```
(*simple example showing how to use Manipulate to animate the solution of
 the Poisson 2D PDE using the Jacobi method. Nasser M. Abbasi feb 18,2012*)

Manipulate[

 Module[{h, u, grid, force, f},
  h = 1 / (n - 1);
  u = Table[0, {n}, {n}];
  grid = makeCoordinates[n];
  f[x_, y_] := -Exp[-(x - 0.25)^2 - (y - 0.6)^2]; (*the force function*)
  force = Map[f[#[[1]], #[[2]]] &, grid, {2}]; (*evaluate it on the grid*)

  Do[ (*solve*)
   jacobiStep[Unevaluated@u, h, force], {n}
  ];

  (*plot the solution*)
  ListPlot3D[u, PlotRange → All,
   PerformanceGoal → "Quality", ImagePadding → 30, ImageSize → 400,
   Mesh → {n, n}, PlotLabel → Text@Row[{"solution to ", Style[TraditionalForm[
         HoldForm[-Power["∇", 2] "u"[x, y] == -Exp[-(x - 0.25)^2 - (y - 0.6)^2]]]]}]
  ]
 ],

 {{n, 9, "number of steps"}, 5, 31, 1, Appearance → "Labeled"},
 TrackedSymbols :→ {n},
 Initialization :→
  {
   jacobiStep[u_, h_, f_] := Module[{nRow, nCol, i, j},
     {nRow, nCol} = Dimensions[u];
     For[i = 2, i < nRow, i++,
      For[j = 2, j < nCol, j++,
       u[[i, j]] =
         (1 / 4) (u[[i - 1, j]] + u[[i + 1, j]] + u[[i, j - 1]] + u[[i, j + 1]] - h^2 f[[i, j]])
       ]
      ]
    ];

   makeCoordinates[n_] := Module[{i, j, h},
     h = 1 / (n - 1);
     N@Table[{i * h, j * h}, {i, 0, n - 1}, {j, 0, n - 1}]
    ];

  }
]
```

number of steps 9

solution to $-\nabla^2 u(x, y) = -\exp(-(x - 0.25)^2 - (y - 0.6)^2)$