CEA's Parallel industrial codes in electromagnetics

Authors : Goudin David, Mandallena Michel, Mer-Nkonga Katherine, Pesqué Jean Jacques, Sesques Muriel Stupfel Bruno, **CEA/CESTA, BP 2, 33114 Le Barp, France**

Abstract

The CEA/CESTA has been working for many years on the simulation of the electromagnetic behavior of 3-dimensional complex targets with inhomogeneous media layers in the frequency domain. The main problems we have to handle with are first the fact we need a very good accuracy in order to reach very low level of Radar Cross Section, and second of all, we want to reach medium or even high frequencies, i.e. body sizes about 20 to 100 wavelengths.

The computational domain is clearly unbounded, one solution consist to truncate it and put an Absorbing Boundary Condition on the outer boundary, at this time the Limit Condition of Berenger seems to be the best but not accurate enough for our applications. We choose instead to take into account the radiation condition without any approximation but by a Boundary Integral Equation on the interface.

Another issue, is about discretization that can leads to solve very large problem' sizes, especially when the frequency became higher and higher. We need to have at least 10 discretization points by wavelength, with the body's sizes mentioned above, problems with several millions of Degrees of Freedom are common. If we want to go further in frequency with the constraints we have, we need to handle with very efficient numerical method as well as high performance computing.

Historically, all our simulation codes were based on standard numerical methods such Boundary Integral Equations and Partial Derivative Equations.

For example *Arlene* is a fully BIE code based on a classical Finite Element approximation of surface Integral Equations such as EFIE and CFIE formulations. All the interfaces between homogeneous materials are meshed by planar triangles and unknown electric and magnetic currents J and M are expanded using first order basis functions. This formulation leads to linear system with a full matrix which is complex non hermitian but symmetric. *Arlene* can compute electromagnetic scattering for three-dimensional complex structures which are composed of combinations of Perfectly Electric Conducting bodies (thin wires, thin metallic plates, thick bodies) and homogenous isotropic media. All connections between wires, patches and thick conducting bodies are possible. A wire object can even penetrate an interface between two dielectric layers. Discrete symmetries of the geometry can also be used to reduce the computation time and memory. *Arlene* has been running on own parallel Terascale machine composed of 2640 processors (HP/Compaq 660 SMP nodes, 4 processors EV68, Node ES45).

Even if this code deals with an anisotropic tensor impedance, a better way to take into account anisotropic media is to use an Hybrid PDE and BIE method as we did with the code *Arlas*.

We consider here a coupling between 2 sub-domains: interior and exterior which represent respectively the penetrable body, and the free space domain. The computational domain is solved by a PDE method (order 2 with the electric field as unknown) with a FE discretization, on the outer boundary a BIE is used to take into account the radiation condition. A strong coupling is realized between these 2 methods. The domain is discretized into tetraedric meshes inside the volume with the electric field E as unknown; on the outer boundary the triangular mesh is extract in order to solve the Integral Equation with the electric and magnetic currents J and M as unknowns. This formulation leads to solve a linear system with a matrix with a full part due to the BIE and a very sparse part from the PED.

Both of this code *Arlene* and *Arlas* used direct linear solver in order to compute the solution.

For symmetric complex and non hermitian matrices, no solver was available, thus we had to develop our own parallel Cholesky-Crout solver, in collaboration with IBM. This solver turns to be very efficient in terms of CPU time and scalability on the super computer we have. As an example we reached the 2 Teraflops with 1536 processors for the factorisation on a matrix with more than 250 000 unknowns.

For *Arlas*, the matrix is composed by a sparse part and a full part. The size of the full and the sparse part is respectively equal to the number of DoF on the outer boundary (electric and magnetic currents) and the number of Dof in the computational domain (electric field). We used a Schur complement method in order to solve this linear system, just by elimination of the electric field.

This resolution can be divided in 6 steps :

- sparse matrix assembly,
- factorisation of the sparse matrix,
- computation of the Schur's complement,
- full matrix assembly,
- factorization and resolution of the full linear system.

The 3 first steps are realized by a high performance parallel software : EMILIO (using the high performance sparse direct solver PaStix), which was developed in collaboration with the team INRIA ScAlApplix. This solver is efficient enough to perform the Cholesky-Crout factorization of our linear system with a reasonable elapse time, up to 3 millions of DoF. A new release of this parallel direct solver could be reach 10 millions of DoF. The step 4 to 5 are realized by the *Arlene'*s solver mentioned above.

As an example, Tab2 shows the CPU time versus the number of processors for the 3 first step on a case which represents 2.6 millions unknowns for the computational domain and 8000 unknowns on the outer boundary.

Tab2 : CPU time vs the number of processors

Task	64 Pe	128 Pe	256 Pe
Sparse matrix Assembly	210s	128s	116s
Sparse matrix factorization	1050s	840s	504s
Computation of the Schur complement	13100s	11800s	12088s

The drawback of this approach is the CPU time for the computation of the Schur complement. This time is due to the fact we have to perform as many resolution (up-and-down computation) as the number of DoF on the outer boundary. This is the prize to pay for having an exact and very accurate solution.

For our recent applications, this numerical approach tends to be inappropriate in terms of number of Degree of Freedom limits, especially because of the Schur complement method and also the direct solver for full matrices we used. The aims are both more than 10 millions DoF for the inner domain and 1 million of DoF for the free space domain on the coupling interface. In order to solve this kind of problem's size, we have combined, in a 3D code named *Odyssee* an original numerical method which is based on a Domain Decomposition Method, and a very efficient parallel algorithm.

The computational domain is partitioned into concentric sub-domains on the interface of which a Robin-type transmission condition are prescribed.

On the outer boundary terminating the computational domain, the radiation condition is taken into account by a new Boundary Integral formulation called EID. This formulation has been used because of its particular algebraic structure of the system which allows us to obtain convergence theorems for iterative algorithms, which is known to be difficult to get with classical Integral Equations. In practical, the EID formulation is efficient enough to reduce drastically the number of iterations during the iterative resolution. In order to improve the elapse time for the resolution of the outer problem, a parallel Multi Level Fast Multipole Algorithm has been developed.

The coupling between each sub-domain is realized by an iterative Gauss-Seidel algorithm. This numerical method, which is intrinsically a sequential algorithm, leads us to focus the whole parallelization effort on the resolution of each sub-domain. It's with this aim in view that each inner problem can be solved by either the parallel direct solver for sparse matrices PaStix or a classical parallel Conjugate Gradient.

However, in case of the resolution by the direct solver, because of the efficiency of the MLFMA and the problem's size between the inner and the outer problem, the optimal number of processors to perform properly each of them is quite different. The number of processors necessary to solve the outer problem could be 10 times less than the one to solve the inner problem. We used this unbalanced number of processors to implement a multilevel parallel algorithm in order to solve at the same time, the outer problem on more than one right hand side, which is useful for example to perform at once the 2 polarizations of an incident wave.

All these numerical methods have been implemented into the code *Odyssee* This code runs actually on our terascale machine. A lot of work has been done in order to perform all kind of simulation with this code.