INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **61**:185–219
Published online 4 November 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1945

Implicit meanflow–multigrid algorithms for Reynolds stress model computation of 3-D anisotropy-driven and compressible flows

G. A. Gerolymos and I. Vallet*, †

Institut d'Alembert, Université Pierre-et-Marie-Curie, Case 161, 4 place Jussieu, 75005 Paris, France

SUMMARY

The present paper investigates the multigrid (MG) acceleration of compressible Reynolds-averaged Navier–Stokes computations using Reynolds-stress model 7-equation turbulence closures, as well as lower-level 2-equation models. The basic single-grid SG algorithm combines upwind-biased discretization with a subiterative local-dual-time-stepping time-integration procedure. MG acceleration, using characteristic MG restriction and prolongation operators, is applied on meanflow variables only (MF–MG), turbulence variables being simply injected onto coarser grids. A previously developed non-time-consistent (for steady flows) full-approximation-multigrid (s–MG) is assessed for 3-D anisotropy-driven and/or separated flows, which are dominated by the convergence of turbulence variables. Even for these difficult test cases CPU-speed-ups $r_{\mathrm{CPUSUP}} \in [3, 5]$ are obtained. Alternative, potentially time-consistent approaches (unsteady u–MG), where MG acceleration is applied at each subiteration, are also examined, using different subiterative strategies, MG cycles, and turbulence models. For 2-D shock wave/turbulent boundary layer interaction, the fastest s–MG approach, with a $\mathrm{V}(2, 0)$ sawtooth cycle, systematically yields CPU-speed-ups of $5 \pm \frac{1}{2}$, quasi-independent of the particular turbulence closure used. Copyright © 2008 John Wiley & Sons, Ltd.

Received 20 June 2008; Revised 4 September 2008; Accepted 5 September 2008

KEY WORDS: second-moment closure; Reynolds stress model; multigrid; compressible RANS methods; 3-D separated flows; secondary flows; shock wave/turbulent boundary layer interaction

1. INTRODUCTION

Several authors have contributed to the development of computational methods for the simulation of complex flows modelled using second-moment closures (SMC) [1–16]. Virtually all of these

*Correspondence to: I. Vallet, Institut d'Alembert, Université Pierre-et-Marie-Curie, Case 161, 4 place Jussieu, 75005 Paris, France.
†E-mail: isabelle.vallet@upmc.fr

methods are upwind-biased and implicit, to achieve the desirable robustness and efficiency, in the implementation of 7-equation full Reynolds stress models (RSMs). Fully converged results for complex aerospace configurations can be obtained by such single-grid (SG) efficient implicit solvers, on large computational meshes with $O(10^7)$ points, in $O(100)$ CPU-h, on a computer/code combination with $4\,\mathrm{Gflop\,s^{-1}}$ sustained performance [17, 18] (the CPU-time may vary by a factor 2, both upward and downward, depending on flow complexity, global boundary conditions well posedness, and Mach number).

Multigrid (MG) convergence acceleration is expected to further enhance computational efficiency, but, to the authors' knowledge, very little work has been done on the MG computation of the Reynolds-averaged Navier–Stokes (RANS) equations with 7-equation Reynolds stress closures, both for incompressible Reynolds-averaged Navier–Stokes (iRANS) [19, 20] and for compressible Reynolds-averaged Navier–Stokes (cRANS) [17, 18, 21]. Careful examination of these previous studies (Table I) shows that evaluation of CPU-speed-up obtained by MG, for computations with SMC, has been reported only by Lien and Leschziner [20] for iRANS (CPU-speed-up-ratio $r_{\mathrm{CPUSUP}}=9$ using $L_{\mathrm{GRD}}=4$ grids, i.e. the fine grid and three coarser grids), and by Gerolymos and Vallet [17, 18] for cRANS ($r_{\mathrm{CPUSUP}}=4$ with $L_{\mathrm{GRD}}=3$). These authors [17, 18, 20] have also reported CPU-speed-ups obtained with the same MG method for 2-equation closures. Lien and Leschziner [20] report slightly higher CPU-speed-ups for the $k-\varepsilon$ MG solver than for the RSM MG solver, viz 14 vs 9. Gerolymos and Vallet [17, 18] observe similar CPU-speed-ups $r_{\mathrm{CPUSUP}}=4$, in both cases. In order to assess the effectiveness of these approaches, one has to evaluate the speed-ups achieved by comparison with published studies on MG methods for 2-equation models, for iRANS [19, 30–36], and cRANS [17, 18, 20, 21, 35, 37–51], respectively (Tables II, III). Although the lists of MG methods for RANS with multi-equation turbulence models (Tables I–III) may be incomplete, they give a representative overview of existing methods and of author-evaluated CPU-speed-ups. Various criteria were used by different authors in reporting CPU-speed-ups e.g. convergence of residuals to machine-precision or until saturation, reduction of residuals by a given number of orders-of-magnitude, convergence of selected local and/or global flow quantities, etc.), so that these CPU-speed-ups (Tables I–III) should be considered as an approximate indication of performance.

Review of MG methods with 2-equation models (Tables II, III) indicates that, in general, $L_{\mathrm{GRD}}\in[3,4]$ are used, both for iRANS and cRANS, but also that the CPU-speed-ups obtained with the pressure-based (PB) iRANS methods are higher, by a factor of $2\frac{1}{2}$, than those obtained with the time marching (TM) cRANS methods. It is not clear whether these higher CPU-speed-ups for PB–iRANS are related to the systematic use of full multigrid (FMG) by these methods or to the nature of the incompressible flow equations and/or the properties of the solvers. Nonetheless, one has to bear in mind that cRANS address generally nonsmooth flows (flows with shock waves), with corresponding serious computational difficulties. Furthermore, Lien and Leschziner [20] have observed that full approximation scheme FAS–FMG computations are faster by a factor of at least 2 compared to FAS–MG without FMG (most iRANS methods use FMG). Notice also that Cornelius *et al.* [34] report, for PB–iRANS FAS–FMG computations, severe deterioration of CPU-speed-up when using highly non-orthogonal grids (from $r_{\mathrm{CPUSUP}}=9$ obtained on quasi-orthogonal grids down to $r_{\mathrm{CPUSUP}}=2$). Concentrating on the cRANS MG-solvers, which are the subject of the present work, reported author-evaluated CPU-speed-ups with 2-equation closures are in the range $r_{\mathrm{CPUSUP}}\cong 4$ (Table III).

DOI: 10.1002/fld

Table I. Overview of existing multigrid methods for RANS with RSM turbulence models, and CPU-speed-ups obtained by multigrid acceleration.

						Multigrid			Speed-up	
Author(s)	Year	Eqs.	RSM	Method	Grid	Strategy	Turbulence	Cycle	L_{GRD}	CPU-speed-up
Demuren [19]	1992	iRANS	[22]	PB	s	FAS–FMG	MF–MG	V(1,1)	3	na
Lien and Leschziner [20]	1994	iRANS	[23]	PB	s	FAS–FMG	RD+PPP	V*	4	9
Gerolymos and Vallet [17, 18]	2005	cRANS	[10, 24–26]	TM	s	FAS	MF–MG	V(1,0)	3	4
Bigarella and Azevedo [21]	2007	cRANS	[13]	TM	u	FAS–FMG	na	V	na	na

*V(1,1) or V(2,2) [20]; PB: pressure-based [27]; TM: time-marching [28]; s: structured grid; u: unstructured grid; FAS: full approximation scheme [29]; FMG: full multigrid [29]; MF–MG: meanflow multigrid [17]; RD: residuals damping [20]; PPP: positivity preserving prolongation fix [20]; L_{GRD}: number of grid levels (including fine grid); speed-up: actual speed-up in CPU-time (to reach some author-dependent convergence criterion); na: not available.

DOI: 10.1002/fld

Table II. Overview of multigrid methods for incompressible RANS with 2-equation turbulence models.

					Multigrid		Speed-up		
Author(s)	Year	Model	Method	Grid	Strategy	Turbulence	Cycle	L_{GRD}	CPU-speed-up
Demuren [19]	1992	[52]	PB	s	FAS–FMG	MF–MG	V(1,1)	3	na
Claus and Vanka [30]	1992	[52]	PB	s	FAS–FMG	MF–MG	V	5	na*
Shyy *et al.* [31, 32]	1993	[52]	PB	s	FAS–FMG	RD+PPP	V	6	na
Lien and Leschziner [20]	1994	[52]	PB	s	FAS–FMG	RD+PPP	V†	4	14
Nowak and Salcudean [33]	1996	[52]	PB	s	NLMG	RD+PPP	V	3	na
Cornelius *et al.* [34]	1999	[52]	PB	s	FAS–FMG	na	V‡	3	9§
Vázquez *et al.* [35]	2004	[53]	SUPG	u	FAS–FMG	RD+PPP+STR	V	3	10
Yan *et al.* [36]	2007	[52, 54]	PB	s	FAS–FMG	RD+PPP	V	4	12
					NLMG–fmg	RD+PPP	V	3	8
					NLMG–fmg	RD+PPP	V	4	36

*Claus and Vanka [30] report a factor 41 CPU-speed-up with $L_{GRD}=5$.

†V(1,1) or V(2,2) [20].

‡$V(N_{V_1}, N_{V_2})$ with [34] $N_{V_1} \in [4, 6]$ and $N_{V_2} \in [1, 4]$.

§For test cases with good grid-orthogonality CPU-speed-ups >9 were obtained, while for a test case with poor grid-orthogonality CPU-speed-up was only 2; PB: pressure-based [27]; SUPG: streamline-upwind/Petrov–Galerkin [55]; s: structured grid; u: unstructured grid; NLMG: nonlinear multigrid [56]; FAS: full approximation scheme [29]; FMG: full multigrid [29]; MF–MG: meanflow multigrid [17]; STR: source terms restriction [37, 41, 42]; RD: residuals damping [20]; PPP: positivity preserving prolongation fix [20]; L_{GRD}: number of grid levels (including fine grid); speed-up: actual speed-up in CPU-time (to reach some author-dependent convergence criterion); na: not available.

One common issue of these MG-techniques (Tables I–III) concerns the particular way the turbulence-transport equations are treated. There are three different approaches:

1. Meanflow–multigrid (MF–MG): In this approach [17–19, 30, 38, 40] turbulence-transport equations are solved on the fine grid only, and turbulence variables are simply injected onto coarser grids.
2. Fully coupled MG with stabilization fixes (FCSF–MG): In this approach [20, 33, 35–37, 39], [41–43, 46, 49, 50] turbulence-transport equations are solved on all grids using FAS [29] or NLMG [56], but several stabilization techniques are used to avoid divergence of the computations, e.g. source terms restriction (STR: the meanflow gradients appearing in the turbulence source terms are computed on the fine grid only, and then restricted onto coarser grids [35, 37, 39, 41–43, 49]), residuals damping (RD: the turbulence-variables residuals are damped by an underrelaxation coefficient in the MG restriction and/or prolongation phase [20, 33, 35, 36, 46, 50]) and specific positivity preserving prolongation fixes (PPP: for physically unconditionally positive variables, only positive increments are permitted in the prolongation from coarser to finer grids, thus guaranteeing that MG does not induce loss of positivity [20, 33, 35, 36, 39, 46, 50]).
3. Linear multigrid (LMG): In this approach [44, 45] MG is applied to the iterative technique used to solve the linear system resulting from the implicit time discretization of the equations [56] (this was also termed Newton MG iteration by Hackbusch [56]), and as such can be applied to turbulence transport without any particular stabilization fix.

DOI: 10.1002/fld

Table III. Overview of multigrid methods for compressible RANS with 2-equation turbulence models (when 2 different techniques [a]/[b] are indicated for time and/or space discretization [a] applies to meanflow and [b] to turbulence transport).

				Basic SG scheme			Multigrid		Speed-up		
Author(s)	Year	Model	Type	Grid	Space	Iteration	Strategy	Turbulence	Cycle	L_{GRD}	CPU-speed-up
Gerolymos [37]	1990	[57]	TM	s	C2	IRS	Ni [58]	STR	V(1,0)	4	na
Yokota [38]	1990	[52]	TM	s	C2	DLU	FAS	MF–MG	W	4	na
Mavriplis and Martinelli [39]	1994	[52, 59]	TM	u	C2/UW1	RK5/IRK5	FAS	STR+PPP	na	na	na
Braaten and Connell [40]	1996	[60]	TM	u	C2	RK3	FAS	MF–MG	V	3	2
Liu and Zheng [41]	1996	[61]	TM	s	MUSCL3-Roe	RK5+IRS	FAS	STR	na	3	na
Gerlinger and Brüggemann [42, 43]	1997	[62]	TM	s	C2	LU–SGS	FAS	STR	V(1, 0)	4	4
Carré *et al.* [44, 45]	1997	[52]	TM	u	MUSCL2-Roe	Jacobi	LMG	LMG	V	$\in[4, 6]$	4
Dick and Steelant [46]	1997	[57, 63, 64]	TM	s	UW1-Roe	GS	FAS	RD+PPP	W	4	na
Ålund *et al.* [47]	1997	[57]	TM	s	C2/HCUW	RK4/TDMA	FAS	na	V	2	na
Demuren and Ibraheem [48]	1998	[65]	TM	s	C2	BW ADI–AF	FAS–FMG	na	V(1,0)	3	2
Park and Kwon [49]	2004	[61, 66]	TM	s	MUSCL3-Roe	DADI	FAS	STR	V(1,0)	3	4
Vázquez *et al.* [35]	2004	[53]	ST–FE	u	EV–GLS	GMRES	FAS–FMG	RD–PPP–STR	V	2	6
Lambropoulos *et al.* [50]	2005	[52]	TM	u	MUSCL2-Roe	point Jacobi	FAS–FMG	RD + PPP	V	3	4
Gerolymos and Vallet [17, 18]	2005	[57]	TM	s	MUSCL3-VL	ADI–AF LDTS	FAS	MF–MG	V(1,0)	4	4
Bigarella *et al.* [21, 51]	2007	[66–69]	TM	u	MUSCL2/UW1 Roe	RK5/bi-CG	FAS–FMG	na	V	3	na

TM: time-marching [28]; ST–FE: space-time finite-elements [70]; s: structured grid; u: unstructured grid; C2: centered $O(\Delta x^2)$ [28]; MUSCL3: $O(\Delta x^3)$ monotone upstream scalar conservation law (MUSCL) [27]; UW1: $O(\Delta x)$ upwind [28]; HCUW: hybrid central upwind [71]; VL: van Leer [27] flux-vector-splitting (FVS); Roe: Roe [27] flux-difference-splitting (FDS); EV: entropy-variables [72]; GLS: Galerkin least-squares [70]; DLU: diagonally-inverted Lower-Upper (LU) [38]; LU–SGS: Lower-Upper (LU) Symmetric Gauss-Seidel [73]; BW: Beam-Warming implicit time-marching [27]; ADI–AF: alternating-directions-implicit approximate-factorization [73]; RK5: 5-stage Runge–Kutta [74]; IRK5: 5-stage implicit Runge–Kutta [39]; IRS: implicit-residual-smoothing [74]; DADI: diagonalized ADI [27]; bi-CG: bi-conjugate-gradient [4, 75]; GS: Gauß-Seidel [75]; Jacobi: [75]; GMRES: generalized minimal residual [75]; TDMA: tridiagonal matrix algorithm [71]; LDTS: local dual-time-stepping [14]; FAS: full approximation scheme [29]; FMG: full multigrid [29]; LMG: linear multigrid [56, 76]; MF–MG: meanflow multigrid [17]; STR: source-terms-restriction [37, 41, 42]; RD: residuals damping [20]; PPP: positivity preserving prolongation fix [20]; L_{GRD}: number of grid levels (including fine grid); speed-up: actual speed-up in CPU-time (to reach some author-dependent convergence criterion); na: not available.

DOI: 10.1002/fld

Linear multigrid (LMG) can be efficient provided the solution of the linear system corresponds to a large percentage of CPU usage for the SG algorithm [76]. MF–MG is very simple to implement, and indeed particularly robust. Nonetheless it is generally accepted [20, 36, 37, 41, 42, 49] that higher CPU-speed-ups can be obtained using fully coupled MG with stabilization fixes (FCSF–MG). Gerlinger and Brüggemann [42], using a 2-equation model [62], have compared CPU-speed-ups obtained by a V(2, 0) FAS–FCSF–MG strategy, with those obtained using MF–MG, on a ($M_\infty = 4$, $\alpha_c = 10°$) compression ramp, and found that, for the density residual, FCSF–MG was faster by a factor of 2.25. Nonetheless, in these computations of Gerlinger and Brüggemann [42], virtually no CPU-speed-up was obtained by MF–MG for the $\sqrt{k}$-residual. On the other hand, the present authors [17, 18], using MF–MG, have observed similar CPU-speed-ups in the convergence of the meanflow and of the turbulence variables.

Although FCSF–MG is expected to further enhance convergence acceleration compared with MF–MG, in the present work we focus on the systematic evaluation of MF–MG, applied to RSM–RANS, for a wide range of flows. In a recent work, the authors [17] have developed an MF–MG technique for RSM–RANS. The method, which uses full approximation MG (FAS–MG), applies successively the subiterative procedure of the basic SG scheme [14, 15] on each grid, using a V(1, 0)-cycle (sawtooth cycle) [74, 77]. The restriction (transfer) and prolongation (interpolation) operators are based on the characteristic MG approach of Leclercq and Stoufflet [78], which maintains the upwind character of the scheme and ensures stability of the multigrid algorithm. The method [17] has been applied to several 2-D and 3-D flows, including shock wave/turbulent boundary layer interactions (SWTBLI) [17, 18, 79–81], flows with large separation [17, 82], and multistage transonic turbomachinery flows [18], and has consistently given CPU-speed-ups in the range $r_{\mathrm{CPUSUP}} \in [3, 4]$.

The purpose of this paper is:

1. to further explore the convergence acceleration possibilities of the MF–MG algorithm [17] for complex 3-D flows, including flows driven by turbulence anisotropy, to determine whether the method still offers significant speed-ups, even when the convergence of meanflow variables is subordinated to the convergence of the turbulence variables,
2. to evaluate alternative MG strategies, and in particular potentially time-consistent algorithms (acceleration of the subiterative convergence) that could also be used for the DTS time integration of unsteady flow problems,
3. to substantiate the conjecture that the present MF–MG algorithms, whose implementation is turbulence-model-independent, achieve comparable CPU–speed-ups whatever the particular turbulence closure used.

After a brief description of the system of equations and of the turbulence model (Section 2), we briefly summarize (Section 3) the standard [17] non-time-consistent FAS–MF–MG algorithm (steady; s–MG), using iteration operators as building-blocks (cf. Appendix A), and assess its performance for (1) developing quasi-incompressible turbulent flow in a 3-D square duct ($M_{\mathrm{CL}_i} = 0.0516$), studied experimentally by Gessner and Emery [83], and (2) transonic ($M_{\mathrm{SW}} \in [1.3, 1.8]$) 3-D SWTBLI in a rectangular nozzle with a swept bump on the lower wall [84, 85].

Then (Section 4), we assess potentially time-consistent algorithms for the acceleration of the subiterative convergence (unsteady; u–MG), for a 2-D compression ramp ($M_{\mathrm{SW}} = 2.85$, $\alpha_c = \Delta\vartheta_{\mathrm{SW}} = 24\,\mathrm{deg}$, $Re_{\theta_0} = 8 \times 10^4$) studied experimentally by Settles *et al.* [86], Dolling and Murphy [87], Settles and Dodson [88].

DOI: 10.1002/fld

Computations for the various configurations were run at the experimental conditions [24, 25], [79, 80], and with quite fine computational grids (Table IV) especially for the 3-D test cases. The possibility of using computational grids, which are sufficiently fine to achieve grid-converged results in complex 3-D configurations is one of the major advantages expected from the development of computationally efficient MG techniques. For the 2-D test case we have systematically run computations using 2 wall normal free WNF–RSMs (the V RSM [82] and the WNF–LSS RSM [25]), and a low turbulence Reynolds number baseline $k-\varepsilon$ model [57]. For the 3-D test cases, because of the high CPU times required, especially for the SG computations, CPU-speed-ups were assessed for the V RSM [82] closure only.

2. RSM–RANS SYSTEM OF EQUATIONS

2.1. Meanflow equations

The flow is modelled by the compressible Favre–Reynolds-averaged 3-D Navier–Stokes equations [10, 24]

$$\frac{\partial \bar{\rho}}{\partial t}+\frac{\partial \bar{\rho}\tilde{u}_\ell}{\partial x_\ell}=0 \tag{1a}$$

$$\frac{\partial \bar{\rho}\tilde{u}_i}{\partial t}+\frac{\partial}{\partial x_\ell}[\bar{\rho}\tilde{u}_i\tilde{u}_\ell+\bar{p}\delta_{i\ell}]-\frac{\partial}{\partial x_\ell}[\bar{\tau}_{i\ell}-\overline{\rho u_i'' u_\ell''}]=0 \tag{1b}$$

$$\frac{\partial}{\partial t}[\bar{\rho}\check{h}_\mathrm{t}-\bar{p}]+\frac{\partial \bar{\rho}\tilde{u}_\ell\check{h}_\mathrm{t}}{\partial x_\ell}-\frac{\partial}{\partial x_\ell}[\tilde{u}_i(\bar{\tau}_{i\ell}-\overline{\rho u_i'' u_\ell''})-(\bar{q}_\ell+\overline{\rho h'' u_\ell''})]=S_{\check{h}_\mathrm{t}} \tag{1c}$$

where t is the time, x_ℓ are the Cartesian space coordinates, u_i are the velocity components, ρ is the density, p is the pressure, δ_{ij} is the Kronecker symbol, $\tilde{(\cdot)}$ denotes Favre averaging, $\bar{(\cdot)}$ denotes non-weighted averaging, $(\cdot)''$ are Favre fluctuations, $(\cdot)'$ are non-weighted fluctuations, $\check{h}_\mathrm{t}=\tilde{h}+\frac{1}{2}\tilde{u}_i\tilde{u}_i$ is the total enthalpy of the mean flow, h is the specific enthalpy, $-\overline{\rho u_i'' u_j''}\equiv-\bar{\rho}\widetilde{u_i'' u_j''}$ are the Reynolds stresses, $k=\frac{1}{2}\widetilde{u_i'' u_i''}$ is the turbulence kinetic energy (TKE), τ_{ij} are the viscous stresses, q_ℓ is the molecular heat flux, $\overline{\rho h'' u_\ell''}\equiv\bar{\rho}\widetilde{h'' u_\ell''}$ is the turbulent heat flux, and $S_{\check{h}_\mathrm{t}}$ is a source term appearing in the meanflow energy equation [10, 24]. The symbol $\check{(\cdot)}$ is used to denote a function of average quantities that is neither a Favre-average nor a non-weighted average. The source term appearing in the meanflow energy equation is obtained by combining the averaged energy equation with the transport equation for the TKE, and reads [10, 89, 90].

$$S_{\check{h}\mathrm{t}}=\left[\Pi_k-P_k+\bar{\rho}\varepsilon_k^{(\tau)}+\overline{u_i''}\frac{\partial \bar{p}}{\partial x_i}+\bar{\tau}_{i\ell}\frac{\partial \overline{u_i''}}{\partial x_\ell}\right] \tag{2}$$

where $\Pi_k=\frac{1}{2}\Pi_{\ell\ell}$ is the velocity–pressure gradient correlation in the TKE transport equation ($\Pi_{ij}=\overline{u_i'\partial_{x_j}p'+u_j'\partial_{x_i}p'}$), $P_k=\frac{1}{2}P_{\ell\ell}$ is the TKE production, and $\varepsilon_k^{(\tau)}=\frac{1}{2}\varepsilon_{\ell\ell}^{(\tau)}$ is the TKE dissipation rate.

DOI: 10.1002/fld

Table IV. Computational grids used for the 2-D and 3-D test-cases studied.

Configuration	Grid	N_i	N_j	N_k	N_{j_s}	r_j	y_w^+	N_{k_s}	r_k	z_w^+	L_x (m)	L_y (m)	L_z (m)
Settles 24° [86]	2-D	[79]	401	201		101	1.1176	0.3	—	—	—	0.31524	0.2	—
Delery [85]	3-D	[10, 24]	421	201	221	81	1.095	$\frac{3}{4}$	89	1.095	$\frac{3}{4}$	0.815	0.100	0.1213
Gessner-Emery [83]	3-D	[82]	801	149	149	96	1.067	$\frac{1}{2}$	96	1.067	$\frac{1}{2}$	25	0.127	0.127

i, j, k: grid directions; N_i, N_j, N_k: number of points in the x, y, z direction, respectively; y_w^+, z_w^+: nondimensional distance of the first grid node away from the wall; r_j, r_k: geometric progression ratio; N_{j_s}, N_{k_s}: number of nodes geometrically stretched near the wall [79]; L_x, L_y, L_z: lengths (m) of the computational box.

DOI: 10.1002/fld

2.2. Reynolds stress transport

The transport equations for the Favre–Reynolds averaged Reynolds stresses are [89, 91]

$$\underbrace{\frac{\partial\overline{\rho u_i'' u_j''}}{\partial t}+\frac{\partial}{\partial x_\ell}(\overline{\rho u_i'' u_j''}\tilde{u}_\ell)}_{\text{convection } C_{ij}}$$

$$=\underbrace{\frac{\partial}{\partial x_\ell}\left(-\overline{\rho u_i'' u_j'' u_\ell''}-\overline{p' u_j'}\delta_{i\ell}-\overline{p' u_i'}\delta_{j\ell}+\mu\overline{\frac{\partial u_i' u_j'}{\partial x_\ell}}\right)}_{\text{diffusion } d_{ij}=d_{ij}^{(u)}+d_{ij}^{(p)}+d_{ij}^{(\mu)}}$$

$$+\underbrace{\overline{p'\left(\frac{\partial u_i'}{\partial x_j}+\frac{\partial u_j'}{\partial x_i}-\frac{2}{3}\frac{\partial u_k'}{\partial x_k}\delta_{ij}\right)}}_{\text{redistribution } \phi_{ij}}+\underbrace{\left(-\overline{\rho u_i'' u_\ell''}\frac{\partial\tilde{u}_j}{\partial x_\ell}-\overline{\rho u_j'' u_\ell''}\frac{\partial\tilde{u}_i}{\partial x_\ell}\right)}_{\text{production } P_{ij}}+\underbrace{\frac{2}{3}\overline{p'\frac{\partial u_k'}{\partial x_k}}\delta_{ij}}_{\text{pressure-dilatation } \frac{2}{3}\phi_p\delta_{ij}}$$

$$+\underbrace{\left(-\overline{u_i''}\frac{\partial\bar{p}}{\partial x_j}-\overline{u_j''}\frac{\partial\bar{p}}{\partial x_i}+\overline{u_i''}\frac{\partial\bar{\tau}_{j\ell}}{\partial x_\ell}+\overline{u_j''}\frac{\partial\bar{\tau}_{i\ell}}{\partial x_\ell}\right)}_{\text{density fluctuation effects } K_{ij}}-\underbrace{\left(\frac{\partial}{\partial x_\ell}[\mu\overline{\frac{\partial u_i' u_j'}{\partial x_\ell}}]-(\overline{u_i'\frac{\partial\tau_{j\ell}'}{\partial x_\ell}+u_j'\frac{\partial\tau_{i\ell}'}{\partial x_\ell}})\right)}_{\text{dissipation } \bar{\rho}\varepsilon_{ij}^{(\mu)}} \quad (3)$$

This form, where a nearly exact term for the viscous diffusion of the Reynolds stresses ($d_{ij}^{(\mu)}$) is used, instead of the original term ($d_{ij}^{(\tau)}$), is the form that is actually modelled [92]. Convection C_{ij} and production P_{ij} are exact terms, while all the other terms ($d_{ij}^{(\mu)}$, $d_{ij}^{(u)}$, $d_{ij}^{(p)}$, ϕ_{ij}, ϕ_p, K_{ij}, and $\varepsilon_{ij}^{(\mu)}$) require modelling.

These equations were closed using WNF RSMs, i.e. SMC which are completely independent of wall topology (distance-from-the-wall vector) [25]. The homogeneous part of the redistribution tensor ϕ_{ij} is modelled [24, 25, 82] by an isotropization-of-production/return-to-isotropy (IPRI) closure, while the strongly inhomogeneous near-wall part is modelled by a WNF approach [25]. The turbulence scale is determined by solving a transport equation for the modified [57] TKE dissipation rate ε^*. The details on the development of the models have been reported elsewhere [24–26, 81, 82] and are beyond the scope of the present paper.

2.3. Numerical model

In summary the flow is modelled by a system of 12 nonlinear evolution equations [24–26, 81, 82, 93]

$$\frac{\partial\underline{u}}{\partial t}+\frac{\partial\underline{F}_\ell}{\partial x_\ell}+\underline{S}=0 \quad (4)$$

where $\underline{u}\in\mathbb{R}^{12}$ is the vector of unknowns (conservative variables), which is split in a vector of meanflow variables $\underline{u}_{\mathrm{MF}}\in\mathbb{R}^5$, and a vector of turbulence variables (Reynolds stresses and

DOI: 10.1002/fld

dissipation rate) $\underline{u}_{\mathrm{RSM}} \in \mathbb{R}^7$

$$\begin{aligned}\underline{u} &= [\underline{u}_{\mathrm{MF}}^{\mathrm{T}}, \underline{u}_{\mathrm{RSM}}^{\mathrm{T}}]^{\mathrm{T}} \\ &= [[\bar{\rho}, \bar{\rho}\tilde{u}, \bar{\rho}\tilde{v}, \bar{\rho}\tilde{w}, \bar{\rho}\breve{h}_{\mathrm{t}} - \bar{p}]; [\overline{\rho u'' u''}, \overline{\rho u'' v''}, \overline{\rho v'' v''}, \overline{\rho v'' w''}, \overline{\rho w'' w''}, \overline{\rho w'' u''}, \bar{\rho}\varepsilon^*]]^{\mathrm{T}}\end{aligned} \tag{5}$$

The fluxes $\underline{F}_\ell \in \mathbb{R}^{12}$ ($\underline{F}_x$, $\underline{F}_y$, $\underline{F}_z$) are the combined convective ($\underline{F}_\ell^{\mathrm{C}}$) and diffusive (viscous; $\underline{F}_\ell^{\mathrm{V}}$) fluxes

$$\underline{F}_\ell = \underline{F}_\ell^{\mathrm{C}} + \underline{F}_\ell^{\mathrm{V}} = \begin{bmatrix} \bar{\rho}\tilde{u}_\ell \\ \bar{\rho}\tilde{u}_\ell\tilde{u} + \bar{p}\delta_{x\ell} \\ \bar{\rho}\tilde{u}_\ell\tilde{v} + \bar{p}\delta_{y\ell} \\ \bar{\rho}\tilde{u}_\ell\tilde{w} + \bar{p}\delta_{z\ell} \\ \bar{\rho}\tilde{u}_\ell\breve{h}\mathrm{t} \\ \tilde{u}_\ell\overline{\rho u'' u''} \\ \tilde{u}_\ell\overline{\rho u'' v''} \\ \tilde{u}_\ell\overline{\rho v'' v''} \\ \tilde{u}_\ell\overline{\rho v'' w''} \\ \tilde{u}_\ell\overline{\rho w'' w''} \\ \tilde{u}_\ell\overline{\rho w'' u''} \\ \bar{\rho}\tilde{u}_\ell\varepsilon^* \end{bmatrix} + \begin{bmatrix} 0 \\ \overline{\rho u_\ell'' u''} - \bar{\tau}_{\ell x} \\ \overline{\rho u_\ell'' v''} - \bar{\tau}_{\ell y} \\ \overline{\rho u_\ell'' w''} - \bar{\tau}_{\ell z} \\ \tilde{u}_i(\overline{\rho u_i'' u_\ell''} - \bar{\tau}_{i\ell}) + (\bar{q}_\ell + \overline{\rho h'' u_\ell''}) \\ -\mathscr{D}_{xx\ell} \\ -\mathscr{D}_{xy\ell} \\ -\mathscr{D}_{yy\ell} \\ -\mathscr{D}_{yz\ell} \\ -\mathscr{D}_{zz\ell} \\ -\mathscr{D}_{zx\ell} \\ -\mathscr{D}_{\varepsilon_\ell} \end{bmatrix} \tag{6}$$

where $\mathscr{D}_{ijk}$ denote the modelled form of the diffusion terms [82] in the Reynolds stress transport (Equations (3)) and $\mathscr{D}_{\varepsilon_i}$ denotes the ε diffusion flux [24]. The source terms vector $\underline{S} \in \mathbb{R}^{12}$

$$\underline{S} = -[0, 0, 0, 0, S_{\breve{h}\mathrm{t}}, S_{uu}, S_{uv}, S_{vv}, S_{vw}, S_{ww}, S_{wu}, S_\varepsilon]^{\mathrm{T}} \tag{7}$$

contains the source terms of the turbulence model ($S_{u_i u_j}$ and S_ε), and the modelled form of the energy equation source term $S_{\breve{h}_{\mathrm{t}}}$ (Equation (2)).

The working medium thermodynamics are approximated by a thermodynamically and calorically perfect gas ($\bar{p} = \bar{\rho} R_g \tilde{T}$ and $c_p = \gamma R_g/[\gamma - 1]$). All computations were run for air ($R_g = 287.04\,\mathrm{m^2\,s^{-2}\,K^{-1}}, \gamma = 1.4$) and used a Sutherland law dependence for viscosity and a corrected Sutherland law for heat conductivity [93, p. 200, Equations (5)].

Notice that, contrary to the above practice (which is in quite general use, and has been followed in the present paper), the Reynolds stresses in the mean-momentum equations, and the term $\tilde{u}_i\overline{\rho u_i'' u_\ell''}$ in the energy equation, should have been included in the convective part of Equations [94], because they are order-1 derivatives.

3. BASIC MG ALGORITHM

3.1. Meanflow–Multigrid

MG is applied on meanflow variables only, while turbulence variables are simply injected on coarser grids, and are only updated on the fine grid (no MG residuals for the turbulence variables). The

DOI: 10.1002/fld

prolongation and restriction operators are based on characteristic MG [78]. The restriction (transfer) operator from the fine to the coarse grid, for the variables, $\mathsf{T}^{2h}_{\mathbf{u};h}$, is a simple injection operator. The restriction (transfer) operator from the fine to the coarse grid, for the residuals, $\mathsf{T}^{2h}_{\mathbf{R};h}$, is a weighted operator respecting the direction of propagation of information along characteristics [78]. The prolongation (interpolation) operator, which is applied only to $\mathbf{u}_{\mathrm{MF}}$, is a simple geometric interpolation operator [78]. The coarse grid scheme is applied on $\mathbf{u}_{\mathrm{MF}}$ only, with frozen turbulence. Full details concerning the restriction and interpolation operators, on structured grids, are given in Gerolymos and Vallet [17]. The MG algorithms are defined using basic operators related to the SG scheme, which is described for completeness in Appendix A.

3.2. *s-V*$(N_{\mathrm{V}_1},0)$ *sawtooth MG cycle*

The steady s–MG algorithm [17] is a standard non-time-consistent FAS method, based on a $\mathrm{V}(N_{\mathrm{V}_1},0)$-cycle (sawtooth cycle), with appropriate forcing terms [74, 77]. Let $\mathscr{G}_h$, $\mathscr{G}_{2h}$, $\mathscr{G}_{4h}$ denote the fine and successively coarser grids. The full-iteration operator N (Equations (A10), (A11)), and the full-iteration frozen-turbulence operator N_{MF} (Equations (A15), (A16)), can be defined on each grid in the MG sequence, $\mathscr{G}_h$ (N_h, $\mathsf{N}_{\mathrm{MF}_h}$), $\mathscr{G}_{2h}$ (N_{2h}, $\mathsf{N}_{\mathrm{MF}_{2h}}$), and $\mathscr{G}_{4h}$ (N_{4h}, $\mathsf{N}_{\mathrm{MF}_{4h}}$). These operators correspond to K_{it} iterations (time steps), with each time step containing M_{it} subiterations (usually determined dynamically by a target convergence-tolerance r_{TRG} criterion), and are defined by the non-dimensional physical time step CFL and the non-dimensional dual time step CFL*, and advance the solution from a given value ${}^0\mathbf{u}$ to a new value $\mathsf{N}({}^0\mathbf{u},\mathrm{CFL},\mathrm{CFL}^*,\mathbf{F};r_{\mathrm{TRG}},K_{\mathrm{it}})$, where $\mathbf{F}$ is the MG forcing term [74, 58], which is 0 on the fine grid. With these definitions, the MG algorithm, for three levels of MG ($L_{\mathrm{GRD}}=3$), and for a dynamic subiterative strategy, reads

$$
\begin{aligned}
&\text{iteration } n\\
&\text{compute on } \mathscr{G}_h: \quad {}^0\mathbf{q}_h={}^n\mathbf{u};\ \mathbf{F}_h=0;\ \mathbf{q}_h=\mathsf{N}_h({}^0\mathbf{q}_h,\mathrm{CFL},\mathrm{CFL}^*,\mathbf{F}_h;r_{\mathrm{TRG}},1)\\
&\text{transfer to } \mathscr{G}_{2h}: \quad \begin{cases} {}^0\mathbf{q}_{2h}=\mathsf{T}^{2h}_{\mathbf{u};h}[\mathbf{q}_h]\\ \mathbf{F}_{\mathrm{MF}_{2h}}=\mathsf{T}^{2h}_{\mathbf{R};h}[\mathbf{L}_{\mathrm{MF}_h}(\mathbf{q}_h)]-\mathbf{L}_{\mathrm{MF}_{2h}}({}^0\mathbf{q}_{2h}) \end{cases}\\
&\text{compute on } \mathscr{G}_{2h}: \quad \begin{cases} \mathbf{q}_{\mathrm{MF}_{2h}}=\mathsf{N}_{\mathrm{MF}_{2h}}({}^0\mathbf{q}_{2h},\mathrm{CFL},\mathrm{CFL}^*,\mathbf{F}_{\mathrm{MF}_{2h}};r_{\mathrm{TRG}},N_{\mathrm{V}_1})\\ \mathbf{q}_{\mathrm{RSM}_{2\mathrm{h}}}={}^0\mathbf{q}_{\mathrm{RSM}_{2h}}=\mathsf{T}^{2h}_{\mathbf{u};h}[\mathbf{q}_{\mathrm{RSM}_h}] \end{cases}\\
&\text{transfer to } \mathscr{G}_{4h}: \quad \begin{cases} {}^0\mathbf{q}_{4h}=\mathsf{T}^{4h}_{\mathbf{u};2h}[\mathbf{q}_{2h}]\\ \mathbf{F}_{\mathrm{MF}_{4h}}=\mathsf{T}^{4h}_{\mathbf{R};2h}[\mathbf{L}_{\mathrm{MF}_{2h}}(\mathbf{q}_{2h})+\mathbf{F}_{\mathrm{MF}_{2h}}]-\mathbf{L}_{\mathrm{MF}_{4h}}({}^0\mathbf{q}_{4h}) \end{cases}\\
&\text{compute on } \mathscr{G}_{4h}: \quad \begin{cases} \mathbf{q}_{\mathrm{MF}_{4h}}=\mathsf{N}_{\mathrm{MF}_{4h}}({}^0\mathbf{q}_{4h},\mathrm{CFL},\mathrm{CFL}^*,\mathbf{F}_{\mathrm{MF}_{4h}};r_{\mathrm{TRG}},N_{\mathrm{V}_1})\\ \mathbf{q}_{\mathrm{RSM}_{4h}}={}^0\mathbf{q}_{\mathrm{RSM}_{4h}}=\mathsf{T}^{4h}_{\mathbf{u};2h}[\mathbf{q}_{\mathrm{RSM}_{2\mathrm{h}}}]=\mathsf{T}^{4h}_{\mathbf{u};2h}\mathsf{T}_{\mathbf{u};\mathrm{h}^2h}[\mathbf{q}_{\mathrm{RSM}_h}] \end{cases}\\
&\text{interpolation}: \quad {}^{n+1}\mathbf{u}=\mathsf{B}_h\{\mathbf{q}_h+\mathsf{I}^h_{2\mathrm{h}}[\mathbf{q}_{2h}-{}^0\mathbf{q}_{2h}]+\mathsf{I}^h_{2h}\mathsf{I}^{2h}_{4h}[\mathbf{q}_{4h}-{}^0\mathbf{q}_{4h}]\}
\end{aligned}
\tag{8}
$$

In these relations ${}^0\mathbf{q}$, $\mathbf{q}$, and $\mathbf{F}$ are internal variables, and were not superscripted by the iteration counter. The variables ${}^0\mathbf{q}_h$, ${}^0\mathbf{q}_{2h}$, and ${}^0\mathbf{q}_{4h}$, are initial values, at the beginning of the computation on the corresponding grid (they are obtained by transfer from the immediately finer grid). The

DOI: 10.1002/fld

variables $\mathbf{q}_h$, $\mathbf{q}_{2h}$, and $\mathbf{q}_{4h}$ are final values, obtained on the corresponding grid from the application of one full iteration N on the fine grid ($\mathbf{q}_h$), and of N_{V_1} meanflow iterations N_{MF} on the coarser grids ($\mathbf{q}_{2h}$ and $\mathbf{q}_{4h}$). The quantities $\mathbf{F}_h \equiv 0$, $\mathbf{F}_{\mathrm{MF}_{2h}}$, and $\mathbf{F}_{\mathrm{MF}_{4h}}$ are the forcing terms on the corresponding grids, ensuring that the procedure is driven by the fine-grid residuals, to obtain MG results identical with fine-grid SG computations [58, 74, 77].

For a higher number of grids the $\mathscr{G}_{4h}$ step would be repeated on $\mathscr{G}_{8h}$. Nonetheless only computations with three levels of MG (h, $2h$, $4h$) will be presented in the following. Tests with four levels of MG, using the present method of characteristic MG and a V-cycle (sawtooth cycle), exhibit stability problems, even in the case of simple subsonic flows. It is believed that this instability is in no way related to turbulence modelling. This conjecture is substantiated, e.g. by the fact that solution breakdown with $L_{\mathrm{GRD}}=4$, for the Settles *et al.* [86] Settles and Dodson [88] $M_{\mathrm{SW}}=2.85$ oblique shock wave/boundary layer interaction (OSWTBLI) test case (cf. Section 4.4), occurs well outside the boundary layer, near the shock wave, in the essentially inviscid flow region. This breakdown occurs quite rapidly, even when the $L_{\mathrm{GRD}}=4$ computations are initialized using fully converged $L_{\mathrm{GRD}}=3$ computations. Theoretical work on characteristic MG with 4 grid-levels ($L_{\mathrm{GRD}}=4$) is required to further analyse this problem. The original work of Leclercq and Stouffflet [78] was limited to a 2-grid analysis, of the 1-D advection equation, with Runge–Kutta (RK) time integration. It concluded that the best results were obtained when one of the operators for the residuals (either prolongation or restriction) was upwind and the other one centered. How these results apply to a 4-grid analysis of the 2-D advection equation is the subject of on-going work.

3.3. Application to 3-D developing anisotropy-driven turbulent flow

The major drawback of the present MF–MG technique is that MG acceleration is not applied on the turbulence variables $\underline{u}_{\mathrm{RSM}}$ (Equation (5)). Several authors [20, 34, 36, 41–43, 49] argue that this will significantly reduce the computational speed-up, because the pace will be set by the slowly convergent $\underline{u}_{\mathrm{RSM}}$. As noted in the Introduction (Section 1), Gerlinger and Brüggemann [42] have quantified this reduction in CPU-speed-up, showing that FCSF–MG can be faster by a factor of 2 compared with MF–MG.

To further investigate this issue, we study developing 3-D turbulent flow in a square duct, investigated experimentally by Gessner and Emery [83]. The experimental configuration [83, 95] consists of a square duct, with quasi-incompressible (inlet Mach number at centerline $M_{\mathrm{CL}_i}=0.0516$) developing turbulent flow at bulk Reynolds number $Re_{\mathrm{B}}=250\,000$ ($Re_{\mathrm{B}}=\bar{u}_{\mathrm{B}}D_h\breve{\nu}^{-1}$, where $\bar{u}_{\mathrm{B}}$ is the bulk velocity, $D_h=2\mathrm{a}$ is the hydraulic diameter of the duct and $\breve{\nu}$ is the kinematic viscosity). Preliminary tests showed that very fine grids were needed to obtain grid-converged results. Computations were run on a 18×10^6 points grid (Table IV), discretizing $\frac{1}{4}$ of the duct, with symmetry conditions at the y- and z-wise symmetry planes [82]. The inflow boundary layer thickness was adjusted (Figure 1) to obtain a close fit to the experimental centerline velocity in the entrance region ($x\in[0, 10D_h]$).

This flow is dominated by anisotropy-driven secondary-flows, and as such is a stringent test case as far as MF–MG is concerned. The sensitivity of the results on the turbulence model used, for this particular test case, has been highlighted by several previous studies on turbulence modelling [25, 82, 95]. These studies have not only confirmed the inadequacy of closures using the Boussinesq assumption, but have also highlighted the importance of both redistribution (ϕ_{ij}; Equations (3)) and turbulence diffusion ($d_{ij}^{(u)}+d_{ij}^{(p)}$; Equations (3)) for the correct prediction of

DOI: 10.1002/fld

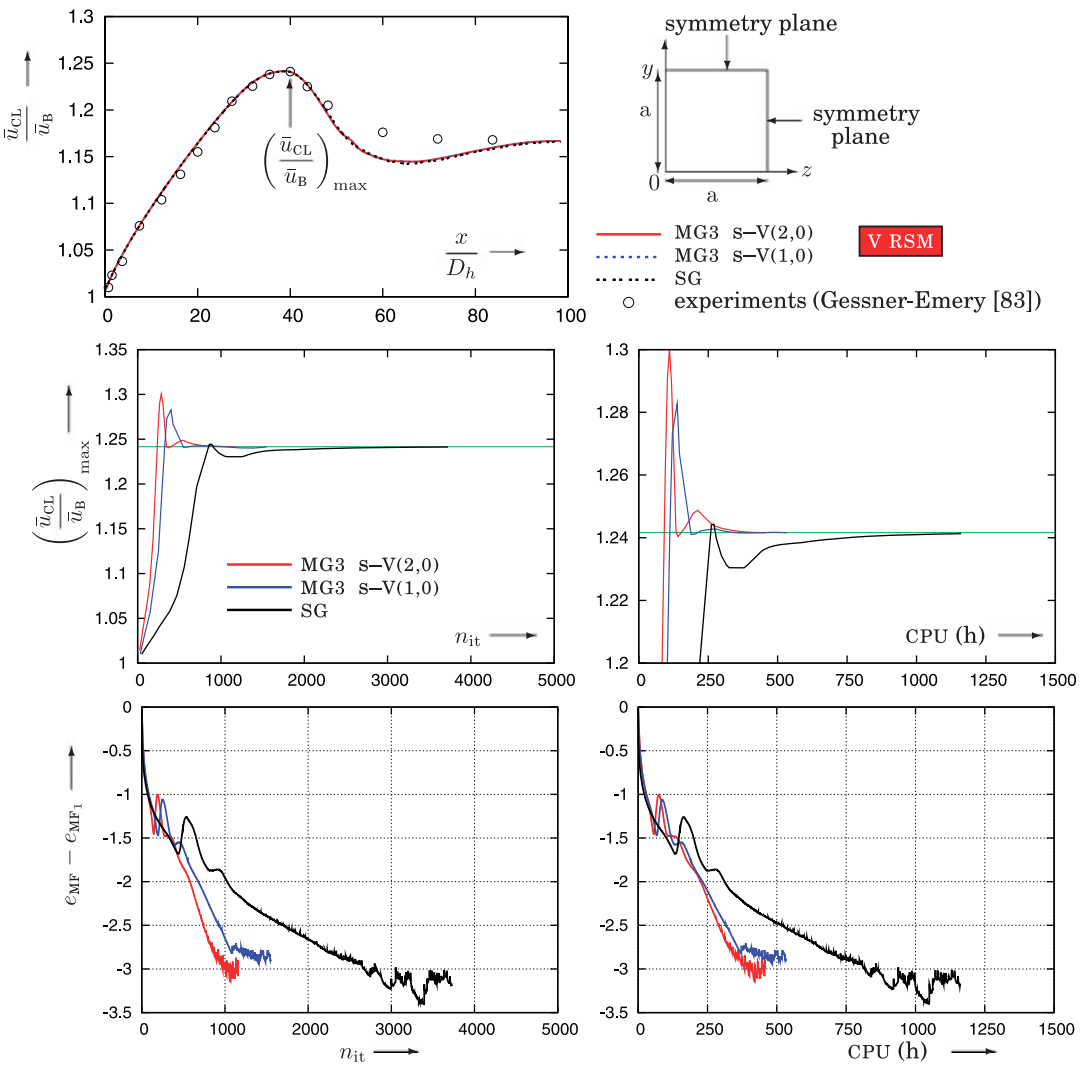

Figure 1. Comparison, for [CFL, CFL*; $M_{\text{it}}, r_{\text{TRG}}$] = [500, 50; −, −1.5], of various multigrid techniques, using 3 levels of multigrid (MG3), with the single-grid algorithm SG, for the convergence of mean-flow error e_{MF} and of maximum centerline velocity $[\bar{u}_{\text{CL}}]_{\max}$ ($x/D_h = 40$; $y = z = \text{a}$), *vs* the number of iterations n_{it} (multigrid-cycles) and *vs* CPU-time (for 3.7 Gflop s^{-1} sustained performance), using the Vallet [82] RSM, for developing turbulent flow in a square duct [83] ($Re_{\text{B}} = 250\,000$; $T_{u_i} = 1\%$; $\ell_{\text{T}_i} = 50$ mm; $\delta_{y_i} = \delta_{z_i} = 0.1$ mm; $\Delta y_w^+ = \Delta z_w^+ < \frac{1}{2}$; 18×10^6 points grid discretizing $\frac{1}{4}$ of the square duct; Table IV; comparison of computed and measured [83] centerline velocity $\bar{u}_{\text{CL}}$).

such global flow features as the duct-centerline velocity $\bar{u}_{\text{CL}}$, which is a direct measure of the blockage of the boundary layers on the duct walls and duct corners. This sensitivity of the results on the predicted $\underline{u}_{\text{RSM}}$ (Equation (5)) suggests that any convergence delay of the turbulence variables should have a significant impact on the overall convergence of the flow. Computations were

DOI: 10.1002/fld

run with a dynamic subiterative strategy ([CFL, CFL*; $M_{\text{it}}, r_{\text{TRG}}$] = [500, 50; −, −1.5]), using the V RSM [82]. CPU-speed-ups were assessed (Figure 1) by comparing SG and MG3 computations, both s-MG3-V(1,0) and s-MG3-V(2,0).

Consideration of the reduction of meanflow error e_{MF} relative to its value at $n_{\text{it}} = 1$ (e_{MF_1}) illustrates the CPU-speed-ups of the MG procedures (Figure 1). As for the Délery_3-D test case (Section 3.4), an alternative criterion was used to evaluate CPU-speed-ups, based on the convergence of a difficult (slowly convergent) flow feature. The Gessner and Emery [83] configuration is dominated by the developing 3-D turbulent boundary layers entrainment, and associated blockage. The peak in the centerline-velocity distribution at $x \cong 40 D_h$ (Figure 1) is a slowly convergent flow feature, and its correct prediction is an indicator of turbulence model performance. This quantity ($[\bar{u}_{\text{CL}}]_{\max}$) was chosen to evaluate CPU-speed-ups (Figure 1). Convergence for $[\bar{u}_{\text{CL}}]_{\max}$ with the s-MG3- V(1,0) strategy is reached within ∼340CPU − h (Figure 1). The s-MG3-V(2,0) computations converge in ∼380CPU − h (Figure 1), for $[\bar{u}_{\text{CL}}]_{\max}$, while the residuals plots suggest a faster convergence using s-MG3-V(2,0) compared with s-MG3–V(1,0). The SG computations reach a similar level of convergence as the s-MG3 strategies in ∼1160CPU − h. MG CPU-speed-ups are (Table V)3.4 for s-MG3-V(1,0) and 3.1 for s-MG3-V(2,0).

Notice that for this configuration, CPU-speed-ups are in general lower than those obtained for the previous test cases, presumably because MF–MG does not handle efficiently this flow, which is dominated by turbulence anisotropy. Nonetheless, CPU-speed-ups are quite substantial, especially taking into account the very large SG CPU-time. Notice also that, despite the absence of low Mach number preconditioning, the method performs quite well for this low Mach number ($M_{\text{CL}_i} = 0.0516$) flow.

Comparison of computed results with measurements [83], along the corner-bisector (Figure 2), demonstrate that the s-MG3 computations give identical results with the SG calculations, both for the mean-velocities and for the Reynolds stresses, in good agreement with measurements (Figure 2).

3.4. Application to 3-D transonic flow

To substantiate the performance of the MF–MG algorithm for 3-D compressible flows with large separation, computations were run for the Délery_3-D transonic nozzle [84, 85, 96], which is a rectangular nozzle with plane parallel sidewalls, where a swept bump fitted on the lower wall induces a 3-D SWTBLI ($M_{\text{SW}} \in [1.3, 1.8]$). A large 3-D separation region is observed in the neighbourhood of the corner between the lower wall and the near wall ($z = 106.3$mm), where the shock wave is strongest (Figure 3). The adjustable throat at the nozzle exit (Figure 3), which serves to generate and position the shock wave, is located relatively near the trailing edge of the swept bump, and strongly influences the separation and reattachment process [10, 24]. Furthermore, it is not improbable that technological details of the experimental set-up, and possible leakage at the exit throat, also influence the flow (the adjustable exit throat is included in the computed geometry, but not the technological/leakage effects). Previous computations of this configuration [10, 24] with various turbulence closures have shown the improvements in predicting this complex flow using SMCs.

Computations were run with a static subiterative strategy ([CFL, CFL*; $M_{\text{it}}, r_{\text{TRG}}$] = [100, 10; 4, −]), using the V RSM [82], on a grid of 19×10^6 points (Table IV), which is quite finer than the grids used in the literature for this configuration [10, 24, 96]. CPU-speed-ups were assessed (Figure 3) by comparing SG and MG3 computations, both s-MG3-V(1,0) and s-MG3-V(2,0).

DOI: 10.1002/fld

Table V. Summary of computational performance of the meanflow–multigrid algorithms, with three levels of multigrid ($L_{\mathrm{GRD}}=3$).

			LDTS-strategy		CPU-speed-up			
Case	Grid	Model	[CFL, CFL*; M_{it}, r_{TRG}]	Criterion	u–V(1,0)	u$'$–V(1,0)	s–V(1,0)	s–V(2,0)
2D OSWTBLI [86, 88]	401×201	LS $k-\varepsilon$ [57]	[100, 10; 4, −]	(1)	2.3	4.4	5	7.7
		WNF–LSS–RSM [25]		(1)	1.7	3.8	3.8	5.4
		V–RSM [82]		(1)	1.7	3.1	3.4	4.6
		LS $k-\varepsilon$ [57]	[∞, 10; −, −1]	(1)	3.9	3.9	3.9	7.1
		WNF–LSS–RSM [25]		(1)	3.5	3.5	3.9	4.4
		V–RSM [82]		(1)	3.5	3.5	4.0	4.7
3-D SWTBLI [85]	$421\times201\times221$	V–RSM [82]	[100, 10; 4, −]	(2)	na	na	3.7	5.5
3-D square duct [83]	$801\times149\times149$	V–RSM [82]	[500, 50; −, −1.5]	(3)	na	na	3.4	3.1

(1): Reach saturation of residuals, at $e_{\mathrm{MF}}\in[-6,-5]$ (Figures 4–5).
(2): Reach full convergence of $[\check{M}_{\mathrm{is}}]_{\mathrm{PLATEAU}}$ in the region of large separation, caused by the interaction of the $M_{\mathrm{SW}}=1.8$ shock wave with the corner boundary-layer (Figure 3).
(3): Reach full convergence of maximum centerline velocity $(\bar{u}_{\mathrm{CL}})_{\max}$ (Figure 1).
CPU-speed-up: actual speed-up in CPU-time (to reach the corresponding convergence criterion); na: not available.

DOI: 10.1002/fld

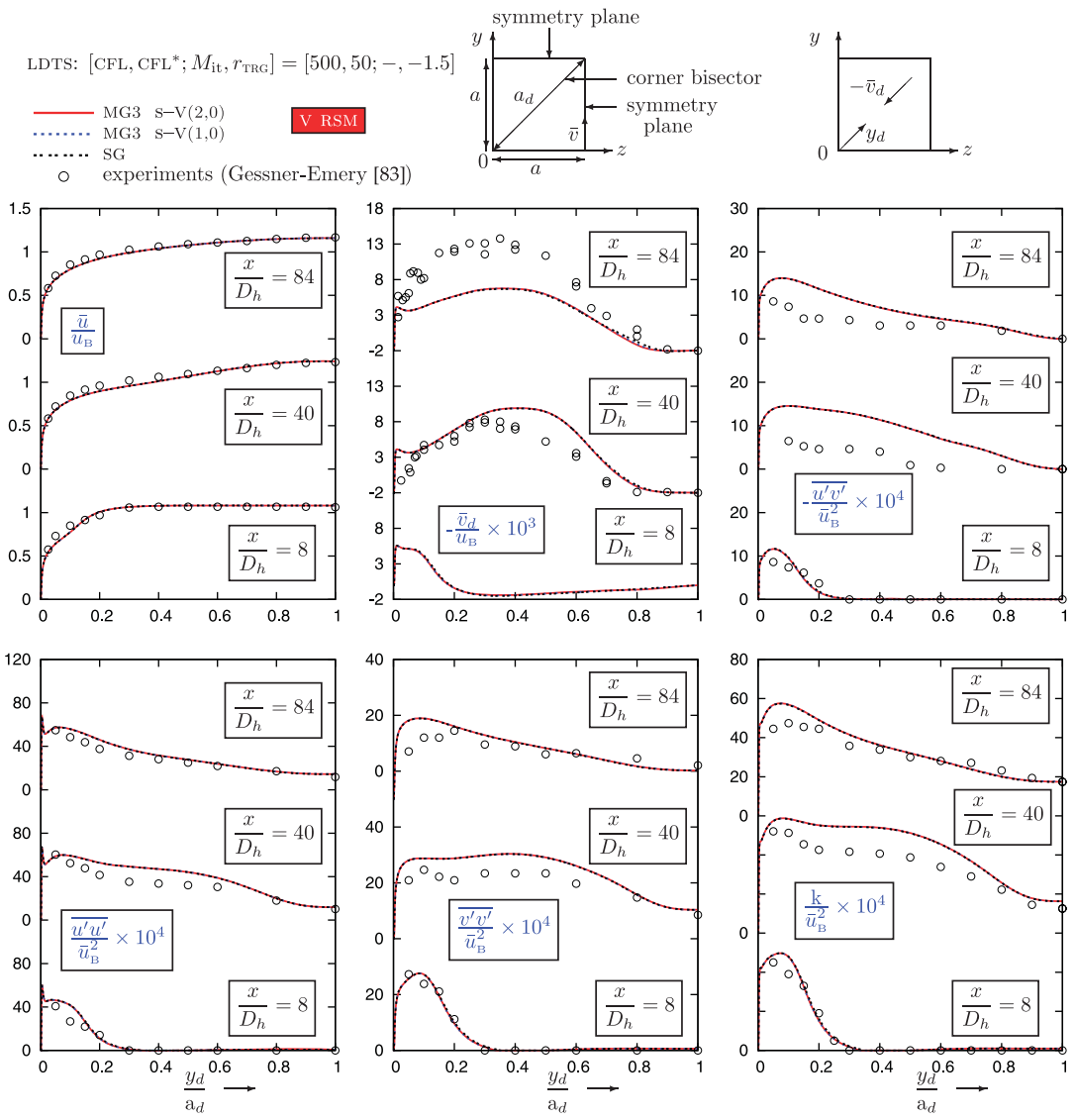

Figure 2. Comparison, for [CFL, CFL*; M_{it}, r_{TRG}] = [500, 50; −, −1.5], of results using the Vallet [82] RSM and 3 levels of multigrid (MG3), with the single-grid algorithm SG, and with measurements [83] along the corner-bisector (y_d), for developing turbulent flow in a square duct [83] ($Re_B = 250000$; $T_{u_i} = 1\%$; $\ell_{T_i} = 50\,\text{mm}$; $\delta_{y_i} = \delta_{z_i} = 0.1\,\text{mm}$; $\Delta y_w^+ = \Delta z_w^+ < \frac{1}{2}$; 18×10^6 points grid discretizing $\frac{1}{4}$ of the square duct; Table IV).

Consideration of the reduction of meanflow error e_{MF} relative to its value at $n_{it} = 1$ (e_{MF_1}) illustrates the CPU-speed-ups of the MG procedures (Figure 3). If saturation of residuals were used as the criterion for evaluating the CPU-speed-ups, it would have been necessary to further continue

DOI: 10.1002/fld

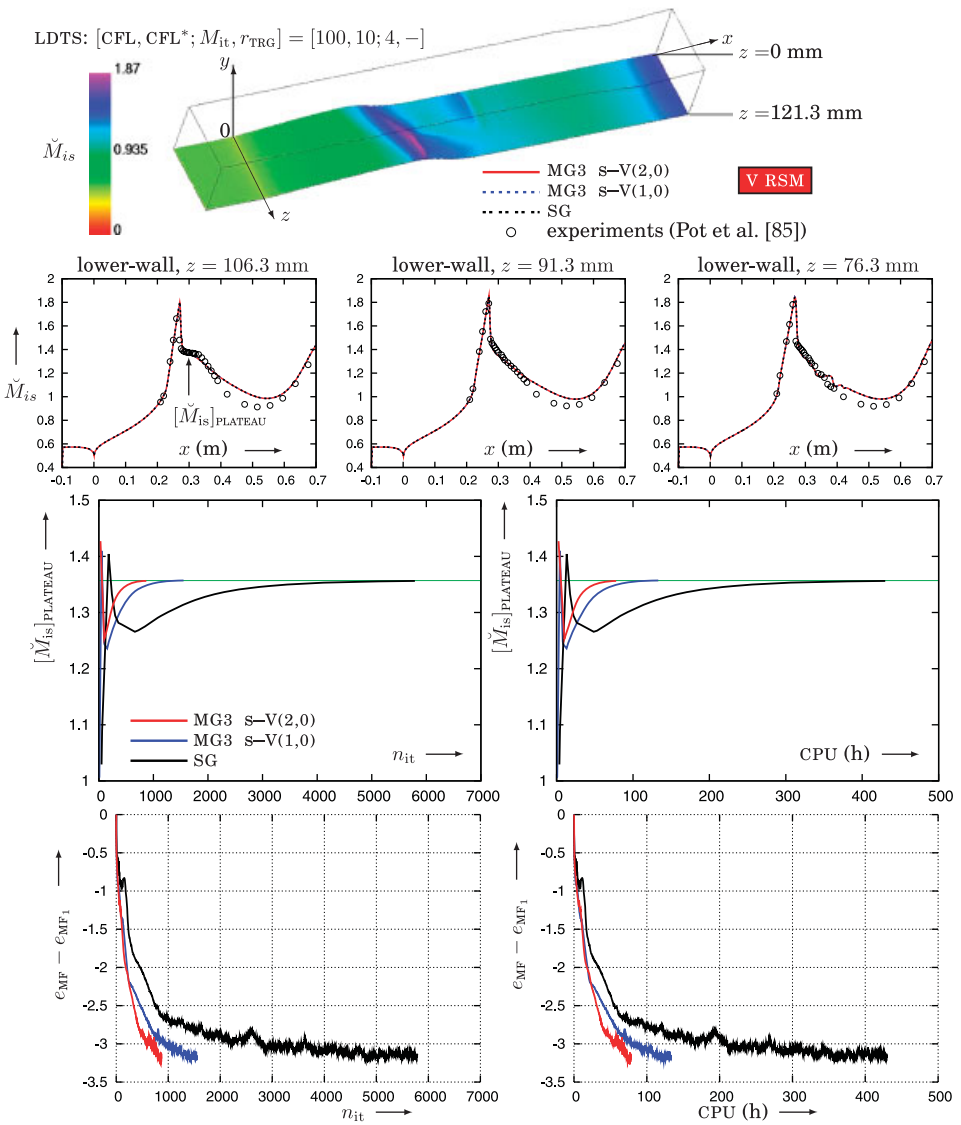

Figure 3. Comparison, for [CFL, CFL*; M_{it}, r_{TRG}] = [100, 10; 4, −], of various multigrid techniques, using 3 levels of multigrid (MG3), with the single-grid algorithm SG, for the convergence of meanflow error e_{MF} and of the isentropic-wall-Mach-number plateau level $[\check{M}_{is}]_{PLATEAU}$, *vs* the number of iterations n_{it} (multigrid-cycles) and *vs* CPU-time (for 3.7 Gflop s^{-1} sustained performance), using the V [82] RSM, for 3-D shock-wave/turbulent-boundary-layer-interaction in a rectangular nozzle [85] ($M_{SW} \in [1.3, 1.8]$; $Re_\chi \times 10^{-6} \in [2.2, 3.3]$; $T_{u_i} = 5\%$; $\ell_{T_i} = 50$ mm; $\delta_{y_i} = \delta_{z_i} = 0$ mm; $\Delta y_w^+ = \Delta z_w^+ < \frac{3}{4}$; 19×10^6 points grid; Table IV; comparison of computed and measured [85] wall pressures on the lower wall; $\check{M}_{is}$ at various locations and $\check{M}_{is}$ contours on the lower wall).

DOI: 10.1002/fld

the computations (especially SG). To avoid this, an alternative criterion was used, based on the convergence of a difficult (slowly convergent) flow feature. The flowfield contains both supersonic attached regions which converge very fast, and the separation region where convergence is much slower. This is especially the case in the large separation zone observed near the corner between the lower wall and the near wall ($z=106.3$ mm), where the shock wave is strongest (Figure 3). Previous computational experience with this configuration suggests [10, 24] that the associated pressure plateau present in the $\breve{M}_{\rm is}$ (isentropic wall Mach number [10]) distribution at $z=106.3$ mm converges slowly, because the associated recirculation is strongly influenced by overall separation and blockage. Using $[\breve{M}_{\rm is}]_{\rm PLATEAU}$ to monitor convergence highlights MG speed-up (Figure 3, Table V). For the s-MG3–V(1,0) strategy the CPU-speed-up is $\sim$3.7, and increases to $\sim$5.5 for the s-MG3-V(2,0) strategy (Figure 3, Table V).

4. ALTERNATIVE MG ALGORITHMS

4.1. Alternative strategies

The above studies (Section 3) confirm previous results [17, 18] on the satisfactory performance of MF–MG with the standard FAS algorithm, even for the difficult case of low Mach number anisotropy-driven turbulent flow (Section 3.3). The s–MG algorithm, with a V(2, 0) cycle gives speed-ups $r_{\rm CPUSUP}\in[3,5]$ for a wide range of complex flows (Section 3). The steady s–MG algorithm uses the baseline single-grid SG algorithm, with the subiterative procedure applied successively on each coarser grid. Alternatively, MG acceleration can be applied at the subiteration level, i.e. with the MG sequence visited at each subiteration. This approach, aiming at the acceleration of the subiterative convergence of the increment, can be made time-consistent, although, in the present work, we are only interested in steady flows. We have studied two alternative MG algorithms:

1. unsteady u–MG: MG acceleration of the subiterative procedure, which can be made time-consistent for unsteady flows computation (in the present work it is used with local-time steps for steady flows),
2. u′–MG: non-time-consistent variant of u–MG, potentially more efficient for steady flows.

These algorithms are applied (Section 4.4) to the computation of 2-D oblique-shock wave/turbulent boundary layer separated flow interaction (OSWTBLI) on a $M_{\rm SW}=2.85$ compression ramp [86–88], and are also compared with the s–MG algorithm (Section 3).

The u–MG and u′–MG algorithms are defined using the subiteration operator S (Equation (A3)), and the corresponding frozen-turbulence subiteration operator $\mathsf{S}_{\rm MF}$ (Equation (A12)) which can be defined on each grid in the MG sequence, $\mathscr{G}_h$ (S_h and $\mathsf{S}_{{\rm MF}_h}$), $\mathscr{G}_{2h}$ (S_{2h} and $\mathsf{S}_{{\rm MF}_{2h}}$), and $\mathscr{G}_{4h}$ (S_{4h} and $\mathsf{S}_{{\rm MF}_{4h}}$). These operators correspond to one subiteration and are defined by the physical time step Δt and the dual pseudo-time step Δt^*. They advance the subiterative solution from ${}^{m,n+1}\mathbf{u}$ to a new value ${}^{m+1,n+1}\mathbf{u}=\mathsf{S}({}^{m,n+1}\mathbf{u},{}^{n}\mathbf{u},\Delta\mathbf{t},\Delta\mathbf{t}^*,\mathbf{F})$, where $\mathbf{F}$ is the MG forcing term [74, 58], which is 0 on the fine grid.

4.2. Potentially time-consistent u-V(1,0) sawtooth MG cycle

In this approach, MG is applied to accelerate the subiterative procedure, which aims at solving the nonlinear system of the discretized equations $\mathbf{R}({}^{n+1}\mathbf{u},{}^{n}\mathbf{u},\Delta\mathbf{t})=0$ (Equation (A5)). At each

DOI: 10.1002/fld

subiteration m_{it}, the solution is advanced on the fine grid using operator S (Equation (A3)), and the result is transferred on the coarse grid, where a meanflow subiteration is performed by applying operator S_{MF} (Equation (A12)). The forcing term $\mathbf{F}_{\mathrm{MF}_{2h}}$ is based on the meanflow residual $\mathbf{R}_{\mathrm{MF}}$ (Equation (A14)). The procedure is repeated on $\mathscr{G}_{4h}$ and increments are interpolated in the fine grid $\mathscr{G}_h$, completing subiteration m_{it}. Obviously, this procedure is expected to be efficient only for a subiterative strategy where the number of subiterations is chosen dynamically, based on an increment-convergence-tolerance criterion (r_{TRG}). The building block of this cycle is a MG-subiteration operator $S^{(\mathrm{u})}_{\mathrm{MG}}$, using a V(1,0) sawtooth cycle, defined as

$$
\left\{
\begin{array}{l}
\text{subiteration } (m_{\mathrm{it}}, n_{\mathrm{it}}) = (m+1, n+1) \\
{}^{n}\mathbf{u}_h = {}^{n}\mathbf{u};\ {}^{n}\mathbf{u}_{2h} = \mathsf{T}^{2h}_{\mathbf{u};h}[{}^{n}\mathbf{u}_h];\ {}^{n}\mathbf{u}_{4h} = \mathsf{T}^{4h}_{\mathbf{u};2h}[{}^{n}\mathbf{u}_{2h}] \\
\text{compute on } \mathscr{G}_h: \quad {}^{0}\mathbf{q}_h = {}^{m,n+1}\mathbf{u};\ \mathbf{F}_h = 0;\ \mathbf{q}_h = \mathsf{S}_h({}^{0}\mathbf{q}_h, {}^{n}\mathbf{u}_h, \Delta\mathbf{t}_h, \Delta\mathbf{t}^{*}_h, \mathbf{F}_h) \\
\text{transfer to } \mathscr{G}_{2h}: \quad \left\{
\begin{array}{l}
{}^{0}\mathbf{q}_{2h} = \mathsf{T}^{2h}_{\mathbf{u};h}[\mathbf{q}_h] \\
\mathbf{F}_{\mathrm{MF}_{2h}} = \mathsf{T}^{2h}_{\mathbf{R};h}[\mathbf{R}_{\mathrm{MF}_h}(\mathbf{q}_{\mathrm{MF}_h}, {}^{n}\mathbf{u}_h, \Delta\mathbf{t}_h)] \\
\qquad\qquad -\mathbf{R}_{\mathrm{MF}_{2h}}({}^{0}\mathbf{q}_{\mathrm{MF}_{2h}}, {}^{n}\mathbf{u}_{2h}, \Delta\mathbf{t}_{2h})
\end{array}
\right. \\
\text{compute on } \mathscr{G}_{2h}: \quad \left\{
\begin{array}{l}
\mathbf{q}_{\mathrm{MF}_{2h}} = \mathsf{S}_{\mathrm{MF}_{2h}}({}^{0}\mathbf{q}_{\mathrm{MF}_{2h}}, {}^{n}\mathbf{u}_{2h}, \Delta\mathbf{t}_{2h}, \Delta\mathbf{t}^{*}_{2h}, \mathbf{F}_{2h}) \\
\mathbf{q}_{\mathrm{RSM}_{2h}} = {}^{0}\mathbf{q}_{\mathrm{RSM}_{2h}} = \mathsf{T}^{2h}_{\mathbf{u};h}[\mathbf{q}_{\mathrm{RSM}_h}]
\end{array}
\right. \\
\text{transfer to } \mathscr{G}_{4h}: \quad \left\{
\begin{array}{l}
{}^{0}\mathbf{q}_{4h} = \mathsf{T}^{4h}_{\mathbf{u};2h}[\mathbf{q}_{2h}] \\
\mathbf{F}_{\mathrm{MF}_{4h}} = \mathsf{T}^{4h}_{\mathbf{R};2h}[\mathbf{R}_{\mathrm{MF}_{2h}}(\mathbf{q}_{\mathrm{MF}_{2h}}, {}^{n}\mathbf{u}_{2h}, \Delta\mathbf{t}_{2h}) + \mathbf{F}_{2h}] \\
\qquad\qquad -\mathbf{R}_{\mathrm{MF}_{4h}}({}^{0}\mathbf{q}_{\mathrm{MF}_{4h}}, {}^{n}\mathbf{u}_{4h}, \Delta\mathbf{t}_{4h})
\end{array}
\right. \\
\text{compute on } \mathscr{G}_{4h}: \quad \left\{
\begin{array}{l}
\mathbf{q}_{\mathrm{MF}_{4h}} = \mathsf{S}_{\mathrm{MF}_{4h}}({}^{0}\mathbf{q}_{\mathrm{MF}_{4h}}, {}^{n}\mathbf{u}_{4h}, \Delta\mathbf{t}_{4h}, \Delta\mathbf{t}^{*}_{4h}, \mathbf{F}_{\mathrm{MF}_{4h}}) \\
\mathbf{q}_{\mathrm{RSM}_{4h}} = {}^{0}\mathbf{q}_{\mathrm{RSM}_{4h}} = \mathsf{T}^{4h}_{\mathbf{u};2h}[\mathbf{q}_{\mathrm{RSM}_{2h}}] = \mathsf{T}^{4h}_{\mathbf{u};2h}\mathsf{T}_{\mathbf{u};\mathrm{h}^2h}[\mathbf{q}_{\mathrm{RSM}_h}]
\end{array}
\right. \\
\text{interpolation}: \quad {}^{m+1,n+1}\mathbf{u} = \mathsf{B}_h\{\mathbf{q}_h + \mathsf{I}^{h}_{2h}[\mathbf{q}_{2h} - {}^{0}\mathbf{q}_{2h}] + \mathsf{I}^{h}_{2h}\mathsf{I}^{2h}_{4h}[\mathbf{q}_{4h} - {}^{0}\mathbf{q}_{4h}]\}
\end{array}
\right.
$$

$$
\Longleftrightarrow {}^{m+1,n+1}\mathbf{u} = \mathsf{S}^{(\mathrm{u})}_{\mathrm{MG}}({}^{m,n+1}\mathbf{u}, {}^{n}\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^{*}, L_{\mathrm{GRD}}) \tag{9}
$$

Using the operator $\mathsf{S}^{(\mathrm{u})}_{\mathrm{MG}}$ (Equation (9)), the u–MG algorithm, with a dynamic subiterative strategy, reads

$$
\begin{array}{l}
\text{do } n_{\mathrm{it}} = 1, N_{\mathrm{it}}, 1;\ n = n_{\mathrm{it}} - 1;\ {}^{0,n+1}\mathbf{u} = {}^{n}\mathbf{u} \\
\text{do } m_{\mathrm{it}} \text{ while } [r_{\mathrm{MF}} \geqslant r_{\mathrm{TRG}}];\ m = m_{\mathrm{it}} - 1;\ M_{\mathrm{it}} = m_{\mathrm{it}} \\
{}^{m+1,n+1}\mathbf{u} = \mathsf{S}^{(\mathrm{u})}_{\mathrm{MG}}({}^{m,n+1}\mathbf{u}, {}^{n}\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^{*}, L_{\mathrm{GRD}}) \\
\text{end do};\ {}^{n+1}\mathbf{u} = {}^{M_{\mathrm{it}},n+1}\mathbf{u};\ \text{end do}
\end{array}
\tag{10}
$$

DOI: 10.1002/fld

The advantage of this procedure is that it can be made time-consistent (by using an homogeneous Δt, and eventually an $O(\Delta t^2)$ discretization) and used for unsteady flows [97]. When applied to steady flows, with local-time steps $\Delta\mathbf{t}$, its major drawback is that the rather costly evaluation of $\mathbf{R}_{\mathrm{MF}_h}$ (Equation (A12)) is performed at every subiteration, contrary to the s–MG algorithm where this evaluation is done at the end of the fine grid subiterative procedure (operator N_h). For this reason, the u–MG procedure is not very efficient when a strategy with a small fixed number of subiterations per iteration M_{it} is used. To improve the performance of u–MG, a modified non-time-consistent variant (u′–MG) was developed.

4.3. u′-V(1,0) sawtooth MG cycle

Since this procedure is interested in the convergence acceleration of steady flows only, it is not designed for the acceleration of the subiterative solution of $\mathbf{R}({}^{n+1}\mathbf{u}, {}^{n}\mathbf{u}, \Delta\mathbf{t})=0$ (Equation (A12)), but rather of $\mathbf{L}({}^{n+1}\mathbf{u})=0$ (Equation (A1)), this being the ultimate goal of TM to the steady state. To this purpose, forcing terms $\mathbf{F}_{\mathrm{MF}_{2h}}$, $\mathbf{F}_{\mathrm{MF}_{4h}}$ (Equations (11)) are based, contrary to u–MG (Equations (9)), on $\mathbf{L}_{\mathrm{MF}}$ instead of $\mathbf{R}_{\mathrm{MF}}$, as in the s–MG algorithm (Equations (8)). Furthermore, the initial state for evaluating the residual in $\mathsf{S}_{\mathrm{MF}_{2h}}$ and $\mathsf{S}_{\mathrm{MF}_{4h}}$ is computed using the transferred results from the immediately coarser grid $\mathbf{q}_{2h}$, $\mathbf{q}_{4h}$ (instead of the transferred iteration n variables ${}^{n}\mathbf{u}_{2h}$, ${}^{n}\mathbf{u}_{4h}$ used in u–MG). This choice is equivalent to cancelling the time derivative in the coarse-grid subiteration operator S_{MF} (Equation (A12)). These non-time-consistent modifications are all motivated by the fact that quasi-time-consistency is expected to slow down the iterative procedure. Notice that both u′–MG and u–MG have the same limit as $\Delta t \longrightarrow \infty$, i.e. for a quasi-Newton iterative procedure [17, 18] The building block of this cycle is a MG-subiteration operator $S_{\mathrm{MG}}^{(\mathrm{u}')}$, using a V(1,0) sawtooth cycle, defined as

$$
\left\{\begin{array}{ll}
\text{subiteration } (m_{\mathrm{it}}, n_{\mathrm{it}})=(m+1, n+1) & \\
\text{compute on } \mathscr{G}_h\text{:} & {}^{0}\mathbf{q}_h={}^{m,n+1}\mathbf{u};\ \mathbf{F}_h=0;\ \mathbf{q}_h=\mathsf{S}_h({}^{0}\mathbf{q}_h, {}^{n}\mathbf{u}_h, \Delta\mathbf{t}_h, \Delta\mathbf{t}_h^*, \mathbf{F}_h) \\
\text{transfer to } \mathscr{G}_{2h}\text{:} & \left\{\begin{array}{l} {}^{0}\mathbf{q}_{2h}=\mathsf{T}_{\mathbf{u};h}^{2h}[\mathbf{q}_h] \\ \mathbf{F}_{\mathrm{MF}_{2h}}=\mathsf{T}_{\mathbf{R};h}^{2h}[\mathbf{L}_{\mathrm{MF}_h}(\mathbf{q}_h)]-\mathbf{L}_{\mathrm{MF}_{2h}}({}^{0}\mathbf{q}_{2h}) \end{array}\right. \\
\text{compute on } \mathscr{G}_{2h}\text{:} & \left\{\begin{array}{l} \mathbf{q}_{\mathrm{MF}_{2h}}=\mathsf{S}_{\mathrm{MF}_{2h}}({}^{0}\mathbf{q}_{\mathrm{MF}_{2h}}, {}^{0}\mathbf{q}_{2h}, \Delta\mathbf{t}_{2h}, \Delta\mathbf{t}_{2h}^*, \mathbf{F}_{2h}) \\ \mathbf{q}_{\mathrm{RSM}_{2h}}={}^{0}\mathbf{q}_{\mathrm{RSM}_{2h}}=\mathsf{T}_{\mathbf{u};h}^{2h}[\mathbf{q}_{\mathrm{RSM}_h}] \end{array}\right. \\
\text{transfer to } \mathscr{G}_{4h}\text{:} & \left\{\begin{array}{l} {}^{0}\mathbf{q}_{4h}=\mathsf{T}_{\mathbf{u};2h}^{4h}[\mathbf{q}_{2h}] \\ \mathbf{F}_{\mathrm{MF}_{4h}}=\mathsf{T}_{\mathbf{R};2h}^{4h}[\mathbf{L}_{\mathrm{MF}_{2h}}(\mathbf{q}_{2h})+\mathbf{F}_{2h}]-\mathbf{L}_{\mathrm{MF}_{4h}}({}^{0}\mathbf{q}_{4h}) \end{array}\right. \\
\text{compute on } \mathscr{G}_{4h}\text{:} & \left\{\begin{array}{l} \mathbf{q}_{\mathrm{MF}_{4h}}=\mathsf{S}_{\mathrm{MF}_{4h}}({}^{0}\mathbf{q}_{\mathrm{MF}_{4h}}, {}^{0}\mathbf{q}_{4h}, \Delta\mathbf{t}_{4h}, \Delta\mathbf{t}_{4h}^*, \mathbf{F}_{\mathrm{MF}_{4h}}) \\ \mathbf{q}_{\mathrm{RSM}_{4h}}={}^{0}\mathbf{q}_{\mathrm{RSM}_{4h}}=\mathsf{T}_{\mathbf{u};2h}^{4h}[\mathbf{q}_{\mathrm{RSM}_{2h}}]=\mathsf{T}_{\mathbf{u};2h}^{4h}\mathsf{T}_{\mathbf{u};\mathrm{h}^2h}[\mathbf{q}_{\mathrm{RSM}_h}] \end{array}\right. \\
\text{interpolation:} & {}^{m+1,n+1}\mathbf{u}=\mathsf{B}_h\{\mathbf{q}_h+\mathsf{I}_{2h}^{h}[\mathbf{q}_{2h}-{}^{0}\mathbf{q}_{2h}]+\mathsf{I}_{2h}^{h}\mathsf{I}_{4h}^{2h}[\mathbf{q}_{4h}-{}^{0}\mathbf{q}_{4h}]\}
\end{array}\right.
$$

$$
\Longleftrightarrow {}^{m+1,n+1}\mathbf{u}=\mathsf{S}_{\mathrm{MG}}^{(\mathrm{u}')}({}^{m,n+1}\mathbf{u}, {}^{n}\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^*, L_{\mathrm{GRD}}) \tag{11}
$$

DOI: 10.1002/fld

Using the operator $\mathsf{S}_{\mathrm{MG}}^{(\mathrm{u}')}$ (Equation (11)), the u′–MG algorithm reads

$$\begin{aligned}
&\text{do } n_{\mathrm{it}}=1, N_{\mathrm{it}}, 1; n=n_{\mathrm{it}}-1; {}^{0,n+1}\mathbf{u}={}^{n}\mathbf{u}\\
&\text{do } m_{\mathrm{it}} \text{ while } [r_{\mathrm{MF}}\geqslant r_{\mathrm{TRG}}]; m=m_{\mathrm{it}}-1; M_{\mathrm{it}}=m_{\mathrm{it}}\\
&{}^{m+1,n+1}\mathbf{u}=\mathsf{S}_{\mathrm{MG}}^{(\mathrm{u}')}({}^{m,n+1}\mathbf{u}, {}^{n}\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^{*}, L_{\mathrm{GRD}})\\
&\text{end do}; {}^{n+1}\mathbf{u}={}^{M_{\mathrm{it}},n+1}\mathbf{u}; \text{end do}
\end{aligned} \tag{12}$$

4.4. Application to 2-D compression ramp

This configuration is an $\alpha_{\mathrm{c}}=24$ deg compression ramp in a $M_{\infty}=2.85$ stream [86, 88], and has been used in assessing s–MG with a V(1,0) sawtooth cycle [17]. The OSWTBLI, which takes place at the compression corner induces large separation of the incoming turbulent boundary layer [86, 88]. In a previous work [79, 80], the authors have conducted systematic grid-convergence and boundary-condition studies using various turbulence closures [10, 24, 25, 57]. These studies indicate that the 401×201 computational grid used in the present work (Table IV) gives satisfactory near-grid-converged results, and that the V RSM [82] (whose results, for this particular configuration are very close to the GV RSM [24]) gives the best prediction of wall pressure $\bar{p}_{\mathrm{w}}$, upstream-influence-length, and separation position (Figure 4). Notice that the model predicts a marked pressure plateau between the first and the second oblique shock waves of the λ-structure (Figure 4), while the measurements show a more gradual increase of $\bar{p}_{\mathrm{w}}$. This is indicative of 3-D effects in the measurements [81].

Computations were first run with a static subiterative strategy ($[\mathrm{CFL}, \mathrm{CFL}^{*}; M_{\mathrm{it}}, r_{\mathrm{TRG}}]=[100, 10; 4, -]$), using the V RSM [82], the WNF–LSS RSM [25] and the Launder–Sharma $k-\varepsilon$ [57] (Figure 4). Both SG computations, and computations with three levels of MG (MG3) were run, using the three MG algorithms (Section 4), with a V(1,0) sawtooth cycle. For the s–MG computations a V(2,0) sawtooth-cycle was also used.

Convergence plots of both meanflow (e_{MF}) and turbulence (e_{RSM}) variables indicate similar trends (Figure 4). They both stall, after reduction of the initial error by ~ 4 orders-of-magnitude (Figure 4). This convergence stall occurs at similar levels both for the SG and MG computations (Figure 4), and is attributed to either the slope limiters in the MUSCL reconstruction [98] or to the AF–ADI factorization used in solving the linear system at each subiteration [99]. CPU-speed-ups (Table V) with the fastest s-MG3-V(2,0) strategy, of 7.7, 5.4, and 4.6, are obtained for the $k-\varepsilon$ [57], WNF–LSS RSM [25], and V RSM [82], respectively. CPU-speed-ups are ~ 1.4 times higher with the s-MG3-V(2,0) strategy, compared with the s-MG3-V(1,0) strategy, whatever the turbulence closure used (Table V). The u–MG3–V(1,0) strategy yields poor CPU-speed-ups (Figure 4, Table V). The simple modifications resulting to the u′–MG are particularly efficient, in that the u′–MG3–V(1,0) strategy CPU-speed-ups are quite close to those of the s-MG3–V(1,0) strategy (Figure 4, Table V).

The same computations were also run (Figure 5) with a dynamic subiterative strategy ($[\mathrm{CFL}, \mathrm{CFL}^{*}; M_{\mathrm{it}}, r_{\mathrm{TRG}}]=[\infty, 10; -, -1]$; quasi-Newton). For SG computations, this dynamic strategy is known to be more efficient than a static one [14, Figure 3, p. 771], because, as convergence is approached, the number of subiterations required to reach r_{TRG} is reduced. For MG computations this is observed on every grid of the MG sequence [18, Figure 2, p. 1219]. Furthermore, only a dynamic subiterative strategy can take full advantage of the u–MG technique,

DOI: 10.1002/fld

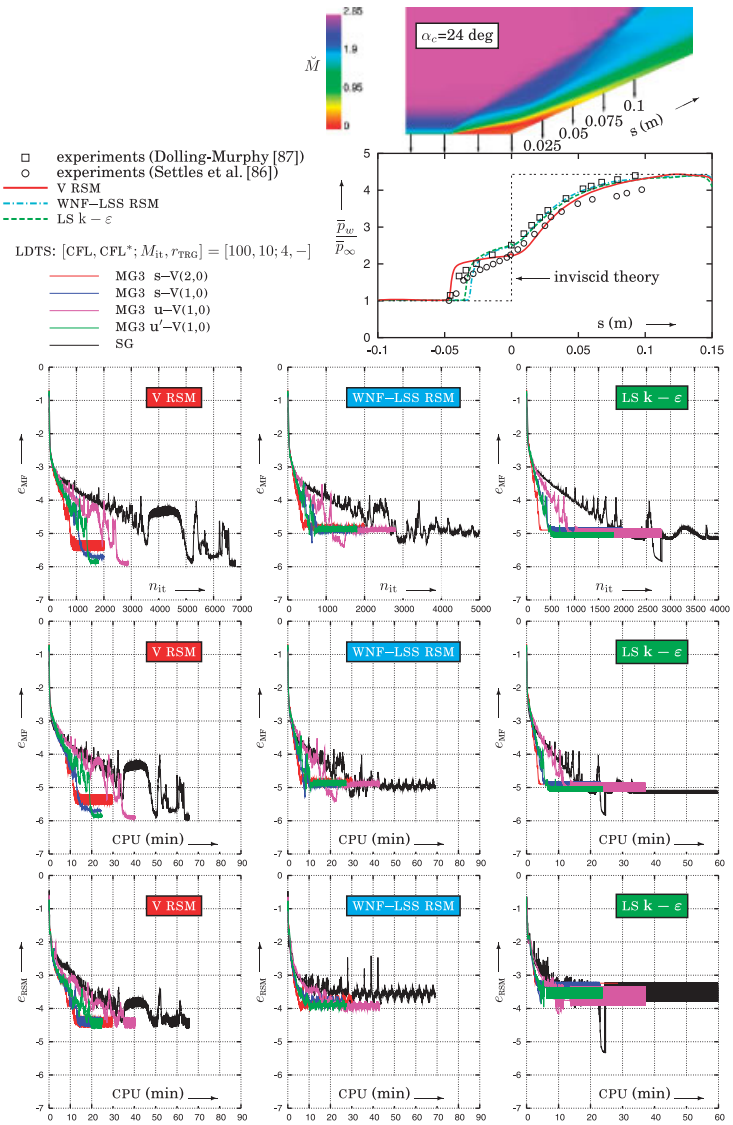

Figure 4. Comparison, for [CFL, CFL*; M_{it}, r_{TRG}] = [100, 10; 4, −], of various multigrid techniques, using 3 levels of multigrid (MG3), with the single-grid algorithm (SG), for the convergence of meanflow error e_{MF} *vs* the number of iterations n_{it} (multigrid-cycles) and *vs* CPU-time (CPU-minutes for 3 Gflop s^{-1} sustained performance), and of the turbulence-variables error e_{RSM} *vs* CPU time, using the V [82] and the WNF–LSS [25] RSMs, and the $k-\varepsilon$ model [57] ($M_\infty = 2.85$; $\alpha_c = 24$ deg; $Re_{\delta_0} = 1.33 \times 10^6$; Settles *et al.* [86, 88] compression ramp; 401×201 grid; Table IV; comparison of computed and measured wall-pressure s-wise distributions and Mach-contours computed with the V RSM).

DOI: 10.1002/fld

which is designed to accelerate subiterative convergence to r_{TRG}, and hence to reduce $M_{\mathrm{it}}(n_{\mathrm{it}})$. As already noted, for the present quasi-Newton computations ($\Delta t \longrightarrow \infty$), u–MG and u′–MG are identical. As far as CPU-speed-ups are concerned (Figure 5, Table V), with this quasi-Newton dynamic subiterative strategy, s-MG3–V(1,0) is only marginally better than u–MG3–V(1,0) or u′–MG3–V(1,0), while s-MG3-V(2,0) is again the fastest technique. We have not tested, in the present work u–MG3-V(2,0) or u′–MG3-V(2,0) strategies.

These computations (Figures 4, 5) indicate that CPU-speed-ups for the $k-\varepsilon$ [57] model, obtained by the s-MG3-V(2,0) technique, are ~ 7, i.e. somewhat higher that those for the RSMs [24, 25], which are $5 \pm \frac{1}{2}$ (Table V), in agreement with the observations of Lien and Leschziner [20]. This may be explained by the fact that, for the $k-\varepsilon$ [57] model, the Reynolds stresses adapt directly to the meanflow gradients through the Boussinesq approximation, but also by the fact that RSMs, especially the V RSM [82], predict much larger separation, in better agreement with measurements [79], and hence a more complex flowfield.

These systematic comparisons suggest that s–MG is the most efficient technique, as far as acceleration of convergence to steady state is concerned. For this reason, 3-D tests (Section 3) were performed for s–MG only. Notice, however, that the interest of the u′–MG technique is that it can be very easily coded as a particular case of the u–MG algorithm, and used as the steady computations option of a u–MG-based unsteady flow solver.

5. CONCLUSIONS

The present study of meanflow–multigrid (MF–MG) strategies for the acceleration of the solution of the 3-D compressible Navier–Stokes equations with near-wall wall normal free (WNF) 7-equation second-moment closures [25] indicates that

1. MF–MG is capable of yielding substantial CPU-speed-ups, even for the most unfavourable conditions, such as low Mach number anisotropy-driven developing turbulent flow in a square duct [83] ($r_{\mathrm{CPUSUP}} > 3$ using s-MG3-V(2,0) on a $\sim 18 \times 10^6$ points grid; Table V), or 3-D SWTBLI ($M_{\mathrm{SW}} \in [1.3, 1.8]$ in a rectangular nozzle [84, 85] ($r_{\mathrm{CPUSUP}} \cong 5\frac{1}{2}$ using s-MG3-V(2,0) on a $\sim 19 \times 10^6$ points grid; Table V).
2. Computations of 2-D OSWTBLI on a $M_{\mathrm{SW}} = 2.85$ compression ramp [86, 88, $\alpha_{\mathrm{c}} = 24\,\mathrm{deg}$], using an alternative, potentially time-consistent, strategy, u–MG (aiming at the MG-acceleration of the subiterative solution of the nonlinear system contained in the full-iteration operator) show that this technique, applied to steady flows, is competitive to the standard s–MG strategy (FAS–MG [29] based on the full-iteration operator), only if the subiterative solution strategy is dynamic, i.e. only if the number of subiterations M_{it} is determined dynamically at each iteration, based on a subiterative convergence tolerance criterion. In that case, u–MG can reduce M_{it} necessary to reach subiterative convergence.
3. To improve upon u–MG applied to steady flows, we have also introduced a modified version, u′–MG, which is not time-consistent, and approaches the speed-ups of s–MG (Table V).
4. In general, for steady flows, the s-MG3-V(2,0) strategy is the fastest, yielding $r_{\mathrm{CPUSUP}} \in [3, 5\frac{1}{2}]$ for the entire range of flows studied.

Although fully coupled MG with stabilization fixes (FCSF–MG) is expected to improve upon the present MF–MG approach, the CPU-speed-ups obtained are quite satisfactory compared with the current state-of-the-art of compressible RANS methods with 2-equation closures (Table III).

DOI: 10.1002/fld

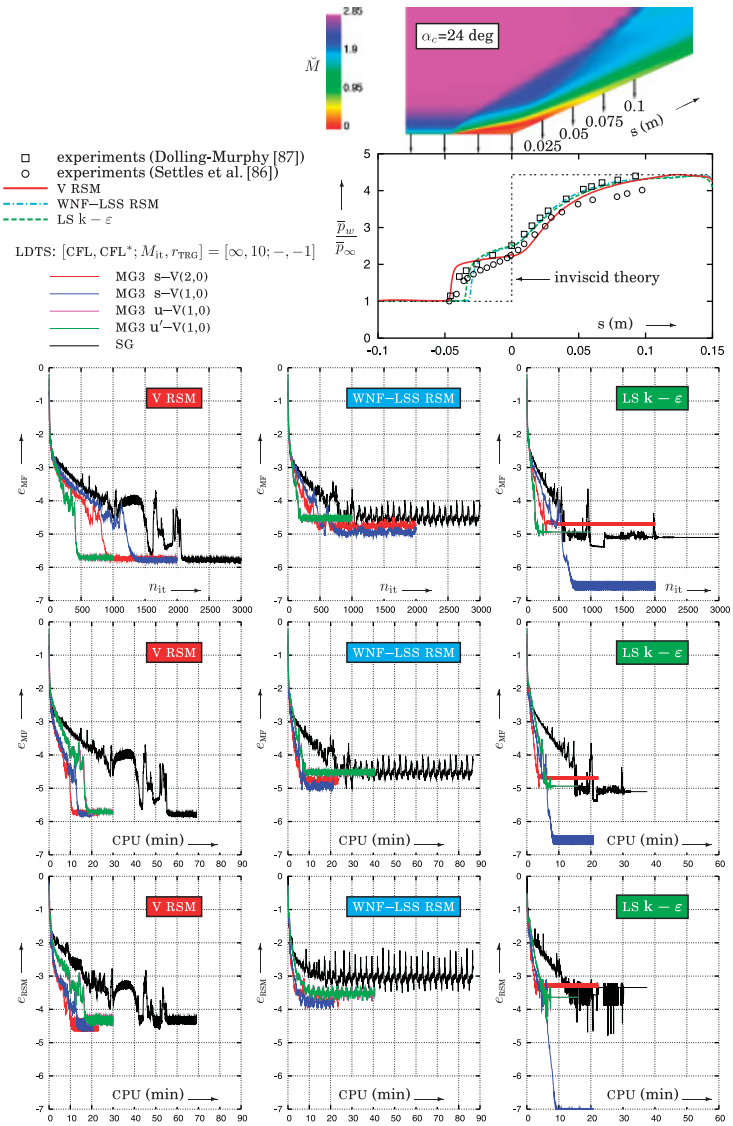

Figure 5. Comparison, for [CFL, CFL*; M_{it}, r_{TRG}] = [∞, 10; −, −1], of various multigrid techniques, using 3 levels of multigrid (MG3), with the single-grid algorithm (SG), for the convergence of meanflow error e_{MF} *vs* the number of iterations n_{it} (multigrid-cycles) and *vs* CPU-time (CPU-minutes for 3 Gflop s^{-1} sustained performance), and of the turbulence-variables error e_{RSM} *vs* CPU time, using the V [82] and the WNF–LSS [25] RSMs, and the $k-\varepsilon$ model [57] ($M_\infty = 2.85$; $\alpha_c = 24$ deg; $Re_{\delta_0} = 1.33 \times 10^6$; Settles *et al.* [86, 88] compression ramp; 401×201 grid; Table IV; comparison of computed and measured wall-pressure s-wise distributions and Mach-contours computed with the V RSM).

DOI: 10.1002/fld

Notice that, to the authors' knowledge, there exist no other published CPU-speed-ups using MG for cRANS with full 7-equation closures. The CPU-speed-ups obtained are quite useful when considering systematic computations of 3-D complex turbulent flows on fine grids, for turbulence-model development and validation studies. In that case, the very slow asymptotic convergence of turbulence quantities, observed in SG computations, can lead to erroneous conclusions, and MG computations are necessary for achieving full convergence, with reasonable CPU times. Furthermore no modification of the MG algorithm is required when different turbulence closures are implemented.

Future research will concentrate on

1. FCSF–MG which is expected to further improve CPU-speed-ups,
2. theoretical studies on characteristic MG to extend the applicability of the present methodology to $L_{\mathrm{GRD}}>3$ levels of MG,
3. to the evaluation of u–MG for the computation of unsteady flows.

APPENDIX A: SINGLE-GRID SCHEME AND OPERATORS

A.1. Single-grid solver

The SG flow solver is described in detail by Chassaing *et al.* [14, 15], and is only briefly summarized in the following. The numerical method presented is designed to be relatively independent of the particular RSM closure used, and is also applicable to 2-equation closures [17]. The equations are discretized on structured multiblock grids using a finite-volume technique, with vertex storage [100]. The divergence of convective fluxes is discretized using the flux-vector-splitting method of Van Leer with $O(\Delta x^3)$ MUSCL (MUSCL3) interpolation [101, 102]. In more recent work [94] hybrid low-diffusion numerical fluxes (e.g. AUSM$^+$, HLLC, Roe, etc.) have been implemented, along with WENO reconstruction of the primitive variables [103]. The MG procedures described in the present work are equally applicable with these more advanced numerical fluxes and reconstruction procedures. The divergence of viscous fluxes is discretized using an $O(\Delta x^2)$ centered stencil [14]. For steady flows, an $O(\Delta t)$ backward-Euler fully implicit scheme is used, which at iteration level n reads [14, 15, 101, 102]

$$\frac{{}^{n+1}\underline{u}_{n_{\mathrm{D}},i,j,k}-{}^{n}\underline{u}_{n_{\mathrm{D}},i,j,k}}{\Delta t_{n_{\mathrm{D}},i,j,k}}+{}^{n+1}\underline{L}_{n_{\mathrm{D}},i,j,k}\cong 0 \quad \forall i,j,k \quad \forall n_{\mathrm{D}}\in[1,N_{\mathrm{D}}]$$

$$\Longleftrightarrow \quad \frac{{}^{n+1}\mathbf{u}-{}^{n}\mathbf{u}}{\Delta \mathbf{t}}+\mathbf{L}({}^{n+1}\mathbf{u})\cong 0 \tag{A1}$$

where n_{D} is the number of the structured block, N_{D} is the number of blocks, $\Delta t_{n_{\mathrm{D}},i,j,k}$ is the time step at grid point (n_{D},i,j,k), $\mathbf{u}=[\underline{u}_1^{\mathrm{T}},\underline{u}_2^{\mathrm{T}},\ldots,\underline{u}_{N_{\mathrm{P}}}^{\mathrm{T}}]^{\mathrm{T}}\in\mathbb{R}^{12N_{\mathrm{P}}}$ is the global vector of the unknowns (N_{P} is the total number of grid points), $\mathbf{L}=[\underline{L}_1^{\mathrm{T}},\underline{L}_2^{\mathrm{T}},\ldots,\underline{L}_{N_{\mathrm{P}}}^{\mathrm{T}}]^{\mathrm{T}}\in\mathbb{R}^{12N_{\mathrm{P}}}$ is the global vector of the space-operators, $\Delta\mathbf{t}=\mathrm{diag}[(\Delta t_1)\underline{\underline{I}}_{12},(\Delta t_2)\underline{\underline{I}}_{12},\ldots,(\Delta t_{N_{\mathrm{P}}})\underline{\underline{I}}_{12}]^{\mathrm{T}}\in\mathbb{R}^{(12N_{\mathrm{P}}\times 12N_{\mathrm{P}})}$ (where $\underline{\underline{I}}_{12}$ denotes the 12×12 identity matrix), and $1/\Delta\mathbf{t}:=\Delta\mathbf{t}^{-1}$.

The nonlinear system obtained from the above discretization (Equation (A1)) for the global vector of unknowns $\mathbf{u}$ is solved using a subiterative approach, based on a local-dual-time-stepping

DOI: 10.1002/fld

LDTS procedure [14], inspired from the corresponding dual-time-stepping (DTS) approach [15] which is widely used for time-consistent unsteady flow computations [104–107]

$$\frac{{}^{m+1,n+1}\mathbf{u}-{}^{m,n+1}\mathbf{u}}{\Delta\mathbf{t}^*}+\frac{{}^{m+1,n+1}\mathbf{u}-{}^{n}\mathbf{u}}{\Delta\mathbf{t}}+\mathbf{L}({}^{m+1,n+1}\mathbf{u})\cong 0 \tag{A2}$$

where $\Delta\mathbf{t}^*=\mathrm{diag}[(\Delta t_1^*)\underline{\underline{I}}_{12},(\Delta t_2^*)\underline{\underline{I}}_{12},\ldots,(\Delta t_{N_\mathrm{P}}^*)\underline{\underline{I}}_{12}]^\mathrm{T}\in\mathbb{R}^{(12N_\mathrm{P}\times 12N_\mathrm{P})}$ is the diagonal matrix of the pseudo-time steps $\Delta t^*_{n_\mathrm{D},i,j,k}$. Defining the subiteration operator S,

$$\begin{aligned}&\mathsf{S}({}^{m,n+1}\mathbf{u},{}^{n}\mathbf{u},\Delta\mathbf{t},\Delta\mathbf{t}^*,\mathbf{F})\\&\quad=\mathsf{RBR}\left\{{}^{m,n+1}\mathbf{u}-\left[\mathbf{I}+\Delta\mathbf{t}^{**}\frac{\partial\mathbf{L}^\mathrm{J}}{\partial\mathbf{u}}({}^{m,n+1}\mathbf{u})\right]^{-1}_{\mathrm{APPRX}}\Delta\mathbf{t}^{**}[\mathbf{R}({}^{m,n+1}\mathbf{u},{}^{n}\mathbf{u},\Delta\mathbf{t})+\mathbf{F}]\right\}\end{aligned} \tag{A3}$$

the corresponding iterative procedure, after linearization, reads [17]

$$\begin{aligned}&\text{do } n_\mathrm{it}=1,N_\mathrm{it},1;\ n=n_\mathrm{it}-1;\ {}^{0,n+1}\mathbf{u}={}^{n}\mathbf{u}\\&\text{do } m_\mathrm{it} \text{ while } [r_\mathrm{MF}\geqslant r_\mathrm{TRG}];\ m=m_\mathrm{it}-1;\ M_\mathrm{it}=m_\mathrm{it}\\&{}^{m+1,n+1}\mathbf{u}=\mathsf{S}({}^{m,n+1}\mathbf{u},{}^{n}\mathbf{u},\Delta\mathbf{t},\Delta\mathbf{t}^*,\mathbf{F})\\&\text{end do};\ {}^{n+1}\mathbf{u}={}^{M_\mathrm{it},n+1}\mathbf{u};\ \text{ end do}\end{aligned} \tag{A4}$$

where

$$\mathbf{R}({}^{m,n+1}\mathbf{u},{}^{n}\mathbf{u},\Delta\mathbf{t})=\left[\frac{{}^{m,n+1}\mathbf{u}-{}^{n}\mathbf{u}}{\Delta\mathbf{t}}+\mathbf{L}({}^{m,n+1}\mathbf{u})\right] \tag{A5}$$

is the residual of the subiterative procedure, $\mathbf{I}\in\mathbb{R}^{(12N_\mathrm{P}\times 12N_\mathrm{P})}$ is the identity matrix, $\Delta\mathbf{t}^{**}:=[\mathbf{I}+\Delta\mathbf{t}^{-1}\Delta\mathbf{t}^*]^{-1}\Delta\mathbf{t}^*$, and $\mathbf{F}\in\mathbb{R}^{12N_\mathrm{P}}$ is a forcing source term, which serves only in the MG iteration, and which is equal to zero for the single-grid scheme ($\mathbf{F}=0$). In the definition of the operator $\mathsf{S}({}^{m,n+1}\mathbf{u},{}^{n}\mathbf{u},\Delta\mathbf{t},\Delta\mathbf{t}^*,\mathbf{F})$, the variables $[{}^{m,n+1}\mathbf{u},{}^{n}\mathbf{u}]$ indicate the states used to compute the residual $\mathbf{R}$ (Equation (A5)).

The Jacobian matrix $\partial\mathbf{L}^\mathrm{J}/\partial\mathbf{u}$ is an approximation to the exact Jacobian $\partial\mathbf{L}/\partial\mathbf{u}$, chosen so as to minimize implicit work for multi-equation turbulence models [14]. The subscript APPRX indicates that the matrix inversion (linear system solution) at each subiteration is approximate. The particular approximations used in the present work [14, 17, 73] are described in Appendix B. To ensure stability at high pseudo-time-steps ($\Delta\mathbf{t}^*$), boundary-conditions are applied implicitly [108], by introducing appropriate modifications of the Jacobians $\partial\mathbf{L}^\mathrm{J}/\partial\mathbf{u}$ to conform with boundary conditions [89]. The operator $\mathsf{B}(\mathbf{u})$ represents (Equation (A3)) the explicit application of boundary-conditions. The operator $\mathsf{R}(\mathbf{u})$ (*cf* Section A.3) represents (Equation (A3)) the explicit application of Reynolds-stress realizability constraints [10], to conform with Reynolds-stress realizability [109]. Both these operators are used in the construction of operator S (Equation (A3)).

The parameters controlling the numerical scheme (time-integration) are the CFL-numbers [14] (CFL for the time-step, and CFL* for the dual pseudo-time-step, assuming that VNN = CFL and VNN* = CFL*) and the number of subiterations performed at each iteration $M_\mathrm{it}(n_\mathrm{it})$. This number can be either fixed, or chosen dynamically, based on an increment-convergence-tolerance criterion.

DOI: 10.1002/fld

The relative variation of the meanflow e_{MF} is given by the following error-L_2-pseudonorm

$$e_{\mathrm{MF}}[\mathbf{u}_{\mathrm{MF}}, \Delta\mathbf{u}_{\mathrm{MF}}] = \log_{10}\sqrt{\frac{1}{5}\left\{\frac{\sum[\Delta\bar{\rho}]^2}{\sum[\bar{\rho}]^2} + \frac{\sum[\Delta(\bar{\rho}\tilde{u}_i)\Delta(\bar{\rho}\tilde{u}_i)]}{\sum[\bar{\rho}\tilde{u}_i\bar{\rho}\tilde{u}_i]} + \frac{\sum[\Delta(\bar{\rho}\breve{h}\mathrm{t}-\bar{p})]^2}{\sum[\bar{\rho}\breve{h}\mathrm{t}-\bar{p}]^2}\right\}} \tag{A6}$$

where $\sum$ implies summation over all grid-nodes, and the summation convention for the Cartesian indices $i, j = 1, 2, 3$ is used. This quantity (e_{MF}) defines approximately the number of digits to which the computation of the meanflow variables is converged. It is used to define the subiterative convergence of the increment by the error reduction between subiterations $[m, n+1]$ and $[m+1, n+1]$ (Equations (A3), (A4))

$$r_{\mathrm{MF}}(m+1, n+1) = \log_{10}\left\{\frac{10^{[e_{\mathrm{MF}}(m+1,n+1)]} - 10^{[e_{\mathrm{MF}}(m,n+1)]}}{10^{[e_{\mathrm{MF}}(m,n+1)]}}\right\} \tag{A7}$$

$$e_{\mathrm{MF}}(m+1, n+1) \equiv e_{\mathrm{MF}}[{}^{n}\mathbf{u}_{\mathrm{MF}}, {}^{m+1,n+1}\mathbf{u}_{\mathrm{MF}} - {}^{n}\mathbf{u}_{\mathrm{MF}}] \tag{A8}$$

The reduction (r_{MF}) indicates approximately the number of digits to which the increment is converged during the subiterations. The time-integration scheme is therefore defined by the triplet $[\mathrm{CFL}, \mathrm{CFL}^*; M_{\mathrm{it}}, r_{\mathrm{TRG}}]$ where either M_{it} or r_{TRG} is specified.

A similar error-pseudonorm [14] is used to monitor the convergence of the turbulence variables

$$e_{\mathrm{RSM}}[\mathbf{u}_{\mathrm{RSM}}, \Delta\mathbf{u}_{\mathrm{RSM}}] = \log_{10}\sqrt{\tfrac{1}{7}\left\{\frac{\sum[\Delta(\overline{\rho u_i'' u_j''})\Delta(\overline{\rho u_i'' u_j''})]}{\sum[(\overline{\rho u_i'' u_j''})(\overline{\rho u_i'' u_j''})]} + \frac{\sum[\Delta\bar{\rho}\varepsilon^*]^2}{\sum[\bar{\rho}\varepsilon^*]^2}\right\}} \tag{A9}$$

A.2. Operators

Using the subiteration operator S (Equation (A3)), it is straightforward to write symbolically the operator N which advances the solution for K_{it} full iterations. This operator can be defined either with a fixed number of subiterations M_{it} ($\mathsf{N}[\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^*, \mathbf{F}; M_{\mathrm{it}}, K_{\mathrm{it}}]$),

$$\left\{\begin{array}{l} \text{do } k = 1, K_{\mathrm{it}}, 1;\ {}^{0,n+k}\mathbf{u} = {}^{n+k-1}\mathbf{u} \\ \text{do } m_{\mathrm{it}} = 1, M_{\mathrm{it}}, 1;\ m = m_{\mathrm{it}} - 1 \\ {}^{m+1,n+k}\mathbf{u} = \mathsf{S}({}^{m,n+k}\mathbf{u}, {}^{n+k-1}\mathbf{u}, \Delta\mathbf{t}, \Delta\mathbf{t}^*, \mathbf{F}) \\ \text{end do};\ {}^{n+k}\mathbf{u} = {}^{M_{\mathrm{it}},n+k}\mathbf{u};\ \text{end do} \end{array}\right\}$$

$$\Longleftrightarrow {}^{n+K_{\mathrm{it}}}\mathbf{u} = \mathsf{N}({}^{n}\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^*, \mathbf{F}; M_{\mathrm{it}}, K_{\mathrm{it}}) \tag{A10}$$

or with the number of subiterations determined dynamically to obtain a given level of error-reduction of the increment r_{MF} [14] ($\mathsf{N}[\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^*, \mathbf{F}; r_{\mathrm{TRG}}, K_{\mathrm{it}}]$

$$\left\{\begin{array}{l} \text{do } k = 1, K_{\mathrm{it}}, 1;\ {}^{0,n+k}\mathbf{u} = {}^{n+k-1}\mathbf{u} \\ \text{do } m_{\mathrm{it}} \text{ while } [r_{\mathrm{MF}} \geqslant r_{\mathrm{TRG}}];\ m = m_{\mathrm{it}} - 1;\ M_{\mathrm{it}} = m_{\mathrm{it}} \\ {}^{m+1,n+k}\mathbf{u} = \mathsf{S}({}^{m,n+k}\mathbf{u}, {}^{n+k-1}\mathbf{u}, \Delta\mathbf{t}, \Delta\mathbf{t}^*, \mathbf{F}) \\ \text{end do};\ {}^{n+k}\mathbf{u} = {}^{M_{\mathrm{it}},n+k}\mathbf{u};\ \text{end do} \end{array}\right\}$$

$$\Longleftrightarrow {}^{n+K_{\mathrm{it}}}\mathbf{u} = \mathsf{N}({}^{n}\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^*, \mathbf{F}; r_{\mathrm{TRG}}, K_{\mathrm{it}}) \tag{A11}$$

In both cases (Equations (A10), (A11)) the subiterations are initialized by ${}^{0,n+1}\mathbf{u} = {}^{n}\mathbf{u}$.

DOI: 10.1002/fld

The MG algorithm is applied to the meanflow variables $\mathbf{u}_{\mathrm{MF}}$ only. To this purpose, it is useful to define a meanflow subiteration operator $\mathsf{S}_{\mathrm{MF}}({}^{m,n+1}\mathbf{u}_{\mathrm{MF}}, {}^{n}\mathbf{u}, \Delta\mathbf{t}, \Delta\mathbf{t}^{*}, \mathbf{F}_{\mathrm{MF}})$, analogous to the subiteration operator $\mathsf{S}({}^{m,n+1}\mathbf{u}, {}^{n}\mathbf{u}, \Delta\mathbf{t}, \Delta\mathbf{t}^{*}, \mathbf{F})$ (Equation (A3))

$$\begin{aligned}&\mathsf{S}_{\mathrm{MF}}({}^{m,n+1}\mathbf{u}_{\mathrm{MF}}, {}^{n}\mathbf{u}, \Delta\mathbf{t}, \Delta\mathbf{t}^{*}, \mathbf{F}_{\mathrm{MF}})\\ &\quad=\mathsf{B}_{\mathrm{MF}}\left\{{}^{m,n+1}\mathbf{u}_{\mathrm{MF}}-\left[\mathbf{I}+\Delta\mathbf{t}^{**}\frac{\partial\mathbf{L}^{\mathrm{J}}_{\mathrm{MF}}}{\partial\mathbf{u}_{\mathrm{MF}}}({}^{m,n+1}\mathbf{u})\right]^{-1}_{\mathrm{APPRX}}\Delta\mathbf{t}^{**}[\mathbf{R}_{\mathrm{MF}}({}^{m,n+1}\mathbf{u}_{\mathrm{MF}}, {}^{n}\mathbf{u})+\mathbf{F}_{\mathrm{MF}}]\right\}\end{aligned} \tag{A12}$$

where $\mathbf{L}_{\mathrm{MF}}$ and $\mathbf{L}^{\mathrm{J}}_{\mathrm{MF}}$ are the meanflow-equations space-operator and the approximate space-operator used for the Jacobians, and

$${}^{m+1,n+1}\mathbf{u}_{\mathrm{RSM}}={}^{m,n+1}\mathbf{u}_{\mathrm{RSM}}={}^{n}\mathbf{u}_{\mathrm{RSM}} \tag{A13}$$

$$\mathbf{R}_{\mathrm{MF}}({}^{m,n+1}\mathbf{u}_{\mathrm{MF}}, {}^{n}\mathbf{u}, \Delta\mathbf{t})=\left[\frac{{}^{m,n+1}\mathbf{u}_{\mathrm{MF}}-{}^{n}\mathbf{u}_{\mathrm{MF}}}{\Delta\mathbf{t}}+\mathbf{L}_{\mathrm{MF}}({}^{m,n+1}\mathbf{u})\right] \tag{A14}$$

Full-iteration meanflow-operators $\mathsf{N}_{\mathrm{MF}}\left(\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^{*}, \mathbf{F}_{\mathrm{MF}}; r_{\mathrm{TRG}}, K_{\mathrm{it}}\right)$ and $\mathsf{N}_{\mathrm{MF}}(\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^{*}, \mathbf{F}_{\mathrm{MF}}; M_{\mathrm{it}}, K_{\mathrm{it}})$ are then defined, in the same way as N (Equations (A10), (A11))

$$\left\{\begin{array}{l}\text{do } k=1, K_{\mathrm{it}}, 1;\ {}^{0,n+k}\mathbf{u}={}^{n+k-1}\mathbf{u}\\ \text{do } m_{\mathrm{it}}=1, M_{\mathrm{it}}, 1;\ m=m_{\mathrm{it}}-1\\ {}^{m+1,n+k}\mathbf{u}_{\mathrm{MF}}=\mathsf{S}_{\mathrm{MF}}({}^{m,n+k}\mathbf{u}_{\mathrm{MF}}, {}^{n+k-1}\mathbf{u}, \Delta\mathbf{t}, \Delta\mathbf{t}^{*}, \mathbf{F}_{\mathrm{MF}})\\ {}^{m+1,n+k}\mathbf{u}_{\mathrm{RSM}}={}^{n}\mathbf{u}_{\mathrm{RSM}}\\ \text{end do};\ {}^{n+k}\mathbf{u}={}^{M_{\mathrm{it}},n+k}\mathbf{u};\ \text{end do}\end{array}\right\}$$

$$\Longleftrightarrow {}^{n+K_{\mathrm{it}}}\mathbf{u}_{\mathrm{MF}}=\mathsf{N}_{\mathrm{MF}}({}^{n}\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^{*}, \mathbf{F}_{\mathrm{MF}}; M_{\mathrm{it}}, K_{\mathrm{it}}) \tag{A15}$$

$$\left\{\begin{array}{l}\text{do } k=1, K_{\mathrm{it}}, 1;\ {}^{1,n+k}\mathbf{u}={}^{n+k-1}\mathbf{u}\\ \text{do } m_{\mathrm{it}} \text{ while } [r_{\mathrm{MF}}\geqslant r_{\mathrm{TRG}}];\ m=m_{\mathrm{it}}-1;\ M_{\mathrm{it}}=m_{\mathrm{it}}\\ {}^{m+1,n+k}\mathbf{u}_{\mathrm{MF}}=\mathsf{S}_{\mathrm{MF}}({}^{m,n+k}\mathbf{u}_{\mathrm{MF}}, {}^{n+k-1}\mathbf{u}, \Delta\mathbf{t}, \Delta\mathbf{t}^{*}, \mathbf{F}_{\mathrm{MF}})\\ {}^{m+1,n+k}\mathbf{u}_{\mathrm{RSM}}={}^{n}\mathbf{u}_{\mathrm{RSM}}\\ \text{end do};\ {}^{n+k}\mathbf{u}={}^{M_{\mathrm{it}},n+k}\mathbf{u};\ \text{end do}\end{array}\right\}$$

$$\Longleftrightarrow {}^{n+K_{\mathrm{it}}}\mathbf{u}_{\mathrm{MF}}=\mathsf{N}_{\mathrm{MF}}({}^{n}\mathbf{u}, \mathrm{CFL}, \mathrm{CFL}^{*}, \mathbf{F}_{\mathrm{MF}}; r_{\mathrm{TRG}}, K_{\mathrm{it}}) \tag{A16}$$

A.3. Realizability constraints and heuristic stabilization

It is quite possible, during the iterations, to obtain Reynolds stresses which do not satisfy the realizability constraints introduced by Schumann [109]. Such anomalous behaviour is systematically

DOI: 10.1002/fld

checked for at every subiteration. If the realizability constraints are not satisfied for a given grid point n_P, then all turbulence variables are set to 0 at this grid point (this is consistent with the wall-boundary-conditions)

$$\text{if } \left\{ \begin{array}{llll} (\widetilde{u''^2})_{n_P}<0 & \vee & [(\widetilde{u''v''})_{n_P}]^2-(\widetilde{u''^2})_{n_P}(\widetilde{v''^2})_{n_P}>0 & \vee \\ (\widetilde{v''^2})_{n_P}<0 & \vee & [(\widetilde{v''w''})_{n_P}]^2-(\widetilde{v''^2})_{n_P}(\widetilde{w''^2})_{n_P}>0 & \vee \\ (\widetilde{w''^2})_{n_P}<0 & \vee & [(\widetilde{w''u''})_{n_P}]^2-(\widetilde{w''^2})_{n_P}(\widetilde{u''^2})_{n_P}>0 & \vee \\ & & (\det[\widetilde{u''_i u''_j}])_{n_P}<0 & \vee \\ \varepsilon^*_{n_P}<0 & \vee & (\ell^*_T)_{n_P}=k_{n_P}^{3/2}\varepsilon_{n_P}^{*\,-1}>\ell_{T_{max}} & \end{array} \right\} : (\underline{u}_{RSM})_{n_P} \longleftarrow \underline{0} \qquad \text{(A17)}$$

where $\ell_{T_{max}}$ is a maximum admissible length-scale (a characteristic order-of-magnitude length of the configuration). Divisions by 0 are avoided throughout the code by adding 10^{-23} to the denominator (for every fraction $b_1/b_2 \cong b_1/(b_2+10^{-23})$). These simple realizability and boundedness fixes (which are completely explicit and as a consequence easy to implement) were developed in Vallet [89], and were found to stabilize the computations for all the cases studied using the single-grid method [14]. In subsequent subiterations turbulence builds up again through diffusion from neighbouring nodes. These explicit realizability constraints will be represented by the operator $\mathsf{R}(\mathbf{u})$ used in constructing the subiteration operator S (Equation (A3)).

APPENDIX B: APPROXIMATE JACOBIANS AND APPROXIMATE FACTORIZATION

A structured multiblock implementation is followed in the present work. Let

$$\mathbf{u}=[\mathbf{u}_1^T,\mathbf{u}_2^T,\ldots,\mathbf{u}_{n_D}^T,\ldots,\mathbf{u}_{N_D}^T]^T \qquad \text{(B1a)}$$

$$\mathbf{L}=[\mathbf{L}_1^T,\mathbf{L}_2^T,\ldots,\mathbf{L}_{n_D}^T,\ldots,\mathbf{L}_{N_D}^T]^T \qquad \text{(B1b)}$$

$$\mathbf{R}=[\mathbf{R}_1^T,\mathbf{R}_2^T,\ldots,\mathbf{R}_{n_D}^T,\ldots,\mathbf{R}_{N_D}^T]^T \qquad \text{(B1c)}$$

be partitions of the vectors of variables $\mathbf{u}\in\mathbb{R}^{(12N_P)}$, of space-operators $\mathbf{L}\in\mathbb{R}^{(12N_P)}$ and of the vector of residuals $\mathbf{R}\in\mathbb{R}^{(12N_P)}$, between structured subdomains $n_D\in[1,N_D]$. The linear system, to be solved at each subiteration (Equation (A3)), reads

$$\left[\mathbf{I}+\Delta\mathbf{t}^{**}\frac{\partial\mathbf{L}^J_{n_D}}{\partial\mathbf{u}_{n_D}}({}^{m,n+1}\mathbf{u}_{n_D})\right][{}^{m+1,n+1}\mathbf{u}_{n_D}-{}^{m,n+1}\mathbf{u}_{n_D}]=-\Delta\mathbf{t}^{**}_{n_D}[{}^{m,n+1}\mathbf{R}_{n_D}] \quad \forall n_D\in[1,N_D] \qquad \text{(B2)}$$

Communication between subdomains uses N_{PH} phantom nodes ($N_{PH}=5$ for MUSCL3 or WENO3 reconstructions), with appropriate Dirichlet and/or von Neumann implicit boundary-conditions.

For the purpose of multi-equation turbulence model development, it is important to design the approximate Jacobians in such a way that the increase of computational cost associated with the increase of the number of turbulence variables (e.g. transport equations for the turbulent-heat-fluxes $\overline{\rho u_i'' h''}$, for the temperature-variance $\widetilde{T''^2}$, or for the density-fluctuations $\overline{\rho'^2},\ldots$) be minimized. Following Vallet [89], Chassaing *et al.* [14] achieved this by constructing an approximate space-discretization operator $\mathbf{L}^J$, whose derivative $\partial\mathbf{L}^J/\partial\mathbf{u}$ is the Jacobian matrix used in constructing

DOI: 10.1002/fld

the subiteration operator S (Equation (A3)). This approximate space-operator $\mathbf{L}^{\mathrm{J}}$ uses approximate fluxes $\underline{F}^{\mathrm{J}}=\underline{F}^{\mathrm{CJ}}+\underline{F}^{\mathrm{VJ}}$ and approximate source terms $\underline{S}^{\mathrm{J}}$, in lieu of the numerical fluxes $\underline{F}^{\mathrm{N}}$ [14, Equation (7), p. 765] and source terms $\underline{S}$ [14, Equation (8), p. 765]. The following approximations were made:

1. source terms-Jacobians were neglected ($\underline{S}^{\mathrm{J}}=0$),
2. an $O(\Delta x)$ MUSCL extrapolation [27, 101] was used for computing the approximate convective fluxes $\underline{F}^{\mathrm{CJ}}$,
3. the approximate viscous fluxes $\underline{F}^{\mathrm{VJ}}$ are designed so that the Jacobians of the viscous fluxes are diagonalized.

With these approximations, which are described in detail in Chassaing *et al.* [14], the linear system (Equation (B2)), after a reorganization of the global vectors of variables (Equations (B1a)) in a meanflow-vector $\mathbf{u}_{\mathrm{MF}}$, and in a separate global vector for each turbulence variable $\mathbf{u}_{uu}, \mathbf{u}_{uv}, \mathbf{u}_{vv}, \mathbf{u}_{vw}, \mathbf{u}_{ww}, \mathbf{u}_{wu}, \mathbf{u}_{\varepsilon}$, and similar reorganizations for $\mathbf{L}$ and $\mathbf{R}$, takes the form

$$\begin{bmatrix} \mathbf{I}_{5N_{\mathrm{P}}}+\mathbf{A}_{\mathrm{MF}} & \mathbf{0} \\ & \begin{bmatrix} \mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}} & & & & & & \\ & \mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}} & & & & & \\ & & \mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}} & & & & \\ & & & \mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}} & & & \\ & & & & \mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}} & & \\ & & & & & \mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}} & \\ & & & & & & \mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}} \end{bmatrix} \\ \mathbf{B}_{\mathrm{MF}} & \end{bmatrix} \begin{bmatrix} {}^{m+1,n+1}\Delta\mathbf{u}_{\mathrm{MF}} \\ {}^{m+1,n+1}\Delta\mathbf{u}_{uu} \\ {}^{m+1,n+1}\Delta\mathbf{u}_{uv} \\ {}^{m+1,n+1}\Delta\mathbf{u}_{vv} \\ {}^{m+1,n+1}\Delta\mathbf{u}_{vw} \\ {}^{m+1,n+1}\Delta\mathbf{u}_{ww} \\ {}^{m+1,n+1}\Delta\mathbf{u}_{wu} \\ {}^{m+1,n+1}\Delta\mathbf{u}_{\varepsilon} \end{bmatrix} = -\Delta\mathbf{t}^{**} \begin{bmatrix} {}^{m,n+1}\mathbf{R}_{\mathrm{MF}} \\ {}^{m,n+1}\mathbf{R}_{uu} \\ {}^{m,n+1}\mathbf{R}_{uv} \\ {}^{m,n+1}\mathbf{R}_{vv} \\ {}^{m,n+1}\mathbf{R}_{vw} \\ {}^{m,n+1}\mathbf{R}_{ww} \\ {}^{m,n+1}\mathbf{R}_{wu} \\ {}^{m,n+1}\mathbf{R}_{\varepsilon} \end{bmatrix} \tag{B3}$$

where $\mathbf{u}_{\mathrm{MF}}=[(\underline{u}^{\mathrm{T}}_{\mathrm{MF}})_1, (\underline{u}^{\mathrm{T}}_{\mathrm{MF}})_2, \ldots, (\underline{u}^{\mathrm{T}}_{\mathrm{MF}})_{N_{\mathrm{P}}}]^{\mathrm{T}} \in \mathbb{R}^{5N_{\mathrm{P}}}$ is the global vector of the meanflow variables, $\mathbf{u}_{uu}=[\overline{(\rho u'' u'')}_1, \overline{(\rho u'' u'')}_2, \ldots, \overline{(\rho u'' u'')}_{N_{\mathrm{P}}}]^{\mathrm{T}} \in \mathbb{R}^{N_{\mathrm{P}}}$ is the global vector of the uu Reynolds-stress component (similar definitions apply to $\mathbf{u}_{uv}, \ldots, \mathbf{u}_{\varepsilon}$, and to $\mathbf{R}_{\mathrm{MF}}, \ldots, \mathbf{R}_{\varepsilon}$), and ${}^{m+1,n+1}\Delta\mathbf{u} := {}^{m+1,n+1}\mathbf{u} - {}^{m,n+1}\mathbf{u}$. Obviously, in the above system (Equations (B3)) the meanflow-variables part is decoupled from the RSM part. Furthermore the RSM part consists of 7 linear systems, with the same matrix $[\mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}}]$, so that the system (Equations (B3)) simplifies to

$$\begin{aligned}
[\mathbf{I}_{5N_{\mathrm{P}}}+\mathbf{A}_{\mathrm{MF}}][{}^{m+1,n+1}\Delta\mathbf{u}_{\mathrm{MF}}] &= -\Delta\mathbf{t}^{**}[{}^{m,n+1}\mathbf{R}_{\mathrm{MF}}] \\
[\mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}}][{}^{m+1,n+1}\Delta\mathbf{u}_{uu}] &= -\Delta\mathbf{t}^{**}[{}^{m,n+1}\mathbf{R}_{uu}] - [\mathbf{B}_{\mathrm{MF}}]_{uu}[{}^{m+1,n+1}\Delta\mathbf{u}_{\mathrm{MF}}] \\
[\mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}}][{}^{m+1,n+1}\Delta\mathbf{u}_{uv}] &= -\Delta\mathbf{t}^{**}[{}^{m,n+1}\mathbf{R}_{uv}] - [\mathbf{B}_{\mathrm{MF}}]_{uv}[{}^{m+1,n+1}\Delta\mathbf{u}_{\mathrm{MF}}] \\
[\mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}}][{}^{m+1,n+1}\Delta\mathbf{u}_{vv}] &= -\Delta\mathbf{t}^{**}[{}^{m,n+1}\mathbf{R}_{vv}] - [\mathbf{B}_{\mathrm{MF}}]_{vv}[{}^{m+1,n+1}\Delta\mathbf{u}_{\mathrm{MF}}] \\
[\mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}}][{}^{m+1,n+1}\Delta\mathbf{u}_{vw}] &= -\Delta\mathbf{t}^{**}[{}^{m,n+1}\mathbf{R}_{vw}] - [\mathbf{B}_{\mathrm{MF}}]_{vw}[{}^{m+1,n+1}\Delta\mathbf{u}_{\mathrm{MF}}] \\
[\mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}}][{}^{m+1,n+1}\Delta\mathbf{u}_{ww}] &= -\Delta\mathbf{t}^{**}[{}^{m,n+1}\mathbf{R}_{ww}] - [\mathbf{B}_{\mathrm{MF}}]_{ww}[{}^{m+1,n+1}\Delta\mathbf{u}_{\mathrm{MF}}] \\
[\mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}}][{}^{m+1,n+1}\Delta\mathbf{u}_{wu}] &= -\Delta\mathbf{t}^{**}[{}^{m,n+1}\mathbf{R}_{wu}] - [\mathbf{B}_{\mathrm{MF}}]_{wu}[{}^{m+1,n+1}\Delta\mathbf{u}_{\mathrm{MF}}] \\
[\mathbf{I}_{N_{\mathrm{P}}}+\mathbf{a}_{\mathrm{RSM}}][{}^{m+1,n+1}\Delta\mathbf{u}_{\varepsilon}] &= -\Delta\mathbf{t}^{**}[{}^{m,n+1}\mathbf{R}_{\varepsilon}] - [\mathbf{B}_{\mathrm{MF}}]_{\varepsilon}[{}^{m+1,n+1}\Delta\mathbf{u}_{\mathrm{MF}}]
\end{aligned} \tag{B4}$$

DOI: 10.1002/fld

These systems are inverted only approximately, for each subdomain $n_D \in [1, N_D]$, by approximately factoring the system matrices to apply alternative-directions-implicit sweeps [73, 101, 102]

$$\left[\mathbf{I}+\Delta\mathbf{t}^{**}\frac{\partial\mathbf{L}^{\mathrm{J}}_{n_\mathrm{D}}}{\partial\mathbf{u}_{n_\mathrm{D}}}\right]^{-1}_{\mathrm{AF-ADI}} = \left[\mathbf{I}+\Delta\mathbf{t}^{**}\left(\frac{\partial\mathbf{L}^{\mathrm{J}}_{n_\mathrm{D}}}{\partial\mathbf{u}_{n_\mathrm{D}}}\right)_{\xi}\right]^{-1} \left[\mathbf{I}+\Delta\mathbf{t}^{**}\left(\frac{\partial\mathbf{L}^{\mathrm{J}}_{n_\mathrm{D}}}{\partial\mathbf{u}_{n_\mathrm{D}}}\right)_{\eta}\right]^{-1} \left[\mathbf{I}+\Delta\mathbf{t}^{**}\left(\frac{\partial\mathbf{L}^{\mathrm{J}}_{n_\mathrm{D}}}{\partial\mathbf{u}_{n_\mathrm{D}}}\right)_{\zeta}\right]^{-1} \quad \forall n_\mathrm{D} \in [1, N_\mathrm{D}] \tag{B5}$$

The previously presented simplifications (Equations (B4)) are applied for each sweep. The solution of each linear system (ξ-, η-, ζ-wise) uses band-LU decomposition [110], which is performed only once for the 7 systems corresponding to the turbulence variables (Equation (B4)).

ACKNOWLEDGEMENTS

The computations presented were run at the Institut pour le Développement des Ressources en Informatique Scientifique (IDRIS), where computer resources were made available by the Comité Scientifique. The authors are listed alphabetically.

REFERENCES

1. Vandromme D, Ha Minh H. About the coupling of turbulence closure models with averaged Navier–Stokes equations. *Journal of Computational Physics* 1986; **65**:386–409.
2. Morrison JH. A compressible Navier–Stokes solver with 2-equation and Reynolds-stress turbulence closure models. *Contr. Report CR-4440*, NASA 1992 (also AIAA Paper 1990-5251, 1990).
3. Lien FS, Leschziner MA. A pressure–velocity solution strategy for compressible flow and its application to shock/boundary-layer interaction using second-moment closure. *ASME Journal of Fluids Engineering* 1993; **115**:717–725.
4. Chuang CC, Chieng CC. Comparison of varients of the bi-conjugate gradient method for compressible Navier–Stokes with 2-moment closure. *International Journal for Numerical Methods in Fluids* 1995; **20**: 233–253.
5. Davidson L. Reynolds-stress transport modelling of shock-induced separated flow. *Computers and Fluids* 1995; **24**:253–268.
6. Leschziner MA. Computation of aerodynamic flows with turbulence-transport models based on 2-moment closure. *Computers and Fluids* 1995; **24**:377–392.
7. Ladeinde F. Supersonic flux-split procedure for second moments of turbulence. *AIAA Journal* 1995; **33**: 1185–1195.
8. Ladeinde F, Intile JC. Calculation of Reynolds-stresses in turbulent supersonic flows. *International Journal for Numerical Methods in Fluids* 1995; **21**:49–74.
9. Zha GC, Knight DD. 3-D shock/boundary-layer interaction using Reynolds stress equation turbulence model. *AIAA Journal* 1996; **34**:1313–1320.
10. Gerolymos GA, Vallet I. Near-wall Reynolds-stress 3-D transonic flows computation. *AIAA Journal* 1997; **35**:228–236.
11. Chenault CF, Beran PS. $k-\varepsilon$ and Reynolds stress turbulence model comparisons for 2-D injection flows. *AIAA Journal* 1998; **36**:1401–1412.

DOI: 10.1002/fld

12. Chenault CF, Beran PS, Bowersox RDW. Numerical investigation of supersonic injection using a Reynolds-stress turbulence model. *AIAA Journal* 1999; **37**:1257–1269.
13. Batten P, Craft TJ, Leschziner MA, Loyau H. Reynolds-stress-transport modeling for compressible aerodynamics applications. *AIAA Journal* 1999; **37**:785–797.
14. Chassaing JC, Gerolymos GA, Vallet I. Efficient and robust Reynolds-stress model computation of 3-D compressible flows. *AIAA Journal* 2003; **41**:763–773.
15. Chassaing JC, Gerolymos GA, Vallet I. Reynolds-stress model dual-time-stepping computation of unsteady 3-D flows. *AIAA Journal* 2003; **41**:1882–1894.
16. Alpman E, Long LN. Separated turbulent flow simulations using Reynolds-stress model and unstructured meshes. *CD-Rom Proceedings of the 43rd AIAA Aerospace Sciences Meeting and Exhibit*, 10–13 January 2005, Reno, NV, U.S.A. AIAA: New York, 2005. AIAA Paper 2005-1094.
17. Gerolymos GA, Vallet I. Mean-flow-multigrid for implicit Reynolds-stress-model computations. *AIAA Journal* 2005; **43**:1887–1898.
18. Gerolymos GA, Vallet I. Robust implicit multigrid Reynolds-stress-model computation of 3-D turbomachinery flows. *ASME Journal of Fluids Engineering* 2007; **129**:1212–1227.
19. Demuren AO. Multigrid acceleration and turbulence models for computations of 3-D turbulence jets in crossflow. *International Journal of Heat and Mass Transfer* 1992; **35**:2783–2794.
20. Lien FS, Leschziner MA. Multigrid acceleration for recirculating laminar and turbulent flows computed with a non-orthogonal collocated finite-volume scheme. *Computer Methods in Applied Mechanics and Engineering* 1994; **118**:351–371.
21. Bigarella EDV, Azevedo JLF. Advanced eddy-viscosity and Reynolds-stress turbulence model simulations of aerospace applications. *AIAA Journal* 2007; **45**:2369–2390.
22. Launder BE, Reece GJ, Rodi W. Progress in the development of a Reynolds-stress turbulence closure. *Journal of Fluid Mechanics* 1975; **68**:537–566.
23. Gibson MM, Launder BE. Ground effects on pressure fluctuations in the atmospheric boundary-layer. *Journal of Fluid Mechanics* 1978; **86**:491–511.
24. Gerolymos GA, Vallet I. Wall-normal-free near-wall Reynolds-stress closure for 3-D compressible separated flows. *AIAA Journal* 2001; **39**:1833–1842.
25. Gerolymos GA, Sauret E, Vallet I. Contribution to the single-point-closure Reynolds-stress modelling of inhomogeneous flow. *Theoretical and Applied Fluid Dynamics* 2004; **17**:407–431.
26. Sauret E, Vallet I. Near-wall turbulent pressure diffusion modelling and influence in 3-D secondary flows. *ASME Journal of Fluids Engineering* 2007; **129**:634–642.
27. Hirsch C. *Numerical Computation of Internal and External Flows* (1st edn), vol. 2. Wiley: New York, NY, U.S.A., 1990; 326–334 (BW, DADI), 494–504 (MUSCL), 420–425 (VL–FVS), 440–443 (VL–FVS), 461–469 (Roe FDS), 661–674 (PB). ISBN 0-471-92452-0.
28. Hirsch C. *Numerical Computation of Internal and External Flows* (1st edn), vol. 1. Wiley: New York, NY, U.S.A., 1988; 161–166 (TM), 167–197 (C, UW). ISBN 0-471-92385-0.
29. Brandt A. Multilevel adaptive solutions to boundary-value problems. *Mathematics of Computation* 1977; **31**: 330–390.
30. Claus RW, Vanka SP. Multigrid calculations of a jet in crossflow. *Journal of Propulsion and Power* 1992; **8**:425–431.
31. Shyy W, Sun CS, Chen MH, Chang KC. Multigrid computation for turbulent recirculating flows in complex geometries. *Numerical Heat Transfer*: *Part A* 1993; **23**:79–98.
32. Thakur S, Wright J, Shyy W, Liu J, Ouyang H, Vu T. Development of pressure-based composite multigrid methods for complex fluid flows. *Progress in Aerospace Sciences* 1996; **32**:313–375.
33. Nowak ZP, Salcudean M. Turbulent flow calculations by the nonlinear multigrid method. *Zeitschrift für Angewandte Mathematik und Mechanik* 1996; **76**:463–469.
34. Cornelius C, Volgmann W, Stoff H. Calculation of 3-D turbulent flow with a finite volume multigrid method. *International Journal for Numerical Methods in Fluids* 1999; **31**:703–720.
35. Vázquez M, Ravachol M, Chalot F, Mallet M. The robustness issue on multigrid schemes applied to the Navier–Stokes equations for laminar and turbulent, incompressible and compressible flows. *International Journal for Numerical Methods in Fluids* 2004; **45**:555–579.
36. Yan J, Thiele F, Xue L. A modified full multigrid algorithm for the Navier–Stokes equations. *Computers and Fluids* 2007; **36**:445–454.
37. Gerolymos GA. Implicit multiple-grid solution of the compressible Navier–Stokes equations using $k-\varepsilon$ turbulence closure. *AIAA Journal* 1990; **28**:1707–1717.

DOI: 10.1002/fld

38. Yokota JW. Diagonally inverted lower-upper factored implicit multigrid scheme for the 3-D Navier–Stokes equations. *AIAA Journal* 1990; **28**:1642–1649.
39. Mavriplis DJ, Martinelli L. Multigrid solution of compressible turbulent flow on unstructured meshes using a 2-equation model. *International Journal for Numerical Methods in Fluids* 1994; **18**:887–914.
40. Braaten ME, Connell SD. 3-D unstructured adaptive multigrid scheme for the Navier–Stokes equations. *AIAA Journal* 1996; **34**:281–290.
41. Liu F, Zheng X. A strongly coupled time-marching method for solving the Navier–Stokes and $k-\omega$ turbulence model equations with multigrid. *Journal of Computational Physics* 1996; **128**:289–300 (also AIAA Paper 1994-2389, 1994).
42. Gerlinger P, Brüggemann D. Multigrid convergence acceleration for turbulent supersonic flows. *International Journal for Numerical Methods in Fluids* 1997; **24**:1019–1035.
43. Gerlinger P, Brüggemann D. An implicit multigrid scheme for the compressible Navier–Stokes equations with low-Reynolds-number turbulence closure. *ASME Journal of Fluids Engineering* 1998; **120**:257–262.
44. Carré G. An implicit multigrid method by agglomeration applied to turbulent flows. *Computers and Fluids* 1997; **26**:299–320.
45. Carré G, Fournier L, Lanteri S. Parallel linear multigrid algorithms for the acceleration of compressible flow calculations. *Computer Methods in Applied Mechanics and Engineering* 2000; **184**:427–448.
46. Dick E, Steelant J. Coupled solution of the steady compressible Navier–Stokes and $k-\varepsilon$ turbulence equations with a multigrid method. *Applied Numerical Mathematics* 1997; **23**:49–61.
47. Ålund A, Lötstedt P, Sillén M. Parallel single grid and multigrid solution of industrial compressible flow problems. *Computers and Fluids* 1997; **26**:775–791.
48. Demuren AO, Ibraheem SO. Multigrid method for the Euler and Navier–Stokes equations. *AIAA Journal* 1998; **36**:31–37.
49. Park SH, Kwon JH. Implementation of $k-\omega$ turbulence models in an implicit multigrid method. *AIAA Journal* 2004; **42**:1348–1357.
50. Lambropoulos NK, Koubogiannis DG, Giannakoglou KC. Acceleration of a Navier–Stokes equation solver for unstructured grids using agglomeration multigrid and parallel processing. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:781–803.
51. Bigarella EDV, Azevedo JLF, Scalabrin LC. Centered and upwind multigrid turbulent flow simulations of launch vehicle configurations. *Journal of Spacecraft and Rockets* 2007; **44**:52–65.
52. Launder BE, Spalding DB. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering* 1974; **3**:269–289.
53. Chen HC, Patel VC. Near-wall turbulence models for complex flows including separation. *AIAA Journal* 1988; **26**:641–648.
54. Lien FS, Leschziner MA. Assessment of turbulence-transport models including nonlinear RNG eddy-viscosity formulation and 2-moment closure for flow over a backward-facing step. *Computers and Fluids* 1994; **23**: 983–1004.
55. Brooks AN, Hughes TJR. Streamline-upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
56. Hackbusch W. Comparison of different multigrid variants for nonlinear equations. *Zeitschrift für Angewandte Mathematik und Mechanik* 1992; **72**:148–151.
57. Launder BE, Sharma BI. Application of the energy dissipation model of turbulence to the calculation of flows near a spinning disk. *Letters in Heat and Mass Transfer* 1974; **1**:131–138.
58. Ni RH. A multiple grid scheme for solving the Euler equations. *AIAA Journal* 1982; **20**:1565–1571.
59. Speziale CG, Abid R, Anderson EC. Critical evaluation of 2-equation models for near-wall turbulence. *AIAA Journal* 1992; **30**:324–331.
60. Turner MG, Jennions IK. An investigation of turbulence modeling in transonic fans including a novel implementation of an implicit $k-\varepsilon$ turbulence model. *ASME Journal of Turbomachinery* 1993; **115**:249–260.
61. Wilcox DC. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal* 1988; **26**:1299–1310.
62. Coakley TJ, Huang PG. Turbulence modeling for high speed flows. *AIAA Paper 1992-0436*, 1983.
63. Lam CKG, Bremhorst KA. Modified form of the $k-\varepsilon$ model for predicting wall turbulence. *ASME Journal of Fluids Engineering* 1981; **103**:456–460.
64. Yang Z, Shih TH. New time-scale-based $k-\varepsilon$ model for near-wall turbulence. *AIAA Journal* 1993; **31**:1191–1198.

DOI: 10.1002/fld

65. Chien KY. Predictions of channel and boundary-layer flows with a low-Reynolds number turbulence model. *AIAA Journal* 1982; **20**:33–38.
66. Menter FR. 2-Equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal* 1994; **32**:1598–1605.
67. Wilcox DC. *Turbulence Modelling for CFD* (2nd edn). DCW Industries, La Cañada, CA, U.S.A., 1998. ISBN 0-9636051-5-1.
68. Shih TH, Liou WW, Shabbir A, Yang Z, Zhu J. A new $k-\varepsilon$ eddy viscosity model for high-Reynolds-number turbulent flows. *Computers and Fluids* 1995; **24**:227–238.
69. Wallin S, Johansson AV. An explicit algebraic Reynolds-stress model for incompressible and compressible turbulent flows. *Journal of Fluid Mechanics* 2000; **403**:89–132.
70. Shakib F, Hughes TJR, Johan Z. A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1991; **89**:141–219.
71. Pantakar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere: Washington, DC, U.S.A., 1980.
72. Hughes TJR, Franca LP, Mallet M. A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier–Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering* 1986; **54**:223–234.
73. Mottura L, Vigevano L, Zaccanti M. Factorized implicit upwind methods applied to inviscid flows at high Mach number. *AIAA Journal* 2000; **38**:1846–1852.
74. Jameson A. Solution of the Euler equations for 2-D transonic flow by a multigrid method. *Applied Mathematics and Computation* 1983; **13**:327–355.
75. Saad Y. *Iterative Methods for Sparce Linear Systems* (2nd edn). SIAM: Philadelphia, PA, U.S.A., 2003. ISBN 978-0-898715-34-7.
76. Mavriplis DJ. An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers. *Journal of Computational Physics* 2002; **175**:302–325 (also AIAA Paper 2001-2573, 2001).
77. Ni RH. Multigrid convergence acceleration techniques for explicit Euler solvers and applications to Navier–Stokes calculations. *Numerical Techniques for Viscous Flow Calculations in Turbomachinery Bladings*. No. LS-1986-02, Lecture Series, VKI, 1986.
78. Leclercq MP, Stoufflet PL. Characteristic multigrid method application to solve the Euler equations with unstructured and unnested grids. *Journal of Computational Physics* 1993; **104**:329–346.
79. Gerolymos GA, Sauret E, Vallet I. Oblique-shock-wave/boundary-layer interaction using near-wall Reynolds-stress models. *AIAA Journal* 2004; **42**:1089–1100.
80. Gerolymos GA, Sauret E, Vallet I. Influence of inflow-turbulence in shock-wave/turbulent-boundary-layer interaction computations. *AIAA Journal* 2004; **42**:1101–1106.
81. Vallet I. Reynolds-stress modelling of $M=2.25$ shock-wave/turbulent-boundary-layer interaction. *International Journal for Numerical Methods in Fluids* 2008; **56**:525–555.
82. Vallet I. Reynolds-stress modelling of 3-D secondary flows with emphasis on turbulent diffusion closure. *ASME Journal of Applied Mechanics* 2007; **74**:1142–1156.
83. Gessner FB, Emery AF. The numerical prediction of developing turbulent flow in rectangular ducts. *ASME Journal of Fluids Engineering* 1981; **103**:445–455.
84. Benay R, Délery JM, Pot T. Analyse expérimentale de l'ecoulement dans un canal transsonique 3-D. *Recherche Aérospatiale* 1986; **6**:399–414.
85. Pot T, Délery JM, Quelin C. Interactions onde-de-choc/couche-limite dans un canal transsonique 3-D. *Rap. Tech. 92-7078-AY-116-A*, ONERA, 1991.
86. Settles GS, Vas IE, Bogdonoff SM. Details of a shock-separated turbulent boundary-layer at a compression corner. *AIAA Journal* 1976; **14**:1709–1715.
87. Dolling DS, Murphy MT. Unsteadiness of the separation shock-wave structure in a supersonic compression ramp flowfield. *AIAA Journal* 1983; **21**:1628–1634.
88. Settles GS, Dodson LJ. Supersonic and hypersonic shock/boundary-layer interaction database. *AIAA Journal* 1994; **32**:1377–1383.
89. Vallet I. Aérodynamique numérique 3-D instationnaire avec fermeture Bas-Reynolds au second ordre. *Doctorat*, Université Pierre-et-Marie-Curie, Paris, FRA, 1995.
90. Gerolymos GA, Vallet I. Implicit computation of the 3-D compressible Navier–Stokes equations using $k-\varepsilon$ turbulence closure. *AIAA Journal* 1996; **34**:1321–1330.
91. Sarkar S, Erlebacher G, Hussaini MY, Kreiss HO. The analysis and modelling of dilatational terms in compressible turbulence. *Journal of Fluid Mechanics* 1991; **227**:473–493.

DOI: 10.1002/fld

92. Pope SB. *Turbulent Flows*. Cambridge University Press: Cambridge (GBR) 2000. ISBN 0-521-59125-2.
93. Gerolymos GA, Vallet I. Wall-normal-free Reynolds-stress model for rotating flows applied to turbomachinery. *AIAA Journal* 2002; **40**:199–208.
94. Gerolymos GA, Vallet I. Low-diffusion approximate Riemann solvers for Reynolds-stress transport. *AIAA Computational Fluid Dynamics Conference*, Miami, FL, U.S.A., 25–28 June 2007. AIAA Paper 2007-4467, 18.
95. So RMC, Yuan SP. A geometry independent near-wall Reynolds-stress closure. *International Journal of Engineering Science* 1999; **37**:33–57.
96. Cahen J, Couailler V, Délery JM, Pot T. Validation of code using turbulence model applied to 3-D transonic channel. *AIAA Journal* 1995; **33**:671–679.
97. Caughey DA. Implicit multigrid computation of unsteady flows past cylinders of square cross-section. *Computers and Fluids* 2001; **30**:939–960.
98. Venkatakrishnan V. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *Journal of Computational Physics* 1995; **118**:120–130.
99. Amaladas JR, Kamath H. Accuracy assessment of upwind algorithms for steady-state computations. *Computers and Fluids* 1998; **27**:941–962.
100. Vinokur M. An analysis of finite-difference and finite-volume formulations of conservation laws. *Journal of Computational Physics* 1989; **81**:1–52.
101. Anderson WK, Thomas JL, Van Leer B. Comparison of finite-volume flux-vector-splittings for the Euler equations. *AIAA Journal* 1986; **24**:1453–1460.
102. Anderson WK, Thomas JL, Rumsey CL. Extension and application of flux-vector-splitting to calculations on dynamic meshes. *AIAA Journal* 1989; **27**:673–674 (also AIAA Paper 1987-1152, 1987).
103. Gerolymos GA, Vallet I. Advances in the numerical computation of complex flows using Reynolds-stress models. *AIAA Computational Fluid Dynamics Conference*, Miami, FL, U.S.A., 25–28 June 2007. AIAA Paper 2007-3963, 18.
104. Rogers SE, Kwak D, Kris C. Steady and unsteady solutions of the incompressible Navier–Stokes equations. *AIAA Journal* 1991; **29**:603–610.
105. Jameson A. Time-dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA Paper 1991-1596*, 1991.
106. Rumsey CL, Sanetrik MD, Bierdon RT, Melson ND, Parlette EB. Efficiency sand accuracy of time-accurate turbulent Navier–Stokes computations. *Computers and Fluids* 1996; **25**:217–236 (also AIAA Paper 1995-1835).
107. Dubuc L, Cantariti F, Woodgate M, Gribben B, Badcock KJ, Richards BE. Solution of the unsteady Euler equations using an implicit dual-time method. *AIAA Journal* 1998; **36**:1417–1424.
108. Chakravarthy SR. Euler equations—implicit schemes and boundary conditions. *AIAA Journal* 1983; **21**:699–706.
109. Schumann U. Realizability of Reynolds-stress turbulence models. *Physics of Fluids* 1977; **20**:721–725.
110. Golub GH, Van Loan CF. *Matrix Computations*. The John Hopkins University Press: Baltimore, MD, U.S.A., 1989; 150–152. ISBN 0-8018-5414-8.

DOI: 10.1002/fld