ELSEVIER Journal of Computational and Applied Mathematics 82 (1997) 379–388

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Multigrid methods for compressible Navier–Stokes equations in low-speed flows

J. Steelant*, E. Dick, S. Pattijn

Department of Mechanical and Thermal Engineering, Universiteit Gent, Sint Pietersnieuwstraat 41, 9000 Gent, Belgium

Received 27 August 1996; received in revised form 1 April 1997

Abstract

The multigrid performance of pointwise, linewise and blockwise Gauss–Seidel relaxations for compressible laminar and turbulent Navier–Stokes equations is illustrated on two low-speed test problems: a flat plate and a backward facing step. The line method is an Alternating Symmetric Line Gauss–Seidel relaxation. In the block methods, the grid is subdivided into geometric blocks of $n \times n$ points with one point overlap. With in the blocks, the solution is obtained by a direct method or with an alternating modified incomplete lower–upper decomposition. The analysis is focused on flows typical for boundary layers, stagnation and recirculation regions. These are characterized by very small Mach numbers, high Reynolds numbers and high mesh aspect ratios.

Keywords: Multigrid methods; Navier–Stokes equations; Turbulence

AMS classification: 65N20; 76G15

1. Introduction

In recent times, conjugate gradient (CG) and GMRES-methods have gained much popularity as fast solvers for Euler and Navier–Stokes equations [5]. These techniques have become serious competitors to the more classic multigrid methods. Viewed as linear system solvers, the three techniques can be seen as different ways to accelerate convergence of basic iterative solvers. For medium-sized linear problems, all three techniques are about equally effective. For sufficiently large problems, multigrid methods, in principle, are faster since the convergence rate is, ideally, independent of the problem size [9]. This last statement is valid for optimal preconditioners (or smoothers) for the three methods. This has been proved numerically by several researchers [1, 2].

*Corresponding author. Tel.: + 32-9-2643296; fax: + 32-9-2643586; e-mail: Johan.Steelant@rug.ac.be.

PII S0377-0427(97)00080-0

Furthermore, GMRES and CG are linear techniques while MG can be applied to nonlinear problems. These linear methods have further the disadvantage of requiring a large amount of memory.

Most multigrid methods for compressible Navier–Stokes equations in the past have been formulated on the basis of simple smoothers, like pointwise Gauss–Seidel methods. Especially for turbulent applications more complex smoothers have not attracted wide attention up to now. This includes our own work [3, 6]. It is clear that for optimal performance much more powerful smoothers are necessary. Especially for low Mach number and high Reynolds number flows, pointwise methods degrade in performance. The degradation due to high Reynolds number was already examined by Kettler [4] and Wesseling [8] based on their study on scalar convection–diffusion equations for high Reynolds number.

From their smoothing analysis it follows that alternating symmetric line Gauss–Seidel (ASLGS) relaxation and incomplete lower–upper (ILU) decomposition methods are the most robust and efficient smoothers. Degradation due to low Mach number cannot be studied by means of scalar model equations as they neglect the presence of different velocity scales, typical for Navier–Stokes equations. Especially, for low-speed flows, there is a large discrepancy between the convective and acoustic velocities. As a consequence the stiffness of the system increases, resulting in a supplementary convergence deceleration.

2. Smoothing characteristics

In order to study the Mach effect, the smoothing analysis of Kettler and Wesseling for scalar model equations was extended by the authors to collective variants of relaxation methods applied to linearized systems of Navier–Stokes equations [7]. By a collective variant is meant that the Navier–Stokes equations are treated as a coupled 4×4 (laminar) or 6×6 (turbulent with k–ε model) system in each node. The results for the scalar equations were confirmed. For low Mach number ($M = 10^{-3}$), high Reynolds number in flow direction (10^6 on cell basis), grid aspect ratios varying from 1 to 1000, the smoothing properties of the ASLGS and AMILU (1) were found to be excellent and by far superior to the smoothing properties of pointwise Gauss–Seidel relaxation (PGS).

A single PGS has very poor smoothing properties. Combinations of PGS-relaxations can lead to better smoothing. A symmetric PGS (PGS_2) is a combination of two single PGS each started in an opposite corner of the grid. The combination of four sequential PGS-steps each started from a different corner (PGS_4) was found to be necessary to improve smoothing and robustness. The following ordering of the relaxations was chosen: (1) starting from the left lower corner marching along vertical lines upwards, line after line in forward order; (2) starting from the right upper corner marching along vertical lines downwards, line after line in backward order; (3) starting from the left upper corner marching along horizontal lines forwards, line after line in downward order; (4) starting from the right lower corner marching along horizontal lines backwards, line after line in upward order.

In the line Gauss–Seidel method, a full geometric line is placed on the new iteration level. For instance, we can choose vertical lines and forward marching direction. A LGS is called symmetric when a forward sweep is followed by a backward sweep (SLGS). An alternating line relaxation

(ALGS) consists of a (forward) vertical line relaxation (VLGS) followed by a (forward) horizontal line relaxation (HLGS). The best smoothing properties were obtained by an alternating symmetric LGS (ASLGS) with the following sequence: (1) HLGS with an upward sweep; (2) VLGS with a backward sweep; (3) HLGS with a downward sweep; (4) VLGS with a forward sweep.

The linearized system of equations obtained from the discretization of the Navier–Stokes equations can be written as $Ax = b$, where A is a sparse matrix with five co-diagonals each having elements of 4×4 (laminar) or 6×6 (turbulent with k–ε model) submatrices. With incomplete factorization a splitting $A = M - N$ is generated where M is a sparse matrix easy to factorize in a lower matrix with unit diagonal L and an upper matrix U: $A = M - N = LU - N$. The elements of the matrices L and U are 4×4 or 6×6 submatrices.

During the factorization either no fill-in is allowed in L and U (ILU(0)) leading to a product LU with 5 co-diagonals or one position of fill-in is allowed in L and U (ILU(1)) leading to a product LU with 7 co-diagonals. By a modified ILU (MILU) is understood that terms generated during the factorization that fall outside allowed positions are not simply thrown away but are partially absorbed in the diagonal of U. The procedure used here is that the absolute values of the terms in a row are added to the diagonal element of that row after multiplication with a factor σ. In order to obtain good smoothing, alternating the visiting order of the nodes has been used (A(M)ILU): (1) starting from the left lower corner marching along vertical line upwards, line after line in forward order; (2) starting from the left upper corner marching along horizontal lines forwards, line after line in downward order. Steps 1 and 2 are applied twice. The choice of the steps is based on the 4 steps in the PGS_4 method. We remark however that a step symmetric to step 1, i.e., marching along vertical lines downwards, line after line in backward order, results in the same fill-in positions as step 1 itself. In the same way, a step symmetric to step 2 gives the same fill-in positions as step 2 itself. Therefore, it does not make much sense to symmetrize the steps. The two steps are repeated to have the same number of steps as in the PGS_4 method.

The optimal σ-factor depends on the grid size and the direction of the flow with respect to the grid lines. For extreme fine grids and flow aligned with the grid, the σ-factor can be as low as $\sigma = 10^{-4}$. In practice, larger factors are to be used.

A specific drawback of the ILU-method is that the L- and U-matrices have to be stored in memory. For larger problems this is prohibitive. The method adopted in this work consists in subdividing the global grid into blocks with one line overlap. The blocks are visited sequentially like in the point Gauss–Seidel method (block Gauss–Seidel: BGS) using the four steps. Within the blocks the approximate solution of the system is obtained by the AMILU-method. In the blocks themselves nodes are visited in the order that accords with the marching direction between the blocks (pattern 1 or 2). In this way, the blockwise relaxation method has the properties of the AMILU(1) for large block size and the properties of the PGS_4 for small block size.

3. Multigrid results

The discretization of the laminar and turbulent flow equations is based on [3, 6]. The multigrid uses four grids in a W-cycle with four iterations on each grid level (pointwise, linewise or blockwise). Full weighting is used for the restriction of the defects, injection for the restriction of the variables and bilinear interpolation as prolongation of the correction. A second-order correction for the convective terms based on a minmod-limiter is put on the right-hand side for all cases. Two versions of the second-order implementation into the multigrid have been considered. In the defect-correction formulation (DC), the second-order correction is updated once on the finest grid after each cycle. In the mixed discretization formulation (MD), the second-order correction is updated at each iteration on the finest grid. Both in the ASLGS and the AMILU(1) methods an underrelaxation factor $\omega = 0.8$ is used for the defect correction cycle whereas $\omega = 0.5$ is used for the mixed discretization.

Fig. 1 shows the convergence behaviour for a laminar flow over a flat plate of 1.34 m length with a sharp leading edge discretized on a 385×97 grid (test case 1). The inlet of the computational domain is 0.25 m upstream of the leading edge. The leading edge is at position (97, 1). This test case was also used in [3, 6]. The uncoming Mach number is $M = 0.015$ and the Reynolds number is $3.3 \cdot 10^5$/m. The residue shown is the maximum over all equations and all grid points. The second-order accuracy is obtained by defect correction. Curve a represents a PGS_4 while curves b, c and d are all BGS-methods with different block sizes: 3×3, 5×5, 13×13. Within the blocks, the solution is obtained here with a direct method. Increasing the size obviously enhances the convergence. This shows the principal strength of more complete solvers. The ASLGS method (curve e) results in almost the same convergence rate as the BGS 5×5. Better convergences rates are obtained for the same test case with mixed discretization (Fig. 2): curve b: BGS 13×13; curve c: ASLGS; curve d: AMILU(1) on a maximum block size of 97×97 and $\sigma = 0.3$. For comparison, the ASLGS in defect correction formulation is given (curve a). The advantage of the mixed discretization is clear. The advantage of the blockwise solver is clear when the performance is evaluated in terms of cycles. The blockwise solver however, is, much more expensive than the linewise solver. The required CPU-time to reach full convergence (reaching of machine accuracy) is shown in Table 1 in a relative way where the PGS_4 method in combination with a defect correction is taken as baseline. Convergence was reached in 154 h for the baseline on a HP730 with a 85 SpecMark89 performance and with a memory capacity of 32 MBytes. The AMILU(1) needs more time for full

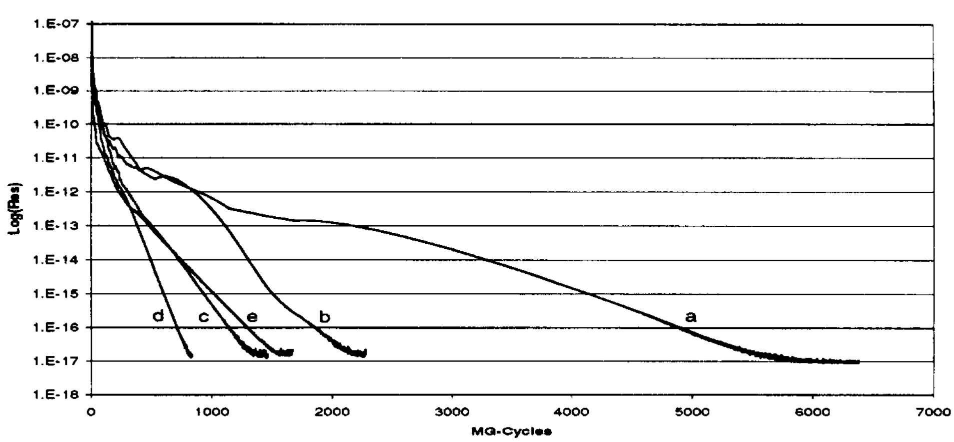

Fig. 1. Convergence histories for laminar flat plate flow (385×97) in defect correction: (a) PGS_4, (b) BGS 3×3; (c) BGS 5×5, (d) 13×13 and (e) ASLGS.

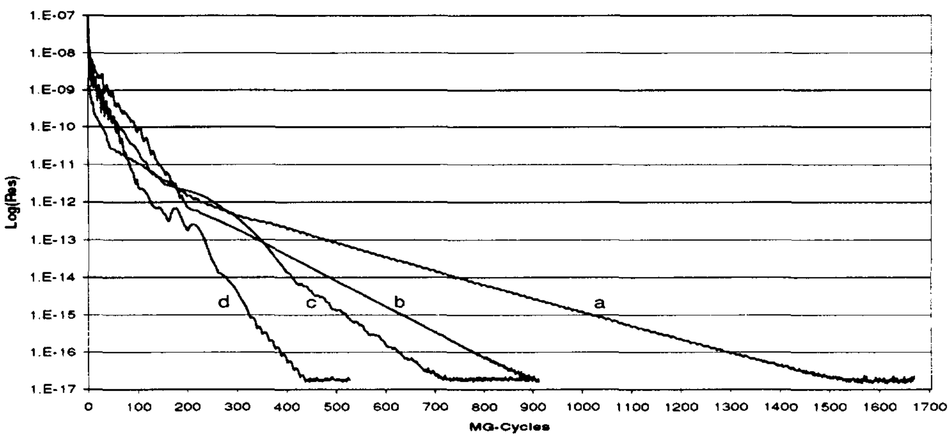

Fig. 2. Convergence histories for laminar flat plate flow (385 × 97) in mixed discretization (MD) and defect correction (DC). (a) ASLGS (DC), (b) BGS 13 × 13 (MD), (c) ASLGS (MD) and (d) AMILU(1) 97 × 97 (MD).

Table 1
Relative CPU cost needed to reach convergence; laminar test cases

Case	PGS_4 (DC)	ASLGS (DC)	ASLGS (MD)	AMILU(1) (MD)	BGS (MD)
1	1	0.751	0.448	0.533	9.050
2	1	1.108	0.880	0.973	NA

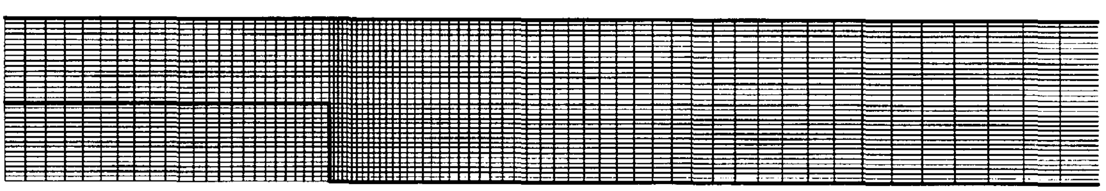

Fig. 3. Backward facing step with expansion ratio of 1 : 2.

convergence than the ASLGS. This means that on the basis of CPU-time, the ASLGS is the most efficient solver.

The second test case is backward facing step with an expansion ratio of 1 : 2 discretized on a 97 × 33 grid (Fig. 3). The inlet velocity profile is parabolic. For the maximum velocity the Mach number is 0.015 and the Reynolds number based on a channel height is 300. In Fig. 4, the influence of the grid size on the MG-performance is demonstrated. A four-level MG on the base grid is compared with a five-level MG on a 193 × 65 grid. Curves (a) and (b) show the same performance ($\simeq$ 30 MG-cycles) for a first-order discretization solved by an ASLGS, as is expected from the

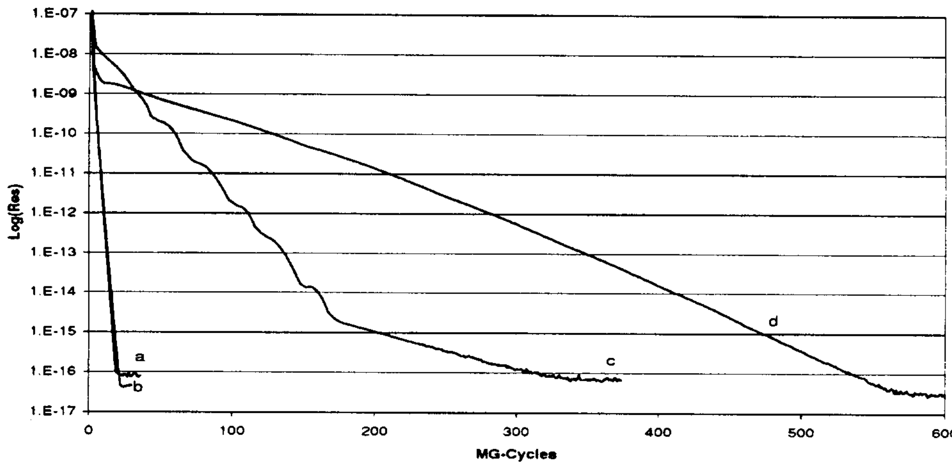

Fig. 4. Convergence histories for laminar backward facing step: (a) ASLGS (97 × 33), (b) ASLGS (193 × 65), (c) ASLGS (DC) (97 × 33) and (d) ASLGS (DC) (193 × 65).

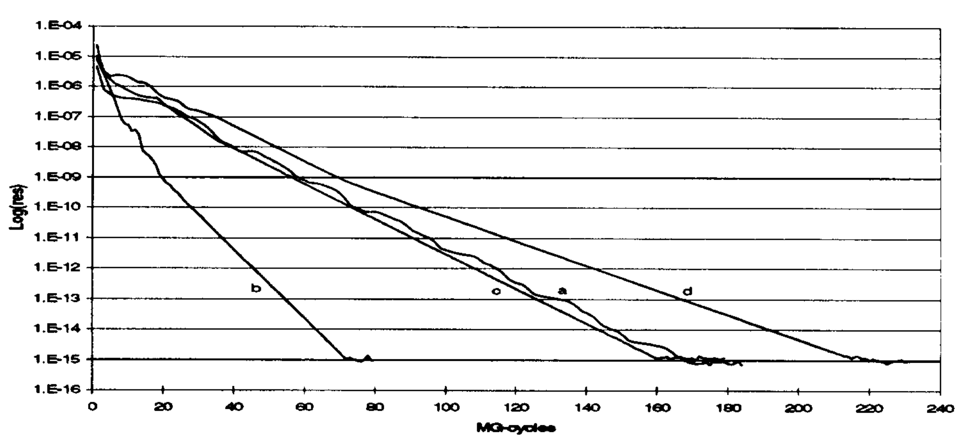

Fig. 5. Convergence histories for laminar backward facing step (97 × 33; $Re_h = 300$): (a) ASLGS (MD), (b) AMILU(1) 97 × 97 (MD), (c) AMILU(1) 33 × 33 (MD) and (d) AMILU(1) 17 × 17 (MD).

theory. However, as the second-order discretization cannot be taken into the MG-cycle, the performance decreases with increasing node numbers. Curves (c) and (d) correspond to the ASLGS-DC implementation for base grid and finer grid. Results for mixed discretization are shown for ASLGS (curve a) and AMILU(1) with $\sigma = 0.1$ in Fig. 5. The effect of block size for the AMILU method is illustrated. Curve b corresponds to an AMILU-method applied to the whole field (still within the memory capacity of the computer system) while curves c and d correspond to the maximum block size of 33 × 33 and 17 × 17. An important loss of performance by reduction of block size is observed. The relative CPU-time, given in Table 1, does not reveal spectacular gains among the different methods. The PGS_4-method, chosen as baseline, reached full convergence in

53 min. Due to the low Reynolds numbers, the smoothing characteristics of the considered methods are comparable resulting in similar CPU-costs. Nevertheless, the ASLGS method in MD-version is the most efficient method.

A similar comparison is made for turbulent flow on a flat plate (test case 1). The Yang-Shih low Reynolds number turbulence model is employed. The same mesh is used as in the laminar test case. The freestream turbulence level is set at Tu = 3%, the Mach number at $M = 0.015$ and the Reynolds number at $3.3 \cdot 10^5$/m. The inclusion of the turbulence equations into the multigrid cycle is not straightforward as corrections from the coarse grids can, especially in the beginning of the convergence, result in negative values for the turbulence quantities k and ε [3]. Two versions are considered here. The first version performs a MG-cycle on the Navier–Stokes equations but not on the turbulence equations which are only solved on the finest grid. In the second version, the turbulence equations are put into a MG-cycle with a damping of the corrections from the coarse to fine grid according to [3]: $\phi_{\text{new}} = (\phi_{\text{old}} + \alpha\delta\phi^+)/(\phi_{\text{old}} - \alpha\delta\phi^-)\phi_{\text{old}}$, where $\alpha = 0.3$ (ϕ represent either k or ε). The values $\delta\phi^+$ and $\delta\phi^-$ are, respectively, the positive or negative corrections. By this formulation, any negative correction will never turn a turbulent quantity into a negative value. In Fig. 6, curves a1 to d1 represent the convergence histories for the Navier–Stokes equations while curves a2 to d2 represent the convergence histories for the turbulence equations. Curves a and b correspond to ASLGS, curves c and d to AMILU(1) with a maximum block size of 49 × 49 and $\sigma = 0.3$.

By bringing the turbulence equations into the multigrid, the convergence of the turbulence quantities is improved for the ASLGS method. However, this only results in a small benefit for the Navier–Stokes part. Further, the principal superiority of the AMILU(1) method over the ASLGS method is clear. The PGS_4 method with k and ε in MG (not shown) reached convergence in 4000 MG-cycles or 154 h in CPU-time. The same remarks hold as for the laminar test cases. The AMILU- and PGS_4-method are more time consuming so that in CPU-time the ASLGS method is advantageous. This can be seen in Table 2 where the PGS_4 with k and ε in MG is taken as baseline for the first turbulent test case.

A symmetric turbulent backward facing step is taken as last case with a step height of 5.4 mm and the downstream channel height $h = 135.0$ mm. The Mach number is $M = 0.01$, the Reynolds

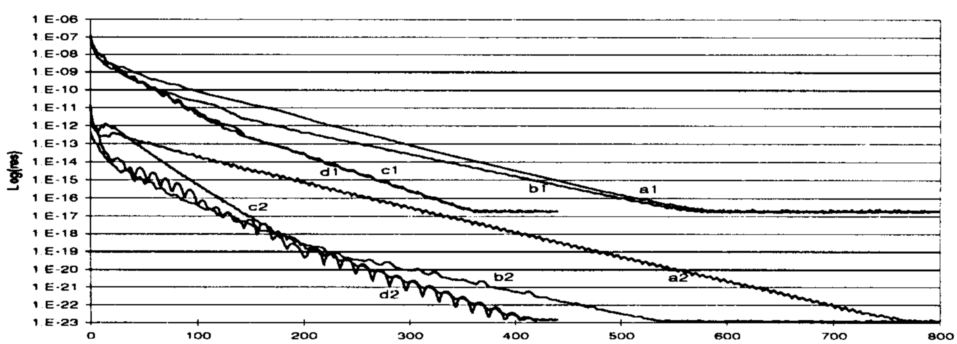

Fig. 6. Convergence histories for turbulent flat plate flow (385 × 97) in mixed discretization: (a) ASLGS k and ε in SG, (b) ASLGS k and ε in MG, (c) AMILU(1) 49 × 49 k and ε in SG and (d) AMILU(1) 49 × 49 k and ε in MG.

Table 2
Relative CPU cost needed to reach convergence; turbulent test cases

Case	PGS_4 (MG)	ASLGS (SG)	ASLGS (MG)	AMILU(1) (SG)	AMILU(1) (MG)
1	1	0.615	0.541	0.653	0.781
2	N.A.	1	Not converged	1.127	Not converged

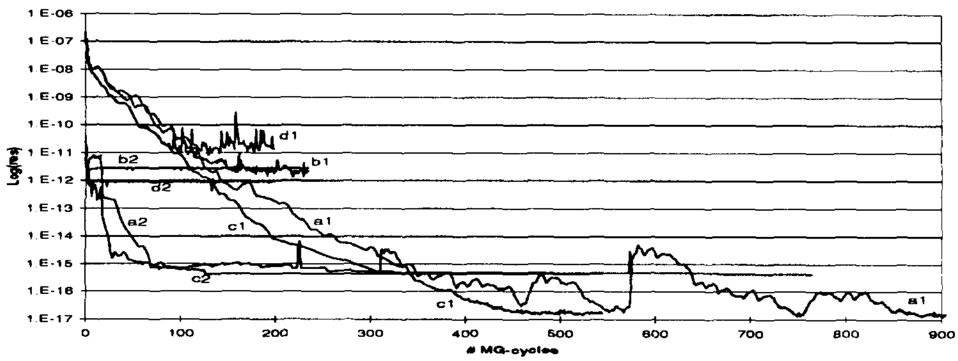

Fig. 7. Convergence histories for turbulent backward facing step (385 × 97) in mixed discretization: (a) ASLGS k and ε in SG, (b) ASLGS k and ε in MG, (c) AMILU(1) 49 × 49 k and ε in SG and (d) AMILU(1) 49 × 49 k and ε in MG.

number $Re_h = 2957$ and the grid size 385 × 97. The grid is similar to the one shown in Fig. 3 but with a large stretching rate in order to obtain grid points sufficiently close to the walls. The upper boundary is a symmetry line. Fig. 7 shows the convergence for the ASLGS-method (curve a and b for the first and second MG-version) and the AMILU(1)-method (curve c and d) with $\sigma = 0.1$ on a maximum block size of 49 × 49. The damping of the coarse grid corrections for k and ε (second MG-version) is apparently not sufficient to guarantee convergence (curves b_2 and d_2). k and ε have limit cycle behaviour (explained below) preventing the convergence of the Navier–Stokes quantities (curves b_1 and d_1). Also in the first MG-version, the residues for k and ε do not decrease as deeply as for the turbulent flat plate. In the recirculation region, the values of k and ε tend toward very small values, often leading to negative values. These are replaced explicitly by small positive values. By this limiting procedure, the residuals of the equations cannot converge to machine zero. The observation is that by refining the grid, the occurrence of negative turbulence quantities is diminished. The Navier–Stokes quantities (curves a_1 and c_1) finally converge to the level of the turbulence quantities (curves a_2 and c_2). The obtained convergence level is sufficient (seven orders of magnitude). Also for this test case the ASLGS method is cheaper than the AMILU(1) method as can be seen in Table 2 where ASLGS (SG) is taken as baseline. The behaviour is the same as for the laminar case (Table 1).

The observation that bringing the turbulence equations into the multigrid formulation does not enhance the convergence, or even obstructs it, is in contrast to the results of earlier work [3, 6]. The difference is that now the more powerful linewise and blockwise methods are used, while in the earlier work it was the pointwise method. For the pointwise method it was found that bringing the turbulence equations into the multigrid formulation was beneficial. An explanation for this different behaviour can be found in the character of the equations for the turbulence quantities which are largely driven by the source terms. As these terms depend on derivatives of flow quantities, their smoothness depends on the smoothness of the flow quantities. For the linewise and blockwise methods the smoothing is strong. So smoothness of the source terms in the turbulence equations is guaranteed. As a consequence, bringing the turbulence equations into the multigrid cannot result in much effect. These turbulence equations have convective and diffusive parts that are less significant then the source part. As the smoothing of a relaxation method has to come from the convective and diffusive parts in the equations, not much supplementary smoothing can be obtained for the turbulence equations. The situation is different if the relaxation method is weak like the pointwise method. In that case, smooth source terms cannot be guaranteed by the relaxation of the Navier–Stokes equations. The only smoothing for the turbulence equations has to come from the treatment of the convective and diffusive terms.

4. Conclusions

For low Mach number flow, an important convergence improvement has been obtained over the PGS_4 method by the ASLGS and AMILU(1) methods. Due to the memory requirements, the size of the blocks within the BGS-method is automatically smaller than for the AMILU(1) method resulting in lower smoothing characteristics. This lower performance in combination with the higher CPU-cost makes the BGS-method unattractive. In most test cases, the AMILU(1) method performs better than the ASLGS method in terms of MG-cycles but not in terms of CPU-cost. Furthermore, the AMILU(1) has a higher memory requirement. Both AMILU(1) and ASLGS allow a mixed discretization which nearly doubles convergence speed with respect to the defect correction implementation. The general conclusion is that the ASLGS method is to be preferred.

A multigrid formulation of the k–ε equations requires damping of the coarse grid corrections. The convergence plots indicate that no acceleration for the flow quantities is obtained by bringing the turbulence equations into the multigrid formulation in the cases where good smoothers are applied, i.e., ASLGS and AMILU(1).

Acknowledgements

The research was granted under contract IUAP/17 by the Federal Services of Scientific, Technical and Cultural Affairs (D.W.T.C.) and under contract G.0283.96 by the Flemish Science Foundation (F.W.O.).

References

[1] W.K. Anderson, R.D. Rausch, D.L. Bonhaus, Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids, AIAA–95–1740–CP, 1995.

[2] P.M. de Zeeuw, Incomplete line LU as smoother and as preconditioner, In: W. Hackbusch, G. Wittum (Eds.), NNFM 41, Incomplete Decompositions (ILU) – Algorithms, Theory and Applications, Vieweg, Braunschweig, 1992, pp. 215–224.

[3] E. Dick, J. Steelant, Multigrid solution of the steady compressible Navier–Stokes equations coupled to the k–ε turbulence equations, AIAA–95–1669–CP, 1995.

[4] R. Kettler, Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods, in: W. Hackbusch, U. Trottenberg (Eds.), Multigrid Methods, Proc. Köln-Porz, 1981, Springer, Berlin, 1982, pp. 502–534.

[5] A. Soria, F. Ruel, Assessment of some iterative methods for non–symmetric linear systems arising in computational fluid dynamics, Internat. J. Numer. Methods for Fluids 21 (1995) 1171–1200.

[6] J. Steelant, E. Dick, A multigrid method for compressible Navier–Stokes equations coupled to the k–ε turbulence equations, Internat. J. Numer. Methods Heat and Fluid Flow 4(2) 99 (1994) 113.

[7] J. Steelant, S. Pattijn, E. Dick, Smoothing properties of different preconditioners for Navier–Stokes equations, in: J.-A. Désidéri et al., (Eds.), Proc. 3rd European CFD Conf. Paris, Wiley, New York, 1996, pp. 616–622.

[8] P. Wesseling, A survey of Fourier smoothing analysis results, in: W. Hackbusch, U. Trottenberg, (Eds.), ISNM 98, Multigrid Methods III, Birkhäuser, Basel, 1991, pp. 105–127.

[9] P. Wesseling, The role of incomplete LU-factorization in multigrid methods, in: W. Hackbusch, G. Wittum, (Eds.), NNFM 41, Incomplete Decompositions (ILU) – Algorithms, Theory and Applications, Vieweg, Braunschweig, 1992, pp. 202–214.