

OpenFEM: an open source finite element toolbox

Etienne Balmès, Dominique Chapelle, J.M. Leclère, Marina Vidrascu

▶ **To cite this version:**

Etienne Balmès, Dominique Chapelle, J.M. Leclère, Marina Vidrascu. OpenFEM: an open source finite element toolbox. 7e colloque national en calcul des structures, CSMA, May 2005, Giens, France. hal-01812916

HAL Id: hal-01812916
https://hal.science/hal-01812916

Submitted on 11 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

OpenFEM : an open source finite element toolbox

E. Balmès[*], **D. Chapelle**[**], **J.M. Leclère**[*], **M. Vidrascu** [**]

[*] *SDTools, 44 rue Vergniaud, 75013 Paris (France)*
balmes@sdtools.com, leclere@sdtools.com

[**] *INRIA Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)*
dominique.chapelle@inria.fr, marina.vidrascu@inria.fr

RÉSUMÉ. *OpenFEM est une boîte à outils éléments finis développée dans les environnements de calculs matriciels MATLAB et SCILAB. Cette toolbox fournit des outils de générations de modèles, d'éléments multi-physiques, de pré- et post-traitement. Cet article décrit son architecture logicielle et ses fonctionnalités.*

ABSTRACT. *OpenFEM is a finite element toolbox designed to be used within a matrix computing environment MATLAB and SCILAB. This toolbox provides general specifications for models, multiphysics elements, and self extending pre- and post-processing tools. This paper describes the software architecture and the existing tools.*

MOTS-CLÉS : *éléments finis, logiciel libre.*

KEYWORDS: *finite elements, OpenSource.*

1re soumission à *Giens'05*, le 18 février 2005.

1 Introduction

OpenFEM is a finite element toolbox designed to be used within the open matrix computing environments MATLAB (6.1 and higher) and SCILAB 3.0. This toolbox has been jointly developed by INRIA and SDTools, based on the existing software packages MODULEF[INR 04b] and SDT (Structural Dynamics Toolbox)[BAL 04]. OpenFEM is distributed under the terms of the "GNU Lesser Public License" (LGPL) and version 2.0 may be freely downloaded from www.openfem.net. Complete C, Fortran, MATLAB and SCILAB source code, documentation and compiled versions are available. Compiled parts of OpenFEM have been tested on Windows, Linux, SGI-IRIX, HP-UX, IBM-AIX, Sun-Solaris and MAC-OS.

Using matrix computing environments gives simple access to optimized numerical libraries and limits the user involvement in memory management issues. This ease of use in development has motivated the wide success of these environments. OpenFEM aims to provide all the basic functionalities needed to deal with multi-physic finite element modeling within theses environments as well as optimized compiled code (written in C and Fortran) needed to obtain performance that is very close to that of custom codes that are not designed for development flexibility.

2 Architecture

OpenFEM is fully integrated in MATLAB and SCILAB development environments. This is key to give the users freedom in developing the specific aspects of their application. Finite element solutions however involve matrix assembly and large matrix operations that need to be highly optimized to obtain performance that would be expected from a dedicated code. To achieve this performance, OpenFEM includes

– compiled C/Fortran code for element function matrix generation, stress and load computations and model assembly. These routines are common to the MATLAB and SCILAB versions (*cf.* section 3),

– the `ofact` object to allow dynamic selection of sparse matrix library routines (*cf.* section 6) for matrix factorization operations,

– optimized self extending functions for model matrix assembly, stress and load computations, as well as time integration. The optimization focuses on memory allocation, argument transmission and automated support of user developments. This optimization strongly limits the loss of performance often found when using high level matrix languages with very flexible memory allocation strategies.

The definition of the models, the boundary conditions, the loads and the results use high level data structures. A finite element model data structure contains fields for

– geometry definition (elements and nodes),

– material components and mechanical properties (for shells, beams, etc.), with automatic unit conversion handling,

– boundary conditions and load cases handling (*cf.* section 5).

OpenFEM provides numerical post-processing tools (stress computations, etc.), basic visualization in MATLAB and SCILAB, and export to the M-EDIT format for visualization [INR 04a]. Paying visualization tools for the MATLAB environment are also available from SDTools.

3 Element level computations

OpenFEM contains three families of elements.

`m`-file elements are written as MATLAB/SCILAB functions. While they run somewhat slower than compiled elements they are the easiest to implement for new users. Current elements implemented as `.m` files are beams (including prestress), bars, springs, concentrated masses, 3 and 4 node shell elements.

`mex`-files are self contained elements. The `of_mk` MEX gateway supports a number of elements where the element mass, stiffness, stress and load computations are written as self-contained FORTRAN programs. These are translated to C using the `f2c` utility [FEL 95] and interfacing calls from OpenFEM are grouped in the `of_mk` gateway. Elements implemented with this strategy include

– 2-D (plane stress, plane strain and axisymmetric) linear and quadratic elements for triangles and quadrangles,

– 3-D linear and quadratic volumes with isotropic, orthotropic elastic behavior,

– shell elements as the MITC4.

The last family of elements, currently only available in the CVS version of OpenFEM, implements a generic strategy for building multiphysic elements. The element is fully described by an `.m` file but all element level operations are compiled. The element description details

– the relation (assumed to be linear) between the strain field and the vector fields and their gradient within the element,

– the topology of the constitutive law matrices,

– the integration rule.

Elements currently implemented within this family are fully anisotropic 3-D solid elements and acoustic fluid elements. Implementation of non linear geometric stiffness and non-linear mechanical elements is currently under way.

4 Model generation, import and manipulations

Meshing tools are provided for finite elements model creation and handling. Parameterized meshes may be automatically generated. Element and node manipulations,

as selection, extrusion, revolution, etc. are possible. This is short example of how to divide and extrude a mesh :

```
FEel0=[Inf abs('quad4');1 2 4 3 1 1];
femesh(';divide 5 3;extrude 10 0 0 .25');
```

Filters to import models from MODULEF and GMSH formats are included in the OpenFEM distribution. Paying filters to import models from NASTRAN, ANSYS, IDEAS, PERMAS are available from SDTools.

5 Boundary condition and loads

Mechanical problems data are handled using script based manipulations of cases defining : boundary conditions, point and distributed loads, physical parameters, ... The various cases are stored in the `Stack` field of the model data structure. It is a `cell array` containing the name, the type and the data for the definition of DOFs or load case.

This is an example of how to define a pressure load on a selected face and fixed boundary condition on $x = 0$:

```
data=struct('sel','x==-.5', ...
            'eltsel','withnode {z>1.25}','def',1,'DOF',.19);
model = fe_case(model,'AddToCase 1',...
                'Pressure','Pressure load',data,
                'FixDof','Fixed boundary condition','x==0');
```

Matrix and RHS asembly are supported by the `fe_mk`, `fe_mknl` and `fe_load` functions.

6 Solvers

The basic bloc of most finite element solvers is the resolution of a problem of the form

$$\boldsymbol{K}\boldsymbol{q} = \boldsymbol{F} \tag{1}$$

which is really solved in multiple steps. Assembly or reassembly of $\boldsymbol{K}$, renumbering and symbolic factorization of $\boldsymbol{K}$, generation of $\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\boldsymbol{T}}$ or $\boldsymbol{L}\boldsymbol{U}$ factors, solution of equation 1 with $\boldsymbol{F}$ and $\boldsymbol{q}$ given in original DOF numbering while the K factors are given in a numbering optimized for the resolution.

Non linear problems typically involve reassembly of a $\boldsymbol{K}$ matrix at multiple steps with minor or no change the matrix topology. The `fe_mknl` functions decouples

assembly in two separate steps. During initialization, DOF numbering, constitutive law and internal state initialization, assembled matrix preallocation, ... are performed. Then at each step, the matrix $\boldsymbol{K}$ can be assembled with a good performance.

The factorization of $\boldsymbol{K}$ needs to be optimized and may be reused a number of times. The sparse matrix libraries provided with MATLAB and SCILAB give poor results in FEM applications and do not provide a mechanism for separating the steps of factorization and solve. OpenFEM thus includes an `ofact` object that allows code development that is independent of the sparse library that will be used eventually.

OpenFEM provides solvers for static response to loading case. Non-linear solvers are currently developed and need optimizations. For dynamics, eigenvalue (Lanczos) and transient (implicit and explicit Newmark, Time discontinuous Galerkin, linear and non linear) solvers are also provided.

7 Sample applications

This section illustrates some specific applications based on the OpenFEM library.

The OSCAR project is a collaboration between SDTools and SNCF (Direction de l'Innovation et de la Recherche) to simulate the pantograph-catenary interaction for high speed lines (see figure 1). Prestress beam elements and Newton algorithm are used to determine the initial catenary state (pretension building phase). Contact research algorithm has been specifically developed and the `fe_time` functions is used (explicit version) to simulate the interaction on hundreds meters of railway (approximately 500000 time steps). The result takes a few minutes.

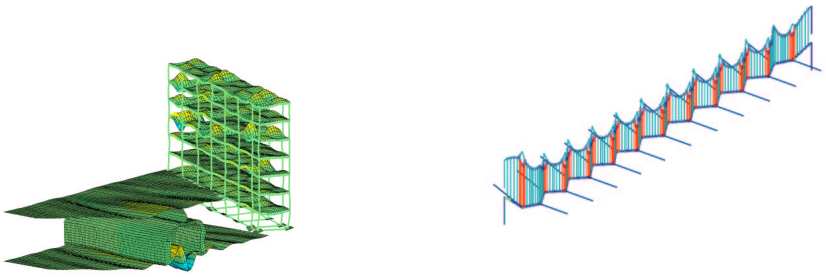


Figure 1 – Soil-structure interaction (ECP-MSSMat). Pantograph-Catenary interaction (SNCF-DIR).

Other applications using OpenFEM may be cited :

– Soil-structure interaction using MISS-OpenFEM interface (see figure 1).

– Biomechanics. Cardiac electromechanical activity modeling using fluid elements of OpenFEM (see figure 2).

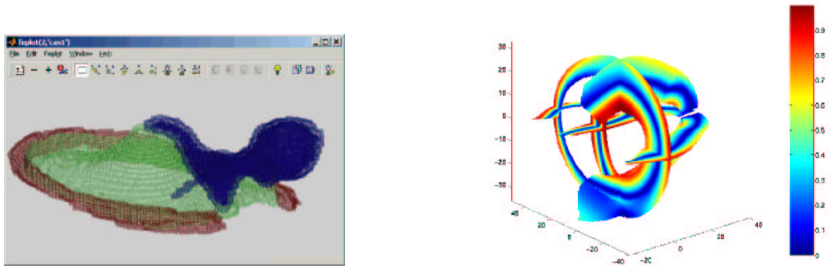


Figure 2 – Internal ear modeling (University Hospital Zuerich). Heart simulation (INRIA, Západoĉeská Univerzita).

– Biomechanics. Anisotropic volume modeling of an internal ear (see figure 2).

8 Conclusion and perspectives

OpenFEM provides a general framework to help the development of FEM solutions with the MATLAB and SCILAB matrix computing environments. The integration within these environments greatly simplifies code development and the development of OpenFEM was efficient with a major focus on optimization to obtain the performance of dedicated compiled codes.

Current topics of active development are time domain simulations of structures with material and geometric non-linearities, building of gyroscopic matrices for rotating machinery, enhancements to the post-processing capabilities in non-linear mechanics, enhanced shell models for composite and piezoelectric applications.

The development team is quite open to external contributions.

9 Bibliographie

[BAL 04] BALMES E., LECLÈRE J., *Structural Dynamics Toolbox 5.2 (for use with MATLAB)*, SDTools, Paris, France, www.sdtools.com, October 2004.

[FEL 95] FELDMAN S., GAY D., MAIMONE M., SCHRYER N., *A Fortran-to-C Converter*, AT&T Bell Laboratories, Murray Hill, NJ 07974, 1995.

[INR 04a] INRIA, *Documentation M-EDIT*, INRIA, Rocquencourt, France, www-rocq.inria.fr/gamma/medit, 2004.

[INR 04b] INRIA, *Documentation MODULEF*, INRIA, Rocquencourt, France, www-rocq.inria.fr/modulef, 2004.