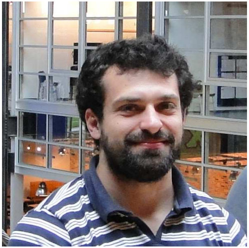
Microwave Imaging for Brain Stroke Detection and Monitoring using High Performance Computing

Victorita Dolean, Frédéric Hecht, Pierre Jolivet,
Frédéric Nataf and Pierre-Henri Tournier

TEAM



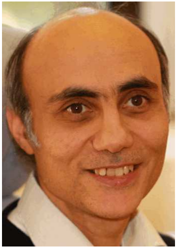



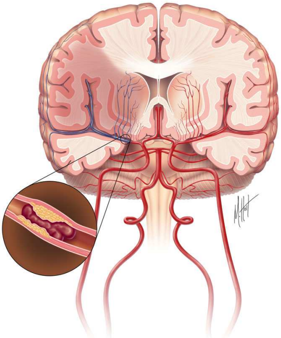

FIGURE: Pierre Jolivet, Frédéric Hecht, Pierre-Henri Tournier, Frédéric Nataf, Victorita Dolean

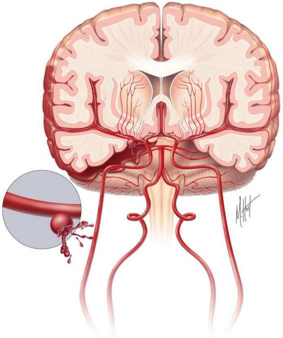
Cerebrovascular accidents (CVA)

120 000 people each year in France

Two types of cerebrovascular accidents :

ischemic (80%) | hemorrhagic (20%)





The therapy strongly depends on the diagnosis :

$\Longrightarrow$ thin the blood

$\Longrightarrow$ lower blood pressure

Imaging of CVAs

In order to distinguish between ischemic and hemorrhagic CVA, CT scan or even better MRI are currently used.
Electromagnetic tomography systems ($\sim$ 1 GigaHertz) are investigated.

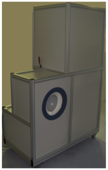
	CT scan	MRI	Microwave imaging
Resolution	very good	excellent	good
Safety	✗	✓	✓
Mobility	$\sim$	✗	✓
Cost	$\sim$ 300 000 €	$\sim$ 1 000 000 €	$<$ 100 000 €
Accessibility	✗	✗	✓
Monitoring	✗	✗	✓

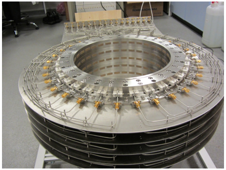
Importance of rapid stroke intervention : *"Time is Brain"*
Monitoring : Clinicians wish to have an image every fifteen minutes

Microwave Imaging

EMTensor company, Vienna, Austria.

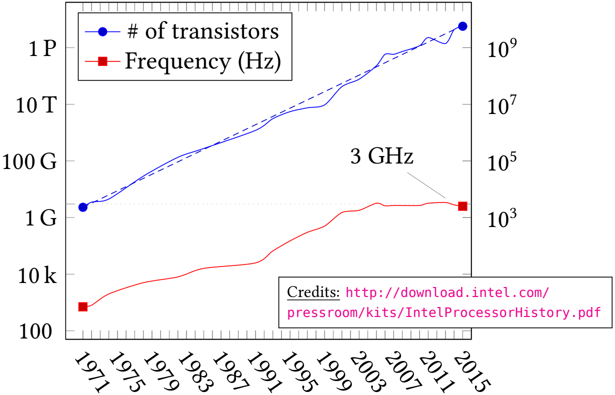
First prototype : cylindrical chamber with 5 rings of 32 antennas each.





New technologies : sensors miniaturization, high capacity mobile networks (4G-5G), massively parallel computations

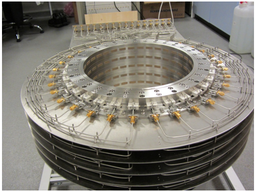
Need for massively parallel computing



Since year 2004 :

- CPU frequency stalls at 3 GHz due to the heat dissipation wall. The only way to improve the performance of computer is to go parallel

Let Ω be a dielectric material, linear, isotrope, non magnetic, dispersive and dissipative.
For each emitter j from 1 to $5x32$, the electric field $\mathbf{E}_j$ satisfies Maxwell system of equations :



$$(1) \quad \begin{cases} \nabla \times (\nabla \times \mathbf{E}_j) - \kappa \mathbf{E}_j = \mathbf{0}, & \text{in } \Omega, \\ \mathbf{E}_j \times \mathbf{n} = \mathbf{0}, & \text{on } \Gamma_{\text{metal}}, \\ (\nabla \times \mathbf{E}_j) \times \mathbf{n} + \mathrm{i}\beta\, \mathbf{n} \times \mathbf{E}_j \times \mathbf{n} = \mathbf{g}, & \text{on } \Gamma_j, \\ (\nabla \times \mathbf{E}_j) \times \mathbf{n} + \mathrm{i}\beta\, \mathbf{n} \times \mathbf{E}_j \times \mathbf{n} = \mathbf{0}, & \text{on } \Gamma_k \text{ with } k \neq j, \end{cases}$$

where $\kappa := \mu_0 \left(\omega^2 \varepsilon + \mathrm{i}\omega\sigma \right)$ where $\mu_0 > 0$, ω, β are known physical parameters, $\varepsilon > 0$ and $\sigma > 0$.
Imaging the brain is recovering the field $\kappa \in \mathbb{C}$ that matches measurements

$\boxed{\text{Challenge : Imaging at least every 15 minutes – Real time HPC}}$

Inverse problem

In order to image the brain, we recover the field $\kappa \in \mathbb{C}$ which minimizes the squared error between measurements and computed values :

$$J(\kappa) = \frac{1}{2} \sum_{i=1}^{160} \sum_{j \neq i} \left| \mathbb{S}_{ij}(\kappa) - \mathbb{S}_{ij}^{obs} \right|^2 dx,$$

where $\mathbb{S}_{ij}(\kappa)$ depend on solutions $\mathbf{E}_j(\kappa)$ to

$$\begin{cases} \nabla \times (\nabla \times \mathbf{E}_j) - \kappa \, \mathbf{E}_j = \mathbf{0}, & \text{in } \Omega, \\ \mathbf{E}_j \times \mathbf{n} = \mathbf{0}, & \text{on } \Gamma_{\text{metal}}, \\ (\nabla \times \mathbf{E}_j) \times \mathbf{n} + \mathrm{i}\beta \, \mathbf{n} \times \mathbf{E}_j \times \mathbf{n} = \mathbf{g}, & \text{on } \Gamma_j, \\ (\nabla \times \mathbf{E}_j) \times \mathbf{n} + \mathrm{i}\beta \, \mathbf{n} \times \mathbf{E}_j \times \mathbf{n} = \mathbf{0}, & \text{on } \Gamma_k \text{ with } k \neq j, \end{cases}$$

This non linear minimization problem is solved in parallel with a BFGS algorithm. Many computations

Challenge : Imaging at least every 15 minutes – Real time HPC

Parallel Software tools : HPDDM and FreeFem++

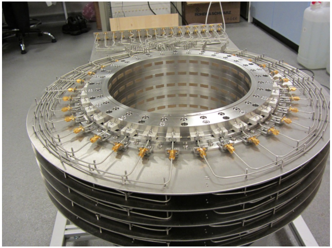
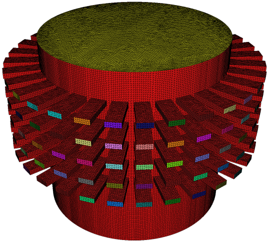
 

FIGURE: Antennas and mesh – interior diameter 28,5 cm

Two in-house open source libraries (LGPL) linked to many third-party libraries :

- HPDDM (High Performance Domain Decomposition Methods) for massively parallel computing
- FreeFem++(-mpi) for the parallel simulation of equations from physics by the finite element method (FEM).

$A\,u = f$? Panorama of parallel linear solvers

Multi-frontal sparse direct solver (I. Duff et al.)

MUMPS (J.Y. L'Excellent), SuperLU (Demmel, . . .), PastiX, UMFPACK, PARDISO (O. Schenk),

Iterative Methods

- Fixed point iteration : Jacobi, Gauss-Seidel, SSOR
- Krylov type methods : Conjuguate Gradient (Stiefel-Hestenes), GMRES (Y. Saad), QMR (R. Freund), MinRes, BiCGSTAB (van der Vorst)

"Hybrid Methods"

- Multigrid (A. Brandt, Ruge-Stüben, Falgout, McCormick, A. Ruhe, Y. Notay, . . .)
- Domain decomposition methods (O. Widlund, C. Farhat, J. Mandel, P.L. Lions,) are a naturally parallel compromise

Why iterative solvers ?

Limitations of direct solvers

In practice all direct solvers work well until a certain barrier :

- **two-dimensional problems** (10^6 unknowns)
- **three-dimensional problems** (10^5 unknowns).

Beyond, the factorization cannot be stored in memory any more.

To summarize :

- below a certain size, direct solvers are chosen.
- beyond the critical size, iterative solvers are needed.

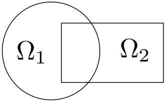
Linear Algebra from the End User point of view

Direct	DDM	Iterative
Cons : Memory Difficult to || Pros : Robustness	Pro : Flexible Naurally ||	Pros : Memory Easy to || Cons : Robustness
solve(MAT,RHS,SOL)	Some black box routines Some implementations of efficient DDM	solve(MAT,RHS,SOL)

Multigrid methods : very efficient but may lack robustness, not always applicable (Helmholtz type problems, complex systems) and difficult to parallelize.

The First Domain Decomposition Method

The original Schwarz Method (H.A. Schwarz, 1870)



$$
\begin{aligned}
-\Delta(u) &= f \quad \text{in } \Omega \\
u &= 0 \quad \text{on } \partial\Omega.
\end{aligned}
$$

Schwarz Method : $(u_1^n, u_2^n) \to (u_1^{n+1}, u_2^{n+1})$ with

$$
\begin{aligned}
-\Delta(u_1^{n+1}) &= f \quad \text{in } \Omega_1 \\
u_1^{n+1} &= 0 \text{ on } \partial\Omega_1 \cap \partial\Omega \\
u_1^{n+1} &= u_2^n \quad \text{on } \partial\Omega_1 \cap \overline{\Omega_2}.
\end{aligned}
\qquad
\begin{aligned}
-\Delta(u_2^{n+1}) &= f \quad \text{in } \Omega_2 \\
u_2^{n+1} &= 0 \text{ on } \partial\Omega_2 \cap \partial\Omega \\
u_2^{n+1} &= u_1^{n+1} \quad \text{on } \partial\Omega_2 \cap \overline{\Omega_1}.
\end{aligned}
$$

Parallel algorithm, converges but very slowly, overlapping subdomains only.
The parallel version is called **Jacobi Schwarz method (JSM)**.

Consider the discretized Poisson problem : $Au = f \in \mathbb{R}^n$.

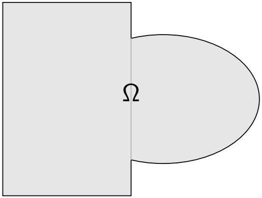
Given a decomposition of $[\![1;n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define :

- the restriction operator R_i from $\mathbb{R}^{[\![1;n]\!]}$ into $\mathbb{R}^{\mathcal{N}_i}$,
- R_i^T as the extension by 0 from $\mathbb{R}^{\mathcal{N}_i}$ into $\mathbb{R}^{[\![1;n]\!]}$.

$u^m \longrightarrow u^{m+1}$ by solving concurrently :

$$u_1^{m+1} = u_1^m + A_1^{-1} R_1 (f - Au^m) \qquad u_2^{m+1} = u_2^m + A_2^{-1} R_2 (f - Au^m)$$

where $u_i^m = R_i u^m$ and $A_i := R_i A R_i^T$.



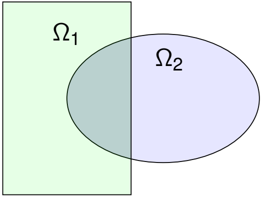
Consider the discretized Poisson problem : $Au = f \in \mathbb{R}^n$.
Given a decomposition of $[\![1;n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define :

- the restriction operator R_i from $\mathbb{R}^{[\![1;n]\!]}$ into $\mathbb{R}^{\mathcal{N}_i}$,
- R_i^T as the extension by 0 from $\mathbb{R}^{\mathcal{N}_i}$ into $\mathbb{R}^{[\![1;n]\!]}$.

$u^m \longrightarrow u^{m+1}$ by solving concurrently :

$$u_1^{m+1} = u_1^m + A_1^{-1} R_1 (f - Au^m) \qquad u_2^{m+1} = u_2^m + A_2^{-1} R_2 (f - Au^m)$$

where $u_i^m = R_i u^m$ and $A_i := R_i A R_i^T$.



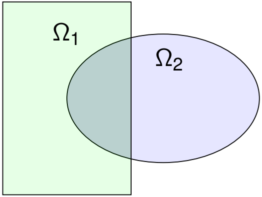
Consider the discretized Poisson problem : $Au = f \in \mathbb{R}^n$.
Given a decomposition of $[\![1;n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define :

- the restriction operator R_i from $\mathbb{R}^{[\![1;n]\!]}$ into $\mathbb{R}^{\mathcal{N}_i}$,
- R_i^T as the extension by 0 from $\mathbb{R}^{\mathcal{N}_i}$ into $\mathbb{R}^{[\![1;n]\!]}$.

$u^m \longrightarrow u^{m+1}$ by solving concurrently :

$$u_1^{m+1} = u_1^m + A_1^{-1} R_1 (f - Au^m) \qquad u_2^{m+1} = u_2^m + A_2^{-1} R_2 (f - Au^m)$$

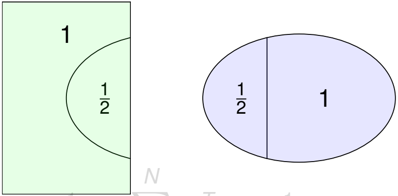
where $u_i^m = R_i u^m$ and $A_i := R_i A R_i^T$.



We have effectively divided, but we have yet to conquer.

Duplicated unknowns coupled via a *partition of unity* :

$$I = \sum_{i=1}^{N} R_i^T D_i R_i.$$



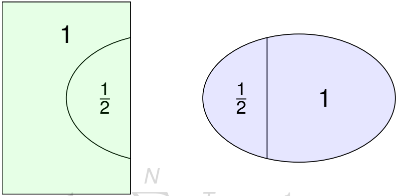
Then, $u^{m+1} = \sum_{i=1}^{N} R_i^T D_i u_i^{m+1}$. $M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i$.

RAS algorithm (Cai & Sarkis, 1999). Weighted Overlapping Block Jacobi method

We have effectively divided, but we have yet to conquer.

Duplicated unknowns coupled via a *partition of unity* :

$$I = \sum_{i=1}^{N} R_i^T D_i R_i.$$



Then, $u^{m+1} = \sum_{i=1}^{N} R_i^T D_i u_i^{m+1}$.

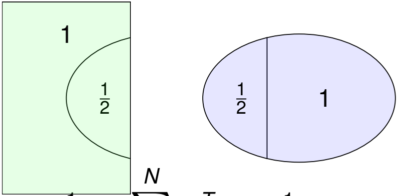
$$M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i.$$

RAS algorithm (Cai & Sarkis, 1999). Weighted Overlapping Block Jacobi method

We have effectively divided, but we have yet to conquer.

Duplicated unknowns coupled via a *partition of unity* :

$$I = \sum_{i=1}^{N} R_i^T D_i R_i.$$



Then, $u^{m+1} = \sum_{i=1}^{N} R_i^T D_i u_i^{m+1}$. $M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i$.

RAS algorithm (Cai & Sarkis, 1999). Weighted Overlapping Block Jacobi method

Algebraic formulation - RAS and ASM

Discrete Schwarz algorithm iterates on a pair of local functions (u_m^1, u_m^2)
RAS algorithm iterates on the global function u^m

Schwarz and RAS

Discretization of the classical Schwarz algorithm and the iterative RAS algorithm :

$$U^{n+1} = U^n + M_{RAS}^{-1} r^n \,, r^n := F - A\, U^n.$$

are equivalent

$$U^n = R_1^T D_1 U_1^n + R_2^T D_2 U_2^n \,.$$

(Efstathiou and Gander, 2002).

Operator M_{RAS}^{-1} is used as a preconditioner in Krylov methods for non symmetric problems.

Algebraic formulation - RAS and ASM

Discrete Schwarz algorithm iterates on a pair of local functions (u_m^1, u_m^2)
RAS algorithm iterates on the global function u^m

Schwarz and RAS

Discretization of the classical Schwarz algorithm and the iterative RAS algorithm :

$$U^{n+1} = U^n + M_{RAS}^{-1} r^n \,, r^n := F - A\, U^n .$$

are equivalent

$$U^n = R_1^T D_1 U_1^n + R_2^T D_2 U_2^n \,.$$

(Efstathiou and Gander, 2002).

Operator M_{RAS}^{-1} is used as a preconditioner in Krylov methods for non symmetric problems.

Many cores : Strong and Weak scalability

How to evaluate the efficiency of a domain decomposition ?

Strong scalability (Amdahl)

"How the solution time varies with the number of processors for a fixed *total* problem size"

Weak scalability (Gustafson)

"How the solution time varies with the number of processors for a fixed problem size *per processor*."

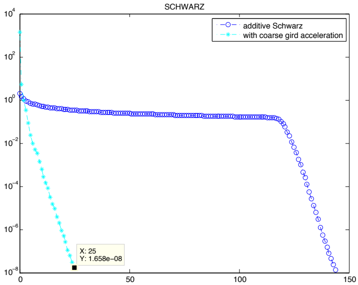
Not achieved with the one level method

Number of subdomains	8	16	32	64
ASM	18	35	66	128

The iteration number increases linearly with the number of subdomains in one direction.

Convergence curves- more subdomains

Plateaus appear in the convergence of the Krylov methods.



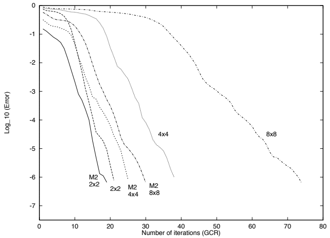


FIGURE: Decomposition into 64 subdomains and into $m \times m$ squares

Solution of a Poisson problem $-\Delta u = f$

Number of subdomains	2x2	4x4	8x8
Number of iterations	20	36	64

Adding a coarse space

One level methods are not scalable for steady state problems.

We add a coarse space correction (*aka* second level)

Let V_H be the coarse space and Z be a basis, $V_H = \text{span}\, Z$, writing $R_0 = Z^T$ we define the two level preconditioner as :

$$M^{-1}_{ASM,2} := R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i.$$

The Nicolaides approach (1987) is to use the kernel of the operator as a coarse space, this is the constant vectors, in local form this writes :

$$Z := (R_i^T D_i R_i \mathbf{1})_{1 \le i \le N}$$

where D_i are chosen so that we have a partition of unity :

$$\sum_{i=1}^{N} R_i^T D_i R_i = Id.$$

Theoretical convergence result

Theorem (Widlund, Dryija)

Let $M^{-1}_{ASM,2}$ be the two-level additive Schwarz method :

$$\kappa(M^{-1}_{ASM,2}A) \leq C\left(1+\frac{H}{\delta}\right)$$

where δ is the size of the overlap between the subdomains and H the subdomain size.

This does indeed work very well

Number of subdomains	8	16	32	64
ASM	18	35	66	128
ASM + Nicolaides	20	27	28	27

HPDDM Library (P. Jolivet and N.)

An implementation of several Domain Decomposition Methods

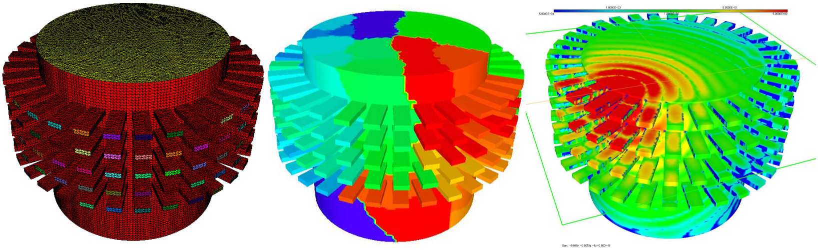
- One-and two-level Schwarz methods
- The Finite Element Tearing and Interconnecting (FETI) method
- Balancing Domain Decomposition (BDD) method
- Implements parallel algorithms : Domain Decomposition methods and Block solvers
- 2 billions unknowns in three dimension solved in 210 seconds on 8100 cores

Library

- Linked with graph partitioners (METIS & SCOTCH).
- Linked with BLAS & LAPACK.
- Linked with direct solvers (MUMPS, SuiteSparse, MKL PARDISO, PASTIX).
- Linked with eigenvalue solver (ARPACK).
- Interfaced with discretisation kernel FreeFem++ & FEEL++
- C++, C, Fortran and Python interface

Forward problem and Synthetic data

- Mesh with 2.3M degrees of freedom ;
- Domain decomposition methods with impedance interface conditions, twice as fast as Dirichlet interface conditions ;
- Parallel computing on 64 cores on SGI UV2000 at UPMC : 3s per emitter, 5 mn as a whole.

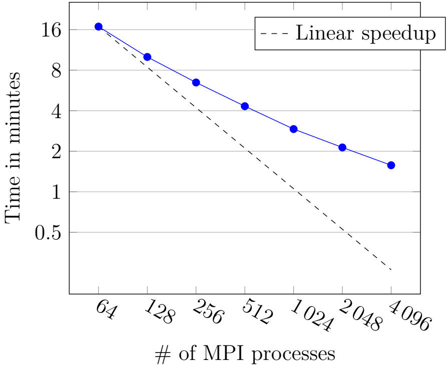


Imaging with massively parallel computations

High Performance Computers Turing (IDRIS, CNRS) and Curie (TGCC, CEA) via GENCI grants.

Several levels of parallelism :

- Sliced image reconstruction
- Cost function evaluation
- Domain Decomposition Methods



Proof of concept

Time to image : $<$ 2 minutes $<$ 15 minutes

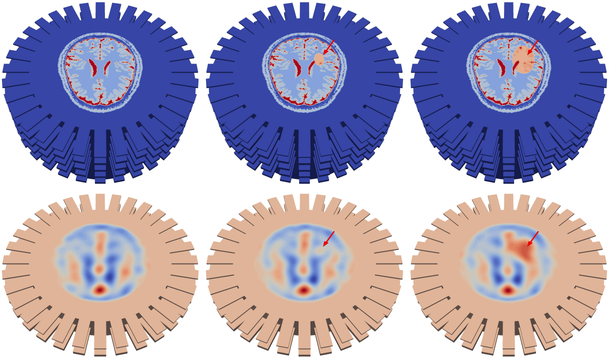


FIGURE: Evolution of a CVA : exact sections of a 3D image and its reconstructed image, 10% noise

Feasibility of a mobile imaging of CVA

Conclusion

- Medical imaging modality based on new technologies :
  - many sensors
  - mobile fast communication
  - and many cores computers
  - parallel flexible tool for scientific simulations
- With IoT (Internet of Things) many other applications can be envisioned e.g. Precision agriculture

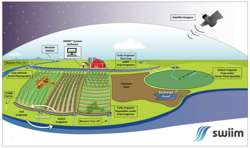


FIGURE: Precision agriculture

- Parallel computers are more and more readily available but software developments still lag behind

Thanks to (1/2)

Our employers

Our laboratories : IRIT (Toulouse), LJLL and Inria Team Alpines (Paris), JAD (Nice)

Funding based on projects

- ANR MEDIMAX 2013-2017 : led by Christian Pichot from Laboratoire d'Electronique, Antennes et Télécommunications (Nice). Collaboration between applied mathematics laboratories (LJLL and MAP5, Paris, JAD, Nice), electrical engineering teams (LEAT, Nice and informally EMTensor, Vienna)
- Computing hours (millions of) and technical support on large HPC supercomputers with hundreds of thousands of cores : Curie (CEA, BULL) and Turing (CNRS,IBM) via GENCI or PRACE calls

Institut des sciences du calcul et des données