FreeFEM days

Freefem and the HPC

Olivier Pironneau

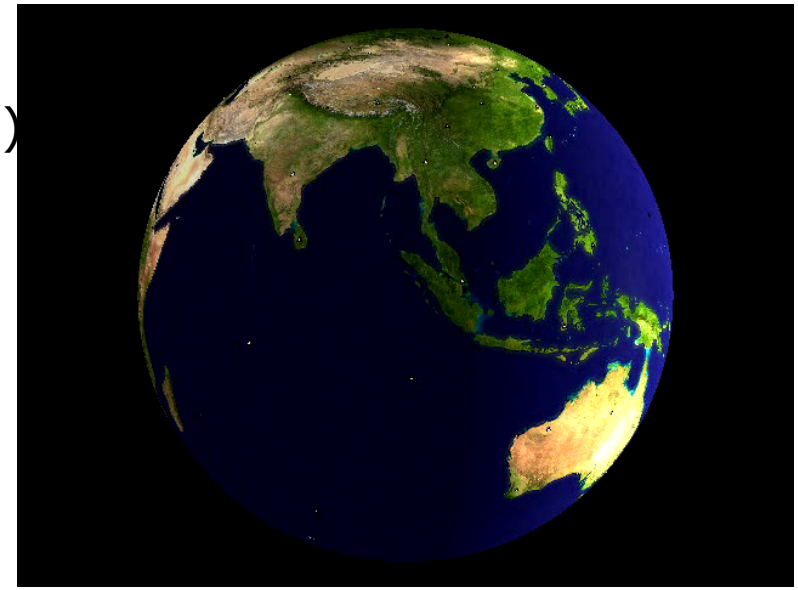
Université Pierre et Marie Curie

Conseil Stratégique pour le Calcul Intensif (CSCI)

Applied mathematicians will not be satisfied till the entire universe has been simulated (Paul Caseau 2001)

Numerical Simulation is a revolution in ALL Sciences

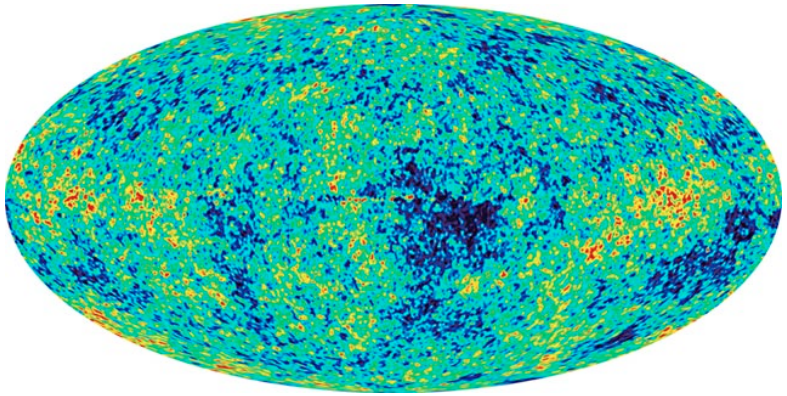
- ***Simulate to*:**
  - Estimate a system before building it
  - Understand what is inaccessible
  - Archive knowledge
  - Assert scientific hypothesis (Astrophysics)
  - Resolve societal challenges(climat, pollution, energy)

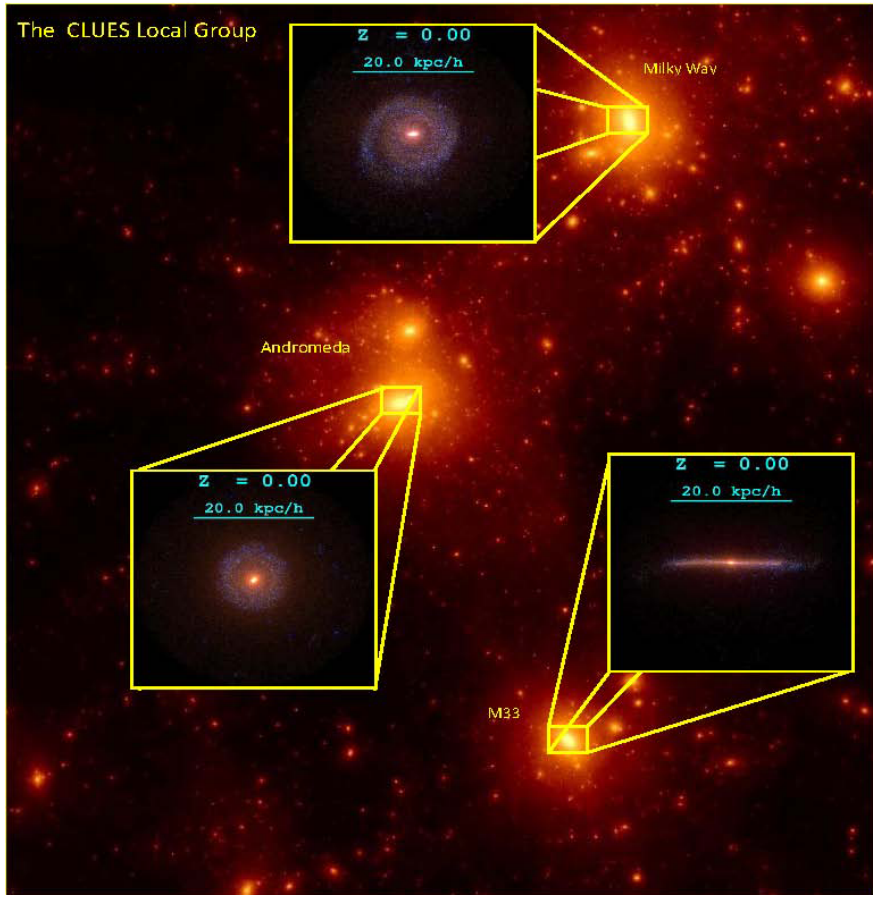


- ***A cultural revolution* :**
- Remplace theory - experience
  By theory - simulation - experience.

Each research team has experts in the 3 domains
e.g. : earthquakes (D. Komastich)

Astrophysic: a revolution





6 dimensional, 10^{55} dark matter particles

This simulation: 100 billion particles

Multi phase – multi scale

Radiative transport

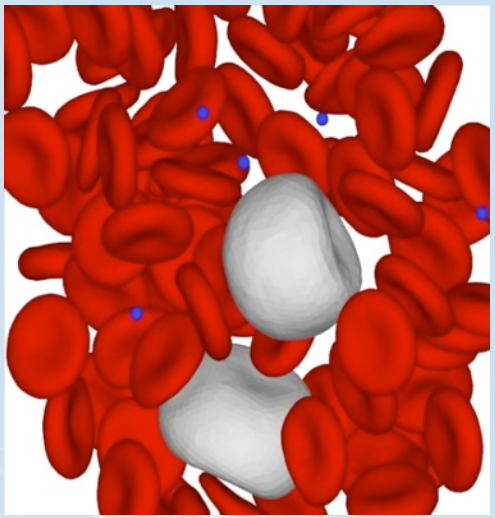
Long range forces, no saturation effects

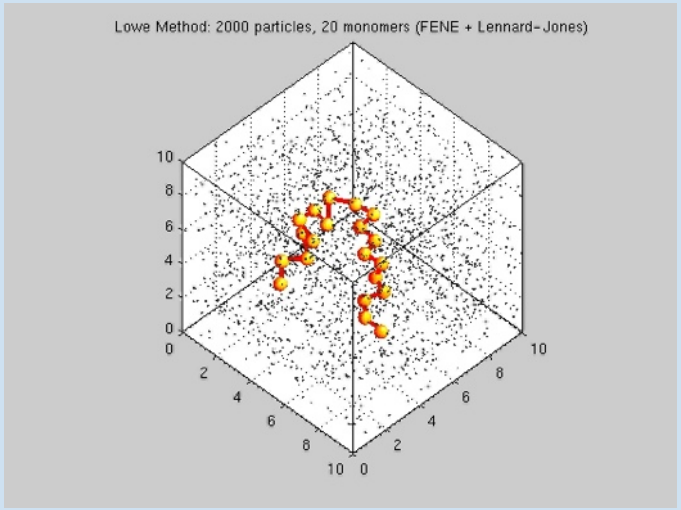
Unresolved ill-understood subgrid physics

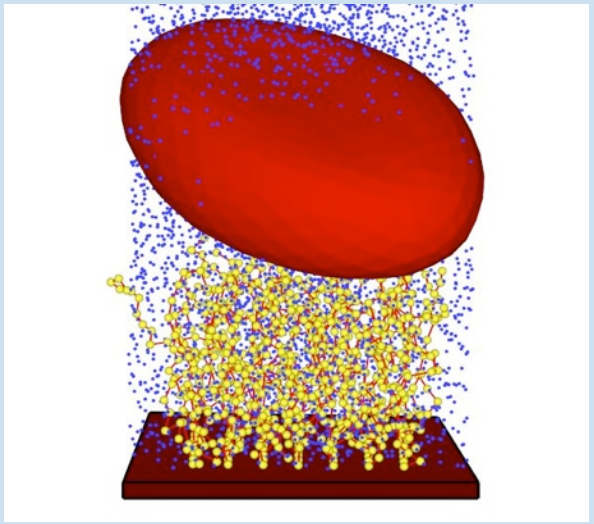
Light cone effects

Huge data volumes to be mined

Multiscale Modeling of Complex Fluids, Soft Matter, and Red Blood Cells







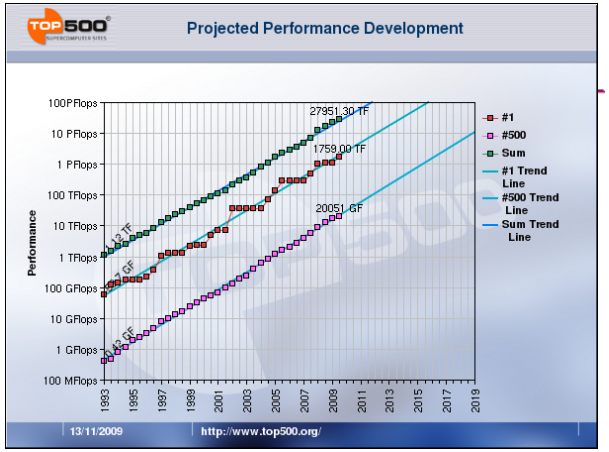
George Em Karniadakis

Applied Mathematics, Brown University

7168 Tesla M2060 et 14,336 CPU

Now 10^6+ hybrid

Limited only by the power consumsion

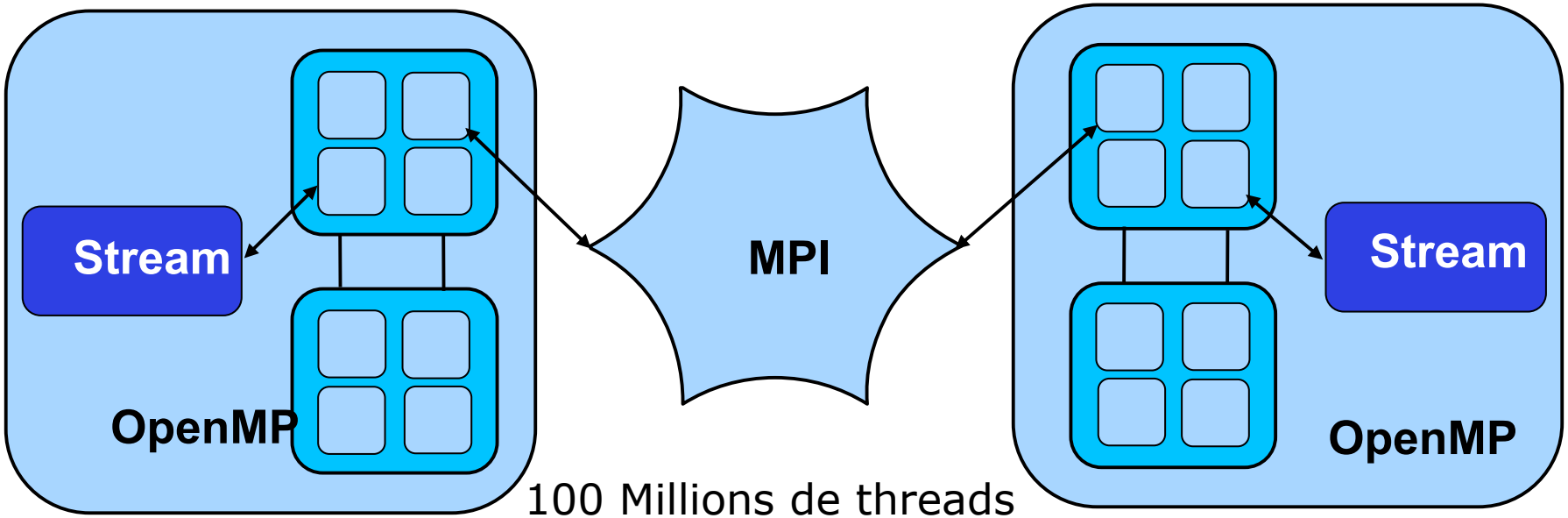


Fujitsu K computer 10.7 Pflops, The machine CURIE in CCRT> 1 Pflops

See www.GENCI.fr

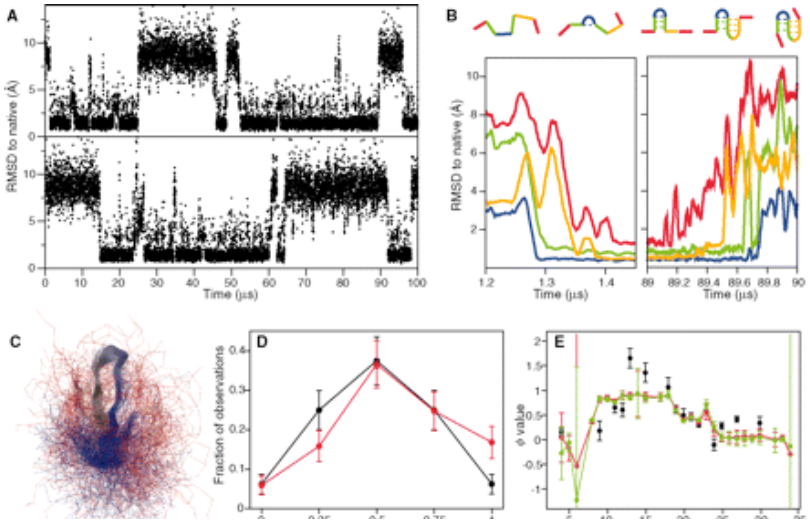
In 2018 exaflops with millions proc and the Pflops in your computer

HPC hardware + HPC middleware => Performance



100 Millions de threads

The ANTON Chip by D.E. SHAW



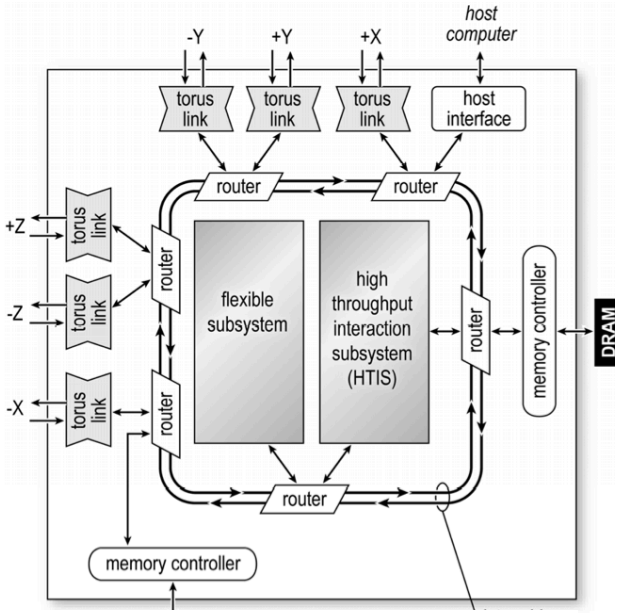
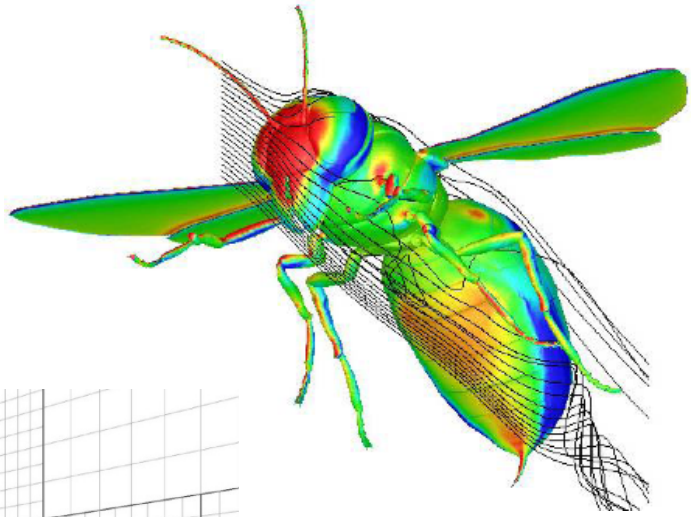


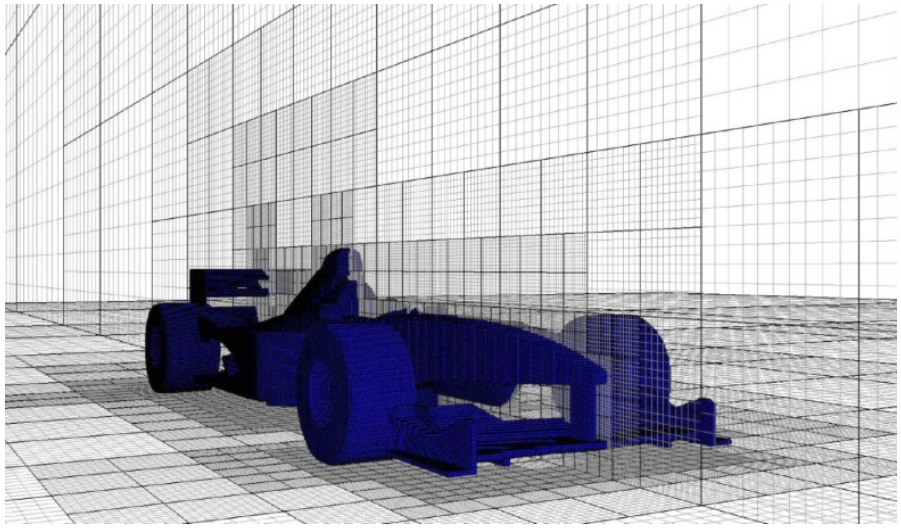
Table 2. Comparison of execution time of a long-range time step for a 512-node Anton machine and a single Xeon processor running GROMACS. Note that the length of the Anton time step is less than the sum of the task times due to parallelism. Benchmark parameters were used from Table 2 of [16].

Phase	Task	GROMACS Time in μs (%)	Anton Time in μs (%)	Speedup
Force Calculation	Range-limited forces	111,000 (61)	2.2 (10)	50,000
	FFT, inverse FFT and Fourier space multiplication	28,700 (16)	7.8 (39)	3,700
	Charge spreading and force interpolation	18,800 (10)	3.5 (17)	5,400
	Bond forces	8,780 (5)	1.9 (9)	4,600
	Correction forces	6,600 (4)	1.8 (9)	3,700
Integration	Position and velocity updates	2,670 (1)	2.2 (11)	1,200
	Constraint calculations	3,000 (2)	2.2 (11)	1,400
	Temperature computation	1,230 (1)	3.1 (15)	400
Other	Atom migration between nodes	NA	3.5 (12)	NA
Entire long-range time step		181,000	20.2	9,000

Algorithmic scaling?

- Schemes adapt to the architecture
- Explicit Schemes?
- Semi-structured Mesh(AMR)
- Industry will follow?





K. Nakahashi

On the Software Side

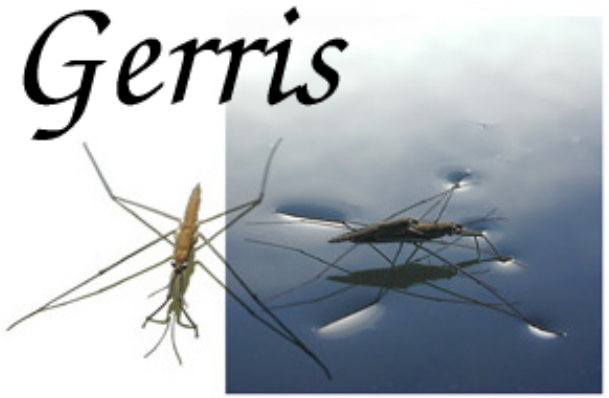
C++, Python, fortran, MPI, CUDA

Matlab, SciLab

COMSOL, FreeFEM++

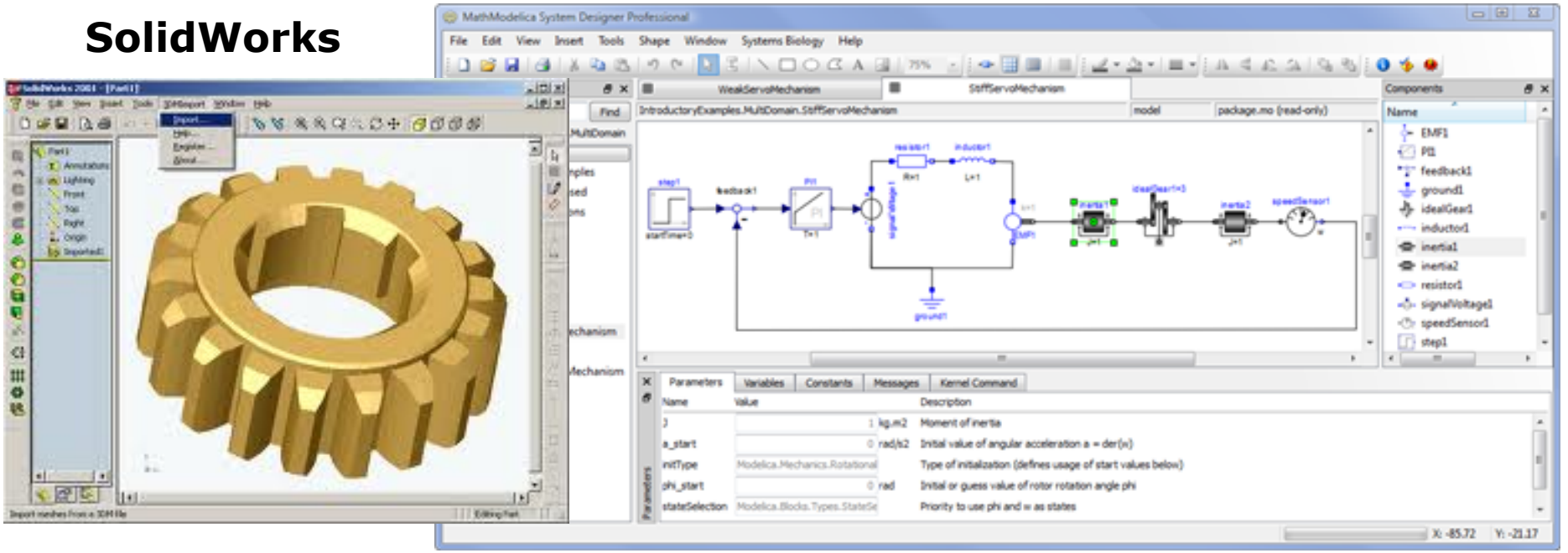
Maple, Mathematica, SAGE

Fluent, Aster, openfoam, gerris, SolidWorks



SolidWorks

Modelica



Position of FreeFEM++

- FreeFEM was started in the eighties, FreeFEM++ in 2001
- (tenth anniversary?) No competition has survived yet

The good

- Superset of C/C++ for mathematicians, (a bit like matlab)
- Open source
- Multiphysics
- Advanced mesh generation and adaptation in 2D
- Fast learning curve for mathematicians
- Now 3D and parallel

The not so good (authors are prof on the FEM fluid incompressible side)

- Variational formulations
- Weak for hyperbolic systems
- Weak in 1D
- No CAD
- Minimal graphics
- Does not take advantage of the hardware of the machine

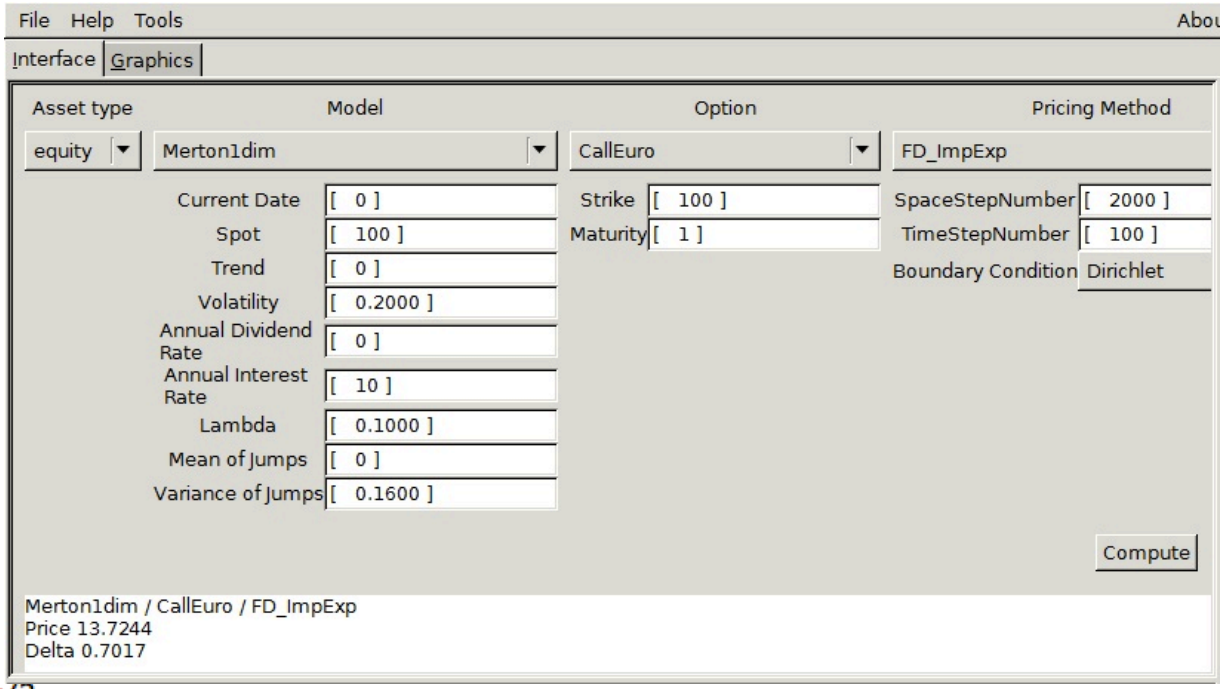
Possible Developments

- **Templates adapted to each fields:** where the vocabulary of the Field is used (Lamé constants for example) with automatic generation of the PDE in variational form

```
// file BlackScholes2D.edp
int m=30,L=80,LL=80, j=100;
real sigmax=0.3, sigmay=0.3,
real rho=0.3, r=0.05, K=40, dt=0.01;
mesh th=square(m,m,[L*x,LL*y]);
fespace Vh(th,P1);

Vh u=max(K-max(x,y),0.);
Vh xveloc, yveloc, v,uold;

for (int n=0; n*dt <= 1.0; n++){
  uold=u;
  solve eq1(u,v,init=j,solver=LU)
  = int2d(th)(  u*v*(r+1/dt)
     + dx(u)*dx(v)*(x*sigmax)^2/2
     + dy(u)*dy(v)*(y*sigmay)^2/2
     + dy(u)*dx(v)*rho*sigmax*sigmay*x*y/2
     + dx(u)*dy(v)*rho*sigmax*sigmay*x*y/2)
     + int2d(th)( -v*convect([xveloc,yveloc],dt,uold)/dt)
     + on(2,3,u=0);
}
plot(u,wait=1,value=1);
```



$$\partial_t u + \frac{1}{2}\sigma_i\sigma_j\rho_{ij}x_i x_j \partial_{x_i x_j} u + (r-\delta)\partial x_i u - ru = 0, \;\; u(x_1, x_2, T) = (K - x_1 - x_2)^+$$

Possible Developments (II)

- **Reintroduce the strong form of the PDE**

$$-\Delta\psi = \omega, \quad -\Delta\omega = f, \quad \psi|_\Gamma = 0 \quad \frac{\partial\psi}{\partial n}|_\Gamma = g$$

```
solve(psi,om){
    pde(psi) om -laplace(psi) = om;
        on(a,b,d) dnu(psi)=0; on(c) dnu(psi) = 1;
    pde(om) - laplace(om) = 0;
        on(a,b,c,d) dnu(om) + psi*1e8 = 0; };
```

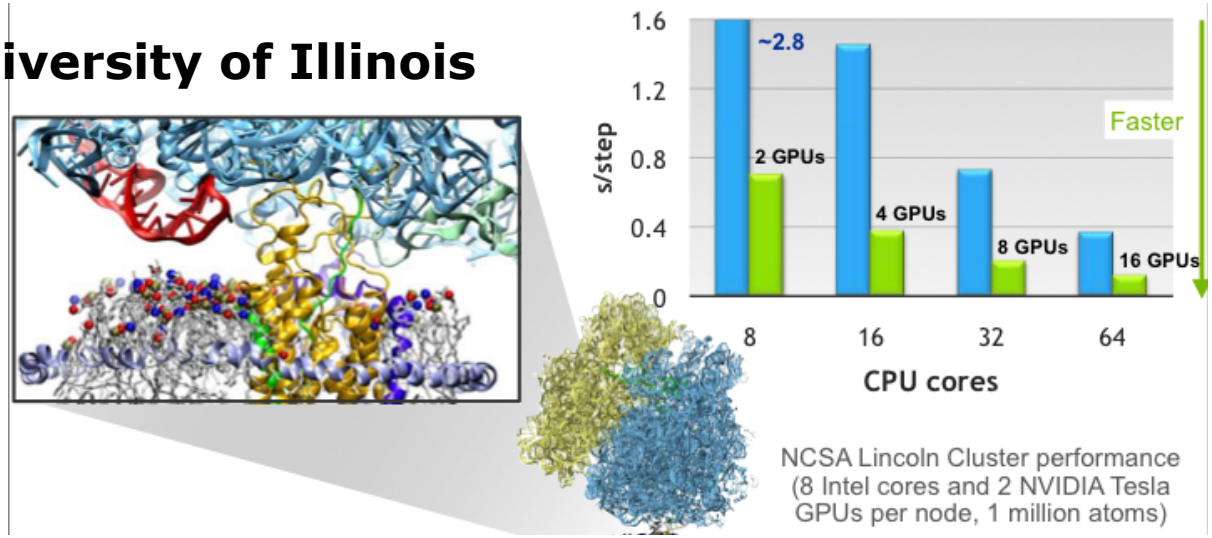
- Ambiguity of boundary conditions
- Another layer away from the linear system solver
- **Have it as an external module** which generates

```
solve([psi,om],[_psi,_om]) = int2d(Th)(dx(psi)*dx(_psi)
    +dy(psi)*dy(_psi)+dx(om)*dx(_om)+dy(om)*dy(_om))
    +int1d(Th,a,b,c,d)(om*_om + 1e8*psi*_om)
-int2d(Th)(om*psihat)-int1d(Th,c)(1*_psi);
```

Possible Developments (III)

- **Automatic detection of the architecture of the computer** and MPI adapted automatically to the number of processors and GPU

Klaus Schulten, University of Illinois



- **Easier interface to add on modules (dll)**
…. And the recompilation of FreeFEM++ is hard…. Hotline with Frédéric needed

- **Web interface, freefem is executed on our machine**

Possible Developments (IV)

Connexion with SAGE
Past attempt in Scilab unconvincing

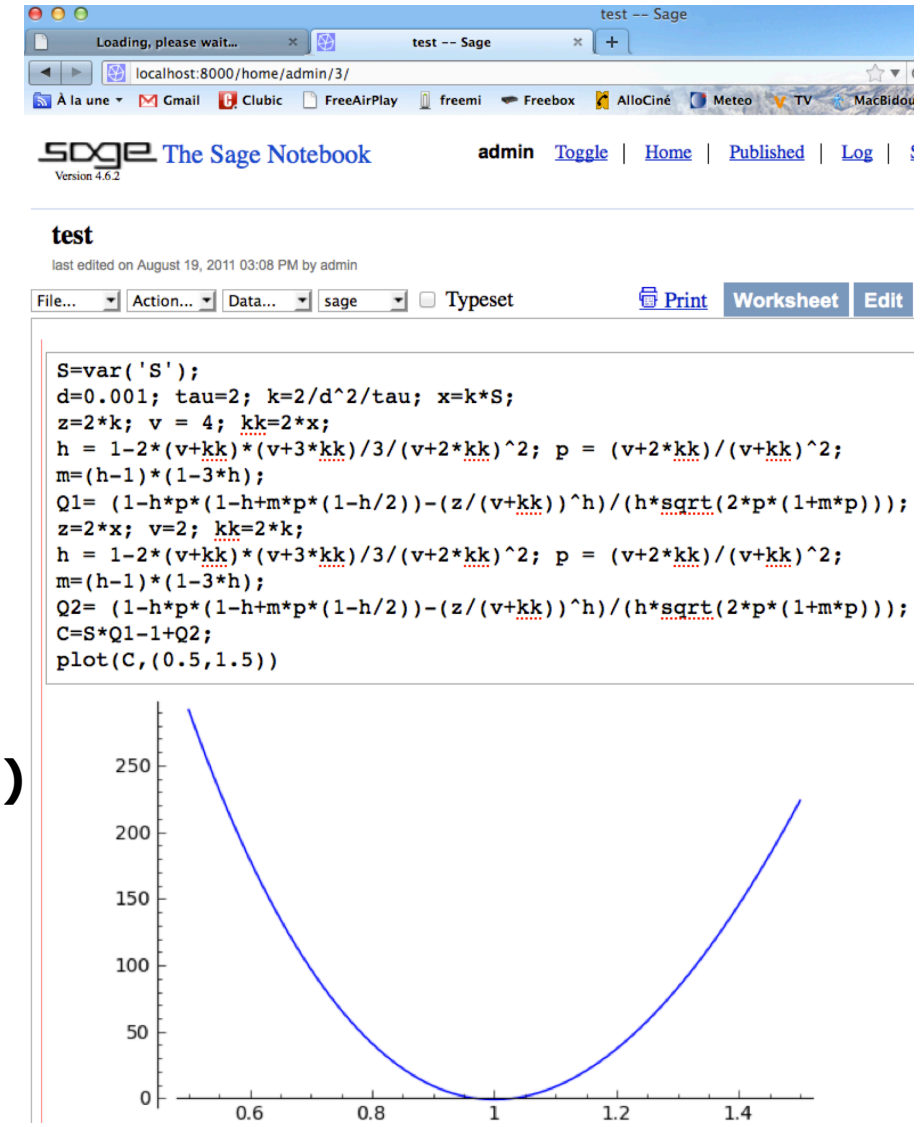
Freefem(« Th=square(10,10); »)

PARI in SAGE shows what to do

For example,

```
pari_absolute_base_polynomial()
x = polygen(ZZ)
K.<a, b> = NumberField([x^2 + 2,
x^2 + 3]);
K.pari_absolute_base_polynomial()
```

y^2+3



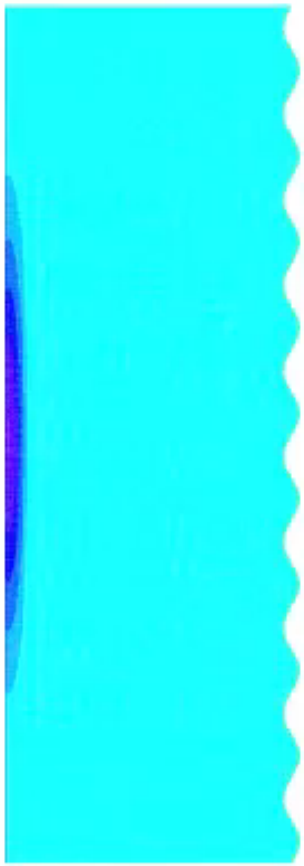
Possible Developments (V)

- **New modules** for spectral methods
  Wiener polynomials…
  New finite elements

  New modules for interfacing other softwares

  Better documentation

  Better tools for generating movies



But there is a limit to what a single man can do

Already much better organized for 3rd parties
e.g. freefem++-cs (Antoine Lehyaric)

Perspective

Who can say where scientific computing is going?

It is currently driven by military applications ... and video games.
Parallel computing is an unsolved challenge

Will FreeFEM++ be copied or improved?

Will it survive if Frédéric Hecht goes on to something else?

... like TeX without D. Knuth, because some things in TeX where never improved, like its line breaking algorithm and here the triangulation algorithm.