Partial Differential Equations

Yubo Tao

May 21, 2019

Outline

- Overview
- Ordinary differential equations (ODE)
  - Initial value problem (初值问题)
  - Error analysis (误差分析)
  - Boundary value problem (边值问题)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

ODE vs PDE

- Ordinary differential equations (函数为单变量)

$$\frac{du}{dx} = 0, u = u(x)$$

- Partial differential equations (函数为多变量)

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, u = u(x, y)$$

ODE

- Initial value problem (初值问题)

$$\frac{dy}{dt} = y$$

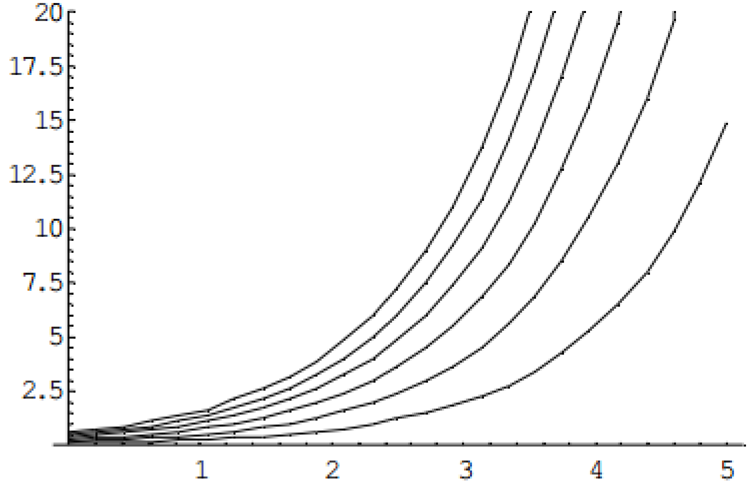
$$y(t_0) = y_0 \text{ (initial value)}$$

- Separation of variables (分离变量法)

$$\frac{dy}{y} = \mathrm{d}t \Rightarrow \int_{y_0}^{y} \frac{dy}{y} = \int_{t_0}^{t} dt$$

$$\ln y - \ln y_0 = t - t_0$$

$$y = y_0 e^{t-t_0} \text{ (a group of solutions)}$$



$t_0=0$, y_0 = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6

PDE

- Three typical PDEs
  - Elliptic equation (椭圆方程)
    $\Delta u = f$ (Poisson方程)
  - Parabolic equation (抛物线方程)
    $u_t - \beta\Delta u = f$ (热传导方程)
  - Hyperbolic equation (双曲线方程)
    $u_{tt} - a^2\Delta u = f$ (波动方程)
- Gradient： $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$
- Laplace operator： $\Delta = \nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

Classification

- 对两个变量的二阶拟线性方程：

$$a(x,y,t)\frac{\partial^2 u}{\partial x^2}+2b(x,y,t)\frac{\partial^2 u}{\partial x\partial y}+c(x,y,t)\frac{\partial^2 u}{\partial y^2}+f(x,y,t,\dots)=0$$

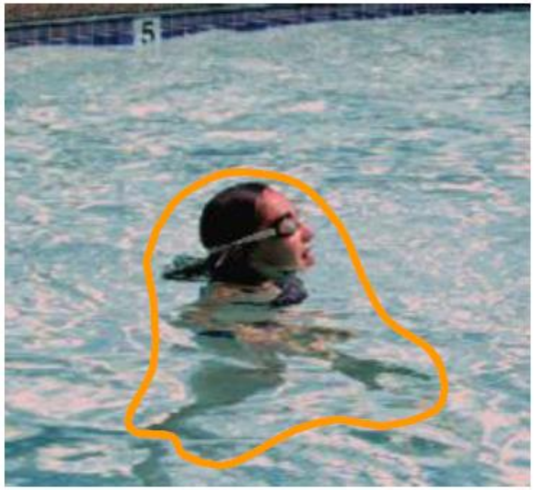
- 对于固定的(x,y,t)，
  - 如果$F=ac-b^2>0$，方程是Elliptic equation(椭圆型)
  - 如果$F=ac-b^2<0$，方程是Hyperbolic equation(双曲型)
  - 如果$F=ac-b^2=0$，方程是Parabolic equation(抛物型)

Classification

- $\sum_{ij} a_{ij} \frac{\partial f}{\partial x_i \partial x_j} + \sum_i b_i \frac{\partial f}{\partial x_i} + cf = 0$
- $(\nabla^T A \nabla + \nabla \cdot b + c) f = 0$
  - If A is positive or negative definite, the system is elliptic
  - If A is positive or negative semidefinite, the system is parabolic
  - If A has only one eigenvalue of different sign from the rest, the system is hyperbolic
  - If A satisfies none of the criteria, the system is ultrahyperbolic (超双曲线)

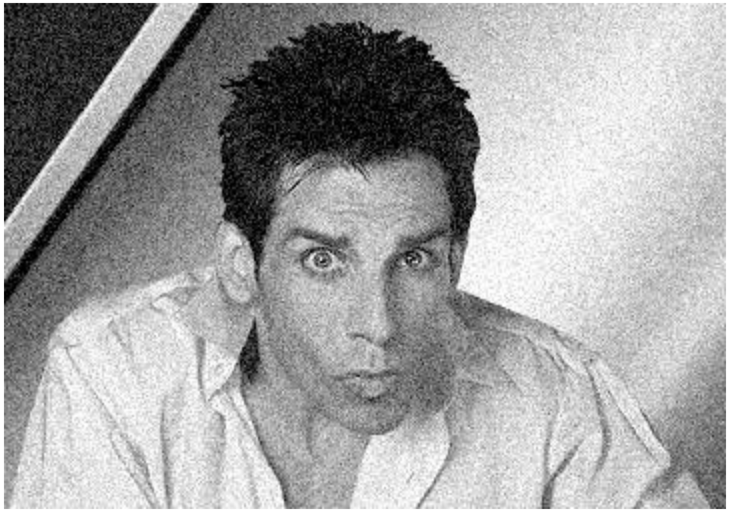
Poisson Equation

- Image synthesis



Heat Diffusion Equation

- Image denoise



t = 0.016

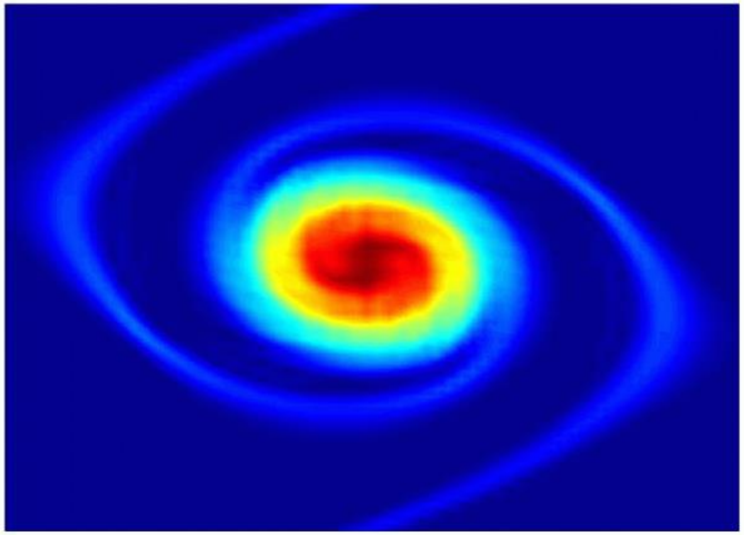
Advection-Diffusion(对流-扩散方程)

- Advection-Diffusion (气流运动)

$$\frac{\partial \omega}{\partial t} + [\psi, \omega] = \nu \nabla^2 \omega$$

$$\nabla^2 \psi = \omega$$

$$[\psi, \omega] = \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} - \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x}$$



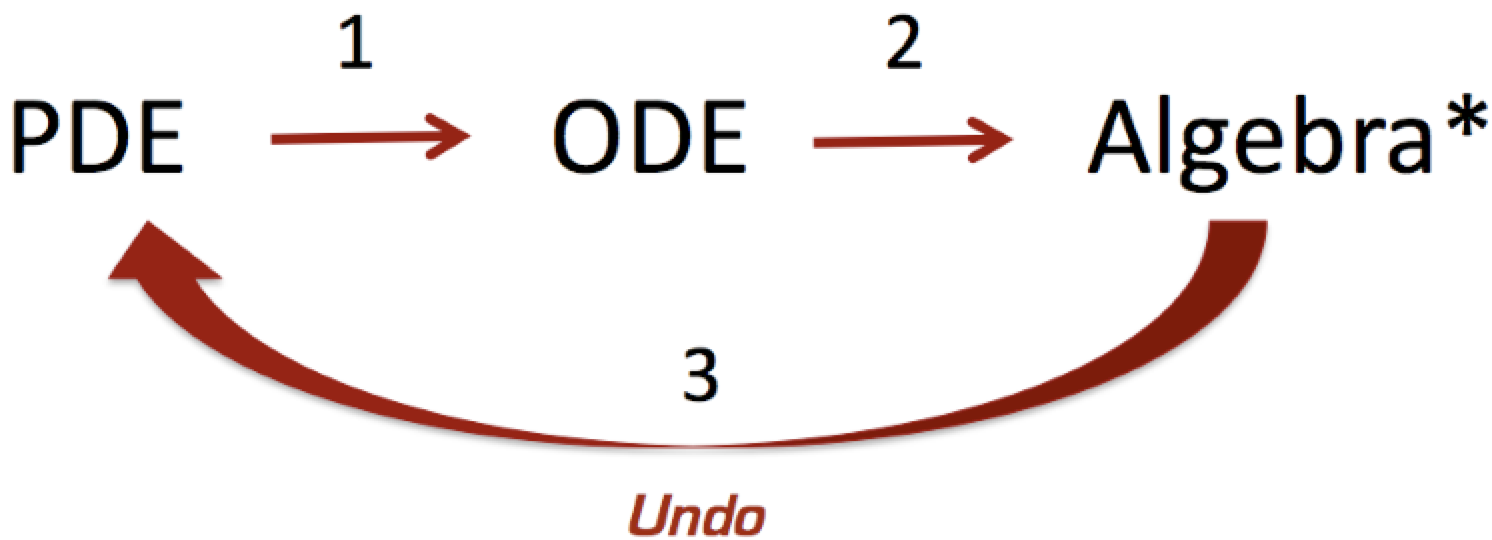
parabolic: $\frac{\partial \omega}{\partial t} = \nu \nabla^2 \omega$

elliptic: $\nabla^2 \psi = \omega$

hyperbolic: $\frac{\partial \omega}{\partial t} + [\psi, \omega] = 0$

偏微分方程数值求解

- 为什么数值求解？
  - 大多数偏微分方程无法通过解析求解
  - 数值求解可以利用真实的测量数据
- 数值求解没有通解，同时有离散误差



Outline

- Overview
- Ordinary differential equations (ODE)
  - Initial value problem (初值问题)
    - Euler, Runge-Kutta, Adams methods
  - Error analysis (误差分析)
    - Discretization error, Round-off, Stability
  - Boundary value problem (边值问题)
    - Shooting method, direct solve, relaxation
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

J. Nathan Kutz, Data-Driven Modeling & Scientific Computation, Methods for Complex Systems & Big Data, Oxford, 2013

Taylor Expansions

- Taylor 用级数的方法来求解微分方程，并提出了有限差分的方法
- 函数f的泰勒展开$h \to 0$

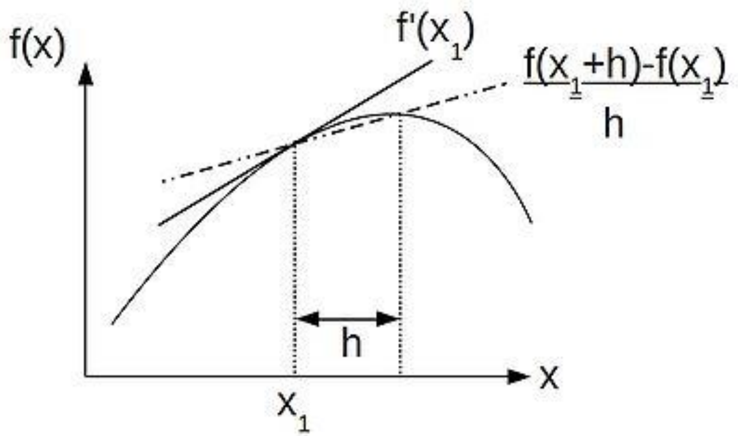
$$f(x+h)=\sum_{k=0}^{n} f^{(k)}(x)\frac{h^k}{k!}+O(h^{n+1})$$

$$f(x+h)=f(x)+hf'(x)+\frac{h^2}{2}f''(x)+O(h^3)$$

Taylor Expansions

$$f(x+h)=f(x)+hf'(x)+\frac{h^2}{2}f''(x)+O(h^3)$$

- 前向差分 $$f'(x)=\frac{f(x+h)-f(x)}{h}+O(h)$$
- 后向差分 $$f'(x)=\frac{f(x)-f(x-h)}{h}+O(h)$$
- 中心差分 $$f'(x)=\frac{f(x+h)-f(x-h)}{2h}+O(h^2)$$
- 二阶导数 $$f''(x)=\frac{f(x+h)-2f(x)+f(x-h)}{h^2}+O(h^2)$$



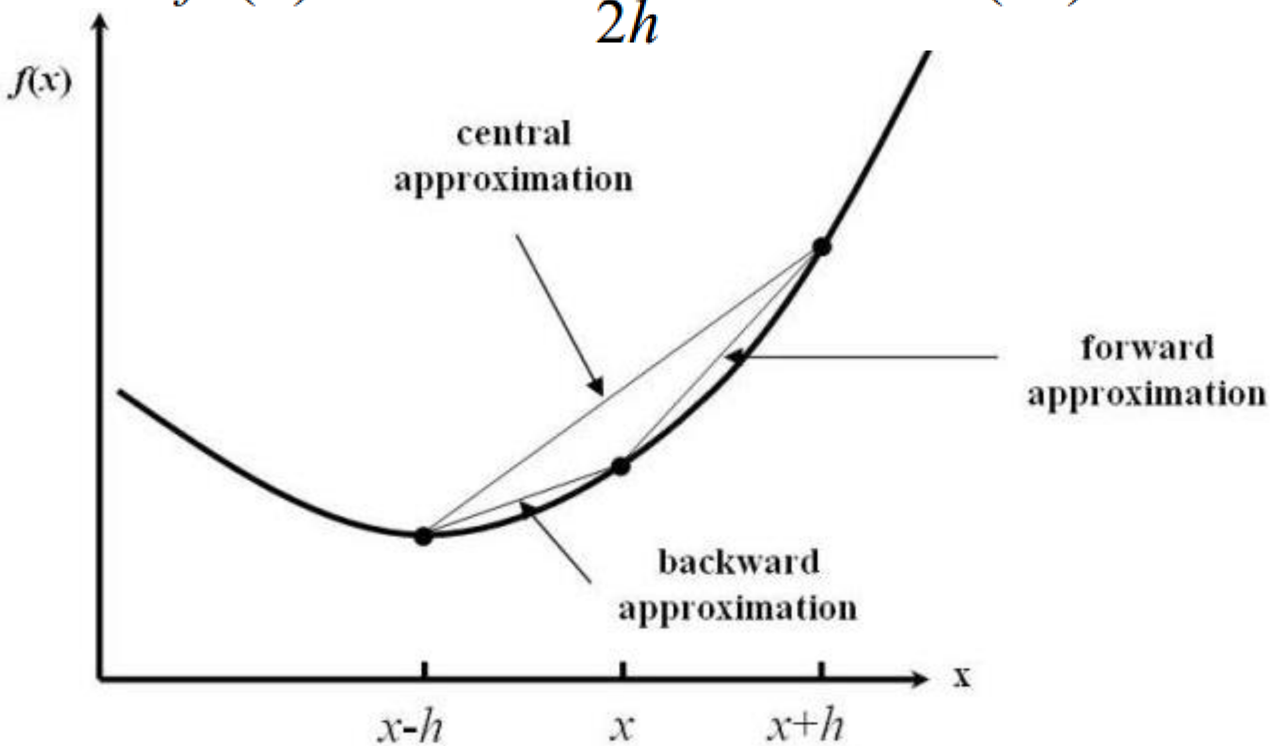
Taylor Expansions

- 前向差分
- 后向差分
- 中心差分

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$$

$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h)$$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$



Initial Value Problem

- ODE

$$\frac{d\boldsymbol{y}}{dt} = f(\boldsymbol{y}, t)$$
$$\boldsymbol{y}(0) = y_0$$
$$t \in [0, T]$$

- Euler Method

$$\frac{d\mathbf{y}}{dt} = \lim_{\Delta t \to 0} \frac{\Delta \mathbf{y}}{\Delta t}$$

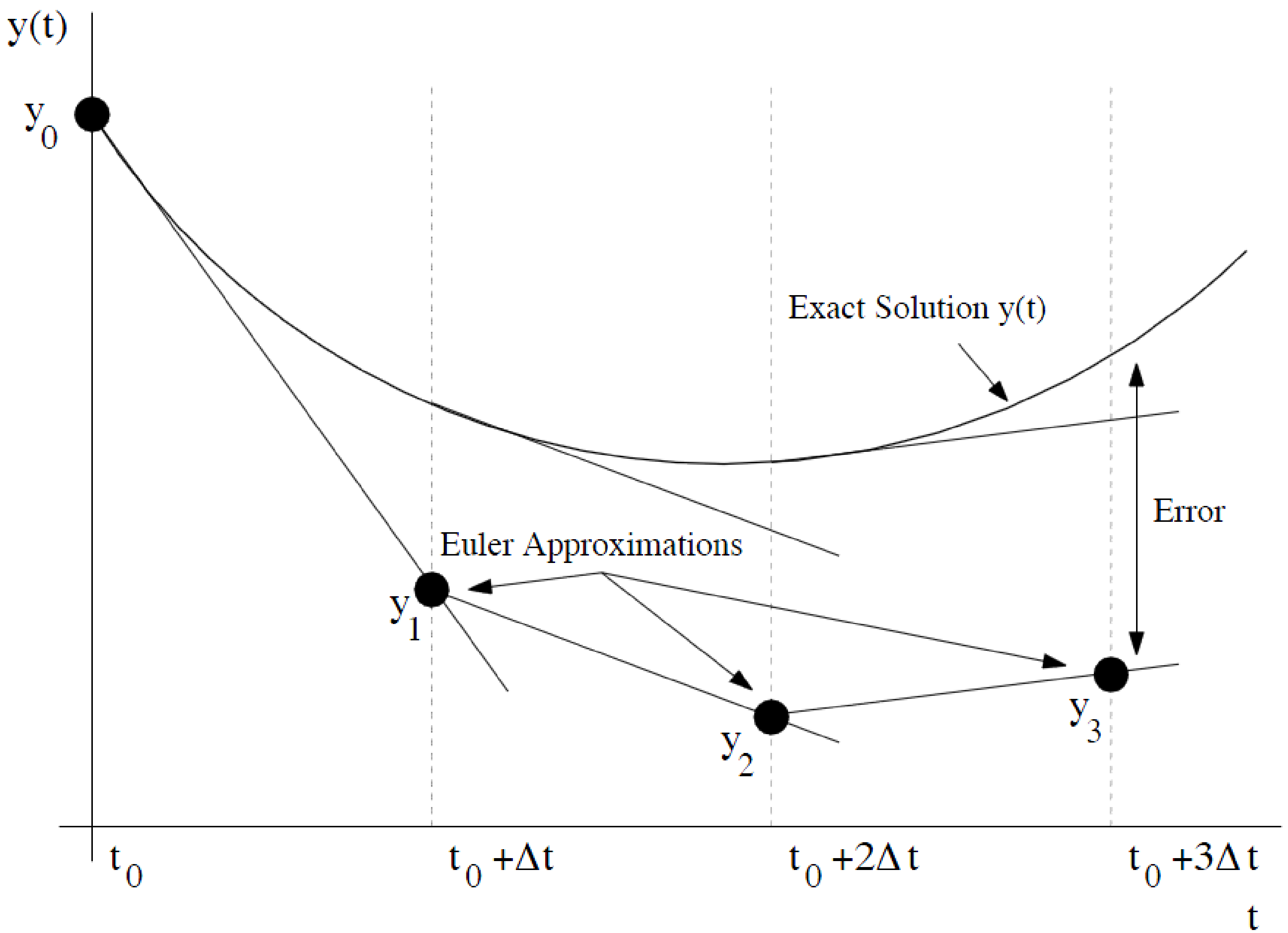
$$\Delta t = t_{n+1} - t_n$$

$$\frac{d\mathbf{y}}{dt} = f(\mathbf{y}, t) \quad \Rightarrow \quad \frac{\mathbf{y}_{n+1} - \mathbf{y}_n}{\Delta t} \approx f(\mathbf{y}_n, t_n)$$

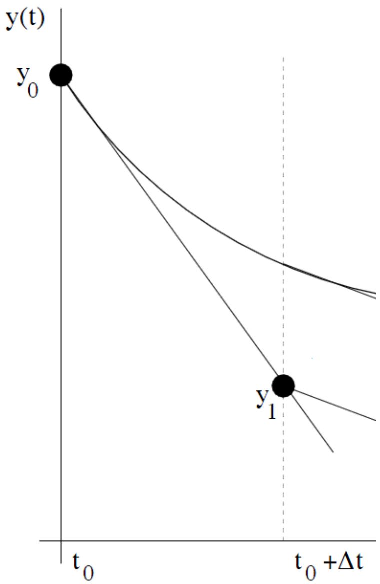
Euler Method

- 迭代求解

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot f(\mathbf{y}_n, t_n)$$



Euler Method



- 迭代求解 $\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot \phi$
  - ϕ降低单个时间步Δt数值误差
  - ϕ是t_n时间点导数
  - ϕ可以扩展为$t_n, t_{n+\frac{1}{2}}, t_{n+1}$等时间点导数混合

$$\mathbf{y}(t+\Delta t) = \mathbf{y}(t) + \Delta t\left[A\boxed{f(t,\mathbf{y}(t))} + B\boxed{f(t+P\cdot\Delta t, \mathbf{y}(t) + Q\Delta t\cdot f(t,\mathbf{y}(t)))}\right]$$

$$f(\boxed{t+P\cdot\Delta t}, \boxed{\mathbf{y}(t)+Q\Delta t\cdot f(t,\mathbf{y}(t))}) =$$
$$f(t,\mathbf{y}(t)) + P\Delta t\cdot f_t(t,\mathbf{y}(t)) + Q\Delta t\cdot f_{\mathbf{y}}(t,\mathbf{y}(t))\cdot f(t,\mathbf{y}(t)) + O(\Delta t^2)$$

$$\mathbf{y}(t+\Delta t) = \mathbf{y}(t) + \Delta t(A+B)f(t,\mathbf{y}(t))$$
$$+PB\Delta t^2\cdot f_t(t,\mathbf{y}(t)) + BQ\Delta t^2\cdot f_{\mathbf{y}}(t,\mathbf{y}(t))\cdot f(t,\mathbf{y}(t)) + O(\Delta t^3)$$

Generalized Euler Method

$$\mathbf{y}(t+\Delta t) = \mathbf{y}(t) + \Delta t(A+B)f(t,\mathbf{y}(t))$$
$$+PB\Delta t^2 \cdot f_t(t,\mathbf{y}(t)) + BQ\Delta t^2 \cdot f_{\mathbf{y}}(t,\mathbf{y}(t)) \cdot f(t,\mathbf{y}(t)) + O(\Delta t^3)$$

$$\mathbf{y}(t+\Delta t) = \mathbf{y}(t) + \Delta t \cdot f(t,\mathbf{y}(t)) + \frac{1}{2}\Delta t^2 \cdot f_t(t,\mathbf{y}(t))$$
$$+\frac{1}{2}\Delta t^2 \cdot f_{\mathbf{y}}(t,\mathbf{y}(t))f(t,\mathbf{y}(t)) + O(\Delta t^3)$$

$$\boxed{\frac{d\boldsymbol{y}}{dt} = f(\boldsymbol{t},\boldsymbol{y})}$$

- Heun’s Method A = ½
- Modified Euler-Cauchy A = 0

$$A+B=1$$
$$PB=\frac{1}{2}$$
$$BQ=\frac{1}{2}$$

Generalized Euler Method

$$\mathbf{y}(t+\Delta t)=\mathbf{y}(t)+\Delta t\left[Af(t,\mathbf{y}(t))+Bf(t+P\cdot\Delta t,\mathbf{y}(t)+Q\Delta t\cdot f(t,\mathbf{y}(t)))\right]$$

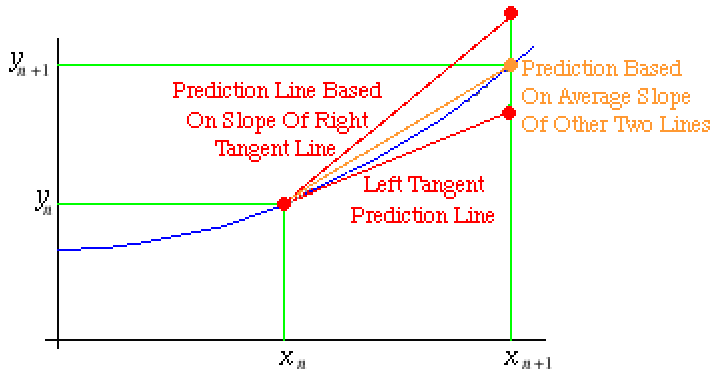
$$A+B=1$$

$$PB=\frac{1}{2}$$

$$BQ=\frac{1}{2}$$

- Heun’s Method A = ½
  - t_n时间点导数和t_{n+1}时间点导数

$$\mathbf{y}(t+\Delta t)=\mathbf{y}(t)+\frac{\Delta t}{2}\left[f(t,\mathbf{y}(t))+f(t+\Delta t,\mathbf{y}(t)+\Delta t\cdot f(t,\mathbf{y}(t)))\right]$$



Generalized Euler Method

$$\mathbf{y}(t+\Delta t) = \mathbf{y}(t) + \Delta t\,[Af(t,\mathbf{y}(t)) + Bf(t+P\cdot\Delta t, \mathbf{y}(t) + Q\Delta t\cdot f(t,\mathbf{y}(t)))]$$

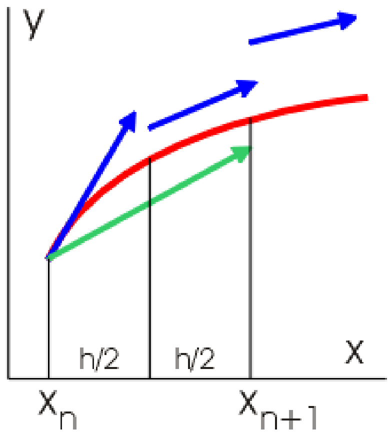
- Modified Euler-Cauchy A = 0
  - $t_{n+\frac{1}{2}}$ 时间点导数
  - Second order Runge-Kutta (龙格-库塔)

$$A + B = 1$$

$$PB = \frac{1}{2}$$

$$BQ = \frac{1}{2}$$

$$\mathbf{y}(t+\Delta t) = \mathbf{y}(t) + \Delta t\cdot f\left(t+\frac{\Delta t}{2}, \mathbf{y}(t) + \frac{\Delta t}{2}\cdot f(t,\mathbf{y}(t))\right)$$



Runge-Kutta Methods

- Runge-Kutta Methods
  - Iterate forward in time given a single initial point
- 4th order Runge-Kutta Method
  - 局部截断误差为$O(\Delta t^5)$，总累计误差为$O(\Delta t^4)$

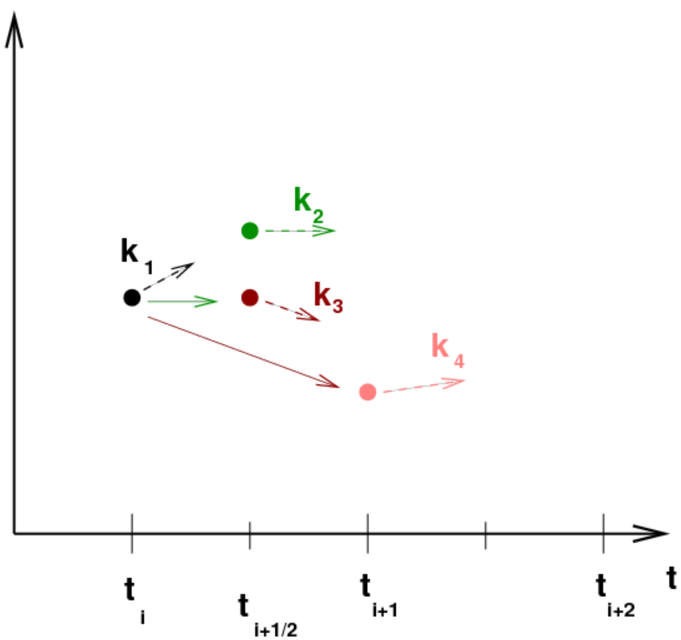
$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{6}\left[f_1 + 2f_2 + 2f_3 + f_4\right]$$

$$f_1 = f(t_n, \mathbf{y}_n)$$

$$f_2 = f\left(t_n + \frac{\Delta t}{2}, \mathbf{y}_n + \frac{\Delta t}{2} f_1\right)$$

$$f_3 = f\left(t_n + \frac{\Delta t}{2}, \mathbf{y}_n + \frac{\Delta t}{2} f_2\right)$$

$$f_4 = f\left(t_n + \Delta t, \mathbf{y}_n + \Delta t \cdot f_3\right)$$



4^{th} order Runge-Kutta Method

- Intermediate time-steps

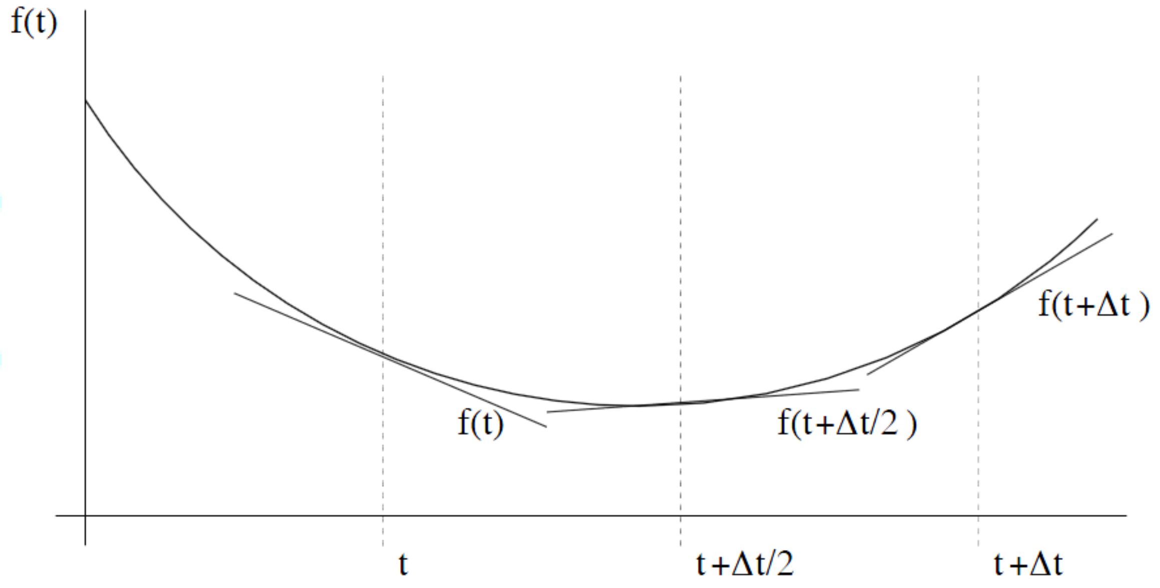
$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{6}\left[f_1 + 2f_2 + 2f_3 + f_4\right]$$

$$f_1 = f\left(t_n, \mathbf{y}_n\right)$$

$$f_2 = f\left(t_n + \frac{\Delta t}{2}, \mathbf{y}_n + \frac{\Delta t}{2} f_1\right)$$

$$f_3 = f\left(t_n + \frac{\Delta t}{2}, \mathbf{y}_n + \frac{\Delta t}{2} f_2\right)$$

$$f_4 = f\left(t_n + \Delta t, \mathbf{y}_n + \Delta t \cdot f_3\right)$$



Adams Method

- Derivative and Taylor expansions
  - Euler method $\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot \phi$
  - Runge-Kutta methods $\Delta t \cdot \phi = \int_{t_n}^{t_{n+1}} f(t, y) dt$
- Theorem of calculus

$$\frac{d\mathbf{y}}{dt} = f(\mathbf{y}, t) \quad \Rightarrow \quad \mathbf{y}(t + \Delta t) - \mathbf{y}(t) = \int_t^{t+\Delta t} f(t, y) dt$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \int_{t_n}^{t_{n+1}} f(t, y) dt$$

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + \int_{t_n}^{t_{n+1}} p(t, y) dt$$

- $f(t,y) \approx p(t,y)$
  - $p(t,y)$ is a polynomial

Adams-Bashforth Scheme

- How to determine *p*(*t*,*y*)
- Adams-Bashforth scheme

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + \int_{t_n}^{t_{n+1}} p(t,y)dt$$

  – Current point and a determined number of past points to evaluate the future solution
  – Order of accuracy relates to *p*(*t*,*y*)
- First-order scheme

$$p_1(t) = \text{constant} = f(t_n, \mathbf{y}_n)$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot f(t_n, \mathbf{y}_n)$$ (Euler Method)

Adams-Bashforth Scheme

- Second-order scheme $\mathbf{y}_{n+1} \approx \mathbf{y}_n + \int_{t_n}^{t_{n+1}} p(t,y)dt$
  - A two-step algorithm requires two initial conditions

$$p_2(t) = f_{n-1} + \frac{f_n - f_{n-1}}{\Delta t}(t - t_{n-1})$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \int_{t_n}^{t_{n+1}} \left(f_{n-1} + \frac{f_n - f_{n-1}}{\Delta t}(t - t_n) \right) dt$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{2}\left[3f(t_n, \mathbf{y}_n) - f(t_{n-1}, \mathbf{y}_{n-1})\right]$$

- High-order schemes
  - Require a boot strap to generate initial conditions

Adams-Moulton Scheme

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + \int_{t_n}^{t_{n+1}} p(t,y)dt$$

- How to determine *p*(*t*,*y*)
- Adams-Bashforth scheme
  - Current point and past points
- Adams-Moulton scheme
  - Current point and future points
- First-order scheme
  - Backward Euler scheme

$$p_1(t) = \text{constant} = f(t_{n+1}, \mathbf{y}_{n+1})$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot f(t_{n+1}, \mathbf{y}_{n+1})$$

Adams-Moulton Scheme

- Second-order scheme
  - A two-step algorithm
  - An *implicit* scheme (vs. *explicit* schemes)

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + \int_{t_n}^{t_{n+1}} p(t,y)dt$$

$$p_2(t) = f_n + \frac{f_{n+1} - f_n}{\Delta t}(t - t_n)$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \int_{t_n}^{t_{n+1}} \left(f_n + \frac{f_{n+1} - f_n}{\Delta t}(t - t_n) \right) dt$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{2}\left[f(t_{n+1}, \boxed{\mathbf{y}_{n+1}}) + f(t_n, \mathbf{y}_n)\right]$$

Predictor-Corrector Method

- Second-order implicit scheme

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{2}\left[f(t_{n+1}, \mathbf{y}_{n+1}) + f(t_n, \mathbf{y}_n)\right]$$

$$\text{predictor (Adams-Bashforth): } \mathbf{y}_{n+1}^P = \mathbf{y}_n + \frac{\Delta t}{2}\left[3f_n - f_{n-1}\right]$$

$$\text{corrector (Adams-Moulton): } \mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{2}\left[f(t_{n+1}, \mathbf{y}_{n+1}^P) + f(t_n, \mathbf{y}_n)\right]$$

Higher Order Differential Equations

- Higher order ODE $y^{(n)} = f(t, y, y', y'', y''', \cdots, y^{(n-1)})$
  - Reduce to a first order system

$$(y_1', y_2' \cdots, y_{n-1}', y_n') = (y_2, y_3, \cdots, y_n, f(t, y_1, \cdots y_n))$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} y \\ y' \\ y'' \\ y''' \\ \vdots \\ y^{(n-2)} \\ y^{(n-1)} \end{pmatrix} = \begin{pmatrix} y \\ y'_1 \\ y'_2 \\ y'_3 \\ \vdots \\ y'_{n-2} \\ y'_{n-1} \end{pmatrix} \qquad \begin{pmatrix} y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \\ \vdots \\ y'_{n-1} \\ y'_n \end{pmatrix} = \begin{pmatrix} y_2 \\ y_3 \\ y_4 \\ y_5 \\ \vdots \\ y_n \\ f(t, y_1, y_2, \ldots, y_n) \end{pmatrix}$$

Higher Order Differential Equations

- Third-order ODE

$$\frac{d^3u}{dt^3} + u^2\frac{du}{dt} + \cos t \cdot u = g(t)$$

$$y_1 = u \qquad y_2 = \frac{du}{dt} \qquad y_3 = \frac{d^2u}{dt^2}$$

$$\frac{dy_1}{dt} = y_2$$

$$\frac{dy_2}{dt} = y_3$$

$$\frac{dy_3}{dt} = \frac{d^3u}{dt^3} = -u^2\frac{du}{dt} - \cos t \cdot u + g(t) = -y_1^2 y_2 - \cos t \cdot y_1 + g(t)$$

$$\frac{d\mathbf{y}}{dt} = \frac{d}{dt}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} y_2 \\ y_3 \\ -y_1^2 y_2 - \cos t \cdot y_1 + g(t) \end{pmatrix} = f(\mathbf{y}, t)$$

Outline

- Overview
- Ordinary differential equations (ODE)
  - Initial value problem (初值问题)
    - Euler, Runge-Kutta, Adams methods
  - Error analysis (误差分析)
    - Discretization error, Round-off, Stability
  - Boundary value problem (边值问题)
    - Shooting method, direct solve, relaxation
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

J. Nathan Kutz, Data-Driven Modeling & Scientific Computation, Methods for Complex Systems & Big Data, Oxford, 2013

Error Analysis

- Initial value problem → time-stepping routines
- Accuracy and stability
- Accuracy $\mathbf{y}(t+\Delta t) = \mathbf{y}(t) + \Delta t \cdot \frac{d\mathbf{y}(t)}{dt} + \frac{\Delta t^2}{2} \cdot \frac{d^2\mathbf{y}(c)}{dt^2}$
  - Truncation error $O(\Delta t^2)$ (local error)

$$\frac{d\boldsymbol{y}}{dt} = f(\boldsymbol{y}, t)$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot f(t_n, \mathbf{y}_n) + O(\Delta t^2)$$

  - Cumulative error (global error)

Discretization Error

- Local discretization error $\epsilon_{k+1} = \mathbf{y}(t_{k+1}) - (\mathbf{y}(t_k) + \Delta t \cdot \phi)$
- Global discretization error $E_k = \mathbf{y}(t_k) - \mathbf{y}_k$
- Euler method Δt, $t \in [a, b], K\ steps$

$$\text{local:} \quad \epsilon_k = \frac{\Delta t^2}{2} \frac{d^2\mathbf{y}(c_k)}{dt^2} \sim O(\Delta t^2)$$

$$\text{global:} \quad E_k = \sum_{j=1}^{K} \frac{\Delta t^2}{2} \frac{d^2\mathbf{y}(c_j)}{dt^2} \approx \frac{\Delta t^2}{2} \frac{d^2\mathbf{y}(\mathbf{c})}{dt^2} \cdot K$$

$$= \frac{\Delta t^2}{2} \frac{d^2\mathbf{y}(\mathbf{c})}{dt^2} \cdot \frac{b-a}{\Delta t} = \frac{b-a}{2} \Delta t \cdot \frac{d^2\mathbf{y}(\mathbf{c})}{dt^2} \sim O(\Delta t)$$

$$\mathbf{y}(t + \Delta t) = \mathbf{y}(t) + \Delta t \cdot \frac{d\mathbf{y}(t)}{dt} + \frac{\Delta t^2}{2} \cdot \frac{d^2\mathbf{y}(c)}{dt^2}$$

Discretization Error

- Accuracy of time-stepping methods

scheme	local error ϵ_k	global error E_k
Euler	$O(\Delta t^2)$	$O(\Delta t)$
2nd order Runge-Kutta	$O(\Delta t^3)$	$O(\Delta t^2)$
4th order Runge-Kutta	$O(\Delta t^5)$	$O(\Delta t^4)$
2nd order Adams-Bashforth	$O(\Delta t^3)$	$O(\Delta t^2)$

- Conclusion: higher accuracy is easily achieved by taking smaller time steps Δt
  - True or False?

Round-off Error

- Round-off error in numerical computations
  - Double precision allows for 16-digit accuracy
- Euler approximation $\frac{d\mathbf{y}}{dt} \approx \frac{\mathbf{y}_{n+1} - \mathbf{y}_n}{\Delta t} + \epsilon(\mathbf{y}_n, \Delta t)$
  - Truncation error $\epsilon(\mathbf{y}_n, \Delta t)$
  - Round-off error $\mathbf{y}_{n+1} = \mathbf{Y}_{n+1} + \mathbf{e}_{n+1}$

Y_{n+1} Value in Computer

$$\frac{d\mathbf{y}}{dt} = \frac{\mathbf{Y}_{n+1} - \mathbf{Y}_n}{\Delta t} + E_n(\mathbf{y}_n, \Delta t)$$

$$E_n = E_{\text{round}} + E_{\text{trunc}} = \frac{\mathbf{e}_{n+1} - \mathbf{e}_n}{\Delta t} - \frac{\Delta t^2}{2} \frac{d^2\mathbf{y}(c)}{dt^2}$$

Round-off Error

- Total error

$$E_n = E_{\text{round}} + E_{\text{trunc}} = \frac{\mathbf{e}_{n+1} - \mathbf{e}_n}{\Delta t} - \frac{\Delta t^2}{2}\frac{d^2\mathbf{y}(c)}{dt^2}$$

$$|\mathbf{e}_{n+1}| \le e_r \qquad M = \max_{c\in[t_n, t_{n+1}]}\left\{\left|\frac{d^2\mathbf{y}(c)}{dt^2}\right|\right\}$$

$$|-\mathbf{e}_n| \le e_r$$

$$|E_n| \le \frac{e_r + e_r}{\Delta t} + \frac{\Delta t^2}{2}M = \frac{2e_r}{\Delta t} + \frac{\Delta t^2 M}{2}$$

$$\frac{\partial |E_n|}{\partial(\Delta t)} = -\frac{2e_r}{\Delta t^2} + M\Delta t = 0$$

$$\Delta t = \left(\frac{2e_r}{M}\right)^{1/3}$$

Step-size

- Step-size in a minimum error

$$\Delta t = \left(\frac{2e_r}{M} \right)^{1/3}$$

- The smallest step-size is not necessarily the most accurate
  - Balance between round-off error and truncation error

Stability

- A stable scheme
  - The numerical solutions do not blow up to infinity
- Example $\frac{dy}{dt} = \lambda y$
  - Analytic solution $y(0) = y_0$
  - Forward Euler method $y(t) = y_0 \exp(\lambda t)$

$$y_{n+1} = y_n + \Delta t \cdot \lambda y_n = (1 + \lambda \Delta t) y_n$$

$$y_N = (1 + \lambda \Delta t)^N y_0$$

$$y_N = (1 + \lambda \Delta t)^N (y_0 + e)$$

$$E = (1 + \lambda \Delta t)^N e$$

Stability

- Forward Euler method $$y_N = (1+\lambda\Delta t)^N y_0$$
  - (1) $\lambda > 0$ $\quad y_N \to \infty$ $$E = (1+\lambda\Delta t)^N e$$
  - (2) $\lambda < 0$ $\quad y_N \to 0$

$$\text{I:} \quad |1+\lambda\Delta t| < 1 \ \text{ then } \ E \to 0$$

$$\text{II:} \quad |1+\lambda\Delta t| > 1 \ \text{ then } \ E \to \infty$$

  - Case I stable
  - Case II unstable if $\Delta t > -2/\lambda$

Stability

- A general theory of stability for one-step time-stepping scheme

$$\mathbf{y}_{n+1} = \mathbf{A}\mathbf{y}_n$$
$$\mathbf{y}_N = \mathbf{A}^N\mathbf{y}_0$$
$$\mathbf{A}^N = \mathbf{S}^{-1}\mathbf{\Lambda}^N\mathbf{S}$$

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_M \end{pmatrix} \rightarrow \mathbf{\Lambda}^N = \begin{pmatrix} \lambda_1^N & 0 & \cdots & 0 \\ 0 & \lambda_2^N & 0 & \cdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_M^N \end{pmatrix}$$

- Instability if $\Re\{\lambda_i\} > 1 \text{ for } i = 1, 2, ..., M$
- Extension: two-step scheme $\mathbf{y}_{n+1} = \mathbf{A}\mathbf{y}_n + \mathbf{B}\mathbf{y}_{n-1}$

Stability

- Forward Euler method $\quad E = (1 + \lambda\Delta t)^N e$
- Backward Euler method

$$y_{n+1} = y_n + \Delta t \cdot \lambda y_{n+1}$$

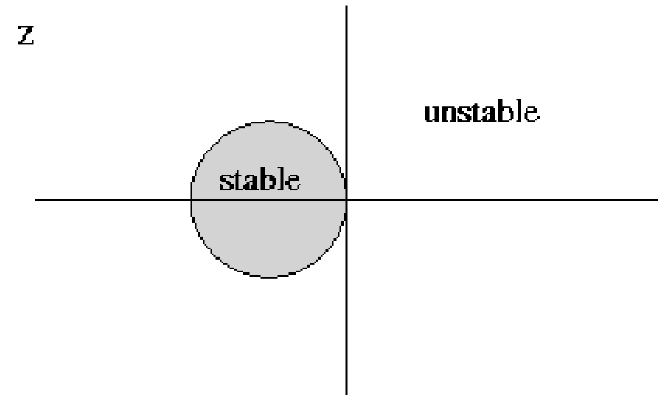
$$y_N = \left(\frac{1}{1 - \lambda\Delta t}\right)^N y_0$$

$$E = \left(\frac{1}{1 - \lambda\Delta t}\right)^N e$$

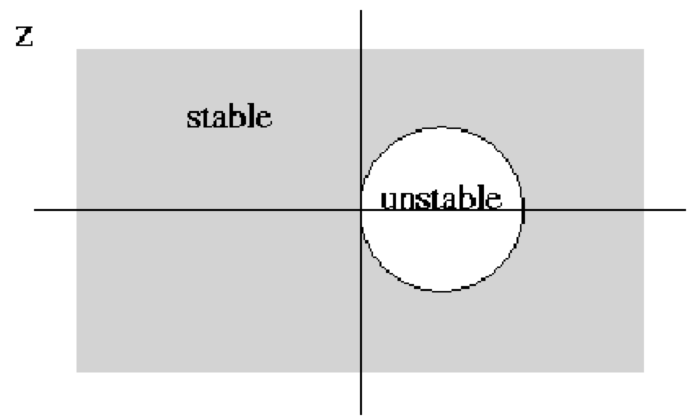
forward Euler: $\quad |1 + z| > 1$

backward Euler: $\quad \left|\frac{1}{1 - z}\right| > 1$

$$z = \lambda\Delta t$$



forward Euler: |1+z|>1



backward Euler: |1/(1-z)|>1

Error Analysis

- Accuracy
  - Truncation error
    - Local discretization error
    - Global discretization error
  - Round-off error
- Stability
- Accuracy vs. Stability

Outline

- Overview
- Ordinary differential equations (ODE)
  - Initial value problem (初值问题)
    - Euler, Runge-Kutta, Adams methods
  - Error analysis (误差分析)
    - Discretization error, Round-off, Stability
  - Boundary value problem (边值问题)
    - Shooting method, direct solve, relaxation
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

Boundary Value Problem

- Initial value problem

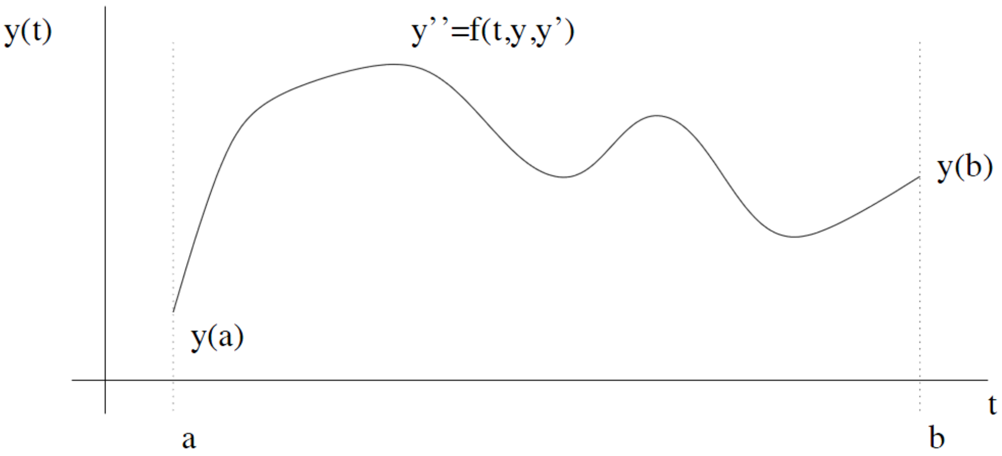
$$\frac{dy}{dt} = \lambda y \qquad y(0) = y_0$$

- Boundary value problem

$$\frac{d^2y}{dt^2} = f\left(t, y, \frac{dy}{dt}\right) \qquad t \in [a, b]$$

$$\alpha_1 y(a) + \beta_1 \frac{dy(a)}{dt} = \gamma_1$$

$$\alpha_2 y(b) + \beta_2 \frac{dy(b)}{dt} = \gamma_2$$



The Shooting Method

- Initial value problem
  - Present time (t = a)
  - Time-stepping schemes
- Boundary value problem
  - Present time (t = a) and future time (t = b)

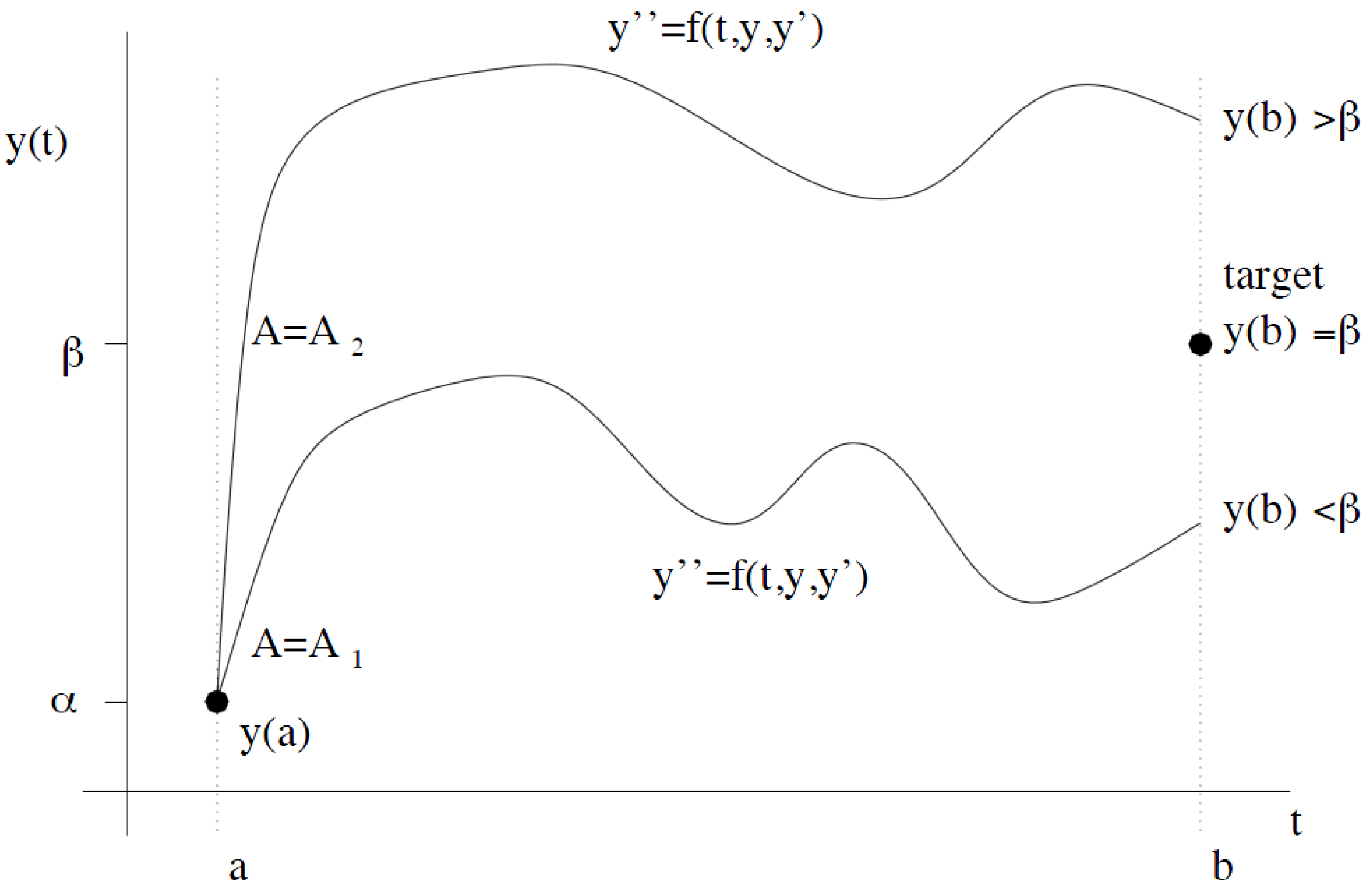
$$\frac{d^2y}{dt^2} = f\left(t, y, \frac{dy}{dt}\right) \quad t \in [a, b] \qquad \begin{array}{l} y(a) = \alpha \\ y(b) = \beta \end{array}$$

  - Two initial conditions $y(a)$ and $y'(a)$
  - Choose the initial conditions $y(a) = \alpha$

$$\frac{dy(a)}{dt} = A$$

The Shooting Method

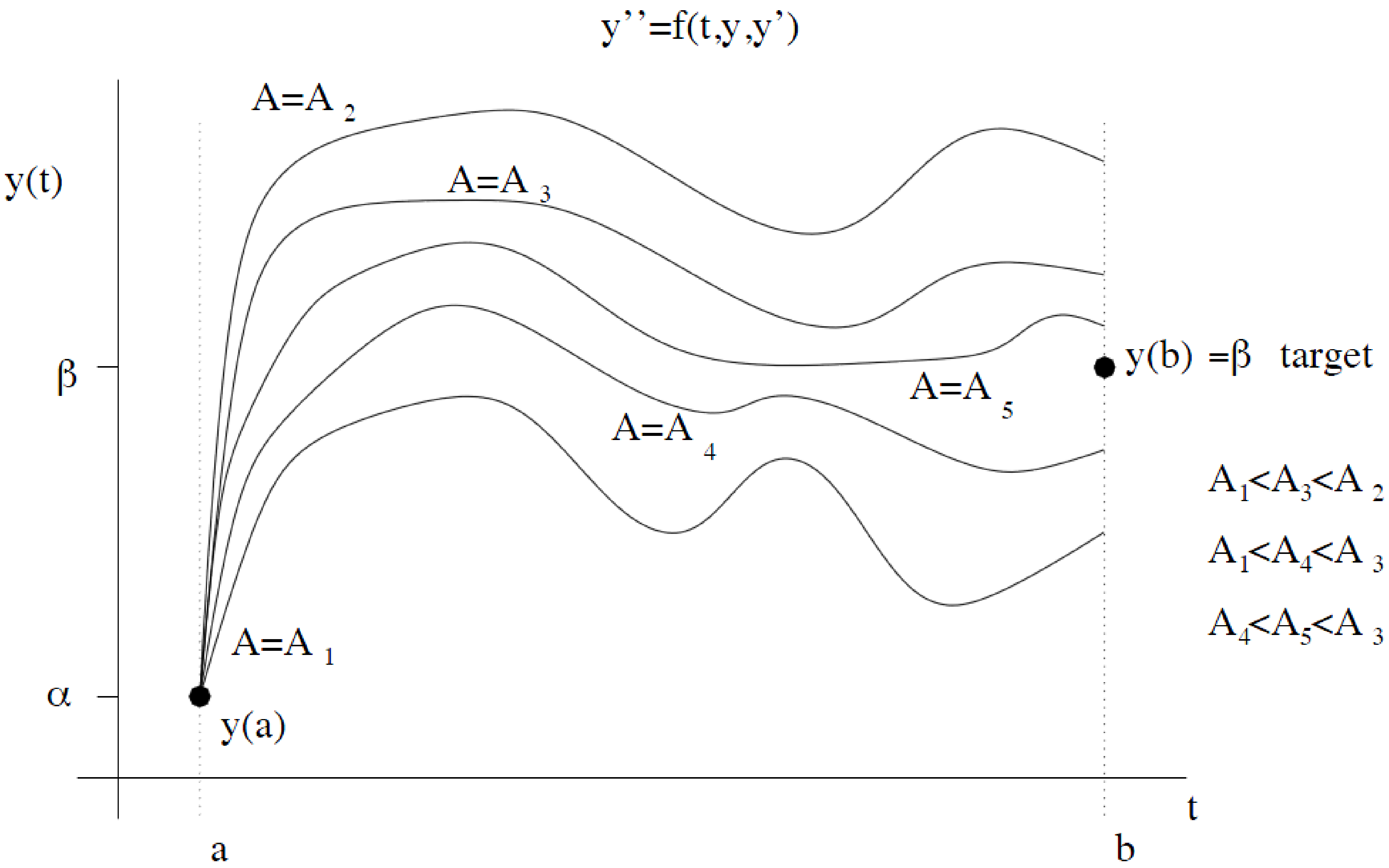
$$\frac{d^2y}{dt^2} = f\left(t, y, \frac{dy}{dt}\right) \quad t \in [a, b] \quad \begin{array}{l} y(a) = \alpha \\ y(b) = \beta \end{array} \quad \begin{array}{l} y(a) = \alpha \\ \dfrac{dy(a)}{dt} = A \end{array}$$



The Shooting Method

- Search for the appropriate value A
  - Solve ODE using a time-stepping scheme with the initial conditions $y(a) = \alpha$ and $y'(a) = A$
  - Evaluate the solution $y(b)$ at $t = b$ and compare this value with the target value of $y(b) = \beta$
  - Adjust the value of A (either bigger or smaller) using a bisection method until a desired level of tolerance and accuracy is achieved.

The Shooting Method



Boundary Value Problem

- The shooting method
  - Iterative scheme
- The direct method $\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot \phi$
  - Taylor expanding ODE
  - Directly solve $y(t_0), y(t_1), y(t_2) \ldots y(t_{N-1}), y(t_N)$
  - Linear problems → Ax = b
  - Nonlinear problems → A relaxation scheme
    - Newton or Secant methods

The Direct Method

- Linear boundary value problem

$$\frac{d^2y}{dt^2} = p(t)\frac{dy}{dt} + q(t)y + r(t) \quad t \in [a,b] \quad \begin{matrix} y(a) = \alpha \\ y(b) = \beta \end{matrix}$$

- Taylor expansions

$$f(t+\Delta t) = f(t) + \Delta t\frac{df(t)}{dt} + \frac{\Delta t^2}{2!}\frac{d^2f(t)}{dt^2} + \frac{\Delta t^3}{3!}\frac{d^3f(c_1)}{dt^3}$$

$$f(t-\Delta t) = f(t) - \Delta t\frac{df(t)}{dt} + \frac{\Delta t^2}{2!}\frac{d^2f(t)}{dt^2} - \frac{\Delta t^3}{3!}\frac{d^3f(c_2)}{dt^3}$$

$$f(t+\Delta t) - f(t-\Delta t) = 2\Delta t\frac{df(t)}{dt} + \frac{\Delta t^3}{3!}\left(\frac{d^3f(c_1)}{dt^3} + \frac{d^3f(c_2)}{dt^3}\right)$$

$$\frac{df(t)}{dt} = \frac{f(t+\Delta t) - f(t-\Delta t)}{2\Delta t} - \frac{\Delta t^2}{6}\frac{d^3f(c)}{dt^3}$$

$$f'''(c) = (f'''(c_1) + f'''(c_2))/2$$ Mean-value theorem

The Direct Method

$O(\Delta t^2)$ center-difference schemes

$$
\begin{aligned}
f'(t) &= [f(t+\Delta t) - f(t-\Delta t)]/2\Delta t \\
f''(t) &= [f(t+\Delta t) - 2f(t) + f(t-\Delta t)]/\Delta t^2 \\
f'''(t) &= [f(t+2\Delta t) - 2f(t+\Delta t) + 2f(t-\Delta t) - f(t-2\Delta t)]/2\Delta t^3 \\
f''''(t) &= [f(t+2\Delta t) - 4f(t+\Delta t) + 6f(t) - 4f(t-\Delta t) + f(t-2\Delta t)]/\Delta t^4
\end{aligned}
$$

$O(\Delta t^2)$ forward- and backward-difference schemes

$$
\begin{aligned}
f'(t) &= [-3f(t) + 4f(t+\Delta t) - f(t+2\Delta t)]/2\Delta t \\
f'(t) &= [3f(t) - 4f(t-\Delta t) + f(t-2\Delta t)]/2\Delta t \\
f''(t) &= [2f(t) - 5f(t+\Delta t) + 4f(t+2\Delta t) - f(t+3\Delta t)]/\Delta t^3 \\
f''(t) &= [2f(t) - 5f(t-\Delta t) + 4f(t-2\Delta t) - f(t-3\Delta t)]/\Delta t^3
\end{aligned}
$$

The Direct Method

$O(\Delta t^4)$ center-difference schemes

$$
\begin{aligned}
f'(t) &= [-f(t+2\Delta t) + 8f(t+\Delta t) - 8f(t-\Delta t) + f(t-2\Delta t)]/12\Delta t \\
f''(t) &= [-f(t+2\Delta t) + 16f(t+\Delta t) - 30f(t) \\
&\qquad +16f(t-\Delta t) - f(t-2\Delta t)]/12\Delta t^2 \\
f'''(t) &= [-f(t+3\Delta t) + 8f(t+2\Delta t) - 13f(t+\Delta t) \\
&\qquad +13f(t-\Delta t) - 8f(t-2\Delta t) + f(t-3\Delta t)]/8\Delta t^3 \\
f''''(t) &= [-f(t+3\Delta t) + 12f(t+2\Delta t) - 39f(t+\Delta t) + 56f(t) \\
&\qquad -39f(t-\Delta t) + 12f(t-2\Delta t) - f(t-3\Delta t)]/6\Delta t^4
\end{aligned}
$$

The Direct Method

- Linear boundary value problem

$$\frac{d^2y}{dt^2} = p(t)\frac{dy}{dt} + q(t)y + r(t) \quad t \in [a,b] \quad \begin{matrix} y(a) = \alpha \\ y(b) = \beta \end{matrix}$$

- Centre-difference schemes
- Discretize [*a*, *b*] into *N+1* steps (*N-1* Unknows)

$$\frac{y(t+\Delta t) - 2y(t) + y(t-\Delta t)}{\Delta t^2} = p(t)\frac{y(t+\Delta t) - y(t-\Delta t)}{2\Delta t} + q(t)y(t) + r(t)$$

$$\left[1 - \frac{\Delta t}{2}p(t)\right] y(t+\Delta t) - \left[2 + \Delta t^2 q(t)\right] y(t) + \left[1 + \frac{\Delta t}{2}\right] y(t-\Delta t) = \Delta t^2 r(t)$$

$$y(t_0) = y(a) = \alpha$$

$$y(t_N) = y(b) = \beta$$

The Direct Method

- Ax = b $\left[1-\frac{\Delta t}{2}p(t)\right]y(t+\Delta t)-\left[2+\Delta t^2 q(t)\right]y(t)+\left[1+\frac{\Delta t}{2}\right]y(t-\Delta t)=\Delta t^2 r(t)$

$$\mathbf{A}=\begin{bmatrix} 2+\Delta t^2 q(t_1) & -1+\frac{\Delta t}{2}p(t_1) & 0 & \cdots & & 0 \\ -1-\frac{\Delta t}{2}p(t_2) & 2+\Delta t^2 q(t_2) & -1+\frac{\Delta t}{2}p(t_2) & 0 & \cdots & \vdots \\ 0 & \ddots & \ddots & \ddots & & \\ \vdots & & & & & \vdots \\ & & & & & 0 \\ \vdots & & & \ddots & \ddots & -1+\frac{\Delta t}{2}p(t_{N-2}) \\ 0 & \cdots & & 0 & -1-\frac{\Delta t}{2}p(t_{N-1}) & 2+\Delta t^2 q(t_{N-1}) \end{bmatrix}$$

$$\mathbf{x}=\begin{bmatrix} y(t_1) \\ y(t_2) \\ \vdots \\ y(t_{N-2}) \\ y(t_{N-1}) \end{bmatrix} \qquad \mathbf{b}=\begin{bmatrix} -\Delta t^2 r(t_1)+(1+\Delta t p(t_1)/2)y(t_0) \\ -\Delta t^2 r(t_2) \\ \vdots \\ -\Delta t^2 r(t_{N-2}) \\ -\Delta t^2 r(t_{N-1})+(1-\Delta t p(t_{N-1})/2)y(t_N) \end{bmatrix}$$

Nonlinear Problems

- Linear problems → Ax = b
- Nonlinear problems → A relaxation scheme

$$y'' = f(t, y, y') \rightarrow \frac{y(t+\Delta t) - 2y(t) + y(t-\Delta t)}{\Delta t^2} = f\left(t, y(t), \frac{y(t+\Delta t) - y(t-\Delta t)}{2\Delta t}\right)$$

- Discretize [*a*, *b*] into *N*+*1* steps (*N*-*1* Unknows)

$$y(t_0) = y(a) = \alpha \text{ and } y(t_N) = y(b) = \beta$$

$$2y_1 - y_2 - \alpha + \Delta t^2 f(t_1, y_1, (y_2 - \alpha)/2\Delta t) = 0$$

$$-y_1 + 2y_2 - y_3 + \Delta t^2 f(t_2, y_2, (y_3 - y_1)/2\Delta t) = 0$$

$$\vdots$$

$$-y_{N-3} + 2y_{N-2} - y_{N-1} + \Delta t^2 f(t_{N-2}, y_{N-2}, (y_{N-1} - y_{N-3})/2\Delta t) = 0$$

$$-y_{N-2} + 2y_{N-1} - \beta + \Delta t^2 f(t_{N-1}, y_{N-1}, (\beta - y_{N-2})/2\Delta t) = 0.$$

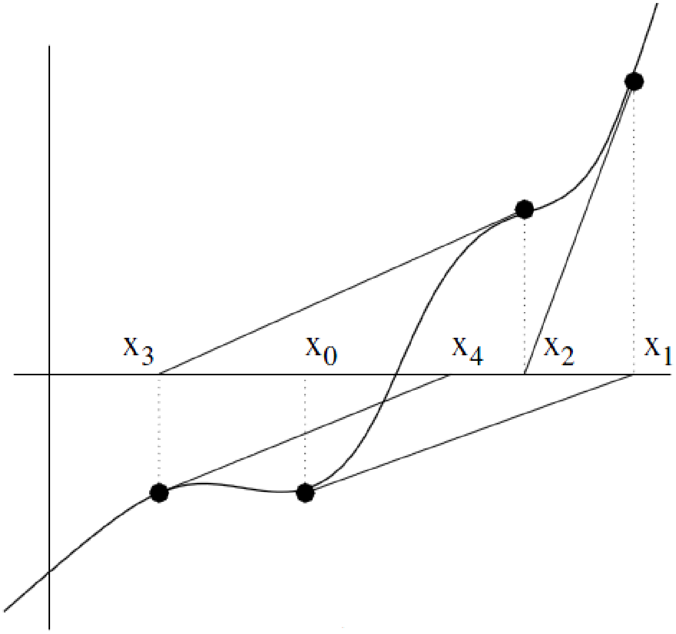
Nonlinear Problems

- (N-1)x(N-1) nonlinear system of equations
  - No guarantees about the existence or uniqueness of solutions
  - The best approach is to use a relaxation scheme based on Newton or Secant method iterations

$$
\begin{aligned}
2y_1 - y_2 - \alpha + \Delta t^2 f(t_1, y_1, (y_2 - \alpha)/2\Delta t) &= 0 \\
-y_1 + 2y_2 - y_3 + \Delta t^2 f(t_2, y_2, (y_3 - y_1)/2\Delta t) &= 0 \\
&\vdots \\
-y_{N-3} + 2y_{N-2} - y_{N-1} + \Delta t^2 f(t_{N-2}, y_{N-2}, (y_{N-1} - y_{N-3})/2\Delta t) &= 0 \\
-y_{N-2} + 2y_{N-1} - \beta + \Delta t^2 f(t_{N-1}, y_{N-1}, (\beta - y_{N-2})/2\Delta t) &= 0.
\end{aligned}
$$

Newton's Method

- A single nonlinear equation $f(x_r) = 0$
  - x_r the root to the equation
- Newton's method (Newton-Raphson method)
  - An iterative scheme based on an initial guess x_0



Newton’s Method

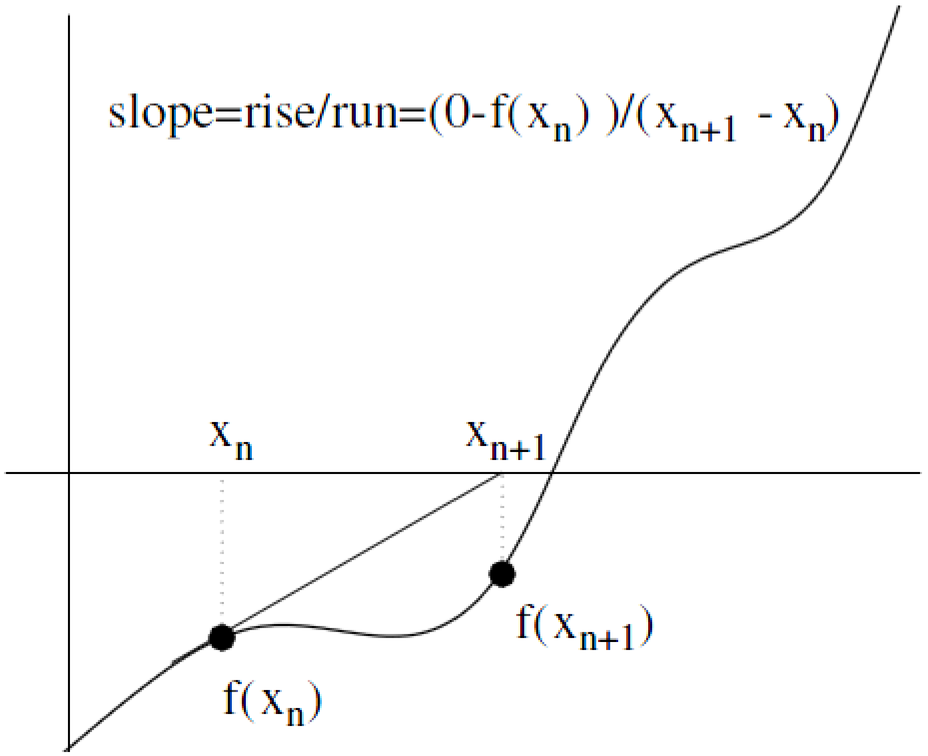
- An iterative scheme based on an initial guess x_0

$$\text{slope} = \frac{df(x_n)}{dx} = \frac{\text{rise}}{\text{run}} = \frac{0 - f(x_n)}{x_{n+1} - x_n}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- Fails if $f'(x_n) = 0$
- For certain guesses,

iterations may diverge



Newton's Method

- Nonlinear problems

$$2y_1 - y_2 - \alpha + \Delta t^2 f(t_1, y_1, (y_2 - \alpha)/2\Delta t) = 0$$

$$-y_1 + 2y_2 - y_3 + \Delta t^2 f(t_2, y_2, (y_3 - y_1)/2\Delta t) = 0$$

$$\vdots$$

$$-y_{N-3} + 2y_{N-2} - y_{N-1} + \Delta t^2 f(t_{N-2}, y_{N-2}, (y_{N-1} - y_{N-3})/2\Delta t) = 0$$

$$-y_{N-2} + 2y_{N-1} - \beta + \Delta t^2 f(t_{N-1}, y_{N-1}, (\beta - y_{N-2})/2\Delta t) = 0.$$

$$\mathbf{F}(\mathbf{x_n}) = \begin{bmatrix} f_1(x_1, x_2, x_3, \ldots, x_N) \\ f_2(x_1, x_2, x_3, \ldots, x_N) \\ \vdots \\ f_N(x_1, x_2, x_3, \ldots, x_N) \end{bmatrix} = \mathbf{0}$$

Newton's Method

- Iteration scheme $$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$\mathbf{x_{n+1}} = \mathbf{x_n} + \mathbf{\Delta x_n}$$

$$\mathbf{J}(\mathbf{x_n})\mathbf{\Delta x_n} = -\mathbf{F}(\mathbf{x_n})$$

$$\mathbf{J}(\mathbf{x_n}) = \begin{bmatrix} f_{1_{x_1}} & f_{1_{x_2}} & \cdots & f_{1_{x_N}} \\ f_{2_{x_1}} & f_{2_{x_2}} & \cdots & f_{2_{x_N}} \\ \vdots & \vdots & & \vdots \\ f_{N_{x_1}} & f_{N_{x_2}} & \cdots & f_{N_{x_N}} \end{bmatrix}$$ Jacobian Matrix

- Initially guessing values for $x_1, x_2, \ldots, x_N$
- Fails if det $\mathbf{J}(x_n) = 0$
- No guarantee that the algorithm will converge

Outline

- Overview
- Ordinary differential equations (ODE)
  - Initial value problem (初值问题)
    - Euler, Runge-Kutta, Adams methods
  - Error analysis (误差分析)
    - Discretization error, Round-off, Stability
  - Boundary value problem (边值问题)
    - Shooting method, direct solve, relaxation
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

J. Nathan Kutz, Data-Driven Modeling & Scientific Computation, Methods for Complex Systems & Big Data, Oxford, 2013

Outline

- Overview
- Ordinary differential equations (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
    - Basic time and space stepping schemes
    - Examples
      - Poisson equations
      - Heat diffusion equations
      - Wave equations
      - Advection-diffusion equation
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

Finite Difference Methods

- Based on Taylor expansions
  - Discretization in time and space

$$f(t+\Delta t) = f(t) + \Delta t \frac{df(t)}{dt} + \frac{\Delta t^2}{2!}\frac{d^2 f(t)}{dt^2} + \frac{\Delta t^3}{3!}\frac{d^3 f(c_1)}{dt^3}$$
$$f(t-\Delta t) = f(t) - \Delta t \frac{df(t)}{dt} + \frac{\Delta t^2}{2!}\frac{d^2 f(t)}{dt^2} - \frac{\Delta t^3}{3!}\frac{d^3 f(c_2)}{dt^3}$$

- Advantages
  - Easy to implement
  - Handle fairly complicated boundary conditions
  - Explicit calculations of the computational error

Finite Difference Methods

- Basic steps
  - Discretize in space and time
  - Solve a large linear system of equations or manipulate large, sparse matrices
- Accuracy and stability
  - Accuracy from the space-time discretization
  - Numerical stability as the solution is propagated in time

Poisson Equation

- Poisson equation $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$
  - $f = 0$时称为Laplace equation或调和方程

$$f^* = \arg\min_f \iint_\Omega \underbrace{\|\nabla f - \mathbf{v}\|^2}_{F} \quad \text{s.t } f^*|_{\partial\Omega} = f|_{\partial\Omega}$$

Problem Reduction

Euler Equation: $F_f - \frac{\partial}{\partial x} F_{f_x} - \frac{\partial}{\partial y} F_{f_y} = 0$

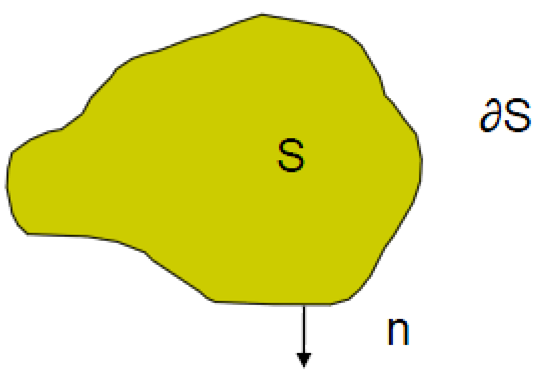
$$\boxed{\Delta f = div(\mathbf{v}) \ \text{ s.t } f^*|_{\partial\Omega} = f|_{\partial\Omega}}$$

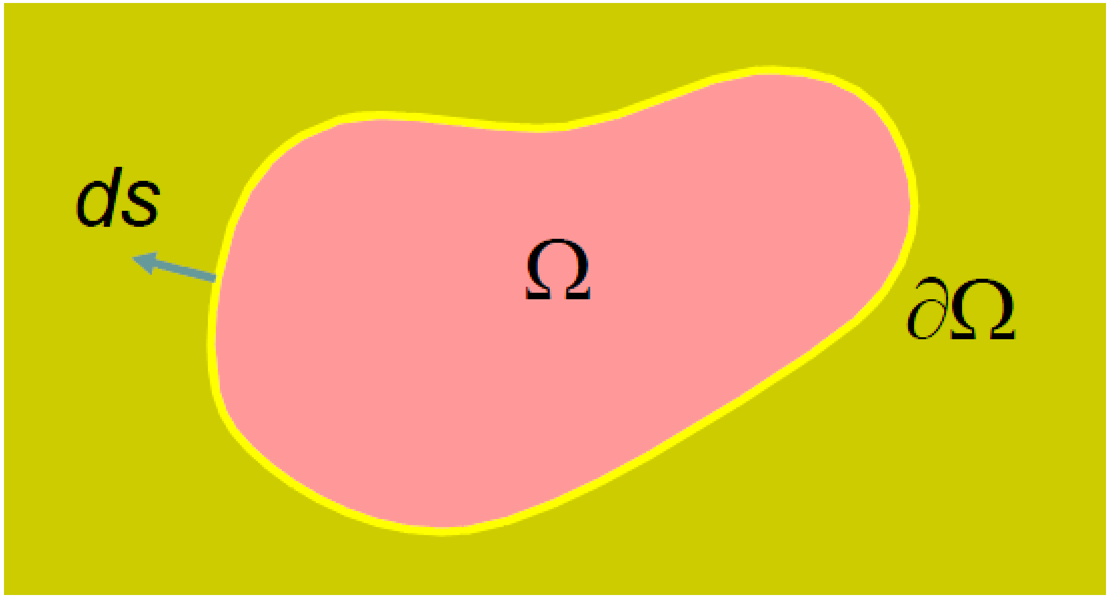
$\mathbf{V}$ is a guidance field, needs not to be a gradient field.

Q: How does this relate to $x^* = argmin_x \|Ax - b\|^2$?

定解问题 - Boundary Conditions

- Dirichlet (狄利克雷)条件：给出u在∂S上的值
- Neumann条件：给出$\frac{\partial u}{\partial n}$在$\partial S$上的值
- Robin条件：给出$au + b\frac{\partial u}{\partial n}$在$\partial S$上的值





泊松方程的导出

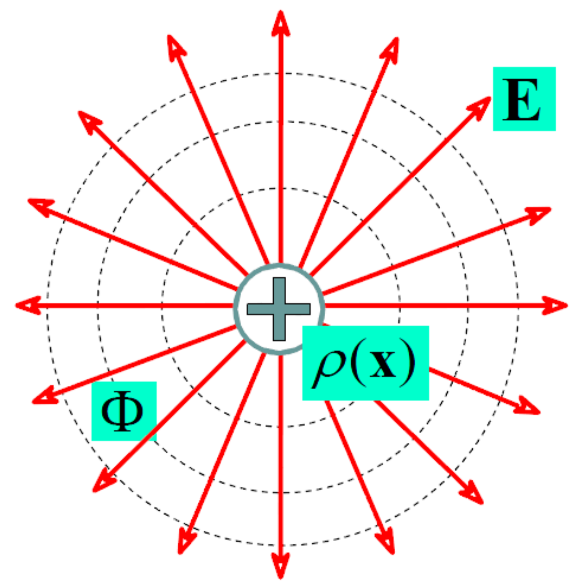
- 静电势 $$\Delta\Phi = -\frac{\rho(\mathbf{x})}{\varepsilon_0}$$

$$\mathbf{F} = \frac{q_1 q_2 \mathbf{r}}{4\pi\varepsilon_0 r^3}$$

- 引力势 $$\Delta\Phi = -4\pi G\rho(\mathbf{x})$$

- 电荷密度$\rho(x)$
- 电势 Φ
- 电场 E

$$\mathbf{E} = -\nabla\Phi$$



泊松方程的导出

Gauss's Law:

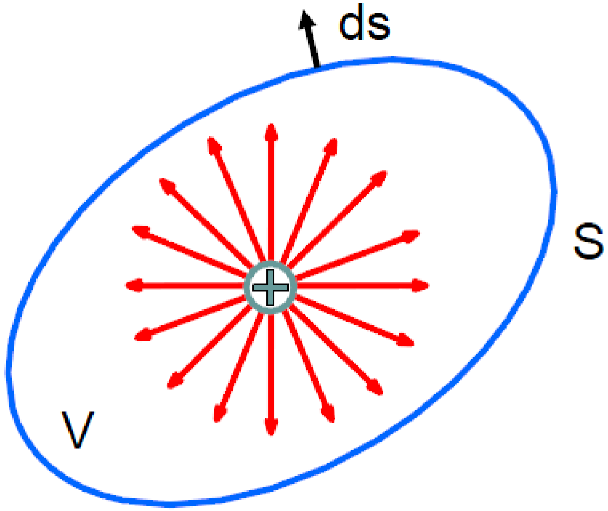
$$\oint_S \mathbf{E} \cdot d\mathbf{s} = \int_V \boxed{\frac{\rho(\mathbf{x})}{\varepsilon_0}} dv$$

Gauss's theorem:

$$\oint_S \mathbf{E} \cdot d\mathbf{s} = \int_V \boxed{\nabla \cdot \mathbf{E}} dv$$

$$\boxed{\nabla \cdot \mathbf{E}} = \boxed{\frac{\rho(\mathbf{x})}{\varepsilon_0}}$$

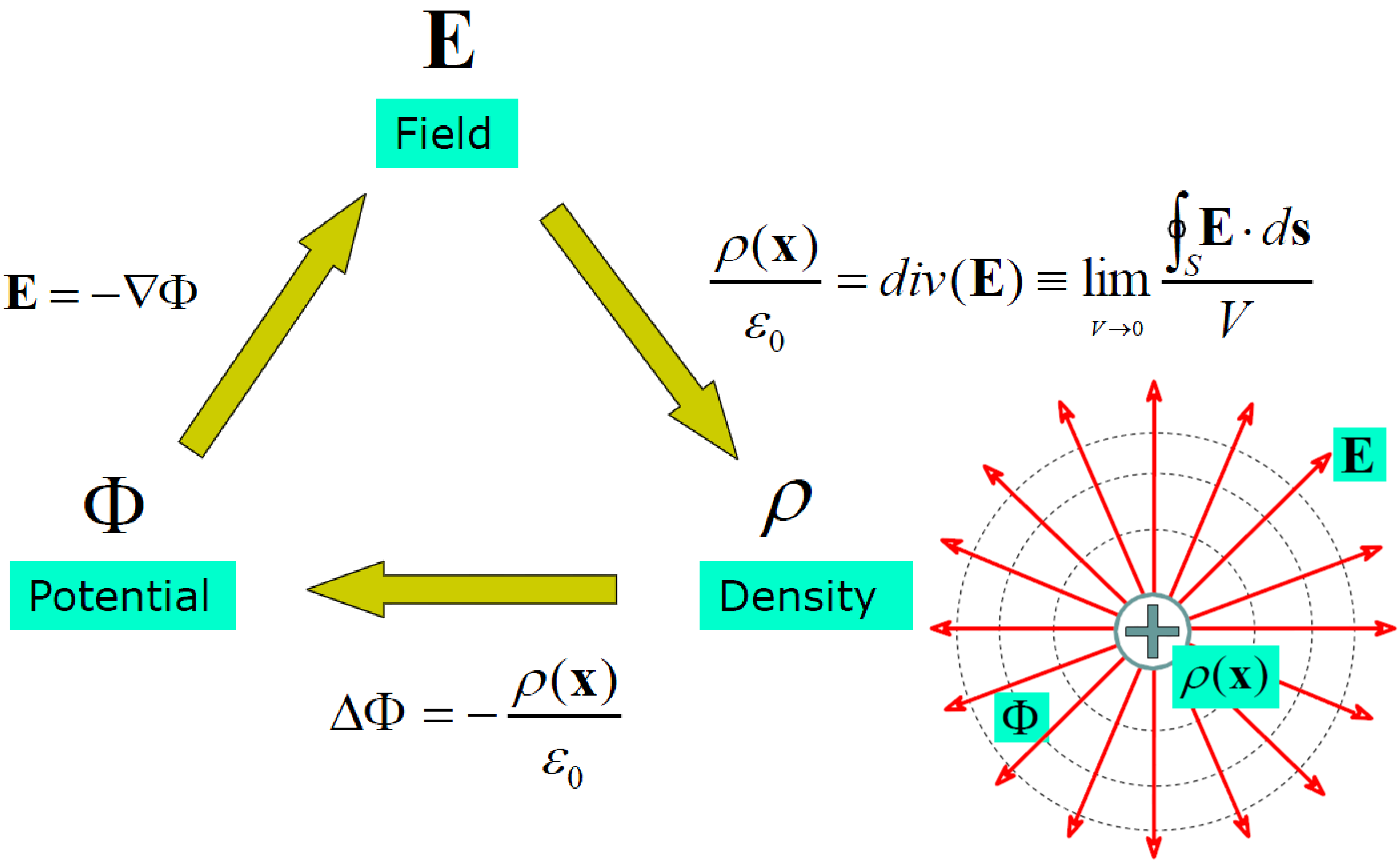
$$\mathbf{E} = -\nabla\Phi$$



$$\boxed{\Delta\Phi = -\frac{\rho(\mathbf{x})}{\varepsilon_0}}$$

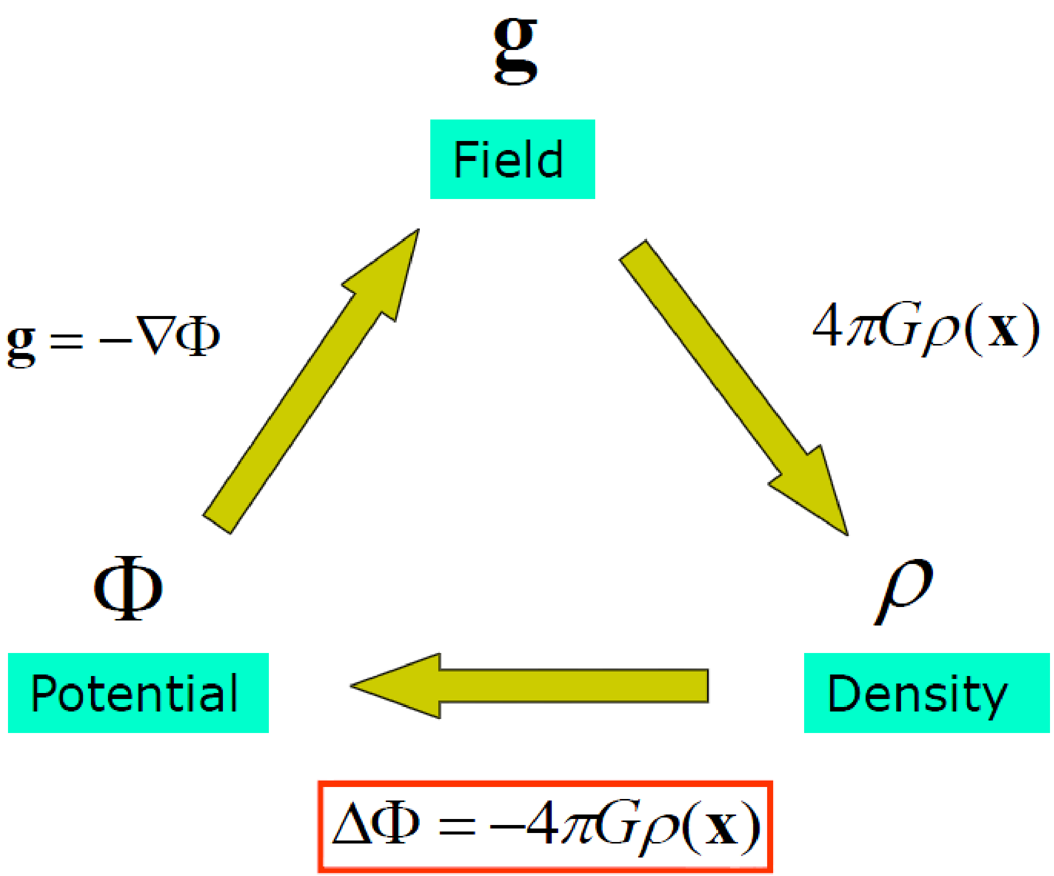
Poisson Equation

密度、势能与场关系

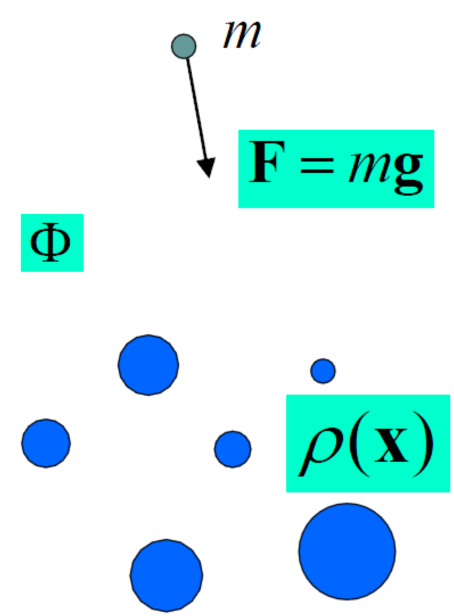


$\mathbf{E}$
Field
$\mathbf{E} = -\nabla\Phi$
$\frac{\rho(\mathbf{x})}{\varepsilon_0} = div(\mathbf{E}) \equiv \lim_{V \to 0} \frac{\oint_S \mathbf{E} \cdot d\mathbf{s}}{V}$
Φ
Potential
ρ
Density
$\Delta\Phi = -\frac{\rho(\mathbf{x})}{\varepsilon_0}$
$\mathbf{E}$
$\rho(\mathbf{x})$
Φ

密度、势能与场关系



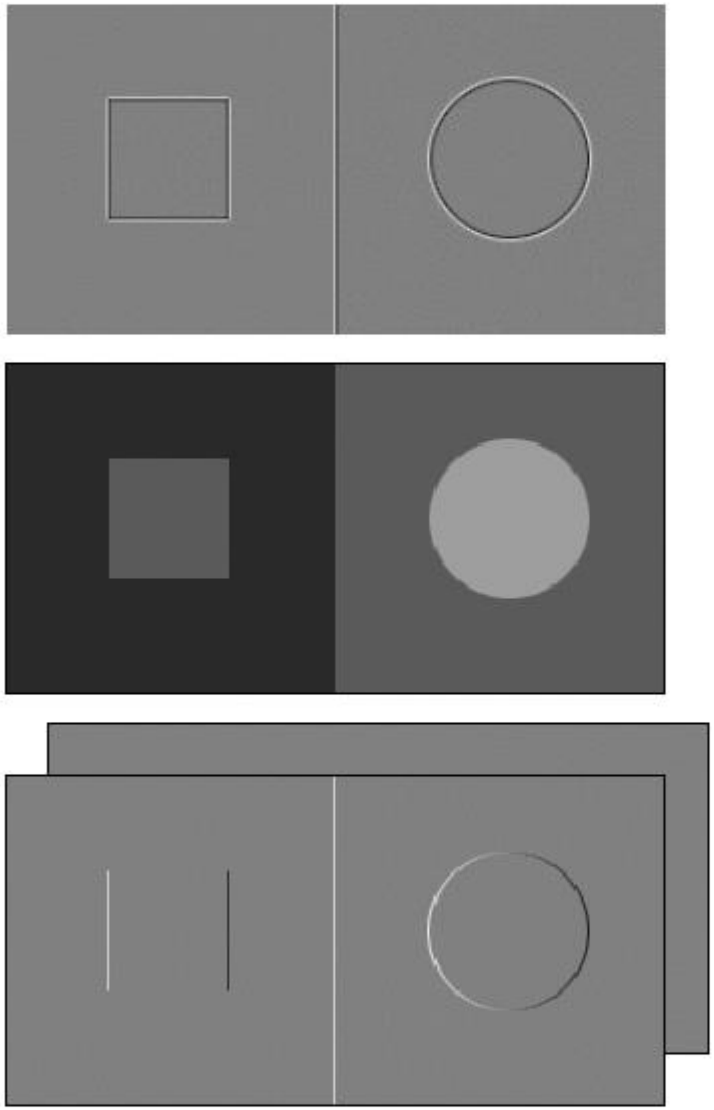
$$4\pi G\rho(\mathbf{x}) = div(\mathbf{g}) \equiv \lim_{V\to 0} \frac{\oint_S \mathbf{g}\cdot d\mathbf{s}}{V}$$



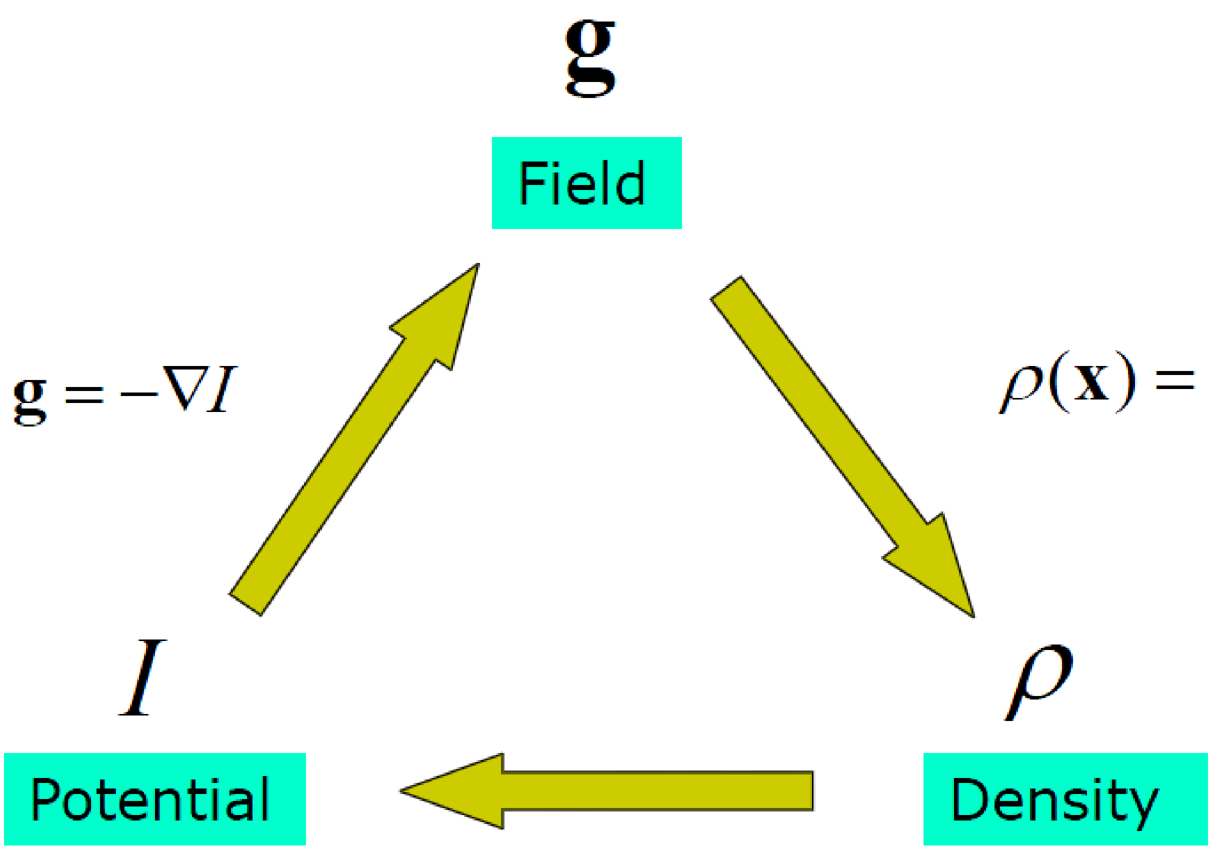
泊松方程在图像处理中的应用

- 图像密度 $\rho(x)$
- 图像(势能) I
- 图像梯度 g

- $g = -\nabla I$



泊松方程在图像处理中的应用

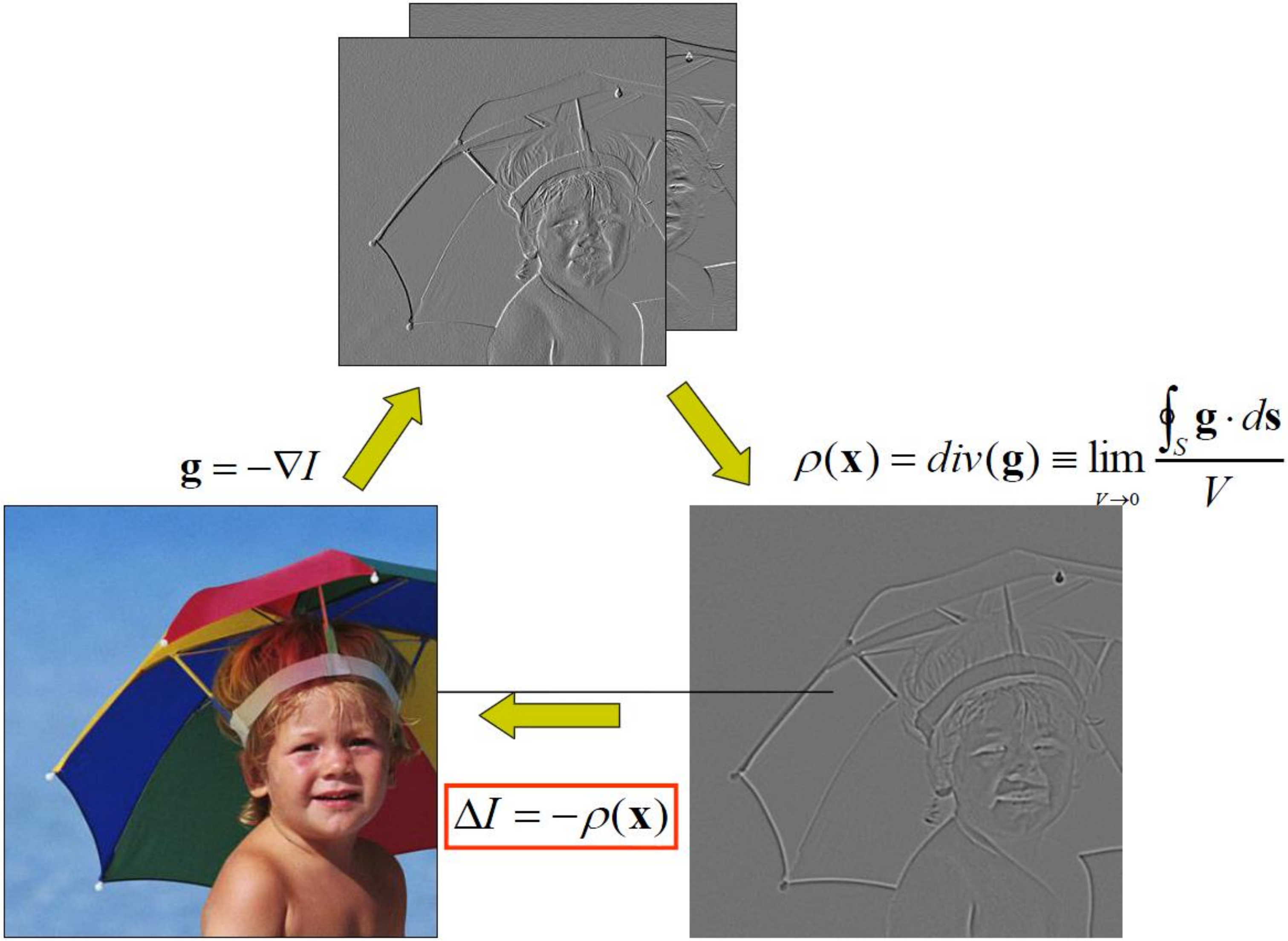


$$\mathbf{g} = -\nabla I$$

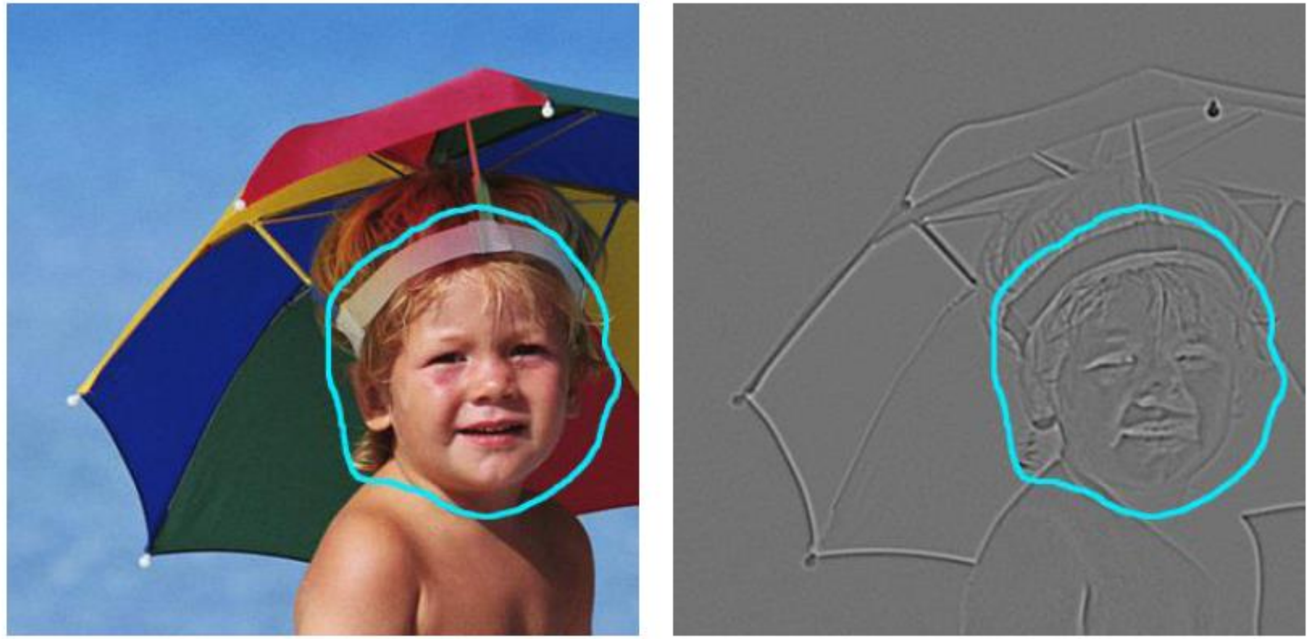
$$\rho(\mathbf{x}) = div(\mathbf{g}) \equiv \lim_{V \to 0} \frac{\oint_S \mathbf{g} \cdot d\mathbf{s}}{V}$$

$$\boxed{\Delta I = -\rho(\mathbf{x})}$$

泊松方程在图像处理中的应用



泊松方程在图像处理中的应用



$$\boxed{\Delta I = -\rho(\mathbf{x})}$$

泊松方程在图像处理中的应用

$$\boxed{\Delta I = div(\nabla I_A) \qquad \text{s.t.} \quad I\,|_{\partial\Omega} = I_B\,|_{\partial\Omega}}$$

I_A $\qquad\qquad$ I_B

Poisson Equation

- Poisson equation $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = -\rho$
  - Elliptic equation
- Finite difference method
  - Discretize the spatial derivative with second-order central difference

$$\Delta f = -\rho$$

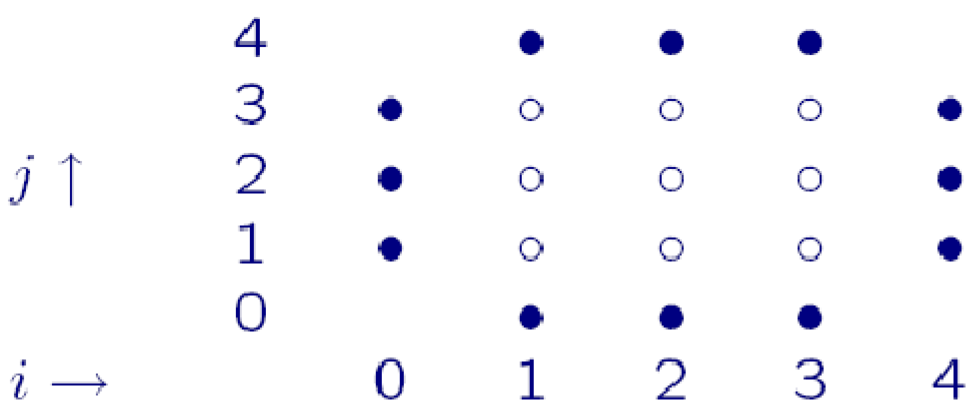
$$\Downarrow$$

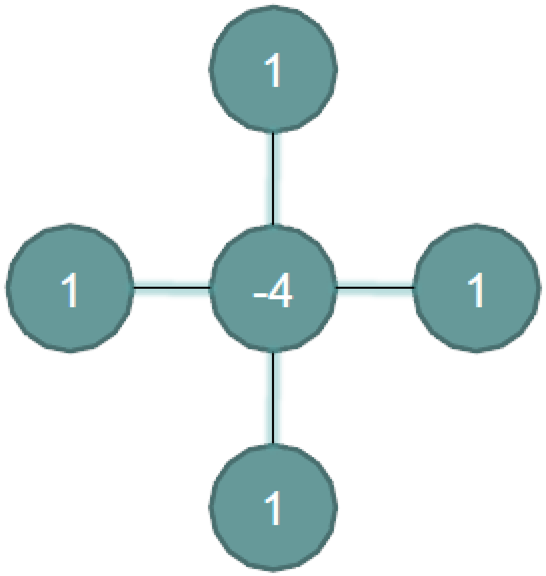
$$\frac{f_{i+1,j} + f_{i-1,j} - 2f_{i,j}}{h^2} + \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{h^2} = \rho_{i,j}$$

Poisson Equation

- Discrete Poisson equation

$$\frac{f_{i+1,j}+f_{i-1,j}-2f_{i,j}}{h^2}+\frac{f_{i,j+1}+f_{i,j-1}-2f_{i,j}}{h^2}=\rho_{i,j}$$

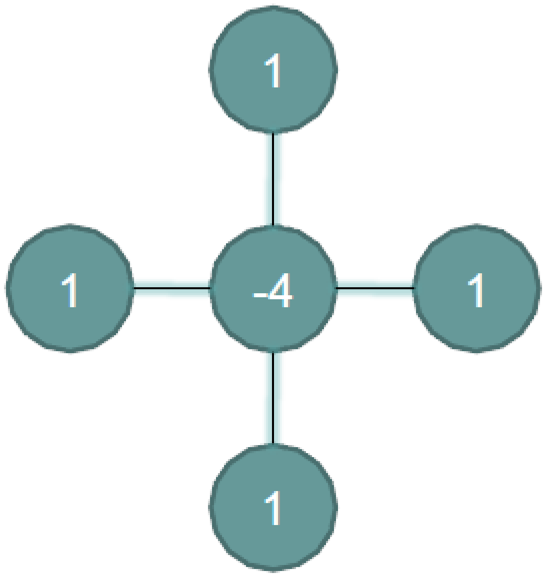




Poisson Equation

- Boundary conditions in **b**

$$
\begin{bmatrix}
-4 & 1 & & & & & & & \\
1 & -4 & 1 & & & & & & \\
 & 1 & -4 & 1 & & & & & \\
 & & 1 & -4 & 1 & & & & \\
0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\
 & & & & 1 & -4 & 1 & & \\
 & & & & & 1 & -4 & 1 & \\
 & & & & & & 1 & -4 & 1 \\
 & & & & & & & 1 & -4
\end{bmatrix}
\begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix}
=
\begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{12} \\ b_{22} \\ b_{32} \\ b_{13} \\ b_{23} \\ b_{33} \end{bmatrix}
$$



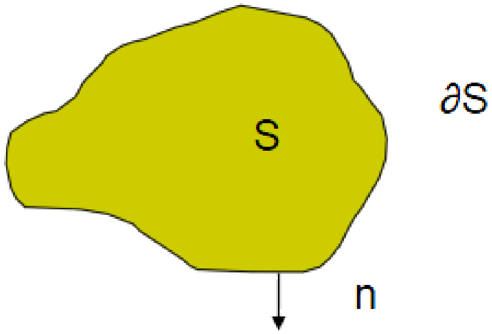
$$\mathbf{Ax = b}$$

5-point method

Big Idea: Derivatives : Functions = Matrices : Vectors

Poisson Equation

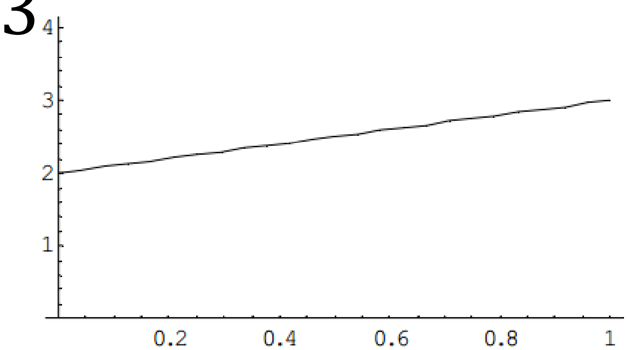
- Poisson equation $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$
  - $f = 0$时称为Laplace equation或调和方程
- Boundary conditions
  - Dirichlet(狄利克雷)条件：给出u在∂S上的值
  - Neumann条件：给出$\frac{\partial u}{\partial n}$在$\partial S$上的值
  - Robin条件：给出$au + b\frac{\partial u}{\partial n}$在$\partial S$上的值
  - One condition for each node on the boundary ∂S



Poisson Equation

Poisson equation	Dirichlet boundary	Periodical boundary	Neumann boundary	Robin boundary
$Ax = \mathrm{b}$ $f(x), x \in [a,b]$	$f(a) = \alpha$ $f(b) = \beta$	$f(a) = f(b)$	$f'(a) = \alpha$ $f'(b) = \beta$	$f(a) = \alpha$ $f'(b) = \beta$
定解问题	唯一解	无数解	无数解或无解	唯一解或无解

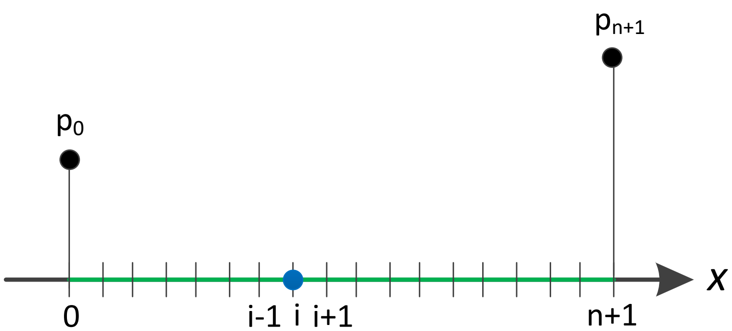
- 定解至少需要一个Dirichlet条件，或仅需要$f'(x)$
- Laplace equation $p_{xx} = 0 \rightarrow p = ax + b, x \in [0,1]$
  - Dirichlet条件 $p(0) = 2, p(1) = 3$
  - Robin条件 $p(0) = 2, p'(0) = 1$
  - Neumann条件
    - $p'(0) = 1, p'(1) = 2$ 无解
    - $p'(0) = 1, p'(1) = 1$ 无数解→Compatibility Condition



Poisson Equation

- $p_{xx} = f \rightarrow p_{i+1} - 2p_i + p_{i-1} = \Delta x^2 f_i$
- Dirichlet boundary p_0 and p_{n+1}
  - A sparse symmetric, negative definite matrix

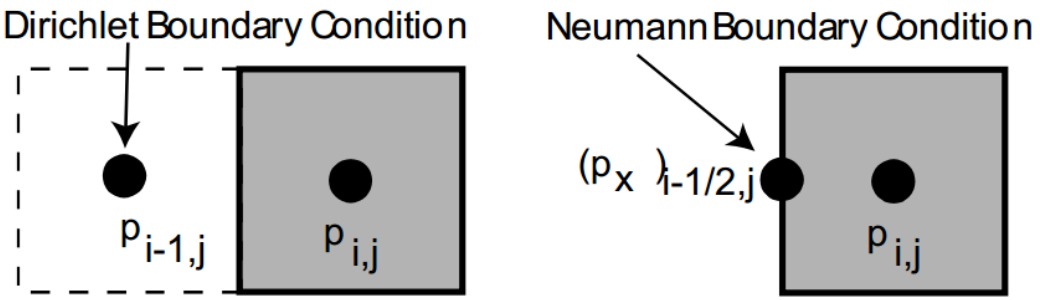
$$1\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}1 \cdot \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix} = \begin{pmatrix} \Delta x^2 f_1 \\ \Delta x^2 f_2 \\ \vdots \\ \Delta x^2 f_{n-1} \\ \Delta x^2 f_n \end{pmatrix}$$



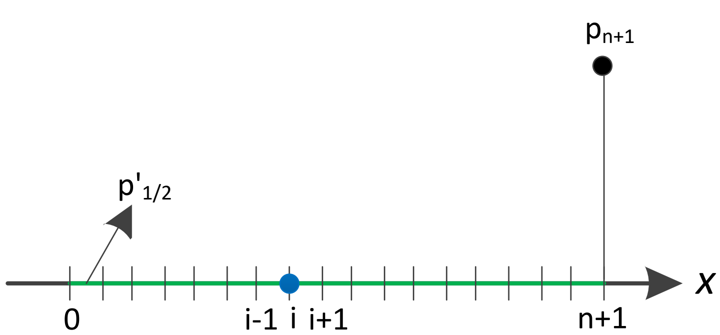
$$\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix} = \begin{pmatrix} \Delta x^2 f_1 - p_0 \\ \Delta x^2 f_2 \\ \vdots \\ \Delta x^2 f_{n-1} \\ \Delta x^2 f_n - p_{n+1} \end{pmatrix}$$

Poisson Equation

- $p_{xx} = f \rightarrow p_{i+1} - 2p_i + p_{i-1} = \Delta x^2 f_i$
- Robin boundary $p'_{1/2}$ and p_{n+1}
  - $\dfrac{\frac{p_2-p_1}{\Delta x} - \frac{p_1-p_0}{\Delta x}}{\Delta x} = f_1$
  - $\dfrac{p_1-p_0}{\Delta x} = p'_{1/2}$
  - $p_2 - p_1 = \Delta x^2 f_1 + \Delta x p'_{1/2}$



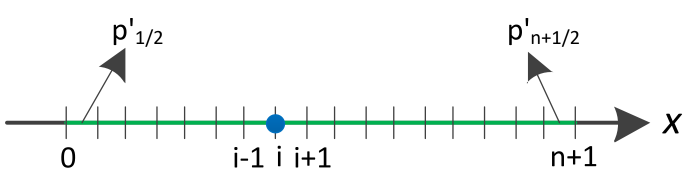
$$\begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix} = \begin{pmatrix} \Delta x^2 f_1 + \Delta x\, p'_{1/2} \\ \Delta x^2 f_2 \\ \vdots \\ \Delta x^2 f_{n-1} \\ \Delta x^2 f_n - p_{n+1} \end{pmatrix}$$



Poisson Equation

- Neumann boundary
  - $A \in R^{n\times n}$ has a null space, $A \cdot (1\ 1\ \cdots\ 1)^T = 0$
  - If p^* is a solution of the PDE, for any vector z on the null space of A ($Az = 0$), $p^* + \alpha z$ is also a solution: $A(p^* +$

$$\begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix} = \begin{pmatrix} \Delta x^2 f_1 + \Delta x\, p'_{1/2} \\ \Delta x^2 f_2 \\ \vdots \\ \Delta x^2 f_{n-1} \\ \Delta x^2 f_n + \Delta x\, p'_{n+1/2} \end{pmatrix}$$

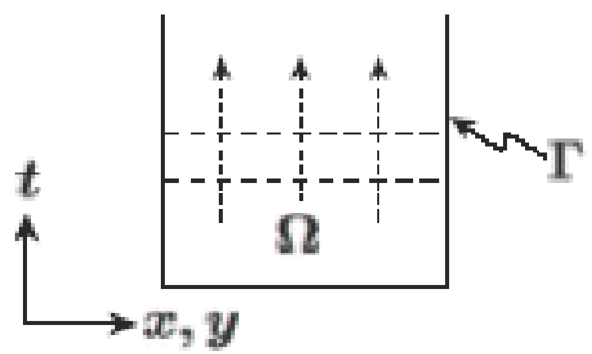


Outline

- Overview
- Ordinary differential equations (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
    - Basic time and space stepping schemes
    - Examples
      - Poisson equations
      - Heat diffusion equations
      - Wave equations
      - Advection-diffusion equation
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

热传导方程的导出

- 在三维空间中，求物体Ω的内部温度分布，设在t时刻(x,y,z)处的温度为$u(x,y,z,t)$，根据热力学中Fourier定律：**热流向量q与温度梯度成正比**：
$$q=-k(x,y,z)\nabla u=-k(u_x,u_y,u_z)$$
- k是物体的热传导系数，负号表示热量从温度高向温度低流



热传导方程的导出

- 在dt时间内流过曲面元dS的热量为dQ，则

$$dQ = (\boldsymbol{q} \cdot \boldsymbol{n})dSdt = -k(\nabla u \cdot \boldsymbol{n})dSdt = -k\frac{\partial u}{\partial \boldsymbol{n}}dSdt$$

$\boldsymbol{n}$表示曲面元dS的法线方向

- 在物体Ω内任取一闭曲面Γ，它所包围的区域为G，则从时刻t_1到t_2流进该曲面的热量

$$Q = \int_{t_1}^{t_2} \oiint_{\Gamma} k\frac{\partial u}{\partial \boldsymbol{n}}dSdt$$

热传导方程的导出

- 同时流入的这些热量就是使物体内部的温度发生变化的热量

$$\iiint\limits_G c(x,y,z)\rho(x,y,z)[u(t_2)-u(t_1)]dxdydz$$

$c(x,y,z)$是比热，$\rho(x,y,z)$是质量

$$Q=\int_{t_1}^{t_2}\oiint\limits_{\Gamma} k(x,y,z)\frac{\partial u}{\partial n}dSdt=$$

$$\iiint\limits_G c(x,y,z)\rho(x,y,z)[u(x,y,z,t_2)-u(x,y,z,t_1)]dxdydz$$

热传导方程的导出

- 利用Gauss公式化简

$$\iiint_{\Omega}\left(\frac{\partial P}{\partial x}+\frac{\partial Q}{\partial y}+\frac{\partial R}{\partial z}\right)dv=\oiint_{\Sigma} P\,dydz+Q\,dzdx+R\,dxdy$$

$$Q=\int_{t_1}^{t_2}\boxed{\oiint_{\Gamma} k(x,y,z)\frac{\partial u}{\partial n}dS}dt=$$

$$\iiint_{G} c(x,y,z)\rho(x,y,z)\boxed{[u(x,y,z,t_2)-u(x,y,z,t_1)]}dxdydz$$

$$\int_{t_1}^{t_2}\boxed{\iiint_{G}\left[\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial u}{\partial z}\right)\right]dxdydz}$$

$$=\iiint_{G} c(x,y,z)\rho(x,y,z)\boxed{\left(\int_{t_1}^{t_2}\frac{\partial u}{\partial t}dt\right)}dxdydz$$

热传导方程的导出

- 由于t_1, t_2, G都是任意的，

$$\int_{t_1}^{t_2}\iiint_G \left[\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial u}{\partial z}\right)\right]dxdydz$$

$$=\iiint_G c(x,y,z)\rho(x,y,z)\left(\int_{t_1}^{t_2}\frac{\partial u}{\partial t}dt\right)dxdydz$$

$$c\rho u_t=(ku_x)_x+(ku_y)_y+(ku_z)_z$$

热传导方程的导出

- 如果物体是均匀的，c, ρ, k都是常数，记$\alpha^2 = \frac{k}{c\rho}$，则 $u_t = \alpha^2(u_{xx} + u_{yy} + u_{zz})$
- 通用表达式$u_t = \nabla(D \cdot \nabla u)$
  - D是张量，在三维上是3×3矩阵
  - D是各向同性张量，则为各向同性扩散方程
  - D是各向异性张量，则为各向异性扩散方程
    - 图像和信号处理大部分都是各向异性扩散方程应用

Heat Diffusion Equation

- 1D heat diffusion equation
  - Parabolic equation $u_t - k\Delta u = f$ $\quad \dfrac{\partial u}{\partial t} = \kappa \dfrac{\partial^2 u}{\partial x^2}$
  - Periodic boundary conditions $u(-L) = u(L)$
- Finite difference method
  - Discretize the spatial derivative with second-order central difference

$$\frac{\partial u}{\partial t} = \frac{\kappa}{\Delta x^2}\left[u(x+\Delta x) - 2u(x) + u(x-\Delta x)\right]$$

  - Reduce the PDE to a system of ODEs

Problem Reduction

Heat Diffusion Equation

- The ODE system

$$
\begin{aligned}
u(-L) &= u_1 \\
u(-L+\Delta x) &= u_2 \\
&\vdots \\
u(L-2\Delta x) &= u_{n-1} \\
u(L-\Delta x) &= u_n \\
u(L) &= u_{n+1}
\end{aligned}
$$

$$u_1 = u_{n+1}$$

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

$$\frac{\partial u}{\partial t} = \frac{\kappa}{\Delta x^2}\left[u(x+\Delta x) - 2u(x) + u(x-\Delta x)\right]$$

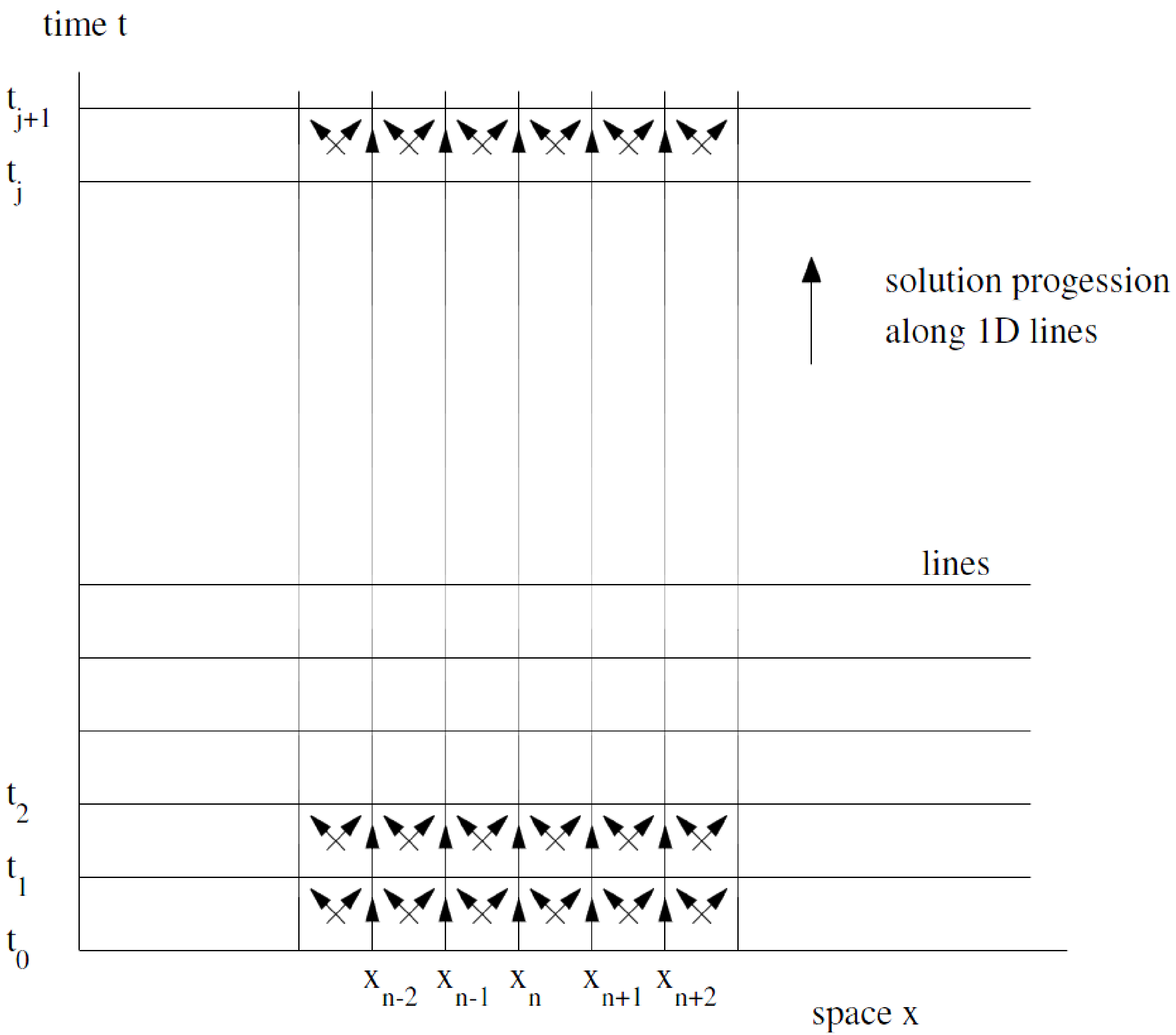
$$\frac{d\mathbf{u}}{dt} = \frac{\kappa}{\Delta x^2}\mathbf{A}\mathbf{u}$$

Heat Diffusion Equation

- The ODE system $\dfrac{d\mathbf{u}}{dt} = \dfrac{\kappa}{\Delta x^2}\mathbf{A}\mathbf{u}$
- Solve the ODE system using time-stepping ODE method, such as Euler, 4th Runge-Kutta

$$\mathbf{A} = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 & 1 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \\ \vdots & & & & & \vdots \\ & & & & & 0 \\ \vdots & \cdots & 0 & 1 & -2 & 1 \\ 1 & 0 & \cdots & 0 & 1 & -2 \end{bmatrix} \qquad \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

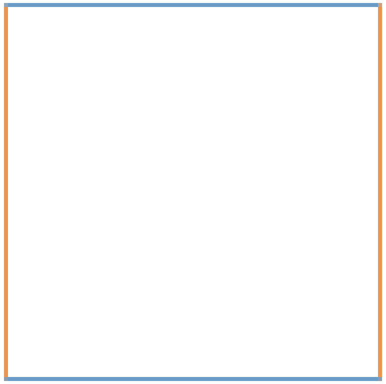
Heat Diffusion Equation



Heat Diffusion Equation

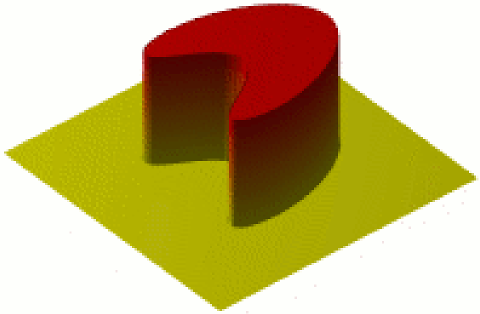
- 2D heat diffusion equation
  - Parabolic equation $u_t - k\Delta u = f$ $\quad \dfrac{\partial u}{\partial t} = \kappa \left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} \right)$
  - Periodic boundary conditions $[-L, L]$

$$u(-L, y) = u(L, y)$$
$$u(x, -L) = u(x, L)$$



- Discretize the spatial derivative with second-order central difference, $\Delta x = \Delta y = \delta$

$$\frac{d\mathbf{u}}{dt} = \frac{\kappa}{\delta^2}\mathbf{A}\mathbf{u}$$



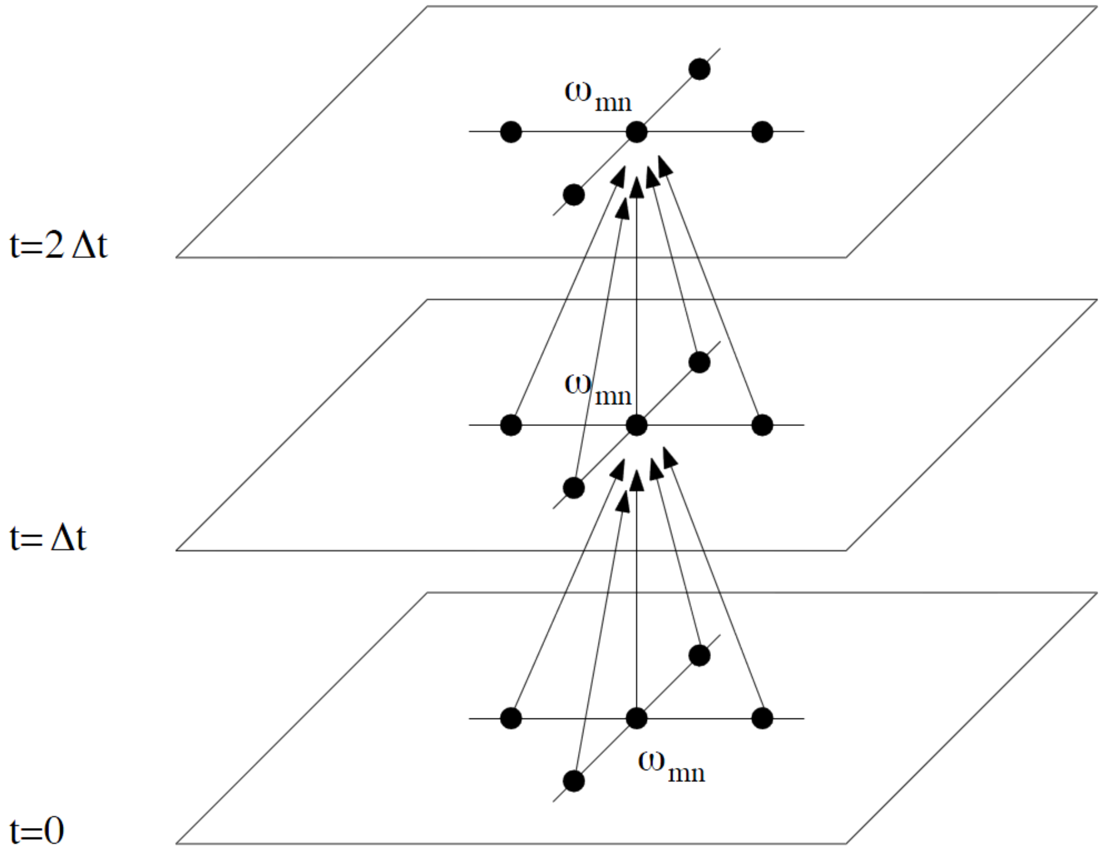
Heat Diffusion Equation

- 2D heat diffusion equation $\frac{\partial u}{\partial t} = \kappa \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$

$$\frac{d\mathbf{u}}{dt} = \frac{\kappa}{\delta^2}\mathbf{A}\mathbf{u}$$

$$\mathbf{u} = \begin{pmatrix} u_{11} \\ u_{12} \\ \vdots \\ u_{1n} \\ u_{21} \\ u_{22} \\ \vdots \\ u_{n(n-1)} \\ u_{nn} \end{pmatrix}$$

$$u_{jk} = u(x_j, y_k)$$



Method of Lines

- Basic procedure
  - Use the data at a single slice of time to generate a solution Δt in the future
  - Use the updated data to generate a solution $2\Delta t$ in the future
- Method of lines
  - Each line is the value of the solution at a given time slice
  - The lines are used to update the solution to a new timeline and progressively generate future solutions

Outline

- Overview
- Ordinary differential equation (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
    - Basic time and space stepping schemes
    - Examples
      - Poisson equations
      - Heat diffusion equations
      - Wave equations
      - Advection-diffusion equation
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

波动方程的导出

- 力学中的弦是指柔软的一维细线，根据Hook定律，其拉伸后所具有的位能，与其长度的增量成正比，比例系数τ称为张力

- 如果以$u(x,t)$表示时刻t时，位于x处的弦离平衡位置的值，横振动是指弦的运动方向垂直于弦的平衡位置，考虑时间t时微小的横振动

波动方程的导出

- 假设x处的斜率$\frac{\partial u(x,t)}{\partial t}$很小，假定ρ为线密度，在时刻$t$时的总动能是

$$T=\frac{\rho}{2}\int_0^l[\frac{\partial u(x,t)}{\partial t}]^2dx=\frac{\rho}{2}\int_0^l u_t^2dx$$

- 弦的伸长所具有的位能是

$$U_1=\tau\left[\int_0^l\sqrt{1+\left(\frac{\partial u}{\partial x}\right)^2}dx-l\right]$$

$$\approx\tau[\int_0^l(1+\frac{1}{2}u_x^2)dx-l]=\frac{\tau}{2}\int_0^l u_x^2dx$$

波动方程的导出

- 若弦上还有以线密度为$F(x,t)$，方向与u轴一致的外力左右，则相应的内能为

$$U_2 = -\int_0^l Fu\,dx$$

- 则

$$f = L = T - U_1 - U_2 = \frac{\rho u_t^2}{2} - \frac{\tau u_x^2}{2} + Fu$$

波动方程的导出

- 应用最小位能原理计算变分（Variational）：

$$J=\int_{t_0}^{t_1} L dt \quad (L=T-U) \qquad \delta J=0$$

$$f=L=T-U_1-U_2=\frac{\rho u_t^2}{2}-\frac{\tau u_x^2}{2}+Fu$$

Euler-Lagrange equations

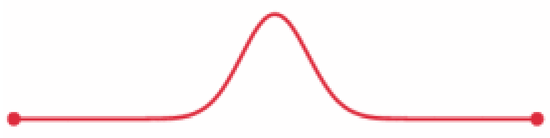
$$f_u-\frac{d}{dt}f_{u_t}-\frac{d}{dx}f_{u_x}-\frac{d}{dy}f_{u_y}-\frac{d}{dz}f_{u_z}=0$$

$$f_u=F, f_{u_t}=\rho u_t, f_{u_x}=-\tau u_x$$

- 欧拉方程 $\rho u_{tt}=\tau u_{xx}+F(x,t)$

Wave Equation

- 1D wave equation $\frac{\partial u}{\partial t} = c\frac{\partial u}{\partial x}$
  - Hyperbolic equation $u_{tt} - c^2 \Delta u = f$
  - $u_{tt} - c^2 \Delta u = 0$



$$\left[\frac{\partial}{\partial t} - c\frac{\partial}{\partial x}\right]\left[\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right] u = 0 \qquad \frac{\partial u}{\partial t} - c\frac{\partial u}{\partial x} = 0 \quad \text{and} \quad \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

  - Advection equations (left and right traveling)
- Finite difference method
  - Discretize by second-order central difference in the *x* direction

$$u_n = u(x_n, t)$$

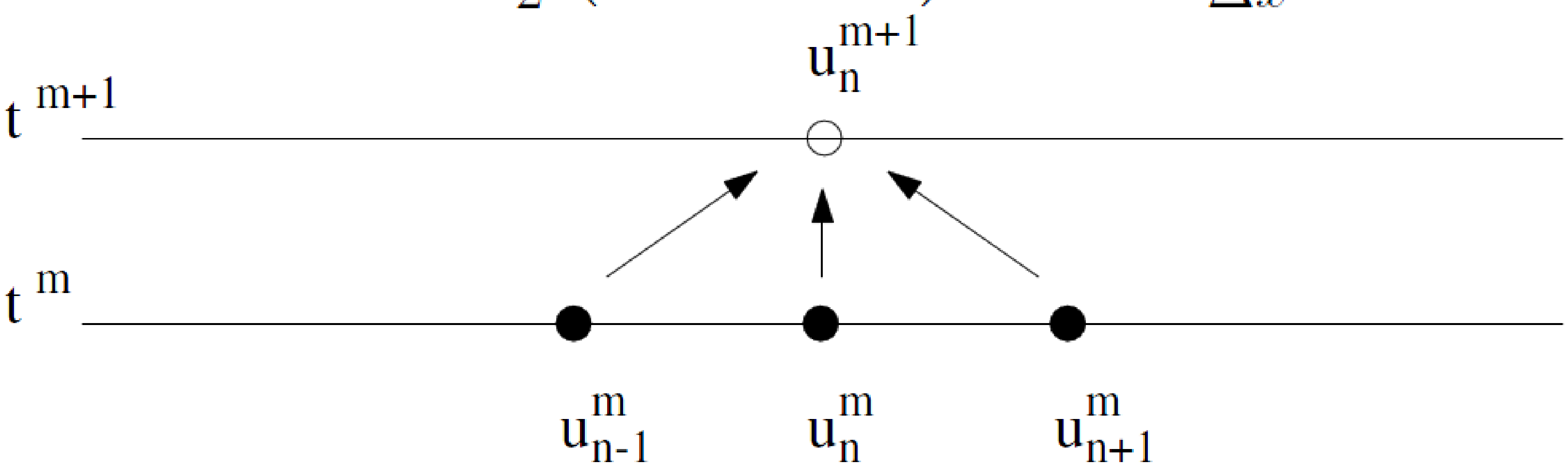
$$\frac{\partial u_n}{\partial t} = \frac{c}{2\Delta x}\left(u_{n+1} - u_{n-1}\right)$$

Wave Equation

- Finite difference method
  - Discretize by second-order central difference in *x*
  - Step forward with an Euler time-stepping method

$$u_n^{(m+1)} = u_n^{(m)} + \frac{c\Delta t}{2\Delta x}\left(u_{n+1}^{(m)} - u_{n-1}^{(m)}\right) \qquad u_n^{(m)} = u(x_n, t_m)$$

$$u_n^{(m+1)} = u_n^{(m)} + \frac{\lambda}{2}\left(u_{n+1}^{(m)} - u_{n-1}^{(m)}\right) \qquad \lambda = \frac{c\Delta t}{\Delta x}$$



Wave Equation

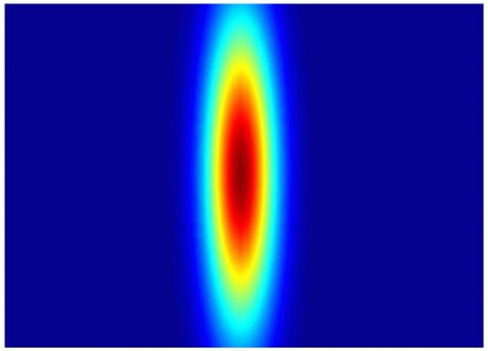
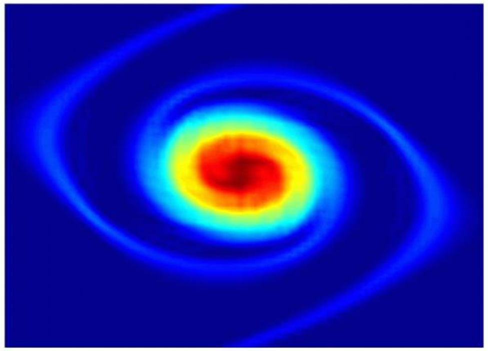
- $\lambda = \dfrac{c\Delta t}{\Delta x}$ is known as the CFL (Courant, Friedrichs, and Lewy) condition
  - The CFL number controls accuracy and stability
  - Given a spatial discretization step-size Δx, choose the time discretization so that the CFL number is kept in check
  - If you indeed you choose to work with very small Δt, then although stability properties are improved with a lower CFL number, the code will also slow down accordingly

Outline

- Overview
- Ordinary differential equations (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
    - Basic time and space stepping schemes
    - Examples
      - Poisson equations
      - Heat diffusion equations
      - Wave equations
      - Advection-diffusion equation
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

Advection-Diffusion Equation

- 对流扩散方程
  - 表征了流动系统的质量传递规律，求解此方程可得出浓度分布
  - 此方程系通过对系统中某空间微元体进行物料衡算而得
  - 对于双组分系统, A组分流入某微元体的量，加上在此微元体内因化学反应生成的量，减去其流出量,即为此微元体中组分A的积累量。考虑到组分A进入和离开微元体均由扩散和对流两种作用造成,而扩散通量是用斐克定律（分子扩散）表述的

Advection-Diffusion Equation

- Quasi-two-dimensional motion of atmosphere
- Advection (hyperbolic) - diffusion (parabolic) behavior
  - Vorticity $\omega(x, y, t)$
  - Streamfunction $\varphi(x, y, t)$

$$\frac{\partial \omega}{\partial t} + [\psi, \omega] = \nu \nabla^2 \omega$$

$$\nabla^2 \psi = \omega$$

$$[\psi, \omega] = \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} - \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x}$$

parabolic: $\frac{\partial \omega}{\partial t} = \nu \nabla^2 \omega$

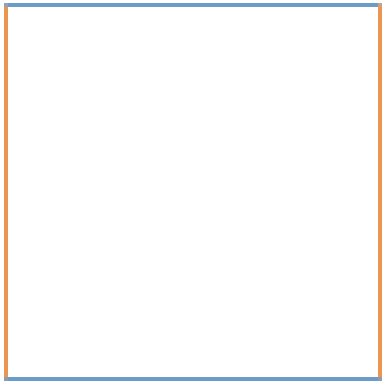
elliptic: $\nabla^2 \psi = \omega$

hyperbolic: $\frac{\partial \omega}{\partial t} + [\psi, \omega] = 0$

Advection-Diffusion Equation

- Advection-diffusion behavior
  - Vorticity $\omega(x, y, t)$
  - Streamfunction $\varphi(x, y, t)$
- Conditions
  - Initial value of vorticity $\omega(x, y, t = 0) = \omega_0(x, y)$
  - Periodic boundary conditions $x, y \in [-L, L]$

$$\omega(-L, y, t) = \omega(L, y, t)$$
$$\omega(x, -L, t) = \omega(x, L, t)$$
$$\psi(-L, y, t) = \psi(L, y, t)$$
$$\psi(x, -L, t) = \psi(x, L, t)$$



Advection-Diffusion Equation

$\psi(x, y, t)$ streamfunction

$\omega(x, y, t)$ vorticity

- Basic algorithm structure
  - Elliptic solve
    - Solve the elliptic problem $\nabla^2\psi = \omega_0$ to find the streamfunction at t_0 $\psi(x, y, t = 0) = \psi_0$
  - Time-Stepping
    - Given initial ω_0 and ψ_0, solve the advection-diffusion problem by time-stepping with a given method

$$\omega(x, y, t + \Delta t) = \omega(x, y, t) + \Delta t \left(\nu\nabla^2\omega(x, y, t) - [\psi(x, y, t), \omega(x, y, t)]\right)$$

  - Loop with the updated value $\omega(x, y, \Delta t)$

$$\frac{\partial\omega}{\partial t} + [\psi, \omega] = \nu\nabla^2\omega \quad \omega(x, y, t = 0) = \omega_0(x, y)$$

$$\nabla^2\psi = \omega$$

Advection-Diffusion Equation

- Elliptic solve $$\nabla^2 \psi = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \omega$$
  - Given ω_0, solve ψ_0
  - Discretize by central difference (second-order)

$$\frac{\psi(x+\Delta x, y, t) - 2\psi(x, y, t) + \psi(x-\Delta x, y, t)}{\Delta x^2} + \frac{\psi(x, y+\Delta y, t) - 2\psi(x, y, t) + \psi(x, y-\Delta y, t)}{\Delta y^2} = \omega(x, y, t)$$

$$\psi_{mn} = \psi(x_m, y_n) \qquad \Delta x^2 = \Delta y^2 = \delta^2$$

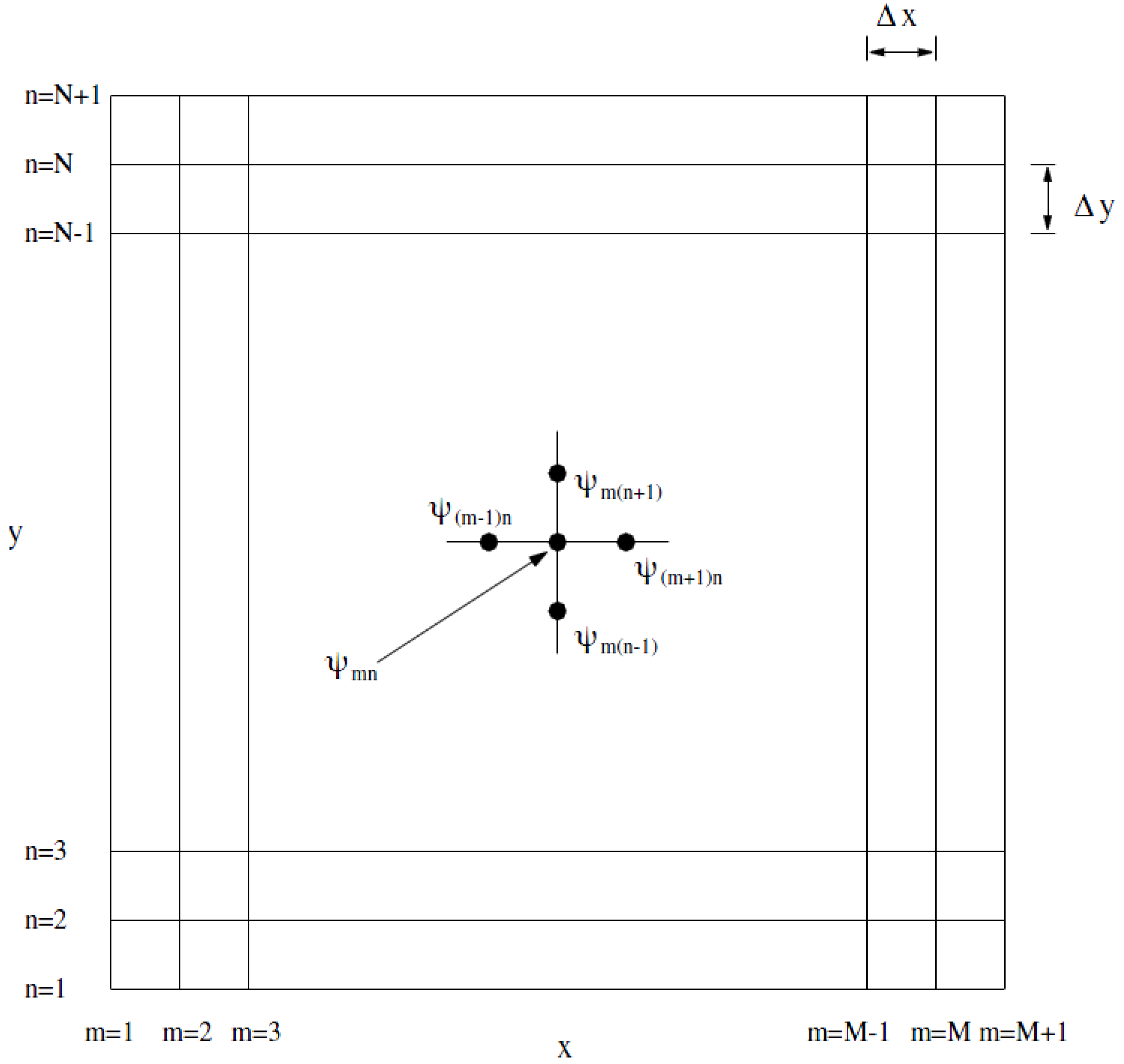
$$-4\psi_{mn} + \psi_{(m-1)n} + \psi_{(m+1)n} + \psi_{m(n-1)} + \psi_{m(n+1)} = \delta^2 \omega_{mn}$$

$$\psi_{1n} = \psi_{(N+1)n}$$

$$\psi_{m1} = \psi_{m(N+1)}$$

Advection-Diffusion Equation

$$-4\psi_{mn} + \psi_{(m-1)n} + \psi_{(m+1)n} + \psi_{m(n-1)} + \psi_{m(n+1)} = \delta^2\omega_{mn}$$



Advection-Diffusion Equation

- Elliptic solve N=4

$$-4\psi_{mn} + \psi_{(m-1)n} + \psi_{(m+1)n} + \psi_{m(n-1)} + \psi_{m(n+1)} = \delta^2\omega_{mn}$$

$$-4\psi_{11} + \psi_{41} + \psi_{21} + \psi_{14} + \psi_{12} = \delta^2\omega_{11} \qquad \psi_{1n} = \psi_{(N+1)n}$$

$$-4\psi_{12} + \psi_{42} + \psi_{22} + \psi_{11} + \psi_{13} = \delta^2\omega_{12} \qquad \psi_{m1} = \psi_{m(N+1)}$$

$$\vdots$$

$$-4\psi_{21} + \psi_{11} + \psi_{31} + \psi_{24} + \psi_{22} = \delta^2\omega_{21}$$

$$\vdots$$

- Sparse banded matrix system $\mathbf{A}\psi = \delta^2\omega$

Advection-Diffusion Equation

- Elliptic solve $\mathbf{A}\psi = \delta^2 \omega$

$$
\mathbf{A} = \begin{bmatrix}
-4 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & -4 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -4 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & -4 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -4 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & -4
\end{bmatrix}
$$

$$\psi = (\psi_{11}\ \psi_{12}\ \psi_{13}\ \psi_{14}\ \psi_{21}\ \psi_{22}\ \psi_{23}\ \psi_{24}\ \psi_{31}\ \psi_{32}\ \psi_{33}\ \psi_{34}\ \psi_{41}\ \psi_{42}\ \psi_{43}\ \psi_{44})^{\mathrm{T}}$$

$$\omega = \delta^2 (\omega_{11}\ \omega_{12}\ \omega_{13}\ \omega_{14}\ \omega_{21}\ \omega_{22}\ \omega_{23}\ \omega_{24}\ \omega_{31}\ \omega_{32}\ \omega_{33}\ \omega_{34}\ \omega_{41}\ \omega_{42}\ \omega_{43}\ \omega_{44})^{\mathrm{T}}$$

Unique solution ?

Advection-Diffusion Equation

- Poisson equation $\nabla^2\psi = \omega_0 \quad \mathbf{A}\psi = \delta^2\omega$
  - The solution of an Poisson Equation is uniquely determined in Ω, if Dirichlet boundary condition
  - Periodic boundary condition $\psi(-L,y,t) = \psi(L,y,t)$
    $\psi(x,-L,t) = \psi(x,L,t)$
    - $\psi = \psi_0 + c$
    - Singular, det(A) = 0
- Advection-diffusion $$\frac{\partial\omega}{\partial t} = \frac{\partial\psi}{\partial y}\frac{\partial\omega}{\partial x} - \frac{\partial\psi}{\partial x}\frac{\partial\omega}{\partial y} + \nu\left(\frac{\partial^2\omega}{\partial x^2} + \frac{\partial^2\omega}{\partial y^2}\right)$$
  - Only use derivative of ψ, no problem for physical model
  - Pin down the value of streamfunction at a single location in the computational domain, i.e.$\varphi(1,1) = 0$

Advection-Diffusion Equation

- Time-Stepping $\frac{\partial \omega}{\partial t} = \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} + \nu\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right)$
  - Given ω_0 and ψ_0, take a time step Δt to solve ω_1
  - Discretize by central difference (second-order)

$$
\begin{aligned}
\frac{\partial \omega}{\partial t} = & \left(\frac{\psi(x, y+\Delta y, t)-\psi(x, y-\Delta y, t)}{2\Delta y}\right)\left(\frac{\omega(x+\Delta x, y, t)-\omega(x-\Delta x, y, t)}{2\Delta x}\right) \\
& -\left(\frac{\psi(x+\Delta x, y, t)-\psi(x-\Delta x, y, t)}{2\Delta x}\right)\left(\frac{\omega(x, y+\Delta y, t)-\omega(x, y-\Delta y, t)}{2\Delta y}\right) \\
& +\nu\left\{\frac{\omega(x+\Delta x, y, t)-2\omega(x, y, t)+\omega(x-\Delta x, y, t)}{\Delta x^2}\right. \\
& \left.+\frac{\omega(x, y+\Delta y, t)-2\omega(x, y, t)+\omega(x, y-\Delta y, t)}{\Delta y^2}\right\}
\end{aligned}
$$

Advection-Diffusion Equation

- Advection term $$[\psi,\omega] = \frac{\partial\psi}{\partial x}\frac{\partial\omega}{\partial y} - \frac{\partial\psi}{\partial y}\frac{\partial\omega}{\partial x}$$

$$\frac{\partial\omega}{\partial x} = \frac{\omega(x+\Delta x, y) - \omega(x-\Delta x, y)}{2\Delta x}$$

$$\frac{\partial\omega_{mn}}{\partial x} = \frac{\omega_{(m+1)n} - \omega_{(m-1)n}}{2\Delta x}$$

$$\partial\omega/\partial x = (1/2\Delta x)\mathbf{B}\omega$$

$$\omega(x_m, y_n) = \omega_{mn}$$

$$\frac{\partial\omega_{11}}{\partial x} = \frac{\omega_{21} - \omega_{n1}}{2\Delta x}$$

$$\frac{\partial\omega_{12}}{\partial x} = \frac{\omega_{22} - \omega_{n2}}{2\Delta x}$$

$$\vdots$$

$$\frac{\partial\omega_{21}}{\partial x} = \frac{\omega_{31} - \omega_{11}}{2\Delta x}$$

$$\vdots\ .$$

Advection-Diffusion Equation

- Advection term $\partial\omega/\partial x = (1/2\Delta x)\mathbf{B}\omega$

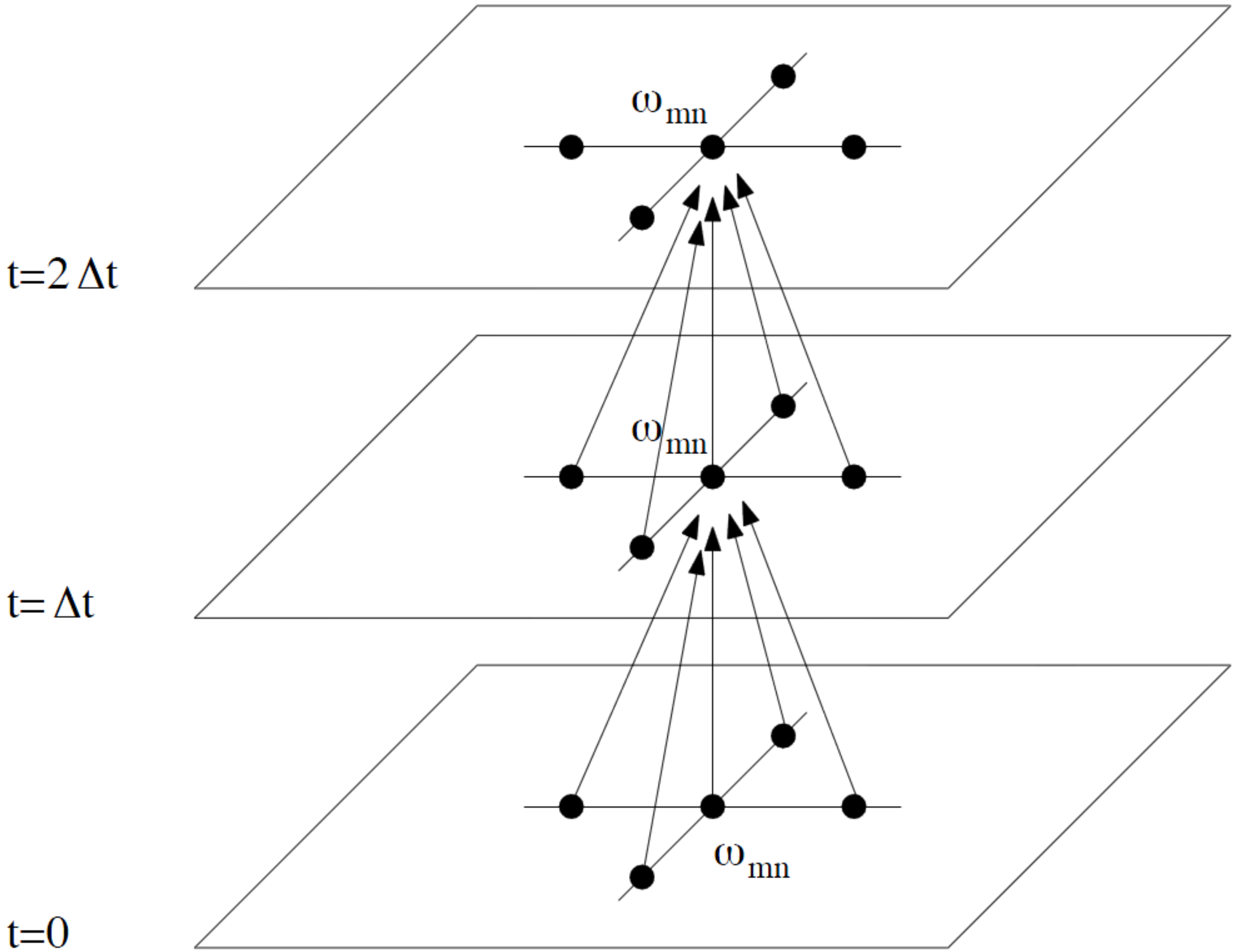
$$\omega = \begin{pmatrix} \omega_{11} \\ \omega_{12} \\ \vdots \\ \omega_{1n} \\ \omega_{21} \\ \omega_{22} \\ \vdots \\ \omega_{n(n-1)} \\ \omega_{nn} \end{pmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \ddots & & \\ \vdots & & & & & \vdots \\ & & & & & 0 \\ \vdots & \cdots & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{I} & \mathbf{0} \end{bmatrix}$$

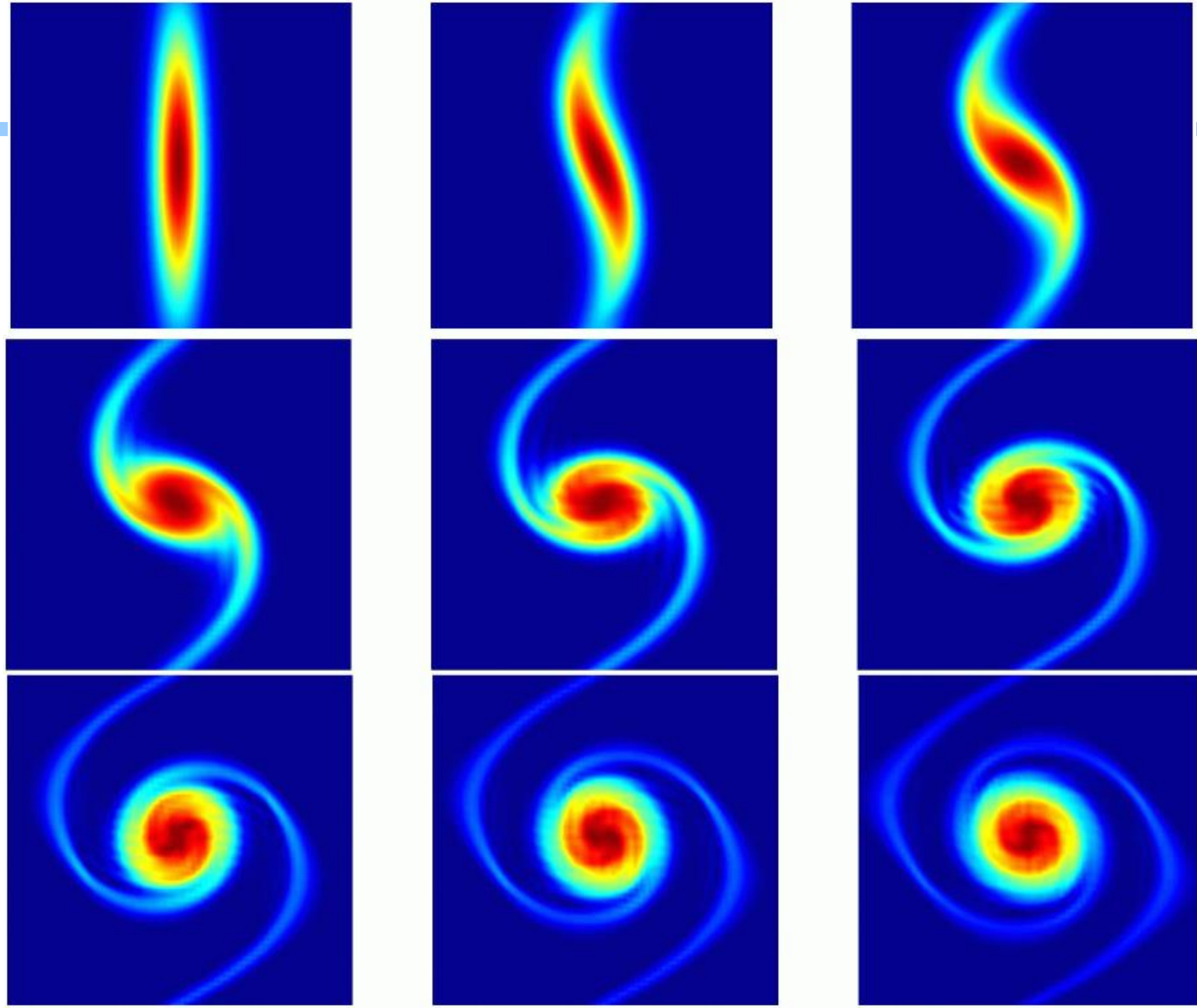
$$[\psi, \omega] = \frac{\partial\psi}{\partial x}\frac{\partial\omega}{\partial y} - \frac{\partial\psi}{\partial y}\frac{\partial\omega}{\partial x} = (\mathbf{B}\psi)(\mathbf{C}\omega) - (\mathbf{C}\psi)(\mathbf{B}\omega)$$

Advection-Diffusion Equation

- Time-Stepping
  - Discretize by central difference (second-order)
  - Time-stepping algorithm to solve a large system of differential equations
    - *N+1* points and periodic boundary conditions → *NxN* coupled system
    - 4^{th} order Runge-Kutta

Advection-diffusion Equation





Finite Difference Methods

- Typical PDEs

典型PDE求解	$\Delta u = f$ 泊松方程	$u_t = k\Delta u$ 热传导方程	$u_t = \boldsymbol{c u_x}$ 波动方程
空间离散差分近似 $\delta = \Delta x = \Delta y$	$Ax = \mathrm{b}$	$\frac{d\boldsymbol{u}}{\mathrm{dt}} = \frac{k}{\delta^2}\boldsymbol{Au}$	$\frac{d\boldsymbol{u}}{\mathrm{dt}} = \frac{c}{2\delta}\boldsymbol{Au}$
求解	(Linear Algebra) LU decomposition Conjugate gradient	(ODE) Euler method Runge-Kutta	(ODE) Euler method Runge-Kutta

$$-4u_{mn} + u_{(m-1)n} + u_{(m+1)n} + u_{m(n-1)} + u_{m(n+1)} = \delta^2 f$$

$$\frac{\partial u}{\partial t} = \frac{k}{\delta^2}[u(x+\Delta x) - 2u(x,y) + u(x-\Delta x, y)$$

$$\frac{\partial u}{\partial t} = \frac{k}{\delta^2}[-4u(x,y) + u(x-\Delta x,y) + u(x+\Delta x,y) + u(x,y-\Delta y) + u(x,y-\Delta y)]$$

$$\frac{\partial u}{\partial t} = \frac{k}{2\delta}[u(x+\Delta x) - u(x-\Delta x, y)$$

Finite Difference Methods

典型PDE求解	$\Delta u = f$ 泊松方程	$u_t = k\Delta u$ 热传导方程	$u_t = \boldsymbol{c}\boldsymbol{u}_x$ 波动方程
空间离散差分近似 $\delta = \Delta x = \Delta y$	$Ax = \mathrm{b}$	$\frac{d\boldsymbol{u}}{\mathrm{dt}} = \frac{k}{\delta^2}\boldsymbol{Au}$	$\frac{d\boldsymbol{u}}{\mathrm{dt}} = \frac{c}{2\delta}\boldsymbol{Au}$
求解	(Linear Algebra) LU decomposition Conjugate gradient	(ODE) Euler method Runge-Kutta	(ODE) Euler method Runge-Kutta

$$
\mathbf{A} = \begin{bmatrix}
-4 & 1 & 0 & 1 & 1 \\
1 & -4 & 1 & 0 & 0 \\
0 & 1 & -4 & 1 & 0 \\
1 & 0 & 1 & -4 & 0 \\
1 & 0 & 0 & 0 & -4 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
-2 & 1 & 0 & \cdots & 0 & 1 \\
1 & -2 & 1 & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \ddots & & \\
\vdots & & & & & \vdots \\
 & & & & & 0 \\
\vdots & \cdots & 0 & 1 & -2 & 1 \\
1 & 0 & \cdots & 0 & 1 & -2
\end{bmatrix}
\begin{bmatrix}
0 & 1 & 0 & \cdots & 0 & -1 \\
-1 & 0 & 1 & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \ddots & & \\
\vdots & & & & & \vdots \\
 & & & & & 0 \\
\vdots & \cdots & 0 & -1 & 0 & 1 \\
1 & 0 & \cdots & 0 & -1 & 0
\end{bmatrix}
$$

Summary

- Overview
- Ordinary differential equations (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
    - Basic time and space stepping schemes
    - Examples
      - Poisson equations
      - Heat diffusion equations
      - Wave equations
      - Advection-diffusion equation
  - Spectral methods (频谱法)
  - Finite elements (有限元法)