Avoimen lähdekoodin ohjelmistot teknisessä laskennassa

- Voiko avoimien lähdekoodin ohjelmistoilla tehdä vakavasti otettavaa tutkimusta tai tuotekehitystä?
- Mikä on avoimen lähdekoodin ohjelmistojen ja yhteisöjen tila?
- Keskeisenä sovellusalueena virtauslaskenta sekä yleisemmin osittaisdifferentiaaliyhtälöiden ratkaisuminen, sekä niiden vaatima esi- ja jälkikäsittely.
- Toiveena on tuoda esille parhaita käytäntöjä, kartoittaa ongelmakohtia, sekä saada ehdotuksia kehityskohteista.
- Osallistujilta toivotaan aktiivista osallistumista keskusteluun.
- Tilaisuus on osa Tekesin osin rahoittamaa LSCFD-projektia

Päivän ohjelma

- 9.30 Ilmoittautuminen & kahvi
- 10.00 Workshopin avaus & tausta
- 10.10 Avoimen lähdekoodin kehitysmallit (Arto Teräs, CSC)
- 10.40 Avoimen lähdekoodin ohjelmistot teknisessä laskennassa - katsaus nykytilanteeseen (Peter Råback, CSC)
- 11.20 Verkongenerointityökalut avoimen lähdekoodin virtauslaskentaan (Juha Kortelainen, VTT)
- 12.00 Lounas (omakustanteinen)
- 13.00 Avoimen lähdekoodin kirjastot ohjelmistokehityksessä - esimerkkinä ElmerGUI (Mikko Lyly, CSC)
- 13.40 Teollisuuden näkökulma avoimen lähdekoodin ohjelmistoihin (Timo Toppila, Fortum)
- 14.20 OpenFOAM - päättymätön ohjemistokehitysprosessi? (Esko Järvinen, CSC)
- 15.00 Kahvitauko
- 15.20 Paneeli aiheesta "Avoimen lähdekoodin mahdollisuudet teknisessä laskennassa"
  (Tommi Karhela pj., Juha Kortelainen, Kati Laakkonen, Peter Råback, Arto Teräs, Timo Toppila)
- n. 16 Tilaisuuden päätös

Tekes-project

LSCFD

Tools for Large Scale Computational Fluid Dynamics

Peter Råback, project manager
CSC, Finnish IT Center for Science

LSCFD: Motivation

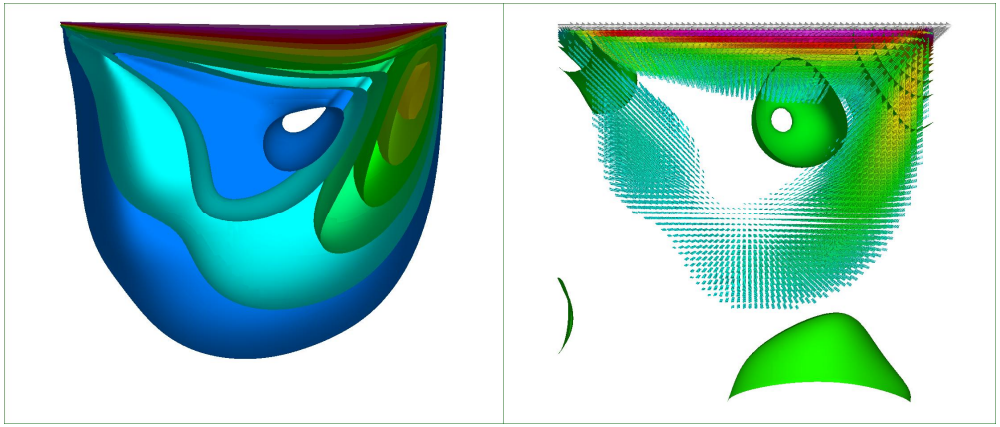
- Supercomputing is underused in the area of CFD (<3%)
  - Internationally CFD is one of the key areas for HPC
  - Modeling of turbulence, particularly of LES remains a grand challenge for many years to come
- Growing interest for Open Source solutions
  - Example: OpenFOAM seminar at CSC with 67 participants
- Some new technologies and methods with high benefits
  - Parallel postprocessing
  - Massively parallel libraries, Hypre
  - Variational multiscale methods for turbulence, VMS
- Information gradients between different communities
  - Acedemia/industry, FV/FE, commercial/open source software

LSCFD: Main goals

- Demonstrate and document the full chain from mesh generation to visualization using best available open source tools
  - Special effort on interfaces (pre->solver, solver->post)
- Study & remove some bottle-necks in the utilization of the tools for large scale computing
  - Creation of meshes consisting of tens of millions of elements
  - Bottle-necks in preconditioning of linear equations
- Implement VMS to massively parallel framework
- Knowledge transfer
- Increase CSC's role as service provider in EU level projects via use of Elmer
- Increase supercomputing in the area of CFD

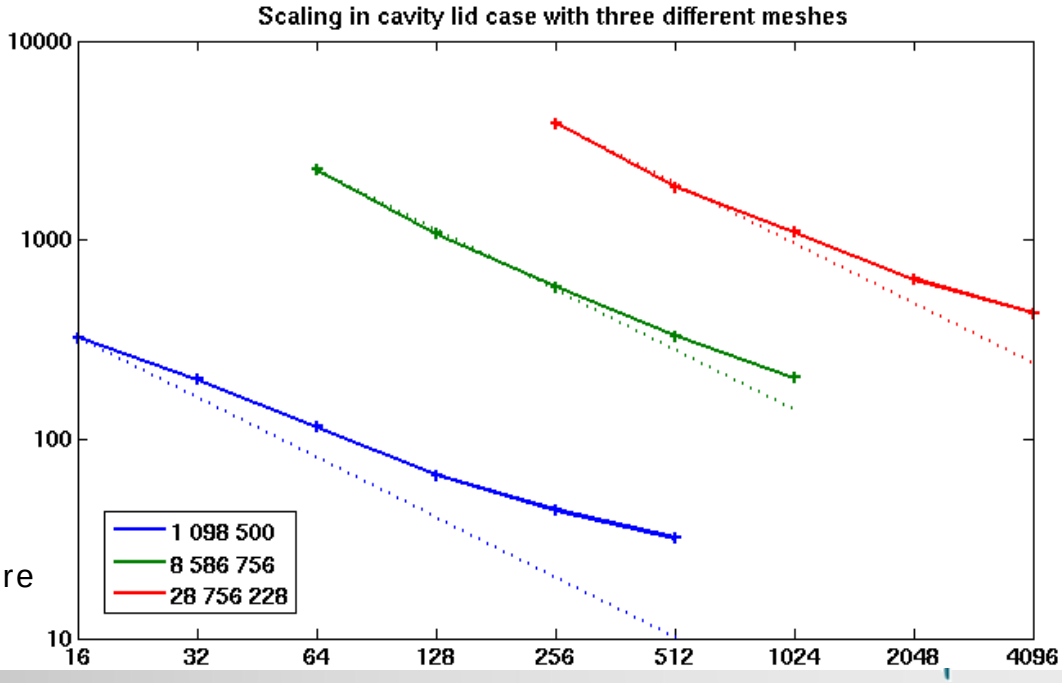
Elmer Parallel performanc

- Partitioning by Metis or simple geometric division
- Parallel assembly and solution by GMG or Krylov subspace methods.
- Parallel performance may scale up to thousands of cores
- Simulation with over one billion unknowns has been performed



Scaling of wall clock time with dofs in the cavity lid case using GMRES+ILU0. Simulation Juha Ruokolainen, CSC, visualization Matti Gröhn, CSC .

Louhi: Cray XT4/XT5 with 2.3 GHz 4-core AMD Opteron. All-in-all 9424 cores and Peak power of 86.7 Tflops.



OpenFOAM parallel performance

- The same cavity lid case with max of 22 Mcells
- Superlinear speedup up to 1024 processors on murska!

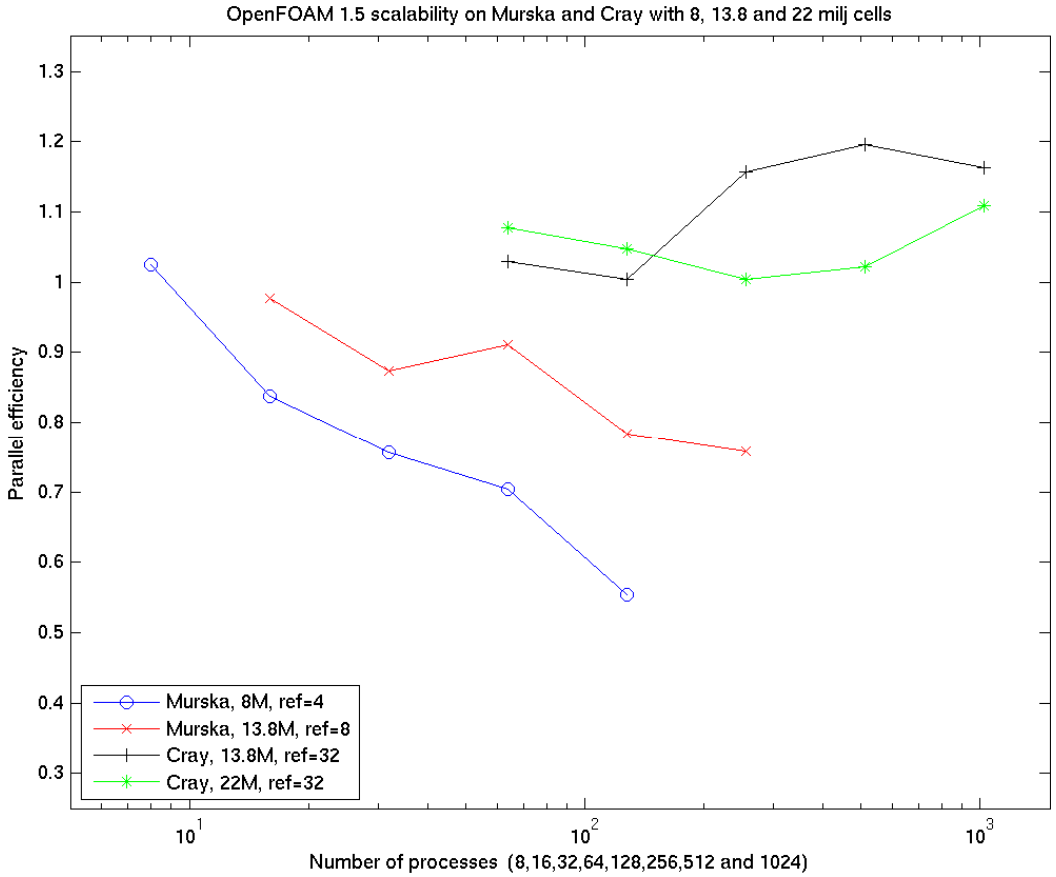


Figure by
E. Järvinen, CSC

Status of Free Software in Computational Engineering

Peter Råback
CSC - IT Center for Science

HTC Keilaniemi, Espoo
25.5.2009

Outline

- Why to use open source software?
- Examples of open source software in computational engineering
  - Some bias towards research
- Things to consider
  - Maturity
  - Life cycle
  - Development models
- Conclusions

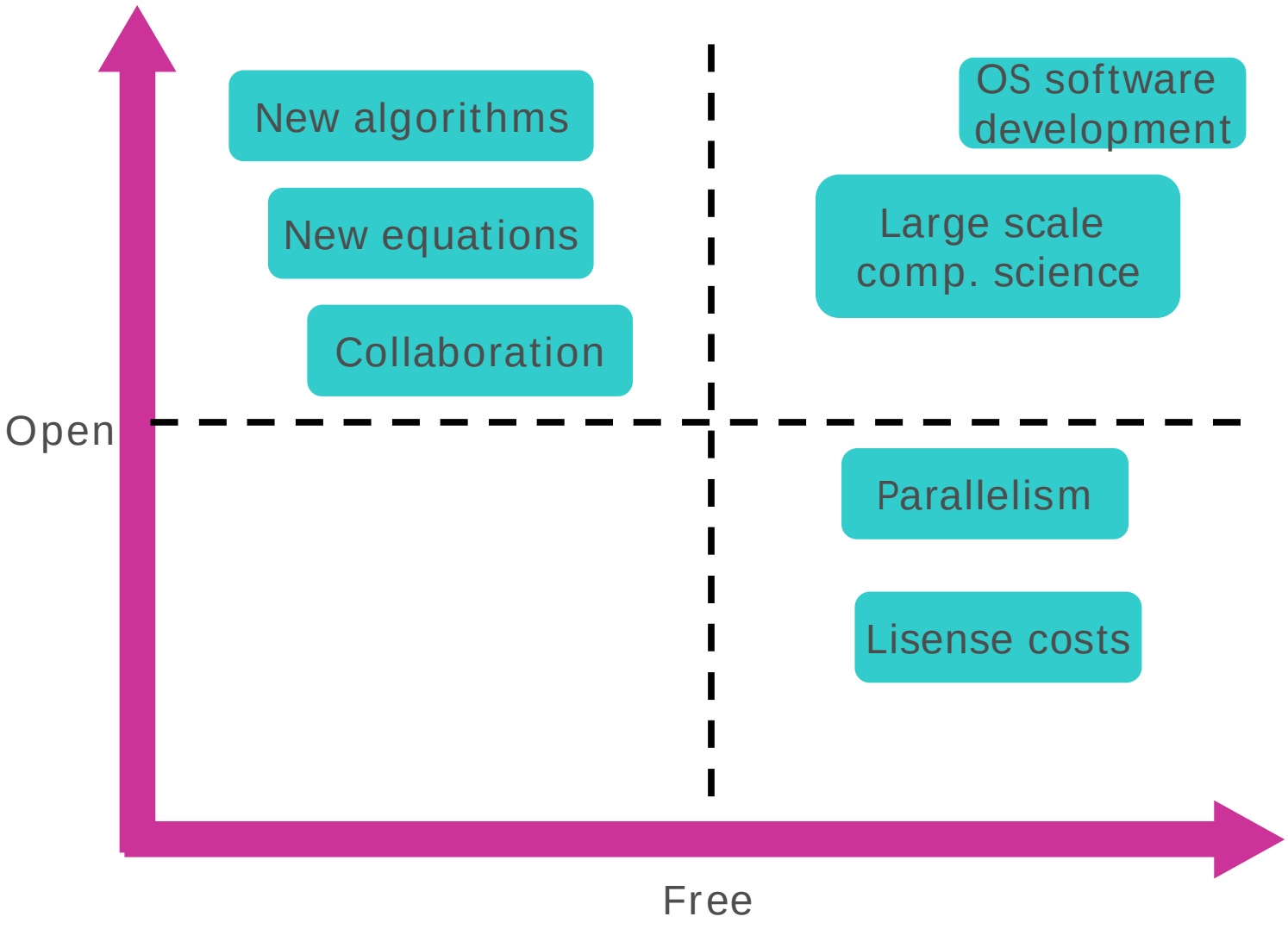
OS software: Users point of view

- Software is free
  - Labour is not, so this is not often the main argument
  - May be particularly important in large scale computations
- At least some control over the intellectual property
  - Own model development not related to proprieraty software
  - Copyleft lisences have a viral effect: own IP cannot be distributed under different license
- In collaboration all parties have access to the software
  - Companies, universities, consultants,...
- Also fundamental ideas may be tested with the software
  - Algorithms, models
  - Compatible with scientific method: falsification
- Open source software has more different roles
  - May be used to attract a wider spectrum of actors

OS software: Developers point of view

- Commercial markets are heavily consolidated and require large initial investments
- Community does tasks that should otherwise be mainly done by the vendor
  - Advertisement, porting, verification, contributions to the code
- OS software may heavily utilize other OS libraries thereby reducing the investment on development
- With copyleft (GPL) licenses the intellectual property may be ensured
  - No commercial exploitation of the code itself by others
- With dual licensing schemes propriarary versions possible
  - Modifications to the software
  - Examples: MySQL, Elmer
- Typically money must come from services
  - Expert services, common projects, courses
  - Examples: OpenFOAM, Chipster
- In-house software in difficult competitive position

Why to use open source software?



Computational Engineering software

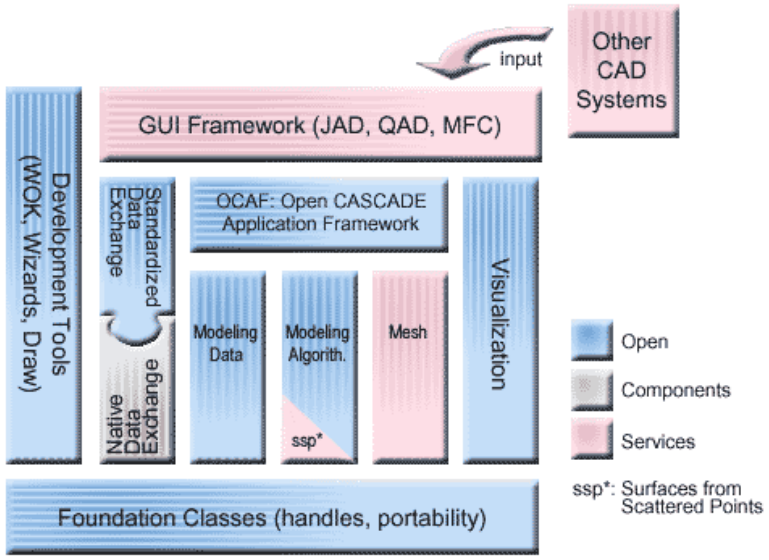
Open Source software with examples

- CAD
  - OpenCASCADE, Salome
- Mesh generation
  - Netgen, Tegen, Gmsh, Triangle (2D)
- Linear algebra:
  - Solvers: umfpack, Hypre
  - Packages: PetSc, Trilinos
- Analysis
  - Structural Mechanics: Code Aster, CalculiX, FELT, Impact
  - Fluid Mechanics: OpenFOAM, Code Saturne, GerrisFlow, Mfix, FeatFlow, FreeCFD
  - Electromagnetics: FastCAP (BEM), NGSolve
  - Multiphysics: Elmer, freefem++
  - FEM Libraries: deal.II, LibMesh
- Visualization
  - Library: VTK
  - End-user programs: Paraview, VisIt, Mayavi, OpenDX

CAD – OpenCASCADE

http://www.opencascade.com/
http://www.opencascade.org/

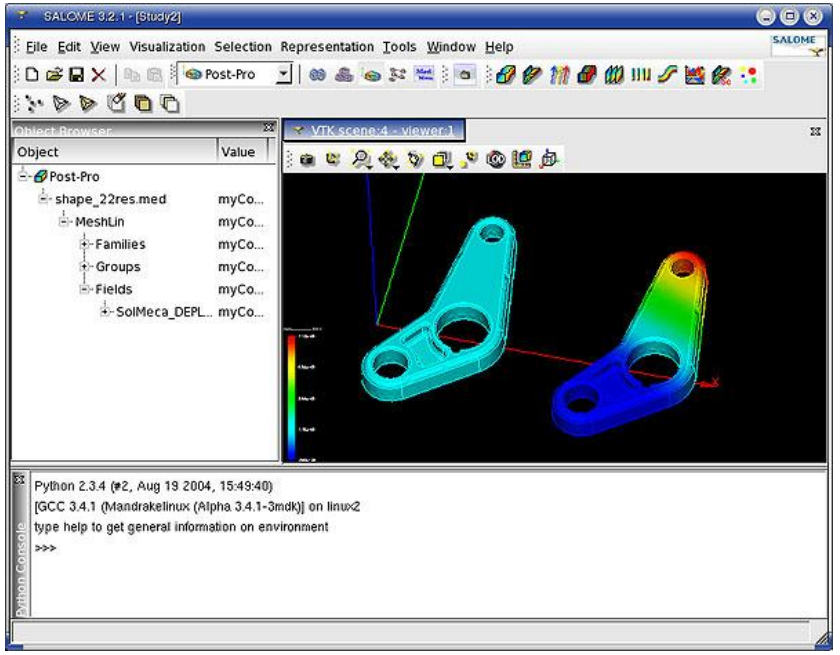
- What is it?
  - Open CASCADE is a powerful CAD/CAM/CAE kernel and development platform for 3D modeling applications.
  - It consists of reusable C++ object libraries and a set of development tools that are available in Open Source
  - Modular structure (see diagram)
- Devolopment history
  - EUCLID-IS CAD/CAM system 1987
  - Published under Open Source in 1999 as OpenCASCADE
  - OpenCASCADE S.A.S. is a service company of 80 developers
  - Curstomers CEA, BMW, SAMTECH, EADS, RINA, Alcatel,...
- The only proper CAD library under Open Source



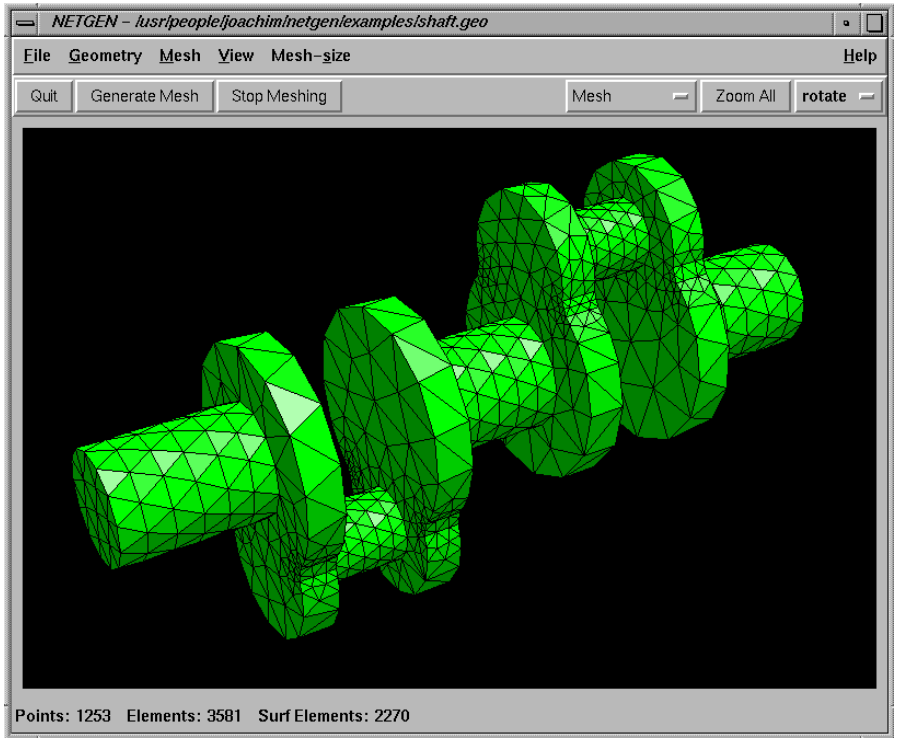
CAD – SALOME

http://www.salome-platform.org/

- What is it?
  - Free software that provides a generic platform for Pre and Post-Processing for numerical simulation.
  - Development started at year 2000 in a large French consortium
  - Goal was to make the use of a number of softwares easier
- Based on a number of free software libraries
  - Qt, OpenCASCADE, Doxygen, Python, VTK
- Main functions
  - Create/modify, import/export (IGES, STEP), repair/clean CAD models
  - Mesh CAD elements, check mesh quality, import/export mesh (MED, UNV, ASCII)
  - Handle physical properties and quantities attached to geometrical items
  - Perform computation using one or more external solvers (coupling)
  - Display computation results (scalar, vectorial)
  - Manage studies (creation, save, reload)



Meshing - Netgen

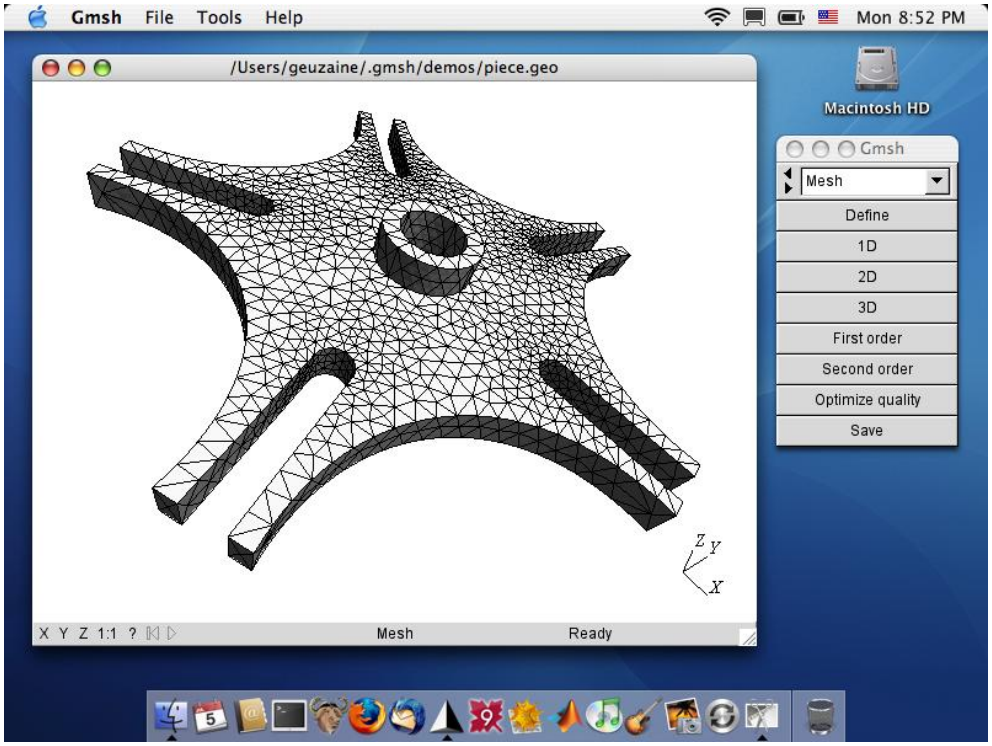


http://www.hpfem.jku.at/netgen/

- What is it?
  - An automatic 2D/3D tetrahedral mesh generator
  - Developed mainly by Joachim Schöberl from year ~2000
- Key features
  - Accepts input from constructive solid geometry (CSG) or boundary representation (BRep) from STL file format
  - Connection to OpenCASCADE for import of IGES and STEP files
  - Contains modules for mesh optimization and hierarchical mesh refinement
  - LGPL library
- Netgen library (netlib) is utilized by a large number of GUI projects
  - Gmsh, SALOME, ElmerGUI,...

Meshing - Gmsh

- Best OS mesh generator for academic use?
- Uses OpenCASCADE, netgen
- More by Juha Kortelainen



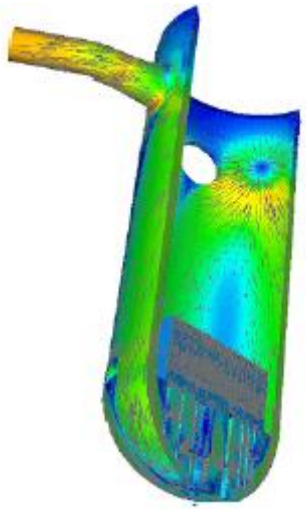
FVM - OpenFOAM

http://www.opencfd.co.uk/openfoam/

- No 1 CFD software under open source
- Features
  - Based on C++ modules which are used to build number of solvers
  - Uses finite volume numerics to solve systems of partial differential equations ascribed on any 3D unstructured mesh of polyhedral cells.
  - Comes with models for fluid flows involving chemical reactions, turbulence and heat transfer
  - Includes some rude utilities for pre- and post-processing
  - Fully parallelizable with iterative solvers
  - License under GPL
- OpenFOAM may be the best example of OS service in CE
  - Started as a PhD project in 1993
  - Now 1000's of serious users
  - Several small consultancy companies base their operation on OpenFOAM

FVM: Code_Saturne

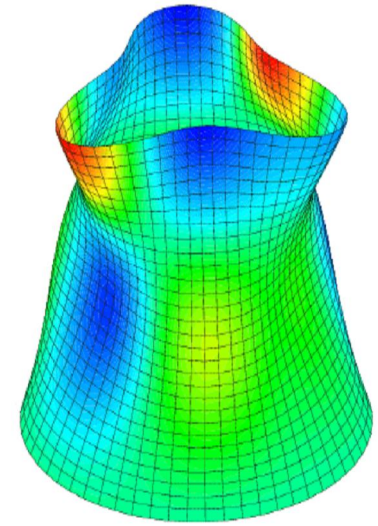
- Developed since 1997 at Électricité de France (EDF)
- Published under open source in 2007
  - Main reasons for publications to ensure that the use of the code in many collaborative research projects
- Big project
  - Validated underEDF'sQuality Assurance, ISO9001
  - 500 000 lines of code
    - 49% Fortran 77, 41% C99, 10% Python
  - Partially translated to english
- Features
  - General purpose computational fluid dynamics software
  - Finite Volume approach accepts meshes with any type of cell grid structure
  - Incompressibleor expandable flows with or without heat transfer and turbulence
  - Dedicated modules in radiative heat transfer, combustion, magneto-hydro dynamics, compressible flows, two-phase flows
  - Parallel scaling demonstrated up to thousands of processors
- Integrated with SALOME but not available publicly yet



FEM – Code_Aster

http://www.code-aster.org/

- Developed since 1989 at EDF
- Published under GPL in 2001
- Big project
  - Core development team: 20 engineers
  - Application development: 40 engineers
  - 200 in-house users
  - 1200 000 lines of code (Fortran & Python)
  - 1256 documents, 13800 pages (mostly in French)
  - 2000 consitancy tests, ISO 9001
- Structural mechanics and thermomechanics with FEM
  - 400 element types, 95 constitutive laws
  - Geometric nonlinearities, contact friction,...
- EDF not in charge of the Free distribution
  - www.caelinux.com -> Salome_Meca



FEM - Elmer

http://www.csc.fi/elmer

- Multiphysical simulation software under open source
  - End-user program rather than library
- Developed at CSC – IT Center for Science since 1995
  - 300 000 lines of code
- Published under open source in 2005
  - Public version control
- Features
  - Around 20 physical models and 20 auxialiary solvers
  - Large number of different element types
  - High level of abstraction ensures easy implementation of new physics
  - A large selection of direct and iterative solvers
  - Parallel scaling achieved up to 4000 processors
- Uses many open source libraries
  - CAD: OpenCASCADE
  - Meshing: Netgen, Tetgen
  - Lin.Alg: Umfpack, MUMPS, Hypre, Lapack, Parpack
  - Visualization: VTK

  (more on Mikko's presentation)

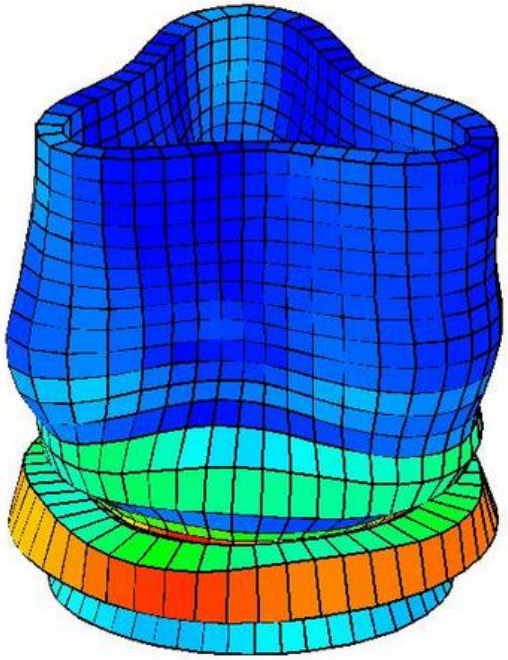
FEM – freefem++

http://www.freefem.org/ff++/index.htm

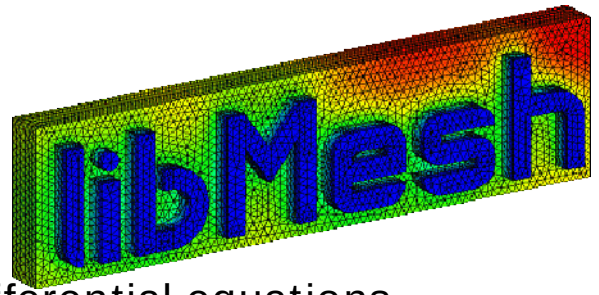
- A language dedicated to the finite element method that enables easy solution of Partial Differential Equations (PDE)
- One of the 1st free softwares
  - Development started in 80's (O. Pironneau, F. Hecht et al.)
  - Idea has been copied (Comsol multiphysics etc.)
- Mainly educational use nowadays
- GUI: FreeFem++-cs

FEM library – deal.II

- What is it?
  - A Finite Element Differential Equations Analysis Library
  - A program library rather than end-user program
  - Computational solution of partial differential equations using adaptive finite elements
  - Uses state-of-the-art programming techniques to offer you a modern interface to the complex data structures and algorithms
  - main aim is to enable rapid development of modern finite element codes
  - Good demonstration of a modern approach taking use of the best available tools



FEM library - libMesh



- What is it
  - Library for the numerical simulation of partial differential equations using arbitrary unstructured discretizations on serial and parallel platforms
  - Provides adaptive mesh refinement computations in parallel
  - libMesh currently supports 1D, 2D, and 3D steady and transient finite element simulations.
  - Makes use of high-quality whenever possible:
    PETSc, LASPack, SLEPc, Metis, Triangle, Tetgen
  - Active development:
    Univ. of Texas at Austin, Technische Universität Hamburg, Sandia National Laboratories, NASA Lyndon B. Johnson Space Center

  - Rather similar approach as in deal.II except emphasis on CFD (vs. CSM)

Linear algebra

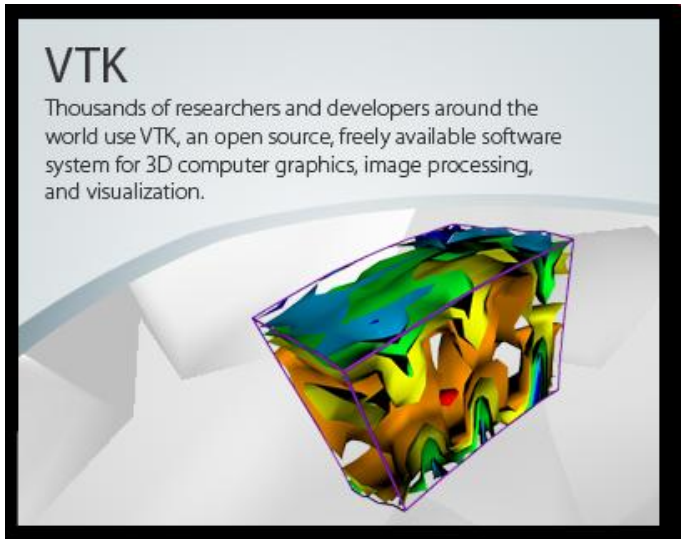
- This area in inherently part of academic developments
  - Many of the best products are published under Open Source
- Linear algebra for dense matrices
  - Lapack
- Direct sparse solvers
  - Umfpack, Mumps, Spools, ...
- Eigenvalue solvers
  - Arpack, Parpack
- Ireative solvers, preconditioners
  - Hypre
- Graph partitioning
  - Metis, Scotch

- Collections of all of the above
  - PETSc, Trilinos

Visualization - VTK

http://www.vtk.org/

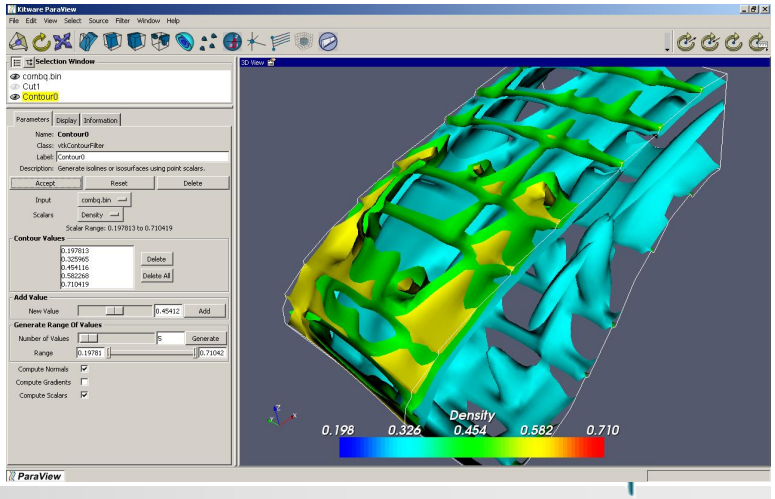
- What Is it?
  - Software system for 3D computer graphics, image processing, and visualization
- Features
  - Consists of a C++ class library and several interpreted interface layers including Tcl/Tk, Java, and Python.
  - VTK supports a wide variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods
  - Supports parallel processing
  - Integrates with various databases on GUI toolkits such as Qt
  - VTK is cross-platform and runs on Linux, Windows, Mac and Unix platforms. An open-source, multi-platform data analysis and visualization application
- Professional support provided by Kitware Inc.
  - Proper documentation not free
  - Supported by a number of large institutions: Los Alamos an Sandia nat.lab.



Visualization - Paraview

http://www.paraview.org/

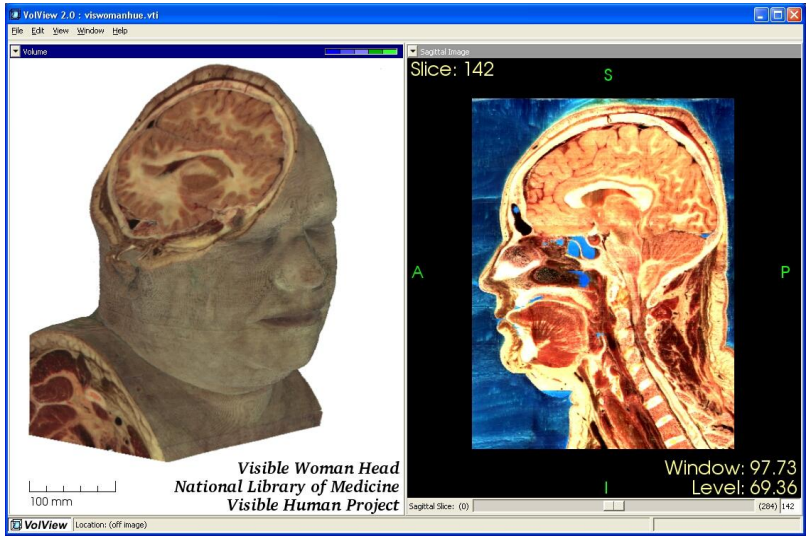
- What Is it?
  - An open-source, multi-platform data analysis and visualization application
  - Developed to analyze extremely large datasets using parallel computing
  - Developers include major Us labs (Los Alamos, Sandia,...)
  - Probably the most popular visualization tool under Open Source
    - Used by OpenFOAM as ParaFOAM
- Features
  - Data exploration may be done interactive or using batch processing
  - Can be run on laptops and supercomputers
  - Based on VTK library



Visualization - VisIT

http://wci.llnl.gov/visit/

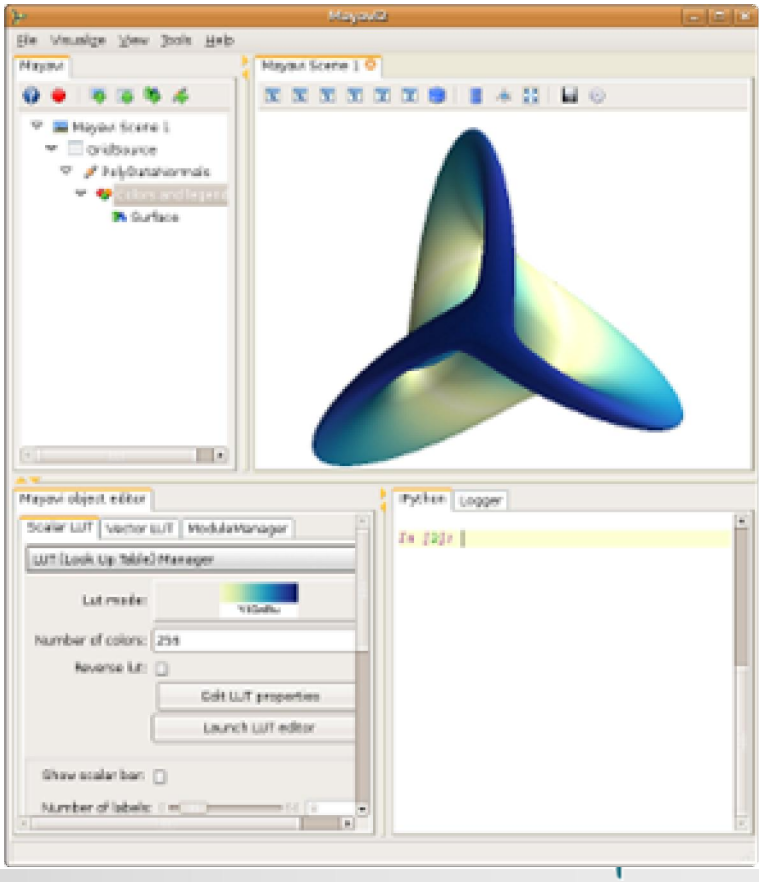
- What is it?
  - Interactive parallel visualization and graphical analysis tool for viewing scientific data on Unix and PC platforms
  - Developed at Lawrence Livermore national laboratoris
  - Rather similar in features as Paraview
  - Distributed under BSD license



Visualization - Mayavi

http://code.enthought.com/projects/mayavi/

- What is it?
  - Easy and interactive visualization of 3-D data
  - Includes a (optional) rich user interface with dialogs to interact with all data and objects in the visualization.
  - A simple and clean scripting interface in Python, including one-liners, or an object-oriented programming interface.
  - Based on the VTK toolkit



CAE Linux

open-source powered engineering

- Many large codes depend on a number of old libraries and may be difficult to compile under different Linux systems
  - OpenCASCADE
- CAELinux is a Linux environment for computational engineering based on PCLinuxOS
- CAELinux has been selected by EDF to be part of the official distributors
- Comes with a large number of precompiled codes
  - SALOME_MECA (=SALOME + Code_Aster)
  - CFD: OpenFOAM, Gerris, Code-Saturne, Openflower
  - CAD/pre/post packages: Paraview, Visit, OpenDX, Netgen, Tetgen, GMSH, Salome, Qcad
  - Elmer, Calculix, Tochnog, Impact, MBDyn
  - Scientific tools: GNU Octave, R & Rkward, Scilab, wxMaxima
  - Developement tools and compilers.
- Problem:
  - Update cycle rather long: 14.10.2005, 15.4.2008,....
  - One person (Joël Cugnoni) in charge?

Status by discipline

- CAD
  - OpenCASCADE the only noteworthy option but the age starts to show
  - Not in par with commercial tools
- Meshing
  - Rather few alternatives mainly on tri/tet meshing
  - More in next presentation
- Linear algebra
  - Plenty of good libraries in open source
  - In massively parallel simulations edge over commercial alternatives
- Solvers
  - Looking at simulations alone there are many good codes
  - Often the user interface rather poor
- Visualization
  - All notable software builds on VTK
  - Paraview most popular
  - Rather competitive with commercial codes
- Workflow
  - Not as smoothly integrated as in commercial tools

Driving force in OS software development

- The standard of software varies a great deal depending on area
- Areas where open source software flourish
  - Academic research; soft funding
  - Things that computer wizards like to do
  - Simplicity
  - Bottom-up planning works
  - Examples: linear algebra, A/V software, Linux
- Areas where commercial software flourish
  - Big business; hard money
  - Things that nobody would do for free
  - Complexity
  - Top-down planning works
  - Examples: CAD, accounting software, Windows
- User need is not a sufficient driving force for open source software
  - E.g. there is no mechanism that would develope the next generation CAD engine under open source

Origins of open source software in CE

- US Governmental research institutes
  - Code generated by public funding may not be commercialized
  - Linear algebra & visualization
    (PetSc, Hypre, VTK, Paraview, ViSit)
- Large French companies
  - Matra Datavision: OpenCASCADE
  - EDL: SALOME, Code_Saturne, Code_Aster
- Other governmental institutes
  - CSC, Finland: Elmer
  - New Zealand: GerrisFlow
- University research groups (individuals)
  - JKU Lintz (Joachim Schöberl): netgen
  - U of Liege (Christophe Geuzaine): gmsh
  - Université Pierre et Marie Curie (Olivier Pironneau/Frederich Hecht): FreeFem++
  - Imperial College (Hrvoje Jasak, Henry Weller): OpenFOAM
  - U of Michigan: FreeCFD
- Independent individuals
  - ?

Cathedral vs. Bazaar development model

- For new developers the different development models make a difference
  - In Cathedral model it is usually rather difficult to get a recognized role and influence the direction development
  - In Bazaar model new developers are measured by their merit alone
- New developers more drawn to bazaar development model
- Many projects have features of both models, or may change in time
  - OpenFOAM: Cathedral gone partially Bazaar?
- For users there is not so much difference
  - The cathedral model may ensure more steady future

Cathedral vs. Bazaar: Examples

Cathedral: Code_Saturne

- History:
  - Developed since 1997 at EDF
  - Published OS in 2007
- Existed long as an in-house project
  - Validated under EDF's Quality Assurance
  - 500 000 lines of (legacy) code
  - Partially translated to english
- Open source start
  - Main reasons for publications to ensure the use of the code in many collaborative research projects
  - Rather passive start, outside contributions of no importance
  - Public version lagging behind

Bazaar: FreeCFD

- History
  - Development since 2008 by two grad students of University of Michigan
- Modern approach
  - Object oriented with C++
  - Many good software libraries adopted
- Initially designed as an open source project
  - First version was published within months
  - It is almost impossible to find references to the U of Michigan
  - Discussion forum already has 428 members

Can the same approach be ideal for industry and grad student?

Continuity and OS software

- The lifespan of software is maybe 5-20 years before it needs to be completely rewritten
- Commercial software vendors ensure backwards compabity
- In open source software continuity is more difficult to ensure
  - 175 261 projects on sourceforge alone!
  - Often a small group of key persons – even one
- Extra care must be set in choosing the right tools:
  - Background organization, available support
  - Activity of user community
  - Technologies used
- From company point of view risks should be minimized
  - Keeping eyes open
  - Favoring standardized formats

Activity of an OS project

- There are many open source software that are no longer actively supported
  - Key person(s) has finished her thesis
  - Funding may have ended
  - The legacy code has become too difficult to further develope
- Signs of activity
  - Solid institution / company in charge (cathedral model)
  - Large & active development community (basaar model)
  - Active user forum
  - New developments (activity in version control system)
  - …
- Signs of a dying code
  - Primary developers no longer active
  - Code has fallen too much behind in technologies used
  - Lot’s of bugs
  - …

CE software - conclusions

- The status of open source software varies a great deal in different disciplines
  - Solver & postprocessing well in par with commercial alternatives
  - It is rather common to combine OS solution with some commercial software
    - E.g. CAD/meshing proprierity + solution open source
  - Field is not that large but may still change quite rapidly
- OS software could be
  - An ideal tool for collaborative research
  - An environment which enable the study of new features
  - Means for saving some money
  - A way to gain some control over own developments
- OS sotware probably won't
  - Offer sufficient complete solution for big business
  - Be an ideal step in a rapid development cycle
  - Be as easy to use as the commercial alternatives

More info

- Karl Fogel: *producing open source software*
  http://producingoss.com/
- CFD Online: www.cfd-online.com
- CAELinux: www.caelinux.com
- Opennovation: http://www.opennovation.org/
- Fernando Varas: Free software for numerical simulation
  http://hermite.mac.cie.uva.es/ehf2008/transparencias/FVaras.pdf

Thank you for your attention!

CSC