Math 277
Ordinary Differential Equations

Dr. Constant J. Goutziers
Department of Mathematics, Computer Science and Statistics
goutzicj@oneonta.edu

Lesson 3

Euler's Method

Initializations

```
> restart;
  with(plots):
```

3.1 A Simple Numerical Technique, Euler's Method

A first-order differential equation defines $\frac{dy}{dt}$ as a function of t and y.

$$\frac{dy}{dt} = f(t, y)$$

Using the first two terms of the Taylor expansion of y, we can write for small values of h

$$y(t+h) \approx y(t) + hf(t, y(t))$$

leading to the iteration scheme

$$t_{n+1} = t_n + h$$

$$y_{n+1} = y_n + hf(t_n, y_n)$$

This process is known as Euler's method. The parameter h is called the stepsize.

Examples

Example 3.1.1

i) Use Euler's method to solve the initial value problem

$$\frac{dy}{dt} = -2\,t\,y^2 \qquad y(0) = 1$$

on the interval $[0, 2]$. Use a stepsize $h = 0.1$.

ii) Plot your result and compare it with the exact solution.

Solution

i) Use Euler's method to solve the initial value problem

$$\frac{dy}{dt} = -2\,t\,y^2 \qquad y(0) = 1$$

on the interval $[0, 2]$. Use a stepsize $h = 0.1$.

Code $f(t, y) = -2\,t\,y^2$, the stepsize h, and the starting values t_0 and y_0, then iterate. Because the variables t and y are used so frequently, it is good programming practice to use a different variable name for the iterates. We will use T and Y.

```
> f:=(t, y)->-2*t*y^2;
  h:=0.1;
```

```
T[0]:=0.0;
Y[0]:=1.0;
```

$$f := (t, y) \rightarrow -2\,t\,y^2$$
$$h := 0.1$$
$$T_0 := 0.$$
$$Y_0 := 1.0 \qquad \textbf{(2.1.1.1)}$$

```
> for k from 0 to 20 do
  T[k+1]:=T[k]+h;
  Y[k+1]:=Y[k]+h*f(T[k], Y[k]);
  print(evalf([k, T[k], Y[k]], 4));
  od:
```

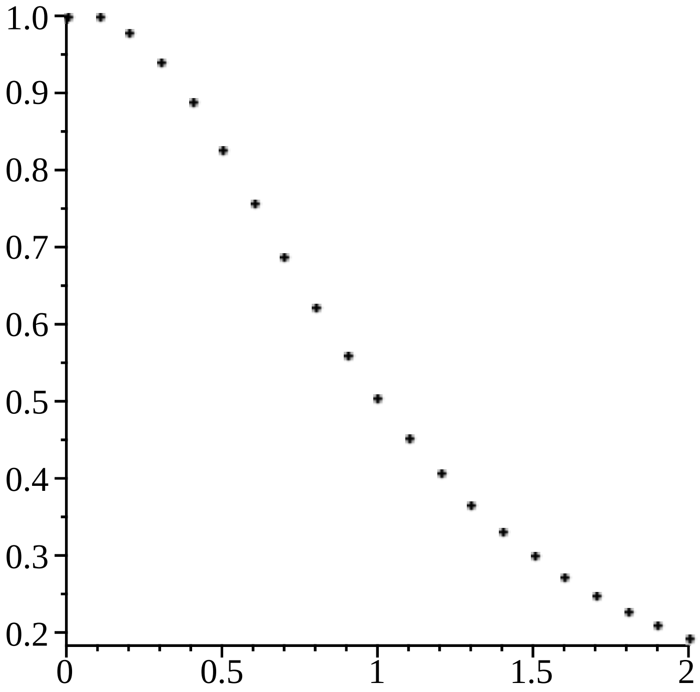
```
[0., 0., 1.0]
[1., 0.1, 1.0]
[2., 0.2, 0.9800]
[3., 0.3, 0.9416]
[4., 0.4, 0.8884]
[5., 0.5, 0.8253]
[6., 0.6, 0.7571]
[7., 0.7, 0.6884]
[8., 0.8, 0.6220]
[9., 0.9, 0.5601]
[10., 1.0, 0.5036]
[11., 1.1, 0.4529]
[12., 1.2, 0.4078]
[13., 1.3, 0.3679]
[14., 1.4, 0.3327]
[15., 1.5, 0.3017]
[16., 1.6, 0.2744]
[17., 1.7, 0.2503]
[18., 1.8, 0.2290]
[19., 1.9, 0.2101]
[20., 2.0, 0.1933]                    (2.1.1.2)
```

ii) Plot your result and compare it with the exact solution.
First, generate the list of points (t_k, y_k), $k = 0, \dots, 20$, then use the **pointplot** routine for visualization.

```
> points:=[seq([T[k], Y[k]], k=0..20)];
```

```
points := [[0., 1.0], [0.1, 1.0], [0.2, 0.9800], [0.3, 0.9415840000], [0.4,          (2.1.1.3)
    0.8883891743], [0.5, 0.8252503483], [0.6, 0.7571465346], [0.7,
    0.6883540296], [0.8, 0.6220176518], [0.9, 0.5601126983], [1.0,
    0.5036419760], [1.1, 0.4529109280], [1.2, 0.4077827001], [1.3,
    0.3678738848], [1.4, 0.3326877741], [1.5, 0.3016970507], [1.6,
    0.2743907176], [1.7, 0.2502978325], [1.8, 0.2289971708], [1.9,
    0.2101188773], [2.0, 0.1933418991]]
```

```
> pointplot(points);
```



The exact solution to this initial value problem is given by

$$y = \frac{1}{1+t^2}$$

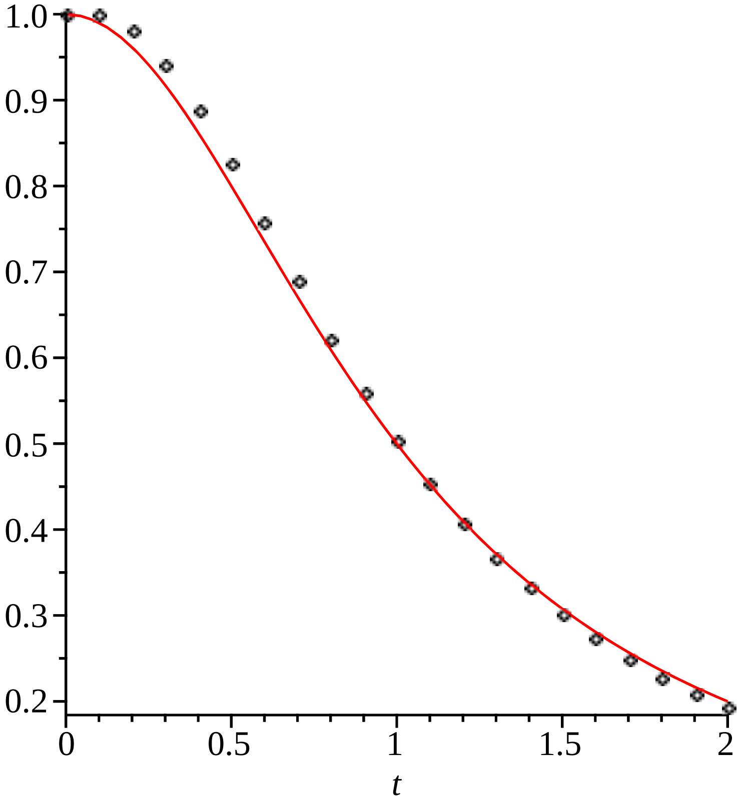
which can be verified by direct substitution.

```
> sol:=1/(1+t^2);
```

$$sol := \frac{1}{1+t^2} \qquad (2.1.1.4)$$

Now, in one picture, we display the exact solution and the points generated by Euler's method.

```
> p1:=pointplot(points):
  p2:=plot(sol, t=0..2):
  display([p1, p2]);
```



Not bad for a first attempt. Later we will introduce more sophisticated methods which provide even better results.