INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2010; **63**:495–516
Published online 10 June 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.2091

Assessment of the finite volume method applied to the v^2–f model

Mikael Mortensen*, †, Bjørn Anders Pettersson Reif and Carl Erik Wasberg

Norwegian Defence Research Establishment (FFI), NO-2027 Kjeller, Norway

SUMMARY

The objective of this paper is to assess the accuracy of low-order finite volume (FV) methods applied to the v^2–f turbulence model of Durbin (*Theoret. Comput. Fluid Dyn.* 1991; **3**:1–13) in the near vicinity of solid walls. We are not (like many others) concerned with the stability of solvers - the topic at hand is simply whether the mathematical properties of the v^2–f model can be captured by the given, widespread, numerical method. The v^2–f model is integrated all the way up to solid walls, where steep gradients in turbulence parameters are observed. The full resolution of wall gradients imposes quite high demands on the numerical schemes and it is not evident that common (second order) FV codes can fully cope with such demands. The v^2–f model is studied in a statistically one-dimensional, fully developed channel flow where we compare FV schemes with a highly accurate spectral element reference implementation. For the FV method a higher-order face interpolation scheme, using Lagrange interpolation polynomials up to arbitrary order, is described. It is concluded that a regular second-order FV scheme cannot give an accurate representation of all model parameters, independent of mesh density. To match the spectral element solution an extended source treatment (we use three-point Gauss–Lobatto quadrature), as well as a higher-order discretization of diffusion is required. Furthermore, it is found that the location of the first internal node need to be well within $y^+=1$. Copyright © 2009 John Wiley & Sons, Ltd.

Received 27 November 2008; Revised 16 April 2009; Accepted 18 April 2009

KEY WORDS: RANS modelling; turbulence modeling; v^2–f model; finite volume method; spectral element method; elliptic relaxation

1. INTRODUCTION

The application of computational fluid dynamics (CFD) for complex flow system analyses relevant for industrial applications continues to be dominated by the Reynolds Averaged Navier–Stokes (RANS) approach. There exist a hierarchy of RANS models ranging from complex differential second moment closures (SMCs) based on dynamical equations governing the individual components of the Reynolds-stress tensor, to simpler algebraic formulas that rely on an explicit relationship

*Correspondence to: Mikael Mortensen, Norwegian Defence Research Establishment (FFI), NO-2027 Kjeller, Norway.
†E-mail: Mikael.Mortensen@ffi.no

Contract/grant sponsor: WALLTURB; contract/grant number: AST4-CT-2005-516008

between the turbulent stresses and the mean flow field. The latter is referred to as eddy-viscosity (EV) models which in its most simplistic form utilizes a linear relationship between the turbulent stresses and the mean rate-of-strain tensor.

The linear EV models vary in complexity and applicability and are usually grouped by the number of additional equations that are required to model the turbulent viscosity. At the top of this EV hierarchy Durbin [1] introduced the elliptic relaxation approach and the so-called v^2–f model, using four additional transport equations. In the elliptic relaxation approach, which is rather different from the traditional damping function approach, elliptic wall effects are accounted for indirectly through the solution of a modified Helmholtz equation, cf. e.g. [1–3]. The v^2–f model is independent of wall topography parameters, which makes it applicable to arbitrarily complex geometries.

The v^2–f model has performed well for a wide variety of turbulent flows, ranging from flat plate boundary layers to massively separated flows [1, 4–6], which clearly demonstrates the importance of near-wall modelling. However, despite its physical appeal and good performance, the v^2–f model is still rarely utilized by the industry. This is often attributed to inherent numerical properties, which makes the model difficult to converge to stable numerical solutions using segregated solvers. Consequently some alternative formulations that focuses on numerical stability and the segregated solver issues have appeared. Lien and Kalitzin [7] altered the definition of f in order to obtain a homogeneous Dirichlet boundary condition on the wall (i.e. $f=0$). Owing to this boundary condition the model is stable when implemented with segregated solvers, but the results are generally not as good as for the original model. Other modifications focusing on the solver issues have been considered by Laurence *et al.* [8], Hanjalić *et al.* [9] and Billard *et al.* [10]. However, as noted by Billard *et al.* [10]: 'numerical solutions brought to make the model of Durbin 'code-friendly' are always done at the expense of prediction accuracy'. For this reason (and the fact that the coupled implicit solvers required by the original model generally perform better than the segregated solvers in terms of stability and convergence rate) we will here focus on the original model. However, we would like to stress that the current paper focuses on the numerical methods used for the discretization of the v^2–f model and not the segregated solver issues.

The integration of the model equations all the way up to a solid wall imposes quite high demands on the stability and accuracy of the numerical scheme. For practical reasons the mesh is highly squeezed while approaching the wall, and extremely high mesh-densities are often required in the wall-normal direction for RANS based models. In the near-wall region the leading transport term is diffusion and a finite volume (FV) scheme is then required to estimate first derivatives on control volume faces. A regular cell- or face-centred FV scheme uses second-order central differencing to approximate face-gradients. Consider then that v^2 approaches the wall as $O(y^4)$, where y is the wall-normal coordinate. The regular FV scheme then in effect tries to represent a fourth-order polynomial by a second-order method, and acceptable results can thus only be produced if a very dense mesh is used as well. Such issues has lead us to question whether FV codes in common use really do give an appropriate representation of the mathematical properties of the v^2–f model, or whether more sophisticated numerical methods are required? The objective of the current work is to properly validate the common FV methods for near-wall modelling of the v^2–f model. To this end we will assess mesh requirements and study the effect of using higher-order face interpolation schemes for diffusion near the wall. For validation of the finite volume method applied to the v^2–f model, we utilize a reference solution based on the highly accurate spectral element method (Patera [11]).

This paper is organized as follows: in Section 2 the original linear v^2–f models including boundary conditions is presented. In Section 3 we focus on the FV and spectral element implementations and in Section 4 the performance of the numerical schemes is discussed. Conclusions are drawn in Section 5.

2. THE v^2–f MODEL

In this section the original v^2–f turbulence model is briefly presented as well as its somewhat irregular boundary conditions.

2.1. *Original model*

The original v^2–f model of Durbin [1] is built on top of the standard two-equation k–ε model, where ε is the dissipation rate of turbulent kinetic energy k. The mean velocity vector $\mathbf{U}=[U, V, W]$ in RANS is computed from

$$\frac{\partial U_i}{\partial t}+U_j\frac{\partial U_i}{\partial x_j}=\frac{\partial}{\partial x_j}\left[\nu\frac{\partial U_i}{\partial x_j}\right]-\frac{1}{\rho}\frac{\partial P}{\partial x_i}-\frac{\partial\overline{u_iu_j}}{\partial x_j} \tag{1}$$

where ν is the kinematic viscosity, P is the mean pressure and u_i is a component of the fluctuating velocity vector. Equation (1) is complemented by the continuity equation, $\nabla\cdot\mathbf{U}=0$. In Equation (1) and throughout this work there is, unless otherwise stated, summation implied by repeating lowercase Roman indices, whereas there is no summation implied by repeating Greek indices.

The Reynolds stresses $\overline{u_iu_j}$ require closure in terms of resolved (known) parameters and the linear EV model reads

$$\overline{u_iu_j}=-2\nu_TS_{ij}+\tfrac{2}{3}k\delta_{ij} \tag{2}$$

where ν_T is the turbulent viscosity, the mean rate of strain tensor $S_{ij}=0.5(\partial U_i/\partial x_j+\partial U_j/\partial x_i)$ and δ_{ij} is the Kronecker delta function that is unity for $i=j$ and zero otherwise.

The standard k–ε model closes the EV ν_T in terms of k and ε, i.e. $\nu_T=C_\mu kT_{k\varepsilon}$, where $T_{k\varepsilon}=k/\varepsilon$ is a turbulent timescale. The modelled transport equations for k and ε are

$$\frac{\partial k}{\partial t}+U_j\frac{\partial k}{\partial x_j}=\frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\nu_T}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]+P_k-\varepsilon \tag{3}$$

and

$$\frac{\partial \varepsilon}{\partial t}+U_j\frac{\partial \varepsilon}{\partial x_j}=\frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\nu_T}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right]+\frac{C_{\varepsilon1}P_k-C_{\varepsilon2}\varepsilon}{T} \tag{4}$$

where σ_k and σ_ε are model constants. The rate of production of turbulent kinetic energy $P_k=-\overline{u_iu_j}\partial U_i/\partial x_j$.

In an effort to improve the prediction of ν_T, the v^2–f model solves two additional scalar equations for v^2 and f

$$\frac{\partial v^2}{\partial t}+U_j\frac{\partial v^2}{\partial x_j}=\frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\nu_T}{\sigma_{v^2}}\right)\frac{\partial v^2}{\partial x_j}\right]+kf-v^2\frac{\varepsilon}{k} \tag{5}$$

DOI: 10.1002/fld

and

$$L^2 \frac{\partial^2 f}{\partial x_j \partial x_j} = f + \frac{C_1 - 1}{T}\left(\frac{v^2}{k} - \frac{2}{3}\right) - C_2 \frac{P_k}{k} \tag{6}$$

where T and L are time- and lengthscales determined by

$$T = \max\left\{\frac{k}{\varepsilon},\ 6\sqrt{\frac{v}{\varepsilon}}\right\} \tag{7}$$

and

$$L = C_L \min\left\{\frac{k^{3/2}}{\varepsilon},\ C_\eta \left(\frac{v^3}{\varepsilon}\right)^{1/4}\right\} \tag{8}$$

respectively. Through Equations (5) and (6) an additional and more appropriate near-wall velocity scale $\sqrt{v^2}$ is obtained and the turbulent viscosity can now be computed as

$$v_T = C_{\mu 1} v^2 T \tag{9}$$

This model has shown much closer agreement with direct numerical simulations and experiments of near-wall turbulence than the common $v_T = C_\mu k T_{k\varepsilon}$, which also is the main motivation for the v^2–f model.

The parameters of the v^2–f model vary somewhat in the literature, but are most commonly set to $C_{\mu 1} = 0.22$, $C_{\text{ed}} = 0.045$, $C_{\varepsilon 2} = 1.9$, $C_1 = 1.4$, $C_2 = 0.3$, $\sigma_k = 1.0$, $\sigma_{v^2} = 1.0$, $\sigma_\varepsilon = 1.3$, $C_L = 0.25$, and $C_\eta = 80$. To avoid using the distance to the nearest wall directly, the last parameter $C_{\varepsilon 1}$ is most often computed as

$$C_{\varepsilon 1} = 1.4\left(1 + C_{ed}\sqrt{\frac{k}{v^2}}\right) \tag{10}$$

which is a slight modification of the original model.

2.2. Boundary conditions

On no slip boundaries (walls) in parallel shear flows it is well known that k, v^2 (here interpreted as the wall normal stress), ε, and f behave asymptotically as [12]

$$k \sim O(y^2), \quad v^2 \sim O(y^4), \quad \varepsilon \sim O(1), \quad f \sim O(1) \tag{11}$$

as the wall is approached ($y \longrightarrow 0$). Here y denotes the coordinate normal to the wall. In other words as $y \longrightarrow 0$ the proper wall boundary conditions are $k = 0$ and $v^2 = 0$. By considering Equations (3) and (5) asymptotically near the wall we can also obtain

$$\varepsilon \longrightarrow 2v\frac{k}{y^2} \tag{12}$$

and

$$kf \longrightarrow -5\frac{v^2 \varepsilon}{k} \quad \text{or} \quad f \longrightarrow -\frac{20 v^2 v^2}{\varepsilon y^4} \tag{13}$$

DOI: 10.1002/fld

Note that the boundary condition for f involves y^4 and ε in the denominator. This boundary condition has been discussed by several authors as being responsible for numerical instabilities, due to the strong coupling with v^2. As a consequence the v^2–f model has usually had to be implemented with a coupled solver (solving systems of equations, k and ε together and then v^2 and f together), and not the more common segregated solvers found in most commercial CFD-softwares. As discussed in the Introduction, the need for a coupled solver has motivated several alternative 'code-friendly' versions of the v^2–f model [13, 8, 9], where f in general has been redefined such that $f_{\text{wall}} = 0$. In the current work the original model has been implemented with a coupled solver that implicitly sets the boundary conditions and numerical instability is thus not an issue here.

The non-standard wall boundary conditions given in Equations (12) and (13) cannot be set directly at the wall since both numerator and denominator here are zero. The alternative in practise is to prescribe the dependent variables at the first internal node. For a second-order FV method this means that the values of f and ε at the wall do not enter the computations. On the other hand, for higher-order FV methods and the spectral element method, the values of f and ε at the wall need to be computed just like any other internal node. Although the boundary conditions (12) and (13) are not singular at the wall in the limit of a converged solution, singularity problems may still arise in the coarse towards a converged state.

2.3. Modifications to the v^2–f model

Two slight modifications have been made to the original v^2–f model. The objective behind these changes has merely been to remove some non-physical features in the near-wall and free-stream regions.

The first modification is a restriction on the source term kf in the v^2 equation (5), suggested in [14]. The restriction reads

$$kf_S = \min\left(kf,\ \frac{C_1-1}{T}\left(\frac{2k}{3} - v^2\right) + C_2 P_k\right) \tag{14}$$

where kf_S is used for kf on the right-hand side of Equation (5). Note that the two terms in Equation (14) constitutes (k times) the right-hand side of the f-equation (6). The physical reasoning behind the restriction is to ensure that kf is not larger than its isotropic value (and thus $\nabla^2 f < 0$) in the source to the v^2 equation. Since $\nabla^2 f < 0$ close to walls the modification only affects the free-stream, where its effect basically is to keep the v^2 less than its isotropic value $2k/3$.

The second modification concerns the max and min operators used in (7), (8) and (14). The arguments for using (7) and (8) is that the appropriate length and time-scales, respectively, are different for the near-wall region as compared with the outer flow region. However, these operators lead to terms that are non-differentiable with respect to the wall normal coordinate at a certain point of intersection. This behaviour is non-physical and degrades the accuracy of the numerical simulation that assumes a piecewise smooth solution exists. To circumvent this problem we utilize a limit representation (see [15, 16]) of the max and min functions that reads

$$\max_{\varepsilon}(a, b) = \lim_{\varepsilon \to 0^+} \varepsilon \ln(e^{a/\varepsilon} + e^{b/\varepsilon}) \tag{15}$$

and

$$\min_{\varepsilon}(a, b) = \lim_{\varepsilon \to 0^-} \varepsilon \ln(e^{a/\varepsilon} + e^{b/\varepsilon}) \tag{16}$$

DOI: 10.1002/fld

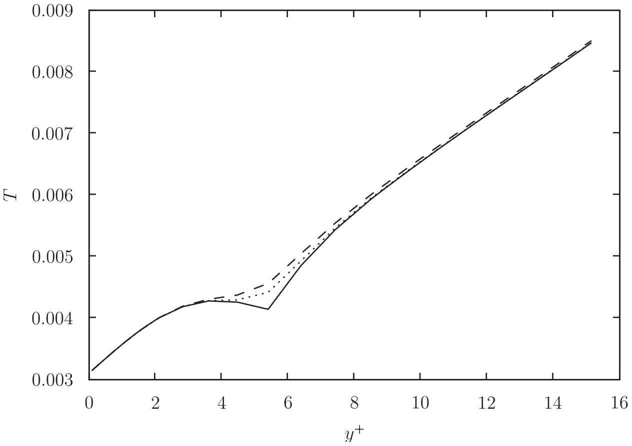

Figure 1. The timescale T is shown here with a solid line for the regular max function, whereas the dashed and dotted lines correspond to T computed from (15) with $\varepsilon = 0.15$ and 0.1, respectively.

These formulas are scaled first with the regular max function and then a finite value of ε is used. The impact on the timescale T is shown for $y^+ < 20$ in Figure 1, where T computed with $\varepsilon = 0.1$ and 0.15 are represented with dotted and dashed lines, respectively. The value $\varepsilon = 0.1$ has been used throughout this paper and it has been confirmed that the particular choice ($\varepsilon \neq 0$) does not influence the numerically converged results significantly.

3. NUMERICAL IMPLEMENTATION

The finite volume (FV) method has since long been the preferred choice amongst the fluid dynamics community for the discretization of the RANS equations. The justification for using the FV method is usually that it is easy to adapt to complex 3D geometries and the method's integral formulation also guarantees conservation of transported quantities, a feature of great importance in fluid flow computations.

In this work we wish to focus on the near-wall behaviour, with a particular emphasize on the numerics, and are thus not concerned with the general 3D discretization. For more complex flows convection will naturally also play an important role for transport, but near a wall the viscous diffusion and viscous dissipation terms will always be of greatest importance, regardless the complexity of the geometry.

To validate the numerics in use by most commercial CFD softwares, the v^2–f model has been implemented using standard numerical methods for FV as well as a higher-order extension. For numerical accuracy validation, a spectral element code of arbitrary order (and accuracy) has been implemented. All computations have been performed with a semi-coupled solver that implicitly updates the boundary conditions described in Section 2.2. Furthermore, dissipation and production terms are accounted for implicitly and explicitly, respectively. The solution procedure is then in a single iteration to first solve and update the mean velocity U, then both k and ε, and finally v^2 and f.

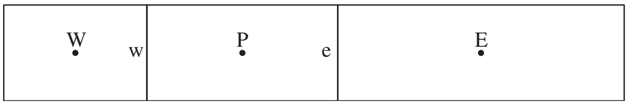

Figure 2. Cell centred finite volume mesh. The nodes are represented with capital letters, whereas the faces have lowercase. The notation W and E represent the nodes to the west and east of the central node P. The closed rectangles surround the three control volumes.

3.1. FV implementation

In this section the standard second-order FV discretization of the v^2–f model in one dimension (1D) is described and a general extension to higher order using Lagrange interpolation is detailed.

In statistically stationary 1D channels the v^2–f model can be simplified to

$$\frac{\partial}{\partial y}\left(\Gamma^{\phi}\frac{\partial \phi}{\partial y}\right)=-S^{\phi} \quad \text{for } \phi=U,k,\varepsilon,v^2 \text{ and } f \tag{17}$$

where $\Gamma^{\phi}=1$ for $\phi=f$ and $\Gamma^{\phi}=v+v_T/\sigma_{\phi}$ for $\phi=U,k,\varepsilon$ and v^2, respectively. Likewise, the sources S^{ϕ} are given in Equations (1), (3), (4), (5), and (6), respectively.

The FV mesh with common notation is shown for a cell-centred scheme in Figure 2. Additional nodes are placed on the faces located at the wall and at the centre of the channel. It should be mentioned that the alternative face-centred scheme that places faces midway between nodes has also been tested. However, the results (at least for a second-order scheme) are very similar between the two schemes, and thus only the cell-centred are reported.

In the FV method, Equation (17) is integrated over a control volume (CV), which transforms the equation into

$$\begin{aligned} \oint_A \Gamma^{\phi}\frac{\partial \phi}{\partial y}\,\mathrm{d}A &= -\int_V S^{\phi}\,\mathrm{d}V \\ \left(\Gamma^{\phi}\frac{\partial \phi}{\partial y}\right)_e-\left(\Gamma^{\phi}\frac{\partial \phi}{\partial y}\right)_w &= -S^{\phi}\,\mathrm{d}y_P \end{aligned} \tag{18}$$

Here $\mathrm{d}y_P=y_e-y_w$ is the size of the CV and the surface gradients need to be computed from the node values (or cell averages). (Note that since the channel is 1D, the area of any face $\mathrm{d}A$ can be set to unity.) The second-order central differencing scheme for the gradient is then

$$\left(\Gamma^{\phi}\frac{\partial \phi}{\partial y}\right)_e=\Gamma^{\phi}_e\frac{\phi_E-\phi_P}{y_E-y_P} \tag{19}$$

To compute the production P_k, the cell-averaged gradient of U is required. With the FV method this term is formally computed as

$$\left(\frac{\partial U}{\partial y}\right)_P=\frac{\displaystyle\int_V \frac{\partial U}{\partial y}\,\mathrm{d}V}{\displaystyle\int_V \mathrm{d}V}=\frac{\oint_A U\,\mathrm{d}A}{\mathrm{d}A\,\mathrm{d}y_P} \tag{20}$$

DOI: 10.1002/fld

which in 1D simplifies to

$$\left(\frac{\partial U}{\partial y}\right)_P = \frac{U_e - U_w}{\mathrm{d}y_P} \tag{21}$$

The face value of any cell-based variable such as Γ_e^ϕ or U_e can then be computed using linear interpolation from the two neighbouring nodes. In the production term the velocity gradients are squared, which amplifies any error in the discretization. On the positive note, the velocity gradient approaches the wall as $O(1)$, and the low-order schemes should thus be sufficient and no special attention is required.

The standard FV discretization scheme described above is found in most regular FV codes. Higher-order schemes associated with FV methods are generally more concerned with convection (e.g. QUICK, TVD, ENO, WENO). However, the FV method can also be extended for diffusion by using more sophisticated operators for the interpolation of gradients to faces. One possibility is then to use a Lagrange interpolating polynomial that can be obtained at any desired order. Each Lagrangian interpolant is an Nth-order polynomial, defined as (no summation implied by repeating Greek indices)

$$h_\alpha(y) = \frac{a_\alpha(y)}{a_\alpha(y_\alpha)}, \quad \alpha = 0, \ldots, N \tag{22}$$

where the polynomials $a_\alpha(y)$ are given by

$$a_\alpha(y) = \prod_{l=0, l\neq\alpha}^{N} (y - y_l), \quad \alpha = 0, \ldots, N \tag{23}$$

In our 1D FV scheme the $N+1$ closest neighbouring nodes are used in the interpolation for any face k. The range of interpolation nodes for face k is thus computed as $j = m_k, \ldots, m_k + N$, where the lower limit, using zero-offset indexing, is computed as

$$m_k = \min\left\{\max\left[0, \left(k - \frac{N}{2}\right)\right], NJ - N - 1\right\} \tag{24}$$

Here it is assumed that the mesh consists of a total of NJ nodes, $NJ-2$ CVs, and integer division is performed with the *floor function* [17].

For the region spanned by the chosen interpolation nodes any function $\xi(y)$ collocated in the nodes (e.g. S^ϕ and Γ^ϕ) can be represented by the interpolating polynomial (with summation over repeating indices)

$$\xi(y) = \xi_i h_i(y), \quad i = 0, \ldots, N \tag{25}$$

and the derivative of this function as

$$\frac{\partial \xi(y)}{\partial y} = \xi_i \frac{\partial h_i(y)}{\partial y}, \quad i = 0, \ldots, N \tag{26}$$

Here the node values are written in short notation as $\xi_i = \xi(y_i)$. It is now useful to define an interpolation matrix $\mathbf{H}$ for the interpolation from nodes to faces. Any function interpolated from $N+1$ surrounding nodes to face k can then be defined as

$$\xi(y_k) = H_{kj}\xi_j = h_j(y_k)\xi_j \tag{27}$$

DOI: 10.1002/fld

The matrix **H**'s column index j runs over NJ nodes and row index k over $NJ-1$ faces. However, there are only $N+1$ non-zero entries on each row, found in the range computed with Equation (24).

A derivative matrix interpolating from $N+1$ surrounding nodes to face k can similarly be defined as

$$\frac{\partial \xi(y_k)}{\partial y} = H'_{kj}\xi_j = \frac{\partial h_j(y_k)}{\partial y}\xi_j, \quad j = m_k, \ldots, m_k+N \tag{28}$$

Note that the matrices **H** and $\mathbf{H}'$ depend only on the mesh and thus can be precomputed, cf. Berrut and Trefethen [18], and used throughout computations.

The FV formulation of Equation (18) of arbitrary order for the face interpolation becomes for CV α (in regular FV notation indices α and $\beta=\alpha+1$ here represent faces west and east, respectively)

$$(H_{\beta i}\Gamma_i^{\phi} H'_{\beta j} - H_{\alpha i}\Gamma_i^{\phi} H'_{\alpha j})\phi_j = -\int_{y_\alpha}^{y_\beta} S^{\phi}\,\mathrm{d}V \tag{29}$$

In the results section the Lagrange interpolants have been tested up to the 8th order, but it is trivial (though not necessarily stable) to extend the method all the way up to $N = NJ-1$.

The final term that requires some attention in the FV discretization of the v^2–f model is the source term. In (18) the source has been discretized by the standard midpoint rule, using only the cell-average:

$$\int_V S^{\phi}\,\mathrm{d}V \approx S_P^{\phi}\,\mathrm{d}y_P \tag{30}$$

It is easy to show that this first-order approximation is not sufficiently accurate for the v^2–f model near a solid wall. Consider the scalar v^2 that approaches a solid wall asymptotically as $O(y^4)$. Performing two derivatives of v^2 then reveals that the source to the v^2 equation will behave as $O(y^2)$ and naturally the cell-centred collocated variable cannot possibly reproduce the correct result. For the first internal cell, we can do some hand calculations and obtain the exact result

$$\int_0^{\triangle y} y^2\,\mathrm{d}y = \frac{(\triangle y)^3}{3} \tag{31}$$

where $\triangle y$ is the size of the first internal CV. Comparing this exact result with the numerical FV integration that uses the midpoint rule

$$\int_0^{\triangle y} y^2\,\mathrm{d}y \approx \left(\frac{\triangle y}{2}\right)^2 \triangle y = \frac{(\triangle y)^3}{4} \tag{32}$$

it is evident that the source term needs special attention close to the wall. A three point Gauss–Lobatto quadrature is exact for polynomials up to order 3 and thus sufficient for the problem under consideration. For a cell-centred FV-scheme the source can then be computed as

$$\int_V S^{\phi}\,\mathrm{d}V \approx \left(\frac{1}{6}(S_w^{\phi} + S_e^{\phi}) + \frac{2}{3}S_P^{\phi}\right)\mathrm{d}y_P \tag{33}$$

which is exact for $S = y^2$. Here the face values can be computed using the existing interpolation matrix.

DOI: 10.1002/fld

At this point it is necessary to mention that even though the face interpolations can be extended to higher order using more neighbouring nodes, the original FV formulations, that is Equations (18) and (20), are still basically two-point differences in 1D. For this reason we find it necessary to use a more advanced technique, the spectral element method, to obtain a reference solution.

3.2. Spectral element implementation

The spectral element method, originally introduced by Patera [11], can be viewed as a finite element method with higher-order polynomials as basis functions. Such a high-order method, with polynomial order typically between 8 and 20, can produce highly accurate results for problem with smooth solutions, with only a small number of computational nodes, which makes it ideally suited for producing reference solutions.

Equation (17) can be written on variational form by multiplying with the test functions v and integrating by parts

$$\int_V \Gamma^\phi \frac{\partial\phi(y)}{\partial y}\frac{\partial v(y)}{\partial y}\,\mathrm{d}V=\int_A v(y)\Gamma^\phi\frac{\partial\phi(y)}{\partial y}\,\mathrm{d}A+\int_V S^\phi(y)v(y)\,\mathrm{d}V \tag{34}$$

Zero gradient (natural) boundary conditions are imposed by ignoring the first term on the right-hand side, while boundary conditions on ϕ (essential) are implemented by modifying the final elemental and stiffness matrices.

Next, we discretize by restricting the solution and the test functions to the piecewise polynomial spaces $X_d=H^1\cap P_{N,K}$ and $X_d^0=H_0^1\cap P_{N,K}$, respectively. Here H^1 is the Sobolov space of functions satisfying the boundary conditions presented in Section 2.2 and H_0^1 is the corresponding space with homogeneous boundary conditions. The computational domain V is decomposed into K elements: $V=\bigcup_{k=1}^K V_k$ and $P_{N,K}$ is the space of piecewise polynomials of degree $\leqslant N$, on element K.

The problem becomes: find the discrete solution $\phi_d\in X_d$ (subscript d indicates a discrete representation) such that

$$\forall v_d\in X_d^0,\quad \int_V \Gamma^\phi\frac{\partial\phi_d(y)}{\partial y}\frac{\partial v_d(y)}{\partial y}\,\mathrm{d}V=\int_V S^\phi(y)v_d(y)\,\mathrm{d}V \tag{35}$$

In order to close this variational form a particular set of basis and test functions needs to be chosen and in the spectral element method these polynomials are generally taken from the Jacobi polynomials. From this family of functions use will be made of the Legendre polynomials, which are given by the recurrence relation

$$L_0(y)=1,\quad L_1(y)=y,\quad L_{k+1}=\frac{2k+1}{k+1}yL_k(y)-\frac{k}{k+1}L_{k-1},\quad k\geqslant 2 \tag{36}$$

where $L_N(y)$ is the Legendre polynomial of degree N. By choosing these Legendre polynomials we can use the Gauss–Lobatto integration formula, which defines integration points z_j and weights w_j, such that the following integration formula holds:

$$p_j w_j=\int_{-1}^{1}p(y)\,\mathrm{d}y,\quad \forall p(y)\in P_{2N-1}[-1,1],\quad j=0,\dots,N \tag{37}$$

where $p_j=p(z_j)$. The integration is exact for polynomials of order up to $2N-1$. The Gauss–Lobatto–Legendre (GLL) points z_j on $[-1,1]$ are defined as the extrema of the Nth-order Legendre polynomial $L_N(y)$, in addition to the endpoints. The GLL points must be computed numerically, cf. Canuto *et al.* [19].

Assuming that each element in V is given as $V_k=[y_k, y_{k+1}]$, the global mapping of z_j and w_j from $[-1,1]$ to V_k is given as

$$z_j^k=y_k+(z_j+1)l_k/2, \quad w_j^k=w_j l_k/2, \quad j=0,\dots,N, \;\; k=1,\dots,K \tag{38}$$

where $l_k=y_{k+1}-y_k$ is the length of element V_k.

Using GLL integration, (35) can be reformulated as (no summation implied by repeating indices since the sums here are explicitly stated)

$$\forall v_d\in X_d, \quad \sum_{k=1}^{K}\sum_{i=0}^{N}\Gamma^{\phi}(z_i^k)\frac{\partial\phi_d(z_i^k)}{\partial y}\frac{\partial v_d(z_i^k)}{\partial y}w_i^k=\sum_{k=1}^{K}\sum_{i=0}^{N}S^{\phi}(z_i^k)v_d(z_i^k)w_i^k \tag{39}$$

In solving for Equation (39) we let the local basis and test functions be the Lagrangian interpolants defined in Equations (22) and (23) at the GLL grid points, i.e. $v_d(y)=h_j(y)$. Global basis and test functions $h_j^k(y)$ are defined on element k as the mapping of $h_j(z)$ from local to global coordinates, and as zero outside element k.

The final formulation of the problem is obtained by inserting for the basis functions, using the representation in (25), and test functions in (39). Using the short hand notation $h_{ij}^k=h_i^k(z_j^k)$ and explicitly stating the summations we obtain

$$\sum_{k=1}^{K}\sum_{m=0}^{N}\sum_{n=0}^{N}\Gamma_m^{\phi,k}\phi_n^k\frac{\partial h_{nm}^k}{\partial y}\frac{\partial h_{im}^k}{\partial y}w_m^k=\sum_{k=1}^{K}\sum_{n=0}^{N}S_n^{\phi,k}\delta_{in}w_n^k \tag{40}$$

Despite the somewhat messy notation that follows from using a higher-order polynomial this equation can be rewritten on compact tensor form as

$$C_{ij}^{\phi,k}\phi_j^k=B_{ij}^k S_j^{\phi,k} \tag{41}$$

where

$$C_{\alpha\beta}^{\phi,\gamma}=D_{l\alpha}^{\gamma}B_{lm}^{\gamma}G_{mn}^{\gamma}D_{n\beta}^{\gamma} \tag{42}$$

$$B_{\alpha\beta}^{\gamma}=\delta_{\alpha\beta}w_{\beta}^{\gamma} \tag{43}$$

$$G_{\alpha\beta}^{\gamma}=\delta_{\alpha\beta}\Gamma_{\beta}^{\phi,\gamma} \tag{44}$$

$$D_{\alpha\beta}^{\gamma}=\frac{2}{l_\gamma}D_{\alpha\beta}=\frac{\partial h_{\beta\alpha}^{\gamma}}{\partial y} \tag{45}$$

Here $\mathbf{C}^{\phi,\gamma}$ and $\mathbf{B}^{\gamma}$ are the elemental stiffness and mass matrices, respectively, for element γ and $\mathbf{G}^{\gamma}$ is the diagonal matrix with $\Gamma^{\phi,\gamma}$ on the diagonal. The derivative matrix $\mathbf{D}$ can be computed for a general set of points on $[-1,1]$ as

$$D_{\alpha\beta}=\begin{cases}\dfrac{a_\alpha(z_\alpha)}{(z_\alpha-z_\beta)a_\beta(z_\beta)}, & \alpha\neq\beta\\[2ex] \displaystyle\sum_{l=0,l\neq\beta}^{N}\frac{1}{z_\beta-z_l}, & \alpha=\beta\end{cases} \tag{46}$$

The nodes located at the interface between adjacent elements need special attention, because these nodes are used by two elements. All other nodes are used solely by the element they belong to. To deal with the interfaces a global system can be assembled by running through all elements and adding up entries that refer to the same point. Hence we can derive a global equation set

$$C^{\phi}_{mn}\phi_n = B_{mn}S^{\phi}_n \tag{47}$$

where the global matrices $\mathbf{C}^{\phi}$ and $\mathbf{B}$ can be assembled as $C^{\phi}_{mn} += C^{\phi,k}_{ij}$ and $B_{mn} += B^{k}_{ij}$. The global indexes m and n are computed from the elemental and local indexes, i.e. $m = m(k,i)$ and $n = n(k,j)$. If we define N_e as a vector that holds the polynomial order of each element and $N_e(0) = 0$, then $m(k,i) = \sum_{l=0}^{k-1} N_e(l) - (k-1) + i$ and $n(k,j) = \sum_{l=0}^{k-1} N_e(l) - (k-1) + j$. The global matrices $\mathbf{C}^{\phi}$ and $\mathbf{B}$ must finally be modified to introduce the Dirichlet boundary conditions at the wall. The natural symmetry conditions on the centreline require no special attention due to the weak formulation of the equations (see Equation (34)).

4. RESULTS

The major objective of this paper is to investigate numerical issues relating to the linear v^2–f model. The model predicts statistical properties of turbulent flows that are integrated all the way up to solid walls. Since the boundary layer in the limit as the wall is approached basically is statistically 1D, even for complex flows, we consider only the fully developed turbulent channel flow. The height of the simulated channel is $2H$ and due to symmetry only half the channel is computed. Unless otherwise stated, the wall unit based Reynolds number used throughout the simulations is $Re_{\tau} = U_{\tau}H/\nu = 2000$, where the wall friction velocity U_{τ} is set to unity, and $H = 0.5$. This Re_{τ} is chosen as a compromise between greater grid density requirements for a higher Re_{τ} and the desire to have a separation between the near-wall region and the outer part of the flow (measured in wall-units).

4.1. The mesh

A turbulent boundary layer in equilibrium is usually divided into four regions, based on the normalized distance to the wall. The wall units used for the normalization is based on the friction velocity, $U_{\tau} = (\sqrt{\nu\partial U/\partial y})_{\text{wall}}$, where U is the tangential wall velocity and y is the wall normal direction. The dimensionless wall normal distance is $y^+ = yU_{\tau}/\nu$. With this common scaling the inner layer is located at $y^+ < 5$, the buffer region for $5 < y^+ < 30$ and the log-layer extends beyond the buffer region into the bulk. The steepest gradients of most relevant quantities, like U, k and ε, are found in the inner layer and the buffer region.

If we want to resolve the inner layer with only a few grid cells, the width of the cells needs to be in the order of the wall units ($\triangle y \approx \nu/U_{\tau}$) in this region. A uniform mesh will then require Re_{τ} cells to cover half the channel width, which is prohibitive except for very low Reynolds numbers. It is evident that a non-uniform mesh that is gradually squeezed in the wall normal direction as the wall is approached will be required. There are several alternative schemes that can be used to mesh a 1D boundary layer. Spectral methods often make use of the Chebyshev points located at $\cos(\pi k/N)$ for $k = 0, 1, \ldots, N$. The disadvantage here is that the shape is fixed and it is not possible to tune it if an even denser mesh near the wall is required. The successive ratio (SR) mesh keeps

DOI: 10.1002/fld

the ratio of successive cells constant and can easily be stretched or squeezed near obstacles, which gives a great deal of flexibility. For this reason the SR mesh has been used throughout this work, both for the FV mesh and for the element boundaries required by the spectral element method (see Section 4.3). The CV faces are then computed as

$$\begin{aligned} y(0) &= 0 \\ y(i) &= \frac{H}{\sum_{j=0}^{NJ-3}\gamma^j}\sum_{k=0}^{i-1}\gamma^k \quad \text{for } i=1,2,\ldots,NJ-2 \end{aligned} \tag{48}$$

where H is half the channel height, γ is the SR and there are $NJ-1$ CV faces and NJ nodes in total. The FV nodes are placed in the centre of the CVs for a cell-centred scheme.

The coarsest mesh used in this work contains $NJ=98$ nodes and is computed with γ ranging from 1.02 to 1.10. To study grid independence a gradually denser mesh is required. To this end we use an array of meshes with $NJ-2=96, 2\cdot 96, 4\cdot 96$ and $8\cdot 96$ CVs. The denser meshes modifies the SR as $\sqrt{\gamma}$, $\sqrt[4]{\gamma}$ and $\sqrt[8]{\gamma}$, respectively, which in each step is effectively to split the coarser mesh's CVs into two parts, using the new SR. In other words, if the first face of the coarse mesh is located at $\triangle y$, then the next mesh density has faces located at $\triangle y/(1+\sqrt{\gamma})$ and $\triangle y$, and every face location of the coarse mesh is effectively reused.

4.2. Validation tools

In a turbulent channel flow, the mean velocity is governed by the equation

$$\frac{\partial}{\partial y}\left((v+v_T)\frac{\partial U}{\partial y}\right)=-\beta \tag{49}$$

where $\beta=-\partial P/\partial x$. Since the pressure gradient is constant, Equation (49) can be integrated analytically from the wall to y, which gives the following exact result:

$$(v+v_T)\frac{\partial U}{\partial y}(y)=1-\beta y \tag{50}$$

where we have set $U_\tau=1$ and $H=0.5$. Equation (50) can be used to validate the numerical representation of the vital mean velocity gradient. A relative error in Equation (50) will then be computed as

$$\varepsilon_u=\left|\frac{(v+v_T)\dfrac{\partial U}{\partial y}-(1-\beta y)}{(1-\beta y)}\right| \tag{51}$$

and an integral measure of this error will be computed as

$$\overline{\varepsilon_u}=\sqrt{\int_0^{2H}\varepsilon_u^2(y)\,\mathrm{d}y} \tag{52}$$

For the spectral element solution the integral will be computed with the GLL quadrature rule applied to each element, whereas the FV method uses the midpoint rule.

In industry, CFD is often used to compute the drag or skin-friction for boundary layer flows. The design is often very restricted and a single percent reduction in drag can be considered a great improvement. Since the skin friction depends on the viscous stress $\nu\partial U/\partial y$ at the wall, it is essential that this term is captured accurately. However, due to the conservative property of the FV method, the term $\nu\partial U/\partial y$ will at the channel wall be exactly (to machine precision) equal to H regardless of numerical order or mesh distribution. Since this term is exact in the channel, the skin friction cannot be used directly to gauge the numerical accuracy of the FV method. The error computed with Equations (51) and (52) off the wall, though, will be representative of the interpolation error introduced when computing the mean velocity gradient.

The 1D channel is at equilibrium and for this reason the energy production across the channel needs to be balanced by the energy dissipation. This can be used to gauge numerical accuracy and an error estimate can then be found from

$$\varepsilon_p = \left| \frac{\int_0^{2H} P_k - \varepsilon \, \mathrm{d}y}{\int_0^{2H} P_k \, \mathrm{d}y} \right| \tag{53}$$

Owing to conservation, the error computed from Equation (53) will be exactly equal to $2\nu\partial k/\partial y|_{\text{wall}}$ for the FV method. As such, Equation (53) will only be used to assess the spectral element method.

4.3. Assessment of the spectral element method

Despite low numerical dissipation and low truncation error, the spectral element method is still just another approximation to the exact solution. The nonlinear coupling between the equations and the absence of analytical solutions for all parameters of the problem makes it difficult to gauge the numerical precision of the method with great confidence. Nevertheless, it is of importance to estimate the error before the spectral element solution can be put to good use.

The spectral element (see Section 3.2) reference solution has been implemented with 48 Gauss–Lobatto–Legendre elements of varying order. The location of the element interfaces have been taken from the SR FV mesh (see Equation (48)) using every $(NJ-2)/48$th node plus the node in the centre of the channel. This mesh has been used to ensure that for all FV mesh densities (see Section 4.1) the FV and reference solution share common nodes that can be used for direct comparison (no interpolations required).

Figure 3(a) and (b) shows the integrated error computed from Equation (52) and (53), respectively, using 48 elements (computed with $NJ=98$ and $\gamma=1.06$) and increasing the element order from 4 to 24. It is evident that the error is vanishing rapidly at higher orders as the spectral element solution approaches the analytical solution. It is interesting to note the spectral convergence that can be observed up to $p=8$ in (a) and for the complete range in (b). In (a) the error seems to stabilize around 10^{-8}, which most likely can be attributed to round-off errors. Note that in (a) we are comparing a derivative with an exact solution, whereas in (b) two computed integral quantities are compared. In both figures the error is thus also a reflection of the accuracy of the GLL quadrature scheme used in Equations (52) and (53).

Finally, it should be mentioned that the near-wall asymptotic behaviours of k and v^2, discussed in Section 2.2, are well captured with the SEM calculations. For v^2 this result should be evident from Figure 9. From the results shown in this section and the fact that the highest-order FV solution seems to converge to the SEM-solution with sufficient resolution (see Figure 8), it should be evident that the spectral element method can indeed give a very accurate representation of the v^2–f model. Note also that increasing the polynomial order to 36 makes no visible changes to

DOI: 10.1002/fld

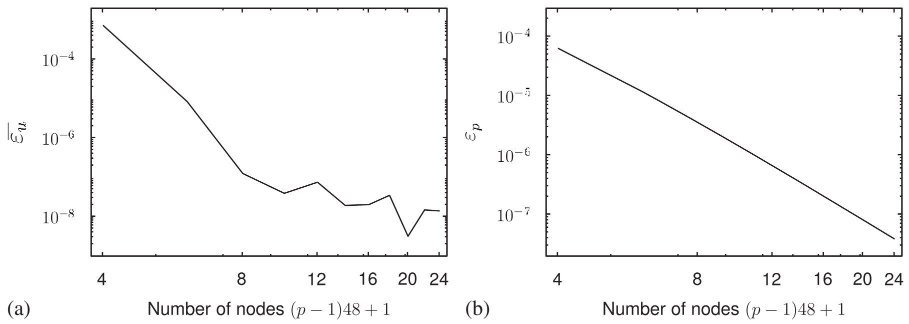

Figure 3. The error computed with Equation (52) (a) and Equation (53) (b) for a spectral element mesh of 48 elements and increasing polynomial order.

the plots made for comparing the SEM and the FV solutions in Section 4.4, which is another validation that the SEM solution has converged with sufficiently high accuracy. Confident that we have a good reference solution it is now time to validate the FV method.

4.4. Assessment of the FV method

The numerical schemes to be used in this validation section were discussed in Section 3.1 and will be referred to as:

$FV_A(N)$: Nth-order diffusion with regular source treatment.
$FV_B(N)$: Nth-order diffusion with extended source treatment.

$FV_A(2)$ is thus the regular cell-centred, second-order FV scheme outlined in Section 3.1. The regular source is computed using a midpoint rule, as given in Equation (30), whereas the extended source treatment uses the Gauss–Lobatto quadrature rule, presented in Equation (33).

The relationships (51) and (52) that were used to validate the spectral element method can also be used to validate the FV method. To this end we use a gradually denser mesh, as discussed in Section 4.1, with $\gamma = 1.04$ and 1.06. Figure 4 shows the errors computed with Equation (52), using schemes $FV_A(2)$ and $FV_A(4)$. Increasing the face interpolation order does not improve the result any further and the plots for higher orders will lie more or less on top of the dashed and dotted curves for $\gamma = 1.04$ and 1.06, respectively. This happens since the FV representation of the derivative is a two-point formula (see Equation (21)), regardless the number of points used for the face interpolation. It is interesting to note that the densest $FV_A(4)$ mesh uses approximately the same number of nodes as the spectral element solution of order 16, but achieves an error that is about three decades higher.

Another interesting observation that can be made from Figure 4 is that the integrated error seems to be slightly smaller for $\gamma = 1.04$, than for $\gamma = 1.06$. This is perhaps contrary to intuition, because the higher γ has better resolution of the inner layer, where the steepest gradients in turbulence parameters are found. To take a closer look at how this is possible, we now investigate the error in Equation (51) more locally. Figure 5 shows the relative error computed with Equation (51) for $FV_A(2)$ with $NJ = 98$ and SRs $\gamma = 1.06, 1.05, 1.04, 1.03$ and 1.02. For this range of γ values the location of the first node closest to the wall will range from $y^+ = 0.22, 0.47, 0.95, 1.9$ and 3.5, respectively. From Figure 5 we see that for a regular second-order FV scheme the two meshes that

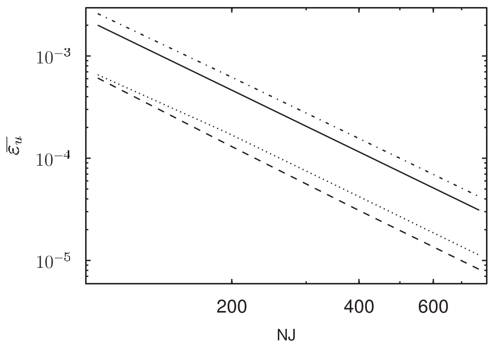

Figure 4. The integral error computed with Equation (52) at various mesh densities. The solid and dashed lines have been computed with $\gamma = 1.04$ for $FV_A(2)$ and $FV_A(4)$, respectively, whereas the dash-dotted and dotted lines represent $FV_A(2)$ and $FV_A(4)$ using $\gamma = 1.06$.

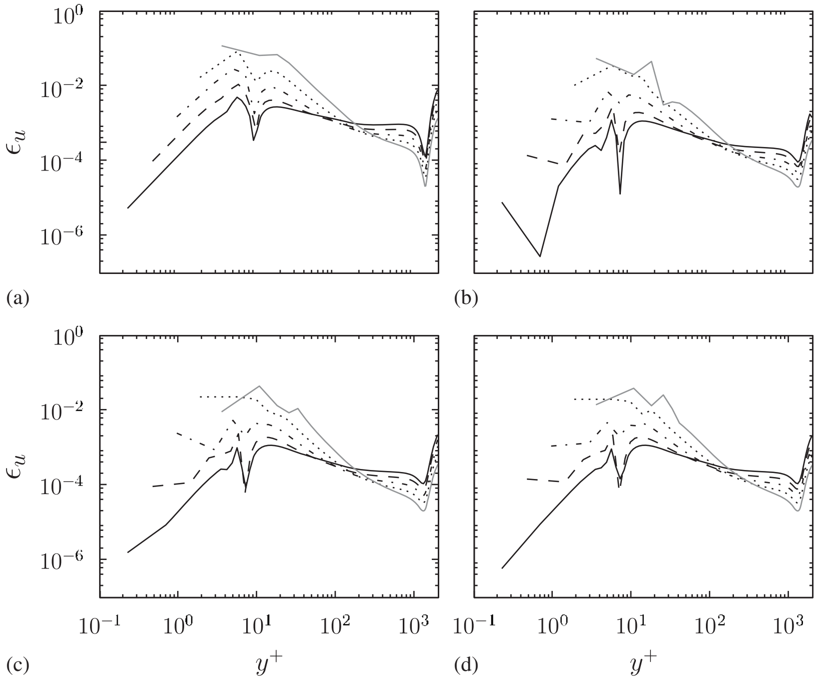

Figure 5. The absolute error computed with Equation (51) for $FV_A(N)$, using $N = 2$, 4, 6 and 8 for (a)–(d), respectively. The solid black, dashed, dash-dotted, dotted and solid grey lines represent meshes with successive ratios $\gamma = [1.06, 1.05, 1.04, 1.03, 1.02]$, respectively.

start with a y^+ higher than unity produce errors in the range of 1–10 % for the inner and buffer region. This clearly demonstrates the need for many grid points within the inner layer. A 1–10 % error in the shear stress is unacceptable by any standards. Not surprisingly the mesh with the smallest SR performs best in the centre of the channel. However, all meshes and schemes produce

DOI: 10.1002/fld

errors below 1% for this region. Looking at Figure 5 it is now possible to understand why $\gamma=1.04$ performed slightly better than $\gamma=1.06$ when the error was integrated across the channel. Even though $\gamma=1.06$ leads to better resolution of the inner layer, the integrated error is larger because $\gamma=1.04$ has greater mesh density and thus smaller errors for $y^+>100$ (outside the log-layer) that makes up for approximately 95% of the total channel width. It is thus important not to forget about mesh requirements for the bulk, even though the gradients here are less steep than near the wall.

It should be noted that the error defined through Equation (51) merely is an interpolation error for the mean velocity gradient, arising because the velocity gradients are naturally located on faces and not in nodes. In fact Equation (51) will be satisfied on faces to machine precision (plotting Figure 5 for face gradients produces errors of $\approx 10^{-14}$), since we are solving Equation (18) and the source is integrated exactly (linear). This can be put to good use and an 'exact' (since linear interpolation here is exact) cell averaged velocity gradient can be computed as

$$\left(\frac{\partial U}{\partial y}\right)_P=\frac{\left(\Gamma^u\dfrac{\partial U}{\partial y}\right)_e+\left(\Gamma^u\dfrac{\partial U}{\partial y}\right)_w}{2(\Gamma^u)_P} \tag{54}$$

This cell-averaged velocity gradient is, according to the test in Equation (51), accurate to machine precision. The result will of course only be exact in a turbulent parallel shear flows, where the linear relationship (50) holds true. However, it may be useful in other configurations as well. Note that even though the errors in computing the cell-averaged gradient of U are only interpolation errors, the term does affect the original solution indirectly through the production P_k, where the gradient is squared and the error amplified.

To more clearly understand what goes on inside the inner layer it is informative to look at the computed energy dissipation rate, which physically represents dissipation of turbulent kinetic energy on the smallest viscous scales. Since it represents the smallest scales, dissipation is the turbulence parameter that imposes the most severe restrictions on the resolution and it constitutes the parameter that changes most rapidly in the inner layer. Figure 6 shows the energy dissipation rate computed with $\mathrm{FV}_A(2)$ in (a) and $\mathrm{FV}_A(4)$ in (b) for a SR mesh of $NJ=98$ nodes and various γ

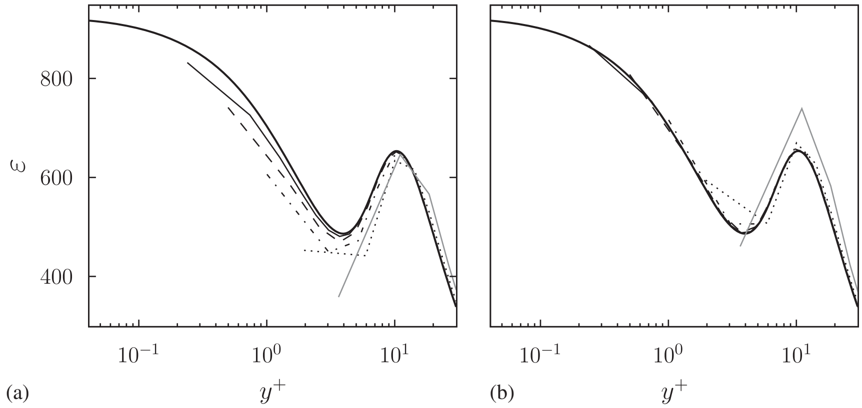

Figure 6. The computed energy dissipation rate for a regular $\mathrm{FV}_A(2)$ in (a) and $\mathrm{FV}_A(4)$ in (b). The reference solution is shown with a solid thick line, whereas the solid black, dashed, dash-dotted, dotted and solid grey lines represent various meshes with $NJ=98$ and $\gamma=[1.06, 1.05, 1.04, 1.03, 1.02]$, respectively.

values, compared with the reference solution. It is evident from this figure that the first node needs to be at least within $y^+ = 1$ and that the two meshes starting farthest from the wall simply do not have sufficient resolution to capture the rapid spatial variations.

Next we look at the mean velocity profile. Very close to a wall it is well known that there is a linear relationship between the velocity and the distance to the wall, i.e. $U^+ \approx y^+$. Since a linear velocity profile is exactly captured by even the lowest-order numerical schemes we expect no additional numerical issues. Still, the velocity profile is affected indirectly by parameters that are more difficult to resolve through the turbulent viscosity. It should perhaps be mentioned here that if the location of the first node is chosen very close to the wall ($y^+ \ll 1$), then the turbulent viscosity in the first nodes will be much smaller than the molecular viscosity and we could expect only minor errors to the velocity profile in this region. The relative error in the mean velocity profile for the coarsest mesh and scheme $\mathrm{FV}_B(N)$, with $N = 2$, 4, 6 and 8 is shown in Figure 7(a). Here we use the extended source treatment to show what can be achieved using higher-order numerics with the FV method. As expected the error near the wall is small. However, in the limit of higher orders it seems that it is not possible to reach the SEM solution and the error stabilizes around 10^{-4} for orders 6 and 8 and to further improve the result a denser mesh will be required. (For example, if we were to make the same plot as Figure 7(a) for $NJ = 2\cdot 96+2$ and $\gamma = \sqrt{1.06}$, then it is found that orders 6 and 8 will stabilize around 10^{-5}.) Figure 7(b) isolates the effect of increasing mesh density for the regular $\mathrm{FV}_A(2)$ scheme. It is noted that convergence is very slow and even for the densest mesh, using $\mathrm{NJ} = 8\cdot 96+2$, the average error is about 10^{-3}. It is apparent that only a combination of higher mesh density and face interpolation order can truly match the reference solution. To further study this combined effect Figure 8 shows the effect of increasing mesh density for both the highest-order FV schemes, $\mathrm{FV}_B(8)$ in (a) and $\mathrm{FV}_A(8)$ in (b) (note how Figure 8(a) is a natural continuation of Figure 7(a)). Figure 8 clearly illustrates the importance of improving the cell-averaged source as the error is consistently at least one order of magnitude better than for the FV scheme that integrates using the midpoint rule. It is also evident

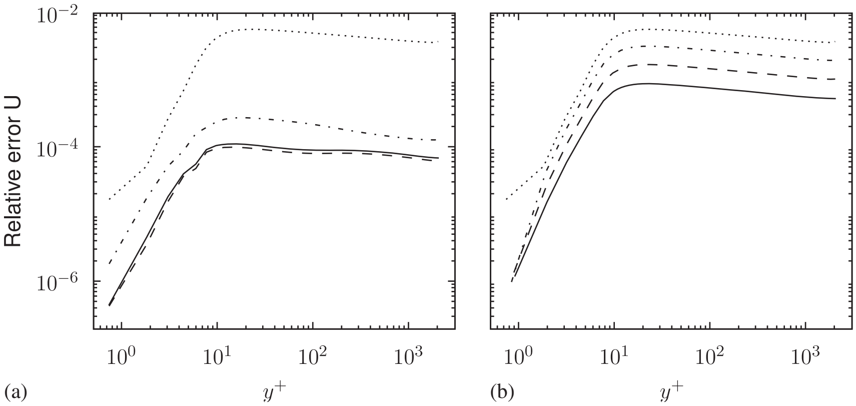

Figure 7. The relative error in the velocity profile using $\mathrm{NJ} = 98$ and $\gamma = 1.06$, with $\mathrm{FV}_B(N)$ is shown in (a), where the solid, dashed, dash-dotted and dotted lines represent $N = 8$, 6, 4 and 2, respectively. (b) is computed using $\mathrm{FV}_A(2)$, $\gamma = 1.06$ and a gradually increased mesh density. The solid, dashed, dash-dotted and dotted lines represent $NJ - 2 = 8\cdot 96$, $4\cdot 96$, $2\cdot 96$ and 96, respectively.

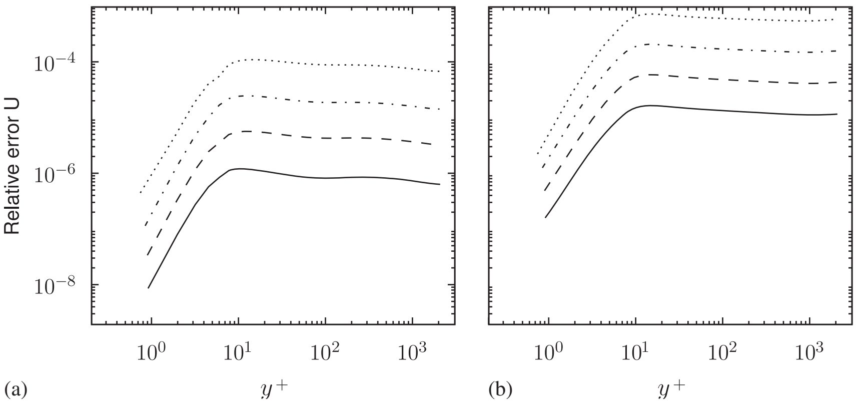

Figure 8. The relative error in the velocity profile for the $\mathrm{FV}_B(8)$ in (a) and $\mathrm{FV}_A(8)$ in (b) using $\gamma = 1.06$ and increasing mesh density. The solid, dashed, dash-dotted and dotted lines have been computed with $\mathrm{NJ}-2 = 8\cdot 96$, $4\cdot 96$, $2\cdot 96$ and 96, respectively.

from Figure 8(a) that the FV solution at high order and mesh density converges towards the SEM solution, which strongly supports the claims made about the accuracy of the reference solution.

To further investigate some numerical issues relating to the v^2–f model, we will now take a closer look at the v^2 profile as the wall is approached. The v^2 profile should follow an asymptotic y^4 curve close to the wall, which is problematic for second-order numerical schemes. Figure 9 shows the results obtained for v^2/y^4 using $\mathrm{NJ} = 98$ and $\gamma = [1.10, 1.08, 1.06, 1.04, 1.02]$. Profiles (1)–(3) represent the $\mathrm{FV}_A(N)$, with $N = 2$, 4 and 6, whereas profiles (4)–(6) represent $\mathrm{FV}_B(N)$, with $N = 2$, 4 and 6, respectively. Evidently, for a regular second-order FV scheme the error seems to be significant for the three nodes closest to the wall, regardless of how close the first node is. The magnitude of the error is alarming, but not surprising. As expected a second-order scheme cannot accurately represent a fourth-order polynomial asymptote and for a second-order scheme the only solution will be to ‘overresolve’ the wall using a very small y^+ for the first internal node. The magnitude of v^2 (and thus the error) in the first internal node will then be small and the turbulent viscosity will be much smaller than the molecular viscosity. As such any numerically induced error in v^2 will not be transferred to the other equations that are coupled to v^2 through the turbulent viscosity.

From Figure 9 it is noteworthy that a fourth-order face interpolation scheme cannot suppress all oscillations in the asymptotic v^2 profile. In theory a fourth-order scheme should be perfect for a y^4 profile. However, in reality this will only be true for a uniform mesh using finite differences, and not the SR FV mesh used here.[‡]

Finally, and to further study the resolution requirements in v^2, Figure 10(a), (b) and (c) shows the relative error in v^2 computed with $\mathrm{FV}_A(2)$, $\mathrm{FV}_A(8)$ and $\mathrm{FV}_B(8)$ respectively, compared with the reference solution. The FV solution is obtained with a gradually increased mesh density using $\gamma = 1.06$. It is evident from (a) that for a second-order scheme the error cannot be eliminated near the wall simply by increasing the mesh density and a higher-order scheme is required. Naturally,

[‡] Even for uniform CVs the first internal node will be three times closer to the wall than the next internal node and the nodes will never be uniformly distributed in a cell-centred FV mesh.

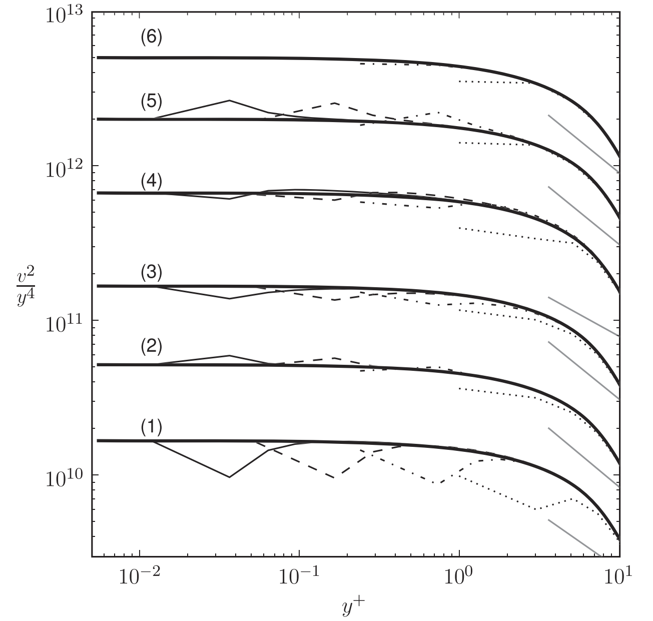

Figure 9. The scalar v^2 scaled by y^4 computed with the FV method using various orders and compared to the reference solution (solid thick line). The profiles (1)–(3) use method $FV_A(N)$, with $N=2$, 4 and 6, respectively, whereas profiles (4)–(6) use method $FV_B(N)$, with $N=2$, 4 and 6. The solid black, dashed, dash-dotted, dotted and solid grey lines represent various meshes with $\gamma=[1.10, 1.08, 1.06, 1.04, 1.02]$, respectively, and for all meshes $NJ=98$. Note that for illustrative purposes the profiles (2)–(6) have been shifted in the vertical direction.

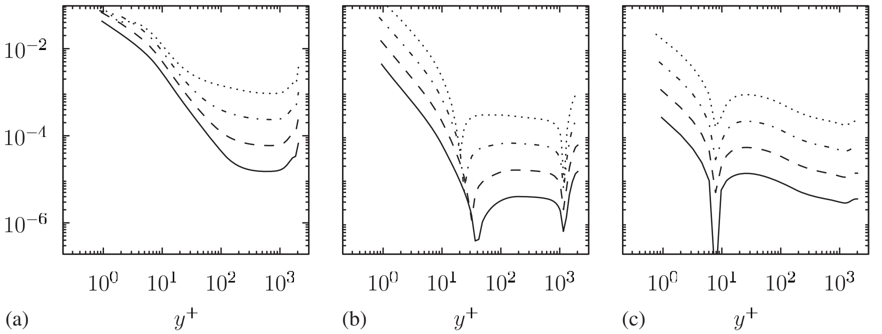

Figure 10. The relative error in v^2 for a regular second-order $FV_A(2)$ in (a) and $FV_A(8)$ and $FV_B(8)$ in (b) and (c), respectively. The solid, dashed, dash-dotted and dotted lines have been computed with $\mathrm{NJ}-2=8\cdot96$, $4\cdot96$, $2\cdot96$ and 96, respectively.

the error in the centre of the channel does reduce with increasing mesh density, even for a second-order scheme. From (b) and (c) it is seen that the error in the v^2 profile can be reduced near the wall if Gauss–Lobatto quadrature is used for the source integration. As expected from the discussion around Equations (31) and (32), the midpoint integration leads to errors that are at least one order of magnitude higher than Gauss–Lobatto quadrature near the wall.

5. CONCLUSIONS

The objective of this work has been to assess the finite volume (FV) method applied to the original v^2–f turbulence model of Durbin. We have tested the FV method in a statistically one-dimensional, wall-bounded, turbulent channel flow, using various mesh densities and discretizations. To compute a higher-order FV solution, a Lagrange face interpolation scheme up to arbitrary order has been described. For validation, the FV results have been compared with a highly accurate spectral element reference solution.

From this work it is apparent that a regular second-order FV scheme cannot fully cope with the steep gradients observed in turbulence parameters near a solid wall. Even for a very dense mesh, where the first grid cell is well within $y^+ = 1$, the error in a second-order scheme is still considerable and cannot be completely eliminated. To match the spectral element reference solution it was found that an extended source integration, using here three point Gauss–Lobatto quadrature, as well as a higher-order diffusion scheme was required. Only with improved source treatment could the FV method give a truly precise representation of all model parameters near a solid wall. It is evident that the location of the first grid-cell needs to be well within $y^+ = 1$. Still, this is a necessary, but not sufficient condition for an accurate near-wall treatment.

A conclusion that can be drawn from this work is that one needs to be careful when interpreting results from simulations of the v^2–f model. It is important that bad simulation results are not falsely attributed to the model, when the real reason is a mismatch between model and numerical scheme—a topic that is easily overlooked.

ACKNOWLEDGEMENTS

This work has been performed under the WALLTURB project. WALLTURB (A European synergy for the assessment of wall turbulence) is funded by the EC under the 6th framework program (Contract No: AST4-CT-2005-516008). The authors also acknowledge support from the Norwegian Research Council through the Norwegian Centre of Excellence: Center for Biomedical Computing.

REFERENCES

1. Durbin PA. Near-wall turbulence closure modeling without ‘damping’ functions. *Theoretical Computational Fluid Dynamics* 1991; **3**:1–13.
2. Manceau R, Wang M, Durbin PA. Assessment of non-local effect on pressure term in RANS modeling using a DNS database. *Proceedings of the Summer Program*, Center for Turbulence Research, 1998; 303–322.
3. Durbin PA, Pettersson Reif BA. *The Elliptic Relaxation Method*. Cambridge University Press: Cambridge, 2002; 127–152.
4. Parneix H, Durbin PA, Behnia M. Computation of 3-d turbulent boundary layers using the v^2–f model. *Flow, Turbulence and Combustion* 1998; **60**:19–46.
5. Parneix H, Behnia M, Durbin PA. Predictions of turbulent heat transfer in an axisymmetric jet impinging on a heated pedestal. *Journal of Heat Transfer* 1999; **121**:43–49.
6. Sveningsson A, Pettersson Reif BA, Davidson L. Modelling the entrance region in a plane asymmetric diffuser by elliptic relaxation. *The Fourth International Symposium on Turbulence and Shear Flow Phenomena*, Williamsburg, Virginia, 2005; 1183–1188.

DOI: 10.1002/fld

7. Lien FS, Kalitzin G. Computations of transonic flow with the v^2–f turbulence model. *International Journal of Heat and Fluid Flow* 2001; **22**:53–61.
8. Laurence DR, Uribe JC, Utyuzhnikov SV. A robust formulation of the v^2–f model. *Flow*, *Turbulence and Combustion* 2004; **73**:169–185.
9. Hanjalić K, Popovac M, Hadžiabdić. A robust near-wall elliptic-relaxation eddy-viscosity turbulence model for CFD. *International Journal of Heat and Fluid Flow* 2004; **25**:1047–1051.
10. Billard F, Uribe JC, Laurence D. A new formulation of the v^2–f model using elliptic blending and its application to heat transfer prediction. *Proceedings of the 7th International ERCOFTAC Symposium on Engineering Turbulence Modelling and Measurements*, Limasol, Cyprus, vol. 1, 2008; 89–94.
11. Patera AT. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of Computational Physics* 1984; **54**:468–488.
12. Durbin PA. A Reynolds stress model for near-wall turbulence. *Journal of Fluid Mechanics* 1993; **249**:465–498.
13. Lien FS, Durbin PA. Non-linear k–e–v2 modelling with application to high lift. *Proceeding of the Summer Program*, Stanford University, 1996; 5.
14. Svenningsson A, Davidson L. Assessment of realizability constraints in v^2–f turbulence models. *International Journal of Heat and Fluid Flow* 2004; **25**(5):785–794.
15. Maximum. http://functions.wolfram.com/01.34.09.0001.01.
16. Minimum. http://functions.wolfram.com/01.35.09.0001.01.
17. Floor. http://functions.wolfram.com/04.01.02.0001.01.
18. Berrut JP, Trefethen LN. Barycentric lagrange interpolation. *SIAM Review* 2004; **46**(3):501–517.
19. Canuto C, Hussaini MY, Quarteroni A, Zang TA. *Spectral Methods in Fluid Mechanics* (1st edn). Springer Series in Computational Physics. Springer: New York, 1988.

DOI: 10.1002/fld