Lecture 10

PDE and Programming – 1

Jung-Il Choi

School of Mathematics and Computing (Computational Science and Engineering)

YONSEI UNIVERSITY

Contents

Lecture 10

- **Partial Differential Equation**
- **1D PDEs**
  - 1D Heat equation
    - ➔ Semi-discretization
    - ➔ Stability analysis
      - ✓ Eigenvalue analysis
      - ✓ Modified wavenumber analysis
      - ✓ von Neumann analysis
    - ➔ Accuracy via modified equation
    - ➔ Example 1
  - 1D Wave equation
    - ➔ Semi-discretization
    - ➔ Stability analysis
      - ✓ Modified wavenumber analysis
    - ➔ Example 2

Lecture 11

- **Multi-dimension**
  - Heat equation
    - ➔ Implicit methods in higher
    - ➔ Approximate factorization
    - ➔ Stability analysis
    - ➔ Alternating direction implicit methods (ADI)
  - Poisson equation
    - ➔ Iterative solution methods
      - ✓ Point Jacobi method
      - ✓ Gauss-Seidel method
      - ✓ Successive over relaxation method (SOR)
  - Non-linear PDEs

Partial Differential Equations

- **Partial Differential Equation (PDE)**
  - An equation stating a relationship between a function of two or more independent variables and the partial derivatives of this function with respect to these independent variables.

➔ $\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 0$: 2D Laplace equation

✓ $(\rightarrow u_{xx} + u_{yy} = 0)$

➔ $\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f(x,y)$: 2D Poisson equation

✓ $(\rightarrow u_{xx} + u_{yy} = f)$

➔ $\dfrac{\partial u}{\partial t} = c^2 \dfrac{\partial^2 u}{\partial x^2}$: 1D Diffusion equation

✓ $(\rightarrow u_t = c^2 u_{xx})$

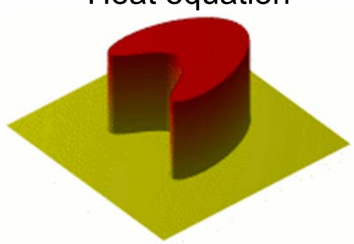
➔ $\dfrac{\partial^2 u}{\partial t^2} = c^2 \dfrac{\partial^2 u}{\partial x^2}$: 1D Wave equation

✓ $(\rightarrow u_{tt} = c^2 u_{xx})$ 이 경우는 일정한 속도와 방향으로 움직임

Diffusion equation

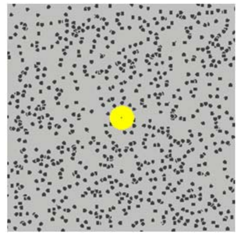
Non linear의 대표적인 예
속도가 빠르면 빠르게,
느리면 느리게
하는 wave equation

Heat equation



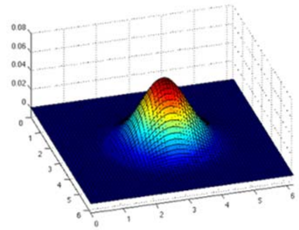
$\dfrac{\partial T}{\partial t} = \alpha \Delta T$

Brownian motion



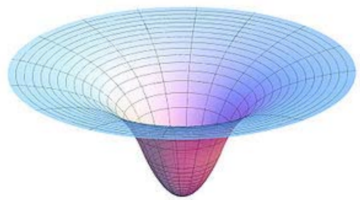
$\dfrac{\partial \rho}{\partial t} = D \Delta \rho$

Burgers' equation



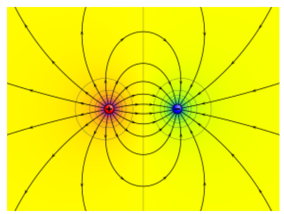
$\dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = 0$

Gravitational potential field



$\Delta \Phi = 4\pi G \rho$

Electrostatic potential field



$\Delta V = -\dfrac{\rho}{\epsilon_0}$

https://en.wilipedia.org/wiki/

Partial Differential Equations

- **Solution and linearity(or nonlinearity) of PDEs**
  - The solution of a PDE in some region R of the domain of interest, $D(\vec{x}, t)$
    - ➔ the particular function, $u(\vec{x}, t)$ satisfies the PDE in R,
    - ➔ and the initial and/or boundary conditions specified on the boundaries of $R \subset D$.

      평균에 대해서는 보장을 받을 수 있음 (analysis가능)

      initial을 무조건 정.확.히 알아야 한다는 것이 아님! 여러번의 trial을 해야함
  - Linear PDE

    https://m.blog.naver.com/pmw9440/221442252220
    - ➔ All partial derivatives appear in a linear form (first degree in the unknown function u and its derivatives)
    - ➔ "AND" none of the coefficients depend on the dependent variable

$$u_{xx} + u_{yy} = 0, \qquad u_t = c^2 u_{xx}, \qquad a u_t + b x u_x = 0$$

Variable coefficient linear PDE
depends on "independent" variable

  - Nonlinear PDE
    - ➔ The derivatives appear in a nonlinear form
    - ➔ "OR" the coefficients depend on the dependent variable

$$u u_x + b u_y = 0, \qquad a u_x^2 + b u_y$$

Partial Differential Equations

- **Order of PDE**
  - The highest-order derivative

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial y} = 0 \quad \rightarrow \quad 2^{nd}\ order$$

- **Homogeneous vs Nonhomogeneous**
  - Homogeneous PDE
    - ➔ each of the terms contains u or the dependent variables or its partial derivatives.

$$u_{xx} + u_{yy} = 0$$

$$uu_x + bu_y = 0$$

$$au_x^2 + bu_y^2 = 0$$

  - Nonhomogeneous PDE

$$\nabla^2 u = u_{xx} + u_{yy} + u_{zz} = f(x, y, z)$$

Partial Differential Equations

- **Classification of PDEs using characteristics analysis**

➔ Consider the general quasilinear 2nd order nonhomogeneous PDE in 2D:

$$Au_{xx} + Bu_{xy} + Cu_{yy} = F(x, y, u, u_x, u_y)$$

Quasilinear : linear in the highest-order derivative

- Parabolic equation ($B^2 - 4AC = 0$)

에너지가 많다면, 온 사방에 균일하게 나눠줌
모래성이 사라지는 그림 생각하면 좋아
모양이 포물선 같아서!

$$\frac{\partial u}{\partial t} = c^2 \nabla^2 u \text{ : Diffusion equation}$$

- Hyperbolic equation ($B^2 - 4AC > 0$)

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u \text{ : Wave equation}$$

파도가 쳐서 오는 모양!
u(x,t)을 y라는 식을 써서 평행이동 해버리자
t축과 x축이 있을 때, 매개체가 c:속도
c에 t를 곱하면 간 거리가 되니, y = x - ct로 하면 1변수로 바뀜
u(x,t) ==> f(x-ct) + f(x+ct) : solution이 dependent함

- Elliptic equations ($B^2 - 4AC < 0$)

$$\begin{cases} \nabla^2 u = 0 \\ \nabla^2 u = f(\vec{x}) \end{cases}$$

: Laplace equation (homogeneous) and Poisson equation (nonhomogeneous)

Heat equation의 steady stationary condition이면 laplace eq.
—> 시간에 대한 요소가 없음
—> diffusion eq. solution이 변하지 않을 때까지 가는 것 (time scale이 없어

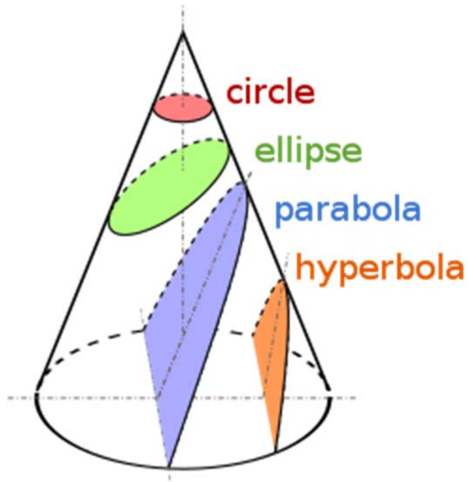
Partial Differential Equations

- **Classification of PDEs using characteristics analysis**
  - The terminology **elliptic**, **parabolic**, and **hyperbolic** chosen to classify PDEs reflects the analogy between the from of **the discriminant** $\boldsymbol{B^2 - 4AC}$
    - ➔ (from the idea of d'Alembert's solution, **methods of characteristics**)
    - ➔ And that which classifies **conic section**.

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

$$Au_{xx} + Bu_{xy} + Cu_{yy} = F(x, y, u, u_x, u_y)$$

Type	Defining condition	Examples
Parabolic	$B^2 - 4AC = 0$	Diffusion equation
Hyperbolic	$B^2 - 4AC > 0$	Wave equation
Elliptic	$B^2 - 4AC < 0$	Laplace/Poisson equation

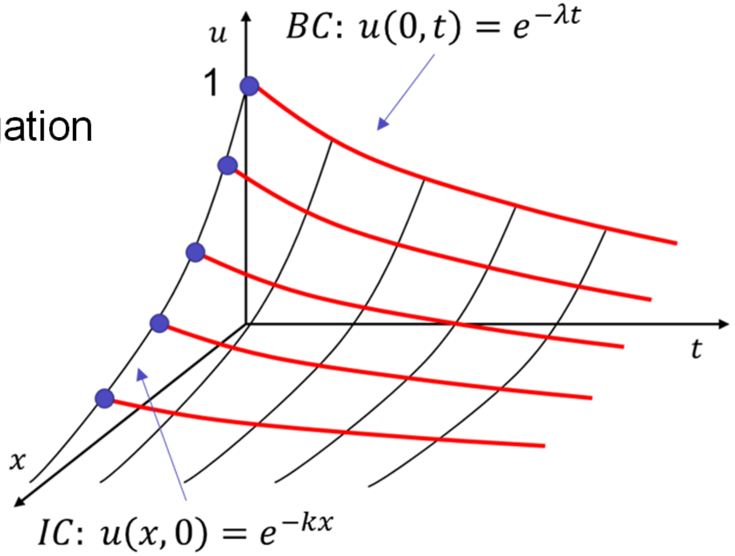


https://en.wilipedia.org/wiki/

Partial Differential Equations

$Au_{xx} + Bu_{xy} + Cu_{yy} = F(x, y, u, u_x, u_y)$		
Type	Defining condition	Examples
Parabolic	$B^2 - 4AC = 0$	Diffusion equation
Hyperbolic	$B^2 - 4AC > 0$	Wave equation
Elliptic	$B^2 - 4AC < 0$	Laplace/Poisson equation

- **Classification of PDEs using characteristics analysis**
  - Characteristics
    - ➔ Propagate behavior of each fixed point on the space at the "Hyper" space($(n+1)$D space for (n)D PDE)
    - ➔ Information, u (velocity, temperature, pressure etc.) propagates along path.
  - Are there any points in the solution domain $D(x, y)$ passing through a general point P along which the second derivatives of $u(\mathrm{x}, \mathrm{y})$ are multivalued or discontinuous (kernel space)?
    - ➔ Homogeneous solution
    - ➔ If there are such paths, they are called path of information propagation
    - ➔ Or Characteristics



Partial Differential Equations

- **Classification of PDEs using characteristics analysis**
  - Chain rule & Homogeneous solution (Kernel space)

$$d(u_x) = u_{xx}dx + u_{xy}dy$$

$$d(u_y) = u_{yx}dx + u_{yy}dy$$

$$\begin{bmatrix} A & B & C \\ dx & dy & \\ & dx & dy \end{bmatrix} \begin{bmatrix} u_{xx} \\ u_{xy} \\ u_{yy} \end{bmatrix} = \begin{bmatrix} F \\ d(u_x) \\ d(u_y) \end{bmatrix} \quad \Rightarrow \quad \det \begin{bmatrix} A & B & C \\ dx & dy & \\ & dx & dy \end{bmatrix} = 0$$

$$\Rightarrow \quad A(dy)^2 - B(dx)(dy) + C(dx)^2 = 0$$

$$\frac{dy}{dx} = \frac{B \pm \sqrt{B^2 - 4AC}}{2A} \quad \Rightarrow$$

Discriminant	Characteristics	Type
$B^2 - 4AC = 0$	Real & Repeated	Parabolic
$B^2 - 4AC > 0$	Real & Distinct	Hyperbolic
$B^2 - 4AC < 0$	Complex	Elliptic

Discriminant	Characteristics	Type
$B^2 - 4AC = 0$	Real & Repeated	Parabolic
$B^2 - 4AC > 0$	Real & Distinct	Hyperbolic
$B^2 - 4AC < 0$	Complex	Elliptic

Partial Differential Equations

- **Classification of PDEs using characteristics analysis**
  - Parabolic PDEs have one real repeated characteristic path (Critical damping, diffusing)
  - Hyperbolic PDEs of two real distinct characteristic paths (Overdamping, diffusing)
  - Elliptic PDEs have no real characteristic paths (Oscillatory)

One-dimensional PDEs

- **1D Heat equation**
  - Semi-discretization
  - Temporal discretization
  - Stability analysis
    - ➔ Eigenvalue/Eigenvector analysis
    - ➔ Modified wavenumber analysis
    - ➔ von Neumann analysis
  - Accuracy via modified equation
  - Example 1

- **1D Wave equation**
  - Semi-discretization
  - Stability analysis
    - ➔ Modified wavenumber analysis
  - Example 2

Full로 하려니 너무 써야하는게 많아..!!

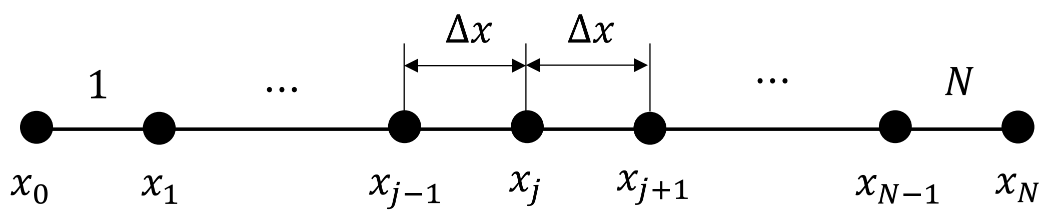
1D Heat equation

- **Semi-discretization - Solving a PDE as a system of ODEs**
  - Numerical methods for PDEs are straightforward extensions of methods developed for initial and boundary value problems in ODEs.
  - That is, a PDE can be converted to a system of ODEs by using finite difference methods for the derivatives in all but one of dimensions.
  - Consider the one-dimensional diffusion(or heat equation)

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

Initial condition : $\phi(x,0) = g(x)$
Boundary condition : $\phi(0,t) = \phi(L,t) = 0$

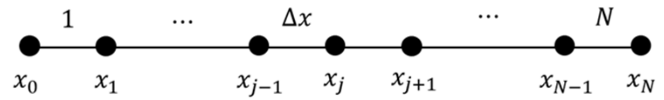
  - Discretization of the Domain with N intervals → $N+1$ uniformly spaced grid points



$$x_j = x_{j-1} + \Delta x$$
$j = 0,\ j = N$ are the boundaries
$j = 1, 2, 3, \cdots, N-1$ are interior points

공간차분 먼저 (x(j)시점)

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



Semi-discretization

- Let's use the second-order central difference scheme to the second derivative.

$$\frac{d\phi_j}{dt} = \alpha \frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta x^2}, \qquad j = 1,2,3,\cdots,N-1$$

Where $\phi_j = \phi(x_j, t)$

- A system of $N-1$ ordinary differential equations

➔ Space derivatives for fixed time (→ Semi-discretization) and solving time marching as solving ODEs.

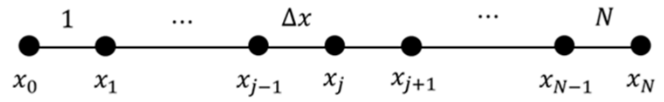
➔ Can be written in matrix form as:

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

➔ Where as ϕ_j are the (time-dependent) elements of the vector ϕ, and A is an $(N-1) \times (N-1)$ tridiagonal matrix.

Semi-discretization

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



$$\frac{d\phi_j}{dt} = \alpha \frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta x^2} \rightarrow \frac{d\phi}{dt} = A\phi$$

$$A = \frac{\alpha}{\Delta x^2}\begin{bmatrix} -2 & 1 & \cdots & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 \end{bmatrix}$$

$\rightarrow (N-1) \times (N-1)$ tridiagonal matrix which is symmetric

- The result is a system of ODEs that can be solved using any of the numerical methods introduced for ODEs, such as Euler methods, RK formulas or multi-step methods.
- However, when dealing with systems, we should be concerned about stability.
- The range of the eigenvalues of A determines whether the system is stable.

$$A\phi = \lambda\phi \rightarrow \det(A - \lambda I) = 0$$

Temporal discretization

$$A = \frac{\alpha}{\Delta x^2}\begin{bmatrix} -2 & 1 & \cdots & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 \end{bmatrix}$$

- **(Recall) Various time advancement schemes**

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

- Forward Euler scheme

$$\frac{\phi^{(n+1)} - \phi^{(n)}}{\Delta t} = A\phi^{(n)}$$

- Backward Euler scheme

$$\frac{\phi^{(n+1)} - \phi^{(n)}}{\Delta t} = A\phi^{(n+1)}$$

- Crank-Nicolson scheme

$$\frac{\phi^{(n+1)} - \phi^{(n)}}{\Delta t} = A\left[\frac{\phi^{(n+1)} + \phi^{(n)}}{2}\right]$$

Stability analysis

- **Eigenvalue/Eigenvector analysis**
  - (Recall) Diagonalization, Eigenvalues, Λ & Eigenvectors(Eigenfunctions), X

    ➔ Diagonalization (Decoupling)

    ✓ Suppose A has the eigenvalues ($ie.\,A$ is diagonalizable),

$$X^{-1}AX = \Lambda \;\rightarrow\; A = X\Lambda X^{-1}$$

$$\frac{d\phi}{dt} = A\phi \;\Rightarrow\; \frac{d\phi}{dt} = (X\Lambda X^{-1})\phi \;\Rightarrow\; X^{-1}\frac{d\phi}{dt} = \Lambda(X^{-1}\phi)$$

$$\frac{d(X^{-1}\phi)}{dt} = \Lambda(X^{-1}\phi) \quad\Rightarrow\quad \frac{d\psi}{dt} = \Lambda\psi, \quad \psi = X^{-1}\phi$$

$$\frac{d\psi_j}{dt} = \lambda_j\psi_j \;\Rightarrow\; \psi_j = c_j e^{\lambda_j t} \Rightarrow \psi = c_1\begin{bmatrix} e^{\lambda_1 t} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + c_2\begin{bmatrix} 0 \\ e^{\lambda_2 t} \\ \vdots \\ 0 \end{bmatrix} + \cdots + c_{N-1}\begin{bmatrix} 0 \\ 0 \\ \vdots \\ e^{\lambda_{N-1} t} \end{bmatrix}$$

$$\psi = \sum_{j=1}^{N-1} c_j e^{\lambda_j t} \;\rightarrow\; \phi = X\psi$$

Stability analysis

$$A = \frac{\alpha}{\Delta x^2}\begin{bmatrix} -2 & 1 & \cdots & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 \end{bmatrix}$$

$\rightarrow (N-1) \times (N-1)$ tridiagonal matrix

$A\phi = \lambda\phi \ \rightarrow \ \det(A - \lambda I) = 0$

- Analytical expressions of eigenvalues of the matrix A

$$\lambda_j = \frac{\alpha}{\Delta \mathrm{x}^2}\left(-2 + 2\cos\frac{\pi j}{N}\right), \qquad j = 1,2,3,\cdots,N-1$$

- The eigenvalue with the smallest ($j = 1$) and the largest magnitude ($j = N-1$) is:

$$\lambda_1 = \frac{\alpha}{\Delta x^2}\left(-2 + 2\cos\frac{\pi}{N}\right), \qquad \lambda_{N-1} = \frac{\alpha}{\Delta x^2}\left(-2 + 2\cos\frac{\pi(N-1)}{N}\right)$$

➔ For large, N, the Taylor series expansion for $\cos\frac{\pi}{N}$ converges rapidly, and $\cos\frac{\pi(N-1)}{N}$ converges to -1.

$$\cos\frac{\pi}{N} = 1 - \frac{1}{2!}\left(\frac{\pi}{N}\right)^2 + \frac{1}{4!}\left(\frac{\pi}{N}\right)^4 + \cdots, \qquad \cos\frac{\pi(N-1)}{N} \approx cos\pi = -1$$

➔ Using the first two terms in the expansion then,

$$\lambda_1 \approx -\frac{\pi^2\alpha}{N^2\Delta x^2}, \qquad \boxed{\lambda_{N-1} \approx -4\frac{\alpha}{\Delta x^2}}$$

Stability analysis

$$\lambda_1 \approx -4\frac{\pi^2\alpha}{N^2\Delta x^2}, \qquad \lambda_{N-1} \approx -4\frac{\alpha}{\Delta x^2}$$

- The ratio of the eigenvalue with the largest modulus to that with the smallest modulus is :

$$\left|\frac{\lambda_{N-1}}{\lambda_1}\right| \approx \frac{4N^2}{\pi^2}$$

➔ For large N, the system is unstable!

Stability analysis

frequency : 1초당 진동수
Wavenumber : wave의 숫자

- **Modified wavenumber analysis** 공간차분을 어떻게 하냐에 대해서 결정됨
  - Let revisit the heat equation,

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

➔ Assuming solution of the form $\phi(x,t) = \psi(t)e^{ikx}$

$$\boxed{\frac{d\psi}{dt} = -\alpha k^2 \psi}$$

➔ Applying to semi-discretized equation

$$\frac{d\phi_j}{dt} = \alpha \frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta x^2}, \qquad j = 1,2,3,\cdots,N-1$$

$$\phi_j(x,t) = \psi(t)e^{ikx_j}$$

$$e^{ikx_j}\frac{d\psi_j}{dt} = \frac{\alpha}{\Delta x^2}\left[e^{ikx_j}e^{ik\Delta x} - 2e^{ikx_j} + e^{ikx_j}e^{-ik\Delta x}\right]\psi_j$$

$$\frac{d\psi_j}{dt} = \frac{\alpha}{\Delta x^2}\left[-2 + e^{ik\Delta x} + e^{-ik\Delta x}\right]\psi = -\frac{2\alpha}{\Delta x^2}[1-\cos(k\Delta x)]\psi_j \boxed{= -\alpha k'^2 \psi_j}$$

Modified wavenumber

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

Stability analysis

$$\frac{d\psi_j}{dt} = -\alpha k'^2 \psi$$

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

➔ $-\alpha k'^2 = \lambda$

$$\psi' = \lambda\psi$$

➔ Using the forward Euler for time advancement,

$$\frac{\psi_j^{(n+1)} - \psi_j^{(n)}}{\Delta t} = -\frac{2\alpha}{\Delta x^2}[1 - \cos(k\Delta x)]\psi_j^{(n)}$$

$$\Delta t \le \frac{2}{|\lambda|} \quad \Rightarrow \quad \Delta t \le \frac{2}{\left|\frac{2\alpha}{\Delta x^2}[1 - \cos(k\Delta x)]\right|}$$

➔ Since, $-1 \le \cos(k\Delta x) \le 1$, the worst-case scenario is :

$$\Delta t_{max} = \frac{\Delta x^2}{2\alpha}$$

Stability analysis

$$\frac{d\psi_j}{dt} = -\alpha k'^2 \psi$$

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

➔ Crank-Nicolson method

$$\frac{\psi_j^{(n+1)} - \psi_j^{(n)}}{\Delta t} = -\alpha k'^2 \left[\frac{\psi_j^{(n+1)} + \psi_j^{(n)}}{2}\right]$$

$$\left(1 + \frac{\alpha k'^2}{2}\right)\psi^{(n+1)} = \left(1 - \frac{\alpha k'^2}{2}\right)\psi^{(n)}$$

✓ For the stability analysis,

$$\psi^{(n+1)} = \sigma\psi^{(n)}$$

$$\text{Where } \sigma = \frac{\left(1 - \frac{\alpha k'^2}{2}\right)}{\left(1 + \frac{\alpha k'^2}{2}\right)} = \frac{1 - \frac{\alpha\Delta t}{\Delta x^2}[1-\cos(k\Delta x)]}{1 - \frac{\alpha\Delta t}{\Delta x^2}[1+\cos(k\Delta x)]} \quad \Rightarrow \quad |\sigma| \le 1$$

➔ Crank-Nicolson is unconditionally stable

Stability analysis

$$\frac{d\psi_j}{dt} = -\alpha k'^2 \psi$$

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

➔ Using backward Euler

$$\frac{\psi_j^{(n+1)} - \psi_j^{(n)}}{\Delta t} = -\alpha k'^2 \psi_j^{(n+1)}$$

$$\left(1 + \frac{\alpha k'^2}{2}\right)\psi^{(n+1)} = \psi^{(n)}$$

✓ For the stability analysis,

$$\psi^{(n+1)} = \gamma \psi^{(n)}$$

Where $\gamma = \dfrac{1}{\left(1 + \dfrac{\alpha k'^2}{2}\right)} = \dfrac{1}{1 - \dfrac{\alpha \Delta t}{\Delta x^2}[1 + \cos(k\Delta x)]} \quad \Rightarrow \quad |\gamma| \leq 1$

➔ Backward Euler is unconditionally stable

✓ However, in contrast to Crank-Nicolson, $\sigma \to 0$ when $\Delta t \to \infty$. That is, the solution does not exhibit undesirable oscillations (although it would be inaccurate).

Stability analysis

- Consider the wave equation,

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \qquad 0 \le x \le L, \qquad t \ge 0$$

➔ Assuming, $u(x,t) = v(t)e^{ikx}$

$$e^{ikx}\frac{dv}{dt} = -ikc\left(e^{ikx}\right)v \;\Rightarrow\; \frac{dv}{dt} = -ikc\,v$$

Modified wavenumber

$$k' = \frac{\sin(k\Delta x)}{\Delta x}$$

➔ Semi-discretized equation with central difference scheme,

$$\frac{du_j}{dt} + c\frac{u_{j+1} - u_{j-1}}{2\Delta x} = 0 \;\Rightarrow\; \frac{dv_j}{dt} = -ic\frac{\sin(k\Delta x)}{\Delta x}v_j = -ick'v_j$$

Stability analysis

- **von Neumann stability analysis**
  - Matrix stability analysis using the eigenvalues of the matrix obtained from a semi-discretization of PDE
    - ➔ This is only available for very simple matrices
  - Consider full discretization of PDE
    - ➔ von Neumann stability analysis does not account for the effect of boundary conditions; periodic boundary conditions are assumed.
    - ➔ Linear, constant coefficient differential equations with uniformly spaced spatial grids.

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

  - Second-order central difference with the explicit Euler method

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

  - Assuming a solution of the form

$$\phi_j^{(n)} = \sigma^n e^{ikx_j}$$

Stability analysis

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

$$\phi_j^{(n)} = \sigma^n e^{ikx_j}$$

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

$$\sigma^{n+1} e^{ikx_j} = \sigma^n e^{ikx_j} + \frac{\alpha \Delta t}{\Delta x^2}\sigma^n\left(e^{ikx_{j+1}} - 2e^{ikx_j} + e^{ikx_{j-1}}\right)$$

Where $x_{j+1} = x_j + \Delta x$ and $x_{j-1} = x_j - \Delta x$.

$$\sigma = 1 + \left(\frac{\alpha \Delta t}{\Delta x^2}\right)[2\cos(k\Delta x) - 2]$$

For stability, $|\sigma| \leq 1$

$$\left|1 + \left(\frac{\alpha \Delta t}{\Delta x^2}\right)[2\cos(k\Delta x) - 2]\right| \leq 1$$

$$\left(\frac{\alpha \Delta t}{\Delta x^2}\right)[2\cos(k\Delta x) - 2] \geq -2 \quad \rightarrow \quad \Delta t \leq \frac{\Delta x^2}{\alpha[1 - \cos(k\Delta x)]}$$

The worst (or the most restrictive) case occurs when $\cos(k\Delta x) = -1$.

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

Accuracy via modified equation

- Second-order central difference with the explicit Euler method :

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

- Define the numerical operator as:

$$L\left[\phi_j^{(n)}\right]=\underbrace{\frac{\phi_j^{(n+1)}-\phi_j^{(n)}}{\Delta t}}_{1}-\alpha\underbrace{\frac{\phi_{j+1}^{(n)}-2\phi_j^{(n)}+\phi_{j-1}^{(n)}}{\Delta x^2}}_{2},\qquad L[\phi_j]=0$$

- Consider the Taylor's expansion,

1 $$\phi_j^{(n+1)}=\phi_j^{(n)}+\Delta t\frac{\partial\phi_j^{(n)}}{\partial t}+\frac{\Delta t^2}{2!}\frac{\partial^2\phi_j^{(n)}}{\partial t^2}+\cdots\ \Rightarrow\ \frac{\phi_j^{(n+1)}-\phi_j^{(n)}}{\Delta t}=\frac{\partial\phi_j^{(n)}}{\partial t}+\frac{\Delta t}{2!}\frac{\partial^2\phi_j^{(n)}}{\partial t^2}+\cdots$$

2 $$\frac{\phi_{j+1}^{(n)}-2\phi_j^{(n)}+\phi_{j-1}^{(n)}}{\Delta x^2}=\left.\frac{\partial^2\phi^{(n)}}{\partial x^2}\right|_j+\frac{\Delta x^2}{2\cdot 3!}\left.\frac{\partial^4\phi^{(n)}}{\partial x^4}\right|_j+\cdots$$

Accuracy via modified equation

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

$$\frac{\phi_j^{(n+1)} - \phi_j^{(n)}}{\Delta t} = \frac{\partial \phi_j^{(n)}}{\partial t} + \frac{\Delta t}{2!}\frac{\partial^2 \phi_j^{(n)}}{\partial t^2} + \cdots$$

$$\frac{\phi_{j+1}^{(n)} - 2\phi_j^{(n)} + \phi_{j-1}^{(n)}}{\Delta x^2} = \left.\frac{\partial^2 \phi^{(n)}}{\partial x^2}\right|_j + \frac{\Delta x^2}{2\cdot 3!}\left.\frac{\partial^4 \phi^{(n)}}{\partial x^4}\right|_j + \cdots$$

- By ① $-\alpha$ ② ,

$$L\left[\phi_j^{(n)}\right] - \left(\frac{\partial \phi_j^{(n)}}{\partial t} - \alpha \left.\frac{\partial^2 \phi^{(n)}}{\partial x^2}\right|_j\right) = -\alpha \frac{\Delta x^2}{12}\left.\frac{\partial^4 \phi^{(n)}}{\partial x^4}\right|_j + \frac{\Delta t}{2}\frac{\partial^2 \phi_j^{(n)}}{\partial t^2} + \cdots$$

$$L[\phi] - \left(\frac{\partial \phi}{\partial t} - \alpha \frac{\partial^2 \phi}{\partial x^2}\right) = -\alpha \frac{\Delta x^2}{12}\frac{\partial^4 \phi}{\partial x^4} + \frac{\Delta t}{2}\frac{\partial^2 \phi}{\partial t^2} + \cdots \quad \Rightarrow \quad \frac{\partial \phi}{\partial t} - \alpha \frac{\partial^2 \phi}{\partial x^2} = \alpha \frac{\Delta x^2}{12}\frac{\partial^4 \phi}{\partial x^4} - \frac{\Delta t}{2}\frac{\partial^2 \phi}{\partial t^2} + \cdots$$

- Let $\tilde{\phi}$ be the exact solution for $\frac{\partial \tilde{\phi}}{\partial t} = \alpha \frac{\partial^2 \tilde{\phi}}{\partial x^2}$,

$$L[\tilde{\phi}] = \varepsilon$$

$$\varepsilon = -\alpha \frac{\Delta x^2}{12}\frac{\partial^4 \tilde{\phi}}{\partial x^4} + \frac{\Delta t}{2}\frac{\partial^2 \tilde{\phi}}{\partial t^2} + \cdots = \left(-\alpha \frac{\Delta x^2}{12} + \alpha^2 \frac{\Delta t}{2}\right)\frac{\partial^4 \tilde{\phi}}{\partial x^4} + \cdots \quad \leftarrow \quad \frac{\partial^2 \tilde{\phi}}{\partial t^2} = \alpha \frac{\partial^3 \tilde{\phi}}{\partial t \partial x^2} = \alpha^2 \frac{\partial^4 \tilde{\phi}}{\partial x^4}$$

- Let set CFL number as:

$$\frac{\alpha \Delta t}{\Delta x^2} = \frac{1}{6}$$

  - ➔ Then, significantly increase the accuracy of method.
  - ➔ Recall) For stability, $CFL \leq 1/2$

Example 1

- Consider the (unsteady) heat equation (or 1D diffusion equation) given by:

$$\boxed{\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}} + \boxed{\left(\pi^2 - 1\right)e^{-t}\sin \pi x} \qquad \begin{matrix} 0 \le x \le 1 \\ t \ge 0 \end{matrix}$$

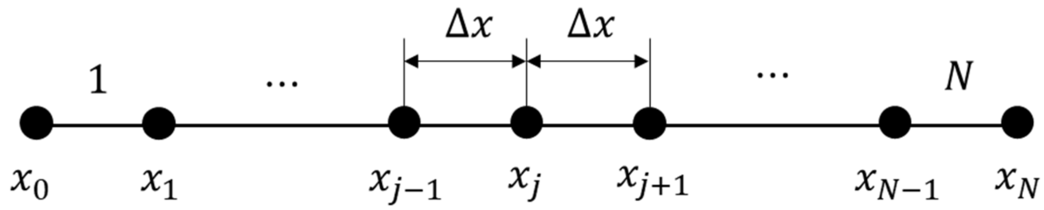
Solution은 여기에 의존

➔ Initial conditions: $T(x,0) = \sin \pi x$

➔ Boundary conditions : $T(0,t) = T(1,t) = 0$

➔ Assume $\alpha = 1$ and $\Delta\mathrm{x} = 0.05$, $N = 21$

➔ $x_j = x_{j-1} + \Delta x$



- Discretized equation:

$$\left.\frac{dT}{dt}\right|_j = \alpha\left(\frac{T_{j+1} - 2T_j + T_{j-1}}{\Delta x^2}\right) + f_j, \qquad j = 1,2,3,\cdots,N-1$$

$$f_j = \left(\pi^2 - 1\right)e^{-t}\sin \pi x_j$$

Example 1

- Discretized equation:

$$\left.\frac{dT}{dt}\right|_j = \overbrace{\underbrace{\alpha\left(\frac{T_{j+1} - 2T_j + T_{j-1}}{\Delta x^2}\right)} + f_j,}^{F} \qquad j = 1,2,3,\cdots,N-1$$

$$A = \frac{\alpha}{\Delta x^2}\begin{bmatrix} -2 & 1 & \cdots & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 \end{bmatrix} \Rightarrow \begin{aligned} \left.\frac{dT}{dt}\right|_1 &= \frac{\alpha}{\Delta x^2}(T_2 - 2T_1) + \left(\pi^2 - 1\right)e^{-t}\sin \pi x_1 \\ \left.\frac{dT}{dt}\right|_2 &= \frac{\alpha}{\Delta x^2}(T_3 - 2T_2 + T_1) + \left(\pi^2 - 1\right)e^{-t}\sin \pi x_2 \\ &\vdots \end{aligned}$$

➔ The PDE has been converted to a system of ODEs

- Time advancement

➔ Using forward Euler,

$$T_j^{(\mathrm{n+1})} = T_j^{(n)} + \Delta t F(T_j^{(n)}, t_n)$$

Example 1

- The stability of the numerical solution for time advancement depends on the eigenvalue of the system having the largest magnitude:

$$\lambda_{N-1} \approx -4\frac{\alpha}{\Delta x^2}$$

➔ When forward Euler is used for real and negative λ:

$$\Delta t_{max} = \frac{2}{|\lambda|_{max}} = \frac{\Delta x^2}{2\alpha}$$

➔ For $\alpha = 1$ and $\Delta x = 0.05$,

$$\Delta t_{max} = 0.00125$$

Example 1

- **Pseudo-code & Code**

Program solve_heat_eq
$x_0 \leftarrow 0,\ x_{max} \leftarrow 1$
$t_0 \leftarrow 0,\quad nt \leftarrow 2000$
$dt \leftarrow 0.001,\ dx \leftarrow 0.05$
$T \leftarrow \sin \pi x$
$T[0], T[x_{max}] \leftarrow 0, 0$

$Call\ Euler(t,\ T,\ nt,\ dt,\ heateq,\ x,\ dx)$
End program

Function Euler$(t_0, y^0, nt, dt, f, x, dx)$
$y = y^0$

$for\ i = 0\ to\ nt - 1\ do$
$\quad t = t_0 + (i+1)dt$
$\qquad rhs \leftarrow f(y, t, x, dx)$
$\quad y = y + rhs * dt$
$end\ for$
$return\ t, y$
End function

Function heateq(T, t, x, dx)
$alpha \leftarrow 1$
$source \leftarrow (\pi^2 - 1)e^{-t} \sin \pi x$

$dudt_i \leftarrow \frac{(T_{i+1} - 2T_i + T_{i-1})}{dx^2} + source_i$
$return\ dudt$
End function

```
x0 = 0
xmax =1

t0 = 0
nt = 2001
dt = 0.001
dx = 0.05
nx = int((xmax-x0)/dx + 1)

x = np.linspace(x0, xmax, nx)

# initial condition
T = np.sin(np.pi * x)
# boundary condition
T[0], T[-1] = 0, 0

# solve Heat Equation
t = t0

t,T= Euler(t,T,nt,dt,heateq,x,dx)
```

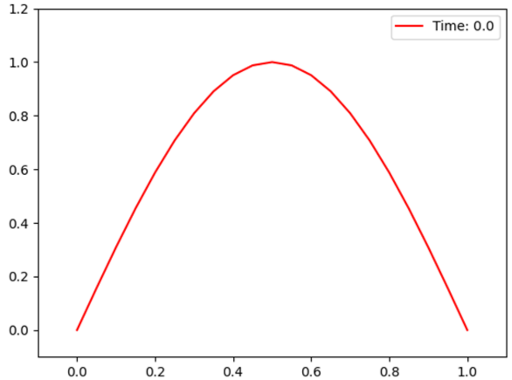
```
def heateq(T,t,x,dx):
    alpha = 1
    dudt = np.zeros(len(T))

    source = (np.pi**2 -1)*np.exp(-t)*np.sin(np.pi*x)
    #central difference
    dudt[1:-1] = (T[:-2] - 2*T[1:-1] + T[2:])/dx**2 +source[1:-1]

    return dudt
```

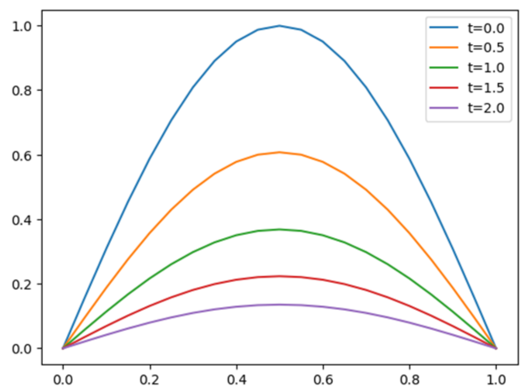
```
def Euler(t0, y0, nt, dt, f, x, dx):
    y = y0

    for i in range(nt):
        t = t0 + dt*(i+1)

        rhs = f(y,t,x,dx)

        y = y + dt*rhs

    return t,y
```

Example 1

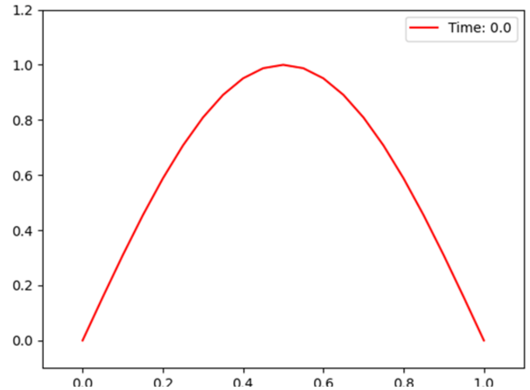
> ➔ For $\alpha = 1$ and $\Delta x = 0.05$,
> $\Delta t_{max} = 0.00125$

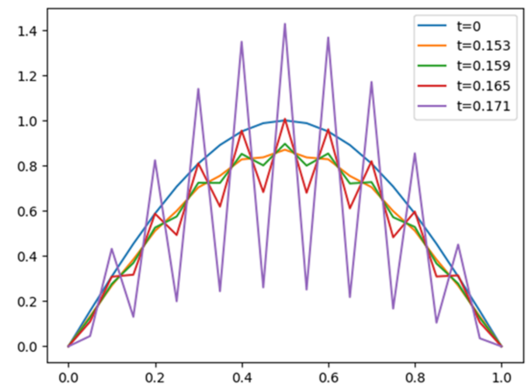
- **result**
  - $\Delta t = 0.001, \Delta x = 0.05,\ \alpha = 1$





  - $\Delta t = 0.0015, \Delta x = 0.05,\ \alpha = 1$





1D Wave equation

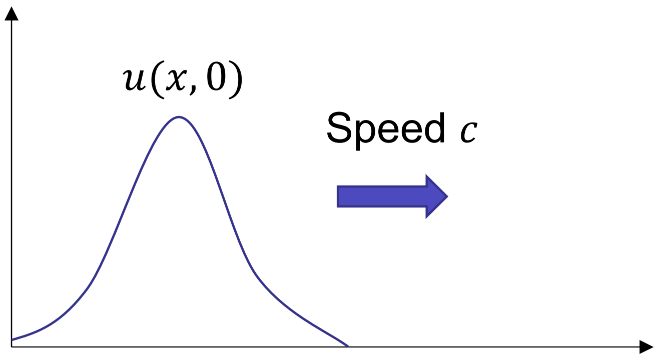
- Consider a semi-discretization of the following first-order wave equation :
  - ➔ aka the convection/transport equation

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

Initial condition : $u(x,0) = f(x)$
Boundary condition : $u(0,t) = \phi(L,t) = 0$

  - ➔ A simple model equation for the convection phenomena.

  - ➔ The exact solution is such that an initial disturbance in the domain ($u(x,0)$) simply propagates with the constant convection speed c in the positive (or negative) x-direction.



Semi-discretization

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

- Semi-discretization
  - ➔ Assume $c > 0$, and using central difference scheme,

$$\frac{du_j}{dt} + c\frac{u_{j+1} - u_{j-1}}{2\Delta x} = 0$$

  - ➔ In matrix from,

$$\frac{du}{dt} = Au$$

$$\text{where } A = -\frac{c}{2\Delta x}\begin{bmatrix} 0 & 1 & \cdots & \\ -1 & 0 & 1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 0 \end{bmatrix} \rightarrow (N-1)\times(N-1) \text{ tridiagonal matrix which is not symmetric}$$

  - ✓ From analytical consideration, no boundary condition is prescribed at $x = L$.
  - ✓ However, a special numerical boundary treatment is required at $x = L$ owing to the use of central differencing in this problem.
  - ✓ A typical well-behaved numerical boundary treatment at $x = L$ slightly modifies the last row of the coefficient matrix A, but we will ignore it for now.

Stability analysis

$$A = -\frac{c}{2\Delta x}\begin{bmatrix} 0 & 1 & \cdots & \\ -1 & 0 & 1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 0 \end{bmatrix}$$

$\rightarrow (N-1)\times(N-1)$ tridiagonal matrix

- **Modified wavenumber analysis**
  - The eigenvalues of A are :

$$\lambda_j = -\frac{c}{\Delta x}\left(i\cos\frac{\pi j}{N}\right), \qquad j = 1,2,3,\cdots,N-1$$

➔ Thus, the eigenvalues of the matrix resulting from semi-discretization of the convection equation are purely imaginary

$$\lambda_j = i\omega_j, \qquad \text{where } \omega_j = -\frac{c}{\Delta x}\left(\cos\frac{\pi j}{N}\right)$$

➔ The solution is a superposition of modes, where each mode's temporal behavior is given by $e^{i\omega_j t}$

➔ Oscillatory or sinusoidal(non-decaying) character.

Stability analysis

$$\frac{dv_j}{dt} = -ick'v_j$$
$$k' = \frac{\sin(k\Delta x)}{\Delta x}$$

➔ Leap flog method for time advancement,

$$\frac{v_j^{(n+1)} - v_j^{(n-1)}}{2\Delta t} = -ic\frac{\sin(k\Delta x)}{\Delta x}v_j^{(n)}$$

$$\Delta t \le \frac{1}{k'c} = \frac{\Delta x}{c\sin(k\Delta x)}$$

✓ The worst-case scenario is,

$$\Delta t_{max} = \frac{\Delta x}{c}$$

$$CFL \le 2.83$$

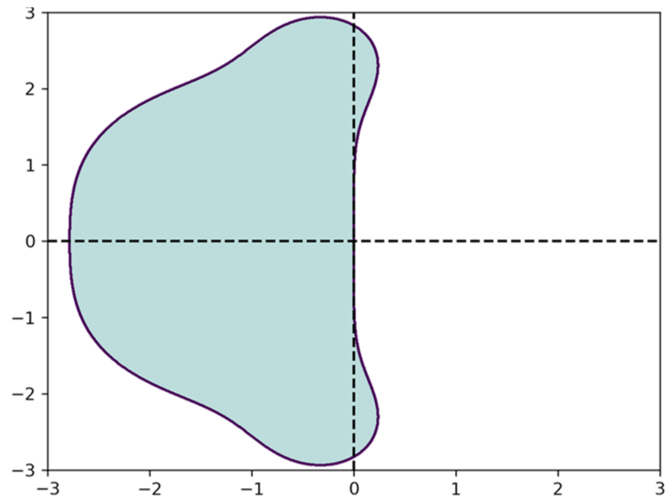
➔ Courant, Friedrich and Lewy (CFL) number is defined:

$$CFL = \frac{c\Delta t}{\Delta x} \le 1$$

➔ RK4

$$CFL \le 2.83$$

Stability region of RK4



Example 2

- Consider the numerical solution to the homogeneous convection equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \qquad \begin{matrix} 0 \le x \le L \\ t \ge 0 \end{matrix}$$

➔ Initial conditions: $u(x,0) = e^{-200(x-0.25)^2}$

➔ Boundary conditions: $u(0,t) = 0$

➔ Although the proper spatial domain for this PDE is semi-infinite, numerical implementation requires a finite domain.

➔ Thus, we arbitrarily truncate the domain to $0 \le x \le 1$

- Semi-discretized equation using a 2nd order central difference scheme:

$$\frac{du_i}{dt} + c\frac{u_{j+1} - u_{j-1}}{2\Delta x} = 0$$

Example 2

- **Pseudo-code & Code**

Program solve_wave_eq
$x_0 \leftarrow 0,\ x_{max} \leftarrow 0.75$
$t_0 \leftarrow 0,\quad nt \leftarrow 21$
$dt \leftarrow 0.01,\ dx \leftarrow 0.01$
$\mathrm{u} \leftarrow \mathrm{e}^{-200(\mathrm{x}-0.25)^2}$
$u[0] \leftarrow 0$

$Call\ Euler(or\ RK4)(t,\ u,\ nt,\ dt,\ waveeq,\ x,\ dx)$
End program

Function waveeq(T, t, x, dx)
$c \leftarrow 1$
$source \leftarrow (\pi^2 - 1)e^{-t}\sin \pi x$

$dudt_i \leftarrow \frac{(u_{i+1} - u_{i-1})}{dx}$
$\#periodic\ boundary\ condition$
$dudt_0 \leftarrow \frac{(u_1 - u_{xmax})}{dx}$
$dudt_{xmax} \leftarrow \frac{(u_0 - u_{xmax-1})}{dx}$

$return\ dudt$
End function

Function Euler$(t_0, y^0, nt, dt, f, x, dx)$
$y = y^0$

$for\ i = 0\ to\ nt - 1\ do$
$t = t_0 + (i+1)dt$
$rhs \leftarrow f(y, t, x, dx)$
$y = y + rhs * dt$
$end\ for$
$return\ t, y$
End function

Function RK4$(t_0, y^0, nt, dt, f, x, dx)$
$y = y^0$

$for\ i = 0\ to\ nt - 1\ do$
$t = t_0 + (i+1)dt$
$\mathrm{k1} \leftarrow dt * f(y, t, x, dx)$
$\mathrm{k2} \leftarrow dt * f(y + \frac{k1}{2}, t + \frac{dt}{2}, x, dx)$
$\mathrm{k3} \leftarrow dt * f(y + \frac{k1}{2}, t + \frac{dt}{2}, x, dx)$
$\mathrm{k4} \leftarrow dt * f(y + k3, t + dt, x, dx)$

$y = y + \frac{k1}{6} + \frac{k2}{3} + \frac{k3}{3} + \frac{k4}{6}$
$end\ for$
$return\ t, y$
End function

```python
def Euler(t0, y0, nt, dt, f, x, dx):
    y = y0

    for i in range(nt):
        t = t0 + dt*(i+1)

        rhs = f(y,t,x,dx)

        y = y + dt*rhs

    return t,y
```

```python
def RK4(t0, y0, nt, dt, f, x, dx):
    y = y0

    for i in range(nt):
        t = t0 + dt*(i+1)

        k1 = dt*f(y, t, x, dx)
        k2 = dt*f(y+k1/2, t+dt/2, x, dx)
        k3 = dt*f(y+k2/2, t+dt/2, x, dx)
        k4 = dt*f(y+k3  , t+dt  , x, dx)

        y = y + k1/6 + k2/3 + k3/3 + k4/6

    return t,y
```

Example 2

- **Pseudo-code & Code**

Program solve_wave_eq
$x_0 \leftarrow 0,\ x_{max} \leftarrow 0.75$
$t_0 \leftarrow 0,\quad nt \leftarrow 21$
$dt \leftarrow 0.01,\ dx \leftarrow 0.01$
$\mathrm{u} \leftarrow \mathrm{e}^{-200(\mathrm{x}-0.25)^2}$
$u[0] \leftarrow 0$

$Call\ Euler(or\ RK4)(t,\ u,\ nt,\ dt,\ waveeq,\ x,\ dx)$
End program

Function waveeq(T, t, x, dx)
$c \leftarrow 1$
$source \leftarrow (\pi^2 - 1)e^{-t}\sin \pi x$

$dudt_i \leftarrow \frac{(u_{i+1}-u_{i-1})}{dx}$
$\#periodic\ boundary\ condition$
$dudt_0 \leftarrow \frac{(u_1-u_{xmax})}{dx}$
$dudt_{xmax} \leftarrow \frac{(u_0-u_{xmax-1})}{dx}$

$return\ dudt$
End function

Function Euler$(t_0, y^0, nt, dt, f, x, dx)$
$y = y^0$

$for\ i = 0\ to\ nt-1\ do$
$\quad t = t_0 + (i+1)dt$
$\qquad rhs \leftarrow f(y, t, x, dx)$
$\quad y = y + rhs * dt$
$end\ for$
$return\ t, y$
End function

Function RK4$(t_0, y^0, nt, dt, f, x, dx)$
$y = y^0$

$for\ i = 0\ to\ nt-1\ do$
$\quad t = t_0 + (i+1)dt$
$\qquad \mathrm{k1} \leftarrow dt * f(y, t, x, dx)$
$\qquad \mathrm{k2} \leftarrow dt * f(y + \frac{k1}{2}, t + \frac{dt}{2}, x, dx)$
$\quad \mathrm{k3} \leftarrow dt * f(y + \frac{k1}{2}, t + \frac{dt}{2}, x, dx)$
$\quad \mathrm{k4} \leftarrow dt * f(y + k3, t + dt, x, dx)$

$\quad y = y + \frac{k1}{6} + \frac{k2}{3} + \frac{k3}{3} + \frac{k4}{6}$
$end\ for$
$return\ t, y$
End function

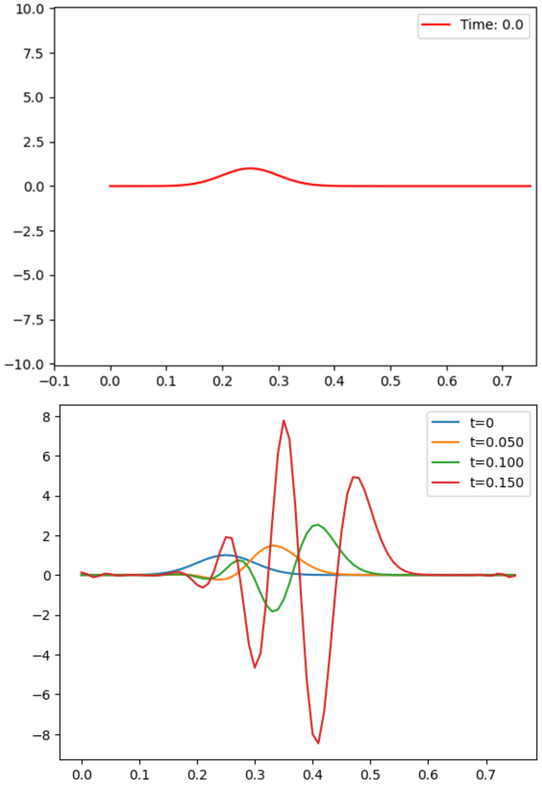
```
x0 = 0
xmax = 0.75

t0 = 0
dt = 0.01
dx = 0.01
nx = int((xmax-x0)/dx + 1)
nt= 21

x = np.linspace(x0, xmax, nx)

# initial condition
u = np.exp(-200*(x-0.25)**2)
# boundary condition
u[0] = 0

# solve Wave Equation
t = t0

t,u = Euler(t,u,nt,dt,waveeq,x,dx)
#t,u = RK4(t,u,nt,dt,waveeq,x,dx)
```

```
def waveeq(u,t,x,dx):
    c = 1
    dudt = np.zeros(len(u))

    # central difference
    dudt[1:-1] = -c*(u[2:]-u[:-2])/dx

    # boundary condition
    dudt[0] = -c*(u[1]-u[-1])/dx
    dudt[-1]= -c*(u[0]-u[-2])/dx

    return dudt
```

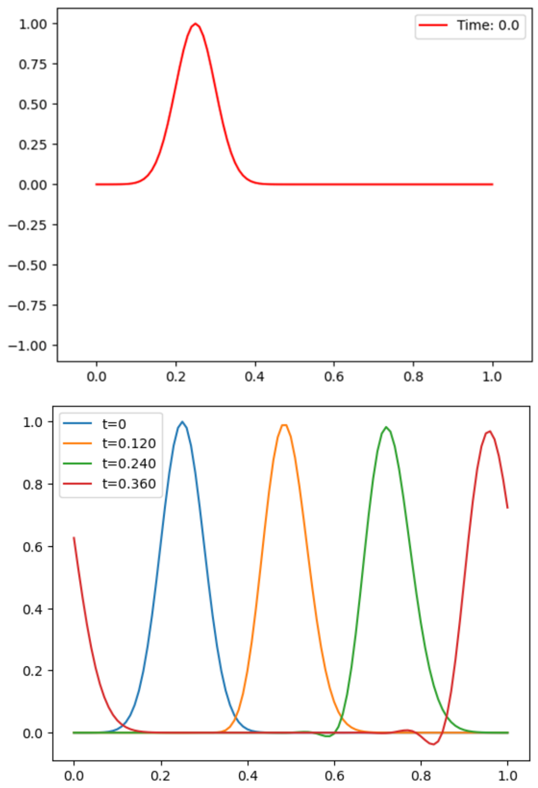
Example 2

$$\begin{aligned} forward\ Euler&: CFL \leq 1 \\ RK4&: CFL \leq 2.83 \end{aligned}$$

- **result** $(\Delta t = 0.01, \Delta x = 0.01,\ c = 1)$
  - Euler method



  - Fourth order Runge-Kutta method



Q&A Thanks for listening