Improved Euler algorithm example

This Maple document, and the mirror Matlab document, have equivalent code for solving initial value problems using Improved Euler's method. By comparing the code you can see some of the main syntax differences between Maple and Matlab. In all examples, we are approximating the solution to the IVP for $y(x)$,

$$y'(x) = y$$
$$y(0) = 1$$

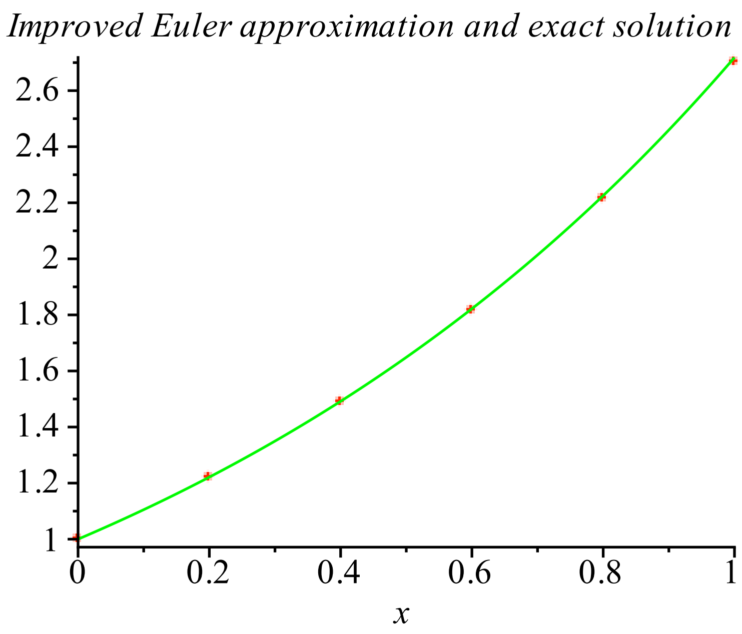
which has exact solution $y(x) = e^x$. You can execute commands successively by placing your cursor in the first math field and then hitting <enter> repeatedly to move through the worksheet. Or, you can use the Edit/Execute/Worksheet option at the top of the Maple Window. (And, you can also remove all output from your worksheet using the Edit/Remove Output/From Worksheet option.)

```
> restart: #clear all memory
> Digits := 6: #decide how many digits to use in numerical computations

> N := 5:   # number of steps (subintervals)
  x0 := 0.:  #choose initial x-value
  xf := 1.0: #final x-value  Note that xf-x0 must be divisible by h in order to end exactly at xf
  h := (xf - x0)/N :# step size
>

> X[0] := 0:  #We choose to use a vector "X" to hold x-values, and this is the initial one
  Y[0] := 1:
      #We choose to use a vector "Y" to hold y-values, and this is the initial one for our IVP
> frhs := (x, y) -> y:  # slope function f(x,y) for our example


> for i from 0 to N - 1 do     #Improved Euler loop
    k1 := frhs(X[i], Y[i]): #left slope estimate
    k2 := frhs(X[i] + h, Y[i] + h·k1): #right slope estimate
    k := (k1 + k2)/2 :  # "trapezoid rule" average slope estimate
    X[i + 1] := X[i] + h:
    Y[i + 1] := Y[i] + h·k
  end do:
>
> for i from 0 to N do      #to print out results
    print(i, X[i], Y[i]);
  end do:
```

```
                    0, 0, 1
              1, 0.200000, 1.22000
              2, 0.400000, 1.48840
```

```
3, 0.600000, 1.81585
4, 0.800000, 2.21534
5, 1.00000, 2.70271                                                    (1)
```

```
> Yexact := x -> e^x :   #exact solution for this IVP
> with(plots) :   #load plotting package
  plot1 := pointplot({seq([X[i], Y[i]], i = 0 ..N)}, color = red) : #Euler points
  plot2 := plot(Yexact(x), x = 0 ..1, color = green) : #exact solution graph
  display({plot1, plot2}, title = `Improved Euler approximation and exact solution`);
```



```
> Error := abs(Yexact(xf) - Y[N]);   #absolute error at final x-value
```
$$RelativeError := \frac{Error}{\text{abs}(Yexact(xf))};\quad \#relative\ error$$

```
Error := 0.01557
RelativeError := 0.00572789                                            (2)
```