SDT & OpenFEM
A technical overview

Etienne Balmes
SDTools

Structural Dynamics Toolbox

- Structural Dynamics Toolbox offers:
  - 3D Finite Element Modeling
  - Experimental Model Analysis
  - Test / Analysis correlation

- With a modular approach
  - MATLAB environment
  - OpenFEM: Core software for Finite Element Modeling (co-developed with INRIA)
  - FEMlink: Import / export industrial modules
  - Runtime SDT: Customized and standalone compiled applications

Current SDTools activities

Funding Beach (spades), SNECMA (full rotor), PSA (damping design), SANF (train-trux), VALEO (joint non-linearities).

Why MATLAB?

Requirements for a long term development environment

- Ease of development
  - Interactive operation on data structures
  - Prior declaration not required (automated memory management)
  - Interactive debugger
  - Profile

- 90% of the code fully portable

- Performance of interpretation (in-line compiler)

Application areas

Mechanics/dynamics problems are at the interface of many scientific/software areas

<table>
<thead>
<tr>
<th>CATIA, IDEAS, ProEngineer, ...</th>
<th>PATRAN, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>NASTRAN, ABAQUS, ANSYS,...</td>
<td>Adams, Simulink,...</td>
</tr>
<tr>
<td>CADA-X, IDEAS Test, ...</td>
<td></td>
</tr>
</tbody>
</table>

Preprocessing
- Mesh manipulations
- Structured meshing
- Property/boundary condition setting

Import
- Modkit, GMESH, E3D
- NASTRAN, IDEAS, ANSYS, PERMAS, SAPCEF, ABAQUS, MISS, GEODYN

FEM core
- Shape function utilities
- Element functions
- Matrix and load assembly
- Factored matrix object
- Dynamic selection of sparse library, additional solvers
- Linear static and time response (linear and non-linear)
- Real eigenvalues
- Optimized solvers for large problems, super-elements, and system dynamics, model reduction and optimization, vibroacoustics, active control
- Other software (NASTRAN, MISS)

Postprocessing
- Stress computations
- Signal processing
- 3D visualization (major extension: contour, object based)
- Export
  - VT, MEDIT
  - NASTRAN, IDEAS, SAMCEF
  - Kinem, SM3D, Gisdyn

OpenFEM, SDTools, MUSIMat

FEM (3): OpenFEM / SDT Architecture

CAD

Meshing

FEM

Simulation

Tools

Runtime SDT

FEMlink

OpenFEM

MATLAB
OpenFEM

OpenFEM toolbox (Collaboration with INRIA, LGPL license)
- Element library (2-D & 3-D, linear & non-linear, mechanics, acoustics, heat, ...), load generation, stress evaluations
- Time and non-linear solvers
- Pre- and post-processing tools

Objectives for the partners:
- Augment use of general purpose FEM in the MATLAB environment
- INRIA: prototyping environment, demonstrate viability of MATLAB/Scilab type environment for FEM computations
- SDTools: share development costs for non-core applications (2-D, non-linear mechanics, ...)

Meshing 1: unstructured

Meshing is a serious business that needs to be integrated in a CAD environment.
OpenFEM is a computing environment.

- IMPORT (MODULEF, GMSH, GID, NASTRAN, ANSYS, SAMCEF, PERMAS, IDEAS)
- Structured meshing
- Run meshing software: GMSH Driver
- 2D quad meshing, 2D Delaunay

SDT/FEMLink

- NASTRAN: read/write/drive
- ABAQUS read/write model and results (.fil)
- ANSYS: read model and element matrices, partial write model
- UFF: read/write model
- PERMAS: read/write model and matrices
- SAMCEF read model and matrices, write topology ...

- Focus on very large model and matrix support

Why NASTRAN, ANSYS ... and SDT?

Typical reasons to use your FEM and SDT

- Post-processing ⇒ extract response from full model output, generate state-space models, visualization, ...
- Pre-processing ⇒ generate parts of your job, for example in a shape optimization process
- Serve as base for your inhouse developments (example Bosch time simulation of squeal)
- Integrate pre- and post-processing in an optimization process (example PSA topology optimization for vibroacoustic response)
- Model reduction capabilities of SDT

Meshing 1: structured example

Structured meshing
- Mapped divisions
- Objects (beam, circle, tube, ...)

Meshing 2: femesh/feutil

- AddFemFEa, AddFem
- AddNode [Node], From ...
- AddTest [NodeShift, Merge]
- DivideDivisions:
  - Divide
  - DivideGroups
  - DivideGroups [ElementSelectors]
  - DivideGroups [ElementSelectors, ElementSelectors]
- Element selectors
- Extrude nRep tx ty tz
- FindDof ElementSelectors
- GetDof
- Find [El0] ElementSelectors
- FindNode Selectors
- GetEdge [Line, Patch]
- GetElemF
- GetLine
- GetNodeSelectors
- GetNormal [Elt, Node]
- GetPatch
- Info ...
- Join [el0] [group i, EName]
- Model [0]
- MatId, ProId, MPID
- ObjectBeamLine i, ObjectMass i
- ObjectHoleInPlate
- Object [Quad, Beam, Hexa] MatId ProId
- Object [Circle, Cylinder, Disk]
- Optim [Model, NodeNum, EltCheck]
- Orient, Orient i [nx, ny, nz, -neg]
- Plot [Elt, El0]
- Quad2Tria, quad42quadb, etc.
- RefineBlock
- Rename [EName] ElementSelectors
- RepeatDiv [nType] nDive origID Ang nx ny nz
- RotateSel [OrigID] Ang nx ny nz
- Sel [Elt, El0] ElementSelectors
- SelGroup [i], SelNode i
- SetGroup [i, name] [Mat j, Pro k, EGID e, Name s]
- String DOF
- SymSel OrigID nx ny nz
- TransSel tx ty tz
- UnJoin [Group 1, Group 2]

Generation, Selection, ...
Meshing 4: fe_gmsh

```matlab
FNode = [1 0 0 0 0 0; 2 0 0 0 0 0; 3 0 0 0 0 0];
model = fe_gmsh('addline',FNode);
model = fe_gmsh('write temp.msh -lc .3 -run -2 -v 0',model);
mo1 = feplot(mo1); delete('temp.msh')

• Good functionality for 2D and 3D
• Limited handling of complex surfaces
• Extensions for TETGEN and NETGEN
```

Meshing 5: selection

Recursive node and element selections

- `GID` = `EltId`
- `Group` = `EltInd`
- `DashNode` = `EltName`
- `Facing` = `x cy z`
- `NodeId > i` = `InNode`
- `rad <= r x y z` = `MatId`
- `Setname` = `SelEdge type`
- `x a` = `SelFace type`
- `x y z` = `WithNode`
- `x y z` = `WithoutNode`

FEM (1): why MATLAB?

MATLAB can be an efficient driver for all of FEM applications
- Very useful versatility in manipulating input to FEM
- Many steps can be performed in MATLAB itself
- MATLAB can be efficiently linked against external libraries (compile only the small part that needs it)
- Debugger and profiler allow quality and optimization

But
- Element level computations need to be compiled
- Argument passing without copy often critical
- MATLAB not perceived as capable of handling large jobs
- A different pricing strategy: cheap licenses but many users because interactive (need for SDT Runtime)

FEM (2): How big?

- 64 bit OS no particular limitation: 2000e6 (int32) nodes/DOFs, typical failure for very large dataset: matrix factor (no out-of-core solver), very long time response, ...
- 32 bit OS: models that have run 400.000 DOF car body, 150.000 engine blade, generally fail due to memory segmentation (largest block becomes small)

Current SDT trends out-of-core routines
- matrix read/multiply 52 bit (4096 TB)
- storage of datasets (_v.handle pointer to data in file)_ for matrices, time responses,
- coupling with external solvers (NASTRAN, MUMPS)
- Java3D rendering considered

Meshing 6: selection

Recursive node and element selections

- `GID = EltId`
- `Group = EltInd`
- `DashNode = EltName`
- `Facing = x cy z`
- `NodeId > i = InNode`
- `rad <= r x y z = MatId`
- `Setname = SelEdge type`
- `x a = SelFace type`
- `x y z = WithNode`
- `x y z = WithoutNode`

FEM (3): OpenFEM / SDT architecture

Preprocessing
- Mesh manipulations
- Structured meshing
- Property/secondary condition setting

Import
- `MeshGML, GMSH, NetGen, GID`
- `NASTRAN, IDEAS, ANSYS, PERMAS, SAMCEF, ABAQUS, MGS, GEF Dyn` (OpenFEM, SDTools, MSSMat)

FEM core
- Element functions
- Shape functions
- Matrix and load assembly
- Factored matrix objects
- Dynamic selection of sparse library, additional solvers
- Linear static and time response (linear and nonlinear)
- Real eigenvalues
- Optimized solvers for large problems, supercomputers
- System dynamics, model reduction and optimization, microelectronics, active control
- Drive other solvers (NASTRAN, MUMPS)

Postprocessing
- Stress computations
- Signal processing
- 3D visualization
- Export
- MATLAB, NASTRAN, IDEAS, SAMCEF
- EnSight, GEF Dyn

FEM (2): How big?

- 64 bit OS no particular limitation: 2000e6 (int32) nodes/DOFs, typical failure for very large dataset: matrix factor (no out-of-core solver), very long time response, ...
- 32 bit OS: models that have run 400.000 DOF car body, 150.000 engine blade, generally fail due to memory segmentation (largest block becomes small)

Current SDT trends out-of-core routines
- matrix read/multiply 52 bit (4096 TB)
- storage of datasets (_v.handle pointer to data in file)_ for matrices, time responses,
- coupling with external solvers (NASTRAN, MUMPS)
- Java3D rendering considered

FEM core
- Element functions
- Shape functions
- Matrix and load assembly
- Factored matrix object
- Dynamic selection of sparse library, additional solvers
- Linear static and time response (linear and nonlinear)
- Real eigenvalues
- Optimized solvers for large problems, supercomputers
- System dynamics, model reduction and optimization, microelectronics, active control
- Drive other solvers (NASTRAN, MUMPS)

Postprocessing
- Stress computations
- Signal processing
- 3D visualization
- Export
- MATLAB, NASTRAN, IDEAS, SAMCEF
- EnSight, GEF Dyn

ofact: gateway to sparse libraries

KeY is central to most FEM problems. Optimal is case/machine dependent. ofact object allows library independent code.

- Method: dynamic selection of method (OpenFEM, SDTools)
  - lu: MATLAB sparse LU solver
  - chol: MATLAB sparse Cholesky solver
  - pardiso: PARDISO sparse solver
  - *umfpack*: UMFPACK solver (NOT AVAILABLE ON THIS MACHINE)
  - spfmex: SDT sparse LDLt solver
  - mtaucs: TAUCS sparse solver
  - sp_util: SDT skyline solver
  - *psldlt*: SGI sparse solver (NOT AVAILABLE ON THIS MACHINE)

- Symfact: symbolic factorization (renumbering, allocation)
- Fact: numeric factorization (possibly multiple for single symfact)
- Solve: forward backward solve (possibly multiple for single fact)
- Clear: free memory
- Not tried: MUMPS, BCS-Lib, ...

- K e y f e a t u r e s:
  - Optimized sparse solvers
  - Factored matrix object
  - Dynamic selection of method
  - Optimization
  - Library independence
  - Efficient storage
  - Runtime in-core/out-of-core
  - Many steps can be performed in MATLAB itself
  - Interfacing with external solvers (NASTRAN, MUMPS, BCS-Lib, ...)
Fact (solve) CPU seconds

- All libraries can be accessible (OpenFEM, SDTools), best is application/machine dependent.
- Memory usage and fragmentation may drive library selection.

### Design criteria

- **Be a toolbox** (easy to develop, debug, optimization only should take time)
- **Optimize ability to be extended by users**
- **Performance identical** to good fully compiled code
- **Solve very general multi-physics FE problems**
- **Be suitable for application deployment**

### Easy user extensions

- Object oriented concepts (specify data structures and methods)
- But non typed data structures (avoid need to declare inheritance properties)

**Example user element**

- **Element name** of `.m` file (beam1.m)
- Must provide basic methods (node, DOF, face, parent, ...)
- Self provide calling format: `eq : beam1('call')`

```
[k1,m1] = beam1(nodeE, elt(cEGI(jElt),:), pointers(:,jElt), integ, const, elmap, node);
```

**Other self extensions**

- **Material functions**
- **Property functions (problem formulations)**
- **Non-standard topology definitions**

### Shape function utilities (integrules)

**Supported topologies are**

- `node1` (0d topology)
- `bar1` (1D linear)
- `beam1,beam3` (1D cubic)
- `quad4` (2D bi-linear), `quadb` (2D quadratic)
- `tria3` (2D affine), `tri6` (2D quadratic)
- `tetra4, tetra10`
- `pentab, penta15`
- `hexa8, 20, 21, 27`

**Standard quadrature rules**

### Formulation library

- **Generic compiled elements**
  - Support any linear element without recompilation
  - 3D elasticity with full anisotropy, 2D plane stress/strain
  - 2 & 3D acoustic fluids, fluid/structure coupling
  - Heat equation
  - Piezo-electric volumes, poro-elasticity
  - Gyroscopic effects

- **Other compiled families**
  - Geometrically non-linear mechanics, mechanical or thermal pre-stress
  - Hyper-elasticity, follower pressure

- **Other elements**
  - Shells, Laminated plate theory, Piezoelectric shells
  - 2D line/points: Bar, Beam, Pre-stressed beam, spring, bush, mass

**SDT**

The OpenFEM specification is designed for multiphysics applications.

- **99 DOF/node**
- **999 internal DOF/element**
User elements

Generic compiled elements

Objective ease implementation of
- linear element families
- arbitrary multi-physic
- good compiled speed
- provisions for non linear extensions

Assumptions
- Strain ε = [B] q linear function of N and ∇N
- Element matrix quadratic function of strain

During assembly init define
- D ↔ constit
- E ↔ EltConst NDN
- K_e ↔ D_e, E (EltConst MatrixIntegrationRule built in integrules MatrixRule)

EltConst = p_solid('constsolid', 'hexa8', [], []);

OpenFEM current developments

- Support shape function tricks (MITC, average strain, ...)
- Extend FieldAtNode interpolation
- 3D material orientation maps
- Better support of analytic expressions for loads

Waiting for real need
- Improve ease of development of NL constitutive laws
- Extend parallel operations beyond assembly

Boundary conditions

Cases define:
- boundary conditions, point and distributed loads,
- physical parameters, ...

Supported boundary conditions
- KeepDOF, FixDOF
- Rigid
- MPC, Uiso

Typical handling by elimination

Application areas

Mechanics/dynamics problems are at the interface of many scientific/software areas

CATIA, IDEAS, ProEngineer, ...
PATRAN, ...
NASTRAN, ABAQUS, ANSYS, ...
Adams, Simulink, ...
CADA-X, IDEAS Test, ...
CAD
Meshing
FEM
Simulation
Testing
SDT & not OpenFEM

- Large model optimization
- feplot (interactive mesh viewer), iiplot (curve viewer)
- Model reduction & superelements
- Sensors
- Experimental modal analysis
- Cyclic symmetry
- Active control with piezoelectric
- Parametric model analysis
- Non-conform mesh matching

MATLAB is perfectly adapted for Experimental Modal Analysis

SDT provides
- Frequency domain identification
- Test / analysis correlation (topology, expansion, correlation, ...)
- GUI based ODS analysis

Current development in the following areas
- Structural modification, hybrid test/analysis models
- Model updating
- In-operation identification
- Improved links with acquisition Photon, Pulse, NI cards, ...

Main limitation (target audience)
- Matlab based Toolbox rather than GUI application (full push-button)

SDT & Superelements

Superelement:
- group of elements identified by a name
- stored as a sub-model
- possibly reduced \( \{ q \} = [T] \{ q_r \} \)
- Possibly reused (sectors, slices, ...)

SDT provided utilities
- Select in model (and dispatch constraints)
- Display full and partial
- Partial recovery for reduced, support reduced sensors
- Reduction (Craig-bampton, free modes, many variants, possibly parameterized)
- Node renumbering

Substructuring & complex systems

Funding Bosch (squeal), SNECMA (full rotor), PSA (damping design, time domain + gyro), SNCF (train/track), VALEO (joint non linearities for lighting systems), ...

External application with SDT base
- Bosch-U. Stuttgart Flexible piping
Model reduction

- Substructuring and complex structures
  - Established
  - Multi-local substructuring, Parametric analysis
  - Current trends
    - Component design within whole
    - Time domain contact/friction
    - Thermo-elastic coupling
    - Time domain damping, self-heating...
    - Stabilize procedures for several million DOFs

Vibroacoustics

- Established techniques
  - Acoustic FEM modeling
  - Coupling matrices
  - Coupled prectures
  - Reduction methodology (critical for heavy fluids)
- Current trends
  - Linear materials
  - Design & robustness
  - Industrial process automation

Damping procedures

- Meshing
- Material handling
- Response predictions (huge challenge, but well established)
- Analysis tools
- Vibroacoustic optimization

Damping projects

- In this area SDTools worked for PSA, SNECMA, EDf, BOSCH, ONERA, GE Wind Energy, METSO ...

Damping issues

- Established techniques undergoing refinement
  - Damping device (re) - machining
  - Reduction methods for direct FRF & model
- Opened issues
  - Device placement & shape optimization techniques
  - Design methodologies (critical evaluation & validation)
  - Viscoelastic behavior & non linearities (time domain simulation, brake lining, engine supports, suspensions...)
  - Thermally challenging environments & self heating, thermoelasticity
  - Rethinking the use of piezoelectric shunts to damp vibrations above 150C
- Improve material knowledge
- Help building convincing prototypes

Vibration & surface contact

- Established
  - Optimized time integration of small displacement contact/friction coupled with superelement reduction
- Current trends
  - Design methodologies for time based simulations (application: squeal with BOSCH)
  - Equivalent linear design parameters for systems with gaps & friction (probable EDf)
  - Weld spot, rivet, bolt... dissipation (unfunded)
- Possible (seen as related)
  - Lubricated joints, high velocity abrasion or machining.

Thanks to...

- kenthiku kouzou kenkyusyo
- Université de Valenciennes, Université du Littoral Côte d'Opale, VALEO SYSTEME D'ESSUYAGE, VATTENFALL Data AB, VTT Manufacturing Technology, Vibration...