Solving the 4 bugs on corners of square or rectangle problem

Nasser M. Abbasi

Fall, 2018 Compiled on September 23, 2024 at 1:35am

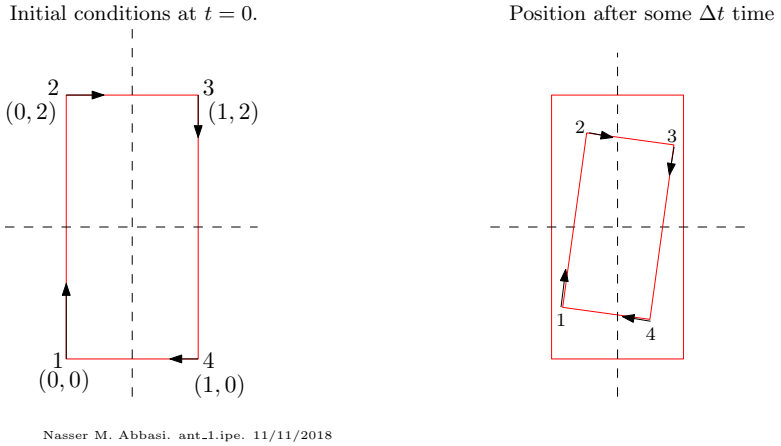
1 Problem description

Four bugs on corners of a rectangle are standing at positions $(0,0),(0,2),(1,2),(1,0)$. Bug 1 starts to move directly towards bug 2, while bug 2 moves directly towards bug 3 and bug 3 moves directly towards bug 4 and bug 4 moves directly towards bug 1. All are moving with unit speed. Motion is clockwise. Find the equations of motions and make animation of the motion.

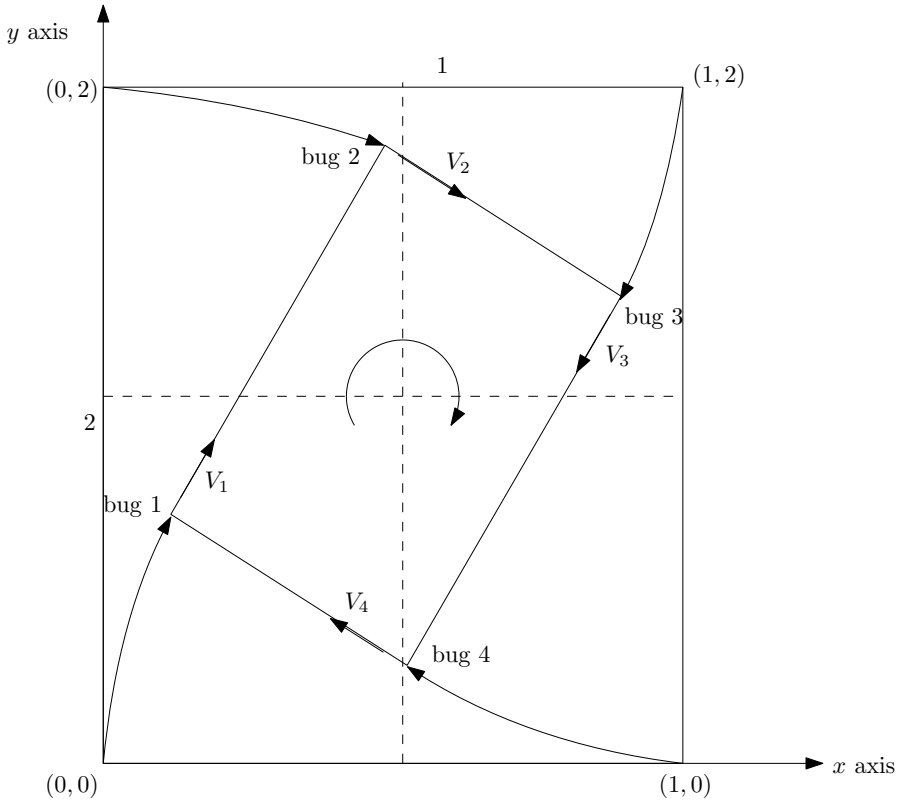
2 Solution

2.1 Analysis of motion

The following diagram shows the initial positions of all four bugs and what happens after Δt has elapsed.



The four bugs are initially located at corners of rectangle. The width is $h = 1$ and the height is $L = 2$. Since each bug moves with the same speed toward the bug next to it (clockwise), then by symmetry, the four bugs will remain on the corners of a parallelogram as time passes, but the parallelogram will become smaller and rotate clockwise as the bugs spiral towards the center of the original rectangle where they all meet. The following diagram illustrates such motion after some Δt has elapsed.

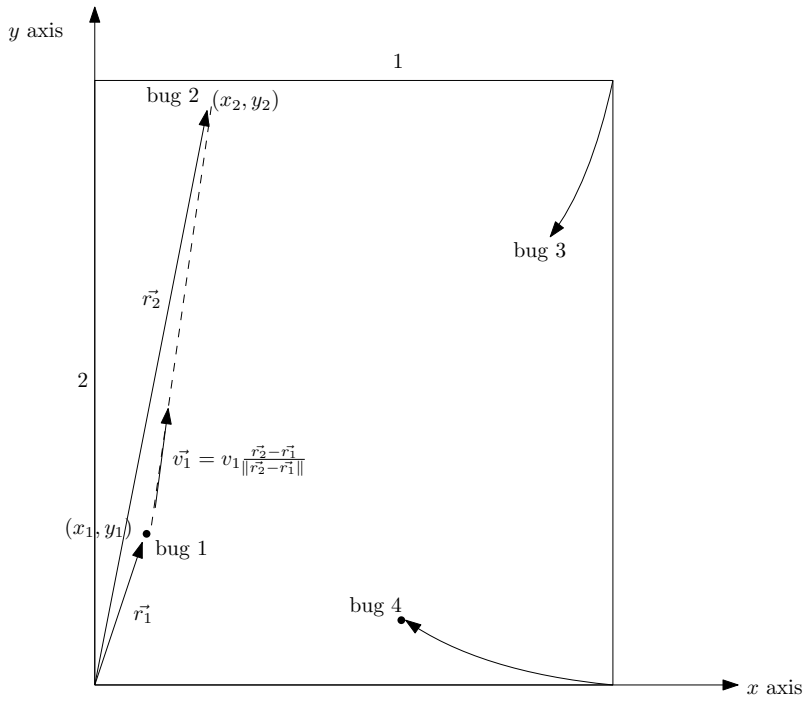


Showing locations of bugs after some Δt.
Rectangle is rotating clockwise and rotating in time. (drawing not to scale)

We see that at each instance of time, each bug remains at the corner of a scaled down parallelogram version of the original rectangle, which is rotating and the bug's velocity is straight toward the bug position it is chasing. What this means is that bug 1 motion is always at 90^0 to the path of bug 2. And bug 2 motion is at 90^0 to the path of bug 3 and so on.

2.2 Equations of motion

To obtain the equation of motion for each bug, we will look at each bug's position relative to the bug it is chasing. Starting with bug 1 relative to bug 2 with the help of the following diagram



Showing relative locations of bug 1 and 2 after some Δt

Nasser M. Abbasi. ant_2.ipe. 11/11/2018

The position vector of bug 1 is $\vec{r}_1(t)$ and the position vector of bug 2 is $\vec{r}_2(t)$. Therefore $\vec{v}_1 = \frac{d\vec{r}_1(t)}{dt} = |\vec{v}_1| \, \hat{r}$ where $\hat{r}$ is unit vector in the direction from bug 1 to bug 2. Hence

$$\frac{d\vec{r}_1(t)}{dt} = |\vec{v}_1| \, \frac{\vec{r}_2(t) - \vec{r}_1(t)}{\| \, \vec{r}_2 \, (t) - \vec{r}_1 \, (t) \, \|}$$

But since $|\vec{v}_1| = 1$ meter per seconds then

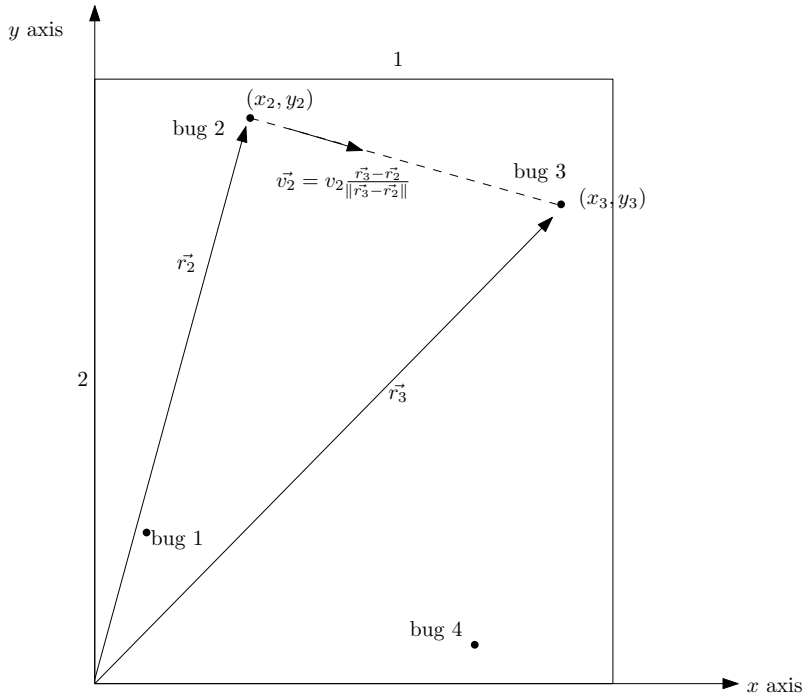
$$\frac{d\vec{r}_1(t)}{dt} = \frac{(x_2 \hat{\imath} + y_2 \hat{\jmath}) - (x_1 \hat{\imath} + y_1 \hat{\jmath})}{\| \, (x_2 \hat{\imath} + y_2 \hat{\jmath}) - (x_1 \hat{\imath} + y_1 \hat{\jmath}) \, \|}$$

$$\left(\frac{dx_1}{dt} \hat{\imath} + \frac{dy_1}{dt} \hat{\jmath} \right) = \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \hat{\imath} + \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \hat{\jmath}$$

Where x_1, y_1 are the coordinates of bug 1 and x_2, y_2 are the coordinates of bug 2. From the above, we obtain equations of motion for bug 1, where now we use $x_1' = \frac{dx}{dt}$ and $y_1' = \frac{dy}{dt}$ for bug 1.

$$\begin{aligned} x_1' &= \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\ y_1' &= \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \end{aligned} \tag{3}$$

The same analysis is now done to obtain $x_2'(t)$ and $y_2'(t)$ expressions similar to (3) above for bug 2.



Showing relative location of bugs 2 and 3 after some Δt

Nasser M. Abbasi. ant_3.ipe. 11/11/2018

The position vector of bug 2 is $\vec{r}_2(t)$ and the position vector of bug 3 is $\vec{r}_3(t)$. Therefore $\vec{v}_2 = \frac{d\vec{r}_2(t)}{dt} = |\vec{v}_2|\,\hat{r}$ where $\hat{r}$ is unit vector in the direction from bug 2 to bug 3. Hence

$$\frac{d\vec{r}_2(t)}{dt} = |\vec{v}_2|\,\frac{\vec{r}_3(t) - \vec{r}_2(t)}{\| \vec{r}_3(t) - \vec{r}_2(t) \|}$$

Since $|\vec{v}_2| = 1$ meter per seconds then

$$\begin{aligned}
\frac{d\vec{r}_2(t)}{dt} &= \frac{(x_3\hat{\imath} + y_2\hat{\jmath}) - (x_3\hat{\imath} + y_2\hat{\jmath})}{\parallel (x_3\hat{\imath} + y_2\hat{\jmath}) - (x_3\hat{\imath} + y_2\hat{\jmath}) \parallel} \\
\left(\frac{dx_2}{dt}\hat{\imath} + \frac{dy_2}{dt}\hat{\jmath} \right) &= \frac{x_3 - x_2}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}}\hat{\imath} + \frac{y_3 - y_2}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}}\hat{\jmath}
\end{aligned}$$

Where x_2, y_2 are the coordinates of bug 2 and x_3, y_3 are the coordinates of bug 3. The above gives the two equations of motion for bug 2, where now we use $x_2' = \frac{dx_2}{dt}$ and $y_2' = \frac{dy_2}{dt}$ for bug 2.

$$\begin{aligned}
x_2' &= \frac{x_3 - x_2}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}} \\
y_2' &= \frac{y_3 - y_2}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}}
\end{aligned} \tag{3}$$

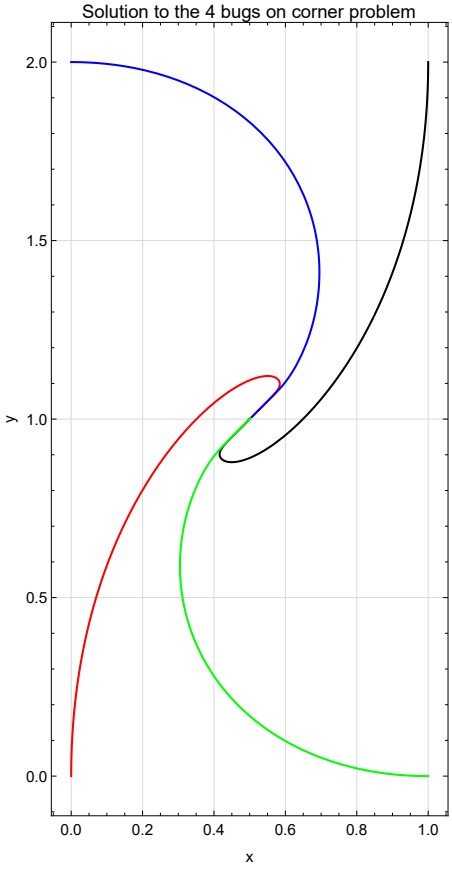
By doing the above again for bug 3 and bug 4, it is clear the result will be similar. The final equations of motions in vector form are

$$\begin{aligned}
\mathbf{x}' &= f(\mathbf{x}) \\
&\begin{pmatrix}
x_1'(t) = \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\
y_1'(t) = \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\
x_2'(t) = \frac{x_3 - x_2}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}} \\
y_2'(t) = \frac{y_3 - y_2}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}} \\
x_3'(t) = \frac{x_4 - x_3}{\sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2}} \\
y_3'(t) = \frac{y_4 - y_3}{\sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2}} \\
x_4'(t) = \frac{x_1 - x_4}{\sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2}} \\
y_4'(t) = \frac{y_1 - y_4}{\sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2}}
\end{pmatrix}
\end{aligned}$$

With initial conditions

$$\mathbf{x}(0) = \begin{pmatrix} x_1(0) \\ y_1(0) \\ x_2(0) \\ y_2(0) \\ x_3(0) \\ y_3(0) \\ x_4(0) \\ y_4(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 1 \\ 2 \\ 1 \\ 0 \end{pmatrix}$$

We can not write this as $\mathbf{x}' = A\mathbf{x}$ since it is not linear system. These ODE's have to solved numerically since they are nonlinear. The following is the result of running the solution for 1.5 seconds with the code listing below it. This shows the bug spiraling down to the center of the original rectangle as expected.

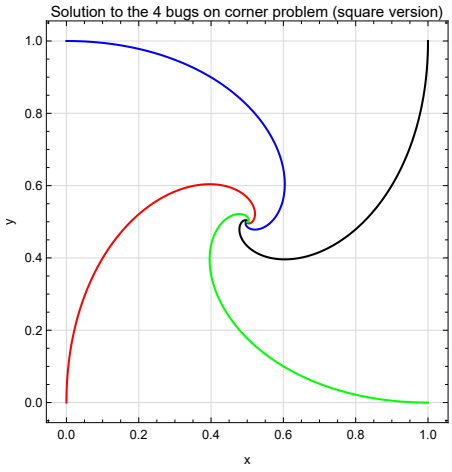


This problem was also solved for a square instead of a rectangle. The only change

needed was to make the initial conditions as follows

$$\mathbf{x}(0) = \begin{pmatrix} x_1(0) \\ y_1(0) \\ x_2(0) \\ y_2(0) \\ x_3(0) \\ y_3(0) \\ x_4(0) \\ y_4(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

The time needed to reach the center in this case is one second.



2.3 Animations

These animations will play in HTML only

Code for animation is

```
ClearAll[t, x1, x2, x3, x4, y1, y2, y3, y4];
ode1 = x1'[t] == (x2[t] - x1[t])/
   Sqrt[(x2[t] - x1[t])^2 + (y2[t] - y1[t])^2];
ode2 = y1'[t] == (y2[t] - y1[t])/
   Sqrt[(x2[t] - x1[t])^2 + (y2[t] - y1[t])^2];
ode3 = x2'[t] == (x3[t] - x2[t])/
   Sqrt[(x3[t] - x2[t])^2 + (y3[t] - y2[t])^2];
ode4 = y2'[t] == (y3[t] - y2[t])/
   Sqrt[(x3[t] - x2[t])^2 + (y3[t] - y2[t])^2];
```

```
ode5 = x3'[t] == (x4[t] - x3[t])/
   Sqrt[(x4[t] - x3[t])^2 + (y4[t] - y3[t])^2];
ode6 = y3'[t] == (y4[t] - y3[t])/
   Sqrt[(x4[t] - x3[t])^2 + (y4[t] - y3[t])^2];
ode7 = x4'[t] == (x1[t] - x4[t])/
   Sqrt[(x1[t] - x4[t])^2 + (y1[t] - y4[t])^2];
ode8 = y4'[t] == (y1[t] - y4[t])/
   Sqrt[(x1[t] - x4[t])^2 + (y1[t] - y4[t])^2];

sol = NDSolve[{ode1, ode2, ode3, ode4, ode5, ode6, ode7, ode8,
    x1[0] == 0, y1[0] == 0, x2[0] == 0, y2[0] == 1, x3[0] == 1,
    y3[0] == 1, x4[0] == 1, y4[0] == 0}, {x1[t], y1[t], x2[t], y2[t],
    x3[t], y3[t], x4[t], y4[t]}, {t, 0, 1}];

Manipulate[
 p1 = ParametricPlot[{x1[t], y1[t]} /. sol, {t, 0, tMax},
    AxesOrigin -> {0, 0}, GridLines -> Automatic,
    GridLinesStyle -> LightGray, Frame -> True,
    FrameLabel -> {{"y", None}, {"x", None}}, ImageSize -> 350,
    PlotStyle -> Red] /. Line[x_] :> {Arrowheads[.04], Arrow[x]};

 p2 = ParametricPlot[{x2[t], y2[t]} /. sol, {t, 0, tMax},
    AxesOrigin -> {0, 0}, GridLines -> Automatic,
    GridLinesStyle -> LightGray, Frame -> True,
    FrameLabel -> {{"y", None}, {"x", None}}, ImageSize -> 350,
    PlotStyle -> Blue] /. Line[x_] :> {Arrowheads[.04], Arrow[x]};

 p3 = ParametricPlot[{x3[t], y3[t]} /. sol, {t, 0, tMax},
    AxesOrigin -> {0, 0}, GridLines -> Automatic,
    GridLinesStyle -> LightGray, Frame -> True,
    FrameLabel -> {{"y", None}, {"x", None}}, ImageSize -> 350,
    PlotStyle -> Black] /. Line[x_] :> {Arrowheads[.04], Arrow[x]};

 p4 = ParametricPlot[{x4[t], y4[t]} /. sol, {t, 0, tMax},
    AxesOrigin -> {0, 0}, GridLines -> Automatic,
    GridLinesStyle -> LightGray, Frame -> True,
    FrameLabel -> {{"y", None}, {"x", None}}, ImageSize -> 350,
    PlotStyle -> Green] /. Line[x_] :> {Arrowheads[.04], Arrow[x]};

 Show[p1, p2, p3, p4, PlotRange -> {{0, 1}, {0, 1}},
  PlotLabel ->
```

```
  Column[{"Solution to the 4 bugs on corner problem",
    "On (0,0),(1,1) square"}], BaseStyle -> 12, ImageSize -> 300],

{{tMax, .01, "time"}, .01, 1, .01, Appearance -> "Labeled"}
]
```