European Journal of Computational Mechanics
Vol. 21, Nos. 1–2, February–April 2012, 22–39

Hierarchical elements for the iterative solving of turbulent flow problems on anisotropic meshes

B.A. Wane[a]*, J.M. Urquiza[a], A. Fortin[a] and D. Pelletier[b]

[a]*Département de mathématiques et de statistique, Université Laval, GIREF, Pavillon Vachon, Québec, G1V 0A6, Canada;* [b]*Département de Génie Mécanique, École Polytechnique de Montréal, Montréal, (QC) H3C 3A7, Canada*

Accurate solution of industrial turbulent flow problems requires very fine meshes resulting in large systems of non-linear equations and huge computational costs. Efficient iterative methods are therefore necessary. Mesh adaptation, and in particular anisotropic mesh adaptation, allows to reduce considerably meshes size while preserving the accuracy of the solution. Unfortunately, iterative methods and anisotropic meshes do not come along easily and convergence problems may occur. In this work, we show how quadratic elements, expressed in a hierarchical basis, can be used to develop efficient iterative methods for the numerical simulation of turbulent flows on strongly anisotropic meshes.

La résolution numérique d'écoulements turbulents résulte en des systèmes d'équations non linéaires de très grande taille et en des coûts de calcul importants, voire prohibitifs. L'utilisation de méthodes itératives est alors nécessaire. L'adaptation de maillage permet de réduire considérablement la taille des maillages et encore plus dans le cas d'adaptation anisotrope où l'on retrouve des éléments de rapport d'étirement très grand. Malheureusement, maillages anisotropes et méthodes itératives ne font pas toujours bon ménage. Dans cet article, nous montrons comment l'utilisation d'éléments quadratiques, exprimés dans une base hiérarchique, permet le développement de méthodes itératives extrêmement efficaces pouvant servir pour la simulation précise d'écoulements turbulents sur des maillages anisotropes.

Keywords: turbulent flows; hierarchical elements; iterative solvers; $k - \epsilon$ turbulence model; logarithmic formulation; anisotropic mesh adaptation

Mots-clés: écoulements turbulents; éléments hiérarchiques; solveurs itératifs; modèle $k - \epsilon$; formulation logarithmique; maillages anisotropes

1. Introduction

Turbulence plays an important role in many mechanical and chemical engineering processes (fluid flow, mass and heat transfer) which are dominated by convective transport. Accurate solution of industrial turbulent flow problems generally requires very fine meshes resulting in huge systems of non-linear equations and thus to large computational costs. Direct solvers are no longer usable, particularly for three-dimensional problems. Iterative methods become mandatory but their convergence is delicate, especially for saddle-point problems such as those

*Corresponding author. Email: bwane@giref.ulaval.ca

ISSN 1779-7179 print/ISSN 1958-5829 online

http://dx.doi.org/10.1080/17797179.2012.702428
http://www.tandfonline.com

encountered in velocity-pressure formulations of the Navier–Stokes equations at large Reynolds numbers. Convergence problems worsen when using higher order finite element discretisations.

A now popular way to reduce computational costs, while preserving the accuracy of the solution, is mesh adaptation which is widely used for computational fluid dynamics (CFD) problems. In most cases, however, isotropic meshes are used. This means that only the size of the elements is controlled but not their elongation. Elements with aspect ratio close to 1 (ratio of the longest to the smallest element lengths) are therefore produced. This is a first important step but it is possible to do better.

Anisotropic mesh adaptation allows to further reduce mesh size by stretching elements in some favourable directions. Elements with large aspect ratios may thus appear in the mesh wherever the solution allows it. In finite element textbooks, elements with large aspect ratio are often described as ill shaped and leading to inaccurate solutions. Moreover, it is well known that these elements have a negative effect on the conditioning of the resulting linear systems. This is why they are seldom used in conjunction with iterative methods.

In this work, we show how expressing higher order classical Lagrange finite element discretisations in a hierarchical basis can be exploited to develop very efficient and robust iterative methods that can be used for the accurate solution of turbulent flows on anisotropic meshes. A classical $k - \epsilon$ model in logarithmic formulation is used for comparison purposes only and can be applied to other Reynolds averaged Navier–Stokes (RANS) equations or large eddy simulations.

This paper is organised as follows. In Section 2, we present the RANS equations and the standard $k - \epsilon$ turbulence model. We also recall the wall boundary conditions which are appropriate to this modelling. We then present the change of variables leading to a logarithmic form of the turbulence equations. In Sections 3 and 4, we discuss the hierarchical finite element formulation and the corresponding iterative methods. In Section 5, we describe the anisotropic mesh adaptation procedure. Finally, we present in Section 6 some numerical results.

2. Turbulent flow equations

2.1. Turbulence model in logarithmic form

We consider the time-averaged continuity and momentum equations for an incompressible fluid in turbulent regime using Boussinesq's hypothesis:

$$\nabla \cdot u = 0, \tag{1}$$

$$\rho u \cdot \nabla u = -\nabla p + \nabla \cdot [2(\mu + \mu_T)D(u)], \tag{2}$$

where ρ is the density, u *is* the velocity, p is the pressure, μ is the viscosity, μ_T is the eddy viscosity of the fluid and $D(u) = (\nabla u + \nabla_u^T)/2$ is the rate of deformation tensor.

We close this system of equations with the $k - \epsilon$ turbulence model of Launder and Spalding (1972). The eddy viscosity is expressed in terms of two turbulence variables, the turbulence kinetic energy k and its rate of dissipation ϵ:

$$\mu_T = \rho C_\mu \frac{k^2}{\epsilon}. \tag{3}$$

The turbulence quantities are governed by the following transport equations:

$$\rho u \cdot \nabla k = \nabla \cdot \left[\left(\mu + \frac{\mu_T}{\sigma_k} \right) \nabla k \right] + \mu_T P(u) - \rho \epsilon, \tag{4}$$

$$\rho u \cdot \nabla \epsilon = \nabla \cdot \left[\left(\mu + \frac{\mu_T}{\sigma_\epsilon} \right) \nabla \epsilon \right] + C_1 \frac{\epsilon}{k} \mu_T P(u) - C_2 \rho \frac{\epsilon^2}{k}, \tag{5}$$

where $P(u) = 2D(u) : D(u)$ is the production of turbulence. As recommended in Launder and Spalding (1972), we set $C_1 = 1.44$, $C_2 = 1.92$, $C_\mu = 0.09$, $\sigma_k = 1.0$ and $\sigma_\epsilon = 1.3$.

It is an understatement to say that the turbulence Equations (4)–(5) may cause some numerical difficulties. One of these is that as the eddy viscosity depends on ϵ and k, it may become negative if ϵ becomes negative, causing the solver to breakdown. Moreover, several source terms contain division by a turbulence variable. With negative or small values, the denominator can cause improper sign or overly large values of the source terms. Enhanced robustness of the algorithm is achieved if one can ensure that turbulence variables remain positive throughout the domain and throughout the course of iterations. Several types of strategies have been developed to this aim (see for instance Kuzmin & Mierka, 2006; Lew, Buscaglia, & Carrica, 2001).

One way to preserve positivity of the dependent variables consists of solving the equations for their logarithms as in Ilinca and Pelletier (1998). This is done by using the change of dependent variables $\mathcal{K} = \ln(k)$ and $\mathcal{E} = \ln(\epsilon)$. The following transport equations for the logarithmic variables are easily deduced from (4) and (5):

$$\rho u \cdot \nabla \mathcal{K} = \nabla \cdot \left[\left(\mu + \frac{\mu_T}{\sigma_k} \right) \nabla \mathcal{K} \right] + \left(\mu + \frac{\mu_T}{\sigma_k} \right) |\nabla \mathcal{K}|^2 + F^1_{\mathcal{K}} P(u) - F^2_{\mathcal{K}},$$

$$\rho u \cdot \nabla \mathcal{E} = \nabla \cdot \left[\left(\mu + \frac{\mu_T}{\sigma_\epsilon} \right) \nabla \mathcal{E} \right] + \left(\mu + \frac{\mu_T}{\sigma_\epsilon} \right) |\nabla \mathcal{E}|^2 + F^1_{\mathcal{E}} P(u) - F^2_{\mathcal{E}},$$

where the source terms are defined by

$$F^1_{\mathcal{K}} = \mu_T e^{-\mathcal{K}}, \; F^2_{\mathcal{K}} = -\rho^2 C_\mu \frac{e^{\mathcal{K}}}{\mu_T}, \; F^1_{\mathcal{E}} = \rho C_1 C_\mu e^{\mathcal{E}-\mathcal{K}}, \; F^2_{\mathcal{E}} = -\rho C_2 e^{\mathcal{E}-\mathcal{K}}.$$

Solving for $\mathcal{K}$ and $\mathcal{E}$ guarantees that $k = e^{\mathcal{K}}$ and $\epsilon = e^{\mathcal{E}}$, as well as the eddy viscosity μ_T will remain positive throughout the computations. Depending on the applications, Dirichlet or free boundary conditions are imposed for u, $\mathcal{K}$ and $\mathcal{E}$ at inflow, outflow or fictitious boundaries. For solid boundaries, we apply the wall boundary conditions described below.

2.2. Wall boundary conditions

The standard $k - \epsilon$ turbulence model is not valid for low values of the turbulent Reynolds number defined as $Re_T = \frac{1}{\mu_T}$. This is often the case in the vicinity of solid walls. The strategy adopted here uses wall functions which describe the solution in the near wall region. In this work, we use the *two-velocity scale wall law* adopted in Chabard (1991) and Lacasse (2004).

The wall law is hereafter evaluated at a distance y from the wall Γ_w in the normal direction, so that the new computational domain is obtained by removing a layer of thickness y from the real domain Ω at the vicinity of Γ_w.

At the new wall, the velocity satisfies the no-penetration boundary condition $u \cdot n = 0$, where n is the unit outward normal vector. It is assumed that k satisfies a free boundary condition at the wall, whereas ϵ satisfies a Dirichlet-type boundary condition:

$$\frac{\partial k}{\partial n} = 0, \qquad \epsilon = \frac{u_\star^3}{\kappa y}, \tag{6}$$

where $u_\star$ is defined by $u_\star = \sqrt{C_\mu^{\frac{1}{2}} k}$. Using the new dimensionless variable $y^+ = y\rho u_\star/\mu$, the two-velocity scale wall law is completed by prescribing the wall shear stress $\boldsymbol{\tau}_w \equiv \boldsymbol{\sigma} \cdot n - (n \cdot \boldsymbol{\sigma} \cdot n)n$, where $\boldsymbol{\sigma} = -pI + 2(\mu + \mu_T)D(u)$ is the Cauchy stress tensor. The wall law is expressed in the form:

$$\boldsymbol{\tau}_w = -\gamma(k)\mathbf{u}, \tag{7}$$

$$\text{where } \gamma(k) = \begin{cases} \frac{\mu}{y}, & y^+ < y_c^+, \\ \frac{\rho u_\star}{\frac{1}{\kappa}\ln(Ey^+)}, & y_c^+ < y^+ < 300 \end{cases}$$

and $\kappa = 0.42$, $E = 9.0$ (Schetz, 1993) and y_c^+ is the value ensuring the continuity of γ that is $y_c^+ = 11.63$. We notice that for a fixed value of k, this is a linear boundary condition of Robin type, much easier to handle than the more classical wall laws. Note also that these boundary conditions can easily be rewritten in terms of the new logarithmic variables as done in the next section with the finite element approximations.

3. Numerical discretisation

In this section, we briefly present the numerical discretisation of the system of equations presented in the previous section. It is based on a finite element approximation using the Taylor–Hood (P_2–P_1) element for velocity and pressure variables and continuous piecewise quadratic polynomials (P_2) for the logarithmic turbulent variables. Finally, the resulting non-linear algebraic system is solved with a fixed point algorithm. It involves the resolution of nonlinear Navier–Stokes and non-linear transport equations solved in a precise sequence described below.

3.1. Finite element approximation

Let $\Omega \subset \mathbb{R}$, m=2 or 3, be the domain occupied by the fluid. As previously described, Γ denotes its boundary and $\Gamma_w \subset \Gamma$ represents a solid wall. For simplicity, in this section, we assume that free boundary conditions are appended in the remaining part of the boundary.

Let $\mathcal{T}_h$ be a triangulation of Ω. If Ω is non-polygonal, $\mathcal{T}_h$ is more precisely a triangulation of a polygonal approximation Ω_h of Ω that has all its boundary vertices in Γ. Let Γ_h be the boundary of Ω_h and let Γ_{wh} be the sub-part corresponding to Γ_w. We denote by n_h the outward unitary normal vector to Γ_h.

For a triangular element $T_k \in \mathcal{T}$, let $P_n(T_k)$ denote the space of polynomials of degree n on T_k. We now define, for $n = 1, 2$:

$$\begin{aligned} W_h^n &= \{q_h \in C^0(\Omega), q_h|_{T_k} \in P_n(T_k), \forall T_k \in \mathcal{T}_h\}, V_h = (W_h^2)^m, \\ W_{0h}^n &= \{q_h \in W_h^n; q_h = 0 \text{ on } \Gamma_w\},\ \ V_{0h} = \{v \in \ V_h; v \cdot n_h = 0 \text{ on } \Gamma_w\}, \end{aligned}$$

and denote by I_h the quadratic interpolation operator associated to W_h^2. The finite element approximation of the continuous problem then reads:

Find $(u_h, p_h, \mathcal{K}_h, \mathcal{E}_h) \in V_{0h} \times W_h^1 \times W_h^2 \times W_h^2$,
such that for all $(v_h, q_h, \psi_{\mathcal{K}_h}, \psi_{\mathcal{E}_h}) \in V_{0h} \times W_h^1 \times W_h^2 \times W_{0h}^2$:

Conservation of momentum:

$$\begin{aligned} &\int_{\Omega_h} (\rho \boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h)\ \mathbf{v}_h d\Omega + \int_{\Omega_h} 2(\mu + \mu_T) \boldsymbol{D}(u_h) : \boldsymbol{D}(\mathbf{v}_h)\ d\Omega + r \int_{\Omega} (\nabla \cdot u_h)(\nabla \cdot \boldsymbol{v}_h)\ d\Omega \\ &\quad - \int_{\Omega_h} p_h \nabla \cdot \boldsymbol{v}_h d\Omega = \int_{\Gamma_{wh}} \gamma(e^{\mathcal{K}_h}) \boldsymbol{u}_h \cdot \boldsymbol{v}_h d\Gamma. \end{aligned} \tag{8}$$

where the Robin-type wall law (7) was applied. The supplementary term involving the (small) parameter r is a discretisation of the term $r\nabla(\nabla \cdot u)$ which is supposed to vanish, thus the formulation is still consistent. Moreover, since r is chosen small and $\nabla \cdot u_h \simeq 0$, it has negligible effect on the numerical solution but enhances drastically the convergence properties of the iterative solver described in Section 4.2.

Conservation of mass:

$$\int_{\Omega} \nabla \cdot u_h\, q_h\, d\Omega = 0. \tag{9}$$

Transport of turbulent variables:
It is well known that solving convection–diffusion equations such as those governing $\mathcal{K}$ and $\mathcal{E}$ is difficult and that the standard Galerkin method is inadequate. Therefore, a Streamline-Upwing-Petrov-Galerkin method introduced by Hughes and Brooks (1979, 1982) was used. The weighting functions denoted as, respectively, $\psi_{\mathcal{K}_h}$ and $\psi_{\mathcal{E}_h}$ are then replaced by:

$$\hat{\psi}_{\mathcal{K}_h} = (\psi_{\mathcal{K}_h} + \tau \frac{u}{\| u \|} \cdot \nabla \psi_{\mathcal{K}_h}) \quad \text{and} \quad \hat{\psi}_{\mathcal{E}_h} = (\psi_{\mathcal{E}_h} + \tau \frac{u}{\| u \|} \cdot \nabla \psi_{\mathcal{E}_h})$$

where the parameter τ depends essentially on the size of the element. Multiplying by these tests functions, one easily obtains the following consistent variational formulations:

$$\begin{aligned} &\int_{\Omega_h} (\rho u_h \cdot \nabla \mathcal{K}_h) \hat{\psi}_{\mathcal{K}_h} d\Omega = \int_{\Omega_h} \left(\mu + \frac{\mu_T}{\sigma_k}\right) (\nabla \mathcal{K}_h \cdot \nabla \mathcal{K}_h) \hat{\psi}_{\mathcal{K}_h} d\Omega \\ &\quad + \int_{\Omega_h} \nabla \cdot \left(\left(\mu + \frac{\mu_T}{\sigma_k}\right) \nabla \mathcal{K}_h \right) \left(\tau \frac{\boldsymbol{u}}{\| \boldsymbol{u} \|} \cdot \nabla \psi_{\mathcal{K}_h} \right) d\Omega \\ &\quad - \int_{\Omega_h} \left(\mu + \frac{\mu_T}{\sigma_k}\right) \nabla \mathcal{K}_h \cdot \nabla \psi_{\mathcal{K}_h} d\Omega + \int_{\Omega_h} (F_{\mathcal{K}}^1 P(\boldsymbol{u}) + F_{\mathcal{K}}^2)\ \hat{\psi}_{\mathcal{K}_h}\ d\Omega, \end{aligned} \tag{10}$$

and similarly for the equation for $\mathcal{E}_h$, we find:

$$\begin{aligned}\int_{\Omega_h}(\rho \boldsymbol{u}_h\cdot\nabla\mathcal{E}_h)\hat{\psi}_{\mathcal{E}_h}d\Omega &= \int_{\Omega_h}\left(\mu+\frac{\mu_T}{\sigma_\epsilon}\right)(\nabla\mathcal{E}_h\cdot\nabla\mathcal{E}_h)\hat{\psi}_{\mathcal{E}_h}d\Omega\\ &+\int_{\Omega_h}\nabla\cdot\left(\left(\mu+\frac{\mu_T}{\sigma_k}\right)\nabla\mathcal{E}_h\right)\left(\tau\frac{\boldsymbol{u}}{\parallel\boldsymbol{u}\parallel}\cdot\nabla\psi_{\mathcal{E}_h}\right)d\Omega\\ &-\int_{\Omega_h}\left(\mu+\frac{\mu_T}{\sigma_\epsilon}\right)\nabla\mathcal{E}_h\cdot\nabla\psi_{\mathcal{E}_h}d\Omega_h+\int_{\Omega_h}(F^1_{\mathcal{E}}P(u)+F^2_{\mathcal{E}})\hat{\psi}_{\mathcal{E}_h}d\Omega,\end{aligned}\tag{11}$$

$$\text{with } \mathcal{E}_h = I_h\left(\log\left(\frac{C_\mu^{3/4}}{\kappa y}\right)+\frac{3\mathcal{K}_h}{2}\right)\quad\text{on}\quad\Gamma_{wh},\tag{12}$$

$$\text{and } \mu_T=\rho\,C_\mu\,e^{2\mathcal{K}_h-\mathcal{E}_h}.\tag{13}$$

3.2. *Fixed point algorithm*

The global non-linear system is solved by the following algorithm:

(1) Initialie variables u, p, $\mathcal{K}$ and $\mathcal{E}$.
(2) Calculate the initial distribution of μ_T using (13).
(3) Until convergence of all variables:

(a) Solve Navier–Stokes Equations (8) and (9) for u and p.
(b) Using u calculated at the previous step:

(i) Solve Equation (10) for $\mathcal{K}$ and update $\mathcal{K}$.
(ii) Solve Equations (11)–(12) for $\mathcal{E}$ and update $\mathcal{E}$.
(iii) Update μ_T using (13).
(iv) Return to Step (3b) if the stopping criterion is not satisfied. Otherwise go to Step (3c).

(a) Return to Step (3a) if the stopping criterion of global convergence is not satisfied. Otherwise, stop the calculations.

Newton's method is used for Equations (10) and (11) and for the RANS Equations (8) and (9). In all cases, the resulting linear systems are solved by the iterative methods described in the following section.

4. Iterative methods based on hierarchical elements

In this section, we introduce two types of iterative methods, the first one for scalar equations such as (10) and (11) and the second one for saddle-point problems such as (8) and (9). The proposed methods were introduced, respectively, in El maliki, Guénette, and Fortin (2011) for scalar equations and in El maliki and Guénette (2010) for the Navier–Stokes equations. In these papers, the proposed iterative methods were tested on a number of rather academic problems and on more or less uniform meshes at low Reynolds numbers. One of our objectives is to modify these methods and test their performance on more realistic applications at high Reynolds numbers and on general anisotropic meshes.

As described in Section 3.1, we are using quadratic (second-order accurate) finite element discretiations and therefore, the different iterative methods must be chosen carefully. All quadratic variables ($u_h, \mathcal{K}_h$ and $\mathcal{E}_h$) are expressed using a hierarchical basis typically on triangles and tetrahedra. The idea is very simple and can be generalised to higher order discretisations as described in Zaki (1993) and Pelletier, Zaki, and Fortin (1994). The hierarchical basis can be described in terms of the barycentric coordinates $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ for a tetrahedral element. The first four (linear) basis functions are given by $\phi_i = \lambda_i$ for $1 \leq i \leq 4$ and are associated to the four vertices of the tetrahedra. The six remaining (quadratic) basis functions associated to midside nodes are:

$$\phi_5 = 4\lambda_1\lambda_2,\ \phi_6 = 4\lambda_2\lambda_3,\ \phi_7 = 4\lambda_3\lambda_4,\ \phi_8 = 4\lambda_1\lambda_3,\ \phi_9 = 4\lambda_1\lambda_4,\ \phi_{10} = 4\lambda_2\lambda_4.$$

On each element, any quadratic finite element field u_h (not necessarily the velocity) can thus be decomposed into a linear part u_l and a quadratic correction u_q, i.e. $u_h = u_l + u_q$, where:

$$u_l = \sum_{i=1}^{4} u_i\phi_i, \qquad u_q = \sum_{i=5}^{10} u_i\phi_i.$$

In other words, the finite element approximation space can be written as $V_h = V_l \oplus V_q$, where V_l is the subspace of continuous piecewise linear polynomials and V_q is the complementary subspace of continuous piecewise quadratic midside shape functions.

Using this decomposition, any linear system $Au_h = b$ can be written in the form:

$$\begin{bmatrix} A_{ll} & A_{lq} \\ A_{ql} & A_{qq} \end{bmatrix} \begin{bmatrix} u_l \\ u_q \end{bmatrix} = \begin{bmatrix} b_l \\ b_q \end{bmatrix}$$

The matrix A_{ll} is exactly the same as one would obtain using a linear discretisation and its size is much smaller than the global matrix (about four times smaller in 2D and seven times in 3D). Moreover, it is shown in Verfürth (1996) that for elliptic problems, the condition number of the matrix A_{qq} is $O(1)$. This means that its condition number does not vary with the size of the matrix. As we will see, it also means that a second-order accurate solution can be obtained at a very small extra cost with respect to a first-order approximation.

4.1. Solving scalar equations

The computation of the turbulent variables and ϵ (or of $\mathcal{K}$ and $\mathcal{E}$) can now be efficiently and accurately (at second order) computed. A first algorithm has the form:

Algorithm 1.1: Hierarchical Iterative Method

0. Put $r = b - Au^0$.

(1) Perform a few iterations of the SOR method on the global system:

$$A\delta = \begin{bmatrix} A_{ll} & A_{lq} \\ A_{ql} & A_{qq} \end{bmatrix} \begin{bmatrix} \delta_l \\ \delta_q \end{bmatrix} = \begin{bmatrix} r_l \\ r_q \end{bmatrix} = r.$$

(2) Compute the new residual $r^* = \begin{bmatrix} r_l^* \\ r_q^* \end{bmatrix} = \begin{bmatrix} r_l - A_{ll}\delta_l - A_{lq}\delta_q \\ r_q - A_{ql}\delta_l - A_{qq}\delta_q \end{bmatrix}$

(3) Solve $A_{ll}\delta^* = r_l^*$ (by a direct method or approximately by a Krylov method).

(4) Update the solution $\begin{bmatrix} u_l^{k+1} \\ u_q^{k+1} \end{bmatrix} = \begin{bmatrix} u_l^k \\ u_q^k \end{bmatrix} + \begin{bmatrix} \delta_l + \delta^* \\ \delta_q \end{bmatrix}$

(5) Update the residual $r = \begin{bmatrix} f_1 - A_{ll}u_l^{k+1} - A_{lq}u_q^{k+1} \\ f_2 - A_{ql}u_l^{k+1} - A_{qq}u_q^{k+1} \end{bmatrix}$

(6) Repeat steps 1–5 until convergence.

Depending on the size of the mesh and on the available computer memory, a direct solver can be used for the linear systems involving A_{ll}. Otherwise, a Krylov space method is more appropriate. Unfortunately, such a simple algorithm converges rather slowly. It is however an excellent preconditioner for a Krylov method on the global matrix A. If the global system is symmetric, a conjugate gradient method can be employed and otherwise, generalised minimal residual method (GMRES) (Saad, 2003) is an excellent choice and was used in the computation of the turbulent variables. We refer to El maliki (2007) and El maliki et al. (2011) for more details.

4.2. Solving the Reynolds averaged Navier—Stokes equations

The Reynolds averaged Navier–Stokes equations in velocity–pressure formulation are discretised by the second-order Taylor-Hood $P_2 - P_1$ element. Here again, the quadratic part (for the velocity) is written in a hierarchical basis as in the previous section. Newton's method is used to linearise the convective term leading to non-symmetric linear systems of the form:

$$\begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f_u \\ f_p \end{bmatrix}$$

Solving saddle-point systems by iterative methods is not a trivial task and we refer to Elman, Sylvester, and Wathen (2005) for a more complete description. We now follow the approach developed in El maliki and Guénette (2010) in the case of laminar flows with constant viscosity. We want a true mixed iterative method in the sense that both the primal (for the velocity) and the dual (for the pressure) problems are solved simultaneously. The key point is the use of a block triangular right preconditioner of the form:

$$P_R = \begin{bmatrix} A & B^T \\ 0 & -S \end{bmatrix} \text{ with } P_R^{-1} = \begin{bmatrix} A^{-1} & A^{-1}B^TS^{-1} \\ 0 & -S^{-1} \end{bmatrix}$$

where $S = BA^{-1}B^T$ is the Schur complement. With this choice, we have:

$$\begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix} P_R^{-1} = \begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix}$$

and clearly this new system is very well conditioned. The definition of P_R^{-1} requires the construction of S^{-1} (and A^{-1}) which is not possible for large systems. The key point is that since they appear in the preconditioner, we only need good working approximations of these inverse matrices that will be denoted as $\tilde{A}^{-1}$ and $\tilde{S}^{-1}$.

Note that the generally non-symmetric matrix A has exactly the same form as in the previous section and therefore, can be decomposed in its linear and quadratic parts and solved in the same way. We thus choose for $\tilde{A}^{-1}$ to make a few iterations of GMRES preconditioned by Algorithm 1.1 presented in Section 4.1. Only a few iterations are performed and thus, linear systems involving A^{-1} are not solved to full accuracy and thus $\tilde{A}^{-1} \neq A^{-1}$.

The construction of the inverse of the Schur complement is more delicate and the lack of space prevents us from giving a complete description. Obviously, building $\tilde{S}$ explicitly is not feasible nor necessary. We only need an easily computable approximation of its inverse. Following El maliki and Guénette (2010) and the references therein, it can be shown that:

$$S^{-1} \simeq M_p^{-1} + \rho M_p^{-1} C_p D_p^{-1} = M_p^{-1}(D_p + \rho C_p) D_p^{-1} = \tilde{S}^{-1}$$

where the above matrices are defined using the pressure basis function ϕ_i

$$M_p^{ij} = \int_\Omega (r + \mu_T + \mu)\phi_j \ \phi_i \ dx, D_p^{ij} = \int_\Omega \nabla\phi_j \cdot \nabla\phi_i \ dx, \ C_p^{ij} = \int_\Omega (\boldsymbol{w} \cdot \nabla\phi_j)\phi_i dx$$

and where $\boldsymbol{w}$ is the velocity field from the previous iteration in Newton's scheme. Each iteration of this preconditioner thus involves the solution of two linear systems, one for M_p and the other for D_p. Note that boundary conditions must be provided when using D_p.

Now that we have defined the preconditioner, a (preconditioned) generalised conjugate residual method (Eisenstat, Elman, & Schultz, 1983) is used for the overall system. This choice is natural since the matrix is generally non-symmetric, but other choices are possible such as flexible GMRES (Saad, 2003). The numerical performances of this algorithm are discussed in El maliki and Guénette (2010) in rather simple situations. Our experience is that the method performs very well in more complex situations as will be seen in the section on numerical results.

5. Anisotropic mesh adaptation

Adaptive remeshing strategies are now commonly used in CFD codes since they allow to control the error level on a given numerical solution thus providing very accurate solutions. The idea is simple: starting with a first (coarse) mesh, the corresponding solution is computed and the error is then estimated using some *a posteriori* error estimator. The mesh is then modified in order to better fit the dependent solution variables and a new solution is computed. This process is repeated many times until the numerical solution and the mesh no longer change and/or some error level has been reached.

Adaptive remeshing strategies allow the concentration of small elements only where needed, that is where the error is estimated large. Moreover, our adaptive method allows for anisotropic meshes where some element can be stretched along preferential directions. Some elements can thus present a large aspect ratio. An important consequence is the reduction of the number of elements needed to obtain a prescribed accuracy.

The anisotropic adaptive strategy used in this work is described in more detail in Belhamadia, Fortin, and Chamberland (2004). The reader is also referred to Habashi et al. (2000), Ait Ali Yahia et al. (2002), Dompierre, Vallet, Bourgault, Fortin, and Habashi (2002) and Hecht and Mohammadi (1997) for a complete presentation. For a linear (P_1) finite element solution, it is well known that the error E has the form $E \simeq C\, e^t H(x) e$, where C is a constant, $e = \boldsymbol{P}_2 - \boldsymbol{P}_1$ is an element edge with vertices $\boldsymbol{P}_2$ and $\boldsymbol{P}_1$ and $H(x)$ is the Hessian matrix of the solution. If the Hessian is positive definite, the term $e^t H(x) e$ can be used to define a new metric (the Euclidian norm corresponding to $H = I$) thus linking the error to this new edge length. The adaptive method therefore tries to obtain a new mesh where all edges are of the same length in the new metric (which varies throughout the domain). As a consequence, the error will essentially be the same on all edges and therefore, the adapted mesh presents some form of equirepartition of the error. For systems of equations, it may be necessary (and in many instances it is essential) to adapt the mesh on the different variables of the problem. These variables (velocity, pressure, temperature, etc.) may be of different nature and units. They are first non-dimensionalised and a new metric is computed for each of them. All these metrics (or more precisely the ellipses associated to the Hessian matrices) are then intersected as proposed in Alauzet and Frey (2003) in order to provide a common metric on which the mesh is adapted taking into account the errors on all variables.

This strategy has to be modified if quadratic (P_2) solutions are used. As proposed in Alauzet and Frey (2003), metrics can be obtained by considering the Hessian matrix of the different components of the gradient of each variable. This requires to recover third-order derivatives which is done by solving least square problems on patches of elements around each node as described in Zhang and Naga (2005). Here again, these metrics have to be intersected.

The main steps of the adaptive procedure are the following:

(1) An initial mesh is provided and the problem is solved providing a numerical solution for u, p, $\mathcal{K}$ and $\mathcal{E}$.
(2) Approximations of the matrix of third-order derivatives of the different variables are computed at each node.
(3) The resulting metrics are intersected in order to provide a unique metric.
(4) The mesh is then modified using local operations. Nodes and edges are swept a few times in order to perform:

- Edge refinement and node suppression to control the edge length and thus, the error level.
- Edge swapping and node displacement to control the quality of the elements.

(5) A new mesh is produced and the process is repeated until the desired error level is reached.

As will be seen in the numerical results, the resulting meshes may present elements with large aspect ratio in regions where the different variables allows it. The meshes may therefore be anisotropic. It must be noted however that the different variables may have conflicting behaviours and that anisotropy is possible only when all variables agree in some direction.

6. Numerical results

We now present numerical results on three different benchmark problems. We first compare our numerical solution with an analytical closed form solution (manufactured solution) of a shear layer. The second test case is the flow over a thin flat plate with finite thickness and the last one is the flow over a backward facing step (BFS) in three dimensions.

6.1. A shear layer with a closed form solution

This problem served as a validation case in Pelletier, Turgeon, and Borggaard (2004) to test the accuracy of their error estimation technique and the efficiency of their isotropic mesh adaptation strategy. This is an example of a so-called *manufactured solution*. The computational domain is the rectangle [100, 300] × −75, 75. The velocity and the turbulence fields depend on *x* and *y*:

$$\begin{aligned} u_x(x,y) &= U_1\left[\left(\tfrac{1+r}{2}\right)+\left(\tfrac{1-r}{2}\right)\,\mathrm{erf}\,\left(\tfrac{\sigma y}{x}\right)\right] \\ u_y(x,y) &= U_1\left(\tfrac{1-r}{2}\right)\tfrac{1}{\sigma\sqrt{\pi}}\exp\left(-\left(\tfrac{\sigma y}{x}\right)^2\right) \end{aligned} \tag{14}$$

$$\begin{aligned} k(x,y) &= k_0\left[c_k+\exp\left(-\left(\tfrac{\sigma y}{x}\right)^2\right)\right] \\ \epsilon(x,y) &= \tfrac{\epsilon_0}{x}[c_k+\exp(-(\tfrac{\sigma\,y}{x})^2)] \end{aligned} \tag{15}$$

and the pressure vanishes ($p(x,y) = 0$). The constant values are:

$$U_1 = 1.0, \quad r = 0.3, \quad \sigma = 13.5, \quad c_k = \frac{10^{-4}}{k_0}, \quad \rho = 1,$$

$$\mu = \frac{1}{Re} = 10^{-4}, \quad k_0 = \frac{343}{75,000}U_1^2(1-r)\frac{\sigma}{\sqrt{\pi}}, \quad \epsilon_0 = \frac{343}{22,500}C_\mu U_1^3(1-r)^2\frac{\sigma^2}{\pi}.$$

The different components of this manufactured solution are illustrated in Figure 1. Dirichlet boundary conditions are applied for all variables on the whole boundary. The above exact solutions are substituted into the Navier–Stokes and turbulence transport equations, and appropriate source terms are determined using Maple and added to the corresponding equations.

Computations were first performed on a series of uniform structured meshes (not illustrated) having, respectively, 3626, 14,504 and 58,016 elements (7425, 29,353 and 116,721 nodes). A series of adapted meshes (Figure 2) was also produced following the procedure explained in the previous section. In the lower and upper parts of the computational domain, the different variables vary very little and the elements are large and isotropic. In the transition region, the elements are strongly stretched as all variables vary in the same direction. In the central region, however, elements are more isotropic in nature. This is due to the fact that strongest variations for each variable take place along different directions, as shown in the isolines of u_x (the x component of u), u_y, $\mathcal{K}$ and $\mathcal{E}$ depicted in Figure 1. This is a nice

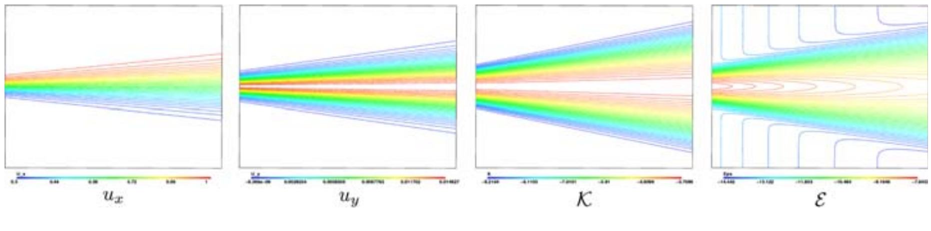
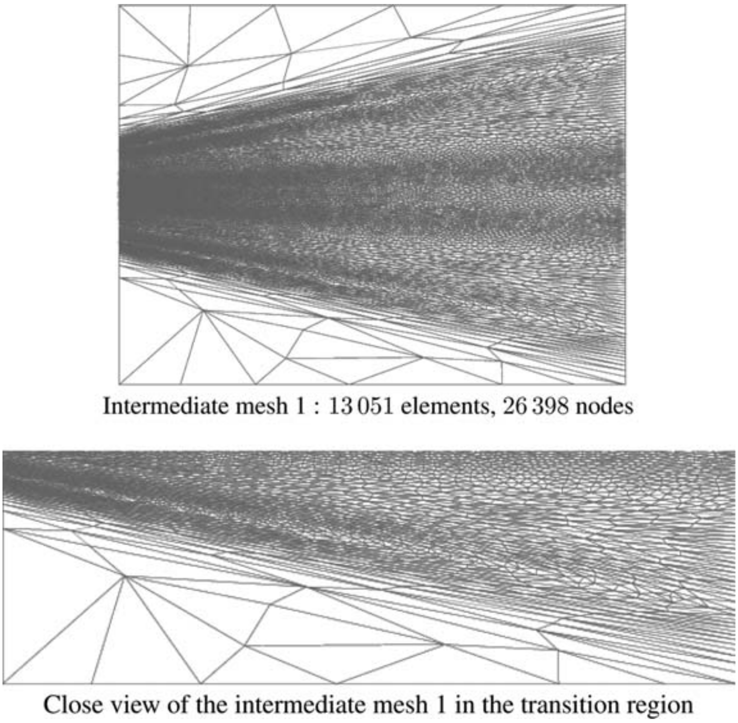


Figure 1. Shear layer test case: manufactured solutions.



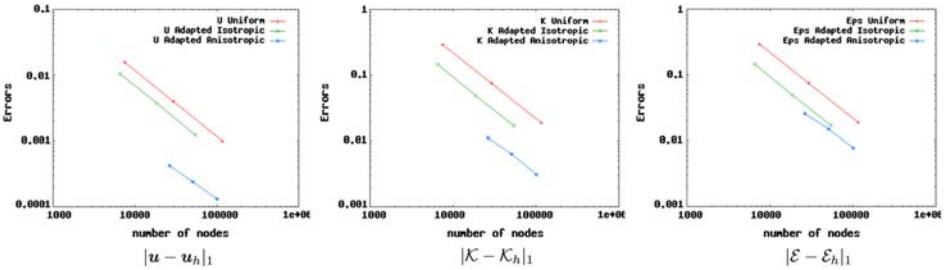
Intermediate mesh 1 : 13 051 elements, 26 398 nodes

Close view of the intermediate mesh 1 in the transition region

Figure 2. Shear layer test case: example of adapted mesh.

example of conflicting interests between the different variables. Anisotropic elements would not be appropriate in the central region.

The errors, as a function of the number of nodes, are presented in Figure 3 for each series of uniform and adapted meshes and compared with those obtained on isotropic meshes from Pelletier et al. (2004). The number of nodes includes each element vertex and the node associated to each element edge, since quadratic finite elements are used. Quadratic convergence was observed for all variables, including the pressure which is not shown. The errors on the anisotropic meshes is lower for all variables, but the gain in accuracy is more pronounced for the velocity where the error is one order of magnitude smaller with respect to isotropic meshes.



$|\boldsymbol{u} - \boldsymbol{u}_h|_1$ $|\mathcal{K} - \mathcal{K}_h|_1$ $|\mathcal{E} - \mathcal{E}_h|_1$

Figure 3. Shear layer test case: comparison of errors (using the usual $H^1(\Omega)$ -seminorm) between anisotropic (this work), isotropic (from Pelletier et al., 2004) and uniform meshes.

6.2. Skin friction on a thin flat plate

For this example, we consider a turbulent flow around a thin flat plate. The computational domain and the boundary conditions are presented in Figure 4. The Reynolds number is set to 200,000.

This test case was studied in Lacasse, Turgeon, and Pelletier (2004) to first observe the effect of the inflow turbulence Reynolds number $Re_T = \frac{\epsilon_0}{\rho C_\mu k_0^2}$, where k_0 and ϵ_0 are the inlet values for k and ϵ, respectively. Re_T is set to 20,000 (10% of Re) with $k_0 = 10^{-6}$ and $\epsilon_0 = 1.8 \times 10^{-9}$.

A few adaptation cycles led to the mesh illustrated in Figure 5 which presents a clustering of elements along the plate. Elements are isotropic in the leading edge region, but are clearly anisotropic downstream along the plate. This is in accordance with, for instance, the production of *k*, which depends on the velocity gradient and which is triggered in the leading edge region (Figure 6). The physically observed stagnation point at the leading edge of the plate is reproduced in the simulation, leading to a large production of turbulent kinetic energy. Finally, as in Lacasse et al. (2004), we observe that the skin friction coefficient $C_f(x) = \frac{\tau_w}{\frac{1}{2}\rho U^2}$, where U is the velocity norm, compares well with White's correlation White (1974) (Figure 7).

6.3. 3D flow over a BFS

In this section, we show that our numerical strategy can be applied in a 3D case. We consider the BFS problem which was studied experimentally in (19). The geometry and the boundary conditions are shown in Figure 8. The Reynolds number based on the inlet mean velocity U

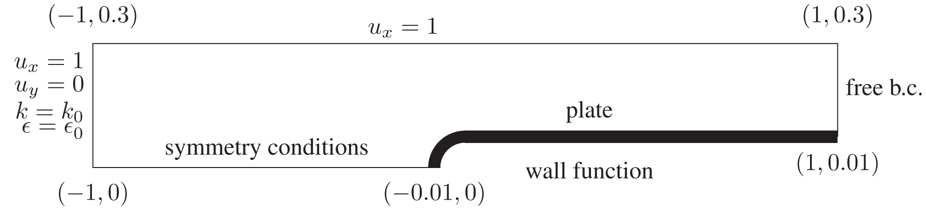


Figure 4. Thin plate problem: geometry and boundary conditions.

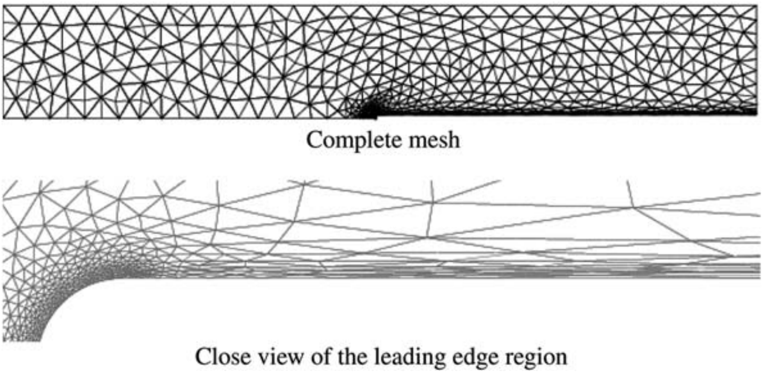


Figure 5. Thin plate problem: adapted mesh (1405 elements).

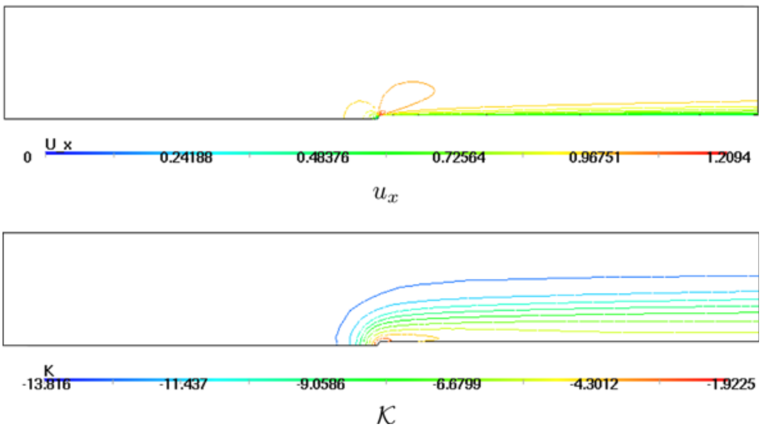


Figure 6. Thin plate problem: isolines of the solution on the adapted mesh.

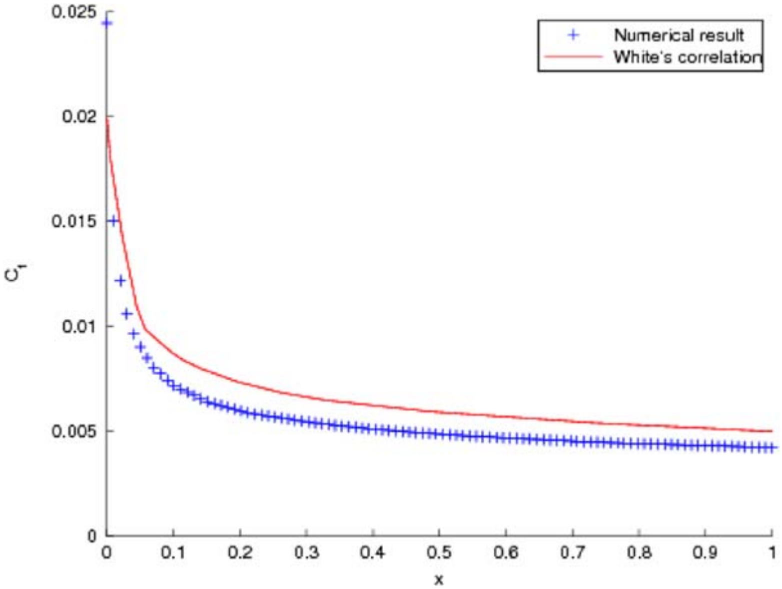


Figure 7. Thin plate problem: profile of the skin friction coefficient.

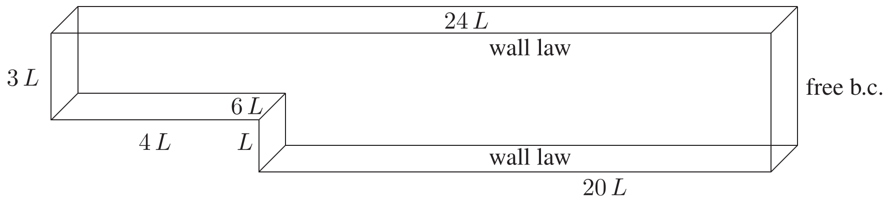
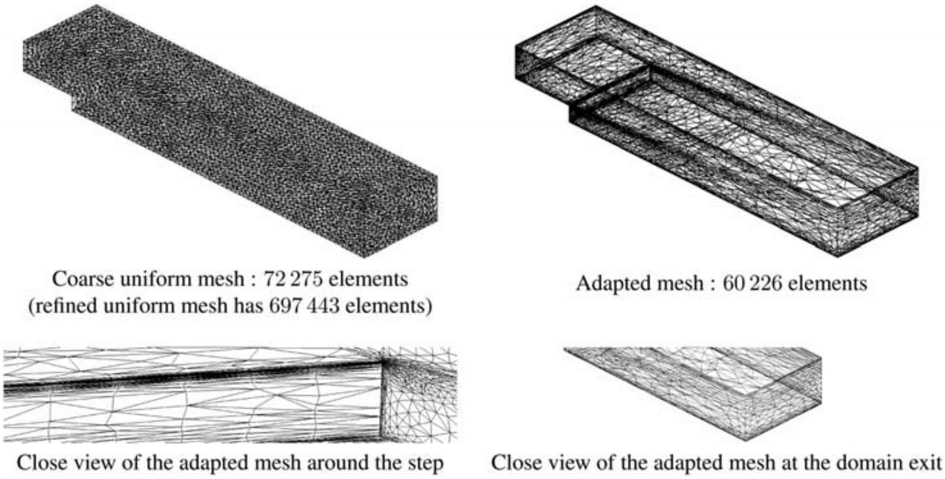


Figure 8. BFS problem: geometry and boundary conditions.



Coarse uniform mesh : 72 275 elements
(refined uniform mesh has 697 443 elements)

Adapted mesh : 60 226 elements

Close view of the adapted mesh around the step

Close view of the adapted mesh at the domain exit

Figure 9. BFS problem: computational meshes.

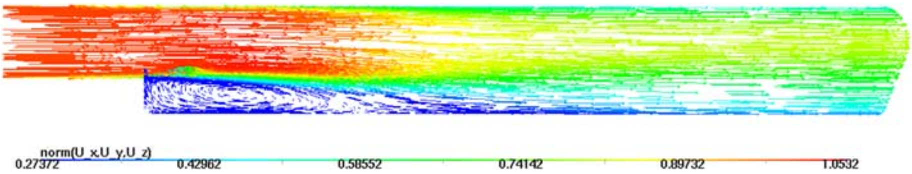


Figure 10. BFS problem: recirculation region on the adapted mesh (median plane).

and the step height L is $Re = UL/\mu = 47,625$. Uniform profiles are specified for all variables at the inlet section. The turbulence intensity at the inlet (ratio between k and U^2) was set to 2%. At the inlet, the following values were thus prescribed: $u_x = 1.0$, $u_y = 0.0$, $k = 0.02$, $\epsilon = 0.01, 524$. At the outlet, free boundary conditions for all the variables are prescribed. At every solid wall, wall boundary conditions are applied with $y = 0.01$ except at the vertical step wall where $y = 0.02$.

The adapted mesh of Figure 9 presents anisotropic features in relevant parts of the domain. The flow field (Figure 10) shows a recirculation whose length is approximately $6.233 \times L$, in good agreement with numerical and experimental results found in the literature (see for instance Kim, 1978; Kuzmin & Mierka, 2006; Le, Moin, & Kim, 1997). Finally, in order to appreciate the advantage of the anisotropic remeshing strategy, we compare in Figure 11 the velocity profile computed with a coarse uniform mesh (72,225 elements), a refined uniform mesh (697,443 elements) and an adapted mesh (60,226 elements). The adapted mesh shows essentially the same profile as the fine structured mesh, with a number of elements almost 10 times smaller. The figure also includes the experimental results from (19) and, as easily seen, the agreement is very satisfactory.

6.4. Performance of the solvers

The numerical performance of the different solvers described in Section 4 was abundantly described in El maliki (2007), El maliki et al. (2011), El maliki and Guénette (2010) but only for isotropic structured meshes. The scalar solver (see Section 4.1) was tested for elliptic and

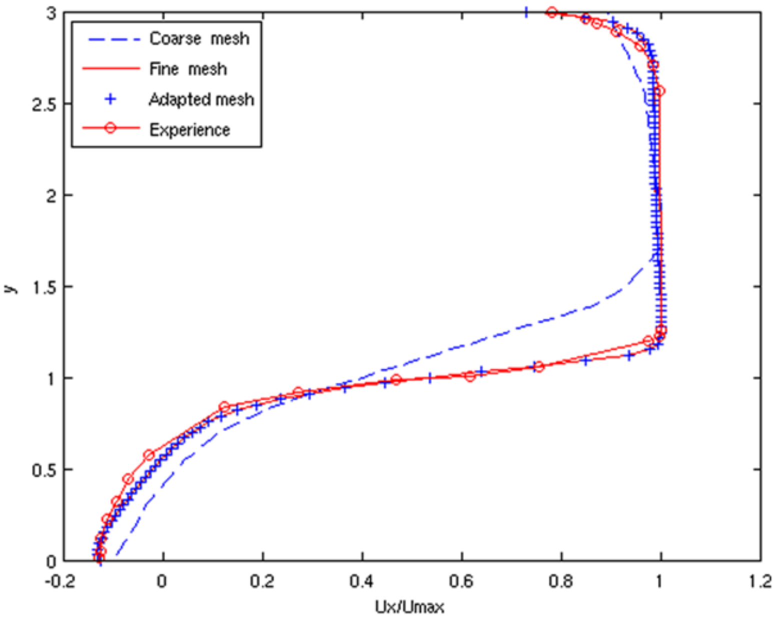


Figure 11. BFS problem: velocity profile at a distance $x = 4L/3$ at the middle plane after the step.

mildly convective convection–diffusion problems while the Navier–Stokes solver (see Section 4.2) was tested for laminar (constant viscosity) flows.

It was not our purpose to redo all the comparisons with other iterative methods. This was done in the above references. We have however shown that they are extremely efficient in much more complex flow problems. In the presented computations, no important difference was observed in the behaviour of the solvers between structured and anisotropic meshes. The number of Krylov iterations is slightly higher for anisotropic meshes but not significantly. This is true for both the scalar and saddle-point solvers. The solution strategy presented in this paper is thus extremely efficient and accurate since it provides second-order accurate convergence of all variables at a very reasonable cost.

It is worth adding a first concluding remark. When starting a new problem, we always use a coarse, regular and more or less uniform mesh since otherwise we may have convergence problems with the different iterative methods. Using a mesh containing elements elongated in inappropriate directions will most probably make our iterative methods break down. This is the case with most iterative methods. Fortunately, mesh adaptation elongates the elements in directions compatible with the solution of the problem and, as our numerical results show, the iterative methods continue to behave very well. This is precisely what we wanted to verify in this work.

7. Conclusions

In this work, we have presented different iterative methods that can be applied to the solution of turbulent flows with a $k - \epsilon$ model. The presented algorithms require a specific hierarchical basis for classical Lagrange finite element discretisations. The resulting iterative solvers perform very well on unstructured grids and can therefore be used in conjunction with an anisotropic mesh adaptation procedure. The resulting scheme is robust and accurate at the same time.

Acknowledgements

The authors wish to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Ait Ali Yahia, D., Baruzzi, G., Habashi, W.G., Fortin, M., Dompierre, J., & Vallet, M.-G. (2002). Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver independent CFD. Part II Structured grids. *International Journal for Numerical Methods in Fluids, 39*, 657–673.

Alauzet, F., & Frey, P. (2003). *Estimateur d'erreur géométrique et métriques anisotropes pour l'adaptation de maillage. Partie I: aspects théoriques (Geometric error estimator and anisotropic metric for mesh adaptation. Part I: theoretical aspects).* (Technical Report No. 4759) INRIA.

Belhamadia, Y., Fortin, A., & Chamberland, É. (2004). Three-dimensional anisotropic mesh adaptation for phase change problems. *Journal of Computational Physics, 201*(2), 753–770.

Chabard, J. (1991). *Projet N3S de mécanique des fluides-manuel théorique de la version 3 (N3S project for fluid mechanics: theoretical manual for version 3)*. (Technical Report n° EDF HE -41=91.30B), Électricité de France.

Dompierre, J., Vallet, M.-G., Bourgault, Y., Fortin, M., & Habashi, W.G. (2002). Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part III: Unstructured meshes. *International Journal for Numerical Methods in Fluids, 39*, 675–702.

Eisenstat, S., Elman, H., & Schultz, M. (1983). Variational iterative methods for nonsymmetric systems of linear equations. *SIAM Journal on Numerical Analysis, 20*, 345–357.

El maliki, A. (2007). *Résolution de problèmes aux limites à l'aide de méthodes itératives hiérarchiques à préconditionneur variable (Solution of boundary value problems using hierarchical iterative methods with variable preconditioner)* (PhD thesis, Département de mathématiques et de statistique, Université Laval, Québec, Canada).

El maliki, A., & Guénette, R. (2010). Efficient preconditioning techniques for finite-element quadratic discretization arising from linearized incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids, 63*(12), 1394–1420.

El maliki, A., Guénette, R., & Fortin, M. (2011). An efficient hierarchical preconditioner for quadratic discretizations of finite element problems. *Numerical Linear Algebra with Applications*, n/a n/a. Published online.

Elman, H., Sylvester, D., & Wathen, A. (2005). *Finite elements and fast iterative solvers with applications in incompressible fluid dynamics*. New York, NY: Numerical mathematics and scientific computation, Oxford science.

Habashi, W.G., Dompierre, J., Bourgault, Y., Ait Ali Yahia, D., Fortin, M., & Vallet, M.-G. (2000). Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solverindependent CFD. Part I: General principles. *International Journal for Numerical Methods in Fluids, 32*, 725–744.

Hecht, M., & Mohammadi, B. (1997). *Mesh adaptation by metric control for multi-scale phenomena and turbulence*. 35th Aerospace Sciences Meeting & Exhibit, n° 97–0859, Reno, USA.

Hughes, T.J.R., & Brooks, A.N. (1979). *A multidimensional upwind scheme with no crosswind diffusion, vol. 34 of finite element methods for convection dominated flows*. New York, NY: American Society of Mechanical Engineers, 19–35.

Hughes, T.J.R., & Brooks, A.N. (1982). A theoretical framework for Petrov-Galerkin methods with discontinuous weighting functions. Applications to the streamline upwind procedure. In R.H. Gallagher (Ed.), *Finite element in fluids*, vol. IV, London: Wiley.

Ilinca, F., & Pelletier, D. (1998). Positivity preservation and adaptive solution for the $k - \varepsilon$ turbulence model. *AIAA J., 36*(1), 44–50.

Kim, J. (1978). *Investigation of separation and reattachment of turbulent shear layer: Flow over a ackward facing step* (PhD thesis, Stanford University).

Kuzmin, D., & Mierka, O. (2006). On the implementation of the $k - \varepsilon$ turbulence model in incompressible flow solvers based on a finite element discretization. In Lude, Rapin (Eds.), *BAIL 2006: International conference on Boundary and Interior Layers*, Göttingen: Georg-August University Göttingen.

Lacasse, D. (2004). *Application d'une méthode d'éléments finis adaptative à des écoulements turbulents (Application of an adaptive finite element method to turbulent flows)* (Master's thesis, École Polytechnique de Montréal, Montréal).

Lacasse, D., Turgeon, É., & Pelletier, D. (2004). On the judicious use of the $k-\varepsilon$ model, wall functions and adaptivity. *International Journal of Thermal Sciences, 43*(10), 925–938.

Launder, B.E., & Spalding, D.B. (1972). *Mathematical models of turbulence*. London: Academic Press.

Le, H., Moin, P., & Kim, J. (1997). Direct numerical simulation of turbulent flow over a backward-facing step. *Journal of Fluid Mechanics, 330*, 349–374.

Lew, A.J., Buscaglia, G.C., & Carrica, P.M. (2001). A note on the numerical treatment of $k-\varepsilon$ the turbulence model. *International Journal of Computational Fluid Dynamics, 14*(3), 201–209.

Pelletier, D., Turgeon, É., & Borggaard, J. (2004). A general continuous sensibility equation formulation for the $k-\varepsilon$ formulation. *International Journal of Computational Fluid Dynamics, 18*(1), 29–46.

Pelletier, D., Zaki, A., & Fortin, A. (1994). Adaptive remeshing for hyperbolic transport problems. *International Journal of Computational Fluid Dynamics, 3*(2), 79–99.

Saad, Y. (2003). *Iterative methods for sparse linear systems* (2nd ed.). Philadelphia, PA: SIAM.

Schetz, J.A. (1993). *Boundary layer analysis*. Upper Saddle River, NJ: Prentice Hall.

Verfürth, R. (1996). *A review of a posteriori error estimation and adaptive mesh-refinement techniques*. Chichester: Advances in numerical mathematics, Wiley-Teubner.

White, F.M. (1974). *Viscous flow*. New York, NY: McGraw-Hill.

Zaki, A. (1993, July). *Simulation numérique des problèmes de convection sur des maillages adaptatifs non structurés du type $h-p$* (PhD thesis, École Polytechnique de Montréal, Montréal, Canada).

Zhang, Z., & Naga, A. (2005). A new finite element gradient recovery method: Superconvergence property. *SIAM Journal on Scientific Computing, 26*(4), 1192–1213.