CHEE 3602 – Topic 7: Ordinary differential equations

Stanislav Sokolenko

Fall 2020

Runge-Kutta methods

Further generalization

Recall the Euler method one more time:

$$y_{i+1} = y_i + f(x_i, y_i)h$$

The Adams-Bashforth method expanded this equation to consider multiple points:

$$y_{i+1} = y_i + (a_1 f(x_i, y_i) + a_2 f(x_{i-1}, y_{i-1}) + \ldots)h$$

The Runge-Kutta method generalizes this idea even further:

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2 + \ldots)h$$

But instead of using past points, $k_1, k_2, \ldots$ comes from multiple steps forward of different size that are all averaged together.

A fourth order example

A fourth order Runge-Kutta method averages together four such values:

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2 + a_3 k_3 + a_4 k_4)h$$

And each k_j comes from an estimate of the derivative at a different (x, y):

$$\begin{aligned}
k_1 &= f(x_i, \quad y_i) \\
k_2 &= f(x_i + b_2 h, \quad y_i + (c_{21} k_1)h) \\
k_3 &= f(x_i + b_3 h, \quad y_i + (c_{31} k_1 + c_{32} k_2)h) \\
k_4 &= f(x_i + b_4 h, \quad y_i + (c_{41} k_1 + c_{42} k_2 + c_{43} k_3)h)
\end{aligned}$$

A fourth order example

Deriving the Runge–Kutta coefficients takes a little longer than the Adams–Bashforth coefficients (although it is not much harder). But this derivation will not be covered here.

The important thing to remember is that the Runge–Kutta coefficients are not unique. It is possible to use different values of a_i, b_j, c_{jk} given a set of rules that are outside the scope of the course.

A fourth order example

The following is just one set of equations:

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

And each k_j comes from an estimate of the derivative at a different (x, y):

$$k_1 = f(x_i, \quad y_i)$$

$$k_2 = f(x_i + \frac{1}{2}h, \quad y_i + \frac{1}{2}k_1 h)$$

$$k_3 = f(x_i + \frac{1}{2}h, \quad y_i + \frac{1}{2}k_2 h)$$

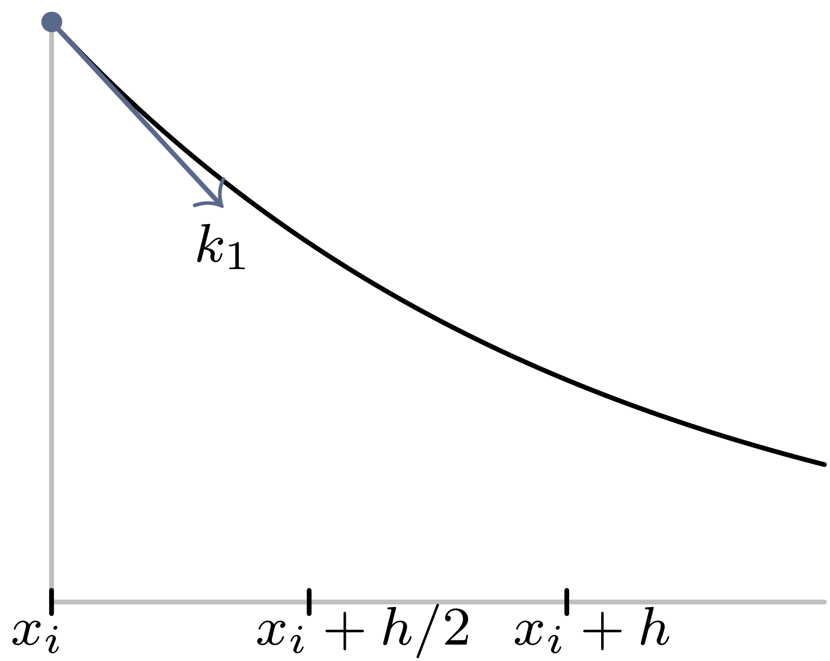
$$k_4 = f(x_i + h, \quad y_i + k_3 h)$$

A fourth order example

Note that these coefficients do not result in the most accurate answer possible. However, all those zero terms save a couple of extra calculations.

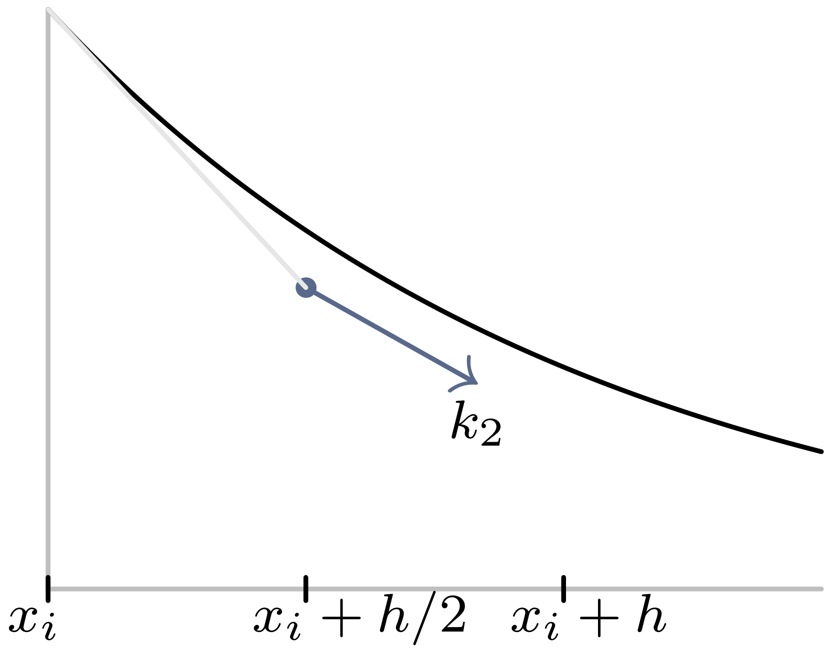
A graphical example

The first estimate of slope (k_1) is equivalent to the Euler method.



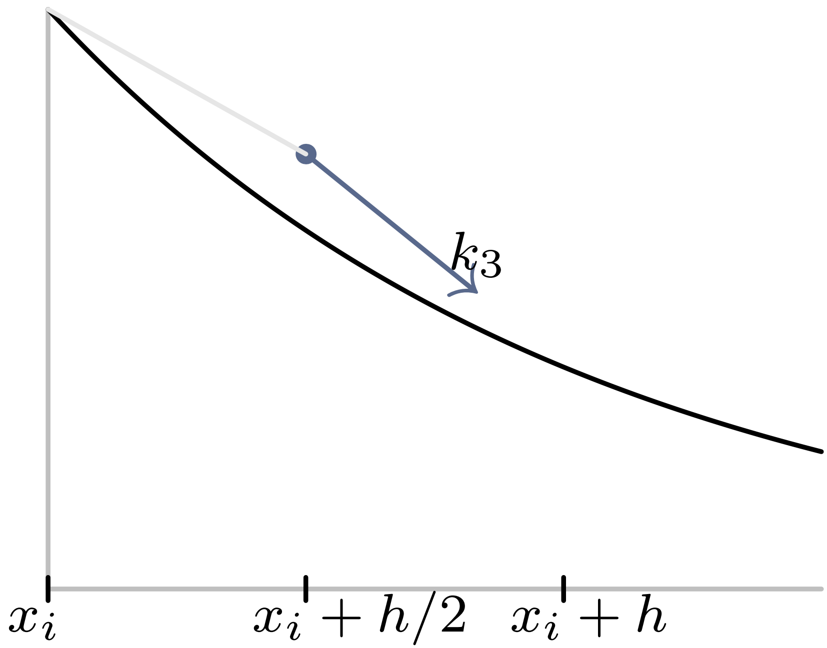
A graphical example

The second estimate (k_2) comes from using the Euler method and extending k_1 halfway to the next point.



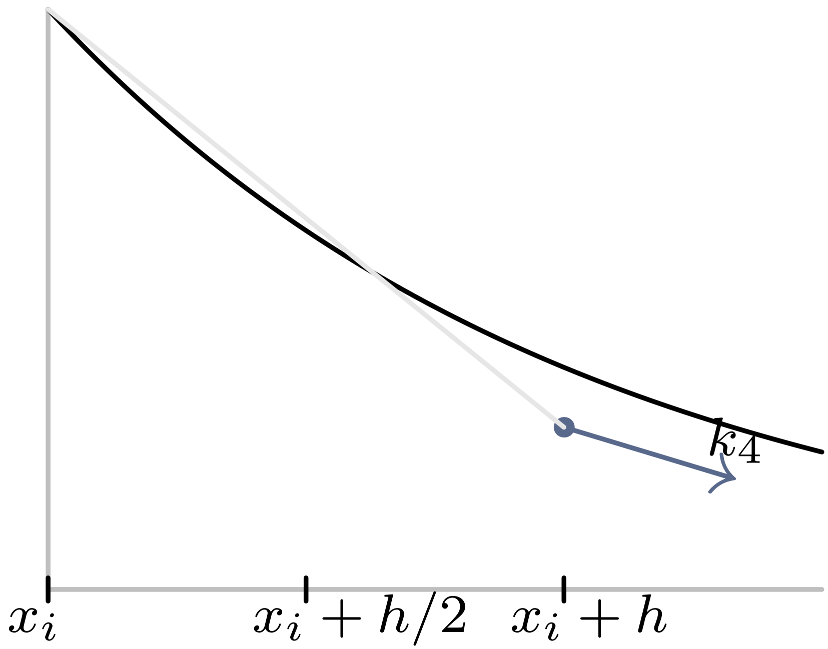
A graphical example

The third estimate (k_3) takes the previous estimate (k_2) back to the starting point and extends it halfway to the next point.



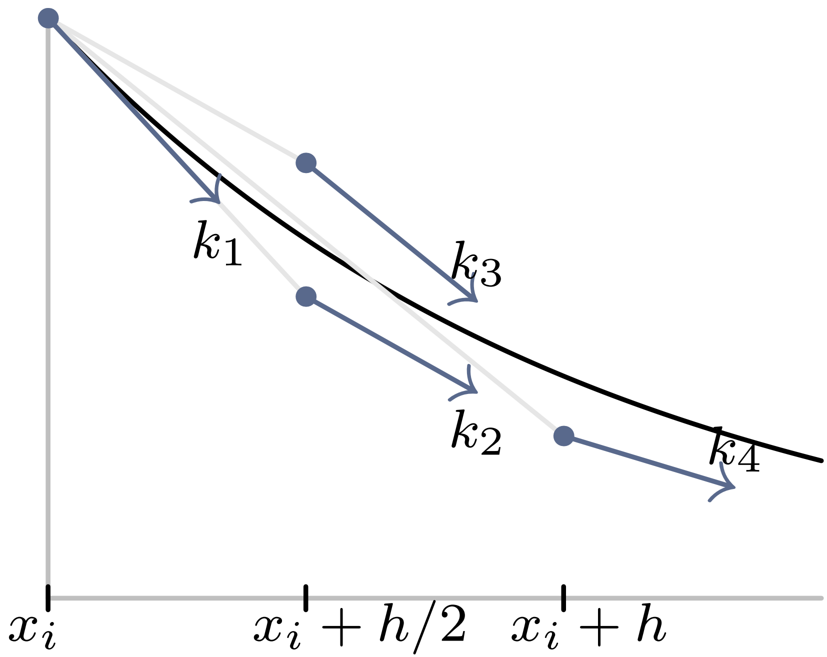
A graphical example

The final estimate (k_4) extends the previous estimate (k_3) from the starting point all the way to the next point.



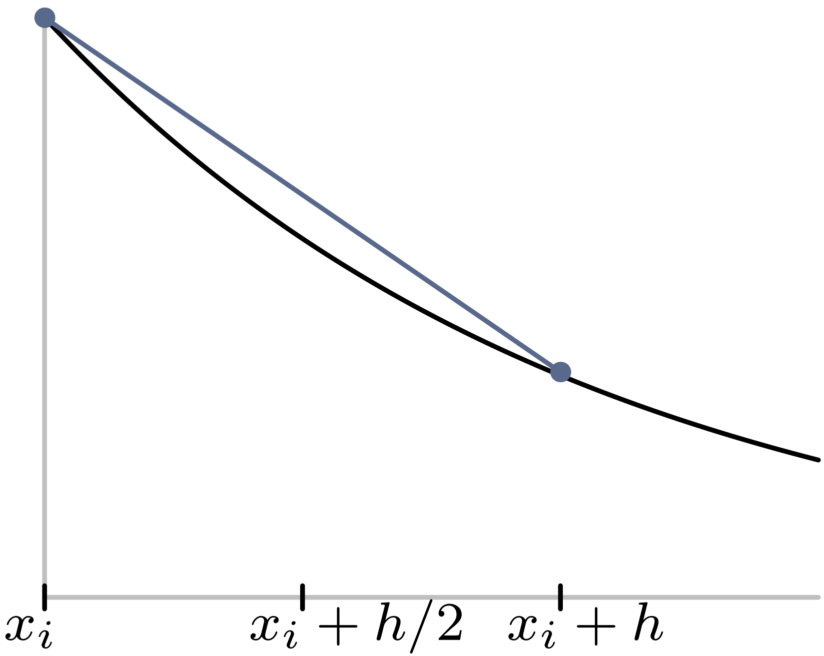
A graphical example

The overall result uses a weighted average of all these estimates.



A graphical example

And achieves a better result than any one of the estimates alone.



Generalizing

Recall the general equation again:

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2 + a_3 k_3 + a_4 k_4)h$$

And each k_j comes from an estimate of the derivative at a different (x, y):

$$\begin{aligned}
k_1 &= f(x_i, \quad y_i) \\
k_2 &= f(x_i + b_2 h, \quad y_i + (c_{21} k_1)h) \\
k_3 &= f(x_i + b_3 h, \quad y_i + (c_{31} k_1 + c_{32} k_2)h) \\
k_4 &= f(x_i + b_4 h, \quad y_i + (c_{41} k_1 + c_{42} k_2 + c_{43} k_3)h)
\end{aligned}$$

Generalizing

All these coefficients can be combined in a tableau:

$$\begin{array}{c|cccc} 0 & & & & \\ b_2 & c_{21} & & & \\ b_3 & c_{31} & c_{32} & & \\ b_4 & c_{41} & c_{42} & c_{43} & \\ \hline & a_1 & a_2 & a_3 & a_4 \end{array}$$

Generalizing

So the example that we considered above would translate to:

$$
\begin{array}{c|cccc}
0 & & & & \\
1/2 & 1/2 & & & \\
1/2 & 0 & 1/2 & & \\
1 & 0 & 0 & 1 & \\
\hline
 & 1/6 & 2/6 & 2/6 & 1/6
\end{array}
$$

Adaptive step size

Despite all the calculations, the Runge-Kutta method is fundamentally a single-step method. Although it lacks the efficiency of using multiple previously calculated steps, this makes it very easy to adapt step size as needed. Which is particularly useful for stiff problems.

Adaptive step size

Adaptive step size methods rely on using two different combinations of k values to generate two different approximations:

1. Estimate error by subtracting two estimates of different orders
2. Check if estimated error meets required tolerance
3. If it does not, calculate optimal step size based on required tolerance and recalculate
4. Take step and repeat

Simplest example

The simplest combination of values compares the Euler $O(h)$ estimate with the so-called Heun $O(h^2)$ method.

$$
\begin{array}{c|cc}
0 & & \\
1 & 1 & \\
\hline
 & 1 & 0 \\
 & 1/2 & 1/2
\end{array}
$$

Since the global accuracy of these methods is $O(h^1)$ and $O(h^2)$, this method is commonly referred to as a Runge-Kutta 1/2 method.

Simplest example

In equation form:

$$y_{i+1}^{(1)} = y_i + (k_1)h$$

$$y_{i+1}^{(2)} = y_i + \frac{1}{2}(k_1 + k_2)h$$

And each k_j comes from an estimate of the derivative at a different (x, y):

$$k_1 = f(x_i, \quad y_i)$$

$$k_2 = f(x_i + h, \quad y_i + k_1 h)$$

Simplest example

Comparing the two y values provides an estimate of error:

$$\varepsilon = |y^{(2)} - y^{(1)}|$$

Which can be compared to the required tolerance.

If $\varepsilon < \varepsilon_{tol}$: take step and increase step size

If $\varepsilon > \varepsilon_{tol}$: decrease step size and recalculate

Calculating step size

Since the Euler method is locally accurate to within $O(h^2)$ and the Heun method is locally accurate to within $O(h^3)$, the difference between the two estimates is the estimate of $O(h^2)$ error.

So decreasing the step size by 2 decreases error by 4:

$$\left(\frac{h_{tol}}{h}\right)^2 = \frac{\varepsilon_{tol}}{\varepsilon}$$

$$h_{tol} = h\left(\frac{\varepsilon_{tol}}{\varepsilon}\right)^{1/2}$$

In practice

Some of the most popular Runge–Kutta methods implement adaptive step sizing with a mixed 4/5 order tableau (implemented in Matlab as `ode45`). This is just one example (Runge–Kutta–Fehlberg):

0						
1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	−7200/2197	7296/2197			
1	439/216	−8	3680/513	−845/4104		
1/2	−8/27	2	−3544/2565	1859/4104	−11/40	
	25/216	0	1408/2565	2197/4104	−1/5	0
	16/135	0	6656/12825	28561/56430	−9/50	2/55

Example

Example 6

Given a first order decay rate of 0.06 h^{-1} and a starting concentration of 1 M at 0 hours, find the concentration of compound A using a single step of 1/2 Runge-Kutta method given a local tolerance of 0.01 M. Use a step size of 6 hours for the initial step.

$$\frac{\mathrm{d}C_A}{\mathrm{d}t} = -0.06C_A$$

The big picture

While adaptive methods are very useful for ensuring accuracy, the resulting data points are not as convenient as an evenly sampled grid for further analysis.

Are there any options to convert unevenly sampled points into evenly sampled ones?