2nd Order Runge Kutta Method

Arash Mafi
Assistant Professor
Department of Electrical Engineering and Computer Science
University of Wisconsin-Milwaukee
11/14/2009

Suppose we would like to numerically solve the following differential equation

$y'[x] == 3\, e^{-4x} - 2\, y[x]$ subject to the boundary condition $y[0] = 1$ using the 2nd order Runge Kutta method.

The following is a simple numerical algorithm in *Mathematica* to solve this differential equation and find $y[1]$ using steps of $dx = 0.1$.

We also compare the result of the Runge Kutta method with that of the *Mathematica* using the NDSolve command.

```
dx = 0.1;  x0 = 0; y0 = 1.0;
Print["x", "      ", "y[x]"];
Do[{
  k1 = -2 y0 + 3 Exp[-4 x0];
  x1 = x0 + dx;
  y1 = y0 + dx * k1;
  k2 = -2 y1 + 3 Exp[-4 x1];
  y0 = y0 + 0.5 (k1 + k2) * dx;
  x0 = x1;
  Print[x0, "    ", y0];
 }, {i, 1, 10}]
```

```
x     y[x]

0.1   1.04055

0.2   1.00109

0.3   0.91999

0.4   0.82082

0.5   0.7176

0.6   0.61828

0.7   0.526997

0.8   0.445549

0.9   0.37434

1.    0.312985
```

```
s = NDSolve[{y'[x] == 3 e^-4x - 2 y[x], y[0] == 1}, y, {x, 0, 1}];
Evaluate[y[1] /. s]
```

```
{0.310865}
```