Applied Mathematical Modelling 40 (2016) 8872–8885

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Polynomial interpolation with repeated Richardson extrapolation to reduce discretization error in CFD

Carlos Henrique Marchi [a,*], Márcio André Martins [b], Leandro Alberto Novak [a], Luciano Kiyoshi Araki [a], Márcio Augusto Villela Pinto [a], Simone de Fátima Tomazzoni Gonçalves [a], Diego Fernando Moro [c], Inajara da Silva Freitas [c]

[a] *Laboratory of Numerical Experimentation (LENA), Department of Mechanical Engineering (DEMEC), Federal University of Paraná (UFPR), Caixa postal 19040, CEP 81531-980, Curitiba, PR, Brazil*
[b] *Rua Padre Salvador, 875 – Santa Cruz, Caixa Postal 730, CEP 85015-430, Guarapuava, PR, Brazil*
[c] *Rua XV de Novembro, 1299, Centro, CEP 80.060-000, Curitiba, PR, Brazil*

ARTICLE INFO

Article history:
Received 22 April 2014
Revised 14 March 2016
Accepted 18 May 2016
Available online 28 May 2016

Keywords:
Richardson extrapolation
Discretization error
Polynomial interpolation
Poisson equation
Burgers equation
Navier–Stokes equation

ABSTRACT

The goal of the present study is to present and test a novel numerical procedure for reducing the discretization error associated with several types of variables of interest in basic Computational Fluid Dynamics (CFD) problems. Variables of interest are classified into five types according to their locations on various grids. According to the current literature, Repeated Richardson Extrapolation (RRE) performs well for only one of the five types of variable, i.e., for global variables or those that otherwise have fixed nodal positions on different grids. RRE does not perform well for the remaining four variable types. Because of this limitation, in this work, polynomial interpolation is applied to various numerical solutions obtained on different grids, followed by RRE. Four problems are used to test the proposed procedure, one linear and three non-linear based on the following equations: 1D Poisson, 2D Burgers and 2D Navier–Stokes. These equations are discretized using the Finite Difference method with approximations of second- and fourth-order accuracy and the Finite Volume method with approximations of first- and second-order accuracy. Polynomial interpolation functions for one- and two-dimensional domains are adopted, and optimization techniques are also adopted in some cases. The discretization error is significantly reduced, and the order of accuracy is also increased: for example, based on a second-order scheme with an error of 1.4×10^{-6}, we obtain 2.1×10^{-27} using six extrapolations on a grid with 1460 elements and an order of accuracy of 14.5. The computational effort (CPU time and memory usage) needed to obtain the solution at a given level of numerical error or using a specific grid is also significantly reduced.

* Corresponding author. Tel.: +55 41 3361 3126; fax: +55 41 3361 3701.
E-mail addresses: chmcfd@gmail.com, machi@pq.cnpq.br (C.H. Marchi), mandre@unicentro.br (M.A. Martins), leandro.novak@ufpr.br (L.A. Novak), lucaraki@ufpr.br (L.K. Araki), marcio_villela@ufpr.br (M.A.V. Pinto), simone.tg@ufpr.br (S.d.F.T. Gonçalves), difmoro@gmail.com (D.F. Moro), inajarafreitas@gmail.com (I.d.S. Freitas).

http://dx.doi.org/10.1016/j.apm.2016.05.029
0307-904X/

1. Introduction

One of the greatest challenges of Computational Fluid Dynamics (CFD) is related to the level of accuracy of the numerical solutions. Although numerical error can never be entirely eliminated, it must be reduced as much as possible. Among all sources of numerical error, the *discretization error* (*Eh*) is usually the most significant one [1]. In the current literature, *Richardson Extrapolation* (RE) is often used to reduce *Eh* and increase the numerical solution's *order of accuracy*. As examples, the following authors applied RE and obtained sixth-order accuracy using a fourth-order scheme: Wang and Zhang [2] did so for the 2D Poisson equation, and Ma and Ge [3] and Wang and Zhang [4] did so for the 3D advection–diffusion equation.

The effect of applying RE can be made more powerful if it is applied recursively, such that each application represents another level of extrapolation. This procedure is known as *Repeated Richardson Extrapolation* (RRE) [5]. Erturk et al. [6] used RRE with two levels of extrapolation in the numerical solution of a classical CFD problem, namely, the two-dimensional steady incompressible lid-driven cavity flow, obtaining sixth-order accuracy based on a second-order approach. For the same problem, Marchi et al. [7] applied RRE with up to nine levels of extrapolation to several types of variables of interest. They ascertained that RRE performs well for some types of variables, performs modestly in the case of variables that involve extreme values (maxima and minima) and is ineffective at finding the coordinate positions of extrema. Burg and Erwin [8] reported similar difficulties in using RRE that are inherent to the process of grid refinement, i.e., in variables whose location depends on the grid that is adopted. In this case, the coordinate values of the variable of interest change for each grid.

RRE's poor performance for variables with coordinates that change with the grid has also been observed by Nicolas et al. [9] in a thermoconvective-flow problem due to the occurrence of a local extreme value for the functional that describes the temperature gradient, where the location of the extremum changes when the grid does. Fig. 1 illustrates this situation for the 1D heat-conduction problem with unit domain (Ω), for which the Finite Difference method (FDM) was employed to solve the Poisson equation, as detailed by Martins [10], in which three variables of interest are considered: (i) $T(1/2)$, which represents the temperature at the specific nodal point that corresponds to coordinate $X=1/2$, (ii) *Tmax*, which indicates the maximum temperature obtained in Ω, and (iii) the location of this maximum temperature (*Xmax*). '*Eh, T*max', '*Eh, X*max' and '*Eh, T*(1/2)' denote, respectively, the error on the solution for *Tmax, X*max' and *T*(1/2) obtained without using RRE. '*Em, T*(1/2)' denotes the error on the solution obtained using RRE. '*Em, Tmax*' is the numerical error associated with the result obtained using RRE for the computation of *Tmax*, and '*Em, Xmax*' is the error associated with the result obtained using RRE to identify the coordinate position of the maximum temperature.

It can be observed in Fig. 1 that the use of RRE significantly reduces the magnitude of *Eh* for *T*(1/2), i.e., '*Eh, T*(1/2)' is much larger than '*Em, T*(1/2)'. However, with regard to the computation of *Tmax*, it is evident that the effectiveness of RRE is severely hindered, i.e., the resulting magnitude of '*Em, Tmax*' is equivalent to that of '*Eh, T*(1/2)' and the result for '*Em, Xmax*' is even worse. Furthermore, '*Epm, Tmax*' and '*Epm, Xmax*' represent the errors on the solution for *Tmax* and its location after the application of the procedure proposed in this work. It is clear that the results thus obtained are qualitatively equivalent to those obtained for the variable *T*(1/2) using RRE, i.e., '*Em, T*(1/2)'.

The goal of the present work is to present and test a novel numerical procedure for reducing the discretization error of variables for which RRE performs poorly, as has been described in the literature. Four problems are used to test this proposed procedure, one linear and three non-linear. The chosen mathematical models are a 1D Poisson equation, a set of two 2D Burgers equations (modified with a specified source term), 2D Navier–Stokes equations (modified with a specified source term) and 2D Navier–Stokes equations for the classical CFD problem, namely, the two-dimensional steady

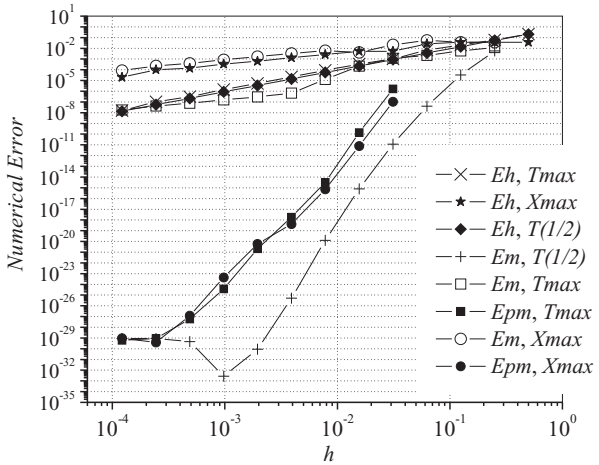

Fig. 1. RRE performance for three variables, 1D Poisson.

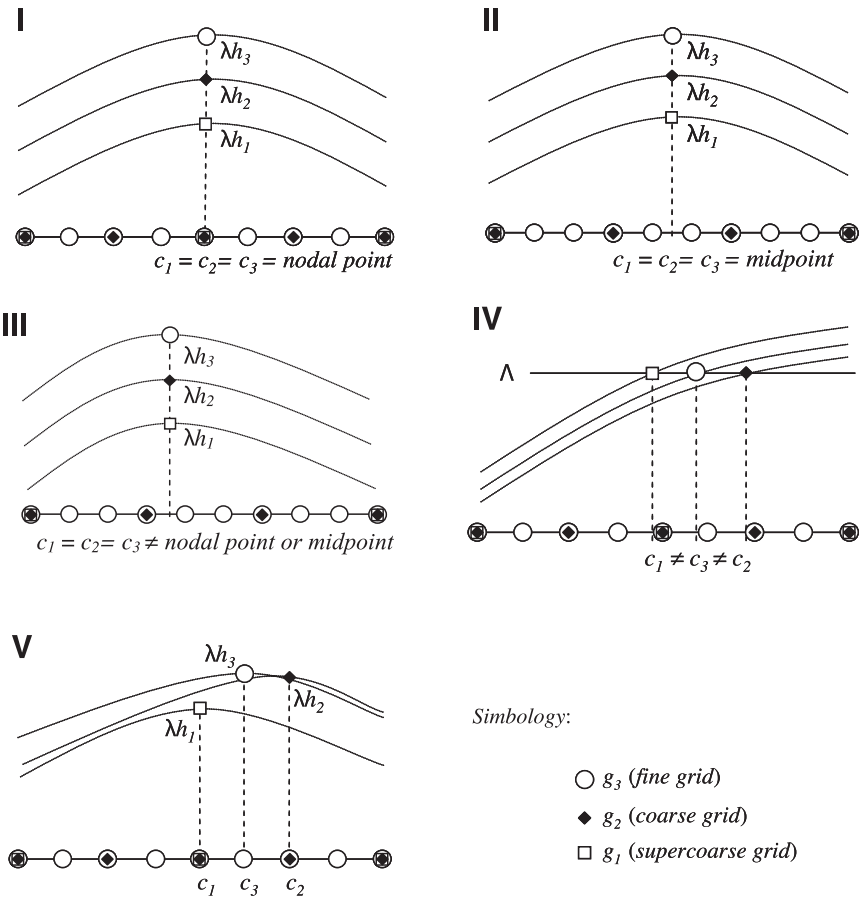

Fig. 2. Types of variables.

incompressible lid-driven cavity flow. The numerical solutions are obtained with the Finite Difference and Finite Volume methods (FDM and FVM) using numerical approximations of first-, second- and fourth-order accuracy. Although the solutions of cited problems are well known in the literature, these problems were chosen because they represent common problems in CFD, i.e., they involve pure diffusion, advection-diffusion and laminar flow. Moreover, in the first problem, the one-dimensional approach is justified by the possibility of using grids with millions of nodes, which permits the detailed analysis of the behavior of the asymptotic error.

This work is organized as follows: in Section 2, the theoretical foundations, the proposed procedure and the definition of the four problems are exposed; in Section 3, the results are presented; and the conclusion is exposed in Section 4.

2. Methodology

2.1. Types of variables

The use of RRE requires obtaining numerical solutions for each variable of interest on a set of distinct grids. However, as illustrated in Fig. 1, the performance of RRE is hindered in the case of variables whose coordinate values change when their solution is computed on different grids. Therefore, to cover the various cases that arise in the application of RRE, the present work proposes the classification of variables into five types, taking the process of grid refinement into account. These types are differentiated by the variable's locations, as described next and as shown in Fig. 2, when three levels of grid refinement are considered: g_1 (supercoarse), g_2 (coarse) and g_3 (fine); the symbols on the horizontal axis represent the nodal points, and the dotted lines (curves) are their respective numerical solutions (λh).

- *Type I*: a global variable, or a local variable whose coordinate (c) is known and coincides with a nodal point (Fig. 2I) in all grids (g) under consideration.
- *Type II*: a local variable for which c is known and is located at the midpoint of two nodes of g, i.e., the coordinate coincides with the arithmetic mean of the nodal coordinates (Fig. 2II).

- *Type III*: a local variable for which c is known and coincides neither with a nodal point of g nor with the midpoint of two such nodes; however, it does have a fixed position (Fig. 2III).
- *Type IV*: a local variable whose value (Λ) is known *a priori*. For this type of variable, the goal is to determine c (Fig. 2IV), which will have a different value for each grid ($c_1 \neq c_2 \neq c_3$) and does not necessarily coincide with any node.
- *Type V*: a variable for which c is unknown, i.e., a variable whose coordinate value changes across different grids (Fig. 2V). Maxima and minima are examples of this type of variable.

Although the representation adopted in Fig. 2 is presented on a one-dimensional domain, this classification (types I through V) is general and is also valid for multidimensional domains. For example, the coordinates of the non-nodal variables of type II are obtained by averaging the coordinates of four or eight nodal points in the two- and three-dimensional cases, respectively. The coordinates of variables of types III through V can be obtained via 1D, 2D or 3D polynomial interpolation of the nodal data.

In this work in particular, we consider one- and two-dimensional domains; the results for these types of variables are given in Section 3. The goal is to reduce *Eh*; in this sense, the performance of RRE for variables of type I has already been demonstrated in the literature in analyses of problems for which the performance is improved using RRE, and these analyses serve as references for the other variable types. Thus, in this study, the behavior for variables of types II, III and IV is analyzed both with RRE alone and with polynomial interpolation followed by RRE. Finally, for type V variables, additional optimization methods are used to obtain maximum and minimum values.

2.2. Numerical solution λh: obtained without interpolation or extrapolation, with error Eh

2.2.1. Numerical error

The numerical error $E(\lambda h)$ on a given variable of interest is defined as the difference between the *analytical solution* (Λ) and the *numerical solution* (λh), i.e.,

$$E(\lambda h) = \Lambda - \lambda h, \tag{1}$$

where $E(\lambda h)$ can arise from four sources [11]: truncation, iteration, round-off and programming. When the other sources are nonexistent or small in comparison with the truncation error, the *discretization error* (Eh) can be computed using Eq. (1). In fact, the absolute value of $E(\lambda h)$ is the value that is plotted in all figures presented in this manuscript.

Similar to the general equation for the truncation error, Eh is given by Roache [12]:

$$Eh = C_0 h^{p_0} + C_1 h^{p_1} + C_2 h^{p_2} + C_3 h^{p_3} + \cdots, \tag{2}$$

where the coefficients C_j, $j = 0, 1, 2, 3, \ldots$ are real-valued functions of the dependent variable and its derivatives but are independent of the grid spacing h, namely, the space between the nodes in the grid used in the discretization process. In the present work, we use uniform grids over a unit domain Ω, such that $h = 1/(N-1)$ (FDM) for N nodal points in 1D and $h = 1/N$ (FVM) for N nodal points in each coordinate axis of the $N \times N$ 2D grid.

By definition, the true order of accuracy corresponds to the exponents of h in Eq. (2), the set of which is denoted by p_T. These exponents are typically integer numbers such that $p_0 < p_1 < p_2 < p_3 < \cdots$. The smallest exponent p_0 is called the asymptotic order or *order of accuracy* of Eh. When $h \to 0$, the first term $C_0 h^{p_0}$ of the right-hand side of Eq. (2) is the primary component of Eh [11].

2.2.2. Problem 1: 1D Poisson equation

The first problem represents one-dimensional heat diffusion with a source, i.e., the classic Poisson equation, which is a linear differential equation. The mathematical model that represents Problem 1 is obtained from the energy-conservation equation by considering a continuous medium, the absence of flow, one-dimensional heat conduction, a steady state and constant properties, yielding:

$$\frac{d^2T}{dx^2} = S, \tag{3}$$

where T is the temperature, x is the space coordinate, $x \in \Omega = [0, 1]$ and S is the source term. To analyze the behavior of Eh, we use $S = S(x) = -9e^{3x}/5$, resulting in:

$$T(x) = -\frac{1}{5}e^{3x} + 3x + 2. \tag{4}$$

The boundary conditions for Eq. (3) are given by $T(0) = 9/5$ and $T(1) = -e^3/5 + 5$.

The numerical model used herein is obtained using the FDM [13], and two cases are formulated and solved: second-order numerical approximations using a central difference scheme (CDS-2, $p_0 = 2$) and a fourth-order compact scheme (CDS-4, $p_0 = 4$) [14]. The linear system obtained from the discretization process is solved using the TDMA (*Tridiagonal Matrix Algorithm*) [13] direct method, so there is no iteration error.

2.2.3. Problem 2: 2D Burgers equations

This problem represents the laminar flow of an incompressible fluid with constant properties in a lid-driven square cavity [15]. The mathematical model corresponds to the equations for the linear momentum in the x and y directions, given by:

$$\frac{\partial u^2}{\partial x}+\frac{\partial (uv)}{\partial y}=-\frac{\partial P}{\partial x}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right), \tag{5}$$

$$\frac{\partial (uv)}{\partial x}+\frac{\partial v^2}{\partial y}=-\frac{\partial P}{\partial y}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)-S(x,y,\mathrm{Re}), \tag{6}$$

where u and v are the components of the velocity vector in the x and y directions, respectively; Re is the Reynolds number ($\mathrm{Re}=u_\infty l\rho/\mu$), where u_∞ and l are the reference velocity and the length scale, respectively, ρ is the density and μ is the dynamic viscosity; Re = 1; and the source term (S) and the static pressure (P) are given by

$$S(x,y,\mathrm{Re})=-\frac{8}{\mathrm{Re}}\left[24F+2f'g''+f'''g\right]-64\left[F_2G_1-gg'F_1\right], \tag{7}$$

$$P(x,y,\mathrm{Re})=\frac{8}{\mathrm{Re}}\left[Fg'''+f'g'\right]+64F_2\left[gg''-\left(g'\right)^2\right], \tag{8}$$

where,

$$\begin{aligned}
&f=f(x)=x^4-2x^3+x^2\Rightarrow f'=4x^3-6x^2+2x\Rightarrow f''=12x^2-12x+2\Rightarrow f'''=24x-12,\\
&F=\int f(x)\,dx=0.2x^5-0.5x^4+x^3/3,\\
&F_1=f(x)f''(x)-\left[f'(x)\right]^2=-4x^6+12x^5-14x^4+8x^3-2x^2,\\
&F_2=\int f(x)f'(x)dx=0.5\left(x^4-2x^3+x^2\right)^2,\\
&g=g(y)=y^4-y^2\Rightarrow g'=4y^3-2y\Rightarrow g''=12y^2-2\Rightarrow g'''=24y,\\
&G_1=g(y)g'''(y)-g'(y)g''(y)=-24y^5+8y^3-4y.
\end{aligned}$$

The domain is $\Omega=\{(x,y)\in\Re^2: 0\le x\le 1 \text{ and } 0\le y\le 1\}$. The boundary conditions adopted for u and v are

$$u(x,0)=u(0,y)=u(1,y)=v(x,0)=v(0,y)=v(1,y)=v(x,1)=0;\ u(x,1)=16\left(x^4-2x^3+x^2\right). \tag{9}$$

The analytical solutions for the two unknowns of this problem, u and v, are:

$$u(x,y)=8\left(x^4-2x^3+x^2\right)\left(4y^3-2y\right), \tag{10}$$

$$v(x,y)=-8\left(4x^3-6x^2+2x\right)\left(y^4-y^2\right). \tag{11}$$

The numerical solution of this problem is obtained using the FVM [16]. The diffusion terms of Eqs. (5) and (6) are approximated using CDS-2, and the advection terms are approximated using two methods: (i) the UDS-1 (*Upwind Differencing Scheme*), which results in a first-order scheme ($p_0=1$), and (ii) the central difference through the deferred correction of the UDS-1, which results in a second-order scheme (CDS-2, $p_0=2$). The boundary conditions are applied using ghost cells. Both systems of equations, for u and v, are solved in a sequential procedure, but coupled by an iterative cycle. Each one of the systems of equations is solved using the lexicographic Gauss-Seidel method, by the use of geometric *multigrid* with a full approximation scheme (FAS), the full-weighting restriction, bilinear interpolation, the V-cycle algorithm and a *full multigrid* [17]. The process iterates twice as many times as necessary for the dimensionless residual of the set of algebraic equations, normalized to the initial estimate, to reach the machine-precision level; this guarantees that the iteration error is controlled and minimized.

2.2.4. Problem 3: 2D Navier–Stokes equations with known analytical solution

The adopted mathematical model involves the mass and momentum conservation laws (Navier–Stokes equations) in the context of the classical two-dimensional steady incompressible lid-driven cavity flow. The adopted assumptions also include: 2D laminar flow in x- and y-directions; constant μ (viscosity); and the neglect of other effects. In this case,

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0, \tag{12}$$

$$\rho\frac{\partial (u^2)}{\partial x}+\rho\frac{\partial (uv)}{\partial y}=\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)-\frac{\partial p}{\partial x}, \tag{13}$$

$$\rho\frac{\partial (uv)}{\partial x}+\rho\frac{\partial (v)^2}{\partial y}=\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)-\frac{\partial p}{\partial y}+S(x,y,\mathrm{Re}), \tag{14}$$

where p represents the pressure and S is the source term presented in Eq. (7) for the analytical solution obtained by the method of manufactured solutions, with Dirichlet boundary conditions given by Eq. (9). The domain is $\Omega=\{(x,y)\in\Re^2:$

$0 \le x \le 1$ and $0 \le y \le 1\}$. In order to solve this problem, we used $u_\infty = 1$ m/s and $l = 1$ m as the reference velocity and the length scale, respectively, $\rho = 1$ kg/m^3 is the density and $\mu = 1$ Pa.s is the dynamic viscosity. Re is the Reynolds number ($\mathrm{Re} = u_\infty l\rho/\mu$), so Re = 1. The analytical solution for the three unknowns of this problem, p, u and v, are given by Eqs. (8), (10) and (11).

The variables of interest for this problem involve the primitive variables u and v, as well as integral forms of these two. More specifically, one considers: $uc = u(1/2, 1/2)$; $\psi\min$ = the minimum value of the streamline function for $(x, y) \in [0, 1]$ and their x- and y-coordinates; $u\min$ = the minimum value of the u-profile at $x = 1/2$ and its respective y-coordinate; $v\max$ = the maximum value of the v-profile at $y = 1/2$ and its respective x-coordinate.

The numerical solution of the mathematical model described by Eqs. (12)–(14) was obtained by using [7,13,16]: (1) the FVM; (2) Central Differencing Scheme (CDS) [13] for approximations of diffusive and pressure terms; (3) CDS with deferred correction for approximations of advective terms; (4) Pressure–velocity coupling by the Semi Implicit Linked Equations Consistent Method (SIMPLEC); (5) co-located uniform grids; (6) boundary conditions applied by using ghost cells; (7) the Modified Strongly Implicit (MSI) method to solve the system of linear equations at each iteration; and (8) time-dependent (fully implicit scheme) formulation, in order to use time as a relaxation parameter for a better convergence rate of the discretized mathematical model. The obtained numerical model does not require the use of pressure boundary conditions [7] and the expressions obtained for internal nodal points are applied to the boundaries. For each grid, the analytical solution was employed as initial guess. The process iterates twice as many times as necessary for the dimensionless residual of the set of algebraic equations, normalized to the initial estimate, to reach the machine-precision level; this guarantees that the iteration error is controlled and minimized.

2.2.5. Problem 4: 2D Navier–Stokes equations with unknown analytical solution

The governing equations of Problem 4 are identical to the ones of Problem 3, Eqs. (12)–(14), excepted by the fact that $S = 0$ and the boundary conditions are given by:

$$u(x, 0) = u(0, y) = u(1, y) = v(x, 0) = v(0, y) = v(1, y) = v(x, 1) = 0;\ u(x, 1) = 1. \tag{15}$$

The analytical solution for this problem is unknown. However, it is widely used in literature [7] for comparisons of schemes and algorithms.

The same algorithm, numerical model and other data described in Section 2.2.4 are applied to Problem 4, excepted by the fact that the used initial guess was $u = v = 0$ and Re = 1000.

The variables of interest for this problem involve the primitive variables u and v, as well as integral forms of these two. More specifically, one considers: $uc = u(1/2, 1/2)$; $\psi\min$ = the minimum value of the streamline function for $(x, y) \in [0, 1]$ and their x- and y-coordinates; and $v\min$ = the minimum value of the v-profile at $y = 1/2$ and its respective x-coordinate.

2.3. Numerical solution λp: obtained using polynomial interpolation, with error Ep

When a problem is solved using numerical methods, the solution is obtained on nodal points determined by the adopted grid. However, in some cases, it may be necessary to obtain solutions at specific locations of the domain Ω that do not coincide with such points. In such an event, polynomial interpolation is a tool that can successfully be applied.

As will be shown in the result Section 3, polynomial interpolation of the results obtained at nodal points is used to reduce the *Eh* of variables of types II through V using RRE. Therefore, λp denotes a *numerical solution obtained using polynomial interpolation of degree p* of the nodal solutions λh, and *Ep* denotes the error associated with λp.

In general, with the knowledge of only a few points for which the value of a function f is known, it is possible to find a polynomial ξ that approximates the function inside some given domain. Among the many techniques for polynomial interpolation, the Newton method [18] is often used in practice.

2.3.1. Polynomial interpolation in 1D

In the one-dimensional case, Newton method is widely applied because it consists of an easy-to-implement recursive process [18], Newton's divided differences (NDD); it is used to determine a polynomial interpolation function ξ_p. An extreme point of ξ_p can be identified by considering the solution of the algebraic equation $d\xi_p/dx = 0$ (the derivative of ξ_p is equal to zero), for which the analytical solution is trivial when $p \le 2$. For $p \ge 3$, Newton's iterative method can be applied for the solution of non-linear algebraic equations [18].

When interpolating a function $f(x)$ by ξ_p in $[x_0, x_p]$, there is an associated error of $e(x) = f(x) - \xi_p(x),\ \forall\, x \in [x_0, x_p]$. For evenly spaced $x_0 < x_1 < x_2 < \cdots < x_p$ (with spacing h), the magnitude of $e(x)$ has an upper bound of [18]:

$$|e(x)| \le \frac{M_{p+1}}{4(p+1)} h^{p+1} = O(h^{p+1}), \tag{16}$$

where $M_{p+1} = \max |{f_p}^{(p+1)}(x)|$, the maximum value of the p+1th order derivative of ξ_p, with $x \in [x_0, x_p]$; i.e., when $f(x)$ is interpolated by ξ_p in $[x_0, x_p]$, the associated error is of at least p+1th order.

2.3.2. Polynomial interpolation in 2D

In the case of a two-dimensional domain, it is possible to obtain the NDD in 2D [19] relative to x and y. An extreme point of ξ_p (2D) can be identified within its domain. In this case, the *optimization problem* [20] can generally be posed as follows: given ξ_p: $\Omega \subset \Re^2 \rightarrow \Re$, we seek either the maximum value of $\xi_p(x,y)$ ($\max \xi_p(x, y)$) or the minimum value of $\xi_p(x,y)$ ($\min \xi_p(x,y)$). We consider the class of objective functions that are non-linear and differentiable, such that gradient-type methods can successfully be applied, especially polynomial functions for which the analytical expression of the gradient can easily be derived [20]. The Armijo line search [20] is often used for this purpose because of its efficiency and ease of implementation.

2.4. Numerical solution λm: obtained via repeated Richardson extrapolation, with error Em

Richardson Extrapolation (RE) can be used whenever an approximation technique having a predictable error term that can be written as a function of h is known [5]. For each value of $h \neq 0$, an expression for λh is obtained such that Λ can be approximated by:

$$\Lambda - \lambda h = k_0 h^{p_0} + k_1 h^{p_1} + k_2 h^{p_2} + k_3 h^{p_3} + \cdots, \tag{17}$$

for a set of unknown constants $k_0, k_1, k_2, k_3, \ldots$. Using RE, approximations can be combined to increase the resulting order of accuracy. Therefore, formulas can be combined for λh by varying h, i.e.,

$$\begin{cases} \lambda h = \Lambda - (k_0 h^{p_0} + k_1 h^{p_1} + k_2 h^{p_2} + k_3 h^{p_3} + \cdots) \\ \lambda(rh) = \Lambda - [k_0 (rh)^{p_0} + k_1 (rh)^{p_1} + k_2 (rh)^{p_2} + k_3 (rh)^{p_3} + \cdots] \end{cases} \Rightarrow \begin{cases} \lambda h = \Lambda - k_0 h^{p_0} + O(h^{p_1}) \\ \lambda(rh) = \Lambda - k_0 (rh)^{p_0} + O(h^{p_1}) \end{cases} \tag{18}$$
$$\Rightarrow \Lambda = \left[\lambda h + \frac{\lambda h - \lambda(rh)}{r^{p_0} - 1}\right] + O(h^{p_1}) = \lambda(h, rh) + O(h^{p_1}),$$

where the refinement ratio r is > 1, λh is the numerical solution obtained over a grid with spacing h between neighboring nodes, $\lambda(rh)$ is the numerical solution obtained over a grid with spacing rh between neighboring nodes, $\lambda(h, rh)$ is a combination of λh and $\lambda(rh)$ and $O(h^{p_1})$ indicates an error of the order p_1.

The process known as *Repeated Richardson Extrapolation* (RRE) consists of the repeated application of Eq. (18). However, p_0 is considered only when RRE is first applied, i.e., at the first extrapolation level. At further levels, the subsequent values of p_T are considered ($p_1, p_2, p_3, \ldots$). RRE becomes particularly simple when geometrically similar grids g, g-1 and g-2 are used [5], i.e., when G distinct grids are generated using a constant refinement ratio ($h_g = h,\ h_{g-1} = rh,\ h_{g-2} = r^2 h, \ldots$). In this case, the *numerical solution obtained using RRE* on a grid g, where g indicates the level of the grid, with m applications of RE is given by Marchi et al. [21]:

$$\lambda_{g,m} = \lambda_{g,m-1} + \frac{\lambda_{g,m-1} - \lambda_{g-1,m-1}}{r^{p_{m-1}} - 1}, \tag{19}$$

where $r = h_{g-1} \,/\, h_g$. Eq. (19) holds for $g = 2, \ldots, G$ and $m = 1, \ldots, g-1$. On any grid g, $m = 0$ corresponds to the numerical solution obtained without using RRE, λh or λp. For $m = 1$, this equation yields the first level of extrapolation with $p_{m-1} = p_0$. For subsequent levels of larger m, p_{m-1} corresponds to the subsequent values of p_T. These values can be determined a priori and confirmed through the concept of the effective order of accuracy (p_E) of the numerical error [21] using, for constant m,

$$(p_E)_{g,m} = \frac{\log\left[\frac{\Lambda - \lambda_{g-1,m}}{\Lambda - \lambda_{g,m}}\right]}{\log(r)}, \tag{20}$$

and, for variable m,

$$(p_E)_{g,m} = \frac{\log\left[\frac{\Lambda - \lambda_{g-1,m-1}}{\Lambda - \lambda_{g,m}}\right]}{\log(r)}, \tag{21}$$

where $g = 2,\ldots,\ G$ and $m = g - 1$.

2.4.1. Computation of λm

Inspection of Eq. (19) reveals that the solution obtained using RRE is in respect to $\lambda_{g,m}$, with several different levels of grid g and extrapolation m. However, the highest level of accuracy is obtained when the largest possible value of m is used in each grid, i.e., $m = g - 1$. Thus, for $g = 2, \ldots, G$, we consider $\lambda m = \{\lambda_{2,1}, \lambda_{3,2}, \ldots, \lambda_{g,g-1}, \ldots, \lambda_{G,G-1}\}$; then, Em denotes the discretization error of λm, computed as the difference between Λ and λm.

Still, in principle, it is valid to use Eq. (19) directly to obtain λm only for type I variables (see Fig. 2I). For the remaining types, II through V (see Fig. 2II through V), λp and λpm must be computed.

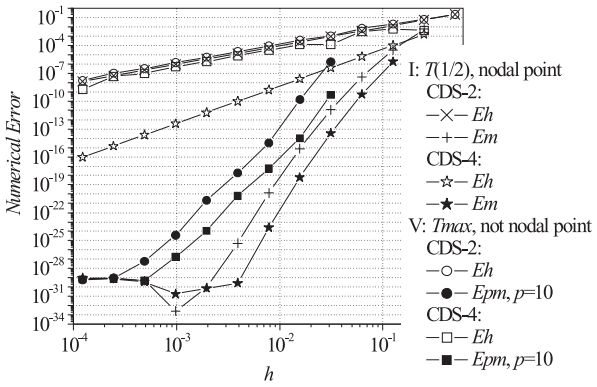

Fig. 3. Effect of the order of the numerical scheme on RRE, $T(1/2)$ and *Tmax*, 1D Poisson.

2.4.2. Computation of λp *and* λpm

This work primarily analyzes the use of RRE on numerical solutions λp obtained via polynomial interpolation over the nodal points. From λp, its numerical error *Ep* can then be computed in a manner analogous to Eq. (1), i.e., through the difference between Λ and λp.

When λp is available on G distinct grids, RRE can be applied via Eq. (19) to obtain λpm in a manner analogous to λm. Therefore, to determine λpm, 1D or 2D polynomial interpolation must first be applied followed by RRE over λp. The numerical error of λpm is thus denoted by *Epm* and is computed as the difference between Λ and λpm. The effective order of the error of λp and λpm can be computed using Eqs. (20) and (21).

3. Results

3.1. Results for Problem 1: 1D Poisson equation

The numerical solutions for Problem 1 are obtained using computational programs written by the authors in Fortran 90 and using the Intel Fortran compiler v. 11.1 with quadruple precision and are run on a computer with a 2.20 GHz dual-core AMD Athlon processor with 2GB of RAM hosting 64-bit Windows XP. Computations are performed on grids with refinement ratios of $r=2$ and 3. For $r=2$, $N=5$ nodal points are used for the coarsest grid and $N=16{,}777{,}217$ nodes for the finest, with a total of 22 grids; the plots presented here, however, go only as far as the grid with 8193 nodes, after which round-off errors become predominant. For $r=3$, $N=3$ nodal points are used for the coarsest grid and $N=9{,}565{,}937$ nodes for the finest, with a total of 15 grids. Analytical solutions for the variables of interest are obtained using Maple 10.0 with 32 decimal places of precision.

In Section 1, results for three variables of interest T(1/2), Tmax and Xmax obtained for the 1D Poisson equation were presented. Numerical studies included two manufactured solutions (an exponential and a sinusoidal ones), the use of FDM and second-order (CDS-2) and fourth-order (CDS-4) approximation schemes, as earlier studied by Marchi and Germer [22]. For both approximation schemes, the effect of applying RRE to the results obtained via cubic and quadratic spline interpolation, which were approximations obtained using the Fast Fourier Transform and Least Squares methods, was further examined. However, the best results were obtained using polynomial interpolation followed by RRE.

The order of the numerical scheme also affects the results obtained for λm and λpm. In Fig. 3, it can be seen that for the type I variable $T(1/2)$ (see Fig. 2I), the magnitude of *Em* is smaller for CDS-4 than for CDS-2. For the type V variable *Tmax* (see Fig. 2V), it is immediately apparent that the order of *Eh* for CDS-4 is degenerated; however, for polynomial interpolation of degree $p=10$, *Epm* exhibits better results for CDS-4 than for CDS-2. We believe that the degenerated behavior of p_E for the *Eh* of *Tmax* in the case of CDS-4 is related to the change in the coordinate position of the variable of interest, which constitutes another source of error that compromises the order of the adopted numerical scheme. In the case of the type V variable considered here, the error incurred in the computation of its coordinate value *Xmax* presents an analogous behavior. The results obtained for p_E are shown in Table 1, where it can be observed that the largest orders correspond to $|Epm, p=10|$; for *Tmax* and *Xmax*, these largest orders are ≈ 15 and ≈ 13, respectively.

3.2. Results for Problem 2: 2D Burgers equations

The numerical solutions for Problem 2 are obtained using computational programs written by the authors in Fortran 90 and using the Intel Fortran compiler v. 11.1 with quadruple precision and are run on a computer with two 3.47 GHz hexa-core Intel Xeon X5690 processors with 192GB of RAM hosting 64-bit Windows 7. Computations are performed on 13 grids with a refinement ratio of $r=2$. $N \times N = 4 \times 4$ nodes are used in the coarsest grid and $N \times N = 16,384 \times 16,384$ nodes in

Table 1
Order of the numerical error for type V variables, *Tmax* and *Xmax*, CDS-2, 1D Poisson.

Variable	*Grid*	$N=33$	$N=129$	$N=1025$
	h	3.13E−02	7.81E−03	9.77E−04
Tmax	p_E *for* $\|Em\|$	0.87	2.45	0.90
	p_E *for* $\|Epm,\ p=2\|$	2.26	2.78	3.00
	p_E *for* $\|Epm,\ p=10\|$	11.10	15.40	12.40
Xmax	p_E *for* $\|Em\|$	0.57	1.34	0.39
	p_E *for* $\|Epm,\ p=2\|$	1.01	4.21	2.02
	p_E *for* $\|Epm,\ p=10\|$	11.30	13.41	8.94

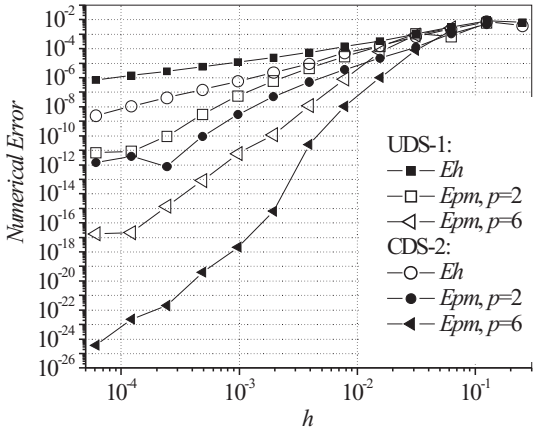

Fig. 4. Type V-2D variable, $\psi\min$, with 2D interpolation, 2D Burgers.

Table 2
Order of the numerical error for type V-2D variables, with 2D polynomial interpolation, $p=6$, 2D Burgers.

Variable	*Scheme*	*Grid*	32×32	256×256	8192×8192
		h	3.13E−02	3.91E−03	1.22E−04
$\psi\min$	CDS-2	p_E *for Eh*	1.77	1.99	1.90
		p_E *for Epm*	4.64	8.70	8.17
	UDS-1	p_E *for Eh*	1.53	1.13	1.01
		p_E *for Epm*	3.78	6.10	6.00
$x(\psi\min)$	CDS-2	p_E *for Epm*	4.97	5.81	9.62
	UDS-1	p_E *for Epm*	4.28	4.73	5.48
$y(\psi\min)$	CDS-2	p_E *for Epm*	2.69	6.50	8.09
	UDS-1	p_E *for Epm*	2.67	5.72	6.11

the finest one. The analytical solutions for the variables of interest are obtained using Maple 10.0 with 32 decimal places of precision.

Further analyses of type V variables are also presented, now taking a two-dimensional approach, i.e., both coordinate values in 2D Ω, x and y, are allowed to change as the grid is refined. The results obtained for $\psi\min$ (minimum value of stream function) of 2D Burgers equations are given in Fig. 4; the analytical solution of $\psi\min$ is $-1/8$. Similar behavior is observed in the *Epm* associated with the remaining variables of interest. The largest values of p_E for *Epm* are obtained using the higher-order numerical scheme and 2D polynomial interpolation of degree $p=6$, which is the highest degree considered in this work. For $\psi\min$, the value of order p_E associated with *Epm* increases up to 6.1 and 8.7 respectively for UDS-1 and CDS-2 (see Table 2).

The procedure for obtaining λpm proposed in this work requires very little computational effort, as discussed below, in comparison with computing λh at the same level of numerical error. Another aspect that should be considered is that the process of grid refinement is a common practice in the analysis of the convergence of solutions in numerical verification procedures. Based on the typical resource requirements of this common task, to obtain λpm, (i) λp must be obtained through polynomial interpolation, which implies post-processing that, in the most effort-consuming cases of $p=10$ (1D) and $p=6$ (2D), requires approximately 10^{-4} s and 2.8 kB of RAM memory independent of the grid being used, and (ii) λpm must be obtained by applying RRE using Eq. (19), which requires approximately 10^{-5} s and 7.1kB of RAM memory for all numerical solutions across 22 distinct grids; for a smaller number of grids, the cost is accordingly smaller.

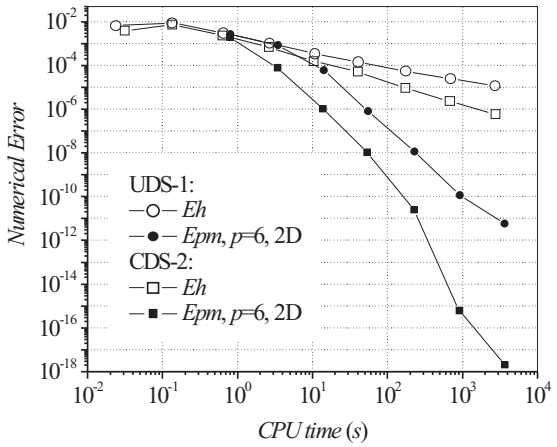

Fig. 5. Comparison among the CPU times required to obtain *Eh* and *Epm* using 2D polynomial interpolation for a type V-2D variable, ψmin, 2D Burgers.

Fig. 5 shows the magnitude of the numerical error associated with the type V variable ψmin of 2D Burgers equations plotted against the CPU time (in seconds) required for the calculation with (*Epm*) and without (*Eh*) the procedure proposed in this work, using both UDS-1 and CDS-2. *Epm* represents the numerical error resulting from RRE associated with the method of 2D polynomial interpolation and optimization. This approach is representative of the costliest case, i.e., the case that incurs the highest computational effort among all cases analyzed herein. The CPU time required to obtain *Epm* on a given grid is the sum of the CPU times over all coarser grids because they are necessary for the computation using Eq. (19). For example, the CPU time required to obtain *Epm* on the 128×128 grid also includes the CPU time required for the 64×64, 32×32, 16×16 and 8×8 grids, i.e., the CPU time required for computations on 5 grids.

According to Fig. 5, *Eh* (the error of the solutions obtained without RRE) decreases linearly (constant slope) as the CPU time increases, whereas the reduction in *Epm* is much more rapid, with a slope that increases with increasing CPU time. The level of reduction in the magnitude of the numerical error obtained for *Epm* ($<10^{-17}$, see Fig. 4) cannot be obtained for *Eh* because of the limitations of the computer used in this study. Moreover, for a given numerical-error magnitude, the CPU time required for the *Epm* calculation is much smaller than that for *Eh* (see Fig. 5). For example, an error magnitude of 10^{-6} can be obtained for *Eh* using CDS-2, and this requires $\approx 3.5 \times 10^3$ s; for the same level of error, *Epm* requires only $\approx$15 s, i.e., 1/233 of the time. Moreover, in terms of CPU time, the relative efficiency of the proposed procedure is increased when a requirement for a higher level of accuracy, i.e., a smaller numerical-error magnitude, is imposed.

3.3. Results for Problem 3: 2D Navier–Stokes equations with known analytical solution

The numerical solutions were obtained using computational codes written by the authors in Fortran 90 and using the Intel Fortran compiler v. 11.1 with double precision and were run on a computer with two 3.47 GHz hexa-core Intel Xeon X5690 processors with 192GB of RAM hosting 64-bit Windows 7. Computations were performed on 10 grids with a refinement ratio of $r=2$. $N \times N = 4 \times 4$ nodes were used in the coarsest grid and $N \times N = 2048 \times 2048$ nodes in the finest one. The analytical solutions for the variables of interest were obtained using Maple 10.0 with 32 decimal places of precision.

The result presented in Fig. 6 refers to the variable *uc*, i.e., to the local value obtained for *u* from Eqs. (12)–(14) at point (0.5; 0.5). Its analytical solution $-1/4$ is obtained from Eq. (10). This result for a type II variable serves as a reference for the remaining types of variables in the problem at hand, i.e., 2D laminar flow. The location of this variable is at the exact midpoint of four central nodal points in all considered grids. In this case, the use of 2D polynomial interpolation of degree $p=1$ (bilinear interpolation) might be considered. In this manner, λp is obtained in each grid and RRE is then applied as in Eq. (19) to obtain λpm. The results for *Ep* and *Epm* associated with *uc* are shown in Fig. 6. With regard to the values attained by p_E, *Ep* has a value that is equivalent to the order of accuracy (2) of the adopted numerical scheme (CDS-2), and the value of order p_E associated with *Epm* increases up to 7.2.

With regard to type V variables (see Fig. 2V), a one-dimensional analysis is first conducted, i.e., when computing λh on distinct grids of 2D Ω, only one of the two coordinate (x or y) values changes, while the other remains fixed. For this approach, the variable of interest is selected: u min1D – minimum value of the velocity profile u on the line $x=1/2$ (fixed x). Its analytical solution u min $1D = -\sqrt{6}/9$ is obtained from Eq. (10). This variable is intentionally located at points with irrational coordinate values. Therefore, they cannot correspond to any nodal point, and an approximate numerical solution must be obtained. To this end, λp is computed via polynomial interpolation. Figs. 7 and 8 show the results obtained for $u$min $1D$ and its $y(u\,\text{min}\ 1D)$ coordinate, respectively; the analytical solution of $y(u\,\text{min}\ 1D)$ is $\sqrt{6}/6$. Analogous results are obtained for all other variables. The best results for *Epm* are obtained for CDS-2 and polynomial interpolation of degree $p=10$, which is the highest degree used in the 1D approach. With regard to the values attained by p_E, *Ep* has a value that is

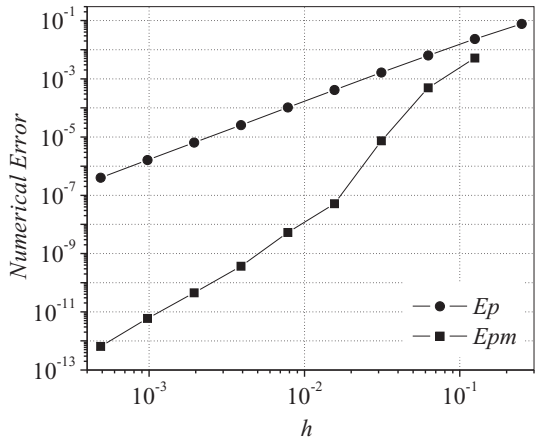

Fig. 6. Type II variable, *uc*, with 2D interpolation of $p=1$, 2D Navier–Stokes (Problem 3).

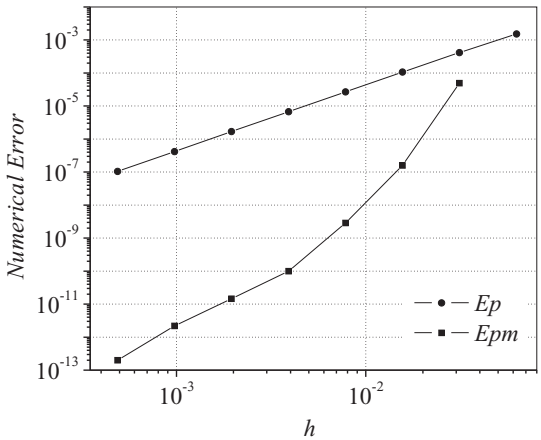

Fig. 7. Type V-1D variable, *u min*1D, with 1D interpolation of $p=10$, 2D Navier–Stokes (Problem 3).

equivalent to the order of accuracy (2) of the adopted numerical scheme (CDS-2), and the value of order p_E associated with *Epm* increases up to 8.3 and 8.4, respectively.

This kind of analysis, with similar results, was conducted for other variables of interest, from Eqs. (12)–(14), namely: v max $1D$, maximum value of the velocity profile v on the line $y=1/2$, whose analytical solution is v max $1D=\sqrt{3}/6$; the x coordinate value of v max $1D$, whose analytical solution is $x(v \text{ max } 1D) = (3-\sqrt{3})/6$; ψmin, minimum value of stream function, and its y coordinate.

When considering a 2D Ω, variables λh whose locations change with resolution across different grids can fall into one of two categories: (i) the values of both the x and y coordinates may change, as in Fig. 4, or (ii) the value of only one, either the x or y coordinate, may change, as in Fig. 7. In the latter case, a simplified approach may be taken in the search for the location of the variable through grid refinement, namely, a one-dimensional approach, because one of the two coordinates remains fixed across all grids under consideration. The results obtained using this approach are comparable to those obtained in Fig. 4, which uses a fully two-dimensional approach.

Marchi et al. [7] observed RRE to be ineffective in reducing *Eh* for variables that involve extreme values (maxima and minima). The variables analyzed in their study coincide with those described in the present work. Based on the presented results, it can be seen that a significant improvement in the performance of RRE can be achieved by applying the procedure proposed herein.

3.4. Results for Problem 4: 2D Navier–Stokes equations with unknown analytical solution

The numerical solutions were obtained using computational codes written by the authors in Fortran 90 and using the Intel Fortran compiler v. 11.1 with double precision and were run on a computer with two 3.47 GHz hexa-core Intel Xeon X5690 processors with 192GB of RAM hosting 64-bit Windows 7. Computations were performed on 10 grids with a refinement ratio of $r=2$. $N \times N = 4 \times 4$ nodes were used in the coarsest grid and $N \times N = 2048 \times 2048$ nodes in the finest one.

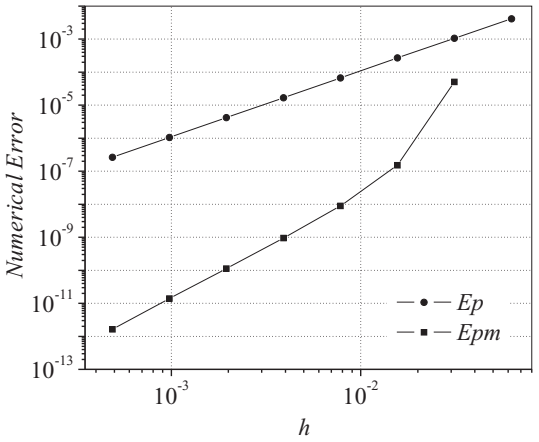

Fig. 8. Type V-1D variable, $y(u\ min\ 1D)$, with 1D interpolation of $p=10$, 2D Navier–Stokes (Problem 3).

Table 3
Comparisons of *uc* with other authors for the Problem 4. Type II-2D variable, with 2D polynomial interpolation, $p=1$.

Reference	*uc*	*U*	p_U
[25]	-0.06080		
[23]	-0.0620561		
[6]	-0.0620		
[26]	-0.06205		
[7]	-0.0620561	±6E-07	2.07
Present	-0.06205613519461	-3E-14	9.41

Table 4
Comparisons of *v*min and its respective *x*-coordinate with other authors for the Problem 4. Type V-1D variable, with 1D polynomial interpolation, $p=10$.

Reference	*v*min	*U*	p_U
[25]	-0.51550		
[23]	-0.5270771		
[7]	-0.52706	±6E-05	2.13
Present	-0.52707730	-4E-08	2.60
Reference	*x*(*v*min)	*U*	p_U
[25]	0.9063		
[23]	0.9092		
[7]	0.9097	±5E-04	1
Present	0.909246996	-4E-09	3.65

Results of the variable *uc* are presented in Table 3, including data of the current work and other ones available in literature. This is the same type of variable found in Problem 3. Moreover, the discretization error estimations (*U*), according to each author, are also presented; in the current work, *U* was evaluated by the use of the estimator developed by Martins [10] for numerical solutions obtained with RRE. Table 3 also includes the apparent order (p_U) according to each author; in the current work, it was evaluated analogously to Eqs. (20) and (21), as described in [21]. It can be observed the fact that the solution of *uc* in the current work agrees to two previous works [7,23] (both with higher precision) in all their six significant digits; however, the current work increases their precision adding seven more significant digits, and noticeably reducing *U* and increasing p_U.

Results of the current work for the variable *v*min and its respective *x*-coordinate are presented in Table 4, which also includes results from other works available in literature. Related to *v*min, it can be noticed that the solution of the current work exactly agrees to one of the previous works [23], which presents higher precision, in its six first significant digits, disagreeing in the last digit, and agrees to the results of Marchi et al. [7] within the *U* interval; however, the current work increases the precision of previous works by adding one more significant digit, and strongly reducing *U*. Besides, it can be noted that for *x*(*v*min), the results of the current work exactly agree to one of the previous works [23], which presents higher precision, in its four significant digits, and agree to results of Marchi et al. [7] within the *U* interval; however, the

Table 5
Comparisons of ψ min and their x- and y-coordinates with other authors for the Problem 4. Type V-2D variable, with 2D polynomial interpolation, $p=6$.

Reference	$-\psi$ min	U	p_U
[25]	0.117929		
[24]	0.118821		
[23]	0.1189366		
[6]	0.118942		
[26]	0.11892		
[7]	0.11893671	±3E-08	2.00
Present	0.1189366104	5E-10	3.57

Reference	$x(\psi$ min)	U	p_U
[25]	0.5313		
[24]	0.5308		
[23]	0.5308		
[6]	0.5300		
[26]	0.53125		
[7]	0.5312	±5E-04	
Present	0.5307901165	2E-10	4.96

Reference	$y(\psi$ min)	U	p_U
[25]	0.5625		
[24]	0.5659		
[23]	0.5652		
[6]	0.5650		
[26]	0.56543		
[7]	0.565	±1E-03	
Present	0.56524055	2E-08	3.04

current work strongly increases the precision of previous works by adding five more significant digits, and considerably reducing U.

Results of the variable ψ min and their x- and y-coordinates are presented in Table 5, which includes data from the current work and the ones from other works available in literature. Related to ψ min, it can be observed that the solution of the current work exactly agrees to the one of the previous works [23], which presents higher precision, in its seven significant digits, and agrees to the results of Marchi et al. [7] in its six first significant digits; however, the current work increases the precision of previous works by adding three more significant digits over [23], and considerably reducing U. Besides, it can be seen that for $x(\psi$ min) the solution of the current work exactly agrees to two previous works [23,24] in all their four significant digits, and agrees to the results of Marchi et al. [7] within the U interval; however, the current work strongly increases the precision of previous works by adding six more significant digits and noticeably reducing U. Finally, related to $y(\psi$ min) it can be noticed that the solution of the current work exactly agrees to one of the previous works [23], which presents higher precision, in its four significant digits, and agrees to the results of Marchi et al. [7] within the U interval; however, the current work strongly increases the precision of previous works by adding four more significant digits, and considerably reducing U.

4. Conclusion

Based on previously published reports of the poor performance of RRE in reducing *Eh* for some types of variables of interest, five types of variables were defined in this work according to their locations on different grids. The performance of RRE when applied to these variables was then analyzed using four test problems (1D Poisson equation, 2D Burgers equations and the 2D Navier–Stokes equations), discretized using Finite Difference and Finite Volume Methods, including numerical schemes of various orders of accuracy. Grids with as many as millions of nodes, several levels of extrapolation, multigrid methods and quadruple and double precision were used.

Based on the obtained results, it was concluded that to employ RRE, the variable of interest must firstly be characterized as one of the five established types. The subsequent procedure for the reduction of *Eh* using RRE is as follows:

(1) For a global or local variable with a fixed location that coincides with a nodal point on every considered grid, RRE can be directly applied. RRE is best suited to this type of variable.
(2) For a variable located at the midpoint of a set of nodal coordinates, linear (or bilinear/trilinear) interpolation should be applied prior to RRE.

(3) For a variable that has a fixed location that is neither a nodal point nor a midpoint thereof, polynomial interpolation of the highest possible degree appropriate to the number of dimensions of the problem under study must be applied prior to RRE.
(4) For a variable whose value is predetermined and whose position is unknown, polynomial interpolation of the highest possible degree should be applied before finding the solution of the associated polynomial equation; only then RRE can be applied.
(5) In the case of a variable that involves finding an extremum, in which the coordinate values may change as the grid is refined, polynomial interpolation of the highest possible degree appropriate to the number of dimensions of the problem under study should be applied, followed by an optimization technique before applying RRE.

The procedure proposed in the present article permits the reduction of the numerical error of RRE in those cases in which RRE is considered to be ineffective according to the current literature. This procedure, which is characterized as a post-processing procedure, entails extremely low computational cost. Given a desired value for the numerical error, the CPU time and RAM memory required to achieve it are much lower when this procedure is used than when it is not. Moreover, it is, in principle, valid for schemes of any order of accuracy that use either Finite Difference or Finite Volume methods and for any equation and number of dimensions, and it permits considerable reduction in the numerical error.

Acknowledgments

The authors thank The Uniespaço Program of the AEB: Anúncio de Oportunidades AO-01/2006 (Brazilian Space Agency), CNPq: 309365/2013-9 (Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil), and CAPES: project #20 of Edital Pró-Estratégia 50/2011 (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brazil) for their financial support. The first author is supported by a CNPq scholarship. The second author thanks the Araucária Foundation (Paraná, Brazil) and the State University of the Center-West (Universidade Estadual do Centro-Oeste, UNICENTRO-PR). The last two authors would thank by their scholarships provided by CAPES: project #20 of Edital Pró-Estratégia 50/2011. The authors thank the reviewers of this work.

References

[1] J.C. Roy, W.L. Oberkampf, A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing, Comput. Methods Appl. Mech. Eng. 200 (2011) 2131–2144.
[2] Y. Wang, J. Zhang, Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D Poisson equation, J. Comput. Phys. 228 (2009) 137–146, doi:10.1016/j.jcp.2008.09.002.
[3] Y. Ma, Y. Ge, A high order finite difference method with Richardson extrapolation for 3D convection diffusion equation, Appl. Math. Comput. 215 (2010) 3408–3417, doi:10.1016/j.amc.2009.10.035.
[4] Y. Wang, J. Zhang, Fast and robust sixth-order multigrid computation for the three-dimensional convection–diffusion equation, J. Comput. Appl. Math. 234 (2010) 3496–3506, doi:10.1016/j.cam.2010.05.022.
[5] G. Dahlquist, A. Bjorck, Numerical Methods in Scientific Computing, SIAM, 2008.
[6] E. Erturk, T.C. Corke, C. Gokçol, Numerical solutions of 2D steady incompressible driven cavity flow at high Reynolds numbers, Int. J. Numer. Methods iFluids 48 (2005) 747–774, doi:10.1002/fld.953.
[7] C.H. Marchi, R. Suero, L.K. Araki, The lid-driven square cavity flow: numerical solution with a 1024 × 1024 grid, J. Braz. Soc. Mech. Sci. Eng. 31 (3) (2009) 186–198 http://dx.doi.org/10.1590/S1678-58782009000300004.
[8] C. Burg, T. Erwin, Application of Richardson extrapolation to the numerical solution of partial differential equations, Numer. Methods Partial Differ. Equ. 25 (2009) 810–832, doi:10.1002/num.20375.
[9] X. Nicolas, M. Medale, S. Glockner, S. Gounand, Benchmark solution for a three-dimensional mixed-convection flow, part 1: Reference solutions, Numer. Heat Transf. Part B 60 (2011) 325–345, doi:10.1080/10407790.2011.616761.
[10] Martins M.A. Repeated Richardson Extrapolations with Polynomial Interpolations to Reduce and Estimate the Discretization Error in CFD [in Portuguese]. Ph.D. thesis. Mechanical Engineering Graduate Course. Federal University of Paraná. Curitiba – PR, Brazil, 2013.
[11] C.H. Marchi, A.F.C. Silva, Unidimensional numerical solution error estimation for convergent apparent order, Numer. Heat Transf. Part B 42 (2002) 167–188, doi:10.1080/10407790190053888.
[12] P.J. Roache, Fundamentals of Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, 1998.
[13] J.H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, Springer, 2002.
[14] P. Pulino, H.C.T. Torres, Esquemas compactos de diferenças finitas de alta ordem para problemas de Poisson: Aplicativos para web. [High-order compact finite-difference schemes for Poisson problems: Web applications], in: Proceedings of the Seventh SIMMEC Computational Mechanics Symposium, Araxá, Brazil, 2006.
[15] T.M. Shih, C.H. Tan, B.C. Hwang, Effects of grid staggering on numerical scheme, Int. J. Numer. Methods Fluids 9 (1989) 193–212.
[16] H.K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics, the Finite Volume Method, Pearson/Prentice Hall, 2007.
[17] U. Trottenberg, C. Oosterlee, A. Schüller, Multigrid, Academic Press, 2001.
[18] R.L. Burden, J.D. Faires, Numerical Analysis, Thomson Learning, 2005.
[19] J.F. Steffensen, Interpolation, Chelsea Publishing Company, 1950.
[20] M.S. Bazaraa, H.D. Sherali, C.M. Shetty, Nonlinear Programming: Theory and Algorithms, Wiley & Sons, 2006.
[21] C.H. Marchi, L.A. Novak, C.D. Santiago, A.P.S. Vargas, Highly accurate numerical solutions with repeated Richardson extrapolation for 2D Laplace equation, Appl. Math. Model. 37 (2013) 7386–7397, doi:10.1016/j.apm.2013.02.043.
[22] C.H. Marchi, E.M. Germer, Effect of ten CFD numerical schemes on Repeated Richardson Extrapolation (RRE), J. Appl. Comput. Math. 2 (3) (2013) 1–8, doi:10.4172/2168-9679.1000128.
[23] O. Botella, R. Peyret, Benchmark spectral results on the lid-driven cavity flow, Comput. Fluids 27 (1998) 421–433.
[24] N.G. Wright, P.H. Gaskell, An efficient multigrid approach to solving highly recirculating flows, Comput. Fluids 24 (1995) 63–79.
[25] U. Ghia, K.N. Ghia, C.T. Shin, High resolutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.
[26] C.H. Bruneau, M. Saad, The 2D lid-driven cavity problem revisited, Comput. Fluids 35 (2006) 326–348.