European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS 2000
Barcelona, 11-14 September 2000

MULTIGRID METHODS APPLIED TO THE H AND P VERSIONS OF THE FEM

Marco L. Bittencourt*, Alberto C. Nogueira Jr†, and Raúl A. Feijóo†

*Departamento de Projeto Mecânico, Faculdade de Engenharia Mecânica, Universidade
Estadual de Campinas, P.O. Box 6051, Campinas, SP, Brazil, 13083-970, e-mail:
mlb@fem.unicamp.br, web page: http://www.fem.unicamp.br/mlb

†Departamento de Projeto Mecânico, Faculdade de Engenharia Mecânica, Universidade
Estadual de Campinas, P.O. Box 6051, Campinas, SP, Brazil, 13083-970,
e-mail:rapadura@fem.unicamp.br, web page: http://www.fem.unicamp.br/comet

†Departamento de Mecânica Computacional, Laboratório Nacional de Computação
Científica, Av. Getúlio Vargas 333, Petrópolis, RJ, Brazil, 25651-070, e-mail:
feij@lncc.br, web page: http://www.lncc.br/feij

Key words: Multigrid, Multi-p, Finite Element Method, Iterative Methods.

Abstract. *In this work, multigrid procedures for the h and p versions of the FEM are discussed. Multigrid methods have an optimal solution cost for linear elliptic problems. In this work, non-nested h-approximation spaces and algebraic multi-p methods are used. For the p finite element version, the approximation spaces are nested and correspond to the nested meshes commonly used in multigrid methods. For h and hp variants, the approximations are non-nested and quadtree and octree data structures are used to map mesh information between two levels. Examples of linear elasticity and Poisson's problems are presented to demonstrate the suitability of the proposed multigrid methods.*

1 INTRODUCTION

The application of the Finite Element Method (FEM) [1] for solving linear elliptic problems requires the solution of the following system of equations,

$$\mathbf{A}\mathbf{u} = \mathbf{b} \tag{1}$$

where $\mathbf{A}$ is a symmetric matrix of order N; $\mathbf{u}$ and $\mathbf{b}$ are the vectors of unknown and independent terms, respectively. For the solution of (1), direct, iterative and multigrid algorithms are usually applied [1, 2, 5].

Basically, h-multigrid methods use several meshes for solving (1), applying elements like nested iterations, coarse grid correction, transfer operators and relaxation schemes. Multigrid methods have as the main feature a theoretical linear asymptotic cost $\mathcal{O}(N)$ for the solution of (1).

Traditionally, multigrid methods have been used with nested meshes, which simplify transfer operators, mesh generation, definition of adaptive refinement criteria [5, 15, 18] and convergence for some class of problems [6, 11, 16]. However, engineering problems have complex geometries, which make it difficult to generate a sequence of truly nested meshes, therefore making the use of non-nested h-approximation spaces an interesting option. Adaptive multigrid algorithms [3] were also developed using the Zhu-Zienkiewicz error estimator [22], adaptive analysis and automatic mesh generation [8, 9].

As another strategy to overcome the problem of generating truly nested meshes, algebraic multigrid methods have been proposed. Some of this methods employ the p-version of the FEM and are called multi-p methods [12, 13]. In this case, the sequence of approximating finite element spaces are naturally nested and the whole algorithm uses just one mesh.

This paper presents concepts related to h and p-multigrid methods. Initially, a brief introduction to the principal elements of multigrid methods for the h and p versions of the FEM is presented. Results for linear elastic problems solved by h-multigrid methods are presented. For the p-version of the FEM, the choice of the basis functions is very important to reach a good condition number of the global matrix. Experiments for some set of functions are presented for the solution of the Poisson's problem.

2 h-MULTIGRID METHODS

Multigrid methods employ some elements such as nested iterations, coarse grid correction and transfers operators.

The nested iteration scheme is used to get a better initial fine mesh solution for an iterative method. It is well known that the performance of an iterative procedure can be improved by using a better initial approximation obtained, for example, by relaxation on a coarser mesh. As the number of unknowns on the coarser mesh is smaller, the computational cost for one relaxation is also much smaller than one on the finer mesh. The following scheme, called nested iterations, obtains a better approximation for the solution of the finest mesh [6]:

```
Relax on $\mathbf{Au} = \mathbf{b}$ on the coarsest mesh
                        $\vdots$
Relax on $\mathbf{Au} = \mathbf{b}$ on $\Omega^{k-2}$ to obtain an initial approximation for $\Omega^{k-1}$
Relax on $\mathbf{Au} = \mathbf{b}$ on $\Omega^{k-1}$ to obtain an initial approximation for $\Omega^{k}$
Relax on $\mathbf{Au} = \mathbf{b}$ on $\Omega^{k}$ to obtain a final approximation to the solution $\mathbf{u}$.
```

Let v be an approximation for the solution u of (1). The residual equation $Ae = r$ defines a relation between the algebraic error $e = u - v$ and the residual $r = b - Av$ and allows iterations on the error e. Relaxing on (1) with some approximation v is equivalent to iterating on the residual equation, where $e = 0$ is the initial approximation. The coarse grid correction scheme is defined in the following way [6]:

```
Relax on $\mathbf{Au} = \mathbf{b}$ on $\Omega^k$ to obtain an approximation $\mathbf{v}^k$
Compute the residual $\mathbf{r} = \mathbf{b} - \mathbf{Av}^k$
    Relax on $\mathbf{Ae} = \mathbf{r}$ on $\Omega^{k-1}$ to obtain an approximation to the error $\mathbf{e}^{k-1}$
Correct the approximation obtained on $\Omega^k$ with the error estimate on $\Omega^{k-1}$ : $\mathbf{v}^k \leftarrow \mathbf{v}^k + \mathbf{e}^{k-1}$
```

The concepts discussed previously are based on transferring functions between coarse and fine meshes. For this purpose, transferring operators must be defined.

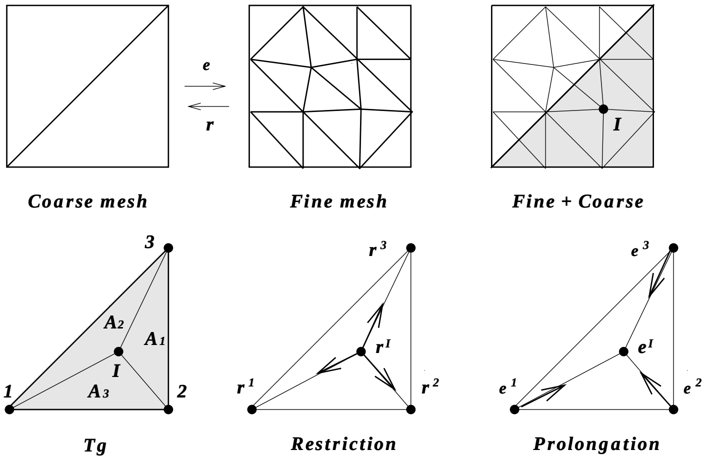


Figure 1: Restriction and prolongation operators.

The first operator, denoted by I_{k-1}^{k}, transforms functions from the coarse mesh Ω^{k-1} to the fine mesh Ω^{k} and is called an interpolation or prolongation operator. It is used to map the error e^{k-1} or the initial approximation v^{k-1} from the coarse mesh to the fine mesh. The restriction operator I_{k}^{k-1} maps functions from the fine mesh Ω^{k} to the coarse mesh Ω^{k-1}, as in the case of the residual projection r^{k} onto Ω^{k-1}.

For nested meshes, the simplest types of restriction are injection and weighting operators [6]. In this paper, however, non-nested h-meshes are considered. Taking a two-dimensional domain discretized by linear triangles, the restriction and prolongation operators are defined using the element shape functions expressed in area coordinates $A_i\,(i=1,2,3)$. Figure 1 shows the operators for linear triangles. For each finer node I, one needs to know the coarser element T_g in which node I is contained and the respective values of the shape functions calculated on the local coordinates of I. For this purpose, efficient quadtree and octree data structures have been used [8].

Using the notation indicated in Figure 1, the restriction operator computes the residual for each node of the triangle T_g by taking the contribution of all finer nodes contained in T_g:

$$\mathbf{r}^{i} = I_{k}^{k-1}\mathbf{r}^{k} = \sum_{l=1}^{N_f} A_i^I \mathbf{r}_l^I \qquad i=1,2,3 \tag{2}$$

where N_f is the number of finer nodes I in T_g with residual r^I and area coordinates A^I.

Analogously, the mapping of the error e^{k-1} on Ω^{k-1} to finer mesh Ω^{k} is based on an interpolation using the shape functions of element T_g:

$$e^{I} = I_{k}^{k-1}e^{k-1} = \sum_{l=1}^{3} A_l^I e^{l} \tag{3}$$

The same procedures can be extended to three-dimensional examples discretized by linear tetrahedral using volume coordinates. A variational description of the restriction and prolongation operators is presented in [4].

Figure 2 shows some commonly used multigrid strategies. Cycles V and W are based on the recursive application of the coarse grid correction scheme. Basically, when going from a finer mesh Ω^{k} to a coarser mesh Ω^{k-1}, ν_1 pre-relaxations are executed on the original equation $Au=b$ or on the error equation $Ae=r$, and residual r is calculated and projected on mesh Ω^{k-1} using the restriction operator I_{k}^{k-1}. In the coarsest mesh, the error equation is solved by a direct or iterative numerical method. Finally, corrections e are mapped to the other levels using the prolongation operator I_{k-1}^{k}, and ν_2 post-relaxations are executed on each mesh.

The FMV algorithm uses the concept of nested iterations. Each V cycle is preceded by one or more smaller V cycles, according to the value of parameter ν_0. Equation $Au=b$ is solved in the coarsest mesh and its solution is prolonged as an initial approximation to the next level. Then ν_0 V cycles are executed, ultimately obtaining a better approximation

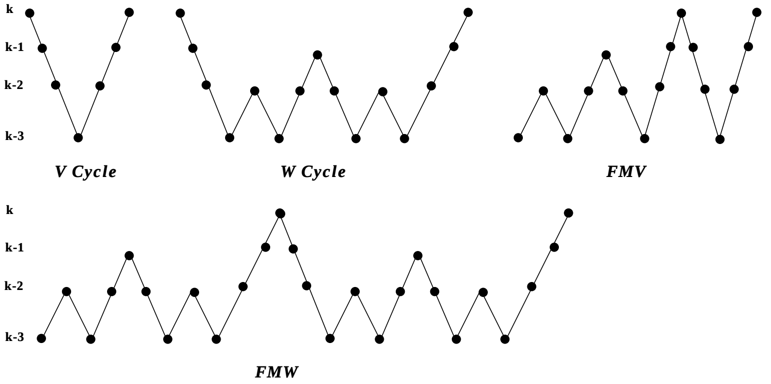


Figure 2: Multigrid strategies.

to the next mesh. This procedure is repeated until reaching the finest mesh. The FMW technique uses smaller W cycles to obtain a better initial approximation to a next level and is analogous to the FMV cycle. These procedures are illustrated in Figure 2 for $\nu_0 = 1$.

3 MULTI-p METHODS

In the p-version of the FEM, hierarchical shape functions are used [7, 14, 19, 20, 21]. In this case, the approximating finite element spaces V_p (including shape functions of order $\leq p$) are nested, i.e.,

$$V_1 \subset V_2 \subset \cdots \subset V_{p-1} \subset V_p.$$

Similarly, the stiffness matrices and load vectors have also hierarchical structures. Hence, if A_p and b_p are the global stiffness matrix and load vector corresponding to the shape functions of order $\leq p$, the resulting algebraic sistem of equations $A_p u = b_p$ can be written as follows

$$\left[\begin{array}{cc} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{array} \right] \left[\begin{array}{c} u_1 \\ u_2 \end{array} \right] = \left[\begin{array}{c} b_1 \\ b_2 \end{array} \right], \tag{4}$$

where A_{11} and b_1 correspond to order $\leq p-1$, i.e., $A_{p-1} = A_{11}$ and $b_{p-1} = b_1$.

The idea of a multigrid scheme for the p-version of the FEM, named multi-p methods [13], derives from this hierarchical properties and the well stablished multigrid features.

In this way, the transferring operators (prolongation and restriction) for the multi-p method are naturally defined by the injection mapping

$$I_{p-1}^{p} = \left[\begin{array}{c} I_{p-1} \\ 0 \end{array} \right], \quad I_{p}^{p-1} = \left[\begin{array}{cc} I_{p-1} & 0 \end{array} \right], \tag{5}$$

where I_{p-1} is the identity matrix of order $N_{p-1} \times N_{p-1}$ with N_p = number of degrees of freedom used in the p-discretization. It is worth noting that, according to this definition, the transferring operators are related by

$$I_p^{p-1} = (I_{p-1}^p)^T.$$

In comparison with h-multigrid methods, the prolongation and restriction operators are very simple and independent of the domain geometry. In addition, it is easy to verify that the transferring operators satisfy the following relation

$$A_{p-1} = I_p^{p-1} A_p I_{p-1}^p .$$

The multi-p V and FMV cycles are built in analogy with the respective h-multigrid strategies. The multi-p V cycle algorithm looks exactly like the h-multigrid V cycle. The only difference is that the multilevel computations are conducted at levels of different degrees of shape functions, not at different meshes. The multi-p FMV cycle also uses the concept of nested iterations. At each ascending level, a V cycle is applied generating an approximate solution to the equation at the next level. A standard multi-p V cycle algorithm is presented below.

ALGORITHM: TWO-LEVEL STANDARD MULTI-p V CYCLE

Let $F(u, b)$ be a linear stationary scheme (as Gauss-Seidel or Jacobi) in matrix form and A_p e b_p defined such as in (4).

Let an inicial guess to the problem $A_p u = b_p$ be $[u_1^0 u_2^0]^T$.

1. Perform ν iterations of $F(x, b)$ on (4) and let the solution be $[u_1^\nu u_2^\nu]^T$;
2. Solve
$$A_{11} u_1 = b_1 - A_{12} u_2^\nu \qquad \text{(restriction)}$$
directly (or iteratively with inicial guess u_1^ν) to obtain $u_1^{\nu+1}$;
3. Perform ν iterations on (4) with inicial guess $[u_1^{\nu+1} u_2^\nu]^T$ (prolongation) to obtain $[u_1^{2\nu+1} u_2^{2\nu}]^T$;
4. If the residual is within the tolerance, stop; otherwise let $[u_1^{2\nu+1} u_2^{2\nu}]^T \Longrightarrow [u_1^0 u_2^0]^T$ and return to 1.

From this algorithm, it can be observed that the lower level computations improve only components in u_1. A slight modification in the algorithm leads to an imediatly extension of the standard multi-p V cycle which also improves the components in u_2. In order to do it, it is nevessary to introduce a new step between steps 2 and 3 of the standard algorithm as follows

MODIFIED ALGORITHM

1. idem to standard multi-p V cycle;
2. idem to standard multi-p V cycle;

3. Perform σ iterations of $F(u,b)$ on the following equations:

   $$A_{22}u_2 = b_2 - A_{12}^T u_1^{\nu+1}$$

   with inicial guess u_2^{ν} and let the solution be $u_2^{\nu+1}$;

4. Perform ν iterations on (4) with inicial guess $[u_1^{\nu+1} u_2^{\nu+1}]^T$ (prolongation) to obtain $[u_1^{2\nu+1} u_2^{2\nu+1}]^T$;

5. If the residual is within the tolerance, stop; otherwise let $[u_1^{2\nu+1} u_2^{2\nu+1}]^T \Longrightarrow [u_1^0 u_2^0]^T$ and return to 1.

4 RESULTS

4.1 h-Multigrid

In this section, the behavior in terms of number of operations and memory requirements for sparse Gaussian elimination (SGE), conjugate gradients, and multigrid is analyzed when applied to three-dimensional linear elastic examples discretized by linear tetrahedral meshes. A flop is defined as a single floating point operation.

For sparse Gaussian elimination on sparse matrices, it is useful to apply a symbolic factorization procedure to determine non-zero coefficients in the matrix factors [17]. Thus, we know in advance the total number of factored matrix coefficients to allocate. The minimum degree renumbering procedure was used to renumber the mesh nodes in order to reduce the fill-in [10].

For iterative and multigrid methods, the convergence criterion $\|\mathbf{Au} - \mathbf{b}\|_2 \,/\, \|\mathbf{b}\|_2 < \xi$ with precision $\xi = 10^{-4}$ was used. The iterative methods are based on conjugate gradient (CG) with diagonal (CGD), SSOR (CGSS), and symmetric Gauss-Seidel (CGGS) preconditioners. Gauss-Seidel was used as the relaxation scheme in all of the multigrid methods.

The first example is a planar elliptic fracture problem in an infinite domain represented by a parallelpiped. There are three meshes (see Figure 3). The second example is another planar fracture problem in a cylindrical bar, a penny-shaped crack. There are four meshes (see Figure 4). In the these two examples, symmetry conditions were considered. The sequences of meshes were generated by an adaptive procedure based on the Zienkiewicz-Zhu error estimator [22].

The mesh features for all examples are given in Table 1. This includes the numbers of nodes, elements, and equations. For a sparse matrix, the table includes both the total number of coefficients for the entire matrix and the average number of coefficients per row. Where relevant, numbers for direct (NCoef^{dir}, m^{dir}) and iterative/multigrid (NCoef^{ite}, m^{ite}) methods are included.

Figures 5 and 6 illustrate the results in terms of the number of operations and the memory space requirements. The solution on the coarsest meshes was obtained by the CGGS method. The high factorization cost on these meshes affects the multigrid performance as illustrated in Figures 5(c) and 5(d). The number iterations on the finest mesh for the fracture problems is based on using CGGS on the coarsest meshes. The FMW results for the cylinder example, however, were obtained by using sparse Gaussian elimination on the coarsest mesh.

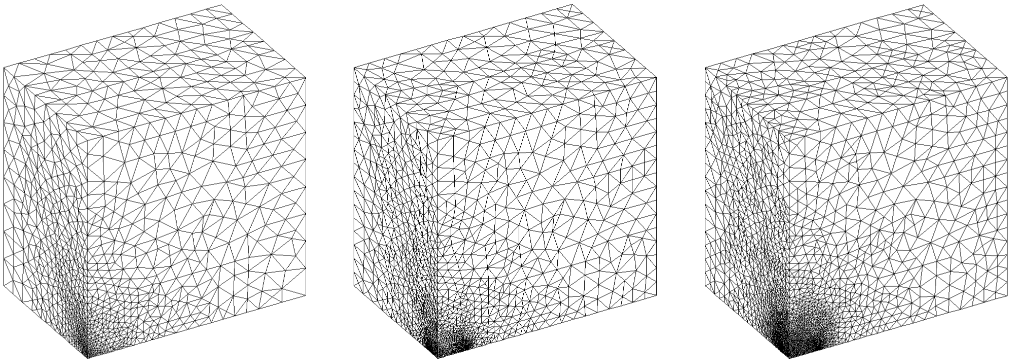


Figure 3: Meshes for the parallelpiped example ($E = 2.1 \times 10^5$; $\nu = 0,3$).

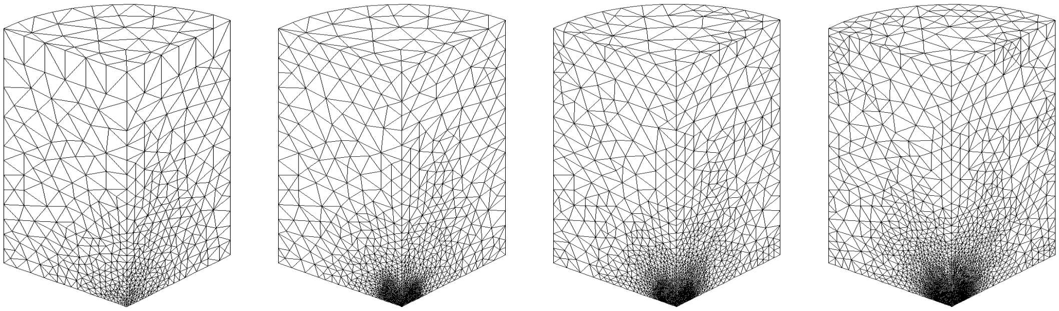


Figure 4: Meshes for the cylinder example ($E = 2.1 \times 10^5$; $\nu = 0,3$).

Iterative techniques are better in three-dimensional problems due to the smaller demand for memory. The acceleration obtained by using multigrid is significantly superior in terms of the number of operations when compared to conjugate gradient-based methods. The increase in memory space caused by multigrid methods is not significant. This allows all data used in the solution process to be loaded into main memory. For linear problems, multigrid is significantly superior to the iterative algorithms based on the conjugated gradient technique. Multigrid is also superior to direct sparse Gaussian elimination.

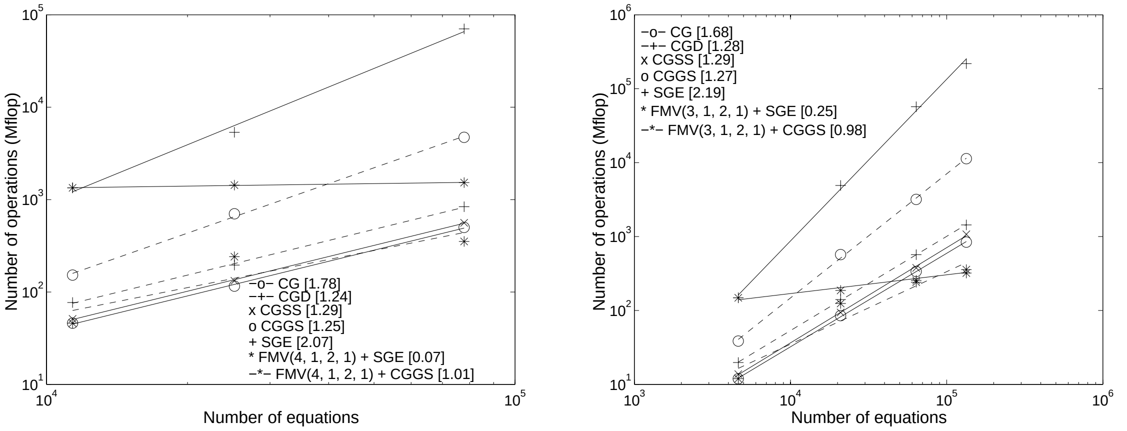
4.2 Multi-p

As mentioned previously, the selection of the shape functions is very important for the p-version of the FEM to reach sparsity and good conditioning of the global matrix. These feature affects strongly the behavior of the iterative and multigrid methods for the

Parallelepiped							
Mesh	Nodes	Elements	Equations	m^{dir}	NCoef^{dir}	m^{ite}	NCoef^{ite}
1	4228	19871	11365	327.0	3716698	19.5	221037
2	9241	44015	25184	422.0	10627486	19.8	495535
3	27652	139348	77862	786.0	61196124	20.5	1596409

Cylinder							
Mesh	Nodes	Elements	Equations	m^{dir}	NCoef^{dir}	m^{ite}	NCoef^{ite}
1	1810	7599	4630	179.4	830385	18.0	83141
2	7753	36746	20933	428.2	8962786	19.6	409267
3	22670	114634	63778	792.9	50566338	20.5	1309692
4	46982	238546	133471	959.3	128036645	20.6	2746298

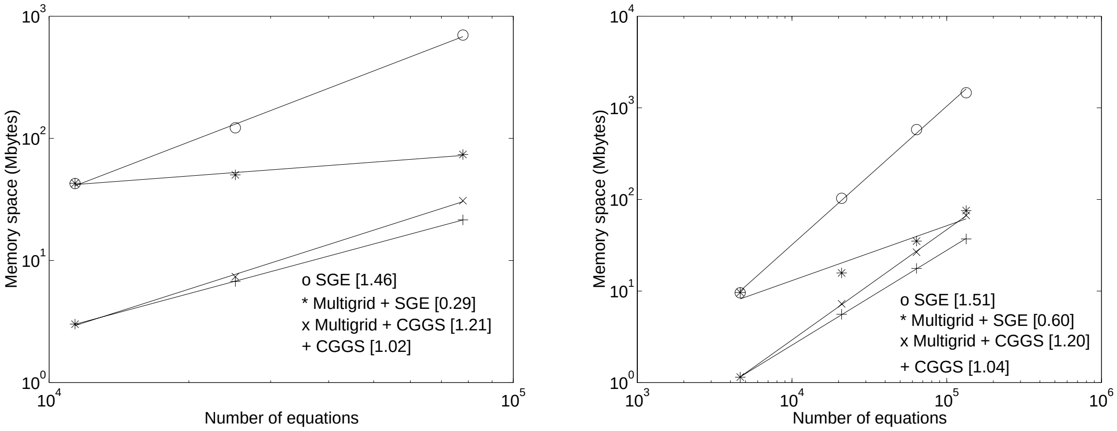
Table 1: Mesh attributes.



(a) Parallelepiped .

(b) Cylinder.

Figure 5: Number of operations.



(a) Parallelepiped.

(b) Cylinder.

Figure 6: Memory space.

solution of the systems of equations. In this paper, results obtained for five sets of p-shape functions will be presented. The multi-p methods have been implemented and the results will be presented during the congress.

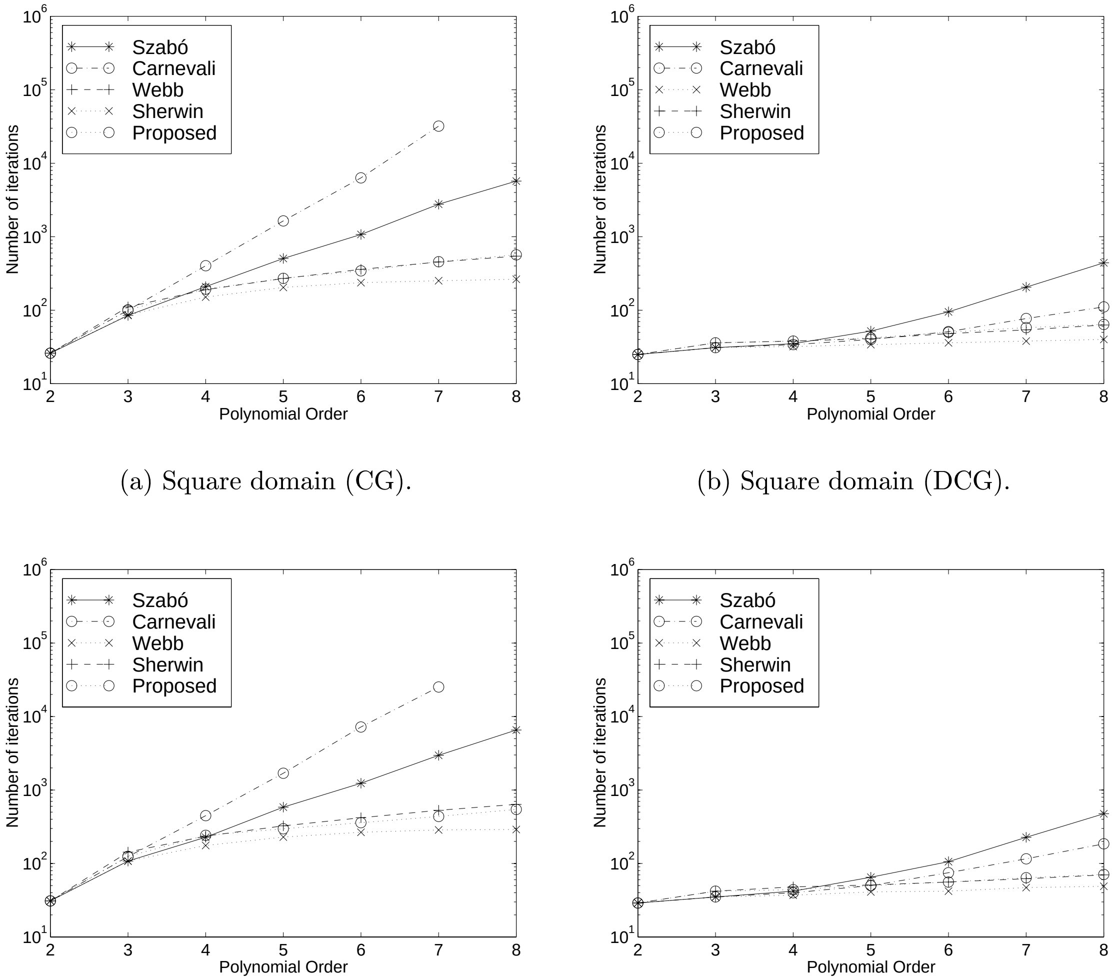
The Poisson's problem $\nabla^2 u + f = 0$ with Dirichlet boundary conditions was solved for a square and a L-shaped domains. The source term was selected in such way to have pronounced local variations on the solution u.

Figure 7 shows the number of iterations of the standard CG and diagonal CG versus the polynomial order for the five sets of p-shape functions [7, 14, 19, 20, 21] used in the solution of the square and L-shaped domains.

It can be observed that the shape functions proposed by Webb [21] have the best behaviour in terms of the number of iterations for the solution of the systems of equations. This behavior is compatible with the results obtained for the local element matrices using the same sets of shape functions [14]. Based on these results, the Webb's functions have been used to implemented the multi-p methods.

Acknowledgments

This work was supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP). The authors are grateful for the software facilities provided by the TACSOM Group (http://www.lncc.br/~tacsom).



(a) Square domain (CG).

(b) Square domain (DCG).

(c) L-shaped domain (CG).

(d) L-shaped domain (DCG).

Figure 7: Number of iterations versus polynomial order.

REFERENCES

[1] O. Axelsson and V.A. Barker. *Finite element solution of boundary value problems – theory and computation.* Academic Press, Orlando, 1984.

[2] M.L. Bittencourt. *Métodos multigrid e iterativos adaptáveis aplicados em malhas não-estruturadas.* PhD thesis, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1996.

[3] M.L. Bittencourt, C.C. Douglas, and R.A. Feijóo. Adaptive non-nested multigrid methods. Relatório de Pesquisa e Desenvolvimento 27/98, LNCC, Rio de Janeiro, 1998.

[4] M.L. Bittencourt and R.A. Feijóo. Non-nested multigrid methods in finite element linear structural analysis. In *Virtual Proceedings of the 8th Copper Mountain Conference on Multigrid Methods*, http://www.mgnet.org, April 1997. The University of Colorado.

[5] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[6] W.L. Briggs. *A multigrid tutorial.* SIAM Books, Philadelphia, 1987.

[7] P. Carnevali, R. B. Morris, Y. Tsuji, and G. Taylor. New basis functions and computational procedures for p-version finite element analysis. *Int. J. Numer. Methods Engrg.*, 36:3759–3779, 1993.

[8] E. Dari. *Contribuciones a la triangulación automática de dominios tridimensionales.* PhD thesis, Instituto Balseiro, Bariloche, Argentina, 1994.

[9] E.A. Fancello, A.C.S. Guimarães, R.A. Feijóo, and M. Venere. Automatic two-dimensional mesh generation using object-oriented programming. In *Proceedings of 11th Brazilian Congress of Mechanical Engineering*, pages 635–638, São Paulo, 1991. Brazilian Association of Mechanical Sciences.

[10] A. George and J.W. Liu. *Computer solution of large sparse positive definite systems.* Prentice-Hall Series in Computational Mathematics, Englewood Cliffs, 1981.

[11] W. Hackbush and U. Trottenberg. *Multigrid Methods.* Springer-Verlag, Berlin, 1982.

[12] N. Hu, X.-Z. Guo, and I. N. Katz. Multi-p preconditioners. *SIAM J. Sci. Comput.*, 18(6):1676–1697, 1997.

[13] N. Hu and I. N. Katz. Multi-p methods: iterative algorithms for the p-version of the finite element analysis. *SIAM J. Sci. Comput.*, 16(6):1308–1332, 1995.

[14] A. C. Nogueira Junior. Relatório anual de atividades - fapesp. Technical report, Universidade Estadual de Campinas (UNICAMP), Campinas, Brazil, 1999.

[15] P. Leinen. Data structures and concepts for adaptive finite element methods. *Computing*, 55:325–354, 1995.

[16] S.F. McCormick. *Multigrid methods*, volume 3 of *Frontiers Series*. SIAM Books, Philadelphia, 1987.

[17] S. Pissanetzky. *Sparse matrix technology*. Academic Press, London, 1984.

[18] U. Rüde. Fully adaptive multigrid methods. *SIAM Journal on Numerical Analysis*, 30(1):230–248, 1993.

[19] S. J. Sherwin and G. Karniadakis. A new triangular and tetrahedral basis for high-order (hp) finite element methods. *Int. J. Numer. Methods Engrg.*, 38:3775–3802, 1995.

[20] B. A. Szabó and I. Babuska. *Finite Element Analysis*. Wiley Interscience, New York, 1991.

[21] J. P. Webb and R. Abouchakra. Hierarchal triangular elements using orthogonal polynomials. *Int. J. Numer. Methods Engrg.*, 38:245–257, 1995.

[22] J. L. Zhu and O. C. Zienkiewicz. A simple error estimator and adaptative procedure for practical engineering analysis. *Int. J. Numer. Methods Engrg.*, 24:337–357, 1987.