Lectures - Week 12
Single step and Multistep Methods for First Order Initial Value Problems

Runge-Kutta Methods

To obtain a more accurate scheme than Forward Euler, we must do additional work such as additional function evaluations. Single step methods take the approach that to approximate the solution at t_{n+1} using only Y^n (and not previously computed values) we obtain an approximation at intermediate steps in (t_n, t_{n+1}) and use these to get Y^{n+1}. We have already seen one such method, the Midpoint Rule, which was written as

$$\begin{aligned} k_1 &= \Delta t f(t_n, Y^n) \\ k_2 &= \Delta t f\Big(t_n + \frac{\Delta t}{2}, Y^n + \frac{1}{2}k_1\Big) \\ Y^{n+1} &= Y^n + k_2 \end{aligned}$$

Example Use the Midpoint Rule to approximate the solution to

$$y'(t) + 2ty^2(t) = 0, \quad y(0) = 1$$

at $T = 0.2$ using $\Delta t = 0.1$. Compare with the results with the exact solution and the approximation we obtained using Euler's Method. Recall that the exact solution is $\dfrac{1}{1+t^2}$.
$Y^0 = 1$ and $f(t, y) = -2ty^2(t)$ so

$$k_1 = .1f(0, Y^0) = .1f(0, 1) = .1(0) = 0$$

$$k_2 = .1f(.05, Y^0 + .5k_1) = .1f(.05, 1) = .1(-2)(.05)(1^2) = -.01$$
$$Y^1 = Y^0 + k_2 = 1 - 0.01 = 0.99$$

To calculate Y^2 we have

$$k_1 = .1f(.1, Y^1) = .1f(.1, 0.99) = .1(-.196) = -0.196$$

$$k_2 = .1f(.15, Y^1 + .5k_1) = .1f(.15, .98903) = -.0293448$$
$$Y^2 = Y^1 + k_2 = 1 - .0293448 = .960655$$

The exact solution is $y(.2) = 0.961538$. With Euler's method we got 0.98 with an error of .0185 whereas here we have a much smaller error of 0.00088.

One can demonstrate that the Midpoint Rule is a second order scheme, i.e., the global error is $\mathcal{O}(\Delta t^2)$. The Midpoint Rule is an example of a family of single-step methods called Runge-Kutta Methods. We will show that the Midpoint Rule has a local truncation error of $\mathcal{O}(\Delta t^3)$ and thus we expect it to have a global error of $\mathcal{O}(\Delta t^2)$. Remember that to find the local truncation error we look at the difference in the exact solution at t_{n+1}

and our approximate solution there where we have used the *exact* solution at t_n. For the Midpoint Rule this gives

$$\begin{aligned} y(t_{n+1}) - \hat{Y}^{n+1} &= y(t_n) + \Delta t y'(t_n) + \frac{\Delta t^2}{2!} y''(t_n) + \frac{\Delta t^3}{3!} y'''(t_n) + \frac{\Delta t^4}{4!} y''''(\xi_n) \\ &\quad - \Big[y(t_n) + \Delta t f\Big(t_n + \frac{\Delta t}{2}, y(t_n) + \frac{1}{2}\Delta t f(t_n, y(t_n))\Big)\Big] \end{aligned}$$

Now as in the case of Euler's method the terms for $y(t_n)$ cancel but now instead of just $f(t_n, y(t_n))$ which would cancel with $y'(t_n)$ we have this term in the square brackets. We need to expand it in terms of a Taylor series so we can simplify terms. However, $f(t, y)$ is a function of two independent variables so we need Taylor series for a function of several variables. This is easy to obtain because we can simply expand in terms of one variable, holding the other fixed, and then expand all terms in the second variable. For example, if $f(x, y)$ then we can get an approximation to $f(x + \Delta x, y + \Delta y)$ by first expanding in terms of x and then with respect to y. We have

$$f(x + \Delta x, y + \Delta y) = f(x, y + \Delta y) + \Delta x f_x(x, y + \Delta y) + \frac{\Delta x^2}{2!} f_{xx}(x, y + \Delta y) + \mathcal{O}(\Delta x^3)$$

and now expanding each $f(x, y + \Delta y)$ or its derivatives in a Taylor series gives

$$\begin{aligned} f(x + \Delta x, y + \Delta y) &= \Big[f(x, y) + \Delta y f_y(x, y) + \frac{\Delta y^2}{2!} f_{yy}(x, y) + \mathcal{O}(\Delta y^3) \Big] \\ &\quad + \Delta x \Big[f_x(x, y) + f_{xy}(x, y)\Delta y + \mathcal{O}(\Delta y^2) \Big] \\ &\quad + \frac{\Delta x^2}{2!} \Big[f_{xx}(x, y) + \mathcal{O}(\Delta y) \Big] + \mathcal{O}(\Delta x^3) \end{aligned}$$

Thus we have

$$\begin{aligned} f(x + \Delta x, y + \Delta y) &= f(x, y) + \Delta x f_x(x, y) + \Delta y f_y(x, y) \\ &\quad + \frac{\Delta y^2}{2} f_{yy}(x, y) + f_{xy}(x, y)\Delta y \Delta x + \frac{\Delta x^2}{2} f_{xx}(x, y) + \mathcal{O}(\Delta x^3, \Delta y^3) \end{aligned}$$

We can now use this to simplify the term $f\big(t_n + \frac{\Delta t}{2}, y(t_n) + \frac{1}{2}\Delta t f(t_n, y(t_n))\big)$; i.e., we are expanding about the point $(t_n, y(t_n)$ and the change in t is $\frac{\Delta t}{2}$ and the change in y is $\frac{\Delta t}{2} f(t_n, y(t_n))$. Note that in our expression this term is multiplied by Δt already so we only need to keep terms through Δt^2 to get the Δt^3 terms.

$$\begin{aligned} f\Big(t_n + \frac{\Delta t}{2}, y(t_n) + \frac{\Delta t}{2} f(t_n, y(t_n))\Big) &= f(t_n, y(t_n)) + \frac{\Delta t}{2} f_t(t_n, y(t_n)) + \frac{\Delta t}{2} f(t_n, y(t_n)) f_y(t_n, y(t_n)) \\ &\quad + \frac{\Delta t^2}{4} f_{tt}(t_n, y(t_n)) + \frac{\Delta t^2}{4} f(t_n, y(t_n)) f_{ty}(t_n, y(t_n)) \\ &\quad + \frac{\Delta t^2}{4} f^2(t_n, y(t_n)) f_{yy}(t_n, y(t_n)) + \mathcal{O}(\Delta t^3) \end{aligned}$$

Using this in our local truncation error and setting $f(t_n, y(t_n) = y'(t_n)$ we have

$$\begin{aligned}
y(t_{n+1}) - \hat{Y}^{n+1} &= \Delta t y'(t_n) + \frac{\Delta t^2}{2!} y''(t_n) + \frac{\Delta t^3}{3!} y'''(t_n) + \frac{\Delta t^4}{4!} y''''(\xi_n) \\
&- \Delta t \Big[y'(t_n) + \frac{\Delta t}{2} f_t(t_n, y(t_n)) + \frac{\Delta t}{2} y'(t_n) f_y(t_n, y(t_n)) \\
&+ \frac{\Delta t^2}{4} f_{tt}(t_n, y(t_n)) + \frac{\Delta t^2}{4} y'(t_n) f_{ty}(t_n, y(t_n)) \\
&+ \frac{\Delta t^2}{4} (y'(t_n))^2 f_{yy}(t_n, y(t_n)) + \mathcal{O}(\Delta t^3)]
\end{aligned}$$

Now the terms involving $y'(t_n)$ cancel. To get the remaining terms to cancel we must obtain $y''(t_n)$ in terms of derivatives of f. To do this we use the Chain Rule from calculus

$$y'(t) = f(t, y(t)) \Rightarrow y''(t) = f_t \frac{dt}{dt} + f_y \frac{dy}{dt} = f_t + f_y y'(t)$$

Using this gives us

$$\begin{aligned}
y(t_{n+1}) - \hat{Y}^{n+1} &= \frac{\Delta t^2}{2!} y''(t_n) + \frac{\Delta t^3}{3!} y'''(t_n) + \frac{\Delta t^4}{4!} y''''(\xi_n) \\
&- \Delta t \Big[\frac{\Delta t}{2} \left[f_t(t_n, y(t_n)) + y'(t_n) f_y(t_n, y(t_n)) \right] \\
&+ \frac{\Delta t^2}{4} f_{tt}(t_n, y(t_n)) + \frac{\Delta t^2}{4} y'(t_n) f_{ty}(t_n, y(t_n)) \\
&+ \frac{\Delta t^2}{4} (y'(t_n))^2 f_{yy}(t_n, y(t_n)) + \mathcal{O}(\Delta t^3) \Big] \\
&= \frac{\Delta t^3}{3!} y'''(t_n) + \frac{\Delta t^4}{4!} y''''(\xi_n) \\
&- \Delta t \Big[+ \frac{\Delta t^2}{4} f_{tt}(t_n, y(t_n)) + \frac{\Delta t^2}{4} y'(t_n) f_{ty}(t_n, y(t_n)) \\
&+ \frac{\Delta t^2}{4} (y'(t_n))^2 f_{yy}(t_n, y(t_n)) + \mathcal{O}(\Delta t^3) \Big] \\
&= \mathcal{O}(\Delta t^3)
\end{aligned}$$

Thus the Midpoint Rule has at least a truncation error of $(\Delta t)^3$ and thus we expect it to be second order. To show that the local truncation error is exactly $\mathcal{O}(\Delta t^3)$ (and not $\mathcal{O}(\Delta t^4)$) we would have to demonstrate that the the third order terms don't cancel. This calculation is tedious but a straightforward application of Taylor series in two independent variables which is very useful.

A general second order Runge-Kutta method takes the form

$$\begin{aligned}
k_1 &= \Delta t f(t_n, Y^n) \\
k_2 &= \Delta t f\Big(t_n + c_2 \Delta t, Y^n + a_{21} k_1\Big) \\
Y^{n+1} &= Y^n + \omega_1 k_1 + \omega_2 k_2
\end{aligned}$$

The Midpoint Rule is a special case where

$$c_2 = \frac{1}{2}, \quad a_{21} = \frac{1}{2}, \quad \omega_1 = 0, \quad \omega_2 = 1\,.$$

Are we free to choose any values of the parameters $c_1, a_{21}, \omega_1, \omega_2$ that we want? Of course the answer is no, if we want a second order scheme. The question then arises if there are any other second order schemes besides the Midpoint Rule. To derive a second order scheme you would proceed just as we did for the Midpoint Rule except you would use the general scheme above starting with $y(t_n)$

$$\hat{Y}^{n+1} = y(t_n) + \omega_1 \Delta t f(t_n, y(t_n)) + \omega_2 \Delta t f\Big(t_n + c_2\Delta t, y(t_n) + a_{21}\Delta t f(t_n, y(t_n))\Big)$$

The calculations are analogous, just a bit more tedious. After the expansion you look at the coefficients of the terms multiplying Δt, $(\Delta t)^2$ and you want this terms to be zero so you get a set of equations involving your four parameters. Without doing all the expansion, lets just look at the terms involving y and y'. We have

$$\begin{aligned} y(t_{n+1}) - \hat{Y}^{n+1} &= y(t_n) + \Delta t y'(t_n) + \cdots \\ &\quad - \Big[y(t_n) + \omega_1 \Delta t f(t_n, y(t_n)) + \omega_2 \Delta t f\Big(t_n + c_2\Delta t, y(t_n) + a_{21}\Delta t f(t_n, y(t_n))\Big)\Big] \\ &= \Delta t y'(t_n) + \cdots \\ &\quad - \omega_1 \Delta t y'(t_n) - \omega_2 \Delta t y'(t_n) + \cdots \end{aligned}$$

Thus we must require that

$$\omega_1 + \omega_2 = 1$$

in order for the terms involving Δt to disappear. If you keep the remaining terms, you will get the additional equations

$$c_2\omega_1 = \frac{1}{2} \qquad c_2 = a_{21}$$

Note that the coefficients for the Midpoint Rule satisfy these three equations. Because we have four coefficients and only three equations there is a free parameter and so we have a whole family of second order Runge-Kutta Methods.

There are families of Runge-Kutta methods of different orders. For each increase of Δt in the order we require an additional function evaluation in the interval (t_n, t_{n+1}). For example, a general fourth order Runge-Kutta is

$$\begin{aligned} k_1 &= \Delta t f(t_n, Y^n) \\ k_2 &= \Delta t f\Big(t_n + c_2\Delta t, Y^n + a_{21}k_1\Big) \\ k_3 &= \Delta t f\Big(t_n + c_3\Delta t, Y^n + a_{31}k_1 + a_{32}k_2\Big) \\ k_4 &= \Delta t f\Big(t_n + c_4\Delta t, Y^n + a_{41}k_1 + a_{42}k_2 + a_{43}k_3\Big) \\ Y^{n+1} &= Y^n + \omega_1 k_1 + \omega_2 k_2 + \omega_3 k_3 + \omega_4 k_4 \end{aligned}$$

Typically, in the literature one simply gives the order of the method used and the coefficients in a table such as below.

$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

The first column gives the values of c_i, the remaining columns (above the horizontal line) give the values of a. The first row gives the coefficients for k_2, the second row for k_3, etc. The last row gives the values of ω_i which always add to one.

Multistep Methods

Another approach to deriving schemes which improve the first order accuracy of Euler's Method is to use our approximate solution at previous points to obtain Y^{n+1}. For example, if we wanted to use Y^{n-1} in addition to Y^n to generate Y^{n+1} as well as $f(t_n, Y^n)$ and $f(t_{n-1}, Y^{n-1})$ we would call this a two-step method and it would have the general form

$$Y^{n+1} = a_1 Y^n + a_2 Y^{n-1} + \Delta t \Big[b_1 f(t_n, Y^n) + b_2 f(t_{n-1}, Y^{n-1}) \Big]$$

This is an example of an *explicit* scheme because we have the unknown Y^{n+1} equal to all known values. We already have $f(t_{n-1}, Y^{n-1})$ so we don't have to do another evaluation besides $f(t_n, Y^n)$ which we would have to do anyway for Euler's Method.

We can also have an *implicit* two-step scheme which is given by

$$Y^{n+1} = a_1 Y^n + a_2 Y^{n-1} + \Delta t \Big[b_0 f(t_n, Y^{n+1}) + b_1 f(t_n, Y^n) + b_2 f(t_{n-1}, Y^{n-1}) \Big]$$

This is actually the general form of a two-step multistep method and if $b_0 = 0$ then it is explicit, otherwise it is implicit.

One of the difficulties with multistep methods over single step methods is the starting values. In single step methods we only need the initial condition to begin whereas in a multistep method we may need the solution at several values before we can implement it. For example, in the two-step method we need Y^0 and Y^1 before we can implement our formula. For a four-step method we would need these values plus Y^2, Y^3. Of course we can just use a single step method to calculate our starting values but we need to be sure we use a scheme with the same (or better) order of accuracy as our multistep method.

Just like Runge-Kutta methods there are a myriad of choices for the coefficients. One of the most popular families of explicit multistep methods are the Adams-Bashforth schemes which use only the solution at the previous step, i.e., Y^n (with $a_1 = 1$) but use the function values at all points considered. For example, in our two-step method above $a_2 = 0$. The analogous implicit multistep methods are the Adams-Moulton schemes. These families of methods can be derived in different ways but one of the most popular is to use an interpolating polynomial. For example, if we are using an explicit two step method we find the line which passes through the points $(t_{n-1}, Y^{n-1}))$ and $((t_n, Y^n)$ and then Y^{n+1} is the value of the line at t_{n+1}.

Example Use the two step Adams Bashforth scheme with $a_1 = 1, a_2 = 0$, $b_1 = 1.5$ and $b_2 = -.5$ to approximate the solution to

$$y'(t) + 2ty^2(t) = 0, \quad y(0) = 1$$

at $T = 0.2$ using $\Delta t = 0.1$. Use the approximation at t_1 from the Midpoint Rule for Y^1. $Y^0 = 1$, $f(t, y) = -2ty^2(t)$ and from the Midpoint Rule example $Y^1 = 0.99$ To calculate Y^2 we have

$$\begin{aligned} Y^2 = Y^1 + .1\Big[1.5f(.1, Y^1) - .5f(0, Y^0)\Big] &= .99 + .1\Big[1.5f(.1, .99) - .5f(0, 1)\Big] \\ = 0.99 + .1\Big[1.5(-.196) - .5(0)\Big] &= 0.99 - 0.029403 = 0.960597 \end{aligned}$$

so our error is $.9615 - .960597 = 0.000903$ which is the same order of magnitude as we found with the Midpoint Rule.

If we used an implicit scheme in the previous example then to find Y^2 we would have an expression like

$$\begin{aligned} Y^2 &= Y^1 + .1\Big[b_0 f(.2, Y^2) + b_1 f(.1, Y^1) + b_2 f(0, Y^0)\Big] \\ &= .99 + .1\Big[b_0(-2)(.2)(Y^2)^2 + b_1 f(.1, Y^1) + b_2 f(0, Y^0)\Big] \end{aligned}$$

The problem with finding Y^2 here is that we have a nonlinear equation for Y^2. In our case it is just a quadratic but in general it is a nonlinear equation which must be solved iteratively. If we want to avoid this computational cost, what can we do? We combine the two methods by predicting with an explicit method and then correcting with an implicit method.

Predictor-Corrector Methods

To avoid the cost of solving an implicit method we combine explicit and implicit multistep methods. The basic idea is if we have the solution at points through t_n and want to obtain an approximation at t_{n+1} then we use an explicit scheme to *predict* the solution at t_{n+1}, call it $\tilde{Y}^{n+1}$. Now we use an implicit scheme to improve, i.e., *correct*, the solution there. When we implement the implicit scheme we use $f(t_{n+1}, \tilde{Y}^{n+1})$ instead of $f(t_{n+1}, Y^{n+1})$ so we no longer have a truly implicit scheme and can solve for Y^{n+1} explicitly without solving a nonlinear equation. As an example, consider the following predictor-corrector pair where we use Forward Euler to predict and a Trapezoidal rule to correct. The Trapezoidal rule is an implicit scheme of the form

$$Y^{n+1} = Y^n + \frac{\Delta t}{2}\Big[f(t_{n+1}, Y^{n+1}) + f(t_n, Y^n)\Big]$$

The predictor-corrector pair is then

$$\tilde{Y}^{n+1} = Y^n + \Delta t f(t_n, Y^n)$$

$$Y^{n+1} = Y^n + \frac{\Delta t}{2}\Big[f(t_{n+1}, \tilde{Y}^{n+1}) + f(t_n, Y^n)\Big]$$

We know that Euler's method is a first order method and so is the Trapezoidal rule but the predictor-corrector combination is second order.

Higher order IVPs

Can we apply the methods we have seen to solve say a second order IVP such as

$$y''(t) + t^2 y'(t) = ty, \quad y(0) = 1, y'(0) = 2$$

The answer is found by converting this second order IVP to a system of first order IVPs by introducing a new variable. For this example we define a new variable $w(t) = y'(t)$. Then our equations become

$$\begin{aligned} y'(t) &= w(t) \\ w'(t) + t^2 w(t) &= ty(t) \\ y(0) &= 1 \quad w(0) = 2 \end{aligned}$$

We can then apply a method such as Forward Euler to this system. We first apply it to the equation for y and then for w at each time step. For example, if we take a step of length $\Delta t = 0.1$ we get

$$Y^0 = 1, \quad W^0 = 2$$

$$\frac{Y^1 - Y^0}{\Delta t} = W^0 \Rightarrow Y^1 = Y^0 + \Delta t W^0 = 1 + .1(2) = 1.2$$

$$\frac{W^1 - W^0}{\Delta t} + t_0^2 W^0 = t_0 Y^0 \Rightarrow W^1 = .1(0 \cdot Y^0 - 0 \cdot W^0) + W^0 = 2$$