

# Using the native mesh partitioning capabilities of Scotch 4.0 in a parallel industrial electro-magnetics code

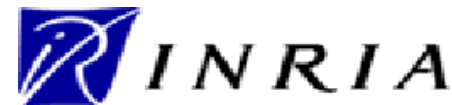
---

François PELLEGRINI

INRIA Futurs, project ScAIApplix

David GOUDIN

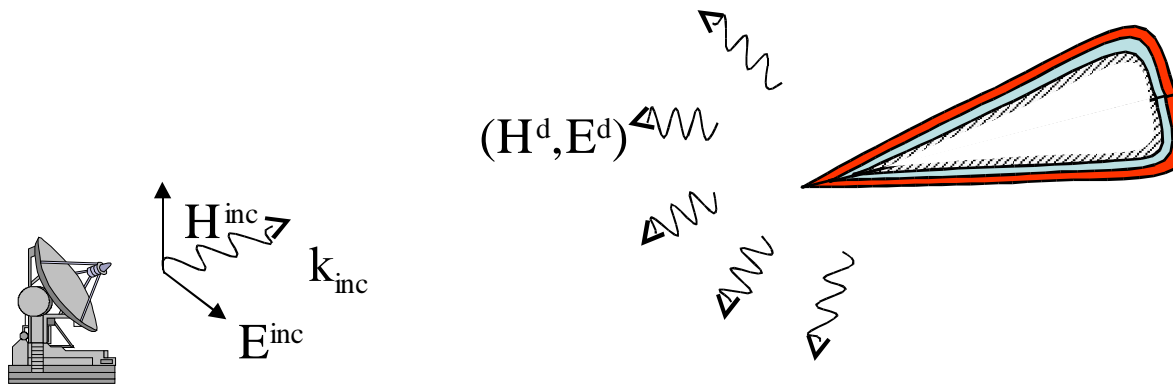
CEA / CESTA



# Problem to solve (1)

---

- Simulation of the electro-magnetic behavior of a 3D complex target in the frequency domain

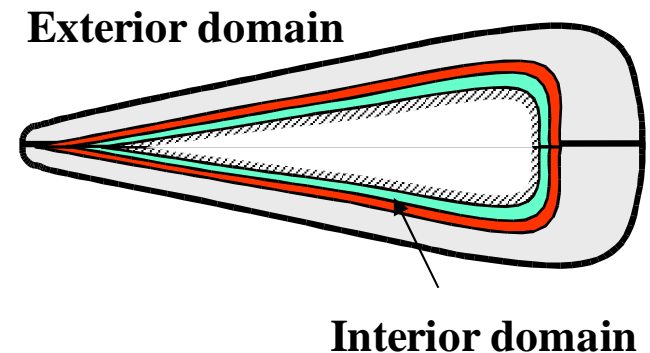


- Numerical resolution of Maxwell's equations in the free space

# Problem to solve (2)

---

- Hybrid method
  - Problem is partitioned into two subdomains
    - Exterior domain
      - Boundary Integral Equations method
      - Surfacic mesh, dense matrix
    - Interior domain
      - Partial Derivative Equations method
      - Volumic mesh, sparse matrix
- Large linear system to solve by direct methods
  - Dense part → ScaLAPack
  - Sparse part → Emilio



# Solving the sparse system in parallel

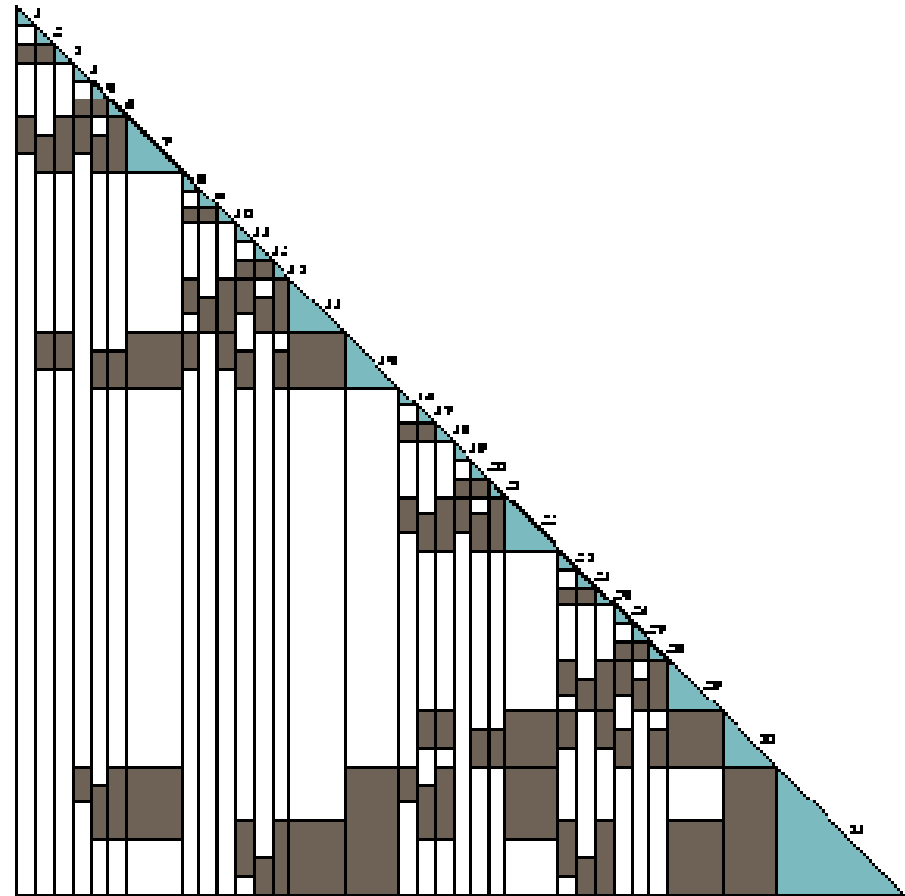
---

- By means of the `Emilio` software chain :
  - Mesh reordering : sequential (`Scotch`)
  - Block symbolic factorization : sequential (`Fax`)
  - Block distribution and mapping : sequential (`Blend`)
  - Mapping of elements : parallel (`Sacha`)
  - Matrix and RHS vector assembly
  - Setting of boundary conditions
  - Factorization and triangular resolution : parallel (`PaStix`)

(See PP04 MS43)

# Block symbolic factorization

- Performed based on the block ordering computed by Scotch
- $Q(G,P) \rightarrow Q(G,P)^* = Q(G^*,P)$
- Linear time and space complexities
- Block data structures
  - Only a few pointers
  - Use of BLAS3 primitives
- Not the main bottleneck...



# Graph block ordering

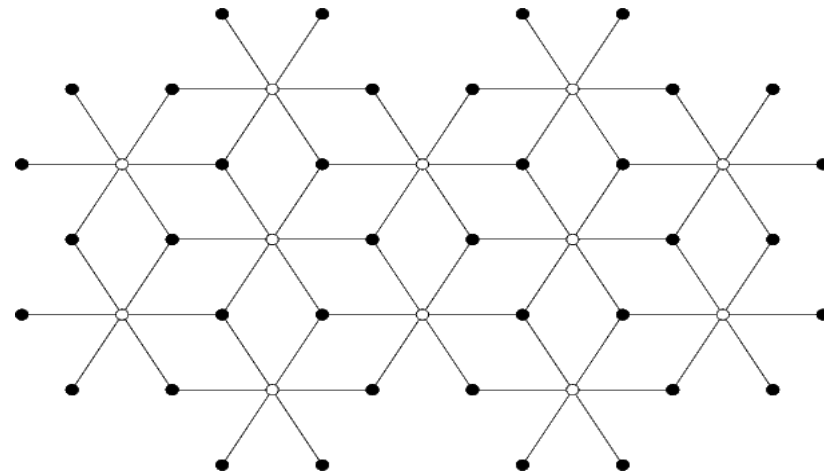
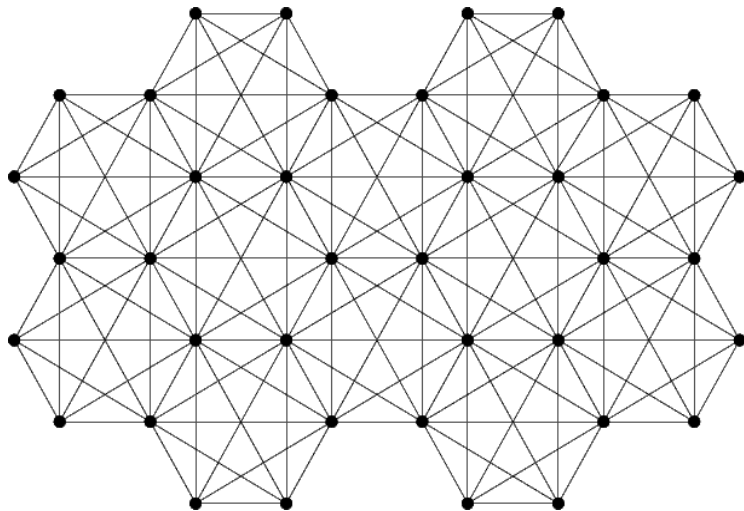
---

- By means of the `Scotch` software
- Originally based on graph heuristics
  - Tight coupling of Nested Dissection and Minimum Degree methods
    - Nested dissection → concurrency
    - Minimum degree → reduction of memory overhead
- All of the nodes of every element must be connected into cliques
  - Obligation to build a nodal graph in addition to the mesh
  - Huge memory overhead for 3D graphs

# Mesh block ordering

---

- Works directly on the mesh structure rather than on the graph structure
  - Bipartite graph : nodes and elements
  - Node-node adjacency computed dynamically from node-element and element-node adjacencies (hashing)



# Expected gains

---

- Let us assume :
  - $\delta$  : graph connectivity
  - $\zeta$  : number of nodes adjacent to a given element
  - $\eta$  : number of elements adjacent to a given node
- Then, for some classical mesh topologies :

<i>Mesh type</i>	$\delta$	$\zeta$	$\eta$	<i>Graph size</i>	<i>Mesh size</i>	<i>Ratio</i>
2D triangles	6	3	6	7 n	15 n	2.14
2D grid, 8 points	8	4	4	9 n	10 n	1.11
3D grid, 26 points	26	8	8	27 n	17 n	0.62
Hexagons	12	6	3	13 n	7.5 n	0.57

- Gains should be important for 3D meshes



# Experimental results

- Three representative test cases provided by CEA

<i>Mesh name</i>	$\delta$	$\zeta$	$\eta$	<i>Nodes</i>	<i>Elements</i>	<i>Mesh edges</i>	<i>Graph edges</i>	<i>Ratio</i>
altr4	12.5	6	3.8	26089	16638	99828	163038	0.75
chanel1m	13.1	6	4.0	80592	54899	329394	526860	0.77
conesphere	15.2	6	5.1	1055039	887428	5324568	8023236	0.80

- Ordering results (Power3-based workstation, 8Gb)

<i>Mesh name</i>	<i>Graph ordering</i>				<i>Mesh ordering</i>			
	NNZ	OPC	Mem	CPU	NNZ	OPC	Mem	CPU
altr4	2.03E+06	3.94E+08	10673	1.7	1.94E+06	3.86E+08	11268	4.0
chanel1m	8.95E+06	2.63E+09	36869	6.4	8.80E+06	2.72E+09	20050	15.8
conesphere	5.58E+08	1.81E+12	155883	122.2	6.01E+08	2.07E+12	52125	402.2

# Analysis of mesh ordering results

---

- Memory gains are very significant
  - One third of memory used for the 1M unknowns case with respect to graph ordering
- Quality is :
  - Better for smaller meshes
  - Becomes worse for larger meshes
    - Related to the quality of the mesh coarsening routine
- Time overhead is significant
  - 2.5 times more for medium case
  - Ratio increases for large graphs
    - Here again related to the quality of mesh coarsening

# Partial conclusion

---

- Validation of the approach in this industrial context
  - Orderings of meshes of more than 2M unknowns have been computed on workstations with 2Gb of memory
  - Can be successfully applied to other domains
- The bottleneck is now the meshing of the test cases
  - Available meshing software can hardly mesh above 3M unknowns
- The `Scotch 4.0` software distribution will soon be available as a LGPL'ed libre software package
  - Usable as a testbed for other partitioning and ordering methods

# Ongoing work within the Scotch project

---

- Design of more efficient mesh coarsening methods
- Design of native minimum degree methods for meshes (collaboration with P. Amestoy, ENSEEITH)
- Efficient coupling of nested dissection and minimum degree methods
  - Criteria for switching between them based on topology
- Mesh static mapping
  - Reduce number of neighbor processors per subdomain
- Parallel graph and mesh partitioning
  - Towards the billion of unknowns...