Fourth Order Runge-Kutta Method and its C Programming

Course: MPHYCC-05 Modeling and Simulation, MPHYCC-09 Lab-II
(M.Sc. Sem-II)

By

Dr. Sanjay Kumar

Assistant Professor
Department of Physics
Patna University

Contact Details: Email- sainisanjay35@gmail.com;
Contact No.- 9413674416

Unit-3 of CC-05

4^{th} Order Runge-Kutta Method

To understand the 4^{nd} order Runge Kutta method, we once again consider the typical first order differential equation:

$$\frac{dy}{dx} = f(x, y) \text{ with the initial condition } y(x = x_1) = y_1$$

Now let's assume h to be the equidistance value of x, i.e.,

$x_2 = x_1 + h$; $x_3 = x_2 + h$; ..... ; $x_{i+1} = x_i + h$

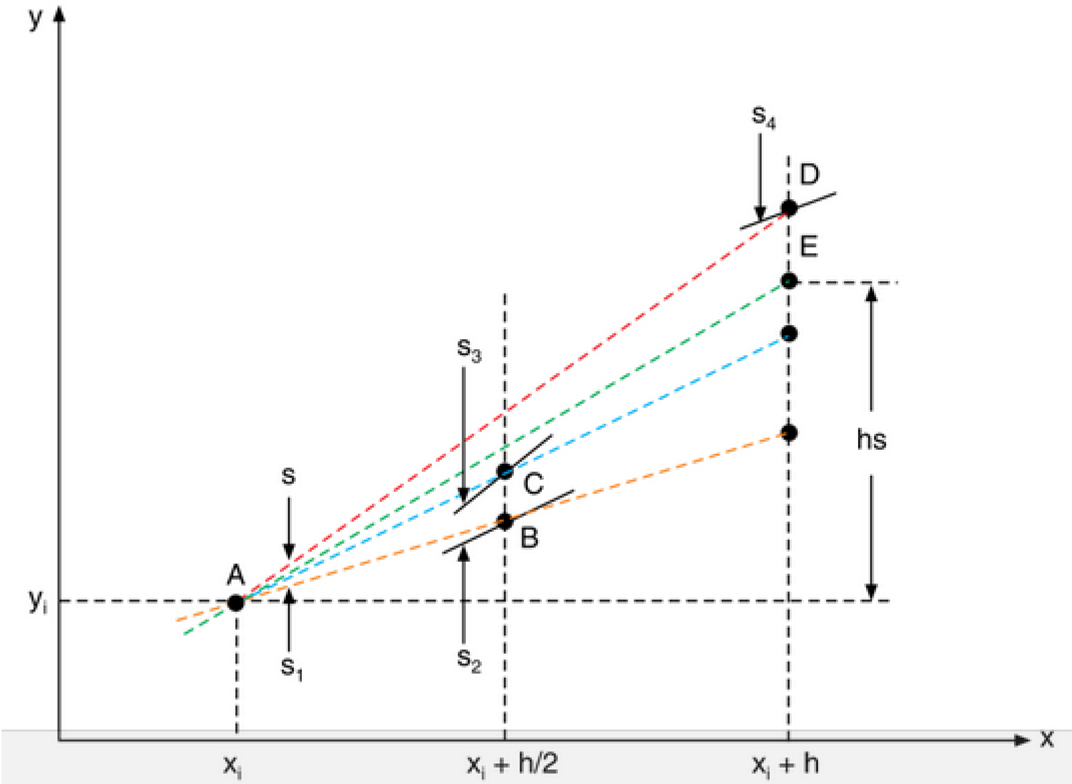

Figure: The figure geometrically illustrates 4^{th} order Runge-Kutta method.

To understand the 4^{th} order Runge-Kutta method refers to the above figure. To enhance the accuracy, in comparison to the 2^{nd} order Runge-Kutta method, here we consider additional point at $x_1 + h/2$ (which is the midpoint of x_1 and $x_1 + h$). In the 4^{th} order Runge-Kutta method, we do the following steps:

1. First of all, we calculate the slope $s_1=f(x_1,y_1)$ of the solution curve y(x) at point (x_1,y_1) (point A in the figure). Then, lets draw a straight line from the initial point (x_1,y_1) with the slope s_1.

2. Lets assume that the straight line cuts the vertical line through $x_1+h/2$ at $(x_1+h/2,\ y_2^{'})$ (point B in the figure).
Note that by definition $s_1=(y_2^{'} - y_1)/(x_1+h/2-x_1)= (y_2^{'} - y_1)/(h/2)$. This implies that $y_2^{'} = y_1 + s_1\, h/2$.
Determine the slope of the solution curve y(x) at the point B. This is given by $s_2=f(x_1 +h/2,\ y_1 + s_1\, h/2)$.

3. Go back to the point (x_1,y_1) and draw a straight line with the slope s_2.
Lets assume that the straight line cuts the vertical line through $x_1+h/2$ at $(x_1+h/2,\ y_2^{''})$ (point C in the figure).
Note that by definition $s_2=(y_2^{''} - y_1)/(x_1+h/2-x_1)=.(y_2^{''} - y_1)/(h/2)$ This implies that $y_2^{''} = y_1 + s_2\, h/2$.
Determine the slope of the solution curve y(x) at the point C. This is given by $s_3=f(x_1 +h/2,\ y_1 + s_2\, h/2)$.

4. Go back to the point (x_1,y_1) and draw a straight line with the slope s_3.
Lets assume that the straight line cuts the vertical line through x_1+h at $(x_1+h,\ y_2^{'''})$ (point D in the figure).
Note that by definition $s_3=(y_2^{'''} - y_1)/(x_1+h-x_1)=.(y_2^{'''} - y_1)/(h)$ This implies that $y_2^{'''} = y_1 + s_3\, h$.
Determine the slope of the solution curve y(x) at the point D. This is given by $s_4=f(x_1 +h,\ y_1 + s_3\, h)$.

5. Now, go back to the point (x_1,y_1) and draw a straight line with a slope $s=(s_1+2s_2 + 2s_3 + s_4)/6$. In the 4nd order Runge-Kutta method, the point y_2 (point E in the figure), where this straight line cuts the vertical line x_1+h, is the approximate solution of the considered differential equation at the point x_1+h. By definition of the slope:

$$s=(y_2 - y_1)/(x_2-x_1)=(y_2 - y_1)/h$$

$$y_2 = y_1 + h\, s$$

$$y_2 = y_1 + h\,(s_1+2s_2 + 2s_3 + s_4)/6$$

where $s_1=f(x_1, y_1)$, $s_2= f(x_1 + h/2, y_1 + s_1\,h/2)$, $s_3= f(x_1 + h/2, y_1 + s_2\,h/2)$ and $s_4= f(x_1 + h, y_1 + s_3\,h)$.

In general, the $(i+1)^{th}$ is obtained from the i^{th} point using the formula:

$$y_{i+1} = y_i + h\,(s_1+2s_2 +2s_3 +s_4)/6$$

where $s_1=f(x_i, y_i)$, $s_2= f(x_i+ h/2, y_i + s_1\,h/2)$, $s_3= f(x_i + h/2, y_i + s_2\,h/2)$ and $s_4= f(x_i + h, y_i + s_3\,h)$.

Assignment

Solve the differential equation using Runge-Kutta 4^{th} order method

$$\frac{dy}{dx}=-y$$

find y for $x \,\varepsilon\, [0, 2]$ with the initial condition $y(x=0)=y_0=1$.

Algorithm to Write a Program of the Runge-Kutta 4^{th} order method

Problem: $\frac{dy}{dx}=f(x,y)$ *with the initial condition* $y(x=x_0)=y_0$ *find y(x) for $x_0 < x < L$*

1. Input x_0, L, y_0, n

2. $h=(x_n-x_0)/n$

3. Do iteraction (i=1,n)
{$s_1=h*f(x_0, y_0)$
$x_1=x_0+h$;
$x_m=x_0 +h/2$;

$s_2=h*f(x_m,y_0+s_1/2)$
$s_3 = h*f(x_m,y_0+s_2/2)$
$s_4 = h*f(x_1,y_0+s_3)$
$y_1=y_0+(1/6)*(s_1+2*s_2 +2*s_3 + s_4)$
write x_1, y_1
$y_0=y_1$
$x_0=x_1$ }

4. end

C- Program of the Runge-Kutta 2nd order method

Problem: $\frac{dy}{dx}=-y$ *with the initial condition* $y(x=0)=1$ *find y(x) for 0 <x<2*

```
#include <stdio.h>
#include <math.h>
int main()
{float k1, k2, k3, k4, x0, l, y0, h, x1, y1, xm;
int n,i;
printf("enter the value of n \n");
scanf("%d",&n);
printf("enter the initial point x0, last point L and initial condition y0:\n");
scanf("%f %f %f",&x0,&l,&y0);
```

```
h=(l-x0)/n;

for(i=1;i<=n;i++)
{x1=x0+h;
 xm=x0+h/2;
k1=-h*y0;
k2=-h*(y0+k1/2);
k3=-h*(y0+k2/2);
k4=-h*(y0+k3);
y1=y0+(1.0/6.0)*(k1+2*k2+2*k3+k4);
printf("x[%d] and y[%d]:%f\t\t%f \n",i,i,x1,y1);
x0=x1;
y0=y1;}

return 0;}
```

Output of the program:

```
enter the value of n
5
enter the initial point x0, last point L and initial condition y0:
0 2 1
x[1] and y[1]:0.400000          0.670400
x[2] and y[2]:0.800000          0.449436
x[3] and y[3]:1.200000          0.301302
x[4] and y[4]:1.600000          0.201993
x[5] and y[5]:2.000000          0.135416
```

\

C- Program of the Runge-Kutta 4th order method using 1D Array

Problem: $\frac{dy}{dx} = -y$ *with the initial condition* $y(x=0)=1$ *find y(x) for 0 <x<2*

```
#include <stdio.h>
#include <math.h>
int main()
```

```
{float k1[100],k2[100],k3[100], k4[100], x[100],y[100],y1[100],h,xm[100];
int n,i;

printf("enter the value of n and h \n");
scanf("%d%f",&n,&h);
printf("enter the value of x[1] and y[1]:\n");
scanf("%f%f",&x[1],&y[1]);

for(i=1;i<=n+1;i++)
{k1[i]=-h*y[i];
x[i+1]=x[i]+h;
xm[i+1]=x[i]+h/2;
k2[i]=-h*(y[i]+0.5*k1[i]);
k3[i]=-h*(y[i]+0.5*k2[i]);
k4[i]=-h*(y[i]+k3[i]);
y[i+1]=y[i]+(1.0/6.0)*(k1[i]+2*k2[i]+2*k3[i]+k4[i]);}

for(i=2;i<=n+1;i++)
printf("x[%d] and y[%d]:%f  %f \n",i,i,x[i],y[i]);

return 0;}
```

Output of the program:

```
enter the value of n and h
5  0.4
enter the value of x[1] and y[1]:
0  1
x[2] and y[2]:0.400000  0.670400
x[3] and y[3]:0.800000  0.449436
x[4] and y[4]:1.200000  0.301302
x[5] and y[5]:1.600000  0.201993
x[6] and y[6]:2.000000  0.135416
```