Chapter 4

Differential equations

An Ordinary differential equations (ODE's) is of the form

$$y' = f(x) \tag{4.1}$$

f is a function. The general solution to (4.4) is of the form

$$y = \int f(x)dx + c$$

usually containing an arbitrary constant c. In order to determine the solution uniquely it is necessary to impose an initial condition.

$$y(x_0) = y_0 \tag{4.2}$$

Example The general solution of the equation

$$y' = sin(x)$$

is

$$y = -cos(x) + c$$

If we specify the condition

$$y(\frac{\pi}{3}) = 2$$

then it is easy to find $c = 2.5$.

Thus the desired solution is

$$y = 2.5 - cos(x)$$

$\diamond$

The more general ODE is of the form

$$y' = f(x, y) \tag{4.3}$$

is approached in a similar fashion.

The most general form is

$$\frac{dy_i(x)}{dx} = f(x, y_1, y_2, ..., y_n) \tag{4.4}$$

ie Find $y_i(x)$ for known f_i
The solution isn't completely specified by the ODE.
Therefore we need BOUNDARY CONDITIONS.
Which leads to 2 types of problem

1. initial value problem
   - all the y_i are specified at some point x_{start} or at a set of x-points eg tabulated intervals

2. two - point boundary problem
   - b.c. specified at 2 points typically at x_{start}, x_{finish} we want to know what happens in between.

$\Rightarrow$ We're going to look at the **initial value problems** (IVP) with

1. Euler's method
2. Runge-Kutta method

4.1 Euler Method

We have an ODE of the form

$$y^{'}(x) = f(x, y)$$

and with the initial condition

$$y(a) = A$$

on the interval $[a, b]$.
Euler generates a table of approximate values for $y(x)$.
Suppose this is done for equally spaced values of x
ie choose N values so the separation between each value is

$$h = \frac{(b-a)}{N}$$

and from this we construct the approximations at

$$x = a + nh, \quad n = 0, 1, ...N$$

DRAWING

To arrive at the numerical recipe for Euler's method. We use the Taylor series expansion of $y(x)$ at $x = x_n$
ie y at a point in $[a, b]$ is

$$y(x_{n+1}) = y(x_n) + hy^{'}(x_n) + \frac{h^2}{2!}y^{''}(x_n) + ...$$

y is at a small distance (one step size) from y at x_n
eg $y(x_2)$ is calculated from $y(x_1)$ when the step size is small.
We have $y^{'}(x) = f(x, y)$, therefore

$$y(x_{n+1}) \approx y(x_n) + hy^{'}(x_n) = y(x_n) + hf(x_n, y_n)$$

So the formula for Euler is

$$y_0 = A$$

$$y_{n+1} = y_n + hf(x_n, y_n)$$

this formula advances the solution from x_n to x_{n+1} to ... to x_N.

In full
y_0 at $x_0 = a + 0.h = a$ boundary
next step
y_1 at $x_1 = a + 1.h = a + h$
y_2 at $x_2 = a + 2.h = a + 2h$
.
.
.
y_N at $x_N = a + N.h = b$ boundary

TYPED NOTES

4.2 There is some error incurred with each step of Euler

Therefore we work out the "local truncation error" by comparing the Euler expression for y_{n+1} with the exact expression, ie

$$error = \text{exact answer} - \text{numerical answer}$$

Euler

$$y_{euler}(x_n + h) = y(x_n) + hy^{'}(x_n)$$

Exact expression for $y(x_n + h)$ keeps all higher order terms in Taylor expansion ie

$$y_{exact}(x_{n+1}) = y(x_n) + hy^{'}(x_n) + \frac{h^2}{2!}y^{''}(x_n) + ...$$

Remember, we 'truncated' this series to arrive at Euler.
The error we made when we truncated

$$\begin{aligned} error &= y_{exact}(x_n+h) - y_{euler}(x_n+h) \\ &= y(x_n) + hy^{'}(x_n) + \tfrac{h^2}{2!}y^{''}(x_n) + ... - [y(x_n) + hy^{'}(x_n)] \\ &= \tfrac{h^2}{2!}y^{''}(x_n) + ... \end{aligned}$$

Implies

$$\begin{aligned} error &= (constant)h^2 + O(h^3) \\ &= Kh^2 + O(h^3) \end{aligned}$$

write it in terms of 'h' since this is the only thing we have control over
So if our interval is

$$x_{start} - x_{Finish}$$

stepsize=h

$$\text{Number of steps taken} = \frac{|x_{start} - x_{Finish}|}{h}$$

and each step introduces an error in $y(x_{Finish})$ is

$$\frac{|x_{start} - x_{Finish}|}{h}(Kh^2 + O(h^3))$$

$$= K(x_{start} - x_{Finish})h + O(h^2)$$

Therefore the error is linear in stepsize.
Euler
Conceptually easiest, not the most accurate though.
Runge-Kutta
Propagates a solution at $x = a$ over an interval using info. from several Euler like steps.
NOTES ON EULER CODE.

4.3 can we do better?

We use an Euler- step to take a 'trial' step to the midpoint of the interval.
We use (x, y) at the MIDPOINT to compute 'real' step across the interval

ie to get from (x_n, y_n) to (x_{n+1}, y_{n+1}) use $(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})$
DRAWING
We want to know: y_{n+1}
We know: y_n
we need $\int_{x_n}^{x_{n+1}} f(x,y)dx$
We do this by using the Taylor expansion.
We expand $f(x, y)$ in a Taylor series about the midpoint of the subinterval $[x_n, x_{n+1}]$
ie about $\frac{x_n+x_{n+1}}{2} = x_{n+\frac{1}{2}} = x_n + \frac{h}{2}$

$$f(x,y) \approx f(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}) + (x - x_{n+\frac{1}{2}})\frac{df}{dx} + ...$$

$$\int_{x_n}^{x_{n+1}} f(x,y) \approx \int_{x_n}^{x_{n+1}} f(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}) + \int_{x_n}^{x_{n+1}} (x - x_{n+\frac{1}{2}})\frac{df}{dx} + ...$$

But if $x = x_{n+\frac{1}{2}}$ ie if the integral is evaluated around the midpoint, $(x - x_{n+\frac{1}{2}}) \to 0$

$$f(x,y) \approx f(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})$$

So, substituting in equation for y_{n+1}

$$y_{n+1} = y_n + hf(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})$$

$$= y_n + hf(x_n + \frac{h}{2}, y_{n+\frac{1}{2}})$$

We need

$$y_{n+\frac{1}{2}}$$

Therefore we approximate a value from Euler

$$\begin{aligned} y_{n+\frac{1}{2}} &= y_n + \delta_x y^{'}(x_n) \\ &= y_n + \tfrac{h}{2}y^{'}(x_n) \\ &= y_n + \tfrac{h}{2}f(x_n, y_n) \end{aligned}$$

From this we get a **2nd order Runge-Kutta Method**

$$\begin{aligned} y_{n+1} &= y_n + hf(x_n + \tfrac{h}{2}, y_{n+\frac{1}{2}}) \\ &= y_n + hf(x_n + \tfrac{h}{2}, y_n + \tfrac{h}{2} f(x_n, y_n)) \end{aligned}$$

And if we break it down into stages we have

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_1}{2}) \end{aligned}$$

$$y_{n+1} = y_n + k_2$$

So to start Runge-Kutta we need:
f at midpoint and endpoints
y at previous point
Therefore we can start from an initial condition
EXAMPLE
CODE 2nd order

So far we have 2nd order Runge-Kutta

$$y_{n+1} = y_n + k_2 + O(h^3)$$

We can similarly derive a higher order formulas. The most popular is **4th order Runge-Kutta**

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_1}{2}) \\ k_3 &= hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_2}{2}) \\ k_4 &= hf(x_n + h, y_n + k_3) \end{aligned}$$

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

CODE 4th order

4.4 Solving Differential Equations

Problems involving ODE's can always be reduced to 1st order differential equations.
Example

$$\frac{d^2y}{dx^2} + q(x)\frac{dy}{dx} = r(x)$$

We call

$$z(x) = \frac{dy}{dx}$$

and substitute

$$\frac{dz}{dx} + q(x)z(x) = r(x)$$

This means the original equation is reduced to 2 coupled 1st order equations

$$\begin{cases} \frac{dy}{dx} = z(x) \\ \frac{dz}{dx} = r(x) - q(x)z(x) \end{cases}$$

As before we need an initial condition for each problem therefore we need two initial conditions.

$$y(0) = \alpha \text{ and } \frac{dy(0)}{dx} = \beta$$

which becomes

$$y(0) = \alpha \text{ and } z(0) = \beta$$

We can now use the Euler of Runge-Kutta to approximate the solution. In the case of Euler we have

$$\begin{cases} y_{i+1} = y_i + hz_i \\ z_{i+1} = z_i + h(r(x_i) - q(x_i)z_i) \end{cases}$$

Example

$$y^{''} + 3y^{'} + 2y = e^t$$

with initial conditions $y(0) = 1$ and $y^{'}(0) = 2$ can be converted to the system

$$\begin{aligned} y^{'} &= z & y(0) &= 1 \\ z^{'} &= e^t - 2y - 3z & z(0) &= 2 \end{aligned}$$

the difference Euler equation is of the form

$$\begin{aligned} y_{i+1} &= y_i + hz_i \\ z_{i+1} &= z_i + h(e^{t_i} - 2y_i - 3z_i) \end{aligned}$$

Example A Specific example:

Harmonic oscillator with Friction

Such a system is:

ie. spring stretched distance x_0, released, the position thereafter described by a second order differential equation.

$$\frac{d^2x}{dt^2} + \omega^2 x(t) = 0$$

$$\frac{d^2x}{dt^2} + 2\beta\frac{dx}{dt} + \omega^2 x(t) = 0$$

Q: can we calculate x at the some time t, after release using numerical techniques?

$\Rightarrow$ Reduce equation to first order coupled equations.

$$\frac{d^2x}{dt^2} + 2\beta\frac{dx}{dt} + \omega^2 x(t) = 0 \tag{4.5}$$

say

$$\frac{dx}{dt} = p(t)$$

substituting into 4.5

$$\frac{dp}{dt} + 2\beta p(t) + \omega^2 x(t) = 0$$

we now have our two equations

$$\begin{cases} \frac{dx}{dt} = p & \text{solving gives } x(t) \\ \frac{dp}{dt} = -2\beta p - \omega^2 x(t) & \text{solving gives } p(t) \end{cases}$$

these equations are coupled ie interdependent.

To start: we need initial conditions

$\boxed{\text{at } t = 0}$

$$\begin{aligned} \text{velocity} &= \tfrac{dx}{dt} = p(0) = 0 \\ \text{position} &= x(0) = 1 \end{aligned}$$

We also choose

$$h = 0.1$$

$$2\beta = 0.2$$

$$\omega^2 = 1$$

We can now numerically solve these equations.

$\boxed{t = 0}$

$$x_0 = 1 \quad p_0 = 0$$

$\boxed{t = 0.1}$

First equation

$$\frac{dx}{dt} = p(t)$$

$$\begin{aligned} Euler: \quad x_1 &= x_0 + hf(x_0, t_0) \\ x_1 &= 1 + 0.1p_0 \\ x_1 &= 1 \end{aligned}$$

Second equation

$$\frac{dp}{dt} = -2\beta p(t) - \omega^2 x(t)$$

$$\begin{aligned} Euler: \quad p_1 &= p_0 + hg(p_0, x_0, t_0) \\ p_1 &= 0 + 0.1[-0.2p_0 - 1x_0) \\ p_1 &= 0 + 0.1[0 - 1] \\ p_1 &= -0.1 \end{aligned}$$

Only now we can solve for $\boxed{t = 0.2}$

$$\begin{aligned} Euler: \quad x_2 &= x_1 + hf(x_1, t_1) \\ x_2 &= 1 + 0.1p_1 \\ x_2 &= 1 + 0.1(-0.1) \\ x_1 &= 0.99 \end{aligned}$$

Therefore each step requires solving 2 ODES.

$$\begin{aligned} Euler: \quad p_2 &= p_1 + hg(p_1, x_1, t_1) \\ p_2 &= -0.1 + 0.1[-0.2p_1 - 1x_1) \\ p_2 &= 0 + 0.1[-0.2(-0.1) - 1(1)] \\ p_2 &= -0.198 \end{aligned}$$

$\boxed{t = 0.3}$

.

.

.

Using RK

We use the same method but you have 4 $f(x,t)$ evaluations for each equation solved.

Aside

Interesting physics:

The answer 'x' at each 't' depends on the values of β and ω.

4 cases:

- $\beta^2 = \omega^2$ critical damping
- $\beta^2 > \omega^2$ over critical damping
- $\beta^2 < \omega^2$ under critical damping
- $\beta - 0$ no damping

Our example here:

$$2\beta = 0.2 \Rightarrow \beta^2 = 0.01$$

$$\omega^2 = 1$$

therefore we had under critical damping.