Linear Computational Cost Graph Grammar Based Direct Solver for 3D Adaptive Finite Element Method Simulations

Anna Paszyńska, Piotr Gurgul, Marcin Sieniek, and Maciej Paszyński

Abstract—In this paper we present a new graph grammar based direct solver algorithm delivering linear O(N)computational cost and linear O(N) memory usage for adaptive finite element method simulations. Classical direct solvers on regular grids deliver $O(N^{1.5})$ complexity for 2D problems and $O(N^2)$ in 3D ones. The linear computational cost of our solver is obtained by generating graph representation of the adaptive mesh and by utilizing dynamic construction prescribing the solver algorithm as graph grammar productions.

Index Terms—Direct solvers, graph grammar, adaptive finite element method.

I. INTRODUCTION

Direct solver is the core part of several challenging engineering applications performed by means of the Finite Element Method (FEM) [1]-[3]. Exemplary problems involve generation of acoustic waves over the model of the human head [4] or borehole resistivity simulations [5]. The process of solving finite element engineering problems starts with generation of the mesh describing the geometry of the computational problem. Next, the physical phenomena governing the problem is described by some Partial Differential Equation (PDE) with boundary and / or initial conditions. Then, PDE is discretized into a system of linear equations using FEM. At this point, the solver algorithm is executed in order to provide the solution to the system of linear equations. The aforementioned engineering problems generate huge linear systems with several million unknowns, and the solver algorithm is the most expensive part of the process in terms of the computational cost. Multi-frontal solver is the state-of-the art algorithm for solving linear systems of equations [6], [7] using the direct solver approach. The multi-frontal algorithm constructs an assembly tree

Manuscript received November 12, 2012; revised March 4, 2013.

The work presented in this paper is supported by Polish National Science Center grants no. NN519447739 and DEC-2011/03/N/ST6/01397. The work of the second author was partly supported by The European Union by means of European Social Fund, PO KL Priority IV: Higher Education and Research, Activity 4.1: Improvement and Development of Didactic Potential of the University and Increasing Number of Students of the Faculties Crucial for the National Economy Based on Knowledge, Subactivity 4.1.1: Improvement of the Didactic Potential of the AGH University of Science and Technology ``Human Assets", No. UDA – POKL.04.01.01-00-367/08-00.

Anna Paszyńska is with the Jagiellonian University, Krakow, Poland (e-mail: anna.paszynska@uj.edu.pl).

Piotr Gurgul, Marcin Sieniek, Maciej Paszyński are with the AGH University of Science and Technology, Krakow, Poland (e-mail: pgurgul@agh.edu.pl, msieniek @agh.edu.pl, paszynsk@agh.edu.pl).

based on the analysis of the connectivity data or the geometry of the computational mesh. Finite elements are merged into pairs and fully assembled unknowns are eliminated within frontal matrices associated to multiple branches of the tree. This process is repeated until the root of the assembly tree is reached. Finally, the common interface problem becomes solved and partial backward substitutions are recursively called on the assembly tree.

Classical direct solvers executed on regular grids deliver $O(N^{1.5})$ complexity for two dimensional problems and $O(N^2)$ complexity for three dimensional problems [8]. In this paper we propose a new graph grammar based direct solver, delivering linear O(N) time and memory complexity for computational problems with point singularities.

II. MODEL PROBLEM

The L-shape domain problem is a model academic problem formulated by Babuška in 1986 [9, 10], to test the convergence of the p and hp adaptive algorithms. The problem consists in solving the temperature distribution over the L-shape domain, presented in Fig. 1 with fixed zero temperature in the internal part of the boundary, and the Neumann boundary condition prescribing the heat transfer on the external boundary.



Fig. 1. The L-shape domain model problem.



Fig. 2. The solution of the L-shape domain model problem.

There is a single singularity in the central point of the domain (the gradient of temperature goes to infinity, compare Fig. 2), so the accurate numerical solution requires a sequence of adaptations in the direction of the central point. The problem can be summarized as follows:

Find the temperature distribution

$$u: R^2 \supset \Omega \ni x \to u(x) \in R \tag{1}$$

such that

$$\sum_{i=1}^{2} \frac{\partial^2 u}{\partial x_i^2} = 0 \text{ in } \Omega$$
 (2)

with boundary conditions

$$u = 0 \text{ on } \Gamma_D \tag{3}$$

$$\frac{\partial u}{\partial n} = g \text{ on } \Gamma_N \tag{4}$$

with *n* being the unit normal outward to $\partial \Omega$ vector, and being defined in the in the radial system of coordinates with the origin point *O* presented in Fig. 1. Equation (5) is actually based on the exact solution to the L-shape problem.

$$g(r,\theta) = r^{\frac{2}{3}} \sin^{\frac{2}{3}} \left(\theta + \frac{\Pi}{2}\right)$$
(5)

III. FICHERA PROBLEM

The Fichera problem constitutes the generalization of the L-shape domain problem into three dimensions. It can be summarized in the following way: Find the temperature distribution $u: R^3 \supset \Omega \ni x \rightarrow u(x) \in R$ over the domain presented in Fig. 3 such that

$$\sum_{i=1}^{3} \frac{\partial^2 u}{\partial x_i^2} = 0 \text{ in } \Omega$$
 (6)

with boundary conditions

$$u = 0 \text{ on } \Gamma_D \tag{7}$$

$$\frac{\partial u}{\partial n} = g \text{ on } \Gamma_N \tag{8}$$

with *n* being the unit normal outward to $\partial \Omega$ vector, and *g* is the exact solution of the L shape problem.



Fig. 3. Domain visualization for the Fichera problem



Fig. 4. The sequence of meshes generated by the self-adaptive *hp*-FEM for the Fichera problem. Different colors denote different polynomial orders of approximations presented in Fig. 3.



Fig. 5. The sequence of meshes generated by the self-adaptive h-FEM for the Fichera problem with fixed p=5.

IV. AUTOMATIC HP-ADAPTATION

A. Exponential Convergence

The presented problems have been solved by both self-adaptive *hp*-FEM and h-FEM (with constant polynomial approximation level p=5). Only hp-adaptive FEM is guaranted to deliver exponential convergence of the numerical error with respect to the mesh size [2], [3]. See Table I to compare convergence rates for both methods.

 TABLE I: CONVERGENCE RATES OF THE SELF-ADAPTIVE H-FEM (LEFT)

 AND HP-FEM (RIGHT) FOR THE FICHERA PROBLEM.

Ν	Error		Ν	Error
1206	5.06		665	9.75
8261	3.18		846	6.18
13726	2.46		1093	4.58
19191	2.23		1577	3.55
35586	2.15]	2247	2.91
		-	3493	2.51

The basic idea behind hp-FEM has been explored further in this paragraph.

B. Mesh Refinements

Generally, the quality of the solution can be improved by the expansion of the approximation base. In FEM terms, this could be done thanks to two kinds of mesh refinements:

- P-refinement increase order of the basis functions on the elements where the error rate is higher than desired. More basis functions in the base mean smoother and more accurate solution but also more computations and the use of high-order polynomials often leads to undesirable side-effects (e.g. Runge effect).
- 2) H-adaptation split the element into two or four in order to obtain finer mesh. This idea arose from the observation that the domain is usually non-uniform and in order to approximate the solution fairly some places require more precise computations than others, where the acceptable solution can be achieved using small number of elements. The crucial factor in achieving optimal results is to decide if a given element should be split into two parts horizontally, into two parts vertically, into four parts (both horizontally and vertically) or not split at all. The refinement process is fairly simple in 1D but in 2D and 3D many refinement rules to follow are being enforced.

C. Automated Hp-Adaptation Algorithm

Neither p- nor h-adaptation guarantee error rate decrease that is exponential with a step number. This can be achieved by combining these two methods. In order to identify the most sensitive areas at each stage dynamically, and improve the solution as much as possible, we employ the self-adaptive algorithm that decides whether a given element should be further refined or is already fine enough for the satisfactory solution. These steps have been summarized in Alg. 1.

1: function adaptive_fem($mesh_{initial}$, $err_{desired}$)

2:
$$mesh_{coarse} = mesh_{initial}$$

- 4: u_{coarse} = compute solution on $mesh_{coarse}$
- 5: $mesh_{fine} = copy \ mesh_{coarse}$
- 6: divide each element of $mesh_{fine}$ into two new elements
- 7: increase order of functions on each element of $mesh_{fine}$ by 1
- 8: u_{fine} = compute the solution on $mesh_{fine}$
- 9: for each element κ of $mesh_{fine}$ do

```
10: err_{K} = compute error decrease rate on K
```

- 11: end do
- 12: $mesh_{adapted} = copy \ mesh_{coarse}$
- 13: for each element κ of $mesh_{adapted}$ do

14: **if**
$$err_{K}$$
 > threshold * err_{max} **then**

- 15: divide K
- 16: end if
- 17: **end do**
- 18: enforce $mesh_{adapted}$ integrity

19:
$$mesh_{coarse} = mesh_{adapted}$$

20: **until**
$$err_{max} < err_{desired}$$

21: return mesh_{fine}

22: end function

Alg. 1. hp-adaptive PBI pseudocode

We iterate until the solution on the given mesh reaches satisfactory error rate (lines 3-20). First, we compute the solution on the initial mesh, called *coarse mesh*. Next, we create its copy called *fine mesh* and perform both h- (line 6) and p-refinement (line 7) on each element κ . Then, we compute the solution *fine mesh* and for each element κ we evaluate relative error decrease. If it is satisfactory (here we can assume threshold = 0.3, see line 14), we keep the hp refinement on that element, since it was a justified decision. Otherwise, we skip the refinement for such element. More details can be found in [2].

V. GRAPH GRAMMAR MODEL

The input for the solver algorithm is the locally refined computational mesh represented as a graph. The mesh is obtained by executing a sequence of graph grammar productions, summarized in Fig. 6 and 7.



Fig. 6. Graph grammar production for generation rectangular finite element

The computational mesh is further h-refined, which is expressed by graph grammar production summarized in Fig. 8. Selected rectangular elements are broken into 8 new son elements with 12 new faces.

In addition to that, the solver algorithm obtains a sequence of element matrices, one matrix for each sub-graph of the mesh representing a single finite element, resulting from discretization of the computational problem. The solver algorithm browses the graph representation of the mesh from bottom elements up to the root elements, and it merges the element matrices into only one frontal matrix. It first identifies fully assembled nodes located within each level of the graph representation of the mesh, eliminates them, and then it iterates the process by going up to the next level. This pattern for elimination ensures that the size of a single frontal matrix involved in the solver algorithm remains constant.

The solver algorithm is expressed as graph grammar productions coloring graph nodes, as it is presented in Fig. 9, 10 and 11.

(P Identify 3D Face)



Fig. 7. Graph grammar production for identification of common faces of two adjacent elements.



Fig. 8. Graph grammar production for adaptation of rectangular finite element.

VI. LINEAR COMPUTATIONAL COST OF THE SOLVER ALGORITHM

Since the cost at each step (level of the elimination tree) is constant, the total cost of the algorithm is proportional to the number of levels, which by grid construction is proportional to the number of unknowns. As a result, we obtain a solver algorithm with linear computational cost with respect to the number of unknowns.

This theoretical estimation has been verified experimentally, as it is presented in Fig. 12. It must be also clearly stated that the theoretical proof of the linear cost has been so far conducted only for h-FEM with fixed p only. For hp-FEM, where p may be different for each element, linear complexity has not been proven yet. On the other hand, experimental results show that solver's complexity is for hp-FEM is very close to linear, compare Fig. 13.



Fig. 9. Coloring of graph nodes for elimination of the deepest level in the adaptive mesh. The red nodes denotes fully assembled nodes, the yellow nodes denotes interface nodes. The size of the frontal matrix is equal to the number of yellow and red nodes, and we eliminate all red nodes.

Although the linear solver code that was used to produce the results was only a proof-of-concept, not optimized implementation, it already has delivered very promising performance.

The presented linear cost solver can help to dramatically lower the computational intensity of the existing h- and hp-FEM solutions. In terms of future work, it is important to transform the solver algorithm into efficient production code that could be easily applied to the large scope of problems.



Fig. 10. Coloring of graph nodes for elimination of the next level in the adaptive mesh



Fig. 11. Coloring of graph nodes for elimination of the top level in the adaptive mesh. This time both red (internal) and yellow (boundary) nodes are fully assembled and can be eliminated.



Fig. 12. Linear computational cost of the h-adaptive solver algorithm



Fig. 13. Almost linear computational cost of the hp-adaptive solver algorithm

REFERENCES

- [1] T. J. R. Hughes, *Linear static and dynamic finite element analysis*, Dover Publications, 2000.
- [2] L. Demkowicz, *Computing with Hp-Adaptive Finite Elements*, vol. 1, One and Two Dimensional Elliptic and Maxwell Problems, Chapmann & Hall / CRC Press, 2006.
- [3] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz, and A. Zdunek, *Computing with Hp-Adaptive Finite Elements*, vol. 2, Frontiers, Three Dimensional Elliptic and Maxwell Problems with Applications, Chapmann & Hall / CRC Press, 2007.
- [4] L. Demkowicz, P. Gatto, J. Kurtz, M. Paszyński, W. Rachowicz, E. Bleszyński, M. Bleszyński, M. Hamilton, C. Champlin, and D. Pardo, "Modeling of bone conduction of sound in the human head using hp-finite elements: code design and verification," *Computer Methods in Applied Mechanics and Engineering*, pp. 1757–1773, vol. 200, Issues 21-22, 2010.
- [5] M. Paszyński and R. Schaefer, "Graph grammar driven parallel partial differential equation solver," *Concurrency and Computations, Practise and Experience*, vol. 22, no. 9, pp. 1063-1097, 2009.
- [6] I. S. Duff and J. K. Reid, "The multifrontal solution of unsymmetric sets of linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 5, pp. 633-641, 1984

- [7] P. Geng, T. J. Oden, and R. A. van de Geijn, "A Parallel Multifrontal Algorithm and Its Implementation," *Computer Methods in Applied Mechanics and Engineering*, vol. 149, pp. 289-301, 2006.
- [8] N. Collier, D. Pardo, L. Dalcin, M. Paszyński, and V. Calo, "The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers," *Computer Methods in Applied Mechanics and Engineering*, vol. 212-216, 2012, pp. 353-361.
- [9] I. Babuška and B. Guo, "The hp-version of the finite element method, Part I: The basic approximation results," *Computational Mechanics*, vol. 1, pp. 21-41, 1986.
- [10] I. Babuška and B. Guo, "The hp-version of the finite element method, Part II: General results and applications," *Computational Mechanics*, vol. 1, pp. 203-220, 1986.



Anna Paszyńska was born in Krakow, Poland, 24.09.1976. She received Master Degree in computer science from Jagiellonian University, Krakow, Poland in 2001. She received her Ph.D. in computer science from the Institute of Fundamental Technological Research of Polish Academy of Sciences in Warsaw, Poland, in 2007. She currently works as an ASSISTANT PROFESSOR at Faculty of Physics, Astronomy and Applied Computer

Science at the Jagiellonian University in Kraków, Poland. Her research interests include evolutionary algorithms, graph grammars and computer aided design. Her recent publications include:

A graph grammar model of the hp adaptive three dimensional Finite Element Method. P. 1 / Anna Paszyńska, Ewa Grabska, Maciej Paszyński // Fundamenta Informaticae ; 2012 vol. 114 no. 2 s. 149–182 IF: 0.522 A graph grammar model of the hp adaptive three dimensional Finite Element Method. P. 2 / Anna Paszyńska, Ewa Grabska, Maciej Paszyński // Fundamenta Informaticae ; 2012 vol. 114 no. 2 s. 183–201 IF: 0.522 Graph grammar based model for three dimensional multi-physics simulations / Maciej Paszyński, Anna Paszyńska, Robert Schaefer, Advances in intelligent modeling and simulation : simulation tools and applications, Berlin ; Heidelberg : Springer-Verlag, 2012, s. 299–324



Piotr Gurgul was born in May 6th 1987 in Krakow, Poland. He graduated from a combined Master of Science and Bachelor of Engineering in Computer Science program at the AGH University of Science and Technology in Krakow, Poland in 2011 and from a Bachelor of Business Administration in Management from Krakow University of Economics in 2012.

Currently he is a PH.D. CANDIDATE and a second year PH.D STUDENT at the AGH University of Science and Technology. His scientific interests focus on hp-adaptive finite element method, and adaptive solvers.

His work has been rewarded with multiple scholarships, including Polish Ministry of Science and Higher Education stipend in 2009, 2010 and 2011 and European Union Scholarship for Young Scientists. He has also completed a bunch of professional internships overseas, including Google in the US and Australia and Microsoft Corp. in USA.



Marcin Sieniek was born in Sep 3rd 1987 in Krakow, Poland. He graduated from a combined Master of Science and Bachelor of Engineering in Computer Science program at AGH University of Science and Technology in Krakow, Poland in 2011 and from a Bachelor of Business Administration in Management program from Krakow University of Economics in 2012. Between 2009 and 2012 he worked professionally with

LP. Currently he is a PH.D. STUDENT in Computer Science at AGH University of Science and Technology and runs a startup project aiming to enhance student collaboration in the educational process. His research interests include finite element method, projection techniques as well as management and transfer of innovations.

Mr. Sieniek has been awarded with multiple stipends and honors, including Polish Ministry of Science and Higher Education stipend (2010), Mayor of Krakow municipal award for scientific impact on the City of Krakow (2008) and a regional Sapere Auso award for special scientific achievements (2008 and 2010).



Maciej Paszynski was born in 12 October 1974 in Krakow, Poland. He got Master Degree in Computer Science from Jagiellonian University, Krakow, Poland in 1999. He got PhD in Computer Science with applications to mathematics in 2003 from Jagiellonian University, Krakow, Poland. He got Habilitation in Computer Science from AGH University of Science and Technology, Krakow, Poland in 2010. He got the

Computer Science Associated Professor position from AGH University of Science and Technology, Krakow, Poland in 2013.He worked as PROGRAMMER / SOFTWARE ENGINEER in RoboBAT company (later bought by AutoDESK) in years 1999-2003 in Krakow, Poland.

In 2003–2005, he worked as a POSTDOCTORAL FELLOW at the Institute for Computational Engineering and Sciences (ICES) at The University of Texas at Austin. In summer 2006 and 2007 he worked as a POSTDOCTORAL FELLOW at the Department of Petroleum and Geosystems Engineering at The University of Texas at Austin. Since 2007 he is a frequent VISITING PROFESSOR at ICES at The University of Texas at Austin, USA, at Basque Center for Applied Mathematics (BCAM), at the Department of Mathematics, Universidad del Pais Vasco, Bilbao, Spain and at King Abdullah University of Science and Technology (KAUST), Saudi Arabia.

Prof. Maciej Paszynski is a main chairman of the International Conference on Computational Science ICCS Workshop "Agent based simulations, adaptive algorithms and solvers" since 2007.