Software Quality Assurance

Vishisht Tiwari and Deepak Agarwal

vmt28 & da475

The code was developed to perform in following configurations:

- Runge-Kutta 23
- Runge-Kutta 34

and

- Bisection Time Stepping
- Constant E.h Time Stepping
- O(h) Time Stepping
- No Time Stepping

The Software was tested using a validation equation shown below.

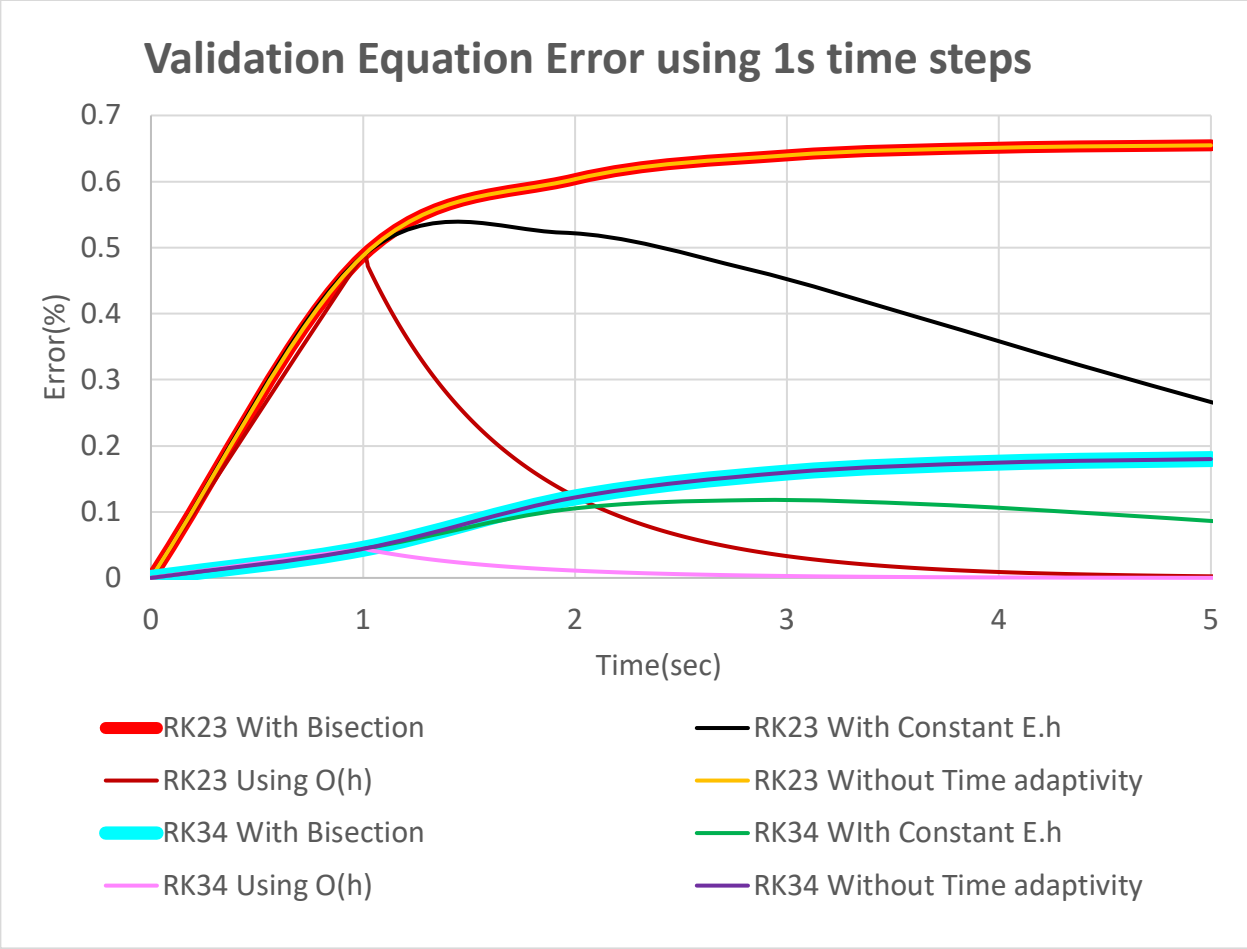
$$\frac{dx}{dt} = 4e^{0.8t} - 0.5x$$

with the equation shown below:

$$x(t) = \frac{4}{1.3}\left(e^{0.8t} - e^{-0.5t}\right) + 2e^{-0.5t}$$

The start time was taken as 0s and stop time as 5s. The ODE Solver was performed on the validation equating using the above methods. Different time steps were also used to validate its effect.

The following graph shows the errors with 1sec time step.



The results were as expected. In all time stepping methods, RK34 performed better than RK23. Among the different time stepping methods, bisection performed as bad as no time stepping. Constant E.h was worse than O(h) time stepping and hence O(h) Time stepping performed the best.

Hence in terms of accuracy O(h) method performed the best followed by Constant E.h and then Bisection.

The speed of these methods has been shown below:

Method	**Speed**
Solving ODE Using RK23 Without Time Adaptivity	0.988ms
Solving ODE Using RK23 With Time Adaptivity Using Bisection	0.85ms
Solving ODE Using RK23 With Time Adaptivity Using Constant E.h	0.641ms
Solving ODE Using RK23 With Time Adaptivity Using O(h)	7.742ms
Solving ODE Using RK34 Without Time Adaptivity	0.810ms
Solving ODE Using RK34 With Time Adaptivity Using Bisection	0.7933ms
Solving ODE Using RK34 With Time Adaptivity Using Constant E.h	0.934ms

Solving ODE Using RK34 With Time Adaptivity Using O(h)	3.00ms

From the above table it can be seen that O(h), with the highest accuracy also takes the longest time to execute. The other methods, however, are quite similar.