Overview of this Class
Introduction to FEM (I)

Kengo Nakajima
Information Technology Center

Technical & Scientific Computing I (4820-1027)
Seminar on Computer Science I (4810-1204)

Today's Class

- Background, Target
- What is FEM (Finite-Element Method) ?
- Schedule, Grades

Descriptions of Class

- Technical & Scientific Computing I (4820-1027)
  - 科学技術計算Ｉ
  - Department of Mathematical Informatics
- Seminar on Computer Science I (4810-1204)
  - コンピュータ科学特別講義Ｉ
  - Department of Computer Science

- Introduction to FEM Programming
  - FEM: Finite-Element Method：有限要素法
  - I: FEM Programming
  - II: Parallel FEM

Instructor: Kengo Nakajima (1/2)

- Current Position
  - Professor, Supercomputing Research Division, Information Technology Center　情報基盤センター
  - Professor, Department of Mathematical Informatics, Graduate School of Information Science & Engineering 数理情報学専攻
  - Visiting Senior Researcher, Advanced Institute for Computational Science (AICS), RIKEN
- Research Interest
  - High-Performance Computing
  - Parallel Numerical Linear Algebra (Preconditioning)
  - Parallel Programming Model
  - Computational Mechanics, Computational Fluid Dynamics
  - Adaptive Mesh Refinement, Parallel Visualization

Instructor: Kengo Nakajima (2/2)

- Education
  - B.Eng (Aeronautics, The University of Tokyo, 1985)
  - M.S. (Aerospace Engineering, University of Texas, 1993)
  - Ph.D. (Quantum Engineering & System Sciences, The University of Tokyo, 2003)
- Professional
  - Mitsubishi Research Institute, Inc. (1985-1999)
  - Research Organization for Information Science & Technology (RIST) (1999-2004)
  - The University of Tokyo
    - Department Earth & Planetary Science (2004-2008)
    - Information Technology Center (2008-)
  - JAMSTEC (2008-2011), part-time
  - RIKEN (2009-), part-time

Overview of the Class (1/2)

- Finite-Element Method (FEM)：有限要素法
  - Numerical method for solving partial differential eqn’s (PDE)
    - 偏微分方程式の数値解法
  - widely-used for solving various types of real-world scientific and engineering problems, such as structural analysis, fluid dynamics, electromagnetics, and etc.
- How to “understand” FEM
  - FEM is based on various types of theories and methods, each of which is not difficult, and you must have learned all of them in high-school and undergraduate classes.
    - fundamental theories
    - programming for real applications
    - related area, such as solving linear equations
  - based on local operations for each “element”
  - Hands-on exercises for programming are important
    - ECCS2012 of Information Technology Center

Overview of the Class (2/2)

- Application
  - 1D & 3D FEM for Static Linear-Elastic Problems

- **Course materials are in English**
- **Lectures are in Japanese**

- SMASH (Science-Modeling-Algorithm-Software-Hardware)

Scientific Computing = SMASH (1/2)

Science

Modeling

Algorithm

Software

Hardware

- You have to learn many things.
- Collaboration (or Co-Design) will be important for future career of each of you, as a scientist and/or an engineer.
  - You have to communicate with people with different backgrounds.
  - It is more difficult than communicating with foreign scientists from same area.
- (Q) Department

Scientific Computing = SMASH (2/2)

- **Science**
- **Modeling**
- **Algorithm**
- **Software**
- **Hardware**

- FEM
  - Parallel FEM using MPI (Winter)
- Science: Solid Mechanics
  - 固体力学
- Modeling: FEM
- Algorithm: Iterative Solvers etc.

- You have to know many components to learn FEM, although you have already learned each of these in undergraduate and high-school classes.

This class covers ...

- Introduction to FEM
- Introduction to Solid Mechanics (Static/Elastic)
- Methods for Solving Linear Equations
  - Solving large-scale linear equations with sparse coefficient matrices is the most expensive and important part of FEM and other methods for scientific computing, such as Finite-Difference Method (FDM) and Finite-Volume Method (FVM). Recently, families of Krylov iterative solvers are widely used for this process. In this classe, details of implementations of Krylov iterative methods are provided along with FEM itself.
- FEM Codes for Static Linear-Elastic Problems
  - 1D
  - 3D

Road to Programming for “Parallel” Scientific Computing

4. Programming for Parallel Scientific Computing (e.g. Parallel FEM/FDM)

3. Programming for Real World Scientific Computing (e.g. FEM, FDM)

<u>Big gap here !!</u>

2. Programming for Fundamental Numerical Analysis (e.g. Gauss-Seidel, RK etc.)

1. Unix, Fortan, C etc.

The third step is important !

- How to parallelize applications ?
  - How to extract parallelism ?
  - If you understand methods, algorithms, and implementations of the original code, it's easy.
  - "Data-structure" is important

4. Programming for Parallel Scientific Computing (e.g. Parallel FEM/FDM)

3. Programming for Real World Scientific Computing (e.g. FEM, FDM)

2. Programming for Fundamental Numerical Analysis (e.g. Gauss-Seidel, RK etc.)

1. Unix, Fortan, C etc.

- How to understand the code ?
  - Reading the application code !!
  - It seems primitive, but very effective.
  - In this class, "reading the source code" is encouraged.

Today’s Class

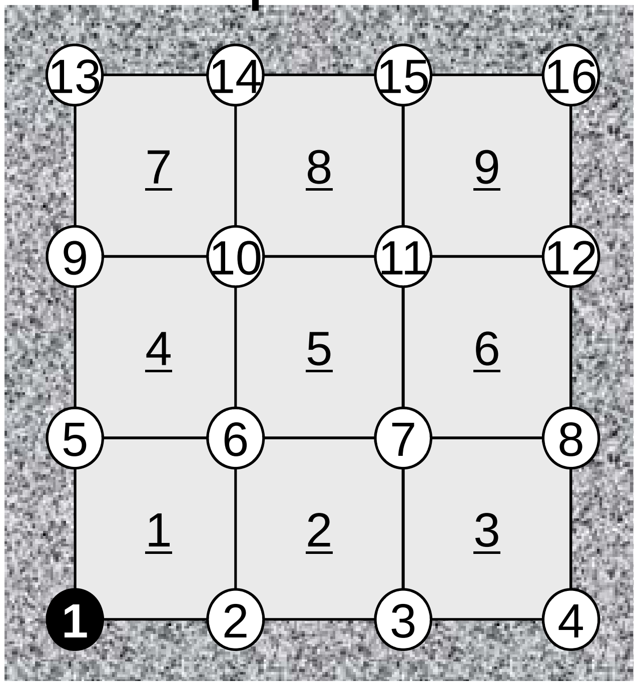
- Background, Target
- **What is FEM (Finite-Element Method) ?**
- Schedule, Grades

Finite-Element Method (FEM)

- One of the most popular numerical methods for solving PDE's (Partial Differential Eqn's, 偏微分方程式)
  - elements (meshes) & nodes (vertices)：要素～節点
- Consider the following 2D heat transfer problem:

$$\lambda\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+Q=0$$

  - 16 nodes, 9 bi-linear elements
  - uniform thermal conductivity (λ=1)
  - uniform volume heat flux (Q=1)
  - T=0 at node 1
  - Insulated boundaries



Galerkin FEM procedures

- Apply Galerkin procedures to each element:

$$\int_V [N]^T \left\{ \lambda \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + Q \right\} dV = 0$$

where $T = [N]\{\phi\}$ in each elem.

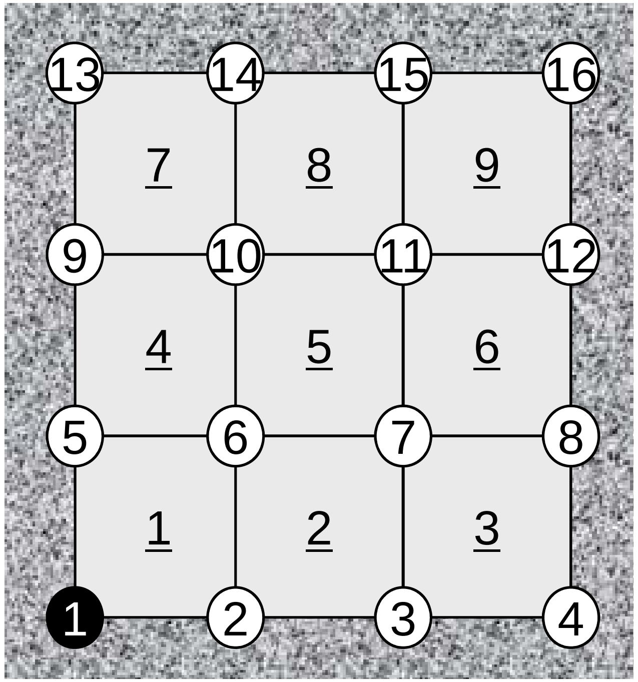
$\{\phi\}$: T at each vertex

$[N]$: Shape function (Interpolation function)

- Introduce the following "weak form（弱形式）" of original PDE using Green's theorem:

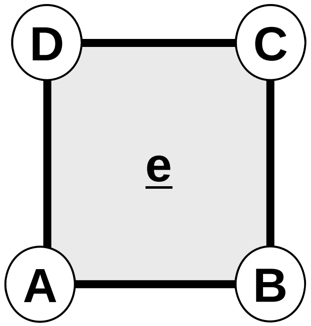
$$-\int_V \lambda \left(\frac{\partial [N]^T}{\partial x} \frac{\partial [N]}{\partial x} + \frac{\partial [N]^T}{\partial y} \frac{\partial [N]}{\partial y} \right) dV \cdot \{\phi\}$$

$$+ \int_V Q [N]^T dV = 0$$



Element Matrix（要素行列）

- Apply the integration to each element and form “element” matrix.



$$-\int_V \lambda \left(\frac{\partial [N]^T}{\partial x}\frac{\partial [N]}{\partial x} + \frac{\partial [N]^T}{\partial y}\frac{\partial [N]}{\partial y} \right) dV \cdot \{\phi\}$$

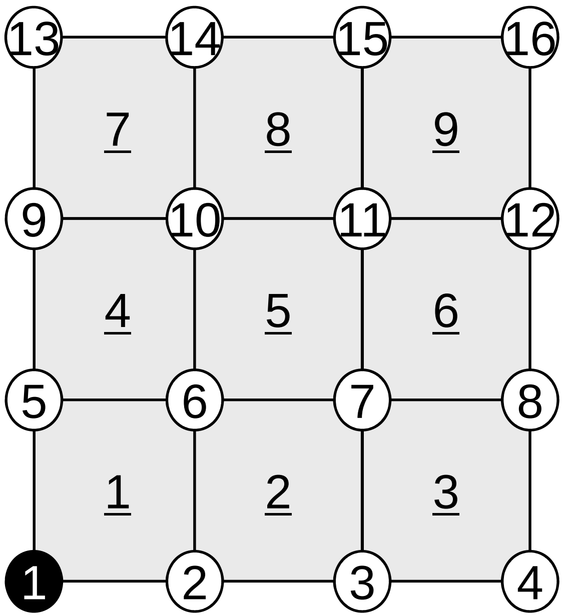
$$+\int_V Q[N]^T dV = 0$$

$$[k^{(e)}]\{\phi^{(e)}\} = \{f^{(e)}\}$$

$$\longrightarrow \begin{bmatrix} k_{AA}^{(e)} & k_{AB}^{(e)} & k_{AC}^{(e)} & k_{AD}^{(e)} \\ k_{BA}^{(e)} & k_{BB}^{(e)} & k_{BC}^{(e)} & k_{BD}^{(e)} \\ k_{CA}^{(e)} & k_{CB}^{(e)} & k_{CC}^{(e)} & k_{CD}^{(e)} \\ k_{DA}^{(e)} & k_{DB}^{(e)} & k_{DC}^{(e)} & k_{DD}^{(e)} \end{bmatrix} \begin{Bmatrix} \phi_A^{(e)} \\ \phi_B^{(e)} \\ \phi_C^{(e)} \\ \phi_D^{(e)} \end{Bmatrix} = \begin{Bmatrix} f_A^{(e)} \\ f_B^{(e)} \\ f_C^{(e)} \\ f_D^{(e)} \end{Bmatrix}$$

Global (Overall) Matrix（全体）

Accumulate each element matrix to “global” matrix.

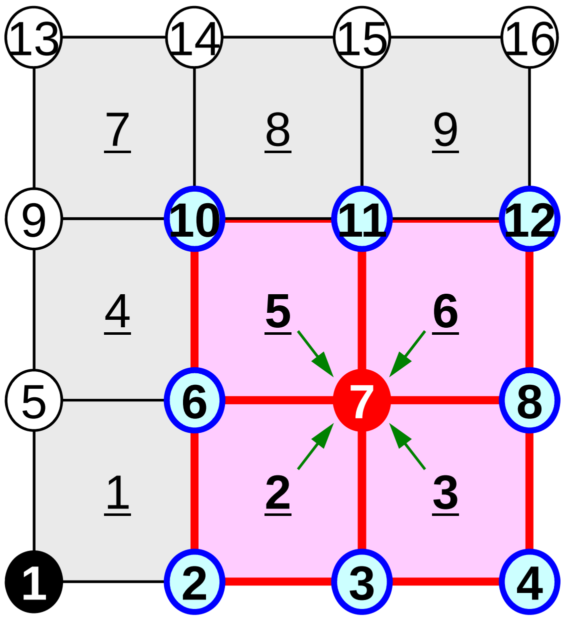


$$[K]\{\Phi\}=\{F\}$$

$$
\begin{bmatrix}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
& X & D & X & & X & X & X & & & & & & & & \\
& & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
& X & X & X & & X & D & X & & X & X & X & & & & \\
& & X & X & & & X & D & & & X & X & & & & \\
& & & & X & X & & & D & X & & & X & X & & \\
& & & & X & X & X & & X & D & X & & X & X & X & \\
& & & & & X & X & X & & X & D & X & & X & X & X \\
& & & & & & X & X & & & X & D & & & X & X \\
& & & & & & & & X & X & & & D & X & & \\
& & & & & & & & X & X & X & & X & D & X & \\
& & & & & & & & & X & X & X & & X & D & X \\
& & & & & & & & & & X & X & & & X & D
\end{bmatrix}
\begin{Bmatrix}
\Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16}
\end{Bmatrix}
=
\begin{Bmatrix}
F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16}
\end{Bmatrix}
$$

To each node …

Effect of surrounding elem’s/nodes are accumulated.



$$[K]\{\Phi\}=\{F\}$$

$$
\begin{bmatrix}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
& X & D & X & & X & X & X & & & & & & & & \\
& & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
& X & X & X & & X & D & X & & X & X & X & & & & \\
& & X & X & & & X & D & & & X & X & & & & \\
& & & & X & X & & & D & X & & & X & X & & \\
& & & & X & X & X & & X & D & X & & X & X & X & \\
& & & & & X & X & X & & X & D & X & & X & X & X \\
& & & & & & X & X & & & X & D & & & X & X \\
& & & & & & & & X & X & & & D & X & & \\
& & & & & & & & X & X & X & & X & D & X & \\
& & & & & & & & & X & X & X & & X & D & X \\
& & & & & & & & & & X & X & & & X & D
\end{bmatrix}
\begin{Bmatrix}
\Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16}
\end{Bmatrix}
=
\begin{Bmatrix}
F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16}
\end{Bmatrix}
$$

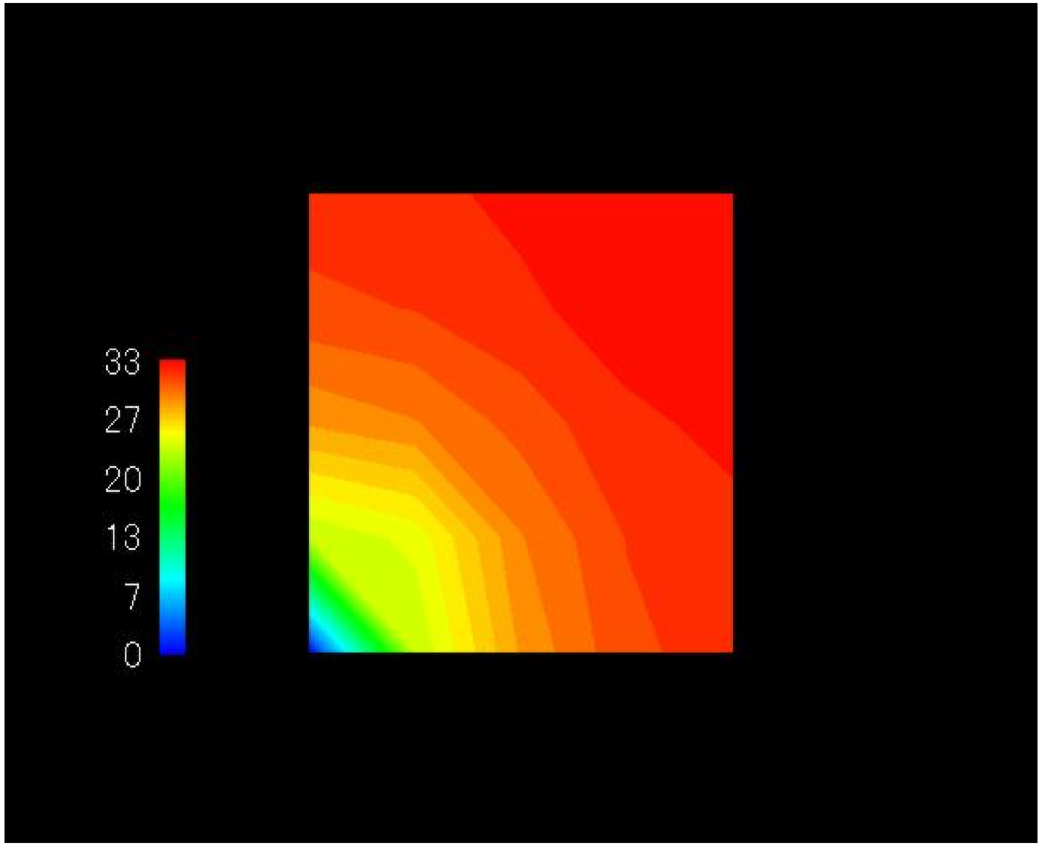
Solve the obtained global eqn’s

under certain boundary conditions

(Φ_1=0 in this case)

$$\left[\begin{array}{cccccccccccccccc}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
& X & D & X & & X & X & X & & & & & & & & \\
& & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
& X & X & X & & X & D & X & & X & X & X & & & & \\
& & X & X & & & X & D & & & X & X & & & & \\
& & & & X & X & & & D & X & & & X & X & & \\
& & & & X & X & X & & X & D & X & & X & X & X & \\
& & & & & X & X & X & & X & D & X & & X & X & X \\
& & & & & & X & X & & & X & D & & & X & X \\
& & & & & & & & X & X & & & D & X & & \\
& & & & & & & & X & X & X & & X & D & X & \\
& & & & & & & & & X & X & X & & X & D & X \\
& & & & & & & & & & X & X & & & X & D
\end{array}\right]
\begin{Bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16} \end{Bmatrix}
=
\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16} \end{Bmatrix}$$

Result …



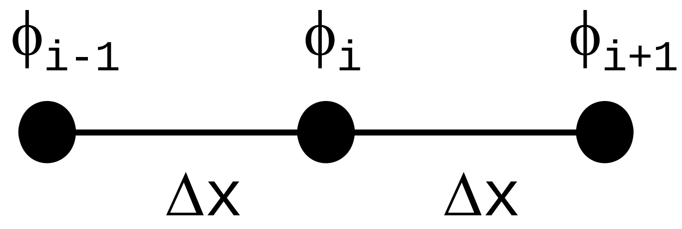
Features of FEM applications

- Typical Procedures for FEM Computations
  - Input/Output
  - Matrix Assembling
  - Linear Solvers for Large-scale Sparse Matrices
  - Most of the computation time is spent for matrix assembling/formation and solving linear equations.
- **HUGE** “indirect” accesses
  - memory intensive
- Local “element-by-element” operations
  - sparse coefficient matrices（疎行列）
  - suitable for parallel computing
- Excellent modularity of each procedure

FDM and FEM

- Numerical Method for solving PDE’s
  - Space is discretized into small pieces (elements, meshes)
- Finite Difference Method (FDM)（（有限）差分法）
  - Differential derivatives are directly approximated using Taylor Series Expansion.
- Finite Element Method（FEM）（有限要素法）
  - Solving “weak form” derived from integral equations.
    - “Weak solutions” are obtained.
  - Method of Weighted Residual (MWR)（重み付き残差法）, Variational Method（変分法）
  - Suitable for Complicated Geometries
    - Although FDM can handle complicated geometries ...

Finite Difference Method (FDM) Taylor Series Expansion



2nd-Order Central Difference

$$\phi_{i+1} = \phi_i + \Delta x \left(\frac{\partial \phi}{\partial x}\right)_i + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i + \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

$$\phi_{i-1} = \phi_i - \Delta x \left(\frac{\partial \phi}{\partial x}\right)_i + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i - \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

$$\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} = \left(\frac{\partial \phi}{\partial x}\right)_i + \underline{\frac{2\times(\Delta x)^2}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots}$$

1D Heat Conduction

- 2nd-Order Central Difference

$$\left(\frac{d^2\phi}{dx^2}\right)_i \approx \frac{\left(\frac{d\phi}{dx}\right)_{i+1/2} - \left(\frac{d\phi}{dx}\right)_{i-1/2}}{\Delta x} = \frac{\frac{\phi_{i+1}-\phi_i}{\Delta x} - \frac{\phi_i - \phi_{i-1}}{\Delta x}}{\Delta x} = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2}$$

- Linear Equation at Each Grid Point

$$\frac{d^2\phi}{dx^2} + BF = 0 \quad \longrightarrow$$

$$\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} + BF(i) = 0 \quad (1 \le i \le N)$$

$$\frac{1}{\Delta x^2}\phi_{i+1} - \frac{2}{\Delta x^2}\phi_i + \frac{1}{\Delta x^2}\phi_{i-1} + BF(i) = 0 \quad (1 \le i \le N)$$

$$A_L(i) \times \phi_{i-1} + A_D(i) \times \phi_i + A_R(i) \times \phi_{i+1} = BF(i) \quad (1 \le i \le N)$$

$$A_L(i) = \frac{1}{\Delta x^2},\ A_D(i) = -\frac{2}{\Delta x^2},\ A_R(i) = \frac{1}{\Delta x^2}$$

FDM can handle complicated geometries: BFC

Handbook of Grid Generation

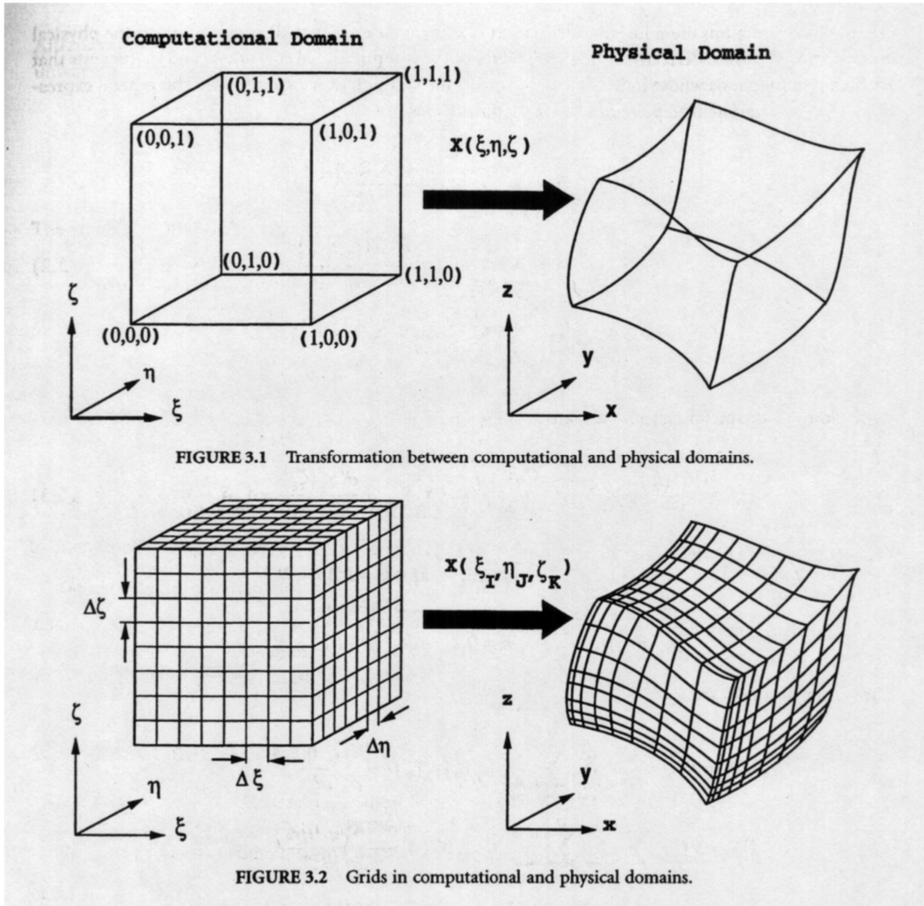
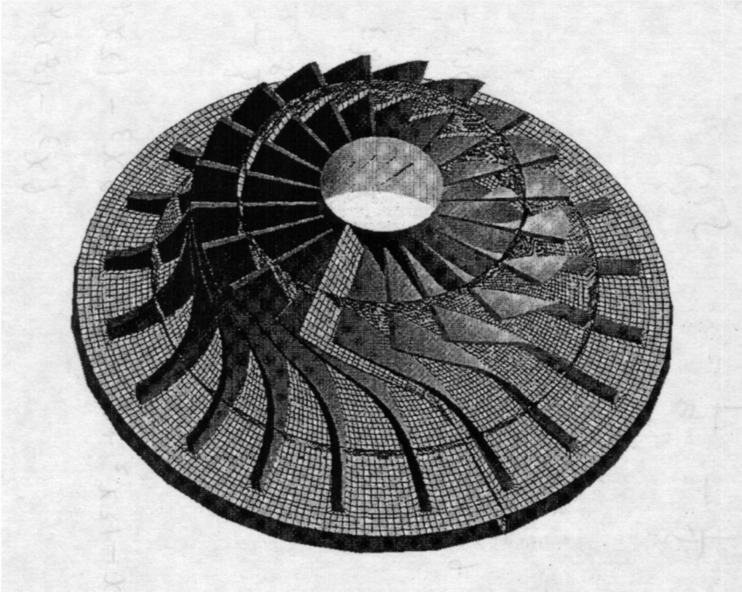


FIGURE 3.1 Transformation between computational and physical domains.

FIGURE 3.2 Grids in computational and physical domains.



History of FEM

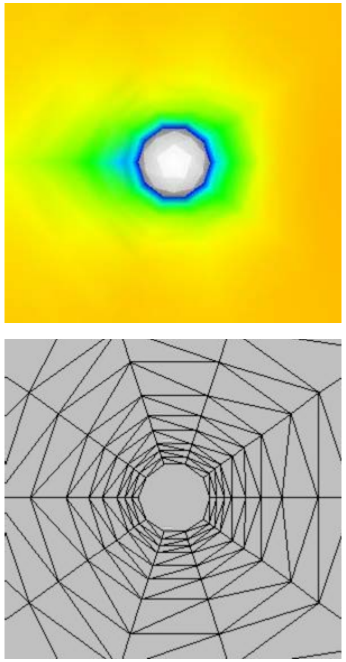
- In 1950's, FEM was originally developed as a method for structure analysis of wings of airplanes under collaboration between Boeing and University of Washington (M.J. Turner, H.C. Martin etc.).
  - "Beam Theory" cannot be applied to sweptback wings for airplanes with jet engines.
- Extended to Various Applications
  - Non-Linear: T.J.Oden
  - Non-Structure Mechanics: O.C.Zienkiewicz
- Commercial Package
  - NASTRAN
    - Originally developed by NASA
    - Commercial Version by MSC
    - PC version is widely used in industries

Recent Research Topics

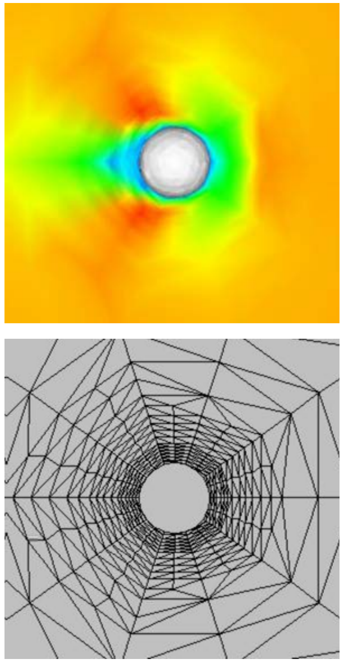
- Non-Linear Problems
  - Crash, Contact, Non-Linear Material
  - Discontinuous Approach
    - X-FEM
- Parallel Computing
  - also in commercial codes
- Adaptive Mesh Refinement (AMR)
  - Shock Wave, Separation
  - Stress Concentration
  - Dynamic Load Balancing (DLB) at Parallel Computing
- Mesh Generation
  - Large-Scale Parallel Mesh Generation

Supersonic Flow around a Sphere

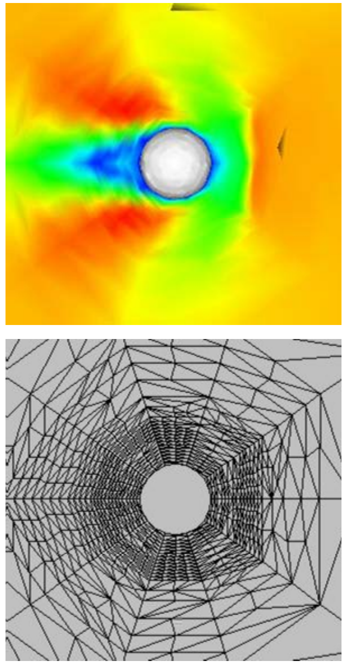
M=1.40, Ideal Gas, Uniform Flow, Re=10^6



Initial Grid



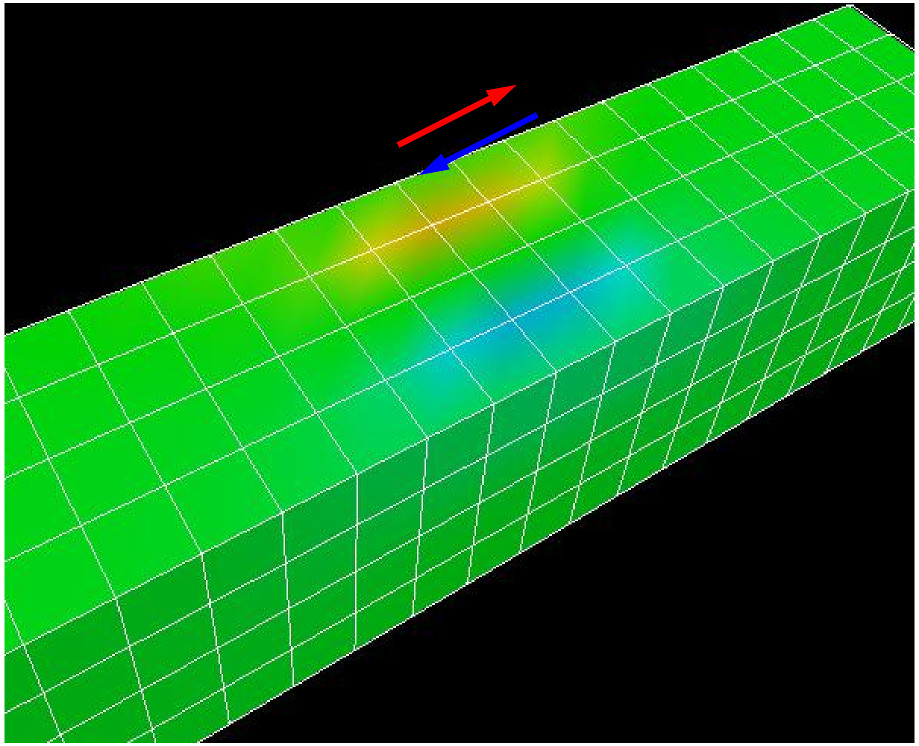
1-Lev. Adapted

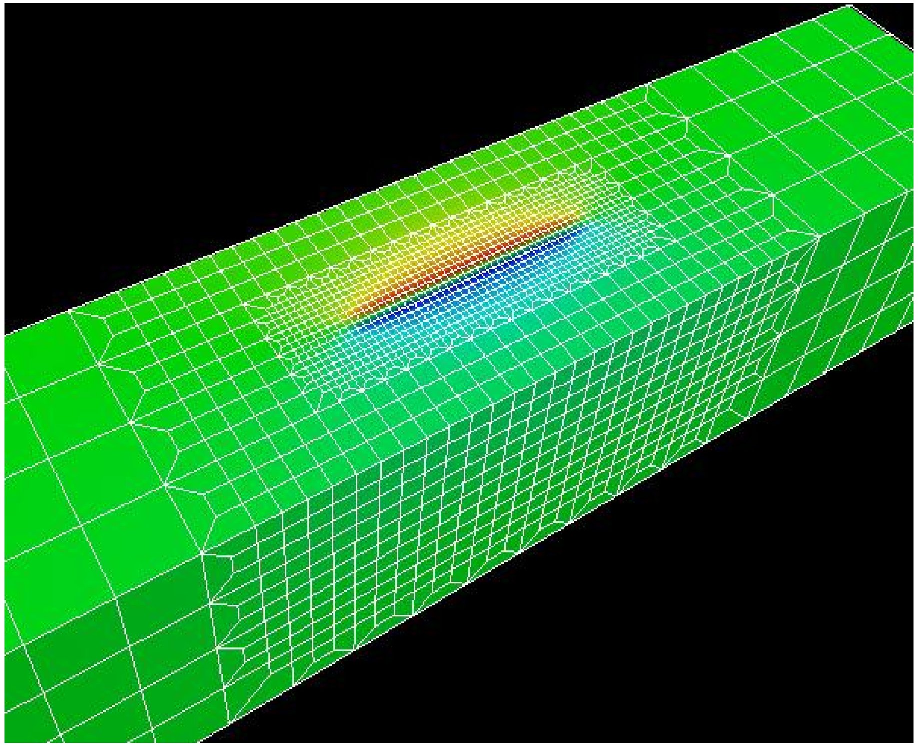


2-Lev. Adapted

	Initial Grid before/after DLB		1-Lev. Adapted before/after DLB		2-Lev. Adapted before/after DLB	
PE0	137	-	793	652	3834	2527
PE1	137	-	696	650	2769	2526
PE2	136	-	668	652	2703	2522
PE3	136	-	448	651	1390	2524

3D Simulations for Earthquake Generation Cycle San Andreas Faults, CA, USA





movie

Today’s Class

- Background, Target
- What is FEM (Finite-Element Method) ?
- **Schedule, Grades**

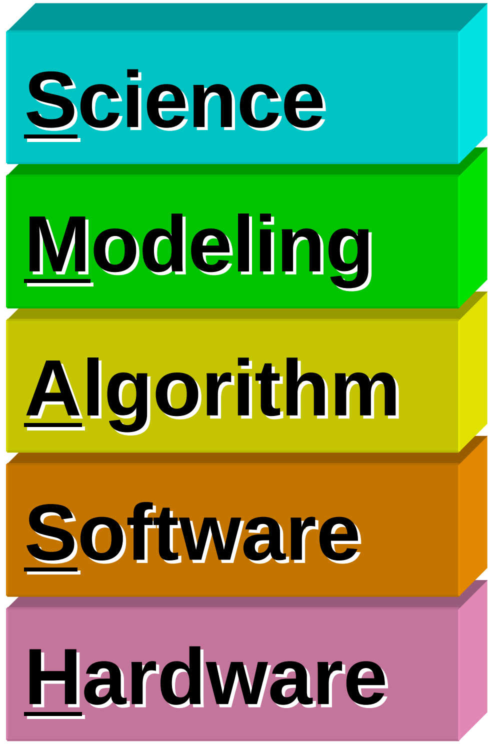
Date	ID	Title
Apr-07 (M)	CS-01	Introduction to FEM (I)
Apr-14 (M)	CS-02	Introduction to FEM (II)
Apr-21 (M)	CS-03	Introduction to Theory of Elasticity
Apr-25 (F)	**CS-04**	**1D Code (I)**
Apr-28 (M)	CS-05	1D Code (II), Linear Solver
May-12 (M)	CS-06	1D Code (III)
May-19 (M)	CS-07	1D Code (IV), Exercise #1
May-26 (M)	CS-08	3D Code (I)
Jun-02 (M)	(canceled)	
Jun-06 (F)	**CS-09**	**3D Code (II)**
Jun-09 (M)	(canceled)	
Jun-16 (M)	CS-10	3D Code (III)
Jun-20 (F)	**CS-11**	**3D Code (IV)**
Jun-23 (M)	(canceled)	
Jun-30 (M)	(canceled)	
Jul-07 (M)	CS-12	**3D Code (V), Exercise #2**
Jul-14 (M)	CS-13	Example for Exercise #1

"Prerequisites"

- Fundamental physics and mathematics
  - Linear algebra, analytics
- Experiences in fundamental numerical algorithms
  - LU factorization/decomposition, Gauss-Seidel
- Experiences in programming by C or Fortran
- Experiences and knowledge in UNIX
- User account of ECCS2012 must be obtained:
  - http://www.ecc.u-tokyo.ac.jp/doc/announce/newuser.html

Strategy (1/2)

- Focused on solid mechanics (linear elastic)
  - FEM's initial motivation
- SMASH
  - Focused on SMASH, but some courses for "science"
- FEM is really useful for solving 2D/3D problems
  - 1D can be done by FDM more easily.
  - You need to reach 3D !!
  - Very tight schedule



Strategy (2/2)

- If you can develop programs by yourself, it is ideal... but difficult.
  - focused on “reading”, not developing by yourself
  - Programs are in C and Fortran
    - Lectures are done by C
- Lecture Materials
  - available at **NOON Friday** through WEB.
  - NO hardcopy is provided
- Starting at 08:40
  - You can enter the building after 08:00
- Taking seats from the front row.
- Terminals must be shut-down after class.

Grades

- 2 Reports on programming
- Homework
  - Answers are available in the next class.

参考文献（1/2）

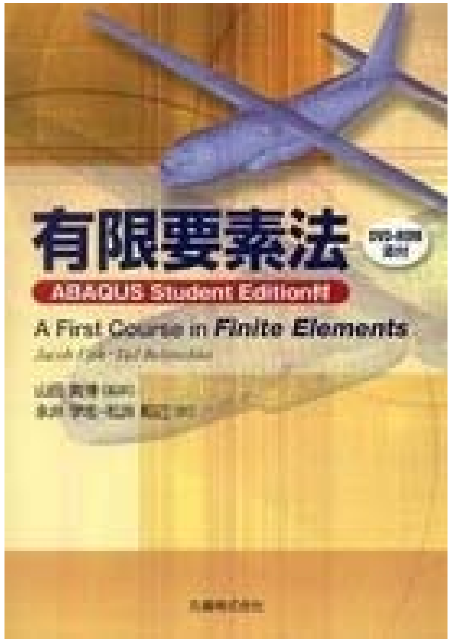
- 菊地「有限要素法概説（新訂版）」，サイエンス社，1999.
- 竹内，樫山，寺田（日本計算工学会編）「計算力学：有限要素法の基礎」，森北出版，2003.
- 登坂，大西「偏微分方程式の数値シミュレーション第2版」，東大出版会，2003.
  - 差分法，境界要素法との比較
- 福森「よくわかる有限要素法」，オーム社，2005.
  - ヘルムホルツ方程式
- 矢川，宮崎「有限要素法による熱応力・クリープ・熱伝導解析」，サイエンス社，1985. （品切）
- Segerlind, L.（川井監訳）「応用有限要素解析 第2版」，丸善，1992.（品切）

参考文献（より進んだ読者向け）

- 菊池，岡部「有限要素システム入門」，日科技連，1986.
- 山田「高性能有限要素法」，丸善，2007.
- 奥田，中島「並列有限要素法」，培風館，2004.

- Smith, I. 他「Programming the Finite Element Method (4th edition)」，Wiley.

References



- Fish, Belytschko, A First Course in Finite Elements, Wiley, 2007
  - Japanese version is also available
  - “ABAQUS Student Edition” included
- Smith et al., Programming the Finite Element Method (4th edition), Wiley, 2004
  - Parallel FEM
- Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover, 2000

If you have any questions, please feel free to contact me !

- Office: 3F Annex/Information Technology Center #36
  - 情報基盤センター別館3F 36号室
- ext.: 22719
- e-mail: nakajima(at)cc.u-tokyo.ac.jp
- **NO specific office hours, appointment by e-mail**

- http://nkl.cc.u-tokyo.ac.jp/14s/