Procedia Computer Science

Volume 51, 2015, Pages 2744–2748

ICCS 2015 International Conference On Computational Science

Telescopic hybrid fast solver for 3D elliptic problems with point singularities

Anna Paszyńska[a], Konrad Jopek[b], Krzysztof Banaś[b], Maciej Paszyński[b, 1],

Piotr Gurgul[b], Andrew Lenerth[c], Donald Nguyen[c], Keshav Pingali[c],

Lisandro Dalcin[d], Victor Calo[d,e,f]

[a]*Jagiellonian University, Krakow, Poland*

[b]*AGH University of Science and Technology, Krakow, Poland*

[c]*Institute for Computational and Engineering Sciences, (ICES), TX, USA*

[d]*Center for Numerical Porous Media*

King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

[e]*Earth Science & Engineering and Center for Numerical Porous Media, (KAUST)*

[f]*Applied Mathematics & Computational Science, (KAUST)*

Abstract

This paper describes a telescopic solver for two dimensional *h* adaptive grids with point singularities. The input for the telescopic solver is an h refined two dimensional computational mesh with rectangular finite elements. The candidates for point singularities are first localized over the mesh by using a greedy algorithm. Having the candidates for point singularities, we execute either a direct solver, that performs multiple refinements towards selected point singularities and executes a parallel direct solver algorithm which has logarithmic cost with respect to refinement level. The direct solvers executed over each candidate for point singularity return local Schur complement matrices that can be merged together and submitted to iterative solver. In this paper we utilize a parallel multi-thread GALOIS solver as a direct solver. We use Incomplete LU Preconditioned Conjugated Gradients (ILUPCG) as an iterative solver. We also show that elimination of point singularities from the refined mesh reduces significantly the number of iterations to be performed by the ILUPCG iterative solver.

Keywords: hybrid solver, multi-frontal solver, h adaptive finite element method, ILUPCG, GALOS

1 Corresponding author. Tel.: +48-12-633-9406; fax: +48-12-633-9406. *E-mail address*: maciej.paszynski@agh.edu.pl.

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015

doi:10.1016/j.procs.2015.05.415

1 Introduction

We use the Finite Element Method (FEM) to solve elliptic problems on arbitrary domain. The classical adaptive algorithm for mesh refinement has been proposed by Demkowicz and co-workers, see [6]. These methods are based on the theoretical observations made by Babuska [1], that careful selection of both *h* and *p* refinements results in the exponential convergence of the numerical error with respect to the number of degrees of freedom, and single *h* or *p* refinements result in algebraic convergence. The theoretical results have been confirmed by the automatic algorithms constructed by Demkowicz [6]. These algorithms construct a sequence of coarse and fine grids. Each fine grid is obtained by performing global *hp* refinement over the corresponding coarse mesh (all elements are broken into four and the polynomial order is increased by one in each element). The fine mesh solution is utilized as the reference solution in order to estimate the numerical error over coarse mesh elements, as well as to select the optimal refinements over the coarse mesh elements where the numerical error is high. The generated sequence of *hp* refined grids indeed delivers exponential convergence of the numerical error with respect to the mesh size. Many authors followed the approach originated by Demkowicz and implemented their own variations of the adaptive algorithms. First, there is a possibility of utilizing different error estimators defined for elliptic [1], parabolic [8,3] or multi-physics problems [13]. In [12], the authors employ modern *h* and *hp* adaptation algorithms for the Girkmann problem. The general problem with adaptive algorithms is need to perform a sequence of global problem solutions over the sequence of grids generated by adaptive algorithms. In general these solutions must be performed by direct solvers [4], since the generated linear systems are not well conditioned, due to presence of elongated elements and the varying polynomial orders of approximation [15]. For the direct solution, the state-of-the-art multi-frontal solver algorithms are usually used [7, 8], and it is more computationally expensive than generation of element frontal matrices [5, 11, 17]. Recently, it has been shown that for the point singularities, it is possible to solve the problem in a linear cost [10]. Straight forward conclusion is that it is also possible to compute the Schur complemet of the part of the mesh with point singularity with linear computational cost. This is the main idea of this paper: to compute Schur complements of local point singularities and later to collect them and submit together to an iterative ILUPCG solver. This idea follows the generalization of the static condensation proposed in [2] to point singularities. A hybrid direct / iterative solver algorithm has been also used in [9]. The elimination is cut at some level, and the remaining Schur complements are submitted to an iterative solver.

2 Algorithm

In this paper we propose a hybrid telescopic solver, performing fast local refinements towards all possible candidates for point singularities, eliminating these singularities using GALOIS solver [16], and submitting the resulting Schur complement matrices to the ILUPCG iterative solver from the SLATEC library [14]. We show that local elimination of point singularities improves the conditioning of the linear system, and the number of iterations performed by the iterative solver remains constant, independently of the number of refinement levels performed. The example of computational grids with point singularities are presented in Figure 1. They result either from point heat sources or from three different material data meeting at the singular points. After having all the candidates for point singularities selected, we execute a hybrid algorithm that can be summarized in the following way: (1) execute multi-frontal solver parallel algorithm that for each point singularity, (2) perform static condensation for other initial mesh elements, and (3) collect resulting Schur complements and call ILUPCG solver.

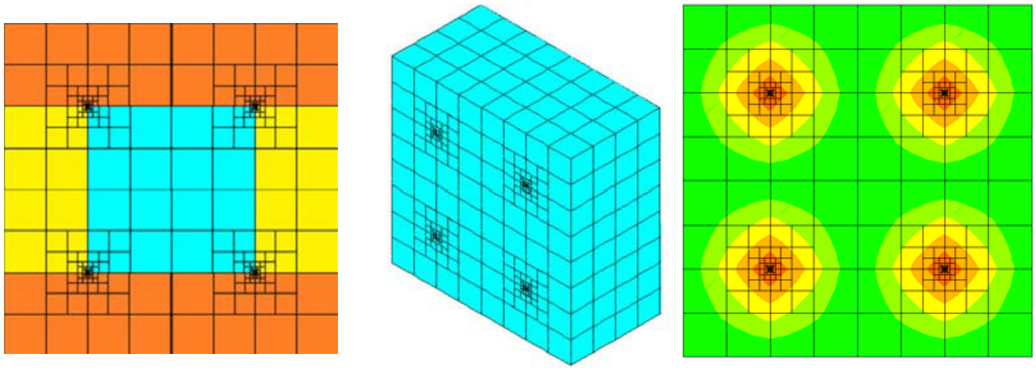

Figure 1. Left panel: Three different material data meeting at the point **Middle panel:** Three dimensional mesh with point singularities enforced by heat sources. **Right panel:** Two dimensional cross-section with heat sources.

3 Numerical results

In this section we report the execution time of the parallel multi-thread GALOIS solver, for elimination of point singularities, for three dimensional problems and the number of iterations of the ILUPCG algorithm.

All the experiments concerns the heat transfer problem, either with different material data or with uniform material but with many point heat sources. We utilize our graph-grammar based multi-frontal solver implemented in GALOIS environment [16]. The GALOIS solver execution is followed by running the ILUPCG algorithm from the SLATEC library [14]. We start by comparing the number of iterations of the ILUPCG solver executed using the hybrid and standard algorithm, where the static condensation is followed by execution of the ILUPCG solver for the entire mesh. The iterations for three dimensional grids with 4x4x4=64 singularities, for quadratic and cubic polynomials are presented on two panels in Figure 2. We can read from these experiments, that for the hybrid algorithm the number of iterations is constant, and it does not depend on the number of refinement levels.

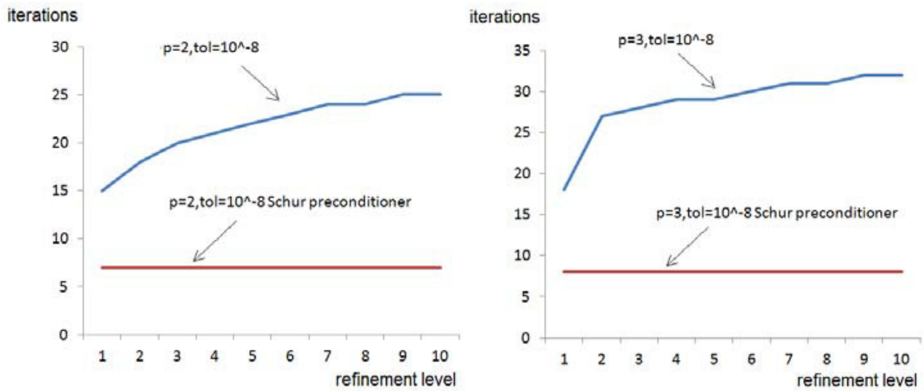

Figure 2. Left panel: Comparison on the number of iterations of ILUPCG solver for three dimensional mesh with quadratic polynomials and 4x4x4 singularities, for hybrid and standard algorithm. **Right panel:** Comparison on the number of iterations of ILUPCG solver for three dimensional mesh with cubic polynomials and 4x4x4 singularities, for hybrid and standard algorithm

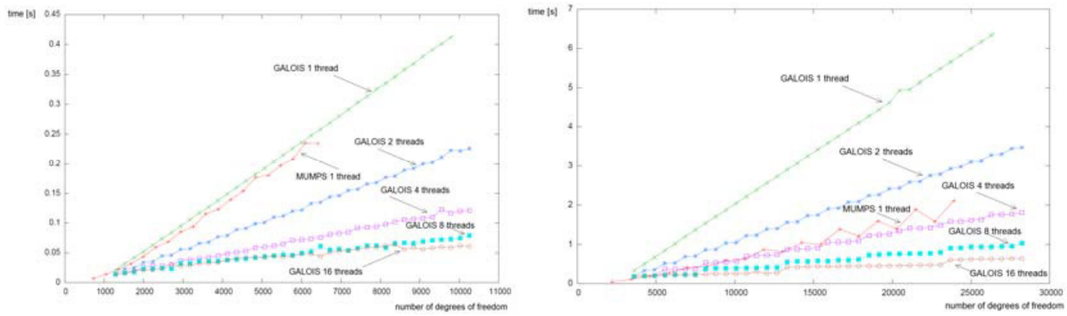

Figure 3. Comparison with serial MUMPS. **Left panel:** Execution time of the GALOIS solver executed over one 3D point singularity with quadratic polynomials, for different number of threads. **Right panel:** Execution time of the GALOIS solver over one 3D point singularity with cubics, for different number of threads.

In Figure 3 we present the execution times for the multi-thread CPU GALOIS solver [16,21] used for elimination of one point singularity over 3D grid with quadratic polynomials, for increasing number of refinement level resulting in increasing number of degrees of freedom, executed in either single core or multi-core mode. We also present the comparison with 1 core MUMPS. We can see that our 1 core GALOIS has similar execution time than 1 core MUMPS solver. We repeat the GALOIS experiments for cubic polynomials, and comparison with MUMPS for one point singularity over 3D grid. This time we need 4 cores of GALOIS solver to be as fast as 1 core MUMPS solver over 3D point singularity with cubic polynomials. All the GALOIS experiments executed on GILBERT a shared memory machine from the Institute for Computational Engineering and Sciences (ICES). The machine has four Intel(R) Xeon(R) CPU E7-4860 with 2.27GHz, each one with 10 cores and additional 10 hyperthreading cores. The total available memory was 128 GB.

Conclusions

In this paper we proposed an algorithm for identification of areas with point singularities over the refined computational mesh. We proposed a telescopic hybrid solver. We perform a selected number of *h* refinements in these areas, and executed multi-thread GALOIS solver for fast elimination of point singularities. The resulting Schur complement matrices are submitted to iterative ILUPCG solver. We show that processing of local point singularities with fast direct solvers allows to improve the conditioning of the linear system and keeps the number of iterations of ILUPCG solver constant, independent on the number of performed local refinements.

Acknowledgement

The work of Anna Paszyńska, Maciej Paszyński, and Krzysztof Banaś have been supported by Polish National Science Center grant no DEC-2012/06/M/ST1/00363. The work of Konrad Jopek has been supported by Dean;s grant. The visits of Anna Paszyńska and Maciej Paszyński at KAUST has been supported by the Center of Numerical Porous Media. The visits of Maciej Paszyński at ICES has been supported by J.T. Oden Research Faculty Fellowship. The work of Keshav Pingali, Andrew Lenhart, Donald Nguyen and vists of Konrad Jopek at ICES has been supported by National Science Foundation grants CNS 1111407, CNS 1406355, XPS 1337281, CCF 1337281, CCF 1218568, ACI 1216701, and CNS 1064956.

References

[1] I. Babuśka, W. Rheinboldt, Error Estimates for Adaptive Finite Element Computations. SIAM Journal of Numerical Analysis, 15, 4 (1978) 736-754.

[2] P. Bientinesi, V. Eijkhout, K. Kim, J. Kurtz, R. van de Geijn, Sparse Direct Factorizations through Unassembled Hyper-Matrices, Computer Methods in Applied Mechanics and Engineering, 199 (2010) 430-438.

[3] K. Banas, A modular design for parallel adaptive finite element computational kernels, Lecture Notes in Computer Science (2004) 3037, 155-162.

[5] V. M. Calo, N. O. Collier, D. Pardo, M. Paszyński, Computational complexity and memory usage for multi-frontal direct solvers used in p finite element analysis. Procedia Computer Science, 4 (2011) 1854–186.

[5] K. Banas, P. Plaszewski, P. Maciol, Numerical integration on GPUs for higher order finite elements, Computers and mathematics with applications (2014) 67(6) 1319-1344

[6] L. Demkowicz, Computing with hp-Adaptive Finite Elements, Vol. II. Frontiers. Three Dimensional Elliptic and Maxwell Problems iwth Applications. Chapman & Hall / CRC Applied Mathematics & Nonlinear Science, (2006).

[7] I. S. Duff, J. K. Reid, The multifrontal solution of indefinite sparse symmetric linear systems. ACM Transactions on Mathematical Software, 9 (1983) 302-325.

[8] K. Banas, L. Demkowicz, Entropy controlled adaptive finite element simulations for compressible gas flow, Journal of computational physics (1996) 126(1) 181-201.

[9] J. Gaidamour, P. Hénon, A Parallel Direct/Iterative Solver Based on a Schur Complement Approach, 2008 11th IEEE International Conference on Computational Science and Engineering, (2008) 98-105.

[10] P. Gurgul, A linear complexity direct solver for h-adaptive grids with point singularities, Procedia Computer Science, 29 (2014) 1090–1099.

[11] F. Kruzel, K. Banas, Vectorized OpenCL implementation of numerical integration for higher order finite elements, Computers and mathematics with applications (2013) 66(10) 2030-2044.

[12] A. Niemi, I. Babuśka, J. Pitkaranta, Demkowicz L. Finite element analysis of the Girkmann problem using the modern hp-version and the classical h-version. Engineering with Computers, 28, 2 (2012) 123-134.

[13] R. H. Nochetto, , K. G. Siebert, A. Veeser, A. Multiscale, nonlinear and adaptive approximation. Springer (2009) 409-542.

[14] SLATEC Common Mathematical Library http://www.netlib.org/slatec/

[15] K. Banaś, Scalability analysis for a multigrid linear equations solver, Lecture Notes in Computer Science (2008) 4967, 1265-1274

[16] A. Paszyńska, M. Paszyński, K. Jopek, M. Woźniak, D. Goik, P. Gurgul, H. AbouEisha, M. Moshkov, V. M. Calo, A. Lenerth,D. Nguyen, K. Pingali, Quasi-Optimal Elimination Trees for 2D Grids with singularities, Scientiffic Programming (2015) Article ID 303024, 1-18.

[17] P. Plaszewski, P. Maciol, K. Banas, Finite Element Numerical Integration on GPUs, Lecture Notes in Computer Science (2010) 6067, 411-420