UNIVERSITY OF ARKANSAS

AM³ Lab
Advanced Manufacturing | Modeling | Materials

Additive Manufacturing – Module 8

Spring 2015

Wenchao Zhou

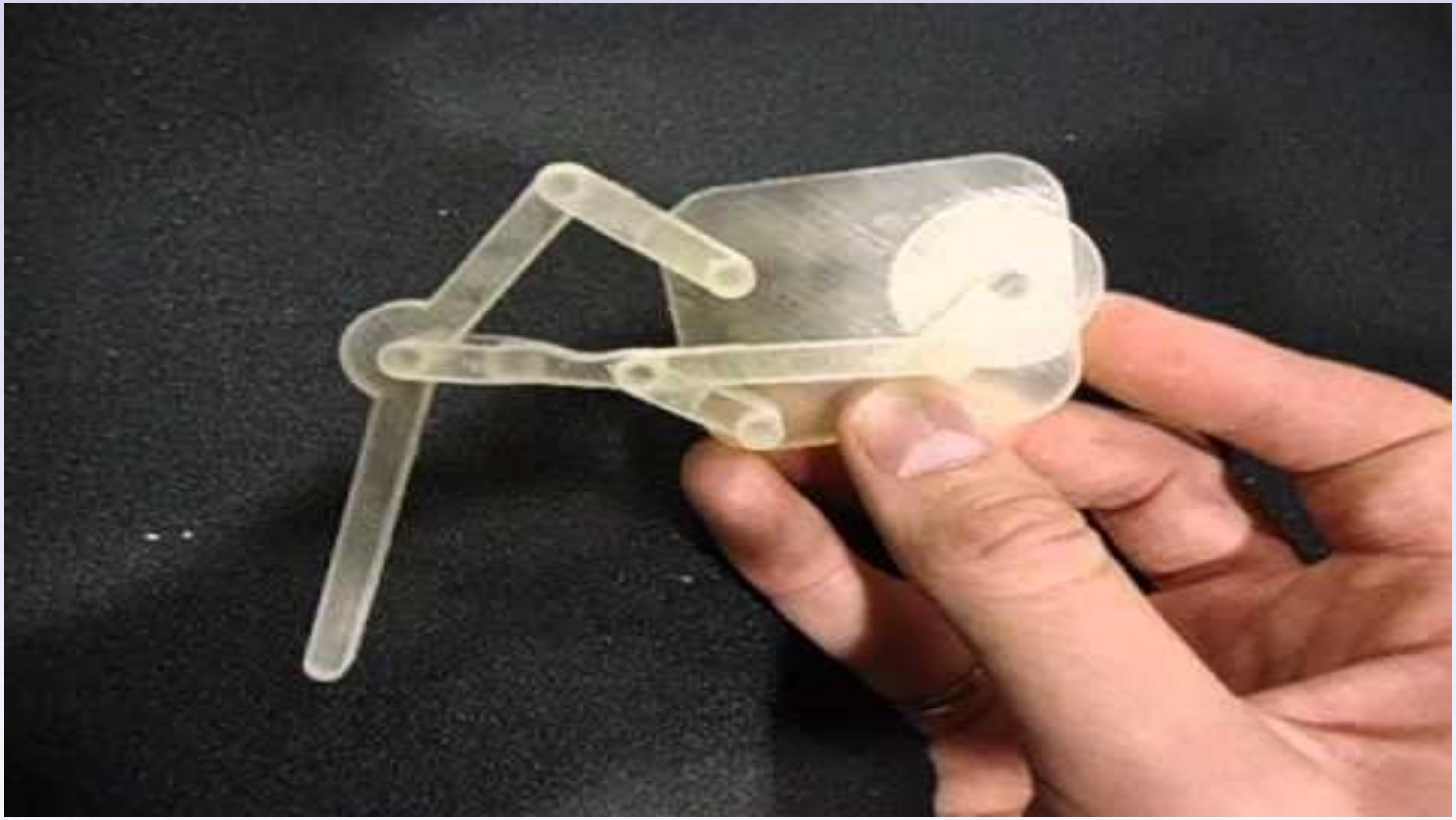
zhouw@uark.edu

(479) 575-7250

The Department of Mechanical Engineering
University of Arkansas, Fayetteville

CAE

Evaluating design

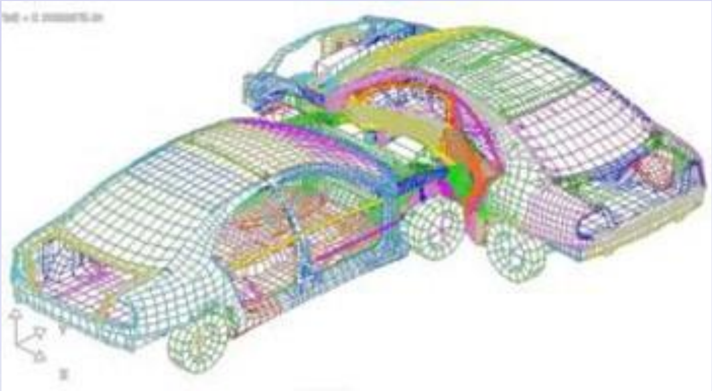


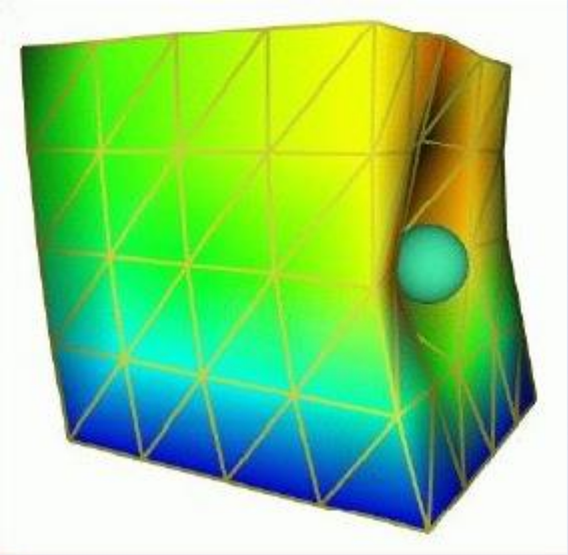
https://www.youtube.com/watch?v=p__-QbQbntI

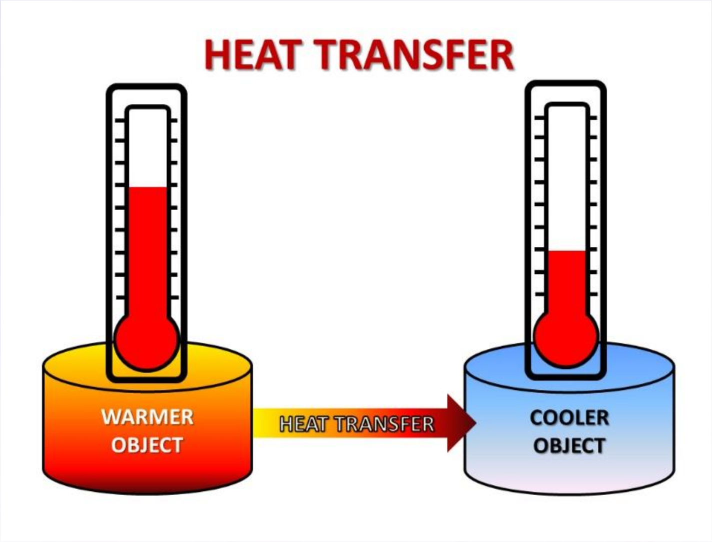
A design of a motion system – motion can be described using kinematic equations when approximated as rigid structures

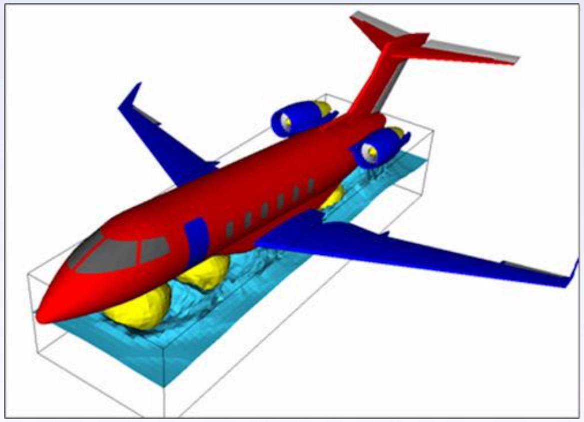
CAE

Evaluating design









What if your design is deformable structure, involves heat transfer, or fluids

CAE

CAE
FDM
FEM

Evaluating design

- Conservation Equations
  - Conservation of Mass
  - Conservation of Energy
  - Conservation of Momentum

⇩

- Partial Differential Equations
  - Describing change in space and time

- Constitutive Models for Materials
  - Hooke's Law
  - Newtonian Fluid
  - Etc.

CAE

Evaluating design

Millennium Prize Problems

P versus NP problem
Hodge conjecture
Poincaré conjecture (solved)
Riemann hypothesis
Yang–Mills existence and mass gap
Navier–Stokes existence and smoothness
Birch and Swinnerton-Dyer conjecture

V · T · E

Even much more basic properties of the solutions to Navier–Stokes have never been proven. For the three-dimensional system of equations, and given some initial conditions, **mathematicians have not yet proved that smooth solutions always exist**, or that if they do exist, they have bounded energy per unit mass.

What about higher order, higher dimension PDEs

$$F\left(x_1, \ldots, x_n, u, \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_n}, \frac{\partial^2 u}{\partial x_1 \partial x_1}, \ldots, \frac{\partial^2 u}{\partial x_1 \partial x_n}, \ldots\right) = 0.$$

Reality: Most PDEs CANNOT be solved analytically!!!

CAE

Evaluating design

- Numerical solutions
  - Discretize and turn PDEs into a system of algebraic equations (mostly linear)

Most popular methods

- Finite difference methods
- Finite element methods

Other methods

- Finite volume method
- Boundary element method
- Discrete element method
- Spectral method
- Particle based methods

http://en.wikipedia.org/wiki/List_of_numerical_analysis_topics#Numerical_methods_for_partial_differential_equations

CAE
FDM
FEM

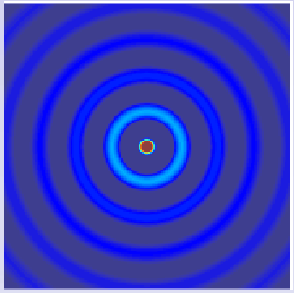
Classification of PDEs

Second-order linear PDEs

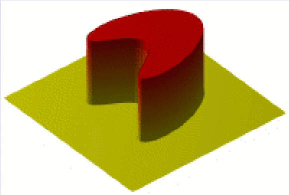
$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu + g = 0$$

are classified based on the value of the discriminant $b^2 - 4ac$

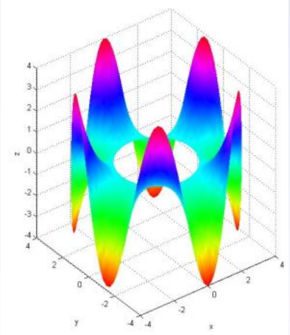
- $b^2 - 4ac > 0$: hyperbolic
  - e.g., wave equation: $u_{tt} - u_{xx} = 0$
  - Hyperbolic PDEs describe time dependent, conservative physical processes, such as convection, that are not evolving toward steady state.
- $b^2 - 4ac = 0$: parabolic
  - e.g., heat equation: $u_t - u_{xx} = 0$
  - Parabolic PDEs describe time-dependent dissipative physical processes, such as diffusion, that are evolving toward steady state.
- $b^2 - 4ac < 0$: elliptic
  - e.g., Laplace equation: $u_{xx} + u_{yy} = 0$
  - Elliptic PDEs describe processes that have already reached steady states, and hence are time-independent.



Wave equation



Heat equation



Laplace equation

Finite Difference Method

$$\partial_x f = \lim_{dx \to 0} \frac{f(x+dx) - f(x)}{dx}$$

Discretize

$$\partial_x f^{+} \approx \frac{f(x+dx) - f(x)}{dx}$$

Forward difference

$$\partial_x f^{-} \approx \frac{f(x) - f(x-dx)}{dx}$$

Backward difference

$$\partial_x f \approx \frac{f(x+dx) - f(x-dx)}{2dx}$$

Centered difference

Finite Difference Method

- **Consider a 1D initial-boundary value problem for heat equation**

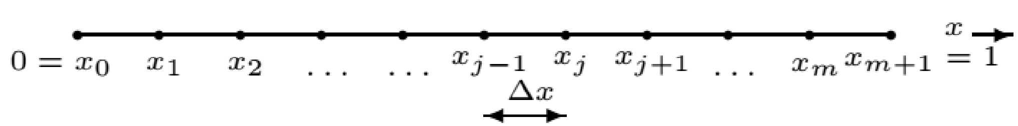
$$u_t = \kappa u_{xx},\ 0 \le x \le 1,\ t \ge 0$$

$$u(0, x) = f(x),\ \text{Initial Condition}$$

$$u(t, 0) = \alpha,\ \text{Boundary Condition at } x = 0$$

$$u(t, 1) = \beta,\ \text{Boundary Condition at } x = 1$$

Discretize the spatial domain $[0, 1]$ into $m + 2$ grid points using a uniform mesh step size $\boxed{\Delta x = 1/(m+1)}$. Denote the spatial grid points by $x_j, j = 0, 1, \ldots m + 1$.



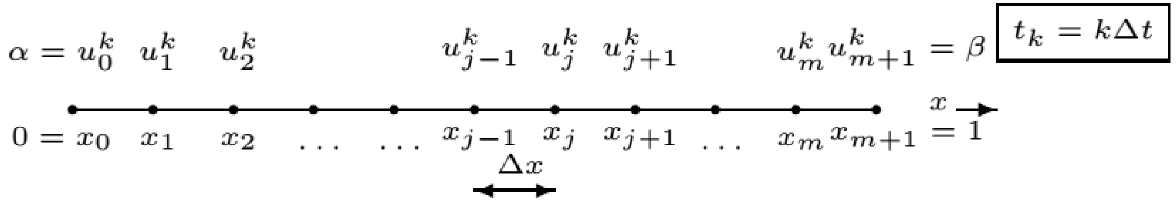
Credit: Vrushali A. Bokil and Nathan L. Gibson @ Oregon State U

Finite Difference Method

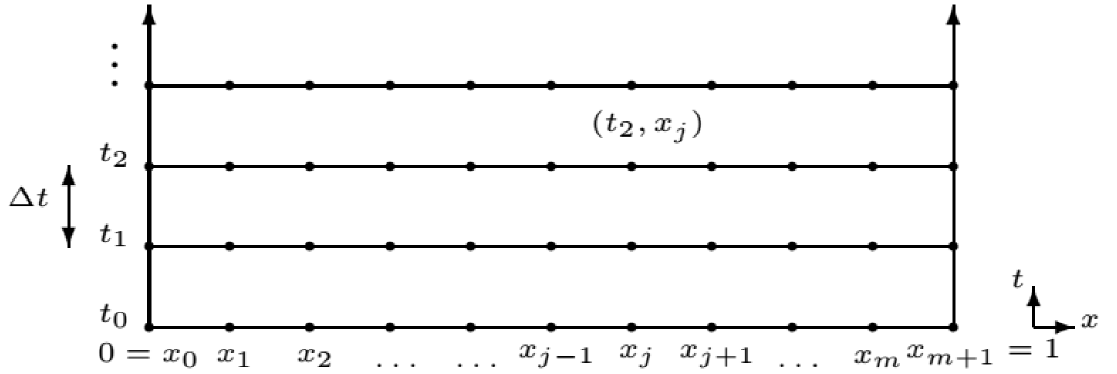
Consider a 1D initial-boundary value problem for heat equation

Similarly discretize the temporal domain into temporal grid points $t_k = k\Delta t$ for suitably chosen time step Δt.

Denote the approximate solution at the grid point (t_k, x_j) as U_j^k.



The space-time grid can be represented as



Credit: Vrushali A. Bokil and Nathan L. Gibson @ Oregon State U

Finite Difference Method

Consider a 1D initial-boundary value problem for heat equation

Replace u_t by a forward difference in time and u_{xx} by a central difference in space to obtain the explicit FDM

$$\frac{U_j^{k+1}-U_j^k}{\Delta t}=\kappa\frac{U_{j+1}^k-2U_j^k+U_{j-1}^k}{(\Delta x)^2}$$

$$\Longrightarrow U_j^{k+1}=U_j^k+\frac{\kappa\Delta t}{(\Delta x)^2}\left(U_{j+1}^k-2U_j^k+U_{j-1}^k\right),\ j=1,2,\ldots m$$

Using Taylor series to determine order of accuracy for the approximation

$$f(x\pm dx)=f(x)\pm \mathrm{dx}f'(x)+\frac{dx^2}{2!}f''(x)\pm\frac{dx^3}{3!}f'''(x)+\frac{dx^4}{4!}f''''(x)\pm\ldots$$

First order accurate in time

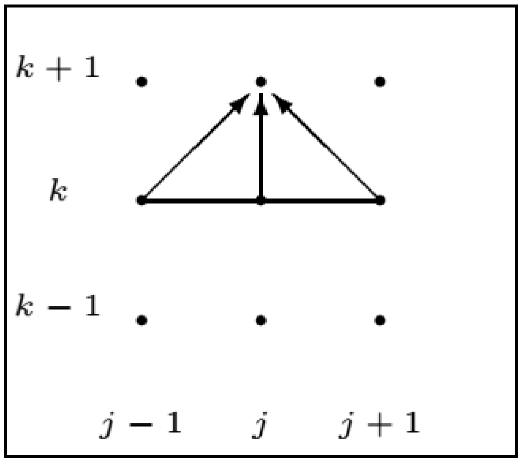
$$\frac{f(x+dx)-f(x)}{dx}=\frac{1}{dx}\left[\mathrm{dx}f'(x)+\frac{dx^2}{2!}f''(x)+\frac{dx^3}{3!}f'''(x)+\ldots\right]$$
$$=f'(x)+O(dx)$$

Second order accurate in space

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i=\frac{u_{i+1}-2u_i+u_{i-1}}{(\Delta x)^2}+O((\Delta x)^2)$$

Finite Difference Method

- **Consider a 1D initial-boundary value problem for heat equation**



Computational Stencil

The local truncation error is $O(\Delta t) + O((\Delta x)^2)$.

- **How to choose Δt and Δx?**

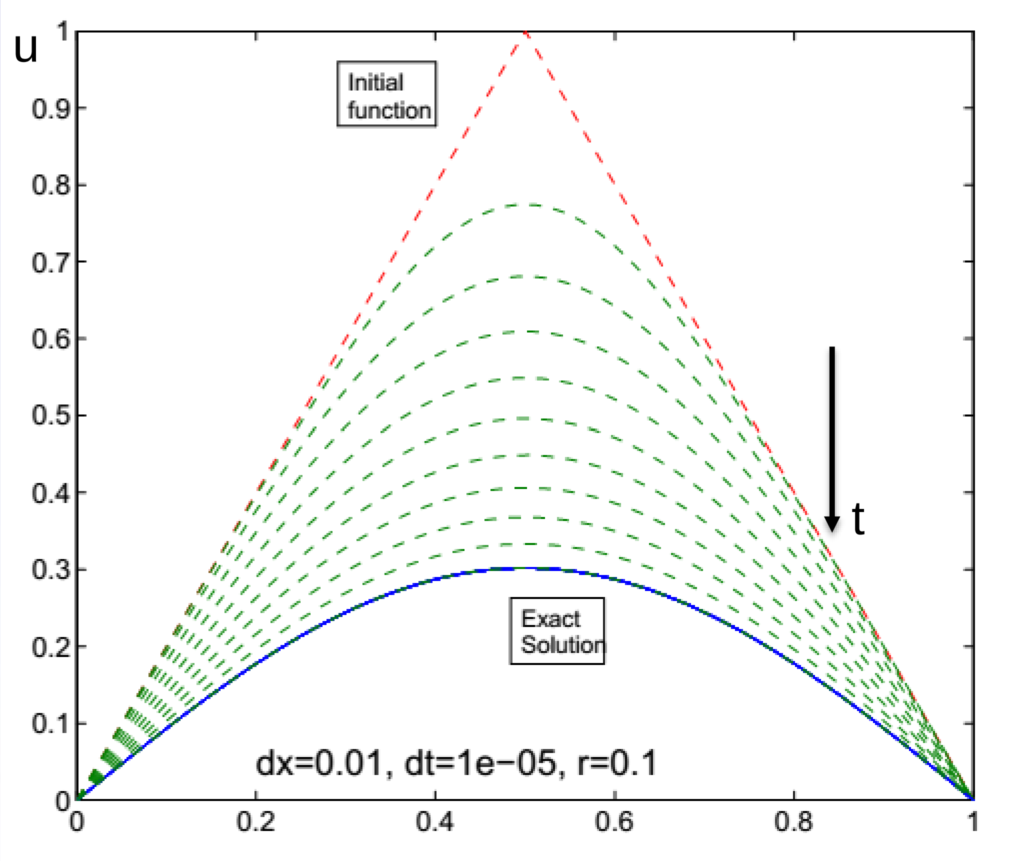
CAE

FDM

FEM

Finite Difference Method

- **How to choose Δt and Δx?**



$$r = \frac{\kappa \Delta t}{\Delta x} \leq \frac{1}{2}$$

With $\Delta x = 0.01$ and $\Delta t = 10^{-5}$

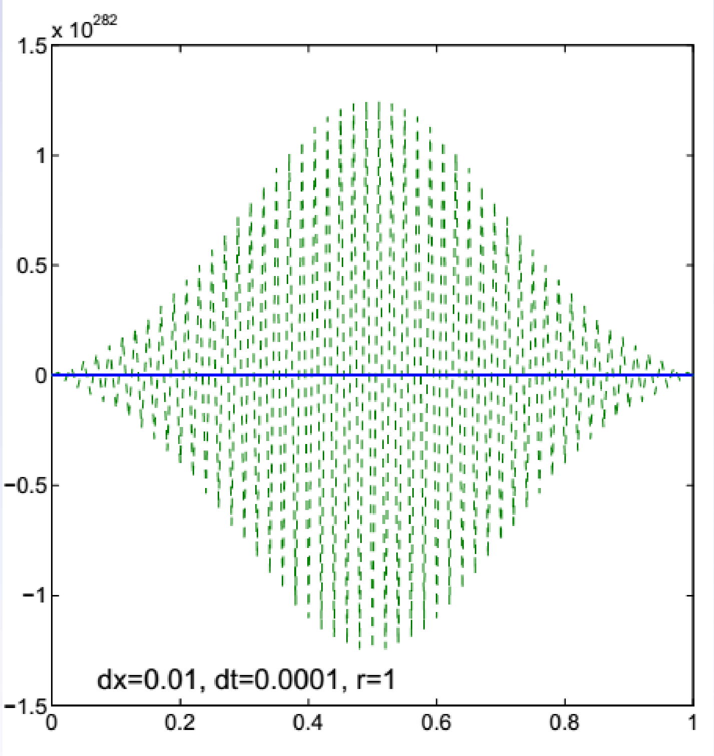
What happens if r is greater than 1/2?

- **Initial condition: discontinuous at x = 0.5**
- **Rapid smoothing of discontinuity as time evolves**
- **High frequency damps quickly. The heat equation is stiff**

Finite Difference Method

- **How to choose Δt and Δx?**

What happens if r is greater than 1/2?



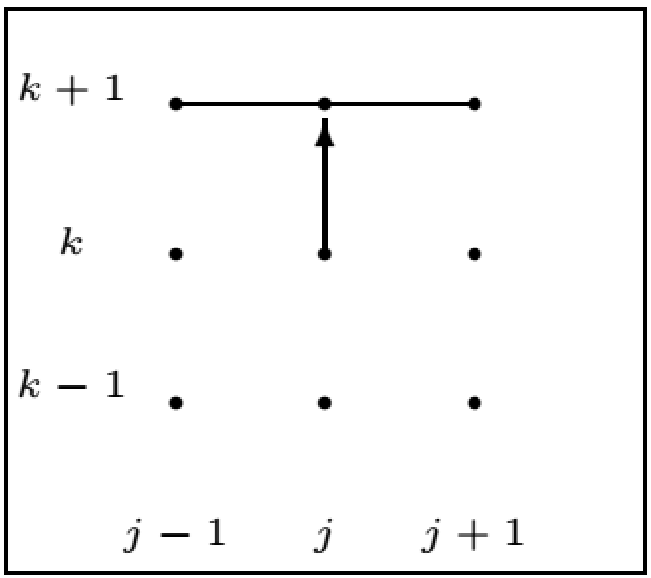
- **Unstable behavior of numerical solution**
- **Δt and Δx cannot be chosen arbitrarily. Must satisfy a stable condition.**

Finite Difference Method

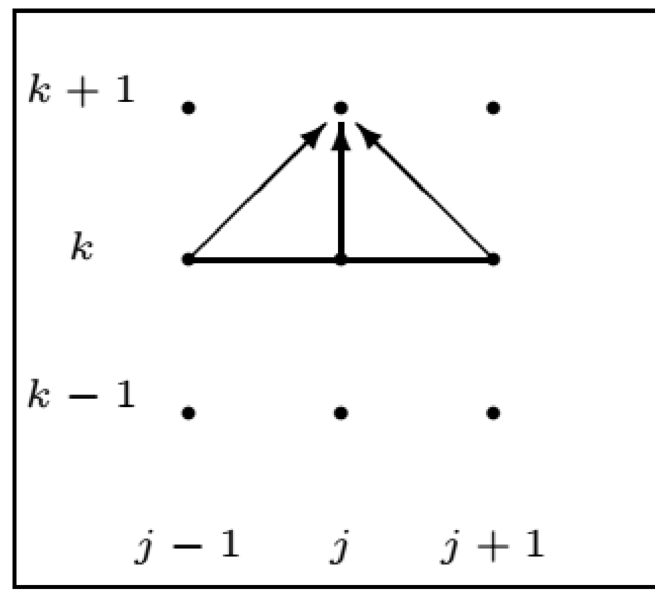
Implicit FDM

$$\frac{U_j^{k+1} - U_j^k}{\Delta t} = \kappa \frac{U_{j+1}^{k+1} - 2U_j^{k+1} + U_{j-1}^{k+1}}{(\Delta x)^2}$$

$$\Longrightarrow \quad U_j^{k+1} = U_j^k + \frac{\kappa \Delta t}{(\Delta x)^2}\left(U_{j+1}^{k+1} - 2U_j^{k+1} + U_{j-1}^{k+1}\right), \; j = 1, 2, \ldots m$$



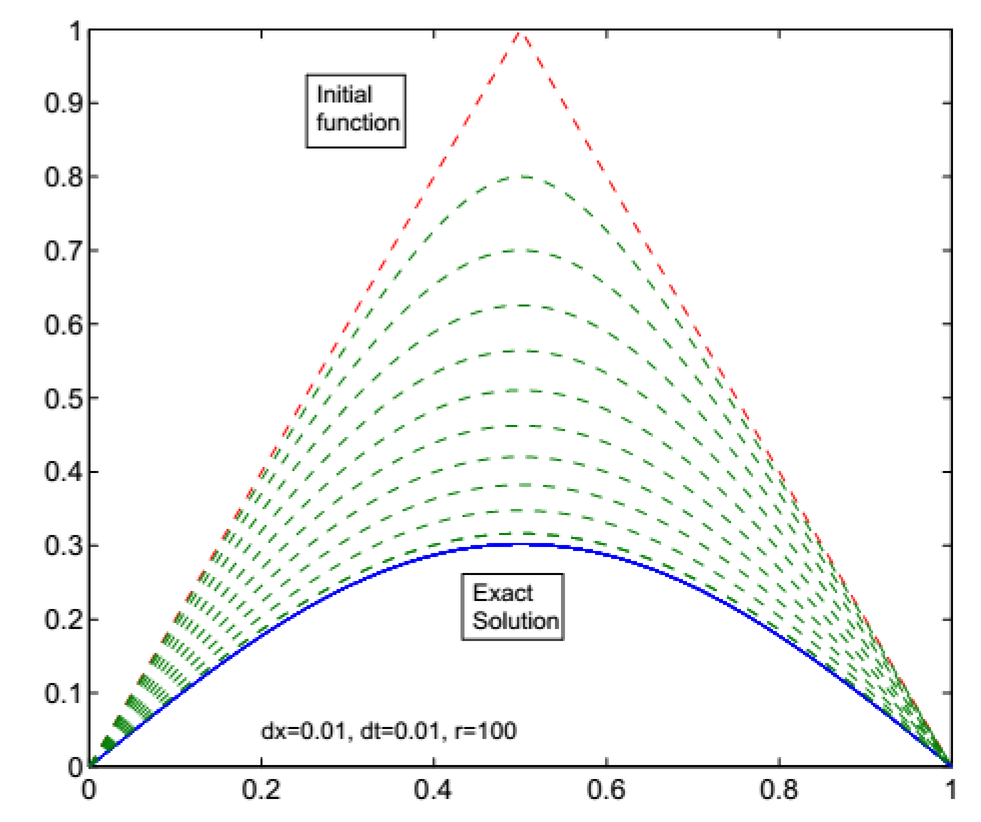
Implicit computational stencil



Explicit computational stencil

Finite Difference Method

Implicit FDM



- **Stable behavior of numerical solution**
- **Δt and Δx cannot be chosen to have the same order of magnitude. Unconditionally stable.**

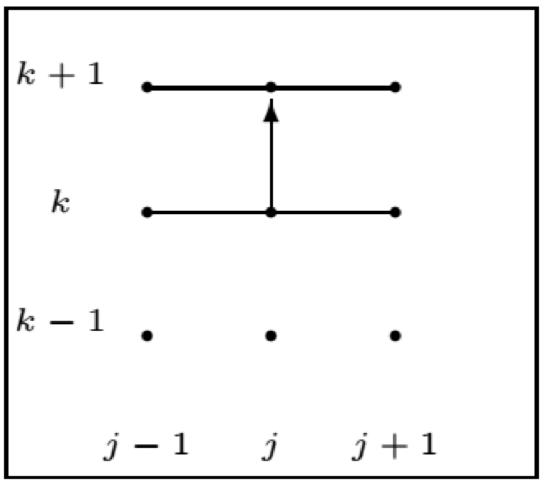
Finite Difference Method

- **Implicit FDM – 2nd order accurate in time – trapezoid rule**

$$\frac{U_j^{k+1}-U_j^k}{\Delta t}=\frac{\kappa}{2}\left(\frac{U_{j+1}^k-2U_j^k+U_{j-1}^k}{(\Delta x)^2}\right)+\frac{\kappa}{2}\left(\frac{U_{j+1}^{k+1}-2U_j^{k+1}+U_{j-1}^{k+1}}{(\Delta x)^2}\right)$$

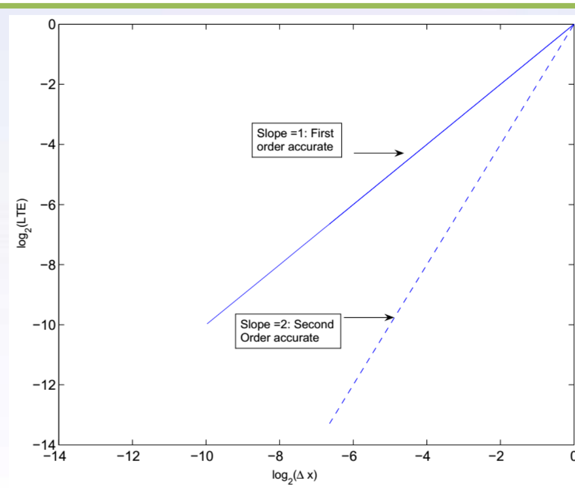
$$\Longrightarrow U_j^{k+1}=U_j^k+\frac{\kappa\Delta t}{2(\Delta x)^2}\left(U_{j+1}^k-2U_j^k+U_{j-1}^k+U_{j+1}^{k+1}-2U_j^{k+1}+U_{j-1}^{k+1}\right),$$

$$j=1,2,\ldots m$$



Computational stencil

- **Unconditionally stable**
- **Second order accurate in time**

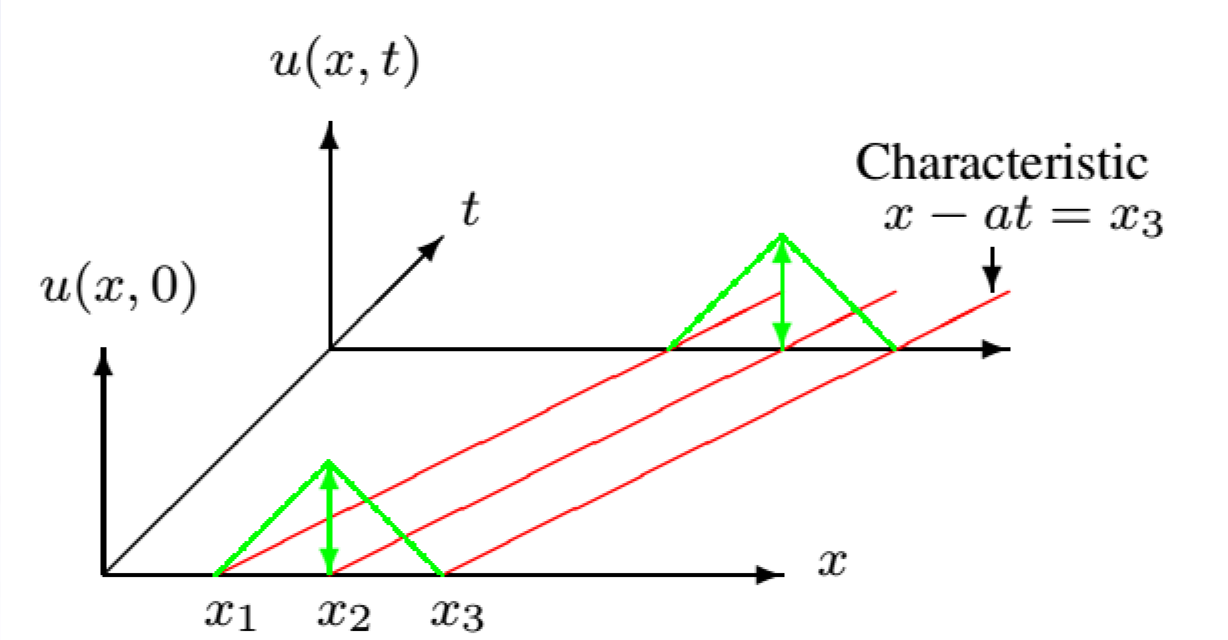


Finite Difference Method

FDM for advection equation

$$u_t + a u_x = 0,\ 0 \le x \le 1,\ t \ge 0$$

$$u(0,x) = f(x),\ \text{Initial Condition}$$



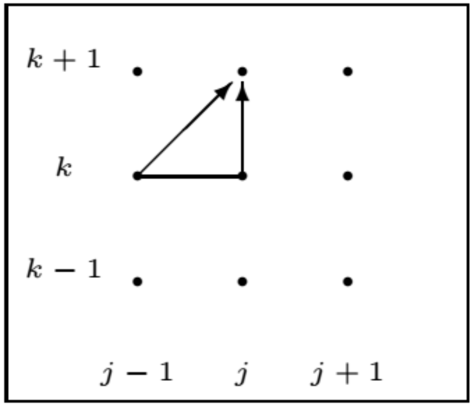
Information propagates along characteristics

Finite Difference Method

FDM for advection equation

$$\frac{U_j^{k+1} - U_j^k}{\Delta t} + a\frac{U_{j+1}^k - U_j^k}{\Delta x} = 0$$

$$\Longrightarrow \quad U_j^{k+1} = U_j^k + \frac{a\Delta t}{\Delta x}\left(U_j^k - U_{j-1}^k\right), \; j = 1, 2, \ldots m$$



Computational stencil

- **Scheme is explicit**
- **First order accurate in space and time**
- **Δt and Δx are related by CFL number: $\nu = a\Delta t / \Delta x$**

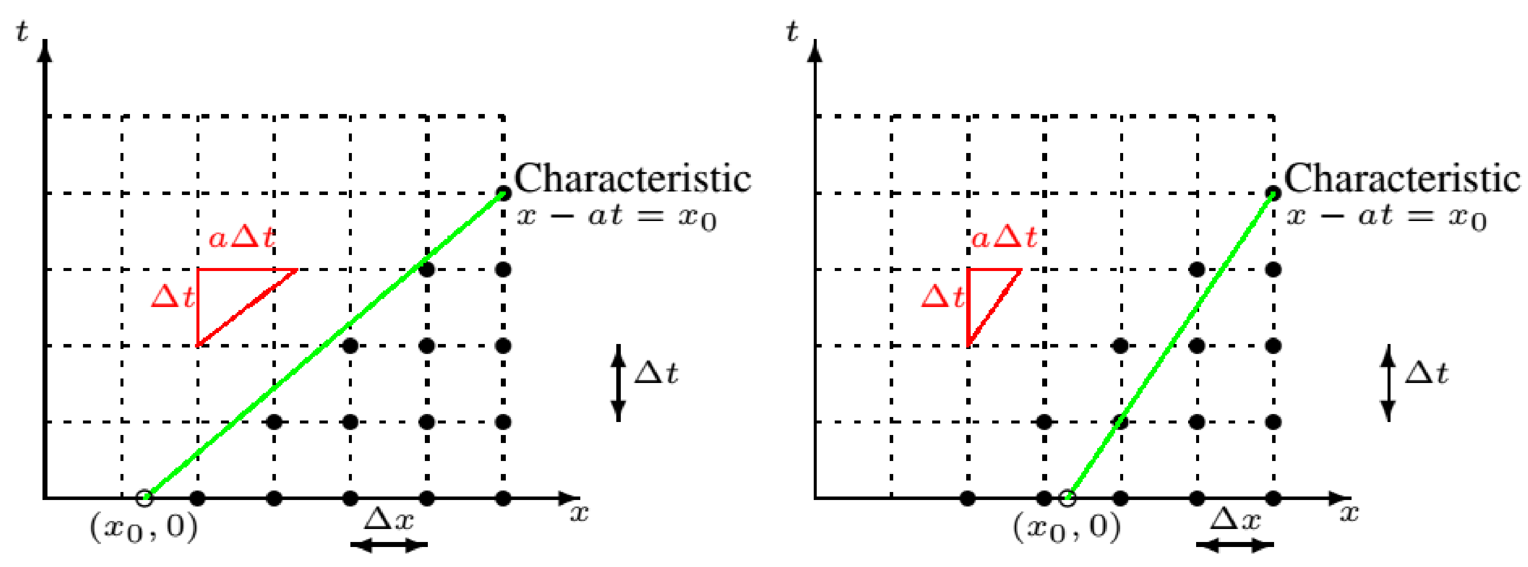
CAE

CAE
FDM
FEM

Finite Difference Method

FDM for advection equation

The CFL Condition: For stability, at each mesh point, the Domain of dependence of the PDE must lie within the domain of dependence of the numerical scheme.

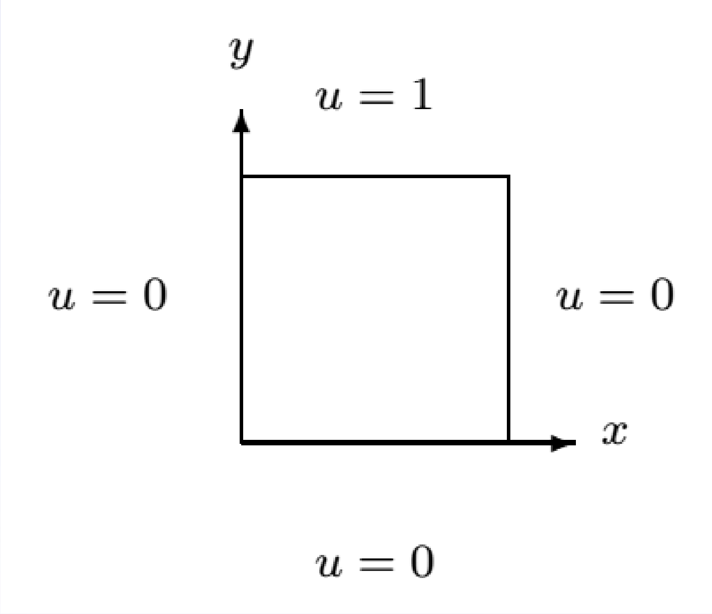


- CFL $\nu <= 1$
- CFL is a necessary condition for stability of explicit FDM applied to Hyperbolic PDEs. It is not a sufficient condition.

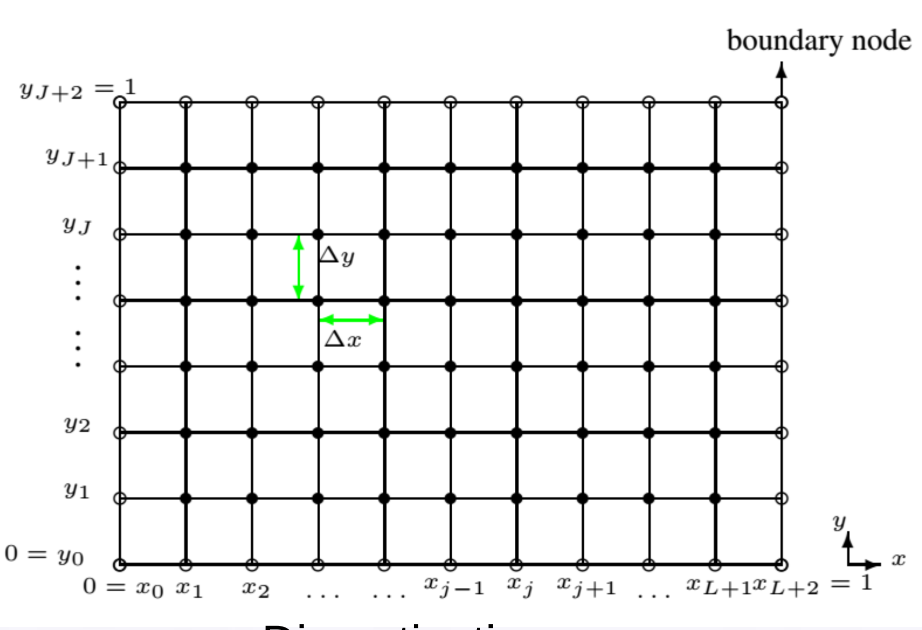
Finite Difference Method

FDM for Laplace equation

$$u_{xx} + u_{yy} = 0,\ 0 \le x \le 1,\ 0 \le y \le 1$$



Boundary conditions



Discretization

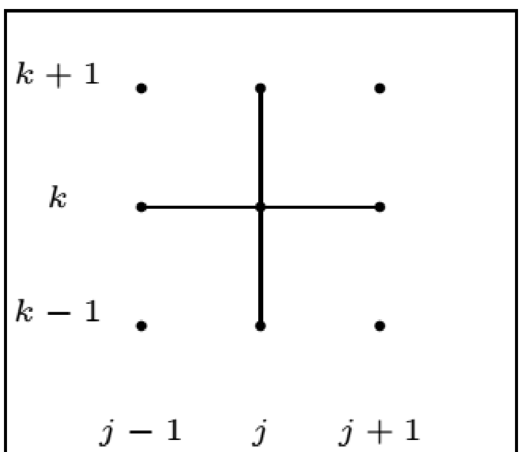
Finite Difference Method

FDM for Laplace equation – centered difference scheme

$$\frac{U_{j+1,k} - 2U_{j,k} + U_{j-1,k}}{(\Delta x)^2} + \frac{U_{j,k+1} - 2U_{j,k} + U_{j,k-1}}{(\Delta y)^2} = 0$$

If $\Delta x = \Delta y$ this becomes

$$U_{j+1,k} + U_{j-1,k} + U_{j,k+1} + U_{j,k-1} - 4U_{j,k} = 0$$



Five-point stencil

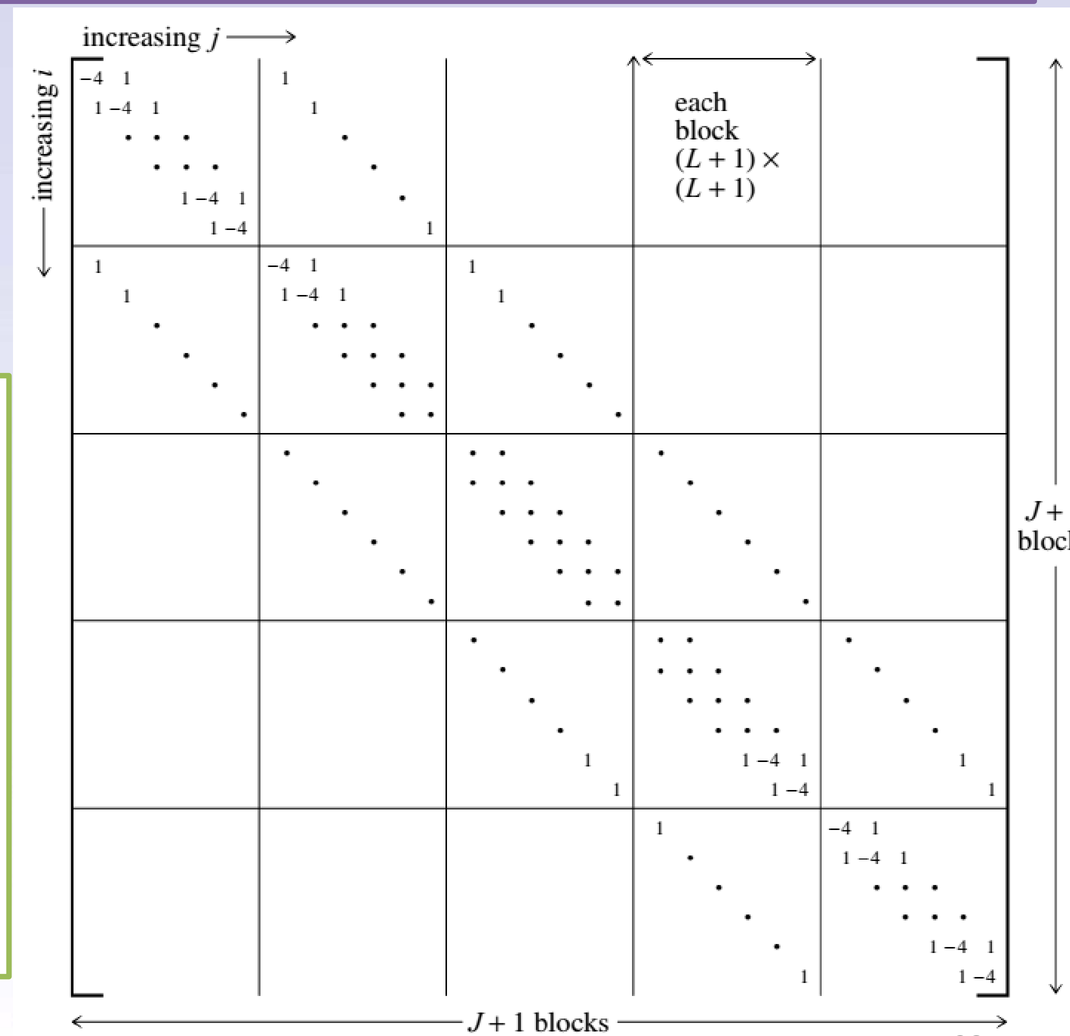
Finite Difference Method

FDM for Laplace equation – form a system of linear equations

$$AU = b$$

$$\begin{bmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 1 & 0 & -4 & 1 \\ 0 & 1 & 1 & -4 \end{bmatrix} \begin{bmatrix} U_{1,1} \\ U_{2,1} \\ U_{1,2} \\ U_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

- **b contains boundary information**
- **A is block tridiagonal**
- **Structure of A depends on the order of grid points**
- **Can be solved using iterative or direct methods, such as Gaussian elimination**



CAE

FDM

FEM

Finite Element Method

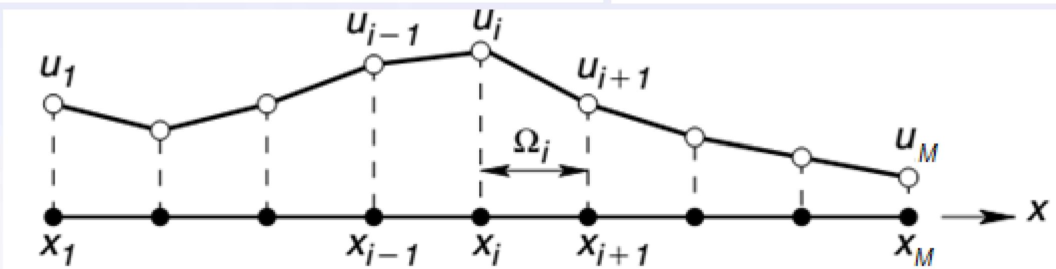
Features

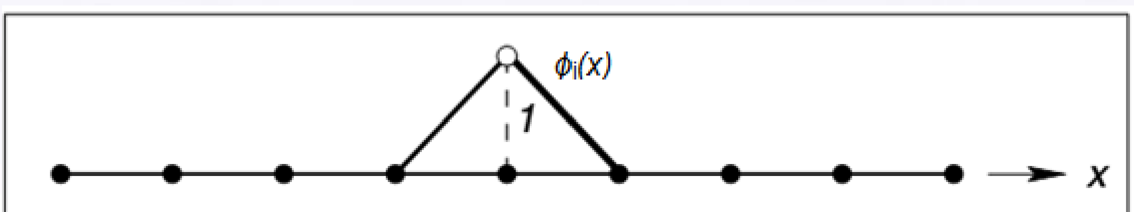
- Complicated geometries
- High-order approximations
- Strong mathematical foundation
- Flexibility

Basic Idea

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_{j=1}^{M} u_j \phi_j(\mathbf{x})$$

- ϕ_j are basis functions
- u_j: M unknowns; Need M equations
- Discretizing derivatives results in linear system





Finite Element Method

Poisson's Equation – Elliptic

$$-\Delta u(x) = f(x)$$

$$\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

Steady state heat transfer

$$\underbrace{kA\frac{\mathrm{d}^2\phi}{\mathrm{d}x^2}}_{\text{Heat conduction}} \underbrace{-hP\phi + hP\phi_f}_{\text{Heat convection}} + \underbrace{q}_{\text{Heat supply}} = 0$$

1D Example

$$-u'' = f, \qquad 0 < x < 1$$

$$u(0) = u(1) = 0$$

- Solution must be twice differentiable
- Unnecessarily strong if *f* (e.g., heat supply) is discontinuous

Finite Element Method

Weak Formulation

Multiply both sides by an arbitrary test function v and integrate

$$\int_0^1 -u''v\,dx = \int_0^1 fv\,dx$$

$$\int_0^1 u'v'\,dx - u'v|_0^1 = \int_0^1 fv\,dx.$$

$$\int_0^1 u'v'\,dx = \int_0^1 fv\,dx.$$

Since v was arbitrary, this equation must hold for all v such that the equation makes sense (v' is square integrable), and $v(0) = v(1) = 0$.

Finite Element Method

- Approximation

$$\int_0^1 u'v'dx = \int_0^1 fvdx.$$

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_{i=1}^{M} \xi_i \phi_i(x)$$

The Basic Idea of FEM

$$\int_0^1 \hat{u}'\hat{v}'dx = \int_0^1 f\hat{v}dx \quad \xrightarrow{\hat{v}=\phi_i(x)} \quad \int_0^1 \sum_{i=1}^{M} \xi_i \phi_i' \phi_j' dx = \int_0^1 f\phi_j dx$$

Thus if $A = (a_{ij})$ with $a_{ij} = \int_0^1 \phi_i'\phi_j' dx$ and $b = (b_i)$ with $b_i = \int_0^1 f\phi_i dx$, then

$$A\xi = b$$

Linear System of Equations

Finite Element Method

- Basis functions

we are looking for functions with the following property

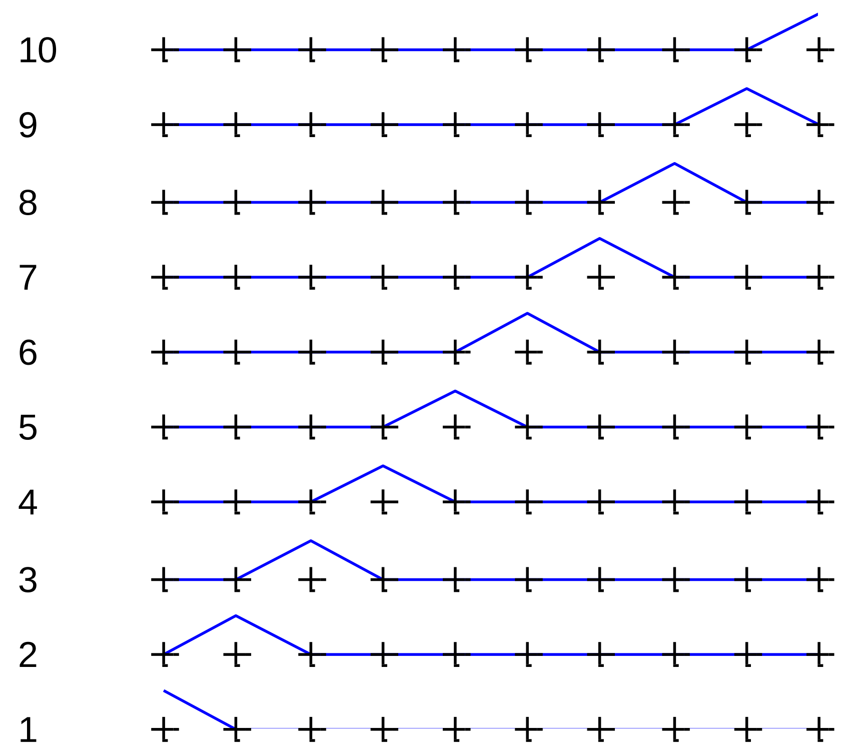
$$\phi_i(x) = \begin{cases} 1 & for \quad x = x_i \\ 0 & for \quad x = x_j,\ j \neq i \end{cases}$$

... otherwise we are free to choose any function ...

The simplest choice are of course linear functions:

+ grid nodes

blue lines – basis functions φ_i

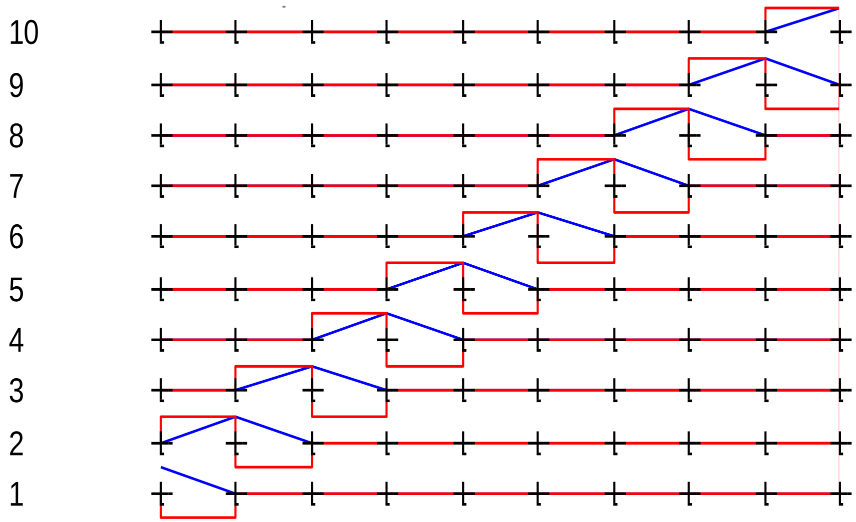


CAE
FDM
FEM

Finite Element Method

Stiffness Matrix

$$A = (a_{ij}) \text{ with } a_{ij} = \int_0^1 \phi_i' \phi_j' dx$$



To assemble the stiffness matrix we need the gradient (red) of the basis functions (blue)

For the special case when $h_j \equiv h$ we have

$$\frac{1}{h}\begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \ddots & & \vdots \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{bmatrix}$$

Finite Element Method

- Compare to FDM

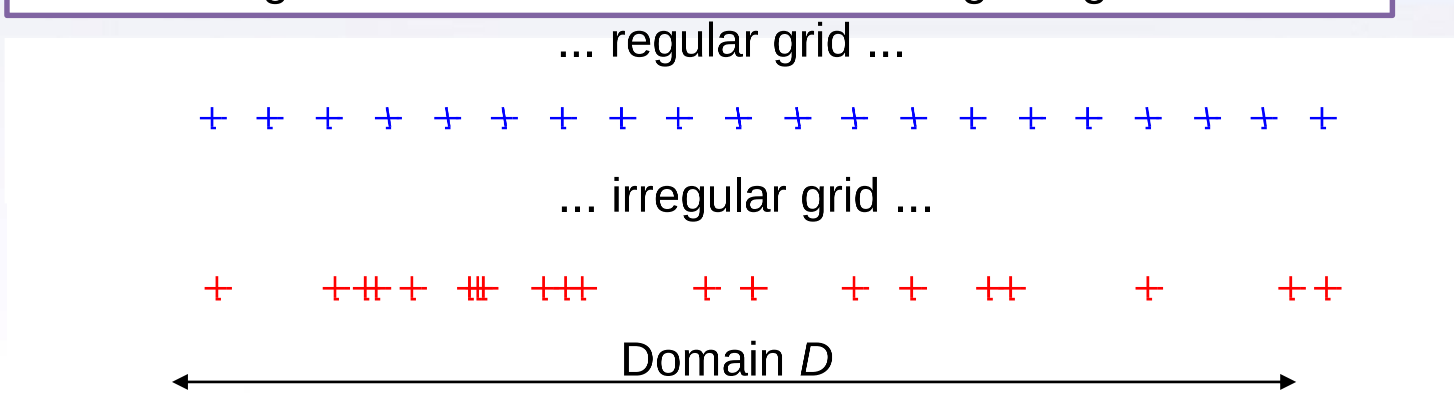
$$b = (b_i) \text{ with } b_i = \int_0^1 f\phi_i dx$$

Note that if Trapezoid rule is used to approximate the right hand side, then $b_i = hf_i$, and therefore the equations determining $\hat{u}$ are

$$\frac{\xi_{i+1} - 2\xi_i + \xi_{i-1}}{h} = hf_i$$

which are exactly the same as FDM.

- Advantage of FEM: do not have to use regular grid



Finite Element Method

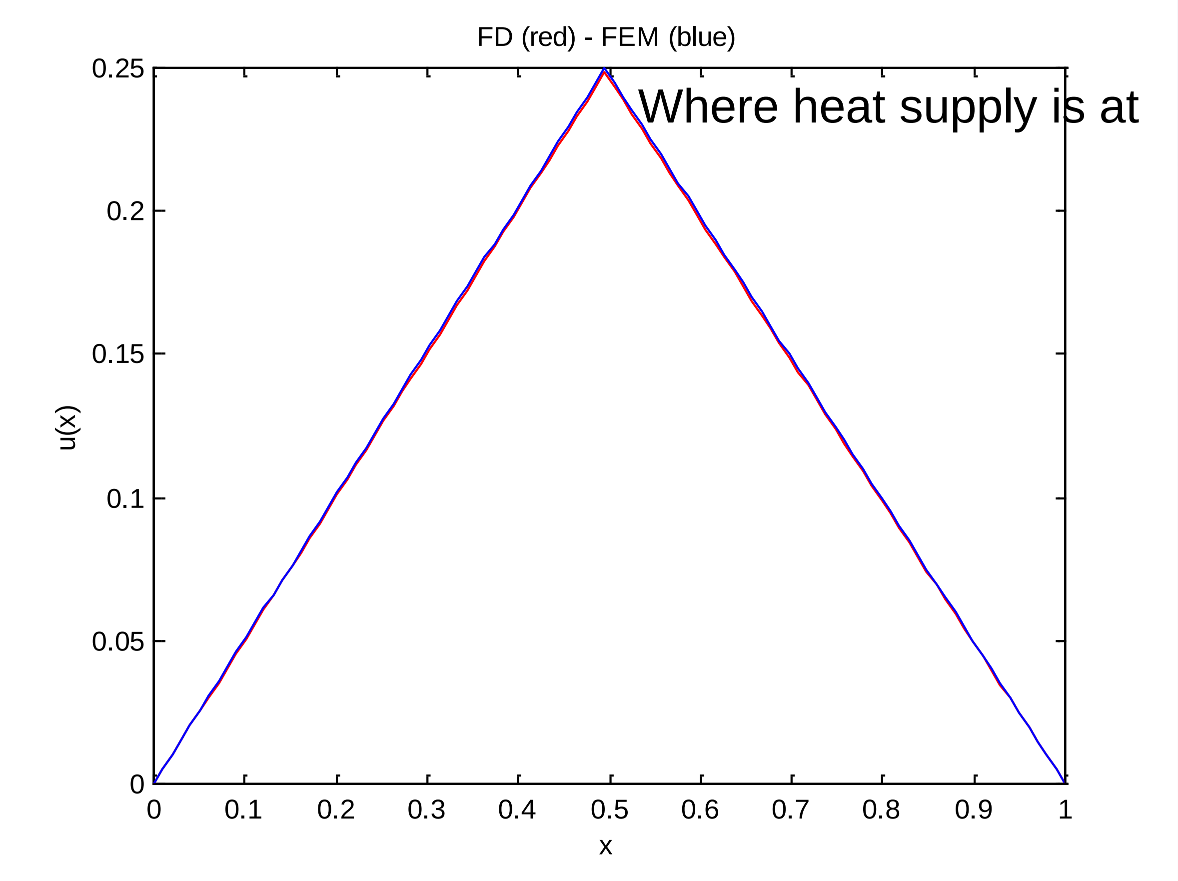
- Simplest Matlab FEM code

```
% source term
b=(1:nx)*0;b(nx/2)=1.;
% boundary left  u_1  int{ nabla phi_1  nabla phij }
u1=0;   b(1) =0;
% boundary right  u_nx int{ nabla phi_nx nabla phij }
unx=0; b(nx)=0;

% assemble matrix Aij

A=zeros(nx);

for i=2:nx-1,
   for j=2:nx-1,
      if i==j,
         A(i,j)=2/dx;
      elseif j==i+1
         A(i,j)=-1/dx;
      elseif j==i-1
         A(i,j)=-1/dx;
      else
         A(i,j)=0;
      end
   end
End
% solve linear system of equations
fem(2:nx-1)=inv(A(2:nx-1,2:nx-1))*s(2:nx-1)'; fem(1)=u1;
fem(nx)=unx;
```

Domain: [0,1]; nx=100;
dx=1/(nx-1);f(x)=d(1/2)
Boundary conditions:
u(0)=u(1)=0

Finite Element Method

- Results



Solving Linear Systems

- Equations

$$
\begin{aligned}
10x_1 - 7x_2 &= 7, \\
-3x_1 + 2x_2 + 6x_3 &= 4, \\
5x_1 - x_2 + 5x_3 &= 6.
\end{aligned}
$$

- Matrix form

$$
\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \\ 6 \end{pmatrix}
$$

- In Matlab

$X = A\backslash B.$ Or $X = A^{-1}B$

Solving Linear Systems

- Linear Algebra (Solving Linear Algebraic Equations)

- Direct(LU factorization)
  - More accurate
  - Maybe cheaper for many time steps
  - Banded matrix
  - Need more memory
  - Typically faster
- Iterative
  - Matrix-free (less memory)
  - Sparse
  - SPD (Symmetric Positive Definite)
  - Converging Issue


THANK
YOU!