

COURSE NOTES

NUMERICAL METHODS FOR ENGINEERING

Hanfeng Zhai

Department of Mechanical Engineering, Stanford University

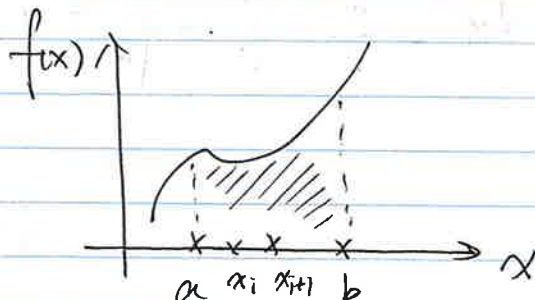
Disclaimer: These notes are intended solely for personal reference and study purposes. They represent my own understanding of the course material and may contain errors or inaccuracies. The content presented here should not be considered as an authoritative source, and reliance solely on these materials is not recommended. If you notice any materials that potentially infringe upon the copyright of others, please contact me at hzhai@stanford.edu so that appropriate action can be taken. Your feedback is greatly appreciated.

October, 2026

Numerical Method Week 3

Integration.

Rectangle Rule.



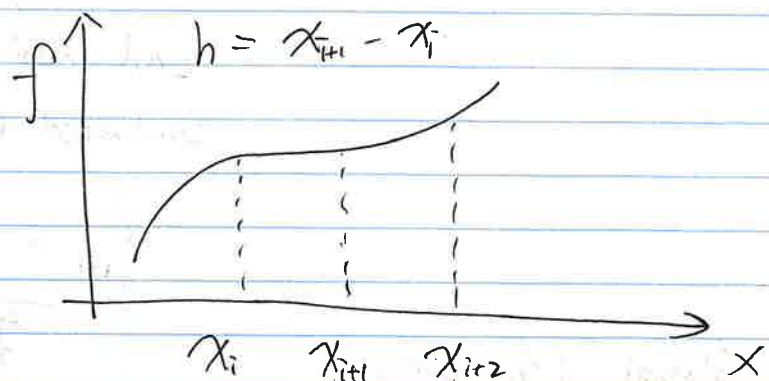
$$\int_{x_i}^{x_{i+1}} f(x) dx = h f(y_i) + \underbrace{\frac{h^3}{24} f''(y_i) + \dots}_{\text{error}}$$

$R(f)$

Trapezoidal rule.

$$T(f) = \int_{x_i}^{x_{i+1}} f(x) dx = \frac{h}{2} [f_{i+1} + f_i] - \frac{h^3}{12} f''(y_i) + \dots$$

Simpson's rule



$$\int_{x_i}^{x_{i+2}} f(x) dx = \frac{h}{3} [f(x_i) + 4f(x_{i+1}) + f(x_{i+2})]$$

$$S(f) = \frac{2}{3} R(f) + \frac{1}{3} T(f) \quad (4^{\text{th}} \text{ Order Method})$$

(Global)

$$= \frac{h}{2} \sum_{i=0}^{n-1} (f_i + f_{i+1}) - \frac{h^2}{12} (f'(b) - f'(a)) + O(h^4)$$

Ex

$$\int_{0.1}^{1.7} \frac{1}{\sqrt{x}} dx$$

9 points.

$$\Delta x = h = 0.2$$

$$x_i = 0.1, 0.3, 0.5, \dots, 1.7.$$

Exact.

$$2x^{1/2} \Big|_{0.1}^{1.7} = (2\sqrt{1.7} - \sqrt{0.1}) = 1.97112$$

Trapezoidal: 2.02018 $\sim$ 2.3% (Error).

Simpson: 1.98558 $\sim$ 0.53%

Trapezoidal w/ n correction:
1.96823 $\sim$ 0.35%.

$\frac{1}{\sqrt{x}}$ is singular near zero, - involving derivatives

brought in troubles $\frac{1}{0}$

$$f(x) = e^x. \quad 0 \leq x \leq 4. \quad \int_0^4 f(x) dx = e^4 - 1 = 53.59815$$

"Rediscovery of the Simpson's Rule".

$$\frac{4\tilde{I}_2 - \tilde{I}_1}{3} = I + \frac{1}{4}C_3 h^4 + \frac{5}{16}C_3 h^6 + \dots \quad (*)$$

this is a 4th order estimate of I

Continue the process.

evaluate the integral with $\frac{h}{4}$ as mesh size

$$\tilde{I}_3 = I - C_1 \frac{h^2}{16} - C_2 \frac{h^4}{256} - C_3 \frac{h^6}{4096} \dots$$

$$\frac{4\tilde{I}_3 - \tilde{I}_2}{3} = I + \frac{h^4}{64} C_2 + \frac{5}{624} h^6 C_3 + \dots \quad (**)$$

Multiply ** by 16 & subtract from *

$$\Rightarrow \dots \mathcal{O}(h^6) \dots$$

$$\mathcal{O}(h^4) \quad \mathcal{O}(h^6)$$

$$\tilde{I}_1$$

$$\tilde{I}_2$$

$$\tilde{I}_3$$

$$I_4$$

$$I_{m,k} = \frac{1}{4^{m-1}-1}$$

$$(4^{m-1} I_{m-1,k+1} - I_{m-1,k})$$

$$m \neq 1$$

Key: With Romberg integration - you can get there exactly !!!

$$\text{Ex.} \int_0^4 \underbrace{\left(\frac{5x^4}{8} - 4x^3 + 2x + 1 \right)}_{f(x)} dx = 72$$

$$I_{11} = \frac{8-0}{2} [f(8) + f(0)] = 2120$$

$$I_{12} = \frac{2120}{2} + \frac{8}{2} f(4) = 712$$

$$I_{13} = 240$$

$$\text{Extrapolate: } I_{21} = \frac{1}{3} [4(712) - 2120] = 242 \frac{2}{3}$$

$$\begin{array}{rcll} 2120 & & & \\ 240 & \searrow & & \\ 712 & \longrightarrow & 242 \frac{2}{3} & \searrow \\ & & & 72 \\ 240 & \searrow & & \\ & \longrightarrow & 82 \frac{2}{3} & \longrightarrow \\ 128 \frac{1}{2} & \searrow & & \\ & \longrightarrow & 72 \frac{2}{3} & \longrightarrow \\ & & & 72 \longrightarrow 72 \end{array}$$

Brute force - you'll never get to the exact soln.
(manually decreasing the mesh size).

Truncation error always exist.

3) Singularity Substitution

$$\int_0^1 x^{-1/2} e^x dx = \int_0^1 \frac{dx}{\sqrt{x}} + \int_0^1 \frac{e^x - 1}{\sqrt{x}} dx$$

"Can do it
analytically"

$$2x^{1/2} \Big|_0^1$$

$$\lim_{x \rightarrow 0} \frac{e^x - 1}{\sqrt{x}} = \lim_{x \rightarrow 0} \frac{\sqrt{x} e^x}{2} = 0.$$

Integrals with infinite limits.

$$\int_0^\infty e^{-x^2} dx.$$

→ map or change of variables.

$$t = \frac{1}{1+x} \quad \text{or} \quad t = e^{-x}$$

Get additional terms due to mapping

(adaptive quadrature — numerical, Gauss Quadrature)

Subdivide the interval (4 panels)

(formula w/ Simpson's rule)

$$S_i^{(2)} = \frac{h_i}{12} \left[f(x_i) + 4f\left(x_i + \frac{h_i}{4}\right) + 2f\left(x_i + \frac{h_i}{2}\right) + 4f\left(x_i + \frac{3h_i}{4}\right) + f(x_i + h_i) \right]$$

Basic idea: - compare the 2 estimates.

$$S_i \text{ \& } S_i^{(2)}$$

⇓

estimate the accuracy.

If accuracy acceptable - take $S_i^{(2)}$ & go to next interval.

If not - Subdivide for next

$$I_i - S_i = c h_i^5 f^{(iv)}\left(x_i + \frac{h_i}{2}\right) + \dots \quad (*)$$

$$I_i - S_i^{(2)} = c \left(\frac{h_i}{2}\right)^5 \left[f^{(iv)}\left(x_i + \frac{h_i}{4}\right) + f^{(iv)}\left(x_i + \frac{3h_i}{4}\right) \right]$$

+ ...

Week 4.

Numerical Soln for ODE.

B.V.P. conditions @ extreme specified.

Initial Value problem. All conditions specified

at same value of independent variable.



ODE: of the form.

$$F(t, y, y', y'', \dots, y^{(i)}) = 0.$$

IVP setup:

$$\frac{dy}{dt} = f(y, t). \quad y(t=0) = y_0.$$

Goal: $y(t)$ for $0 \leq t \leq t_f$.

Numerical Soln: Marching.

Given a soln on $0 \leq t \leq t_n$, what is

$$y_{n+1}' = y_n + h f(y_n, t_n) + \frac{h^2}{2} y_n''$$

2nd order approx.

"Forward Euler!"

$$y'' = \frac{dy'}{dt} = \frac{df'}{dt} \stackrel{\text{chain rule}}{=} \frac{\partial f}{\partial t} \frac{dt}{dt} + \frac{\partial f}{\partial y} \underbrace{\frac{dy}{dt}}_f = f_t + f f_y$$

$$y''' = f_{tt} + 2 f f_{ty} + f_t f_y + f f_y^2 + f^2 f_{yy}$$

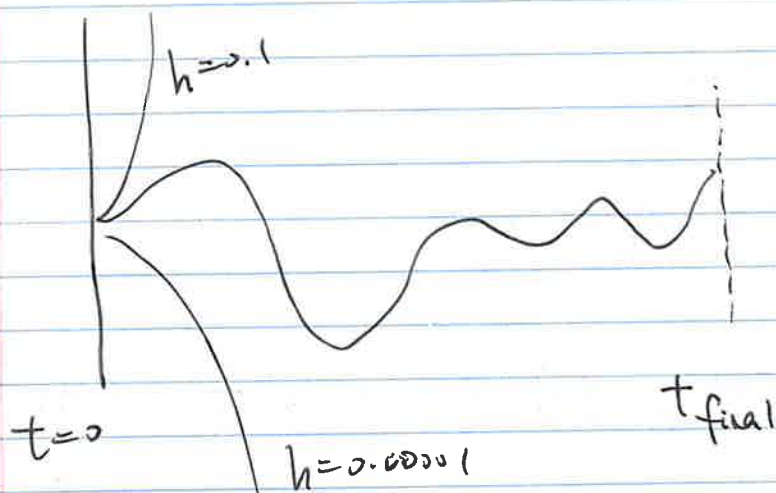
of terms increases rapidly

when we consider higher order approximation.

$$f_t = \frac{\partial f}{\partial t}, \quad f_y = \frac{\partial f}{\partial y}, \quad \dots$$

$$y' = \frac{dy}{dt} \quad f_t = \frac{\partial^2 f}{\partial y \partial t}$$

Stability: Assuming that we are solving for a $y(t)$ that does not blow up.



↙ this numerical method
is not stable

→ Stable, unstable, conditional stable.

Stable: for any choice of parameters, does not blow up. ↗ (at)

Unstable: for any choice of parameters, → blow up.

implications remain roughly unchanged.

Hence, $y' = \lambda y$... the model problem.

$$y' = \lambda y, \text{ I.C.s } y(0) = 1 \rightarrow y(t) = e^{\lambda t}.$$

Consider $y'' + \omega^2 y = 0 \rightarrow$ exact solution

$$y = C_1 \cos(\omega t) + C_2 \sin(\omega t)$$

$$y_1 = y$$

$$y_2 = y'$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}' = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad \text{since } \lambda^2 + \omega^2 = 0$$

$\lambda_{\pm} = \pm i\omega$
 $\hookrightarrow$ eigenvalues are imaginary.

$$\lambda_+ = i\omega$$

$$\lambda_- = -i\omega$$

$$\begin{pmatrix} i\omega & 0 \\ 0 & -i\omega \end{pmatrix}$$

$$A = S \Lambda S^{-1}$$

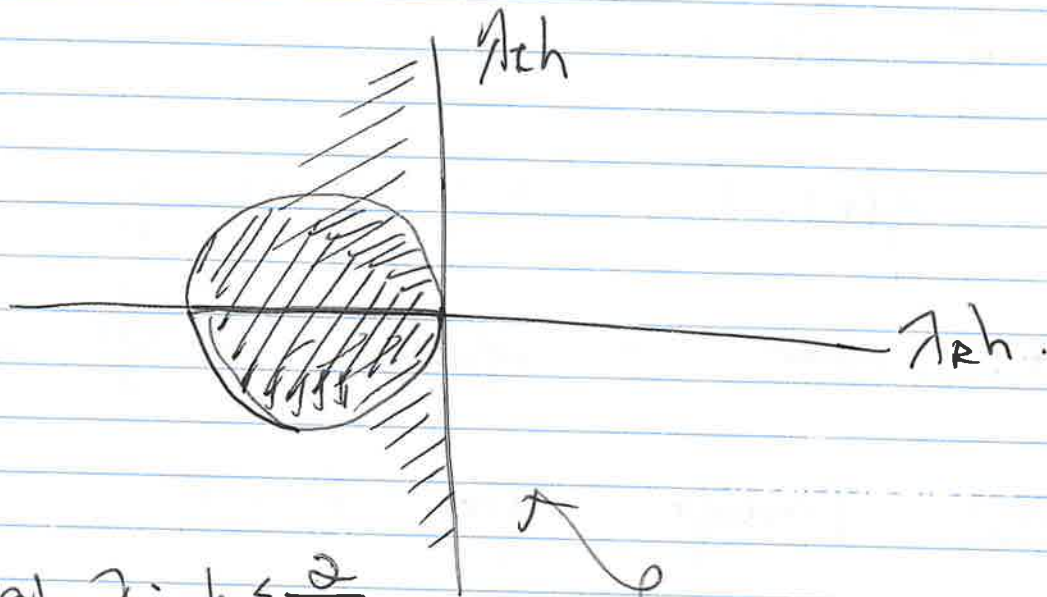
$\nwarrow$ eigenvalues

$$\vec{y}' = S \Lambda S^{-1} \vec{y}$$

$$\vec{z}' = S^{-1} \vec{y}$$

$$S^{-1} \vec{y}' = \Lambda S^{-1} \vec{y} = \vec{z}' = \Lambda \vec{z}$$

Actual stability



Real λ : $h \leq \frac{2}{|\lambda|}$

Explicit Euler

stability region

$$y'' = \frac{d^2 y}{dt^2} = \frac{dy'}{dt} = \frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} \cdot \frac{dy}{dt}$$

Too many terms for high order.

Euler Method:

$$y_{n+1} = y_n + h f(y_n, t_n).$$

Locally, 2nd order.

→ Globally, 1st order.

Other Methods.

Runga - Kutta Methods.

Sol'n at t_{n+1} is obtained as a function of y_n , $f(y_n, t_n)$ & $f(y, t)$ evaluated at intermediate points.

Multi-Step Methods.

Formula involves

$$y_n, y_{n-1}, y_{n-2}, \dots$$

$$f_{n-1}, f_{n-2}, \dots$$

Exact Sol'n.

$$y = y_0 e^{\lambda t}$$

$$= y_0 [e^{\lambda_R t + i \lambda_I t}]$$

$$= y_0 e^{\lambda_R t} \cdot \underbrace{e^{i \lambda_I t}}_{\text{oscillatory}}$$

E.g.

$$y'' + \alpha y' + \beta y = 0$$

Stability,

Euler $y_{m+1} = y_n + h f(y_n, t_n).$

$$= y_n + \lambda h y_n + (1 + \lambda h) y_n.$$

$$\sigma = 1 + \lambda h.$$

amplification factor.

$$y_n = \sigma^n y_0$$

Sol'n of diff eqn.

exact

$$y = y_0 e^{\lambda t}, \quad y_n = y_0 e^{\lambda n h}$$

$$|\lambda h| \leq 2 \Rightarrow h \leq \frac{2}{|\lambda|}$$

Stability Analysis. for Backward Euler.

$$y_{n+1} = y_n + h f(y_{n+1}, t_{n+1})$$

$$y' = \lambda y.$$

$$y_{n+1} = y_n + \lambda h y_{n+1}$$

$$(1 - \lambda h) y_{n+1} = y_n \Rightarrow y_{n+1} = \underbrace{\frac{1}{1 - \lambda h}}_{\sigma} y_n.$$

$$\sigma = \frac{1}{1 - \lambda h - i \lambda h}, \text{ For stability. } |\sigma| \leq 1$$

Quick Rev - complex numbers.

$$Z_1 = a + ib = \sqrt{a^2 + b^2} e^{i\theta}$$

$$\hookrightarrow \theta = \tan^{-1} \frac{b}{a}.$$

$$|e^{i\theta}| = \cos^2 \theta + \sin^2 \theta = 1.$$

Using model equation $y' = \lambda y$

for assessment of accuracy in addition to stability.

Exact sol'n: $y = y_0 e^{\lambda t} = y_0 e^{\lambda n h} = (y_0 e^{\lambda h})^n$

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \dots$$

α
↓

→ Explicit Euler $= y_0 \left(1 + \lambda h + \frac{\lambda^2 h^2}{2} + \frac{\lambda^3 h^3}{6} + \dots \right)^n$

$$y_n = y_0 (1 + \lambda h)^n \rightarrow \text{convention:}$$

compare amplification factor

to the exponential expansion

→ α^{th} order if one leaves: up to & including $\frac{\lambda^{\alpha} h^{\alpha}}{\alpha!} \dots$

Backward Euler

$$D = \frac{1}{1 - \lambda h}$$

$$G = A e^{i\theta}$$

Phase error.

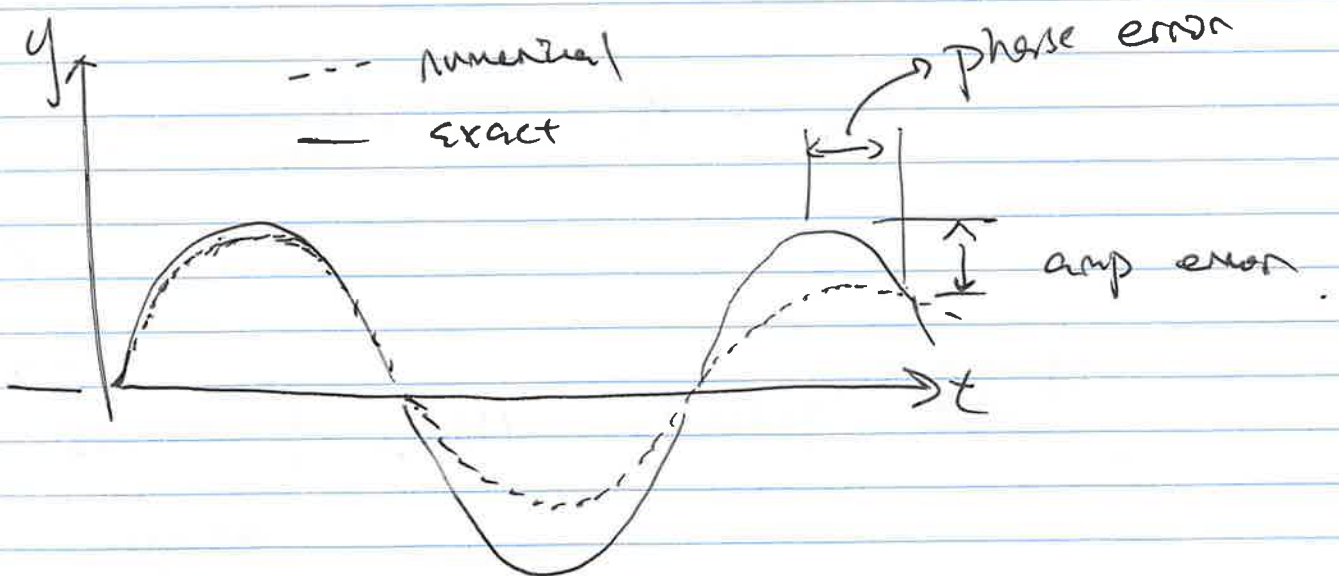
$$= \omega h - \theta$$

$$\theta = \tan^{-1} \omega h$$

$$= \omega h - \frac{(\omega h)^3}{3} + \frac{(\omega h)^5}{5} + \dots$$

Phase error.

$$P.E. = \frac{(\omega h)^3}{3} + \dots$$



P.E. > 0 → phase lag.

$$\lambda = \lambda_R + i\lambda_I.$$

$$e^{\lambda h} = 1 + \lambda h + \frac{\lambda^2 h^2}{2} + \frac{\lambda^3 h^3}{3} + \dots$$

Exact
sol'n.

first order accurate.

Forward & Backward Euler

Phase error

$$\lambda = i\omega. \quad \text{Exact sol'n. } e^{i\omega t}$$

$$\text{Amp} = 1$$

More accuracy ... trapezoidal method.

$$y_{n+1} = y_n + \frac{h}{2} [f(y_{n+1}, t_{n+1}) + f(y_n, t_n)]$$

Apply to the model equation.

$$y' = \lambda y \rightarrow y_{n+1} = y_n + \frac{h}{2} [\lambda y_{n+1} + \lambda y_n]$$

$$|\sigma| = \frac{|A|}{|B|} < 1 \quad \text{Since } |B| > |A|$$

trapezoidal method is unconditionally stable

σ is always ≤ 1 .

Accuracy

$$\sigma = \frac{1 + \lambda h/2}{1 - \lambda h/2}$$

$$= \left(1 + \frac{\lambda h}{2}\right) \left[1 + \frac{\lambda h}{2} + \frac{\lambda^2 h^2}{4} + \frac{\lambda^3 h^3}{8} + \dots\right] \frac{1}{1 - x} = 1 + x + x^2 + \dots$$

$$= 1 + \frac{\lambda h}{2} + \frac{\lambda^2 h^2}{4} + \frac{\lambda^3 h^3}{8} + \dots + \frac{\lambda h}{2}$$

$$+ \frac{\lambda^2 h^2}{8} + \frac{\lambda^3 h^3}{16} + \dots$$

$$= 1 + \lambda h + \frac{\lambda^2 h^2}{2} + \frac{1}{4} \lambda^3 h^3 + \dots \quad \text{2nd order}$$

compare to $e^{\lambda h} = 1 + \lambda h + \frac{\lambda^2 h^2}{2} + \frac{\lambda^3 h^3}{6} + \dots$

↑ real & negative.

$\lim_{h \rightarrow \infty} \sigma = -1 \Rightarrow$ soln oscillates, but
does not blow up.

Backward Euler. $\sigma = \frac{1}{1 - \lambda h}$.

Explicit Euler $\sigma = 1 - \lambda h$.

↖ blow up

Implicit Scheme

$$y_{n+1} = y_n + h f(y_{n+1}/t_{n+1})$$

↖

@ every time step

(if nonlinear) you have to solve
the ablin. eqn. (iterative - N.R., ...)

Predictor - Corrector Scheme. (RK4)

$$= y_n + \frac{h}{2} \left(\lambda(1+\lambda h) y_n + \lambda y_n \right)$$

$$= y_n \left(1 + \lambda h + \frac{\lambda^2 h^2}{2} \right)$$

2nd order accuracy.

→ ... Stability, accuracy, phase error.

Runge-Kutta: 2nd order.

$$y_{n+1/2}^* = y_n + \frac{h}{2} f(y_n, t_n).$$

$$y_{n+1} = y_n + h f(y_{n+1/2}^*, t_{n+1})$$

Apply to model eqn.

midpoint, -inspired.

$$y' = \lambda y, \quad y_{n+1/2}^* = y_n + \frac{\lambda h}{2} y_n = \left(1 + \frac{\lambda h}{2} \right)$$

Most famous algorithm of this scheme,

4th order R.K. (RK4)

$$y_{n+1/2}^* = y_n + \frac{h}{2} f(y_n, t_n)$$

$$y_{n+1/2}^{**} = y_n + \frac{h}{2} f(y_{n+1/2}^*, t_{n+1/2})$$

$$y_{n+1}^{***} = y_n + h f(y_{n+1/2}^{**}, t_{n+1/2})$$

$$y_{n+1} = y_n + \frac{1}{6} h \left[f(y_n, t_n) + 2f(y_{n+1/2}^*, t_{n+1/2}) + 2f(y_{n+1/2}^{**}, t_{n+1/2}) + f(y_{n+1}^{***}, t_{n+1}) \right]$$

4 function evaluation
@ each step.

Week 5. Lecture 2.

Runge - Kutta Method. (another notation).

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

$$k_1 = hf(y_n, t_n).$$

$$k_2 = hf(y_n + \beta k_1, t_n + \alpha h).$$

unknown coefficients.

$$y_{n+1} = y_n + \gamma_1 k_1 + \gamma_2 k_2.$$

γ_1 & γ_2

To get the coefficients, are unknown

Compare to Taylor series.

$$k_2 = h \left[f(y_n, t_n) + \beta k_1 \frac{\partial f}{\partial y} \Big|_n + \alpha h \frac{\partial f}{\partial t} \Big|_n + \dots \right]$$

3 eqns for 4 unknowns.

Choose α as a free parameter

$$\gamma_2 = 1/2\alpha, \quad \beta = \alpha, \quad \gamma_1 = 1 - \frac{1}{2\alpha}$$

R-K formula:

$$K_1 = hf(y_n, t_n)$$

$$K_2 = hf(y_n + \alpha K_1, t_n + \alpha h)$$

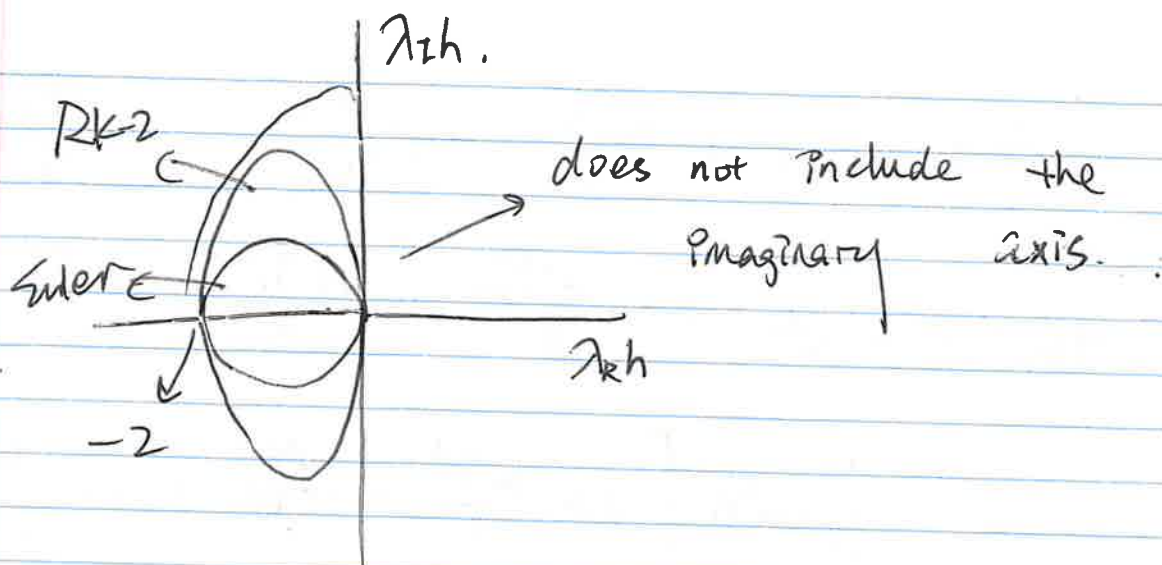
$$y_{n+1} = y_n + \left(1 - \frac{1}{2\alpha}\right) K_1 + \frac{1}{2\alpha} K_2$$

$\alpha = 1/2 \Rightarrow$ the R-K formula from last lecture

Apply this to Model equation. $y' = \lambda y$.

$$K_1 = h \lambda y_n$$

$$\begin{aligned} K_2 &= \lambda h [y_n + \alpha K_1] = \lambda h [y_n + \alpha h \lambda y_n] \\ &= y_n (\lambda h + \alpha \lambda^2 h^2) \end{aligned}$$



$$\text{Euler: } \sigma = 1 + \lambda h \Rightarrow |\sigma| = \sqrt{1 + \omega^2 h^2}$$

Sol'n after 100 timesteps.

$$\rightarrow y_0 |\sigma|^{100}$$

ωh always < 1 .
in num. meth.

* (accuracy, stability, phase error)

$$\sigma = |\sigma| e^{i\theta}$$

$$\theta = \tan^{-1} \left[\frac{\text{img}(\sigma)}{\text{real}(\sigma)} \right] = \tan^{-1} \left[\frac{\omega h}{1 - \frac{\omega^2 h^2}{2}} \right]$$

$$\text{P.E.} = \omega h - \tan^{-1} \left[\frac{\omega h}{1 - \frac{\omega^2 h^2}{2}} \right]$$

$$k_3 = hf(y_n + \frac{1}{2}k_2, t_n + h/2)$$

$$k_4 = hf(y_n + k_3, t_n + h)$$

$$y_{n+1} = y_n + \frac{1}{6}k_1 + \frac{1}{3}(k_2 + k_3) + \frac{1}{6}k_4$$

main cost: function evaluations.

↳ Multi-Step Methods.

Formula includes information from time

Steps $n-1, n-2$, in addition to $n, n+1$

implicit

... Leap-frog scheme

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

$$\frac{y_{n+1} - y_{n-1}}{2h} = f(y_n, t_n)$$

Week 6 Lecture 2.

Multistep Methods.

$$y' = f(y, t), \quad y(0) = y_0$$

$t_n \rightarrow t_{n+1}$ leap frog scheme

$$\frac{y_{n+1} - y_{n-1}}{2h} = f(y_n, t_n)$$

$$y_{n+1} = y_{n-1} + 2h f(y_n, t_n)$$

∴ Not self-starting.

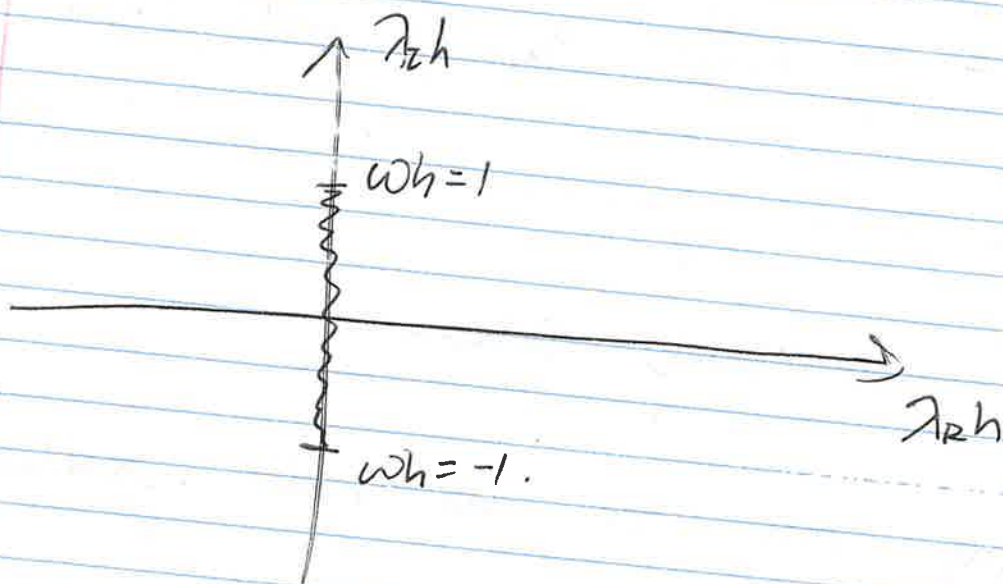
Use another method (e.g., Euler).

$t_0 \rightarrow t_1$

apply this to model problem. $y' = \lambda y$

$$C_1 = \frac{y_1 - y_0 \sigma_2}{\sigma_1 - \sigma_2}$$

$$C_2 = \frac{\sigma_1 y_0 - y_1}{\sigma_1 - \sigma_2}$$



$$\lambda = i\omega$$

2nd order Adam Bashforth.

Recall: no amp. error.

$$y_{n+1} = y_n + h y'_n + \frac{h^2}{2} y''_n + \frac{h^3}{6} y'''_n + \dots$$

$$\begin{array}{ccc} \swarrow & & \downarrow \\ f(y_n, t_n) & & \frac{h^2}{2} \left[\frac{y'_n - y'_{n-1}}{h} \right] \end{array}$$

$$y_{n+1} = y_n + h f(y_n, t_n) + \frac{h^2}{2} [f(y_n, t_n) - f(y_{n-1}, t_{n-1})] + \mathcal{O}(h^3)$$

func eval. + cost + stability.

Quadratic:

↑
order

$$\sigma = \frac{1}{2} \left[1 + \frac{3\lambda h}{2} \pm \sqrt{1 + \lambda h + \frac{9}{4}\lambda^2 h^2} \right]$$

Recall $\sqrt{1+x} = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{3}{64}x^3 + \dots$

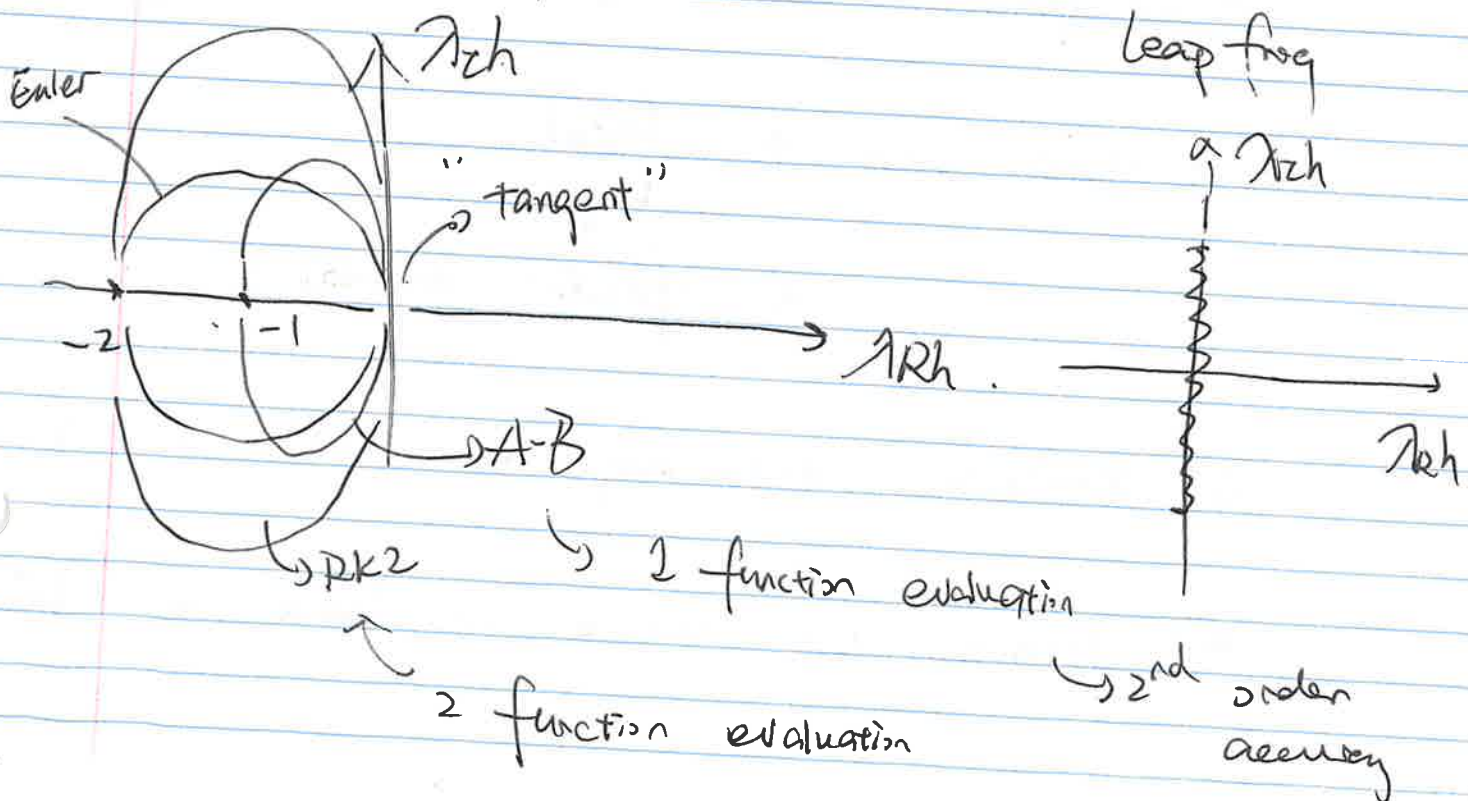
$$\sigma_1 = 1 + \lambda h + \frac{\lambda^2 h^2}{2} + \mathcal{O}(h^3) \quad \text{near root}$$

$$\sigma_2 = \frac{1}{2}\lambda h - \frac{1}{2}\lambda^2 h^2 + \mathcal{O}(h^3)$$

Note that $\sigma_2 \rightarrow 0$ as $h \rightarrow 0$

... good !!!

Stability analysis.



M-evaluation numbers. #. (cost).

i.e.g., $R-k=4$ is a 4-evaluation method per step

$R-k=2$ is a 2-eval. ...

e.g., $R-k=2$ $h = \frac{\pi}{M/2}$ $M/2$ steps.

$$\text{Amp error} = 1 - |\sigma(kwh)|^{M/k}$$

k : # of eval

Phase error

of the method per step.

$$\frac{M}{k} \left[kwh - \tan^{-1} \left(\frac{\sigma_z(kwh)}{\sigma_p(kwh)} \right) \right]$$

Leapfrog is a 2-eval method.

System of differential equations.

ordinary.

High order reduce to system

$$\left. \begin{array}{l} y_1 = f \\ y_2 = f' \\ y_3 = f'' \end{array} \right\} \rightarrow \left\{ \begin{array}{l} y_1' = y_2 \\ y_2' = y_3 \\ y_3' = y_1 y_2 \end{array} \right.$$

Single eqn.

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

System

$$\vec{y}' = f(\vec{y}, t).$$

$$\frac{dy_i}{dt} = f_i(y_1, y_2, \dots, y_m, t)$$

$$i = 1, 2, 3, \dots, m.$$

Model eqn.

$$y' = Ay$$

$$\frac{d\vec{y}}{dt} = A\vec{y}$$

↓

A is $m \times m$ matrix

going back to the model eqn.

for systems. $\vec{y}' = A \vec{y}$

$$A = S \Lambda S^{-1} \quad \Leftarrow \text{constant matrix}$$

no t 's.

$$\frac{d\vec{y}}{dt} = S \Lambda S^{-1} \vec{y}$$

$$S^{-1} \frac{d\vec{y}}{dt} = \Lambda S^{-1} \vec{y}$$

let $S^{-1} \vec{y} = \vec{u}$

$$\frac{d\vec{u}}{dt} = \Lambda \vec{u} \Rightarrow \frac{du_i}{dt} = \lambda_i u_i$$

eigenvalues of A

Soln: $u_i = C_i e^{\lambda_i t}$

modes.

$$\vec{y} = S \vec{u} = C_1 e^{\lambda_1 t} \vec{s}_1 + C_2 e^{\lambda_2 t} \vec{s}_2 + \dots$$

linear combinations.

Linear case.

$$\vec{y}' = A \vec{y}$$

$\underbrace{\hspace{1.5cm}}_{n \times n \text{ matrix.}}$

Single ODE.

$$y' = \lambda y \rightarrow \lambda = \frac{dy}{dt}$$

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

Euler:

$$\vec{y}^{n+1} = \vec{y}^n + h A \vec{y}^n$$

$$= (I + h A) \cdot \vec{y}^n$$

$\uparrow$ identity matrix.

$$\vec{y}^n = \underbrace{(I + h A)^n}_B \vec{y}_0$$

Single eqn.

$$y^n = \underbrace{(1 + \lambda h)^n}_{\text{eigenvalue}} y_0$$

$$\vec{y}^n = B^n \vec{y}_0 \rightarrow \text{eigs of } B$$

Stability. $| \sigma | \leq 1$.

For stability $| \lambda_i | \leq 1$

If need to integrate the equation, in time
& time-final is large

$$\text{if } \frac{|\lambda|_{\max}}{|\lambda|_{\min}} \gg 1 \Rightarrow \text{Problem is Stiff}^n$$

many time steps are required to
reach t_f

Ex. 9. System of 2 eqns.

$$u' = 998u + 1998v \quad u(0) = 1$$

$$v' = -999u - 1999v \quad v(0) = 1$$

$$A = \begin{bmatrix} 998 & 1998 \\ -999 & -1999 \end{bmatrix}$$

$$\lambda_1 = -1 \quad \& \quad \lambda_2 = -1000$$

$$\text{For Euler: } h_{\max} = \frac{2}{1000} = \frac{1}{500}$$

you can change time-steps sizes, but
the round-off error will initiate the
second mode $\rightarrow$ not good!

... you'll have to go

with implicit method

Soln to Stiffness problem

but you'll have to solve

nonlinear eqn per time steps

iterative methods, e.g., Newton Raphson

For non-linear $f(y, t)$, have to solve a

nonlinear eqn @ each step.

$\Rightarrow$ inconvenience & costly.

Iterative soln methods ...

$$y_{n+1} = y_n + \Delta t y'_n + \frac{\Delta t^2}{2} y''_n + \dots$$

trapezoidal:

$$y_{n+1} - y_n = \frac{\Delta t}{2} \left[f(y_n, t_{n+1}) + \frac{\partial f}{\partial y}(y_{n+1} - y_n) + f(y_n, t_n) \right] + \mathcal{O}(\Delta t^3)$$

$$\left[1 - \frac{\Delta t}{2} \frac{\partial f}{\partial y} \Big|_n \right] (y_{n+1} - y_n) = \frac{\Delta t}{2} \left[f(y_n, t_{n+1}) + f(y_n, t_n) \right]$$

$$y_{n+1} = y_n + \frac{\Delta t}{2} \frac{f(y_n, t_{n+1}) + f(y_n, t_n)}{1 - \frac{\Delta t}{2} \frac{\partial f}{\partial y} \Big|_n}$$

Apply this linearization to a system of equations

Coupled set of ODEs

$$\frac{d\vec{u}}{dt} = \vec{f}(\vec{u}) = \vec{f}(u_1, u_2, \dots, u_m)$$

↑ linearize

elements of $A_n =$
$$\begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \dots & \frac{\partial f_1}{\partial u_n} \\ \vdots & & & \vdots \\ \frac{\partial f_m}{\partial u_1} & \frac{\partial f_m}{\partial u_2} & \dots & \frac{\partial f_m}{\partial u_n} \end{bmatrix}$$

Apply to trapezoidal method

$$\vec{u}' = \vec{f}(\vec{u})$$

$$\frac{\vec{u}^{n+1} - \vec{u}^n}{\Delta t} = \frac{1}{2} [\vec{f}(\vec{u}^{n+1}) - \vec{f}(\vec{u}^n)]$$

$$\vec{u}^{n+1} - \vec{u}^n = \frac{\Delta t}{2} \left[\vec{f}(\vec{u}^n) + A_n (\vec{u}^{n+1} - \vec{u}^n) \right]$$

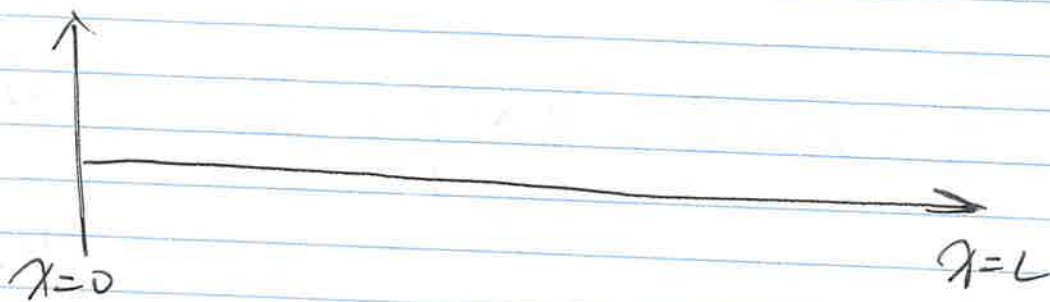
A_n change in time

At each time step $\rightarrow$ solve linear system:
$$+ \vec{f}(\vec{u}^n)]$$

$$\left(I - \frac{\Delta t}{2} A_n \right) \vec{u}^{n+1} = \left(I - \frac{\Delta t}{2} A_n \right) \vec{u}^n + \Delta t \vec{f}(\vec{u}^n)$$

solve @ every step $\leftarrow$ linear system of algebraic eqn.

$$y(0) = y_0, \text{ \& \> } y(L) = y_L$$



Convert the 2nd order ODE $\rightarrow$ 2 1st order ODEs,

$$u = y$$

(or in general,

$$v = y'$$

or system of 1st order ODE)

$$\begin{cases} u' = v \\ v' = -f(y) \end{cases} \quad \begin{matrix} u(0) = y_0 \\ u(L) = y_L \end{matrix}$$

guess $v(0) = v_0$

Solve from $x=0$ to $x=L$ & check to see
if $u(L) = y_L$ if not guess again ...

$$y_L = C_1 y_1(L) + C_2 y_2(L)$$

2 eqns, for C_1 & C_2

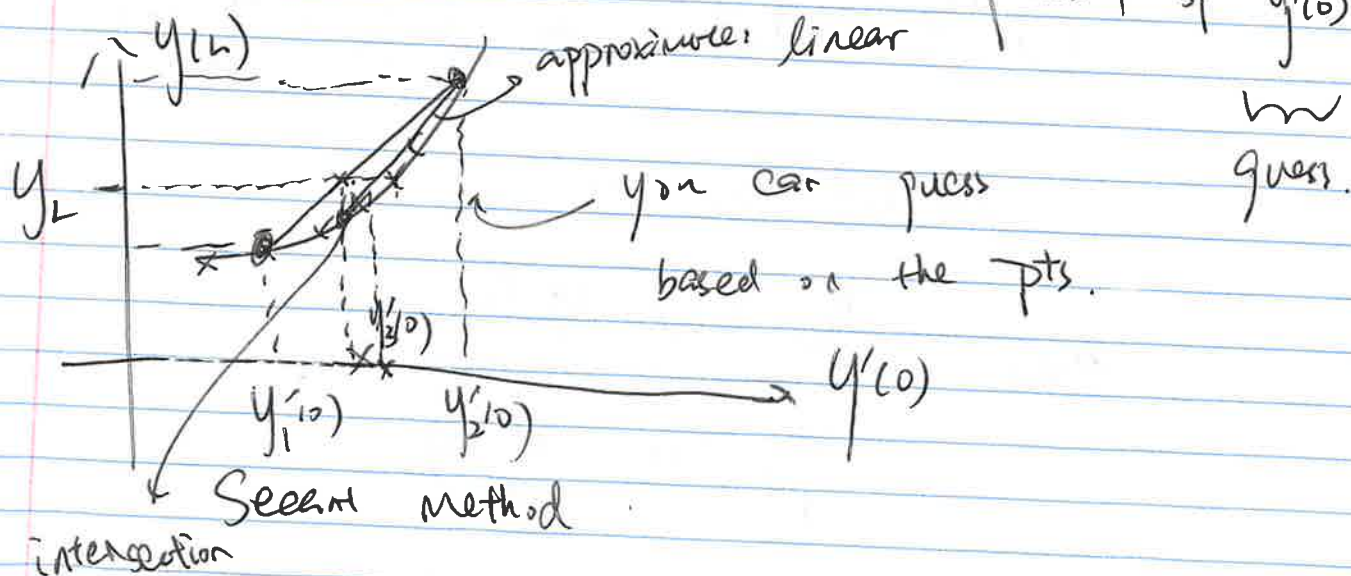
$$\left\{ \begin{aligned} C_1 &= \frac{y_L - y_2(L)}{y_1(L) - y_2(L)} \end{aligned} \right.$$

$$C_2 = 1 - C_1$$

Non-linear eqns. $\rightarrow$ have to iterate

e.g. secant method

Consider $y(L)$ as a nonlinear function of $y'(0)$



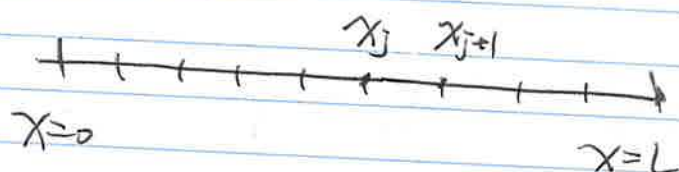
Direct Method

Approximate the derivatives in the diff. eqn. with finite differences, and solve the system of algebraic eqns.

Example $y''(x) + A(x)y'(x) + B(x)y(x) = f(x)$

$$y(0) = y_0$$

$$y(L) = y_L$$



$$h = x_{j+1} - x_j$$

Second order finite differences.

$$\frac{y_{j+1} - 2y_j + y_{j-1}}{h^2} + A(x_j) \frac{y_{j+1} - y_{j-1}}{2h}$$

$$+ B(x_j)y_j = f_j$$

" $N+1$ pts including the 2 boundaries"

$$j = 1, 2, 3, \dots, N-1$$

$$\begin{pmatrix}
 \beta_1 & \alpha_1 & & & 0 \\
 \gamma_2 & \beta_2 & \alpha_2 & & \\
 0 & \gamma_3 & \beta_3 & \alpha_3 & \\
 & & & & \\
 & 0 & \dots & & \\
 & & & \gamma_{N-1} & \beta_{N-1}
 \end{pmatrix}
 \begin{pmatrix}
 y_1 \\
 y_2 \\
 y_3 \\
 \vdots \\
 y_{N-1}
 \end{pmatrix}
 =
 \begin{pmatrix}
 f_1 - \alpha_1 f_0 \\
 f_2 \\
 f_3 \\
 \vdots \\
 f_{N-1} - \alpha_{N-1} y_1
 \end{pmatrix}$$

... $3N$ operations for the tridiagonal matrix.

- Mixed boundary conditions.

$$a y_0 + b y'_0 = g$$

1) use one side of finite-diff for y'_0 .

$$y'_0 = \frac{-3y_0 + 4y_1 - y_2}{2h}$$

$$a y_0 + b \frac{-3y_0 + 4y_1 - y_2}{2h} = g \rightarrow \text{solve for } y_0 \text{ in terms of } y_1 \text{ \& } y_2$$

P.D.E. $\rightarrow$ semi-discretization

finite difference spatial derivatives



System of O.D.E.s.

Example

diffusion eqn. (heat eqn.)

$$\frac{\partial \phi}{\partial t} = \alpha \frac{\partial^2 \phi}{\partial x^2}$$

$$\phi(0, t) = \phi(L, t) = 0$$

$$\phi(x, 0) = g(x)$$

F.V in space



$$\frac{d\phi_i}{dt} \leftarrow \frac{\partial \phi_i}{\partial t} = \alpha \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2}, \quad i=1, 2, 3, \dots, N-1$$

System of O.D.E.s

$$\frac{d\vec{\phi}}{dt} = A\vec{\phi}$$

A is $(N-1) \times (N-1)$ matrix



... stiffness !

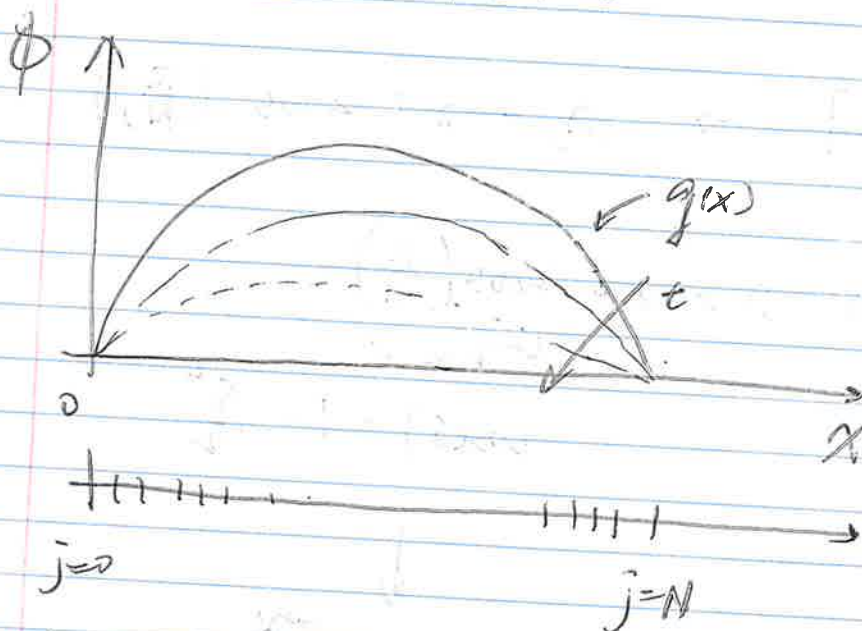
Week 8

Tue

PDEs. $\frac{\partial \phi}{\partial t} = \alpha \frac{\partial^2 \phi}{\partial x^2}$

$$\phi(x, 0) = g(x).$$

$$\phi(0, t) = \phi(L, t) = 0.$$



$$\frac{\partial \phi_j}{\partial t} = \alpha \frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta x^2}$$

$$j = 1, 2, \dots, N-1.$$

$$\frac{d\vec{\phi}}{dt} = A\vec{\phi}, \quad \vec{\phi} = \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \end{Bmatrix}, \quad A = \frac{\alpha}{\Delta x^2} \begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & & \\ & & \ddots & \ddots & \\ 0 & & & 1 & -2 \end{bmatrix}$$

$B[1, -2, 1]$

Stiffness $\frac{|\lambda|_{\max}}{|\lambda|_{\min}} = \frac{4\alpha/\Delta x^2}{\alpha\pi^2/\Delta x^2 N^2} = \frac{4N^2}{\pi^2}$

Stiffness increase
with grid resolution.

For explicit Euler, $\lambda \in \mathbb{R}$ negative

$$\Delta t \leq \frac{2}{|\lambda|_{\max}} = \frac{2}{4\alpha/\Delta x^2} \rightarrow \Delta t \leq \frac{\Delta x^2}{2\alpha}$$

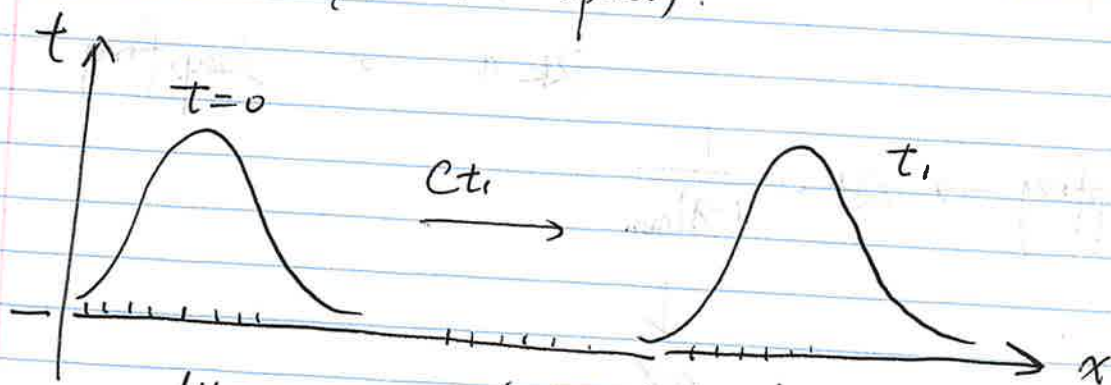
Advection eqn.

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$$



$$c > 0$$

(advection speed)



$$\frac{du_j}{dt} = -c \left(\frac{u_{j+1} - u_{j-1}}{2\Delta x} \right)$$

For advection problems, stability bounds

$$\frac{C \Delta t}{\Delta x} \leq \text{const.}$$

→ function of time-stepping scheme.

↪ CFL number

(Courant - Friedrich - Lewy)

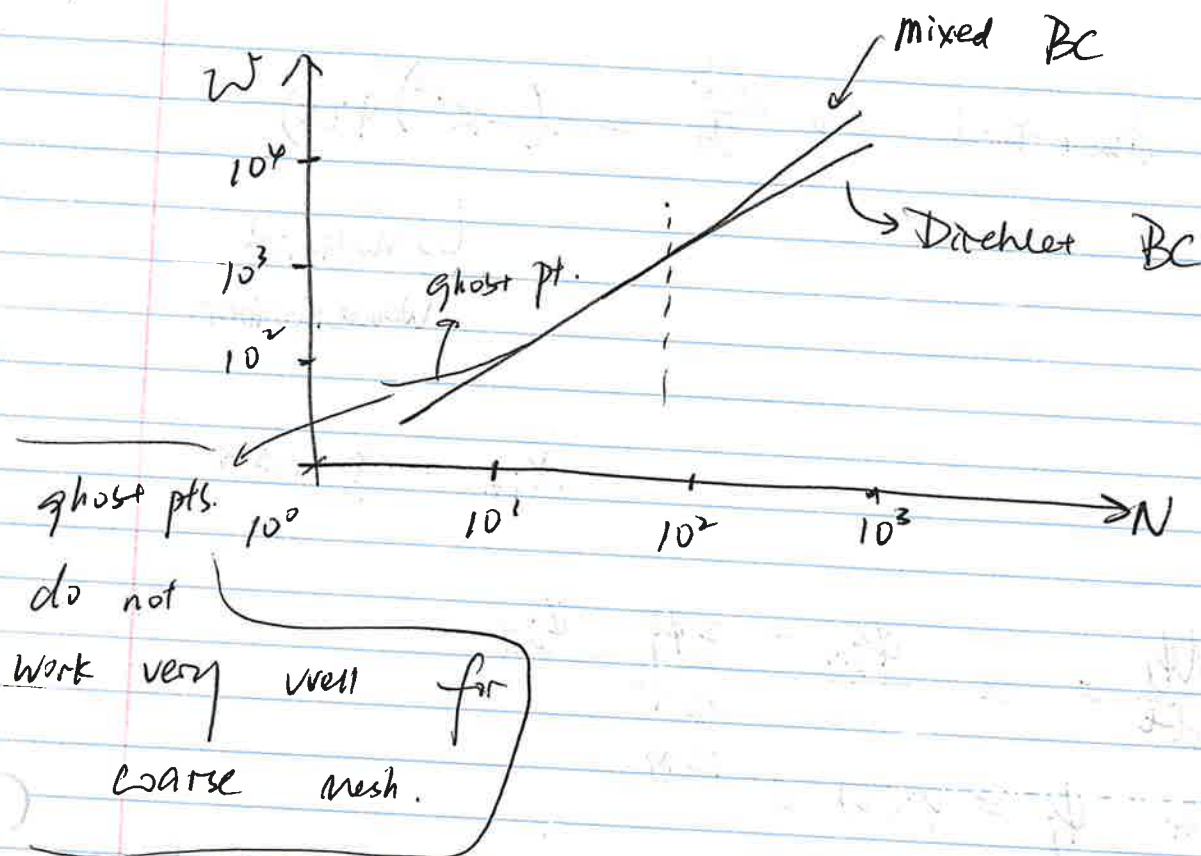
For a very slight modification to the system (e.g., B.C.s) the analysis (stability analysis) changes.

Effect of boundary conditions, on stability.

$$\frac{\partial \phi}{\partial t} = \frac{\partial^2 \phi}{\partial x^2}, \quad \phi(x, 0) = g(x)$$

$$\phi(0, t) = 0,$$

$$\alpha \phi(L, t) + \frac{\partial \phi}{\partial x}(L, t) = 0$$



Modified wavenumber analysis (Assume P.B.C.s)

$$\frac{\partial \phi}{\partial t} = \alpha \frac{\partial^2 \phi}{\partial x^2} \quad \text{let } \phi(x, t) = \psi(t) e^{ikx}.$$

plug in P.D.E.

$$\frac{d\psi}{dt} e^{ikx} = \alpha (-k^2) \psi(t) e^{ikx}$$

(using the ansatz)

$$\rightarrow \frac{d\psi}{dt} = \alpha (-k^2) \psi(t).$$

System decomposed !!!

$$\text{CFL } k' \Delta x = \pi \rightarrow (k' \Delta x)_{\max}^2 = 4$$

$$k'^2_{\max} = \frac{4}{\Delta x^2}$$

$$\text{Explicit Euler } \Delta t \leq \frac{2}{|\lambda|_{\max}} = \frac{2}{\alpha |k'|_{\max}} = \frac{2}{\alpha \left(\frac{4}{\Delta x} \right)}$$

$$\rightarrow \Delta t \leq \frac{\Delta x^2}{2\alpha}$$

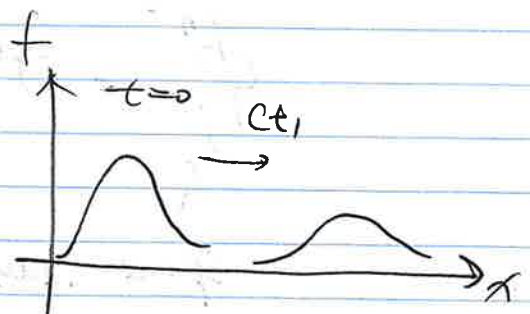
$$\text{RK4. } \Delta t \leq \frac{2.88}{\Delta t |\lambda|_{\max}} = \frac{2}{\alpha |k'|^2_{\max}} = \frac{2.77}{6} \frac{\Delta x^2}{\alpha}$$

4th-order Padé

$$(k \Delta x)^2 = \frac{12(1 - \cos(k \Delta x))}{5 + \cos(k \Delta x)} = 6 \rightarrow k^2_{\max} = \frac{6}{\Delta x^2}$$

Advection-Diffusion

$$\frac{\partial \phi}{\partial t} = -c \frac{\partial \phi}{\partial x} + \alpha \frac{\partial^2 \phi}{\partial x^2}$$



$$\frac{d\psi}{dt} = (-ikx - \alpha k^2) \psi$$

Central
1st order

pick your
scheme
(eg Padé)

Modified wave number eqn. $\frac{d\psi}{dt} = (-ik_1' - \alpha k_2'^2) \psi$

Week 8.

Thurs

P.D.E.

$$\frac{\partial \psi}{\partial t} = \alpha \frac{\partial^2 \psi}{\partial x^2}$$

$$\frac{d\psi_i}{dt} = \alpha \frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{\Delta x^2}$$

$$i = 1, 2, 3, \dots, N-1.$$



$$\vec{\psi} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{N-1} \end{bmatrix}$$

semi-discretized.

$$\frac{d\vec{\psi}}{dt} = A\vec{\psi} \quad \text{system of ODEs.}$$

Euler time adv.

$$\vec{\psi}^{n+1} = \vec{\psi}^n + \Delta t A \vec{\psi}^n$$

Similar to $y' = \lambda y$

$$\lambda = -\alpha k'^2$$

For 2nd order central (for 2nd derivative).

$$k'^2 = \frac{2}{\Delta x^2} [1 - \cos k \Delta x]$$

For time adv. Euler.

$$\Delta t \leq \frac{2}{\frac{2\alpha}{\Delta x^2} [1 - \cos k \Delta x]}$$

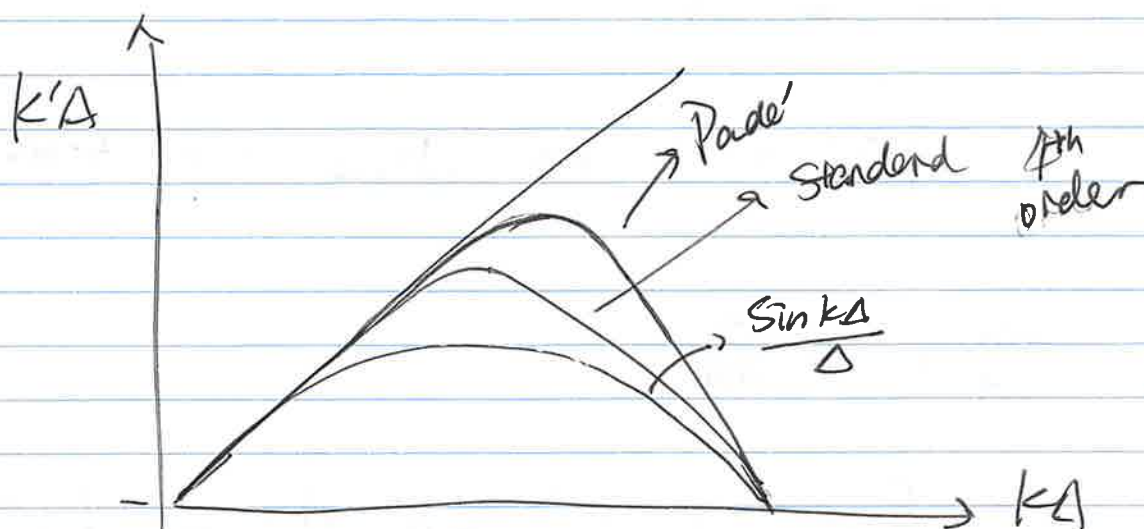
Worst case scenario. $\cos k \Delta x = -1$

$$\Delta t \leq \frac{\Delta x^2}{2\alpha}$$

RK-4: $\Delta t \leq \frac{2.79 \Delta x^2}{4\alpha}$

Problem: $\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$ (Convection)

For Padé's scheme $\rightarrow$ substitute new k'



Von Neumann Stability analysis

$$\frac{\partial \phi}{\partial t} = \alpha \frac{\partial^2 \phi}{\partial x^2}$$

Euler

$$\phi_j^{n+1} = \phi_j^n + \alpha \frac{\Delta t}{\Delta x^2} \left[\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n \right]$$

Assume $\phi_j^n = \sigma^n e^{ikx_j}$

& seek $|\sigma| \leq 1$

Heat eqn. $\frac{\partial \psi}{\partial t} = \alpha \frac{\partial^2 \psi}{\partial x^2}$

$$\frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} = \frac{\alpha}{2} \left[\frac{\partial^2 \psi^{n+1}}{\partial x^2} + \frac{\partial^2 \psi^n}{\partial x^2} \right]$$

$$\psi_j^{n+1} - \psi_j^n = \frac{\alpha \Delta t}{2} \left[\frac{\psi_{j+1}^{n+1} - 2\psi_j^{n+1} + \psi_{j-1}^{n+1}}{\Delta x^2} + \frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{\Delta x^2} \right]$$

$$\frac{\alpha \Delta t}{2 \Delta x^2} = \beta$$

$$-\beta \psi_{j+1}^{n+1} + (1+2\beta) \psi_j^{n+1} - \beta \psi_{j-1}^{n+1} = \beta \psi_{j+1}^n$$

$$j = 1, 2, 3, \dots, N-1 \quad + (1-2\beta) \psi_j^n + \beta \psi_{j-1}^n$$

Modified eqn. analysis.

Accuracy

$$\frac{\partial \tilde{\varphi}}{\partial t} = \alpha \frac{\partial^2 \tilde{\varphi}}{\partial x^2} \quad \tilde{\varphi} \text{ is the exact sol'n.}$$

φ_j^n is the numerical sol'n.

(evaluated at spatial & temporal grid)

φ (e.g., polynomial) continuous function for

the numerical sol'n φ_j^n .

$\varphi(x, t)$ is analytical function

↳ φ_j^n of the discrete pts.

What eqn. does φ satisfy.

Define

↳ $[\varphi]$ difference operator

If ϕ is sol'n to discrete eqn.

$$\cancel{L[\phi]}$$

the modified eqn.

$$\dots - \left(\frac{\partial \phi}{\partial t} - \alpha \frac{\partial^2 \phi}{\partial x^2} \right) = - \alpha \frac{\Delta x^2}{12} \frac{\partial^2 \phi}{\partial x^4} \dots$$

which the numerical sol'n satisfies. is

$\tilde{\phi}$ exact sol'n.

$$L[\tilde{\phi}] \neq 0 = \underbrace{\frac{\alpha \Delta x^2}{12} \frac{\partial^2 \phi}{\partial x^4} - \frac{\Delta t}{2} \frac{\partial^2 \phi}{\partial x^2}}_{\text{error}}$$

truncation error
in the continuous case

Showed:

$$L[\psi]$$

$$\frac{\partial \psi}{\partial t} - \alpha \frac{\partial^2 \psi}{\partial x^2} = \alpha \cdot \frac{\Delta x^2}{12} \frac{\partial^4 \psi}{\partial x^4} - \frac{\Delta t}{2} \frac{\partial^2 \psi}{\partial t^2} + \dots$$

modified eqn.

Continuous eqn. that the discrete (numerical)
soln satisfies.

2nd order in space & 1st order in time

$$L[\tilde{\psi}] \neq 0. \quad \tilde{\psi} \neq \psi; @ \text{ grid pts.}$$

$$L[\psi] = 0 \quad \text{if } \psi = \psi_j @ x_j$$

$$L[\tilde{\psi}] = \underbrace{\frac{\partial \tilde{\psi}}{\partial t} - \frac{\partial^2 \tilde{\psi}}{\partial x^2}}_{=0} - \alpha \frac{\Delta x^2}{12} \frac{\partial^4 \tilde{\psi}}{\partial x^4} + \frac{\Delta t}{2} \frac{\partial^2 \tilde{\psi}}{\partial t^2} + \dots$$

!!!
0

(Exact Soln)

$$L[\tilde{\psi}] = -\alpha \frac{\Delta x^2}{12} \frac{\partial^4 \tilde{\psi}}{\partial x^4} + \frac{\Delta t}{2} \frac{\partial^2 \tilde{\psi}}{\partial t^2} + \dots$$

Exam $y' = \lambda y$

λ is real & negative.

→ looping will be unstable.

P.D.Es. ... modified wavenumber analysis.

$$\Rightarrow \frac{d\phi}{dt} = -k^2 \phi$$

Ansatz: $\phi = \hat{\phi}(t) e^{ikx}$

$$\frac{\phi_j^{n+1} - \phi_j^n}{2\Delta t} = \alpha \frac{\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n}{\Delta x^2} + \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2)$$

... unstable

D.F. $\phi_j^n = \frac{\phi_j^{n+1} + \phi_j^n}{2} + \mathcal{O}(\Delta t^2)$

Resulting eqn. → unconditionally stable

Rearrange: $\beta = \alpha \frac{\Delta t}{\Delta x^2}$

$$(1+2\beta) \phi_j^{n+1} = (1-2\beta) \phi_j^n + 2\beta \phi_{j+1}^n + 2\beta \phi_{j-1}^n$$

Stability analysis for discrete form: von Neumann

$$\frac{u^{n+1} - u^n}{\Delta t} = \underbrace{u^n \frac{\partial u^n}{\partial x}}_{\text{Euler}} + \underbrace{v \frac{\partial^2 u^{n+1}}{\partial x^2}}_{\text{backward Euler}}$$

Euler

backward Euler

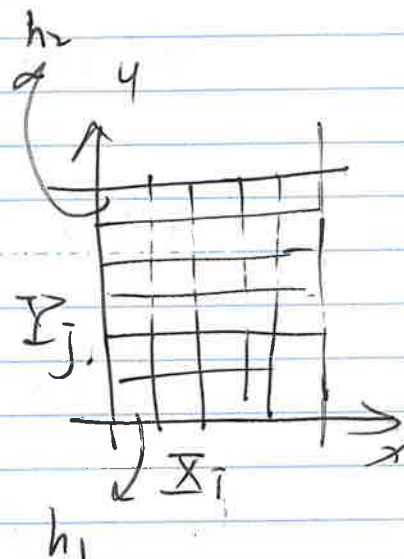
$$\dots \Delta t \leq \frac{\Delta x^2}{2v}$$

... higher dimensions.

$$\frac{\partial \phi}{\partial t} = \alpha \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right)$$

Use M internal pts. (there is $\bar{x}_j$

not on boundaries) &



N internal pts in y .

Euler in time & 2nd order F.D. in space

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} = \alpha \left[\frac{\phi_{i+1,j}^n - 2\phi_{ij}^n + \phi_{i-1,j}^n}{h_1^2} \right.$$

$$\left. + \frac{\phi_{i,j+1}^n - 2\phi_{ij}^n + \phi_{i,j-1}^n}{h_2^2} \right]$$

$$\Delta t < \frac{1}{2\alpha(k_1^2 + k_2^2)}$$

$$\Delta t < \frac{2}{\alpha \left[\frac{2 - 2\cos k_1 h_1}{h_1^2} + \frac{2 - 2\cos k_2 h_2}{h_2^2} \right]}$$

Worst Case Scenario: $\cos k_1 h_1$ & $\cos k_2 h_2 = -1$.

gives you: $\Delta t \leq \frac{1}{2\alpha \left(\frac{1}{h_1^2} + \frac{1}{h_2^2} \right)}$

If $h_1 = h_2 = h$.

$$\Delta t < \frac{h^2}{4\alpha}$$

In 3D: $\Delta t < \frac{h^2}{6\alpha}$.

Implicit:

- Trapezoidal.

- Crank-Nicholson.

$$\frac{\varphi^{n+1} - \varphi^n}{\Delta t} = \frac{\alpha}{2} \left[\frac{\partial^2 \varphi^{n+1}}{\partial x^2} + \frac{\partial^2 \varphi^{n+1}}{\partial y^2} + \frac{\partial^2 \varphi^n}{\partial x^2} + \frac{\partial^2 \varphi^n}{\partial y^2} \right]$$

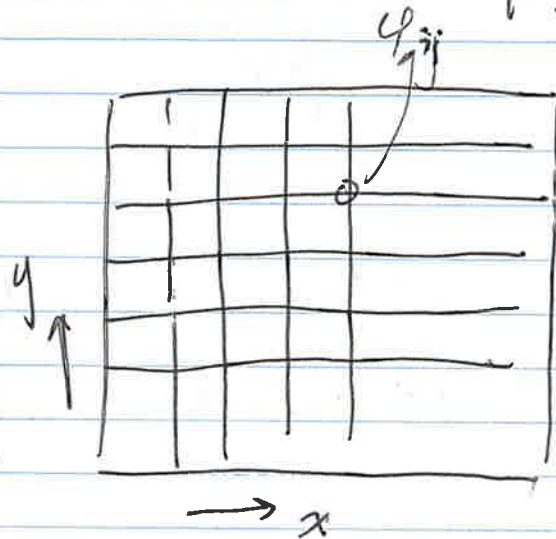
Week 9

Thurs.

Multi-dimensions

$$\frac{\partial \varphi}{\partial t} = \alpha \left(\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} \right)$$

Implicit. $\rightarrow$ have to
solve a system
of algebraic equations
at each step.



in 2D & 3D. $\rightarrow$ large matrix (sparse).

Crank - Nicholson (Trapezoidal).

$$\frac{\varphi^{n+1} - \varphi^n}{\Delta t} = \frac{\alpha}{2} \left[\frac{\varphi_{i+1,j}^{n+1} - 2\varphi_{i,j}^{n+1} + \varphi_{i-1,j}^{n+1}}{h_1^2} + \frac{\varphi_{i,j+1}^{n+1} - 2\varphi_{i,j}^{n+1} + \varphi_{i,j-1}^{n+1}}{h_2^2} \right]$$

Block-tri-diagonal.

$$\begin{array}{c}
 M \times M \\
 \text{matrices}
 \end{array}
 \begin{bmatrix}
 B & C & & & \\
 A & B & C & & \\
 & 0 & A & B & C \\
 & & & \dots & \\
 & & & & A & B & C \\
 & & & & & A & B
 \end{bmatrix}
 \begin{bmatrix}
 \varphi_{11} \\
 \varphi_{21} \\
 \varphi_{31} \\
 \vdots \\
 \varphi_{12} \\
 \varphi_{22} \\
 \vdots
 \end{bmatrix}^{n+1} = \begin{bmatrix}
 \vdots \\
 \vdots \\
 \vdots \\
 F_{ij}^{(n)} \\
 \vdots \\
 \vdots
 \end{bmatrix}$$

N blocks on the diagonal.

... A & C can be the same

(heat eqn.)

of arithmetic operations. $\mathcal{O}(M^2 N)$

in symbolic notation:

$$\frac{\varphi^{n+1} - \varphi^n}{\Delta t} = \frac{1}{2} \left[\frac{\partial^2 \varphi^{n+1}}{\partial x^2} + \frac{\partial^2 \varphi^{n+1}}{\partial y^2} + \frac{\partial^2 \varphi^n}{\partial x^2} + \frac{\partial^2 \varphi^n}{\partial y^2} \right]$$

$$\begin{aligned}
 & \left[I - \frac{\alpha \Delta t}{2} A_x - \alpha \frac{\Delta t}{2} A_y \right] \vec{\varphi}^{n+1} \\
 &= \left[I + \alpha \frac{\Delta t}{2} A_x + \alpha \frac{\Delta t}{2} A_y \right] \vec{\varphi}^n \\
 & \quad + \mathcal{O}(\Delta t) \left[\mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2) \right]
 \end{aligned}$$

Approx. factorization.

$$\begin{aligned}
 & \left(I - \frac{\alpha \Delta t}{2} A_x \right) \left(I - \frac{\alpha \Delta t}{2} A_y \right) \vec{\varphi}^{n+1} - \frac{\alpha^2 \Delta t^2}{2} A_x A_y \vec{\varphi}^{n+1} \\
 &= \left(I + \alpha \frac{\Delta t}{2} A_x \right) \left(I + \alpha \frac{\Delta t}{2} A_y \right) \vec{\varphi}^n - \frac{\alpha^2 \Delta t^2}{2} A_x A_y \vec{\varphi}^n \\
 & \quad + \mathcal{O}(\Delta t) \left[\mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2) \right]
 \end{aligned}$$

$$\begin{aligned}
 & \left(I - \frac{\alpha \Delta t}{2} A_x \right) \left(I - \frac{\alpha \Delta t}{2} A_y \right) \vec{\varphi}^{n+1} = \\
 & \quad \left(I + \frac{\alpha \Delta t}{2} A_x \right) \left(I + \frac{\alpha \Delta t}{2} A_y \right) \vec{\varphi}^n \\
 & \quad + \frac{\alpha^2 \Delta t^2}{2} A_x A_y (\vec{\varphi}^{n+1} - \vec{\varphi}^n) + \mathcal{O}(\Delta t) \\
 & \quad + \mathcal{O}(\Delta t) \left[\mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2) \right] \underbrace{\hspace{10em}}_{1^{st} \text{ order T.S.}}
 \end{aligned}$$

To get ϕ_{ij} , use defn of Σ

$$\left(I - \frac{\alpha \Delta t}{2} A_y\right) \vec{\phi}^{n+1} = \vec{\Sigma}$$

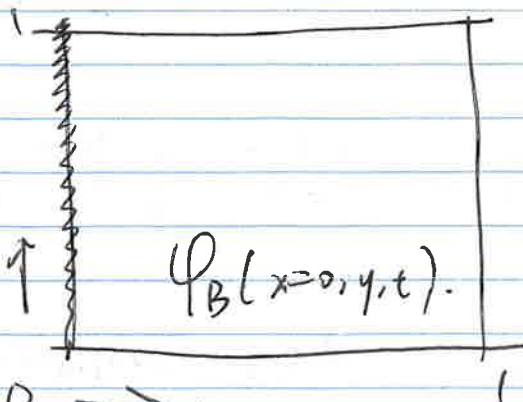
Solve the tri-diagonal for $\vec{\phi}$.

$$\phi_{ij} - \frac{\alpha \Delta t}{2} \frac{\phi_{i,j+1} - 2\phi_{ij} + \phi_{i,j-1}}{\Delta y^2} = \Sigma_{ij}$$

For each i , solve the tri-diagonal
the boundary conditions for Σ in terms of
B.C.s of ϕ ...?

$\rightarrow \Sigma_{0,j}$

Recall defn of Σ .



$$\Sigma_{ij} = \phi_{ij}^{n+1} - \frac{\alpha \Delta t}{2} \frac{\phi_{i,j+1}^{n+1} - 2\phi_{ij}^{n+1} + \phi_{i,j-1}^{n+1}}{\Delta y^2}$$

Alternating Direction Implicit Methods (A.D.I.)

$$\frac{\partial \phi}{\partial t} = \alpha \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right)$$

- 1) advance 1 step with implicit Euler acting on $\frac{\partial^2 \phi}{\partial x^2}$ & explicit Euler on $\frac{\partial^2 \phi}{\partial y^2}$

$$\frac{\phi^{n+1/2} - \phi^n}{\Delta t/2} = \alpha \left(\frac{\partial^2 \phi^{n+1/2}}{\partial x^2} + \frac{\partial^2 \phi^n}{\partial y^2} \right)$$

- 2) Advance $1/2$ Step with the roles of explicit & implicit scheme reversed.

$$\frac{\phi^{n+1} - \phi^{n+1/2}}{\Delta t/2} = \alpha \left(\frac{\partial^2 \phi^{n+1/2}}{\partial x^2} + \frac{\partial^2 \phi^{n+1}}{\partial y^2} \right)$$

Each step involves simple tri-diagonal (1D soln)

... Same as approx. factorization.

Elliptic P.D.E.

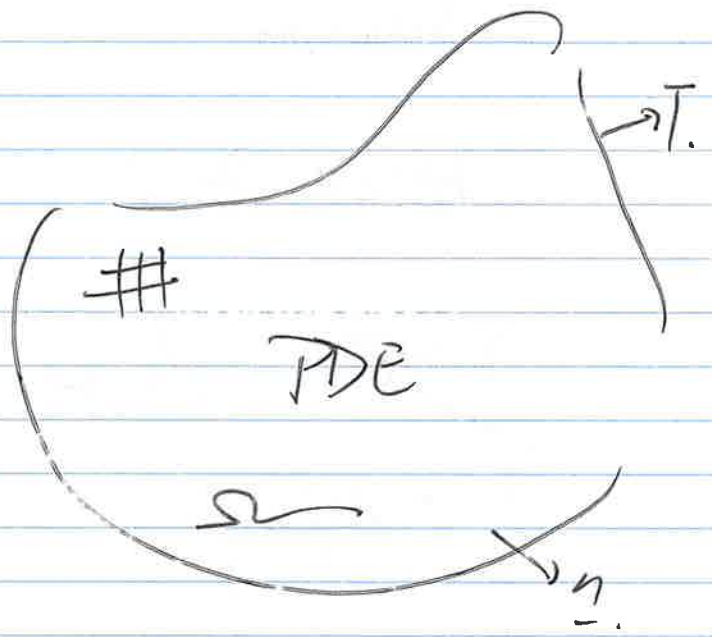
Equilibrium (Steady-state).

Soln at all pts in

Space are coupled.

Information travels.

at ∞ speed.



Laplace eqn.

$$\nabla^2 \phi = 0. \quad \Leftrightarrow \quad \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0.$$

Helmholtz eqn.

$$\nabla^2 \phi + \alpha^2 \phi = 0.$$

Poisson eqn. $\nabla^2 \phi = -f(x, y)$

Iterative Sol'n Method $\rightarrow$ Iterative methods.

$$A\vec{x} = \vec{b} \quad (1)$$

$$A = A_1 - A_2$$

Iterative Scheme

$$A_1 x^{(k+1)} = A_2 x^{(k)} + b \quad (2)$$

$\nearrow$ iteration index.

Starting with initial guess $x^{(1)}$

$k = 0, 1, 2, 3, \dots$ until convergence

Requirements,

(1) A should be easily iterable.

(2) Converges $\lim_{k \rightarrow \infty} x^{(k)} = x$

$$\text{Error } \epsilon^{(k)} = x - x^{(k)}$$

Subtract (2) from (1)

$$\Rightarrow D^{-1} = A^2 \begin{bmatrix} -1/4 & & & \\ & -1/4 & & \\ & & \ddots & \\ & & & 0 \\ & 0 & & & \ddots \\ & & & & & -1/e \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & \\ & & & & \ddots \end{bmatrix}$$

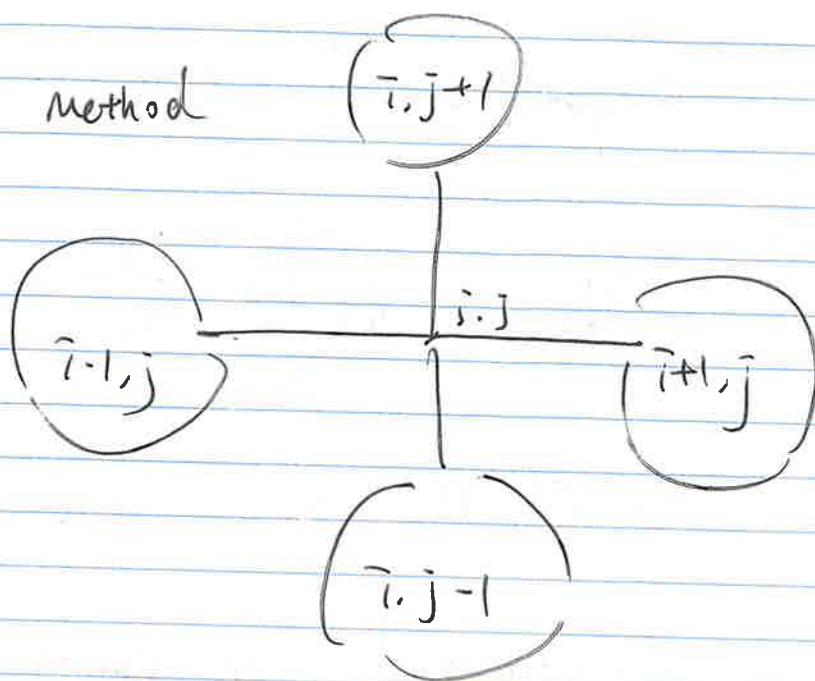
Eigenvalues of $A_1^T A_2$

$$\lambda_{m,n} = \frac{1}{2} \left[\cos \frac{m\pi}{M} + \cos \frac{n\pi}{N} \right] \quad \begin{matrix} m=1,2,3 \dots M-1 \\ n=1,2,3 \dots N-1 \end{matrix}$$

$$\Rightarrow |\lambda_{m,n}|_{\max} < 1$$

$\Rightarrow$ Point Jacobi converges

... Gauss-Seidel method

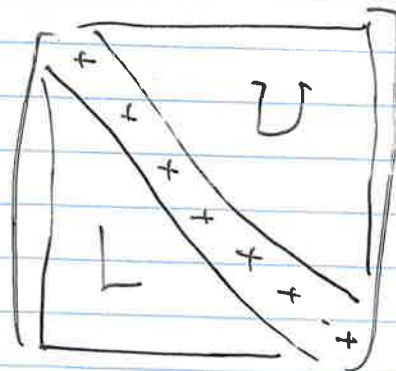


$$\phi_{ij}^{(k+1)} = \frac{1}{4} \left[\phi_{i-1,j}^{(k)} + \phi_{i+1,j}^{(k+1)} + \phi_{i,j-1}^{(k)} \right.$$

$$\left. + \phi_{i,j+1}^{(k+1)} \right] - \frac{\Delta^2}{4} f_{ij}.$$

In matrix notation

$$A = \underbrace{D}_{A_1} - \underbrace{L+U}_{A_2}$$



$$\omega = 1.7 \sim 1.9$$

↑ fantastic improvement
on convergence

Example

$$M = N = 30.$$

	$\lambda_{\max}$	itr. #. required (factor of 1000)
Jacobi.	.9945	1270
G-S.	.989	633
SOR		46

objective: minimize $\lambda_{\max}$

