$$u'(t) = f(t, u(t)) \qquad u(a) = u_0$$

ordinary diff. equation

initial condition

Goal: approx on interval $[t_0, T]$.

$t_i = t_0 + hi \qquad h = \frac{T - t_0}{n}$

time grid

$$u(t_{i+1}) - u(t_i) = \int_{t_i}^{t_{i+1}} f(t, u(t))\, dt$$

now approximate this integral...

left endpoint

$$\int_{t_i}^{t_{i+1}} f(t,u(t))\,dt \approx \int_{t_i}^{t_{i+1}} f(t_i, u(t_i))\,dt = h\, f(t_i, u(t_i))$$

assume the interval $[t_i, t_{i+1}]$ is so small that $f(t, u(t))$ is almost constant on that interval.

thus

$$u(t_{i+1}) - u(t_i) \approx h\, f(t_i, u(t_i))$$

Solve for $u(t_{i+1})$ in terms of $u(t_i)$...

$$u(t_{i+1}) \approx u(t_i) + h\, f(t_i, u(t_i))$$

Euler's forward method or explicit method.

Errors here can propagate in out of control way when h is too large.

right endpoint

$$\int_{t_i}^{t_{i+1}} f(t,u(t))\,dt \approx \int_{t_i}^{t_{i+1}} f(t_{i+1}, u(t_{i+1}))\,dt = h\, f(t_{i+1}, u(t_{i+1}))$$

$$u(t_{i+1}) - u(t_i) \approx h\, f(t_{i+1}, u(t_{i+1}))$$

Solve for $u(t_{i+1})$ in terms of $u(t_i)$..

$$u(t_{i+1}) - h\, f(t_{i+1}, u(t_{i+1})) \approx u(t_i)$$

Euler's backwards method or implicit method.

Errors don't go out of control when h is large, but solving the non-linear equations is hard.

Consider the function $g_i(u) = u - h\, f(t_{i+1}, u) - u(t_i)$

Solve $g_i(u) = 0$ to find $u = u(t_{i+1})$

whole bunch of equations to solve...

Try Newton's method to solve..

$$u^{k+1} = u^k - \frac{g_i(u^k)}{g_i'(u^k)}$$

Newton iteration..

$u^0 = u(t_i)$

this should be a good guess because $u(t_{i+1})$ is not far from $u(t_i)$ because u is continuous

Note, that taking h small enough that Newton's method with $u^0 = u(t_i)$ converges to $u(t_{i+1})$ means h can't be too big.

If h is big, maybe use some bisection steps followed by Newton's steps to ensure convergence..

So one can create non-linear root solver's based on Newton's method that don't require h to be that small...

One more idea.., substitute one method into the other

$$u(t_{i+1}) \approx u(t_i) + h\, f(t_i, u(t_i))$$ Explicit

$$u(t_{i+1}) - u(t_i) \approx h\, f(t_{i+1}, u(t_{i+1}))$$ Implicit method

Make implicit method explicit by substituting.

$$u(t_{i+1}) - u(t_i) \approx h\, f(t_{i+1}, u(t_i) + h\, f(t_i, u(t_i)))$$

Solve

composition of f inside f.

$$u(t_{i+1}) \approx u(t_i) + h\, f(t_{i+1}, u(t_i) + h\, f(t_i, u(t_i)))$$

This is another approximation that's explicit and different from Euler's explicit method.

$$u(t_{i+1}) - u(t_i) = \int_{t_i}^{t_{i+1}} f(t, u(t))\, dt$$

now approximate this integral...

Trapezoid method.

$$\int_{t_i}^{t_{i+1}} f(t, u(t))\,dt \approx h\, \frac{f(t_i, u(t_i)) + f(t_{i+1}, u(t_{i+1}))}{2}$$

$$u(t_{i+1}) \approx u(t_i) + h\, \frac{f(t_i, u(t_i)) + f(t_{i+1}, u(t_{i+1}))}{2}$$

Solve for $u(t_{i+1})$ ...

more acurate because used trapezoid rule..

$$\boxed{\underbrace{u(t_{i+1}) - \frac{h}{2} f(t_{i+1}, u(t_{i+1}))}_{\text{implicit method have to solve}} \approx u(t_i) + \frac{h}{2} f(t_i, u(t_i))}$$

$$g(u) = u - \frac{h}{2} f(t_{i+1}, u) - u(t_i) - \frac{h}{2} f(t_i, u(t_i)) = 0$$

for u to find $u(t_{i+1})$.

Now substitute Explicit Euler to obtain an explicit rule...

$$u(t_{i+1}) \approx u(t_i) + h\, f(t_i, u(t_i))$$ bad approximation

$$u(t_{i+1}) - \frac{h}{2} f(t_{i+1}, u(t_{i+1})) \approx u(t_i) + \frac{h}{2} f(t_i, u(t_i))$$ better approximation

f composed with f.

Thus

$$u(t_{i+1}) \approx u(t_i) + \frac{h}{2} f(t_i, u(t_i)) + \frac{h}{2} f(t_{i+1}, u(t_i) + h\, f(t_i, u(t_i)))$$

substituting the bad approximation into the good ones does not spoil the good approximation because where we substituted it is multiplied by h which makes the error smaller.

Next time discuss Butcher Tableau which make a general way of writing compositions of f with f for a approximation

over all s

$$k_i = f\left(t_n + c_i h, y_n + h\sum_{j=1}^{s} a_{ij} k_j\right).$$

Each method listed on this page is defined by its Butche
the method in a table as follows:

$$\begin{array}{c|cccc}
c_1 & a_{11} & a_{12} & \dots & a_{1s} \\
c_2 & a_{21} & a_{22} & \dots & a_{2s} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\
\hline
 & b_1 & b_2 & \dots & b_s
\end{array}$$

en.wikipedia.org/wiki/List_of_Runge–Kutta_methods