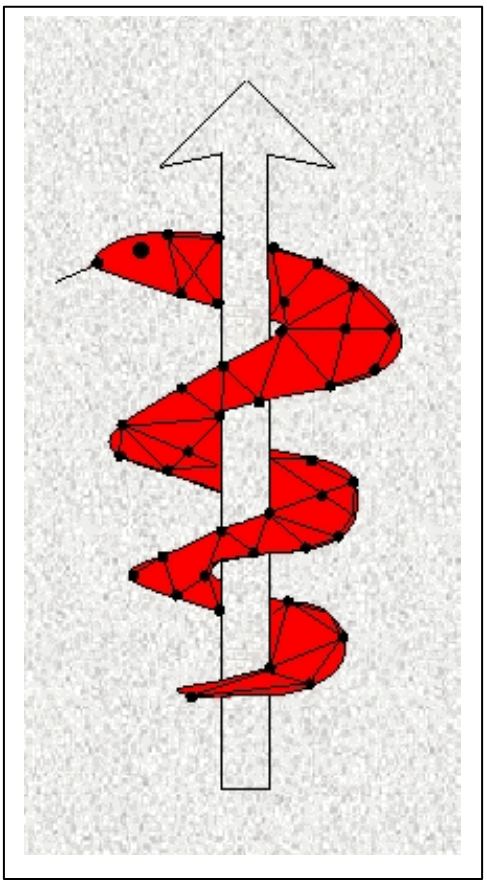

## The IST Programme
**Project No. 10378**

# SimBio

## SimBio - A Generic Environment for Bio-numerical Simulation

**http://www.simbio.de**



## Deliverable 3c
## Numerical Solution System
## Final Release Notes

| | |
|---|---|
| Status: | Draft |
| Version: | 2.0 |
| Security: | Public |
| Responsible: | NEC |
| Authoring Partners: | NEC, MPI, ESI |

Release History

| Version | Date |
|---|---|
| 1.0 | 03.04.2002 |
| 2.0 | 14.10.2002 |

**The SimBio Consortium:**
NEC Europe Ltd. – UK
A.N.T. Software – The Netherlands
CNRS-DR18 – France
Sheffield University – UK
MPI of Cognitive Neuroscience – Germany
Biomagnetisches Zentrum Jena – Germany
ESI Group – France
Smith & Nephew - UK

# Table of Contents

# 1. Introduction

The numerical solution system (NSS, work-package 3) is a core module of the generic SimBio environment. Figure 1 shows the interaction of the NSS with other SimBio modules.

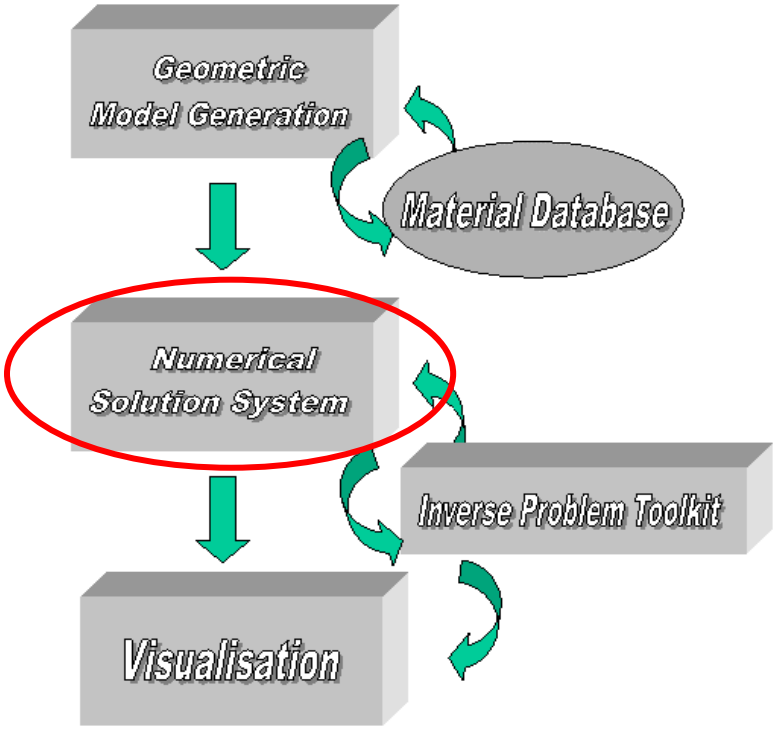



**Figure 1: Position of the numerical solution system in the SimBio project.**

This document is focused on the description of the final software release. It summarises the technical progress of the NSS components and their integration and the installation, verification and use of the software. The content of the numerical solution system and a detailed explanation of each tool together with preliminary and intermediate results have been presented in detail in deliverables D3a and D3b. The final version of the NSS comprises:

**I. Linear Solver Libraries and Partitioning Tool**

1. **DRAMA**, a tool for partitioning finite element meshes for efficient parallel execution.
2. **PEBBLES**, public domain library for parallel equation solving.
3. **PILUTS**, a highly efficient NEC linear equations solver library (**P**arallel **I**ncomplete **LU** with **T**hreshold preconditioned **S**olvers). The solver routines can be executed as parallel standalone tools or can be called directly as parallel subprogram from an application code parallelised by domain partitioning.

**II. Full Problem Solvers**

4. **NeuroFEM (was CAUCHY)**, a full FE code for parallel execution with coupling for parallel solvers provided.
5. An interface for **PAMSAFE**, a fully non-linear FE code for biomechanical applications.
6. **HeadFEM**, a non-linear fully parallel FE code for special biomechanical applications.

The remainder of this deliverable is organised as follows. For each of the components listed above the final status of implementation of the software is illustrated by a description of the functionality of its final version and an explanation how to access, install and use the tool(s).

# 2. DRAMA Repartitioning Tool

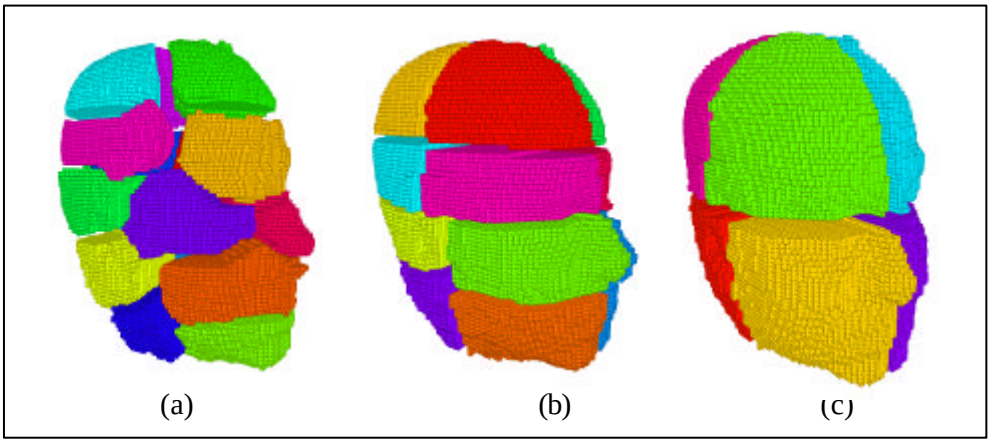



**Figure 2.1: Three partitioned head meshes, (a) PJOSTLE 16 sub-domains, (b) RCB 16 sub-domains, (c) RCB 8 sub-domains.**

## *2.1 Introduction*

The DRAMA library supports dynamic load balancing for parallel message-passing mesh-based simulation codes. Although the library was originally developed for applications with dynamic, solution-adaptive mechanisms, it can equally well be applied to static problems. Based on a very general cost model, DRAMA provides access to a range of parallel partitioning algorithms through a simple and effective mesh-based interface. The repartitioner tool that is based on the DRAMA library provides load-balancing, matrix partitioning and basic data-migration capabilities to the NSS components.

## *2.2 Final Implementation*

The repartitioner tool allows one to use the VISTA file format for platform independent, compact binary input and output. It supports different material models in the form of material labels for the elements. The DRAMA tool consists of the following subroutines:

- drama_simbio_part
  - read_vista_header
  - allocate DRAMA mesh
  - read_vista_mesh
  - repartition
    - read DRAMA options
    - DRAMA_INIT
    - set DRAMA options
    - cost function evaluation
    - DRAMA geometric/graph/mesh migration
    - cost function evaluation
    - migrate data
    - write_pmvis_mesh (optional, see [pmvis])
    - DRAMA_FINALIZE
  - write_vista_mesh

The tool is able to exploit the features of the DRAMA library, which give special support to the SimBio applications for the partitioning of sparse symmetric matrices. The repartitioner tool uses file interfaces based on a common ASCII or VISTA format definition. It can be linked with VGrid, NeuroFEM, HeadFEM and the PILUTS solver (standalone) tool in a linear chain (see Figures 5.3 and 7.2). The VISTA mesh file format definition is a platform independent, compact file coupling format. It starts with an ASCII header followed by lists of binary data. The details of the format definition are described in D1.2b section 2.1.4.

## *2.3 How to access, install and use the partitioner tool*

The use of the partitioner tool requires that the DRAMA library is installed on the system. The entire software can be obtained in the following two steps:

1. Download the DRAMA library from its homepage: http://www.ccrl-nece.de/DRAMA/.
2. Download the partitioner tool (DRAMAtool.tar.gz) from the protected part of the SimBio webpage: http://www.simbio.de/.

The following steps are necessary to install and verify the software:

1. Unpack the release: tar –xzf DRAMAtool.tar.gz
2. Change directory to desired working directory (i.e. $DRAMAtool_dir)
3. Edit the Makefile and adjust the system dependent information.
4. Build the partitioner: make
5. Set appropriate options in the DRAMA.options file.
6. Verify installation: run_test_suite (shell script provided with the final release).

The verification step involves the execution of 7 different test examples. The actual results are compared to reference data and a status (“passed” or “failed”) is reported. For an implicit solver the file DRAMA.options typically should have the following content:

```
!-1-----------------------------------------------------------------------------------------
! 2   Copyright 2000 by NEC Europe Ltd.
! 3   C&C Res. Lab. Sankt Augustin
! 4   file DRAMA.options,        version date 17.07.2000 JF
!-5-----------------------------------------------------------------------------------------
  1              number of computational phases
  0              detailed
  4              graph type
 10              partitioner type
  0              module type
  1              DRAMA_cost_prediction_choice    0: off   1: on
 92              geoinp           91: GEOnodes 92:GEOelements
991              geomod         991:bucket   992:simple
9992             geoversion    9991:DREDUCT 9992:DMOVE
  0              ncflags   0: const   1: const+deg. dep.  2: nopn(1:nnodes)
  0 1 0 1 0      costmodel switch
  0.0 90.0 0.0   cost model parameters
  0              computing environment
  0              renumbering strategy
```

To run the tool follow the following 3 steps:

1. Create VISTA input file: input.v
2. Submit a parallel job: mpirun –np 4 drama_simbio_part.exe
3. The partitioned output mesh is found in the file: output.v

# 3. AMG-preconditioned CG (PEBBLES-Solver Library)

## *3.1 Introduction*

The kernel of the NeuroFEM-simulator, described within ST4.1, is the FE-solver system. Thousands of large equation systems with symmetric positive definite stiffness matrix have to be solved within the inverse source localisation procedure. Preconditioned conjugate gradient methods are under the most efficient methods for solving such large equation systems. Since the geometry and thus the stiffness matrix stays the same for an inverse source localisation for one patient, the effort for the set-up of the preconditioner can be neglected compared to the solver part. Since a high resolution of the FE-head model is necessary with respect to the inclusion of tensor-valued material properties of e.g. skull and white matter, a multigrid-approach seemed to be appropriate for the construction of such a preconditioner. An algebraic approach (see [rei00, haa00]) is very attractive compared to a purely geometrical multigrid because the generation of a hierarchical grid together with an optimal tuning of the operator and the inclusion of anisotropic tensor-valued material properties within a geometrical approach would be quite difficult. A first comparison between a serial AMG-preconditioner and different serial threshold-factorisation preconditioners in realistic head models showed a superior behaviour of the AMG-CG approach especially for very high relative solution accuracies [wol00]. In that paper, the coupling of the solver methods to the FE-code (at that time a pure serial FORTRAN77-code) was realised by means of a "loose" file-coupling, where the files included the stiffness-matrix in compact row format. For these comparisons, no knowledge about the starting vectors was assumed (**0** starting vector).

Since it should be possible to calculate better starting vectors through the use of techniques for multiple right-hand sides [chan97] and since a final statement about the necessary absolute solver accuracy can only be given after having tested the resolution possibilities of FE-based source localisation and the sensitivity towards the material properties in a later state of the SIMBIO-project, a parallel development/integration of the parallelised algebraic multigrid as well as parallelised threshold-factorisation preconditioners, described in section 4, was important for the NeuroFEM-kernel.

## *3.2 Final Implementation*

The parallel version of the algebraic multigrid preconditioned CG method (parPEBBLES, [haa00]) has now been "strongly" integrated into the parallelised version of NeuroFEM, parNeuroFEM, by means of an element-wise coupling. A partitioning of the dual graph of the FE-head-model is calculated through the use of the public domain METIS partdmesh-function. The FE-mesh together with the tensor-valued material properties are distributed element-wise to the processors. The definition of parPEBBLES basic class within the basic NeuroFEM simulator class, which organises a dynamical memory management of the AMG-CG structures, opens the possibility for the use of deeper PEBBLES-structures, derived from this basic class. In the setup-phase of parNeuroFEM, the connectivity structure of the FE-geometry (element-nodes list) and in a further step the element stiffness-matrices are handed over to PEBBLES class structures, organising a storage in compact row format of the stiffness matrix. This process is fully parallel. The Dirichlet-node (EEG reference electrode) information is send to all processors which are part of the partition-index-set of that node and implemented in the local sub-matrices by means of a penalty-approach, leading to a regular matrix and thus a unique solution for the potential. The hierarchy of stiffness- and prolongation/interpolation-matrices, again a fully dynamical process, are determined exclusively through the use of the algebraic information in the stiffness-matrix on the finest level (strong neighbour relationship between the nodes). A linear interpolation of the potential for each finer grid node through all neighbouring coarse grid nodes was chosen, which turned out to be most efficient in our simulations compared to more sophisticated interpolation

techniques (see [wol00, wol02a]). Since the calculation of the source load is extremely fast compared to the local potential computations, the root-process calculates the right-hand side vector and then distributes it to the processors. After allocation of a starting vector (currently the **0** vector), the solver module is called using one smoother step as Pre- and Post-Smoother in each iteration. We use a Gauss-Seidel smoother for the inner nodes and a Jacobi smoother for the interface nodes [haa99]. The factorisation on the coarsest grid level is carried out by an $LL^t$ factorisation. Figure 3.1, taken from [wol02a, wol02b], shows the wall-clock time from 1 to 12 processors for the solver part of the parallel AMG-CG compared to a parallel Jacobi-CG up to an accuracy of $10^{-8}$ for two different head models, calculated on an SGI Origin. The number of necessary iterations is shown over the curves.

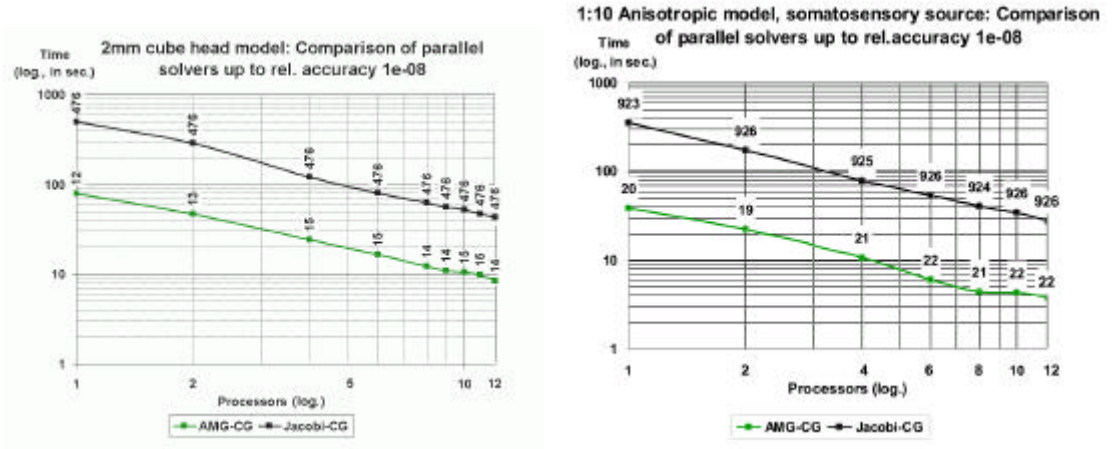



*Figure 3.1: SGI-Origin: Wall-clock time from 1 to 12 processors for the solver part of the parallel AMG-CG compared to a parallel Jacobi-CG up to an accuracy of $10^{-8}$, taken from [wol02a,wol02b]. The number of necessary iterations is shown over the curves. Left: Realistic isotropic 2mm cube head model with 325384 nodes. Right: Realistic anisotropic tetrahedral head model with 147.287 nodes.*

The Jacobi-CG is a well-known solver method in FE-based source localisation. As shown in Figure 3.1, we achieved a nearly linear speed-up for the presented isotropic cubic head model as well as for different tested isotropic and anisotropic high resolution tetrahedral head models [wol02a, wol02b]. For the anisotropic model in Figure 3.1, right, the anisotropy ratio for the head tissue layers skull (radial: tangential to skull surface) and white matter (transverse: longitudinal to the fibres directions) were set to 1:10. More information about the anisotropy modelling will be given in subsection 5.2.

## *3.3 How to access, install and use the Software*

The solver library as kernel of the NeuroFEM-simulator is integrated as FE-simulation tool in the ST4.1 release. It can be accessed via the interface functions defined in the release notes of ST4.1. The software is tested and optimised on Linux and SGI-Irix using the Gnu C++-compiler. The ST4.1 release contains libraries and a binary command line tool for Linux and SGI-Irix. Additionally, a NeuroFEM version is released as a standalone-application with file-coupling to the inverse toolbox via lead field matrix computation.

# 4. NEC Solver Library PILUTS

## *4.1 Introduction*

The PILUTS library includes parallel sparse solvers for real symmetric positive definite (spd), general real symmetric and real non-symmetric matrices. As basic iterative methods, the **C**onjugate **G**radient (CG) algorithm, a **sym**metric variant of the **Q**uasi-**M**inimal **R**esidual method (symQMR) and the **Bi**-**C**onjugate **G**radient **stab**ilised (BiCGstab) algorithm are provided [bas00, bas99, bas96, saad96]. CG is applied to equation systems with spd matrices, QMR usually to systems with general symmetric matrix, and BiCGstab usually to systems with non-symmetric matrix. For convergence acceleration of the basic iterations, a selection of preconditioning methods is available. The preconditioners include scaling methods, symmetric or non-symmetric incomplete block factorisations with threshold and **D**istributed **S**chur **C**omplement (DSC) algorithms [bas00, saad99, saad96].

## *4.2 Final Implementation*

All PILUTS methods are implemented in such a way that the solver routines can be executed as parallel standalone tools or can be called as parallel subprograms in an application code parallelised by domain partitioning. In the former case, the equation system data is read piece-wise from a file and distributed to the processors available. The solution process itself is performed fully in parallel afterwards. In the latter case, the parallel application code provides the distributed equation system data to a local PILUTS subroutine which then solves the matrix problem in parallel.

The DRAMA tool can determine the data distribution for the PILUTS CG solver. The solver can read the partitioning information from a small DRAMA tool file (default name: “dist.dat”) containing the number of nodes in each sub-domain. With this information the solver can distribute the matrix data accordingly.

The PILUTS environment additionally provides the following features: First, the distributed equation system data can be re-partitioned by ParMETIS and re-distributed in order to reduce the couplings between sub-domains and in order to accelerate the convergence of block factorisations or DSC preconditioners. Second, the distributed matrix data can be re-ordered in order to reduce fill-in for local, incomplete decompositions.

The PILUTS methods CG, symQMR, and BiCGstab can be called by using the same interface. The specific iterative solver is selected by a parameter.

For preconditioning CG, simple diagonal scaling, block **I**ncomplete **C**holesky decomposition with **T**hreshold (ICT), block **I**ncomplete $\boldsymbol{LDL^T}$ factorization with **T**hreshold (ILDLT), both with preceding diagonal scaling, DSC preconditioning with preceding diagonal scaling using ILDLT for local diagonal matrix blocks, and DSC using complete $LDL^T$ decompositions of the local diagonal matrix blocks, are available [bas00, saad99, saad96].

Simple diagonal scaling, block ILDLT with preceding diagonal scaling, DSC preconditioning with preceding diagonal scaling using ILDLT for local diagonal matrix blocks, or DSC using complete $LDL^T$ decompositions of the local diagonal matrix blocks can precondition the current PILUTS symQMR solver.

For preconditioning BiCGstab, simple diagonal scaling, simple row and column scaling, block **I**ncomplete $\boldsymbol{LU}$ factorisation with **T**hreshold (ILUT), block ILUT with preceding

diagonal scaling, block ILUT with preceding row and column scaling, DSC preconditioning using ILUT for local diagonal matrix blocks, DSC preconditioning with preceding row and column scaling using ILUT for local diagonal matrix blocks, and DSC using complete *LU* decompositions of the local diagonal matrix blocks are available [bas00, saad99, saad96].

The final PILUTS library includes the possibility of the re-use of methods (re-partitioning, fill-in reducing re-ordering, communication scheme, preconditioners) if the solver is called several times. This means the permutation vectors for re-partitioning and fill-in reducing re-ordering or all information on the communication scheme for matrix-vector operations have to be computed only once during the first solver call. This proceeding makes sense if the sparsity pattern of the matrix does not change in a non-linear iteration or in successive time steps. In addition, a once computed preconditioner can be applied several times. This is especially useful if systems are solved with the same matrix but different right hand sides as in the NeuroFEM case.

The PILUTS library was successfully integrated and tested in the simulation codes HeadFEM and NeuroFEM.

### *4.3 How to access, install and use the NEC PILUTS library*

The PILUTS software can be obtained from the SimBio web page. The following steps are necessary to install the software:

1. Download the PILUTS package (piluts.tar.gz) from the protected part of the SimBio webpage: http://www.simbio.de/
2. Unpack the release: tar –xzf piluts.tar.gz
3. Change directory to piluts
4. Edit the Makefile and adjust the system dependent information.
5. Build the PILUTS library: make
6. To use the library link the object code with *libpiluts.a*.

The PILUTS library comes with a standalone test environment and a set of sample matrices. The matrix data are provided in files. Within this environment, all PILUTS features including re-partitioning and re-use of methods can be tested. You find sample calls of all PILUTS modules and sample parameter settings for all PILUTS solvers.

# 5. NeuroFEM

## *5.1 Introduction*

The inverse problem in EEG and MEG amounts to finding a realistic source distribution in the human brain for a given set of field observations on the surface of the head. This requires the repeated simulation of the field propagation for a given dipolar source in the brain using a volume-conduction model of the head. For most realistic modelling, the different tissues have to be segmented and assigned individual conductivity tensor material parameters. As a basis for the development of the NeuroFEM-software, the software package CAUCHY'97, described under
http://www.rwth-aachen.de/neurologie/Ww/Neurologie/cauchy/CauchyFunctionality.htm
and in [buc97, rien97, wol99], was taken and strongly redesigned.

## *5.2 Final Implementation*

A C++ class-structured software replaces old FORTRAN77 CAUCHY'97 kernel routines and enables the development of the SIMBIO-software on parallel platforms. The NeuroFEM-simulator has been derived from an abstract simulator class defined in the design report of ST4.1. A hierarchic class structure is used to reduce the number of interfaces and to keep them clean of implementation details. The class structure of the inverse toolbox is described in the release notes of ST4.1. The toolbox of ST4.1 provides abstract class interfaces for grid generators, forward simulators using a discrete and continuous search space. NeuroFEM tools are interfaced to classes that are derived from these abstract classes.

In stating, that head tissue conductivity anisotropy has a non-negligible influence on the field propagation and should therefore be taken into account, our findings in [wol02b] are in agreement with published results of [mar98, hau02]. In [wol02b], we presented non-invasive measurement techniques and methods from image processing for obtaining a high resolution anisotropic map of the two tissue layers skull and white matter. Our presented head model was constructed from an MRI segmentation of the five tissue layers skin, skull, cerebrospinal fluid and brain grey and white matter. The modelling of skull anisotropy was based on dual-echo MRI data, allowing especially an improved segmentation of the inner skull surface [bur02]. The radial eigenvectors of the skull conductivity tensors were obtained by means of the surface normals of a smooth surface spongiosa model, yielding accurate eigenvector directions for the resulting conductivity tensors. Whole head DT-MRI measurements were exploited in order to model the conductivity tensor eigenvectors for each point in the white matter layer by means of the corresponding diffusion tensor eigenvectors We used high resolution realistic finite element head modelling, where a measured/modelled tensor-valued conductivity was assigned to each finite element in the skull and white matter layer. Fig. 5.1 shows conductivity tensors in the barycentre of the finite elements for an anisotropic head model, using VM from WP5 as a visualisation tool.

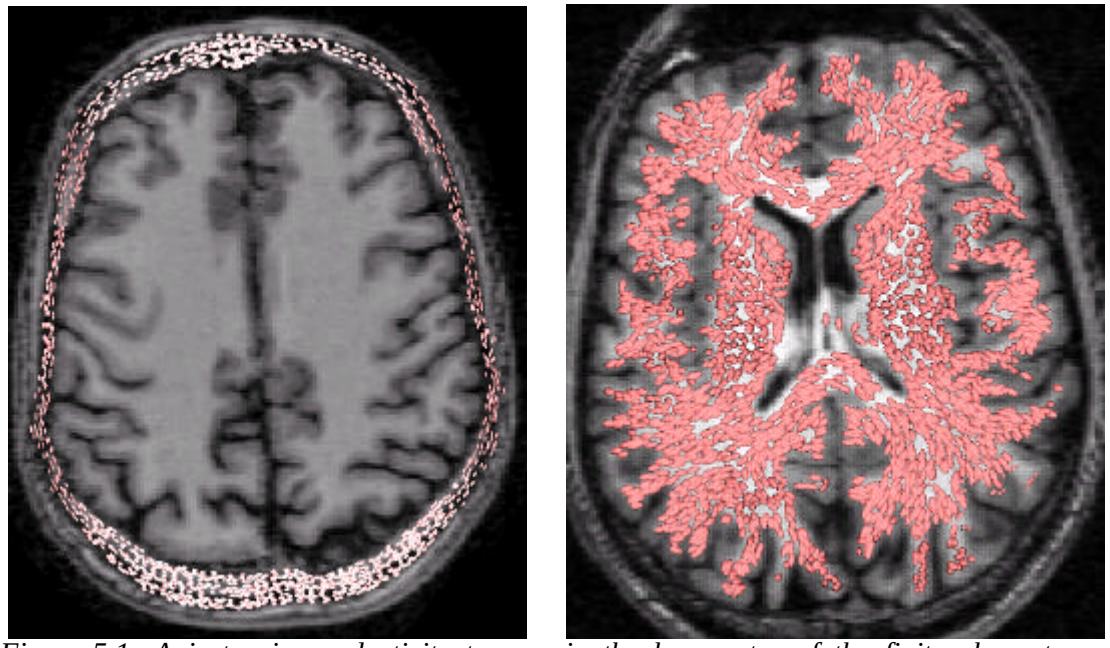


*Figure 5.1: Anisotropic conductivity tensors in the barycentre of the finite elements on underlying T1-MRI: Left: Tensors of the skull. Right: Tensors of white matter layer. The tensors were visualised with the VM-tool of WP5.*

In order to test the influence of anisotropy on the EEG, different anisotropy ratios were simulated. Our results in [wol02b] agree to a large extent with published results of [mar98, hau02].

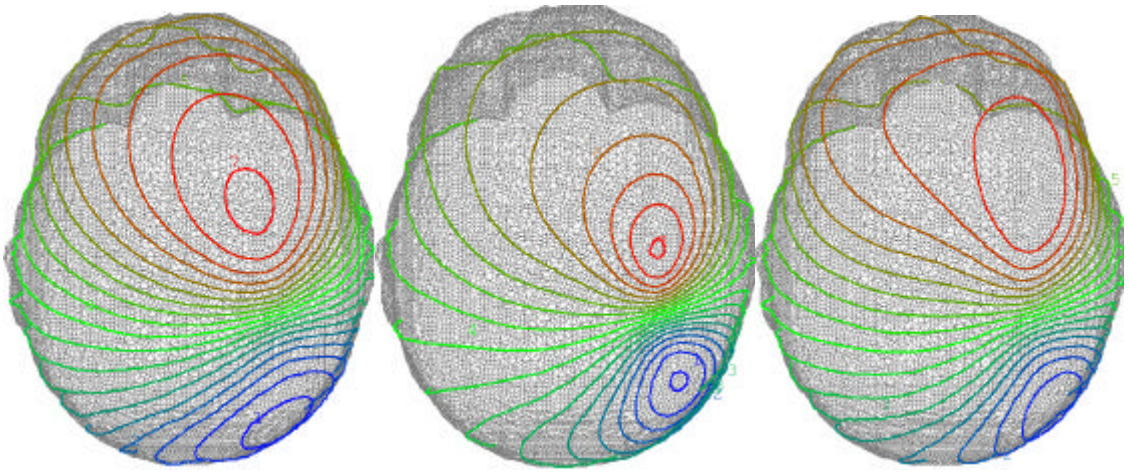


*Figure 5.2: Topography of an iso-potential distribution of a mainly tangentially oriented eccentric source on the upper part of the head. Left: Isotropic skull, from –0.9 to 0.2 uV. Middle: 1:10 anisotropic skull, 10 times higher tangential conductivity, from –2.2 to 0.9 uV. Right: 1:10 anisotropic skull, 10 times lower radial conductivity, from –0.7 to 0.1 uV.*

As an example, Fig. 5.2 shows the changes in the topography of the isopotential distribution of a mainly tangentially oriented eccentric source on the upper part of the head for isotropic (left) and anisotropic (middle and right) skull modelling. We conclude, that for robust EEG-based dipole source reconstruction in the human brain, realistic tissue conductivity anisotropy of the skull as well as the white matter layer has to be taken into account [wol02b]. This is not possible with a Boundary Element Method approach or computations in a multilayered sphere model [mun92], so that NEUROFEM's fast high resolution finite element head modelling approach is necessary. The solver process was shown (see subsection 3.2.2) to be stable with respect to realistic tissue anisotropy (see Fig. 3.1). The presented methods and software

concepts enable the study of the influence of tissue conductivity anisotropy on the inverse source localization results in future examinations.

In EEG/MEG-source localisation, the source is usually modelled as a mathematical equivalent current dipole, i.e. a current source and a sink which are infinitively close together in the human cortical layer. This point-like equivalent current dipole has been shown to be an adequate model for the synchronous polarisation of a cortical surface of about 30mm$^2$. The point-like source directly leads to a singularity in the related potential which has to be treated numerically. One possibility is the "blurred dipole" where current monopoles are placed at neighbouring FE-mesh nodes around the dipole location such that the resultant moment matches that of the mathematical dipole [buc97]. Another possibility is the subtraction method where the "singularity-potential" for a mathematical dipole in an unbounded homogeneous conductor is calculated analytically and the correction is carried out numerically on the realistic geometry ([rango96, mar98] ). The correction is calculated with the FE method so that the sum of "singularity-" and "correction-", the "total-potential", obeys the charge continuity law within the head and the Neumann boundary conditions at the surface. The subtraction method and the node-shift (ns) mesh generation approach (see ST1.2. report) have been validated in a 4-layer sphere model where a spherical harmonics series expansion of the dipole potential can be derived for anisotropically conducting layers [mun88, mun93]. To validate the subtraction method, we assumed the following isotropic conductivities in the 4 layer model: 0.33, 0.0049, 1.0 and 0.34 S/m. Together with node-shifted FE-mesh, the magnification error (optimum 1) was 1.053 and the relative difference measure (optimum 0) 0.023 for 6 electrodes at all extreme sphere surface positions. The results were visualized in D3b.

The integrated class structure allows comparisons with boundary-element-based forward simulations and analytical series expansion formulas for spherical shell geometries, which both are also derived from the abstract simulator-class. It now enables influence-studies of tissue anisotropy on the various inverse algorithms of the ST4.1. toolbox. A dynamical memory management has been introduced throughout NeuroFEM, which replaces the former static allocations and enables a user-friendly application on distributed memory platforms. The NeuroFEM-simulator can now be used as a forward simulator for discrete and continuous inverse source localisation methods.

For the tight coupling between ST4.1 and WP3 in the case of a continuous parameter space for the inverse reconstruction described here, the NeuroFEM tool was extended to provide a source simulation also for dipoles which are not on the discrete grid, but in the continuous space. In the case of a discrete parameter space, the coupling between WP3 and ST4.1 is on a simple file level or via close coupling. The lead-field matrix can also be directly handed to the inverse tools.

### *5.3 Software Integration*

The FE mesh is stored in a Vista graph file (see D1.2a) that also contains the tensor-valued material properties. After a partitioning by means of the DRAMA-tool, NeuroFEM reads the partitioned geometry and creates the lead field matrix in binary Vista format. The lead field matrix is stored as a two-dimensional Vista image file. The tool was extended to read a separate surface grid as influence space, extracted from segmented volume data by means of a *marching tetrahedra* algorithm described in the release notes of ST1.2. This surface grid represents the brain surface or the cortex on which the lead-field matrix is computed.

For a tight coupling of NeuroFEM and the inverse toolbox, the dipole source reconstruction result is written in a vista grid structure and can then be visualised with tools from WP5. The inverse toolbox contains a variety of state of the art inverse source reconstruction algorithms.

Figure 5.3 shows the software chain from mesh-generation to source reconstruction visualisation.

| Task | Programme Name | seq/par |
|---|---|---|
| Meshing | **VGrid** | seq |
| | ↓ | |
| Re-partitioning | **DRAMA** | seq or par |
| | ↓ | |
| Simulator (forward/inverse) | **NeuroFEM** (+PEBBLES/PILUTS/ST41) | seq or par |
| | ↓ | |
| Visualisation | **VM** | seq |

*Figure 5.3: A linear software chain that generates a forward solution.*

## *5.4 How to access, install and use NeuroFEM*

The actual NeuroFEM tool is integrated in the ST4.1 software release. Additionally, a NeuroFEM version is released as a standalone-application with file-coupling to the inverse toolbox via lead field matrix computation.

To provide access to NeuroFEM tools, three interface levels are defined in the release notes of ST4.1. The homogeneous interface between the functions of the inverse toolbox and NeuroFEM allows to use NeuroFEM as stand-alone application to compute the lead-field matrix and as integrated FEM-simulation tool in the inverse source reconstruction. One level of the interfaces enables the calling of inverse methods on a command line level. This level is integrated in a complete SimBio application environment.

The software is tested and optimised on Linux and SGI-Irix using the Gnu C++- compiler. The ST4.1 and the WP3 releases contain libraries and a binary command line tool for Linux and SGI-Irix.

# 6. PAM-SAFE

## *6.1 Introduction*

Since 90‘s years, the FEM code, explicit non-linear solver developed by ESI has been completed with airbag, seatbelt and dummy models to make PAM-SAFE™ a comprehensive occupant safety package. From this safety solver, the first studies in biomechanical field have been performed ([pam95], [bea95]). That answered to the safety user request to model the behaviour of the human and not only the one of the dummy ([pam98], [pam99]). The development of the biomechanical models for the health began with the European project, KNEES-UP ([knee]), where the first 3D FE knee model has been developed for, in particular, the study of the menisci to improve menisci prosthesis. This one year project, in collaboration with Sheffield University, permitted us to bring to light the specific problems met with this kind of biomechanical models (very confine environment and very small components,…) and in this specific simulations in the health with a large simulation time like a gait cycle simulation (in the order of 1 second).

## *6.2 Final Implementation*

Thus, these kinds of biomechanical models and of simulations need some improvements of PAM-SAFE™ package, the solver and its pre-processor, GENERIS™, to facilitate the modelling of the mesh, to improve the modelling of the biological materials, in particular the menisci and to decrease the CPU time of the simulation.

### 6.2.1 The Generis™ Pre-processor

6.2.1.1 Visualisation of the fibre direction in elements, in particular solids.
This option enables one to visualise global or local direction of a solid element. The first two directions (r; s) are displayed. The third direction is obtained as the cross-product of the two first ones.

6.2.1.2 Automatic definition of the 3D-curve defining the meniscus centre line.
By defining a group (keyword GROUP) containing the nodes and null-element shells shown on the picture, the 3D-center line of the structure is automatically computed.

6.2.1.3 Reading and writing of the tetrahedron element definition.
Generis™ now reads and writes 4-nodes and 10-nodes tetrahedral elements.

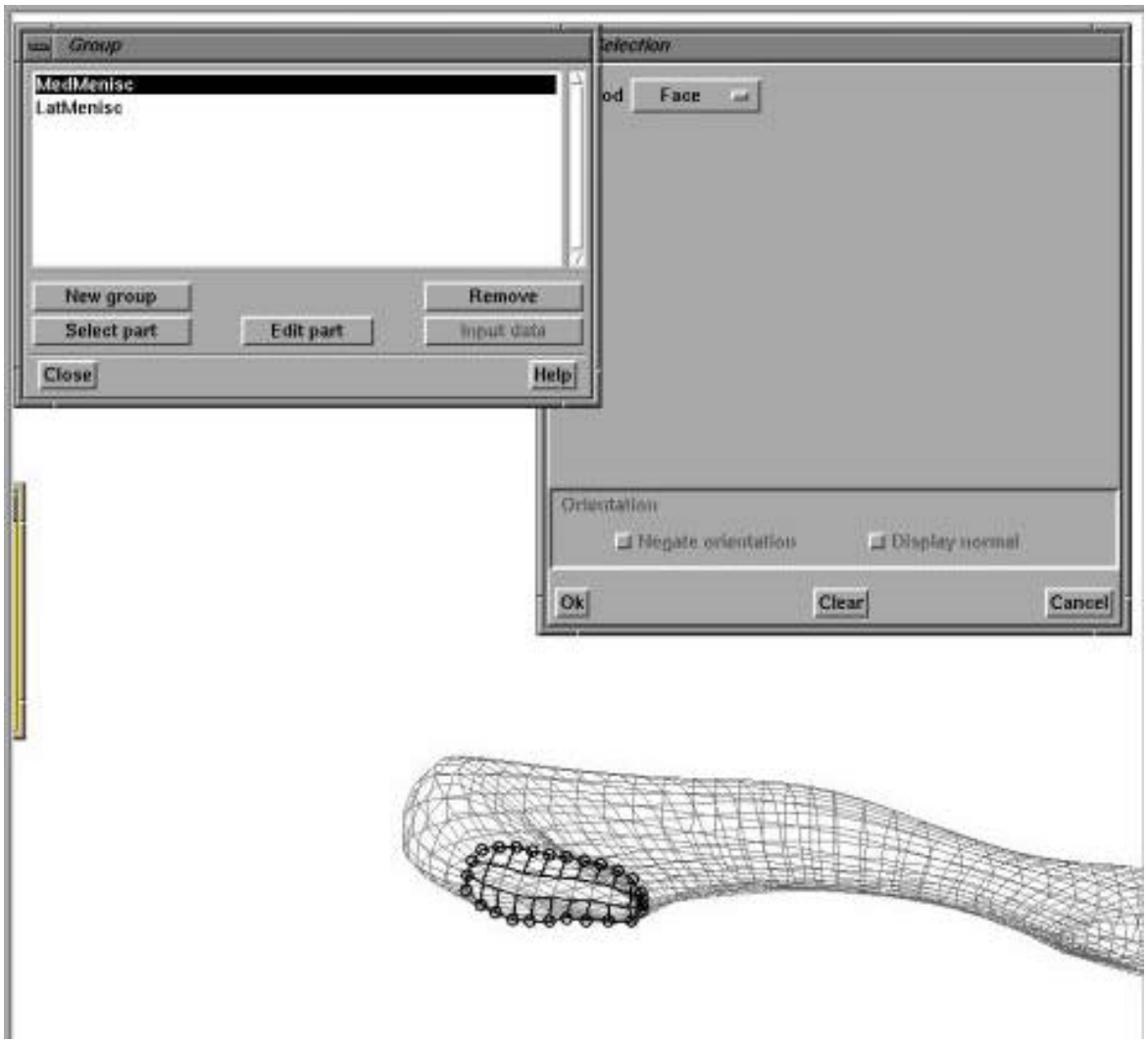



**Figure 6.1: Neutral fibre definition.**

## 6.2.2 PAM-SAFE™ Solver

6.2.2.1 Evaluation of current tetrahedral formulation and its improvement.
Until now, the PAM-SAFE™ solver needed to work with 8-nodes solid elements to obtain an accurate response of the mesh. This was a very restraining condition. In fact, some software of automatic mesh exist, but the most efficient software creates 4-nodes elements, (i.e. tetrahedra). In particular, the human geometry, which is especially complex, is difficultly meshed with hexahedra.

Moreover, the WP1 of SimBio develops an automatic mesh generation from MRI to obtain tetrahedra especially. The tetrahedral formulation must be improved in PAM-SAFE™ solver.

The recent development of the tetrahedral element formulation enables to use the most advanced mesh generation technique software. Two types of terahedra elements were developed: 4-nodes and 10-nodes tetrahedra. While 4-nodes tetrahedra have a node per top, 10-nodes tetrahedra have an additional node on each edge. A simple traction test was performed on a meniscus sample with a linear elastic material law. Results are compared for four types of element:

- Hexahedra
- Degenerated hexahedra, which are tetrahedra with an hexahedral formulation of the structure mechanics equations.

- 4-nodes tetrahedra
- 10-nodes tetrahedra.

| Hexahedra | Degenerated hexahedra | Tetra - 4 | Tetra - 10 | Type of element |
|---|---|---|---|---|
| 13.67 | 14.34 | 16.63 | 13.38 | Modulus of elasticity (Mpa) |
| 2.873 | 0.5914 | 0.8449 | 0.62 | Time step (µs) |
| 698 | 8695 | 2341 | 19440 | Simulation time (s) |

**Table 6.1: Performance results**

Compared to the hexahedral reference, 10-nodes tetrahedral elements give the best response. Nevertheless, their calculation time is very long, partly due to the fact that one single hexahedron element was cut into 4 tetrahedral elements. 4-nodes tetrahedral have a 10% over-evaluation of the result, but at a much quicker pace.

6.2.2.2 Improvement of some materials in the case of specific parameters of menisci.
Due to the complexity of the biological material and due to the difficulty to perform experimental tests for their characterisation, it is important to take into account the microstructure of the biological material, in particular for the knee model, the microstructure of the menisci and of the ligaments. Thus, the fibre direction is the main characteristics of the microstructure of these biological components. Currently, the user defines the fibre direction in the global frame or in the local frame of each element constituting the material. The first solution cannot be used for a too complex fibre direction as in the menisci. The second solution is very heavy, because no automatic tool exists.

The two former options when it comes to defining meniscus mechanical properties are the following.

- Use a linear elastic material representing the meniscus matrix and superpose stiff bars on the same mesh for representing the collagen fibres.
- Use a ply material where the user defines both matrix and fibres properties in the same material. The fibre orientation in that material is restricted to global coordinates system or local element coordinates. So to control accurately this fibre direction the mesh must follow this direction: this is extremely restraining for the mesh generation.

These two options are mesh-dependent and this is not suitable for automatic generation, in particular when working with tetrahedral elements whose local element direction is dependent on their shape. Therefore, an orientation definition is developed that projects the 3D-medial line of the structure on each element.

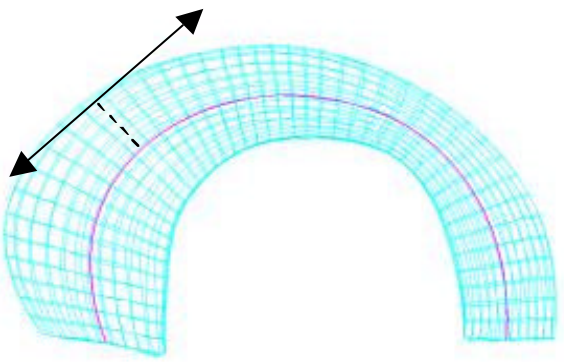


Figure 6.2: Meniscus with medial line.

The 3D-medial line is defined as three parameterised functions of a meaningless parameter s: x(s), y(s), z(s). This option enables the definition of mesh-free fibre reinforced structures such as the meniscus. Its input is the 3D curve defined by the pre-processor.

6.2.2.3 Performance evaluation of the current parallel version.

6.2.2.4 Settings of new (or future) developments necessary for the knee behaviour study in the parallel version.

6.2.2.5 Setting new features in the splitter tool.

6.2.2.6 Work on the simulation time reduction for this kind of simulation.

The simulation time for this kind of simulation is due to two main factors:
-A long physical time for explicit finite element analysis
-A low time step.

The first option for raising the time step is to get rid of all stiff structures. In the SimBio project, both the trabecular and the cortical bones are modelled as rigid. The time-step is thus imposed by the soft structures. This has little influence on the soft structures response as verified by the following study: A condyle of the femoral and tibial bones and cartilage are modelled by a ball and socket as seen in Figure 6.3. Only one quarter of a sphere is kept and symmetry conditions are imposed.

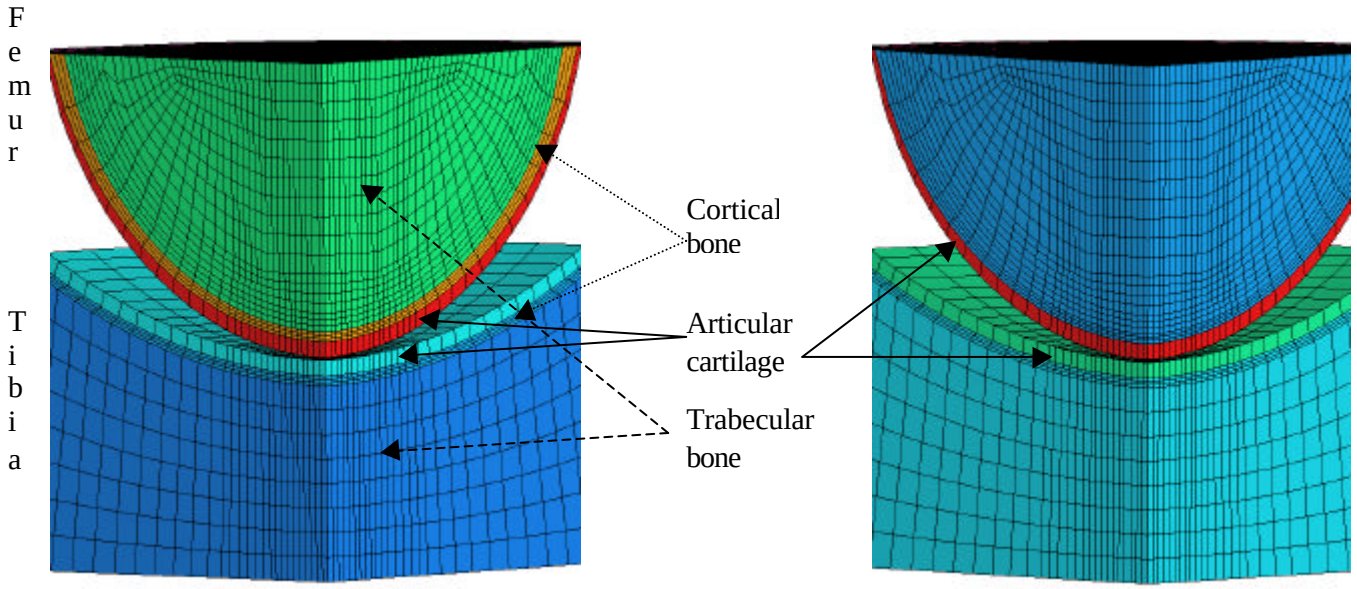



**Figure 6.3: Basic knee model**

In Figure 6.3, case 1, all structures are considered as linear elastic deformable materials. Femoral and tibial articular cartilages are represented by only one layer of elements each. In case 2, only the layers of articular cartilages are kept as linear elastic deformable materials with the same input. The rest of the structure is set as rigid, with the same inertia as the deformable part. The base of the tibial part is then fixed for both models and a load representative of standing is applied on the upper face of the femoral parts. Nodal pressure of the femoral articular cartilage are then compared in Figure 6.4.

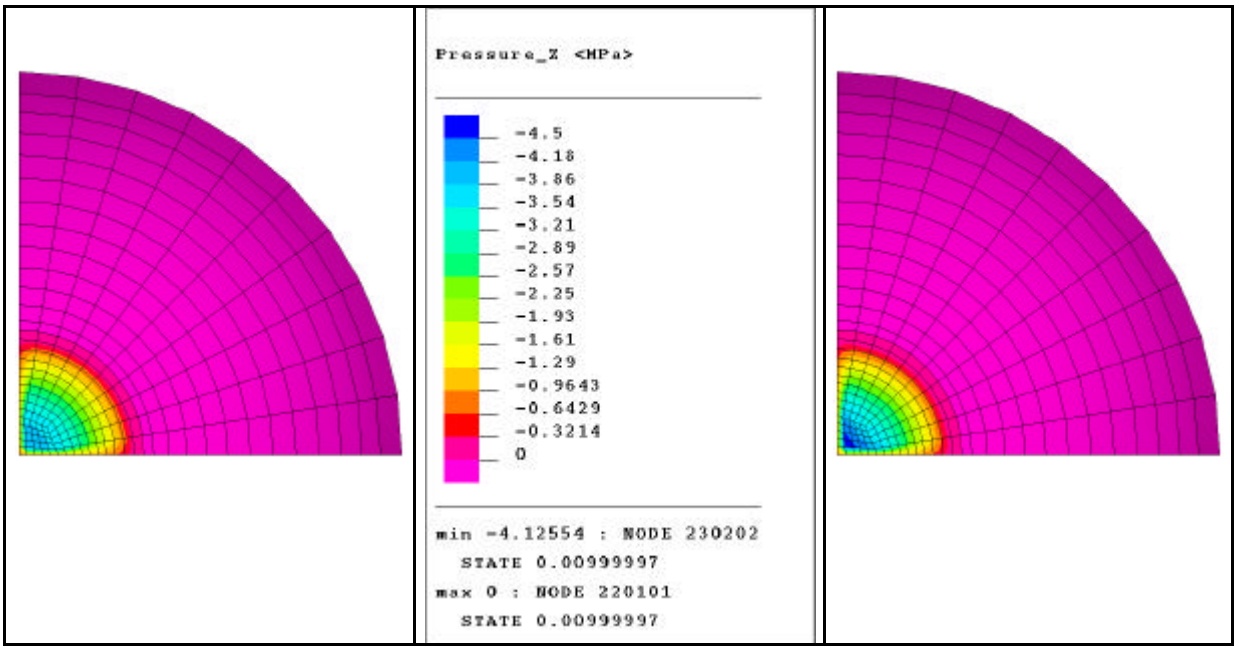



**Figure 6.4: Results**

The results are similar in shape and range. The maximum obtained for case 2 being slightly higher than for case 1 at –4.80 MPa compared to –4.12 MPa for case 1. Table 6.2 summarizes the results and calculation characteristics.

| | Case 1 | Case 2 |
|---|---|---|
| Maximal nodal pressure (MPa) | -4.12 | -4.80 |
| Time step (µs) | 0.26 | 1.02 |
| Calculation time (s) | 6992 | 265 |

**Table 6.2: Results**

The calculation time gain is more than 25 for a 16.5% error.

The next step is to use mass scaling with care to raise the time step without raising the mass too much and in known places. Concerning the high physical time of the studied phenomenon, one can still try and reduce the gait cycle inputs from two seconds to one second for example, without it having too much consequence on the structural level. It is still a gait cycle. The next step is of course extensive parallelisation of the calculation.

### 6.2.3 Converters

Moreover to permit the interaction with the SimBio environment, translators for SimBio data have been developed:

1. Mesh Vista to Pam-SAFE™ format
2. PAM-SAFE™ output DSY, THP to Vista format
3. Integration of the tool in the SimBio environment

### 6.2.4 Interaction with other SimBio components

Figure 6.5 shows the interaction between the PAM-SAFE™ package and other SimBio components. Obviously the main interaction is with the mesh generation tool.

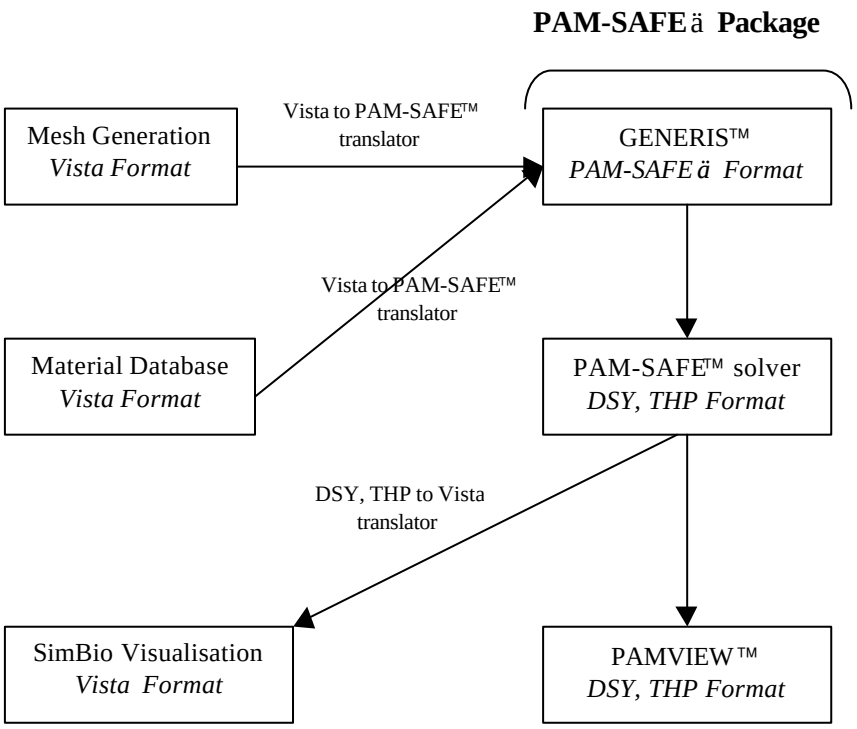



Figure 6.5: Component Interaction.

In the aim to define the best integration of the ESI tool (GENERIS™, PAM-SAFE™, PAMVIEW™) in the SimBio environment, some scenarios were written and given to the WP6.

### *6.3 Summary – PAM-SAFE ä*

A version of PAM SAFE version V2002-beta was delivered for the NEC CLUSTER. To achieve that, the following was performed:

- A new version of the software based on FORTRAN 90 (the late version was F77) was created. This needed code changes especially with some features in order to have a completely portable version.
- This software version included a complete porting on the NEC LINUX architecture.
- New routines about time counting were implemented.
- The optimisations on NEC by using the INTEL60 compiler were made. Problems with new compiler were found and resolved.
- The parallel platform used was the SCORE which is the NEC tool. It was necessary to understand and run the SCORE parallel platform.
- The licence FLEXLM system for LINUX was incorporated (new feature).
- The METIS decomposition method was also compiled and incorporated into the version DMP.
- The routines related to the "Fibres Option 4" that were created for the arbitrary definition of fibres for the meniscus were incorporated.
- Some tests were run to validate the executable. These test cases included full knee simulations under different number of processors to access the scalability of the software.

# 7. HEAD-FEM

## *7.1 Introduction*

HeadFEM is a fully parallel Finite Element (FE) code for applications specialized on head-mechanics. The code allows static/dynamic linear elastic and non-linear hyper-elastic FE analysis suitable for biomechanical head modelling. HeadFEM exploits the PILUTS solver library for the parallel solution of the linear system of equations. The typical calculation time for a static linear / non-linear FE analysis is less than 5 / 30 minutes for a problem with 60.000 elements corresponding to a spatial resolution of 4 mm on a single CPU.

## *7.2 Final Implementation*

HeadFEM was designed as an object oriented modular code using standard interfaces (Finite Element Interface, FEI, defined by Sandia National Labs [fei]) to allow a clean integration of solver libraries and to facilitate further enhancements of the software. The FEI matrix interface is used to couple HeadFEM in memory to the linear solver library PILUTS.

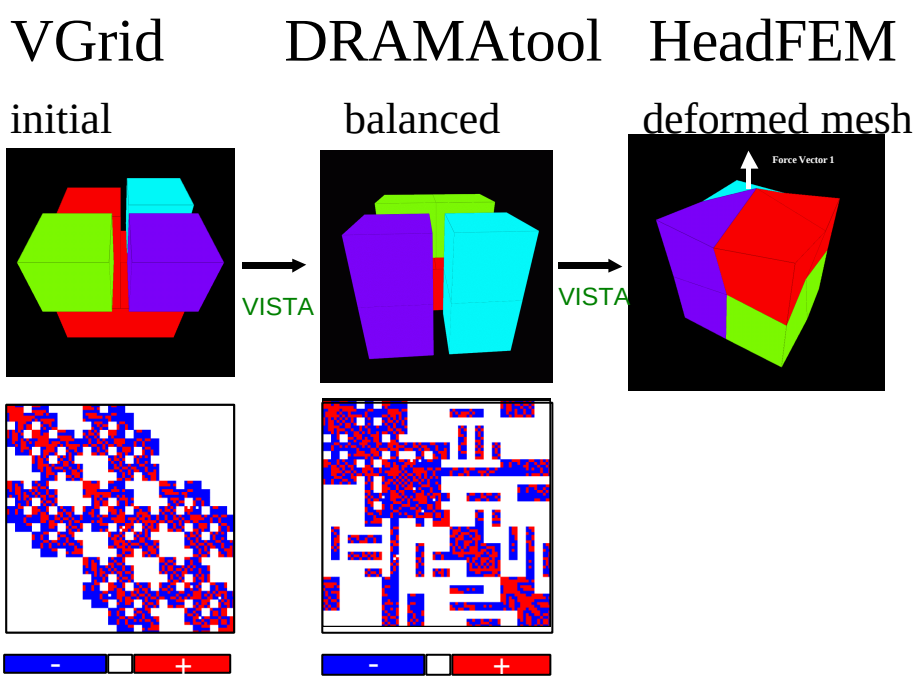



**Figure 7.1: Component interaction example. Matrix visualization with MatView from Oak Ridge National Lab [matview], (blue =negative values, red=positive entry).**

| Task | Programme Name | sequential/parallel |
|---|---|---|
| Meshing | **VGrid** | seq |
| | ↓ | |
| Re-partitioning | **DRAMA tool** | seq or par |
| | ↓ | |
| Simulator | **HeadFEM** | seq or par |
| | ↓ | |
| Visualisation | **VM** | seq |

**Figure 7.2: SimBio component interaction for HeadFEM.**

HeadFEM is running in parallel with an input mesh generated by VGrid and partitioned by the DRAMA tool. The partitioning information is stored as two additional vectors for elements and nodes separately in the VISTA-file. The results of a linear or non-linear static FE-analysis with HeadFEM are vectors of nodal displacements and forces and element based stresses and strains. The output mesh is in VISTA format that is accepted by the SimBio visualisation module VM.

The HeadFEM code supports linear tetrahedral and hexahedral elements. Non-linear analysis is implemented by a full Newton-Raphson method based on a calculation of the tangential stiffness matrix. The available material models include isotropic linear elasticity and hyper-elasticity. A mixed displacement / pressure formulation implemented by the mean dilatation method allows one to model compressible and nearly incompressible materials like soft tissues. In addition, suitable non-linear stress/strain measures have been implemented in the code.

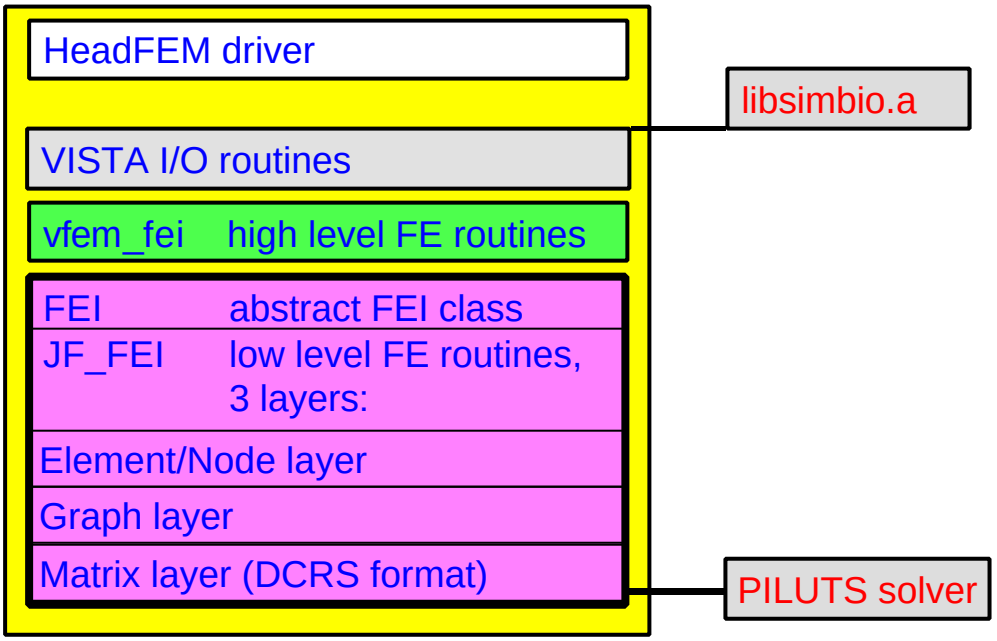



**Figure 7.3: Structure of the HeadFEM implementation.**

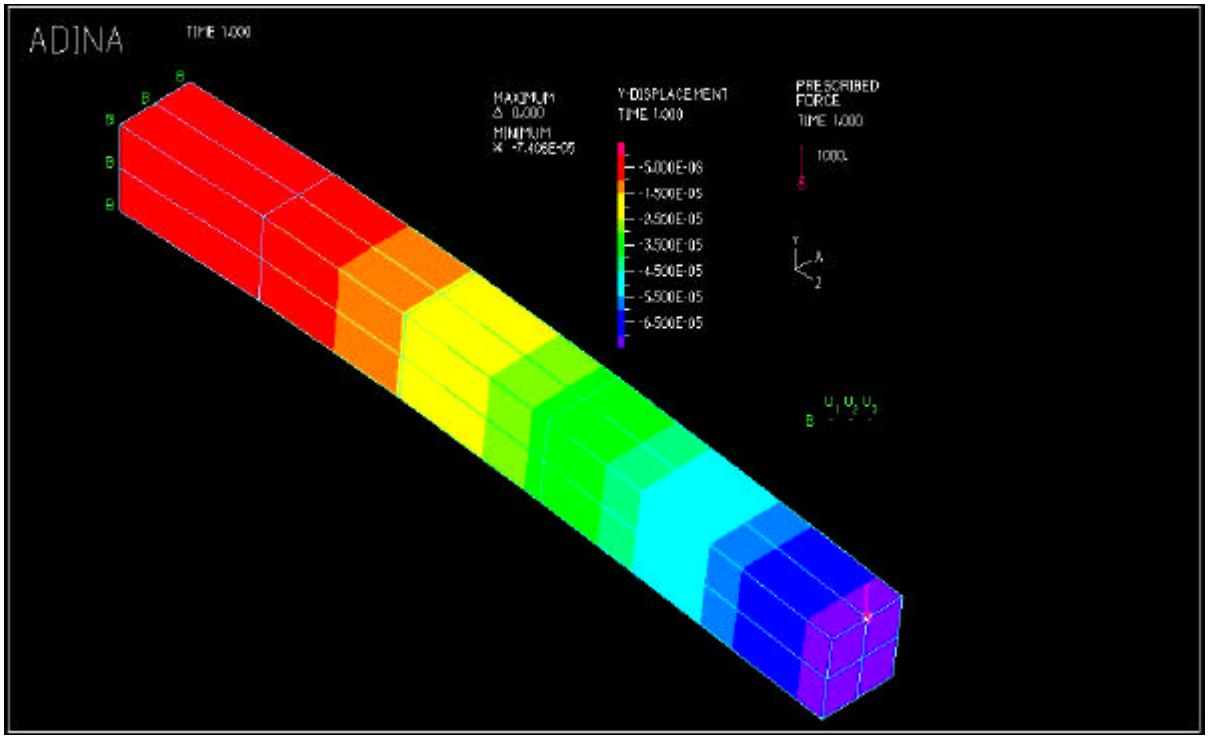
The implementation has been verified against the commercial code ADINA™ using three CAD models which were directly accessible to both codes:

Example 1 : Linear analysis of a tip loaded beam with:
- 20 hexahedral elements,
- 54 tetrahedral elements.

Example 2: Linear analysis of a cylinder with internal pressure load (analytic solution) with:
- 400 hexahedral elements,
- 2400 tetrahedral elements.

Example 3: Non-linear analysis of a tip loaded beam with 20 to 40000 elements.

Example1: Y-displacements of a tip loaded beam with ADINA™ and HeadFEM.

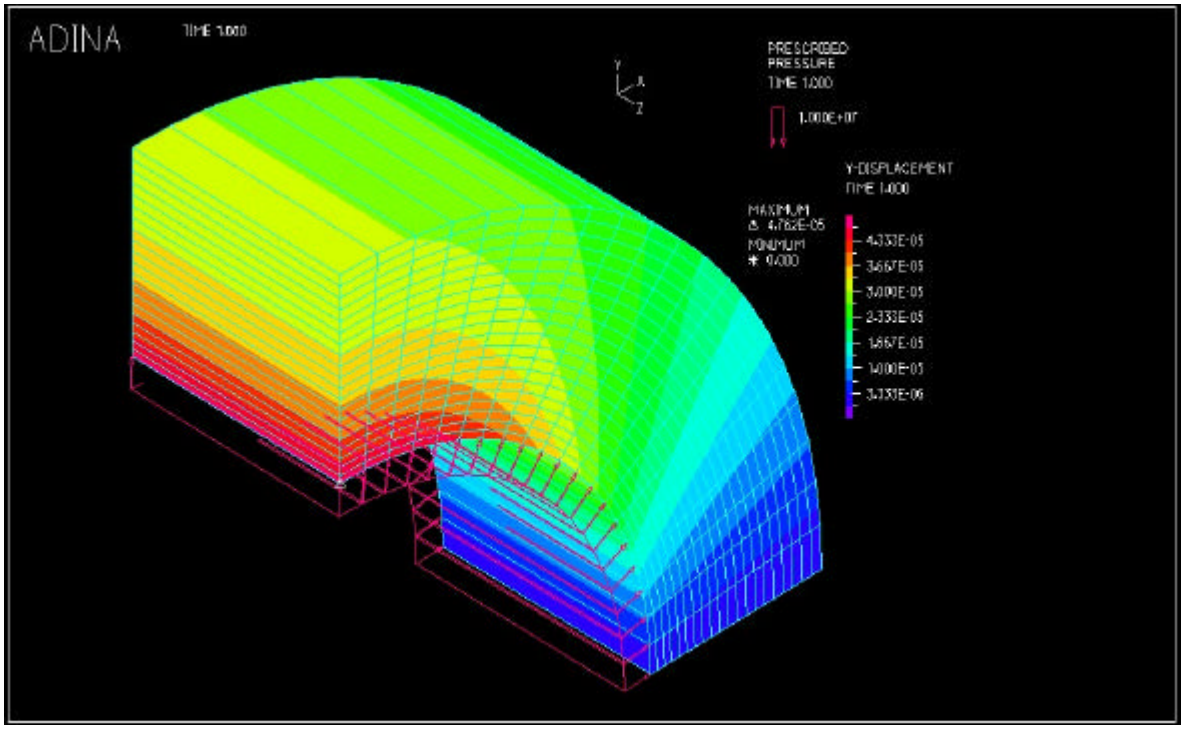



| Displacements (at node ID=52) | | | | | | |
|---|---|---|---|---|---|---|
| | 20 hexahedral elements | | | 54 tetrahedral elements | | |
| | x | y | z | X | y | z |
| 1PE ADINA™ | 3.95212e-17 | -7.40615e-05 | 5.56015e-06 | 9.28873e-06 | -3.62608e-05 | 2.74549e-06 |
| 1PE HeadFEM | -9.02289e-18 | -7.40628e-05 | 5.55377e-06 | 9.28876e-06 | -3.62611e-05 | 2.74389e-06 |
| 4PE HeadFEM | -4.35608e-18 | -7.40628e-05 | 5.55377e-06 | 9.28876e-06 | -3.62611e-05 | 2.74389e-06 |

| Global stiffness matrix | | | | | | | |
|---|---|---|---|---|---|---|---|
| 20 hexahedral elements | | | 54 tetrahedral elements | | | Matrix indices | |
| ADINA™ | HeadFEM | | ADINA™ | HeadFEM | | | |
| 1 PE | 1 PE | 4 PEs | 1 PE | 1 PE | 4 PEs | row | col |
| 1.58449e+10 | 1.58496e+10 | 1.58496e+10 | 3.29765e+10 | 3.29864e+10 | 3.29864e+10 | 158 | 158 |
| 0.00000e+00 | 2.38419e-06 | 2.38419e-06 | -6.36766e+09 | -6.36957e+09 | -6.36957e+09 | 157 | 158 |
| 3.97979e+08 | 3.98098e+08 | 3.98098e+08 | 0.00000e+00 | 0.00000e+00 | 0.00000e+00 | 156 | 158 |
| 1.28680e+09 | 1.28718e+09 | 1.28718e+09 | -2.67442e+09 | -2.67522e+09 | -2.67522e+09 | 155 | 158 |
| 5.09413e+08 | 5.09566e+08 | 5.09566e+08 | 3.69324e+09 | 3.69435e+09 | 3.69435e+09 | 154 | 158 |
| 6.36766e+07 | 6.36957e+07 | 6.36957e+07 | -6.68605e+08 | -6.68805e+08 | -6.68805e+08 | 153 | 158 |
| -3.87764e+09 | -3.87880e+09 | -3.87880e+09 | -9.04208e+09 | -9.04479e+09 | -9.04479e+09 | 152 | 158 |
| -3.18383e+09 | -3.18479e+09 | -3.18479e+09 | -6.36766e+09 | -6.36957e+09 | -6.36957e+09 | 151 | 158 |

The comparison between ADINA™ and HeadFEM results shows excellent agreement of all significant digits both for displacements and global stiffness matrix.

Example 2: Cylinder under internal pressure load.




Input material parameters and pressure load:
E = 2.0e11 N/$m^2$
ν = 0.3
p = 1.0e7 N/ $m^2$

| Radial displacements --- 400 hexahedral elements | | | | | |
|---|---|---|---|---|---|
| | | Theory | ADINA™ | HeadFEM | |
| y [m] | Node ID | | 1 PE | 1 PE | 8 PEs |
| 0.5 | 440 | 4.7667e-05 | 4.76247e-05 | 4.76592e-05 | 4.76592e-05 |
| 1.0 | 420 | 3.0333e-05 | 3.03123e-05 | 3.03343e-05 | 3.03343e-05 |
| Radial displacements --- 2400 tetrahedral elements | | | | | |
| | | Theory | ADINA™ | HeadFEM | |
| y [m] | Node ID | | 1 PE | 1 PE | 8 PEs |
| 0.5 | 440 | 4.7667e-05 | 4.81441e-05 | 4.82780e-05 | 4.82780e-05 |
| 1.0 | 420 | 3.0333e-05 | 3.04874e-05 | 3.05826e-05 | 3.05826e-05 |

Example 3: Tip loaded beam (steel) with non-linear FE analysis (full Newton-Raphson).

| XxYxZ | elements | max. y-displacement | |
|---|---|---|---|
| | | HeadFEM | ADINA™ |
| 2x2x5 | 20 | 7.4061e-5 | 7.4062e-5 |
| 4x4x10 | 160 | 1.3717e-4 | 1.3718e-4 |
| 10x10x40 | 4000 | 1.8910e-4 | 1.8911e-4 |
| 12x12x60 | 8640 | 1.9233e-4 | 1.9233e-4 |
| 16x16x80 | 20480 | 1.9392e-4 | 1.9393e-4 |
| 20x20x100 | 40000 | 1.9495e-4 | 1.9496e-4 |

Again, results show excellent agreement between solutions from ADINA™ and HeadFEM.

## *7.3 How to access, install and use HeadFEM*

The use of the HeadFEM software requires that the SimBio library *libsimbio.a* and the PILUTS solver library *libpiluts.a* have been installed on the system. The following sections explain the installation, verification and use of the HeadFEM package.

### 7.3.1 System hardware and software requirements

The prerequisites to run HeadFEM as part of the SimBio environment are:

1. parallel platform
2. system libraries: libmpi.a
3. libraries: * libsimbio.a [wp1]
   * libheadfem-utilities.a [Utility functions for HeadFEM]
   * libmetis.a [sequential Metis library from University of Minnesota]
   * libpiluts.a [NEC PILUTS solver library]
   * for specialized diff operations: GrAL
   (contact: berti@ccrl-nece.de),
   and the gral-simbio module.
4. tools: vtoa (from wp1/pgms/converters) for verification
5. compilers: mpicc, mpiCC, mpif90 or mpif77 (for PILUTS)

To compile the PILUTS library change to the directory $(SIMBIO)/solver and run make. This produces the solver library libpiluts.a.

To compile the HeadFEM utilities change the directory headfem-utilities and run make. This produces libheadfem-utilities.a. Use “gmake DEBUG=opt” to obtain an optimised version.

### 7.3.2 Compilation of HeadFEM

To compile the HeadFEM code you need to:

1. Edit Makefile and set local options.
2. gmake [DEBUG=opt to obtain an optimised version]
3. copy bicgstab.in.default to bicgstab.in,
   and/or likewise for cg.in, symqmr.in
   (these files control the corresponding PILUTS solver, see corresponding section below).

Now you can optionally verify the HeadFEM installation:

4. run_test_suite (see also section 7.3.5 below).

### 7.3.3 Perform a Simulation with HeadFEM

To run HeadFEM use the following steps:

1. Edit file HeadFEM.options to set:
   olevel -- integer controlling the level of output (0=no additional output)
   epsNR -- stopping criterion for Newton-Raphson (NR): resid < epsNR
   switchNR -- 0: static / 1: modified NR / 2: full NR

2. Copy input VISTA mesh to input.v in the local working directory.

3. Start HeadFEM (executable name is *driver*):
   a) interactive: mpirun -np 8 *driver* <options> (example on 8 processors).
   b) batch: submit -p 8 -t 20 -b *script* (example on SUCCESS)
   Here, *script* is an adequate wrapper script, see e.g. ./script or ./runchecks.pl

### 7.3.4 Command-line options for HeadFEM

The following command line options are available in the final version of HeadFEM:

Usage: ./driver <options>, where <options> includes:

| | |
|---|---|
| -help | Prints usage information. |

Options for I/O:

| | | |
|---|---|---|
| -out <string> | Name of result file. | Default: output.v |
| -in <string> | Name of input file. | Default: input.ascii |
| -log <string> | Name of logging file. | Default: output.<p> where p is the process number. |
| -opt <string> | Name of options file. | Default: HeadFEM.options |
| -material <string> | Name of material properties file. | Default: mat.dat |
| -saveMatrix <string> | If set, the system matrices will be saved to the given file.<p>. | Default: matrix.<p> |
| -olevel <integer> | Verbosity level. | Default: 1 |
| -dumpMatrix <int> | toggle dumping of system matrix to save memory. | Default: 0 |

Options for the linear solver:

| | | |
|---|---|---|
| -solvetype cg \| symqmr \| bicgstab | Solver algorithm. Depending on solvetype, the solver reads in a file, cg.in \| symqmr.in \| bicgstab.in | Default: bicgstab. |
| -choicerhs user \| rowsum | Choose right-hand side, (rowsum is useful for debugging only, because then the solution is known: (1,..,1)). | Default: user |
| -choicestart zero \| scaledrhs \| user \| ilu | Choose starting vector. | Default: zero |

### 7.3.5 Verification of the Installation

A test-suite is provided for verification purposes. The tests are base on a comparison of reference output with output from a current run. There are five options:

1. Compile the specialised tools for file comparison in GridDiff/ and MatrixDiff/, which use the GrAL library. Note that the GrAL library has to be installed first (edit the corresponding Makefiles, variable $(GRAL) ),

2. Use the usual Unix *diff* command: set the Makefile-variables GRIDDIFF and MATRIXDIFF to *diff*. This has the drawback that there is no way to ignore insignificant deviations in floating point values,

3. ./run_test_suite to run all 24 tests (this option will take some time!),

4. ./run_test_suite_mini (1 test),

5. ./run_test_suite_small (9 tests).

## *7.4 Example: Skull Mechanics Simulation with HeadFEM*

The following pictures show an example of the model preparation (interactive halo device positioning) for a skull mechanics simulation with HeadFEM. The finite element model created by the VGrid mesher from voxel data (MRI scan) shown in Figure 7.4 (a) together with the medical halo device consisted of 60.000 hexahedral elements corresponding to a spatial resolution of 4 mm.

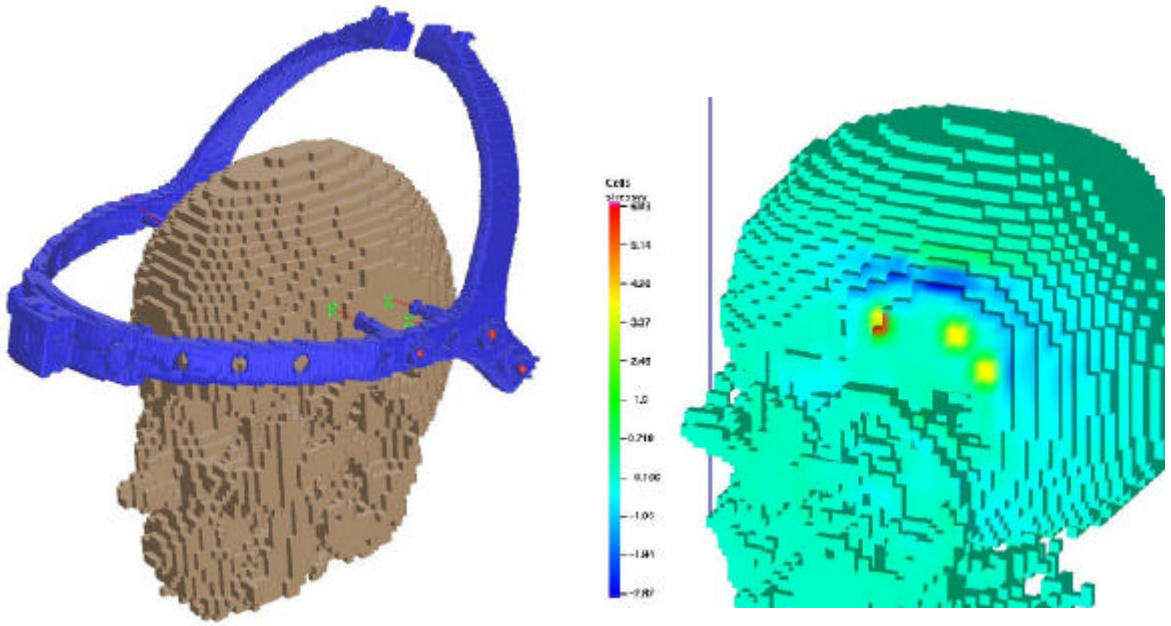


Figure 7.4 (a) (b)

Figure 7.4 (b) shows, as expected, that the distribution of stresses is strongly localised around the position of the initial forces.

## References


[bas00] Basermann, A., “Parallel Block ILUT/ILDLT Preconditioning for Sparse Eigenproblems and Sparse Linear Systems”, to appear in Numerical Linear Algebra with Applications, special issue of the Sparse99 meeting, Minneapolis, 1999.

[bas96] Basermann, A., "QMR and TFQMR methods for sparse non-symmetric problems on massively parallel systems", The Mathematics of Numerical Analysis, series: Lectures in Applied Mathematics, vol. 32, pp. 59-76, 1996.

[bas99] Basermann, A., "Parallel Preconditioned Solvers for Large Sparse Hermitian Eigenvalue Problems", J. Dongarra and V. Hernandez, eds: VECPAR’98 – Third international conference for vector and parallel processing, Lecture Notes in Computer Science, Springer, Berlin, vol. 1573, pp. 72-85, 1999.

[bea95] Beaugonin, M.; Haug, E., Munck, G., Cesari, D., *"A Preliminary Numerical Model of the Human Ankle under Impact Loading"* ; PLEI Conference,Washington, D.C., USA, 1995.

[buc97] Buchner, H., Knoll, G., Fuchs, M., Rienacker, A., Beckmann, R., Wagner, M., Silny, J., and Pesch, J. (1997) Inverse localization of electric dipole current sources in finite element models of the human head. *Electroencephalography and Clinical Neurophysiology*, **102**:267-278.

[bur02] Burkhardt, S., Wolters, C.H. and Saupe, D. (2002) Segmentation of the Human Skull Surfaces from Dual-Echo MRI volumes. *Med.Imag.Anal.*, submitted for publication.

[chan97] Chan, T.F. and Wan, W.L. (1997) Analysis of Projection Methods for solving linear systems with multiple right hand sides. *SIAM J.Sci.Comput.*, **18(6)**:1698-1721.

[fei] The FEI interface definition from Sandia: http://z.ca.sandia.gov/fei/.

[haa99] Haase, G. (1999) Parallelisierung numerischer Algorithmen fuer partielle Differentialgleichungen, B.G.Teubner Stuttgart Leipzig.

[haa00] Haase, G., Kuhn, M. and Reitzinger, S. (2000) Parallel AMG on distributed memory computers, Preprint submitted to Elsevier Preprint.

[hau02] Haueisen, J., Tuch, D.S., Ramon, C., Schimpf, P.H., Wedeen, V.J., George, J.S., Belliveau, J.W. (2002) The Influence of Brain Tissue Anisotropy on Human EEG and MEG. *NeuroImage*, **15**, 159-166.

[knee] Knees-UP Project, ESPRIT HPCN TTN Action, Realistic 3D Model of the Human Knee. World-Wide Web document: http://kiefer.gmd.de:8002/popcorn/projects/KNEESUP/www-kneesup.htm

[mar98] Marin, G., Guerin, C., Baillet, S., Garnero, L. and Meunier, G. (1998) Influence of skull anisotropy for the forward and inverse problem in EEG: Simulation studies using FEM on realistic head models, *Human Brain Mapping*, **6**, 250-269.

[matview] Scalable Matrix Viewer: http://www.epm.ornl.gov/~kohl/MatView/.

[mun88] de Munck, J.C. (1988) The potential distribution in a layered anisotropic spheroidal volume conductor, *J.Appl.Phys.*, **64** (2), 464-470.

[mun93] de Munck, J.C. and Peters, M. (1993), A fast method to compute the potential in the multi sphere model, *IEEE Trans. Biomed. Eng.*, **40**(11), 1166-1174.

[pam95] Haug, E., *"Biomechanical Models in Vehicle Accident Simulation"*, PAM-User's Conference in Asia; PUCA'95, pp. 233-256, Shinyokohama, Japan, November 16-17, 1995.

[pam98] Haug, E., Beaugonin, M. and Tramecon, A. (1998) Current Status of Articulated and Deformable Human Models for Impact and Occupant Safety Simulation at ESI Group, *PAM'98 User's Conference.*

[pam99] Haug, E., Beaugonin, M., Godest, A. C., Tramecon, A., Allain, J. C., Hyncik, L., Choi, L. H., Lee, I. H. (1999) Recent Trends in Biomechanical Simulation and Applications at ESI Group, *PAM'99 User's Conference.*

[pmvis] Partitioned Matrix Visualiser: http://www-users.cs.umn.edu/~oztekin/pmvis/pmvis.html.

[rango96] von Rango, J., Schlitt, H.A., Halling, H. and Mueller-Gaertner, H.-W. (1996) Finite Integration Techniques for the MEG Forward Problem. , *In: Witte, H., Zwiener, U., Schack, B. and Doering, A. (eds.), Quantitative and Topological EEG and MEG Analysis, Druckhaus Mayer Verlag GmbH Jena-Erlangen,1997, pp.*336-338.

[rei00] Reitzinger, S. (2000) PEBBLES: Parallel and element based gray box linear equation solver http://www.sfb013.uni-linz.ac.at/~reitz/pebbles.html.

[rien97] Rienaecker, A., Buchner, H. and Knoll, G., Comparison of Regularized Inverse EEG Analyses in Finite Element Models of the Individual Anatomy, *In: Witte, H., Zwiener, U., Schack, B. and Doering, A. (eds.), Quantitative and Topological EEG and MEG Analysis, Druckhaus Mayer Verlag GmbH Jena-Erlangen, 1997, pp. 459—463.*

[saad96] Saad, Y., "Iterative methods for sparse linear systems", PWS Publishing Company, Boston, 1996.

[saad99] Saad, Y. and Sosonkina, M., *Distributed Schur Complement Techniques for General Sparse Linear Systems*, SIAM J. Sci. Comput., 21(4):1337-1356, 1999.

[wol00] Wolters, C., Reitzinger, S., Basermann, A., Burkhardt, S., Hartmann, U., Kruggel, F. and Anwander, A. (2000), Improved tissue modelling and fast solver methods for high resolution FE-modelling in EEG/MEG-source localisation, *Proceedings of the 12th International Conference on Biomagnetism, Helsinki, Finland, August 12-17,* http://biomag2000.hut.fi/papers_forward.html

[wol02a] Wolters, C.H., Kuhn, M. Anwander, A. and Reitzinger, S. (2002), A parallel algebraic multigrid solver for finite element method based source localization in the human brain, *Comp.Vis. Sci., in press.*

[wol02b] Wolters, C.H., Anwander, A., Koch, M., Reitzinger, S., Kuhn, M. and Svensen, M. (2002), Influence of Human Head Tissue Conductivity

Anisotropy on the EEG using Fast High Resolution Finite Element Modeling, based on a Parallel Algebraic Multigrid Solver, *accepted as a bookchapter-contribution to "Forschung und wissenschaftliches Rechnen", http://www.cns.mpg.de/billing/billing.html, 2001*.