Numerical Analysis.
Day 22.

Taylor and Runge-Kutta Methods.

- Start by considering higher order Taylor Approximations - recall Euler was first order.

Do this by example

Ex $y' = y\cos x \quad y(0) = 1$

Write out a Taylor series for the solution near $x = 0$

$$y(h) = y(0) + hy'(0) + \frac{h^2}{2}y''(0) + \frac{h^3}{6}y'''(0) + \cdots$$

(we know this!)

now we need values for y' y'' y''' and so on...

$y'(0) \rightarrow \quad y'(x) = y(x)\cos(x) \longrightarrow y'(0) = y(0)\cos(0) = 1 \cdot 1$

From the ODE! plug in zero. ↑ initial condition.

Take derivative of ODE.

$y''(0) \rightarrow \quad y'' = \frac{d}{dx}\, y\cos(x) = y'(x)\cos(x) - y(x)\sin(x)$

plug in zero $\quad y'(0)\cos(0) - y(0)\sin(0)$

$= y(0)\cos(0)\cos(0) - y(0)\sin(0) = 1 \cdot 1 \cdot 1 - 0$

$= 1$

$y'''(0) \rightarrow y''' = y''(x)\cos(x) - y'(x)\sin(x) - y'(x)\sin(x) - y(x)\cos(x)$

plug in zero $\quad y''(0)\cos(0) - y'(0)\sin(0) - y'(0)\sin(0) - y(0)\cos(0)$

$= 1 \cdot 1 - 1 \cdot 1 = 0$

so $y(h) = 1 + h + \frac{h^2}{2} + 0 + \ldots$ and we have our first step...

we could just keep going... until we are ready to truncate the series.

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

then $y(3h)$, $y(4h)$, etc...

Lets think about this in general: SECOND ORDER TAYLOR METHODS:

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \underbrace{\frac{h^3}{6}y'''(\xi_n)}_{\text{truncation error}}$$

So our approximation is

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

$$y'(x_n) = y(x_n)\cos(x_n)$$

$$y''(x_n) = y'(x_n)\cos(x_n) - y(x_n)\sin(x_n)$$

$$= y(x_n)\cos^2(x_n) - y(x_n)\sin(x_n)$$

So

$$y(x_{n+1}) = y(x_n) + hy(x_n)\cos(x_n) + \frac{h^2}{2}\left[y(x_n)\cos^2(x_n) - y(x_n)\sin(x_n)\right]$$

This is a second order approximation. ∎

Lets consider a higher order example.

Ex Compute a fourth order approximation for $y' = -y \quad y(0) = 1$

You Try... ~~worksheet~~.

1 Write out the Taylor Polynomial to $O(h^5)$ "truncate at h^4"

2. Calculate all the derivatives.

3. Put it all back together.

SOLN: $y_{n+1} = y_n + hy_n' + \frac{h^2}{2}y_n'' + \frac{h^3}{6}y_n''' + \frac{h^4}{4}y_n^{(4)}$

$$= \left(1 - h + \frac{h^2}{2} - \frac{h^3}{6} + \frac{h^4}{24}\right)y_n$$

NOW COMPLETELY GENERAL...

$y' = f(x,y)$: Need this later...

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{6}y'''(x_n) + \cdots$$

so $y'(x_n) = f(x_n, y_n) = f$

$$y''(x_n) = f_x(x_n,y_n) + f_y(x_n,y_n)y'(x_n).$$
$$= f_x + f_y f$$
$$y'''(x_n) = f_{xx}(x_n,y_n) + 2f_{xy}(x_n,y_n)y'(x_n) + f_{yy}y'(x_n)y'(x_n) + f_y(x_n,y_n)f_x(x_n,y_n) + f_y(x_n,y_n)f_y(x_n,y_n)f$$
$$= f_{xx} + f_{xy}f + \underbrace{f_{xy}f + f_{yy}f^2 + f_y f_x + f_y^2 f}_{\text{product and chain rules.}}$$

We could just keep going like this... but it get very messy.

Use to be thought impossible because all was done by hand... but now that computers can do symbolic arithmetic these methods can be automated and are more realistic.

EXPLICIT
RUNGE KUTTA methods – another one step method handles the problem of messy computation AND higher order.

There are some of the most popular for solving ODE's.

We could write any of our taylor methods as

$$y_{n+1} = y_n + h\,F(x_n, y_n, h)$$

where F is a specially chosen approximation to $f(x,y)$ such that this method behaves as a higher order Taylor Method.

NOTE $F(x_n, y_n; h) = f(x_n, y_n) \longrightarrow$ EULER ... first order Taylor.

Let's see if we can find an $F(x_n, y_n; h)$ that is more accurate than EULER.

There are LOTS of ways to pick F...

ORDER 2 Choose

$$F(x,y,h) = \gamma_1 f(x,y) + \gamma_2 f(x+\alpha h,\ y+\beta h f(x,y))$$

(using two points to approximate the slopes... weighted.)

we want to choose γ_1, γ_2, α, and β so that the truncation error is $O(h^3)$. "Weights give us flexibility"

Force This ↓

TRUNCATION ERROR $= T_{n+1} \equiv y_{n+1} - [y_n + hF(x_n, y_n, h)]$

(y_{n+1}: real value; $y_n + hF(x_n, y_n, h)$: value approximated by our method.)

$$T_{n+1} = y_{n+1} - \left[y_n + h\left[\gamma_1 f(x,y) + \gamma_2 f(x+\alpha h,\ y+\beta h f(x,y))\right]\right]$$

USE TAYLOR EXPANSIONS TO COMPUTE ERROR ... out to $O(h^3)$

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2}y''_n + O(h^3)$$

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

$y''_n = f_x + f_y f$ plug in

$$y_{n+1} = y_n + hf + \frac{h^2}{2}(f_x + f_y f) + O(h^3)$$

we also need to expand $f(x+\alpha h,\ y+\beta h f(x,y))$ ... in both y and x ... out to $O(h^2)$ because it is mult by h.

EXPAND IN y ...

$$f(x+\alpha h,\ y+\beta h f(x,y)) = f(x+\alpha h, y) + \beta h f \cdot f_y(x+\alpha h, y) + O(h^2)$$

THEN EXPAND THE RESULT IN x ... term by term ...

$$= f(x,y) + \alpha h f_x(x,y) + \beta h f \cdot f_y(x,y) + O(h^2)$$

(anything w/ h^2 goes here!!)

$$= f + \alpha h f_x + \beta h f f_y + O(h^2) \quad \checkmark$$

PLUG THIS ALL IN ... to T_{n+1}

$$T_{n+1} = \cancel{y_n} + hf + \frac{h^2}{2}(f_x + f_y f) - \cancel{y_n} - h\left[\gamma_1 f + \gamma_2(f + \alpha h f_x + \beta h f f_y)\right] + O(h^3)$$

cancel and gather like terms.

$$T_{n+1} = h(1 - \gamma_1 - \gamma_2)f + \frac{h^2}{2}\left[(1 - 2\gamma_2\alpha)f_x + (1 - 2\gamma_2\beta)ff_y\right] + O(h^3)$$

Force these terms to zero so that $T_{n+1} = O(h^3)$

$$\left.\begin{aligned} 1 - \gamma_1 - \gamma_2 &= 0 \\ 1 - 2\gamma_2\alpha &= 0 \\ 1 - 2\gamma_2\beta &= 0 \end{aligned}\right\}$$ Three eqns for FOUR unknowns...

$$\gamma_2 \neq 0 \;;\; \gamma_1 = 1 - \gamma_2 \;;\; \alpha = \beta = \frac{1}{2\gamma_2}$$

we can choose γ_2 to be anything...

USUALLY $\gamma_2 = 1/2 \Rightarrow \gamma_1 = 1/2$ and $\alpha = \beta = 1$

Leading to... RUNGE KUTTA: (second Order)

$$y_{n+1} = y_n + h\left[\frac{1}{2}f(x_n, y_n) + \frac{1}{2}f(x_n + h,\; y_n + hf(x_n, y_n))\right] \quad n \geq 0$$

$$\text{Error} \approx \frac{1}{3}\left[y_h(x_n) - y_{2h}(x_n)\right]$$

TO PROGRAM THIS...
AT EACH GRID POINT...

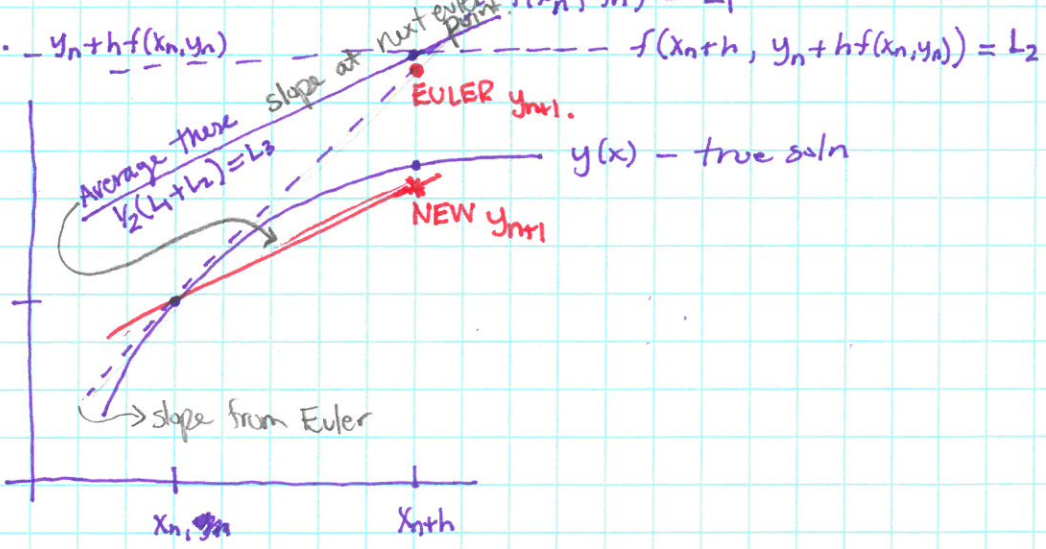
$V_1 = f(x_n, y_n)$ - calculate $f(x,y)$ at the past node

ye $= y_n + h V_1$ - find the y-estimate $ye = y_n + hf(x,y)$

$V_2 = f(x_{n+1}, V_1)$ - find $f(x_n + h,$ y-estimate$)$

$y_{n+1} = y_n + \frac{h}{2}[V_1 + V_2]$ - plug all in... calculate the next step.

GEOMETRICALLY...



A POPULAR METHOD ... fourth order RUNGE-KUTTA.

$$V_1 = f(x_n, y_n)$$
$$V_2 = f(x_n + h/2, \; y_n + h/2 V_1)$$
$$V_3 = f(x_n + h/2, \; y_n + h/2 V_2)$$
$$V_4 = f(x_n + h, \; y_n + h V_3)$$
$$y_{n+1} = y_n + \frac{h}{6}[V_1 + 2V_2 + 2V_3 + V_4]$$

Derived in a similar way.

ERROR DISCUSSION

If a Runge-Kutta method satisfies

$$T_n(Y) = O(h^p)$$

with $p \geq 2$, then it can be shown that

$$|Y(x_n) - y_n| \leq ch^{p-1}, \qquad x_0 \leq x_n \leq b$$

when solving the initial value problem

$$y' = f(x, y), \quad x_0 \leq x_n \leq b, \qquad y(x_0) = Y_0$$

We can also go further and show that

$$Y(x_n) - y_h(x_n) = D(x_n)h^{p-1} + O\left(h^p\right), \quad x_0 \leq x_n \leq b$$

This can then be used to justify Richardson's extrapolation.

For example, if $T_n(Y) = O(h^3)$, then Richardson's error estimate is

$$Y(x_n) - y_h(x_n) \approx \frac{1}{3}\left[y_h(x_n) - y_{2h}(x_n)\right]$$