Mitch Pryor
mpryor@utexas.edu

LECTURE #3

Solving ODEs Numerically

"Now that we have all systems in a common format (s-s form) let's review some methods to solve these systems..."

Recall $\dot{\vec{z}} = \mathbb{A}\vec{z} + \mathbb{B}\vec{u}$ $\quad \vec{y} = \mathbb{C}\vec{z} + \mathbb{D}\vec{u}$ ← "in general, we can handle this even though mostly deal w/ SISO systems"

If discrete

$$\vec{z}[k+1] = \mathbb{A}\vec{z}[k] + \mathbb{B}\vec{u}[k] \quad \vec{y}[k] = \mathbb{C}\vec{z}[k] + \mathbb{D}\vec{u}[k]$$

solve sets of 1st order ODEs (start w/ one)

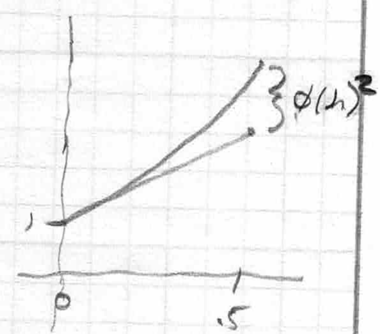
given $\frac{dy}{dt} = f(y,t)$; $y(0) = y_0$ find $y(t) = ?$

Euler's Method (simplified Taylor method)

$$y(t+h) = y_0(t) + h\,y'(t_0) + \underbrace{\mathcal{O}(h^2)}_{\text{neglecting higher order terms.}}$$

Ex) $dy/dx = x + y$ find $y(0.5)$ if $y_0 = 1$

note analytical answer is $2e^x - x - 1$



$\Rightarrow \quad y_e = 1 + (.5)(0+1) = 1.5$

$\quad\quad y_a = 2e^{.5} - .5 - 1 = 1.7974$

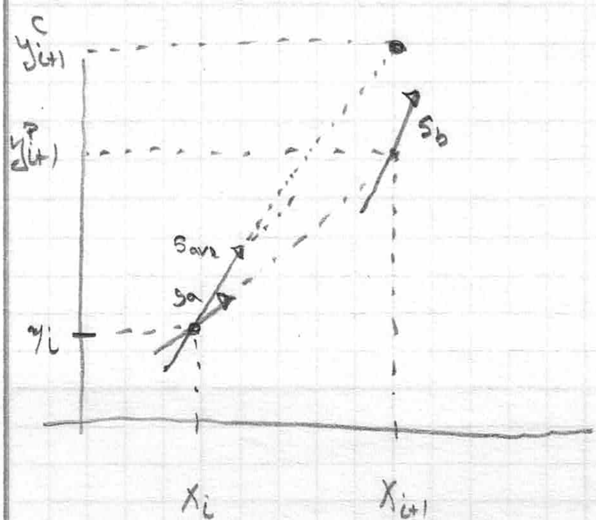
how to improve?
- smaller h
- multiple steps.

try 5 steps w/ $h = 0.02$

Step	x_i	y_i	$y_i' = x_i + 1$
0	0	1	1
1	.02	1.02	1.04
2	.04	1.0408	1.0808
3	.06	1.0624	1.1224
4	.08	1.0848	1.1648
5	1.0	1.1081 ← analytical = 1.1103	

How to make even better?
- smaller steps
- better estimate of the slope.

Heun's Method ≡ determine slope at beginning and end of the interval.



Steps

1) calculate slope @ beginning of the interval.

$$S_a = f(x_i, y_i)$$

2) find y_{i+1} using this slope.

$$y_{i+1}^p = y_i + h\,y_i'$$

3) find slope @ the end of the interval.

$$S_b = f(x_{i+1}, y_{i+1})$$

4) find the average slope

$$S_{ave} = \frac{S_a + S_b}{2}$$

5) use this slope to "correct" the initial guess.

$$y_{i+1}^c = y_i + h\,S_{ave}$$

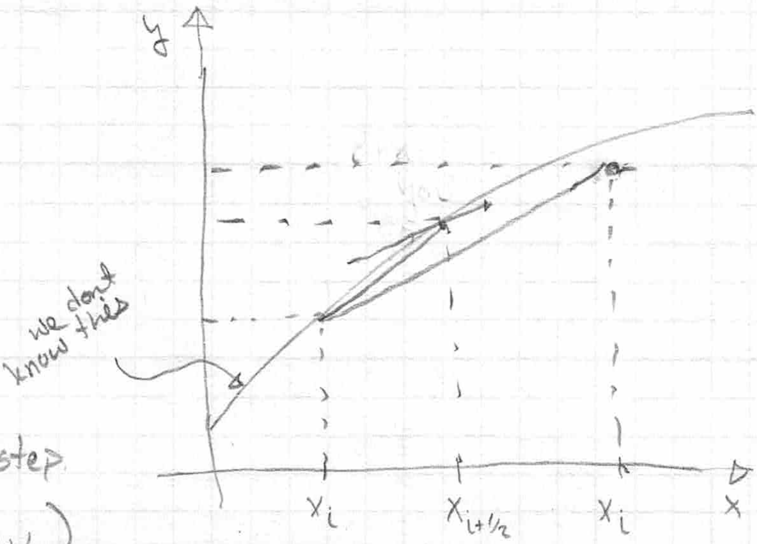
from algorithm → equation

$$y_{i+1} = y_i + h\,S_{ave} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_{i+1}^p)\right)$$

where $y_{i+1}^p = y_i + h y_i'$

"Instead of slope at beginning & end, how about just midway"

Midway Method.



1) Project ½ along interval

$$y_{i+1/2} = y_i + \frac{h}{2} f(x_i, y_i)$$

2) find slope @ midpoint

$$y'_{i+\frac{1}{2}} = f(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}})$$

3) use ~~this~~ slope to find next step

$$y_{i+1} = y_i + h\, f(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}})$$

"Now, combine these strategies into an infinite # of possibilities"

$$y_{i+1} = y_i + h \underbrace{F(x_i, y_i, h)}$$

some generalized estimate of the slope over the interval of interest.

Runga-Kutta Methods.

Ex) find the RK sol^n that includes slopes at the beg., middle, and end of an interval where the middle interval is given twice as much importance.

$$y_{i+1} = y_i + \frac{h}{4}(k_1 + 2k_2 + k_3)$$

where $k_1 = f(x_i, y_i)$

$k_2 = f(x_{i+1/2}, y_i + \frac{1}{2} h k_1)$

$k_3 = f(x_i, y_i + h k_2)$

Switch to slides now!!!

Classical 4th order Runga Kutta Method.

$$y_{i+1} = y_i + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

where $k_1 = f(x_i, y_i)$

$k_2 = f(x_{i+1/2}, y_i + \frac{1}{2}hk_1)$

$k_3 = f(x_{i+1/2}, y_i + 1/2 hk_2)$

$k_4 = f(x_{i+1}, y_i + hk_3)$

EX) given $\frac{dy}{dx} = x + y$; $y(0) = 1$ find $y(.1) = ?$

$k_1 = 0 + 1 = 1$

$k_2 = f(0.05, 1 + \frac{1}{2}(.1)(1)) = 1.1$

$k_3 = f(0.05, 1 + \frac{1}{2}(.1)(1.1)) = 1.105$

$k_4 = f(0.1, 1 + .1(1.105)) = 1.2105$

$$\Rightarrow y(.1) = 1 + \frac{.1}{6}\left(1 + 2(1.1) + 2(1.105) + 1.2105\right) = 1.11034$$

"look familar?"

"That's great but we need to solve systems of 1st order systems..."

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2 \cdots y_n)$$

$$\vdots$$

$$\frac{dy_n}{dx} = f_n(x, y_1, y_2 \cdots y_n)$$

Ex) find y_1 & y_2 @ $x = 1.0$ using Euler's method w $h = 0.5$

$$\frac{dy_1}{dx} = -0.5y_1 \qquad y_1(0) = 4$$

$$\frac{dy_2}{dx} = 4 - 0.3y_2 - 0.1y_1 \qquad y_2(0) = 6$$

solve

step 1

$$\begin{cases} y_1(0.5) = 4 + 0.5(-.5(4)) = 3 \\ y_2(0.5) = 6 + 0.5(4 - 0.3(6) - 0.1(4)) = 6.9 \end{cases}$$

step 2

$$\begin{cases} y_1(1.0) = 3 + 0.5(-.5(3)) = 2.25 \\ y_2(1.0) = 6.9 + 0.5(4 - .3(6.9) - .1(3)) = 7.715 \end{cases}$$

Ex) Repeat using 4th order RK ...

{posted on the web site}

Solving sets of ODEs using MATLAB

Ex)

$$\dot{y}_1 = y_2 \qquad y_1(0) = 4$$

$$\dot{y}_2 = 0.3y_2 - 0.1y_1 + 4\cos(t) \qquad y_2(0) = 0$$

$$[T, Y] = \text{ode45}(\underbrace{@mySystem}_{\text{name of function in working directory}}, \underbrace{[T\_span]}_{\text{time span of interest}}, \underbrace{[Y\_init]}_{\text{initial conditions}})$$

T vector w/ n values

Y - state values at each time step

$$T = \begin{bmatrix} t_0 \\ \vdots \\ t_f \end{bmatrix} \qquad Y = \underbrace{\begin{bmatrix} & \\ & \\ & \end{bmatrix}}_{m \text{ columns w/ } m = \# \text{ states}} \Big\} n \text{ rows}$$

mystem.m

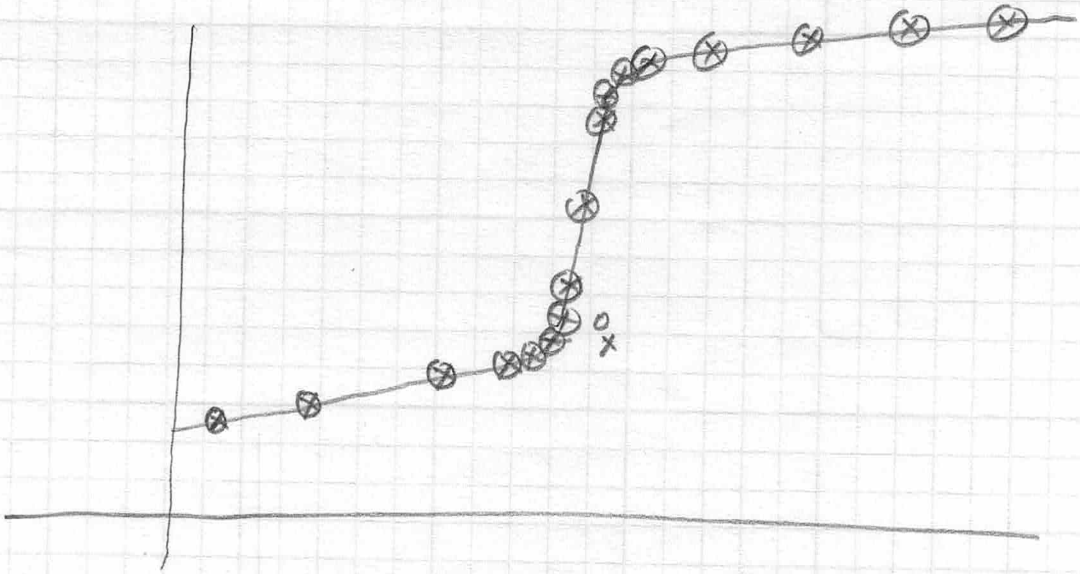
```
function yprime = mysystem (t, y)
    yprime = [ y(2);
              0.3*y(2) - 0.1*y(1) + 4*cos(t)
    ];
```

"so what does ODE45 do?" (

Cash-Karp RK w/ Adaptive step sizing

4th order $y_{i+1} = y_i + h\left[\frac{37}{378}k_1 + \frac{250}{621}k_3 + \frac{125}{594}k_4 + \frac{512}{1771}k_6\right]$

5th order $y_{i+1} = y_i + h\left[\frac{2825}{27{,}848}k_1 + \frac{18575}{48384}k_3 + \frac{13525}{55295}k_4 + \frac{277}{14336}k_5 + \frac{1}{4}k_6\right]$



InitTimeStep
MaxTimeStep
} ODEGET
ODESET

$$h_{new} = h_{present} + \left|\frac{\Delta_{desired}}{\Delta_{actual}}\right|^{\alpha}$$

where

$\alpha = .2$ if $\Delta_d < \Delta_a$

$= .25$ if $\Delta_d > \Delta_a$

"so in homework 1, we learned how to represent any linears system as set of 1st order ODEs, and in homework 2 we will learn to solve any set of 1st order ODEs (numerically)