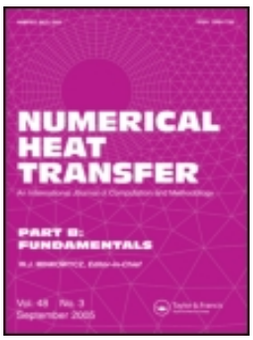


Taylor & Francis
Taylor & Francis Group

Numerical Heat Transfer, Part B: Fundamentals
An International Journal of Computation and Methodology

ISSN: 1040-7790 (Print) 1521-0626 (Online) Journal homepage: http://www.tandfonline.com/loi/unhb20

Residual Least-Squares Error Estimate for Unstructured h-Adaptive Meshes

Duarte M. S. Albuquerque , José M. C. Pereira & José C. F. Pereira

To cite this article: Duarte M. S. Albuquerque , José M. C. Pereira & José C. F. Pereira (2015) Residual Least-Squares Error Estimate for Unstructured h-Adaptive Meshes, Numerical Heat Transfer, Part B: Fundamentals, 67:3, 187-210, DOI: 10.1080/10407790.2014.950074

To link to this article: http://dx.doi.org/10.1080/10407790.2014.950074

Published online: 10 Dec 2014.

Submit your article to this journal

Article views: 76

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
http://www.tandfonline.com/action/journalInformation?journalCode=unhb20

Download by: [b-on: Biblioteca do conhecimento online UL] **Date:** 29 May 2017, At: 20:33

Numerical Heat Transfer, Part B, 67: 187–210, 2015
Copyright © Taylor & Francis Group, LLC
ISSN: 1040-7790 print/1521-0626 online
DOI: 10.1080/10407790.2014.950074

Taylor & Francis
Taylor & Francis Group

RESIDUAL LEAST-SQUARES ERROR ESTIMATE FOR UNSTRUCTURED *h*-ADAPTIVE MESHES

Duarte M. S. Albuquerque, José M. C. Pereira, and José C. F. Pereira
LAETA, IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal

An* a posteriori *error estimate suitable for finite-volume adaptive computations is presented. The error estimate combines the least-squares method regressions with the residual computation, which provides information from the grid quality and the governing equations for a better local adaptation of the unstructured grid.

The decision algorithm uses the information provided by the error estimate and does not require problem-dependent constants; it also uses a grid interface correction step to provide a smoother and a high-quality adaptive grid.

The proposed error estimate and the adaptive refinement algorithm are verified against analytic solution for different two-dimensional problems. In addition, calculations of three-dimensional laminar flows with different types of unstructured grids have demonstrated the applicability of the adaptive method.

1. INTRODUCTION

Computational fluid dynamics (CFD) applications of complex geometric configurations require too much effort to generate the mesh. In addition, the change of flow parameters or minor geometric configurations require modifications to the grid by either increasing or moving the local cell resolution in order to efficiently capture all relevant flow phenomena.

Automatic adaptive mesh refinement (AMR) techniques have the potential of improving the solution accuracy; see, e.g., [1]. The adaptive refinement algorithm requires sensors, classified as error indicators and error estimates, to provide information for the local grid adaptation. The selection of error indicators based on large flow gradients of intuitive physical features, such as vorticity, shock waves, and boundary layers, does not necessarily guarantee a corresponding reduction in the overall discretization error. The termination criterion for refinement cannot be based

Received 5 March 2014; accepted 16 June 2014.

Address correspondence to José C. F. Pereira, Mechanical Engineering Department/LASEF, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisbon, Portugal. E-mail: jcfpereira@tecnico.ulisboa.pt

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/unhb.

NOMENCLATURE

a_p, a_l	momentum matrices entries	$\mathbf{u}$	velocity vector
A, B	analytic constants	u, v	velocity components
b	coefficient vector (WLS)	U_f	face velocity
C	contribution of different schemes	V_P	cell volume
		w_P	weighted function of cell P
$\mathbf{d}_n$	distance vector between cells P and P_n	W	weight matrix (WLS)
		$\mathbf{x}_p$	cell P centroid
D, L	reference diameter, length	x, y	Cartesian coordinates
e	least-squares residual vector (WLS)	X	least-squares rectangular matrix (WLS)
E	numerical error	y	observation vector (WLS)
$\mathbf{E}_R$	residual least-squares error estimate	α_u, α_p	momentum and pressure relaxation factors
E_T	Taylor-series error estimate	Γ	diffusion coefficient
h	hydraulic radius (mesh)	ζ_1, ζ_2, ϕ_1, ϕ_2, ψ	analytic functions
K_0	modified Bessel function of second kind and zero order	ν	kinematic viscosity
		ρ	density
M_j	jet momentum	ϕ	transported variable
p	pressure		
p'	pressure correction	**Subscripts and Superscripts**	
P	cell or volume control (mesh)		
r, γ	polar coordinates	$\bar{a}$	interpolated values of a
Re	Reynolds number	i, j	coordinate indexes
$\mathbf{R}_M$, R_C	momentum and continuity residuals	L	level of refinement
		n	iteration number
S	source magnitude	1, 2, 3	index numbers for faces and cell neighbors
S_f	cell's face (mesh)	*	approximated velocity

on error indicators because they do not provide information on error levels in the domain. Indicator-based adaptive refinement algorithms need user-defined parameters that are problem-dependent. Error estimates allow one to guess an error distribution in the domain and can therefore be exploited to derive a suitable termination criterion.

In the framework of the finite-element method (FEM), the adaptive procedures usually rely on *a posteriori*–type error estimates of residuals or quantities of interest: global error-based estimates, adjoint-based estimates; and residual estimates. Summaries of different error estimates has been published; see, e.g., [2–11]. For fluid dynamics there has been a large amount of research devoted to *a posteriori* error analysis and optimal grid adaptation for functional outputs; see [12–14]. The error assessments made in predicting integral quantities representing engineering output have established an implicit *a posteriori* procedure for computing upper and lower bounds on functional outputs of finite-element solutions.

In the case of the finite-volume method (FVM), the Richardson extrapolation [15, 16] is the most popular error estimate in the FVM context but requires solutions on two meshes with different spacings, which can be difficult to obtain in 3-D industrial flow configurations. There have been some attempts at single-mesh error estimates, based on energy conservation and angular moment conservation

equations; see Haworth et al. [17]. An error estimate based on high-order face interpolation was proposed by Muzaferija and Gosman [18], and later, Jasak and Gosman [19–21] proposed two error estimates; the first one based in the Taylor-series truncation error and the second one based on the conservation of the second moment of the transported variable. Yahia et al. [22] applied the Taylor-series truncation error to edges integrals in the framework of *r*-adaptivity.

Residual error estimates have not been extensively studied, particularly in the context of finite-volume computations. The residual error estimates can provide an estimate of the local truncation error, which is the extent to which the discrete algebraic equation differs from the partial differential equation it models. An error estimate based on the residual error from the governing equations was investigated by Jasak and Gosman [23] who proposed an element residual estimate for finite-volume discretizations, and Juretic [24] extended it to a face-based error estimate. Also, some techniques were used to extend error estimates from the FEM to a cell-centered FVM by Nicaise [25] using the Morley interpolation or with the approach of self-equilibrium fluxes developed by Jasak and Gosman [26]. Aftsomis and Berger [27] computed the local truncation error by using τ-extrapolation, this method requires the residual evaluation of the numerical solution in a coarser grid. Hay and Visonneau [28] proposed the use of a higher-order reconstruction operator to evaluate the residual, while Karni and Kurganov [29] introduced the use of a localized test function which proved to have the same behavior as the truncation error. Roy and Sinclair [30] provide a two-dimensional spline-fitting approach based on the method of nearby problems to estimate discretization error. Ganesh et al. [31, 32] proposed an estimate called the $\Re$-parameter which is an approach to residual estimation in a finite-volume framework.

Most of the methods still have limitations in complex meshes regarding the automatic adaptive refinement because they require some empirical value based on the numerical error distribution that requires previous knowledge of the solution. This is not suitable, when building an automatic refinement procedure.

Very often, second-order accuracy on adaptive grids is achieved by applying projection schemes in FVM problems with adaptive octree grids for Euler equations by Martin and Colella [33] or for incompressible Navier-Stokes equations by Martin et al. [34]. These schemes are not extendable for triangular, tetrahedral, and prismatic adaptive grids, which are used in problems where the geometric complexity is relevant.

The main objective of this work is to develop a FVM error estimate in the framework of unstructured meshes. The residual least-squares (RLS) error estimate has two main advantages when compared to other approaches. The polynomial reconstruction done with the weighted least-squares (WLS) method has the versatility required to be used on unstructured grids which can have an arbitrary cell distribution. The residual re-evaluation has information from the governing equations and the grid quality. By treating the adaptive grids as unstructured ones and applying the FV schemes suitable for unstructured grids, a local second-order accuracy in the interfaces between the cells with different levels of refinement is guaranteed.

During the development of the adaptive algorithm, four important conditions were considered to create a robust and universal adaptive method:

1. The FV numerical schemes used are second-order for both the mean and maximum error. This is achieved by considering the adaptive grid as an unstructured one and using a face least-squares interpolation for both the convective and diffusive terms.
2. The RLS error estimate can be used in any unstructured grid and has information from both the governing equations and the grid quality.
3. The selected cells for refinement do not depend on user-defined constants which can vary with the type of numerical problem. All cases studied in this work use the same decision algorithm, which proves its applicability and versatility.
4. The grid interface correction step improves the distribution of the cell's levels of refinement in accord to the numerical error different scales, and also enhances the overall numerical method efficiency and the adaptive grid quality and smoothness.

The article is organized as follows. Section 2 starts with an exposition of the numerical methods used in the FV discretization, and afterwards the error estimates and adaptive decision algorithm used in this work are explained. The results are described in Section 3, which starts with a second-order verification of the FVM, through the quantification of the error decay in an adaptive grid with a double grid interface. Afterwards, the adaptive algorithm is verified for three different problems: the Laplace equation, the convection-diffusion equation, and the Navier-Stokes equations. For each of these benchmark problems, an analytical solution exists, and consequently it is possible to measure the adaptive algorithm efficiency. The proposed adaptive algorithm is applied to three different laminar flows: the flow past a 2-D circular cylinder, the confined flow past a 3-D square cylinder, and the flow around a sphere.

2. NUMERICAL METHOD

2.1. Governing Equations and Unstructured Grids Formulation

The steady isothermal flow of an incompressible fluid is governed by the mass and momentum conservation laws, being expressed by the incompressibility constraint and the Navier-Stokes equations:

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

$$\nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = \nabla \cdot \left(\nu \nabla \mathbf{u} + \nu \nabla^T \mathbf{u}\right) - \frac{1}{\rho} \nabla p \tag{2}$$

where $\mathbf{u}$ is the velocity vector, ν is the kinematic viscosity, ρ is the fluid's density, and p is the fluid's pressure. The governing equations are discretized on unstructured meshes made of cells with arbitrary topology. Each cell P is a polyhedron with a closed boundary ∂P which is composed by a variable number of faces $\partial P = \{S_1, S_2, \ldots, S_F\}$. Each face S_k is a plane polygon of arbitrary orientation which connects

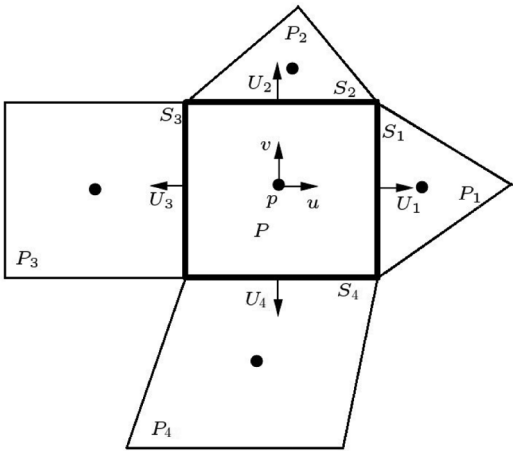


Figure 1. General 2-D polyhedral control volume.

P and its neighbor cell P_k, see Figure 1, each face has a set of different vertices in a well-defined orientation. The computational points are located at the centers of each control volume corresponding to the so-called collocated arrangement.

2.2. Pressure–Velocity Coupling Algorithm

The SIMPLE algorithm [35] is used for the pressure–velocity coupling. It starts by computing an approximate velocity field $\mathbf{u}^*$, which satisfies the momentum equations by using the values from the previous iteration n. The finite-difference equation is obtained after the linearization of the convective contribution:

$$\sum_{f=1}^{F} U_f^n \mathbf{u}_f^* - \nu_f \sum_{f=1}^{F} (\nabla \mathbf{u}^*)_f \cdot \mathbf{S}_f = -\frac{V_P}{\rho} \nabla p^n \tag{3}$$

where U_f^n is the face velocity defined by $\mathbf{u}_f^n \cdot \mathbf{S}_f$, and $\mathbf{S}_f$ is the face surface vector defined by $S_f \mathbf{n}_f$, with the normal unit vector of the face f designated by $\mathbf{n}_f$ and V_P denoting the cell P volume. A system of linear equations is assembled in this form:

$$\frac{1}{\alpha_u} a_p \mathbf{u}_p^* + \sum_{l=1}^{F} a_l \mathbf{u}_l^* = -\frac{V_P}{\rho} \nabla p^n + \frac{1-\alpha_u}{\alpha_u} a_p \mathbf{u}_p^n \tag{4}$$

where α_u is the underrelaxation factor for the momentum equations. The face velocity U_f^* is computed with Rhie-Chow interpolation [36]:

$$U_f^* = \overline{\mathbf{u}_p^*} \cdot \mathbf{S}_f - \overline{\frac{\alpha_u V_P}{\rho a_p}} \left[(\nabla p^n)_f - \overline{(\nabla p^n)} \right] \cdot \mathbf{S}_f \tag{5}$$

where the values with overbars are obtained by interpolation from the two adjacent cells which have the common face f (each face f belongs to two different cells, except

for boundary faces) and a_p are the momentum system matrix's main diagonal elements. The face velocities are used to solve the pressure-correction equation:

$$\sum_{f=1}^{F} \frac{\overline{\alpha_u V_P}}{\rho a_p} (\nabla p')_f \cdot \mathbf{S}_f = \sum_{f=1}^{F} U_f^* \tag{6}$$

where p' is the pressure correction. After solving the Poisson equation, Eq. (6), the velocity values are corrected with the p' values, therefore satisfying the continuity equation, Eq. (1). The face velocities are computed by

$$U_f^{n+1} = U_f^* - \frac{\overline{\alpha_u V_P}}{\rho a_p} (\nabla p')_f \cdot \mathbf{S}_f \tag{7}$$

and the cell velocities are corrected by

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\overline{\alpha_u V_P}}{\rho a_p} (\nabla p') \tag{8}$$

The pressure is updated,

$$p^{n+1} = p^n + \alpha_p p' \tag{9}$$

where α_p is an underrelaxation factor for the pressure update. The momentum residuals $\mathbf{R}_M$ are computed by the following equation:

$$\mathbf{R}_M = \frac{a_p \mathbf{u}_p^{n+1} + \sum_{l=1}^{F} a_l \mathbf{u}_l^{n+1} + (V_P/\rho)\nabla p}{a_p \max(\mathbf{u})} \tag{10}$$

To the authors' knowledge from previous works [37], it is important to use a residual adimensionalization, due to the resulting different cell volumes during the adaptive refinement algorithm, in order to give them the same residual contribution. The continuity residual R_C is computed by

$$R_C = \sum_{f=1}^{F} U_f^* \tag{11}$$

and the second norm is computed for all residuals. If all norms are lower than a prescribed value, the cycle ends; if not, the computation advances to the next iteration, back to Eq. (4).

2.3. Numerical Schemes

The weighted least-squares (WLS) method is used for the computation of the diffusive and convective terms on the arbitrary unstructured grids. A linear polynomial is built, centered at the face centroid, with the discrete information of

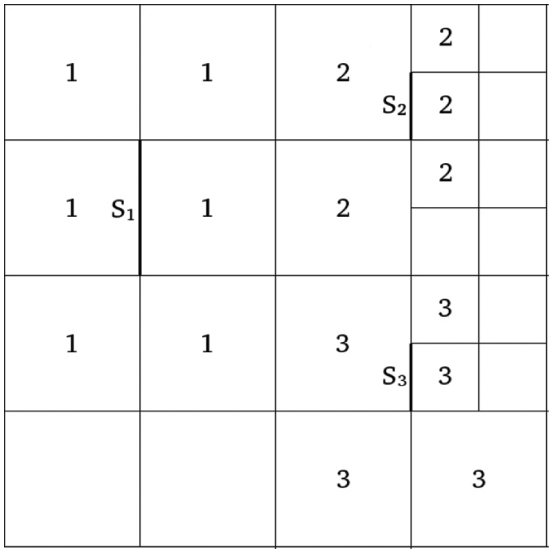


Figure 2. Examples of different stencils used in the WLS schemes.

the surrounding cells. This allows us to deal with the several orthogonality and skewness deviations that exist in the interfaces between the refined and nonrefined cells.

Both diffusive and convective terms are computed with a single regression, using a linear polynomial and the values from the cells that have the face's vertices. Figure 2 shows examples of different stencils used with these schemes. In each example, the regression k is centered in the face S_k and each cell of the stencil is marked with the respective number k.

The computational cost of solving the WLS problem for each face is high, but it is only required to be solved once. The matrix values can be saved and used in future iterations. All the least-squares regressions use a weight function W_P, given by the inverse square of the distance:

$$w_P = |\mathbf{x}_P - \mathbf{x}_{\text{ref}}|^{-2} \tag{12}$$

where $\mathbf{x}_P$ is the cell P centroid coordinates and $\mathbf{x}_{\text{ref}}$ is the coordinates of the regression reference. The convective and diffusive schemes used in this work may create a non-positive-definite matrix, so the deferred-correction approach from Ferziger [38] was required. A stable scheme is computed implicitly and the WLS-based schemes are computed explicitly. The deferred-correction finite-difference equation reads:

$$\frac{1}{\alpha_u} a_p \mathbf{u}_p^* + \sum_{l=1}^{F} a_l \mathbf{u}_l^* = -\frac{V_P}{\rho} \nabla p^n + \frac{1-\alpha_u}{\alpha_u} a_p \mathbf{u}_p^n + [C_{\text{stable}}(\mathbf{u}^n) - C_{\text{WLS}}(\mathbf{u}^n)] \tag{13}$$

where C is the contribution of the stable and the WLS schemes, respectively. As stable schemes, the central differences, Eq. (14), scheme is used for the diffusion

terms, while the convective fluxes are approximated by the first-order upwind, Eq. (15).

$$(\nabla \mathbf{u})_f = \mathbf{S}_f(\mathbf{u}_{P_1} - \mathbf{u}_P)/(\mathbf{S}_f \cdot \mathbf{d}_1) \tag{14}$$

$$\mathbf{u}_f = \begin{cases} \mathbf{u}_P & \Leftarrow U_f \geq 0 \\ \mathbf{u}_{P_1} & \Leftarrow U_f < 0 \end{cases} \tag{15}$$

where $\mathbf{d}_1$ is the distance vector between the centroids of the cells P and P_1 and $\mathbf{u}_{P_1}$ is the velocity value from cell P_1 which shares the face f with cell P. The deferred correction implies that at least two correction steps are required for the pressure-correction computation in order to avoid divergence.

3. ADAPTIVE ALGORITHM AND ERROR ESTIMATORS

3.1. Adaptive Algorithm

Error estimates are required for the adaptive mesh refinement procedure, and they try to give a good approximation of the numerical error and its distribution in the computational domain. With this information it is possible to compute an estimation of the ideal hydraulic diameter h_i distribution for a desired local error E_0. In the case of a second-order method in space, the following equations are valid:

$$|E| \sim Kh^2 \tag{16}$$

$$|E_0| \sim Kh_i^2 \tag{17}$$

where E is the error estimation and K is an unknown constant. After some algebraic manipulation,

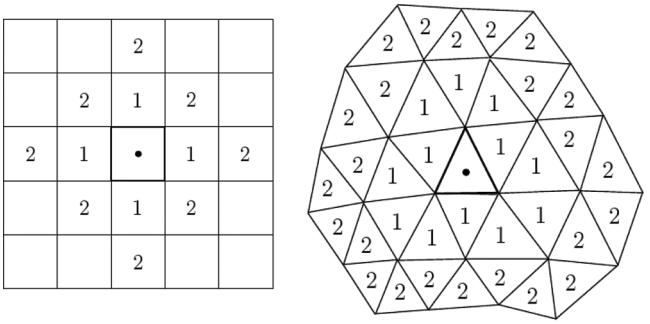
$$h_i \sim h\sqrt{E_0/E} \tag{18}$$

Ideally, Eq. (18) can be used to create adaptive grids with an approximate constant error distribution if the adaptive algorithm is combined with an automatic grid generator.

The adaptive procedure used in this work is based on the maximum value of the error estimate. The cells with an error higher than $\lambda \max(|E|)$ are selected for h-refinement, where λ is a factor that depends on the method's order. In the case of a second-order method this value λ is equal to 0.25, which is the reduction factor $(h_L/h_{L+1})^2$ of the local error for each grid refinement.

3.2. Regressions of High-Order Polynomial

The authors' own code, SOL, has the capability to make different types of regressions from the discrete cell values by solving a weighted least–squares problem. Different types of polynomials and cell sets can be used in these regressions. Figures 3*a* and 3*b* show examples of different cell sets: Figure 3*a* shows a cell set



(*a*) Cell sets by face. (*b*) Cell sets by vertex.

Figure 3. Different examples of cell sets used in the WLS polynomial regressions.

defined by the first and second cell's neighbors by face in a Cartesian grid, and Figure 3*b* shows another cell set defined by the first and second cell's neighbors by vertex in a grid with triangles.

These high-order regressions are required in the computation of the error estimate values, the WLS method has the versatility required in the context of unstructured grids. The computational stencil defined by the first and second cell's neighbors by vertex is used in these regressions, because this type of stencil provides more reliable information around the reference cell. The weighted function w_p, Eq. (12), is required in order to guarantee that the closest points to the reference have a greater contribution than the other points. The computational value from the reference cell is used to constrain the polynomial regression.

3.3. Taylor Series Truncation Error

The Taylor series expansion theorem states that any smooth function $\phi \in C^{\infty}$ in the vicinity of a point P can be approximated as a sum of its derivatives at P, in multidimensional form:

$$\phi(x_i; x_{i,P}) = \phi_P + \left(\frac{\partial \phi}{\partial x_i}\right)_P (x_i - x_{i,P}) + \frac{1}{2}\left(\frac{\partial^2 \phi}{\partial x_i \partial x_j}\right)_P (x_i - x_{i,P})(x_j - x_{j,P}) + \dots \quad (19)$$

where Einstein summation is implied above. Since the present FV numerical method has second-order accuracy, the discretization error is dominated by the second-order terms. We use the definition of the average error,

$$E_T = \frac{1}{2V_P} \int_P \left(\frac{\partial^2 \phi}{\partial x_i \partial x_j}\right)_P (x_i - x_{i,P})(x_j - x_{j,P})\, dV \quad (20)$$

where E_T is the error estimate based on a Taylor series. Assuming that the Hessian matrix is constant inside the cell P, which is a valid approximation for a

second-order method, the integration of Eq. (20) results in the following equation:

$$E_T = \frac{1}{2V_P} \left| \left(\frac{\partial^2 \phi}{\partial x_i \partial x_j} \right)_P \right| (M_{ij})_P \tag{21}$$

where $(M_{ij})_P$ is the inertia tensor of the cell P. The Hessian matrix values are computed from a regression made with a quadratic polynomial from the cell's first and second neighbors by vertex. Due to the assumption of linear variation inside the computational cells, zones with lower errors will have lower values of the Hessian matrix.

3.4. Residual Least Squares

A regression is done with a cubic-order polynomial and considering the cell's first and second neighbors by vertex. The new face's values and gradients are computed and compared with the values from the convective and diffusive schemes. One way to do this, is by recomputing new residual values, which indicate whether these new values satisfy the governing equations. The residual least squares (RLS) vector for each cell is computed by using the following formula:

$$\mathbf{E}_R = \frac{\sum_{f=1}^{F} U_f^n \mathbf{u}_f - \nu_f \sum_{f=1}^{F} (\nabla \mathbf{u})_f \cdot \mathbf{S}_f + (V_P/\rho) \nabla p^n}{a_p} \tag{22}$$

where the values $\mathbf{u}_f$ and $(\nabla \mathbf{u})_f$ are computed with a cubic regression, and a_p is the matrix value used for the momentum equations, which is required to give the RLS error estimate the same dimensions as the velocity or the transported variable. This error estimate gives an indication of the local error if the differences between the cubic profile and the numerical discretization affect the governing equations.

3.5. Grid Interface Correction

It is known from the literature that the interface between cells with different levels of refinement causes a degradation in the grid quality and consequently in the local numerical solution. The main problem is how to control the local error increase due to the lower grid quality at these interfaces. To solve this problem, a grid interface correction was used in the adaptive algorithm.

The selected cells for refinement are added to the same group if they are neighbors of each other. Each of these groups has an extended layer of one cell, since there is always an increase of the numerical error near the grid interface. If in one of these groups there exist cells with different levels of refinement, only the cells with the lowest level of refinement are refined. This procedure increases the number of refinement steps but prevents the accumulation of the grid interface in the same zone, resulting in grids with a lower number of cells for the same error level. Figure 4 shows a typical example of the cell selection algorithm, where each set of selected cells is highlighted with a bold line. In the last step of the algorithm, only the cells with the lowest level of refinement are refined, as it can be seen between the first and second grids of the example.

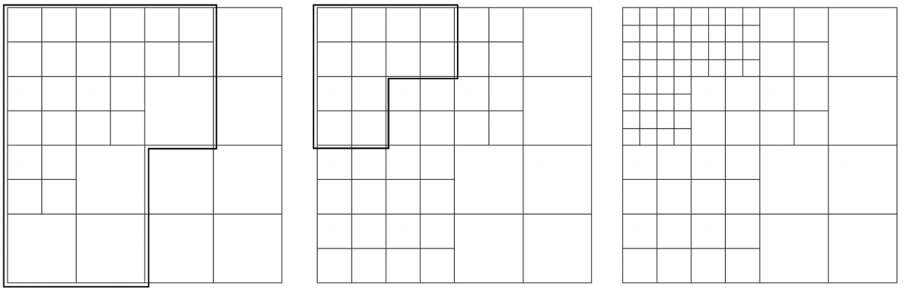


Figure 4. Example of the cell selection algorithm for refinement.

4. RESULTS

4.1. Analytic 2-D Cavity

To study the numerical error from the FV discretization at the adaptive grid interface, the analytic 2-D cavity problem was selected. This benchmark appeared, previously, in Shin et al. [39], Kobayashi et al. [40], and Pereira et al. [41], where details of the problem can be found. In this example, $\nu = 1.0$ was used, which corresponds to a Reynolds number of $\mathrm{Re} = UL/\nu = 1.0$.

The 2-D momentum equations of the flow are considered with an additional source term given by

$$B(x,y) = \frac{8}{\mathrm{Re}}\left[24\int \zeta_1(x)dx + 2\zeta_1'(x)\zeta_2''(y) + \zeta_1'''(x)\zeta_2(y)\right] - 64\left[\Phi_2(x)\Psi(y) - \zeta_2(y)\zeta_2'(y)\Phi_1(x)\right] \tag{23}$$

where $'$ is the differential operator and the functions $\zeta_1(x)$, $\zeta_2(y)$, $\Phi_1(x)$, $\Phi_2(x)$, and $\Psi(y)$ are defined by

$$\zeta_1(x) = x^4 - 2x^3 + x^2 \tag{24}$$

$$\zeta_2(y) = y^4 - y^2 \tag{25}$$

$$\Phi_1(x) = \zeta_1(x)\zeta_1''(x) - \zeta_1'(x)\zeta_1'(x) \tag{26}$$

$$\Phi_2(x) = \int \zeta_1(x)\zeta_1'(x)\,dx \tag{27}$$

$$\Psi(y) = \zeta_2(y)\zeta_2'''(y) - \zeta_2'(y)\zeta_2''(y) \tag{28}$$

The problem is solved in a square domain $[0;1]^2$, a Dirichlet boundary condition is imposed in all boundaries with zero velocity, except for the upper boundary ($y = 1.0$), where Eq. (29) is imposed:

$$u(x,1) = 16\zeta_1(x) \tag{29}$$

The analytical velocity field is independent of the Reynolds number and is given by

$$u(x,y) = 8\zeta_1(x)\zeta_2'(y) \tag{30}$$

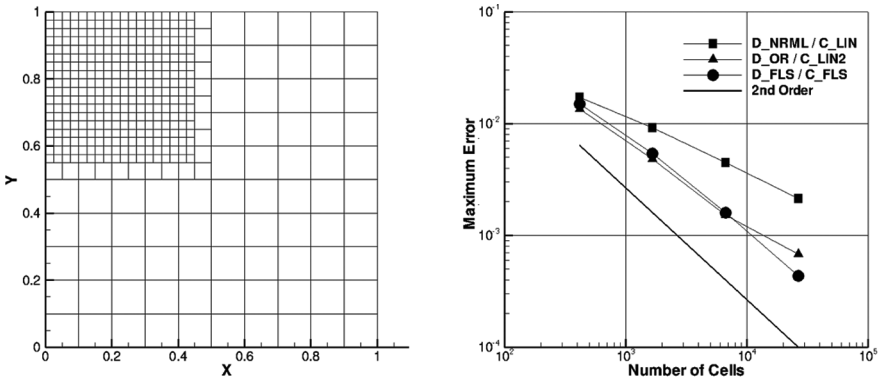
$$v(x,y) = -8\zeta_1'(x)\zeta_2(y) \tag{31}$$

and the pressure field is given by

$$p(x,y) = \frac{8}{\mathrm{Re}}\left[24\int \zeta_1(x)\,dx\zeta_2'''(y) + 2\zeta_1'(x)\zeta_2'(y)\right] + 64\Phi_2(x)[\zeta_2(y)\zeta_2''(y) - \zeta_2(y)\zeta_2(y)] \tag{32}$$

The second-order accuracy of the overall numerical scheme is verified for a prescribed adaptive grid in order to study the interface impact in the numerical accuracy, which is shown in Figure 5*a*. From this starting grid, all cells are refined three times for each pair of numerical schemes: normal diffusive and linear interpolation convective schemes which are typically used in Cartesian grids (D_NRML,C_LIN); overrelaxed diffusive (Jasak and Gosman [19]) and linear interpolation with gradient-correction convective scheme (D_OR,C_LIN2); and the face least-squares diffusive and convective schemes (D_FLS,C_FLS).

Figure 5*b* shows the maximum error for the three pairs of schemes used in this interface study. The mean error (which is not shown, for simplification) has second-order reduction for the three schemes sets, where D_FLS,C_FLS is the one with the lowest error. The maximum error also has second-order reduction except for the D_NRML,C_LIN schemes, which are close to first-order. From these results, the lowest numerical error was achieved with the D_FLS,C_FLS schemes. These were the schemes selected for the future subsections of this work.



(*a*) Initial grid used in the interface study. (*b*) Maximum error over the number of cells.

Figure 5. Analytical cavity: initial grid and maximum error over the number of cells for three pairs of diffusive and convective schemes.

4.2. Laplace Equation in an L-Shaped Domain

For this test case, the Laplace equation $\nabla^2\phi = 0.0$ was solved in an L-shaped domain $[-1, 1]^2 \backslash ([0, 1] \times [-1, 0])$. Dirichlet boundary conditions are prescribed at all boundaries, and the analytic solution is given by the following equation:

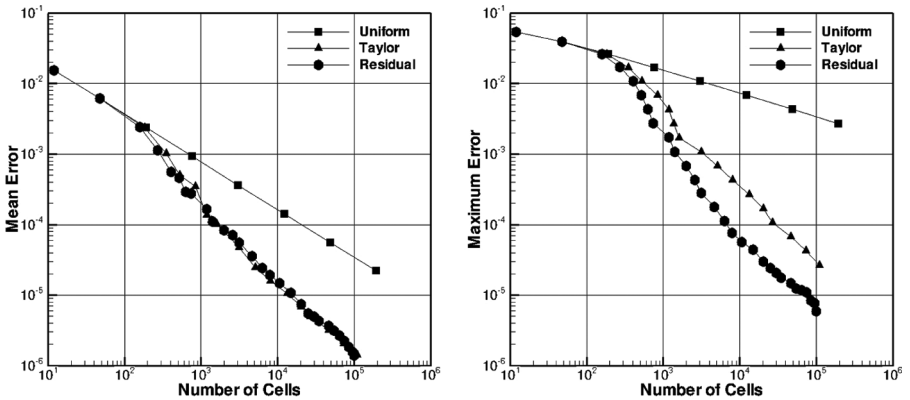
$$\phi(x, y) = r^{2/3} \sin(2\varphi/3) \quad \text{with } (x, y) = r(\cos \varphi, \sin \varphi) \tag{33}$$

The computations start with a Cartesian grid of 12 cells. Three types of refinement are applied to this grid: one with uniform refinement and two with the adaptive algorithm using the classic Taylor series and the RLS as error estimates. The goal is to study the main differences between the two error estimates and evaluate their effectiveness. Figure 6 shows the mean and maximum error after 15 levels of refinement for the Taylor series and 22 levels of refinement for the RLS error estimates.

For the uniform-grid case, the mean and maximum error slope have order of $4/3$ and $2/3$, respectively. Although at the singularity point $(x, y) = (0, 0)$ the analytic solution is zero, the analytic gradient value is infinite, which causes the method to have an order of accuracy lower than 2 for the uniform grids.

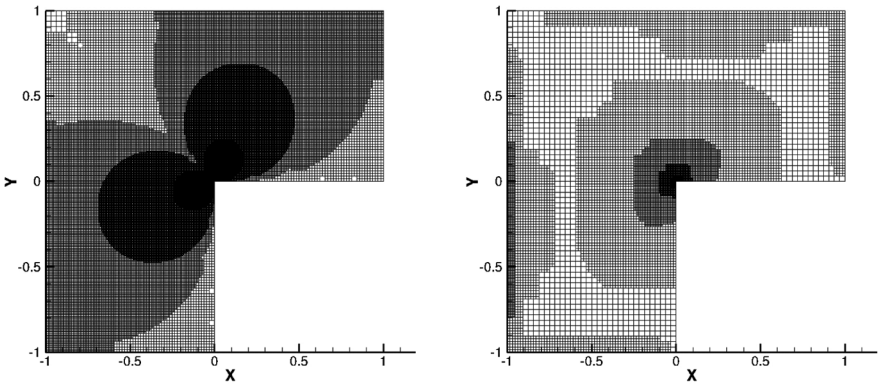
The mean error has second-order error decay for both error estimates of the adaptive grids. As a consequence, the mean error of the final adaptive grids is much lower than for the uniform grid (more than 10 times).

Figure 6*b* shows the maximum error for the three cases studied. The adaptive grid with the Taylor series (TS) estimate has maximum error 100 times lower than the error of the uniform grid, and the adaptive grid with the RLS estimate shows a substantial improvement over the TS estimate because the maximum error is 1,000 times lower. The ratio between the maximum and mean errors, which is a measure of the adaptivity efficiency, is 0.053 for the TS estimate and 0.2336 for the RLS estimate.



(*a*) Mean error. (*b*) Maximum error.

Figure 6. Laplace equation: mean and maximum error over number of cells for uniform and adaptive grids (squared grid example).

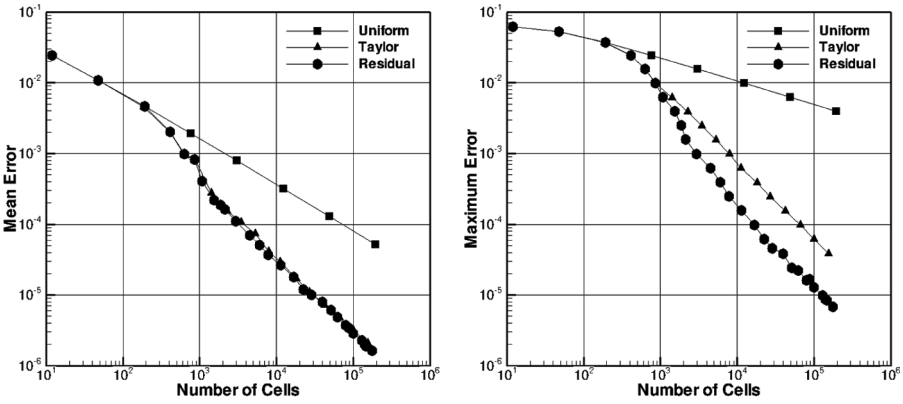


(*a*) Adaptive grid using the Taylor estimate. (*b*) Adaptive grid using the RLS estimate.

Figure 7. Laplace equation: adaptive grids with square cells with the Taylor and RLS estimates.

Figure 7 shows the two final adaptive grids obtained with the TS and RLS error estimates. The adaptive grid with TS has a circular pattern and more refined cells than the RLS adaptive grid. This happens due to the loss of accuracy of the TS error estimate at the interfaces of the adaptive grids. The adaptive grid with RLS has a lower number of cells and a rectangular pattern is observed with an over-estimation of the numerical error at the boundaries of the computational domain—see Figure 6*b*.

Figure 8 shows the mean and maximum error for triangular grids with a initial number of 12 cells. For the same number of cells, the error is slightly higher than for the squared-based ones. From the comparison between the adaptive and uniform grids, the same conclusions as for the previous case can be drawn.



(*a*) Mean error. (*b*) Maximum error.

Figure 8. Laplace equation: mean and maximum error over number of cells for uniform and adaptive grids (triangle grid example).

4.3. Convection-Diffusion Equation—Point Source in Cross-Flow

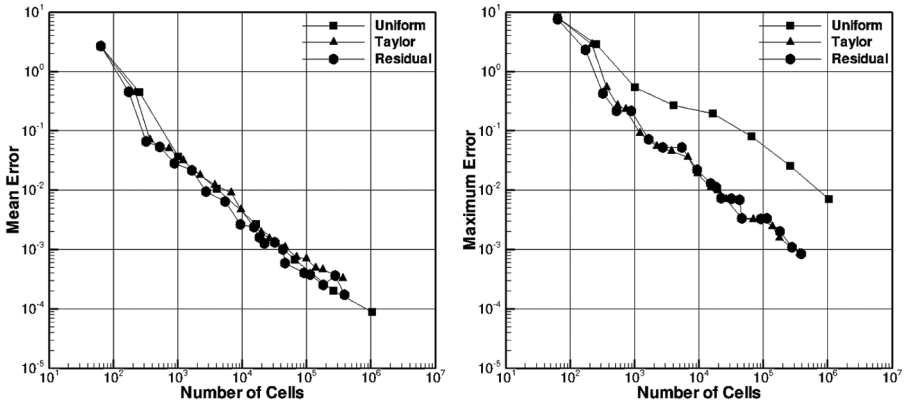
The convection-diffusion equation $U\partial\phi/\partial x = \Gamma\nabla^2\phi$ is frequently used for benchmarking of adaptive grid refinement algorithms; see, for example, Jasak and Gosman [19, 20]. For the point source in cross-flow case, the analytical solution is given by

$$\phi(x,y) = \frac{S}{2\pi\Gamma} K_0\left(\frac{U\sqrt{x^2+y^2}}{2\Gamma}\right) e^{0.5xU/\Gamma} \tag{34}$$

where $S = 16.67$ [ϕ/s] is the source magnitude, $\Gamma = 0.05$ [m^2/s] is the diffusion coefficient, $U = 1.0$ [m/s] is the imposed velocity on the x axis, and K_0 is the modified Bessel function of second kind and zero order. This problem is solved in a rectangular domain $[0.0, 4.0] \times [-0.5, 0.5]$ and the line-source origin is located at 0.05 m from the left boundary in order to avoid numerical problems from the singularity point. Dirichlet boundary conditions are prescribed at all boundaries, except for the right boundary ($x = 4.0$), where a null gradient is imposed. The starting grid has 16×4 cells and 20 levels of refinement are considered; Figure 9 shows the mean and maximum errors for the uniform and adaptive grids.

Figure 9*a* shows that the mean error displays the same slope for the three type of grids at study. The adaptive grids do not show improvements in the mean error when compared with the uniform grid. The TS criterion displays a mean error curve at a slightly upper position relative to the RLS and uniform cases. However, the adaptive grids have a lower maximum error than the uniform grid. The final adaptive grid has a maximum error 10 times lower than the uniform grid. In conclusion, the RLS has better efficiency than the TS, but the differences are smaller than in the previous case.

The error slope is not always constant due to the grid interface correction, which prevents the accumulation of the grid interfaces between different levels of refinement and avoids the loss of grid quality and of the solution overall accuracy.



(*a*) Mean error. (*b*) Maximum error.

Figure 9. Line source: mean and maximum error over number of cells for uniform and adaptive grids.

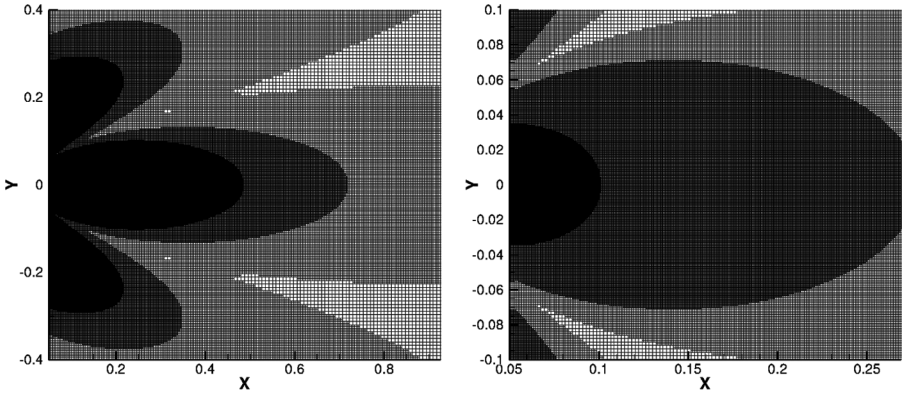


Figure 10. Line source: adaptive grid obtained for the RLS estimate after 20 refinement steps.

Figure 10 shows two different zooms of the grid obtained with the RLS estimate after 20 refinement steps. The cell zones with different refinement levels are well defined due the grid interface algorithm, which prevents the accumulation of refined cells from the previous refinement steps. The present results are a significant improvement over Jasak and Gosman [20] because of the grid interface correction and the different refinement decision algorithm, which creates much smoother adaptive grids with higher quality.

4.4. Point Jet

The point jet analytic case is obtained by freeing a point flow with a fixed momentum M_j in a large domain. This solution has a singularity point at the origin, and consequently the computational domain was located at 0.005 m downstream of the jet origin to avoid accuracy problems from the singularity point. The computational domain $[0.0, 2.5] \times [0.0, 0.1]$ is used for these computations, where the domain is extended in the x direction to avoid numerical errors near the outlet boundary. Dirichlet boundary conditions are applied at all boundaries, except for $x = 2.5$, where an outlet boundary condition is applied. The initial grid is covered with 100×4 cells.

The velocity field of this solution is defined by Eqs. (35)–(36):

$$u(x,y) = \frac{A}{B} x^{-1/3} \operatorname{sech}^2\left(\frac{y}{B} x^{-2/3}\right) \tag{35}$$

$$v(x,y) = -\frac{A}{3} x^{-2/3} \tanh\left(\frac{y}{B} x^{-2/3}\right) + \frac{2A}{3B} y x^{-4/3} \operatorname{sech}^2\left(\frac{y}{B} x^{-2/3}\right) \tag{36}$$

where A, B, and M_j are constants defined by

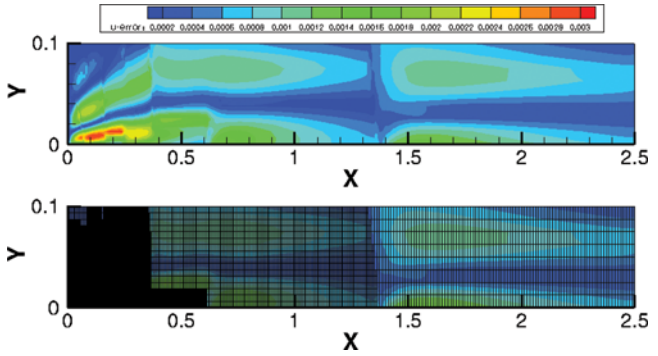


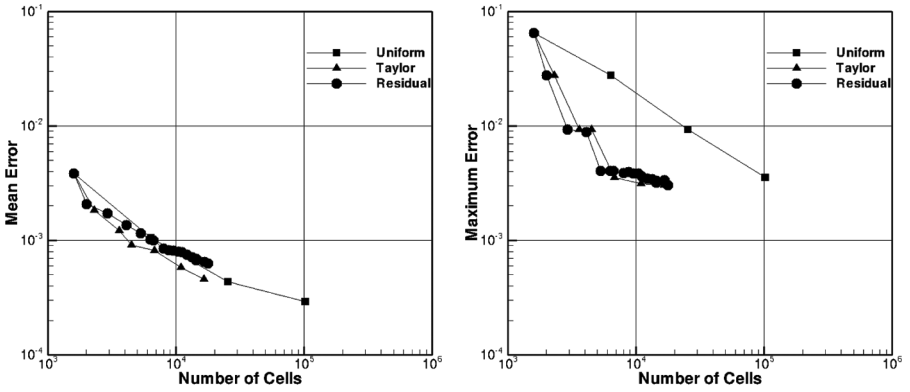
Figure 11. Point jet: grid obtained after 14 levels of refinement with RLS error estimate.

$$A = \left(\frac{9}{2}\nu M_j\right)^{1/3} \quad B = \left(\frac{48\nu^2}{M_j}\right)^{1/3} \quad M_j = U_0^2\, h = 0.1\left[\mathrm{m}^3/\mathrm{s}^2\right] \tag{37}$$

and ν is set to 0.0012 $[\mathrm{m}^2/\mathrm{s}]$.

In the case of incompressible flows, the two error estimates are applied to each component of the velocity field. Figure 11 shows the analytic error distribution and the grid after 14 levels of refinement using the RLS. The error is distributed along the cells with different levels of refinement and the cells with higher error are located near the singularity of the problem, which is the zone with higher numerical error for a uniform mesh. The discontinuities in the error field are located at the interfaces between the cells with different refinement levels which are caused by the volume discontinuity between these cells.

Figure 12 shows the mean and maximum errors as a function of the number of cells. For each level of refinement, the three grids have approximately the same



(*a*) Mean error. (*b*) Maximum error.

Figure 12. Point jet: mean and maximum error over number of cells for uniform and adaptive grids.

maximum error. Since both adaptive grids have fewer cells than the correspondent uniform grid, the curves from the adaptive grids are below the curve of the uniform grid, see Figure 12*b*.

4.5. Flow over a 2-D Cylinder

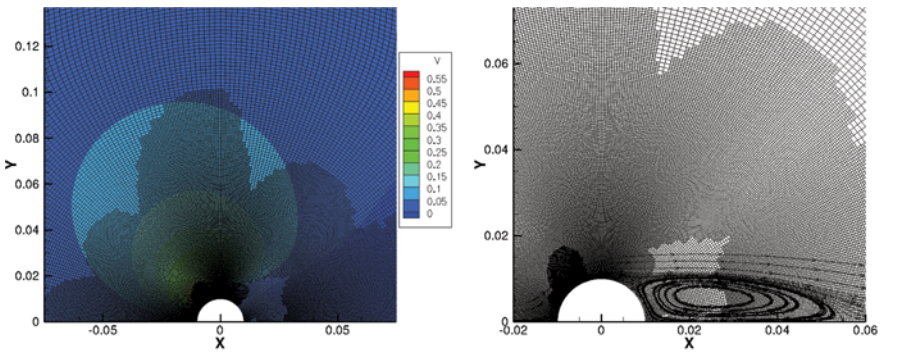
The Navier-Stokes equations were solved for the steady flow over a 2-*D* cylinder for a Reynolds number of 40. The Reynolds number for this flow is defined by $\mathrm{Re} = UD/\nu$, where D is the cylinder diameter. The computational domain has a length of 43 diameters in the longitudinal direction, with the cylinder located 13 diameters from the inlet. The domain extends up to 13 diameters in the vertical direction.

Figures 13*a* and 13*b* show respectively the V-velocity contour plot and the streamlines for the adaptive mesh obtained after 6 levels of refinement using the RLS error estimate. Different levels sets of the error distribution can be observed in this flow due to the different cell refinement levels. The refinement zone is located in a circle above the cylinder where the v velocity is maximum; this detail can be observed in the grid zoom (Figure 13*b*). The flow reattachment length is equal to 2.21 D which is consistent with the value 2.18 D from Calhoun [42].

4.6. 3-D Flow around a Squared Cylinder in a Confined Channel

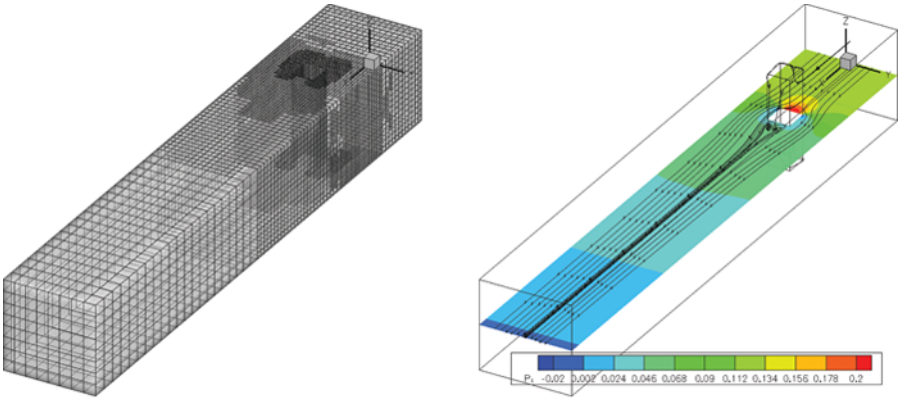
The 3-D Navier-Stokes equations were solved for the case of a confined flow in a channel with a squared cylinder when the Reynolds number is equal to 20. For this subcritical Reynolds number the flow remains steady, without the onset of a Karman street. This problem was previously studied by Schafer and Turek [43] and by Braack and Richter [44], where details of the computational domain and of the boundary conditions can be found.

The starting grid has 2,912 cells. Figures 14 and 15 shows the adaptive mesh obtained after 7 levels of refinement with 83,706 cells, where the streamlines and



(*a*) V-velocity contour plot. (*b*) Flow streamlines and adaptive grid zoom.

Figure 13. Cylinder flow Re = 40: contour plot of the v velocity and streamlines in the cylinder wake after 6 levels of refinement.



(*a*) Squared cylinder - 3D grid view. (*b*) Squared cylinder - pressure contour plot.

Figure 14. Adaptive grid for squared cylinder level of refinement 6: 3-D view and pressure contour plot.

the pressure contour plots can be observed. The adaptive grid divides the domain into four zones where the cells have different refinement levels. The zone with the highest refinement level includes the four squared corners, which are considered singularities points of this problem.

4.7. Flow over a Sphere

Two initial meshes were made, one with hexahedron and another with tetrahedron cells, in order to calculate the three-dimensional flow over a sphere at $\mathrm{Re} = 200$. The initial hexahedron grid has 46,800 cells, and its domain has a cylindrical form; the computational domain of the initial tetrahedral grid is a squared prism with $39 \times 13 \times 13$ diameters comprising 126,182 cells.

Figure 16 shows the obtained adaptive grids for two levels of refinement, with both the hexahedral and tetrahedral grids. The final hexahedrical and tetrahedrical meshes have 1,331,216 and 2,707,026 cells, respectively, which corresponds to meshes with 55.5% and 66.5% less than the uniform refinement case. Both adaptive grids are refined near the sphere wall and in the flow's wake, which are the primary features of this problem. The cone formed by the refined cells in the flow's wake is bigger in the tetrahedral grid, since the cells that are far away

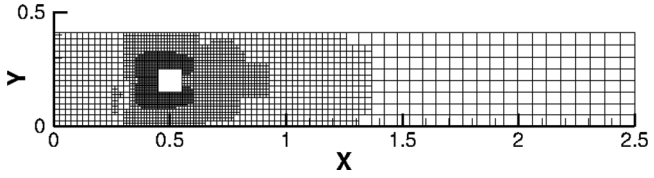


Figure 15. Adaptive grid for squared cylinder level of refinement 6: 2-D plot.

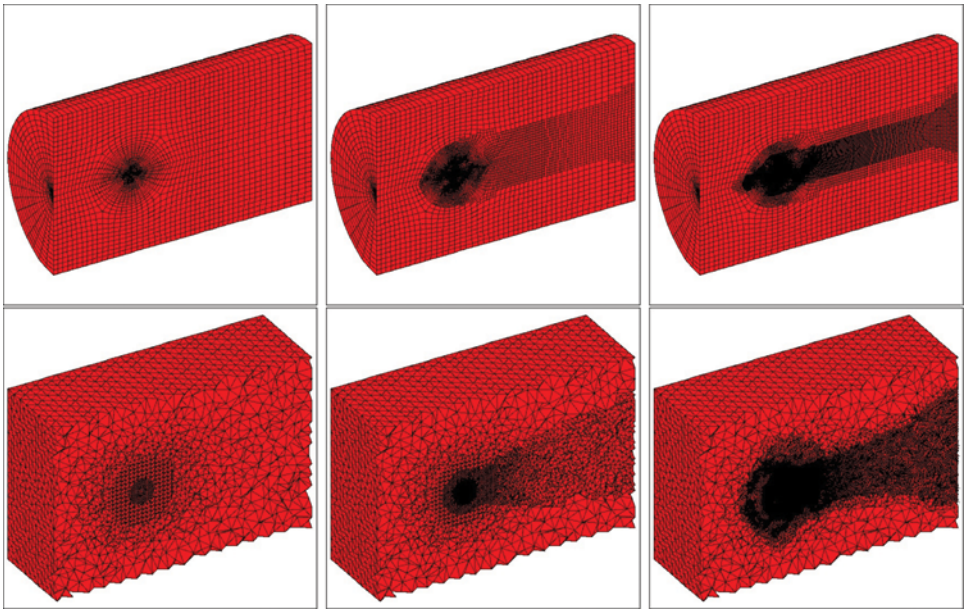


Figure 16. Example of refinement in the sphere flow with hexahedral and tetrahedral grids.

from the sphere have a higher hydraulic diameter or local grid length than in the hexahedral grid.

5. CONCLUSIONS

In this article, we present a framework for goal-oriented *a posteriori* error estimate for finite-volume methods. The residual least-squares error estimate has beeen shown to be suitable for FVM adaptive refinement on unstructured grids. The RLS error estimate is applicable to arbitrary unstructured grids, and its calculation is strongly coupled to the governing equations and the grid quality.

The second-order FV discretization is obtained considering the adaptive grid as an unstructured one and using a face-centered weighted least-squares diffusive and convective schemes. These schemes maintain second-order accuracy despite the mesh non-orthogonality and skewness deviation that occurs at the grid interface between cells of different levels of refinement.

To demonstrate the validity of the *a posteriori* error estimate, we conducted numerical experiments for several equations with analytical solution. The proposed adaptive algorithm for FVM is independent of user-defined parameters and can deal with the problem of the grid quality loss at the grid interface. It produces smoother adaptive grids and with higher quality, making it an alternative to other algorithms existing in the literature.

The adaptive grid refinement method has been extended to three-dimensional Navier-Stokes incompressible laminar-flow simulations. The proposed adaptive error estimate is expected to be particularly beneficial for three-dimensional Navier-Stokes applications.

FUNDING

This work was supported by Fundação para a Ciência e Tecnologia (FCT - Portugal) under the project PEst-OE/EME/LA0022/2013. The first author would like to thank for the support received during his PhD by the Fundação para a Ciência e Tecnologia (FCT - Portugal) grant SFRH/BD/48150/2008 co-financed by POPH/FSE.

REFERENCES

1. D. J. Mavriplis, Adaptive Meshing Techniques for Viscous Flow Calculations on Mixed Element Unstructured Meshes, *Int. J. Numer. Meth. Fluids*, vol. 24, pp. 93–111, 2000.
2. M. Ainsworth and J. T. Oden, A Posteriori Error Estimation in Finite Element Analysis, *Comput. Meth. Appl. Mech. Eng.*, vol. 148, pp. 1–88, 1997.
3. T. Gratsch and K. Bathe, A Posteriori Error Estimation Techniques in Practical Finite Element Analysis, *Comput. Struct.*, vol. 83, pp. 235–265, 2005.
4. K. Segeth, A Review of Some a Posteriori Error Estimates for Adaptive Finite Element Methods, *Math. Comput. Simul.*, vol. 80, pp. 1589–1600, 2008.
5. R. Verfurth, *A Review of Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, John Wiley, Newyork, 1996.
6. T. Plewa, T. Linde, and V. G. Weirs, *Adaptive Mesh Refinement—Theory and Applications*, Springer-Verlag, Berlin, 2005.
7. J. Peraire, M. Vahdati, and K. Morgan, Adaptive Remeshing for Compressible Flow Computations, *J. Comput. Phys.*, vol. 72, pp. 449–466, 1987.
8. M. Ainsworth and J. Oden, A Unified Approach to a Posteriori Error Estimation Using Element Residual Methods, *Numer. Math.*, vol. 65, pp. 23–50, 1993.
9. I. Babuska and A. Miller, A Feedback Finite Element Method with a Posteriori Error Estimation. Part I. The Finite Element Method and Some Basic Properties of the a Posteriori Error Estimator, *Comput. Meth. Appl. Mech. Eng.*, vol. 61, 1987.
10. P. Morin, R. Nochetto, and K. Siebert, Convergence of Adaptive Finite Element Methods, *SIAM Rev.*, vol. 44, pp. 631–658, 2002.
11. R. Verfurth, A Posteriori Error Estimation and Adaptive Mesh Refinement Techniques, *J. Comput. Appl. Math.*, vol. 50, pp. 67–83, 1994.
12. L. Machiels, J. Peraire, and A. T. Patera, A Posteriori Finite Element Output Bounds for the Incompressible Navier-Stokes Equations: Application to a Natural Convection Problem, *J. Comput. Phys.*, vol. 172, pp. 401–425, 2001.
13. K. J. Fidkowski and D. L. Darmofal, Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics, *AIAA J.*, vol. 49, pp. 673–694, 2011.
14. D. A. Venditti and D. L. Darmofal, Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows, *J. Comput. Phys.*, vol. 187, pp. 22–46, 2003.
15. M. J. Berger and J. Oliger, Adaptive Mesh Refinement Hyperbolic Partial Differential Equations, *J. Comput. Phys.*, vol. 53, pp. 484–512, 1984.
16. M. C. Thompson and J. H. Ferziger, An Adaptive Multigrid Technique for the Incompressible Navier-Stokes Equations, *J. Comput. Phys.*, vol. 82, pp. 94–121, 1989.
17. D. C. Haworth, E. L. Thary, and M. S. Huebler, A Global Approach to Error Estimation and Physical Diagnostics in Multidimensional Fluid Dynamics, *Int. J. Numer. Meth. Fluids*, vol. 17, pp. 75–97, 1993.
18. S. Muzaferija and D. Gosman, Finite-Volume CFD Procedure and Adaptive Error Control Strategy for Grids of Arbitrary Topology, *J. Comput. Phys.*, vol. 138, pp. 766–787, 1997.

19. H. Jasak and A. D. Gosman, Automatic Resolution Control for the Finite Volume Method, Part 1: A-Posteriori Error Estimates, *Numer. Heat Transfer B*, vol. 38, pp. 237–56, 2000.
20. H. Jasak and A. D. Gosman, Automatic Resolution Control for the Finite Volume Method, Part 2: Adaptive Mesh Refinement and Coarsening, *Numer. Heat Transfer B*, vol. 38, pp. 257–71, 2000.
21. H. Jasak and A. D. Gosman, Automatic Resolution Control for the Finite Volume Method, Part 3: Turbulent Flow Applications, *Numer. Heat Transfer B*, vol. 38, pp. 273–90, 2000.
22. D. Ait-Ali-Yahia, G. Baruzzi, W. G. Habashi, M. Fortin, J. Dompierre, and M. G. Vallet, Anisotropic Mesh Adaptation: Towards User-Independent, Meshindependent and Solver-Independent CFD. Part I: General Principles, *Int. J. Numer. Meth. Fluids*, vol. 32, pp. 725–744, 2002.
23. H. Jasak and A. D. Gosman, Residual Error Estimate for the Finite-Volume Method, *Numer. Heat Transfer B*, vol. 39, pp. 1–19, 2001.
24. F. Juretic, Error Analysis in Finite Volume CFD. Ph.D. thesis, Imperial College, University of London, 2004.
25. S. Nicaise, A Posteriori Error Estimations of Some Cell-Centered Finite Volume Methods, *SIAM J. Numer. Analy.*, vol. 43, pp. 1481–1503, 2006.
26. H. Jasak and A. D. Gosman, Elemental Residual Error Estimate for the Finite Volume Method, *Comput. Fluids*, vol. 32, pp. 223–248, 2003.
27. M. J. Aftosmis and M. J. Berger, Multilevel Error Estimation and *h*-Refinement for Cartesian Meshes with Embedded Boundaries, 40th *AIAA Aerospace Sciences Meeting and Exhibit 14–17 January, Reno NV, 2002–0863*, 2002.
28. A. Hay and M. Visonneau, Adaptive Finite-Volume Solution of Complex Turbulent Flows, *Comput. Fluids*, vol. 36, pp. 1347–1363, 2007.
29. S. Karni and A. Kurganov, Local Error Analysis for Approximate Solutions of Hyperbolic Conservation Laws, *Adv. Comput. Math.*, vol. 178, pp. 79–99, 2005.
30. C. Roy and M. Sinclair, On the Generation of Exact Solutions for Evaluating Numerical Scheme and Estimating Discretization Error, *J. Comput. Phys.*, vol. 228, pp. 1790–1802, 2009.
31. N. Ganesh, Nikhil V. Shende, and N. Balakrishnan, R-Parameter: A Local Truncation Error Based Adaptive Framework for Finite Volume Compressible Flow Solvers, *Comput. Fluids*, vol. 38, pp. 1799–1822, 2009.
32. N. Ganesh and N. Balakrishnan, A *h*-Adaptive Algorithm Using Residual Error Estimates for Fluid Flows, *Commun. Comput. Phys.*, vol. 13, pp. 461–478, 2013.
33. D. Martin and P. Colella, A Cell-Centered Adaptive Projection Method for the Incompressible Euler Equations, *J. Comput. Phys.*, vol. 163, pp. 271–312, 2000.
34. D. Martin, P. Colella, and D. Graves, A Cell-Centered Adaptive Projection Method for the Incompressible Navier-Stokes Equations in Three Dimensions, *J. Comput. Phys.*, vol. 227, pp. 1863–1886, 2008.
35. S. V. Patankar and D. B. Spalding, A Calculation Procedure for Heat, Mass and Momentum Transfer in Three Dimensional Parabolic Flows, *Int. J. Heat Mass Transfer*, vol. 15, p. 1787, 1972.
36. C. M. Rhie and W. L. Chow, Numerical Study of the Turbulent Flow past an Airfoil with Trailing Edge Separation, *AIAA J.*, vol. 21, pp. 1525–1532, 1983.
37. J. P. Magalhães, D. M. S. Albuquerque, J. M. C. Pereira, and J. C. F. Pereira, Adaptive Mesh Finite-Volume Calculation of 2D Lid-Cavity Corner Vortices, *J. Comput. Phys.*, vol. 243, pp. 365–381, 2013.
38. J. H. Ferziger and M. Peric, *Computational Method for Fluid Dynamics*, Springer-Verlag, Berlin/New York, 1996.
39. T. M. Shih, C. H. Tan, and B. C. Hwang, Effects of Grid Staggering on Numerical Schemes, *Int. J. Numer. Meth. Fluids*, vol. 9, p. 193, 1989.

40. M. Kobayashi, J. M. C. Pereira, and J. C. F. Pereira, A Conservative Finite-Volume Second-Order Accurate Projection Method on Hybrid Unstructured Grids, *J. Comput. Phys.*, vol. 150, pp. 40–75, 1999.
41. J. M. C. Pereira, M. Kobayashi, and J. C. F. Pereira, A Fourth-Order-Accurate Finite Volume Compact Method for the Incompressible Navier-Stokes Solutions, *J. Comput. Phys.*, vol. 167, pp. 217–243, 2001.
42. D. Calhoun, A Cartesian Grid Method for Solving the Two-Dimensional Stream Function-Vorticity Equations in Irregular Region, *J. Comput. Phys.*, vol. 176, pp. 231–275, 2002.
43. M. Schafer and S. Turek, Benchmark Computations of Laminar Flow Around a Cylinder, in *Flow Simulation with High Performance Computers II, DFG Priority Research Program Results*, vol. 52, pp. 547–66, 1996.
44. M. Braack and T. Richter, Solutions of 3D Navier-Stokes Benchmark Problems with Adaptive Finite Elements, *Comput. Fluids*, vol. 35, pp. 372–392, 2006.

APPENDIX: WEIGHTED LEAST-SQUARES METHOD

Assuming a base profile for the dependent variable ϕ in each cell P,

$$\phi(x, y, z; P) = \phi_P + \frac{\partial \phi}{\partial x}(x - x_P) + \frac{\partial \phi}{\partial y}(y - y_P) + \frac{\partial \phi}{\partial z}(z - z_P) + \dots \tag{38}$$

similar to a Taylor series expansion, which can be generalized as

$$Y = b_0 + \sum_{k=1}^{n} b_k X_k \tag{39}$$

where y is a linear combination of n variables X_k and the parameters b are unknowns. Using the polynomial approach of the Taylor series, the b parameters would be the gradient, Hessian, third, and higher derivatives. An observation is defined by

$$y_i \sqrt{w_{i,i}} = (b_0 + b_1 X_{1,i} + b_2 X_{2,i} + \dots + b_n X_{n,i}) \sqrt{w_{i,i}} + e_i \tag{40}$$

Considering all cell values from the stencil, an overdetermined system of equations is formed, given by $\mathbf{yW} = \mathbf{XbW} + \mathbf{e}$, where $\mathbf{y}$ is a vector with dimension $m \times 1$ containing the observations of y, $\mathbf{W}$ is a diagonal matrix with dimension $m \times m$ containing the weight function for each observation, $\mathbf{X}$ is a matrix $m \times (n+1)$ containing the values of the several explaining variables for each observation, $\mathbf{b}$ is a vector with dimension $n+1$ with the parameters to be determined, and $\mathbf{e}$ is the vector with the correlation error for each of the observations:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & X_{1,1} & X_{2,1} & \cdots & X_{n,1} \\ 1 & X_{1,2} & X_{2,2} & \cdots & X_{n,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{1,m} & X_{2,m} & \dots & X_{n,m} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix} \tag{41}$$

$$\mathbf{W} = \begin{bmatrix} \sqrt{w_{1,1}} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \sqrt{w_{m,m}} \end{bmatrix} \tag{42}$$

The correlation error is defined as the difference between the observed $\mathbf{y}$ and the estimated value $\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{b}}$:

$$e_i = y_i - \hat{y}_i \tag{43}$$

The solution vector $\hat{\mathbf{b}}$ obtained by the weighted least-squares method is the one that minimizes the sum

$$\min_{\hat{\mathbf{b}}} \sum_{i=0}^{m} w_i^2 e_i^2 \quad \Leftrightarrow \quad \min_{\hat{\mathbf{b}}} (\mathbf{W}^T \mathbf{e}^T)(\mathbf{We}) \tag{44}$$

The least-squares solution of this system is obtained by forming the normal system

$$\mathbf{X}'\mathbf{WWy} = \mathbf{X}'\mathbf{WWXb} + \mathbf{X}'\mathbf{We} \tag{45}$$

or (by neglecting the residuals)

$$\mathbf{b} = (\mathbf{X}'\mathbf{WWX})^{-1}\mathbf{X}'\mathbf{WWy} \tag{46}$$

If no weight function is used, the previous equation becomes:

$$\mathbf{b} = \underbrace{(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'}_{\mathbf{X}_{\text{left}}^{-1}}\mathbf{y} \tag{47}$$

where $\mathbf{X}_{\text{left}}^{-1}$ is the left inverse of the rectangular matrix $\mathbf{X}$.