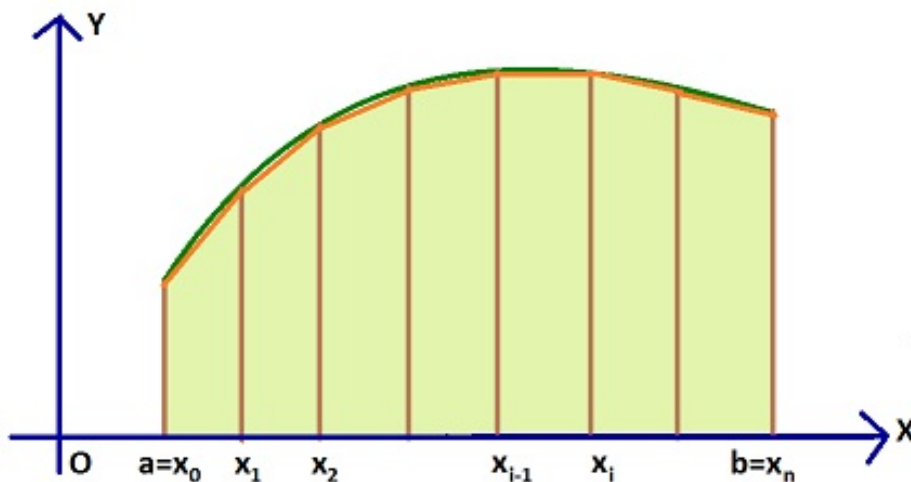


EE2-8C Mathematics
Numerical Analysis of Differential Equations

Introductory Exercise: Trapezoidal Rule I

Not every integral has an exact solution. To approximate a solution we can employ numerical techniques, the simplest of which is the Trapezoidal Rule. Consider the integral of $f(x)$ over the interval $[a, b]$. We begin by dividing the interval into n segments of equal length $h = \frac{b-a}{n}$:



Set $a = x_0$ and $b = x_n$, then the n intervals are $[x_i, x_{i+1}]$ for $i = 0 \dots n-1$.

Introductory Exercise:Trapezoidal Rule II

The area under the curve $f(x)$ now consists of integrating over each segment

$$\int_{x_i}^{x_{i+1}} f(x) dx$$

and summing. For each segment, we approximate the integral with the area of a trapezoid of left height $f(x_i)$ and right height $f(x_{i+1})$, giving an approximate area:

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx \frac{1}{2}h (f(x_i) + f(x_{i+1}))$$

Introductory Exercise:Trapezoidal Rule III

To see a simple case, we let $n = 3$, so $a = x_0$ and $b = x_3$, and we have

$$\frac{1}{2}h (f(x_0) + f(x_1)) + \frac{1}{2}h (f(x_1) + f(x_2)) + \frac{1}{2}h (f(x_2) + f(x_3))$$

which can be tidied up as

$$\frac{1}{2}h (f(x_0) + 2f(x_1) + 2f(x_2) + f(x_3))$$

so in the general case we can write

$$\int_a^b f(x)dx \approx \left(\frac{b-a}{2n}\right) \left[f(x_0) + 2 \left(\sum_{i=1}^{n-1} f(x_i) \right) + f(x_n) \right]$$

Exercise I

1.) Use the Trapezoidal Rule to obtain

$\mathcal{I} = \int_0^1 e^x dx = e - 1 \approx 1.718281828 \dots$. Let \mathcal{I}_n be the approximate solution with n segments; begin by obtaining \mathcal{I}_4 . Increase n to improve precision. Let ϵ_n be the error made with segment length $h = (b - a)/n$. For each n , calculate $\epsilon_n = |\mathcal{I} - \mathcal{I}_n|$.

2.) Tabulate your results and compare n and the error ϵ_n . Can you deduce any relationship between the error and the number of segments?

3.) Write a simple Matlab code to investigate this further. (i) Given a function $f(x)$, an interval $[a, b]$ and a number of segments n , your code should integrate the function numerically over $[a, b]$ with the interval divided into n segments. Compare with the exact solution. (ii) Next, incorporate the code you've written into a loop so you can investigate the effect of increasing n on the error. Integrate for $n = 2 \dots N$ where N is chosen to be suitably large. Plot the error against n . What can you conclude? Hint: try a log-log plot for ϵ_n and h . Test this on several functions and intervals.

Exercise II

Now consider Simpson's rule which is a refinement of the trapezoidal rule. Take n an even number and proceed as follows:

$$\int_a^b f(x) dx \approx \left(\frac{b-a}{3n} \right) \left[f(x_0) + 2 \left(\sum_{i=1}^{n/2-1} f(x_{2i}) \right) + 4 \left(\sum_{i=1}^{n/2} f(x_{2i-1}) \right) + f(x_n) \right]$$

To see what is going on, let $n = 6$, then we have

$$\frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + 4f(x_5) + f(x_6)] .$$

Repeat the previous exercise and error analysis with Simpson's rule. What do you observe?

For further reading, google "numerical integration": the wikipedia entry is well-explained and there are, as usual, many more.

Or be daring: go to the Library, find a text on Numerical Methods or Numerical Analysis (there are dozens) and look for Numerical Integration.

ODEs

An ordinary differential equation (ODE) of the form

$$y' = f(x, y)$$

may have an analytical (exact) solution. For example,

$$y' = x + y,$$

with initial condition $y(0) = 2$, is linear. Use integrating factor e^{-x} to find the solution

$$y = 3e^x - x - 1.$$

But more complicated functions $f(x, y)$ make this difficult or impossible. Numerical Methods are necessary.

Higher order equations

The methods used for first-order equations apply to higher order equations as well. Can always rewrite as systems of first-order equations. In the second-order equation

$$y'' = f(x, y, y'),$$

let $z = y'$, thus obtaining a system of coupled first-order equations:

$$z' = f(x, y, y'), \quad y' = z.$$

For example

$$y'' - 2y' + 3y = \sin x$$

becomes

$$y' = z, \quad z' = 2z - 3y + \sin x.$$

Circuit example

The simplest LR-circuit is governed by the ODE: $L \frac{di}{dt} + Ri = V$.

(inductance L , resistance R , current i , voltage source V)

With a capacitor we have a simple LCR-circuit with a 2nd-order ODE

$$L \frac{di}{dt} + Ri + \frac{1}{C} \int i dt = V \Rightarrow L \frac{d^2i}{dt^2} + R \frac{di}{dt} + \frac{1}{C} i = \frac{dV}{dt}.$$

(capacitance C)

All of these can be solved numerically using the methods to follow.

Euler's Method I

The simplest is Euler's method. Recalling that y' is the slope of a function, we use it to calculate the slope at a point x_i and then estimate the value at a nearby point. Let (x_0, y_0) be the known initial condition. Then for a nearby point (x_1, y_1)

$$y_1 = y_0 + \left. \frac{dy}{dx} \right|_{x=x_0} (x_1 - x_0)$$

is a *linear* approximation. The smaller $h = x_1 - x_0$, the better the approximation. If we need to know some specific value $y(x_f)$, or the behaviour of the function on the interval $[x_0, x_f]$ we divide the interval into N subintervals of size h and calculate

$$y_{i+1} = y_i + \left. \frac{dy}{dx} \right|_{x=x_i} (x_{i+1} - x_i) = y_i + h f(x_i, y_i),$$

where $N = i + 1$ completes the solution.

Euler's Method II

Example

For the previous example, we have $y' = x + y$ so

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

where $(x_0, y_0) = (0, 2)$. If we would like to obtain the behaviour of y on $[0, 1]$, we set $x_f = 1$. Subdivide the interval into four equal pieces of length $h = 1/4$. Then

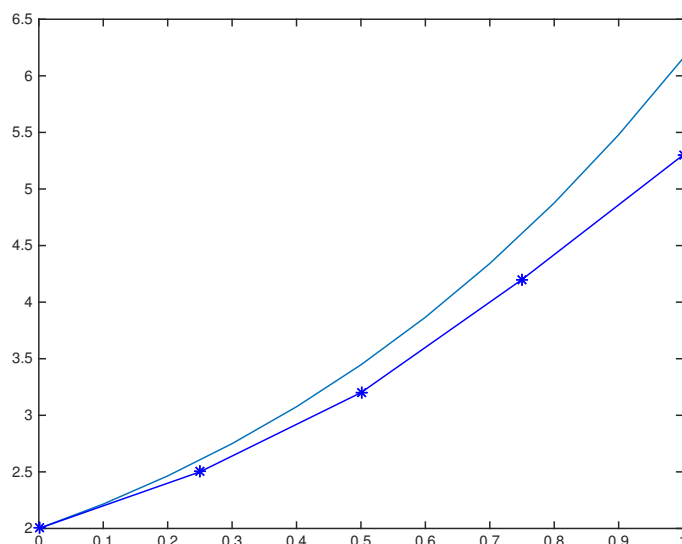
$$y_1 = y_0 + h(x_0 + y_0) = 2 + \frac{1}{4}(0 + 2) = 2.5$$

$$y_2 = y_1 + h(x_1 + y_1) = 2.5 + \frac{1}{4}(0.25 + 2.5) = \frac{51}{16} \approx 3.2$$

$$y_3 = y_2 + h(x_2 + y_2) = 51/16 + \frac{1}{4}(0.5 + 51/16) = \frac{263}{64} \approx 4.2$$

$$y_4 = y_3 + h(x_3 + y_3) = 263/64 + \frac{1}{4}(0.75 + 263/64) = \frac{1363}{256} \approx 5.3$$

Euler's Method III



blue: numerical solution

green: exact solution

Error I

Recall the Taylor series of a function $g(x)$ around $x = a$:

$$g(x) = g(a) + g'(a)(x - a) + \frac{g''(a)}{2}(x - a)^2 + \dots$$

First exchange x and a :

$$g(a) = g(x) + g'(x)(a - x) + \frac{g''(x)}{2}(a - x)^2 + \dots$$

Now let $a = x + h$, so that $a - x = h$ and we have

$$g(x + h) = g(x) + g'(x)h + \frac{g''(x)}{2}h^2 + \dots$$

We can use this last expression to write $y(x)$ as a Taylor series around h :

$$y(x + h) = y(x) + hy'(x) + h^2 \frac{y''(x)}{2} + \dots$$

Error II

In the notation used in our implementation of Euler's method, we take $y(x_{i+1}) = y(x_i + h)$ and recall $y' = f(x, y)$, so that

$$y_{i+1} = y_i + hf(x_i, y_i) + \frac{h^2}{2}f'(x_i, y_i) + \dots$$

so the error at each step is proportional to h^2 :

$$\frac{h^2}{2}f'(x, y) = O(h^2)$$

this is called the *local* truncation error, incurred in a single iteration, from x_i to x_{i+1} . As we move from x_0 to x_f some of these errors may cancel each other out, as in some cases they may be above, in some cases below the true value of y , but this can not be relied on. In the worst case, all errors will be of the same sign, giving an estimate of the *global* truncation error as proportional to Nh^2 and since $h = \frac{x_f - x_0}{N}$ we see that the global error is $O(h)$.

Matlab code to evaluate Euler's method:

```
x=0; % set initial value of x0
y=2; % set yinitial condition y at x0
h=0.03; %set step-size
xf=1; %set final value of x
N=round((xf-x)/h); %nr of steps: (interval size)/(step
size)
plot(x,y,'*'); %plot initial condition
hold on; %figure open for more data
for i=1:N %loop for N steps
y=y + h*(x+y); %next value of y
x=x+h; %increase x by stepsize
plot(x,y, 'b*') %plot now values of x,y
end
xx=0:h:xf; %same interval, stepsize
yy=3*exp(xx)-xx-1; %calculate exact solution
plot(xx,yy,'b'); %plot exact solution
```

Predictor-Corrector Methods I

An improvement on Euler's method is Heun's method, also called the "Improved Euler Method".

In Euler's method, the gradient at the beginning of an interval $[x_i, x_{i+1}]$:

$$y'_i = f(x_i, y_i)$$

is used to make a prediction of the value of y_{i+1} at the end of the interval:

$$y_{i+1}^p = y_i + hf(x_i, y_i).$$

We use the superscript p to indicate the prediction and set this as the first-step of a *predictor-corrector* method, where we now estimate the gradient at the end of the interval as

$$y'_{i+1} = f(x_{i+1}, y_{i+1}^p).$$

Predictor-Corrector Methods II

We now have two estimates of the gradient, one at each end of the interval, so we can estimate a better over-all gradient on $[x_i, x_{i+1}]$ as the average of the two:

$$y'_{av} = \frac{y'_i + y'_{i+1}}{2} = \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^p)}{2}.$$

Predictor-Corrector Methods III

Using this average gradient, we can now use Euler's method to estimate y_{i+1} :

$$y_{i+1} = y_i + h \left(\frac{y'_i + y'_{i+1}}{2} \right) = y_i + h \left(\frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^p)}{2} \right),$$

which is called the *corrector* equation.

The method is summed up as

$$y_{i+1}^p = y_i + hf(x_i, y_i) \quad \text{Predictor equation}$$

$$y_{i+1} = y_i + h \left(\frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^p)}{2} \right) \quad \text{Corrector equation}$$

Note that the corrector equation includes two distinct values of the unknown $y(x_{i+1})$, the first estimate y_{i+1}^p and the second y_{i+1} . This can be seen as an iteration: use the corrector equation repeatedly to improve the estimate of y_{i+1} . This will not necessarily converge on the exact value of y but can significantly reduce the local truncation error.

Second-order Runge-Kutta I

Runge-Kutta methods have several orders, there is one of order 1: the Euler method, but there are infinitely many of order 2, of which we have seen only two.

In general, we write

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

where ϕ is called the *increment* function and can be interpreted as an optimal representation of the gradient of y over the interval $[x_i, x_{i+1}]$. The increment function can be written as

$$\phi = a_1k_1 + a_2k_2 + \dots a_nk_n$$

Second-order Runge-Kutta II

where the a_i are constants and the k_i are

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

$$k_2 = f(x_i + p_1h, y_i + q_{11}k_1h)$$

$$k_3 = f(x_i + p_2h, y_i + q_{21}k_1h + q_{22}k_2h)$$

.

.

.

$$k_n = f(x_i + p_{n-1}h, y_i + q_{n-1,1}k_1h + q_{n-1,2}k_2h + \dots + q_{n-1,n-1}k_{n-1}h)$$

for an n^{th} order RK method. The k_i are obtained recurrently, easy to implement in any programming environment. For $n = 1$ there is only one term k_1 , giving Euler's method. For $n = 2$ we require constants p_1 and q_{11} and weights a_1, a_2 . Using Taylor expansion we can obtain three equations for these, so they are not uniquely determined: there are infinitely many second-order RK methods.

Derivation of Second-order RK I

We have $y_{i+1} = y_i + h(ak_1 + bk_2)$, where $k_1 = f(x_i, y_i)$, and $k_2 = f(x_i + ph, y_i + qk_1h)$, where we drop indices for optical simplicity and seek a, b, p, q . We begin with the Taylor expansion we saw earlier:

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2!}y''(x) + \dots \text{ giving}$$

$$y_{i+1} = y_i + hf(x_i, y_i) + \frac{h^2}{2}f'(x_i, y_i) + \dots \text{ where we use partial differentiation}$$

to obtain $f'(x, y) = \frac{\partial f(x, y)}{\partial x} + \frac{\partial f(x, y)}{\partial y} \frac{dy}{dx}$, so that

$$y_{i+1} = y_i + hf(x_i, y_i) + \frac{h^2}{2} \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} \right) + \dots \quad (**)$$

Derivation of Second-order RK II

Now consider $k_2 = f(x_i + ph, y_i + qk_1h)$. Recall the Taylor expansion for a function of two variables:

$$F(x+h, y+k) = F(x, y) + h \frac{\partial F}{\partial x} + k \frac{\partial F}{\partial y} + O(h^2, k^2)$$

which we apply to obtain

$$k_2 = f(x_i + ph, y_i + qk_1h) = f(x_i, y_i) + ph \frac{\partial f}{\partial x} + qk_1h \frac{\partial f}{\partial y} + O(h^2).$$

Substituting for k_1 and k_2 into $y_{i+1} = y_i + h(ak_1 + bk_2)$ we have

$$\begin{aligned} y_{i+1} &= y_i + h a k_1 + h b k_2 \\ &= y_i + h a f(x_i, y_i) + h b \left[f(x_i, y_i) + p h \frac{\partial f}{\partial x} + q f(x_i, y_i) h \frac{\partial f}{\partial y} + O(h^2) \right] \end{aligned}$$

and collect terms:

$$= y_i + (a+b)hf(x_i, y_i) + h^2 \left(b p \frac{\partial f}{\partial x} + b q \frac{\partial f}{\partial y} f(x_i, y_i) \right) + O(h^3)$$

Derivation of Second-order RK III

Now compare this last equation with (**), not forgetting that $dy/dx = f(x_i, y_i)$, to see that we require

$$\begin{aligned}a + b &= 1 \\bp &= 1/2 \\bq &= 1/2\end{aligned}$$

which is a system of 3 equations in 4 unknowns, with infinitely many solutions.

Choose for example $b = 1/2$, so that $a = 1/2$ and $p = q = 1$. This gives

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

where $k_1 = f(x_i, y_i)$, and $k_2 = f(x_i + h, y_i + k_1 h)$. Noting that k_1 , resp. k_2 , is the gradient at the beginning, resp. end of the interval, this is just the Heun method with a single correction y_{i+1}^p .

Derivation of Second-order RK IV

Each choice of b gives a new variation, each with the same rough global error $O(h^2)$ but differences in finer detail, see any text on Numerical Analysis for more detail.

Midpoint method I

Another, similar, approach improves Euler's method, based on using Euler's method to estimate the gradient at the midpoint of the interval $[x_i, x_{i+1}]$ and using this to obtain a better estimate of y at the endpoint.

In terms of the RK2 notation used above, we let $b = 1$, so that $a = 0$ and $p = q = 1/2$ giving

$$y_{i+1} = y_i + hk_2$$

where k_1 is as usual, and $k_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1h\right)$, which is the midpoint method.

$k_1 =$	$f(x_i, y_i)$	gradient at x_i using Euler
$k_2 =$	$f(x_i + 0.5h, y_i + 0.5hk_1)$	gradient at $x_i + h/2$ using Euler
$y_{i+1} =$	$y_i + hk_2$	

Higher-order Runge-Kutta Methods

There are many, here is just one of the most common in use, the:

Classic fourth order Runge-Kutta method

$$\begin{aligned}k_1 &= f(x_i, y_i) \\k_2 &= f(x_i + 0.5h, y_i + 0.5k_1h) \\k_3 &= f(x_i + 0.5h, y_i + 0.5k_2h) \\k_4 &= f(x_i + h, y_i + k_3h) \\&\text{combined with} \\y_{i+1} &= y_i + h \left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \right)\end{aligned}$$

Error Analysis I

If the ODE is of the form $y' = f(x)$, a function of x only, the predictor step is not required, as we can obtain the average gradient directly from x_i and x_{i+1} so that a single application of the corrector step gives

$$y_{i+1} = y_i + h \left(\frac{f(x_i) + f(x_{i+1})}{2} \right), \quad (*)$$

which looks remarkably like the trapezoidal method of numerical integration. Let's explore this. We can solve the ODE $y' = f(x)$ by integrating over the appropriate interval:

$$\int_{y_i}^{y_{i+1}} dy = \int_{x_i}^{x_{i+1}} f(x) dx \Rightarrow y_{i+1} - y_i = \int_{x_i}^{x_{i+1}} f(x) dx.$$

Error Analysis II

Recall that the trapezoidal method approximates

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx h \left(\frac{f(x_i) + f(x_{i+1})}{2} \right)$$

where $h = x_{i+1} - x_i$. Hence the error in $(*)$ is given by the error in the Trapezoidal method: a heuristic error analysis of this method gave the global truncation error as $O(h^2)$, and the same applies to Heun's method, though we do not show this analytically. It can be shown that for the general case $y' = f(x, y)$, the same error holds.

Higher order ODE - Coupled first order ODEs I

A second order, or higher system, can be written as a system of first-order equations:

$$\frac{d^2x}{dt^2} + 2\frac{dx}{dt} - 3x = \sin t$$

with initial conditions $x(0) = x'(0) = 0$. Taking $y = x'$ we have $y' = x''$ giving the system of two coupled first-order ODEs:

$$x' = y = f_1(t, x, y), \quad y' = \sin t + 3x - 2y = f_2(t, x, y),$$

with initial conditions $x(0) = y(0) = 0$, which can be solved by any of the RK methods. Though f_1 is not actually a function of t, x , we write it with all three arguments, for the case of where we have two coupled first-order equations which can be functions of all three variables. The method can be generalized in an obvious way for an ODE of order n giving n coupled first-order ODEs.

Higher order ODE - Coupled first order ODEs II

We illustrate by employing Euler's and Heun's method to solve the system. Note that where we had one dependent variable y and one independent variable x , we now have two dependent variables x, y and one independent variable t . So we will have increments in time: $t_{i+1} = t_i + h$ giving increments in x and y :

$$\begin{aligned}x_{i+1} &= x_i + h \phi_x(t_i, x_i, y_i) \\y_{i+1} &= y_i + h \phi_y(t_i, x_i, y_i)\end{aligned}$$

The increments for Euler and second-order RK are given as

$$\begin{aligned}\phi_x &= ak_{1x} + bk_{2x} \\ \phi_y &= ak_{1y} + bk_{2y},\end{aligned}$$

where $a = 1, b = 0$ for Euler's method and $a = b = 1/2$ for Heun's method.

Higher order ODE - Coupled first order ODEs III

For Euler's method, the easiest case, we have

$$\begin{aligned}k_{1x} &= f_1(t_i, x_i, y_i) = y_i \\k_{1y} &= f_2(t_i, x_i, y_i) = \sin t_i + 3x_i - 2y_i\end{aligned}$$

which means we can skip the $\phi = ak_1$ stage and write

$$\begin{aligned}x_{i+1} &= x_i + h y_i \\y_{i+1} &= y_i + h(\sin t_i + 3x_i - 2y_i)\end{aligned}$$

with the usual $x_0 = 0$ and $y_0 = 0$, for $t_0 = 0$. We divide up the interval, say $[0, 1]$ into n equal segments of length $h = 1/n$ and iterate as before, calculating two new values, x_{i+1} and y_{i+1} at each stage, increasing t_i by h until $t_n = 1$. To see this in action, let $h = 0.2$, and begin Euler's iteration:

Higher order ODE - Coupled first order ODEs IV

$$\begin{aligned}x_1 &= x_0 + h y_0 = 0 + 0.2(0) = 0 \\y_1 &= y_0 + h(\sin t_0 + 3x_0 - 2y_0) = 0\end{aligned}$$

which looks odd, but keep on iterating. Increase t to $t_1 = 0.2$, then

$$\begin{aligned}x_2 &= x_1 + h y_1 = 0 + 0.2(0) = 0 \\y_2 &= y_1 + h(\sin t_1 + 3x_1 - 2y_1) = 0 + 0.2(\sin 0.2 + 0 - 0) \approx 0.03973\end{aligned}$$

and increase t to $t_2 = 0.4$:

$$\begin{aligned}x_3 &= x_2 + h y_2 \approx 0 + 0.2(0.03973) \approx 0.007947 \\y_3 &= y_2 + h(\sin t_2 + 3x_2 - 2y_2) \approx 0.03973 + 0.2(\sin 0.4 + 0 - 2(0.03973)) \\&\approx 0.1017\end{aligned}$$

Higher order ODE - Coupled first order ODEs V

and so on. Doing this with increasing order of RK method just requires care. We illustrate it with Heun's method. Here we have

$$\begin{aligned}k_{1x} &= f_1(t_i, x_i, y_i) = y_i \\k_{1y} &= f_2(t_i, x_i, y_i) = \sin t_i + 3x_i - 2y_i\end{aligned}$$

as before, and

$$\begin{aligned}k_{2x} &= f_1(t_i + ph, x_i + hqk_{1x}, y_i + hqk_{1y}) \\&= y_i + hk_{1y} \\k_{2y} &= f_2(t_i + ph, x_i + hqk_{1x}, y_i + hqk_{1y}) \\&= \sin(t_i + h) + 3(x_i + hk_{1x}) - 2(y_i + hk_{1y})\end{aligned}$$

(recall that for Heun $p = q = 1$) giving the increments, and so

$$\begin{aligned}x_{i+1} &= x_i + h(ak_{1x} + bk_{2x}) = x_i + \frac{h}{2}(k_{1x} + k_{2x}) \\y_{i+1} &= y_i + h(ak_{1y} + bk_{2y}) = y_i + \frac{h}{2}(k_{1y} + k_{2y})\end{aligned}$$

as $a = b = 1/2$ for Heun.

Finite Differences for ODE I

Finite Difference methods take a different approach to those seen up to now. For a second-order, or higher, ODE, we can distinguish between an IVP, where all the information is known at the same (initial) point and a Boundary Value Problem, BVP where, for example $y'' - 2y' + 3y = \sin x$, with $y(0) = 0$ and $y(1) = 0$. We have two conditions on y , needed to specify a second-order system but now we have information on y at two different points, usually thought of as the boundaries of an interval of interest. We may similarly have $y'(0) = -1$, $y'(1) = 1$ or other variations. Note: Unlike IVPs, solutions to BVPs do not necessarily exist, see any text on Differential equations, for example "Elementary Differential Equations and Boundary value problems" by Boyce/Di Prima. We will not concern ourselves with this problem and deal only with BVPs where solutions exist.

Finite Differences for ODE II

The approach followed previously will not work here. If we represent the second-order equation as a system of two first-order equations in y and $z = y'$, the boundary conditions $y(0) = 0$ and $y(1) = 0$ do not tell us anything about z so we have no z_0 to start off the process. Worse, even if we did, we can start off the process at y_0 but cannot guarantee in any sensible way that the iteration would lead to $y_n = y(1)$ as given. Similar problems arise for other possible combinations of the boundary conditions. What is to be done? A burning question!

Finite Differences for ODE III

One solution (among several) lies in breaking up the interval in question, in this case $[0, 1]$. We begin by finding an approximation for the second derivative. Define the *central difference*

$$\delta y(x_i) = y\left(x_i + \frac{h}{2}\right) - y\left(x_i - \frac{h}{2}\right),$$

and note that

$$\frac{1}{h}\delta y(x_i) = \frac{y(x_i + \frac{h}{2}) - y(x_i - \frac{h}{2})}{h} \approx \left.\frac{dy}{dx}\right|_{x=x_i}$$

Finite Differences for ODE IV

Similarly, the second central difference is $\delta^2 y(x_i) =$

$$\begin{aligned} &= \delta(\delta y(x_i)) \\ &= \delta y\left(x_i + \frac{h}{2}\right) - \delta y\left(x_i - \frac{h}{2}\right) \\ &= y\left[\left(x_i + \frac{h}{2}\right) + \frac{h}{2}\right] - y\left[\left(x_i + \frac{h}{2}\right) - \frac{h}{2}\right] \\ &\quad - \left\{ y\left[\left(x_i - \frac{h}{2}\right) + \frac{h}{2}\right] - y\left[\left(x_i - \frac{h}{2}\right) - \frac{h}{2}\right] \right\} \\ &= y(x_i + h) - 2y(x_i) + y(x_i - h) \end{aligned}$$

Since the second derivative y'' is the first derivative of y' , a similar argument gives

$$\frac{1}{h^2} \delta^2 y(x_i) \approx \left. \frac{d^2 y}{dx^2} \right|_{x=x_i}$$

Finite Differences for ODE V

To approximate the first derivative in terms of $x_i \pm h$, we proceed slightly differently. Take the central difference this time with

$\Delta y(x_i) = y(x_i + h) - y(x_i - h)$ so that

$$\frac{1}{2h} \Delta y(x_i) = \frac{y(x_i + h) - y(x_i - h)}{2h} = \frac{y(x_{i+1}) - y(x_{i-1}))}{2h} \approx \left. \frac{dy}{dx} \right|_{x=x_i}$$

We now take $u_i \approx y(x_i)$ so we can approximate

$$\left. \frac{dy}{dx} \right|_{x=x_i} \approx \frac{u_{i+1} - u_{i-1}}{2h}, \quad \text{and} \quad \left. \frac{d^2 y}{dx^2} \right|_{x=x_i} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}.$$

Example I

We take the model convection-diffusion equation

$$y'' - 20y' = -1, \quad \text{with } y(0) = 1, y(1) = 0.$$

We divide the interval of interest $[0, 1]$ into five equal segments of length $h = 1/5$, giving $x_0 = 0, x_1 = 0.2, x_2 = 0.4, x_3 = 0.6, x_4 = 0.8, x_5 = 1$. Estimating the derivatives using the formulae in the previous slide, we get

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - 20 \left(\frac{u_{i+1} - u_{i-1}}{2h} \right) = -1$$

and rearrange to

Example II

$$\left(\frac{10}{h} + \frac{1}{h^2} \right) u_{i-1} - \frac{2}{h^2} u_i + \left(-\frac{10}{h} + \frac{1}{h^2} \right) u_{i+1} = -1$$

or simply

$$au_{i-1} + bu_i + cu_{i+1} = -1,$$

where

$$a = 10/h + 1/h^2 = 75, b = -2/h^2 = -50 \text{ and } c = -10/h + 1/h^2 = -25$$

for $i = 1, 2, 3, 4$, and $u_0 = 1, u_5 = 0$.

Example III

Each $i = 1..4$ is an equation involving u_{i-1}, u_i, u_{i+1} :

$$\begin{array}{rcll} au_0 + & bu_1 + & cu_2 & = -1 \\ & au_1 + & bu_2 + & cu_3 & = -1 \\ & & au_2 + & bu_3 + & cu_4 & = -1 \\ & & & au_3 + & bu_4 + & cu_5 & = -1 \end{array}$$

As we know u_0 and u_5 the first and last equations involve only two unknowns, and we can rewrite in matrix form

$$\begin{pmatrix} b & c & 0 & 0 \\ a & b & c & 0 \\ 0 & a & b & c \\ 0 & 0 & a & b \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} -1 - au_0 \\ -1 \\ -1 \\ -1 - cu_5 \end{pmatrix}$$

Example IV

The solution is easily obtained with Gaussian elimination. In this case, $u_1 \approx 0.9928$, $u_2 \approx 1.0544$, $u_3 \approx 0.9095$, $u_4 \approx 1.3843$.

A matrix of this type is called "tridiagonal". The solution of larger matrices is best done using numerical algorithms, of which many exist.