IMPLEMENTATION OF NUMERICAL METHODS

Note that particular versions of Maple and Mathematica packages may have slightly different commands than described here. Always consult help topics for more information. Algorithms will be illustrating on the following initial value problem

$$\frac{dy}{dx} = x + y, \quad y(0) = 1$$

on the interval $[0, 1]$.

1. Implementing Euler's Method

With Maple,

```
f := (x, y) → x + y;          # defines the function f(x,y)
x0 := 0 :  y0 := 1 :          # enters initial values
xf := 1 :                     # enters the final value of x
n := 10 :                     # desired number of steps, say n=10
h := evalf((xf − x0)/n);      # computes resulting step size
x := x0 :  y := y0 :          # initializing the values for x and y
for i from 1 to n do          # starting the loop
k := f(x, y) :                # the left-hand slope
y := y + h ∗ k :              # Euler step to update y
x := x + h                    # update x
print(x, y);                  # display current values
od :
```

If you only want to print the final endpoint result, remove the print command from the loop and put it right after the loop. To run the program use Execute Worksheet command from Edit menu.

Using Mathematica is similar.

```
f[x_, y_] := x + y
x0 = 0;  y0 = 1;
xf = 1;
n = 10;
h = (xf - x0)/n;
x = x0;  y = y0;
Do[k = f[x, y];
y = y + h * k;
x = x + h;
Print[x, "  ", y],
{i, 1, n}]
```

If you only want to print the final endpoint result, remove the print command from the loop and put it right after the loop.

2. Improved Euler Implementation

With Maple,

```
f := (x, y) → x + y;          # defines the function f(x,y)
x0 := 0 :  y0 := 1 :          # enters initial values
xf := 1 :                     # enters the final value of x
n := 10 :                     # desired number of steps, say n=10
h := evalf((xf - x0)/n);      # computes resulting step size
x := x0 :  y := y0 :          # initializing the values for x and y
for i from 1 to n do          # starting the loop
k1 := f(x, y) :               # the left-hand slope
k2 := f(x + h, y + h * k1) :  # the right-hand slope
k := (k1 + k2)/2 :            # the average slope
y := y + h * k :              # Euler step to update y
x := x + h                    # update x
print(x, y);                  # display current values
od :
```

With Mathematica,

$$
\begin{aligned}
&f[x\_, y\_] := x + y \\
&x0 = 0;\ y0 = 1; \\
&xf = 1; \\
&n = 10; \\
&h = (xf - x0)/n; \\
&x = x0;\ y = y0; \\
\mathrm{Do}[&k1 = f[x, y]; \\
&k2 = f[x + h, y + h * k1]; \\
&k = (k1 + k2)/2; \\
&y = y + h * k; \\
&x = x + h; \\
&\mathrm{Print}[x, \text{''}\ \ \text{''}, y], \\
&\{i, 1, n\}\,]
\end{aligned}
$$

3. Runge-Kutta Implementation

Using Maple,

```
f := (x, y) → x + y;                    # defines the function f(x,y)
x0 := 0 :  y0 := 1 :                    # enters initial values
xf := 1 :                               # enters the final value of x
n := 10 :                               # desired number of steps, say n=10
h := evalf((xf − x0)/n);                # computes resulting step size
x := x0 :  y := y0 :                    # initializing the values for x and y
for i from 1 to n do                    # starting the loop
k1 := f(x, y) :                         # the left-hand slope
k2 := f(x + h/2, y + h ∗ k1/2) :        # 1st midpoint slope
k3 := f(x + h/2, y + h ∗ k2/2) :        # 2nd midpoint slope
k4 := f(x + h, y + h ∗ k3) :            # the right-hand slope
k := (k1 + 2 ∗ k2 + 2 ∗ k3 + k4)/6 :    # the average slope
y := y + h ∗ k :                        # Euler step to update y
x := x + h                              # update x
print(x, y);                            # display current values
od :
```

With Mathematica,

$$
\begin{aligned}
& f[x\_, y\_] := x + y \\
& x0 = 0;\ y0 = 1; \\
& xf = 1; \\
& n = 10; \\
& h = (xf - x0)/n; \\
& x = x0;\ y = y0; \\
\mathrm{Do}[& k1 = f[x, y]; \\
& k2 = f[x + h/2, y + h * k1/2]; \\
& k3 = f[x + h/2, y + h * k2/2]; \\
& k4 = f[x + h, y + h * k3]; \\
& k = (k1 + 2k2 + 2k3 + k4)/6; \\
& y = y + h * k; \\
& x = x + h; \\
& \mathrm{Print}[x, \text{''}\ \ \text{''}, y], \\
& \{i, 1, n\}\,]
\end{aligned}
$$