Calcolo (2011) 48: 33–45
DOI 10.1007/s10092-010-0026-5

On the application of the minimum polynomial extrapolation method to incompressible flows with heat transfer

Roberto Bertelle · Maria Rosaria Russo · Manolo Venturin

Received: 12 November 2009 / Accepted: 23 April 2010 / Published online: 22 September 2010

Abstract In this paper, the Minimum Polynomial Extrapolation method (MPE) is used to accelerate the convergence of the Characteristic–Based–Split (CBS) scheme for the numerical solution of steady state incompressible flows with heat transfer. The CBS scheme is a fractional step method for the solution of the Navier–Stokes equations while the MPE method is a vector extrapolation method which transforms the original sequence into another sequence converging to the same limit faster then the original one without the explicit knowledge of the sequence generator. The developed algorithm is tested on a two-dimensional benchmark problem (buoyancy–driven convection problem) where the Navier–Stokes equations are coupled with the temperature equation. The obtained results show the feature of the extrapolation procedure to the CBS scheme and the reduction of the computational time of the simulation.

Keywords Fractional step method · Characteristic Galerkin · CBS scheme · Vector extrapolation methods · Minimum polynomial extrapolation

Mathematics Subject Classification (2000) 65B05 · 65N30

1 Introduction

The computation of steady state numerical solution for incompressible flows with heat transfer is a very important problem in many engineering applications such as

This paper is part of a special issue devoted to the 2nd Dolomites Workshop on Constructive Approximation and Applications, 2009.

R. Bertelle · M.R. Russo
Dip. Matematica Pura ed Applicata, Università degli Studi di Padova, Via Trieste 63, 35121 Padova, Italy

M. Venturin (✉)
Dip. di Informatica, Università di Verona, Ca' Vignal 2, Strada Le Grazie, 37134 Verona, Italy
e-mail: manolo.venturin@univr.it

thermal insulation, solar energy, cooling phenomena, etc. The mathematical model is based on the solution of the Navier–Stokes equations coupled with the temperature equation for the heat transfer. The numerical solution of this problem is generally done by iterative schemes and may require long simulation times which may not be acceptable in the development of engineering projects. Thus, in all these cases, it is important to improve the convergence rate of the overall iterative scheme which may be obtained, for instance, by vector extrapolation methods.

A possible approach to solve the Navier–Stokes equations is through the use of the Characteristic–Based–Split (CBS) scheme which is a fractional step method based on a finite difference velocity–projection scheme where the convective terms are treated by using the idea of the Characteristic–Galerkin method (see [1–4] for details). In particular, for the solution of incompressible flows the CBS scheme essentially uses the semi-implicit formulation which needs a small time step and, as a consequence, may require large simulation time. An alternative approach based on an artificial compressibility concept has been proposed in [6] to permit explicit time integration and, hence, having shorter CPU times.

The Minimum Polynomial Extrapolation (MPE) is a well–known polynomial acceleration technique which is used to accelerate the convergence of sequences without the explicit knowledge of the sequence generator. This algorithm has been successfully applied to several problems in different fields, see for example [8–16].

In this paper, we present a modified version of the CBS algorithm where the MPE acceleration technique is applied to improve the rate of convergence of the original sequence generated by the CBS scheme where we consider non-isothermal incompresible flow problems. Furthermore, we refer to [5] for the application of the algorithm to other fluid problems.

The paper is organized as follows. In Sect. 2, we review the problem with its governing equations and the corresponding approximation by the CBS scheme. Then in Sect. 3 we briefly present the MPE method and its implementation in the developed algorithm. Finally, in Sect. 4, we show the obtained numerical results and we give some concluding remarks.

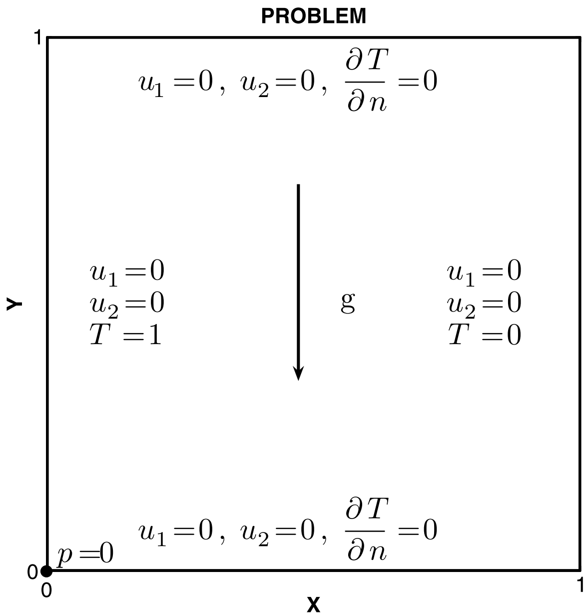
2 Statement of the problem and CBS approximation

This problem shows the natural convection where the Navier–Stokes equations are coupled with the temperature equation for heat transfer. A local temperature difference induces a local density difference within the fluid and this produces the fluid motion.

The geometry of the problem, as well as the boundary conditions, are shown in Fig. 1. The two vertical walls of the square domain $\Omega = (0, 1) \times (0, 1)$ are kept at the constant temperatures T_{lw} and T_{rw}, respectively for the left and the right one; the other two horizontal walls are adiabatic, i.e., $\partial T/\partial \mathbf{n} = 0$ where $\mathbf{n}$ denotes the outward normal unit vector. Moreover, no–slip velocity conditions are applied on all the walls and the pressure in the left bottom corner is fixed at zero for each time t.

The governing equations may be written in conservation form as [3, 7]

Fig. 1 Geometry and boundary conditions for the natural convection problem with no slip velocity at the walls



Continuity equation

$$\frac{\partial u_i}{\partial x_i}=0$$

Momentum equations

$$\frac{\partial u_i}{\partial t}+\frac{\partial}{\partial x_j}(u_j u_i)=-\frac{\partial p}{\partial x_i}+\frac{\partial}{\partial x_j}\left(\mu\frac{\partial u_i}{\partial x_j}\right)+g_i$$

Temperature equation

$$\frac{\partial T}{\partial t}+\frac{\partial}{\partial x_i}(u_i T)=\frac{\partial}{\partial x_i}\left(\alpha\frac{\partial T}{\partial x_i}\right)$$

where Einstein's summation ($i=1,2$) convention is used. The unknown variables are the velocity field u_i, the pressure p and the temperature T. In the previous system of equations, μ is the kinematic viscosity term of the fluid, g_i takes into account gravitational forces acting on the system the system and $\alpha=k/(\rho c_p)$ is the thermal diffusivity which depends on the fluid density ρ, on the thermal conductivity k and on the specific heat c_p. The system is completed with both boundary and consistent initial conditions.

The CBS scheme requires a dimensionless form of the previous equations which may be obtained using the following scale factors

$$x_i^\star=\frac{x_i}{L},\qquad u_i^\star=\frac{u_i L}{\alpha},\qquad p^\star=\frac{pL^2}{\alpha^2},\qquad t^\star=\frac{t\alpha}{L^2}$$

$$T^{\star} = \frac{T - T_L}{T_H - T_L}, \qquad \mathrm{Pr} = \frac{\mu}{\alpha}, \qquad \mathrm{Ra} = \frac{g\gamma(T_H - T_L)L^3}{\mu\alpha}$$

where the superscript $\star$ denotes a non-dimensional quantity. Pr and Ra are the Prandtl and Rayleigh numbers, L is the characteristic length, T_H and T_L are, respectively, the higher and the lower temperatures, g is the gravity acceleration and γ is the thermal expansion coefficient. Then, the equations in their dimensionless form can be written as (omitting, for brevity, the superscript $\star$).

Continuity equation

$$\frac{\partial u_i}{\partial x_i} = 0$$

Momentum equations

$$\frac{\partial u_i}{\partial t} = -\frac{\partial}{\partial x_j}(u_j u_i) - \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \mathrm{Ra}\,\mathrm{Pr}\,T$$

$$\tau_{ij} = \mathrm{Pr}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)$$

where, δ_{ij} is the Kronecker delta.

Temperature equation

$$\frac{\partial T}{\partial t} = -\frac{\partial}{\partial x_i}(u_i T) + \frac{\partial^2 T}{\partial x_i^2}.$$

The numerical solution of the above set of equations is done using the CBS scheme as presented in [1, 2, 7, 18]. It consists of a fractional procedure for the solution of the momentum and the continuity equations followed by another procedure based on the Characteristic–Galerkin method for the solution of the advection equation for the temperature. The discretization of the problem is obtained via the Finite Element Method with linear elements for all the variables; furthermore, the algorithm has been used in its semi-implicit form ($\theta_1 = 1$ and $\theta_2 = 1$). The n-th temporal step of the CBS scheme can be summarized as follows:

Step 1: Solve the intermediate momentum equation obtained from the momentum equation without the pressure gradient term:

$$\tilde{u}_i^n = u_i^n + \Delta t\left[\frac{\partial \tau_{ij}}{\partial x_j} - \mathrm{Ra}\,\mathrm{Pr}\,T + \frac{\Delta t}{2}u_k\frac{\partial^2}{\partial x_k \partial x_j}(u_i u_j)\right]^n$$

Step 2: Solve the modified continuity equation for the pressure:

$$\left(\frac{\partial^2 p}{\partial x_i \partial x_i}\right)^{n+1} = \frac{1}{\Delta t}\left[\frac{\partial \tilde{u}_i}{\partial x_i}\right]^n$$

Step 3: Correct the intermediate velocity field $\tilde{u}_i^n$ using the new values of the pressure computed at step 2:

$$u_i^{n+1} = \tilde{u}_i^n - \Delta t \left[\frac{\partial p}{\partial x_i} \right]^{n+1}$$

Step 4: Solve the advection–diffusion equation for temperature:

$$T^{n+1} = T^n + \Delta t \left(-\frac{\partial (u_i T)}{\partial x_i} + \frac{\partial^2 T}{\partial x_i \partial x_j} + \frac{\Delta t}{2} u_j \frac{\partial^2}{\partial x_i \partial x_j} (u_i T) \right)^n .$$

In particular, in the splitting part summarized in step 1, the pressure gradient is omitted from the momentum equations and an intermediate velocity field is estimated by the Characteristic–Galerkin method. By removing the pressure term from the momentum equations the fractional step method introduces a first-order error into the momentum equations [17]. This permits to circumvent the Ladyshenskaya–Bubuska–Brezzi (LBB) condition allowing equal-order interpolations [19] for both velocity and pressure fields and enhancing pressure stability of the CBS scheme. As an alternative, it is possible to consider the pressure gradient into the momentum equations as a source term but, as discussed in [17], this approach is not recommended. Then, the continuity equation is solved using the intermediate vector field values of the previous step obtaining the new pressure values. Finally, the velocity field is corrected with the new computed pressure term. Then, the advection–diffusion scalar equation for temperature, is solved using the Characteristic–Galerkin method. In particular, the Characteristic–Galerkin method stabilizes advective terms using a Taylor series expansion of the temporal derivative along the characteristic leading to the introduction of consistent stabilization artificial diffusion terms.

In this work, linear finite elements with direct integration are used to further reduce the computational cost. Moreover, we use the semi-implicit version of the scheme with global time-stepping. Denoting by h the mesh size and by $\|u\|$ the norm of the velocity field, the CBS algorithm in this form is subject to the critical time step given by $\Delta t_c = h/\|u\|$ which is due to the linear stability analysis of the one dimensional convection–diffusion equation using the Characteristic–Galerkin method.

3 The extrapolation algorithm

In this section, the Minimum Polynomial Extrapolation (MPE) acceleration technique is briefly introduced and applied to the CBS algorithm.

The MPE is a map $\mathcal{F}_{MPE}$ that transforms a given convergent N-dimensional vector sequence $\{x_n\}$, $n \in \mathbb{N}$, into another sequence $\{y_n\}$, $n \in \mathbb{N}$, convergent to the same limit. The purpose of this map is to get a better convergence rate for the sequence $\{y_n\}$, such that

$$\lim_{n \to +\infty} \frac{\|y_n - l\|}{\|x_n - l\|} = 0$$

where l is the common limit of the sequences $\{x_n\}$, $\{y_n\}$ and $\|\cdot\|$ is a vector norm. For a given positive integer parameter $k \ll N$, firstly the vector $\zeta = [\zeta_0, \ldots, \zeta_{k-1}]^T \in \mathbb{R}^k$ is computed as the least square solution of the overdeterminated linear system

$$U\zeta = -u_{n+k} \quad \text{where } U = [u_n, u_{n+1}, \ldots, u_{n+k-1}]$$

and $u_n = x_{n+1} - x_n$. Then, the vector y_n is obtained as

$$y_n = \sum_{j=0}^{k} \left\{ \frac{\zeta_j}{\sum_{i=0}^{k} \zeta_i} x_{n+j} \right\}$$

where it is assumed that $\zeta_k = 1$. It is interesting to remark that the linear system is inexpensive to solve due to $k \ll N$. However, the proper choice of the parameter k to obtain an improvement of the convergence rate is a difficult task and involves a preliminary training session which depends on the considered problem.

The CBS is an iterative algorithm which, starting from an initial guess, computes the sequence of solutions $\mathcal{S}^{(n)} = \{T^{(n)}, u_1^{(n)}, u_2^{(n)}, p^{(n)}\}$ evaluated at each time step. The stationary solution of the problem is denoted by $\mathcal{S}^{(\infty)}$, *i.e.*

$$\mathcal{S}^{(\infty)} = \lim_{n \to +\infty} \mathcal{S}^{(n)}.$$

The remaining part of this section presents a modified version of the CBS algorithm to improve the rate of convergence of $\mathcal{S}^{(n)}$ towards $\mathcal{S}^{(\infty)}$. With this aim the MPE is applied to the sequence $\mathcal{S}^{(n)}$, that is for some positive parameters k and i, *i.e.* $\mathcal{S}^{(n)}$, $n = i, \ldots, i + k - 1$ terms of the sequence are collected and then the extrapolation step is performed as

$$v^{(i+k-1)} \leftarrow \mathcal{F}_{MPE}\big(v^{(i)}, \ldots, v^{(i+k-1)}\big), \quad \forall v \in \{T, u_1, u_2, p\}$$

where $\mathcal{F}_{MPE}$ is the map associated to the MPE method. The main problem is the choice of the parameters k and i, *i.e.* how the collecting sequence starts and how many collecting vectors are necessary for extrapolation. Since, the literature does not give any guideline for this kind of evolution problems the following heuristic collecting criteria is adopted.

Collecting criteria *Let $\mathcal{V} \subseteq \{T, u_1, u_2, p\}$ be a subset of the problem variables. Let $v \in \mathcal{V}$. Then, denoting by $\mathcal{N}$ the mesh nodes, k consecutive terms $\mathcal{S}^{(n)}$, $n = i, \ldots, i + k - 1$, are collected if for each $v \in \mathcal{V}$*

$$\max_{N \in \mathcal{N}} \left| v_N^{(j+1)} - v_N^{(j)} \right|, \quad j = i, \ldots, i + k - 2$$

occurs at the same mesh node, denoted by N_v. Note that N_v may be different for each variable $v \in \mathcal{V}$.

The collecting criteria is shown in Fig. 2 for the case $\mathcal{V} = \{T, u_1, u_2, p\}$ where there are four differencies that have to be taken into account. Let N_{k1}, N_{k2}, N_{k3}, N_{k4}

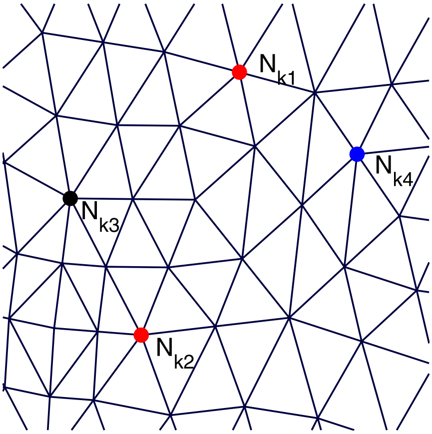


Fig. 2 The collecting criteria for the case $\mathcal{V} = \{T, u_1, u_2, p\}$. For each variables $v \in \mathcal{V}$, the node N_v of the mesh $\mathcal{N}$ that satisfies $\max_{N \in \mathcal{N}} |v_N^{(n+1)} - v_N^{(n)}|$ at time i must remain the same for $n = i, \ldots, i+k-1$. Each variable of $\mathcal{V}$ may have a different node N_v; for example N_{k1}, N_{k2} are associated to u_1 and u_2, and N_{k3} to T, N_{k4} to p

be the nodes of the mesh that fulfills the relation of the collecting criteria at some time step i for, T, u_1, u_2 and p respectively. Then, these nodes have to be the same for all the next steps $i+1, \ldots, i+k-2$ in order to collect $\mathcal{S}^{(n)}$, $n = i, \ldots, i+k-1$. For example, node N_{k1} must satisfy the equation of the collecting criteria for the variable T for all the time steps from i to $i+k-2$.

Note that the extrapolation is performed for all the variables of the problem but the set $\mathcal{V}$ used in the collecting criteria may not include all these variables. Moreover, it is not clear a priori which is the optimal choice for $\mathcal{V}$. The previous collecting criteria is a restrictive condition that like to mimic the behavior of iterative solutions near stationary points where the shape of the solutions change small between two consecutive iterations. As a consequence the differencies between consecutive iterations are proportional and hence it is reasonably that they appear to the same nodes of the mesh.

The collecting criteria is used to build the following accelerated CBS algorithm.

Algorithm

1. select $\mathcal{V}$
2. set k_s, k_c, k, k_r
3. skip the first k_s CBS iterations
4. collect k_c iterations satisfying the collecting criteria
5. apply MPE to the last k of the k_c collected iterations
6. take the extrapolated solution as the new starting point
7. perform k_r CBS iterations
8. return to step 4 until convergence.

At step 1 the set $\mathcal{V}$ is chosen. At step 2 the four algorithm parameters k_s, k_c, k, k_r are introduced. At step 3, the first k_s iterations obtained by the CBS algorithm are skipped since they are the most far away from the stationary solution and, hence, not so good to perform an extrapolation step. At step 4, k_c consecutive CBS iterations are collected. In the case that the collecting criteria breakdowns before k_c consecutive

Table 1 The three different cases of the problem taken into account in the numerical simulations

	Case 1	Case 2	Case 3
Ra	10^3	10^4	10^5
Pr	0.71	0.71	0.71

iterations are collected the step 4 restarts. At step 5 the extrapolation is performed only on the last k iterations of the k_c collected iterations. There are two reasons for this. The first one is to maintain the number of extrapolation iterations, and thus the CPU time, low in solving the linear system associated with the MPE. The second one is to check the collecting criteria inside a more long sequence to avoid the collection of less significant iterations for the extrapolation. Step 6 classifies this algorithm as a restarting one since the extrapolated solution acts as a new starting point for the CBS algorithm. At step 7, since the extrapolated solution is typically not feasible for the evolution problem (i.e., it does not satisfy the equations), k_r iterations are given to the CBS algorithm to relax towards a feasible solution before looking back again to step 4. In step 8 the following stopping criteria, based on the maximum absolute error of the scaled variables, is checked

$$e < \epsilon \quad \text{with } e = \frac{\max_{v\in\{p,u_1,u_2,T\}} \|v^{(i)} - v^{(j)}\|_\infty}{\Delta t},$$

where ϵ is some prefixed tolerance, Δt is the time-step of the CBS algorithm and $j = i + m$ ($m = 10$ in our simulations).

Steps 1 and 2 are critical since it is not clear a priori which are the optimal choices for $\mathcal{V}$ and for the parameters of the algorithm. In the next section we present the results obtained using different $\mathcal{V}$ as well as for different parameters k_s, k_c, k, k_r. Finally, to avoid breakdown of the proposed accelerated algorithm, it is necessary that the underlying CBS scheme is robust and stable.

4 Numerical results

The equations are discretized using the Finite Element Method with an unstructured mesh consisting of 2375 nodes and 4584 triangular elements. The scaled global time-stepping of the semi-implicit CBS algorithm is $\Delta t = 10^{-5}$.

The simulations are performed on a computer, running Linux, equipped with 4 GB of RAM and an Intel Core Duo processor with 2.26 GHz and 3 MB of L2 cache memory. The post-processing is done using Matlab.

We consider three cases of the problem accordingly to the Rayleigh and Prandtl numbers shown in Table 1.

We have done seven simulations for each one of the previous cases. The differences from one simulation to another are in following features

- choice of the variables used to check the collecting criteria;
- choice of the extrapolation parameters.

Table 2 Collecting criteria and extrapolation parameters (from *left* to *right*, k_s, k_c, k, k_r)

Simulation	Description	
	Extrap. criteria	Parameters
sim0	No extrapolation	
sim1	(T, u_1, u_2, p)	(0, 200, 100, 100)
sim2	(T, u_1, u_2, p)	(0, 200, 50, 100)
sim3	(T, u_1, u_2, p)	(0, 200, 100, 200)
sim4	(T, u_1, u_2)	(0, 200, 100, 100)
sim5	(T, u_1, u_2)	(0, 200, 50, 100)
sim6	(T, u_1, u_2)	(0, 200, 100, 200)

Table 3 Simulation times obtained for a Rayleigh number of 10^3

Simulation	Stopping tolerance		
	1e-06	1e-07	1e-08
sim0	1921 (100%)	2022 (100%)	2115 (100%)
sim1	632 (33%)	678 (34%)	714 (34%)
sim2	661 (35%)	720 (36%)	782 (37%)
sim3	643 (34%)	678 (34%)	729 (35%)
sim4	724 (38%)	759 (38%)	770 (37%)
sim5	720 (38%)	742 (37%)	813 (39%)
sim6	658 (35%)	690 (35%)	776 (37%)

Table 2 summarizes the seven simulations named `sim0` to `sim6`. `sim0` refers to the simulation without any acceleration. For the other ones, the "extrap. criteria" column shows the variables adopted in the collecting criteria. The "parameters" column reports the four extrapolation parameters k_s, k_c, k, k_r used by the algorithm. For example, `sim1` uses all the four variables in the collecting criteria and has $k_s = 0$, $k_c = 200$, $k = 100$, $k_r = 100$. Regarding the stopping tolerance ϵ, we have considered, for each simulation, the three different values 10^{-6}, 10^{-7}, 10^{-8}.

Tables 3, 4, 5 show, for simulations `simj`, $j = 1, \ldots, 6$, both the CPU times t_j, in seconds, and the ratios t_j/t_0 where t_0 is the CPU time of `sim0`. These tables clearly prove that the proposed algorithm improves the rate of convergence. Furthermore, there are no relevant differences among the obtained results for the six simulations, that is for this kind of problem, the acceleration performs always well and is quite insensitive both to the choice of the variables used in the collecting criteria and to the extrapolation parameters. We remark that for a given problem a training session is typically required to find a proper set of the extrapolation parameters that improves the rate of convergence.

Figure 3 shows the results for the three cases. The Rayleigh number increases from the most left column to the right one. The upper row represents the streamline of the velocity, the middle and the lower ones are the pressure and the temperature, respectively.

Table 4 Simulation times obtained for a Rayleigh number of 10^4

Simulation	Stopping tolerance		
	1e-06	1e-07	1e-08
sim0	4061 (100%)	4199 (100%)	4358 (100%)
sim1	1025 (25%)	1083 (26%)	1173 (27%)
sim2	861 (21%)	932 (22%)	999 (23%)
sim3	1101 (27%)	1164 (28%)	1258 (29%)
sim4	973 (24%)	1025 (24%)	1067 (24%)
sim5	878 (22%)	952 (23%)	1026 (24%)
sim6	975 (24%)	1033 (25%)	1075 (25%)

Table 5 Simulation times obtained for a Rayleigh number of 10^5

Simulation	Stopping tolerance		
	1e-06	1e-07	1e-08
sim0	1798 (100%)	1854 (100%)	1896 (100%)
sim1	1218 (68%)	1272 (69%)	1313 (69%)
sim2	1727 (96%)	1765 (95%)	1824 (96%)
sim3	1135 (63%)	1183 (64%)	1227 (65%)
sim4	1212 (67%)	1220 (66%)	1224 (65%)
sim5	1574 (88%)	1603 (87%)	1647 (87%)
sim6	1190 (66%)	1199 (65%)	1203 (63%)

Figure 4 shows the convergence process of `sim4` at different Rayleigh numbers, one per column. This qualitative behavior is common for all the simulations. The error for each variable $v \in \{u_1, u_2, p, T\}$ reported in this figure is defined as

$$e_v = \frac{1}{\Delta t} \frac{\|v^{(n)} - v^*\|_2}{\|v^*\|_2}$$

where $v^{(n)}$ is the solution at the end of the n-th iteration and v^* is the corresponding reference solution computed with a high accuracy ($\epsilon = 10^{-10}$) via the standard CBS algorithm without acceleration. This formula is adopted because the algorithm procedure is time dependent; so, the difference between two iterates is proportional to the time step Δt and to the residual of the solution. It is interesting to note the great impact of the extrapolation even if the number of extrapolation steps, denoted by circles, are at most three. Furthermore, the pressure seems to be the variable that benefits less from the extrapolation due to a quite large error immediately after the extrapolation step. However, the global behavior of the solution in not influenced by this.

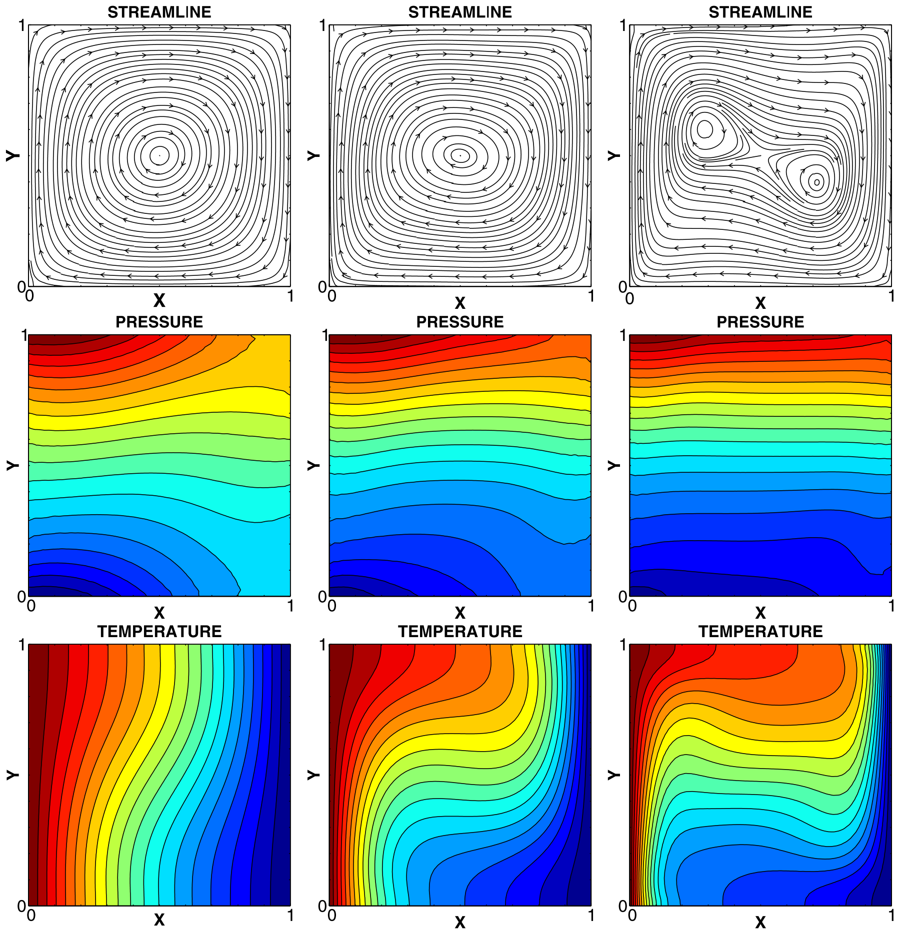


Fig. 3 Behavior of the solution at different Rayleigh numbers (*first column*: $Ra = 10^3$, *second column*: $Ra = 10^4$, *third column*: $Ra = 10^5$). The *first row* plots the velocity streamline, the *second row* the pressure and the *third row* the temperature

5 Conclusions

In this work, the acceleration technique known as Minimal Polynomial Extrapolation (MPE) is applied to increase the rate of convergence of the numerical solution of an isothermal incompressible flow problem where the governing equations are solved by the Characterist ic–Based–Split algorithm. The results show a considerable reduction of the CPU time and the acceleration technique is quite robust. Thus, for this kind of problem, the MPE may be used as a remarkable feasible tool to speed up the overall convergence process.

Acknowledgements Authors are pleasure to acknowledge professors M. Redivo Zaglia, S. De Marchi and M. Morandi Cecchi for their constant suggestions and remarks. We would like to thank the referees for their helpful comments which greatly improved the quality of the paper.

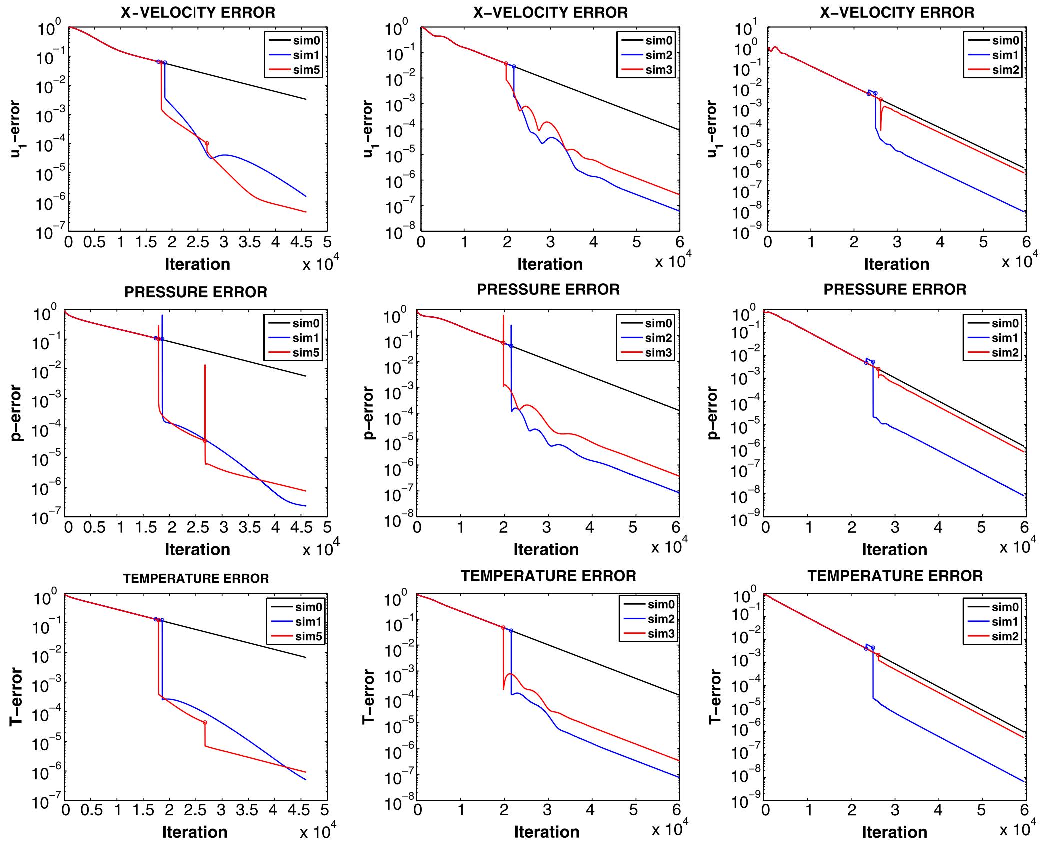


Fig. 4 Convergence behavior of the best (*blue*) and worst (*red*) simulation for three different Rayleigh numbers, one for *each column* (left $Ra = 10^3$, middle $Ra = 10^4$, right $Ra = 10^5$). *First row* reports the error on the x component of the velocity, u_1, the *second row* the error associated to the pressure p and the *third row* the error of the temperature T

References

1. Zienkiewicz, O.C., Codina, R.: A general algorithm for compressible and incompressible flow. I. The split, characteristic–based scheme. Int. J. Numer. Methods Fluids **20**(8–9), 869–885 (1995)
2. Zienkiewicz, O.C., Nithiarasu, P., Codina, R., Vázquez, M., Ortiz, P.: The Characteristic–Based–Split procedure: an efficient and accurate algorithm for fluid problems. Int. J. Numer. Methods Fluids **31**(1), 359–392 (1999)
3. Zienkiewicz, O.C., Taylor, R.L., Nithiarasu, P.: The Finite Element Method for Fluid Dynamics, vol. 3, 6th edn. Elsevier, Butterworth (2005)
4. Morandi–Cecchi, M., Venturin, M.: Characteristic-based split (CBS) algorithm finite element modelling for shallow waters in the Venice lagoon. Int. J. Numer. Methods Eng. **66**(10), 1641–1657 (2006)
5. Venturin, M., Bertelle, R., Russo, M.R.: An accelerated algorithm for Navier–Stokes equations. Simul. Model. Pract. Theory **18**, 217–229 (2010)
6. Nithiarasu, P.: An efficient artificial compressibility (AC) scheme based on the Characteristic Based Split (CBS) method for incompressible flows. Int. J. Numer. Methods Fluids **56**(13), 1815–1845 (2003)
7. Massarotti, N., Nithiarasu, P., Zienkiewicz, O.C.: Characteristic–Based–Split (CBS) algorithm for incompressible flow problems with heat transfer. Int. J. Numer. Methods Heat Fluid Flow **8**(8), 969–990 (1998)
8. Sidi, A.: Vector extrapolation methods with application to solution of large system of equations and to PageRank computations. Comput. Math. Appl. **56**, 1–24 (2008)
9. Jbilou, K., Sadok, H.: Vector extrapolation methods. Applications and numerical comparison. J. Comput. Appl. Math. **122**, 149–165 (2000)
10. Rajeevan, N.: Vector-extrapolated fast maximum likelihood estimation algorithms for emission tomography. IEEE Trans. Med. Imaging **11**(1), 9–20 (1992)
11. Bertelle, R., Russo, M.R.: An approach to the Gummel map by vector extrapolation methods. Numer. Algorithms **45**, 331–343 (2007)
12. Dascal, L., Rosman, G., Kimmel, R.: Efficient Beltrami filtering of color images via vector extrapolation. In: SSVM07. LNCS, vol. 4485, pp. 92–103. Springer, Berlin (2007)
13. Shimano, K., Yonezu, S., Enomoto, Y.: Acceleration of unsteady incompressible flow calculation using extrapolation methods (fluids engineering). J. Trans. Jpn. Soc. Mech. Eng. B **74**(745), 1896–1902 (2008) (in Japanese)
14. Brezinski, C., Redivo-Zaglia, M.: Extrapolation Methods Theory and Practice. North-Holland, Amsterdam (1991)
15. Smith, D.A., Ford, W.F., Sidi, A.: Extrapolation methods for vector sequences. SIAM Rev. **29**, 199–233 (1987)
16. Jbilou, K., Sadok, H.: Some results about vector extrapolation methods and related fixed–point iterations. J. Comput. Appl. Math. **36**, 385–398 (1991)
17. Nithiarasu, P., Zienkiewicz, O.C.: Analysis of an explicit and matrix free fractional step method for incompressible flows. Comput. Methods Appl. Mech. Eng. **195**, 5537–5551 (2006)
18. Venturin, M.: A finite element stabilization system for advection–diffusion problems. PhD Thesis, University of Padua (2006)
19. Domino, S.P.: A comparison of various equal-order interpolation methodologies using the method of manufactured solutions. Center for Turbulence Research. Proceedings of the Summer Program (2008)