Building a sparse matrix in coupled Finite Element/Electric Circuit for large scale Magnetic Field Modeling

E. Rodriguez[1], Y. Le Floch[1], B. Paya[2], O. Pateau[2]
[1]Cedrat, 15 chemin de Malacher, F-38420 Meylan
[2]EDF R&D, Avenue des Renardieres, F-77250 Moret sur Loing
E-mail: eric.rodriguez@cedrat.com

Abstract — Simulation of modern electromagnetic applications with Finite Element Analysis (FEA) requires to solve large sparse linear systems, that may reach up to millions of unknowns variables. Dealing with large matrix involves the use of sparse data structures. A transient matrix is necessary to gather information from the mesh or other equations in order to finally produce a matrix in optimized sparse format. Classical approaches to design sparse patterns are not efficient in large scale applications. In this paper, we focus and propose a generic approach for both building and ordering sparse matrix, in particular for coupled fem/electric circuit modeling.

I. INTRODUCTION

The capability of modern computer**s** allows scientists to study more challenging case studies. Dealing with complex geometr**ies** and finest mesh involves large amount of data manipulated in the solving process and thus requires the use of fast algorithms. Algorithm accuracy can be measured through its ability to scale and preserve performance when the amount of data increases. In Finite Element Analysis, it is necessary to solve linear systems A.x=b, usually by using iterative solvers. When storing and manipulating sparse matrices, it is useful and often necessary to use data structure**s** that take advantage of the sparse structure of the matrix. According to mesh information, a matrix is framed in a transient format, which is efficient for an incremental construction and for creating the final sparse matrix. List of lists (LIL) algorithm offers facilities and effectiveness in finite element case**s**. The request for modelling design**s** with soli**d** conductors connected to electric circuit**s** leads to the creation of strongly coupled formulation**s** FE-Electric Circuit introduced by [3] and [4]. The Electric circuit rarely exceeds thousand of equations. Sparsity property differs in lonely finite element problem and drastically affects the performance of the matrix building algorithm.

II. CLASSICAL APPROACH

Large linear systems are often solved by iterative **solvers** thanks to their low memory requirement**s**. Preconditioning technique**s** can improve the reliability and the efficiency of iterative methods. Reordering strategies are used to reduce fill-in in complete LU decomposition, it can also significantly improve the quality of incomplete LU factorizations. Thus, we include reordering pre-processing in our building **of the** sparse matrix algorithm Reverse Cuthill-McKee, (RCM) [1]. In FEA software**s**, finite elements and electric circuit components are often stored in a database, the transient matrix is built by questioning this database. The Matrix is framed with a reduced fill-in profile: reducing bandwidth reordering and finite element unknown variables first, in order to design a look-like down arrow shape. The classical approach STD_LIL (fig.1) consists **in** using a list of lists: a list of non zero column position**s** per row. For each finite element, we compute the pattern of the local stiffness matrix, and put non zero terms in the list of lists LIL.

1: For all finite elements
Form non-zero terms of the local stiffness matrix in a list of position (I,J)
=> Insert (I,J) in the LIL global matrix
2: Form the list of electric circuit and coupled FE/EC non zero term in (I,J)
=> Insert (I,J) in the LIL global matrix
3: Get permutation from RCM algorithm (FE unknowns in front)
4: Apply permutation in the LIL matrix
5: Create the CRS matrix (Compressed Row Storage)

Fig. 1. Standard algorithm STD_LIL

The set of problems used for the numerical experiment**s** is representative of linear systems solved in electromagnetic computation. In order to assess scalability, we compare the time of computation for various mesh refinements. In every case **the** STD_LIL algorithm scales poorly. Bad accuracy particularly appears in coupled finite element/electric circuit (FE/EC) applications. **The** matrix pattern has both sparse and dense parts. RCM algorithm and reordering task are costly in the dense part.

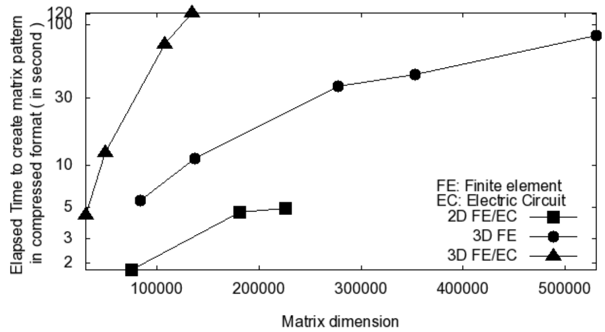

Fig. 2. Performance of STD_LIL algorithm

III. SUBMATRIX APPROACH

A. *Principle*

Using sorted LIL is the first most obvious way to optimize performance, this helps minimizing look up in searching, sorting and inserting tasks. The structure pattern suggests using a block approach which seems natural. The two different patterns have their own specific data structure**s**. An ordered LIL data structure is well adapted to the finite element pattern which has a native reduced bandwidth profile.

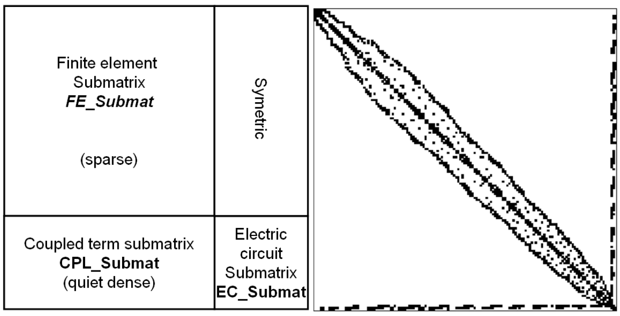

Fig. 2. Structure of coupled matrix

The down arrow shape needed in global matrix patters for an efficient incomplete factorization is preserved. Using blocks has the two main « divide and conquer » benefits . A natural data parallelism appears, and the size of **the** array manipulated decreases, thus leading to faster computation. A global LIL is necessary to gather at last sub-matrices, we neglect the memory requirement over-cost to focus on computation time.

1: Create the finite element ordered LIL submatrix: EF_Submat
2: Create the electric circuit ordered LIL submatrix: EC_Submat
3: Form FE/EC non zero list in **UNORDERED** array: CPL_Submat
4: Get permutation of submatrix FE_Submat and EC_Submat
5: Sort the **permuted** arrays of coupled submatrix CPL_Submat
6: Fill the LIL global matrix a) with **CPL_Submat, upper part**
b) with **FE_Submat**
c) with **EC_Submat**
d) with **CPL_Submat, lower part**
Fast Look up
7: Create the **CRS matrix** (Compressed Row Storage)

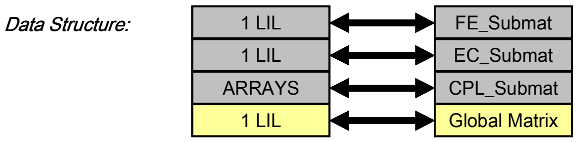

Fig. 3. SBM_LIL algorithm and data structure

As Cpl_Submat is excluded of the reordering process, unordered LIL is adapted to collect non zero positions and fill in the global LIL optimally. Step 5 improves significantly insertion of large array in sorted list. STD_LIL and SBM_LIL algorithms were implemented in FEA software Flux®, distributed by Cedrat. The two approaches provide the same matrix pattern. Scaling performances are shown in fig. 4. In 3D FE simulation, CPU time is reduced by 3, which is mainly due to the use of ordering in the LIL data structure. For FE/EC simulation, MTD_LIL is 40x faster and scalability is globally better for the whole problems set.

B. *Industrial case*

In order to improve the efficiency of electrical device, EDF R&D uses finite element analysis simulations to optimize their transverse flux heating devices, shown in Fig. 5.

The steady state AC magnetic 3D simulation leads to the solving of a linear system of 2,2 million of unknowns. About 300 millions of non zero values may be found in the matrix, and more than 50% of these are found in the CPL_Submat.

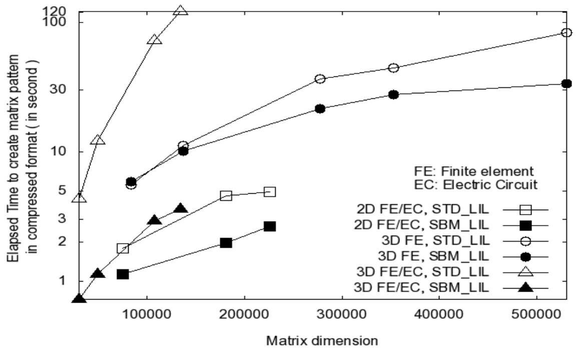

Fig. 4. SBM_LIL algorithm and data structure

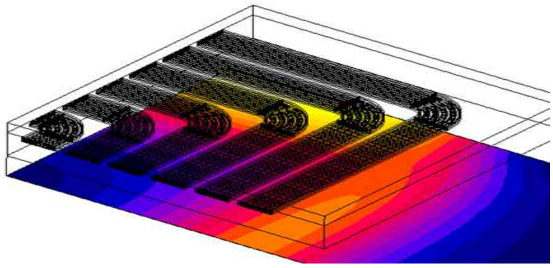

Fig. 5. Transverse flux heating device (courtesy of EDF)

This is due to fine mesh of coil conductor connected to electric circuit variables. The basic approach takes 2 full days to build the matrix, mainly due to the reordering task in electric circuit part of the global matrix. The new method presented is very fast , it takes around 3 minutes to compute the sparse matrix. Excluding CPL_Submat of reordering task and adapting optimization (step 5-6) leads to a decrease in time of lookup for data insertion, which is even more the case when dealing with huge data arrays.

C. *Conclusion and future works*

We propose a fast and flexible algorithm to build and reorder sparse matrix in finite element solving processes. We measure the impact of ordering strategy and provide a fast method for coupled FE/EC model which leads to important time saving when dealing with industrial applications. We should focus on minimizing memory requirements, and exploiting intrinsic task/data parallelism of the method on multicore architecture.

IV. References

[1] L E. Cuthill and J. McKee, "Reducing the bandwidth of sparse symmetric matrices," Proceedings of 24th national conference, pp. 157-172, New York, NY, USA, 1969.

[2] Y. Saad, *Iterative method for sparse linear systems*, 2nd ed., SIAM, 2003.

[3] G. Meunier, C. Guérin and Y. Le Floch, "Circuit coupled t0-f formulation with surface impedance," *IEEE Trans. Magn.*, vol. 44, no.6, pp. 730-733, 2008.

[4] G. Meunier, Y. Le Floch and C. Guérin, "A nonlinear circuit coupled t-t0-f formulation for solid conductors," *IEEE Trans. Magn.*, vol. 39, no.3, pp. 1729-1732, 2000.