EXTRA: $\boxed{\text{Ch 9,10}}$ "Special Topics"

A Brief overview of numerical approximation of differential equations:

- scalar 1st order ODE $\frac{dy}{dt} = f(x,y)$ IVP
- BVP: 2nd order ODE
  $y'' = f(x,y,y')$; $y(a) = \alpha$, $y(b) = \beta$
- PDE: 2nd order with IC & BC.
  eg. $\frac{\partial y}{\partial t} = \alpha \frac{\partial^2 y}{\partial x^2}$, $y(x,0) = y_0(x)$
  $y(a,t) = y_a(t)$
  $y(b,t) = y_b(t)$

Key concepts: $\left\{\begin{array}{l}\text{Convergence,}\\ \text{Consistency,}\\ \text{Stability.}\end{array}\right.$ ALL NECESSARY!

Part II: Numerical Solution of IVP's of ODE's

- Euler's methods
- Runge-Kutta methods
- Multi-step methods
- Error analysis (consistency, convergence, stability)

Initial Value Problems (IVP's) of ODE

- 1st-order (scalar) ODE:

$$\begin{cases} y' = f(x,y), & x \in [a,b] \\ y(a) = y_0 \in \mathbb{R} & \text{(initial value)} \end{cases}$$

Question: Under what conditions on f does this problem have a unique solution?

Existence and uniqueness theorem: Assume that
(i) $f(x,y)$ is continuous for all $x \in [a,b]$ and $y \in \mathbb{R}$;
(ii) there is a constant L (independent of x, y) such that

$$|f(x,y_1) - f(x,y_2)| \le L|y_1 - y_2| \quad \text{(Lipschitz condition)}$$

for all $x \in [a,b]$ & $y_1, y_2 \in \mathbb{R}$ ↳ Lipschitz constant.

Remark: $f(x,y)$ satisfies the Lipschitz condition if $\frac{\partial f}{\partial y}$ is bounded. In fact,

$$|f(x,y_1) - f(x,y_2)| = \Big| \underbrace{\frac{\partial f}{\partial y}(x,\xi)}_{\text{bounded by } L} (y_1 - y_2) \Big| \le L |y_1 - y_2|$$

Ex. The IVP $\begin{cases} y' = x + \frac{x^2}{1+x^2} \sin y \\ y(0) = 0 \end{cases}$ has a unique sol^n.

Ex. The IVP $\begin{cases} y' = \sqrt{y} \\ y(0) = 0 \end{cases}$ has at least 2 sol^{n}s: $y = 0$, $y = \frac{1}{4}x^2$.

This is because the function $\sqrt{y}$ does not satisfy the Lipschitz condition

$$\left| \frac{\sqrt{y_1} - \sqrt{y_2}}{y_1 - y_2} \right| = \frac{1}{\sqrt{y_1} + \sqrt{y_2}} \quad \text{not bounded as } y_1, y_2 \to 0.$$

- System of 1st-order ODE's: N is called the dimension of the system

$$\begin{cases} y_1' = f_1(x, y_1, \ldots, y_N) \\ y_2' = f_2(x, y_1, \ldots, y_N) \\ \quad \vdots \\ y_N' = f_N(x, y_1, \ldots, y_N) \end{cases} \iff \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}'}_{\substack{y \\ \text{(a vector)}}} = \underbrace{\begin{bmatrix} f_1(x, y_1, \ldots, y_N) \\ f_2(x, y_1, \ldots, y_N) \\ \vdots \\ f_N(x, y_1, \ldots, y_N) \end{bmatrix}}_{f(x,y) \text{ (a vector function)}}$$

$$\iff \boxed{y' = f(x,y)}$$

(same form as 1st-order scalar ODE)

- Higher order scalar ODE's

$$\frac{d^N y}{dx^N} = f(x, y, y', \ldots, y^{(N-1)})$$

convert it to a system of 1st-order ODE's by change of variables:

$$\begin{cases} y_1 = y \\ y_2 = y' \\ y_3 = y'' \\ \quad \vdots \\ y_N = y^{(N-1)} \end{cases} \xrightarrow{\text{then}} \begin{cases} y_1' = y' = y_2 \\ y_2' = y'' = y_3 \\ y_3' = y'' = y_4 \\ \quad \vdots \\ y_N' = y^{(N)} = f(x, y_1, \ldots, y_N) \end{cases}$$

$$\Longleftrightarrow \begin{bmatrix} y_1' \\ y_2' \\ \vdots \\ y_N' \end{bmatrix} = \begin{bmatrix} y_2 \\ y_3 \\ \vdots \\ y_N \\ f(x, y_1, \dots, y_N) \end{bmatrix} \quad \text{or} \quad \boxed{y' = f(x,y)}$$

Remarks: ① The existence and uniqueness theorem also holds for vector ODE's. However in the Lipschitz condition the absolute sign $|\cdot|$ should be replaced by vector norms $\|\cdot\|$.

② IVP's with more than one solution are difficult to solve numerically. Thus, in what follows we always assume each IVP has a unique solution!

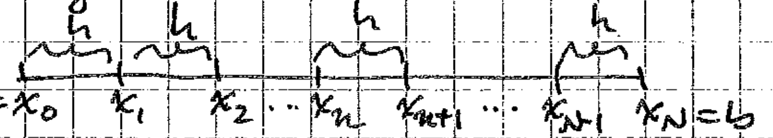
③ For simplicity, only scalar ODE's will be fully discussed. Numerical methods for vector ODE's are similar to those for scalar ODE's.

Euler's Method (§5.3 of text)

We try to solve $\begin{cases} y' = f(x,y) \\ y(a) = y_0 \end{cases}$, $x \in [a, b]$ numerically.

Notation: $h = \frac{b-a}{N} \longrightarrow$ step size.

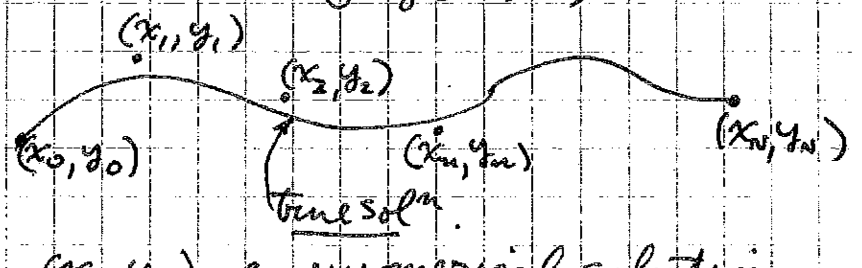
$x_n = a + nh$, $n = 0, 1, 2, \dots, N$,



$y(x) \longrightarrow$ exact (true) solution

y_n denotes the approximate value of $y(x_n)$, ie

$y_n \approx y(x_n)$



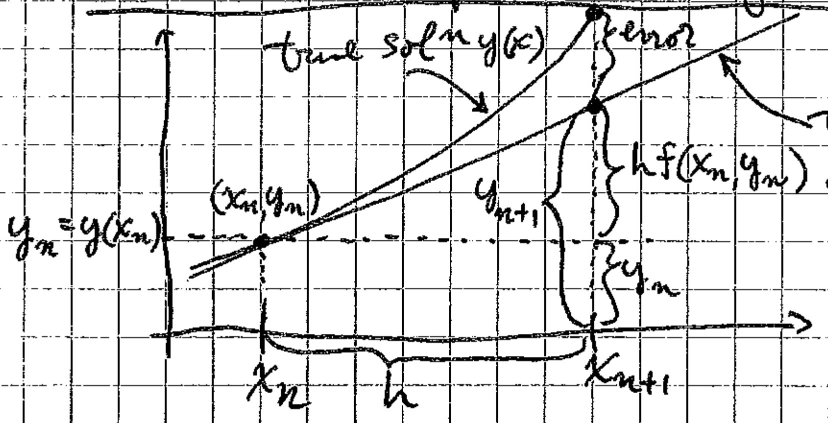
We call $(x_0, y_0), (x_1, y_1), \dots, (x_N, y_N)$ a numerical solution

We derive Euler's method as follows:

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + h\underbrace{y'(x_n)}_{f(x_n,\, y(x_n))} + \frac{h^2}{2!} y''(\xi)$$

$$\Longrightarrow \boxed{y_{n+1} = y_n + h f(x_n, y_n)}$$

Geometrical Interpretation of the Euler Method



the tangent line at (x_n, y_n) of $y(x)$ with slope $y'(x_n) = f(x_n, y(x_n)) = f(x_n, y_n)$

Its eqn. is:

$$y - y_n = f(x_n, y_n)(x - x_n)$$

$$\Rightarrow y_{n+1} - y_n = f(x_n, y_n)(x_{n+1} - x_n) = hf(x_n, y_n)$$

- Error analysis:

Def^n: ① The per-step (or local) truncation error is $\boxed{e_{n+1}(h) = y(x_{n+1}) - y_{n+1}}$ when y_n is exactly $y(x_n)$

② The accumulated (or global) truncation error is $\boxed{E_{n+1}(h) = y(x_{n+1}) - y_{n+1}}$ with no restriction on y_n.

We estimate $e_{n+1}(h)$ as follows:

$$\begin{aligned} e_{n+1}(h) &= y(x_{n+1}) - y(x_n) - hf(x_n, y(x_n)) \\ &= y(x_n + h) - y(x_n) - hf(x_n, y(x_n)) \\ &= \cancel{y(x_n)} + h\cancel{y'(x_n)} + \frac{h^2}{2} y''(\xi) \\ &\quad - \cancel{y(x_n)} - h\cancel{f(x_n, y(x_n))} \quad \text{(cancel)} \\ &= \frac{h^2}{2} y''(\xi) = O(h^2) \end{aligned}$$

$\Rightarrow$ the local truncation error is of order $\boxed{2}$

We estimate $E_{n+1}(h)$ as follows (assume no round-off errors)

$$\begin{cases} y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n)) + \underbrace{e_{n+1}(h)}_{\frac{h^2}{2} y''(\xi)} \\ y_{n+1} = y_n + hf(x_n, y_n) \end{cases}$$

$$\Rightarrow E_{n+1}(h) = y(x_{n+1}) - y_{n+1} = y(x_n) - y_n + h\left(f(x_n, y(x_n)) - f(x_n, y_n)\right) + e_{n+1}(h)$$

$$\Rightarrow |E_{n+1}(h)| \le |y(x_n)-y_n| + h\,|f(x_n,y(x_n))-f(x_n,y_n)| + |e_{n+1}(h)|$$

$$\le |E_n(h)| + Lh\,|\underbrace{y(x_n)-y_n}_{E_n(h)}| + \frac{M}{2}h^2$$

where $\boxed{M=\max_{\xi}|y''(\xi)|}$, $\boxed{L=\max_{\xi}\left|\frac{\partial f}{\partial y}(x_n,\xi)\right|}$

$$= \underbrace{(1+Lh)}_{\alpha}\,\underbrace{|E_n(h)|}_{z_n} + \underbrace{\frac{M}{2}h^2}_{\beta}$$

$$\Leftrightarrow\quad z_{n+1} \le \alpha z_n+\beta \le \alpha(\alpha z_{n-1}+\beta)+\beta = \alpha^2 z_{n-1}+\beta(1+\alpha)$$

$$\le \alpha^2(\alpha z_{n-2}+\beta)+\beta(1+\alpha) = \alpha^3 z_{n-2}+\beta(1+\alpha+\alpha^2)$$

$$\le \cdots \le \alpha^{n+1}z_0 + (\cancel{\alpha^n}\,1+\alpha+\cdots+\alpha^{n-1}+\alpha^n)\,\beta$$

$$= \alpha^{n+1}z_0 + \frac{1-\alpha^{n+1}}{1-\alpha}\,\beta$$

$$\Rightarrow |E_{n+1}(h)| \le (1+Lh)^{n+1}|E_0(h)| + \frac{1-(1+Lh)^{n+1}}{1-(1+Lh)}\cdot\frac{M}{2}h^2$$

$$\le e^{(n+1)hL}|E_0(h)| + \frac{M}{2}h\left(\frac{e^{(n+1)h}-1}{L}\right)$$

(using $(1+Lh)\le e^{hL}$)

$$\left(\begin{array}{l} x_{n+1}=a+(n+1)h \\ E_0(h)=y(0)-y_0 \end{array}\right) = \boxed{e^{L(x_{n+1}-a)}|E_0(h)| + \frac{M}{2L}h\,e^{L(x_{n+1}-a)}}$$

Def^n: A numerical solution $\{(x_n,y_n)\}$ is said to be convergent if $\lim_{\substack{h\to0\\ y_0\to y(0)}} y_n = y(x_n)$ at any fixed $x_n\in[a,b]$

Remark: ① Euler's method is convergent since $E_{n+1}(h)\to0$ as $E_0(h)\to0$ and $h\to0$.

② The local error of the Euler method $= O(h^2)$

" global " " " " " $O(h^1)$

$\Rightarrow$ it is generally true that the global error order is one less than the local error order.

Example: Consider $y' = \lambda y$, $y(0) = y_0$. The analytic solution is $\underline{y(x) = e^{\lambda x} y_0}$.

By Euler's method: $y_{n+1} = y_n + h f(x_n, y_n) \Rightarrow$

$$y_{n+1} = y_n + h(\lambda y_n) = (1+h\lambda) y_n$$
$$= (1+h\lambda)(1+h\lambda) y_{n-1} = (1+h\lambda)^2 y_{n-1}$$

$\Rightarrow \boxed{y_{n+1} = (1+h\lambda)^{n+1} y_0}$ (numerical solution)

Note that $(1+h\lambda)^{n+1} y_0 = (1+h\lambda)^{\frac{x_{n+1}-0}{h}} y_0 = \left[(1+\lambda h)^{\frac{1}{h}}\right]^{x_{n+1}} y_0$

$\rightarrow e^{\lambda x_{n+1}} y_0 = y(x_{n+1})$ as $h \to 0$.

$\Rightarrow \{(x_n, y_n)\}$ is convergent for all $\lambda \in \mathbb{R}$.

However, if $\lambda \ll 0$ (i.e. $\lambda < 0$ and $|\lambda|$ is large), e.g. $\lambda = -10$, $h = 0.5$ then:

The exact solution: $y(x) = e^{-10x} y_0 \to 0$ as $x \to \infty$

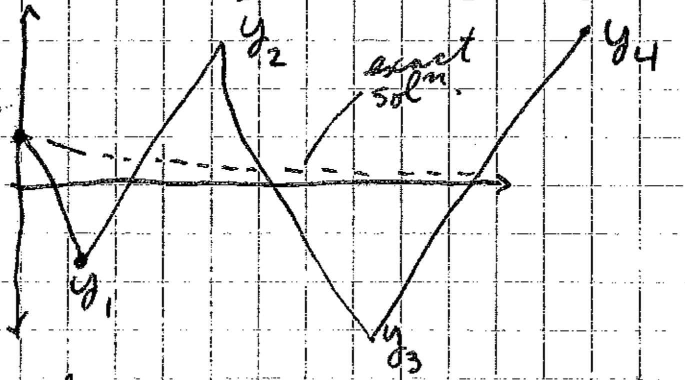
The numerical solution: $y_1 = (1+\lambda h) y_0 \to -4 y_0$

$y_2 = (1+\lambda h)^2 y_0 \to (-4)^2 y_0 = 4^2 y_0$

$\vdots$

$y_n = (-4)^n y_0$

$\rightarrow$ oscillates without bound.



Conclusion: Convergence is necessary for a numerical method to be good, but not sufficient. We need another concept called stability to determine if a method is useful or not.

Definition: A numerical method is said to be stable if small changes in y_0 lead to small changes in y_n for all $0 \le h \le h_0$.

Question: How to determine whether a numerical method is stable?

<u>A useful stability analysis method</u>: Suppose that a numerical method, when it is applied to the <u>model equation</u> $y' = \lambda y$, leads to

$$\circledast \qquad y_{n+1} = E(\lambda h)\, y_n, \quad n = 0, 1, 2, \ldots, N-1$$

Then it is a <u>stable method</u> if $|E(\lambda h)| \leq 1$. The <u>stability region</u> is all such λ in the complex plane that satisfy

$$\boxed{|E(\lambda h)| \leq 1.} \quad \text{or} \quad \{\lambda : |E(\lambda h)| \leq 1\}$$

<u>Remark</u>: ① From $\circledast$ we can see that $y_{n+1} = (E(\lambda h))^{n+1} y_0$. Thus if y_0 is slightly changed to $\tilde{y}_0$, its perturbed sol^n is $\tilde{y}_{n+1} = (E(\lambda h))^{n+1} \tilde{y}_0$ and $|y_{n+1} - \tilde{y}_{n+1}| = |E(\lambda h)^{n+1}|\, |y_0 - \tilde{y}_0|$

$\leq |y_0 - \tilde{y}_0|$ if $|E(\lambda h)| \leq 1$, $\Rightarrow y_{n+1} \sim \tilde{y}_{n+1}$ if $y_0 \sim \tilde{y}_0$ and thus we have stability

② There are two reasons why $y' = \lambda y$ is chosen for the stability analysis:

(i) if a numerical method is unstable when it is applied to such a simple ~~problem~~ equation, it is unlikely to be stable for more complicated problems.

(ii) Any complicated ODE can be locally linearized and put into this form:

$$\begin{aligned} y' &= f(x, y) \\ &= \underbrace{f(x_n, y_n) + (x - x_n)\frac{\partial f}{\partial x}(x_n, y_n)} + (y - y_n)\frac{\partial f}{\partial y}(x_n, y_n) + \cdots \\ &\approx (y - y_n)\frac{\partial f}{\partial y}(x_n, y_n) + y_n' \end{aligned}$$

$$\Rightarrow \underbrace{(y - y_n)}_{Y'}{}' \approx \underbrace{(y - y_n)}_{Y}\, \underbrace{\frac{\partial f}{\partial y}(x_n, y_n)}_{\lambda}$$

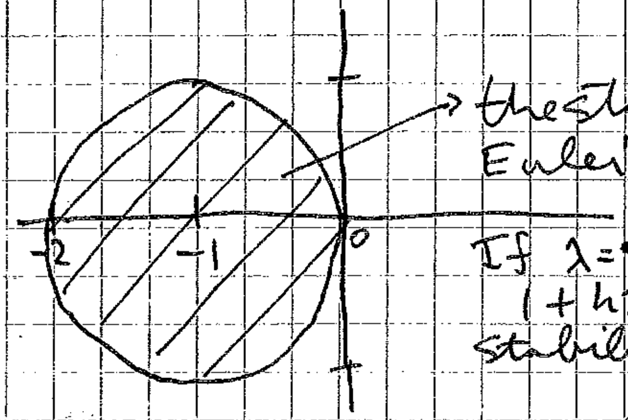
$$\Rightarrow \boxed{Y' = \lambda Y}$$ – the model equation!

Ex. Consider the Euler method $y_{n+1} = y_n + hf(x_n, y_n)$. We apply it to the model equation $y' = \lambda y$, i.e. $f(x,y) = \lambda y$. It leads to

$$y_{n+1} = y_n + h\lambda y_n = (1+h\lambda) y_n \Rightarrow \boxed{E(\lambda h) = 1 + h\lambda}$$

For stability we need $|1+h\lambda| \leq 1 \Leftrightarrow |h\lambda - (-1)| \leq 1$

Consider also complex $\lambda \rightarrow \lambda h = z$, $z = x+iy$)



the stability region for Euler's method.

If $\lambda = -10$, $h = 0.5$ then $1 + h\lambda = -4$ is not in the stability region.

Conclusion: For a numerical method to be useful, it must be stable.

Generalization / Improvements of Euler's Method

Objective: ① To obtain higher order methods (more accurate).
② To obtain methods with a larger stability region.

Overall Picture:

$$\boxed{\text{Euler's Method}}$$

Single step Methods
- Improved Euler
- Implicit Euler
- Modified Euler
- θ- method
- Runge-Kutta methods ⊛

Multi-step Methods
- Adams - Bashforth
- Adams - Moulton

Predictor/Corrector Methods

~~#82~~ ~~#80~~

Derive Numerical Methods from numerical Quadrature:

$$\frac{dy}{dx} = f(x,y) \Longrightarrow dy = f(x,y)\,dx$$

$$\Rightarrow \int_{x_n}^{x_{n+1}} dy = \int_{x_n}^{x_{n+1}} f(x, y(x))\,dx$$

$$\Rightarrow \boxed{y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y(x))\,dx}$$

- Left-point integration formula gives:

$$\int_{x_{n\cancel{0}}}^{x_{n+1}} f(x, y(x))\,dx = f(x_n, y(x_n))h + O(h^2)$$

$$\Rightarrow \boxed{y_{n+1} = y_n + h f(x_n, y_n)} \rightarrow \text{the Euler method.}$$

- Mid-point integration formula gives

$$\int_{x_n}^{x_{n+1}} f(x, y(x))\,dx = f(x_n + \tfrac{1}{2}h,\ y(x_n + \tfrac{1}{2}h))\,h + O(h^3)$$

$$\approx f(x_n + \tfrac{1}{2}h,\ y(x_n) + \tfrac{1}{2}h f(x_n, y(x_n))))$$

$$\Longrightarrow \boxed{y_{n+1} = y_n + h f(x_n + \tfrac{1}{2}h,\ y_n + \tfrac{1}{2}h f(x_n, y_n))}$$

called 'Modified Euler' $\left\{\begin{array}{l}\text{global error} = 2 \\ \text{function evaluations} = 2\end{array}\right.$

- Trapezoidal Rule gives

$$\int_{x_n}^{x_{n+1}} f(x, y(x))\,dx = \frac{h}{2}\left[f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))\right] + O(h^3)$$

$$\Longrightarrow \boxed{y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right]}$$

called 'implicit" Euler " $\left\{\begin{array}{l}\text{global error} = O(h^2) \\ \text{Disadvatage: have to solve} \\ \text{for } y_{n+1}\end{array}\right.$

Remark: the stability region of a numerical method solving a system of ODE's is obtained by applying the method to the model equations

$$\frac{dy}{dx} = \lambda_i y , \quad i = 1, 2, \dots, N$$

so all $\lambda_i h$ should lie inside the stable region if the method is to be stable.

Difficulties in solving stiff equations:
Usually stable region is bounded, to keep method stable $\lambda_i h$ should also be bounded. Now, if λ_{max} is large → h must be small → expensive computation and is also likely to lead to roundoff errors. Implicit methods ~~usually~~ usually have large stability domains → good for stiff problems.

Boundary Value problems (Baby elliptic PDE's) eg. $\Delta u = 0$ Laplace's eqn.
We consider the 2nd-order nonlinear/linear BVP:

$$\text{BVP} \begin{cases} y'' = f(x, y, y') , & x \in [a, b] \\ y(a) = \alpha , \quad y(b) = \beta & \text{(Boundary conditions of 1st kind (Dirichlet))} \end{cases}$$

A theoretical result: The above BVP has a unique solution if:

(i) $f(x, y, z)$ is continuous
(ii) f_y, f_z exist and are continuous
(iii) $f_y > 0$ and f_z is bounded.

eg. $f(x, y, y') = y - 4x e^x$, $f(x, y, y') = (1 + x^2) y - y'$ etc.

Question: How do we find the solution numerically? (Finite difference method).

[Note: an PDE $A u_{xx} + 2B u_{xy} + C u_{yy} = F(x, y, u, u_x, u_y)$
is elliptic if $AC - B^2 > 0$, parabolic if $AC - B^2 = 0$, hyperbolic $AC - B^2 < 0$.
(Laplace's eqn) (Heat eqn.) (Wave eqn.)

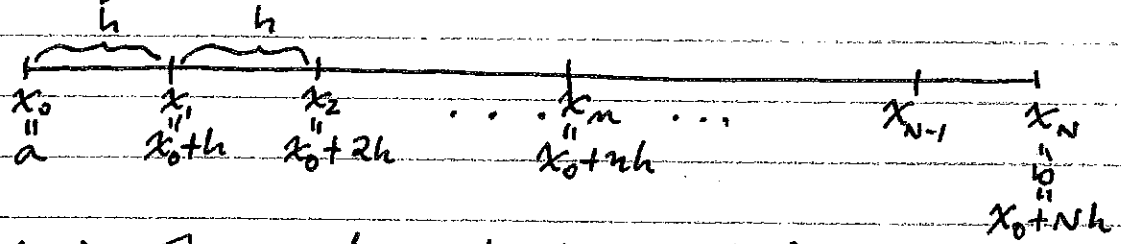
Shooting Method: satisfy ODE initially & iterate until B.C. is satisfied

Could also satisfy B.C. & then iterate until ODE is satisfied:
→ F.D., Galerkin methods

System of algebraic equations rather than sequence of IVP's (as in shooting method)

Idea: Discretize the space $[a,b]$ and replace the derivatives by difference quotients:

$h = \frac{b-a}{N}$



Let $y_n \approx y(x_n)$. Then at node x_n, $y_n'' = f(x_n, y_n, y_n'')$.

Note that:

$$y_n' = \begin{cases} (y_{n+1} - y_{n-1})/2h & \text{(central difference)} \\ (y_n - y_{n-1})/h & \text{(backward difference)} \\ (y_{n+1} - y_n)/h & \text{(forward difference)} \end{cases}$$

$$y_n'' = \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} \quad \text{(central difference)}$$

$$\text{BVP} \Rightarrow \begin{cases} \dfrac{y_{n+1} - 2y_n + y_{n-1}}{h^2} = f\left(x_n, y_n, \dfrac{y_{n+1} - y_{n-1}}{2h}\right), \; n = 1:N-1 \\ y_0 = \alpha, \; y_N = \beta \to \text{incorporate in RHS.} \end{cases}$$

$$\Rightarrow \underbrace{\begin{bmatrix} -2 & 1 & 0 & \bigcirc \\ 1 & -2 & 1 & \\ & \ddots & \ddots & 1 \\ \bigcirc & & 1 & -2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix}}_{y} = \underbrace{h^2 \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix} - \overset{\text{from B.C.'s}}{\begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ \beta \end{bmatrix}}}_{\gamma(y)}$$

where $f_n = f(x_n, y_n, (y_{n+1} - y_{n-1})/2h)$, $n = 1, \ldots, N-1$

$$\Rightarrow \boxed{Ay = \gamma(y)} \longrightarrow$$ a system of N-1 nonlinear algebraic equations for N-1 unknowns: $y_1, \ldots, y_{N-1}$.

Linear BVP - A special case

$$\text{(i)} \quad \begin{cases} y'' + Q(x)y = F(x), & x \in [a,b] \\ y(a) = \alpha, \quad y(b) = \beta \end{cases}$$

$$\rightarrow f(x, y, y') = -Q(x)y + F(x), \quad Q(x) < 0.$$

$$\Rightarrow \gamma(y) = h^2 \begin{bmatrix} f_1 \\ \vdots \\ f_{N-1} \end{bmatrix} - \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ \beta \end{bmatrix} = h^2 \begin{bmatrix} -Q(x_1)y_1 + F(x_1) \\ \vdots \\ -Q(x_{N-1})y_{N-1} + F(x_{N-1}) \end{bmatrix} - \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ \beta \end{bmatrix}$$

$$= \underbrace{h^2 \begin{pmatrix} -Q(x_1) & & \bigcirc \\ & \ddots & \\ \bigcirc & & -Q(x_{N-1}) \end{pmatrix}}_{B} \underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_{N-1} \end{pmatrix}}_{y} + \underbrace{\begin{bmatrix} h^2 F(x_1) \\ \vdots \\ h^2 F(x_{N-1}) \end{bmatrix} - \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ \beta \end{bmatrix}}_{C}$$

$$\Rightarrow Ay - By - c = 0 \Rightarrow \boxed{(A - B)y = c}$$

where

$$A - B = \begin{bmatrix} -2 + h^2 Q_1 & 1 & & \bigcirc \\ 1 & -2 + h^2 Q_2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & & 1 \\ \bigcirc & & 1 & -2 + h^2 Q_{N-1} \end{bmatrix}$$

$\therefore$ A-B is <u>symmetric</u>, <u>tridiagonal</u>, <u>diagonal dominant</u> !!
→ ~~use~~ Jacobi, Gauss-Seidel, S.O.R. methods converge!

<u>Newton's method</u> for solving <u>nonlinear algebraic systems</u>: Find $\bar{x}$ s.t. $G(\bar{x}) = 0$.

Finite Difference Method for solving Heat Equation

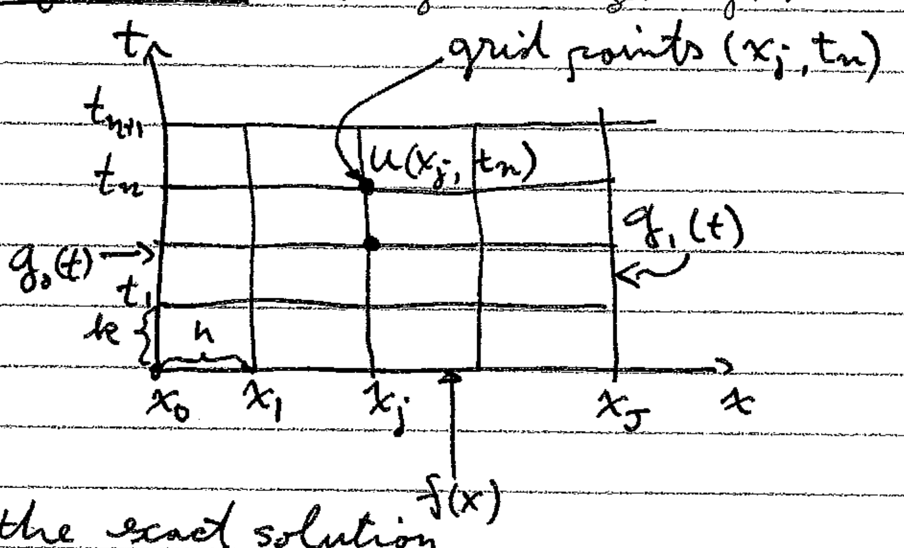
– fully discretized by F.D. in x + t

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \ 0 < x < 1$$

$$u(x,0) = f(x), \ 0 \le x \le 1$$

$$u(0,t) = g_0(t), \ t > 0$$

$$u(1,t) = g_1(t), \ t > 0$$



Notation: $U(x,t)$ is the exact solution

$u(x_j, t_n)$ is the exact value of $u(x,t)$ at the gridpoint (x_j, t_n).

U_j^n is the approximate value of $u(x_j, t_n)$,

i.e. $U_j^n \approx u(x_j, t_n)$

Note that: $$\left.\frac{\partial u}{\partial t}\right|_{(x_j, t_n)} = \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{k} + O(k)$$

(1st-order forward difference in time)

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_j, t_n)} = \frac{u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n)}{h^2} + O(h^2)$$

(2nd-order central difference in space)

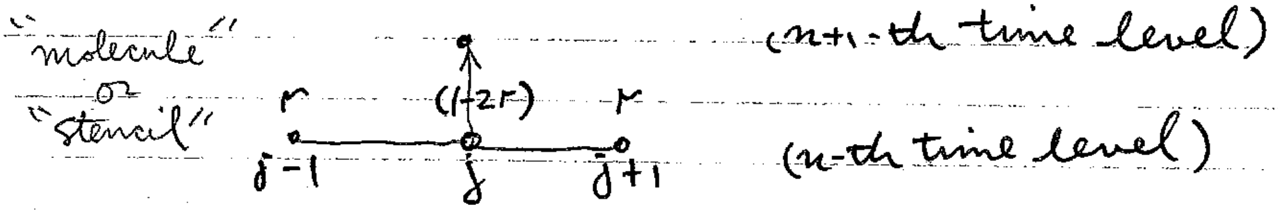
$$\rightarrow \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{k} + O(k) = \alpha\left(\frac{u(x_{j+1}, t_n) - 2u(x_j, t_n) - u(x_{j-1}, t_n)}{h^2} + O(h^2)\right)$$

$$\rightarrow \frac{U_j^{n+1} - U_j^n}{k} = \alpha\left(\frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{h^2}\right)$$

$$\Rightarrow \boxed{u_j^{n+1} = r\,u_{j-1}^{n} + (1-2r)\,u_j^{n} + r\,u_{j+1}^{n}, \quad r = \frac{\alpha k}{h^2}}$$

This is a Forward-Time Centred-Space (FTCS) finite difference approximation equation

⇒ <u>Two-level explicit</u> formula $GTE = O(k+h^2)$



<u>Example</u>: choose $f(x)=1$, $g_0(t)=g_1(t)=0$.

(i) $h=1/4$, $k=1/32$, $\alpha=1 \rightarrow r = \frac{\alpha k}{h^2} = \frac{1\cdot\frac{1}{32}}{\frac{1}{16}} = \frac{1}{2}$ $\boxed{r=1/2}$

$$\Rightarrow \boxed{u_j^{(n+1)} = \frac{1}{2}\left(u_{j-1}^{n} + u_{j+1}^{n}\right)}$$

smaller stepsize in time more expensive, slower!

(ii) $h=1/4$, $k=1/16$, $\alpha=1 \rightarrow \boxed{r=1}$

$$\Rightarrow \boxed{u_j^{(n+1)} = u_{j-1}^{n} - u_j^{n} + u_{j+1}^{n}}$$

larger stepsize in time, faster, no division operations

0	1/4	1/4	1/4	0
0	1/4	1/2	1/4	0
0	1/2	1/2	1/2	0
0	1/2	1	1/2	0
0	①	1	1	0
①	1	①	1	1

$k = 1/32$

$h =$ 1/4 1/4 1/4 1/4

Case (i) with r=1/2
An exponentially decaying solution (approaches the equilibrium 0)

0	5	-7	5	0
0	-2	3	-2	0
0	1	-1	1	0
0	0	1	0	0
0	①	1	1	0
①	①	①	1	1

1/16

1/4

Case (ii) with r=1
An <u>explosive oscillatory</u> solution (physically impossible)
→ <u>unstable</u>, doesn't converge

Convergence, consistency, stability

Key concepts for evaluating a numerical method for a PDE.

Convergence: A solution u_j^n to a finite difference equation which approximates a given PDE is said to be convergent if

$$\lim_{\substack{h\to 0\\ k\to 0}} u_j^n = u(x_j, t_n) \qquad (x_j, t_n \text{ fixed})$$

Conclusion: If $r \le 1/2$ the FTCS method is convergent. Why?

Let $e_j^n = u(x_j, t_n) - u_j^n$ be the ~~truncated~~ truncation error. Then

$$u_j^n = u(x_j, t_n) - e_j^n \text{. Substitute into FTCS}$$

$$\to u(x_j, t_{n+1}) - e_j^{n+1} = r\left(u(x_{j-1}, t_n) - e_{j-1}^n\right) + (1-2r)\left(u(x_j, t_n) - e_j^n\right) + r\left(u(x_{j+1}, t_n) - e_{j+1}^n\right)$$

$$\Rightarrow e_j^{n+1} = r\, e_{j-1}^n + (1-2r)\, e_j^n + r\, e_{j+1}^n + \Big[u(x_j, t_{n+1}) - u(x_j, t_n) - r\big(u(x_{j+1}, t_n) + u(x_{j-1}, t_n) - 2u(x_j, t_n)\big)\Big]$$

$$[\ \ ] = [O(k) + O(h^2)]\,k \text{ (using FTSC approx.)}$$

$$\text{or } [\ \ ] \le M(k^2 + kh^2)$$

$\boxed{\text{If } r \le 1/2}$ $\to (1-2r) \ge 0$

$$\Rightarrow |e_j^{n+1}| \le r\,|e_{j-1}^n| + (1-2r)\,|e_j^n| + r\,|e_{j+1}^n| + M(k^2 + kh^2)$$

Let $e_{\max}^n = \max_{j=1:J-1} |e_j^n|$. Then

$$e_j^{n+1} \le r\, e_{\max}^n + (1-2r)\, e_{\max}^n + r\, e_{\max}^n + Mk(k+h^2)$$

$$\to e_j^{n+1} \le \cancel{r}\, e_{\max}^n + Mk(k+h^2)$$

$$\le e_{\max}^{n-1} + 2Mk(k+h^2) \le \dots \le \overset{\text{initial error is zero}}{e_{\max}^{0}} + M\overbrace{(n+1)k}^{t_{n+1}}(k+h^2)$$

$$(e_{\max}^n \le e_{\max}^{n-1} + Mk(k+h^2))$$

$= M t_{n+1}(k+h^2) \longrightarrow 0$ as $h, k \to 0$

$\Rightarrow$ GTE $= O(k+h^2)$

can also show that $r > 1/2$ is not convergent in a similar way

Thus the FTCS method is <u>convergent</u> if $r \leq 1/2$.

what about $r > 1/2$? → we will show this is unstable and hence <u>not</u> convergent.

<u>Consistency</u>: A finite difference formula is said to be <u>consistent</u> with a PDE if as $h, k \to 0$, the difference <u>equation</u> becomes the PDE at each point in the solution domain.

<u>Conclusion</u>: The FTCS method is consistent with the heat equation <u>for any $r > 0$</u> because the (global) truncation error is $O(k+h^2)$ which $\to 0$ as $h, k \to 0$. More specifically, using Taylor series, we find the (global) truncation error is

$$e_j^n = -\frac{\alpha h^2}{2}\left(r - \tfrac{1}{6}\right)\frac{\partial^4 u}{\partial x^4}(x_j, t_n) + O(k^2+h^4)$$

→ $r = 1/6$ is the optimum value!

→ the solution of the FTCS finite difference equation approaches the exact solution more rapidly when $r = 1/6$ than for other values of $r \leq 1/2$.

$\Rightarrow$ Consistency is <u>necessary</u>, but <u>not</u> <u>sufficient</u> to ~~give a~~ have a correct method.

<u>Stability</u>: A finite difference method is said to be <u>stable</u> if a small round-off error at one stage is not amplified by later computations

Note that we have 3 types of solutions:

$u(x,t)$ - the exact solution

u_j^n - the approximate solution (exact solution of FTCS)

U_j^n - the numerical (or computer) solution

} ~~truncation~~ Truncation error → convergence

} roundoff error → stability

$$\Rightarrow \boxed{\xi_j^n = u_j^n - U_j^n}$$ is called the roundoff error

We need $|\xi_j^n|$ to be bounded or tend to zero as the computation goes on.

Note that U_j^n also satisfies the FTCS ~~as~~ scheme, i.e.

$$U_j^{n+1} = rU_{j-1}^n + (1-2r)U_j^n + rU_{j+1}^n$$

Thus the error ξ_j^n also satisfies

$$\boxed{\xi_j^{n+1} = r\xi_{j-1}^n + (1-2r)\xi_j^n + r\xi_{j+1}^n}$$

Example: Consider again $f(x)=1$, $g_0(t)=g_1(t)=c$.

(i) $h=1/4$, $k=1/32$, $\alpha=1$, $r=1/2$

$$\longrightarrow \xi_j^{(n+1)} = \frac{1}{2}\left(\xi_{j-1}^n + \xi_{j+1}^n\right)$$

(ii) $h=1/4$, $k=1/16$, $\alpha=1$, $r=1 \longrightarrow \xi_j^{n+1} = \xi_{j-1}^n - \xi_j^n + \xi_{j+1}^n$

& Assume errors at boundary are zero.

0	$\varepsilon/8$	0	$\varepsilon/8$	0
0	0	$\varepsilon/4$	0	0
0	$\varepsilon/4$	0	$\varepsilon/4$	0
0	0	$\varepsilon/2$	0	0
0	$\varepsilon/2$	0	$\varepsilon/2$	0
0	0	ε	0	0
0	0	0	0	0

} Error propagation due to introduction of error $\xi_2^1 = \varepsilon$ with $r=1/2$ → stable (error does not grow)

0	29ε	41ε ~~[illegible]~~	29ε	0
0	-12ε	17ε	-12ε	0
0	5ε	-7ε	5ε	0
0	-2ε	3ε	-2ε	0
0	ε	$-\varepsilon$	ε	0
0	0	ε	0	0
0	0	0	0	0

} $r=1$ error grows ↓ UNSTABLE

se results illustrate:

ax's Equivalence Theorem: Given a properly posed linear initial value problem and a finite difference approximation to it that satisfies the consistency condition, then

stability ⟺ (equivalent to) convergence

\+ Consistency

or i.e. stability and consistency together are a necessary & sufficient condition for convergence.

Remarks:

① well-posedness of an IVP means that the solution of the PDE depends continuously on the given initial data. The diffusion (heat) equation ands its associated IVP & BVP ~~are~~ is well-posed.

② We need only show the stability of the method to prove it converges, which is good since stability is easier to show provided the PDE is well-posed and the method is consistent with the PDE.

Stability Analysis Methods:

(1) Discrete perturbation stability method
(2) Matrix method
(3) Von Neumann's method (also called Fourier series method) ⟶ Most useful in practice.