Numerical solutions of initial value problems with Matlab

© 2015, Yonatan Katznelson

In this note I’ll discuss using MATLAB to calculate numerical approximations to the solutions of first order initial value problems of the form

\[ y' = f(t,y); \quad y(t_0) = y_0. \]

Specifically, this note will cover MATLAB’s built-in solver, ode45, as well as simple implementations of Euler’s method and the improved Euler method.

1. ode45

The built-in function ode45 is a sophisticated numerical solver that calculates the slope at each step of the approximation by taking a weighted average of several slopes, and furthermore chooses variable step sizes to improve the approximation. Technically, ode45 is a variable step Runge-Kutta algorithm, as described in Section 8.3 of the textbook.

Assuming that the function \( f = f(t,y) \) is defined (in MATLAB),† then the command

\[
\texttt{>> ode45(}f, [t0 t1], y0)\]

generates the plot of a numerical approximation to the initial value problem

\[ y' = f(t,y); \quad y(t_0) = y_0 \]

on the interval \([t_0,t_1]\).

A plot is the default output of ode45. If you want to look at the actual values, then you use the command

\[
\texttt{>> [t y]=ode45(}f, [t0 t1], y0);\]

which stores the points \( t_j \) where the approximations were computed in the vector \( t \),‡ and stores the corresponding approximate values \( y_j \) in the vector \( y \). Having done this, you can use the command

\[
\texttt{>> y(end)}\]

to see the last entry in the vector \( y \), for example, which is the value of the (approximate) solution at the right-hand endpoint (\( t_1 \)) of the given interval.

Example 1. Consider the initial value problem

\[
y' = \frac{t^2 + 1}{y + 2}, \quad y(0) = 1. \tag{1.1}\]

The pair of commands

\[
\texttt{>> f=@(t,y) (t.^2+1)./(y+2);} \\
\texttt{>> ode45(}f, [0 3],1);\]

generates the plot in Figure 1

On the other hand, if we want to compute the approximate value of the solution to this problem at \( t = 3 \), we can use the sequence of commands

\[
\texttt{>> f=@(t,y) (t.^2+1)./(y+2);} \\
\texttt{>> f(3,1)}\]

†How one defines a function in MATLAB is described in the first MATLAB note.
‡Remember, these points are not (necessarily) evenly spaced.
Figure 1: A basic ode45 plot

```matlab
>> [t y]=ode45(f,[0 3],1);
>> y(end)
```
which produces the output
```
ans =
    3.7446.
```
If you also want to see the plot of the approximate solution produced by this sequence of commands you can simply type

```matlab
>> plot(t,y)
```
This produces the plot, shown in Figure 2, without the little circles that appear in the default `ode45` output.

The explicit solution of the initial value problem in Example 1 is given by

$$y = -2 + \sqrt{\frac{2}{3}t^3 + 2t + 9},$$

as you can verify. The value of this function at $t = 3$ is $y(3) = -2 + \sqrt{33} \approx 3.7446$, so `ode45` does a good job in this case.

\[\square\]

2. **Euler’s method**

Recall that Euler’s method for approximating the solution of the initial value problem

$$y' = f(t, y), \quad y(t_0) = y_0$$

(2.1)

uses the ‘tangent-line’ approximation

$$y(t + h) \approx y(t) + y'(t) \cdot h.$$  \hspace{1cm} (2.2)

To compute the approximate value of $y(t_f)$, the approximation in (2.2) is used repeatedly in small steps to get from $t_0$ to $t_f$, and uses $f(t, y)$ to compute $y'(t)$. The algorithm is simple and short:
Choose the step size $h$. In practice, we usually choose the number of steps, $n$, and set $h = (t_f - t_0)/n$.

2. For $0 \leq j \leq n$, let $t_j = t_0 + j \cdot h$.

3. For $1 \leq j \leq n$, let $y_j = y_{j-1} + h \cdot f(t_{j-1}, y_{j-1})$.

The final value in the sequence of $y_j$'s is the approximate value we seek. I.e., if $y = \phi(t)$ is the solution of the initial value problem in (2.1), then $y_n$ is the approximation that Euler’s method produces for $\phi(t_f)$.

This algorithm is also easy to implement in MATLAB. The mfile in Figure 3 defines the function `euler` whose input is the function $f(t, y)$, the initial and final points, $t_0$ and $t_f$, of the interval on which you want to approximate the solution of the differential equation, the initial value $y_0$ and the number of steps $n$. The output is two vectors: the vector of $t_j$'s where the approximate values are computed, and the vector of $y_j$'s are the corresponding approximate values.

**Example 2.** I’ll use the function `euler` to find an approximate solution of the initial value problem in Example 1. The sequence of commands

```matlab
>> f=@(t,y) (t.^2+1)./(y+2);
>> [t y]=euler(f,0,3,1,10);
>> y(end)
```

produces the output

```
ans =
  3.5892
```

which isn’t as good as the approximation produced with `ode45`, but isn’t bad for only 10 steps (and step size $h = 0.3$). Increasing the number of steps to $n = 50$, yields the approximate value $y_{50} = 3.7138$, which is better, and increasing the step size to $n = 1000$ produces the approximation $y_{1000} = 3.7430$, which is even better, but still not as good as the one that `ode45` generated.

It is worth noting that `ode45` used only 40 steps to generate its output. On the other hand, each step in the algorithm used by `ode45` involves many more sub-steps than the single computation in Euler’s method.
function [t y]=euler(func,t0,t1,y0,num)

% This function produces approximate values of the solution y=g(t) of the
% initial value problem y’=func(t,y), y(t0)=y0, on the interval [t0, t1]
% using Euler's method with 'num' steps. The output are the num+1 points,
% (t_j,y_j), corresponding to the approximate values.
% determines the step size:
h=(t1-t0)/num;

% initiate the vector of t values
 t=t0:h:t1;
% initiates the vector of y values
 y=zeros(1,num+1);
y(1)=y0;
% Euler's "tangent-line" method...
 for j=1:num
     y(j+1)=y(j)+h*func(t(j),y(j));
 end;

Figure 3: MATLAB function euler.

The output of the function euler, may also be plotted in a graph, using the command plot(t,y). The plot in Figure 4 includes the graph of the 10 step Euler’s method approximation (blue), the graph of the ode45 approximation (red), and green asterisks that plot the values of the (true) solution $y = -2 + \sqrt{2t^3/3 + 2t + 9}$. As you can see, ode45 produces an approximation that is indistinguishable from the true solution, but even the 10-step Euler’s method yields a fairly good plot.

Figure 4: Plots of Euler method approximation, ode45 approximation and true values.