Some Finite Element Computational Strategies for Large-Scale Flow Problems in High Performance Computers

L. Catabriga[1,2], C. M. Dias[1], M. A. D. Martins[1], A. L. G. A. Coutinho[1] and J. L. D. Alves[1]
[1]Center for Parallel Computations - Dept. of Civil Engineering
COPPE/Federal University of Rio de Janeiro
21945-970 Rio de Janeiro, Brazil
[2]Dept. of Informatics - Federal University of Espírito Santo
29069-900 Vitória, Espírito Santo, Brazil
Key words: Finite elements methods, reduced integration, iterative solvers, edge-based data structures

Resumo

Neste trabalho apresentamos algumas estratégias computacionais especialmente projetadas para a solução de problemas de Mecânica dos Fluidos de grande porte pelo Método dos Elementos Finitos em computadores de alto desempenho. Estudam-se técnicas de integração reduzida e estruturas de dados baseadas em arestas. Apresentamos também novos pré-condicionadores para esquemas implícitos baseados em agrupamentos de super-arestas. O desempenho dessas novas estratégias computacionais é avaliado nos computadores paralelos Cray J90 e SGI/Origin 2000.

Abstract

In this paper we present some computational strategies tailored for the Finite Element solution of large-scale flow problems in high performance computers. Reduced Integration techniques and edge-based data structures are studied. We also introduce new preconditioners for implicit schemes based on superedges clustering. The performance of these new computational strategies is evaluated on the Cray J90 and SGI/Origin 2000 parallel computers.

1 Introduction

Parallel supercomputers are widely used in large-scale finite element flow simulation. Because the supercomputer architecture is different than uniprocessor computer, the performance of this machine increase when the codes are designed to take advantage of the new design. Low order elements are used in finite element analysis by their simplicity and adaptability to any domain. For these kind of elements we present here some finite element computational strategies for large-scale flow problems [1]. These strategies are separated in two groups. The first one is the element technology. For quadrilateral and hexaedra linear elements, a Gaussian numerical integration is often necessary to evaluate the element matrix. This procedure is responsible for a large amount of computational effort. Then we may use the one-point quadrature or reduced integration to reduce this cost. The reduced integration can yield hourglass modes. To control these spurious modes two techniques are used without compromising accuracy [2] [3]. These reduced integration schemes decrease the computational cost on evaluating such matrices and reduce code complexity, thus improving vectorization and parallelization [4].

The second problem is the solution of the resulting linear systems of equations at each time step or iteration. For most problems of practical interest, solution methods based on direct methods lead to massive storage demands and large computer times. Iterative methods present, comparatively, low storage requirements and, when associated

with suitable preconditioners, provide a powerful computational strategy [5]. In large-scale problems, the solution of nonlinear systems of equations may involve millions of unknowns. Inspired by Finite Volume Methods, edge-based data structures have been introduced for explicit finite element computations of compressible flows on unstructured grids composed by triangles and tetrahedra [6] [7] [8]. They may be also viewed as a representation of the nodal graph of a grid composed by triangles and tetrahedra. Thus, the edge representation is an alternative data structure for computing the global matrix-vector products needed in Krylov space iterative techniques. These schemes have major reduction in indirect addressing operations and memory requirements. However, to reduce the effects of indirect addressing on total CPU costs, Lohner [9] suggested several alternative edge-based data structures. Analysing their main characteristics, Martins et al. [10] have shown that the superedges represent the best compromise. Here we extend the edge concept and present edge-based preconditioners for Krylov update techniques.

The remainder of this paper is organized as follows. In the next section we present the element technology for quadrilateral elements and numerical examples, assessing both accuracy and performance . In the following section we briefly review the edge-based finite element scheme and also present the superedge concept. Next, we present the derivation of clustered edge-by-edge preconditioners. Additional numerical experiments are shown, assessing the performance of the edge-based data structures and preconditioners. Concluding remarks are gathered in the end of the paper.

2 Element Technology

The one-point Gauss quadrature for low-order quadrilaterals and hexaedra can yield spurious oscillations, or hourglass modes. It is necessary to control such hourglass modes in order to generate low cost and accurate solutions. Here we show two reduced integration schemes that are able to produce good solutions for convective-diffusive problems.

2.1 Hourglass Control by Perturbation Techniques

The first scheme studied is widely used in diffusive problems. The idea behind this scheme is to add to the element matrix, evaluated by one-point Gaussian quadrature, stabilization terms derived through the Hu-Washizu variational formulation. These added terms correct the rank defficiency of the element matrix obtained by the reduced integration [2]. In the SUPG formulation for steady-state convection-diffusion [11], the element matrix $\mathbf{K}^e$ can be written as,

$$\mathbf{K}^e = \mathbf{K}_d + \mathbf{K}_c + \mathbf{K}_{pg} \tag{1}$$

where the subscripts d, c and pg correspond respectively to the diffusion, convection and Petrov-Galerkin terms. Then, using the Perturbation Technique each term can be computed as the sum of the reduced integration part and the stabilization part [4],

$$\mathbf{K} = \mathbf{K}_{RI} + \mathbf{K}_{ST} \tag{2}$$

where the subscripts RI and ST refer to the one-point quadrature matrix and to the correction term. Details about the construction of these correction terms can be seen in [2] [4]. The resulting code corresponding to (2) is a single do-loop for all elements, which can be parallelized and/or vectorized. This is much simpler and faster than a standard finite element implementation, which involves several inner loops through the Gauss integration points.

2.2 Alternative Integration

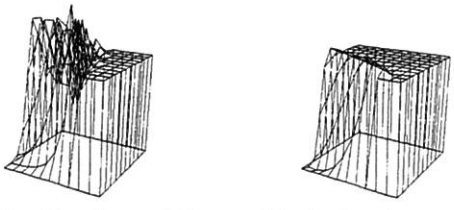
In the scheme proposed by Hansbo [3], each term of $\mathbf{K}$ can be evaluated by an alternative form, where material properties tensor $\mathbf{D}$, the Jacobian $\mathbf{J}$, and the source term are computed at the midpoint of the element. Then, the diffusion term can be written as,

$$\mathbf{K}_d = \int_\Omega \mathbf{B}^T \mathbf{D}_{(0,0)} \mathbf{B} \mathbf{J}_{(0,0)} d\Omega \tag{3}$$

and the other terms are obtained in the same way. $\mathbf{B}$ is the gradient of trial functions. In both schemes, the integrals may be evaluated using a symbolic algebra software like Maple. Studies showed that these schemes have the same accuracy for regular meshes. Implementation also results in a single do-loop.

2.3 Computer Performance

The convergence properties of the stabilization techniques have been extensively studied elsewhere [2] [3]. To illustrate the accuracy of these schemes, we show in Figure 1 the solution of the problem of pure convection of a scalar in a direction skew to the mesh, with unitary constant velocity making 67.5^o with the horizontal, on a 10×10 regular mesh [11]. We may observe that reduced integration without hourglass control produces a high oscillatory solution, whereas both hourglass control schemes yield accurate solutions. The



(a) without hourglass control (b) using hourglass control

Figure 1: Advection skew to the mesh

computer performance of these reduced integration schemes can be observed in Table 1. This Table lists the CPU time in seconds spent in the generation of 262,144 bilinear quadrilateral element matrices in two different machines, a SGI/Indy workstation and a Cray J94 running in vector mode. We may observe that both stabilization techniques were much faster than the standard 2×2 Gaussian integration. Analysing the parallel content of each reduced integration procedure by Cray's *Atexpert* tool, we observed that both schemes were 99.9% parallel.

3 Edge-based Data Structures for Solving Finite Element Equations

3.1 Basic Ideas and Sparse Matrix-Vector Multiplication

The application of implicit finite element formulations to complex flow problems requires the solution of a linear system of equations at each time step and/or iteration, that is,

$$Ax = b \tag{4}$$

PÁGINA FALTANDO

Esse trabalho foi disponibilizado online como parte do esforço de publicação de todo o histórico do evento, entretanto, a versão do trabalho recuperada não continha essa página.

PÁGINA FALTANDO

Esse trabalho foi disponibilizado online como parte do esforço de publicação de todo o histórico do evento, entretanto, a versão do trabalho recuperada não continha essa página.

PÁGINA FALTANDO

Esse trabalho foi disponibilizado online como parte do esforço de publicação de todo o histórico do evento, entretanto, a versão do trabalho recuperada não continha essa página.

PÁGINA FALTANDO

Esse trabalho foi disponibilizado online como parte do esforço de publicação de todo o histórico do evento, entretanto, a versão do trabalho recuperada não continha essa página.

The boundary mesh has 35,307 nodes and 70,937 triangular elements as can be seen in Figure 3. The complete model comprises 448,695 nodes and 2,815,158 tetrahedral elements, which results in 3,314,611 edges, being 5,92% grouped in *superedge*3's and 52,59% in *superedge*6's. We observed in this mesh that in the average, 3.2 tetrahedra share an edge.

Figure 3: Automobile surface mesh

The CPU times in seconds and the relative times for Jacobi-Preconditioned Conjugate Gradient (J-PCG) solution are gathered in Table 4, for the EBE, edges and superedge versions of matrix-vector product. We also show in this Table the number of J-PCG iterations needed to reach convergence for a tolerance of 10^{-6}. The number of iterations is slightly different for each solution scheme due to ill-conditioning. In this problem, the ratio between maximum and minimum values of the system matrix main diagonal is 1.8×10^{7}.

Scheme	Iterations	CPU Time (s)	Relative Time
EBE	25,875	28,934	1.00
edges	25,850	8,689	0.30
superedge	25,818	6,973	0.24

Table 4: Iterations and CPU Times for the automobile solution.

The *Mflop/s* rates and parallel speed-up's for the sparse matrix-vector multiplications needed in J-PCG, for the different data structures are shown in Table 5. The *Mflop/s* were measured on single CPU runs, employing CRAY's *perfview* tool. Parallel speed-up's were measured by CRAY's tool *atexpert* on a J90 with four processors. Results were extrapolated by the same tool to 16 CPU's.

3.3.2 Non-symmetric case - Pollutant Dispersion in Guanabara Bay

In this case, we consider a single time step in the simulation of pollutant dispersion in Guanabara Bay. It is a two-dimensional simulation of the convection-diffusion equation. The mesh has 34,317 triangular elements and 18,481 nodes, and it is shown in Figure 4. It has 52,800 edges, being 78.82 % grouped as *superede3's*. We show in Table 6 the

Scheme	*Mflop/s* single CPU	Speed-up 4 CPU's	Speed-up 16 CPU's
EBE	140.1	3.57	14.17
Edges	105.6	3.60	14.55
superedge	123.8	3.98	15.41

Table 5: Performance data for the automobile solution.

number of GMRES(25) iterations needed to reach convergence for a tolerance of 10^{-6}. The LU and diagonal preconditioners as well as the EBE and superedges schemes are compared. We observed that the solution with less iterations and CPU time corresponds to the superedge LU-preconditioner. The memory required to hold the coefficients for the left LU clustered edge-by-edge preconditioner was 124 *Mwords*. For the edges that could not be grouped into superedges, we employed the corresponding edge-by-edge preconditioner, that needs a storage area of just 45 *Mwords*. Considering that the left LU element-by-element preconditioner requires a storage area of 308 *Mwords*, we observe that the edge-based scheme reduces the storage area to hold the preconditioner roughly by a factor of two.

Figure 4: Guanabara Bay

	Diagonal		LU	
	Iterations	Time(s)	Iterations	Time(s)
Element	312	81.4	80	36.2
Superedge	311	68.4	89	29.2

Table 6: Computer Performance Elements × Superedges in the SGI-Indy.

4 Concluding Remarks

In this paper we presented some strategies to reduce computational costs and memory requirements in large-scale flows simulations. Reduced integration techniques for

quadrilateral elements yielded fast and low complexity codes without compromising accuracy. Several edge-based data structures were employed to implement matrix-vector products needed in Krylov subspace iterative techniques. They are around three times faster than standard EBE techniques in large three-dimensional problems. The resulting code achieved good performance on Cray J90 and SGI/Origin 2000 parallel machines. Clustered edge-based preconditioners were introduced. Numerical experiments have shown that they are faster and require less memory than EBE preconditioners. The strategies studied in this paper are presently being incorporated in simulation programs for porous media and compressible flows, and results will be reported in the near future.

5 Acknowledgments

Computer time on the CRAY J90 was provided by the Center for Parallel Computations of the Federal University of Rio de Janeiro (NACAD). This research is partially supported by a RECOPE/FINEP/MCT research grant. We are indebted to SGI/Cray Research Division by computer time and support in the utilization of their Origin 2000 machine at Eagan, MN, USA.

References

[1] Aliabadi, S., and Tezduyar, T., Parallel Fluid Dynamics Computations in Aerospace Applications, Int. J. for Num. Meth. in Fluids, 21 (1995), pp. 783-805.

[2] Belytschko, T., Ong, J. S. J., Liu, W. K., and Kennedy, J. M., Hourglass Control in Linear and Nonlinear Problems, Comp. Methods Appl. Mech. Engng., 43, (1984), pp. 251-276.

[3] Hansbo, P., A New Approach to Quadrature for Finite Elements Incorporating Hourglass Control as a Special Case, Comp. Methods Appl. Mech. Engng., to appear, 1998.

[4] Dias, C. M., and Coutinho, A. L. G. A., Integração Reduzida para Problemas Advectivos-difusivos Escalares Discretizados pela Formulação SUPG com Operador de Captura de Descontinuidades, Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería, Vol. 14, 2, 145-166 (1998).

[5] Coutinho, A. L. G. A, and Alves, J. L. D., Parallel Finite Element Simulation of Miscible Displacements in Porous Media, SPE Journal, 1(4), SPE37399, (1996), pp. 487-500.

[6] Peraire, J., Morgan, K., and Peiro, J., Unstructured Grid Methods for Compressible Flows, AGARD Special Course on Unstructured Grid Methods for Advection Dominated Flows, 787, 5.1-5.39 (1992).

[7] Baum, J. D., Luo, H., and Lohner, R., Edge-based Finite Element Scheme for Euler Equations, AIAA J., 32, 1183-1190 (1994).

[8] Peraire, J., Lyra, P. R. M., Morgan, K., and Peiro, J., TVD Algorithms for the Solution of the Compressible Euler Equations on Unstructured Meshes, Int. J. Num. Meth. Fluids, 19, 849-863 (1994).

[9] Lohner, R., Edges, Stars, Superedges and Chains, Comp. Meth. Appl. Mech. and Engng, 111, 255-63, 1994.

[10] Martins, M. A. D., Coutinho, A. L. G. A., and Alves, J. L. D., Parallel Iterative Solution of Finite Element Systems of Equations Employing Edge-Based Data Structures, SIAM Conf. on Parallel Processing for Scientific Computing, Minneapolis, USA, March, 1997.

[11] Brooks, A. N., and Hughes, T. J. R., Streamline Upwind Petrov-Galerkin Formulation for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equation, Comp. Methods Appl. Mech. Engng., 32, (1982), pp. 199-259.

[12] Catabriga, L., Martins, M. A. D., Coutinho, A. L. G. A., and Alves, J. L. D., Clustered Edge-Based Preconditioners for Non-Symmetric Finite Element Equations, Proc. 4^{th} World Congress on Computational Mechanics, CD-ROM, Ed. S.Idelsohn, E. Oñate, E. Dvorkin, CIMNE, Barcelona, 1998.