PT-204: Numerical computation method

UNIT IV

Numerical Solution of

Ordinary Differential Equations

or

First Order and First Degree Differential Equations

Ruge-Kutta Method

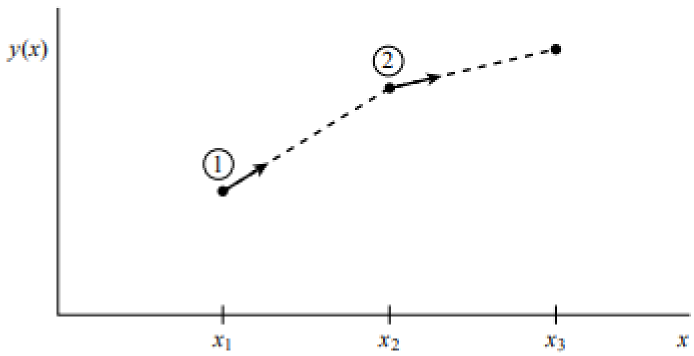
2^{nd} order Runge-Kutta Method

The formula for the Euler method is

$$y_{n+1} = y_n + hf(x_n, y_n) \quad (1)$$

which advances a solution from x_n to $x_{n+1} \equiv x_n+h$.

The formula is unsymmetrical as it advances the solution through an interval h, but uses derivative information only at the beginning of that interval.

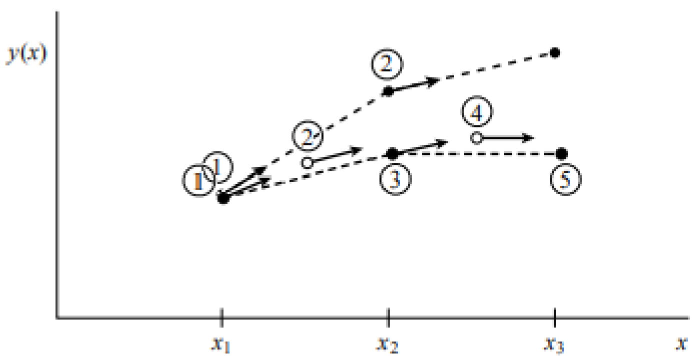


Euler's method is not recommended for practical use as there are several reasons :

(i) The method is not very accurate when compared to other, fancier, methods run at the equivalent stepsize

(ii) It is first order method that is its error is of the order of h^2, ($O\ h^2$).

(iii) It is not a stable method as it changes value with the change on sep size.



Consider, however, the use of a step like (1) to take a "trial" step to the midpoint of the interval. Then use the value of both x and y at that midpoint to compute the "real" step across the whole interval. Figure 2 illustrates the idea.

In equations,

$$k_1 = hf(x_n, y_n)$$

$k2 = hf(xn + ½ h, yn + ½ k1)$ (2)

$yn+1 = yn + k2 + O(h^3)$

As indicated in the error term, this symmetrization cancels out the first-order error term, making the method second order.
[A method is conventionally called nth order if its error term is $O(h^{n+1})$.]

In fact, equation (2)is called the midpoint method. We needn't stop there. There are many ways to evaluate the right-hand side f(x, y) that all agree to first order, but that have different coefficients of higher-order error terms. According to second-order Runge-Kutta method, equations may be proposed as follows:

$h = xn+1 - xn$

$k1 = hf(xn, yn)$

$k2 = hf(xn + h, yn + k1)$ (3)

$yn+1 = yn +(k1+ k2)/2$

Algorithm: This algorithms provides Runge-Kutta 2^{nd} order solution to an ordinary differential equation of first order and first degree which one of the initial condition is known. Let given information is dy/dx= f(x,y) with y(x0) = y0.

Step 1	Read xo	initial limit of x
	Read yo	boundary condition for y at xo
	Read xn	last point upto which solution is to be obtained
	Read n	number of calculations points
Step 2	h <= (xn-xo)/n	step size
	X <= xo	
	Y <= yo	
	print xo, yo	Print initial point information
Step 3	for i = 1 to n	
	k1= h* f(x,y)	
	x2= x+h	
	k2= h* f(x2,y+k1)	
	y2 = y+(k1+k2)/2	
	print x2, y2	Print intermediate point information
	x= x2	
	y=y2	
	end of for loop	
Step 4	stop	End of the algorithm

4^{th} order Runge-Kutta Method

By far the most often used is the classical fourth-order Runge-Kutta formula, which has a certain sleekness of organization about it. The equations may be proposed as follows:

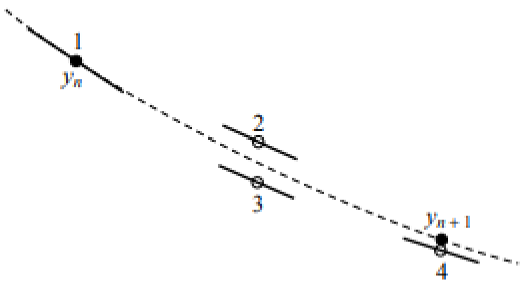
$$
\begin{aligned}
k1 &= hf(xn, yn)\\
k2 &= hf(xn + h/2, yn + k1/2)\\
k3 &= hf(xn + h/2, yn + k2/2) \qquad (4)\\
k4 &= hf(xn + h, yn + k3)\\
yn+1 &= yn + (1/6(k1+ 2*k2 + 2*k3 + k4)
\end{aligned}
$$

The fourth-order Runge-Kutta method requires four evaluations of the right hand side per step h. In this method the order of error is (h^5). This will be superior to the midpoint method (2) if at least twice as large a step is possible with (4) for the same accuracy. Middle points are given two times weight as compared to end points.

Is that so? The answer is: often, perhaps even usually, but surely not always!

This takes us back to a central theme, namely that high order does not always mean high accuracy. The statement "fourth-order Runge-Kutta is generally superior to second-order" is a true. But you should recognize it as a statement about the contemporary practice of science rather than as a statement about strict mathematics. For many scientific users, fourth-order Runge-Kutta is not just the first word on ODE integrators, but the last word as well.

Fourth-order Runge-Kutta method. Graphical representation of the calculations.



In each step the derivative is evaluated four times: once at the initial point, twice at trial midpoints, and once at a trial endpoint. From these derivatives the final function value (shown as a filled dot) is calculated.

Algorithm: This algorithms provides Runge-Kutta 4^{th} order solution to an ordinary differential equation of first order and first degree which one of the initial condition is known. Let given information is dy/dx= f(x,y) with y(x0) = y0.

Step 1	Read xo	initial limit of x
	Read yo	boundary condition for y at xo
	Read xn	last point upto which solution is to be obtained
	Read n	number of calculations points
Step 2	h <= (xn-xo)/n	step size
	X <= xo	
	Y <= yo	
	print xo, yo	Print initial point information
Step 3	for i = 1 to n	
	k1= h* f(x,y)	
	k2= h* f(x+h/2,y+k1/2)	
	k3= h* f(x+h/2,y+k2/2)	
	x2= x+h	
	k4= h* f(x2,y+k3)	
	y2 = y+(k1+2*k2+2*k3+k4)/6	
	print x2, y2	Print intermediate point information
	x= x2	
	y=y2	
	end of for loop	
Step 4	stop	End of the algorithm

Prpoblem Obtain the approximate solution y(t) of IVP using 2^{nd} and 4^{th} order Runge-Kutta methods. Obtain approximate solution at x= 0.1 correct to 4 places of decimal.

$$y' = 1 + xy, \quad y(0)=1$$

Compare results using both the methods.