Open source CAE system:
ADVENTURE

Hiroshi Kawai
Univ. of Tokyo, Japan
(ADVENTURE Project)
Contents.

- ADVENTURE
- Open source software for “Missing Middle”
- How to optimize FEM code on supercomputer?
- Supercomputing for “Big Engineering”
ADVENTURE Project
since 1997 -

General purpose CAE System for Large Scale Analysis & Design, named **ADVENTURE** has been developed, released and applied.

(1) Finite element method, 10 - 100 million DOF mesh
(Few million DOF at most)

(2) High-parallel performance of 90 % even in 1000 PE
(Rarely worked on MPP)

(3) Portability : WS, PC cluster, MPP, ES, (K-supercomputer)
(WS or small scale cluster)

(4) Open-source
(Expensive license fee)

(5) Module-based architecture, common I/O
(Closed architecture)
ADVENTURE : Open-source CAE package

Module-based Architecture : 20+ modules
CAD-connection, Mesh Generation,
Domain Decomposition, FE-Solvers,
Visualization, Optimization, etc.

AdvIO, AdvTriPatch, AdvTetMesh,
AdvBCtool, AdvSolid, AdvVisual,
AdvCAD, AdvThermal,
AdvMagnetic, AdvForge,
AdvImpact, AdvFluid, AdvIAgent,
AdvOpt, AdvShape, AdvMaterial,
AdvAuto,

http://adventure.sys.t.u-tokyo.ac.jp/

It’s open-source !
Module architecture of ADVENTURE system

Utilities:
- ADV_iAgent
- ADV_Material
- ADV_FEMAPtool
- ADV_DecisionMaker
- ADV_CAD
- ADV_TriPatch
- ADV_TetMesh
- ADV_BCtool
- ADV_Metis
- ADV_Solid / ADV_Forge / ADV_Impact / ADV_sFlow
- ADV_Thermal / ADV_Fluid / ADV_Magnetic
- ADV_Visual
- ADV_Opt / ADV_Auto
- ADV_Io

Blue: parallel
Italic: Ver.1
Roman: Ver. β
Setting boundary conditions using BCtool
BC tool : geometry model
BC tool : mesh
BC tool: displacement constraint on face
BC tool: load on face
Visual: geometry model
Visual : mesh
Visual : domain decomposition
Visual: deformation and stress contour plot
Visual: deformation and contour with mesh
Domain Decomposition Method (DDM)
---a kind of Schur complement-type, iterative sub-structuring method
Parallel FE analysis techniques

• Domain Decomposition-based solver
  ex.) DDM
  Preconditioner: BDD, FETI, ...

• Parallel matrix solver
  ex.) Parallel CG solver
  Preconditioner: ILU, IC, AMG, ...
Domain Decomposition Method (DDM)

- Analysis domain
- Domain decomposed into multiple sub-domains

reduce the original problem into the interface problem
Flow of Basic DDM Algorithm

Enforced displacement on interface

FEM FEM FEM ...

Reaction force on interface

Converged ?

Update displacement B.C. on interface

END
Finite Element Analysis and Mesh Subdivision

例:
二次元問題、三角形要素の場合
Domain Decomposition: into subdomains
Converting to interface problem
DDM using local Schur complement

Perform static condensation of interior DOF of each sub-domain

\[
[S] = [K_{bb}] - [K_{ib}]^T [K_{ii}]^{-1} [K_{ib}]
\]

Evaluate local Shur complement at the first DDM iteration

\[
\{f_b\} = [S] \{u_b\}
\]

For each DDM iteration, evaluate reaction force over interface
Subdomain-wise FEM

Matrix equation of each subdomain  \( (i:\text{interior}, b:\text{boundary}) \)

\[
\begin{bmatrix}
[K_{ii}] & [K_{ib}]
\end{bmatrix}
\begin{bmatrix}
[K_{ib}]^T & [K_{bb}]
\end{bmatrix}
\begin{Bmatrix}
{u_i} \\
{u_b}
\end{Bmatrix}
= \begin{Bmatrix}
{f_i} \\
{f_b}
\end{Bmatrix}
\]

Enforce displacement B.C. on interface

\[
[K_{ii}]{u_i} = \{f_i\} - [K_{ib}]{u_b}
\]

Obtain displacement (solution)

\[
\{f_b\} = [K_{ib}]^T \{u_i\} + [K_{bb}]{u_b}
\]

Evaluate reaction force on interface
Pre-conditioner for DDM: Balancing Domain Decomposition (BDD)
Iterative methods with high convergence rate.

- ICCG method
- Multigrid (Brandt '77)
- AMG (Ruge and Stuben '87)
- FETI (Farhat '92)
- BDD (Mandel '93)
- FETI-DP (Farhat et al. '00)
- BDDC (Dohrmann '03)

All of them including the coarse grid correction

IBDD-DIAG method for huge scale problems

**ICCG**: Incomplete Cholesky Conjugate Gradient, **AMG**: Algebraic MultiGrid
**FETI**: Finite Element Tearing and Interconnecting, **FETI-DP**: Dual-Primal FETI
**BDD**: Balancing Domain Decomposition, **BDDC**: BDD based on Constraints
Preconditioner for DDM:

**Balancing Domain Decomposition (BDD)**

- **DDM**, which solve the interface problem using preconditioned CG (PCG) method.

- **Neumann-Neumann (N-N) preconditioner**, which precondition to the residual vector as a local subdomain preconditioner.

- **Coarse grid correction**, which delete a coarse space component from the preconditioned vector.

Coarse Grid Correction in BDD pre-conditioner
Flow of DDM with BDD pre-conditioner

Initialization

DDM body

BDD pre-conditioner

Coarse grid correction

Converged? yes no

END

Coarse grid correction using Direct solver

subdomain-wise FEM

DDM CG loop
CAE Education
-- filling “missing middle” --
e-learning system for CAE
CAE education

- There are 4 procedures on FEM analysis
  - Mesh generation
    - Keep balance of the numerical accuracy and computing time
  - Boundary condition
    - Specify appropriate load and constraint
  - Material properties
    - Specify appropriate material property parameters
  - Post processing (Visualization)
    - The result of the analysis is examined by visualizing the stress distribution and the deformation plot

Developing an e-learning system to educate above 4 skills

http://www.woelfel.de/wbieng/fem/fem-berechnung.html
• Local PC: easy operation
• Remote Server: large scale simulation
Web-based CAE System

ADVENTURE Modules

Web(HTTPS) or SSH

Local PC (browser)

Visual

BCtool

TriPatch

TetMesh

Data communication

4 Modules on Remote Parallel Computer

Parallel

Metis

Solid

Remote PC Cluster
Visualization
JSME Certification of Computational Mechanics Engineer since 2003

JSME : Japan Society of Mechanical Engineering
What’s Happening Now !!

CAE systems are getting Black-box

Ordinary Engineers (Non-expert engineers) use Computational Mechanics (CM) technologies in various engineering fields.
Due to nice GUI and robust technology of commercial CAE systems, even a beginner can get some nice looking result.

However, there is some possibility that the nice looking result is totally wrong. Furthermore, it is not an easy task to prove that the result is reliable.
Seminar for CAE Software at Keio Univ.
How to optimize FEM code on supercomputer?
Classification of HPC systems

• Multi-processing and parallelization
  – Hardware component: core, CPU chip, chassis, rack
  – Memory model: shared memory / distributed memory
  – Programming model: MPI / OpenMP / hybrid-parallel

• Type of processor and instruction set
  – Processor: scalar / vector / GPU
  – Instruction set: scalar / vector / SIMD

• In short ...
  – vector supercomputer
  – scalar MPP (Massively parallel processors), PC cluster
  – GPU, GPU cluster
Scalar MPP / PC cluster

• Core : Multiple FP instructions per clock
  – Accelerated by SIMD instruction (multi-media extension)
    • SIMD Vectorization

• CPU : Multi-core
  – Shared memory, OpenMP parallel

• Chassis : Multi-CPU socket
  – Non-uniform memory access (NUMA)
    • Physically distributed, logically shared memory space

• System : Multi-chassis, Multi-rack
  – Distributed memory, MPI parallel
  – High speed network : Infiniband, Myrinet
How to handle multi-core scalar CPU?
Modern Scalar CPU
from multi-core to many-core

Num. transistors
Clock frequency

core

Memory

cache

core

cache
Subdomain-wise FEM

Matrix equation of each subdomain (i:interior, b:boundary)

\[
\begin{bmatrix}
[K_{ii}] & [K_{ib}] \\
[K_{ib}]^T & [K_{bb}]
\end{bmatrix}
\begin{bmatrix}
\{u_i\} \\
\{u_b\}
\end{bmatrix}
= \begin{bmatrix}
\{f_i\} \\
\{f_b\}
\end{bmatrix}
\]

Enforce displacement B.C. on interface

\[ [K_{ii}] \{u_i\} = \{f_i\} - [K_{ib}] \{u_b\} \]

Obtain displacement (solution)

\[ \{f_b\} = [K_{ib}]^T \{u_i\} + [K_{bb}] \{u_b\} \]

Evaluate reaction force on interface
How to solve domain-wise FEM? (1)

- Type of linear algebraic solver
  - **Direct solver**
    - LDL decomposition, forward reduction/backward substitution, skyline memory storage
  - **Iterative solver**
    - CG, preconditioning, non-zero component memory storage

- Memory saving strategy
  - In addition to mesh, store any other data (**Storage** type)
    - Coefficient matrix (Skyline / Non-zero component)
    - LDL decomposition
    - Data for preconditioning
  - Mesh only (**Storage-Free** type)
How to solve domain-wise FEM ? (2)

- Direct solver-based
  - Storage type: **DS** (*Direct-solver based matrix Storage*)
    - Store original coefficient and factorized matrix on memory
    - Each DDM loop, FR/BS only
  - Storage-free type: **DSF** (*Direct ... Storage-Free*)
    - Each DDM loop, factorization and FR/BS

- Iterative solver-based
  - Storage type: **IS** (*Iterative-solver based matrix Storage*)
    - Store coefficient and precondition matrix on memory
    - Each DDM loop, solve by PCG
  - Storage-free type: **ISF** (*Iterative ... Storage-Free*)
    - Each DDM loop, build coefficient & preconditioning matrix and solve by PCG
Subdomain-wise FEM on Scalar CPU
-- subdomain size vs computational cost --

Calculation Time

Num.DOF per subdomain

Out-of Cache

DSF $N^{7/3}$

DS $N^{5/3}$

IS, ISF $N^{4/3}$

N

500 1000 2000 3000
## Benchmark on PC cluster

<table>
<thead>
<tr>
<th>Total DOF</th>
<th>Domain DOF</th>
<th>Num. Domains</th>
<th>DS</th>
<th>DSF</th>
<th>ISF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>memory (GB)</td>
<td>Time (min.)</td>
<td>memory (GB)</td>
</tr>
<tr>
<td>25M</td>
<td>380</td>
<td>100K</td>
<td>1.0</td>
<td>2.5</td>
<td>0.3</td>
</tr>
<tr>
<td>37M</td>
<td>530</td>
<td>100K</td>
<td>1.0</td>
<td>2.8</td>
<td>0.3</td>
</tr>
<tr>
<td>100M</td>
<td>1300</td>
<td>100K</td>
<td>5.6</td>
<td>4.7</td>
<td>0.5</td>
</tr>
<tr>
<td>220M</td>
<td>2700</td>
<td>100K</td>
<td></td>
<td>Out of memory</td>
<td>Out of cache</td>
</tr>
<tr>
<td>490M</td>
<td>5600</td>
<td>100K</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

- DS, 100M DOF, 5 minutes
- ISF, 200M DOF, 30 minutes
- ISF, 500M DOF, 2 hours

### Peak performance ratio:

- DSF: 40 - 60%
- ISF: 15 - 40%
How to handle GPU?
DDM using local Schur complement

Perform static condensation of interior DOF of each sub-domain

\[
[S] = [K_{bb}] - [K_{ib}]^T [K_{ii}]^{-1} [K_{ib}]
\]

Evaluate local Shur complement at the first DDM iteration

\[
\{f_b\} = [S]\{u_b\}
\]

For each DDM iteration, evaluate reaction force over interface

- Matrix vector product ([S] : symmetric dense matrix)
- Direct Storage (DS)

⇒ on Vector / GPU, vectorize in terms of num. sub-domains
DDM hot spot benchmark on GPU

Matrix-vector product for each sub-domain

\[
\{ f_{b_{dom}} \} = [S_{dom}]\{ u_{b_{dom}} \}
\]

for \( \text{dom} = 1 \ldots \text{number of domains} \)

37 G flops x10 faster than Intel Core i7

- Outer loop unrolling (5x5)
- Reduce memory access because of symmetric matrix
- Vector length : Num. Domains \( \approx 500 \)
Japanese Next Gen. Supercomputer “K”
K-Supercomputer

• From vector to scalar
  – Monstrous scalar machine (Millions of core !!!)

• Fujitsu: The next gen. Scalar CPU (*Venus*)
  – Core: SPARC64 VIIIfx, 16 G flops
    • SIMD instruction extension (*HPC-ACE*)
  – CPU: 8 cores (128 G flops in total)
  –
HW configuration of “K-computer”

• 10 “Peta” flops with scalar CPU only ??!
  – Core : 16 G flops
  – CPU : 8 cores, 128 G flops
  – Computational node : 1 CPU, 128 G flops
  – System : 80,000 chassis ==> 10 P flops (est.)

• Parallelization
  – Core : Peak 16 G flops with SIMD Vectorization
  – CPU / Node : shared memory parallel
  – System : 80,000 distributed memory parallel
<table>
<thead>
<tr>
<th>Subdomain DOF</th>
<th>Num. subdomains</th>
<th>(size of coarse matrix)</th>
<th>Total DOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>30K</td>
<td>180K</td>
<td>10M</td>
</tr>
<tr>
<td></td>
<td>100K</td>
<td>600K</td>
<td>30M</td>
</tr>
<tr>
<td></td>
<td>300K</td>
<td>1.8M</td>
<td>100M</td>
</tr>
<tr>
<td>1,000</td>
<td>30K</td>
<td>180K</td>
<td>30M</td>
</tr>
<tr>
<td></td>
<td>100K</td>
<td>600K</td>
<td>100M</td>
</tr>
<tr>
<td></td>
<td>300K</td>
<td>1.8M</td>
<td>300M</td>
</tr>
<tr>
<td>3,000</td>
<td>30K</td>
<td>180K</td>
<td>100M</td>
</tr>
<tr>
<td></td>
<td>100K</td>
<td>600K</td>
<td>300M</td>
</tr>
<tr>
<td></td>
<td>300K</td>
<td>1.8M</td>
<td>1G</td>
</tr>
<tr>
<td>10,000</td>
<td>30K</td>
<td>180K</td>
<td>300M</td>
</tr>
<tr>
<td></td>
<td>100K</td>
<td>600K</td>
<td>1G</td>
</tr>
<tr>
<td></td>
<td>300K</td>
<td>1.8M</td>
<td>3G</td>
</tr>
<tr>
<td></td>
<td>1M</td>
<td>6M</td>
<td>10G</td>
</tr>
<tr>
<td>30,000</td>
<td>30K</td>
<td>180K</td>
<td>1G</td>
</tr>
<tr>
<td></td>
<td>100K</td>
<td>600K</td>
<td>3G</td>
</tr>
<tr>
<td></td>
<td>300K</td>
<td>1.8M</td>
<td>10G</td>
</tr>
<tr>
<td></td>
<td>1M</td>
<td>6M</td>
<td>30G</td>
</tr>
</tbody>
</table>
Flow of DDM with BDD pre-conditioner

Initialization

DDM CG loop

Coarse grid correction using Direct solver

Converged? yes

END

DDM body

BDD pre-conditioner

Coarse grid correction

subdomain-wise FEM
Parallelization of Band / skyline Solver

Blocking
1) on-cache algorithm
   --- scalar optimization / OpenMP parallelization
2) reduction in MPI communication cost

\[ \text{Bandwidth} = N^3 \times N^2 \]

Wider bandwidth: \( \leftarrow \) (more subdomains)
more blocks available, better load balancing
larger block size, reduction of communication latency cost
EASIER to parallelize
Huge Scale FE Analysis on T2K

Environment: Univ. Tokyo T2K (Hitachi HA8000 cluster)
   CPU: AMD Quad-core Opteron 8356 (2.3GHz) x 4 CPU
       (16 core per node)
   Network: Myrinet-10G

DDM code: BDD-diag pre-conditioning, MPI-OpenMP hybrid parallel

<table>
<thead>
<tr>
<th>Total DOF</th>
<th>Subdomain DOF</th>
<th>Num. Subdomains</th>
<th>Num. Nodes</th>
<th>Num. Cores</th>
<th>Make Coarse Matrix</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>58M</td>
<td>1300</td>
<td>58k</td>
<td>16</td>
<td>256</td>
<td>104</td>
<td>38 (0.95)</td>
</tr>
<tr>
<td>103M</td>
<td>1300</td>
<td>102k</td>
<td>16</td>
<td>256</td>
<td>251</td>
<td>62 (1.54)</td>
</tr>
<tr>
<td>100M</td>
<td>2200</td>
<td>58 k</td>
<td>64</td>
<td>1024</td>
<td>106</td>
<td>38 (0.86)</td>
</tr>
<tr>
<td>180M</td>
<td>2200</td>
<td>102 k</td>
<td>64</td>
<td>1024</td>
<td>250</td>
<td>70 (1.60)</td>
</tr>
</tbody>
</table>
Practical Applications
Seismic analysis of ancient structure
Pantheon in Roma (45M DOF)
Industry Applications: Train
Industry Applications: Car wheel
Industry Applications: Semiconductor
Other applications

Manufacturing (Stamp)

HTTR reactor

Cellar Phone

Magnetic analysis (Trans)
Pressure Vessel Model

--- from CAD to mesh ---

CAD model of nuclear pressure vessel

Mesh of pressure vessel (60 M DOF)
60 Millions DOF Mesh (details)

Size of Mesh:
- 2mm (Min.)
- 10mm (Base)
**BWR Model**
-- add internal structures --

**Internal structures:**
- A core shroud
- Control rod guide tubes
- Control rod housings
- Fuels
- Moisture separator

---

BWR: Boiling Water Reactor
CAD Model of BWR

RPV model

Cross section

Shroud

CRD housing
CRD: Control Rod Derive
CAD Model of BWR (internal structures)
Fuel rod
Fuel support
CRD Guide tube
CRD stab tube
CRD housing

(side view)

(isometric view)
(bottom view)

(side view)
200 Millions DOF
BWR
Pressure Vessel Model

Mesh (tetra-quadratic)

Elements: 39,746,750
Nodes: 67,910,224
DOFs: 203,730,672
Subdomains: 34,816
Quasi-Static Analysis of Pressure Vessel

Boundary & Loading Conditions
- Bottom surface of its skirt portion is fixed

Part Decomposition
Stress Distribution with Deformation (x 4,000)
Next Challenge: Seismic Response Analysis

--- It’s (linear / non-linear) dynamic! ---

- NS direction, Elcentro Earthquake in 1940
- Data: time increment 0.02 sec, samples 2688 points (53.74 sec)
- Analysis: NS direction → Y direction, 229 steps (4.58 sec)

![Graph showing seismic response analysis with Elcentro Earthquake data and analysis steps.]
New Target: Full-scale Power Plant Model
Building + Pressure Vessels
Acknowledgements

• Allied Engineering
• Insight
• Hitachi
• Tokyo Electric Power Co.
• Earth Simulator Center (JAMSTEC)
• Fujitsu
• Riken
• ADVENTURE Project
ADVENTURE : Open-source CAE package

Module-based Architecture : 21 modules, currently
CAD-connection, Mesh Generation,
Domain Decomposition, FE-Solvers,
Visualization, Optimization, etc.


• The result of this study is released as a solid analysis module of ADVENTURE system : ADVENTURE_Solid.
• download ADVENTURE system from the web site:

http://adventure.sys.t.u-tokyo.ac.jp/

It's open-source!
FLASH-based e-Learning system

- **Language**
  - ActionScript ver.3.0

- **Development environment**
  - Flex ver.2 (SDK for free)

- **User environment**
  - Flash player ver.9

In this e-learning system, computation is **performed** on client side (in Web browser).

1) Flash swf file is placed on web server.
2) Flash swf file is downloaded into web browser.
3) Flash swf file is executed in Flash player.
Huge scale visualization

Supercomputer

On-line visualization

High speed network

Server-side visualization
(off-line rendering)

Result data

Image files

Client-side visualization
(walkthrough rendering)

Result data

Client terminal in our laboratories
Server-side off-line visualization

- Contour and deformation on FE model surface
- Run on computational server
  - The Earth Simulator (ES)
  - PC cluster
- Time sharing system (TSS) or batch environment
  - As soon as analysis job finished
  - During analysis
- Invoke from command line or script, and generate image data
  - Image or animation files
- Polygon rendering by software only
  - Acceleration of polygon rendering using look-up table
  - Vectorization on ES
  - Parallelization (MPI)
Walkthrough visualization

• Contour and deformation plot on FE model surface

• **Run on client terminal**
  – PC or WS
  – Graphics cluster

• **Use interactively** on GUI environment
  – Need to transfer result data on model surface through network

• Visual simulation techniques (**walk-through**)
  – Visibility detection (View volume culling, occlusion culling)
  – HDDM data for bounding volume
  – **LOD**, fog

• Efficient use of graphics hardware
  – Re-use polygons on video RAM
  – Parallel rendering using graphics cluster
Pantheon Model in Roma
Image data generated by off-line rendering on The Earth Simulator

Deformation and contour plot of Pantheon, Roma
Walk-through in Pantheon
Solving Interface System

Original problem

\[ Ku = f \]

\[
\left\{ \sum_{i=1}^{\text{NDOM}} R_B^{(i)T} \left( K_{BB}^{(i)} - K_{IB}^{(i)T} K_{II}^{(i)-1} K_{IB}^{(i)} \right) R_B^{(i)} \right\} u_B
\]

\[
= \sum_{i=1}^{\text{NDOM}} R_B^{(i)T} \left( f_B^{(i)} - K_{IB}^{(i)T} K_{II}^{(i)-1} f_I^{(i)} \right)
\]

\[ K^{(i)} : \text{local stiffness matrix} \]
\[ f^{(i)} : \text{local RHS vector} \]
\[ R^{(i)} : \Omega \rightarrow \Omega^{(i)} \text{ restriction operator} \]
\[ I: \text{interior dofs} \]
\[ B: \text{interface dofs} \]

Interface problem

\[ S(i) = K_{BB}^{(i)} - K_{IB}^{(i)T} \left( K_{II}^{(i)} \right)^{-1} K_{IB}^{(i)} \]

\[ S = \sum_i R_B^{(i)T} S(i) R_B^{(i)} \]
Educational program

- Goal of education
  - Acquisition of basic knowledge necessary to use simulation (FEM)

- Learning contents
  - Mesh generation (Number of elements, Aspect ratio)
  - Boundary condition (Constraint condition, Loading condition)
  - Material properties (Young's modulus, Poisson's ratio)

- Prerequisite knowledge
  - Outline of FEM theory and procedures
  - How to use the E-learning system

Blended e-learning

- e-learning system
- + classroom based education
System architectures for e-learning

• CAI (Computer Aided Instruction)
  – Stand alone system
  – Client / Server (C/S) system

• e-learning or WBT (Web-based training)
  – HTML web application
  – Rich client
Educational program (5)

Advanced problem 1 (Cantilever beam bending problem)

Learners work on these advanced problems by themselves.

Advanced problem 2 (Simply supported beam bending problem)
The beam model is divided into quadrilateral elements.

Please note that the computing time becomes long when the number of elements is increased. Simulation accuracy improves as the aspect ratio approaches 1.
Setting loading conditions.

If you input a loading value at the center of a beam, please input the number of elements on direction x by an even number. If you don’t need a load value, please input 0 for load value.
e-learning system (Result – deformation plot)

<table>
<thead>
<tr>
<th>Mesh Generation</th>
<th>Workspace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraint Condition</td>
<td>Show deformation figure</td>
</tr>
<tr>
<td>Loading Condition</td>
<td>Magnification 100</td>
</tr>
<tr>
<td>Material Property</td>
<td>xx Show stress distribution</td>
</tr>
<tr>
<td>Analysis</td>
<td>yy Show strain distribution</td>
</tr>
<tr>
<td>Result</td>
<td>xy Show strain distribution</td>
</tr>
<tr>
<td>Comparison</td>
<td></td>
</tr>
</tbody>
</table>

Deformation plot is displayed on view area.

Displaying result of simulation.

- Show deformation figure: each component of the displacement vector is displayed.
- Show stress distribution: each component of stress is displayed by contour plot.
- Show strain distribution: each component of strain is displayed by contour plot.

If you input 100 for deformation magnification, the displacement is enlarged 100 times.
Displaying result of simulation.

Show deformation figure: displacement vector is displayed.
Show stress distribution: each component of stress is displayed by contour plot.
Show strain distribution: each component of strain is displayed by contour plot.

If you input 100 for deformation magnification, the displacement is enlarged 100 times.
Learners compare their FEM solution with theoretical solution.

Displaying Y direction displacement at loading point.

Let us compare the FEM solution with the theoretical solution.
Certification of Engineer in Computational Mechanics Field

(A) Popularization
Expansion of CM Technology into Various Eng. Fields
- A number of commercial CAE systems
- High-performance but less expensive PCs
- High quality GUI

CAE systems are getting Black-box

(B) Advancement
R&D of New CM Technologies
- HPCN (High-performance Computing & Networking)
- Nonlinear Algorithms
- Automatic mesh generation
- Adaptive technology
- Meshless technology
Grades of Certified Computational Mechanics Engineer

3rd Grade: Perform fundamental CM analyses under the supervision by higher grades CM engineers

2nd Grade: Perform reliable linear elastic analyses

1st Grade: Perform reliable nonlinear analyses

Analyst Grade: Understand theory, lead CM analysis projects

Methods: (FEM, FDM, BEM, ...)
Application Fields: (Solid, Thermal-Fluid, Vibration, ...
Image of Whole System

Objective・Reliable・Individual

CM Engineer

Provide Human Resources
Provide Human Resources (Theory・Programming)
Provide Human Resources Education (OJT) Practices

Education
Education (Software) How to use software

Examinations
3rd, 2nd, 1st, Analyst

Comprehensive Education Program (Standard Textbook, Seminar etc)

JSME
Committee of Certification of CM Engineers

Certificate

Application

Certified CAE Software Seminar

Univ.

Industry

CAE Vender

Lab.

Academic Societies
Seminar for CM Analysis at Keio Univ.
Examination in Tokyo, Nagoya, Osaka, Fukuoka
Common patterns in Simulation

• Categorization of (time marching) scheme
  – Explicit scheme
    • \([M]\) \{a\}_\text{new} = [K] \{u\}_\text{prev} + [C] \{v\}_\text{prev} + \{f\}
    • \([M]\) is a diagonal matrix \implies\text{matrix-vector product only}
  – Implicit scheme
    • \([K]\) \{u\}_\text{new} = [M] \{a\}_\text{prev} + [C] \{v\}_\text{prev} + \{f\}
    • \([K]\) is NOT diagonal \implies\text{solve a linear equation}

• How to solve a linear equation ? (implicit scheme)
  – Using a direct solver (difficult to parallelize)
  – Using an iterative solver
    • \{q\} := [A] \{p\} \text{ (matrix-vector product)}
    • alpha = \{p\}\{q\} / \{p\}\{p\}
    • \{x\} := \{x\} + beta \{p\}

All you need is Matrix-Vector Product!
Assembly of element-wise matrix into global matrix

Element-wise matrix
How matrix-vector product can be performed?

\[ \{q\} := [A] \{p\} = \sum_e [A]_e \{p\}_e = \sum_e \{q\}_e \]

- Global matrix = assembly of element-wise matrices
  - Matrix-vector product can be performed in an element-by-element manner
- Element-by-element (EBE) parallelization
  - Sub-domain = collection of elements
    - Domain-by-domain parallelization

*Element-by-element => Domain-by-domain*
Sparse matrix in FEM

node number $I$

node number $J$

nodal block matrix
How to store sparse matrix?

Sparse matrix in FEM

Non-zero component storage (CRS format)
Compressed Row Storage (CRS) format

Row Index Array (offset for each row)

Column Index Array (column ID of each non-zero component)

Value Array (non-zero component only)

A row in matrix [A] (in case of full matrix)
Automatic domain decomposition

- Algorithms for mesh partitioning
  - Recursive spatial bisection
  - Recursive spectral bisection
  - Graph partitioning

- METIS
  - Free software (You can download from The Internet.)
  - What is this?
    - Graph-partitioning library
    - Can be used for mesh partitioning also
  - How to use it?
    - Prepare mesh file
    - Use a command, “partdmesh”
    - Result: sub-domain ID for each element

**METIS : anyway, download and just use it !**
Porting to *The Earth Simulator* (Vector-type, 40T FLOPS)
The Earth Simulator: The fastest computer in the world (several years ago...)

System Configuration
- Total num. of processors: **5,120**
- Total num. of nodes: **640**
- Num. of processors per node: **8**
- Interconnected Network: full-crossbar switches

Linpack Benchmark
- **35.86** Tflops
- CPU efficiency **87.5%**

- Total peak performance: **40 Tflops**
- Total main memory: **10 TB**
- Peak performance/PN: **64 Gflops**
- Peak performance/AP: **8 Gflops**
- Shared memory/PN: **16 GB**

http://www.es.jamstec.go.jp/