A linelet preconditioner for incompressible flow solvers

Orlando Soto, Rainald Löhner and Fernando Camelli
SCS/Laboratory for Computational Fluid Dynamics, George Mason University, Fairfax, VA, USA

Received March 2002
Revised August 2002
Accepted August 2002

Keywords *Navier-Stokes equations, Finite elements, Incompressible flow*

Abstract *A parallel linelet preconditioner has been implemented to accelerate finite element (FE) solvers for incompressible flows when highly anisotropic meshes are used. The convergence of the standard preconditioned conjugate gradient (PCG) solver that is commonly used to solve the discrete pressure equations, greatly deteriorates due to the presence of highly distorted elements, which are of mandatory use for high Reynolds-number flows. The linelet preconditioner notably accelerates the convergence rate of the PCG solver in such situations, saving an important amount of CPU time. Unlike other more sophisticated preconditioners, parallelization of the linelet preconditioner is almost straighforward. Numerical examples and some comparisons with other preconditioners are presented to demonstrate the performance of the proposed preconditioner.*

1. Introduction

Due to the widely varying scales characterizing fluid dynamics problems, it is common to employ anisotropic meshes to resolve the boundary layers arising in viscous flows at high Reynolds numbers. Streamwise length scales are often three or four orders of magnitude larger than the normal length scales (Hassan *et al.*, 1990; Martin and Löhner, 1992; Mavriplis, 1998a, b). This mesh stretching induces stiffness to the numerical problem, leading to ill-conditioned linear systems of equations. It is well known that the convergence rate of any iterative solver strongly depends on the condition number of the system matrix A. Therefore, anisotropic grids considerably deteriorate the convergence rate of such schemes.

In the present work, a linelet preconditioning matrix P has been implemented in order to improve the convergence rate of the preconditioned conjugate gradient (PCG) solver for anisotropic meshes. The idea is to construct "lines" in the mesh in the normal direction to the grid stretching. Then, the preconditioning matrix is built by assembling the diagonal entries of the system matrix ($A_{i,i}$), and the non-diagonal entries ($A_{i,j}\ i \neq j$) of the edges belonging to these linelets. An important condition for the final structure of the preconditioner is that a nodal point can only belong to one linelet, i.e. a linelet does not cross any other one. Then, if the nodal points are renumbered following the linelets, the preconditioning matrix associated to the degrees of freedom belonging to a line is tri-diagonal. In addition, as it will be shown later, diagonal preconditioning automatically holds for the degrees of freedom associated to the points of the mesh that do not belong to any linelet.

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 13 No. 1, 2003
pp. 133-147

0961-5539
DOI 10.1108/09615530310456796

Linelet preconditoners have been used to accelerate the convergence rate of flow schemes to steady state (Beam and Warming, 1978; Briley and McDonald, 1977; Hassan *et al.*, 1990; Martin and Löhner, 1992; Mavriplis, 1998a, b; Rogers *et al.*, 1985) In this work, they are used as a general preconditoner to solve ill-conditioned systems of equations induced by highly stretched grids.

The present paper is organized as follows. Section 2 presents the schemes used to discretize the Navier-Stokes equations. Section 3 presents the method implemented to construct the linelets. Some examples and comparisons with other preconditioners are shown in Section 4. Finally, some conclusions are drawn in Section 5.

2. Finite element discretizations

The Navier-Stokes describing an incompressible flow may be written as:

$$\frac{\partial u}{\partial t} + (u\cdot\nabla)u - \nu\Delta u + \nabla p = f \ \text{ in } \ \Omega\times(0,t_f), \tag{1}$$

$$\nabla\cdot u = 0 \ \text{ in } \ \Omega\times(0,t_f) \tag{2}$$

where Ω is the flow domain, t is the time variable, $(0,t_f)$ the time interval for the simulation, u the velocity field, ∇ the gradient operator, ν the kinematic viscosity, Δ the Laplacian operator, p the pressure and f the external body forces (i.e. the gravity and the Boussinesq forces).

Let σ be the viscous stress tensor and n the unit outward normal to the boundary $\partial\Omega$. Denoting by an overbar prescribed values, the boundary conditions for equations (1) and (2) to be considered here are:

$$\begin{aligned} u = \bar{u} \ \text{ on } \ \Gamma_{\mathrm{du}}, \ \ p = \bar{p} \ \text{ and } \ n\cdot\sigma = \bar{t} \ \text{ on } \ \Gamma_{\mathrm{nu}}, \\ u\cdot n = \bar{u}_n, \ \ n\cdot\sigma\cdot g_1 = \bar{t}_1 \ \text{ and } \ n\cdot\sigma\cdot g_2 = \bar{t}_2 \ \text{ on } \ \Gamma_{\mathrm{mu}} \end{aligned} \tag{3}$$

for $t\in(t_0,t_f)$. The boundary $\partial\Omega$ has been considered split into three sets of disjoint components Γ_{du}, Γ_{nu} and Γ_{mu}, the latter being the part where mixed conditions are prescribed: normal velocity and the tangent stresses. Vectors g_1 and g_2 (for the 3D case) span the space tangent to Γ_{mu}. Finally, Γ_{du} and Γ_{nu} are the two disjoint components of $\partial\Omega$ where Dirichlet and Neumann boundary conditions for the velocity are prescribed, respectively. Suitable initial conditions have to be added to equations (1) and (2).

For the solution of the incompressible flow, two schemes were implemented: an explicit fractional step type formulation, and an implicit monolithic orthogonal subscale stabilization (OSS) scheme. In the first approach (Löhner *et al.*, 1997; Ramamurti and Löhner, 1996) the pressures are implicitly integrated, correctly reflecting the infinite propagation speed of sound. A complete time step for such a method consists of the following three parts:

(a) *Convective-diffusive prediction*: $u^n \rightarrow u^*$

$$\left[\frac{1}{\delta t} - \nu\Delta\right](u^* - u^n) = \nu\Delta u^n - (u^n \cdot \nabla)u^n - \nabla p^n + f^n \tag{4}$$

(b) *Pressure correction*: $p^n \rightarrow p^{n+1}$

$$\nabla \cdot u^{n+1}; \quad \frac{u^{n+1} - u^*}{\delta t} + \nabla(p^{n+1} - p^n) = 0 \tag{5}$$

which results in

$$\delta t \Delta(p^{n+1} - p^n) = \nabla \cdot u^* \tag{6}$$

(c) *Velocity correction*: $u^* \rightarrow u^{n+1}$

$$u^{n+1} = u^* - \delta t \nabla(p^{n+1} - p^n) \tag{7}$$

Then, from the given or computed u^n and p^n, one can obtain u^* from equation (4), p^{n+1} from equation (6), and, finally, u^{n+1} from equation (7). In the earlier equations, as in the rest of the paper, δt is the time step size. The boundary conditions of the earlier scheme are the standard ones: velocities prescribed at inflow and solid boundaries, and the pressures set at outflow boundaries.

The spatial discretization of the earlier equations is performed by the standard Galerkin method, using a lumped mass approximation for the temporal terms. The convective instabilities are treated by mean of a second order Roe type solver (Löhner *et al.*, 1997) combined with a MUSCL limiting procedure to obtain a monotonic scheme (van Leer, 1974). The incompressibility or pressure stabilization (projection equation (6)), is treated by adding to the final Galerkin variational form a consistent fourth order pressure term.

For the implicit approach, a OSS type formulation was adopted (Codina, 2000, 2002; Soto and Löhner, 2001; Soto *et al.*, 2001. The final variational form of such scheme may be written as follows: given u_h^n, find $(u_h^{n+1}, p_h^{n+1}, \pi_h^{n+1}, \xi_h^{n+1})$ in $V_h \times Q_h \times \tilde{V}_h \times \tilde{V}_h$ such that

$$\begin{aligned}
&\frac{1}{\delta t}\left(u_h^{n+1,i} - u_h^n, v_h\right) + \left(u_h^{n+\theta,i-1} \cdot \nabla u_h^{n+\theta,i}, v_h\right) + \left(\nu \nabla u_h^{n+\theta,i}, \nabla v_h\right) \\
&\quad - \left(p_h^{n+1,i-1}, \nabla \cdot v_h\right) + \left(\tau\left(u_h^{n+\theta,i-1} \cdot \nabla u_h^{n+\theta,i} - \pi_h^{n+\theta,i-1}\right), u_h^{n+\theta,i-1} \cdot \nabla v_h\right) \\
&\quad = \left(f^{n+\theta}, v_h\right) + \left(\sigma^{n+\theta,i-1} \cdot n, v_h\right)\Gamma_{\mathrm{nu}}
\end{aligned} \tag{8}$$

$$\delta t\left(\nabla p_h^{n+1,i} - \nabla p_h^{n+1,i-1}, \nabla q_h\right) + \left(\tau\left(\nabla p_h^{n+1,i} - \xi^{n+1,i-1}\right), \nabla q_h\right) = -\left(\nabla \cdot u_h^{n+1,i}, q_h\right) \tag{9}$$

$$\left(\pi_h^{n+\theta,i}, \tilde{v}_h\right) = \left(u_h^{n+\theta,i} \cdot \nabla u_h^{n+\theta,i}, \tilde{v}_h\right) \tag{10}$$

$$\left(\xi_h^{n+1,i}, \tilde{v}_h\right) = \left(\nabla p_h^{n+1,i}, \tilde{v}_h\right) \tag{11}$$

$\forall (v_h, q_h, \tilde{v}_h, \tilde{v}_h) \in V_h \times Q_h \times \tilde{V}_h \times \tilde{V}_h$, which represent the finite element (FE) spaces where the different fields are interpolated. In the earlier equations, the superscript i stands for the sub-iteration number into the timestep, θ is the trapezoidal rule parameter (temporal discretization), the subscript h refers to the discrete problem, π_h and ξ_h are the projection of the convective term and the gradient of pressures into the FE space, respectively, and the functional form (a, b) is defined as $(a, b) = \int_\Omega a \cdot b \, \mathrm{d}\Omega$. Finally, τ is the local time step for a Courant number of one, which is computed element by element using the following expression:

$$\tau = \frac{h^2}{4\nu + 2\|u_e\|h} \tag{12}$$

where h is the element size, and $\|u_e\|$ the elemental velocity norm. In Soto *et al.* (2001) a detailed deduction of the earlier formulation is described.

3. Linelet preconditioner

As mentioned before, linelets are built in the normal direction to the grid stretching. Let A be the matrix associated to the variational form (equation (9)) or, which is the same, to the system of equations resulting from the discretization of the second step (equation (6)) for the projection scheme. The linelet preconditioning matrix P is defined as:

$$P_{i,i} = A_{i,i}; \qquad P_{i,j} = A_{i,j} \;\; i \neq j \text{ if } i,j \text{ belongs to a linelet}$$
$$P_{i,j} = 0 \;\; i \neq j \text{ if } i,j \text{ does not belong to any linelet} \tag{13}$$

After a nodal renumeration following the linelets, the structure of P consists of sets of decoupled tri-diagonals. Each set is associated to a linelet. For the nodal points that do not belong to any linelet a diagonal structure is obtained.

Figure 1 shows an example of the structure of P, for the case of two linelets. In this example, five nodal points (four edges) belong to the first linelet, and four points (three edges) to the second one. The rest of the nodes do not belong

to any linelet. This set of decoupled tri-diagonal equations allows a fast solution of the preconditioning equations by a direct method (e.g. via the Thomas Algorithm).

3.1 Linelet construction

In order to obtain a good preconditioner, the construction of a proper set of linelets is crucial. The idea is to start with a set of source points in the mesh. For each source point i, the surrounding edge (i, j) with minimum length is chosen to grow the linelet. Clearly, (i, j) is the direction normal to the maximum mesh stretching, and therefore, the direction of the highest stiffness. The new set of point j's will be the new set of source points. The procedure is repeated until no more linelets can be grown. A more detailed description of the methodology is presented below:

(1) Choose a set of linelet sources: these are the initial linelet points. This task is accomplished by marking the points surrounded by sufficiently distorted edges, i.e. the points in which the ratio between its smallest surrounding edge over its largest one is smaller than a given value. Then, the sources belonging to the same linelet are eliminated as follows (see Figure 1 for guidance). For each linelet source i, its minimum length connection (point j), and the minimum length connection of j (point k) are found. Then two situations can occur: if k and i are different, i is eliminated because the edge (j, k) is smaller than the edge (i, j), and therefore the linelet probably starts in k and goes through j and i. The second option is when i and k are the same point, then the source point can be i or j. In this case, the point with the smallest second smallest surrounding edge has to be eliminated, because the linelet automatically will include this edge as it grows.

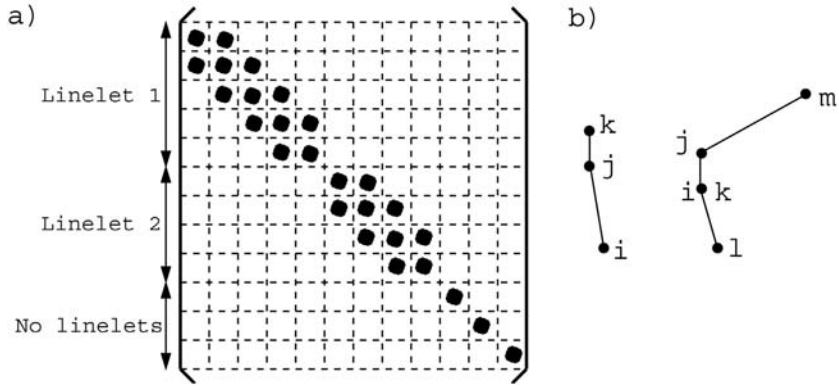

Figure 1.
From left to right: (a) structure of the preconditioning matrix, and (b) guidance for the source elimination process

(2) Linelets growing: for each point i marked as a linelet source, its closest neighbor j that does not belong to any other linelet and that fulfills the condition:

$$\frac{l_{ij}}{l_i^m} \leq \epsilon \tag{14}$$

is chosen. Here l_{ij} denotes the length of the edge (i, j), and l_i^m the maximum length of the edges surrounding i-j will be the new point in the corresponding linelet, and the new source point for such a linelet. After all linelets are updated by the described procedure, this step is repeated, taking the new linelet points as new source points (points j). The algorithm stops when no further linelets can be grown.

(3) Growing in the other direction: repeat step (2) but starting with the second close neighbor of the initial source points. This is to take into account the regions where linelets can grow in two directions (i.e. wake regions).

(4) Nodal point renumeration: the nodal points must be renumbered following the linelets. This is crucial to obtain the desired tri-diagonal structure of P. First, the nodes of a linelet are renumbered from one end to the other. Thereafter, a second linelet is renumbered, and so on until all the linelets have been covered. Finally, the rest of points have to be renumbered too.

3.2 PCG solver

The final objective is to use P to precondition the system of linear equations resulting from the FE discretization of the incompressibility or pressure constraint (equation (9)). The system to be solved can be written as:

$$Ax = b \tag{15}$$

where A is the system matrix, x the unknown vector, and b the right hand side term. If a standard PCG solver is used (Saad, 1996), a system of equations of the form:

$$Pz = r \tag{16}$$

has to be solved in each iteration, where $r = b - Ax$ is the residual vector. The solution of the previous system of equations was implemented using the Thomas Algorithm in order to exploit the tri-diagonal structure of P. The scheme is basically a LU decomposition applied to a tri-diagonal (or block tri-diagonal) matrix as follows: $(P = LU)$:

$$\begin{bmatrix} D_1 & F_1 & 0 & \dots & 0 \\ E_2 & D_2 & F_2 & \dots & 0 \\ 0 & E_3 & D_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & E_n & D_n \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ L_2 & I & 0 & \dots & 0 \\ 0 & L_3 & I & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & L_n & I \end{bmatrix} \times \begin{bmatrix} U_1 & F_1 & 0 & \dots & 0 \\ 0 & U_2 & F_2 & \dots & 0 \\ 0 & 0 & U_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 0 & U_n \end{bmatrix} \tag{17}$$

It is easy to corroborate that L and U can be computed by using the following algorithm:

Thomas Algorithm

$U_1 := D_1$

do $i := 2, n$

Obtain U_{i-1}^{-1}

$L_i := E_i U_{i-1}^{-1}$

$U_i := D_i - L_i F_{i-1}$

end do

Once L and U have been obtained, the solution of the system $Pz = r$ is accomplished by performing the following forward and backward substitutions:

$$y_i = b_i - L_i y_{i-1}, \quad (y_0 = 0) \quad i = 1, \dots, n; \quad r_i = U_i^{-1}\left[y_i - F_i r_{i+1}\right], \quad (r_{n+1} = 0) \quad i = n, \dots, 1 \tag{18}$$

As shown before, the tri-diagonal subsystems ($Pz = r$) are totally decoupled, i.e. each tri-diagonal subsystem is associated with its respective linelet (Figure 1). Therefore, they can be solved in an independent manner, which constitutes an extremely important property for parallelization purposes. Groups of linelets (or tri-diagonal subsystems) can be sent to each processor in

order to be solved. The number of linelets in each group depends on the number of available processors.

As a final remark, it is important to note that the *LU* decomposition is done only once. In each PCG iteration, just the forward and backward substitutions (equation (18)) must be performed.

4. Numerical examples

Some numerical results, and some comparisons with diagonal and incomplete LU decomposition (ILU) preconditioners (Saad, 1996) are presented later. Both formulations, the projection scheme and the OSS implicit methodology, showed important savings in the solution of the pressure (equation (9)), and therefore in the total CPU time, when the linelet preconditioner was used on problems with highly stretched meshes.

4.1 Flow over a backward-facing step – Implicit OSS scheme

The first example that was been considered is the 2D laminar flow over a backward-facing step. The computational domain is shown in Figure 2(a). A fully parabolic velocity profile was prescribed at the inflow, a no-slip condition at the walls ($u = 0$), and the pressures were set to zero at the outflow. The Reynolds (Re) number of the problem based on the total cross section of the channel, and on the average velocity at inflow was $\mathrm{Re} = 10^3$.

The OSS implicit technique was used for the spatial discretization, and the non-linear terms in equation (8) were treated via a Picard linearization. The final discrete system of equations was solved with a standard GMRES method using diagonal preconditioning (Saad, 1996). For the pressure Laplacian (equation (9)), a standard PCG algorithm with three different preconditioners was tested: the standard diagonal one, the standard ILU(0) (Saad, 1996) preconditioner, and the linelet preconditioner described in this work with $\epsilon = 1$ for the linelet growing criterium (equation (14)). The set of linelets for this example may be observed in Figure 2(b). The Courant number of the problem

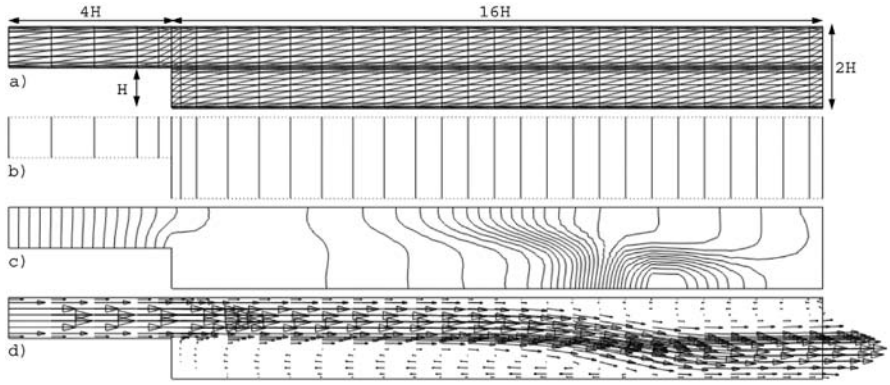

Figure 2.
From top to bottom: (a) computational domain and mesh of 990 linear elements and 544 nodal points, (b) set of linelets, (c) pressure field $\mathrm{Re} = 1,000$: peaks (-0.06147, 0.0231), and (d) velocity field $\mathrm{Re} = 1,000$: main vortex length 14 H

was set to 10, and two sub-iterations were performed at each time step to arrive at the steady state solution. Residuals were reduced to five orders of magnitude.

The steady state pressure distribution and the velocity field can be observed in Figure 2(c) and (d), respectively. These results are in good agreement with other numerical and experiment data (Armaly *et al.*, 1983; Codina, 1993; Kim and Moin, 1985; Kim, 1987; Sohn, 1986; Soto and Codina, 1997). The experimental value of the main vortex reattachment-point is around 14.3 H (measured from the step), while the present numerical result is around 14 H. This is an excellent agreement taking into account the 3D effects that take place in the experiment.

Figure 3 shows the number of PCG iterations in each time step sub-iteration (two per time step), for the different preconditioners. A tolerance of 10^{-4} was taken as solver termination criterion. Clearly, the linelet preconditioner needs less iterations than the other schemes to converge. However, the CPU time savings are not so impressive. The linelet preconditioner saved 25 per cent of CPU with respect to the diagonal one, and only a 3 per cent with respect to the ILU(0) preconditioner. This is due to the additional operations required for the construction and linelet use, the size of the problem, and the rather mild stretching of the elements (1 : 20 was the maximum aspect ratio). As will be observed later, for 3D problems with many more elements and larger aspect ratios, the CPU savings become more important. All the simulations were run on four R-10000 SGI processors.

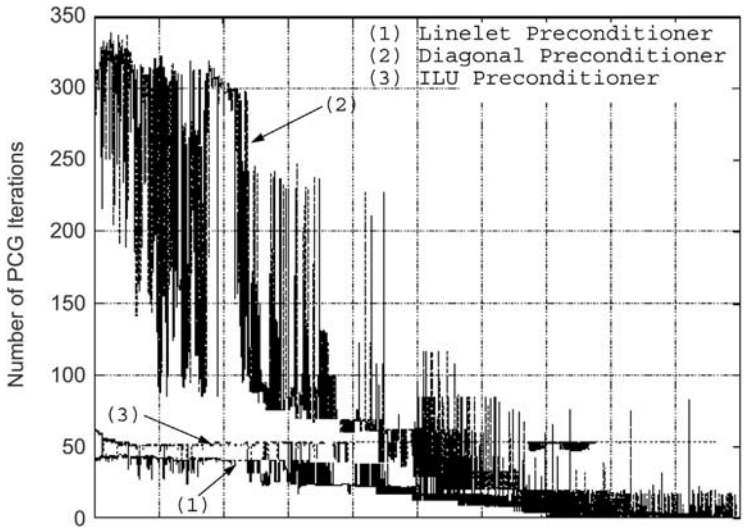

Figure 3.
Backward-facing step. PCG iterations in each time step sub-iteration

4.2 Flow over a bump – explicit fractional step scheme

In this second numerical example, the high Reynolds flow ($\mathrm{Re} = 10^5$) over a bump is computed. A cut of the 3D mesh is shown in Figure 4. The boundary conditions of the example were: velocity prescribed to $u = (1, 0, 0)$ at inflow, normal velocities set to zero at the top and bottom walls, and $u = (0, 0, 0)$ on the bump surface. At outflow, the pressures were imposed to zero. The explicit fractional step method described earlier, combined with the standard Baldwin Lomax turbulence model (Camelli and Löhner, 2002), was used to carry out this numerical example.

The steady-state pressure field is shown in Figure 4, and the comparison between the number of PCG iterations performed by the linelet and diagonal preconditioners is shown in Figure 5. The linelet preconditioner reduced the PCG iterations in each time step at least 50 per cent, which allowed to attain a total CPU time reduction of 40 per cent.

4.3 Cube in a channel flow – implicit OSS scheme

The example considers the flow around a cube in a channel at $\mathrm{Re} = 4 \times 10^5$. The hybrid Baldwin Lomax-Smagorinsky model (Camelli and Löhner, 2002) was utilized to take into account the turbulence effects. The initial and boundary conditions were taken from the experimental data (Krajnović and Davidson, 2001). Figure 6 shows details of the FE surface and volume mesh. In the same figure the positions where experimental velocity profiles are available are also shown. It is important to note the high stretching of the elements in the boundary layer region (aspect ratio of 1 : 50,000). For this case it was found that the ILU preconditioner was totally inadequate. It was never able to decrease the

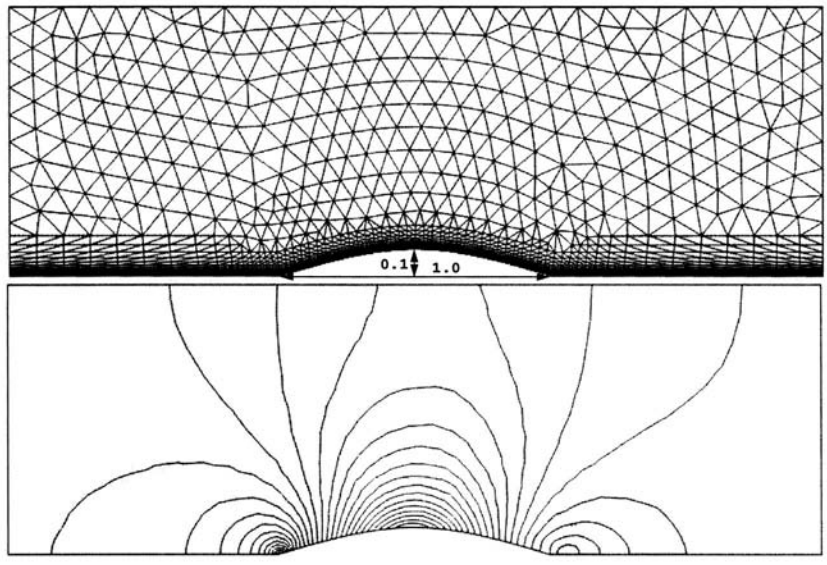

Figure 4.
From top to bottom: mesh of 23,785 linear elements and 5,674 nodal points. Pressure field, peaks (− 0.3590, 0.2816)

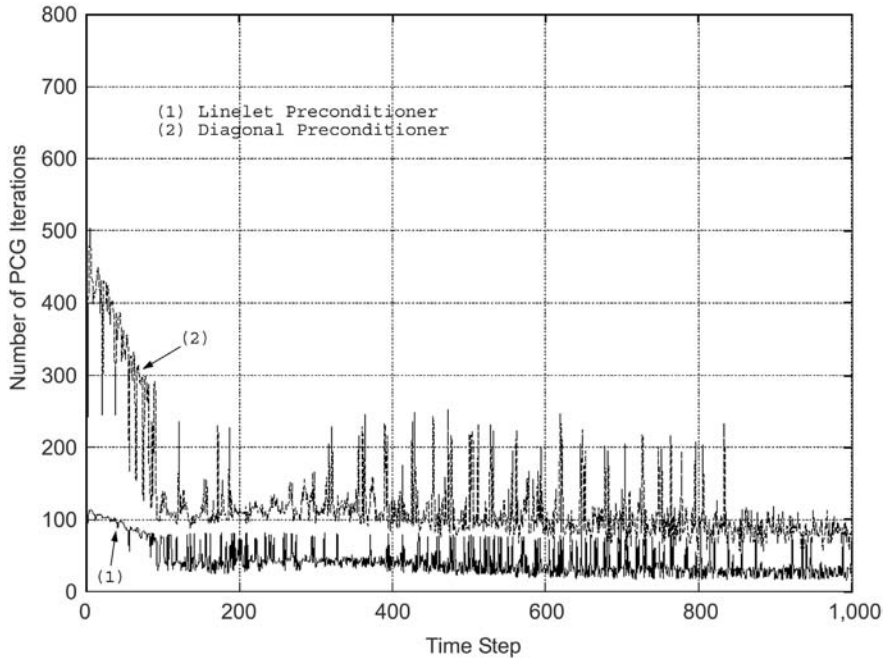

Figure 5.
Flow over a bump. PCG iterations in each time step

residual of the Laplacian equation (9) for four orders of magnitude (the specified tolerance), and, hence, the PCG solver always finished after the allowed maximum number of iterations were performed. A similar behavior was observed for the diagonal preconditioner.

On the other hand, the linelet preconditioner showed a very good convergence rate for this case. It was always able to reduce the residual of the PCG solver four orders of magnitude in 20-170 iterations, as shown in Figure 7. In Figure 8, a comparison of the mean numerical velocity profiles and the experimental ones by Krajnović and Davidson (2001) and Martinuzzi and Tropea (1993) is shown. As may be observed, they agree fairly well.

Conclusions

A linelet preconditioner has been presented and used to solve the discrete pressure Laplacian equation obtained, when stabilized or fractional step methods are used to solve incompressible flow problems. For all the cases considered, the linelet preconditoner reduced the number of PCG iterations with respect to other common preconditioners (Diagonal, ILU). This behavior was most notable for meshes with highly stretched elements. For isotropic meshes, the numerical experience indicates that such a reduction is not important, and an ILU type preconditioner is superior. In addition, it is important to remark that due to the additional operations that are necessary for the linelet preconditioner, it is most effective in terms of CPU time savings for large scale

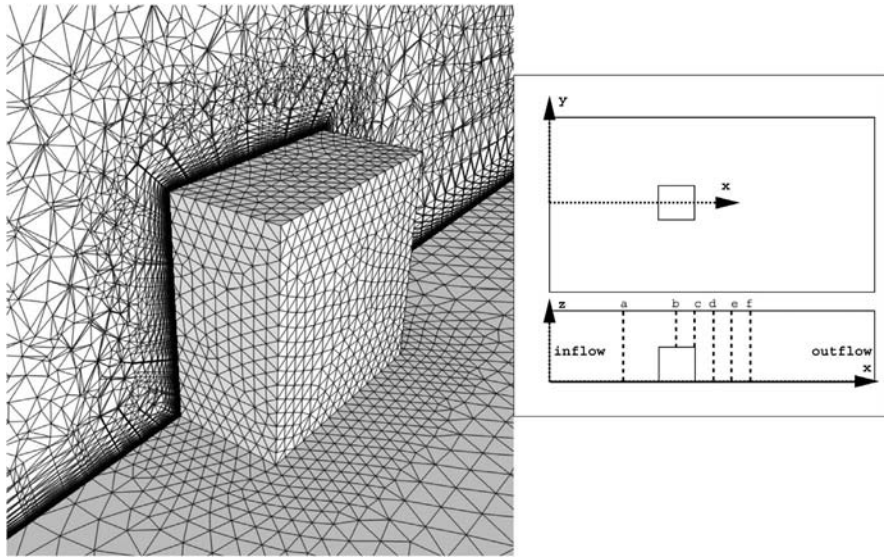

Figure 6.
From left to right: detail of the 2,062,908 linear elements and 357,859 nodal points mesh. Locations of the experimental mean velocity profiles

problems. We also remark that in previous work some comparisons have been performed with a multigrid preconditioner for the pressure Laplacian equation (9). Experience indicates that even though multigrid reduces the solver iterations, the CPU time for large scale simulations is almost the same as using

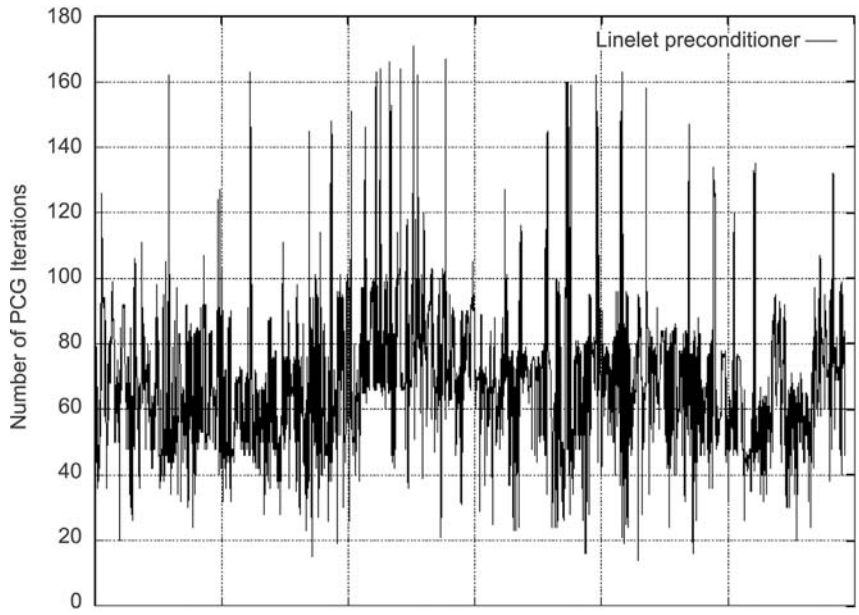

Figure 7.
Cube in a channel. PCG iterations in each time step sub-iteration

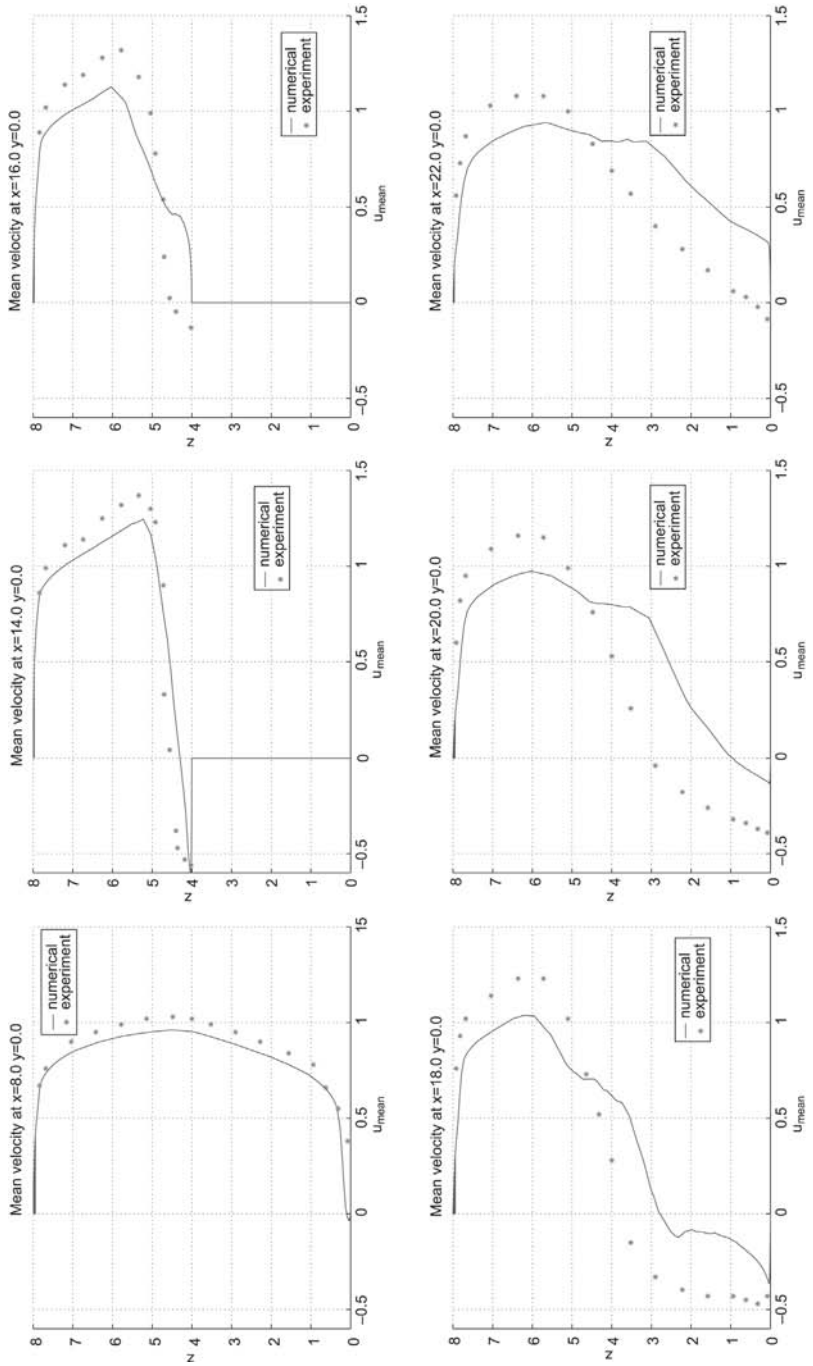

Figure 8.
Comparisons among the numerical and experimental mean velocity profiles at a, b, c, d, e and f, respectively (see Figure 6)

a standard diagonal preconditioner, due to the overhead of operations in the multigrid solver (Waltz, 2000).

References

Armaly, B., Durst, F., Pereira, J. and Schonung, B. (1983), "Experimental and theoretical investigation of backward-facing step flow", *J. Fluid Mech.*, Vol. 127, pp. 473-6.

Beam, R.M. and Warming, R.F. (1978), "An implicit finite difference algorithm for hyperbolic systems in conservation-law form", *J. Comp. Phys.*, Vol. 22, pp. 87-110.

Briley, W.R. and McDonald, H. (1977), "Solution of the multi-dimensional compressible Navier-Stokes equations by a generalized implicit method", *J. Comp. Phys.*, Vol. 21, pp. 372-97.

Camelli, F. and Löhner, R. (2002), "Combining the Baldwin Lomax and Smagorinsky turbulence models to calculate flows with separation regions", *AIAA*-2002-0426.

Codina, R. (1993), "A finite element formulation for viscous incompressible flows", *CIMNE Monograph* No. 16.

Codina, R. (2000), "Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods", *Comp. Meth. Appl. Mech. Eng.*, Vol. 190, pp. 1579-99.

Codina, R. (2002), "Analysis of a stabilized finite element approximation of the Ossen equations using orthogonal subscales", *Numerische Mathematik* (in press).

Hassan, O., Morgan, K. and Peraire, J. (1990), "An implicit finite element method for high speed flows", *AIAA* Paper 90-0402.

Kim, J. and Moin, P. (1985), "Application of a fractional-step method to incompressible Navier-Stokes equations", *J. Comp. Phys.*, Vol. 59, pp. 308-23.

Kim, S. (1987), "A finite element computational method for high Reynolds number laminar flows", *NASA CR*-179135.

Krajnović, S. and Davidson, L. (2001), "Large Eddy simulation of the flow around a three-dimensional bluff body", *AIAA* Paper 2001-0432.

Löhner, R., Yang, C., Gñate, E. and Idelsohn, S. (1997), "An unstructured grid-based, parallel free surface solver", *AIAA*-97-1830.

Martin, D. and Löhner, R. (1992), "An implicit linelet-based solver for incompressible flows", *AIAA*-92-0668.

Martinuzzi, R. and Tropea, C. (1993), "The flow around surface-mounted prismatic obstacles placed in a fully developed channel flow", *ASME J. Fluid Mech.*, Vol. 115, pp. 85-91.

Mavriplis, D.J. (1998a), "Directional agglomeration multigrid techniques for high reynolds number viscous flow solvers", *AIAA* Paper 98-0612.

Mavriplis, D.J. (1998b), "On convergence acceleration techniques for unstructured meshes", *AIAA* Paper 98-2966.

Ramamurti, R. and Löhner, R. (1996), "A parallel implicit incompressible flow solver using unstructured meshes", *Computer and Fluids*, Vol. 5, pp. 119-32.

Rogers, S.D., Kwak, D. and Kaul, U. (1985), "On the accuracy of the pseudo-compressibility method in solver the incompressible Navier-Stokes equation", *AIAA* Paper 85-1689.

Saad, Y. (1996), *Iterative Methods for Sparse Linear Systems*, PWS Pub. Co., Boston.

Sohn, J. (1986), *Evaluation of Fidap on Some Classical Laminar and Turbulent Benchmarks*, Universities Space Research Association, Fluid Dynamics Branch, Nasa/Marshall Space Flight Cente, Huntsville, AL 35812.

Soto, O. and Codina, R. (1997), "Estabilización de la solución por elementos finitos de problemas de flujo incompresible con rotación, turbulencia, superficie libre y temperatura", *CIMNE Monograph* No. 41.

Soto, O. and Löhner, R. (2001), "An implicit monolithic time accurate finite element scheme for incompressible flow problems", *AIAA*-2001-2616.

Soto, O., Löhner, R., Cebral, J. and Codina, R. (2001), "A time-accurate implicit-monolithic finite element scheme for incompressible flow problems", *Proceedings of ECCOMAS CFD 2001*, 4-7 September 2001, Swansea, Wales.

van Leer, B. (1974), "Towards the ultimate conservative scheme. II. Monotonicity and conservation combined in a second order scheme", *J. Comp. Phys.*, Vol. 14, pp. 361-70.

Waltz, J. "Unstructured multigrid for time dependent incompressible fluid flow", PhD thesis, School of Computational Sciences of George Mason University, 4400 University Drive, MS 4C7 - George Mason University - Fairfax, VA 22030-24444.