Euler’s Method in Maple
MA 341S, Spring, 2002

First here is the Euler script for the initial value problem $y' = y$, $y(0) = 1$, with $h = 0.1$ and 10 steps to approximate $y(1)$, which should be 2.718281828.

```maple
> x:=0;
> y:=1;
> deltax:=0.1;
> for i to 10
> do
>   slope:= y;
>   x := x + deltax;
>   y := y + deltax*slope;
> end do;

x := 0
y := 1
deltax := .1
slope := 1
x := .1
y := 1.1
slope := 1.1
x := .2
y := 1.21
slope := 1.21
x := .3
y := 1.331
slope := 1.331
x := .4
y := 1.4641
slope := 1.4641
x := .5
y := 1.61051
slope := 1.61051
x := .6
y := 1.771561
slope := 1.771561
x := .7
y := 1.9487171
slope := 1.9487171
x := .8
y := 2.1435881
```
\[ \text{slope} := 2.14358881 \]
\[ x := .9 \]
\[ y := 2.357947691 \]
\[ \text{slope} := 2.357947691 \]
\[ x := 1.0 \]
\[ y := 2.593742460 \]

Now let's write a Maple procedure to calculate with. The procedure will have as input \( x_{\text{start}}, x_{\text{end}}, y_{\text{start}}, \text{step size}, \text{the function}. \)

The value of a Maple procedure is the value of the last command executed inside the procedure. For this reason I will include something extra here, namely a list of the points calculated and call the list \text{newpoint}. The value of newpoint[0] will be the starting value, and it is set in the 7th line of the procedure. The last line just before end proc is a loop that prints out the values of newpoint. Since its the last thing executed when the procedure is called, it will be the value of the procedure.

\[
\text{euler_method} := \text{proc}(x_{\text{start}}, x_{\text{end}}, y_{\text{start}}, \text{step size}, \text{the function}) \\
\text{local } N, x, y, f, j, \text{slope}, \text{newpoint}, k; \\
N := (x_{\text{end}} - x_{\text{start}})/\text{step size}; \\
x := x_{\text{start}}; \\
y := y_{\text{start}}; \\
f := \text{the function}; \\
\text{newpoint}[0] := (x, y); \\
\text{for } j \text{ to } N \\
\text{ do} \\
\text{slope} := f(x, y); \\
x := x + \text{deltax}; \\
y := y + \text{deltax} \times \text{slope}; \\
\text{newpoint}[j] := (x, y); \\
\text{end do}; \\
\text{for } k \text{ from } 0 \text{ to } N \\
\text{ do} \\
\text{print(newpoint}[k]); \\
\text{end do}; \\
\text{end proc};
\]
The function `euler_method` is defined as follows:

```plaintext
euler_method := proc(x_start, x_end, y_start, step_size, the_function)
local N, x, y, f, j, slope, newpoint, k;
N := (x_end - x_start)/step_size;
x := x_start;
y := y_start;
f := the_function;
newpoint[0] := x, y;
for j to N do
    slope := f(x, y); x := x + deltax; y := y + deltax * slope; newpoint[j] := x, y
end do;
for k from 0 to N do print(newpoint[k]) end do
end proc
```

The call to `euler_method(0,1,1,0.1,(a,b)-> b)` produces the following output:

```
0, 1
 .1, 1.1
 .2, 1.21
 .3, 1.331
 .4, 1.4641
 .5, 1.61051
 .6, 1.771561
 .7, 1.9487171
 .8, 2.14358881
 .9, 2.357947691
1.0, 2.593742460
```