The 4th -order Runge-Kutta method for a system of ODEs

By Gilberto E. Urroz, Ph.D., P.E.
January 2010

Problem description

Consider the case of a system of two first-order ODEs given by:

$$\frac{dy_1}{dx} = f1\left(x, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right) = f1(x, y)$$

$$\frac{dy_2}{dx} = f2\left(x, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right) = f2(x, y)$$

subject to the initial conditions:

$$ys_1 = y_1(xs_1) \quad \text{and} \quad ys_2 = y_2(xs_2)$$

This system of equations can be re-written as a single ODE in which y and f are column vectors, i.e.,

$$\frac{dy}{dx} = f(x, y) \quad , \text{ with } \quad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad \text{and} \quad f(x, y) = \begin{pmatrix} f1(x, y) \\ f2(x, y) \end{pmatrix}$$

The initial conditions are given by the vector: $ys = \begin{pmatrix} ys1 \\ ys2 \end{pmatrix}$

Once the system of equations is written as a single ODE, the Runge-Kutta algorithms presented for a single ODE can be used to solve the equation. This illustrated in the following example.

Example

Solve the system of first-order ODEs:

$$\frac{dy_1}{dx} = \sin(x) + \cos(y_1) + \sin(y_2)$$

$$\frac{dy_2}{dx} = \cos(x) + \sin(y_2)$$

Subject to the initial conditions:

$$y1(0) = -1 \qquad \text{and} \qquad y2(0) = 1$$

Solve the ODEs in the interval: $0 \le x \le 20$ using 100 intervals.

Solution (version 1):

First, define the vector function f(x,y):

$$f(x, y) := \begin{pmatrix} \sin(x) + \cos(y_1) + \sin(y_2) \\ \cos(x) + \sin(y_2) \end{pmatrix}$$

The initial conditions are: $xs := 0$ $\quad ys := \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

The end of the solution interval is: $xe := 20$

Use 100 intervals: $n := 100$

Calculate the increment size, Δx:

$\Delta x := \mathrm{eval}\left(\frac{xe - xs}{n}\right)$ $\quad \Delta x = 0.2$

Create the x solution vector: $xsol := \mathrm{eval}(xs, xs + \Delta x .. xe)$

The y-solution vector gets initialized as follows:

$ysol := ys$ $\quad ysol = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

The following "for" loop calculates the Runge-Kutta algorithm (version 1) to produce the solution:

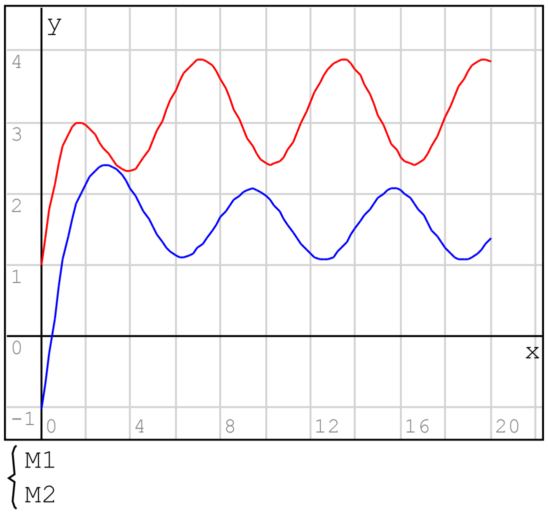
```
for $k \in 1..n$
    $x0 := \mathrm{eval}(xsol_k)$
    $y0 := \mathrm{eval}(\mathrm{col}(ysol, k))$
    $xM := \mathrm{eval}\left(x0 + \frac{1}{2} \cdot \Delta x\right)$
    $K1 := \mathrm{eval}(\Delta x \cdot f(x0, y0))$
    $yM := \mathrm{eval}\left(y0 + \frac{1}{2} \cdot K1\right)$
    $K2 := \mathrm{eval}(\Delta x \cdot f(xM, yM))$
    $yM := \mathrm{eval}\left(y0 + \frac{1}{2} \cdot K2\right)$
    $K3 := \mathrm{eval}(\Delta x \cdot f(xM, yM))$
    $y1 := \mathrm{eval}(y0 + K3)$
    $x1 := \mathrm{eval}(xsol_{k+1})$
    $K4 := \mathrm{eval}(\Delta x \cdot f(x1, y1))$
    $y1 := \mathrm{eval}\left(y0 + \frac{1}{6} \cdot (K1 + 2 \cdot K2 + 2 \cdot K3 + K4)\right)$
    $ysol := \mathrm{augment}(ysol, y1)$
```

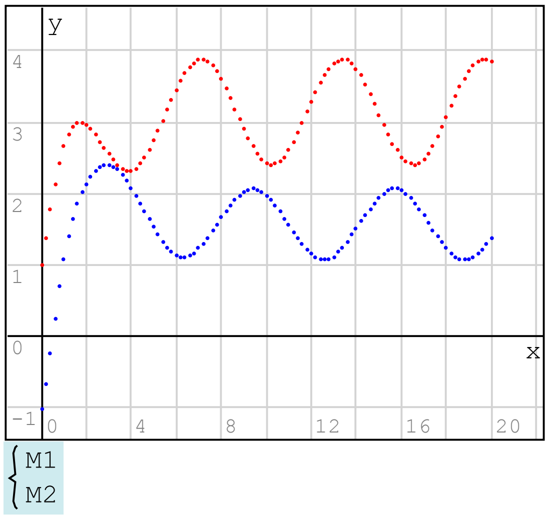
After completing the iterative process, the solution is stored in a row vector called "ysol". This vector can be transposed to put together the graph of the two solutions as illustrated here:

$ysol := ysol^T$

$M1 := \text{augment}(xsol, \text{col}(ysol, 1))$

$M2 := \text{augment}(xsol, \text{col}(ysol, 2))$





The plot to the left uses the "Graph by lines" option in the "Plot" palette, while the plot to the right uses the "Graph by points" option in the "Plot" palette.

Solution (version 2):

First, define the vector function f(x,y):

$$f(x, y) := \begin{pmatrix} \sin(x) + \cos(y_1) + \sin(y_2) \\ \cos(x) + \sin(y_2) \end{pmatrix}$$

The initial conditions are: $xs := 0$ $\quad ys := \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

The end of the solution interval is: $xe := 20$

Use 100 intervals: $n := 100$

Calculate the increment size, Δx:

$\Delta x := \text{eval}\left(\frac{xe - xs}{n}\right)$ $\quad \Delta x = 0.2$

Create the x solution vector: $xsol := \text{eval}(xs, xs + \Delta x .. xe)$

The y-solution vector gets initialized as follows:

$ysol := ys$ $\quad ysol = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

The following "for" loop calculates the Runge-Kutta algorithm (version 1) to produce the solution:

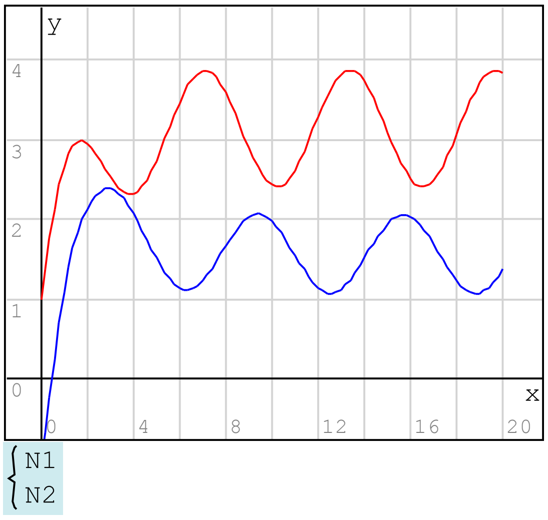
```
for k ∈ 1 .. n
    $x0 := \mathrm{eval}\left(xsol_k\right)$
    $y0 := \mathrm{eval}\left(\mathrm{col}(ysol, k)\right)$
    $x13 := \mathrm{eval}\left(x0 + \frac{1}{3}\cdot\Delta x\right)$
    $x23 := \mathrm{eval}\left(x0 + \frac{2}{3}\cdot\Delta x\right)$
    $K1 := \mathrm{eval}\left(\Delta x\cdot f(x0, y0)\right)$
    $y13 := \mathrm{eval}\left(y0 + \frac{1}{3}\cdot K1\right)$
    $K2 := \mathrm{eval}\left(\Delta x\cdot f(x13, y13)\right)$
    $y23 := \mathrm{eval}\left(y13 + \frac{1}{3}\cdot K2\right)$
    $K3 := \mathrm{eval}\left(\Delta x\cdot f(x23, y23)\right)$
    $y1 := \mathrm{eval}\left(y0 + K1 - K2 + K3\right)$
    $x1 := \mathrm{eval}\left(xsol_{k+1}\right)$
    $K4 := \mathrm{eval}\left(\Delta x\cdot f(x1, y1)\right)$
    $y1 := \mathrm{eval}\left(y0 + \frac{1}{8}\cdot(K1 + 3\cdot K2 + 3\cdot K3 + K4)\right)$
    $ysol := \mathrm{augment}(ysol, y1)$
```

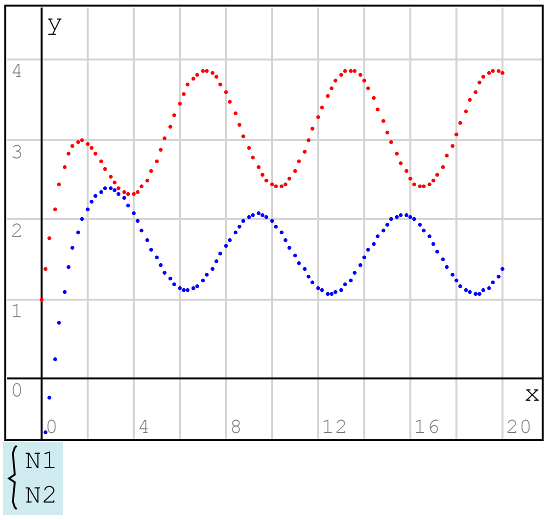
After completing the iterative process, the solution is stored in a row vector called "ysol". This vector can be transposed to put together the graph of the two solutions as illustrated here:

$ysol := ysol^T$

$N1 := \mathrm{augment}(xsol, \mathrm{col}(ysol, 1))$

$N2 := \mathrm{augment}(xsol, \mathrm{col}(ysol, 2))$





The plot to the left uses the "Graph by lines" option in the "Plot" palette, while the plot to the right uses the "Graph by points" option in the "Plot" palette.