Math 685. Lecture 12

Project 3 #2

$\tilde{P}\vec{x} = \vec{x}$ $\quad \lambda_{max}(\tilde{P}) = 1$ in Page Rank

or $\lambda_{max}(P) \neq 1$

$\downarrow$ dom. eigenvector

Markov chain: $P\vec{x} = \vec{x}$

$$\begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} \qquad \begin{aligned} P x^{(k)} &= x^{(k+1)} \\ P^{(k)} x_0 &= x^{(k)} \end{aligned}$$

Project 4. "pplane" tool.

Example of ODE system analysis

$$\begin{cases} \dot{x} = x(K-ay) = \overbrace{Kx - axy}^{f(x,y)} \\ \dot{y} = y(L-bx) = \underbrace{Ly - bxy}_{g(x,y)} \end{cases} \qquad \begin{aligned} a&=0.1, K=2 \\ b&=10, L=1 \end{aligned}$$

Equilibrium: $\begin{cases} \dot{x} = 0 \\ \dot{y} = 0 \end{cases}$ $\quad \begin{cases} x(K-ay) = 0 \\ y(L-bx) = 0 \end{cases}$

1) $x = 0 \quad y = 0$

2) $y = \frac{K}{a} = 20 \quad x = \frac{L}{b} = 0.1$

To classify: linearization

$$J = \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix} = \begin{pmatrix} K-ay & -ax \\ -by & L-bx \end{pmatrix}$$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = J \begin{pmatrix} x \\ y \end{pmatrix}$$

linear

$$J = \begin{pmatrix} K-ay & -ax \\ -by & L-bx \end{pmatrix}$$

1) $x=0 \quad y=0 \quad : \quad J = \begin{pmatrix} K & 0 \\ 0 & L \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$

$\lambda_1 = 2 \quad \lambda_2 = 1$

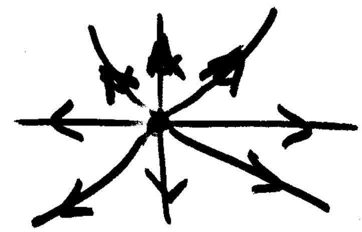
$$\begin{pmatrix} x \\ y \end{pmatrix} = C_1 v_1 e^{+2t} + C_2 v_2 e^{t} \quad \boxed{\longrightarrow +\infty \text{ as } t \to +\infty}$$

eig. vectors

$\longrightarrow 0$ as $t \to -\infty$

Real eig.values, Both positive

$\Longrightarrow$ unstable node

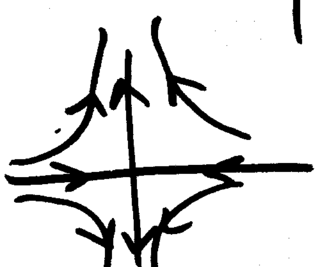


2) $y = \frac{K}{a} = 20, \quad x = \frac{L}{b} = 0.1$

$$J = \begin{pmatrix} 0 & -0.01 \\ -200 & 0 \end{pmatrix}$$

$$\begin{vmatrix} -\lambda & -0.01 \\ -200 & -\lambda \end{vmatrix} = \lambda^2 - 2 = 0.$$

$\lambda = \pm\sqrt{2}$ → real



saddle.

$b \neq 0$ Complex

$\lambda = a \pm \underline{\underline{b}}\, i$

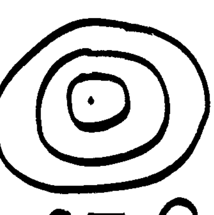
$$\begin{pmatrix} x \\ y \end{pmatrix} = C_1 v_1 e^{at} \cos bt + C_2 v_2 e^{at} \sin bt$$

Spiral | OR center

$a<0$ asymp stable

$a>0$ unstable

$a \neq 0$



$a = 0$

Stable

Methods for num. soln of ODE systems

$\rightarrow \quad y'' + 5y = 0 \quad \Longrightarrow \quad \begin{cases} x_1' = x_2 \\ x_2' = -5x_1 \end{cases} \quad \vec{y}\,' = \vec{f}(t, \vec{y})$

$\begin{pmatrix} x_1 = y \\ x_2 = y' \end{pmatrix}$

$\rightarrow$ Explicit

Forward Euler

$$y_{k+1} = y_k + h f(t_k, y_k)$$

$h = 0.1 \quad \vec{y} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \vec{f} = \begin{pmatrix} x_2 \\ -5x_1 \end{pmatrix}$

$y_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad f(0, y_0) = \begin{pmatrix} 1 \\ -5 \end{pmatrix}$

$y_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.1 \begin{pmatrix} 1 \\ -5 \end{pmatrix}$

If appied to $\boxed{y' = \lambda y, \ \lambda < 0}$

~~Conv. region~~ Stability: $|1 + h\lambda| \le 1$

h has to be <u>small enough</u>

Implicit Schemes

Backward Euler

$$y_{k+1} = y_k + h f(t_{k+1}, y_{k+1})$$

$h = 0.1$

$y_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad f(0, y_0) = \begin{pmatrix} 1 \\ -5 \end{pmatrix} \leftarrow$ not needed

$f(0.1, y_1) = ?$

$\vec{y}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.1\, \vec{f}(0.1, y_1)$

$\begin{pmatrix} y_1^{(1)} \\ y_1^{(2)} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.1 \begin{pmatrix} y_1^{(2)} \\ -5y_1^{(1)} \end{pmatrix}$

$\begin{cases} x = 1 + 0.1y \\ y = 1 - 0.5x \end{cases} \rightarrow \vec{y}_1$ etc.

requires more work, <u>but</u>

Stability ~~Conv. region~~: $\left| \dfrac{1}{1 - h\lambda} \right| \le 1$

<u>h</u> can be anything!

Accuracy still requires h reasonably small.

1) Forward Euler

2) Backward Euler (implicit)

options: Trapezoidal : average of BE & FE

Heun's method - 2step method

$$K_1 = y_k + h_k f(t_k, y_k) \leftarrow \text{FE}$$

$$y_{k+1} = y_k + \frac{h_k}{2}\left(f(t_k, y_k) + K_1 \cdot h_k\right)$$

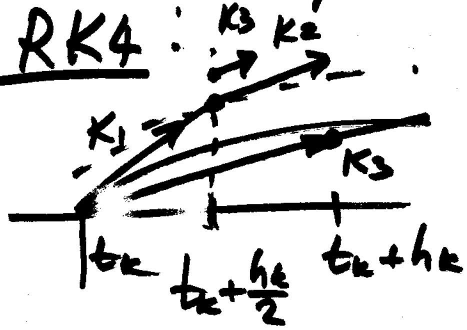
3) Taylor Series method of higher order

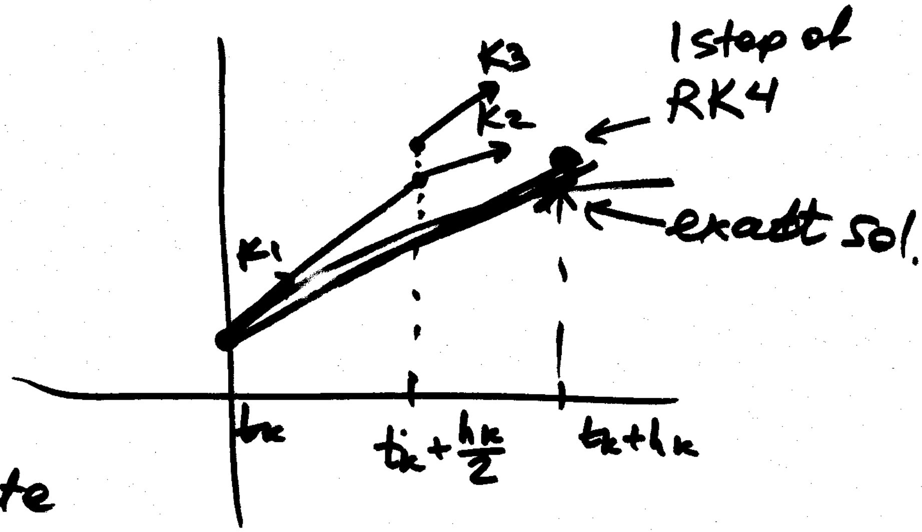
$$y'' = f_t + f_x \cdot f$$

$$y_{k+1} = y_k + h_k f + \frac{h_k^2}{2}(y'') \leftarrow O(h_k^3) \text{ trunc. error}$$

more accurate but more computations are required : f_x, f_t in addition to f.

4) RK4 :



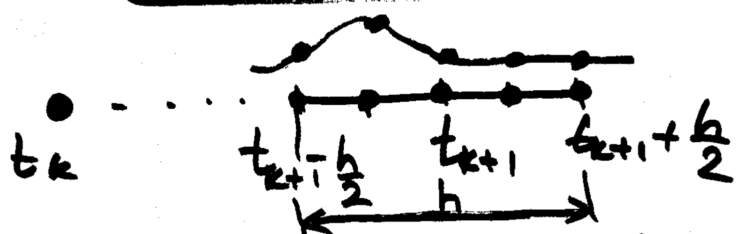


⊕ : very accurate

4th order

⊖ : rigorous stability region estimation is not possible ⇒ step size control is lacking.

5) Extrapolation methods:



(t_j, y_j) close to t_{k+1} where you want to get

$(t_j, y_j) \leftarrow$ fit a poly to this data $\underline{\hat{y}(h)}$

$y_{k+1} \approx \hat{y}(0)$.

$\oplus$ very high accuracy of solution

$\ominus$ no flexibity, no choice of node positions
complexity is high.

6) Multistep method.

PECE

$$y_{k+1} = y_k + \frac{h}{2}(3y'_{k} - y'_{k-1}) \quad (*)$$

$$y_{k+1} = y_k + \frac{h}{2}(y'_{k+1} + y'_k) \quad (**)$$

a) compute $\underline{y_0, y_1}$ somehow (any method), y_0', y_1'
compute predictor: $(*)$ y_2
Compute $y_2' = f(t_2, y_2) \leftarrow$ slope
compute corrector: $(**)$ $y_2 = y_1 + \frac{h}{2}(y_2' + y_1')$
iterate. predictor

$-a < h\lambda < b$, $a, b > 0$
$\uparrow$ stability threshold.

If you pick high accuracy explicit method, $|h\lambda| \ll 1$ becomes too tough to handle.

$$e_{k+1} = C\, O(e_k^p)$$

$\uparrow$ error constant

<u>Multistep methods</u>: changing step size is not easy since coming from uniform interpolation nodes.

<u>Multivalue ~~vals~~ methods</u>:

$$y' = f(t,y)$$

$$\hat{y}_{k+1}(t) = y(t_k+h) = \left[y(t_k) + h y' + \frac{h^2}{2} y'' + \frac{h^3}{6} y''' \right]$$

Taylor

$$\vec{y}_k = \begin{bmatrix} y_k \\ h y_k' \\ \frac{h^2}{2} y_k'' \\ \frac{h^3}{6} y_k''' \end{bmatrix} \qquad \boxed{\hat{\vec{y}}_{k+1} = B \vec{y}_k}$$

$$\vec{y}_{k+1} = \begin{bmatrix} y_{k+1} \\ h y_{k+1}' \\ \frac{h^2}{2} y_{k+1}'' \\ \frac{h^3}{6} y_{k+1}''' \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_k \\ h y_k' \\ \frac{h^2}{2} y_k'' \\ \frac{h^3}{6} y_k''' \end{bmatrix}$$

$$y_{k+1}' = y' + h y'' + \frac{h^2}{2} y'''$$

$$h y_{k+1}' = h y' + \underbrace{h^2 y''}_{2\left(\frac{h^2}{2} y''\right)} + \underbrace{\frac{h^3}{2} y'''}_{3\left(\frac{h^3}{6} y'''\right)}$$

ODE $\underline{y_{k+1}' = f(t_{k+1}, y_{k+1})}$, $\qquad \vec{y}_{k+1} = \hat{\vec{y}}_{k+1} + \alpha \vec{r}$

Line search:

$y_{k+1} = y_k + \alpha_k s$ $\qquad 0 \le \alpha_k \le 1,\ \alpha = 0:0.01:1$

$$\min(f(y_k + \alpha s))$$

$$\alpha = \underset{\alpha}{\operatorname{argmin}}(f(y_k + \alpha s)) = \begin{pmatrix} 0.8 \\ 0.1 \\ 10^{-6} \end{pmatrix}$$

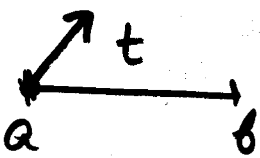
$\boxed{\text{BVP}}$

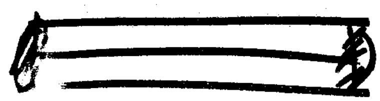
$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \begin{cases} \vec{y}\,' = \vec{f}(t, \vec{y}) \\ \vec{y}(0) = \vec{y}_0 \end{cases}$$

IVP

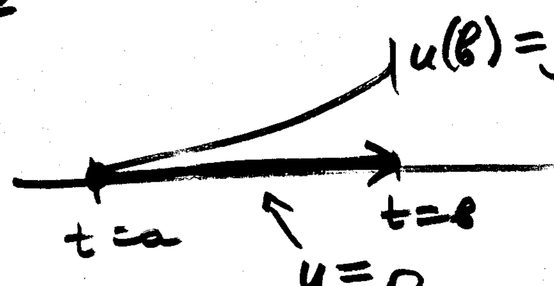
$$\begin{cases} \vec{y}\,' = f(t, \vec{y}) \\ y(a) = y_1 \\ y(b) = y_2 \end{cases}$$

BVP





$$\begin{cases} y(0) = y_0 \\ y'(0) = y_0' \end{cases}$$



← contradicts uniqueness of ODE solutions.

Num. Methods:

1) Shooting : replace BVP with a series of IVPs and correct initial slopes as you go.

$$\begin{cases} u'' = f(t, u, u') \\ u(a) = \alpha,\ u(b) = \beta \end{cases} \longrightarrow \begin{cases} u'' = f(t, u, u') \\ u(a) = \alpha,\ u'(a) = \underline{\underline{\gamma}} \end{cases}$$

BVP

choose γ s.t. $u(b) = \beta$