integrand then Maple first computes the antiderivatives and takes the limits afterwards.

```
> int(expand(fDf),x=0..infinity): normal(%);

                                   468027  Pi
                                 - ------- ---
                                   1048576  25
                                          a
```

7.86 bug in integration exp(I*a*t) (31.8.98)

7.86.1 Al Rosethal

What's going on here?

```
>q:=int(exp(I*a*t)/(t^2+b^2),t=-infinity..infinity);

                                q := 0

> qr:=int(cos(a*t)/(t^2+b^2),t=-infinity..infinity);

                             signum(a) Pi sinh(b a)
                   qr := - ----------------------
                                      b
```

Why doesn't Maple return the second result for the first integral?

7.86.2 John Roumeliotis(3.9.98)

Even more curious (?)

```
> q:=int(cos(a*t)/(t^2+b^2),t=-infinity..infinity);

                           signum(a~) Pi sinh(b a~)
                  q := - ------------------------
                                      b

> simplify( subs(a=5,b=3,%) );

                           - 1/3 Pi sinh(15)

> q:=int(cos(5*t)/(t^2+3^2),t=-infinity..infinity);

              q := - 1/3 Pi sinh(15) + 1/3 Pi cosh(15)
```

7.86.3 HARALD PLEYM (3.9.98)

You can try this:

```
>q:=Int(exp(I*a*t)/(t^2+b^2),t=-infinity..infinity);
                            infinity
                           /
                          |          exp(I a t)
                   q :=   |          ---------- dt
                          |            2    2
                         /            t  + b
                            -infinity

> q:=map(evalc,q);

                        infinity
                       /
                      |          cos(t a)   I sin(t a)
              q :=    |          -------- + ---------- dt
                      |           2    2      2    2
                     /           t  + b      t  + b
                       -infinity

> value(%);

                    (-1 - I) Pi sinh(b a) signum(a)
                    ------------------------------
                                 b
```

7.87 bug in integration in Maple 6 and fix (16.5.01)

7.87.1 Josef Hekrdla

I am really horrified! If you integrate elementary function `1/(9x^2+6x+2)`, you obtain the wrong result `arctan(1+3x)` instead of the correct `1/3*arctan(1+3x)`! It is only for Maple6. Maple5 returns the correct result.

It is corrected with Maple 7. (U. Klein)

7.87.2 Carl DeVore(18.5.01)

This bug results from trying to program a great variety of simple cases rather than just using a handful of algorithms for the general cases. In this problem, the special case that is handled in procedure `‘int/ratpoly/ratpoly‘` is when the denominator is exactly one more than a perfect square of first-degree binomial. In other words,

```
> int(c/((a*x+b)^2+1), x)
```

will give the wrong answer. Change that 1 to anything else and you'll get the right answer. This case is so special – why do they even bother with it? It's hard to imagine that it is worth the bother to treat it separately.

The general case algorithms need to be there anyway. Does it really save that much time to sort out and handle the simple cases? When there are so many special cases, it makes it that much more likely that a trivial bug will go undetected for a long time. Does Maple have an official programming philosophy?

In this case, the bug is truly trivial – essentially a typo – and easy for the user to correct. The fact that someone did not notice this bug sooner is a sign that it is not worth the trouble to treat this case separately. I've traced it to lines 21 and 23 of procedure `‘int/ratpoly/ratpoly‘`. I show the fix after the code below.

```
> restart;
> showstat(`int/ratpoly/ratpoly`);

`int/ratpoly/ratpoly` := proc(f)
local ans, den, num, d, n, k, const, q, r, rest, g, const2, A, B, answer;
   1   if type(f,polynom(anything,_X)) then
```

```
   2     RETURN(`int/polynom`(args))
       end if;
   3   if type(f,`+`) then
   4     return map(procname,f)
       end if;
   5   num := 1;
   6   const := 1;
   7   if type(f,`*`) then
   8     rest, const := selectremove(has,f,_X);
   9     num, den := numer(rest), denom(rest)
       else
  10     const, num, den := 1, 1, denom(f)
       end if;
  11   den, n := op(`if`(den::`^`,den,[den, 1]));
  12   d := degree(den,_X);
  13   answer := FAIL;
  14   if d = 1 then
  15     const := const*num/coeff(den,_X,1);
  16     if n = 1 then
  17       answer := const*ln(den)
         else
  18       answer := -const/(n-1)/(den^(n-1))
         end if
       elif n = 1 and degree(num,_X) = d-1 and rem(num,diff(den,_X),_X,q) = 0 then
  19     answer := const*q*ln(den)
       elif n = 1 and d = 2+2*degree(num,_X) and type(den,polynom(polynom(rational),_X
            and psqrt(den-1) <> _NOSQRT and rem(diff(den-1,_X),num,_X) = 0 then
  20     r := psqrt(den-1);
  21     answer := const/content(r,_X)*arctan(r)
^^^^^^^^^^^^^ The bug is that they divide by the common factor (the "content") of the

       elif n = 1 and d = 2+2*degree(num,_X) and type(den,polynom(polynom(rational),_X
            and psqrt(den+1) <> _NOSQRT and rem(diff(den+1,_X),num,_X) = 0 then
  22     r := psqrt(den+1);
  23     answer := const/content(r,_X)*arctanh(r)
                       ^^^^^^^^^^^^
```

Same bug here.

```
      elif d = 2 then
 24     answer := const*`int/ratpoly/quadratic`(num,den,n-1)
      elif d = 3 then
 25     r := `int/ratpoly/cubic`(num,den,n);
 26     answer := `if`(r = FAIL,FAIL,const*r)
      elif 0 < ldegree(num,_X) and `int/ratpoly/subs`(num,den,'k') then
 27     num := collect(num/(_X^(k-1)),_X);
 28     r := `int/indef1`(subs(_X = _X^(1/k),num/(den^n)));
 29     answer := 1/k*const*subs(_X = _X^k,r)
      elif irem(d,2,'q') = 0 and degree(num,_X) = q-1 and
           `int/ratpoly/arctan`(num,den^(n+1),'q') then
 30     answer := const*q
      elif d = 4 then
 31     r := `int/ratpoly/quartic`(num,den,n);
 32     answer := `if`(r = FAIL,FAIL,const*r)
      elif d = 5 then
 33     r := `int/ratpoly/quintic`(num,den,n-1);
 34     answer := `if`(r = FAIL,FAIL,const*r)
      elif d = 6 then
 35     r := `int/ratpoly/sextic`(num,den,n-1);
 36     answer := `if`(r = FAIL,FAIL,const*r)
      end if;
 37   if answer = FAIL then
 38     if type(f,'ratpoly(rational,_X)') then
 39       const*`int/risch/ratpoly`(num/(den^n),_X)
        else
 40       g, const2, A, B := `int/ratpoly/horowitz`(num/(den^n),_X);
 41       if A = 0 then
 42         const*g
          else
 43         const*(g+const2*sum(subs(_X = _R,A/diff(B,_X))*ln(_X-_R),_R = RootOf(B,_
          end if
        end if
      else
 44     answer
      end if
end proc
```

An example of the bug in action:

```
> int(c/((a*x+b)^2+1), x);

                          c arctan(a x + b)
```

Before you test my fix, make sure to do a restart to clear the erroneous answers from the remember tables.

```
> restart;
```

Since the "content" command is used only on the erroneous lines, you can easily correct this bug:

```
> `int/ratpoly/ratpoly`:= subs(content= lcoeff, eval(`int/ratpoly/ratpoly`)):
```

You could also use "diff" instead of "lcoeff". That would be more natural.

Test it:

```
> int(c/((a*x+b)^2+1), x);

                          c arctan(a x + b)
                          -----------------
                                  a

> int(1/(9*x^2+6*x+2), x);

                         1/3 arctan(1 + 3 x)
```

In release 5, these special cases are not treated. They are merely passed on to ‘int/ratpoly/quadratic‘ as in line 24 above. Why were these cases singled out in release 6?

I am not absolutely sure that my fix works in all cases. Be wary when doing integrals with a quadratic denominator.

7.88 bug in integration in Maple 6 and Maple 7 (17.9.01)

7.88.1 James R. FitzSimons

Maple 6 can not integrate this.

```
> Digits:=32;
> evalf(3*Int(sqrt(689*t^4-1492*t^3+1076*t^2-292*t+29),t=0..1));

                              Digits := 32

Error, (in evalf/int) unable to handle singularity
```

I tried int and it hung up so long I gave up.

I tried this on a demo of MuPad and it had no problem with this integral.

Does Maple 7 do better?

With Maple 7 you get no error message, but also no result (U. Klein)

It is corrected with Maple 8 (U. Klein)

7.88.2 Koch-Beuttenmueller (19.9.01)

I tried your integral in Maple7 with different Maple methods, different settings of Digits (between 25 and 35, but inside Int) and different integration boundaries (from minima to minima etc). I got a lot of confusing results when Maples will calculate it and when not. When I have a little bit more time, I think I will send it to the support.

Integration with `method =_Sinc` worked best.

7.88.3 Carl DeVore (19.9.01)

The function has no singularity. The polynomial is never even close to 0 (min of poly is about 1.58). I haven't looked too deeply in the code, but I suspect that evalf/int will report a singularity simply because it cannot control the error at the high setting of Digits – some preset maximum number of function evaluations is exceeded. At the time of failure, it is trying a Newton-Cotes rule with a fixed precision of about 15 digits (see `?int,numerical`), so there's no way it will get 32 Digits (so why does it waste time trying?) Setting `infolevel[all]:= 5` is quite instructive for this problem.

A very close reading of `?int,numerical` leads to the following:

```
> evalf(3*Int(t-> sqrt(689*t^4-1492*t^3+1076*t^2-292*t+29), 0..1));

                  7.237223368328592885826619210956022413967067986017504163362886501215 4
```

Very oddly, this syntax means to try the double exponential algorithm `(_Dexp)` without any singularity checking. How's that for an inconsistent `syntax?` This leaves the question of why the `_Dexp` reported a singularity in the first place. I haven't looked. It had only tried about 140 function evaluations before it gave up.

How does that compare with the MuPad answer?

7.88.4 Jim FitzSimons (19.9.01)

It was a MuPAD demo which has expired. The result using DERIVE is

```
refint:=7.237223368328592885826619210956022413967067986017504163362886501215 4~
88869007574127674389921879675643065604886427388322169063078 7
```

This is the same value I got using MuPAD.

7.89 bug in integration in Maple 6 and Maple 7 (27.11.00)

7.89.1 Halvor Mehlum

I get the following inconsistency on my Windows 98 running Maple 6

As part of my work I executed the following code:

```
a1 := .9*(.9+8.*exp(-1))*exp(.1*tt)/(1+8*exp(-1)):
b1:=int(exp(-tt)*ln(a2),tt=t..1):
c1:=evalf(subs(t=0.5,b1));
a2:=simplify(a1):
b2:=int(exp(-tt)*ln(a1),tt=t..1):
c2:=evalf(subs(t=0.5,b2));
The result was as follows

    c1 := -.01387124925
    c2 := -.02000179884
```

which is surprising as they are supposed to be identical. The expressions b1 and b2 are also different. Does the same problem arise on other machines?

It is corrected with Maple 8 (U. Klein)

7.89.2 Don Hartig (27.11.00)

I assume that the a2 appearing in the expression for b1 should actually be a1. If that is the case, then I get the same values for c1 and c2. I also get the same values for b1 and b2.

I am running Maple 6.1 on a Macintosh G4.

7.89.3 Robert Israel (28.11.00)

Yes, it's a bug. It shows up in a simpler form here:

```
> f:= exp(-t)*ln(exp(t/2)/(1+exp(1)));
  int(f,t);

                    2                  2
  - 1/2 exp(- 1/2 t)  + 2 exp(- 1/2 t)  ln(1 + exp(1))

                              2
         - 1/2 t exp(- 1/2 t)
```

The 2 in front of the second term should be 1. The bug seems to be new in Maple 6: it doesn't affect Maple V Releases 4 to 5.1.

This is corrected with Maple 7. (U. Klein)

7.89.4 Heike Koch-Beuttenmueller (28.11.00)

Things seems to be even much stranger:

I tried the following on Compac Alpha:

```
d1 := 9/10*(9/10+8*exp(-1))*exp(1/10*tt)/(1+8*exp(-1));

int(exp(-tt)*log(d1),tt=1/2..1);

-1/5*exp(-1)-2*exp(-1)*ln(3)+3*exp(-1)*ln(2)+3*exp(-1)*ln(5)-2*exp(-1)
*ln(9+80*exp(-1))+2*exp(-1)*ln(1+8*exp(-1))+3/20*exp(-1/2)+2*exp(-1/2)
*ln(3)-3*exp(-1/2)*ln(2)-3*exp(-1/2)*ln(5)+2*exp(-1/2)*ln(9+80*exp(-1)

evalf(%,30);
                  -.020001797140674999584745129 72

int(exp(-tt)*log(d1),tt=0.5..1);

                           -.01387125093
```

Mathematica gives the last result. Numerical integration with Maple as well.

7.90 bug in integration in Maple 6 (11.6.01)

7.90.1 Liu Yuting

```
> int(log(sin(t)), t=0..Pi);
```

Maple 6 gives 0 (wrong).

```
> int(log(sin(x)), x=0..Pi);
```

Maple 6 gives `-Pi*ln(2)` (correct).

It seems that only 'x' works. Would anyone like to fix it?

7.90.2 Jason Schattman (15.6.01)

This bug has been fixed in Maple 7. Both of the problems given below output `-Pi*ln(2)`.

7.90.3 Helmut Kahovec (16.6.01)

On Wed, 16 May 2001 at `12:05:34` GMT I wrote in comp.soft-sys.math.maple (answering the same question):

The bug is in the global table `` `int/itable` ``, which is used in a table lookup at line 2 of `` `int/indef1`() ``. There is a typo in the entries for `ln(cos(_X)) and ln(sin(_X))` where an x appears instead of an `_X`:

```
> eval(`int/itable`[ln(cos(_X))]);

                                                              2
        1/2 I polylog(2, -exp(2 I _X)) - x ln(2) - 1/2 I _X
                                           ^

> eval(`int/itable`[ln(sin(_X))]);

                                                                  2
   -1/2 I polylog(2, exp(-2 I _X)) - x ln(2) + 1/2 I (1/2 Pi - _X)
                                      ^
```

I am quite sure that the people at Maplesoft already know this bug.

In order to fix this bug we only have to modify those two entries:

```
> restart;

> `int/itable`[ln(cos(_X))]:=
>   1/2*I*polylog(2,-exp(2*I*_X))-_X*ln(2)-1/2*I*_X^2:
                                    ^^
                                    ||
_____________________________________||

> `int/itable`[ln(sin(_X))]:=
>   -1/2*I*polylog(2,exp(-2*I*_X))-_X*ln(2)+1/2*I*(1/2*Pi-_X)^2:
                                     ^^
                                     ||
_____________________________________||
```

Now we get in turn:

```
> int(log(sin(t)),t=0..Pi);

                                -Pi ln(2)

> int(log(sin(x)),x=0..Pi);

                                -Pi ln(2)

> int(log(cos(t)),t=-Pi/2..Pi/2);

                                -Pi ln(2)

> int(log(cos(x)),x=-Pi/2..Pi/2);

                                -Pi ln(2)
```

7.90.4 Carl DeVore (19.6.01)

This integral is done as a simple table lookup. Someone typed x instead of _X into the table. To fix it:

```
> `int/itable`:= subs(x= _X, eval(`int/itable`));
```

This brings up a more general and severe point. There needs to be a more mechanized way of checking these things, of finding these bugs. This is essential if Maple is ever going to be used for some critical, real-time, life-or-death application – for example, traffic control. If you misspell a variable name in Maple, that error is not detected. It just assumes that you mean some other undefined "name". There should be an option that produces a warning whenever the parser detects an undeclared name; and, of course, to match that, there should be a way to declare names other than procedure parameters.

7.90.5 Humberto Jose Bortolossi (22.6.01)

What about a free patch for Maple 6.x owners? Always we have to upgrade (and pay more money) to get bugs fixed???

7.90.6 Robert Israel (22.6.01)

Carl DeVore wrote: This brings up a more general and severe point. ...

I really hope nobody ever uses Maple for critical, life-or-death applications. It's much too complicated to ever be sure that there are no more bugs, even if much more effort was put into searching for and correcting them. That type of application needs a much different style of programming.

Sufficiently complicated software may be impossible to certify as bug-free anyway - cf. the debate some years ago in connection with the "Star Wars" missile defence system.

Sun's Java Development Kit comes with a disclaimer:

Software is not designed or intended for use in on-line control of aircraft, air traffic, aircraft navigation or aircraft communications; or in the design, construction, operation or maintenance of any nuclear facility. Licensee warrants that it will not use or redistribute the Software for such purposes.

Maybe Maple should too.

7.90.7 Joe Riel (home) (27.6.01)

Carl DeVore wrote: If you misspell a variable name in Maple, ...

The solution is to use mint, it detects unused and undeclared variables.

7.91 bug in integration in Maple 7 and correction (15.4.02)

7.91.1 Marko Horbatsch

In case this bug coincides with a previous report, my apologies, but it seems that the developers are not checking their software updates with an appropriate suite of integrals, they should add the following:

```
int(ln(x)/(1-x),x=0..1);
```

should result in `-Pi^2/6`

but gives infinity (maple6 is OK on this one).

If one uses the anti-derivative, and takes the limit for the lower limit, the correct result is returned; it is however the right-hand integration limit which seems to throw maple7, because changing that to something else than 1, makes the integral work.

Of course, this hits a class of integrals, i.e., including powers of `ln(x)` produces similarly erroneous results.

for

```
f:=ln(x)^3/(1-x)^3;

int(f,x=0..1);
```

isn't wrong, but doesn't evaluate, even though taking the limit on the anti-derivative works,

```
limit(int(f,x=a..1),a=0,right);
```

works out. This complaint applies to both maple7 and maple6, i.e., apparently maple does not try on these improper integrals taking the antiderivative and then performing limits to obtain the definite integral.

It is corrected with Maple 8 (U. Klein)

7.91.2 Helmut Kahovec (25.4.02)

There seems to be a bug in ‘int/cook/IIntd1c‘ of Maple7, which defines two different versions of ‘int/cook/ngritty‘. The second one evaluates the derivative of the Beta(z,w) function at w=0 although the value of w may be zero. This evaluation should be replaced by the limit lim w->0. Then we get the following results:

```
> `int/cook/IIntd1c`:=proc(
    t,ta,tb,param,model,triglabel,ans,fltype,fail
  )
  local cof,a,b,c,d,dl,a0,a1,f,m,w,np1,p,u,u2,r,s,s2,sp,ucomplex,
        v,z,btype,ptype,a0type,a1type,wtype,dtype,dltype,a0fl,
        a1fl,bfl,wfp1l,dfl,dlfl,pfm1l,B,S1,S2,D,DL,A0,A1,C1,M,
        N,P,R,Ucplex,U2,MODEL,failtemp,fltypetemp,res,res1,res2,
        i,sig,dc1,dc2,dc3,dc4,`int/cook/ngritty`;
    failtemp:=0;
    cof:=param[1];
    b:=param[2];
    u:=-op(1,param[3]);
    u2:=-op(2,param[3]);
    s:=op(1,param[4]);
    s2:=op(2,param[4]);
    dl:=param[5];
    m:=param[6];
    w:=param[7];
    p:=param[8];
    a0:=param[9];
    a1:=param[10];
    c:= param[11];
    r:=param[12];
    d:=param[13];
    sp:=param[14];
    f:=param[15];
    MODEL:=exp(-Ucplex*t^S1)*t^N*ln(B*t^DL)^M*cos(C1*t^R)/
      ((A0+A1*f^D)^P);
    d:=`int/cook/secure`(d,'dfl','dtype');
    p:=`int/cook/secure`(p-1,'pfm1l','ptype')+1;
    w:=`int/cook/secure`(w+1,'wfp1l','wtype')-1;
    b:=`int/cook/secure`(b,'bfl','btype');
```

```
dl:=`int/cook/secure`(dl,'dlfl','dltype');
a0:=`int/cook/secure`(a0,'a0fl','a0type');
a1:=`int/cook/secure`(a1,'a1fl','a1type');
if d<>0 then
  np1:=(w+1)/d
else
  fail:=2;
  `int/cook/nogo1`(MODEL,model,t,ta,tb,s);
  return
end if;
if not type(pfm1l,constant) then
  fail:=2;
  `int/cook/nogo2`(MODEL,model,t,ta,tb,p);
  return
end if;
if not type(wfp1l,constant) then
  fail:=2;
  `int/cook/nogo2`(MODEL,model,t,ta,tb,w+1);
  return
end if;
if type([a0,a1],[constant,constant]) and
   type([a1fl,a0fl],[numeric,numeric]) and
   abs(a0fl)<abs(a1fl)
then
  `int/cook/nogo1`(MODEL,model,t,ta,tb,abs(a1)-abs(a0));
  return
end if;
if not type(dfl,constant) then
  fail:=2;
  `int/cook/nogo2`(MODEL,model,t,ta,tb,d);
  return
end if;
if u=0 and u2=0 and m=0 and c=0 and sp=0 and
   f=t and pfm1l<1 and 0<dfl and abs(a0/a1)<>1
then
  sig:=signum(a0);
  if not type(p,integer) and (type(sig,'function') and
     op(0,sig)=('signum') or sig=-1)
  then
```

```
      fail:=2;
      if 2<printlevel then
        `int/cook/nogo1`(MODEL,model,t,ta,tb);
        printf("--> Does not fit into this sub-class\n")
      end if;
      return
    end if;
    s:=d;
    `int/cook/ngritty`:=proc(s,dc1,np1,p,dc2,a0,a1,dc3,dc4)
    local lam,b,Kt,F;
      lam:=np1;
      b:=p;
      Kt:=-a1/a0;
      F:=Beta(lam,1)*hypergeom([b,lam],[lam+1],Kt);
      simplify(a0^(-p))*simplify(F)/s
    end proc
  elif u=0 and u2=0 and a1=-a0 and c=0 and sp=0 and f=t and
       pfm1l<=0 and 0<dfl and type(m,integer) and 0<=m and
       type(d,freeof(t))
  then
    if ta=0 and wfp1l<=0 then
      z:=limit(t*model,t=0,'right');
      if hastype(z,'infinity') then
        v:=limit((t-1)*model,t=1,'left');
        if signum(z)=signum(v) or not hastype(v,'infinity') then
          ans:=z;
          fail:=4;
          return
        elif has(v,'infinity') then
          fail:=3;
          if 2<printlevel then
            `int/cook/nogo1`(MODEL,model,t,ta,tb);
            printf("integral is most likely divergent\n");
            printf("Cauchy principal value ?\n")
          end if;
          return
        end if
      end if
    end if;
```

```
    s:=d;
    z:=np1;
    v:=-p+1;
    `int/cook/ngritty`:=proc(s,m,np1,p,z0,a0,a1,dc2,v)
    local z,w,F;
      F:=Beta(z,w);
      if 0<m then F:=diff(F,`$`(z,m)) end if;
      try
        if v=0 then                                              #<==
          simplify(                                              #<==
            a0^(-p))*simplify(limit(eval(F,z=z0),w=v))/(s^(m+1)) #<==
          )                                                      #<==
        else                                                     #<==
          simplify(                                              #<==
            a0^(-p))*simplify(eval(F,{z=z0,w=v}))/(s^(m+1)       #<==
          )                                                      #<==
        end if                                                   #<==
      catch "numeric exception: division by zero":
        a0^(-p)/(s^(m+1))*infinity
      end try
    end proc:
  else
    fail:=2;
    if 2<printlevel then
      `int/cook/nogo1`(MODEL,model,t,ta,tb);
      printf("--> Does not fit into sub-classes\n")
    end if;
    return
  end if;
  if (bfl<>`int/cook/insecure`(1,btype) or 3<btype) and
      bfl<>0 and dlfl<>0
  then
    res:=0;
    for i from 0 to m do
      res:=res+
        binomial(m,i)*ln(b)^(m-i)*dl^i*
        `int/cook/ngritty`(s,i,np1,p,z,a0,a1,ucomplex,v)
    end do
  elif bfl=`int/cook/insecure`(1,btype) and dlfl<>0 then
```

```
      res:=dl^m*`int/cook/ngritty`(s,m,np1,p,z,a0,a1,ucomplex,v)
    else
      res:=`int/cook/ngritty`(s,m,np1,p,z,a0,a1,ucomplex,v)
    end if;
    res:=cof*res;
    if has(res,abs) then res:=`int/cook/abs`(res) end if;
    if has(res,'infinity') then failtemp:=4 end if;
    if fltypetemp=1 then res:=evalf(res) end if;
    ans:=eval(res);
    fltype:=fltypetemp;
    fail:=failtemp
  end proc:
```

```
> int(ln(x)/(1-x),x=0..1);

                                    2
                               - 1/6 Pi

> int(ln(x)^2/(1-x)^2,x=0..1);

                                     2
                                1/3 Pi

> j:=ln(x)^3/(1-x)^3;

                                      3
                                   ln(x)
                            j := --------
                                       3
                                 (1 - x)

> J:=applyop(expand,1,int(j,x=0..1));

                                    2   1/2       ln(1 - exp(1/y))
  J :=   lim    -3 Zeta(3) + 1/2 Pi  + --- - 3/2 ----------------
       y -> 0+                         3                 2
                                       y                 y
```

```
          3 polylog(2, exp(1/y))                                  3/2
        - ---------------------- + 3 polylog(3, exp(1/y)) + ---
                    y                                          2
                                                              y

          3 ln(1 - exp(1/y))
        - ------------------ - 3 polylog(2, exp(1/y))
                  y
```

Maple cannot compute that limit symbolically. However, it will return the numerical value of J:

```
> simplify(fnormal(evalf(J)),zero);

                              -8.540972910
```

Let us check this result:

```
> int(j,x=a..1);

                                 1
                                /        3
                               |    ln(x)
                               |   -------- dx
                               |          3
                              /    (1 - x)
                                a

> limit(%,a=0,right);

                                    2
                        - 1/2 Pi  - 3 Zeta(3)

> evalf(%);

                              -8.540972911
```

7.92 bug in integration in Maple V.4 and Mathematica 3.0 (22.8.97)

7.92.1 luca ciotti

I think I found a (serious?) bug in definite integration both in Maple Release 4 (unix version) and Mathematica 3.0 (linux version).

The two softwares produce different answers, both wrong.

The integral is `I=\int_0^{pi/2} dx/sqrt{sin(x)+cos(x)`

Maple 3, Maple 4 and Mathematica 3 all give the same numerical result doing numerical integration, i.e.

```
I=1.3974....
```

If you change the input from I := int(...) to evalf(Int(...)) Maple V Release 5 yields to 1.770261042. (U. Klein)

Symbolic integration in Maple 4 produces a result in term of elliptic integrals, that evaluated with evalf() is exactly

`-1.3974 ....`

With Maple V Release 5 you get: 1.77026.... (U. Klein)

Symbolic integration in Mathematica 3 produces a result in term of HypergeometricPFQ, that evaluated with `N[]` is

`-3.01236`

Changing variable of integration as `y=tan(x/2)` produces a symbolic result in Maple and Mathematica equal to `1.3974....`

7.92.2 Sjoerd W. Rienstra (27.8.97)

It could be a bug, but very likely it is an inherent risk of symbolic calculation. A branch of the multivalued function, other than expected or meant, may be used by the system. Or the primitive of a multivalued function (like a square root, log, arcsin, arctan, etc.) is evaluated at both sides of the branch cut, so that the result includes the additional jump over the branch cut.

Never forget that a computer program is ONLY reliable when it hasn't passed the compiler yet. When the compiler doesn't complain anymore and the program produces numbers, it sometimes occurs (I heard some rumours of rare occasions, long ago ....) that the results are allright.

7.93 bug in invphi, Maple V.3 to Maple 8 (20.6.02)

7.93.1 Neil J. A. Sloane

Is this a known bug?

```
FRY$ maple
    |\^/|     Maple 6 (SGI MIPS UNIX)
._|\|   |/|_. Copyright (c) 2000 by Waterloo Maple Inc.
 \  MAPLE  /  All rights reserved. Maple is a registered trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> with(numtheory):
Warning, the protected name order has been redefined and unprotected

> invphi(1610510);
                                                    []

> phi(1771561);
                                                 1610510
\begin{MAPLEinline}

In other words \verb|invphi(1610510);| gave an incorrect answer.

%%%%%%%%%%%%%%%%%%%
\subsection{\href{mailto:meade@math.sc.edu}{Douglas B. Meade       }(25.6.02)}

I have not worked with Maple's invphi function for several
years (Maple V, Release 2 in 1993) but remember that it was not
very efficient and was not perfect. At that time I wrote some
general pre-image routines for integer-valued functions. These have not
been updated to later versions of Maple but should not be difficult to
update. The PS and PDF versions of the technical report I wrote at that
time can be downloaded from the list of publication on my website
\url{http://www.math.sc.edu/~meade/publ.html}.

I do not have time to update the code but will be happy to answer
any questions that you might have.
```

```
%%%%%%%%%%%%%%%%%%%
\subsection{\href{mailto:devore@math.udel.edu}{Carl Devore      } (27.6.02)}

My investigation so far shows that this bug will occur for
\verb|invphi((p-1)*p^n) for n>1| and p any of the primes
11, 23, 29, 31, 47, 53, 59, 67, 71, 79, 83, 89, 103, 107, 113, 131, ....

I can't see the pattern of the primes, except that it appears
that if p is such a prime, then so are all primes of the
form \verb|(4*n+2)*p+1, n= 0, 1, ....|

Tracking down the bug and verifying the above conjecture might
lead to an interesting result. Perhaps it is one of your
Integer Sequences. So the smallest example that I can find
for this bug is \verb|invphi(10*11^2)|.


%%%%
\clearpage\section{bug in is, Maple 8   (10.6.02)}
%%%%%%%%%%%%%%%%%%%
\subsection{\href{mailto:HJaramillo@houston.westerngeco.slb.com}{Herman Jaramillo}}

In Maple 7.

\begin{MAPLEinline}
 > is(cos(a)*cos(a)-sin(b)*sin(b)=cos(b)*cos(b)-sin(a)*sin(a));

                          FAIL

 > simplify(cos(a)*cos(a)-sin(b)*sin(b)-cos(b)*cos(b)+sin(a)*sin(a));

                            0
```

Is this a bug?.

The same result for Maple V.3, V.5 and 6 (U. Klein)

7.93.2 Robert Israel (11.6.02)

Perhaps more of a weakness than a bug. "is" does not take identities among transcendental functions into account when checking equalities.

Unfortunately the help page does not mention this. For example:

```
> is(exp(2*x) = exp(x)^2);

                FAIL
```

Interestingly, however:

```
> is(exp(2*x)<=exp(x)^2) assuming x::real;

                true

> is(exp(2*x)>=exp(x)^2) assuming x::real;

                true
```

BTW: does anybody have an example where "_EnvTry:= hard" makes a difference in the result of "is"?

7.93.3 Carl Devore (12.6.02)

The FAIL indicates that "is" could not figure out the answer, so I would very rarely consider an answer of FAIL to be a bug.

However, it seems like "is" has been terribly messed up in Maple 8. In Maple 8, the above "is" returns false.

7.93.4 RICHARD YOUNG (12.6.02)

Maple V release 4 returns a true.

7.94 bug in iscont/piecewise in Maple V.4 (28.8.97)

7.94.1 Jean Brillet

Is it a bug in iscont function ?

```
> p:=x->piecewise(x>=0,cos(x),exp(-x));
> iscont(p(x),x=-infinity..infinity);

                          false
```

Also, it seems that, sometimes, piecewise functions are not evaluated correctly when arguments have not numeric type :

```
>fx:=x->piecewise(x>-2 and x<2,cos(x),x>=2,cos(2),exp(-abs(x))):
>fx(Pi);

       PIECEWISE([-1, -2-Pi < 0 and Pi-2 < 0],[cos(2), 2 <= Pi],[exp(-Pi),
                  otherwise])

>fx(ln(2));

       PIECEWISE([cos(ln(2)), -2-ln(2) < 0 and ln(2)-2 < 0],[cos(2), 2 <=
                  ln(2)],[1/2, otherwise])
Results of fx(evalf(Pi)) and fx(evalf(ln(2)) are correct
```

However,

```
> fz:=z->piecewise(z<Pi/2,1,2):
> fz(0);
                  1
> fz(exp(1));
                  2
```

What happens ? Is it due to MAPLE difficulty to evaluate bolean expressions with symbolic constants

The bug is removed with Maple V Release 5. (U. Klein)

7.94.2 Robert Israel (4.9.97)

| Is it a bug in iscont function ?

Sort of. Really you might call it a bit of false advertising. The help for "iscont" claims

> `The iscont function returns true if the expression is continuous on the interval (thus having no poles), or false if the expression is not continuous. If iscont cannot determine the result it returns FAIL.`

However, this is not really true. "iscont" basically uses the "discont" function to find the real discontinuities of your expression, and then checks whether any of these is in the interval given. But the help for "discont" says

> `discont returns a set of values where it is possible (not necessarily certain) that discontinuities occur.`

So a "false" result from iscont is only an indication that there might be a discontinuity, not a guarantee that there is one. On the other hand, a "true" result is not necessarily reliable either. In many cases Maple tries to use "solve" to find, say, a zero of the denominator of a fraction, and the capabilities of "solve" are limited. For example,

```
> iscont(1/(cos(x)-x^2), x = -Pi .. Pi);
                              true
```

From examining the code for `'discont/discontR'`, it seems to me that for any piecewise expression (unless the expressions in two intervals are actually the same), "discont" will always claim there is a discontinuity at the boundaries of the intervals. Moreover, for an expression that is not a rational function but is a sum or product of terms, "discont" will report any point that is a discontinuity of any of those terms; it does not try to check whether there is a cancellation of some sort.

Conclusion: be very skeptical about the results of "iscont" or "discont".

| Also, it seems that, sometimes, piecewise functions are not evaluated ...

This is not really a failure to evaluate correctly, it is a failure to simplify.

```
> simplify(fx(Pi));

                cos(2)

> simplify(fx(ln(2)));
```

```
            cos(ln(2))
```

Perhaps the "and" causes Maple some confusion:

```
> fxp:= x -> piecewise(x  <= -2, exp(-abs(x)), x >= 2, cos(2), cos(x)):
> fxp(Pi);
            cos(2)
> fxp(ln(2))
            cos(ln(2))
```

But here's a genuine bug:

```
> simplify(fx(x));
                                  {     3 cos(x) - 3 cos(2)          x <= -2
                                  {
                                  { 4 cos(x) - exp(x) - 3 cos(2)     x < 2
                                  {
                                  {       cos(2) - exp(2)            x = 2
                                  {
                                  {  2 cos(x) - exp(x) - cos(2)      2 < x
```

| However, ...

What's your complaint here? These results are perfectly correct.

7.95 Bug in Jacobi elliptic function, Maple V to Maple 8 (2.11.98)

7.95.1 Robert Michael Sinclair

The expressions returned by Maple VR5 for the partial derivatives of `JacobiSN(z,k)`, `JacobiCN(z,k)` and `JacobiDN(z,k)` with respect to k are not correct for all values of z. The reason is that they contain the subexpression `EllipticE(JacobiSN(z, k), k)` instead of the integral `int(JacobiDN(v,k)^2,v=0..z)`. While it is true that both are equal for `-EllipticK(k) <= z <= EllipticK(k)`, they are not in general equal for z outside this interval.

The example below illustrates the problem. The function d1 is Maple's expression, d2 is the correct expression using the integral given above, and d3 is a difference quotient approximation to the partial derivative. In the first numerical comparison, z is less than `EllipticK(k)` `(1.6<1.68...)`, but z is greater than `EllipticK(k)` in the other two. In the final comparison, both the sign and order of magnitude of Maple's expression are incorrect.

```
    |\^/|     Maple V Release 5 (WMI Campus Wide License)
._|\|   |/|_. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> d1:=unapply(diff(JacobiSN(z,k),k),z,k):
> d2:=(z,k)->k/(1-k^2)*JacobiSN(z,k)-k/(1-k^2)*JacobiSN(z,k)^3
>     +z/k*JacobiCN(z,k)*JacobiDN(z,k)
>     -1/k/(1-k^2)*JacobiCN(z,k)*JacobiDN(z,k)*int(JacobiDN(v,k)^2,v=0..z):
> d3:=(z,k)->(JacobiSN(z,k+1e-10)-JacobiSN(z,k))/1e-10:
> Digits:=50:
> EllipticK(0.5);
                  1.6857503548125960428712036577990769895008008941411

> zk:=1.6,0.5: d1(zk), evalf(d2(zk)), d3(zk);
    -.031182163668468368837824341392897355040055016301907,
    -.031182163668468368837824341392897355040055016301903,
    -.031182163680558307555413752432763256031 5

> zk:=1.7,0.5: d1(zk), evalf(d2(zk)), d3(zk);
```

```
     .005282011499481807709953789681659890276144429069717,
     .005891184123212836060745180249782058869208078728551,
     .005891184112453688742108429299771793799 4

> zk:=10.0,0.5: d1(zk), evalf(d2(zk)), d3(zk);
  -19.459115391501625785648272200645954534230450114114,
    3.223614644063210632808934334083732420054336805507,
    3.223614644547415117621187123986166083552 4
```

See also: Bug in derivative of JacobiDN

7.96 bug in Laplace, Maple V.5 (18.2.00)

7.96.1 Ray Vickson

The following behavior is almost unbelievable:

```
> restart:
> with(inttrans):
> f1:=(s+1)/(s*(s^2+6*s+9));

                                    s + 1
                        f1 := ----------------
                                     2
                                s (s  + 6 s + 9)

> f2:=1/(s*(s^2+6*s+9));

                                      1
                        f2 := ----------------
                                     2
                                s (s  + 6 s + 9)

> invlaplace(f1,s,t);

                 1/9 + 2/3 t exp(-3 t) - 1/9 exp(-3 t)

> invlaplace(f2,s,t);
Error, (in tablelook) division by zero
```

Try as I might, I cannot trace the source of the problem. How can something so simple fail so badly?

It is corrected with Maple 6. (U. Klein)

7.96.2 SCHADE, GEORGE (21.2.00)

I tried this on windows 95, release 4, and it worked fine.

7.96.3 Harald Pleym (21.2.00)

I don't know why, but I know that Maple6 (version 16401) produce the same unbelievable error.

A work around is

```
> invlaplace(convert(f2,parfrac,s),s,t);

                                  1/9-1/3*t*exp(-3*t)-1/9*exp(-3*t)
```

7.96.4 jmw (21.2.00)

Yet when f2 is expressed differently, things seem to work:

```
with(inttrans):

f2:=1/(s*(s+3)^2);
                                       1
                              f2 := ----------
                                           2
                                    s (s + 3)
 > invlaplace(f2,s,t);

                        1/9 - 1/9 (1 + 3 t) exp(-3 t)
```

7.96.5 Willard, Daniel Dr (21.2.00)

Just for kicks, I tried `f3:=1/((s+a)*(s^2+6*s+9))` and took `(limit(invlaplace(f3,s,t),a=0))`. That works.

7.96.6 Wilhelm Werner (24.2.00)

Maybe the following gives a hint to what is happening:

```
>with(inttrans): f2:=1/(s*(s^2+a*s+b));invlaplace(f2,s,t);

                    exp((- 1/2 a + 1/2 %1) t)   exp((- 1/2 a - 1/2 %1) t)
                    ------------------------- - -------------------------
                        - 1/2 a + 1/2 %1            - 1/2 a - 1/2 %1
               1/b + ------------------------------------------------------
                                          2
                                       sqrt(a  - 4 b)

                                            2
                                  %1 := sqrt(a  - 4 b)
```

It seems to me that Maple looks into a table containing the above formula and then substitutes `a=6 and b=9` (instead of using something like `l'Hospitals` rule in this case).

7.97 Bug in limit (3.3.97)

7.97.1 Preben Alsholm

I stumbled on the following bug in limit.

Let f and x in f(x) both be unassigned names:

```
limit( f(x)*exp(-x), x=infinity);
```

Maple answers 0, but should obviously return unevaluated.

It seems that when it doesn't know enough about f(x), then it returns zero:

```
limit( -I*erf(I*x)*exp(-x), x=infinity);
```

returns 0, but should return infinity (or unevaluated).

The procedure for limit is complicated. I didn't get very far in tracking the bug, but did find that

```
`limit/series`( f(1/x)*exp(-1/x),x,right);
```

correctly returns FAIL. `'limit/series'` is called from within `'limit/limit'` which in turn is called from limit.

The bug

```
limit( -I*erf(I*x)*exp(-x), x=infinity);

                              0
```

may be due to the following feature

```
limit( f(x)*g(x), x=0);

                          f(0)g(0)
```

which e.g. results in

```
limit( f(x)*x, x=0);
```

```
0
```

I would prefer that all three returned unevaluated.

7.98 Bug in limit(c/x/ln(x)) in Maple V.4 and Maple V.5 (13.8.97)

7.98.1 Andreas Jung

There is a sign bug in limit in the following cases (Maple V Release 4 under Solaris on a Sun workstation):

```
> limit(42/x/ln(x), x=0, right);
                                    -infinity
```

Ok, that's right. But:

```
> limit(c/x/ln(x), x=0, right);
                                signum(c) infinity
```

The correct result should be `-signum(c)*infinity`. (Note that not the complex limit should be taken, but the real one). Interestingly, applying a minus sign has no influence on the result:

```
> limit(-c/x/ln(x), x=0, right);
                                signum(c) infinity
```

So the result isn't always wrong. It seems as if the numerator isn't completely considered if it contains a variable. For example, try `17^e` instead of c.

Assuming c to be real has no effect.

It is corrected with Maple 6. (U. Klein)

7.99 bug in limit-taylor-series, MapleVr5 and Maple 6 (31.5.00)

7.99.1 Ivan Huerta

In the following example limit() fails when x is numeric. Same results with MapleV 5 and Maple 6 on Win98. If x is a variable the results are ok.

```
> f:= x-> sin(2*Pi*x+Pi/2);

  f := x -> sin(2 Pi x + 1/2 Pi)

> limit( (f(41/100+h)-f(41/100))/h,h=0);

                                    0
```

Curiously enough the series expansion for this expression gives a leading term `(-cos(9/50 Pi) + sin(8/25 Pi)) 1/h`, but `(-cos(9/50 Pi) + sin(8/25 Pi)=0`, which fools taylor.

```
> series( (f(41/100+h)- f(41/100) )/h,h=0);

                                -1                                          2
(-cos(9/50 Pi) + sin(8/25 Pi)) h   - 2 sin(9/50 Pi) Pi + 2 cos(9/50 Pi) Pi  h

                          3  2                       4  3
     + 4/3 sin(9/50 Pi) Pi  h  - 2/3 cos(9/50 Pi) Pi  h  -

                        5  4      5
    4/15 sin(9/50 Pi) Pi  h  + O(h )


> taylor((f(41/100+h)-f(41/100))/h,h=0);

   Error, does not have a taylor expansion, try series()


> series((f(x+h)-f(x))/h,h=0);
```

```
                                        2                          3  2
 - 2 sin(2 Pi x) Pi - 2 cos(2 Pi x) Pi  h + 4/3 sin(2 Pi x) Pi  h  +

                      4  3                        5  4      5
   2/3 cos(2 Pi x) Pi  h  - 4/15 sin(2 Pi x) Pi  h  + O(h )

> limit((f(x+h)-f(x))/h,h=0);

                        -4 Pi sin(Pi x) cos(Pi x)
```

7.99.2 Robert Israel (5.6.00)

Well, I think you provided the clue yourself: "limit" looks at what it thinks is the leading term of a series expansion. It's not so obvious why, if Maple thinks there's a `1/h` term in the series, it comes up with 0 rather than `+infinity` or `-infinity` or undefined: I think that's because it looks at the signum of the coefficient of the leading term in order to decide which of these to return, and "signum" is clever enough to know that `-cos(9/50 Pi) + sin(8/25 Pi)=0`.

The way that series tests whether something is zero is specified in the environment variable `Testzero`, which is usually set to just use "normal". In this case it will work better if you use "is" instead:

```
> restart;
  Testzero:= proc(x) is(x,0) end;
  f := x -> sin(2*Pi*x + Pi/2);
  limit( (f(41/100+h)-f(41/100))/h,h=0);

       -2 sin(9/50 Pi) Pi
```

7.99.3 Helmut Kahovec (6.6.00)

That is an interesting bug. As I found out by using the Maple debugger, this bug seems to be unavoidable if the procedure limit() and the procedures called by it cannot determine whether a trig expression is zero or not. Look at the following short Maple session:

```
> restart;
> expr:=x+(-cos(9/50*Pi)+sin(8/25*Pi))/h;
```

```
                              -cos(9/50 Pi) + sin(8/25 Pi)
                  expr := x + ----------------------------
                                            h

> limit(expr,h=0);

                                        0
```

This is certainly wrong as the result should be x. The procedure limit() cannot verify that `-cos(9/50 Pi) + sin(8/25 Pi)` equals zero. However, testeq() can:

```
> zero_trig_expr:=-cos(9/50*Pi)+sin(8/25*Pi);

              zero_trig_expr := -cos(9/50 Pi) + sin(8/25 Pi)

> testeq(zero_trig_expr=0);

                                      true
```

Thus, as a first bugfix, limit() and the procedures called by it should make use of testeq(), too.

7.100 bug in LinearAlgebra,GramSchmidt, Maple 6 to Maple 8 (13.9.02)

7.100.1 Jack-Michel CORNIL

When working on an exercise for my students in order to reduce a symmetric matrix with an orthogonal transition matrix, I fell on the following error :

```
> with(LinearAlgebra):

> A:=Matrix(3,3,[[2,-2,-2],[2,-2],[2]],scan=triangular,shape=symmetric);

> T:=Eigenvectors(A);

> T2:=T[2];
> B:=[Column(T2,1..3)];

> U:=GramSchmidt(B,'normalized');
Error, (in LinearAlgebra:-LA_Main:-Normalize) datatype of in-place Vector
(algebraic) and datatype of result (anything) do not agree
```

It is the option 'normalized' which causes the trouble.

I think that the problem comes from the "datatype" of the elements of the resulting matrix T2 as we can see below but it is impossible to modify this option.

```
> VectorOptions(B[1]);

  shape = [], datatype = algebraic, orientation = column,

        storage = rectangular, order = Fortran_order

> MatrixOptions(T2);

  shape = [], datatype = algebraic, storage = rectangular,

        order = Fortran_order
```

7.101 bug in LinearSolve, Maple 7 to Maple 8 (24.10.02)

7.101.1 Luis Goddyn

LinearSolve appears to be broken in both Maple 7 and Maple 8, and across several platforms.

The problem appears occationally when the matrix has a real entry and the solution is not unique.

Is this problem known? Does anyone have a fix or a workaround? (I must solve a 20 by 40 system, and I am neither amused nor impressed!)

Two small examples of the LinearSolve bug

```
> with(LinearAlgebra):
> <<1,1,0>|<1,1,0>|<0,0,1>>,<1,1,0>; LinearSolve(%); # Works correctly
                              [1    1    0]  [1]
                              [           ]  [ ]
                              [1    1    0], [1]
                              [           ]  [ ]
                              [0    0    1]  [0]

                                 [1 - _t0[2]]
                                 [           ]
                                 [  _t0[2]   ]
                                 [           ]
                                 [    0      ]

> <<1,1,0>|<1,1,0>|<0,0,1.>>,<1,1,0>: LinearSolve(%); # Make one entry real
 Error, (in LinearAlgebra:-LA_Main:-BackwardSubstitute)
 Matrix must be in row-echelon form, zero rows at the bottom

> <<1.,1.,0.>|<0.,0.,0.>|<0.,0.,1.>>,<1.,1.,0.>: LinearSolve(%); # Example 2
 Error, (in LinearAlgebra:-LA_Main:-BackwardSubstitute)
 Matrix must be in row-echelon form, zero rows at the bottom

> interface(version);
     TTY Iris, Maple 8.00, SGI MIPS UNIX, Apr 22 2002 Build ID 110847
```

7.101.2 Robert Israel (24.10.02)

It seems to work with

```
> LinearSolve(%,method='solve',outputoptions=[datatype=anything]);
```

(in this example, anyway).

7.101.3 Denis Sevee (27.10.02)

Yes, it's annoying. If you haven't noticed yet, the same problem turns up with GaussianElimination and LUDecomposition, i.e. Maple doesn't move the rows of zeroes to the bottom.

7.102 bug in Linsolve in Maple V.4 (7.8.97)

7.102.1 W. Edwin Clark

Be careful with Linsolve mod 2. Here's an example of what can happen. (with Maple VR4 on both Mac and Solaris)

[By the way can someone tell me how to look at the code for the procedure Linsolve mod p. The obvious method doesn't work, ie, using `interface(verboseproc=2); eval(Linsolve);`

```
> with(linalg):
> A:=matrix([[0,0,0,1,0,1],[1,1,1,0,0,0],[0,0,1,1,1,0],[0,0,0,1,1,1]]);

                      [0    0    0    1    0    1]
                      [                          ]
                      [1    1    1    0    0    0]
                A := [                           ]
                      [0    0    1    1    1    0]
                      [                          ]
                      [0    0    0    1    1    1]

> b:=vector([1,1,1,1]);

                         b := [1, 1, 1, 1]

> x:=Linsolve(A,b) mod 2;

        x := [_t[2] + _t[6], _t[2], _t[6], _t[6], 1, _t[6]]

> map(`mod`,evalm(A &* x - b),2);

                            [1, 1, 0, 0]
```

Clearly this should be zero.

Another problem is that if A has more rows than columns `Linsolve(A,b) mod 2` returns an error message.

The bug is removed with Maple V Release 5. (U. Klein)

Here's a work-around for the first problem. For the second one can apply Gausselim

mod 2 to the augmented matrix and then delete the zero rows.

```
> B:=augment(A,transpose(matrix([[1,1,1,1]])));

                    [0    0    0    1    0    1    1]
                    [                               ]
                    [1    1    1    0    0    0    1]
               B := [                               ]
                    [0    0    1    1    1    0    1]
                    [                               ]
                    [0    0    0    1    1    1    1]

> Gausselim(B) mod 2;

                  [1    1    1    0    0    0    1]
                  [                               ]
                  [0    0    1    1    1    0    1]
                  [                               ]
                  [0    0    0    1    0    1    1]
                  [                               ]
                  [0    0    0    0    1    0    0]

> x:=linsolve(delcols(B,7..7),col(B,7));

     x := [1 - _t[1] - _t[2], _t[1], _t[2], 1 - _t[2], 0, _t[2]]

> map(`mod`,evalm(A &* x - cb),2);

                             [0, 0, 0, 0]
```

7.102.2 Robert Israel (13.8.97)

The bug is not just in mod 2, but for any modulus when the matrix A is not square. For example, with Edwin's example,

```
> A:=matrix([[0,0,0,1,0,1],[1,1,1,0,0,0],[0,0,1,1,1,0],[0,0,0,1,1,1]]);
> b:=vector([1,1,1,1]);
> x:= Linsolve(A,b) mod 10;
        x := [9 _t[2] + 9 _t[6], _t[2], _t[6], 9 _t[6], 1, _t[6]]
```

```
> map(`mod`,evalm(A &* x - b),10);
                              [9, 9, 0, 0]
```

| [By the way can someone tell me how to look at the code ...

Try this:

```
> interface(verboseproc=2);
> readlib(`mod/Linsolve`);
```

The bug appears to be near the end of the procedure, in the line

```
x[j] := Normal((B[i, m + 1] - t)*s) mod p
```

which should be

```
x[j] := Normal((B[i, n + 1] - t)*s) mod p
```

After making this change in the procedure, it seems to work correctly.

```
> x:= Linsolve(A,b) mod 10;
      x := [1 + 9 _t[2] + 9 _t[6], _t[2], _t[6], 1 + 9 _t[6], 0, _t[6]]

> map(`mod`,evalm(A &* x - b),10);
                  [0, 0, 0, 0]
```

7.103 bug in Map, Maple 6 to Maple 8 (21.10.02)

7.103.1 Jean Brillet

It seems there is a small bug in the function Map of the LinearAlgebra package ("inplace" map for Matrix and Vector).

Let A be a Matrix with numerical entries and with `"datatype=anything"`.

Then `Map(evalf[n],A)` applies `evalf[Digits]` whatever is n. The same occurs for `Map(x->evalf[n](x),A)`. `Map(evalf,A,n)` does nothing. However `Map(x->evalf(x,n),A)` works fine.

7.103.2 Carl Devore (23.10.02)

The problem with your final solution, `Map(x->evalf(x,n), A)`, is that it generates 1 extra high-level procedure call per represented entry of A.

It is hard to imagine a situation where this would be worth the extra time compared to the extra memory that is required by regular map.

7.104 Bug in Maple 6.02a PostScript Win 2k Pro (3.5.02)

7.104.1 Jeff Wright

In case you are suffering with getting a workable postscript picture out of Maple 6.02a (on win2k pro, at least), your ps.dll file is bad.

Replacing the ps.dll file with the same-named file from Maple 6.01 seems to solve the problem.

Another way to solve the problem is not to include titles in your plots. (Of course all my plots are so clear and intuitive as to render titles completely unnecessary - NOT!)

7.105 bug in Maple V.5 affecting Newton in share lib. (4.3.98)

7.105.1 Ross Taylor

The R5 version of Newton, an implementation of Newton's method for systems of nonlinear equations, does not work.

As far as I can determine this is due to a bug in linalg[norm] which no longer is able to compute '1' norms (even though the help page says it can).

As a work around until linalg[norm] is fixed you can edit Newton.mpl in `/share/newton`. There are two lines of code that need to be changed. The lines that contain the command:

```
norm(something,1)
```

should be changed to read

```
norm(something,2)
```

This change in the type of norm used in the calculation does not have any real effect on the way Newton behaves.

If you do carry out this edit you may find (as I did) that with(share) etc. no longer is able to load `Newton.mpl`.

In that case you will need to load Newton using a standard read command.

My thanks to George Luna for telling me of this problem and I hope that this workaround helps other Newton users out there.

The bug is removed with the Library Update of Maple V.5. (U. Klein)

7.105.2 Joe Riel (5.3.98)

Concerning Ross's fix to his Newton package in the share library. To get everything properly updated, you should do the following.

1. Open the Newton.mpl file and modify per Ross's instructions.
2. Go to the bottom of the file and remove the # sign in front of the savelib statement.
3. Add the line savelibname `:= sharename:` just above the savelib statement.

4. It's a good idea to put a comment statement in Newton.mpl indicating that you've changed it.

5. While you're at it, you might want to open the Newton.msl file and correct the typo in the `"Description = ... " statement;` there should be a space just before the backslash at the end of the line.

6. Save the files and open a Maple worksheet. Execute the following commands,

```
restart:
with(share):
currentdir("".sharename."/Newton"):
read "Newton.mpl":
```

That should do it. Restart maple and load the package using

```
with(share):
with(Newton):
```

You might consider write protecting the share libary (along with your main maple library) so you that cannot accidentally write to it.

7.105.3 Robert Israel (5.3.98)

A temporary fix for the '1' norm problem is to set `_Envsignum0:= 1`.

Of course this won't work very well if you want to use some other setting of `_Envsignum0` (i.e. you want `signum(0)` to return something other than 1). But you can do the following:

```
> with(linalg):
> oldnorm:= eval(norm):
> unprotect(norm):
> norm:= proc() _Envsignum0 := 1; oldnorm(args) end:
> protect(norm):
```

And now

```
> norm([1,2,3],1);

                        6
```

and this won't affect signum outside of norm:

```
> signum(0);

        0
```

7.106 Bug in matrix of power n, expand in Maple 4 to Maple 8 and simplify in Maple 4 to Maple 6 (5.2.01)

7.106.1 HARALD PLEYM

Can anyone explain this strange behaviour, both in R4 / R5 and Maple 6.

```
>with(linalg):
>A:=diag(1,2,3)^n:
> expand(A);
                                        n
                                     bb
```

The same result with Maple 7 and Maple 8. Corrected with Maple 9. (U. Klein)

```
> simplify(A);
                                       n
                                    b
```

This is corrected with Maple 7. (U. Klein)

I expected only the matrix A as output.

7.106.2 Dr. TANAKA, Kazuo (13.2.01)

This is not the answer with which you are satisfied. This is my comments.

A command "expand" is a reverse of "factor". Therefore, I think that we should not apply "expand" to matrix, vector and so on.

A command "simplify" is a simplification rule for various types of expressions, which, I am afraid, do not include matrix, vector nor,....

Please try as followings;

```
>with(linalg):
>A:=diag(1,2,3);
>evalm(A^4);
```

7.106.3 Colin Birch (13.2.01)

I cannot explain where bb comes from, but if A is a matrix, in R4 / R5 the expressions `A^n, A*A or A+A` are syntax errors, which won't do what you want. You have to use multiply() and add(). In Maple 6 you can use the LinearAlgebra package instead:

```
> with(LinearAlgebra): A:=DiagonalMatrix([1,2,3])^2: expand(A);

                              [1    0    0]
                              [           ]
                              [0    4    0]
                              [           ]
                              [0    0    9]
```

but I am afraid you still get:

```
> with(LinearAlgebra): A:=DiagonalMatrix([1,2,3])^n: expand(A);

                              [1    0    0]n
                              [           ]
                              [0    2    0]
                              [           ]
                              [0    0    3]
```

My impression is that linalg is only being maintained for the sake of compatibility between releases.

7.106.4 Renato Portugal (13.2.01)

This is a bug!

Look at the following commads, which are independent on linalg or package LinearAlgebra of Maple 6:

```
> Z := array(1 .. 3, 1 .. 3,[(1, 1)=1,(3, 2)=0,(2, 1)=0,(2, 2)=2,(3, 3)=3,
            (1, 2)=0,(2, 3)=0,(1, 3)=0,(3, 1)=0])^m;

> simplify(%);

                                    m
```

```
                                    b
> expand(%%);

                                     m
                                   bb
```

The bug seems to be in procedure

‘simplify/commonpow‘ and ‘expand/power‘

which have b and bb as local variables respectively. It seems that it is a problem of evaluation to name of these local variables.

7.107 bug in matrixplot coloring, Maple 4 to Maple 8 (10.3.00)

7.107.1 Thom Mulders

I want to do a matrixplot and let the color of the bars depend on the value of the matrix entry that is plotted.

I would expect to do this as follows:

```
A:=linalg[matrix]([[1,2,3],[1,5,6]]);
col:=proc(x,y) A[x,y]/7 end;
with(plots):
matrixplot(A,heights=histogram,axes=frame,style=patch,color=col);
```

The graph I get gives the same color for the bars corresponding to the entries `(1,1) and (2,2)` but not for the bars corresponding to the entries `(1,1) and (2,1)` (what I expect).

7.107.2 Carl Devore(25.4.02)

The order that the colors are generated for the histogram plot is transposed from the order that the bars are generated. In Maple 7, you can see this in lines 59 and 67 of procedure `‘plots/matrixplot‘`.

In earlier versions of Maple, the procedure is named `‘plots/matplot‘`. In Maple `5 & 6`, the relevant lines are `39 & 47`. In Maple 4, lines `38 & 46`.

Since this bug would have been so trivial to fix, and you wrote the above over two years ago, it seems that no-one at WMI who is in a position to do something about these bugs is actually doing anything.

7.108 Bug in maximize with trig functions in Maple V.3 to Maple V.5 (6.2.97)

7.108.1 Jay Bourland

There seems to be a bug in minimize/maximize for the trig functions:

```
> maximize(sin(x),x,0..1);

                                  1

> evalf(sin(1));

                              .8414709848

> minimize(sin(x),x,-1..0);
                                -sin(1)
```

It is corrected with Maple 6. (U. Klein)

7.108.2 David L. Johnson

MO, the maximize() function from release 3 (which I guess might be what you are using here) is not very good.

Here is another example:

```
> maximize(sin(x)+cos(x),x,0..0.2);

                                      1/2
                                    2
```

I don't, however, consider that to be too much of a problem, since you can do all the steps to really find the maximum easily and reliably.

I try not to let students know about the maximize() function, since I want them to understand the process. That and its unreliable performance, that is.

7.108.3 Robert Israel

Don't use "maximize" and "minimize"! They are unreliable at best and misleading at worst.

It's not so much the bugs as the unrealistic expectations. "minimize" and "maximize" work in basically the same way as a calculus student would: look at the function at its critical points, singular points and endpoints, and take the minimum or maximum value. But critical points can only be found by solving the equation $f'(x) = 0$. Unfortunately, except for rational functions "solve" can't be relied on to find all the solutions: the most it will usually do is to find one. The result is that "maximize" and "minimize" will give wrong answers if the actual max or min is at a different critical point.

I realize that the bug in this case is a bit different: Maple knows that $\sin(x)$ has a critical point at `x=Pi/2`, but maybe somehow it doesn't realize that `Pi/2` is not in the interval 0 to 1.

7.109 Bug in multilinear in Maple V.5 (25.5.99)

7.109.1 Tullia Dymarz

Why doesn't this work...

```
> define(h,multilinear);
> h(h(2*a,b),c);
Error, (in h) too many levels of recursion
```

while this works...

```
> define(n,linear);
> n(n(2*a,b),c);

                              2 n(n(a, b), c)
```

It is corrected with Maple 6. (U. Klein)

7.109.2 Robert Israel

It's a bug. The programmers evidently didn't expect that the arguments to h might contain h. The result is that whenever x or y contains h, h(x,y) attempts to evaluate itself, resulting in an infinite recursion.

The "linear" code doesn't contain this bug, but it has another one:

```
> n(a*b);
 Error, (in n) too many levels of recursion
```

Here is a way to patch h so that it will work:

```
h := readlib(procmake)(subs('`&function`(op,`&expseq`(`&local`[2]))' =
 '`&function`(op,`&expseq`(`&function`(subs,`&expseq`(_FUNCNAME =
 `&args`[-2], `&local`[2])))))' , readlib(procbody)(h)));
```

If h is a binary function, you could also define it to be multilinear "by hand" as follows:

```
> define(h,
  h(a::nonunit(algebraic)+b::nonunit(algebraic),c::anything) = h(a,c)+h(b,c),
```

```
  h(a::anything,b::nonunit(algebraic)+c::nonunit(algebraic)) = h(a,b)+h(a,c),
  h(a::nonunit(constant)*b::nonunit(algebraic),c::anything) = a*h(b,c),
  h(a::anything,b::nonunit(constant)*c::nonunit(algebraic)) = b*h(a,c));
```

7.109.3 Andre Heck (9.6.99)

| From: Robert Israel <israel@math.ubc.ca>

As the author of the original code for defining multilinear mappings (1991), I can only confirm that it is a bug and that it was introduced later (1996) by WMI programmers. The original code was still present in Maple V Version Release 4:

```
    |\^/|     Maple V Release 4 (University of Amsterdam)
._|\|   |/|_. Copyright (c) 1981-1996 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> define(multilinear(h));
> h(h(2*a,b),c);
                                2 h(h(a, b), c)
```

Interested in the old code?

```
> interface(verboseproc=3):
> print(`define/multilinear`);
proc(OpName)
local t;
global _X;
option
`Copyright (c) 1991 by the University of Waterloo. All rights reserved.`;
    if nargs <> 1 or not type(OpName, 'string') then
        ERROR(`invalid arguments`)
    fi;
    if assigned(_X) then _X := '_X' fi;
    t := proc()
        local func, k, term, terms, cf, coefficient;
        option remember, `Copyright (c) 1991 by the University of Waterl\
        oo. All rights reserved.`;
```

```
if nargs < 1 then ERROR(`no element`)
elif nargs = 1 then
    if type(args[1], 'constant') then RETURN(args[1]*'_X(1)')
    elif type(args[1], `+`) then RETURN(map(_X, args[1]))
    elif type(args[1], `*`) and hastype(args[1], 'constant')
    then
        cf := select(type, args[1], 'constant');
        term := args[1]/cf;
        RETURN(cf*_X(term))
    fi
elif 2 <= nargs then
    if type(args[1], `+`) then
        RETURN(map(_X, args[1 .. nargs]))
    else for k from 2 to nargs do
            if type(args[k], `+`) then
                func := subs({_KK = k, 'F' = _X}, proc()
                        F(args[2 .. _KK], args[1],
                        args[_KK + 1 .. nargs])
                    end);
                RETURN(map(func, args[k], args[1 .. k - 1],
                    args[k + 1 .. nargs]))
            fi
        od
    fi;
    terms := NULL;
    coefficient := 1;
    for k to nargs do
        if type(args[k], 'constant') and args[k] <> 1 then
            coefficient := coefficient*args[k]; term := 1
        elif
        type(args[k], `*`) and hastype(args[k], 'constant')
        then
            cf := select(type, args[k], 'constant');
            coefficient := coefficient*cf;
            term := args[k]/cf
        else term := args[k]
        fi;
        terms := terms, term
    od;
```

```
                if coefficient <> 1 then RETURN(coefficient*_X(terms)) fi
            fi;
            RETURN('_X(args)')
        end;
    if member(OpName, {'func', 'k', 'term', 'terms', 'cf', 'coefficient'})
    then t := subs(OpName = 'b', eval(t))
    fi;
    if type(OpName, protected) then
        unprotect('OpName'); lprint(`Warning: new definition for`, OpName)
    fi;
    OpName := subs(_X = OpName, eval(t));
    NULL
end
```

This piece of code will work if you change the name of the procedure, say to ‘mlinear‘, and change the first type checking into `if nargs <> 1 or not type(OpName, 'symbol')`

```
    |\^/|     Maple V Release 5 (WMI Campus Wide License)
._|\|   |/|_. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> read mlinear;
> mlinear(h);
> h(h(2*a,b),c);
                                2 h(h(a, b), c)
```

Note that I have not tested whether other problems arise with this code in Maple V Release 5.

7.110 bug in networks[connectivity], from Maple 6 to Maple 8 (19.11.01)

7.110.1 Edwin Clark

Apparently a bug was introduced when going from Maple V r5.1 to Maple 6 and retained in Maple 7 in networks[connectivity]

Here's a simple example:

Consider the graph G defined below which clearly has connectivity (that is, edge-connectivity) 1:

```
> interface(version);

  Maple Worksheet Interface, Maple 7.00, APPLE_PPC_MAC, Tue, May 2\
        9, 2001 Build ID 96223

> restart:
> with(networks):
> V:={1,2,3,4,5, 6}:
> E:={{1,2},{2,3}, {3,1}, {3,4}, {4,5}, {5,6}, {4,6}}:
> G:=graph(V,E):
>
> connectivity(G, 1, s1), s1;

                          2, {{5, 6}, {4, 5}}
```

On the other hand in Maple V r5.1 we get the correct answer:

```
> interface(version);

  Maple Worksheet Interface, Release 5.1, APPLE_PPC_MAC, Aug 20 19\
        98

> restart:
> with(networks):
> V:={1,2,3,4,5, 6}:
> E:={{1,2},{2,3}, {3,1}, {3,4}, {4,5}, {5,6}, {4,6}}:
> G:=graph(V,E):
```

```
>
> connectivity(G, 1, s1); s1;

                                    1

                                {{3, 4}}
```

The bug retained in Maple 8, too (U. Klein)

7.111 Bug in nullspace, Maple V.5 (9.11.98)

7.111.1 Luis Goddyn

Looks like the nullspace (kernel) of a matrix having a float entry fails. How do I fix/work around? I need it!

```
> interface(version);
               TTY Iris, Release 5, SGI MIPS UNIX, Jan 11 1998

> with(linalg):

> nullspace( matrix(1,2, [ 1 , 3/2 ] ) );
                                  {[-3/2, 1]}

> nullspace( matrix(1,2, [ 1 , 1.5 ] ) );
                                       {}
```

It is corrected with Maple 6. (U. Klein)

7.111.2 Robert Israel (11.11.98)

Nullspace appears to work for square matrices, but not for non-square ones. So a work-around would be to use the following instead of nullspace(A):

```
> AA:= evalm( htranspose(A) &* A);
> nullspace(AA);
```

7.111.3 Denis Sevee (11.11.98)

(1) You can try (for your example)

```
    A:=matrix(1,2,[1,1.5]):
    linsolve(A,[0]);
```

But then you have to factor out the parameters to get the basis.

(2) You can multiply your matrix by a power of 10 to eliminate the decimals. This has no effect on the nullspace.

(3) If rref gives `[ I F ]`, then the basis for the nullspace is the columns of

```
[ -F ]
[  I ]
```

But if rref doesn't give this form you have to some column swapping.

(4) (The best?) You can do `evalf(Svd(A,U,V))`. The columns of V corresponding to the zero singular values (i.e., the last n columns of V where n is the nullity of `A, n=# of columns` of `A - rank(A)`, will be a basis for Nul A. An orthonormal basis, in fact.

7.111.4 Wilhelm Werner (12.11.98)

Maybe this fixes your problem:

```
> with(linalg):
> map(convert,matrix(1,2,[1,1.5]),rational);

                              [1    3/2]

> nullspace(%);

                             {[-3/2, 1]}
```

7.111.5 Helmut Kahovec (12.11.98)

On a PC with NT 4.0 Release 4 did it right, but Release 5 did not.

7.111.6 Jim Gunson (12.11.98)

The nullspace of a square matrix (for example) contains only the nullvector, unless the matrix can be reduce by Gaussian elimination to a matrix with a row/rows of zeros.

Using floats to approximate the entries will make this unlikely. What tends to happen is that you get very small numbers in the reduced matrix, which should be zero. (Perhaps you know this already.) This same problem occurs when finding eigenvalues.

I ran accross this in writing a linalg text for my students here at Kwantlen.

My solution is to find the reduced matrix and set to zero any elements at the ends

of the lower rows that are extremely small (eg `10^-18`). This can be done manually, by inspection or by applying a `"filter"` that zeros any element smaller than some amount.

The result should be an close approximation to the null-space. Hopefully this will do the job for you.

The following proc. does this:

```
> newnullspace:=proc(m,threshold)local filter,ht,augmat,redmat;

>  filter:=(x,t)-> if abs(x)<t then 0 else x;fi;

>   ht:=linalg[coldim](m);
>   augmat:=linalg[augment](m,vector(ht,0));
>   redmat:=linalg[gausselim](augmat);
>   redmat:=map(filter,evalm(redmat),threshold);
>   RETURN(linalg[backsub](redmat));
> end:

> linalg[nullspace](matrix(3,3,[1,2,4,2,3,5,6,6,6.001]));

                                {}

> newnullspace(matrix(3,3,[1,2,4,2,3,5,6,6,6.001]),0.01);

            [1.999500000 _t[1], -2.999666667 _t[1], _t[1]]
```

7.111.7 Joe Riel (12.11.98)

The problem lies in the procedure ‘`linalg/kernel/float`‘. It computes the nullity of a matrix by counting the number of singular values that are 0. Alas, if the matrix has less rows than columns, this count does not indicate the nullity.

One workaround is to make the matrix square by stacking it onto a zero matrix,

```
> M := matrix(1,2,[1,1.5]):
> M0 := stackmatrix(M,[0$2]):
> kernel(M0);
                    {vector([-.8320502943, .5547001962])}
```

A second is to convert the entries to rational values; however, this may be undesireable for performance reasons.

A third is to correct the offending procedure. Following is what I believe to be a corrected version [no guarantees].

```
`linalg/kernel/float` := proc(A, nullity)
local K, p, m, n, S, U, V;
option `Copyright (c) 1996 Waterloo Maple Inc. All rights reserved.`;
    m := linalg[rowdim](A);
    n := linalg[coldim](A);
    S := evalf(Svd(A, 'U', 'V'));

# Determine the position, p, of the first zero in S.
# Because the values of S are in decreasing order,
# the remaining terms must also be zero.
# If S has no zeros then p exits the loop assigned min(m,n)+1.
# Columns p to n of V form a basis for the kernel of A.

    for p to min(m,n) while
        S[p] <> 0
        and 10^(Digits - 1)*abs(S[p]) >= abs(S[1])
        do od;
    if nargs > 1 then nullity := n-p+1 fi;
    K := {linalg['col'](V, p .. n)};
    if type(A, 'array'(2)) then K else map(convert, K, list) fi
end:
```

Comment: The loop that counts the number of singular values that are 0 assumes that the singular values in S are arranged in decreasing order. This seems to be true, but the help page for Svd does not mention this fact. Presumably that is how linpack does it...

```
> kernel(M);
                    {vector([-.8320502943, .5547001962])}


> evalm(M &* %[]);
                           vector([0])
```

If you want to test this procedure, you should execute `readlib('linalg/kernel')` before assigning it, otherwise it will be replaced by the old version.

To install this procedure into your own library use

```
save `linalg/kernel`, `linalg/kernel/float`,
     ``.yourlibrary.`/linalg/kernel.m`:
```

where yourlibrary expands to the path of your personal library (not the main maple library).

If it precedes the main maple library in your libname assignment, then subsequent invocations of kernel will work properly [assuming my patch is correct].

7.111.8 Michael Monagan (13.11.98)

Sorry, my mistake. I put in place a code to do an SVD decomposition of the matrix but made an simple error. A workaround is to pad the matrix with enough zero rows to make it square. I.e.

```
> nullspace( matrix( [[1,1.5],[0,0]] ) );
                          {[-.8320502943, .5547001962]}
```

7.111.9 Adri van der Meer (13.11.98)

It seems that the problem only occurs in the case of 1 x n -matrices. linsolve works well, but returns the nullspace in parametric form (with _t[i]'s). To convert this to a base you could use (general case):

```
> fnullspace := proc(B::matrix)
>   local i,r,q;
>   q :=linsolve(B,vector(rowdim(B),[0$rowdim(B)]));
>   r := `union`(seq(indets(q[i]),i=1..vectdim(q)));
>   {seq(map(coeff,q,r[i]),i=1..nops(r))}
> end:

> A := matrix(1,2,[1,1.5]):
> fnullspace(A);
```

```
                          {[1, -.6666666667]}

> B := matrix(3,3,[[1,1.5,1],[1,1.5,1],[1,1.5,1]]):
> fnullspace(B);

              {[1, -.6666666667, 0], [0, -.6666666667, 1]}
```

7.111.10 Bob Wright (15.11.98)

This is in response to a recent demonstration of a bug in linalg. I've lost the original so can't reply directly to it. The problem is caused by incorrect calculation of the nullity (N) in `'linalg/kernel/float'`, introduced in release 5.

Release 4 doesn't make this error. Use of the SVD is a good idea, but please developers, lets do the implementation right! Below is an illustration of the problem, a rerun using a fixed version of `'linalg/kernel/float'`, and the fixed version of `'linalg/kernel/float'`.

```
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
> A:= matrix(1,2,[1,-1.5]);

                           A := [1    -1.5]

> kernel(A);

                                  {}

> B:= matrix(2,3,[[1,-1.5,0],[1,-1.5,0]]);

                               [1    -1.5    0]
                          B := [               ]
                               [1    -1.5    0]

> kernel(B);

                               {[0, 0, 1.]}
```

```
> read `patch.lna`; #new `linalg/kernel/float`
> kernel(A);

                    {[-.8320502943, -.5547001962]}

> kernel(B);

              {[0, 0, 1.], [-.8320502943, -.5547001962, 0]}
```

New calculation of N (nullity) and new use of it in new ‘linalg/kernel/float‘.

```
> eval(`linalg/kernel/float`);

  proc(A, nullity)
  local K, N, i, m, n, S, U, V;
    m := linalg[rowdim](A);
    n := linalg[coldim](A);
    S := evalf(Svd(A, 'U', 'V'));
    N := max(n-m,0); #Note new calc of N
    for i to min(m, n) do
        if S[i] = 0 or 10^(Digits - 1)*abs(S[i]) < abs(S[1])
        then N := N + min(m, n) - i + 1; break #Note different use of N
        fi
    od;
    if nargs = 2 then nullity := N fi;
    K := {linalg['col'](V, n + 1 - N .. n)};
    if type(A, 'array'(2)) then K
    else map(convert, K, list)
    fi
  end
```

7.111.11 Federico Rocchi (12.11.98)

An analogous problem seems to occur with version 4.0:

```
with(linalg):

kernel(matrix(1,2,[1,3/2]));

                              {[-3/2, 1]}

kernel(matrix(1,2,[1,1.5]));

                           {[1, -.6666666667]}
```

Does anyone know how to solve/fix it even for 4.0 version?

7.111.12 Robert Israel (18.11.98)

This is neither analogous nor a problem. It is simply correct (the basis vector in the second case being a multiple of that of the first case).

On the other hand, Release 4 does have numerical problems with `"kernel"` for matrices with floats. For example, about half the time it returns for

```
> kernel(matrix([[3.,4,5],[6,7,8],[9,8,7]]));
```

where a correct result would be `{[1, -2, 1]}`.

The problem is that Release 4 uses essentially the method we're taught in elementary linear algebra courses: solve the system `A x = 0` by Gaussian elimination (using linsolve). Unfortunately, rounding error can get you into trouble when A is singular. In the case of `A = matrix([[3.,4,5],[6,7,8],[9,8,7]])`, Gaussian elimination is likely to return a result such as

```
[ 9       8          7        ]
[ 0  1.666666666 3.333333333  ]
[ 0       0          x        ]
```

where x is nearly, but (because of rounding error) not quite, 0.

The much better method which Release 5 uses (but doesn't quite get right for matrices

with more columns than rows) is to use the singular value decomposition as suggested by Denis Sevee: do `evalf(Svd(A,U,V))`, and then if A has n columns and there are r nonzero singular values (where values close enough to 0 to be probable results of rounding error are considered as 0), take the last `n-r` columns of V as a basis of kernel(A).

The bug in Release 5 is that it takes the last `min(m,n)-r` columns rather than the last `n-r` columns.

Another work-around for this bug is to use "stackmatrix" to add enough rows of 0's to make a square matrix. You could define:

```
> newkernel:= proc(M)
      local m,n;
      m:= linalg[rowdim](M);
      n:= linalg[coldim](M);
      if m >= n then linalg[kernel](args)
      else linalg[kernel](linalg[stackmatrix](M, matrix(n-m,n,0)),args[2..-1])
      fi
   end;
```

7.111.13 Tom Holly (20.11.98)

| On Wed, 18 Nov 1998, Robert Israel wrote: ...

This is very interesting. In release 4, I can get any of the following results:

```
> kernel(matrix([[3.,4,5],[6,7,8],[9,8,7]]));
                      {[-.5000000000, 1, -.5000000000]}

> kernel(matrix([[3.,4,5],[6,7,8],[9,8,7]]));
                                      {}

> kernel(matrix([[3.,4,5],[6,7,8],[9,8,7]]));
                      {[-.4999999999, 1, -.5000000001]}

> kernel(matrix([[3.,4,5],[6,7,8],[9,8,7]]));
                     {}
```

Release 5 consistently gives the result:

```
> kernel(matrix([[3.,4,5],[6,7,8],[9,8,7]]));

                {[.4082482905, -.8164965809, .4082482905]}
```

7.112 bug in numeric dsolve in Maple 7 (7.9.01)

7.112.1 Douglas B. Meade

Matt Miller and I have recently encountered the same bug with numeric solutions of initial value problems in Maple 7. This bug has been reported to Maple Support; the incident number is 347182. I am posting this message on MUG as an alert to other Maple users. The bug might appear to be rather specific, but the two of us discovered the same bug within the same 24 hour period. (To be honest, Matt showed me the problem he was having, I diagnosed the problem, then encountered the same situation in the process of updating one of my worksheets later in the day.)

The importance of this error is that it occurs in situations where users are often lazy about checking Maple's output. An unsuspecting user is not likely to discover this problem without much effort.

The problem seems to arise when the procedure returned by `dsolve,numeric` is evaluated in the same command in which it is defined. Here is a simple example to illustrate the problem:

```
> restart;
> interface( version );

  Maple Worksheet Interface, Maple 7.00, SUN SPARC SOLARIS, May 28\
         2001 Build ID 96223

> ode := diff( y(t), t ) = y(t);

                              d
                      ode := -- y(t) = y(t)
                              dt

> ic1 := y(0) = 1;
> ic2 := y(0) = 2;

                          ic1 := y(0) = 1

                          ic2 := y(0) = 2

> dsolve( { ode, ic1 }, numeric )(1);   # exact value should be exp(1)
```

```
                    [t = 1., y(t) = 2.71828131105043]

> dsolve( { ode, ic2 }, numeric )(1);   # exact value should be 2*exp(1)

                    [t = 1., y(t) = 2.71828131105043]

> sol1 := dsolve( { ode, ic1 }, numeric ):
> sol2 := dsolve( { ode, ic2 }, numeric ):

> sol1(1);

                   [t = 1., y(t) = 2.71828131105041937]

> sol2(1);

                   [t = 1., y(t) = 5.43656262948751578]
```

Another workaround, suggested by Brian Fox in Maple's Technical Support, is to include `output=listprocedure` as an additional argument to dsolve:

```
> dsolve ({ode,ic1}, y(t), numeric, output=listprocedure)(1);

                [t(1) = 1, y(t)(1) = 2.71828131105041937]

> dsolve ({ode,ic2}, y(t), numeric, output=listprocedure)(1);

                [t(1) = 1, y(t)(1) = 5.43656262948751490]
```

Note, however, the unfortunate appearance of the left-hand sides of the equations. The use of this output would require modifications to nonstandard notation in subsequent eval or subs commands.

I have not attempted to uncover the full extent or precise source of this problem. Is the bug-generating syntax used in other Maple commands or packages? I suspect it has something to do with the remember table, but don't have the time to track this down. (Hint, hint!)

You've been warned! `(:-))`

It is corrected with Maple 8 (U. Klein)

7.112.2 Carl DeVore (17.9.01)

I haven't looked at your specific problem very closely, but I'll bet that it is related to this remember table bug that I discovered last week in Maple 6. Under certain circumstances, a returned local procedure has a persistent remember table.

```
> restart;
> kernelopts(version);
      Maple 6.01, SUN SPARC SOLARIS, June 9 2000 Build ID 79514
> interface(verboseproc=3);
> P:= proc(ic)
>    local f;
>    f:= proc() 0 end;
>    f(ic[1]):= f(ic[1])+ic[2];
>    f
> end proc:
> P([0,1])(0);
                                  1
> P([0,1])(0);
                                  2
```

Unlike your example, in this case the bug still occurs if the returned procedure is assigned (why the difference?).

```
> g:= P([2,2]):
> g(2);
                                  2
> h:= P([2,2]):
> h(2);
                                  4
> g:= P([2,2]):
> g(2);
                                  6
```

Clear the table.

```
> g:= subsop(4= NULL, op(g));
```

```
                          g := proc() 0 end proc
> g:= P([2,2]);
                                  g := f
> g(2);
                                    2
> print(h);
                            proc() 0 end proc # (2) = 2
```

h has acquired a remember table.

```
> g:= subsop(4= NULL, op(g));
                          g := proc() 0 end proc
> print(h);
                            proc() 0 end proc
```

h has lost its table.

```
> addressof(eval(g));
                                  3339372
> addressof(eval(h));
                                  3339372
```

h and g are really the same thing.

This latter fact is true even if there is no remember table.

```
> restart;
> P:= proc() proc() 0 end end:
> g:= P();
> h:= P();
> addressof(eval(g));
                                  3521784
> addressof(eval(h));
                                  3521784
```

The bug can be corrected like this:

```
> P:= proc(ic)
>    local f;
>    f:= proc() 0 end;
```

```
>    f:= subsop(4= NULL, eval(f)); #Clear table
>    f(ic[1]):= f(ic[1])+ic[2];
>    f
> end:

> P([1,1])(1);
                                   1
> P([1,1])(1);
                                   1
```

But this attempted correction will not work. (Why?)

```
> P:= proc(ic)
>    local f;
>    f:= proc() option remember, system; 0 end;
>    gc(); #Attempt to remove tables from "system" procedures
>    f(ic[1]):= f(ic[1])+ic[2];
>    f
> end:

> P([1,1])(1);
                                   1
> P([1,1])(1);
                                   2
```

I believe that this subs trick is the "official" way of avoiding these problems. Subs is capable of making a true copy of a procedure (including its remember table – although that's not relevant here). (Thanks to Robert Israel for teaching me that a long time ago.)

```
> P:= proc(ic)
>    local f;
>    f:= subs(_= 0, proc() _ end);
>    f(ic[1]):= f(ic[1])+ic[2];
>    f
> end:

> P([0,1])(0);
                                   1
```

```
> P([0,1])(0);
                                    1

> g:= P([0,1]):
> h:= P([0,1]):
> addressof(eval(g));
                                 3509456
> addressof(eval(h));
                                 3509584
```

7.112.3 Carl DeVore (17.9.01)

(continuing my previous article on the same subject...)

I have fixed the bug. Yes, it was related to the ideas I gave in the article above, but it was a little more subtle than described there because it only affected the remember table of the procedure named ‘unknown‘.

Correction of the Bug about Unknown Remember Tables in `dsolve/numeric`

```
> restart;
> kernelopts(version);
      Maple 7.00, SUN SPARC SOLARIS, May 28 2001 Build ID 96223

> ode:= diff(y(t), t) = y(t):
> dsolve({ode, y(0)=1}, numeric, method= rkf45)(1);

                  [t = 1., y(t) = 2.71828131105043]
```

The problem is that when a procedure is returned, but not assigned to a name, it is assigned to the name "unknown". Unknown can have a remember table. These remember tables suffer from the same globality properties as the remember tables that I showed in my previous article on this subject.

```
> T:= op(4, eval(unknown));

  T := table(["complex" = true,

                                          [ 1..25 1-D Array      ]
       "right_comp_soln_data" = [1, 25, [ Data Type: complex   ],
```

```
                                        [ Storage: rectangular ]
                                        [ Order: C_order       ]

       [ 1..25 x 1..1 2-D Array ]
       [ Data Type: complex     ], 4, false, 37, 0, true, 15, 3]
       [ Storage: rectangular   ]
       [ Order: C_order         ]

       ])

> interface(rtablesize=25);
> T["right_comp_soln_data"][3];

  [0., .0138810615785643, .0277621231571285, .0416431847356928,
       .0555242463142571, .104703338600836, .153882430887414,
       .203061523173993, .252240615460571, .302611511380123,
       .352982407299675, .403353303219226, .453724199138778,
       .503996888692608, .554269578246439, .604542267800270,
       .654814957354100, .697963087684838, .741111218015575,
       .784259348346312, .827407478677050, .870555609007788,
       .913703739338525, .956851869669262, 1.]
```

We see that that is the array of t-values.

```
> convert(T["right_comp_soln_data"][4], listlist);

  [[1.], [1.01397785097315], [1.02815108215469], [1.04252242449701],
         [1.05709464738379], [1.11038122051575], [1.16635381252075],
         [1.22514781794812], [1.28690559131751], [1.35338861554478],
         [1.42330612136986], [1.49683553403133], [1.57416362907186],
         [1.65532416183425], [1.74066902853877], [1.83041395956687],
         [1.92478603124900], [2.00965481087161], [2.09826560191513],
         [2.19078339230133], [2.28738057535581], [2.38823708346178],
         [2.49354052961592], [2.60348698245198], [2.71828131105043]]
```

And that is the array of y-values. These rather complicated remember tables are only used in the new rkf45 and rosenbrock methods of dsolve/numeric.

To correct the bug, we unassign unknown at the start of dsolve/numeric.

```
> p:= Patch(`dsolve/numeric`):
```

The lengthy procedure/module Patch is at the end of this article.

```
> with(p);
                  [&-, &r, &s, Recompile, c, oo, s]

> &-1;

   1   userinfo(1,{dsolve, `dsolve/numeric`},'entering');

> 0&r"`Start patch: Carl 12Sep2001`; \
>     proc(p) \
>        if p then \
>           userinfo(1, {dsolve, `dsolve/numeric`}, `Unassigning protected unknown.`);
>           unprotect(':-unknown') \
>        fi;
>        unassign(':-unknown'); \
>        if p then protect(':-unknown') fi \
>     end \
>     (evalb(:-unknown::protected)); \
>     `End patch`; \
>    ";

   1 `Start patch: Carl 12Sep2001`;  proc(p)  if p then userinfo(1,
{dsolve, `dsolve/numeric`}, `Unassigning protected unknown.`);
unprotect(':-unknown')  fi;  unassign(':-unknown');  if p then
protect(':-unknown') fi end (evalb(:-unknown::protected));  `End patch`;
userinfo(1,{dsolve, `dsolve/numeric`},'entering');
Note the trick above for including *permanent* comments in code.

> Recompile();
```

The Patch/Recompile procedure does not currently install the patched code in a library. That is being worked on. For the time being, you may install it in a library yourself.

Test the patch:

```
> infolevel[all]:= 1:
```

```
> dsolve({ode, y(0)=1}, numeric)(1);
dsolve/numeric:   entering
                  [t = 1., y(t) = 2.71828131105043]

> dsolve({ode, y(0)=2}, numeric)(1);
dsolve/numeric:   entering
                  [t = 1., y(t) = 5.43656262948750]

> f:= dsolve({ode, y(0)=1}, numeric);
dsolve/numeric:   entering

> f(1);
                [t = 1., y(t) = 2.71828131105041937]

> f:= dsolve({ode, y(0)=2}, numeric);
dsolve/numeric:   entering

> f(1);
                [t = 1., y(t) = 5.43656262948751578]
```

Here is the procedure Patch. This is a work-in-progress that will soon be distributed as a patch-library-maintainence system.

A Patching System for Maple's Internal Source Code

Author: `Carl Devore <devore@math.udel.edu>`

12 September 2001

Thanks to Joseph Riel for the idea that it might be possible to recompile the modified output of showstat. Thanks to Robert Israel for the idea that local procedures can be extracted from modules.

```
> restart;
> Patch:= proc(ProcName::evaln, m::`module`)
>    option `Copyright (C) 2001, Carl James DeVore, Jr.  All rights reserved.`;
>    global _PatchLock;
>    local M, IsModule, _TheProc, S, `&->`, p, LockCheck, findline, findlines, strip,

>    # If this module is the Patcher, then this procedure below is the Snatcher.
>    # It will retrieve any export or local from any module.  Note that in the
```

```
>    # case of exports, this procedure works in cases where ":-" does not.

>    `&->`:= proc(M::`module`, Item::evaln)
>       local items, p, stringer, item;

>       stringer:= proc(item)
>          local s, p, newp;
>          s:= sprintf("%A", item);
>          p:= 1;
>          do  # Might need to remove some module name prefixes
>             newp:= searchtext(":-", s, p..-1);
>             if newp=0 then return s[p..-1] else p:= p+newp+1 fi
>          od
>       end proc;

>       items:= [op(3, eval(M))];  #The locals
>       item:= stringer(Item);
>       if not member(item, map(stringer, items), p) then
>          items:= [op(1, eval(M))];  # The exports
>          if not member(item, map(stringer, items), 'p') then error "%1 not found in
>       fi;
>       eval(items[p])
>    end proc;

>    # Find index in S of a line number.
>    findline:= (n::nonnegint)-> `if`(n=0, 0, searchtext(cat("\n", " " $ 4-length(n),

>    # Find indices into S that delimit a pair of line number.
>    findlines:= proc(m::nonnegint, n::posint)
>       local p;
>       p:= findline(m);
>       if p=0 and m<>0 then error "linenumber %1 is too large", m fi;
>       p+1..findline(n)-1
>    end proc;
>
>    # Strip replaces \n's with blanks.  When a single line is displayed, this
>    # format is used.  That way the arrows in search string markers and compile
>    # error markers will make sense.
```

```
>    strip:= proc(In::string)
>       # This proc can be done better in Maple 7 with the StringTools.
>       local Out;
>       Out:= In;
>       do try Out:= ReplaceText(Out, "\n", " ") catch: return Out end od
>    end proc;

>    # ReplaceText finds the nth occurence of s1 in s and replaces it with s2.
>    # n defaults to 1.
>    # This procedure is a companion to Maple's SearchText.
>    # The fourth argument is not currently used in this module.

>    ReplaceText:= proc(s::string, s1::string, s2::string, n::posint)
>       local i,p,l;
>       i:= 0;
>       l:= length(s);
>       to `if`(nargs=4, n, 1) do
>          if i=l then error "String not found: %1", s1 fi;
>          p:= SearchText(s1, s, i+1..-1);
>          if p=0 then error "String not found: %1", s1 else i:= i+p fi
>       od;
>       cat(s[1..i-1], s2, s[i+length(s1)..-1])
>    end proc;

>    IsModule:= evalb(nargs>1);
>    if IsModule then
>       M:= m;
>       if not eval(M)::`module` then error "2nd argument must the name of a module" f
>       _TheProc:= eval(M)&->ProcName;
>       # At this point, it might me good to look for lexical vars, and return
>       # an error in that case.
>    else
>       M:= ``;
>       _TheProc:= ProcName
>    fi;

>    if not eval(_TheProc)::procedure then
>       error "First argument must be the NAME of a procedure"
>    fi;
```

```
>
>    # The only data structure in this module is a copy of the procedure
>    # specifed by P that is put into a string
>    # and has had \n's (newlines) and statement numbers added by debugopts.
>
>    S:= sprintf("%s", debugopts(procdump= _TheProc));
>    # Note the above statement causes "_TheProc" to appear in an assignment
>    # statement in S.

>    p:= SearchText("|Calls Seconds  Words|\nPROC |", S);
>    if p>0 and p<SearchText("\n   1", S) then
>       error "Turn off statement-level tracing before patching."
>    fi;
>
>    # Since the parse statement operates at the global level, we only allow
>    # one patch-in-progress at a time.
>    # I could do some complicated stuff to get around this, but I can't think
>    # of any good reason why someone would want to do this.  A patch is
>    # considered to be "in progress" if it has not been recompiled.
>    if assigned(:-_PatchLock) and :-_PatchLock then
>      WARNING("can only patch one proc at a time.  Previous patch will be scrapped.")
>    fi;
>    :-_PatchLock:= true;

>    LockCheck:= proc()
>       if not :-_PatchLock then error "Already Recompiled.  Reload module to patch ag
>    end proc;

>    module()
>       export `&-`, `&s`, `&r`, Recompile, oo, c, s;
>       local findcurrentlineno, i, Relink;
>
>       s:= ``;
>       c:= ``;
>       # Find the last line number.  Set to oo.
>       for i do if S[-i]="\n" then try oo:= sscanf(S[-i+1..-i+5], "%d")[]; if oo::pos

>       findcurrentlineno:= proc(S::string)
>          local i,n;
```

```
>          for i to length(S) do
>             if S[i]="\n" then try n:= sscanf(S[i+1..i+5], "%d")[]; if n::posint then
>          od;
>          oo
>       end proc;

>       # The procedure Recompile:
>       #   1. Replaces all showstat formatting codes and statement numbers by blank
>       #      The length is not changed to facilitate returning to the exact point
>       #      that caused on error. The original S is not changed, a copy is made.
>       #   2. Compiles string S to a procedure using parse. Sometimes debugopts
>       #      creates syntax that parse won't accept.
>       #      The only example I know is "catch NULL:". I attempt to figure out if
>       #      the compilation error is caused by this, and then change these to
>       #      simply "catch     :"
>       #   3. If compilation fails, then the user is returned to the editor with
>       #      an "^" pointing to the position that parse is complaining about.
>       #   4. If the procedure came from module, then it is relinked to the module.
>       #
>       # Note that parse statements operate at the global level. That means that
>       # the name of the procedure is assigned by the parse statement.
>
>       Recompile:= proc()
>          local In, Out, B, N, Protected, LE;
>          LockCheck();
>          In:= S;
>          Out:= "";
>          do
>             B,In:= sscanf(In, "%[^\n]%[^\r]")[];
>             Out:= cat(Out, B, " ");
>             if In="\n" then break fi;
>             B,N,In:= sscanf(In[2..-1], "%[ ]%[0-9]%[^\r]")[];
>             Out:= cat(Out, " " $ length(B)+length(N))
>          od;
>          Protected:= evalb(_TheProc::protected);
>          if Protected then unprotect(_TheProc) fi;
>          try
>             parse(cat(Out, ";"), statement)
>          catch:
```

```
>             do try Out:= ReplaceText(Out, "catch NULL:", "catch     :") catch: break
>             try
>                parse(cat(Out, ";"), statement)
>             catch:
>                LE:= [lastexception];
>                if nops(LE)=4 and LE[-1]::posint then
>                   # Display the compilation error
>                   &-findcurrentlineno(S[LE[-1]..-1]);
>                   printf("\n%s", cat(" " $ LE[-1]-findline(c)-2, "^"));
>                   error LE[2][1..-5], LE[3..-2]
>                fi;
>                error LE[2..-1]
>             end try
>          finally
>             if Protected then protect(_TheProc) fi
>          end try;
>          if IsModule then
>             # The _TheProc in the next statement is the global _TheProc.  Note
>             # how this trick allows the assignment to
>             # any export of any module as long as the name of module is global.
>             try parse(sprintf("%A:-%A:= _TheProc;", M, ProcName), statement)
>             catch: Relink(Procname, eval(:-_TheProc), eval(M))
>             end
>          fi;
>          # If the compilation and relink were successful, then the current patch
>          # is not "in progress"
>          :-_PatchLock:= false;
>          [][]
>       end proc;

>       # Relink a procedure local to a module.
>       Relink:= proc(N::name, P::procedure, M::`module`)
>          # Waiting for Helmut Kahovec to write this procedure.
>          error "Relinking local procedures to their parent modules not yet implement
>       end;
>
>       # Search command. Sets the current line and the current search string.
>       `&s`:= proc(N::{nonnegint,string}, Search::string)
>          local n,p,pos;
```

```
>          LockCheck();
>          if nargs=0 then  #repeat current search, advancing one line
>             if not s::string then
>                error "Use &s\"string\"; or s:= \"string\"; to set the current search
>             fi;
>             n:= `if`(c::nonnegint, c, 0)
>          elif nargs=1 then
>             if not N::string then error "String expected: %1", N fi;
>             s:= N;
>             n:= `if`(c::nonnegint, c, 0)
>          else
>             if N::string then error "Line number expected for left argument." fi;
>             n:= N;
>             s:= Search
>          fi;
>          if n=oo then n:= 0 fi;  #Recycle the search
>          #If current line is 0, check it; otherwise advance 1 line before searchin
>          p:= `if`(n<1, 1, findline(`if`(nargs=0, n+1, n)));
>          pos:= SearchText(s, S, p..-1);
>          if pos=0 then error "Not found." fi;
>          &-findcurrentlineno(S[p+pos-1..-1]);
>          printf("\n%s", cat(" " $ p+pos-findline(c)-2, "^" $ length(s)))
>       end proc;

>       # Line display command. If used with a single argument, it also sets the
>       # current line.
>       `&-`:= proc(m::nonnegint, n::posint)
>          local s;
>          LockCheck();
>          s:= S[findlines(m, `if`(nargs=1, m+1, min(10^5, n+1)))];
>          if nargs=1 then s:= strip(s) fi;
>          printf("%s", s);
>          c:= `if`(nargs=1, m, ``);
>          [][]
>       end proc;

>       # Replace command.  Only works on the current line.
>       `&r`:= proc(N::{string, identical(0), identical(1)}, Replace::string)
>          local p,q,replace,n,new;
```

```
>          LockCheck();
>          if nargs=1 and not s::string then
>             error "Use &s\"string\"; or s:= \"string\"; to set current search."
>          fi;
>          if not c::nonnegint then error "Use &-n; or c:=n; to set line number to n b
>          if nargs=1 then replace:= N; n:= `` else n:= N; replace:= Replace fi;
>          p,q:= op(findlines(c, c+1));
>          if n="" then new:= replace                                    #replace whole l
>          elif n=`` then new:= ReplaceText(S[p+6..q], s, replace)       #replace search
>          elif n=0 then new:= cat(replace, S[p+6..q])                   #prepend to line
>          elif n=1 then new:= cat(S[p+6..q], replace)                   #append to line
>          else error "Left argument can only be \"\", 0, or 1, but received %1", n
>          fi;
>          S:= cat(S[1..p+5], new, `if`(c=oo, "", S[q+1..-1]));
>          &-c
>       end proc

>    end module
> end proc:
```

A bit of help about the various editting commands:

Let's test it on a procedure local to a module. Note, thus, that we are looking at code that is ordinarily difficult to look at.

```
> p:= Patch(`Map/Internal`, eval(LinearAlgebra)):

> with(p);
                  [&-, &r, &s, Recompile, c, oo, s]
```

List the entire proc. Lines start at 0 (the procedure header), and go to oo.

```
> 0&-oo;

_TheProc := proc(MV, f)
local x, i, j, rows, cols;
   1   if type(MV,{'Vector', 'Matrix'}) then
   2     if member(rtable_options(MV,storage),{sparse, sparse[upper], sparse[lower]})
   3       for x in rtable_elems(MV) do
```

```
 4          try
 5            MV[lhs(x)] := f(rhs(x),args[3 .. -1])
            catch NULL:
 6            NULL
            end try
          end do
        elif type(MV,'Vector') then
 7        for i to op(1,MV) do
 8          try
 9            MV[i] := f(MV[i],args[3 .. -1])
            catch NULL:
10            NULL
            end try
          end do
        else
11        rows, cols := op(1,MV);
12        for i to rows do
13          for j to cols do
14            try
15              MV[i,j] := f(MV[i,j],args[3 .. -1])
              catch NULL:
16              NULL
              end try
            end do
          end do
        end if;
17      MV
      else
18      f(MV,args[3 .. -1])
      end if
end proc
```

Note that the line numbers are actually the statement numbers as reported by showstat. These numbers are fixed throughtout a single editting session.

c refers to the current line number.

```
> c;
There is no current line.
```

Search for 1st occurence of "MV"

```
> &s"MV";

_TheProc := proc(MV, f) local x, i, j, rows, cols;
                  ^^
Check current line and search.

> c;
                                     0
> s;
                                   "MV"
```

Set current line to 3 and display.

```
> &-3;
   3       for x in rtable_elems(MV) do
```

Replace entire line.

```
> ""&r" `junk`;";
   3   `junk`;
```

Search for "cols" starting on the this line.

```
> &s"cols";
  11       rows, cols := op(1,MV);
                ^^^^
```

Put something at the beginning of the line.

```
> 0&r"junk2; ";
  11  junk2;      rows, cols := op(1,MV);
```

Search for "MV", starting at line 7.

```
> 7&s"MV";
   7       for i to op(1,MV) do
```

MPut something at the end of the line.

```
> 1&r"junk 3";
   7       for i to op(1,MV) dojunk 3
```

Set current line to 0.

```
> &-0;
_TheProc := proc(MV, f) local x, i, j, rows, cols;
```

Replace current search string in current line.

```
> &r"junk 4";
_TheProc := proc(junk 4, f) local x, i, j, rows, cols;
```

Search again for the current search string.

```
> &s();
   1   if type(MV,{'Vector', 'Matrix'}) then
               ^^


> Recompile();
_TheProc := proc(junk 4, f) local x, i, j, rows, cols;
                      ^

Error, (in Recompile) incorrect syntax in parse: unexpected number
```

Note that the position of the compilation error is indicated.

Scrap this editting session. Let's write our own module to edit.

```
> mymodule:= module()
>    export myproc;
>    myproc:= proc(P) try print(`Hello, world.`) catch: end end
> end;

      mymodule := module _m649292 () export myproc;  end module

> p:= Patch(myproc, mymodule):
Warning, can only patch one proc at a time.  Previous patch will be scrapped.
```

```
> with(p);
Warning, these names have been rebound: &-, &r, &s, Recompile, c, oo, s
                  [&-, &r, &s, Recompile, c, oo, s]

> 0&-oo;

_TheProc := proc(P)
   1   try
   2     print(`Hello, world.`)
       catch NULL:
   3     NULL
       end try
end proc
```

Note the "`catch NULL:`" This is incorrect syntax generated by showstat. The Recompile handles it gracefully.

```
> &s"Hello";

   2     print(`Hello, world.`)        catch NULL:
               ^^^^^


> &r"Goodbye";

   2     print(`Goodbye, world.`)        catch NULL:

> Recompile();
> mymodule:-myproc();
                            Goodbye, world.
```

That proves that the module was actually changed. Also note that the procedure was automatically reinserted into its parent module. The full version of Patch will also reinsert local procedures into their parent modules.

Test the protection feature:

```
> P:= proc() "junk" end;
> protect(P);
```

```
> evalb(P::protected);
                                  true
> p:= Patch(P):
> with(p);
Warning, these names have been rebound: &-, &r, &s, Recompile, c, oo, s
                 [&-, &r, &s, Recompile, c, oo, s]

> &s"junk";

   1   "junk" end proc
        ^^^^

> &r"new junk";

   1   "new junk" end proc

> Recompile();
> P();
                               "new junk"
> evalb(P::protected);
                                  true
```

7.113 bug in numerical integration for elementary function in Maple V.4 and Maple V.5 (27.11.97)

7.113.1 Harold P. Boas

Motivated by a recent posting of M. L. Glasser, I found the following example of an elementary function for which the default numerical integration algorithm in Maple V Release 4 comes horribly unstuck.

```
f := x -> 1/(1+log(1/x));
                                    1
                    f := x -> ------------
                              1 + log(1/x)
```

Observe that $f(x)$ approaches the limit 0 when x tends to 0 from the right, $f(1) = 1$, and f is increasing on the interval $[0, 1]$. Consequently, the integral of f on $[0, 1]$ is between 0 and 1. However, Maple evaluates the integral to a value of approximately 0.6 times ten to the sixteenth power!

```
evalf(Int(f(x), x=0..1));
                                       16
                          .6066382611 10
```

Setting `infolevel[‘evalf/int‘]:=1;` reveals where Maple goes wrong in this example. Near $x = 0$, where Maple thinks that there is a singularity, Maple expands the integrand in a geometric series in powers of `log(x)` and truncates the series, failing to notice that the series is badly divergent.

Diverting Maple into a different branch of its numerical algorithm produces the correct value of about 0.596. For example, each of the following commands gives a correct result.

```
evalf(Int(1.0*f(x), x=0..1));
evalf(Int(f(x), x=0..1, 10, _Dexp));
evalf(int(1/(1-log(x)), x=0..1));
```

It is corrected with Maple 6. (U. Klein)

7.114 bug in odeplot with Mac OS X, Maple 7 (8.9.02)

7.114.1 Roberto Sussman

I found a nasty plot using Maple 7 with a Mac (G4 Tower 400 Mhz, Mac OS X version 10.1). Since Maple is not native it runs under the "Classic" Mac OS 9.2 emulation.

First I run the following simple commands related to solving numericaly a system of two ODE's

```
> eqMN:=(beta0,lambda0,x)->diff(M(x),x)=piecewise(x>0,
            x^2*(exp(beta0-beta(x))+lambda0), x=0, 0);
> eqbN:=(lambda0,x)->diff(beta(x),x)=piecewise(x>0,
> (M(x)/x^2-lambda0*x)*beta(x), x=0, 0);
> S0:=dsolve({eqMN(10^6,0,x),eqbN(0,x),M(0)=0,beta(0)=10^6},
            {M(x),beta(x)},type=numeric);
> seq(S0(.5*i), i=1..5);
> plots[odeplot](S0,[x,log10(M(x))],0..2.5,color=black);
```

Up to this point everything is fine. But then, I want to redo the ODE's with a different method, therefore I edit the dsolve commands adding (say) `"method = lsode"` (the problem is independent of the method).

I do so and then re-run all the commands. It does the numerical solutions and tabulates numerical values, but displays an error message as I try to plot them with odeplot. The message is

```
Error, (in StringTools:-Remove) mapped expression must return `true'
or `false'
```

Though, you can obtain the plot by forming a list of lists with the points by using instead of odeplot the command

```
plot([seq([i/10,log10(rhs(S0(i/10)[2]))], i=0..25)], color=black);
```

Afterwards, since I could not get any plot with odeplot I quit Maple and tried to start a new session, but got the error message

```
"The application Maple has unexpectedly quit... etc"
```

Hence I had to force-quit the whole Classic environment. This annoying bug does not happen if running the same commands under Mac OS 9.x. Hopefuly the Mac OS X native version of Maple might come soon. I write below the transcript of the Maple session (after re-executing the group of commands)

```
> restart;
> eqMN:=(beta0,lambda0,x)->diff(M(x),x)=piecewise(x>0,
            x^2*(exp(beta0-beta(x))+lambda0), x=0, 0);

  eqMN := (beta0, lambda0, x) -> diff(M(x), x) = piecewise(0 < x,

         2
        x  (exp(beta0 - beta(x)) + lambda0), x = 0, 0)

> eqbN:=(lambda0,x)->diff(beta(x),x)=piecewise(x>0,
             (M(x)/x^2-lambda0*x)*beta(x), x=0, 0);

  eqbN := (lambda0, x) -> diff(beta(x), x) =

                          /M(x)             \
        piecewise(0 < x, |---- - lambda0 x| beta(x), x = 0, 0)
                          |  2              |
                          \ x               /

> S0:=dsolve({eqMN(10^6,0,x),eqbN(0,x),M(0)=0,beta(0)=10^6},
            {M(x),beta(x)},type=numeric, method=lsode);

                 S0 := proc(x_lsode)  ...  end proc

> seq(S0(.5*i), i=1..5);

                                          -5
  [x = .5, M(x) = .107253635455685075 10   ,

                                          7
        beta(x) = .100001172170238756 10 ], [x = 1.0,

                                        -5
        M(x) = .207236310045923610 10   ,
```

```
                                          7
        beta(x) = .100001321418795199 10 ], [x = 1.5,

                                        -5
        M(x) = .300759851927367088 10   ,

                                          7
        beta(x) = .100001405604256631 10 ], [x = 2.0,

                                        -5
        M(x) = .391263046364007452 10   ,

                                          7
        beta(x) = .100001463943739084 10 ], [x = 2.5,

                                        -5
        M(x) = .480790975386358341 10   ,

                                          7
        beta(x) = .100001508190117218 10 ]

 > plots[odeplot](S0,[x,log10(M(x))],0..2.5,color=black);
Error, (in StringTools:-Remove) mapped expression must return `true'
or `false'

 > plot([seq([i/10,log10(rhs(S0(i/10)[2]))], i=0..25)], color=black);
```

7.115 Bug in orthopoly in Maple V.4 to Maple 6 (8.11.96)

7.115.1 Gerald A. Edgar

Maple V, release 4 (PowerMac version).

I got a strange sum the other day...

```
> sum((k-2)*x^k/k!,k=0..infinity);

                          exp(x) orthopoly[L](1, -3, -x)
```

But Maple does these two fine:

```
> sum(k*x^k/k!,k=0..infinity);
                                      x exp(x)
> sum(2*x^k/k!,k=0..infinity);
                                      2 exp(x)
```

It seems to be a bug in orthopoly package...

```
> with(orthopoly):
> L(1,1,x);
                                       -x + 2
> L(1,a,x);
                                     1 + a - x
> L(1,-3,x);
                                     L(1, -3, x)      ????????
> L(1,-3.0,x);
                                     -2.0 - x
```

It is corrected with Maple 7. (U. Klein)

7.116 bug in parallelepiped, Maple 7 (13.12.01)

7.116.1 Harry Kneppers

In Maple 7, the `geom3d[parallelepiped]` command doesn't do what I want and I don't know what mistake I am making.

If I do

```
with(geom3d):
point(A,1/2,0,0), point(B,1,0,0), point(C,1/2,2,0), point(D,1/2,0,3):
dsegment(d1,[A,B]), dsegment(d2,[A,C]), dsegment(d3,[A,D]):
parallelepiped(pp,[d1,d2,d3]);
draw(pp);
```

it doesn't look like a parallelepiped. I get this kind of trouble as soon as A is not the origin.

I also tried to start with a parallelepiped at the origin and then use geom3d[translation], but that did not help me out. It gives a nice picture, but it is not what I want.

7.116.2 Robert Israel (15.12.01)

There's a simple mathematical bug in "parallelepiped". The expressions $B+C-2*A, E+C-2*A, B+C+E-3*A$ and $B+E-2*A$ in lines 25 to 28 should be $B+C-A, E+C-A, B+C+E-2*A$ and $B+E-A$ respectively.

It is corrected with Maple 8 (U. Klein)

7.117 bug in pdsolve (second order PDE) in Maple 6 (17.8.00)

7.117.1 E. Elbraechter

For the following second order PDE `(A&*Laplace_Operator(F) + a&*gradient(F))` I get a result witch contains two non global variables ($X[1], X[2]$). The check with pdetest is negative.

```
> restart;
> map(interface, [version, patchlevel]);

  [Maple Worksheet Interface, Maple 6, IBM INTEL LINUX22, Jan 31 2\

        000 Build ID 16401, 0]


> pde := a11*diff(F(x,y), x,x)  + a22*diff(F(x,y), y,y)
>         +  b1*diff(F(x,y), x)    +  b2*diff(F(x,y), y):
> sol  := pdsolve(pde, build);

  sol := F(x, y) = _F1(

        (2 a22 b1 - 2 sqrt(-a11 a22) b2 + 4 a11 a22 X[2])

        /                y a11     \
        |1/2 x - 1/2 --------------| -
        \            sqrt(-a11 a22)/

        (-b1 sqrt(-a11 a22) + b2 a11 + 4 a11 a22 X[1])

        /sqrt(-a11 a22) x    \
        |---------------- + y|)
        \      a11           /

> 'pdetest(sol, pde) = 0':     '%' =  evalb(%);

         (pdetest(sol, pde) = 0) = false
```

```
> indets(sol, indexed) minus {X[1], X[2]};  # non global variables X[1],X[2]

                              {X[1], X[2]}
```

If I set either `a11=a22=0` (no Laplace-Operator) or `b1=b2=0` (no gradient term) the checks with pdetest are positive.

It is corrected with Maple 7. (U. Klein)

7.117.2 Edgardo S. Cheb-Terrab (18.8.00)

You are correct - the result you show is wrong - those `X[1], X[2]` should not be there. The correct result is

```
> pdsolve(pde,build);

                        2 /sqrt(-a11 a22) x    \         4
                    _c[1]  |---------------- + y| a11 a22
                           \      a11           /
  F(x, y) = _C1 exp(-------------------------------------)
                                   %1

                  /sqrt(-a11 a22) x    \
              _c[1] |---------------- + y| b2 a11
                    \      a11          /
        exp(-----------------------------------)
                           %1

                                 /                y a11     \    2
              _c[1] sqrt(-a11 a22) |1/2 x - 1/2 --------------| b2
                                 \             sqrt(-a11 a22)/
        exp(-------------------------------------------------------)
                                   %1

                /              /                 y a11     \    2
                |    _c[1] a22 |1/2 x - 1/2 --------------| b1
              / |              \             sqrt(-a11 a22)/
        _C2  /  |exp(---------------------------------------)
            /   \                          %1
```

```
                /sqrt(-a11 a22) x    \                    \
           _c[1] |---------------- + y| b1 sqrt(-a11 a22) |
                 \      a11           /                    |
       exp(-----------------------------------------------)|
                                %1                         /

  %1 := 4 a11 a22 _c[1] - b1 sqrt(-a11 a22) + b2 a11

> pdetest(%,pde);
                                0
```

An alternative to have pdsolve in your machine free of this problem would be for you to download the PDEtools package available on the web at `http://lie.uwaterloo.ca/pdetools.htm`. There you will find the latest version of PDEtools. To have the downloaded package active and so Maple's pdsolve updated it suffices to drop the package in any directory and add that directory to the libname (see instructions in the readme.txt file distributed in the zip). This is not an official Maple patch, but the version of the package on the web is "free of known bugs up-to-today"| and maintained with the help of feedback from people all around.

7.117.3 Willard, Daniel Dr (21.8.00)

Try Release 5.1:

```
>with(PDEtools):
>pde:=a11*diff(F(x,y),x,x)+a22*diff(F(x,y),y,y)+b1*diff(F(x,y),x)
                          +b2*diff(F(x,y),y):
>pdsolve(pde, F, build);
>simplify(%);
```

The result comes easily.

7.118 Bug in plot in Maple V.5 (15.3.99)

7.118.1 Luis Goddyn

Why does plot get stuck on this constant function? Is it trying to simplify before plotting? Be sure to save your work first!

```
> restart; interface(version); kernelopts(version);

         TTY Iris, Release 5, SUN SPARC SOLARIS, Jan 11 1998
  Maple V, Release 5, SUN SPARC SOLARIS, Jan 11 1998, CAW-55CD-951412-7

> p :=  (sqrt(2)*exp(t) + 1) * (sqrt(2)*exp(t) - 1);

                      1/2                1/2
              p := (2    exp(t) + 1) (2    exp(t) - 1)

> q :=  2*exp(t)^2 - 1;
                                      2
                          q := 2 exp(t)  - 1

> simplify(expand(p)/q);  # Note: simplify(p/q) fails, even with options!

                                    1

> plot(p/q, t=0..1);
```

[Comments: TTY freezes here, `^C` to get out. X11 and MWI interfaces also freeze.]

Plotting error, `buffer_read`: read failed

It is corrected with Maple 6. (U. Klein)

7.118.2 Robert K Wright (24.3.99)

The problem is not related to the use of `exp(t)`. See below:

```
> interface(version);
   Maple Worksheet Interface, Release 5, IBM INTEL NT, May 11 1998
> kernelopts(version);
   Maple V, Release 5, IBM INTEL NT, May 08 1998, WIN-55CD-314159-2

> p:= (a*x-1)*(a*x+1);

> q:= a^2*x^2-1;

> a:= 3:

> plot(p/q,x=1..2);
```

No problem for a nonradical.

Now assign `a:= sqrt(2) and try plot(p/q,x=1..2):`

The Maple session hangs. In windows I can't save any of this if Maple is hung, so I'm saving now.

(If you try the same example on an HP workstation with Maple V Release 5.1 you get the same result. Don't try to stop the Maple session by typing `^C`! You are not able to stop the process! You have to reboot the workstation or you have to wait for the core dump.

I found the bug some months before. As a workaround you can use `a := evalf(sqrt(2));` (instead of `a := sqrt(2);`) and you will get the plot immediately. U. Klein)

7.118.3 Helmut Kahovec (24.3.99)

If you execute the following statements in a fresh Maple session then everything works fine. However, if you remove the `#` character in the definition of `my_args` then Release 5 crashes when it processes `INTERFACE_PLOT(my_args)`.

```
> restart;

> my_args:=
>   CURVES(
```

```
>     [
>       [0.0, 1.],
> #       [0.5, 1.0000000000000001],
>       [1.0, 1.]
>     ],
>     COLOUR(RGB,1.0,0,0)
>   ),
>   AXESLABELS("t",``),
>   VIEW(0 .. 1.,DEFAULT):
> INTERFACE_PLOT(my_args);
```

`INTERFACE_PLOT()` is called in `` `print/PLOT`() `` as you may verify by debugging any plot() statement and setting the breakpoint by `stopat(`print/PLOT`)`. Its source code is:

```
> `print/PLOT`:=proc()
>   subs(
>     {('undefined')=('FAIL'),('INTERFACE_TEXT')=('TEXT')},
>     INTERFACE_PLOT(args)
>   )
> end:
```

`` `print/PLOT`() `` is called at the final processing stage of any plot() statement when the plot structure has already been computed successfully. `INTERFACE_PLOT()` transfers the data of the plot structure from the Maple kernel to the Maple front-end program. Apparently, that part of the Release 5 software cannot correctly handle a mixture of data points with ordinate values of 1. and `1.+EPS`. EPS is the smallest value different from 0 recognized by the CPU.

Since Maple cannot (automatically) simplify the expression

```
> expr:=(sqrt(2)*exp(t)+1)*(sqrt(2)*exp(t)-1)/(2*exp(2*t)-1);

                  (sqrt(2) exp(t) + 1) (sqrt(2) exp(t) - 1)
         expr := ------------------------------------------
                              2 exp(2 t) - 1
```

to 1 plotting expr gives a lot of data points with ordinate values of `1.+EPS`. Since they are mixed with 1. your plot statement crashes at that very point.

7.119 bug in plots[display] under Unix, Maple 6 and Maple 7 (31.1.02)

7.119.1 William Slough

I am using Maple 6 under Linux and I'm trying to animate a sequence of point plots. With insequence=false, everything looks fine but when I set `insequence=true`, the animation has just a single point: the last frame.

Here's a simple example:

```
with(plots);
p1 := plot([[1,2]], style=POINT, view=[0..5,0..5], scaling=constrained):
p2 := plot([[2,4]], style=POINT, view=[0..5,0..5], scaling=constrained):
display([p1,p2], insequence=true);
```

It is corrected with Maple 8 (U. Klein)

7.119.2 Mark Fitch (1.2.02)

I have experienced this, but only under Solaris and Linux. The problem does not occur under Windows systems. I presume this is a bug in the point plotting routine in the unix versions.

7.119.3 Dr Francis J. Wright (5.2.02)

Using Maple 7 under Windows NT it seems to work fine. The animation initially shows the first point, and when I run the animation it flips to the second point.

If I single step, it flips back and forth. However, I have experienced a lot of problems with point plots in Maple 6. I think there are problems driving some displays, which vary from minor corruption of the display to completely crashing Maple.

I was able to avoid this problem on a machine running Windows 98 by reducing the display acceleration level, which had no discernible effect other than to avoid crashing Maple.

I discovered very recently on our teaching machines, which run Windows ME, that Maple 6 would crash if asked to display a single point alone, but not if the point plot was combined with another (non-point) plot. So, if you experiment along these lines you might be able to find a work-around.

The problem might be fixed in Maple 7; I haven't noticed it in Maple 7, but I haven't tried Maple 7 on any of the machines where there were problems using Maple 6.

[Incidentally, Maple 6 also seems to crash when it tries to open a Maple 7 worksheet containing 3D plots! A workaround for that is to delete the plots USING MAPLE 7 and then re-execute the worksheet using Maple 6.]

7.119.4 Koch-Beuttenmueller (7.2.02)

The problem exists for Maple 7 as well. I tried under Compaq Tru 64.

7.119.5 Theodore Kolokolnikov (13.2.02)

Another problem under linux version 7 (i'm not sure if this is version-7 specific) is that it takes relatively long time to display even a very simple plot (for example `plot(sin(x),x=0..10))`.

It seems that the Linux version of Maple was not well tested, at least when compared to windows.

7.119.6 David L. Johnson (20.2.02)

| Theodore Kolokolnikov wrote: ...

I don't think you have found the source of your problem. I would guess that it might be specific to certain hardware. At any rate plotting for me is quite fast with Maple 7 `(debian linux 2.2, PII@400mHz, s3-virge card)`.

7.120 bug in plotting electric field of a dipole in Maple V.2 (24.10.97)

7.120.1 J. David Wright

I would like to plot the electric field of a dipole using Maple V RII for the macintosh.

A simplified vector form for the electric field of a dipole is:

```
[2*cos(theta)/r^3,sin(theta)/r^3]
```

As a first attempt I tried the following:

```
> Edipole:=[2*cos(theta)/r^3,sin(theta)/r^3];
> fieldplot( Edipole,r=-10..10,theta=0..Pi,coords=polar);
Error, (in plot/options2d) unknown or bad argument, coords = polar
```

That method obviously didn't work and I couldn't figure out how to make a plot directly from the polar coordinates.

Then I tried transforming to cartesian coordinates.

```
>Edipole:=subs({r=(x^2+y^2)^(1/2),theta=arctan(y/x)},
                [2*cos(theta)/r^3,sin(theta)/r^3]);
> fieldplot( Edipole,x=-10..10,y=-10..10);
```

This method produced a plot but it did not look like a typical textbook picture of a dipole's electric field.

Then I tried to plot the gradient of the electric potential of a dipole

```
> Vdipole:=subs({r=(x^2+y^2)^(1/2),theta=arctan(y/x)},cos(theta)/r^2);
> gradplot( Vdipole,x=-10..10,y=-10..10);
```

This method produced a plot which looked exactly like that of the previous method.

I also seem to remember getting 'division by `zero`' errors as well while trying to use field plot for this particular problem. However, I cannot seem to reproduce such an error at this moment.

I wonder how one generally deals with 'division by zero' errors in plotting routines.

Further, I would appreciate any suggestions concerning plotting the electric field of a dipole.

It is corrected with Maple V.3. (U. Klein)

7.120.2 Michael Komma (28.10.97)

`http://userwst1.fh-reutlingen.de/~komma/physth/feldli1.ms`

(should be updated, commented and translated ;-)

7.120.3 Robert Israel (28.10.97)

Release 2! I'm using Release 4 right now, but I think my remarks still apply.

| A simplified vector form for the electric field of a dipole is: ...

Release 4 doesn't produce an error message here, but the result is not correct.

| Then I tried transforming to cartesian coordinates. ...

Three problems here:

1) Since the arrows grow in size as `r -> 0`, you only get a few big arrows and the rest very tiny. You might try it without the `r^3` in the denominator, to at least get the directions if not the magnitudes of the arrows.

2) `theta = arctan(y/x)` only works in the right half plane. Use the two-variable form `theta = arctan(y,x)` which works in the whole plane.

3) That's `_not_` the electric field of a dipole! What you want (for the cross-section of a three-dimensional dipole) is `[ 3*cos(2*theta)-1, 3*sin(2*theta)]/r^3`

| I also seem to remember getting 'division by zero' errors as well ...

I've run into it just now in R4 as well. For example,

```
> fieldplot([0,0], x = -1 .. 1, y = -1 .. 1);
Error, (in plots/fieldplot/arrowsf2d) division by zero
```

This is a bug.

| I wonder how one generally deals with 'division by zero' errors ...

If it comes from arrows of 0 length, try to avoid having arrows of 0 length. If they occur only at a discrete set of points or on a curve, you might shift the x and/or y endpoints

a bit to miss these points. Or cheat by adding a small nonzero vector (too small to actually be seen in the plot) to the field.

7.120.4 Michael Komma (28.10.97)

| Error, `(in plot/options2d)` unknown or bad argument, `coords = polar`

This option is available in R4.

| Then I tried transforming to cartesian coordinates. ...

Besides the transformation of coordinates you must transform the fieldvector: `E[x]=E[r]*cos(theta)-E[theta]*sin(theta), E[y]=E[r]*sin[theta]+E[theta]*cos[theta]`

| I wonder how one generally deals with 'division by zero' errors ...

There are at least two reasons for 'division by zero'.

1. The singularity at r=0: shift the plot ranges, so that this point is not evaluated in the grid (example: `-1.001..1.002`) or add a small constant in the denominator `(1/(r^3+.0001))`.

2. If terms in the vector occur, that cannot be evaluated, Maples fieldplot reacts with a 'division by zero' (due to zero-step-size I guess).

Plotting electrostatic fields (fieldlines) is best done with numerical methods, since the analytical solutions are the exception. The 'few' closed solutions can be obtained by multipole-expansion of the *potential*.

7.121 Bug in print(), Maple V.5 to Maple 8 (5.8.02)

7.121.1 Helmut Kahovec

There seems to be a bug in the print() function in all recent releases of Maple. print() does not make use of the continuation character when an expression has many closing parentheses:

```
> `evalapply/e`:=proc(X,Y) op(X)(op(X)(op(Y))) end proc:

> expr:=e(e(e)(e))(e)(e):
```

The output of the following input statement is correct. I have added the continuation character '\' myself in order to make the output of lprint() more readable.

```
> lprint(expr);
e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e\
(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(\
e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e\
(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(\
e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e\
(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(\
e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e\
(e(e(e(e(e(e(e(e(e(e(e(e(e(e)))))))))))))))))))))))))))))))))))))))))\
))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))\
))))))))))
```

However, when expr is print-ed then many closing parentheses are missing when working with the worksheet interface version of Maple.

```
> expr;

  e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(

        e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(

        e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(
```

```
e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(

e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(

e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(

e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(

e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(e(

e(e(e(e(e(e))))))))))))))))))))))))))))))
```

It is correct in Maple V.3 (U. Klein)

7.122 bug in product, Maple V to Maple 8 (18.11.98)

7.122.1 CORNIL Jack Michel

A new bug ?

```
> product(1/(i-k),k=0..i-1);

                                        0
```

I do not think it is possible.

7.122.2 Robert Israel (20.11.98)

Actually I think it's quite an old bug. The explanation in a nutshell is this: Maple first evaluates the indefinite product:

```
   product(1/(i-k), k) = (-1)^k/GAMMA(k-i).
```

This is true in the sense that if you call this F(k), then `F(k+1) = F(k)/(i-k)` (as a function of i). Of course GAMMA has poles at the integers `<= 0`, so it doesn't really work for the case Maple wants: it wants to calculate the definite product as `F(i)/F(0)`. Now `1/F(0) = GAMMA(-i)` which is undefined (but at this point Maple doesn't realize that fact because it doesn't know i is supposed to be a positive integer).

On the other hand, `F(i) = (-1)^i/GAMMA(0)` and Maple does know that GAMMA has a pole at 0. So it evaluates `limit(1/F(k),k=0)` and correctly finds 0. Therefore the answer is evaluated as `(1/GAMMA(-i))*0 = 0`.

On one level the problem is one of recognition of singularities (or zeros): Maple doesn't realize that `GAMMA(-i)` is undefined. I would have thought that it might help to explicitly assume i is a positive integer:

```
> assume(i, posint);
```

Unfortunately, that doesn't work: even if you assume i is a positive integer, `GAMMA(-i)` just returns unevaluated.

The problem, I think, is in the code for GAMMA which checks for (actual) integers <=

0, but doesn't worry about values that are known to be integers <= 0: it uses type(a, integer) rather than is(a, integer).

7.122.3 Helmut Kahovec (22.11.98)

it's a bug. When calculating

```
> restart;
> product(1/(i-k),k=0..i-1);

                                  0
```

Maple (Release 4) executes the following statements:

```
> p:=product(1/(i-k),k);

                                      k
                                   (-1)
                        p := -------------
                             GAMMA(-i + k)

> p1:=eval(subs(k=0,p));

                                    1
                          p1 := ---------
                                GAMMA(-i)

> p2:=eval(subs(k=i-1,p));
Error, (in GAMMA) singularity encountered

> p2:=limit(p,k=i-1);

                               p2 := 0

> p2/p1;

                                  0
```

You may verify this with the help of the Maple debugger (some extensive output

shortened):

```
> stopat(product,73);

                                [product]

> product(1/(i-k),k=0..i-1);
(-1)^k/GAMMA(-i+k)
product:
  73*       if r = lasterror then
              ...
            else
              ...
            fi
> step
(-1)^k/GAMMA(-i+k)
product:
  75          r := `product/subscheck`(r,i,a,b);
> step
`product/subscheck`:
   1    nexpr := expr;
> step
(-1)^k/GAMMA(-i+k)
`product/subscheck`:
   2    tp1 := traperror(eval(subs(i = a,nexpr)));
> step
1/GAMMA(-i)
`product/subscheck`:
   3    if tp1 = lasterror or tp1 = 0 or tp1 = FAIL then
          ...
        fi;
> step
1/GAMMA(-i)
`product/subscheck`:
   5    tp2 := traperror(eval(subs(i = b,nexpr)));
> next
Error, singularity encountered
`product/subscheck`:
   6    if tp2 = lasterror or tp2 = 0 or tp2 = FAIL then
```

```
          ...
        fi;
> step
Error, singularity encountered
`product/subscheck`:
   7      tp2 := limit(nexpr,i = b)
> step
limit:
   1    if nargs < 2 or 3 < nargs or ... then
          ...
        fi;
> return
0
`product/subscheck`:
   8    if tp1 = undefined or ... then
          ...
        else
          ...
        fi
> return
0
product:
  76          if r = FAIL then
                ...
              else
                ...
              fi
> step
0
Warning, statement number may be incorrect
product:
  12        RETURN(r)
> step
```

7.122.4 Willard, Daniel, Dr. (30.11.98)

That matter of `"type(a, integer)"` versus `"is(a, integer)"` bedevils many other programs, for example, the computation of Bessel functions, which will not provide a series solution if the index is not `"type(n, integer)"` though there are perfectly good series expansions for arbirary real or complex n. I have been complaining about this since the first release of Maple V, without any effective response from Maple or Waterloo.

7.123 bug in Random Tools in Maple 7 (28.11.01)

7.123.1 Mike May

I have been playing with some of the new packages with Maple release 7. I am having some trouble with the RandomTools package. As the examples below show the digits option does weird things on the range used.

1) Is this a known bug?

2) When I try verboseproc to look at code, I am having trouble getting to the code called with specific options. Any suggestions on how to do that?

```
>  data := sort([seq(Generate(float(range=0.0321..162.0,
                digits=2,method=uniform)), i=1..2000)]):
>  data[1];data[2000];

                                .032

                                1.6

>  data := sort([seq(Generate(float(range=0.0321..162.0,
                digits=3,method=uniform)), i=1..2000)]):
>  data[1];data[2000];

                               .0337

                                16.2

>  data := sort([seq(Generate(float(range=0.0321..162.0,
                digits=4,method=uniform)), i=1..2000)]):
>  data[1];data[2000];

                               .2092

                               161.9

>  data := sort([seq(Generate(float(range=0.0321..162.0,
                digits=5,method=uniform)), i=1..2000)]):
>  data[1];data[2000];
```

```
                                .34690

                                1619.4

>  data := sort([seq(Generate(float(range=0.0321..162.0,
                digits=6,method=uniform)), i=1..2000)]):
>  data[1];data[2000];

                               1.61150

                               16197.6

>  data := sort([seq(Generate(float(range=0.0321..162.0,
                digits=10,method=uniform)), i=1..2000)]):
>  data[1];data[2000];

                             16630.71940

                                          9
                           .1619774165 10
```

It is corrected with Maple 8 (U. Klein)

7.123.2 Preben Alsholm (3.12.01)

It appears to me that the bug can be found in line 37 of the code for the float-flavor:

```
showstat(op(1,GetFlavor(float)));
```

There it says

```
evalf(Float(rand(manL .. manR*10^Digits)(),expL))
```

I think it should say

```
evalf(Float(rand(manL .. manR*10^(expR-expL))(),expL))
```

And incidentally:

In this procedure 'logarithmic' is consistently misspelled 'logrithmic'. This has the following amusing effect:

```
Generate(float(range=0.0321..162.0,method=logarithmic)):
Error, (in ProcessOptions) argument 'method = logarithmic' invalid: rhs
```

should be of type identical(uniform), identical(logrithmic)

It is corrected with Maple 8 (U. Klein)

7.123.3 Mike May (3.12.01)

Thanks for the help. Since I did not get a copy of the message posted to MUG, it seems that I somehow got deleted from the e-mail list. I have re-subscribed.

A couple of follow-up questions:

1) Is there a standard procedure for reporting bugs in the code?

2) I was able to follow your example to get the code with showstat. I am not clear on the construction of the command. Is there a reason I should suspect I want

```
showstat(op(1,GetFlavor(float)));
```

rather than

```
showstat(GetFlavor(float));
```

3) Looking at the code there is repeated reference in the RandomTools package to the FlavorTable. Is there any way to print out the FlavorTable?

I am also running into "features" of the CurveFitting package. When I run the same data through linear regression on the `stats[fit, leastsquare]` command and the `CurveFitting[LeastSquares]` command, my coefficients on the new command have extra digits. When I do a least squares fit to a higher degree equation the coefficients only agree to 6 significant digits.

```
>  mdata := [65.5, 73.3, 41, 61.6, 42.9, 51, 62.3, 70, 56.9, 52.3]:
>  fdata := [72.7, 79.6, 42, 63.3, 46.1, 55.4, 67.6, 76.8, 56, 59.7]:
>  UNESCO1 := [seq([mdata[i],fdata[i]], i=1..10)]:
>  with(stats[fit]): with(CurveFitting):
>  reglineold := leastsquare[[x,y]]([mdata, fdata]);
```

```
            reglineold := y = -3.192082489 + 1.128850251 x

>  regline := y = LeastSquares(UNESCO1,x);

          regliner := y = -3.192082489 + 1.12885025120340932 x

>  leastsquare[[x,y], y=a+b*x+c*x^2]([mdata, fdata]);

                                                                  2
           y = 5.744726622 + .8029638020 x + .002871857378 x

>  LeastSquares(UNESCO1, x, curve=a+b*x+c*x^2);

                                                                    2
     5.744723280 + .802963923918129096 x + .00287185630353611324 x
```

This leads to the questions:

4) Is this a known bug?

5) Any suggestions on how to determine which command is giving the correct answer?

6) Any way to get the newer package to print the normal number of digits?

I did some more playing that leads me to believe the old stats/fit command is the more accurate:

```
>  with(CurveFitting):with(ListTools):with(stats[fit]):
>  data1 := [[0,0],[1.,2],[2,2],[3,2],[4,4]]:
>  data2 := Transpose(data1):
>  LeastSquares(data1,x);

                  .4000000000 + .799999999999999932 x

>  leastsquare[[x,y]](data2);

                    y = .4000000000 + .8000000000 x
```

7.123.4 Preben Alsholm (7.12.01)

| Mike May wrote:

| 1) Is there a standard procedure for reporting bugs in the code?

No, as far as I know. I used to report bugs to 'support', but have stopped doing that. Now I just use the newsgroup `comp.soft-sys.math.maple`. I would be very surprised, if the Waterloo Maple people don't read that. The advantage is that the bug is made public.

| 2) I was able to follow your example to get the code with showstat.

GetFlavor(float) is a module, which can be seen by doing:

```
eval(GetFlavorFloat);
```

It exports the procedure Main. The exports make up the first operand of a module. The code for Main can also be accessed by doing:

```
interface(verboseproc=2):
eval(GetFlavor(float):-Main);
```

or instead

```
showstat(GetFlavor(float):-Main);
```

| 3) Looking at the code there is repeated reference in the ...

FlavorTable is local to the module RandomTools. The locals form the third operand:

```
op(3, eval(RandomTools));
```

To see the module structure try

```
op(2,eval(RandomTools));
```

To print the FlavorTable do

```
eval(op([3,1],eval(RandomTools)));
```

| I am also running into "features" of the CurveFitting package. ...

Try increasing Digits to e.g. 20. Then it appears that there is considerable agreement between CurveFitting and stats[fit].

| 4) Is this a known bug?

I don't think there is a bug.

| 6) Any way to get the newer package to print the normal number of digits?

On the result you could use

```
evalf[5](%);
```

if you want 5 digits, say.

7.123.5 Carl Devore (13.12.01)

| Mike May wrote:

| I am also running into "features" of the CurveFitting package. ...

The LeastSquares computation is using hardware floating point double precision, which is usually 14 or 15 decimal digits.

| When I do a least squares fit to a higher degree equation ...

And the LeastSquares is the more accurate one. You can verify this by setting `Digits:= 20`, and redoing the (old) leastsquare. (Or just check the residulas).

| 4) Is this a known bug?

No bug. The old packge was doing the computation to 10 digits, and the newer was using more.

| 5) Any suggestions on how to determine which command is giving ...

Look at the residuals. See my worksheet "Contour plots for three-ingredient mixing problems" at `www.mapleapps.com` for more info.

| 6) Any way to get the newer package to print the normal number of digits?

Run the results through evalf.

| I did some more playing that leads me to believe the old stats/fit ...

It is just a rare coincidence that in this example the coefficients can be computed exactly in base 10 arithmetic. The software floats that Maple uses are base 10. The hardware floats used by LeastSquares are base 2.

7.124 bug in series, Maple 6 to Maple 8 (15.10.01)

7.124.1 Michael Rubinstein

Any explanations as to why the former works but the latter doesn't?

```
>  linux174$ maple
>       |\^/|     Maple 6 (IBM INTEL LINUX)
>  ._|\|   |/|_. Copyright (c) 2000 by Waterloo Maple Inc.
>   \  MAPLE  /  All rights reserved. Maple is a registered trademark of
>    <____ ____>  Waterloo Maple Inc.
>         |       Type ? for help.
>  > series(1/(1-x),x);
>                                    2    3    4    5      6
>                          1 + x + x  + x  + x  + x  + O(x )
>
>  > series(1/(1-x)^1.,x);
```

7.124.2 Dr Francis J. Wright (16.10.01)

Using floating-point exponents is not something I would want to do without very good reason. However, it seems to me that the second expansion is well defined, and indeed Maple (7) seems quite happy to do it in two steps:

```
subs(alpha=1., series(1/(1-x)^alpha,x));

                  2                3                4                5
  1 + 1. x + 1. x  + 1.000000000 x  + 1.000000000 x  + 1.000000000 x

              6
         + O(x )
```

So I guess it's a bug (in Maple 7 also). Since series is a builtin function, there is not much more that we mere mortals to do to investigate it.

7.125 Bug in simplify(Dirac) in MapleV.4 and Maple V.5 (21.10.98)

7.125.1 Lev A. Melnikovsky

It seems I don't understand pretty much. Either in Maths or in Maple. Here's what I do (Version 4.00b, library patch level 2)

```
> Dirac(2*x+y);
                                  Dirac(2 x + y)

> simplify(");
                                1/2 Dirac(2 x + y)
```

Could someone, please, help me? Is it a feature or a bug? Do I miss something? It seems I'm going to become crazy with it...

P.S. Look at `'simplify/Dirac/expression'`! It looks rather strange...

It is corrected with Maple 6. (U. Klein)

7.125.2 Robert Israel (27.10.98)

It's a bug. At first I thought it was fixed in Release 5, but then I realized it was not: the bug may or may not occur, depending on whether Maple happens to look at the x or y variable first. If you try

```
> simplify(Dirac(2*x+3*y));
```

you'll always get an incorrect answer, but sometimes it will have a factor 1/2 and sometimes 1/3.