Initial Value Problems for ODEs

Some applications of Ordinary Differential Equations (they really arise in many places):

- Chemical Reactor Dynamics
- Predator-Prey problems
- Atmospheric Fluid Dynamics
- Electrical circuits
- Advanced mass-spring systems
- Pendulum Dynamics
- Biological Dynamics e.g., glucose uptake and insulin production
- Finance

Some ODEs (actually most ODEs) cannot be solved analytically. For example, the ODE may be nonlinear or the right hand side makes it too hard to integrate. E.g. pendulum equation $\frac{d^2\theta}{dt^2} - \frac{g}{L}\sin\theta \;=\; 0$. For small θ we linearize the ODE ($\sin x \simeq x$ for small x), but for larger values of θ we usually solve numerically.

We'll study the first order ODE

$$\frac{dy}{dt} \;=\; f(y,t) \quad \text{with} \quad y(a) \;=\; \alpha$$

(Any higher order ODE that can be solved for the highest derivative term can be written as a system of first-order ODEs. Show how to do this with a 3rd order ODE.)

We would like simply to integrate the above ODE to get $y(t) \;=\; \int_a^t f(y(t),t)\;dt$. However, we do not know the integrand since we do not yet know $y(t)$.

We'll only consider problems with "reasonable" functions f. If f changes too rapidly, the integration (numerical) method might have difficulty.

A *Well Posed Initial Value Problem* has a unique solution. Small changes in f or $y(a)$ lead to small changes in the solution (otherwise, round-off error leads to inaccurate solutions - e.g., chaos might arise).

The idea of ALL numerical methods for solving (integrating) ODEs is to approximate an average slope on an interval

$$\overline{f(y,t)}$$

Elementary Theory of ODE Initial Value Problems (IVPs)

A function $f(y,t)$ satisfies a *Lipschitz condition* in y on a region in the (t,y)-plane if a constant L exists such that $|f(y_1,t) - f(y_2,t)| \leq L|y_1 - y_2| \;\forall\; (t,y_1),(t,y_2)$ in the region.
That is, f does not change very rapidly when y changes by only a little bit (for fixed t).

If $f(y,t)$ is continuous and Lipschitz on a region in the (t,y) plane ($a \leq t \leq b$, $-\infty < y < \infty$), then the IVP $\frac{dy}{dt} = f(y,t)$ with $y(a) = \alpha$ has a unique solution, $y(t)$ for $a \leq t \leq b$. (If f_y is only continuous at $a \leq t \leq b$, there is a unique solution, but it is not guaranteed to exist on the entire interval).

Taylor Series Method

Since $y(t) = y(a) + y'(a)(t-a) + \frac{1}{2}y''(a)(t-a)^2 + \ldots$ and we are given $y(a)$ and a recipe for $\frac{dy}{dt} = f(y,t)$, we can plug in and compute $y'(a)$. We can take the derivatives of the original ODE

$$\frac{d^2y}{dt^2} = f_y\frac{dy}{dt} + f_t = f_y f + f_t$$

$$\frac{d^3y}{dt^3} = f_{tt} + f_{ty}f + f_y(f_t + ff_y) + f(f_{yt} + f_{yy}f)$$

We plug in a and find $y''(a)$. We can continue in this manner and find more and more terms in our Taylor series.

Disadvantages:

- The method is not easy to code since we'd need some sort of automatic partial differentiation.
- The number of terms grows as we take more derivatives.
- The method is only any good if convergence is fast around $t = a$, so that we would only need a few terms.
- The method is only any good if we are not interested in values of $y(t)$ for t far from a.

Example: $\frac{dy}{dt} = y^2e^{3t}$ where $y(0) = 1$

Example: $\frac{dy}{dt} = -2t - y$ where $y(0) = -1$ (from text)

The error can be related to the first abandoned term of the Taylor Series. Suppose we keep terms through $\frac{1}{n!}\frac{d^n y}{dt^n}(t-a)^n$. Then we can estimate the error as $\frac{d^{n+1}y(\xi)}{dt^{n+1}}\frac{(t-a)^{n+1}}{(n+1)!}$. Unfortunately, we have no idea what our error is since we don't know y, although if our Taylor Series terms are shrinking, we can be pretty confident we are not leaving out too much.

We expect the computed solution to be good around $t = a$. We can piece together solutions by computing an approximate solution for a small region around $t = a$ and connecting to next region. Work out a few terms for the higher order ODE IVP.
Example: $y'' = 3 + x - y^2$ with $y(0) = 1$ and $y'(0) = -2$.

Euler's Method

This method is easy to understand, apply and code up. It is not used much (because there are other easy to apply methods that have smaller error), but presents the main ideas of alot of these methods in a simple form.

- Let $t_i = a + ih$, where $h = \frac{b-a}{N}$, dividing the interval from a to b into N pieces.
- We have $\frac{dy}{dt} = f(y,t)$
- So, $y(t_{i+1}) = y(t_i + h) = y(t_i) + hf(y_i, t_i) + \dots$ by Taylor's Theorem.
- We ignore the higher order terms, giving us the difference equation:
  $w_{i+1} = w_i + hf(w_i, t_i)$
- (We've called the computed values w and will continue to do so throughout).
- Since the first term ignored has order h^2, in one step, the error has order h^2.
- Since going from $t = a$ to $t = b$ takes $\frac{b-a}{h}$ steps, the total error is $h^2\frac{b-a}{h} \simeq h$.
- It can be shown more carefully that the error

$$|y_i - w_i| \leq \frac{hM}{2L}[e^{hLi} - 1]$$

  where L is the Lipschitz constant and M is an upper bound for $y''(\xi)$ where $\xi \in [a,b]$.

How to arrive at this error estimate for Euler's method

- Since $y(t_{i+1}) = y(t_i+h) = y(t_i) + hf(y_i,t_i) + \frac{h^2}{2}y''(\xi)$ and $w_{i+1} = w_i + hf(w_i,t_i)$
- $|y_{i+1} - w_{i+1}| \leq |y_i - w_i| + h|f(y_i,t_i) - f(w_i,t_i)| + \frac{h^2}{2}|y''(\xi_i)|$.
- If f is Lipschitz, with constant L, and has second derivative less than M, we have $|y_{i+1} - w_{i+1}| \leq |y_i - w_i| + hL(y_i - w_i)| + \frac{h^2}{2}M$.
- This gives the difference equation (letting $\rho_i = |y_i - w_i|$): $\rho_{i+1} \leq (1+hL)\rho_i + \frac{h^2}{2}M \Longrightarrow \rho_{i+1} \leq e^{ihL}(\rho_0 + \frac{h^2}{2}\frac{M}{Mh}) - \frac{h^2M}{2hL}$
- (Using $e^x \leq 1+x$ and if $a_{i+1} \leq (1+s)a_i + t$, then $a_{i+1} \leq e^{(i+1)s}(a_0 + t/s) - t/s$)
- So, if the initial error is zero, $\rho_{i+1} \leq (e^{ihL} - 1)\frac{hM}{2L}$.

Local truncation error:
Write the difference method as $w_0 = \alpha$ and $w_{i+1} = w_i + h\phi(w_i,t_i)$.
Let $\tau_i(h) = \frac{y_i - y_{i-1}}{h} - \phi(y_{i-1},t_{i-1})$ where the ys are the exact values. This can be written $\tau_{i+1}(h) = \frac{y_{i+1} - w_{i+1}}{h}$.
We assume we have the exact value at the previous step.
For Euler's method we have

$$\begin{aligned} \tau_i(h) &= \frac{y_i - y_{i-1}}{h} - \phi(w_{i-1},t_{i-1}) \\ &= \frac{y_{i-1} + hf(y_{i-1},t_{i-1}) + \frac{h^2}{2}y''(\xi) - y_{i-1}}{h} - f(y_{i-1},t_{i-1}) \\ &= \frac{h}{2}y''(\xi) \\ \text{So, } |\tau_i(h)| &\leq \frac{hM}{2} \end{aligned}$$

and the local truncation error is $O(h)$.

Pictorially, we are approximating the ODE be a tangent line of length h (in the t direction). We then repeat the procedure from the new (computed) point, $w(t_{i+1})$.

Example: Use Euler's method for $\frac{dy}{dt} = -2t - y$ where $y(0) = -1$ using $h = 0.1$ to approximate $y(0.4)$.

Local truncation error for the Taylor Series Method:
Write the difference method as $w_0 = \alpha$ and $w_{i+1} = w_i + h\phi(w_i,t_i)$.
Let $\tau_i(h) = \frac{y_i - y_{i-1}}{h} - \phi(y_{i-1},t_{i-1})$ where the ys are the exact values.
For the Taylor Series method including through nth derivatives (of y), we have
$\tau_i(h) = \frac{y_i - y_{i-1}}{h} - \phi(y_{i-1},t_{i-1})$
Here, $\phi(y_{i-1},t_{i-1}) = y'(t_{i-1}) + \frac{h}{2}y''(t_{i-1}) + ... + \frac{h^{n-1}}{n!}y^{(n)}(t_{i-1})$
$\tau_i(h) = \frac{y_{i-1} + hy'(t_{i-1}) + ... + \frac{h^n}{n!}y^{(n)}(t_{i-1}) + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\xi) - y_{i-1}}{h} - \phi(y_{i-1},t_{i-1})$

$= \frac{h^n}{(n+1)!} y^{(n+1)}(\xi)$.
and the local truncation error is $O(h^n)$.

Runge Kutta Methods

We have seen above that if we have our method match the terms of the Taylor series method through order n, the local truncation error (related to the global error) will be $O(h^n)$.

2 point methods: Midpoint Method, Heun's Method and Modified Euler

We consider methods of the form

$$w_{i+1} \;=\; w_i + ak_1 \;+\; bk_2 \text{ where } k_1 = hf(w_i, t_i) \text{ and } k_2 = hf(w_i + \beta k_1, t_i + \alpha h)$$

That is, we allow for two evaluations of f, one of which is at the current point. Let's see how much of the Taylor series we can match up.

$y_{i+1} = y_i + h\dfrac{dy}{dt} + \dfrac{h^2}{2}\dfrac{d^2y}{dt^2} + \dfrac{h^3}{3!}\dfrac{d^3y}{dt^3} + \ldots$ where all terms are evaluated at (y_i, t_i)

$y_{i+1} = y_i + hf + \dfrac{h^2}{2}(f_t + f_y f) + \dfrac{h^3}{3!}(f_{yy}f^2 + 2ff_{yt} + f_y^2 f + f_y f_t + f_{tt}) + \ldots$

Next, expand w_{i+1}

$w_{i+1} = w_i + ak_1 + bk_2$

$\quad = w_i + ahf(y_i, t_i) + bhf(y_i + \beta hf, t_i + \alpha h)$.

$\quad = w_i + ahf + bh(f + \beta hff_y + \alpha hf_t + \frac{(\beta hf)^2}{2!}f_{yy} + \frac{2\alpha h\beta h}{2!}ff_{ty} + \frac{(\alpha h)^2}{2!}f_t t + \ldots.$

where f alone denotes $f(y_i, t_i)$.

Now, compare y_{i+1} with w_{i+1}, assuming we start at the correct value (i.e., $w_i = y_i$).

Term	y		w				
y_i :	1	$=$	1				
f :	h	$=$	$ah + bh$	$\Rightarrow$	$a + b$	$=$	1
f_t :	$\frac{h^2}{2}$	$=$	$\alpha\beta h^2$	$\Rightarrow$	αb	$=$	$\frac{1}{2}$
f_y :	$\frac{1}{2}h^2 f$	$=$	$b\beta h^2 f$	$\Rightarrow$	βb	$=$	$\frac{1}{2}$

Note that there are too many h^3 terms to match. (We have 3 equations in 4 unknowns but would have to match up $f_{yy}f^2,\ ff_{yt},\ f_y^2 f,\ f_y f_t,\ f_{tt}$ to make 8 equations in 4 unknowns and it cannot be done – not consistent, overdetermined).

All methods satisfying these conditions have their first truncation errors in the h^3 term of the Taylor series, so the local truncation error is $O(h^2)$.

There are many choices we can make satisfying the 3 conditions.

- When we let $a = 0$, $b = 1$, $\alpha = 1/2$ and $\beta = 1/2$ we have the *midpoint method*

$$w_{i+1} \;=\; w_i \;+\; hf(w_i \;+\; \frac{1}{2}hf(w_i, t_i), t_i + \frac{1}{2}h)$$

  Here, half an Euler step was performed to allow us to approximate the derivative in the middle of the interval.

- When we let $a = \frac{1}{2}$, $b = \frac{1}{2}$, $\alpha = 1$ and $\beta = 1$ we have the *Modified Euler method.*

$$w_{i+1} \;=\; w_i \;+\; \frac{h}{2}[f(w_i, t_i) \;+\; f(w_i \;+\; hf(w_i, t_i), t_i + h)]$$

  That is, we average the slope at the beginning and the end of the interval.

- When we let $a = \frac{1}{4}$, $b = \frac{3}{4}$, $\alpha = \frac{2}{3}$ and $\beta = \frac{2}{3}$ we have *Heun's method*

$$w_{i+1} \;=\; w_i \;+\; \frac{h}{4}[f(w_i, t_i) \;+\; 3f(w_i \;+\; \frac{2}{3}hf(w_i, t_i), t_i + \frac{2}{3}h)]$$

Error Analysis for the Midpoint Method

Here, the method is $w_{i+1} \;=\; w_i \;+\; hf(w_i \;+\; \frac{1}{2}hf(w_i, t_i), t_i + \frac{1}{2}h)$.
Thus, $\phi(w_i, t_i) \;=\; f(w_i \;+\; \frac{1}{2}hf(w_i, t_i), t_i + \frac{1}{2}h)$ and
$\tau_{i+1}(h) \;=\; \frac{y_{i+1} - y_i}{h} - \phi(w_i, t_i)$
$\quad = \frac{y_i + hf(y_i, t_i) + \frac{h^2}{2}(f_t + ff_y) + O(h^3) - y_i}{h} - \Big(f(y_i, t_i) + \frac{1}{2}hf_t(y_i, t_i) + f_y(y_i, t_i)\frac{h}{2}f(y_i, t_i) + O(h^2)\Big)$
where the $O(h^3)$ terms in the fraction are $\frac{h^3}{3!}[f_t t + f_{ty}f + f_y(f_t + f_y f) + f(f_{yt} + f_{yy}f)]$
and the $O(h^2)$ terms from ϕ are $\left(\frac{h}{2}\right)^2 f_{tt} + \frac{h}{2}\frac{h}{2}ff_{ty} + \left(\frac{hf}{2}\right)^2 f_{yy}$.
These terms cannot possibly match up so $\tau_{i+1}(h) \;=\; O(h^2)$.

General Setup of Runge-Kutta Methods

Third order Runge-Kutta methods are rarely used. It might be helpful if we set up the pattern for second, third and fourth order Runge-Kutta methods.

- For second order

$$w_{i+1} \;=\; w_i + a_1 k_1 \;+\; a_2 k_2$$

  where

$$\begin{aligned} k_1 &= hf(w_i, t_i) \\ k_2 &= hf(w_i + q_{11}k_1, t_i + p_1 h) \end{aligned}$$

  This has 4 unknowns: a_1, a_2, p_1 and q_{11}.

- For third order

$$w_{i+1} = w_i + a_1 k_1 + a_2 k_2 + a_3 k_3$$

where

$$\begin{aligned} k_1 &= hf(w_i, t_i) \\ k_2 &= hf(w_i + q_{11}k_1, t_i + p_1 h) \\ k_3 &= hf(w_i + q_{21}k_1 + q_{22}k_2, t_i + p_2 h) \end{aligned}$$

This has 8 unknowns: a_1, a_2, a_3, p_1, p_2, q_{11}, q_{21} and q_{22}. (There arise 6 conditions in matching up terms giving, local truncation error $O(h^3)$).

- For fourth order

$$w_{i+1} = w_i + a_1 k_1 + a_2 k_2 + a_3 k_3 + a_4 k_4$$

where

$$\begin{aligned} k_1 &= hf(w_i, t_i) \\ k_2 &= hf(w_i + q_{11}k_1, t_i + p_1 h) \\ k_3 &= hf(w_i + q_{21}k_1 + q_{22}k_2, t_i + p_2 h) \\ k_4 &= hf(w_i + q_{31}k_1 + q_{32}k_2 + q_{33}k_3, t_i + p_3 h) \end{aligned}$$

This has 13 unknowns: a_1, a_2, a_3, a_4, p_1, p_2, p_3, q_{11}, q_{21}, q_{22}, q_{31}, q_{32} and q_{33}. (There arise 11? conditions in matching up terms, giving local truncation error $O(h^4)$).

Fourth Order Runge-Kutta

We see that by including more function evaluations, we can increase the order of Runge-Kutta methods. For the second order RK methods, we had 2 function evaluations. In order to get fourth order derivatives to match, we need 4 function evaluations. Again, the method is not unique. However, there is one version that has become the standard for 4th order Runge Kutta computations.

We let $w_{n+1} = w_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ where

$$\begin{aligned} k1 &= hf(w_n, t_n) \\ k2 &= hf(w_n + \tfrac{1}{2}k_1, t_n + \tfrac{h}{2}) \\ k3 &= hf(w_n + \tfrac{1}{2}k_2, t_n + \tfrac{h}{2})) \\ k4 &= hf(w_n + k_3, t_n + h) \end{aligned}$$

The local truncation error for this method is $O(h^4)$ so, although there are 4 function evaluations per time step, the error is usually much smaller than for second order RK, even for a larger step size.

Some notes on error:

- The error in one step of the above methods is one order better than the "local truncation error". E.g. One step of Euler has error $O(h^2)$ but the local truncation error is $O(h)$. We always lose an order (as we did in integration) when we go "global". Again, in a handwaving manner, there will be order $\frac{1}{h}$ steps times the one-step error to justify losing the order. As we saw for Euler's method, it is really more complicated than that, but it is good enough for us.

- Variable Step Size. One can use the ideas of adaptive integration to vary step size. If y has flat regions and regions where it varies greatly, it might be useful to use smaller step sizes in the regions of greater variability. One can use a numerical method with, say, step size h over parts of the region and then recompute using step size $h/2$. In those regions where the answers differ by more than a certain amount, use a smaller step size. In this way, different parts of your computed solutions will be obtained using different step sizes, but on each region, the error will be roughly the same.

  For example, since Runge-Kutta is fourth order accurate, if we halve the step size, we expect one-sixteenth the error. So if the difference between the coarse and fine results is no more than 15 times the tolerance for this time step, we can be pretty confident that the finer Runge-Kutta value is acceptable.

Predictor-Corrector Methods – Multi-Step Methods

An m-step multistep method is a method of solving $\frac{dy}{dt} = f(y(t), t)$ with $y(a) = \alpha$ of the form:

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \ldots + a_0 w_{i+1-m}$$
$$+ h[b_m f(w_{i+1}, t_{i+1}) + b_{m-1} f(w_i, t_i) + \ldots + b_0 f(w_{i+1-m}, t_{i+1-m})].$$

We need m starting values. Typically, all but the first one needed comes from using one step methods like Runge-Kutta Fourth Order.

Explicit m-step methods: (when $b_m = 0$) $w_{i+1} = F(w_i, w_{i-1}, \ldots, w_{i+1-m})$
Implicit m-step methods: $w_{i+1} = F(w_{i+1}, w_i, w_{i-1}, \ldots, w_{i+1-m})$

Idea: Use an interpolating (Lagrange) polynomial on the m known values of f (e.g. values 0,1,...,m-1) and integrate.

Example: Suppose we know 2 points on the solution, (t_i, y_i) and (t_{i+1}, y_{i+1}). Find (t_{i+2}, y_{i+2}).

- Knowing (t_i, y_i) and (t_{i+1}, y_{i+1}), gives us f_i and f_{i+1} (i.e. $f(y_i, t_i)$ and $f(y_{i+1}, t_{i+1})$).

- We can find the Lagrange polynomial through the points (t_i, f_i) and (t_{i+1}, f_{i+1}).

- Since $\frac{dy}{dt} = f(y(t), t)$ we have $y = \int f(y(t), t)dt$.
- Let $t_0 = 0$, $t_1 = h$ and $t_2 = 2h$. The Lagrange polynomial through (t_i, y_i) and (t_{i+1}, y_{i+1}) is

$$L(t) = f_0 \frac{t - t_1}{-h} + f_1 \frac{t - t_0}{h}$$

- Thus, $y(t_2) \simeq y(t_1) + \int_{t_1}^{t_2} f(y(t), t)dt \simeq y(t_1) + \frac{1}{h} \int_{t_1}^{t_2} [f_0(t_1 - t) + f_1(t - t_0)]dt$
- $y(t_2) \simeq y(t_1) + \frac{h}{2}[3f_1 - f_0]$
- So, the 2 step method derived here is:

$$w_{i+1} = w_i + \frac{h}{2}[3f(w_i, t_i) - f(w_{i-1}, t_{i-1})]$$

We use local truncation error in the form: $\tau_{i+1}(h) = \frac{y_{i+1} - w_{i+1}}{h}$. This becomes, for multistep methods: $\tau_{i+1}(h) = \frac{y_{i+1} - (a_{m-1}w_i + a_{m-2}w_{i-1} + ... + a_0 w_{i+1-m})}{h}$
$-[b_m f(w_{i+1}, t_{i+1} + b_{m-1} f(w_i, t_i) + \ldots + b_0 f(w_{i+1-m}, t_{i+1-m})]$

Thus, the local truncation error is $O(h^2)$ since the Lagrange polynomial has error $O(h^2)$, the integral is over length h making it $O(h^3)$ and we lose a power as always when we apply a method on an interval from a to b. The method just derived is called the *Adams-Bashforth Two Step Predictor.* Notice that this is an explicit scheme.

Note: We will often use the notation f_i to represent $f(w_i, t_i)$.

One can now use the three known values, (t_i, f_i), (t_{i+1}, f_{i+1}) and (t_{i+2}, f_{i+2}) to find a quadratic Lagrange polynomial:
$y(t_{2_{update}}) \simeq y_i + \int_{t_1}^{t_2} [f_0 \frac{(t-h)(t-2h)}{2h^2} - f_1 \frac{t(t-2h)}{h^2} + f_2 \frac{t(t-h)}{2h^2}]dt$
$\simeq y_1 + \frac{h}{12}[5f_2 + 8f_1 - f_0]$ with local truncation error $O(h^3)$.
In our usual notation, this becomes

$$w_{i+1} = w_i + \frac{h}{12}[5f(w_{i+1}, t_{i+1}) + 8f(w_i, t_i) - f(w_{i-1}, t_{i-1})]$$

This method is called the *Adams-Moulton Two Step Corrector.* Notice that this is an implicit scheme.

Note: A simple check one can make that will catch many careless errors in deriving these methods is that they must be able to be simplied to something like $w_{i+1} = w_i + h *$ (average slope) or $w_{i+1} = w_{i-k} + (k+1)h *$ (average slope). If more than one old value of w is used, it is more more complicated.

Example: Consider $\frac{dy}{dt} = -2t - y$ with $y(0) = 1$ and $y(.1) = -0.9145$ from fourth order Runge-Kutta.

- Find $y(.2)$ using Adams-Bashforth 2-step and then update using Adams-Moulton 2-step.
- Since this ODE is linear, we can compute Adams-Moulton without Adams-Bashforth. Find an approximation for $y(0.2)$ using only Adams-Moulton without Adams-Bashforth.

If we try to apply Adams-Moulton alone for a nonlinear ODE, we often cannot solve explicitly. (That's why it is called an *implicit* method.)
For example, consider $\frac{dy}{dt} = \sin y$
We'll get $w_2 = w_1 + \frac{h}{12}[5\sin y_2 + 8\sin y_1 - \sin y_0]$, which we cannot solve for y_2 since the equation is nonlinear and implicit. Using the predictor together with the corrector, gives the higher order error of Adams-Moulton.

<u>Some Adams Formulas:</u>

Steps	Adams Bashforth Explicit	One step error
2	$w_{n+1} = w_n + \frac{h}{2}[3f_n - f_{n-1}]$	$\frac{5}{12}y'''h^3$
3	$w_{n+1} = w_n + \frac{h}{12}[23f_n - 16f_{n-1} + 5f_{n-2}]$	$\frac{3}{8}y^{(4)}h^4$
4	$w_{n+1} = w_n + \frac{h}{24}[55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}]$	$\frac{251}{720}y^{(5)}h^5$

Steps	Adams Moulton Implicit	One step error
2	$w_{n+1} = w_n + \frac{h}{12}[5f_{n+1} + 8f_n - f_{n-1}]$	$\frac{-1}{24}y^{(4)}h^4$
3	$w_{n+1} = w_n + \frac{h}{24}[9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}]$	$\frac{-19}{720}y^{(5)}h^5$
4	$w_{n+1} = w_n + \frac{h}{720}[251f_{n+1} + 646f_n - 264f_{n-1} + 106f_{n-2} - 19f_{n-3}]$	$\frac{-3}{160}y^{(6)}h^6$

A variation on this is to "recorrect", i.e., after using Adams-Moulton, use Adams-Moulton again and again until w_{n+1} stops changing.

Notice that all the Adams methods have $w_{i+1} = w_i + \ldots$. The data at the newest point is used along with slopes at several known points.

One can derive other methods similar to the Adams methods by starting at a point further back and using Newton-Cotes integration formulas. Predictor methods can be based on Newton-Cotes open formulas, while corrector methods can be based on Newton-Cotes closed formulas. For the predictor, the points $x_{i-(n-1)}, \ldots, x_{i-1}, x_i$ and their corresponding f values $f_{i-(n-1)}, \ldots, f_{i-1}, f_i$ are used to predict y_{i+1} based on fitting a Lagrange polynomial through the known (x, f) values and integrating with an open Newton-Cotes formula. Some examples are:

$$
\begin{aligned}
w_{n+1} &= w_{n-1} + 2hf_n \text{Midpoint Method} \\
w_{n+1} &= w_{n-2} + \tfrac{3h}{2}[f_n + f_{n-1}] \\
w_{n+1} &= w_{n-3} + \tfrac{4h}{3}[2f_n - f_{n-1} + 2f_{n-2}]
\end{aligned}
$$

The corrector methods can be based on closed Newton-Cotes formulas. Some examples are:

$$
\begin{aligned}
w_{n+1} &= w_n + hf_{n+1} \text{ An Adams – Moulton Zero Step Backward Euler} \\
w_{n+1} &= w_n + \tfrac{h}{2}[f_{n+1} + f_n] \text{ An Adams – Moulton One Step Trapezoidal} \\
w_{n+1} &= w_{n-1} + \tfrac{h}{3}[f_{n+1} + 4f_n + f_{n-1}] \text{ Implicit Simpson's Method} \\
w_{n+1} &= w_{n-2} + \tfrac{h}{8}[f_{n+1} + 3f_n + 3f_{n-1} + f_{n-2}] \text{Simpson's 3/8 Rule}
\end{aligned}
$$

Another similar method is called Milne's method: $w_{i+1} = w_{i-3} + \frac{4h}{3}[2f_i - f_{i-1} + 2f_{i-2}]$. Here, using the values at $t_{i-2},\ t_{i-1}$ and t_i, a Lagrange polynomial is fit to the values of f and then $w_{i+1} = w_{i-3} + \int_{t_{i-3}}^{t_{i+1}} \textit{Lagrange polynomial } dt$. Milne's method is often used with the Implicit Simpson's Method corrector above which fits a quadratic for f through $t_{i+1},\ t_i$ and t_{i-1}.

Another example is sometimes given Heun's name. The predictor is the Midpoint method $w_{n+1} = w_{n-1} + 2hf_n$ and the corrector is Improved Euler $w_{n+1} = w_n + \frac{h}{2}[f_{n+1} + f_n]$.

Another set of methods are called backward difference formulas. For the derivation, one considers Lagrange polynomials through $y_{n+1}, y_n ...)$ andtakes the derivative at t_{n+1} to get a method of the form

$$
y_{n+1} = \sum_{k=0}^{p-1} \mu_k y_{n-k} + \nu f_{n+1}
$$

These methods are used for stiff ODEs.

Another Way to Define Truncation Error – Epperson

The presentation in Epperson seems better than what I have above.

The residual of a numerical ODE method is given by

$$
R_n = y_{n+1} - \sum_{k=0}^{p} a_k y_{n-k} - hF(y_{n+1}, y_n, ..., y_{n-p}; f_{n+1}, f_n, ...f_{n-p})
$$

i.e., it is the true y_{n+1} minus the computed value.

The truncation error is $\tau_n = \frac{1}{h}R_n$.

The method is consistent if $\lim_{h\to 0} max_{t_n \to T}|t_n| = 0$.

Examples:
Euler's Method:

$$R_i = y_{i+1} - [y_i + hf(y_i, t_i)] = \frac{h^2}{2}y_i''(\xi)$$

and the truncation error is $\frac{h}{2}y_i''(\xi)$ and the method is consistent.

Similarly for Taylor series

$$R_i = y_{i+1} - [y_i + hf(y_i, t_i) + ... + \frac{h^n}{n!}y^{(n)}(t_{i-1})] = \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\xi)$$

and the truncation error is $frach^n(n+1)!y^{(n+1)}(\xi)$ and the method is consistent (for $n \geq 1$).

Consistency, Convergence and Stability

- *Consistent* - The method (i.e., the difference scheme) must be reducible to the ODE as $h \to 0$. $\tau_{i+1}(h) = \frac{y_{i+1}-y_i}{h} - \phi(w_i, t_i) = O(h^p)$ where the method is $w_{i+1} = w_i + h\phi(w_i, t_i)$ and $w_i = y_i$. We say the method is consistent if $p \geq 1$. Recall, another way to write $\tau_{i+1}(h)$ is $\frac{y_{i+1}-w_{i+1}}{h}$.

- *Convergent* – $|y_i - w_i| \to 0$ as $h \to 0$. E.g. Euler's method has $|y_i - w_i| \leq \frac{Mh}{2L}(e^{L(b-a)} - 1) \to 0$ as $h \to 0$. Usually, convergence is very hard to prove.

- *Stable* - Will numerical solutions grow for ODEs with solutions that don't grow? E.g. $y' = -\lambda y$ with $\lambda > 0$. Will small errors in initial conditions yield small changes in the solution for problems in which the exact solution does not grow? A stable method depends continuously on the initial data – similar to well-posedness.

- *Theorem: Convergent* $\Leftrightarrow$ *Consistent* (for one-step methods). E.g. Euler's method has $\tau_{i+1}(h) \simeq O(h)$ so this method is consistent and, therefore, convergent. Modified Euler has $\tau_{i+1}(h) \simeq O(h^2)$ so this method is consistent and, therefore, convergent.

Stability with respect to step size

We consider the stable ODE $y' = -\lambda y$ with $\lambda > 0$ so that solutions go to zero as $t \to \infty$. Let us try to understand (for this ODE) what step size can be used when applying Euler's method.

$w_{i+1} = w_i + hf(w_i, t_i) = w_i - \lambda h w_i = (1 - \lambda h)w_i$. This implies that w_i grows if $|1 - \lambda h| > 1$ and the method is stable for this ODE if $-1 < 1 - \lambda h < 1$ or $0 < h < \frac{2}{\lambda}$. Of course, the answer might not be so accurate if h is close to $\frac{2}{\lambda}$, but at least the computed solution won't grow.

Modified Euler for $y' = -\lambda y$

$$\begin{aligned} w_{i+1} &= w_i + \tfrac{h}{2}(f(w_i, t_i) + f(w_i + hf(w_i, t_i), t_{i+1})) \\ w_{i+1} &= w_i + \tfrac{h}{2}(-\lambda w_i - \lambda w_i - \lambda h(-\lambda w_i)) \\ w_{i+1} &= w_i(1 - h\lambda + \lambda^2 \tfrac{h^2}{2}) \end{aligned}$$

So we need $-1 < 1 - h\lambda + \lambda^2\frac{h^2}{2} < 1$ or $-2 < -\lambda h + \frac{h^2\lambda^2}{2} < 0$.
The inequality on the left always holds. The one on the right is good only if $0 < h < \frac{2}{\lambda}$. So the method is stable for $0 < h < \frac{2}{\lambda}$.

Trapezoidal (Implicit) Method for $y' = -\lambda y$

$$\begin{aligned} w_{i+1} &= w_i + \tfrac{h}{2}(f(w_i, t_i) + f(w_{i+1}, t_{i+1})) \\ w_{i+1} &= w_i + \tfrac{h}{2}(-\lambda w_{i+1} - \lambda w_i) \\ (1 + \tfrac{h}{2}\lambda)w_{i+1} &= (1 - \tfrac{h}{2}\lambda)w_i \\ \text{So, } \; w_{i+1} &= \frac{(1 - \frac{h}{2}\lambda)}{(1 + \frac{h}{2}\lambda)} w_i \;\; \text{but} \\ \left|\frac{(1 - \frac{h}{2}\lambda)}{(1 + \frac{h}{2}\lambda)}\right| &< 1 \;\forall h, \lambda > 0. \end{aligned}$$

So, the Trapezoidal rule is stable for all step sizes (where h and λ are greater than 0). Such a method is called **A-stable**. It turns out that the Trapezoidal method is the only A-stable multi-step method.

Multi-Step Stability Analysis

Suppose the ODE of interest is $\frac{dy}{dt} = f(y, t)$. We will assume that $f \equiv 0$ in order to perform stability analysis. We will analyze whether the numerical scheme will cause small (round-off or truncation) errors to build up, decrease or remain the same.

We assume that the solver (difference equation) has solutions $w_n = \lambda^n$.

Example: Adams-Bashforth Four Step:
$w_{i+4} = w_{i+3} + \frac{h}{24}[55f_{i+3} - 59f_{i+2} + 37f_{i+1} - 9f_i] \Rightarrow \lambda^4 = \lambda^3 + \frac{h}{24}0$.
The roots are $\lambda = 1, 0, 0, 0$.

- If all the $|\lambda|s < 1$ except for one and that one is $\lambda = 1$, the method is *STRONGLY STABLE.*

- If all the $|\lambda|s \leq 1$ and more than one value of λ has $|\lambda| = 1$ but these λs with magnitude 1 are distinct, the method is *WEAKLY STABLE*.
- All other cases give a method that is *UNSTABLE*

Notice that any method that attempts to solve the ODE should give a decent answer for $\frac{dy}{dt} = 0$ so it must have $\lambda = 1$ (i.e., solution is constant).

Since for the Adams-Bashforth Four Step method analyzed above, the polynomial has roots $\lambda = 1, 0, 0, 0$, this method is Strongly Stable.

Example: Milne's Method:
$w_{i+4} = w_i + \frac{4h}{3}[2f_{i+3} - f_{i+2} + 2f_{i+1}] \Rightarrow \lambda^4 = 1 + \frac{4h}{3}0.$
The roots are $\lambda = 1, -1, i, -i$. So Milne's method is Weakly Stable.

Example: Implicit Trapezoidal method:
$w_{i+1} = w_i + \frac{h}{2}(f_i + f_{i+1}) \Rightarrow \lambda = 1 + \frac{h}{2}0.$
This gives $\lambda - 1 = 0 \Rightarrow \lambda = 1 \Rightarrow$ method is Strongly Stable. The trapezoidal method is stable but not very accurate for large step sizes. The local truncation error is $\tau_i(h) = O(h^2)$.

Theorem: If a multi-step method is consistent (reduces to the ODE as $h \to 0$), then the method is stable if and only if it is convergent.
Consider Adams-Bashforth four-step again. The method reduces to $w_{n+1} = w_n + hf_n + \ldots$ or $\frac{dy}{dt} = f(y,t)$ as $h \to 0$, so the method is consistent. It is also (strongly) stable since $\lambda = 1, 0, 0, 0$. Thus, according to our theorem, the method is convergent.

Analyzing the local truncation error of a multi-step method:
To analyze the local truncation error of the method

$$\begin{aligned} w_{i+2} &= 4w_{i+1} - 3w_i - 2hf(w_i, t_i) \\ \tau_{i+2}(h) &= \frac{y_{i+2} - w_{i+2}}{2h} \\ &= \frac{y_i + 2hy_i' + \frac{(2h)^2}{2!}y_i'' + \frac{(2h)^3}{3!}y_i''' + \ldots - 4[y_i + hy_i' + \frac{h^2}{2!}y_i'' + \frac{h^3}{3!}y_i''' + \ldots] + 3y_i + 2hf_i}{2h} \\ &= \frac{4h^3 y_i'''}{6 \cdot 2h} + \ldots = O(h^2) \end{aligned}$$

A-Stability and Stability Regions

We consider the ODE $y' = \lambda y$ with $y(0) = 1$. We let $\xi = \lambda h$ and plot on the real and imaginary axis (in λh) where the method is stable using difference equations and roots of the relevant polynomial.

The stability region is the part of the complex plane where the method is absolutely stable. $|r| < 1$. In general as the accuracy of the method increases, the size of the stability region decreases. (See pp. 361-2, Epperson)

A method is called A-stable if its stability region includes the left half plane, i.e. $\lambda < 0$.

Ex. Trapezoidal method for this ODE becomes: $y_{n+1} \;=\; y_n \;+\; \frac{1}{2}h[\lambda y_n \;+\; y_{n+1}]$
So, $(1 - \frac{1}{2}\xi)y_{n+1} \;=\; (1 + \frac{1}{2}\xi)y_n$
And then, $y_{n+1} \;=\; \dfrac{1+\frac{1}{2}\xi}{1-\frac{1}{2}\xi}y_n$. Since $|\dfrac{1+\frac{1}{2}\xi}{1-\frac{1}{2}\xi}| < 1 \;\forall\lambda \; where \;\; Re(\lambda) < 0$, the Trapezoidal method is A-stable.

Systems of ODEs

All the methods we have used can be applied to systems of ODEs in a straightforward manner. We must solve for all the values at a particular time before moving onto the next time.

For example, in order to use Runge-Kutta for a system, you need to compute all the k_1s before starting on the k_2s since the k_2s can depend on the k_1s from the other equations.
Example: Consider

$$y'' \;-\; 2y' \;+\; 2y \;=\; e^{2t}\sin t \;\;\text{on}\;\; 0 \le t \le 1$$
$$\text{with}\;\; y(0) \;=\; -0.4 \;\;\text{and}\;\; y'(0) \;=\; -0.6$$

- Let $u_1 \;=\; y$ and $u_2 \;=\; y'$
- Then, $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}' \;=\; \begin{pmatrix} u_2 \\ 2u_2 - 2u_1 + e^{2t}\sin t \end{pmatrix}$
- Use Improved Euler (an RK-2 method) to demonstrate working through one time step.

A Lipschitz condition similar to that for the case of a single first order ODE applies to systems and guarantees a unique solution exists for a system of n ODEs if we have $|f(y_1, y_2, \ldots, y_n, t) - f(z_1, z_2, \ldots, z_n, t)| \le L\sum_{i=1}^{n}|y_i - z_i|$.

The error analysis for systems is similar to that for a single ODE.

Gragg Extrapolation for ODEs

This is the same as Richardson Extrapolation (for numerical differentiation) and Romberg Integration (for numerical integration) but when it is used for ODEs it is called Gragg Extrapolation. For example, the Midpoint method has error of the form $k_1h^2 + k_2h^4 + ...$ We can eliminate these terms using different step sizes of h, to improve the results. Of course, we can only improve the results at grid points common to all the step sizes.

Stiff ODEs

ODEs with rapidly decaying transients or with varying time scales can cause problems for many methods. Usually, we are interested only in the steady or near-steady aspects of the solution. One can use VERY small time steps or use an implicit method.

Such problems arise in springs, control systems and chemical kinetics problems.

Example (B&F § 5.11):

$$\begin{array}{llll} u_1' & = 9u_1 + 24u_2 + 5\cos t - \frac{1}{3}\sin t & u_1(0) & = \frac{4}{3} \\ u_2' & = -24u_1 - 51u_2 - 9\cos t + \frac{1}{3}\sin t & u_2(0) & = \frac{2}{3} \\ \text{Exact Solution} & & & \\ u_1(t) & = 2e^{-3t} - e^{-39t} + \frac{1}{3}\cos t & & \\ u_2(t) & = -e^{-3t} + 2e^{-39t} - \frac{1}{3}\cos t & & \end{array}$$

Obviously, the e^{-39t} terms $\to 0$ very quickly. However, they also have very large derivatives so a step size that is too large can cause the ODE solver to overshoot 0 and go very negative and the next step might give a very large negative or, positive but unrealistic, slope and the solution can get worse and worse.

For the above system, one obtains good results with Runge-Kutta (4th order) for step size 0.05 but extremely poor results for $h = 1$.

Generally, one uses implicit multi-step methods for stiff linear ODEs, since they tend to have good stability properties.

Consider $\frac{dy}{dt} = -ay$ using the Backward Euler (aka Implicit Euler).

$$y_{i+1} = y_i + hf(y_{i+1}, t_{i+1}) = y_i - ahy_{i+1}$$

which yields

$$y_{i+1} = \frac{y_i}{1 + ah}$$

Solving a stiff nonlinear ODE: Implicit methods may be combined with Newton's method or Secant method (with w_i as a starting guess, or using a one-step method as a starting guess) for nonlinear ODEs. E.g., consider the Trapezoidal method.

- $w_{i+1} = w_i + \frac{h}{2}[f(w_{i+1}, t_{i+1}) + f(w_i, t_i)]$ but w_{i+1} appears on both sides and is unknown.

- Start with an initial guess and then solve:
  $F(w_{j+1}) = w_{j+1} - w_j - \frac{h}{2}[f(w_{j+1}, t_{j+1}) + f(w_j, t_j)] = 0$ for w_{j+1} using Newton's method.

- This becomes,

$$w_{i+1}^{(k)} = w_{i+1}^{(k-1)} - \frac{F(w_{i+1})}{F'(w_{i+1})}$$

$$w_{i+1}^{(k)} = w_{i+1}^{(k-1)} - \frac{w_{i+1}^{(k-1)} - w_i - \frac{h}{2}[f(w_{i+1}^{(k-1)}, t_{i+1}) + f(w_i, t_i)]}{1 - \frac{h}{2} f_y(w_{j+1}, t_{j+1})}$$

  where the superscript indicates the Newton method iteration step and the subscript indicates the space and time location, as usual.

In some sense, the multi-step methods are more efficient than one-step methods in that for Runge-Kutta, functions are evaluated once and never again used, whereas multi-step methods use the values of the function and its derivative several times.

Some nice problems on multistep methods - from Burden and Faires

1. Given the multistep method

$$w_{i+1} = -\frac{3}{2}w_i + 3w_{i-1} - \frac{1}{2}w_{i-2} + 3hf(w_i, t_i)$$

   for $i = 2, \ldots, N-1$ with starting values w_0, w_1 and w_2. Find the local truncation error. Comment on consistency, stability and convergence. (Assume all previous values are exact and write out Taylor series, match up terms and consider whether $\frac{y_{i+1} - w_{i+1}}{h}$ is $O(h^p)$ where $p \geq 1$).

2. Obtain an approximate solution to the differential equation $y' = y$ on $0 \leq t \leq 10$ where $y(0) = 1$ using Milne's method with $h = 0.1$ and $h = 0.01$, with exact starting values $w_0 = 1$ and $w_1 = e^{-h}$. How does the step size influence the number of correct digits in the computed solutions at $t = 1$ and $t = 10$?

3. Investigate the stability of the difference method:
   $w_{i+1} = -4w_i + 5w_{i-1} + 2h[f(w_i, t_i) + 2hf(w_{i-1}, t_{i-1})]$.

4. Consider the problem $y' = 0$ for $0 \leq t \leq 10$ with $y(0) = 0$, which has solution $y \equiv 0$. Consider the propagation of error for 6 steps for the difference scheme $w_{i+1} = 4w_i - 5w_{i-1}$ where $w_0 = 0$ and $w_1 = \epsilon$.

Boundary Value Problems (BVPs) for ODEs

An application: Heat flow in a rod
Linear

$$T_{xx} + h(T_a - T) = 0 \text{ where } T(0) = T_1 \text{ and } T(L) = T_2$$

Nonlinear

$$T_{xx} + h(T_a - T)^4 = 0 \text{ where } T(0) = T_1 \text{ and } T(L) = T_2$$

Many other applications arise by performing a technique known as Separation of Variables on PDEs.

Here we are given information about the solution to the ODE at the endpoints $y(0)$ and $y(L)$, for instance, for a second order ODE, but not the derivative at $x = 0$.

Linear Shooting Method
Consider $y'' = f(x, y, y')$ on $a \leq x \leq b$ where $y(a) = \alpha$ and $y(b) = \beta$.

Theorem: If f is continuous on the region $\{(x, y, y') | a \leq x \leq b, -\infty \leq y \leq \infty, -\infty \leq y' \leq \infty\}$ andf_y and $f_{y'}$ are also continuous on the region and $f_y > 0$ while $f_{y'} \leq M$ $\forall (x, y, y')$ in the region, then the Boundary Value Problem has a unique solution.

Example: $y'' + e^{-xy} + \sin y' = 0$ on $1 \leq x \leq 2$ with $y(1) = y(2) = 0$.

For the general linear second order ODE, $y'' = p(x)y' + q(x)y + r(x)$, with $y(a) = y_\alpha$ and $y(b) = y_\beta$, where p, q and r are continuous on $[a, b]$ and $q(x) > 0$ on $[a, b]$, the above theorem implies that the ODE BVP has a unique solution.

To investigate how to solve such a problem, consider the ODEs:
$y_1'' = p(x)y_1' + q(x)y_1 + r(x)$ on $[a, b]$ with $y(a) = \alpha$ and $y'(a) = 0$ and
$y_2'' = p(x)y_2' + q(x)y_2$ on $[a, b]$ with $y(a) = 0$ and $y'(a) = 1$

- We can solve these ODE (IVPs) using, for example, Runge-Kutta for systems.
- Notice that letting $Y = y_1 + c_2 y_2$ gives us $Y(a) = \alpha$ and $Y'(a) = c_2$.
- Notice also that Y is a solution of the original ODE. (Just consider $Y'' = y_1'' + c_2 y_2''$ and plug in).
- We would like $Y(b) = \beta$, i.e., require $y_1(b) + c_2 y_2(b) = \beta \Rightarrow c_2 = \frac{\beta - y_1(b)}{y_2(b)}$
- Thus,

$$Y(x) = y_1(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2(x)$$

Nonlinear Shooting Method

Superposition does not hold for solutions to nonlinear ODEs so we cannot use the Linear Shooting Method. Now, we will solve the ODE (by our favorite numerical method, e.g. Runge-Kutta) with $y(a) = \alpha$ but with two guesses for $y'(a)$, say, $y'(a) = t_0$ and $y'(a) = t_1$. We will try to obtain a sequence of $t_0, t_1, \ldots$ such that $\lim_{k\to\infty}[y(b, t_k) - \beta] = 0$, where $y(b, t_k)$ is the computed value of y at $x = b$ using $y'(a) = t_k$.

We apply the Secant method or Regula Falsi on $F(t) = y(b, t) - \beta$ to obtain t_2, then on t_1 and t_2 (or t_0 and t_2 if this brackets the root) to obtain t_3, etc. Here are the details:

- We solve for the line through $y(b, t_0) = y_0$ and $y(b, t_1) = y_1$.
- This line has equation:
  $y - y_0 = m(t_2 - t_0)$
  $y - y_0 = \frac{y_1 - y_0}{t_1 - t_0}(t_2 - t_0)$
- Letting $y = \beta$, we find $t_2 = t_0 + \frac{(t_1 - t_0)(\beta - y_0)}{y_1 - y_0}$.
- So, in general, $t_k = t_{k-2} + \frac{(t_{k-1} - t_{k-2})(\beta - y_{k-2})}{y_{k-1} - y_{k-2}}$ or $t_k = t_{k-1} + \frac{(t_{k-1} - t_{k-2})(\beta - y_{k-1})}{y_{k-1} - y_{k-2}}$.

A variant of this method using Newton's method also can be derived, but another derivative (or an approximation to it) is needed ($\frac{dy}{dt}|_{b,t}$) and must be approximated, so it is more complicated and Secant method is often applied. The Newton's method is

$$t_k = t_{k-1} - \frac{y_{k-1} - \beta}{\frac{dy}{dt}|_{b,t_{k-1}}}$$

Finite Difference Methods for BVPs

The idea here is to apply the difference methods we learned earlier to approximate derivatives in order to solve an ODE-BVP by turning the problem into a system of linear (or nonlinear) equations.

Recall that $y''(x_i) \approx \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2}$ and $y'(x_i) \approx \frac{y(x_{i+1}) - y(x_{i-1})}{2h}$.

The Linear Problem

We *discretize* the ODE on the interval: e.g. $y'' = p(x)y' + q(x)y + r(x)$ with $y(a) = \alpha$ and $y(b) = \beta$. Now, $y(x_0) = y(a) = \alpha$ and $y(x_n) = y(b) = \beta$ are known.

Write the ODE as

$$\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} \;-\; p(x_i)\frac{y(x_{i+1}) - y(x_{i-1})}{2h} - q(x_i)y(x_i) \;=\; r(x_i)$$

The equations become (for x_1, x_i and x_{n-1} where $i = 2, 3, 4, \ldots, n-2$):

$$-\left(\frac{2}{h^2} + q(x_1)\right) y(x_1) + \left(\frac{1}{h^2} \;-\; \frac{p(x_1)}{2h}\right) y(x_2) \;=\; r(x_1) - \frac{\alpha}{h^2} - p(x_1)\frac{\alpha}{2h}$$

$$\left(\frac{1}{h^2} \;+\; \frac{p(x_i)}{2h}\right) y(x_{i-1}) - \left(\frac{2}{h^2} + q(x_i)\right) y(x_i) + \left(\frac{1}{h^2} \;-\; \frac{p(x_i)}{2h}\right) y(x_{i+1}) \;=\; r(x_i)$$

and

$$\left(\frac{1}{h^2} \;+\; \frac{p(x_{n-1})}{2h}\right) y(x_{i-2}) - \left(\frac{2}{h^2} + q(x_{n-1})\right) y(x_{n-1}) \;=\; r(x_{n-1}) - \frac{\beta}{h^2} + p(x_{n-1})\frac{\beta}{2h}$$

This is just a linear system of equations. It is tridiagonal and, thus, extremely efficient to solve using Gaussian elimination. The solution will exist and be unique if $h < \frac{2}{L}$, where $L = max_{a \leq x \leq b}|p(x)|$ and $q(x) > 0$ (By Gershgorin's theorem). The truncation errors in these approximations for y' and y'' are $O(h^2)$ so the method is order h^2. So, in this sense, it is not as good as Runge-Kutta.

We can obtain better approximations by adding more points. Another option is to use higher order discretizations, but this leads to wider bands in the matrix and means something special must be done for more of the points near the endpoints – one-sided derivatives and special cases, which make coding more complicated. It is easier to reduce the step size and perform Richardson extrapolation since we know the order is h^2.