# Applications of the FETI-DP-RBS-LNA Algorithm on Large Scale Problems with Localized Nonlinearities

Jun Sun[1], Pan Michaleris[2], Anshul Gupta[3], and Padma Raghavan[4]

[1] Department of Mechanical and Nuclear Engineering, 307 Reber Building,
Pennsylvania State University, University Park, PA 16802, USA.
Tel:(814)8650059, Email:junsun@psu.edu

[2] Department of Mechanical and Nuclear Engineering, 232 Reber Building,
Pennsylvania State University, University Park, PA 16802, USA.
Tel:(814)8637273, Fax:(814)8634848, Email:pxm32@psu.edu

[3] IBM T. J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY
10598, USA. Tel:(914)9451450, Fax:(914)9453434,
Email:anshul@watson.ibm.com

[4] Department of Computer Science and Engineering, 343K IST Building,
Pennsylvania State University, University Park, PA 16802, USA.
Tel:(814)8659233, Fax:(814)8653176, Email:raghavan@cse.psu.edu

**Summary.** Large scale computing is a well-known research topic since it is heavily desired by many science and engineering disciplines to simulate complex and sophisticated problems. However, due to the unprecedented amount of data and computations involved, it also poses challenges for current available numerical algorithms and computer hardware. In this paper, the Dual-Primal Finite Element Tearing and Interconnecting method (FETI-DP) is carefully investigated, and a reduced back-substitution (RBS) algorithm is proposed to accelerate the time consuming preconditioned conjugate gradient (PCG) iterations involved in the interface problems. Linear and nonlinear identification analysis (LNA) is also proposed for large scale problems with localized nonlinearities. This combined approach is named as the FETI-DP-RBS-LNA algorithm. Serial CPU time of this approach is measured and compared with a direct sparse solver and the standard FETI-DP method on a welding problem. Parallelism of the FETI-DP-RBS-LNA algorithm is also implemented by using MPI and the performance is reported. The empirical results demonstrate the effectiveness of the proposed computational approach for welding applications, which is representative of a large class of three dimensional linear-nonlinear large scale problems with localized nonlinearities.

## 1 Introduction

Large scale finite element analysis is an important research area due to its wide applicability in modeling and simulating complicated scientific and engineering applications, such as structural mechanics, heat transfer, and biomechanics. For realistic and sophisticated models, high density meshes are required to capture the underlying physics in areas that are of particular interest or with complex geometry or loading. Accordingly, the total degrees of freedom in systems discritized by finite element method may easily exceed millions, and it poses many computational challenges for current available numerical algorithms as well as computer hardware.

Extensive research has been conducted to develop efficient and reliable numerical methods which have the capabilities to solve large scale systems arising from various disciplines. Two well-known approaches in this field are direct and iterative methods. Direct sparse solvers are recognized as robust and efficient choices for most of the applications, and they are widely employed in many commercial finite element softwares. However, the high memory demands and the not-so-well parallel scalability of direct sparse solvers restrict its applications to large scale problems [1]. Traditional iterative solvers are excellent from the memory point of view. However, they are problem dependent and the convergence is not guaranteed. For complex ill-conditioned engineering problems, they are not as reliable as direct sparse solvers.

Several novel approaches, such as Domain Decomposition (DD) methods and adaptive meshing methods[2, 3], have also been studied extensively for their possible applications to solve large scale systems. DD methods are based on the native divide and conquer concept, they partition the physical domain into subdomains with either overlapping or non-overlapping interfaces. Coarse-grain parallel processing can then be applied to the computations of these subdomains to reduce overall analysis time. Adaptive meshing refines or coarsens the meshes in different regions of the model during the analysis based on their corresponding resolution requirements. Therefore, this approach is capable to reduce the computational costs while still maintain the quality of the solution.

The objective of this paper is to present the FETI-DP-RBS-LNA algorithm [4] and to investigate its serial and parallel performance for large scale problems with localized nonlinearity. The FETI-DP-RBS-LNA algorithm is based on one type of DD methods, the Dual-Primal Finite Element Tearing and Interconnecting method (FETI-DP)[5, 6]. Reduced Back-Substitution (RBS) algorithm is proposed to accelerate costing local back-substitutions, and Linear and Nonlinear Analysis (LNA) is introduced to reduce unnecessary refactorizations of linear subdomains in the analysis. The distributed version of this algorithm is implemented with Message Passing Interface (MPI) and the performance is measured on a distributed PC cluster for a welding mechanical analysis problem with one million degrees of freedom.

## 2 Review of The FETI-DP-RBS-LNA Algorithm

### 2.1 The FETI-DP Algorithm

FETI-DP can be viewed as a combination of direct and iterative methods. Based on the underlining divide and conquer concept, the physical domain is divided into subdomains with non-overlapping interfaces. The related nodes after finite element discretization can be classified into three groups based on their locations, and they are marked as corner nodes, non-corner interface nodes and internal nodes in Figure 1, respectively. More details of FETI-DP can be found in Ref [4, 5, 6].

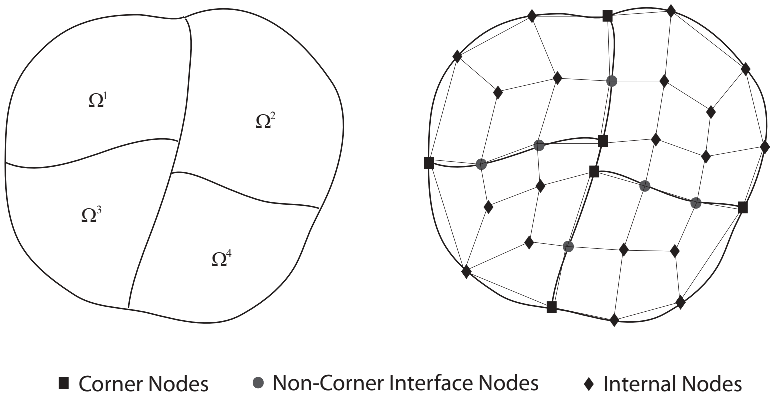

**Fig. 1.** Subdomains with non-overlapping interfaces, their meshes and nodes classification

Through the similar concepts of super elements and substructures, the high level interface problem can be first formulated and solved by an iterative Preconditioned Conjugate Gradient (PCG) method. Once the interface solution is available, corner information can be further solved. After that, all the low level subdomains are independent and can be solved by direct sparse solvers in a parallel fashion. These procedures are shown in Figure 2.

### 2.2 Reduced Back-Substitution Algorithm

Based on the CPU statistics in Ref [7] and the welding simulation problem in this paper, the PCG iterations for large interface problems are found to be the time consuming part in the FETI family algorithms. Within the PCG costs, a high percentage (around 64.3% for the mechanical analysis of the welding problem in this paper) of the CPU time is actually consumed by the local back-substitutions inside the PCG iterations. Therefore, the reduction

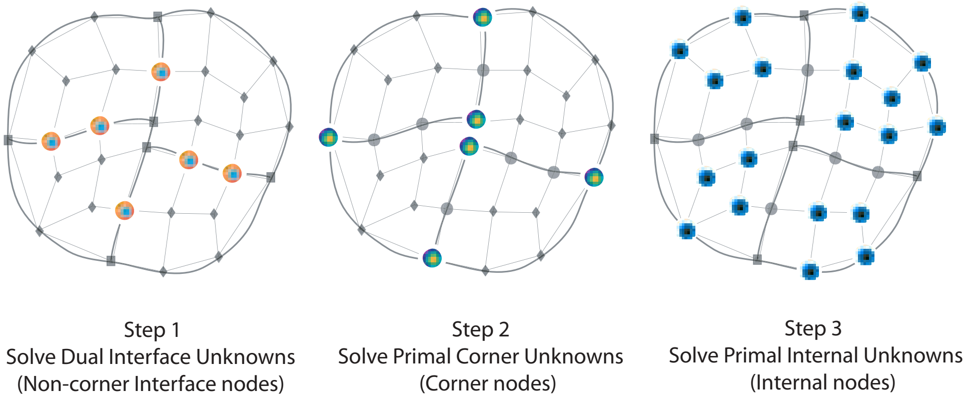
Step 1
Solve Dual Interface Unknowns
(Non-corner Interface nodes)
Step 2
Solve Primal Corner Unknowns
(Corner nodes)
Step 3
Solve Primal Internal Unknowns
(Internal nodes)

**Fig. 2.** Solution Scheme of FETI-DP

of computations in local back-substitutions will greatly improve the overall performance of the FETI-DP algorithm.

During each PCG iteration, the most time consuming steps are calculating the following two matrix-vector multiplications listed in Equation (1). Each multiplication has several back-substitutions involved.

$$(F_{I_{rr}} + F_{I_{rc}} {K^{*}_{cc}}^{-1} {F_{I_{rc}}}^{T}) \cdot \lambda \qquad \text{and} \qquad {F^{D}_{I_{rr}}}^{-1} \cdot \lambda \tag{1}$$

Taking one sub-step from the first multiplication $F_{I_{rr}} \cdot \lambda$ as an example, after substituting the detailed expression of $F_{I_{rr}}$ [5, 6], it yields the following equation:

$$F_{I_{rr}} \cdot \lambda = \sum_{s=1}^{n_s} B^{s}_{r} {K^{s}_{rr}}^{-1} {B^{s}_{r}}^{T} \lambda \tag{2}$$

In the FETI-DP algorithm, ${B^{s}_{r}}^{T}$ is first applied on $\lambda$ through scatter operations to get ${B^{s}_{r}}^{T}\lambda$, then ${K^{s}_{rr}}^{-1}({B^{s}_{r}}^{T}\lambda)$ is solved as a whole through the back-substitution at the subdomain level, where ${K^{s}_{rr}}^{-1}$ is the inverse of subdomain matrix which has already been factorized with its factorized information stored. Finally, $B^{s}_{r}$ is applied on the solution vector ${K^{s}_{rr}}^{-1}({B^{s}_{r}}^{T}\lambda)$ through gather operations to form $B^{s}_{r}({K^{s}_{rr}}^{-1}({B^{s}_{r}}^{T}\lambda))$ and summed over all the subdomains. The reason this process requires much computational time lies in the relatively large number of equations in each subdomain, as the back-substitution is actually performed on each subdomain internal and non-corner interface degrees of freedom (equations). The left part graph of Figure 3 shows the nodes involved in this standard back-substitution.

${B^{s}_{r}}^{T}$ and $B^{s}_{r}$ connect subdomain level information to global domain information through scatter and gather operations. If written in matrix format, their representations are sparse matrices. Based on the analysis in Ref [4], assuming the number of equations corresponding to non-corner interface degrees of freedom is $m$, and these equations are numbered last. Only the last

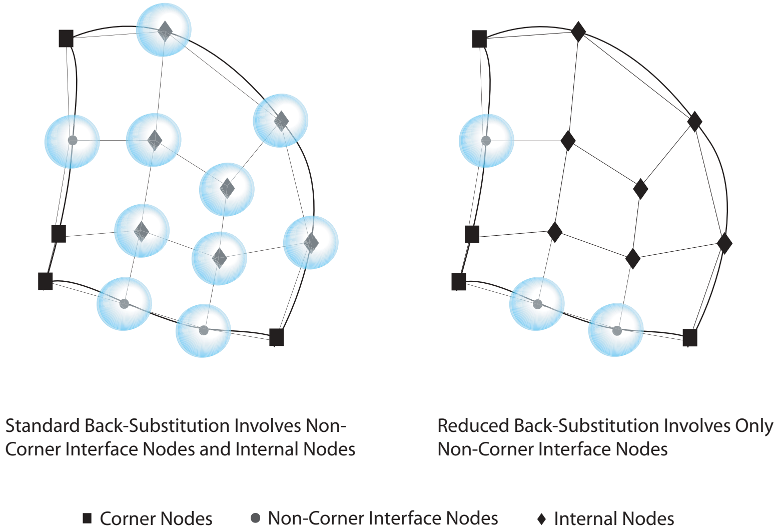
Standard Back-Substitution Involves Non-Corner Interface Nodes and Internal Nodes
Reduced Back-Substitution Involves Only Non-Corner Interface Nodes
■ Corner Nodes
● Non-Corner Interface Nodes
♦ Internal Nodes

**Fig. 3.** Standard Back-Substitution and Reduced Back-Substitution for Subdomain $\Omega^2$ in Figure 1

$m$ components from $\lambda$ are required as the input for the back-substitutions in Equation (2) since $B_r^s$ zeros the rest components, and only the last $m$ components from the back-substitution result ${K_{rr}^s}^{-1} {B_r^s}^T \lambda$ are required as the output due to the same reason. Thus the back-substitution is actually performed on the last $m$ equations. $m$ is a much smaller number compared to the sum of subdomain internal degrees of freedom and non-corner interface degrees of freedom. Therefore, much time can be saved based on this reduced back-substitution (RBS). The nodes involved in this RBS algorithm are shown in the right part of Figure 3. Compared to standard back-substitution, many internal nodes are not necessary to be included anymore.

It must be mentioned that the proposed reduced back-substitutions will affect the ordering scheme since it poses the restriction to re-number the related equations to the end of the entire equations. This re-numbering introduces additional time costs in the numeric factorization stage compared to the situation with a good ordering scheme, such as the nest-dissection scheme. More detailed discussion on this issue is in Ref [4].

### 2.3 Linear-Nonlinear Analysis

Linear-nonlinear analysis (LNA) is a well-known and efficient strategy to solve problems with localized nonlinearity. It exploits information about which subdomain remains linear during a nonlinear analysis. Therefore, repeated factorizations of linear subdomains can be avoided and computation costs can be saved. More implementation details on LNA can also be found in Ref [4].

## 3 Serial and Distributed Performance Results

### 3.1 Software and Hardware

The software and hardware implementation for the serial performance measurement is shown in Ref [4]. The standard MPICH libarary has been implemented in the in-house code for distributed computing. The distributed computing simulations are performed on the Penn State LION-XM cluster, which consists 168 computing nodes, and each node has 2 Intel Xeon (3.2 GHz) Processors and 4 GB memory.

### 3.2 16-Subdomain Hollow Beam Model and Simulation Information

A 16-Subdomain hollow beam model is chosen to be the large scale welding problem for performance measurements in this paper. The model and welding information can be found in [4]. The total number of Hex20 element in this model is 65664, and the total number of equations is 1007634. The number of interface equations is 8460 and the number of corner equations is 174.

### 3.3 Serial Performance Results

| CPU Time (s) | Serial Direct | FETI-DP | FETI-DP | FETI-DP | FETI-DP |
|---|---|---|---|---|---|
| | Sparse Solver | | RBS | LNA | RBS & LNA |
| IO & SF | 42.11 | 81.45 | 103.58 | 80.99 | 103.58 |
| NF | 47262.12 | 26525.69 | 40601.13 | 1849.01 | 2582.91 |
| BS | 1273.22 | — | — | — | — |
| PCG (LBS) | — | 58759.03 | 8879.37 | 58335.77 | 8900.07 |
| | — | (54880.92) | (5083.20) | (54497.29) | (5110.52) |
| TOTAL | 48577.45 | 85366.17 | 49584.08 | 60265.77 | 11586.56 |

**Table 1.** Mechanical Analysis Serial Performance, First 50 Time Increments

The serial CPU costs of the IBM Watson direct sparse solver, FETI-DP, FETI-DP-RBS, FETI-DP-LNA and FETI-DP-RBS-LNA in the mechanical analysis are measured and compared in Table 1, where IO stands for solver initialization and ordering, SF is symbolic factorization, NF is numeric factorization, BS is back-substitution, PCG is Preconditioned Conjugate Gradient iterations, LBS is local back-substitution in PCG, LNA is Linear-Nonlinear Analysis, and RBS is Reduced Back-Substitution. Detailed analysis of serial CPU time is listed in Ref [4].

| Wallclock Time (s) | UNISYS, 1 Processor | LION-XM, 16 Nodes | SpeedUp |
|---|---|---|---|
| | (16 Subdomains) | (1 Subdomain Per Node) | |
| NF | 288.53 | 20.07-30.72 | 9.4 |
| PCG | 54.64 | 6.51 | 8.4 |

**Table 2.** Mechanical Analysis Distributed Performance and SpeedUp, First Iteration

### 3.4 Distributed Performance Results

Distributed computing performance results are measured for the numeric factorization and PCG iterations during the first iteration, as shown in Table 2. 16 computing nodes of the LION-XM cluster are used in the simulation and each computing node contains one subdomain.

The subdomain level computations, such as, forming the subdomain stiffness matrices, local numeric factorizations, local back-substitutions and residual computations are all performed on each individual computing node in a parallel fashion. MPI is mainly implemented to gather and broadcast the intermediate results during the procedure of solving the interface problem by the PCG method.

The speedup gained during numeric factorization is 9.4. Perfect scalability is not achieved due to the fact that the number of interface DOFs of each subdomain is different. Therefore, the computational cost of each subdomain is also not the same. Some subdomains have large interfaces and require more time to be factorized. The MPI wallclock time is measured based on the longest factorization time.

The speedup gained during PCG iterations is 8.4. In the total 6.51s wallclock time, around 2.6s is spent on inter-processor communications to gather and broadcast the intermediate solution results during the interface solves. Therefore, from the computational point of view, the numerical scalability is very good and higher speedup can be expected when high-speed network interconnect is implemented.

## 4 Conclusion and Future Work

In this paper, a fast implementation of the FETI-DP algorithm: the FETI-DP-RBS-LNA algorithm is proposed for solving large scale problems with localized nonlinearity. Serial performance of the FETI-DP-RBS-LNA algorithm is tested to give a correct estimation of floating point performance. Distributed performance is also evaluated for the first iteration to measure the speedup gained from distributed computing. The future work will be to continue the investigation of the distributed performance of the FETI-DP-RBS-LNA algorithm when linear nonlinear analysis is applied.

## 5 Acknowledgments

The authors would like to acknowledge the funding from Office of Naval Research, award number N00014-00-1-0645 and ONR program manager Julie Christodoulou. The authors would also like to acknowledge Unisys, Inc. and Pennsylvania State University for providing the Unisys ES7000 systems; Jeff Nucciarone from Academic Services and Emerging Technologies, Pennsylvania State University, for his help and suggestions during the execution of the simulation on the Unisys system.

## References

1. R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM journal on numerical analysis*, 16:346–358, 1979.
2. N.S. Prasad and T. K. Sankaranrayanan. Estimation of residual stresses in weldments using adaptive grids. *Computers and Structures*, 60(6):1037–1045, 1996.
3. H. Runnemalm and S. Hyun. Three-dimensional welding analysis using an adaptive mesh scheme. *Computer Methods in Applied Mechanics and Engineering*, 189:515–523, 2000.
4. Jun Sun, Pan Michaleris, Anshul Gupta, and Padma Raghavan. A Fast Implementation of the FETI-DP Method: FETI-DP-RBS-LNA and Applications on Large Scale Problems with Localized Nonlinearitie. *International Journal for Numerical Methods in Engineering*, To appear.
5. C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numerical Linear Algebra with Applications*, 7:687–714, 2000.
6. C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method - part I: A faster alternative to the two-level FETI method. *International Journal for Numerical Methods in Engineering*, 50:1523–1544, 2001.
7. C. Farhat, K. Pierson, and M. Lesoinne. The second generation of FETI methods and their application to the parallel solution of large-scale linear and geometrically nonlinear structural analysis problems. *Computer Methods in Applied Mechanics and Engineering*, 184:333–374, 2000.