2.29 Numerical Fluid Mechanics
Fall 2011 – Lecture 12

REVIEW Lecture 11:

- End of (Linear) Algebraic Systems
  - Gradient Methods
  - Krylov Subspace Methods
  - Preconditioning of $\mathbf{Ax}=\mathbf{b}$
- FINITE DIFFERENCES
  - Classification of Partial Differential Equations (PDEs) and examples with finite difference discretizations
    - Parabolic PDEs
    - Elliptic PDEs
    - Hyperbolic PDEs

FINITE DIFFERENCES - Outline

- Classification of Partial Differential Equations (PDEs) and examples with finite difference discretizations
  - Parabolic PDEs, Elliptic PDEs and Hyperbolic PDEs
- Error Types and Discretization Properties
  - Consistency, Truncation error, Error equation, Stability, Convergence
- Finite Differences based on Taylor Series Expansions
  - Higher Order Accuracy Differences, with Example
  - Taylor Tables or Method of Undetermined Coefficients
- Polynomial approximations
  - Newton's formulas
  - Lagrange polynomial and un-equally spaced differences
  - Hermite Polynomials and Compact/Pade's Difference schemes
  - Equally spaced differences
    - Richardson extrapolation (or uniformly reduced spacing)
    - Iterative improvements using Roomberg's algorithm

References and Reading Assignments

- Part 8 (PT 8.1-2), Chapter 23 on "Numerical Differentiation" and Chapter 18 on "Interpolation" of "Chapra and Canale, Numerical Methods for Engineers, 2010/2006."
- Chapter 3 on "Finite Difference Methods" of "J. H. Ferziger and M. Peric, Computational Methods for Fluid Dynamics. Springer, NY, 3rd edition, 2002"
- Chapter 3 on "Finite Difference Approximations" of "H. Lomax, T. H. Pulliam, D.W. Zingg, *Fundamentals of Computational Fluid Dynamics (Scientific Computation).* Springer, 2003"

Classification of Partial Differential Equations

(2D case, 2nd order)

Quasi-linear PDE for $\phi(x,y)$

$$A\phi_{xx} + B\phi_{xy} + C\phi_{yy} = F(x,y,\phi,\phi_x,\phi_y)$$

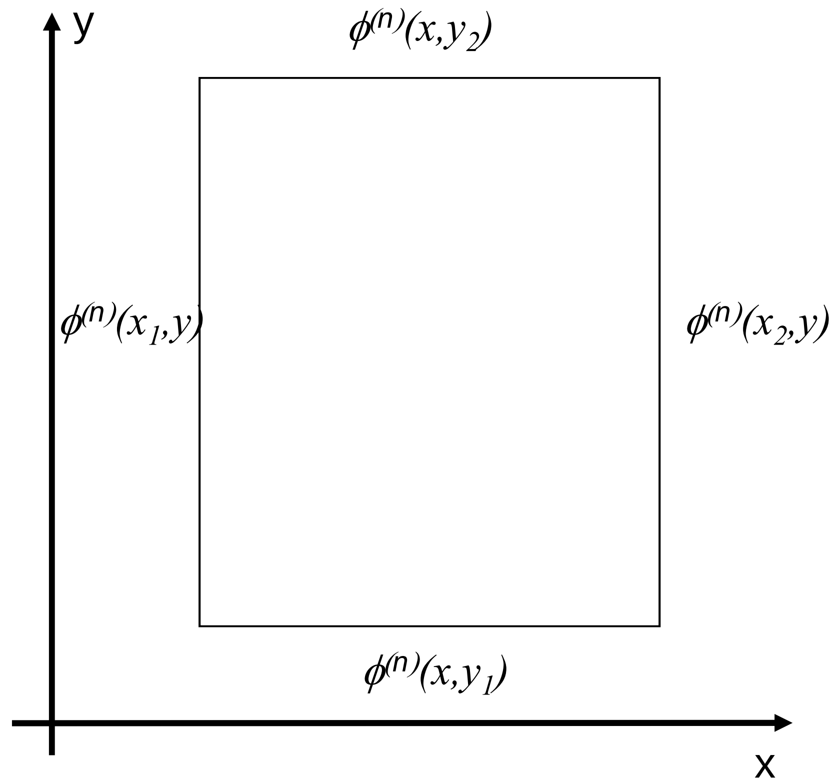
A,B and C Constants

$B^2 - 4AC > 0$ Hyperbolic

$B^2 - 4AC = 0$ Parabolic

$B^2 - 4AC < 0$ Elliptic

(Only valid for two independent variables: x,y)



- In general: A, B and C are function of: $x, y, \phi, \phi_x, \phi_y$
- Equations may change of type from point to point if A, B and C vary with $x, y,$. etc
- Navier-Stokes, incomp., const. viscosity: $\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\,\nabla^2\mathbf{u} + \mathbf{g}$

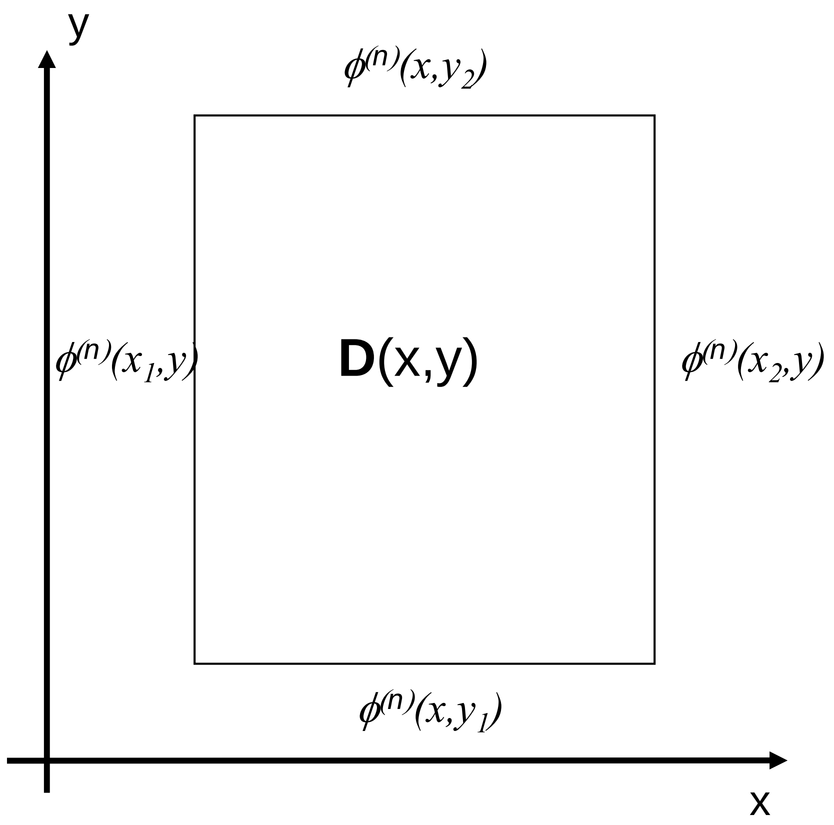
Partial Differential Equations
ELLIPTIC: $B^2 - 4AC < 0$

Quasi-linear PDE for $\phi(x, y)$

$$A\phi_{xx} + B\phi_{xy} + C\phi_{yy} = F(x, y, \phi, \phi_x, \phi_y)$$

A,B and C Constants

$B^2 - 4AC$	> 0	Hyperbolic
$B^2 - 4AC$	$= 0$	Parabolic
$B^2 - 4AC$	$\boxed{< 0}$	Elliptic



Partial Differential Equations
Elliptic PDE

Laplace Operator $\nabla^2 \equiv u_{xx} + u_{yy}$

Examples:

$$\nabla^2 u = 0$$

Laplace Equation – Potential Flow

$$\nabla^2 u = g(x,y)$$

Poisson Equation
- Potential Flow with sources
- Heat flow in plate

$$\nabla^2 u + f(x,y)u = 0$$

Helmholtz equation – Vibration of plates

$$\mathbf{U} \cdot \nabla \mathbf{u} = \nu \nabla^2 \mathbf{u}$$

Convection-Diffusion

- Smooth solutions ("diffusion effect")
- Very often, steady state problems
- Domain of dependence of u is the full domain **D**(x,y) => "global" solutions
- Finite differ./volumes/elements, boundary integral methods (Panel methods)

Partial Differential Equations
Elliptic PDEs

$$0 \le x \le a\ ,\ \ 0 \le y \le b;$$

Equidistant Sampling

$$\begin{aligned} h &= a/(n-1) \\ h &= b/(m-1) \end{aligned}$$

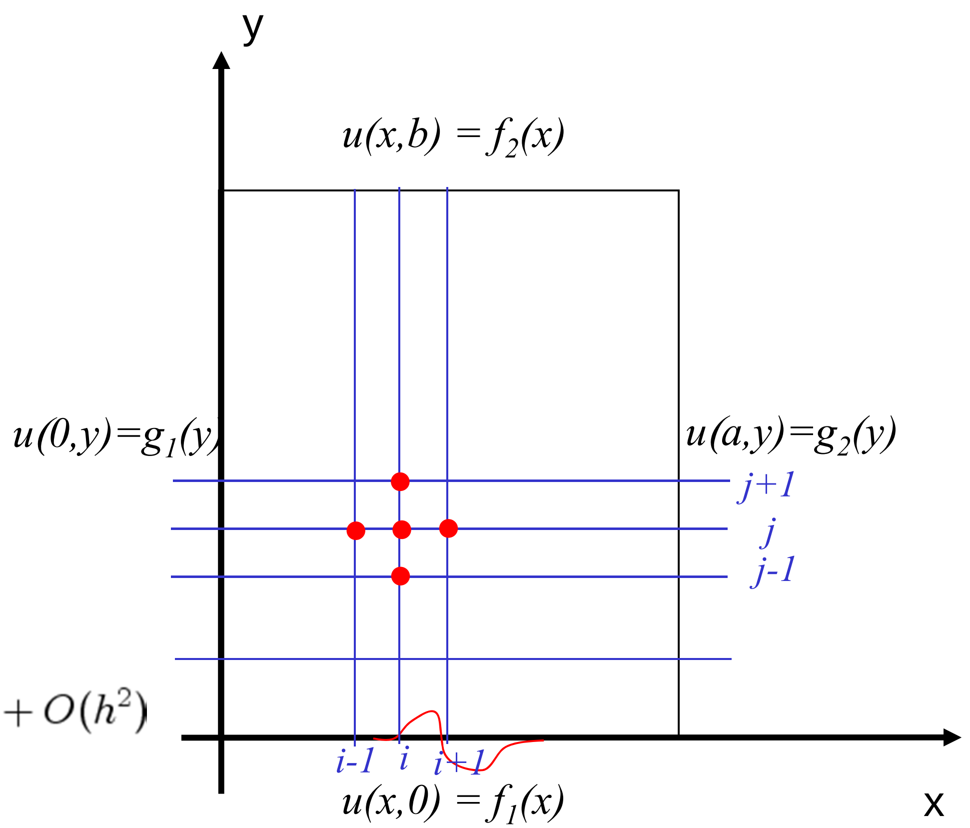
Discretization

$$\begin{aligned} x_i &= (i-1)h,\ i = 1, \ldots, n \\ y_j &= (j-1)h,\ j = 1, \ldots, m \end{aligned}$$

Finite Differences

$$u_{xx}(x,t) = \frac{u(x_{i-1},y_j) - 2u(x_i,y_j) + u(x_{i+1},y_j)}{h^2} + O(h^2)$$

$$u_{yy}(x,t) = \frac{u(x_i,y_{j-1}) - 2u(x_i,y_j) + u(x_i,y_{j+1})}{h^2} + O(h^2)$$



Partial Differential Equations
Elliptic PDE

Discretized Laplace Equation

$$\nabla^2 u = \frac{u(x_{i-1},y_j) + u(x_i,y_{j-1}) - 4u(x_i,y_j) + u(x_{i+1},y_j) + u(x_i,y_{j+1})}{h^2} = 0$$

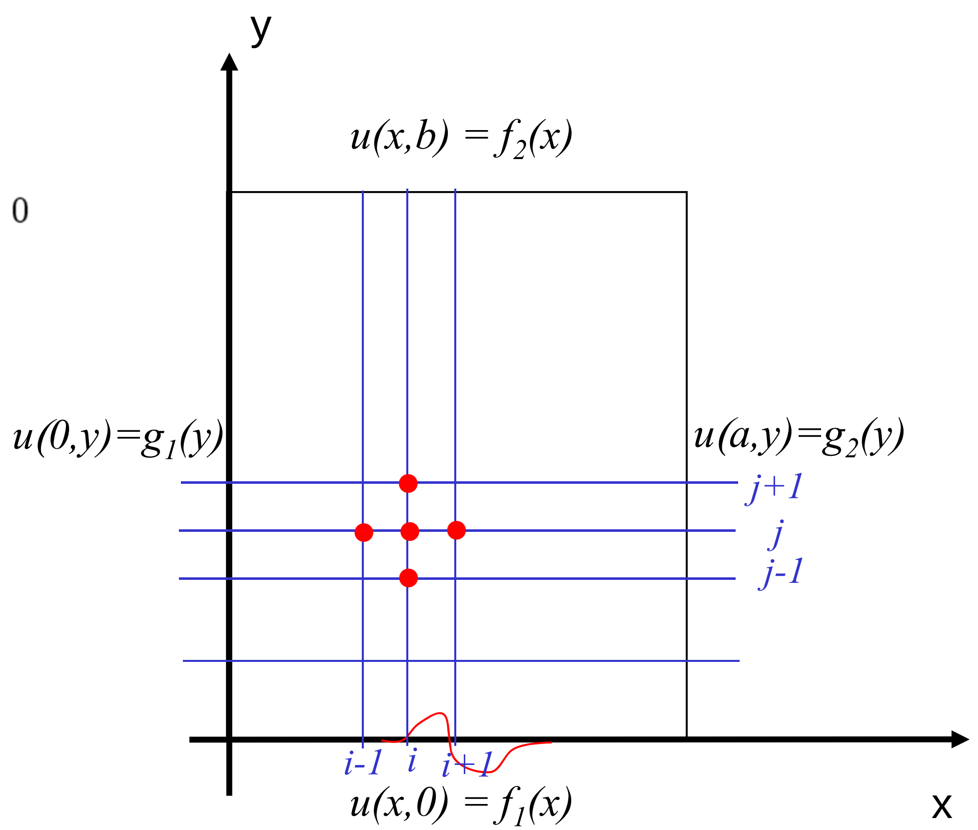
$$u_{i,j} = u(x_i,t_j)$$

Finite Difference Scheme

$$u_{i+1,j} + u_{i-1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = 0$$

Boundary Conditions

$$\begin{aligned}
u(x_1,y_j) &= u_{1,j}\ , \quad 2 \le j \le m-1 \\
u(x_n,y_j) &= u_{n,j}\ , \quad 2 \le j \le m-1 \\
u(x_i,y_1) &= u_{i,1}\ , \quad 2 \le i \le n-1 \\
u(x_i,y_n) &= u_{i,n}\ , \quad 2 \le i \le n-1
\end{aligned}$$



$\boxed{\text{Global Solution Required}}$

Elliptic PDE:
Poisson Equation

$$\nabla^2 u = g(x,y)$$

$$g_{i,j} = g(x_i, y_j)$$

SOR Iterative Scheme, with Jacobi

$$\begin{aligned}
u_{i,j}^{k+1} &= u_{i,j}^{k} + \omega r_{i,j}^{k} \\
&= u_{i,j}^{k} + \omega \frac{u_{n+1,j}^{k} + u_{n-1,j}^{k} + u_{n,j+1}^{k} + u_{n,j-1}^{k} - 4u_{n,j}^{k} - h^2 g_{i,j}}{4} \\
&= \boxed{(1-\omega)u_{i,j}^{k} + \omega \frac{u_{n+1,j}^{k} + u_{n-1,j}^{k} + u_{n,j+1}^{k} + u_{n,j-1}^{k} - h^2 g_{i,j}}{4}}
\end{aligned}$$

Partial Differential Equations

Hyperbolic PDE: $B^2 - 4\,A\,C > 0$

Examples:

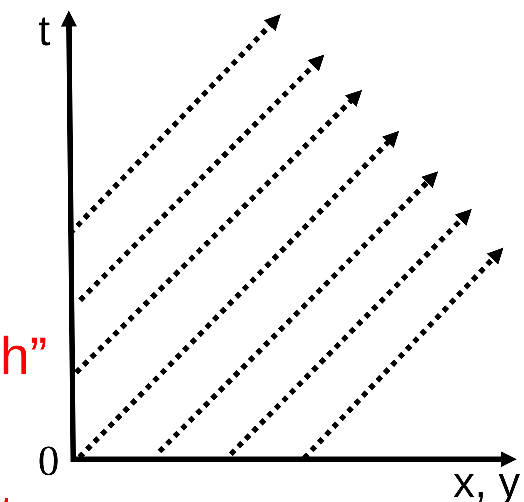
(1) $\dfrac{\partial^2 u}{\partial t^2} = c^2 \dfrac{\partial^2 u}{\partial x^2}$ ← Wave equation, 2nd order

(2) $\dfrac{\partial u}{\partial t} \pm c \dfrac{\partial u}{\partial x} = 0$ ← Sommerfeld Wave/radiation equation, 1st order

(3) $\dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{U} \cdot \nabla)\,\mathbf{u} = \mathbf{g}$ ← Unsteady (linearized) inviscid convection (Wave equation first order)

(4) $(\mathbf{U} \cdot \nabla)\,\mathbf{u} = \mathbf{g}$ ← Steady (linearized) inviscid convection

- Allows non-smooth solutions
- Information travels along characteristics, e.g.:
  - For (3) above: $\dfrac{d\,\mathbf{x_c}}{dt} = \mathbf{U}(\mathbf{x_c}(t))$
  - For (4), along streamlines: $\dfrac{d\,\mathbf{x_c}}{ds} = \mathbf{U}$
- Domain of dependence of $\mathbf{u}(\mathbf{x},T)$ = "characteristic path"
  - e.g., for (3), it is: $\mathbf{x}_c(t)$ for $0 < t < T$
- Finite Differences, Finite Volumes and Finite Elements



Partial Differential Equations
Hyperbolic PDE

Waves on a String

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2} \qquad 0 < x < L, \quad 0 < t < \infty$$

Initial Conditions

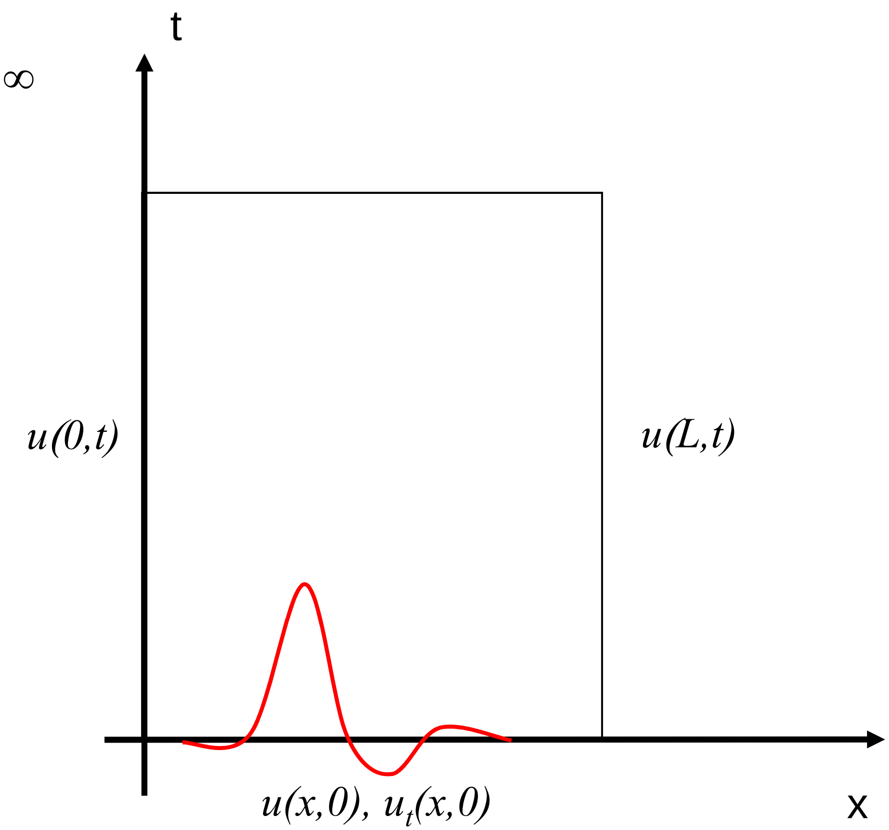
$$\begin{aligned} u(x,0) &= f(x), \;\; 0 \leq x \leq L \\ u_t(x,0) &= g(x), \;\; 0 < x < L \end{aligned}$$

Boundary Conditions

$$\begin{aligned} u(0,t) &= 0, \;, \; 0 < t < \infty \\ u(L,t) &= 0, \;, \; 0 < t < \infty \end{aligned}$$

Wave Solutions

$$u = \begin{cases} F(x-ct) & \text{Forward propagating wave} \\ G(x+ct) & \text{Backward propagating wave} \end{cases}$$



Typically Initial Value Problems in Time, Boundary Value Problems in Space
Time-Marching Solutions: Explicit Schemes Generally Stable

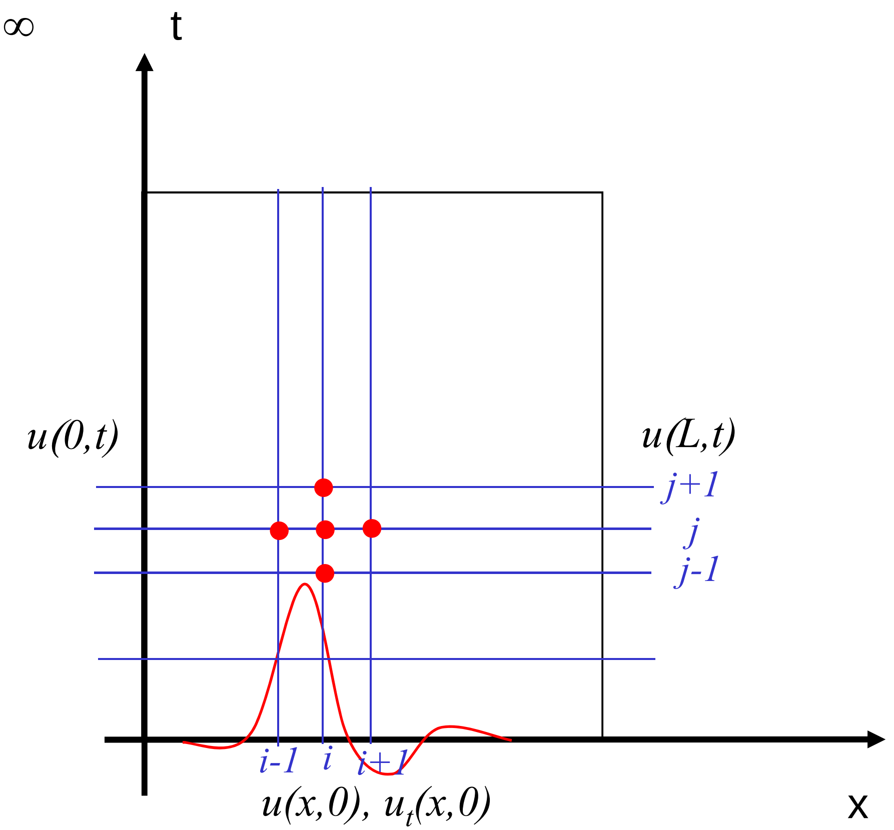
Partial Differential Equations
Hyperbolic PDE

Wave Equation

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2} \qquad 0 < x < L, \quad 0 < t < \infty$$

Discretization:

$$\begin{aligned} h &= L/n \\ k &= T/m \\ x_i &= (i-1)h, \; i = 2, \ldots, n-1 \\ t_j &= (j-1)k, \; j = 1, \ldots, m \end{aligned}$$



Finite Difference Representations

$$u_{tt}(x,t) = \frac{u(x_i, t_{j-1}) - 2u(x_i, t_j) + u(x_i, t_{j+1})}{k^2} + O(k^2)$$

$$u_{xx}(x,t) = \frac{u(x_{i-1}, t_j) - 2u(x_i, t_j) + u(x_{i+1}, t_j)}{h^2} + O(h^2)$$

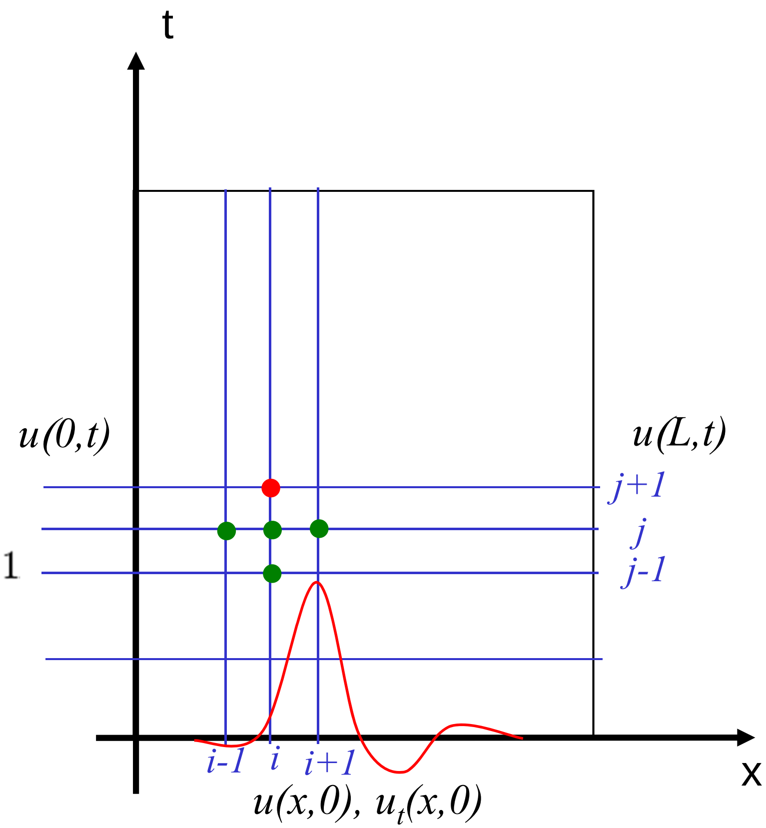
$$u_{i,j} = u(x_i, t_j)$$

Finite Difference Representations

$$\frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{k^2} = c^2 \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2}$$

Partial Differential Equations
Hyperbolic PDE

Introduce Dimensionless Wave Speed $C = \frac{ck}{h}$



Explicit Finite Difference Scheme

$$u_{i,j-1} - 2u_{i,j} + u_{i,j+1} = C^2(u_{i-1,j} - 2u_{i,j} + u_{i+1,j})$$

$$\boxed{u_{i,j+1}} = \boxed{(2-2C^2)u_{i,j} + C^2(u_{i+1,j} + u_{i-1,j}) - u_{i,j-1}} \quad i = 2, \ldots n-1$$

Stability Requirement: $C = \frac{ck}{h} < 1$

$$C = \frac{c\,\Delta t}{\Delta x} < 1 \quad \text{Courant-Friedrichs-Lewy condition (CFL condition)}$$

Physical wave speed must be smaller than the largest numerical wave speed, or,
Time-step must be less than the time for the wave to travel to adjacent grid points:

$$c < \frac{\Delta x}{\Delta t} \quad \text{or} \quad \Delta t < \frac{\Delta x}{c}$$

Error Types and Discretization Properties: Consistency

Consider the differential equation (L symbolic operator)

$$\boxed{L(\phi) = 0}$$

and its discretization for any given difference scheme

$$\boxed{\hat{L}_{\Delta x}(\hat{\phi}) = 0}$$

❖ **Consistency** (Property of the discretization)

– The discretization of a PDE should asymptote to the PDE itself as the mesh-size/time-step goes to zero, i.e

for all smooth functions ϕ: $\boxed{\left| L(\phi) - \hat{L}_{\Delta x}(\phi) \right| \rightarrow 0 \quad \text{when } \Delta x \rightarrow 0}$

(the truncation error vanishes as mesh-size/time-step goes to zero)

Error Types and Discretization Properties: Truncation error and Error equation

❖ Truncation error

$$\boxed{\tau_{\Delta x} = L(\phi) - \hat{L}_{\Delta x}(\phi)}$$

Remember:
ϕ does not satisfy the FD eqn.

- Since $L(\phi) = 0$, the truncation error is the result of inserting the exact solution in the difference scheme
- If the FD scheme is consistent: $\boxed{\tau_{\Delta x} = L(\phi) - \hat{L}_{\Delta x}(\phi) \rightarrow O(\Delta x^p) \quad \text{for} \;\; \Delta x \rightarrow 0}$
- p (>0) is the order of accuracy for the FD scheme $\hat{L}_{\Delta x}$
- Order p indicates how fast the error is **reduced** when the grid is **refined**

❖ Error evolution equation

- From $\hat{L}_{\Delta x}(\hat{\phi}) = 0$ and $\phi = \hat{\phi} + \varepsilon$ where ε is the discretization error, for linear problems, we have: $\tau_{\Delta x} = L(\phi) - \hat{L}_{\Delta x}(\hat{\phi} + \varepsilon) = -\hat{L}_{\Delta x}(\varepsilon)$

$$\Rightarrow \hat{L}_{\Delta x}(\varepsilon) = -\tau_{\Delta x}$$

- The truncation error acts as a source for the discretization error, which is convected and diffused by the operator $\hat{L}_{\Delta x}$

Error Types and Discretization Properties: Stability

❖ Stability

- A numerical solution scheme is said to be stable if it does not amplify errors that appear in the course of the numerical solution process
- For linear(-ized) problems, since $\hat{L}_{\Delta x}(\varepsilon) = -\tau_{\Delta x}$, stability implies:

  $\boxed{\left\| \hat{L}_{\Delta x}^{-1} \right\| < \text{Const.}}$ with the Const. not a function of Δx

  - If inverse was not bounded, discretization errors ε would increase with iterations
- In practice, infinite norm $\left\| \hat{L}_{\Delta x}^{-1} \right\|_{\infty} < \text{Const.}$ is often used.
- However, difficult to assess stability in real cases due to boundary conditions and non-linearities
  - It is common to investigate stability for linear problems, with constant coefficients and without boundary conditions
  - A widely used approach: von Neumann's method (see lectures 13-14)

Error Types and Discretization Properties: Convergence

❖ Convergence

– A numerical scheme is said to be convergent if the solution of the discretized equations tend to the exact solution of the (P)DE as the grid-spacing and time-step go to zero

– Error equation for linear(-ized) systems: $\varepsilon = -\hat{L}_{\Delta x}^{-1}(\tau_{\Delta x})$

– Error bounds for linear systems:

$$\|\varepsilon\| = \left\|\hat{L}_{\Delta x}^{-1}(\tau_{\Delta x})\right\| \le \left\|\hat{L}_{\Delta x}^{-1}\right\| \; \|\tau_{\Delta x}\|$$

$$\text{For a consistent scheme: } \|\tau_{\Delta x}\| \to O(\Delta x^p) \quad \text{for } \; \Delta x \to 0$$

$$\text{Hence } \|\varepsilon\| \le \left\|\hat{L}_{\Delta x}^{-1}\right\| \; \|\tau_{\Delta x}\| \le \alpha \, O(\Delta x^p)$$

Convergence <= Stability + Consistency (for linear systems)

= Lax Equivalence Theorem (for linear systems)

– For nonlinear equations, numerical experiments are often used
  - e.g., iterate or approximate true solution with computation on successively finer grids, and compute resulting discretization errors and order of convergence

Finite Differences - Basics

- Finite Difference Approximation idea directly borrowed from the definition of a derivative.

$$\phi'(x_i) = \lim_{\Delta x \to 0} \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x}$$

- Geometrical Interpretation
  - Quality of approximation improves as stencil points get closer to x_{i}
  - Central difference would be exact if ϕ was a second order polynomial and points were equally spaced

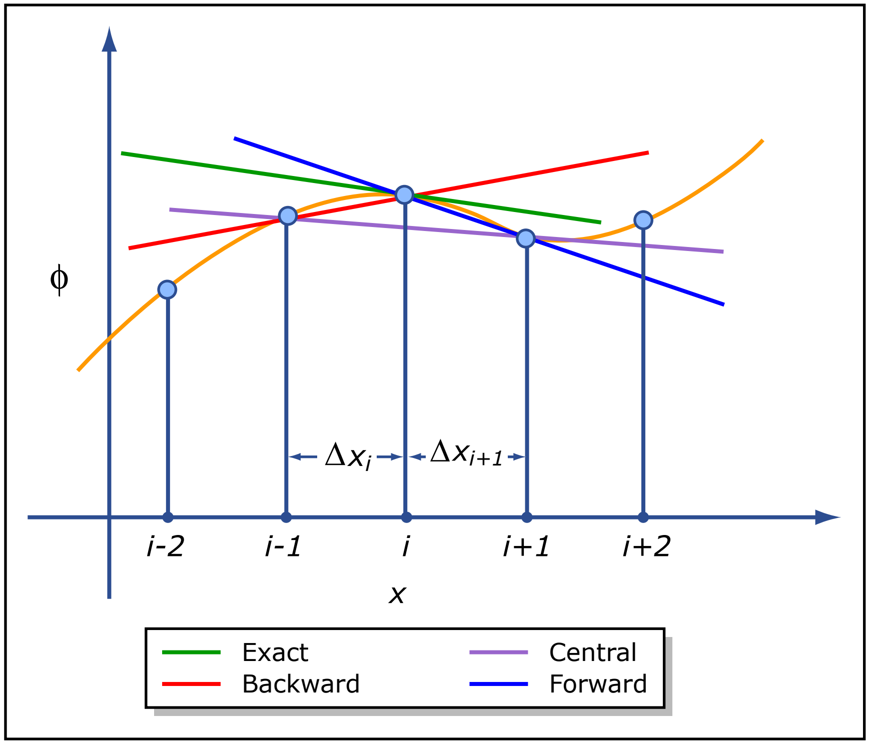


Image by MIT OpenCourseWare.

FINITE DIFFERENCES: Taylor Series, Higher Order Accuracy

How to obtain differentiation formulas of arbitrary high accuracy?

1) First approach: Use Taylor series, introducing higher-order terms

$$f(x_{i+1}) = f(x_i) + \Delta x\, f'(x_i) + \frac{\Delta x^2}{2!} f''(x_i) + \frac{\Delta x^3}{3!} f'''(x_i) + \ldots + \frac{\Delta x^n}{n!} f^n(x_i) + R_n$$

$$R_n = \frac{\Delta x^{n+1}}{n+1!} f^{(n+1)}(\xi)$$

- For example, how can we derive the forward finite-difference estimate of the first derivative at x_i with second order accuracy?

$$\left. f(x_{i+1}) = f(x_i) + \Delta x\, f'(x_i) + \frac{\Delta x^2}{2!} f''(x_i) + O(\Delta x^3) \right\} \longrightarrow f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{\Delta x} - \frac{\Delta x}{2!} \underline{f''(x_i)} + O(\Delta x^2)$$

- If we retain the second-derivative, and estimate it with first-order accuracy, the order of accuracy for the estimate of $f'(x_i)$ will be $p=2$

FINITE DIFFERENCES:
Taylor Series, Higher Order Accuracy Cont'd

- Using $$f(x_{i+1}) = f(x_i) + \Delta x\, f'(x_i) + \frac{\Delta x^2}{2!} f''(x_i) + \frac{\Delta x^3}{3!} f'''(x_i) + \ldots + \frac{\Delta x^n}{n!} f^n(x_i) + R_n$$

$$R_n = \frac{\Delta x^{n+1}}{n+1!} f^{(n+1)}(\xi)$$

- Estimate the second-derivative with forward finite-differences at first-order accuracy:

$$\left.\begin{aligned} f(x_{i+1}) &= f(x_i) + \Delta x\, f'(x_i) + \frac{\Delta x^2}{2!} f''(x_i) + O(\Delta x^3) \\ f(x_{i+2}) &= f(x_i) + 2\Delta x\, f'(x_i) + \frac{4\Delta x^2}{2!} f''(x_i) + O(\Delta x^3) \end{aligned}\right\} \left.\begin{matrix} *(-2) \\ *(1) \end{matrix}\right\} \Rightarrow f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{\Delta x^2} + O(\Delta x)$$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{\Delta x} - \frac{\Delta x}{2!} f''(x_i) + O(\Delta x^2)$$

$$\Rightarrow \boxed{\underline{f'(x_i)} = \frac{f(x_{i+1}) - f(x_i)}{\Delta x} - \frac{\Delta x}{2!} \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{\Delta x^2} + O(\Delta x^2) = \underline{\frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2\Delta x} + O(\Delta x^2)}}$$

Forward Differences

Forward finite-divided-difference formulas:

First Derivative | **Error**

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} \qquad O(h)$$

$$f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h} \qquad O(h^2)$$

Second Derivative

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} \qquad O(h)$$

$$f''(x_i) = \frac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2} \qquad O(h^2)$$

Third Derivative

$$f'''(x_i) = \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{h^3} \qquad O(h)$$

$$f'''(x_i) = \frac{-3f(x_{i+4}) + 14f(x_{i+3}) - 24f(x_{i+2}) + 18f(x_{i+1}) - 5f(x_i)}{2h^3} \qquad O(h^2)$$

Backward Differences

Backward finite-divided-difference formulas:

First Derivative **Error**

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} \qquad O(h)$$

$$f'(x_i) = \frac{3f(x_i) - 4f(x_{i-1}) + 3f(x_{i-2})}{2h} \qquad O(h^2)$$

Second Derivative

$$f''(x_i) = \frac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2} \qquad O(h)$$

$$f''(x_i) = \frac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3})}{h^2} \qquad O(h^2)$$

Third Derivative

$$f'''(x_i) = \frac{f(x_i) - 3f(x_{i-1}) + 3f(x_{i-2}) - f(x_{i-3})}{h^3} \qquad O(h)$$

$$f'''(x_i) = \frac{5f(x_i) - 18f(x_{i-1}) + 24f(x_{i-2}) - 14f(x_{i-3}) + 3f(x_{i-4})}{2h^3} \qquad O(h^2)$$

Centered Differences

Centered finite-divided-difference formulas:

Error

First Derivative

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

O(h)

$$f'(x_i) = \frac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})}{12h}$$

O(h^2)

Second Derivative

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}$$

O(h)

$$f''(x_i) = \frac{-f(x_{i+2}) + 16f(x_{i+1}) - 30f(x_i) + 16f(x_{i-1}) - f(x_{i-2})}{12h^2}$$

O(h^2)

Third Derivative

$$f'''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2})}{2h^3}$$

O(h)

$$f'''(x_i) = \frac{-f(x_{i+3}) + 8f(x_{i+2}) - 13f(x_{i+1}) + 13f(x_{i-1}) - 8f(x_{i-2}) + f(x_{i-3})}{8h^3}$$

O(h^2)

FINITE DIFFERENCES
Taylor Series, Higher Order Accuracy: **EXAMPLE**

Problem: Estimate 1st derivative of f = -0.1*x^4 - 0.15*x^3-0.5*x^2-0.25*x +1.2 at x=0.5, with a step size of 0.25 and using successively higher order schemes. How does the solution improve?

L11_FD.m

```
%Define the function
f=@(x) -0.1*x^4 - 0.15*x^3-0.5*x^2-0.25*x +1.2;
%Define Step size
h=0.25;
%Set point at which to evaluate the derivative
x = 0.5;
%% Using forward difference
%First order:
df=(f(x+h)-f(x)) / h;
fprintf('\n\n First order Forward difference: %g, with error:%g%% \n',df,abs(100*(df+0.9125)/0.9125))
%Second order:
df=(-f(x+2*h)+4*f(x+h)-3*f(x)) / (2*h);
fprintf('Second order Forward difference: %g, with error:%g%% \n',df,abs(100*(df+0.9125)/0.9125))
%% Backwards difference
%First order:
df=(-f(x-h)+f(x)) / (h);
fprintf('First order Backwards difference: %g, with error:%g%% \n',df,abs(100*(df+0.9125)/0.9125))
%Second order:
df=(f(x-2*h)-4*f(x-h)+3*f(x)) / (2*h);
fprintf('Second order Backwards difference: %g, with error:%g%% \n',df,abs(100*(df+0.9125)/0.9125))
```

```
%% Central difference
%Second order:
df=(f(x+h)-f(x-h)) / (2*h);
fprintf('Second order Central difference: %g, with error:%g%% \n',df,abs(100*(df+0.9125)/0.9125))
%Fourth order:
df=(-f(x+2*h)+8*f(x+h)-8*f(x-h)+f(x-2*h)) / (12*h);
fprintf('Fourth order Central difference: %g, with error:%g%% \n',df,abs(100*(df+0.9125)/0.9125))
```

Output

First order Forward difference: -1.15469, with error:26.5411%

Second order Forward difference: -0.859375, with error:5.82192%

First order Backwards difference: -0.714063, with error:21.7466%

Second order Backwards difference: -0.878125, with error:3.76712%

Second order Central difference: -0.934375, with error:2.39726%

Fourth order Central difference: -0.9125, with error:2.43337e-014%

Why is the 4th order "exact"?

FINITE DIFFERENCES: Taylor Series, Higher Order Accuracy Summary

- Approach:
  - Incorporate more higher-order terms of the Taylor series expansion than strictly needed and express them as finite differences themselves
    - e.g. for finite difference of m^{th} derivative at order of accuracy p, express the $m+1^{\text{th}}$, $m+2^{\text{th}}$, $m+p-1^{\text{th}}$ derivatives at an order of accuracy $p-1, .., 2, 1$.
  - General approximation: $$\boxed{\left(\frac{\partial^m u}{\partial x^m}\right)_j - \sum_{i=-r}^{s} a_i \, u_{j+i} = \tau_{\Delta x}}$$
  - Can be used for forward, backward, skewed or central differences
  - Can be computer automated
  - Independent of coordinate system and extends to multi-dimensional finite differences (each coordinate is usually treated separately)
- Remember: order p of approximation indicates how fast the error is **reduced** when the grid is **refined** (not the magnitude of the error)

FINITE DIFFERENCES: Interpolation Formulas for Higher Order Accuracy

2nd approach: Generalize Taylor series using interpolation formulas

- Fit the unknown function solution of the (P)DE to an interpolation curve and differentiate the resulting curve. For example:
  - Fit a parabola to data at points $x_{i-1}, x_i, x_{i+1} \quad (\Delta x_i = x_i - x_{i-1})$, then differentiate to obtain:

$$f'(x_i) = \frac{f(x_{i+1})\left(\Delta x_i\right)^2 - f(x_{i-1})\left(\Delta x_{i+1}\right)^2 + f(x_i)\left[\left(\Delta x_{i+1}\right)^2 - \left(\Delta x_i\right)^2\right]}{\Delta x_{i+1}\,\Delta x_i\,(\Delta x_i + \Delta x_{i+1})}$$

  - This is a 2nd order approximation
  - For uniform spacing, reduces to centered difference seen before
  - In general, approximation of first derivative has truncation error of the order of the polynomial
- All types of polynomials or numerical differentiation methods can be used to derive such interpolations formulas
  - Polynomial fitting, Method of undetermined coefficients, Newton's interpolating polynomials, Lagrangian and Hermite Polynomials, etc

2.29 Numerical Fluid Mechanics
Fall 2011