Debian for Simulation and Numerical Modeling

Applications to High Magnetic Field Magnets Design

C. Daversin[1], C. Prudhomme[2], C. Trophime[1] and S. Veys[2]

[1] Laboratoire National des Champs Magnétiques Intenses, Grenoble
[2] Laboratoire Jean Kuntzmann, Grenoble

United States

- Thallahassee (FL)
- Los Alamos (NM)
- Gainesville (FL)

Europe

- Grenoble / Toulouse
- Nijmegen (Netherlands)
- Dresden (Germany)

Asia

- Tsukuba (Japan)
- Sendai (Japan)
- Hefei (China)
- Wuhan (China)

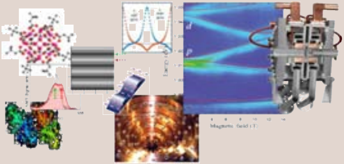
LNCMI a User Facility run by the CNRS

Pulsed field installation TOULOUSE : 14 MJ, 24 kV, 1 GW, 80 Tesla

Continuous field installation GRENOBLE : 24 MW, 35 Tesla

Research

- Condensed matter
- Chemistry and Biochemistry
- Applied Superconductivity
- MagnetoSciences
- Magnet development
- Instrumentation under B



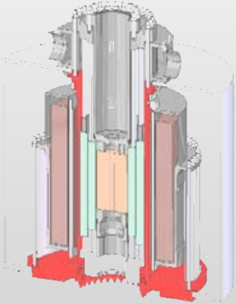
Facilities (10000 hours / year)

- High Pressure
- EPR,NMR
- Lasers
- Low Temp. : down to 20 mK
- High Temp : up to 1600Â°C

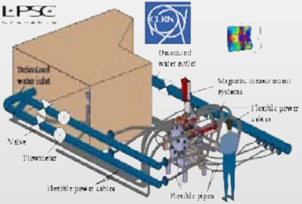
Access

- Call for Magnet Time / 2 x year
- 140 projects / year

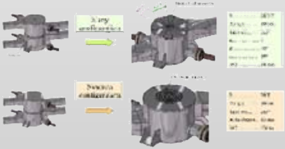
Projets



Hybrid Magnet : 43 T

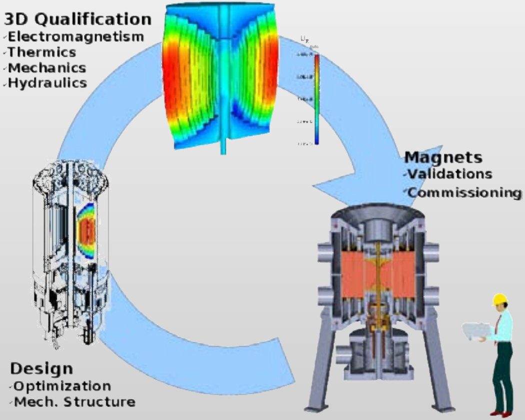


ECR Ion source (Euro ν)



Scattering under Magnetic Field

From Design to Commissioning



High Field Magnet Design

Challenges

- Multiphysics Modeling,
- Non-Linearities and Coupling,
- Complex geometries,
- Optimization

Needs

- 3D Numerical Modeling,
- Fast and reliable methods,
- Control Quantity of Interest (B, $< T >$, stress, ...),
- Uncertainties quantifications

Our choice

- Use open source software ("state-of-the-art"),
- Use Linux as a platform for development and computation,
- Need for HPC (from meso centers to national centers).

First attempts

- Use of RedHat/Fedora,
- Few Librairies/Software for numerical modeling,
- (Re)build packages for used/tested software,
- Difficult to get new packages into distribution,
- Difficult System Upgrade.

Why Debian ?

Debian

- Large choice of software/librairies for numerical modeling in Debian,
- Easier to get new packages into distribution,
- Easier system upgrade.
- Bring full programming and runtime environments for science in minutes

Debian for Numerical Modeling and Simulation

- Debian Scientific Computing Project (scicomp) (C. Prudhomme et al.),
- Debian Science (S. Ledru et al.),
- Most scicomp packages have now been merged into Debian science

Debian - A Large offer

General Finite Element Analysis (FEA)

- General Finite Element Analysis (FEA),
- Numerical libraries,
- Pre- and post-processing frameworks and tools

See

- http://pkg-scicomp.alioth.debian.org/
- http://wiki.debian.org/DebianScience

Debian - HPC ressources

CIMENT meso center (Grenoble)

- 3000 cores (> 5000 in 2013),
- 40 % using Debian,
- Storage grid running Debian.

Grid5000 (Grenoble)

- Heavy Debian user (services and grid nodes)

National(/European) facilities

- Rely unfortunately on poor programming environments : eventually requires compiling down to the compilers, often requires to recompile the numerical libraries stack when one uses modern software
- Suggest Debian@Genci and Debian@Prace

Debian - HPC Ressources

Active development of HPC SysAdmin in Grenoble

- OAR batch scheduler administration tools package
- Taktuk efficient, large scale, parallel remote execution of commands,
- CiGri grid manager tool (to be packaged)
- Xkaapi Lib for parallel programming targeting Hybride archs (CPUs/GPUs)

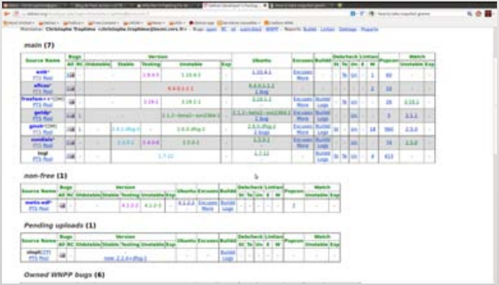
Contributors (imag.fr)

- CIMENT / CiGri : Bruno.Bzeznik
- Grid5000 : Pierre.Neyron
- OAR : Olivier.Richard
- Taktuk : Guillaume.Huard
- Parrain debian : Vincent.Danjean
- DD Philippe.Le-Brouster
- Xkaapi : Thierry.Gautier

Debian - How to contribute ?

How and why I became a DM ?

- Use/test some new libraries / software
- (Re)build / Update the packages (eg : gmsh, mumps, petsc, ...)
- Capitalize my efforts by submitting the package (eg : getdp)



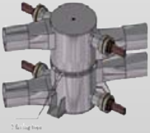
Debian - How to contribute ?

My experiences / advices

- Try to package libraries / software,
- Upload your work into Debian Science svn/git,
- Fill Bug reports, Provide patches,
- Share your work,
- Simple, Save time, May help other,
- Benefits

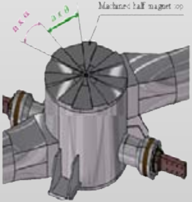
Magnet Development for ESRF

Started with « ESRF Up » FP7



X-ray configuration

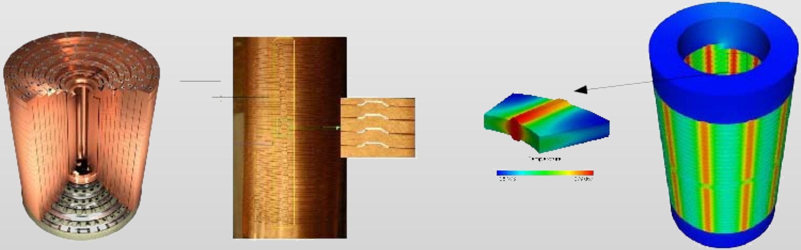
A light for science ESRF



B	28.8 T
Air gap	10 mm
Take off	2x3°
n	7
∂	36°
α	15°
WB	34 mm

High Field Split Magnet Design

- Radially cooled helices insert,
- Temperature within each helices,
- Huge attraction forces.



Thermo-Electrical model and Field calculation with gmsh/getdp

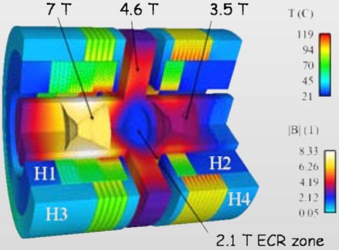


Figure: Initial mesh

Anisotropic mesh Adaption with Feel++ and Gmsh

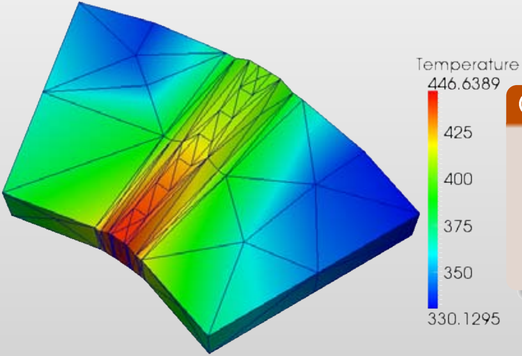


Figure: Initial mesh

Control Interpolation error

- Estimate of Hessian (based on grad recovery tech.)
- Hessian-based metric adapt. Control length of mesh edges

Anisotropic mesh Adaption with Feel++ and Gmsh

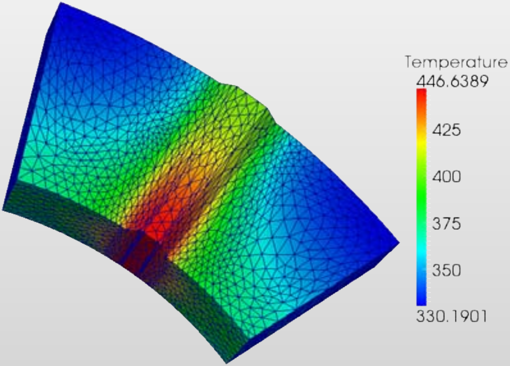
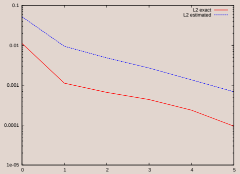


Figure: 3rd iteration

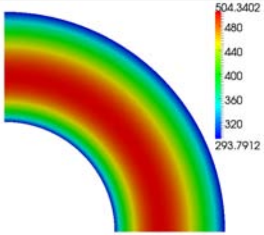
Interpolation error



Model order reduction with Feel++ and OpenTurns

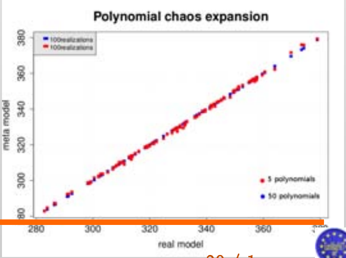
Approximations

- FEM :
  - Dimension of FEM space : 2 312 (P1)
  - NL iterations ≈ 20 (rel. tol 10^{-8})
- RB :
  - Dimension of RB space : 40
  - NL iterations ≈ 20 (rel. tol 10^{-8})
  - EIM terms ≈ 40 (rel. tol 10^{-10})



Meta model

- Y = F(X)
  - X and Y are stochastic variables
  - X follows uniform distribution
  - use polynomial chaos expansion



Sobol indices and quantiles with Openturns

Parameters range

- $\sigma_0 \in [...]$
- $k_0 \in [...]$
- $V_D \in [0.05, 0.2](V)$
- $h \in [6.10^4, 9.10^4](W.m^{-2}.K^{-1})$
- $T_w \in [273, 323](K)$

Mean, Standard deviation

```
-- Mean = [328.473]
-- Standard deviation = 22.0728297966
```

Sobol indices

...
-- Sobol 4 = 0.633300283167 (V_D)
-- Sobol 5 = 1.76545493422e-05
-- Sobol 6 = 0.362812453403 (T_w)

Quantiles

Determine a threshold $q(\gamma)$ such that $P(Y_i < q(\gamma)) > \gamma$

```
99.0 -quantile = [374.123]
80.0 -quantile = [354.55]
```

Conclusions and Perspectives

From Debian view point

- Add specific packages to Debian
- Contribute to packaging efforts (code-aster)
- Enlarge offer for Engineering and Design

From a scientific view point

- Improve our numerical model (error control, uncertainty quantification)
- Apply model reductions to our problem,
- Include more physics
- More complex geometries (from a complete helix to an insert)