SIMULATION OF HEAT AND MASS TRANSPORT IN A SQUARE LID-DRIVEN CAVITY WITH PROPER GENERALIZED DECOMPOSITION (PGD)

A. Dumon[1], C. Allery[1], and A. Ammar[2]

[1]LaSIE (FRE-CNRS), Pôle Sciences et Technologie, La Rochele, France
[2]Arts et Metiers ParisTech, Angers, France

The aim of this study is to apply proper generalized decomposition (PGD) to solve mixed-convection problems with and without mass transport in a two dimensional lid-driven cavity. PGD is an iterative reduced order model approach which consists of solving a partial differential equation while seeking the solution in separated form. Comparisons with results in the literature and with results from a standard solver are make. For the case of a mixed-convection problem without mass transfer, three Richardson numbers are considered, $Ri = 0.1$, $Ri = 1$, and $Ri = 10$. In this case, PGD is seven times faster than the standard solver with $Ri = 10$ with a similar accuracy. For the case with mass transfer, simulations are done with different Lewis numbers, $Le = 5$, $Le = 25$, and $Le = 50$, and with different value of the ratio N between the solutal and the thermal Grashoff numbers. In this case, too, PGD is ten times faster than the standard solver.

I. INTRODUCTION

The study of flow in mixed convection is a relevant research topic that has been treated by many authors in recent years. In fact, these type of flows can be found in many engineering applications—for example, cooling of electronic devices, lubrication and drying technologies, heat exchangers, and ventilation of rooms to name just a few of these applications. Two different problems can be treated. The first one, which has been extensively studied in the literature, is the case where the cavity is differentially heated with a single-sided [1–5] or double moving lid [6–8]. The second case consists of introducing some particles in the cavity and treating the mixed convection with mass transport [9–13].

Mixed-convection problems are not convenient to solve for optimization problems where multiple solutions are usually required, or for feedback control problems where real-time solutions are requested. In fact, such problems give rise to very large systems that cannot be easily solved numerically. Consequently, in order to decrease the computation time, it is necessary to develop and use fast numerical methods such as reduced-order models (ROMs).

The most popular reduced-order modeling technique is proper orthogonal decomposition (POD). This technique has been used to treat problems of particle dispersion [14, 15] and mixed-convection problems in cavities [16–19]. POD uses a set of snapshots generated by evaluating the computational solution of a transient problem at several time points. The POD basis is given by the most representative vector corresponding to the most dominant eigenvalue of the snapshot vector matrix. A low-order dynamic system is then obtained by the Galerkin projection of the problem onto this basis. Since N is small (a few dozen), the resolution of this system is fast.

The main drawback of POD is the need for snapshots of the solution in order to construct the reduced basis. A lengthy computing time may be required for the calculation of these snapshots. Furthermore, the basis depends on the information contained in the snapshots. As a result, the basis is not necessarily valid to study a phenomenon in a range of parameters different from the initially set parameters. To overcome these disadvantages it is possible to use the proper generalized decomposition (PGD) method, which consists of constructing iteratively a reduced basis without "a priori" knowledge of the solution. At each iteration of the iterative process, the basis is improved by adding a new vector.

This method involves looking for a solution to a partial differential equation as a product sum of the functions of each variables. For example, if we search a field w dependent on N variables, this can be expressed by

$$w(x_1, \cdots, x_N) \approx w_Q(x_1, \cdots, x_N) = \sum_{i=1}^{Q} \prod_{k=1}^{N} F_{ki}(x_k) \tag{1}$$

(x_i can be any scalar or vector variable involving space, time, or any other parameter of the problem, such as the conductivity coefficient, for example). Thus, if M degrees of freedom are used to discretize each variable, the total number of unknowns involved in the solution is $Q \times N \times M$ instead of the M^N degrees of freedom involved in mesh-based usual discretization techniques. In most cases, where the field is sufficiently regular, the number of terms Q in the finite sum is generally quite small (a few dozen), and in all cases the approximation converges toward the solution of the full grid description [20, 21.] It must be emphasized here that the functions are not known "a priori." These functions are adaptively computed by introducing the separated approximation of the representation into the model and then solving the resulting nonlinear problem.

PGD philosophy, originally called "radial time-space approximation," was first introduced by Ladevéze [22, 23] in the context of the LATIN method (Large Time Increment method) to reduce calculation cost as well in terms of memory storage as in terms of simulation time. The LATIN method was developed to solve unsteady nonlinear mechanical problems. It consists of solving iteratively a nonlinear local problem followed by a global linear problem, until convergence to the solution of the problem. Solving the full problem can be very costly; the introduction of PGD to solve this problem allowed the use of the LATIN method in cases where a classical resolution would have been impossible. In this case the solution $w(\mathbf{x}, t)$ of the problem defined on a

space-time domain is searched in the following form:

$$w(\mathbf{x}, t) \approx w_m(\mathbf{x}, t) = \sum_{i=1}^{m} F_1^i(\mathbf{x}) F_2^i(t)$$

The use of PGD combined with the LATIN method is also effective for solving mechanical problems involving different scales. It provides very satisfactory results with consequent reduction of the computation time and storage [24, 25]. More recently, and within the framework of homogenization problems, Lamari et al. [26] have applied PGD in order to determine, in the linear case, the homogenized thermal conductivity tensor of a microstructure.

PGD has also been applied to solve stochastic differential partial equations. In this context the solution w of a given stochastic problem is sought in the following form:

$$w \approx \sum_{i=1}^{m} \lambda^i F^i$$

where F^i are deterministic functions weighted by random coefficients λ^i. Nouy [27, 28] applies the method to solve stochastic linear equations (in this work, the PGD is initially named generalized spectral decomposition). Nouy shows that the method saves computing time and storage space compared with Krylov-type methods traditionally used in this context. Finally, Nouy and Le Maitre [29] proposed an extension of PGD for solving stochastic nonlinear differential equations.

Another area where PGD has shown its capacity is in the solving of multidimensional problems. In this case, the solution $u(x_1, \ldots, x_n)$ of a given problem is sought in totally separated form. So, as already mentioned, the approximation of order m of the solution is defined as

$$u_m(x_1, \cdots, x_n) = \sum_{i=1}^{m} F_1^i(x_1) \cdots F_N^i(x_n)$$

where $x_1, \ldots, x_n$ variables can represent spatial, temporal, or controls parameters (such as the Reynolds number) of the problem. In fact, seeking a solution as a decomposition on each variable can make the problem much easier (in terms of storage and computation time) to solve. Therefore, Ammar et al. [30–32] introduced PGD to treat the problems of description of fluids on a microscopic scale through the resolution of the Fokker-Planck equation. In this type of description, the number of degrees of freedom associated with a molecule (represented as a simple spring) is seven (three spatial dimensions, one temporal dimension, and three coordinates used to define the orientation and elongation of the spring). The method solved the Fokker-Planck equation using a number of unknowns equal to 10^5, so that the resolution would have necessitated a classic mesh size 10^{35}, which is impossible to treat with standard method. Chinesta et al. [33] browse different physical descriptions involving different scales (quantum mechanics, molecular dynamics, Brownian dynamics, kinetic theory) and focus on the difficulties encountered due to the large

number of degrees of freedom, and thus the benefits of the PGD method to address these problems. Recently, PGD was also used to solve problems of fluid mechanics [34, 35]. In this context, PGD is applied to solve the Navier-Stokes equation in order to reduce CPU time, by decreasing the cost of the space representation, and to obtain a sufficiently accurate solution. An incremental approach is keeping and the separation is doing over the physical space at each time step.

The aim of this work is to study the ability of PGD to treat the problems of mixed convection with and without mass transport. In the same way as what has been done in [34], a fractional step algorithm is used to decouple the velocity and pressure.

This article is organized as follows. First, we summarize briefly the numerical strategy used to solve the problem, including the governing equations and the discretization used to solve the problem. Furthermore, the PGD method is presented in this section. Then the PGD method is applied to solve mixed-convection problems with and without mass transfer in a 2-D square lid-driven cavity. Results obtained by PGD are compared to these obtained with a standard solver and with these of the literature.

2. GOVERNING EQUATIONS

In the following, the 2-D nonisothermal Navier-Stokes equations with mass transport will be described in dimensionless form. The fluid is considered Newtonian, incompressible, and with constant properties. The flow is laminar and the mixture is assumed to be perfect. Boussinesq approximation is performed, i.e., the fluid properties are assumed to be constant except for the density, ρ, which is assumed to vary linearly with both temperature and concentration [36, 37] such that $\rho = \rho_0[1 - \beta_T(T - T_c) - \beta_C(C - C_c)]$, where T_C and C_C are temperatures imposed by the boundary conditions. ρ_0 is a characteristic density; β_T and β_C are volumetric expansion coefficients with temperature and mass (concentration) fraction, respectively. The equations of the problem will be written in dimensionless form using the new dimensionless variables as follows:

$$\Theta = \frac{T - T_c}{T_h - T_c} \quad \Phi = \frac{C - C_c}{C_h - C_c} \quad \tau = \frac{tU_0}{d} \quad P = \frac{p}{\rho_0 U_0^2}$$
$$X = \frac{x}{d} \quad Y = \frac{y}{d} \quad U = \frac{u}{U_0} \quad V = \frac{v}{U_0}$$

(2)

where x and y are the Cartesian coordinates measured along the horizontal and vertical axes of the cavity, $\mathbf{u} = (u, v)$ is the velocity vector, T is the temperature of the fluid, C is the concentration of the species, and t is the time. U_0 and d are the characteristic velocity and length of the studied problem, $\mathbf{U} = (U, V)$ is the dimensionless velocity, Θ represents the dimensionless temperature, and Φ the dimensionless concentration. The equations of the problem in dimensionless form can be written as follows:

$$\begin{cases} \nabla \cdot \mathbf{U} = 0. \\ \frac{\partial \mathbf{U}}{\partial \tau} - \frac{1}{\mathrm{Re}}\nabla^2 \mathbf{U} + (\mathbf{U} \cdot \nabla)\mathbf{U} = -\nabla P + \mathrm{Ri}\left(\Theta + \frac{\mathrm{Gr}_C}{\mathrm{Gr}_T}\Phi\right) \cdot \mathbf{y} \\ \frac{\partial \Theta}{\partial \tau} - \frac{1}{\mathrm{Re}\,\mathrm{Pr}}\nabla^2 \Theta + (\mathbf{U} \cdot \nabla)\Theta = 0 \\ \frac{\partial \Phi}{\partial \tau} - \frac{1}{\mathrm{Re}\,\mathrm{Sc}}\nabla^2 \Phi + (\mathbf{U} \cdot \nabla)\Phi = 0 \end{cases}$$

(3)

where Re is the Reynolds number, Sc the Schmidt number, Ri the Richardson number, and Pr the Prandtl number. Gr_T and Gr_c are the thermal and solutal Grashof numbers, respectively. These dimensionless numbers are defined by

$$\mathrm{Re} = \frac{U_0 d}{\nu} \quad \mathrm{Sc} = \frac{\nu}{D} \quad \mathrm{Pr} = \frac{\nu}{\alpha} \quad \mathrm{Ri} = \frac{\mathrm{Gr}_T}{\mathrm{Re}^2}$$
$$\mathrm{Gr}_T = \frac{g\beta_T (T_h - T_c)d^3}{\nu^2} \quad \mathrm{Gr}_C = \frac{g\beta_C (C_h - C_C)d^3}{\nu^2} \tag{4}$$

where ν is the kinematic viscosity of the fluid, D the diffusion coefficient, and α the thermal diffusivity.

The Richardson number characterizes the importance of natural convection relative to the convection induced by forced training of the cavity on the top, while the thermal Grashoff number represents the strength of natural convection and the Reynolds number the strength of the forced convection. For a very small value of Richardson number we are in forced-convection regime, while Richardson is great when we are in a so-called natural convection.

3. RESOLUTION TECHNIQUE

3.1. Discretization of the Problem

The main difficulty encountered in the resolution of the Navier-Stokes equations lies in the velocity–pressure coupling within the continuity equation. In this work these fields are decoupled using a fractional step method which consists of solving the momentum equation in order to find an estimated velocity and then solving Poisson equation from this estimated velocity, thus providing a pseudo-pressure. These two equations are solved at each time step. Solving these two equations gives a reconstruction of the velocity and the pressure within these two estimated fields with respect to the continuity equations. In this work we use the Van-Kan algorithm (see [38]), which will be described in the following.

A Crank-Nicholson discretization is applied to the diffusion term, and the convective term, is discretized with an Adams-Bashforth scheme. The temperature and the concentration are discretized in an explicit way. Then, at time step $n+1$, we are looking for $\mathbf{U}^{n+1}$ and P^{n+1} such as

$$\begin{cases} \dfrac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\delta t} - \nabla^2 \left(\dfrac{\mathbf{U}^{n+1}}{\mathrm{Re}} \right) + \nabla P^{n+1} = -\dfrac{3}{2} \nabla \cdot (\mathbf{UU})^n + \dfrac{1}{2} \nabla \cdot (\mathbf{UU})^{n-1} \\ \qquad\qquad + \mathrm{Ri}\left(\Theta^n + \dfrac{\mathrm{Gr}_C}{\mathrm{Gr}_T} \Phi^n \right) \cdot \vec{\mathbf{y}} \quad \text{in } \Omega \\ \nabla \cdot \mathbf{U}^{n+1} = 0 \quad \text{in } \Omega \\ \mathbf{U}^{n+1} = \mathbf{f} \quad \text{on } \partial\Omega \end{cases} \tag{5}$$

where $\mathbf{f} = (f_x, f_y)$ corresponds to the imposed boundary conditions on $\partial\Omega$. The intermediate velocity is then defined by

$$\mathbf{U}^{n+1} = \widetilde{\mathbf{U}} - \frac{\delta t}{2}\nabla\widetilde{P} \tag{6}$$

where $\widetilde{P} = P^{n+1} - P^n$ is the pressure correction.

Injecting (6) in problem (5) and neglecting the second-order term $\left(\frac{\delta t}{\text{Re}}\right)\nabla^2(\nabla\widetilde{P})$, the intermediate velocity $\widetilde{U}$ satisfies

$$
\begin{cases}
\dfrac{\widetilde{\mathbf{U}} - \mathbf{U}^n}{\delta t} - \nabla^2\left(\dfrac{\widetilde{\mathbf{U}} - \mathbf{U}^n}{2\,\text{Re}}\right) = -\nabla P^n - \dfrac{3}{2}\nabla\cdot(\mathbf{UU})^n + \dfrac{1}{2}\nabla\cdot(\mathbf{UU})^{n-1} \\
\qquad\qquad\qquad\qquad + \text{Ri}\left(\Theta^n + \dfrac{\text{Gr}_C}{\text{Gr}_T}\Phi^n\right)\cdot\vec{\mathbf{y}} \quad \text{in } \Omega \\
\widetilde{\mathbf{U}} = \mathbf{f} \quad \text{on } \partial\Omega
\end{cases}
\tag{7}
$$

Then, by taking the divergence of Eq. (6), we get the following problem for the pressure correction:

$$
\begin{cases}
\nabla^2\widetilde{P} = \frac{2}{\delta t}\nabla\cdot\widetilde{\mathbf{U}} \\
\frac{\partial\widetilde{P}}{\partial n} = 0 \quad \text{on } \partial\Omega
\end{cases}
\tag{8}
$$

Therefore, in order to compute $\mathbf{U}^{n+1}$ and P^{n+1}, systems (5) and (8) must be solved successively to obtain the intermediate velocity $\widetilde{\mathbf{U}}$ and therefore the pressure correction $\widetilde{P}$. Finally the velocity and the pressure at time step $n+1$ are updated using Eq. (6) and the definition of the pressure correction.

In the same way, by using a Cranck-Nicholson scheme to discretize the diffusive term and an Adams-Bashforth scheme to discretize the convective term, the temperature is a solution of

$$
\begin{cases}
\dfrac{\Theta^{n+1}}{\delta t} - \dfrac{1}{2\,\text{Re}\,\text{Pr}}\nabla^2\Theta^{n+1} = \dfrac{\Theta^n}{\delta t} + \dfrac{1}{2\,\text{Re}\,\text{Pr}}\nabla^2\Theta^n - \dfrac{3}{2}(\mathbf{U}^n\cdot\nabla)\Theta^n \\
\qquad\qquad + \dfrac{1}{2}(\mathbf{U}^{n-1}\cdot\nabla)\Theta^{n-1} \quad \text{in } \Omega \\
\Theta^{n+1} = \Theta_1 \text{ on } \partial\Omega_1 \\
\nabla\Theta^{n+1}\cdot\mathbf{n} = \Theta_2 \quad \text{on } \partial\Omega_2
\end{cases}
\tag{9}
$$

and the concentration is a solution of

$$
\begin{cases}
\dfrac{\Phi^{n+1}}{\delta t} - \dfrac{1}{2\,\text{Re}\,\text{Sc}}\nabla^2\Phi^{n+1} = \dfrac{\Phi^n}{\delta t} + \dfrac{1}{2\,\text{Re}\,\text{Sc}}\nabla^2\Phi^n \\
\qquad\qquad - \dfrac{3}{2}(\mathbf{U}^n\cdot\nabla)\Phi^n + \dfrac{1}{2}(\mathbf{U}^{n-1}\cdot\nabla)\Phi^{n-1} \quad \text{in } \Omega \\
\Phi^{n+1} = \Phi_1 \quad \text{on } \partial\,\Omega_1 \\
\nabla\Phi^{n+1}\cdot\mathbf{n} = \Phi_2 \quad \text{on } \partial\Omega_2
\end{cases}
\tag{10}
$$

with $\partial\Omega_1 \bigcup \partial\Omega_2 = \partial\Omega$ and $\partial\Omega_1 \bigcap \partial\Omega_2 = \emptyset$.

3.2. PGD Algorithm

3.2.1. Strategy. In the context of transient problems, in order to compute the unknown w, PGD can be applied using one of the three following decompositions.

1. The first possibility consists of a time space decomposition:

$$w(t, \mathbf{x}) = \sum_i F_t^i(t) F_\mathbf{x}^i(\mathbf{x})$$

In this case we are looking for a similar representation to an "a priori" reduced model but obtained using a nonincremental approach. We are looking directly for a time/space solution. The main drawback encountered here is that a full grid description is required to define the F_x functions over the physical space.

2. To circumvent this difficulty, the second possibility consists of writing a full decomposition involving the two dimensions of the physical space:

$$w(t, x, y) = \sum_i F_t^i(t) F_x^i(x) F_y^i(y)$$

In practice, making such a decomposition leads to a significant increase in the number of terms required in the sum.

3. The third possibility consists of keeping the incremental approach and doing the separation over the physical space at each time step n:

$$w^n(x, y) = \sum_i F_x^i(x) F_y^i(y)$$

This last decomposition will be used in this work.

Thus, in this article, the velocity fields, the pressure, the temperature, and the mass concentration are searched at each time step on the following separated form:

$$\widetilde{U}^{n+1}(x, y) = \sum_{i=1}^{n_{\widetilde{U}}} \alpha_{\widetilde{U}}^i F_{\widetilde{U}}^i(x) G_{\widetilde{U}}^i(y) \quad \widetilde{V}^{n+1}(x, y) = \sum_{i=1}^{n_{\widetilde{V}}} \alpha_{\widetilde{V}}^i F_{\widetilde{V}}^i(x) G_{\widetilde{V}}^i(y)$$

$$\widetilde{P}^{n+1}(x, y) = \sum_{i=1}^{n_{\widetilde{P}}} \alpha_{\widetilde{P}}^i F_{\widetilde{P}}^i(x) G_{\widetilde{P}}^i(y) \quad \Theta^{n+1}(x, y) = \sum_{i=1}^{n_\Theta} \alpha_\Theta^i F_\Theta^i(x) G_\Theta^i(y) \quad (11)$$

$$\Phi^{n+1}(x, y) = \sum_{i=1}^{n_\Phi} \alpha_\Phi^i F_\Phi^i(x) G_\Phi^i(y)$$

where each function F_w, G_w and integer n_w (w could be $\widetilde{U}$, $\widetilde{V}$, $\widetilde{P}$, Θ, or Φ) depend of the time $n+1$. In order to alleviate the notation, this dependency is not written explicitly.

All the unknowns of the problem, except the pressure, are each the solution, after temporal discretization, of equations which all have the same generic structure, namely,

$$\frac{w^{n+1}}{\delta t} - \frac{\beta_w}{2} \nabla^2 w^{n+1} = g_w^{n+1} \quad (12)$$

Thus, for the sake of clarity, the PGD procedure will be described to solve the general equation (12). PGD, which is an iterative method, consists of finding, at time

step $n+1$, an approximation of the solution $w^{n+1}(x, y) \in \Omega = X \times Y \subset \mathbb{R}^2$ with $x \in X \subset \mathbb{R}$ and $y \in Y \subset \mathbb{R}$ as

$$w^{n+1}(x,y) \approx w_m^{n+1}(x,y) = \sum_{i=1}^{m} \alpha_w^i F_w^i(x) G_w^i(y) \tag{13}$$

where $w_m^{n+1}(x, y)$ is the approximation of the solution of order m. At each iteration, the solution is enriched with an additional term $\alpha_w^{m+1} F_w^{m+1}(x) G_w^{m+1}(y)$. PGD should be decomposed in three steps. During the first step, called "the enrichment step," the F_w^{m+1} and G_w^{m+1} functions are obtained by solving a small-size nonlinear problem. Then, for the second step, called the "projection step", in order to improve the quality of the reconstruction, the $m+1$ α_w^{m+1} coefficients are determined by solving a linear system of size $(m+1)$. Finally, the "check convergence step" consists of computing the norm of the full model residual in order to decide if the solution needs more enrichment or not. In the following these three steps will be described in detail.

3.2.2. Enrichment Step. At the mth stage, the solution approximation of order $m-1$ is supposed to be known. At this step, we search to compute the functions $F_w^m(x)$ and $G_w^m(y)$. We search $w_m^{n+1}(x, y)$ as

$$w_m^{n+1}(x,y) = w_{m-1}^{n+1}(x,y) + F_w^m(x) G_w^m(y) = \sum_{i=1}^{m-1} \alpha_w^i F_w^i(x) G_w^i(y) + F_w^m(x) G_w^m(y) \tag{14}$$

Introducing Eq. (14) into Problem (12) gives

$$\frac{F_w^m G_w^m}{\delta t} - \frac{\beta_w}{2} \left(\frac{d^2 F_w^m}{dx^2} G_w^m + F_w^m \frac{d^2 G_w^m}{dy^2} \right) = S_w^m + \text{Res}_w^m \tag{15}$$

where Res_w^m is a residual due to the fact that w_m^{n+1} is an approximation of the solution. The second member is given by

$$S_w^m = g_w^{n+1} - \sum_{i=1}^{m-1} \alpha_w^i \left[\frac{F_w^i G_w^i}{\delta t} - \frac{\beta_w}{2} \left(\frac{d^2 F_w^i}{dx^2} G_w^i + F_w^i \frac{d^2 G_w^i}{dy^2} \right) \right] \tag{16}$$

Equation (15) is then projected onto each of the unknowns F_w^m and G_w^m and the residual Res_w^m is forced to be orthogonal to each of these functions. This gives the two following problems:

$$\left\langle \frac{F_w^m G_w^m}{\delta t} - \frac{\beta_w}{2} \left(\frac{d^2 F_w^m}{dx^2} G_w^m + F_w^m \frac{d^2 G_w^m}{dy^2} \right), F_w^m \right\rangle_{\widehat{L^2(X)}} = \langle S_w^m, F_w^m \rangle_{\widehat{L^2(X)}} \tag{17}$$

$$\left\langle \frac{F_w^m G_w^m}{\delta t} - \frac{\beta_w}{2} \left(\frac{d^2 F_w^m}{dx^2} G_w^m + F_w^m \frac{d^2 G_w^m}{dy^2} \right), G_m \right\rangle_{\widehat{L^2(Y)}} = \langle S_w^m, G_w^m \rangle_{\widehat{L^2(Y)}} \tag{18}$$

where $<\cdot,\cdot>\widehat{_{L^2(x)}}$ ($<\cdot,\cdot>\widehat{_{L^2(y)}}$, respectively) are the scalar products on L^2, in the **x** direction, (in the **y** direction, respectively).

Equations (17) and (18) are solved using the fixed-point method. After convergence of the fixed-point method, the functions F_w^m and G_w^m are normalized. It is important to note that the resolution of this system of two sets of nonlinear equations by a fixed-point method corresponds to the resolution of two linear systems of size N. Instead of solving a large 2-D problem, PGD consists of solving many 1-D problems. For example, if we consider a problem with N nodes of discretization in each direction, PGD consists of solving some linear systems of size N, while a classic approach consists of solving a linear system of size N^2. Thus, if N is large enough, the PGD will be faster than traditional methods.

3.2.3. Projection Step. In order to increase the accuracy of the decomposition, the m α_w^i coefficients are now searched in such a way that the residual is orthogonal to each of the m products of the $F_w^i G_w^i$ functions. At this step, we search $w_m^{n+1}(x, y)$ as

$$w_m^{n+1}(x, y) = \sum_{i=1}^{m} \alpha_w^i F_w^i(x) G_w^i(y) \tag{19}$$

Then, by introducing this expression into Eq. (12),

$$\sum_{i=1}^{m} \alpha_w^i \left[\frac{F_w^i G_w^i}{\delta t} - \frac{\beta_w}{2} \left(\frac{d^2 F_w^i}{dx^2} G_w^i + F_w^i \frac{d^2 G_w^i}{dy^2} \right) \right] = g_w^{n+1} + \text{Res}_w^m \tag{20}$$

By projecting the above equation according to the $F_w^i G_w^i$, a system satisfied by the α_w^i coefficients is obtained. For $1 \le k \le m$,

$$\left\langle \sum_{i=1}^{m} \alpha_w^i \left[\frac{F_w^i G_w^i}{\delta t} - \frac{\beta_w}{2} \left(\frac{d^2 F_w^i}{dx^2} G_w^i + F_w^i \frac{d^2 G_w^i}{dy^2} \right) \right], \quad F_w^k G_w^k \right\rangle \widehat{_{L^2(\Omega)}} \tag{21}$$
$$= \langle g_w^{n+1}, F_w^k G_w^k \rangle \widehat{_{L^2(\Omega)}}$$

Equation (21), which represents a small linear system (of size m), can be easily solved by using a classical solver. At this point we know the approximation of the solution of order m, $w_m^{n+1}(x, y)$.

3.2.4. Check Convergence Step. If the L^2 norm of this residual, defined by

$$\text{Res}_w^m = \sum_{i=1}^{m} \alpha_w^i \left[\frac{F_w^i G_w^i}{\delta t} - \frac{\beta_w}{2} \left(\frac{d^2 F_w^i}{dx^2} G_w^i + F_w^i \frac{d^2 G_w^i}{dy^2} \right) \right] - g_w^{n+1}(x, y) \tag{22}$$

is lower than a coefficient ϵ set by the user, the PGD algorithm is converged. Otherwise, one more iteration at least is needed, and the enrichment and projection steps are repeated until convergence, which increases the value m gradually.

3.2.5. Summary of the Algorithm. The PGD resolution algorithm combining the enrichment phase and the computation of the coefficients using a projection based on an approximation called is summarized below.

1. **for** $m = 1$, m_{max} **do**
2. Solving equations (17) and (18) by a fixed point method in order to compute F_w^m and G_w^m.
3. Computation of $\{\alpha_w^i\}_{i=1}^m$ by solving equation (21)
4. Then $w_m = \sum_{i=1}^m \alpha_w^i F_w^i G_w^i$
5. Computation of the residual defined by Eq. (22)
6. Check convergence: if $\|\mathrm{Res}_w \leq \varepsilon\|$ then **stop**
7. **end for**

In the following, the only equations to solve are the unsteady and the steady diffusion equation. For the sake of clarity, PGD was described in a continuous form; however, this is the algebraic form of the algorithm which is used in practice. That is why an algebraic version of the resolution of these equations is given in the Appendix.

4. APPLICATION OF PGD TO SOLVE A MIXED-CONVECTION PROBLEM

4.1. Problem Description

We consider an incompressible Newtonian fluid within a square cavity. Vertical walls are differentially heated and isothermal, while the left wall has a higher temperature than the right wall and horizontal walls are kept adiabatic. The top lid of the cavity moves in the $\vec{x}$ direction with constant velocity U_0. In this case the problem (3) is solved regardless of the equation governing the concentration and taking that $Gr_C = 0$. Figure 1 presents the studied case.
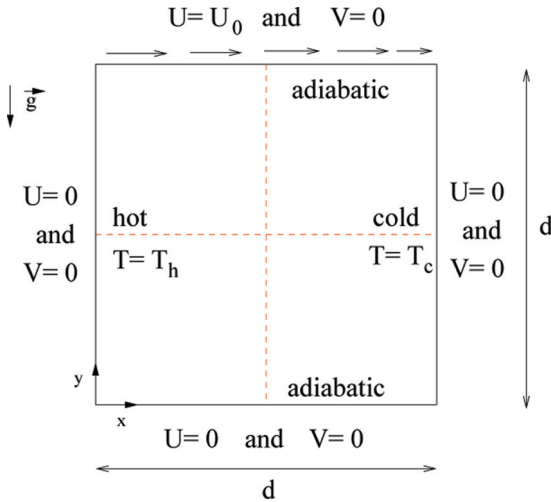

Figure 1. Configuration for studied case (color figure available online).

Simulations have been done with a time step equal to $\delta t = 10^{-3}$. Since we were looking for a stationary flow, we defined the following convergence criteria:

$$\frac{\|u_i^k - u_i^{k-1}\|}{\|u_i^k\|} \le \epsilon_u \quad \text{for } i = 1, 2 \quad \frac{\|p^k - p^{k-1}\|}{\|p^k\|} \le \epsilon_p \quad \frac{\|T^k - T^{k-1}\|}{\|T^k\|} \le \epsilon_T$$

In the following cases the values $\epsilon_u = 10^{-10}$, $\epsilon_p = 10^{-8}$, and $\epsilon_T = 10^{-10}$ were considered. The simulations were performed using the PGD method and using a traditional solver (bi-conjugate gradient) for a Grashoff number set at 10^5 and a Prandtl number equal to 0.71 (characteristic of the air). In the following, three Richardson numbers will be considered (Ri = 0.1, Ri = 1, and Ri = 10). N_h represents the number of nodes in each direction.

4.2. Results

Figures 2 and 3 compare the velocity field and isotherms obtained by PGD with those obtained by Sun et al. [39]. We note that for Ri = 10, the flow is mainly due to the temperature gradient and we can see a double vortex at the center of the cavity. For the case where the Richardson number is equal to 1, forced convection due to imposed velocity on the top of the cavity starts to become important, it shows especially by observing the size of the boundary layer temperature on the left wall of the cavity. Finally, for a Richardson number equal to 0.1, we observe that the forced convection becomes dominant in relation to temperature. In this case the results are very similar to those obtained for the 2D isothermal lid driven cavity for Re = 1000[1]. Concerning the isotherms, we observe that more Richardson number is important more the temperature tends to stratify. These figures show that whatever the Richardson number considered, PGD results are very closed to the results of the literature.

In order to compare more precisely the PGD results with those of the literature, the Nusselt numbers on the right wall (east) (Nu_e) and on the left wall (west) (Nu_w) are defined as

$$\text{Nu}_e = -\frac{(\partial \Theta / \partial X)_e}{\Theta_h - \Theta_c} \quad \text{Nu}_w = -\frac{(\partial \Theta / \partial X)_w}{\Theta_h - \Theta_c} \tag{23}$$

and the average Nusselt numbers, calculated by integrating the local Nusselt along the wall, as

$$\overline{\text{Nu}_e} = \int_0^1 \text{Nu}_e \, dY \quad \overline{\text{Nu}_w} = \int_0^1 \text{Nu}_w \, dY \tag{24}$$

Average Nusselt numbers at convergence of the chosen method (PGD or standard solver without separated representation) defined in Eq. (24) are compared with those obtained by Sun et al. [39]. Results are listed in Table 1. It can be noticed that the average Nusselt number increases with decreasing

[1]Recall that $\text{Re} = \sqrt{\text{Gr}/\text{Ri}}$ and thus $\text{Ri} = 0.1 \Rightarrow \text{Re} = 1000$
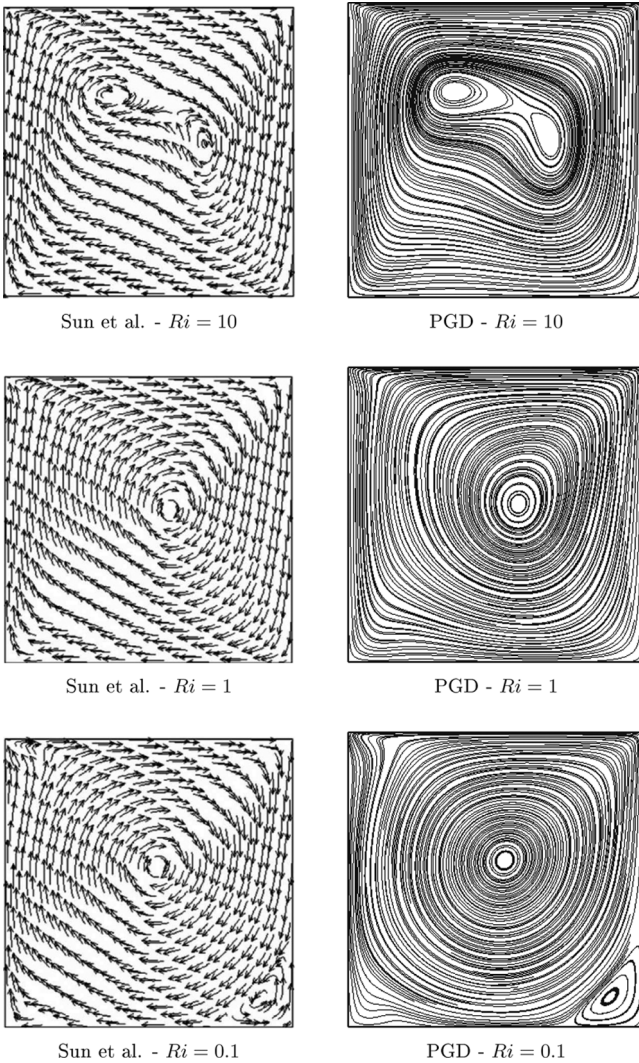
Figure 2. Velocity vectors from Sun et al. [39] (left) and streamlines from PGD (right) for different Richardson numbers and $N_h = 100$.

Richardson number, which indicates that the more dynamic effects are stronger the higher the heat transfer on the left and right borders of the cavity is important. We can also note that the average Nusselt numbers obtained by PGD are similar to those by Sun et al., even if a slight difference is observed for $\mathrm{Ri} = 1$ and $\mathrm{Ri} = 10$. This difference is not due to PGD, since the same results are obtained with the standard solver.

Figure 4 shows also a good match between the simulations and the literature concerning the Nusselt numbers. Indeed, Nusselt numbers at the west and at east calculated with the standard solver and with the PGD are the same.
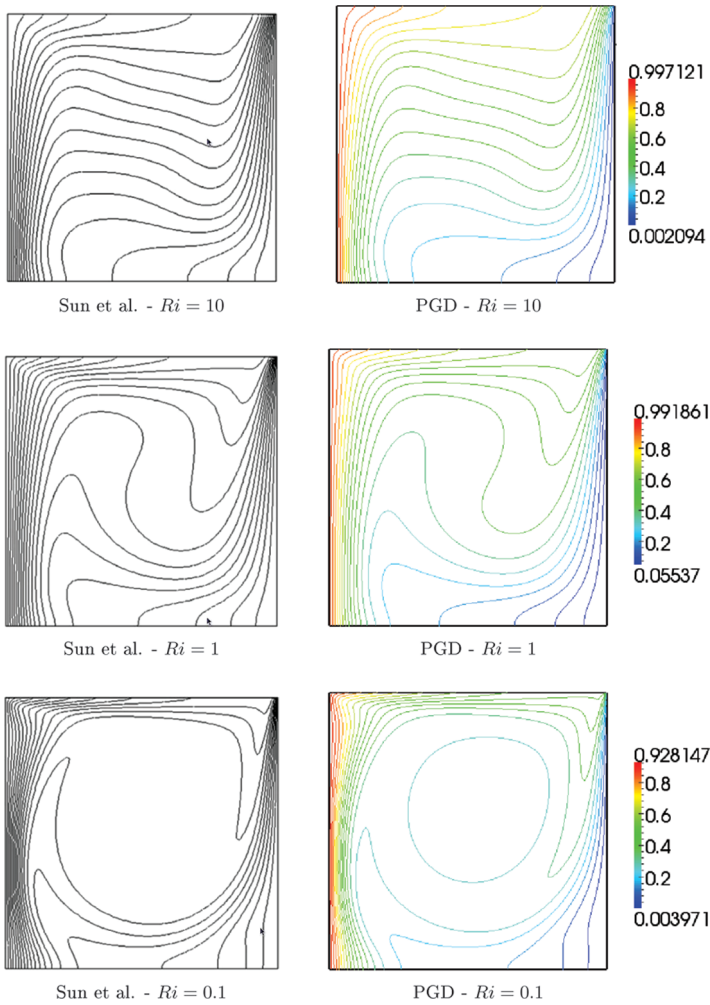
Figure 3. Isotherms from Sun et al. [39] and from PGD for Richardson numbers equal to 10, 1, and 0.1 (color figure available online).

4.3. Computational Time Comparison

The ratio of computation time between the standard solver and the PGD method is recorded in Table 2 for simulations done with a number of nodes N_h ranging from 61 to 301. First, it can be noticed that from 200 nodes in each direction, PGD is faster than the standard solver for each of the Richardson numbers considered. Second, the higher the Richardson number, the more the time saving by using PGD is important. Indeed, with $N_h = 301$, at $Ri = 10$, PGD is six times faster than the standard solver, whereas with the same number of nodes, at $Ri = 0.1$, PGD is four times faster than the standard solver. From these results it can be concluded that the higher the Richardson number (the more the effect of temperature is important in comparison with forced convection), the more the PGD method is efficient.

Table 1. Values of average Nusselt numbers on the right wall (east) and on the left wall (west) from the literature, from PGD and from standard solver for Richardson numbers equal to 0.1, 1, and 10

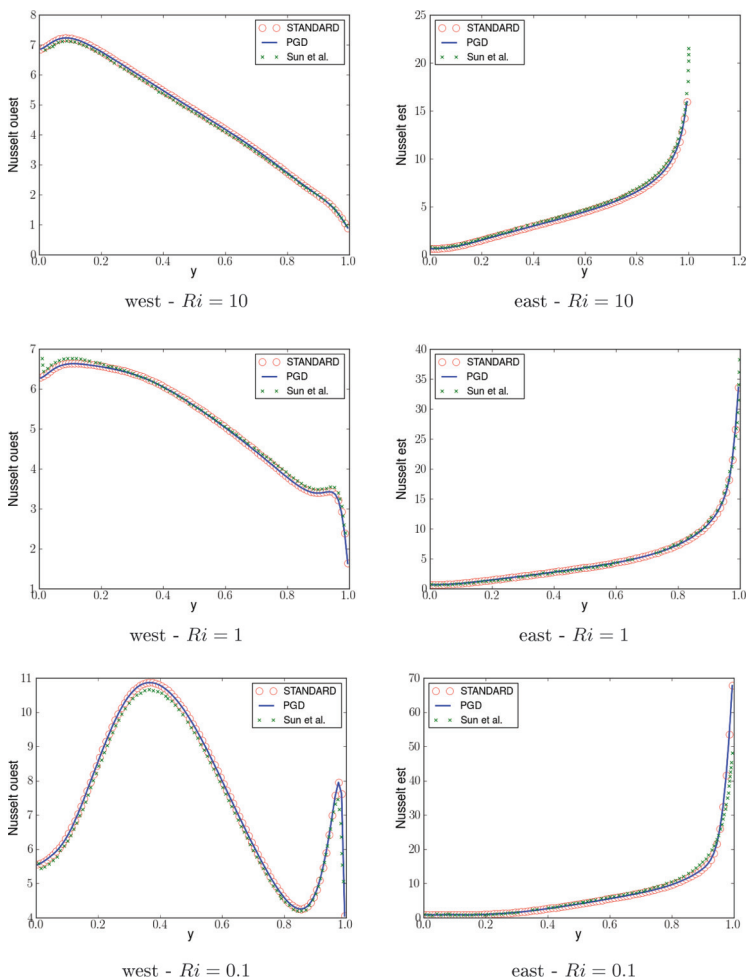|	$\overline{Nu}_{east}$			$\overline{Nu}_{west}$		
	Sun et al.	PGD	Standard	Sun et al.	PGD	Standard
$Ri = 10$	4.61	4.49	4.49	4.61	4.54	4.54
$Ri = 1$	5.29	5.19	5.20	5.29	5.20	5.20
$Ri = 0.1$	7.46	7.44	7.45	7.46	7.48	7.48

Figure 4. Values of Nusselt numbers on the western and eastern walls from the literature, the standard model, and PGD for Richardson numbers equal to 0.1, 1, and 10 (color figure available online).

Table 2. Ratio of computation time for solving the Navier-Stokes isothermal equations

Number of nodes in each direction	(Standard computation time)/(PGD computation time)		
	Ri = 10	Ri = 1	Ri = 0.1
61	0.2	0.14	0.17
101	0.5	0.7	0.57
151	1.25	1.06	0.82
201	4.4	3.67	1.60
301	6.33	5.20	3.7

4.4. Optimality of the PGD

Singular value decomposition (SVD) [40, 41] is a procedure which decomposes optimally a 2-D field into 1-D product functions. So, if the PGD method is optimal, the number of PGD functions should be the same as (or very close to) the number of
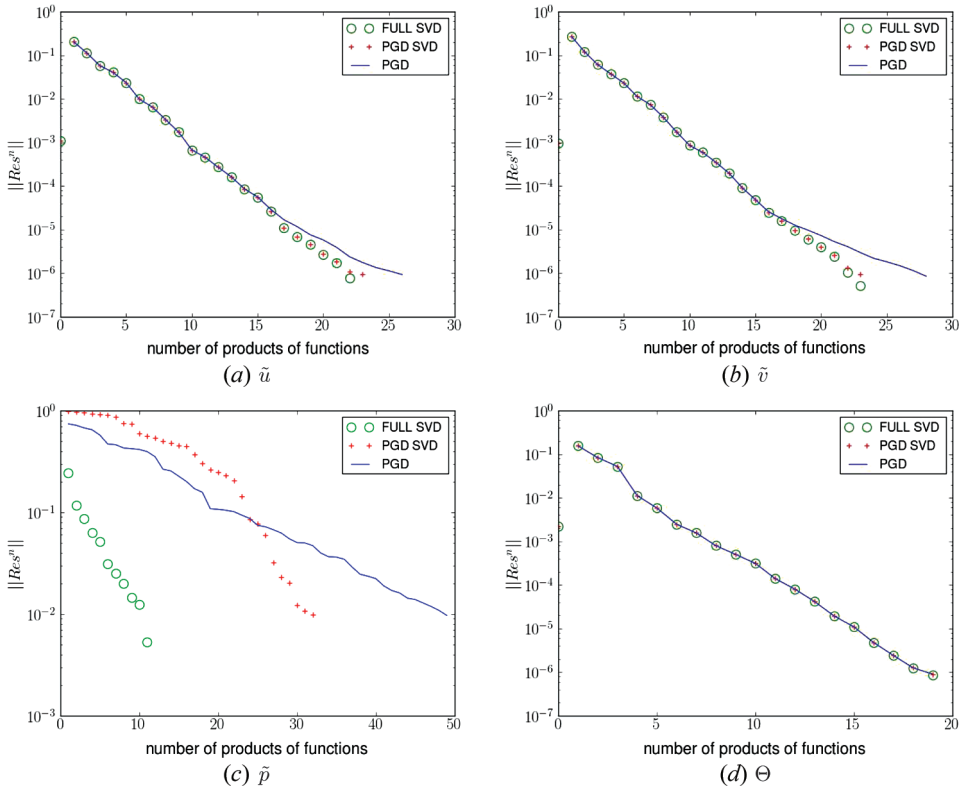

Figure 5. Norm of the residual depending of the number of functions keeping for each of PGD decomposition, SVD on the PGD result (PGD SVD) and SVD on the standard model results (FULL SVD) for the horizontal (a) and vertical (b) velocity and for the pressure correction (c) and the temperature (d) with $Ri = 1$ and $N_h = 201$ (color figure available online).

functions obtained by SVD. To discuss the optimality of the PGD, the number of PGD functions is compared with the number of functions from a SVD. To achieve this, the evolution of the norm of the residual of each equations solved with regard to the number of PGD functions or SVD functions kept is considered. The residual is defined as

$$\text{Res}_w^m = \mathcal{A}_w \left[\sum_{i=1}^{m_w} F_w^i G_w^i \right] - SM_w \tag{25}$$

where w corresponds to the different unknowns of the problem, namely, the horizontal and vertical velocities, the pressure, or the temperature. $\mathcal{A}_w$ (respectively SM_w) is the operator (resp. the second member) associated to the equation which is used to calculate the variable w. Functions F_w^i, G_w^i are the PGD functions or the SVD functions of the unknown w. In the following, we de-note by "FULL-SVD" the results obtained with a SVD on the standard solver solution, and by "PGD-SVD" the results obtained with a SVD on the PGD solution.

The norm of the residual associated to the equation satisfied by the velocity, the pressure, and the temperature with regard to the number of functions m_u, m_v (resp. m_p, m_Θ) retained are plotted in Figure 5 with $N_h = 201$ and $\text{Ri} = 1$. The residual associated to the velocity and temperature obtained by PGD, PGD-SVD, and FULL-SVD gives the same curve of residual with regard to the number of products of function kept. It proves that, for these variables, the PGD gives the same optimal representation as can be provided by a SVD of the full solution.

Concerning the pressure, Figure 5 c shows that the PGD is able to find a solution with a norm of the residual inferior to 10^{-2} with approximatively 50 products
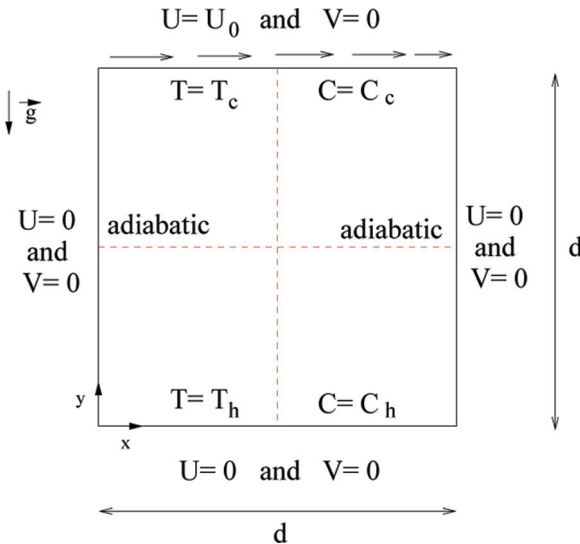

Figure 6. Geometry of the nonisothermal cavity with mass transport (color figure available online).

of functions, while FULL-SVD needs only 10 products of functions to reach the same norm of the residual. In this case it is clear that the PGD is not optimal. Moreover, the PGD-SVD gives a norm of the residual different from the FULL-SVD, which implies that the solution obtained by PGD is slightly noisy compared with the solution obtained by the standard solver. Results obtained for $Ri = 0.1$ and $Ri = 10$ are quite the same as those obtained with $Ri = 1$, which is why they are not plotted here. The performance of the global PGD algorithm is somewhat reduced by the resolution of the pressure problem. In fact, it is not possible to be more optimal for the resolution of the other equations.
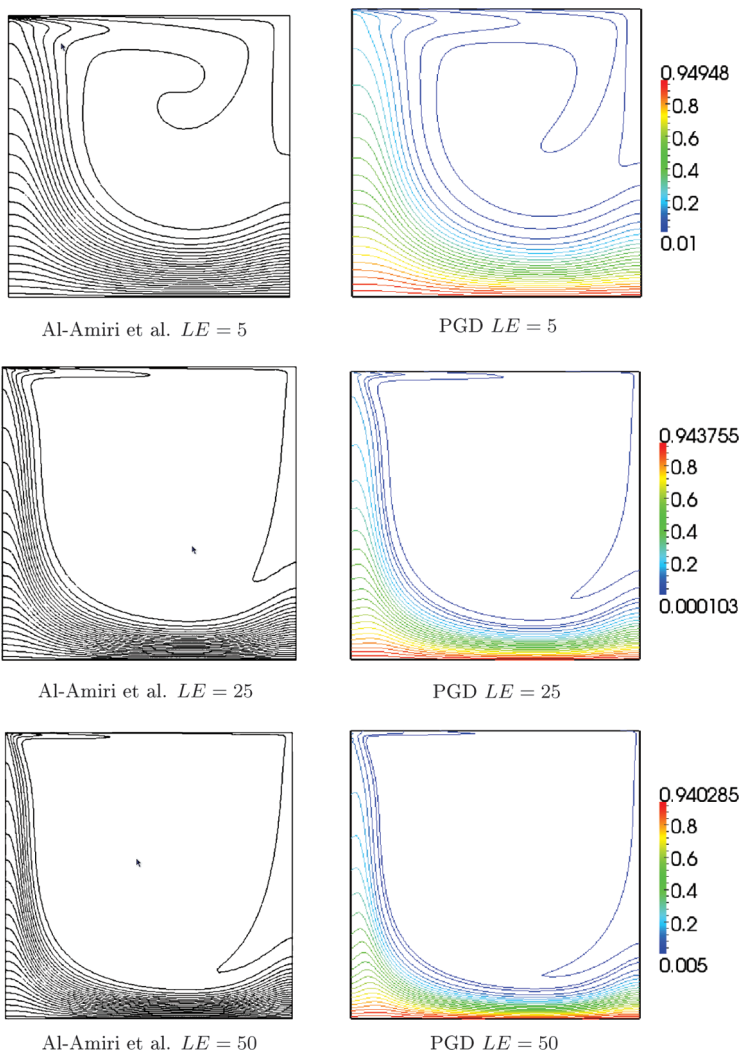

Figure 7. Isoconcentrations from literature and from the PGD for different Lewis numbers with $N_h = 250$ (color figure available online).

5. APPLICATION OF PGD TO SOLVE A MIXED-CONVECTION PROBLEM WITH MASS TRANSPORT

5.1. Problem Description

The cavity studied here is such that its north wall is animated by a constant horizontal velocity equal to U_0, while the velocity on all other walls is equal to zero. A temperature gradient from south to north is imposed, i.e., the temperature T_h imposed on the south side is higher than the temperature imposed on the north face T_c. In the same way, the concentration of species on the south side C_h is greater than the concentration imposed on the north side, denoted C_C. The east and west sides are considered adiabatic and impermeable. Figure 6 shows the cavity studied.

The following simulations are done with $Pr = 1$ and $Re = 100$. N_h represents the number of nodes in each direction. A time step equal to $\Delta t = 10^{-3}$ and a convergence criterion on the concentration is added to those defined above in the isothermal case [Eq. (23)]:

$$\frac{\|C^k - C^{k-1}\|}{\|C^k\|} \le \epsilon_C \tag{26}$$

Here the convergence criteria have been taken equal to $\epsilon_u = 10^{-10}$, $\epsilon_p = 10^{-10}$, $\epsilon_T = 10^{-10}$, and $\epsilon_C = 10^{-10}$.

Streamlines Al-Amiri et al. $Le = 5$ Streamlines PGD $Le = 5$

Isotherms Al-Amiri et al. $Le = 5$ Isotherms PGD $Le = 5$

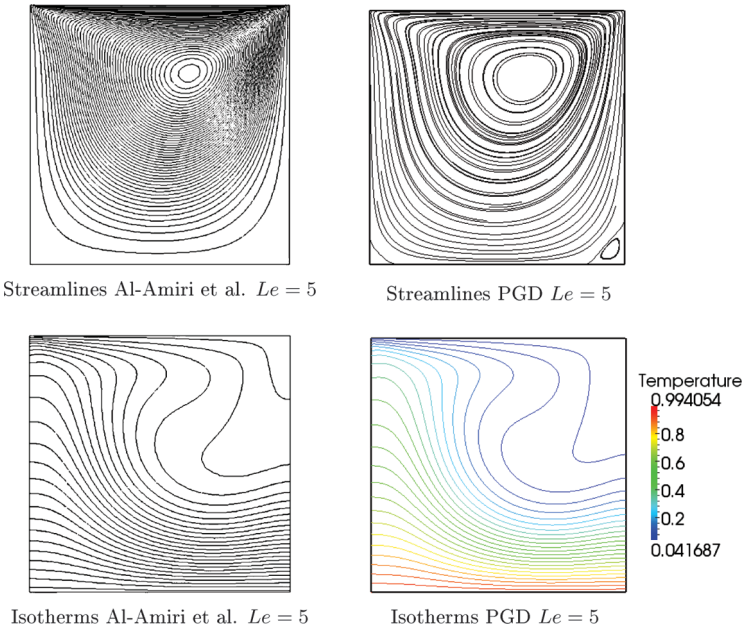Temperature
0.994054
0.8
0.6
0.4
0.2
0.041687

Figure 8. Streamlines and isotherms from Al-Amiri et al. [42] and from PGD for a Lewis number equal to 5 (color figure available online).

5.2. Results

In a first time, we fix the Richardson number equal to 0.01 and the Grashof number to 10^2. The Lewis number, defined by $Le = Sc/Pr$, varies from 5 to 50. Figure 7 shows a good match between the isoconcentrations obtained by PGD and those of Al-Amiri et al. [42]. In this case, we see that the boundary layer on the southern wall appears to decrease with increase of the Lewis number, which is due to the fact that when the Lewis number increases, the diffusivity of the species

Al-Amiri et al. $Gr_C/Gr_T = 0$ PGD $Gr_C/Gr_T = 0$

Al-Amiri et al. $Gr_C/Gr_T = -25$ PGD $Gr_C/Gr_T = -25$

Al-Amiri et al. $Gr_C/Gr_T = -50$ PGD $Gr_C/Gr_T = -50$

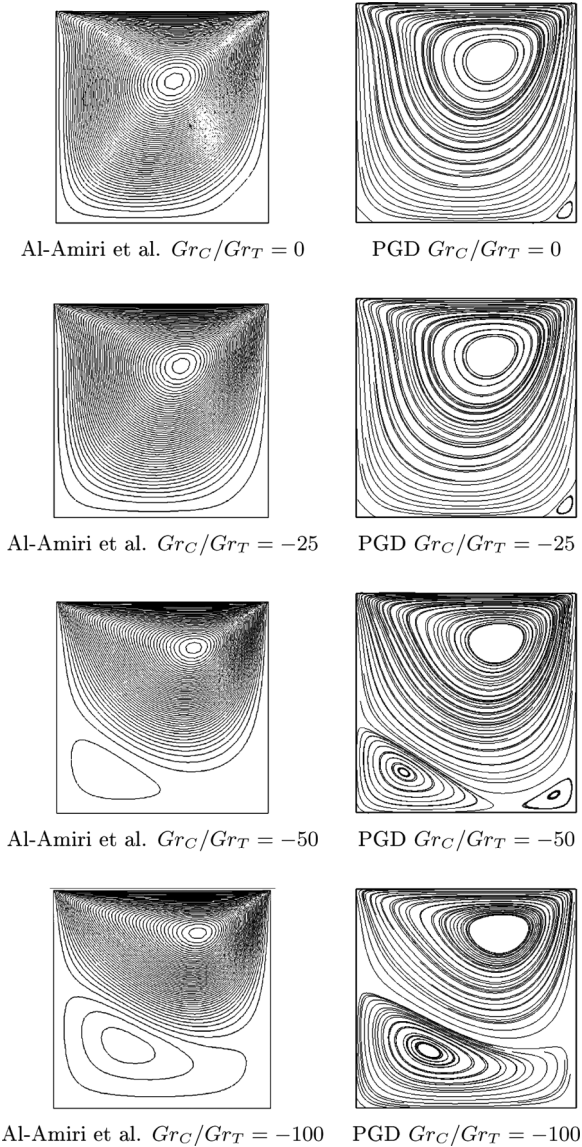Al-Amiri et al. $Gr_C/Gr_T = -100$ PGD $Gr_C/Gr_T = -100$

Figure 9. Streamlines from the literature and from PGD for different value of the ratio Gr_C/Gr_T.

Table 3. Ratio of computation time for solving the Navier-Stokes isothermal equations with mass transport

Number of nodes in each direction	(Standard computation time)/(PGD computation time)			
	$N = 0$	$N = -25$	$N = -50$	$N = -100$
61	0.27	0.18	0.13	0.1
101	0.53	0.45	0.31	0.3
201	1.93	1.83	7.1	7.1
301	5.47	6.42	9.42	9.42

decreases. In addition, regardless of the Lewis number (Le = 5, 25, or 50), the velocity field and temperature remain unchanged, because of the dominant forced convection, which had already been noted by Al-Amiri et al. In Figure 8 we plot the streamlines and isotherms for Le = 5. Once more, we see that the PGD results are similar to those of the literature.

In a second time, we fix the Richardson number to 0.01, the Reynolds number at 100, which requires a value of thermal Grashoff number equal to 100. The ratio Gr_T/Gr_c is taken equal to 0, -25, -50 and -100. This implies that the solutal Grashoff numbers varies from 0 to $-10, 000$. The streamlines obtained by PGD are compared with those obtained by Al-Amiri et al. [42] in Figure 9. This figure shows that the PGD results are very close to the results of the literature. In this case the temperature fields are the same as the concentration fields, so they are not represented in the following. Indeed, if Le = 1 and Pr = 1, it implies that Sc = 1 and equations which gives the temperature and the concentrations are exactly the same.

5.3. Computation Time

The ratio of the computation time with the standard solver on computation time required for PGD are reported in Table 3. The simulations were performed for a number of nodes ranging from 61 to 301 in each direction and with Le = 1. We observe that from 200 nodes in each direction, PGD is faster than the standard solver for each of the Richardson numbers considered. The more ratio N is important, the more the time saving by PGD is important. Indeed, with $N_h = 301$ and $N = 0$, PGD is six times faster than the standard solver, while with $N = -100$, PGD is 10 times faster than the standard solver. So, the more the solutal effects are important, the more the PGD is efficient in comparison to the standard solver. In this case, optimality curves are not presented because the results are identical to those obtained for mixed convection without mass transport; in this case again, only the pressure equation is not optimal.

6. SUMMARY AND CONCLUSIONS

To conclude, one can note the performance of PGD to treat mixed-convection problems with and without mass transfer. Indeed in the first test case (i.e., without

mass transfer), we have demonstrated that PGD is able to provide good values of average Nusselt in comparison with the standard solver and with results of the literature. Moreover, values, of Nusselt numbers at west and at east are very close to these given by the literature. In addition, the time saved by using the PGD is relatively high compared to traditional solvers. Indeed, for a mesh of size 301×301 there is time savings of about 7 for a Richardson number equal to 10 and about 4 for a Richardson number equal to 0.1. In the second test case, where the mass transfer is taken into account, PGD results are close to these of the literature. In this case, too, application of PGD allows one to save time. Indeed, PGD should be 10 times faster than the standard solver for a mesh of size 301×301 with a ratio N equal to -100.

This work is the first step in dealing with a 3-D situation, where the time coordinate can be included in the decomposition, or where a 3-D problem (with x, y, z space coordinates) can be solved. Indeed, in 2D, the full model has N^2 degrees of freedom (N is the number of nodes in each direction), while PGD consists of successive enrichments of 2 times N degrees of freedom. Moreover, in 3D, full description involves N^3 degrees of freedom. However, with PGD, successive enrichments involving 3 times N degrees of freedom are required. In such a situation, the benefit of the method goes beyond simple saving of CPU and provides a feasible solution for otherwise intractable problems.

REFERENCES

1. R. Nasrin, Influences of Physical Parameters on Mixed Convection in a Horizontallid-driven Cavity with an Undulating Base Surface, *Numer. Heat Transfer A*, vol. 61, pp. 306–321, 2012.
2. S. Sivasankaran and K. L. Pan, Numerical Simulation on Mixed Convection in a Porous Lid-drivencavity with Nonuniform Heating on Both Side Walls, *Numer. Heat Transfer A*, vol. 61, pp. 101–121, 2012.
3. M. M. Rahman, H. F. Öztop, N. A. Rahim, R. Saidur, and K. Al-Salem, MHD Mixed Convection with Joule Heating Effect in a Lid-Driven Cavity with a Heated Semi-circular Source using the Finite Element Technique, *Numer. Heat Transfer A*, vol. 60, pp. 543–560, 2011.
4. S. B. Paramane and A. Sharma, Consistent Implementation and Comparison of FOU, CD, SOU, and QUICK Convection Schemes on Square, Skew, Trapezoidal, and Triangular Lid-Driven Cavity Flow, *Numer. Heat Transfer B*, vol. 54, pp. 84–102, 2008.
5. V. C. Mariani and A. T. Prata, A Eulerian-lagrangian Method Applied to Fluid Flow in Lid-driven Cavities with Irregular Bottom Walls, *Numer. Heat Transfer B*, vol. 53, pp. 206–233, 2008.
6. H. F. Oztop and I. Dagtekin, Mixed Convection in Two-Sided Lid-Driven Differentially Heated Square Cavity, *Int. J. Heat Mass Transfer*, vol. 47, pp. 1761–1769, 2004.
7. D. Z. Noor, P. R. Kanna, and M.-J. Chern, Flow and Heat Transfer in a Driven Square Cavity with Double-Sided Oscillating Lids in Anti-phase, *Int. J. Heat Mass Transfer*, vol. 52, pp. 3009–3023, 2009.
8. E. Vishnuvardhanarao and M. K. Das, Laminar Mixed Convection in a Parallel Two-Sided Lid-Driven Differentially Heated Square Cavity Filled with A Fluid-Saturated Porous Medium, *Numer. Heat Transfer A*, vol. 53, pp. 88–110, 2007.
9. D. S. Kumar, K. Murugesan, and H. R. Thomas, Effect of the Aspect Ratio of a Heated Block on the Interaction between Inertial and Thermosolutal Buoyancy Forces in a Lid-Driven Cavity, *Numer. Heat Transfer A*, vol. 60, pp. 604–628, 2011.

10. M. Bhuvaneswari, S. Sivasankaran, and Y. J. Kim, Numerical Study on Double Diffusive Mixed Convection with a Soret Effect in a Two-Sided Lid-Driven Cavity, *Numer. Heat Transfer A*, vol. 59, pp. 543–560, 2011.

11. B. John, X.-J. Gu, and D. R. Emerson, Investigation of Heat and Mass Transfer in a Lid-Driven Cavity under Nonequilibrium Flow Conditions, *Numer. Heat Transfer B*, vol. 58, pp. 287–303, 2010.

12. D. Senthil Kumar, K. Murugesan, and H. R. Thomas, Numerical Simulation of Double Diffusive Mixed Convection in a Lid-Driven Square Cavity Using Velocity-vorticity Formulation, *Numer. Heat Transfer A*, vol. 54, pp. 837–865, 2008.

13. A. Al-Amiri, K. Khanafer, J. Bull, and I. Pop, Effect of Sinusoidal Wavy Bottom Surface on Mixed Convection Heat Transfer in a Lid-Driven Cavity, *Int. J. Heat Mass Transfer*, vol. 50, pp. 1771–1780, 2007.

14. C. Allery, C. Béghein, and A. Hamdouni, Applying Proper Orthogonal Decomposition to the Computation of Particle Dispersion in a Two-Dimensional Ventilated Cavity, *Commun. Nonlinear Sci. Numer. Simulation*, vol. 10, pp. 907–920, 2005.

15. C. Allery, C. Beghein, and A. Hamdouni, On Investigation of Particle Dispersion by a Pod Approach, *Int. Appl. Mech.*, vol. 44, pp. 133–142, 2008.

16. A. Sempey, C. Inard, C. Ghiaus, and C. Allery, Fast Simulation of Temperature Distribution in Air Conditioned Rooms by using Proper Orthogonal Decomposition, *Building and Environment*, vol. 44, pp. 280–289, 2009.

17. D. Alonso, A. Velazquez, and J. Vega, Robust Reduced Order Modeling of Heat Transfer in a Back Step Flow, *Int. J. Heat Mass Transfer*, vol. 52, pp. 1149–1157, 2009.

18. H. Gunes, Low-Order Dynamical Models of Thermal Convection in High-Aspect Ratio Enclosures, *Fluid Dynam. Res.*, vol. 30, pp. 1–30, 2002.

19. C. Leblond, C. Allery, and C. Inard, An Optimal Projection Method for the Reduced-order Modeling of Incompressible Flows, *Comput. Meth. Appl. Mech. Eng.*, vol. 200, pp. 2507–2527, 2011.

20. A. Ammar, F. Chinesta, P. Diez, and A. Huerta, An Error Estimator for Separated Representations of Highly Multidimensional Models, *Comput. Meth. Appl. Mech. Eng.*, vol. 199, pp. 1872–1880, 2010.

21. A. Ammar, F. Chinesta, and A. Falco, On the Convergence of a Greedy Rank-One Update Algorithm for a Class of Linear Systems, *Arch. Comput. Meth. Eng.*, vol. 17, pp. 473–486, 2010.

22. P. P. Ladevéze, New Approaches and Non-incremental Methods of Calculation, in *Nonlinear Computat. Struct. Mech.*, Springer-Verlag, Berlin.

23. P. Ladeveze, J.-C. Passieux, and D. Neron, The LATIN Multiscale Computational Method and the Proper Generalized Decomposition, *Comput. Meth. Appl. Mech. Eng.*, vol. 199, pp. 1287–1296, 2010.

24. P. Ladevéze and A. Nouy, On a Multiscale Computational Strategy with Time and Space Homogenization for Structural Mechanics, *Comput. Meth. Appl. Mech. Eng.*, pp. 192.

25. A. Nouy and P. Ladevéze, Multiscale Computational Strategy with Time and Space Homogenization: A Radial-Type Approximation Technique for Solving Micro Problems, *Int. J. Multiscale Comput. Eng.*, pp. 170.

26. H. Lamari, A. Ammar, P. Cartraud, G. Legrain, F. Chinesta, and F. Jacquemin, Routes for Efficient Computational Homogenization of Nonlinear Materials Using the Proper Generalized Decompositions, *Arch. Comput. Meth. Eng.*, vol. 17, pp. 373–391, 2010.

27. A. Nouy, A Generalized Spectral Decomposition Technique to Solve a Class of Linear Stochastic Partial Differential Equations, *Comput. Meth. Appl. Mech. Eng.*, vol. 196, pp. 4251–4537, 2007.

28. F. Chinesta, A. Ammar, and P. Joyot, The Nanometric and Micrometric Scales of the Structure and Mechanics of Materials Revisited: An Introduction to the Challenges of

Fully Deterministic Numerical Descriptions, *Int. J. Multiscale Comput. Eng.*, vol. 6, pp. 191–213, 2008.

29. A. Nouy and O. Le Maître, Generalized Spectral Decompositions for Stochastic Nonlinear Problems, *J. Comput. Phys.*, vol. 8, pp. 283–288, 2007.

30. A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, A New Family of Solvers for Some Classes of Multidimensional Partial Differential Equations Encountered in Kinetic Theory Modeling of Complex Fluids, *J. Non-Newtonian Fluid Mech.*, vol. 139, pp. 153–176, 2006.

31. A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, A New Family of Solvers for Some Classes of Multidimensional Partial Differential Equations Encountered in Kinetic Theory Modelling of Complex Fluids: Part II: Transient Simulation Using Space-time Separated Representations, *J. Non-Newtonian Fluid Mech.*, vol. 144, pp. 98–121, 2007.

32. A. Ammar, M. Normandin, F. Daim, D. Gonzalez, E. Cueto, and F. Chinesta, Non Incremental Strategies Based on Separated Representations: Applications in Computational Rheology, *Commun. Math. Sci.*, vol. 8, no. 3, Sp. Iss.SI, pp. 671–695, 2010.

33. F. Chinesta, A. Ammar, and P. Joyot, The Nanometric and Micrometric Scales of the Structure and Mechanics of Materials Revisited: An Introduction to the Challenges of Fully Deterministic Numerical Descriptions, *Int. J. Multiscale Comput. Eng.*, vol. 6/3, pp. 191–213, 2008.

34. A. Dumon, C. Allery, and A. Ammar, Proper General Decomposition (PGD) for the Resolution of Navier-Stokes Equations, *J. Comput. Phys.*, vol. 230, pp. 1387–1407, 2011.

35. A. Dumon, C. Allery, and A. Ammar, Proper Generalized Decomposition Method for Incompressible Flows in Stream-Vorticity Formulation, *EJCM—Comput. Meth. Coupled Fluid Struct. Probl.*, vol. 19, no. 5-6-7, pp. 591–617, 2010.

36. J. Weaver and R. Viskanta, Natural Convection Due to Horizontal Temperature and Concentration Gradients—1. Variable Thermophysical Property Effects, *Int. J. Heat Mass Transfer*, vol. 34, pp. 3107–3120, 1991.

37. J. Weaver and R. Viskanta, Natural Convection Due to Horizontal Temperature and Concentration Gradients—2. Species Interdiffusion, Soret and Dufour Effects, *Int. J. Heat Mass Transfer*, vol. 34, pp. 3121–3133, 1991.

38. J. Guermond, P. Minev, and J. Shen, An Overview of Projection Methods for incompressible Flows, *Comput. Meth. App. Mech. Eng.*, vol. 195 pp. 6011–6045, 2006.

39. C. Sun, B. Yu, H. F. Oztop, Y. Wang, and J. Wei, Control of Mixed Convection in Lid-Driven Enclosures Using Conductive Triangular Fins, *Int. J. Heat Mass Transfer*, vol. 54, pp. 894–909, 2011.

40. Q. Liang, P. H. Taylor, and A. G. Borthwick, Particle Mixing and Reactive Front Motion in Unsteady Open Shallow Flow—Modelled using Singular Value Decomposition, *Comput. Fluids*, vol. 36, pp. 248–258, 2007.

41. Z. Luo, X. Yang, and Y. Zhou, A Reduced Finite Difference Scheme Based on Singular Value Decomposition and Proper Orthogonal Decomposition for Burgers Equation, *J. Comput. Appl. Math.*, vol. 229, pp. 97–107, 2009.

42. A. M Al-Amiri, K. M. Khanafer, and I. Pop, Numerical Simulation of Combined Thermal and Mass Transport in a Square Lid-Driven Cavity, *Int. J. Thermal Sci.*, vol. 46, pp. 662–671, 2007.

APPENDIX: ALGEBRAIC FORMULATION OF PGD

A.1. Preliminaries

PGD is an iterative procedure which can be described as follows. At each iteration, the solution is enriched with an additional term,

$$\alpha_w^m F_w^m(x) G_w^m(y)$$

The F_w^m and G_w^m functions are obtained by solving a small-size nonlinear problem $(N_x + N_y)$, where N_x is the number of nodes in the direction x and N_y is the number of nodes in the direction y. Then, m α_w^i coefficients are determined by solving a linear system of size (m).

The PGD algorithm is summarized in three steps:

1. We assume we are at iteration m. The first step consists of computing the new F_w^m and G_w^m functions, and will be referred to as the "*enrichment*" step.
2. Once these functions have been calculated, the m coefficients α_w^i have to be updated. This is the "*projection*" step.
3. Finally, the convergence has to be checked. If the residual norm exceeds a given tolerance, enrichment is performed again, until convergence is achieved. This is the "*checking convergence*" step.

The problem (12) can be written in a discrete form:

$$\mathcal{L}_h(w_h) = \mathcal{G}_h \qquad (27)$$

with

$$\mathcal{L}_h = \sum_{j=1}^{n_{\mathcal{L}}} \mathbb{A}_x^j \otimes \mathbb{A}_y^j \quad \mathcal{G}_h = \sum_{j=1}^{n_{\mathcal{G}}} \mathbf{f}_x^j \otimes \mathbf{f}_y^j \quad w_h = \sum_{i=1}^{m} \alpha_w^i \mathbf{F}_w^i \otimes \mathbf{G}_w^i \qquad (28)$$

The operator $\mathcal{L}$ is discretized as a tensor product of operators $\mathbb{A}_x^j$ and $\mathbb{A}_y^j$ in the directions x and y, respectively The discretized operator $\mathbb{A}_x^j$ (resp. $\mathbb{A}_y^j$) is a square matrix whose size is N_x (resp. N_y). The second term is decomposed as products of the sum of vectors $\mathbf{f}_x^j$ and $\mathbf{f}_y^j$ of size N_x and N_y. Finally, the unknown w_h is calculated as a product sum of vectors $\mathbf{F}_w^i$ and $\mathbf{G}_w^i$ of size N_x and N_y using a weight coefficient α_w^i. $n_{\mathcal{L}}$ (resp. $n_{\mathcal{G}}$) represents the number of tensor products required to represent the separated form of the initial operator $\mathcal{L}$ (resp. the second member $\mathcal{G}$).

Taking into account the property of the tensor product, Eq. (27) can be written as:

$$\sum_{k=1}^{n_{\mathcal{L}}} \sum_{i=1}^{m} \alpha_w^i \left(\mathbb{A}_x^k \mathbf{F}_w^i \otimes \mathbb{A}_y^k \mathbf{G}_w^i \right) = \sum_{j=1}^{n_{\mathcal{G}}} \mathbf{f}_x^j \otimes \mathbf{f}_y^j \qquad (29)$$

The three steps of enrichment, projection, and checking convergence will now be described with these notations.

A.2. The Enrichment Step

At this stage, the solution w_h^{m-1} is supposed known and can be written as

$$w_h^{m-1} = \sum_{i=1}^{m-1} \alpha_w^i \mathbf{F}_w^i \otimes \mathbf{G}_w^i \qquad (30)$$

The unknown w_h^m is searched as

$$w_h^m = w_h^{m-1} + \mathbf{R} \otimes \mathbf{S} \qquad (31)$$

where $\mathbf{R}$ and $\mathbf{S}$ are unknowns. Introducing this new approximation of the solution into Eq. (29), we have to solve

$$\sum_{k=1}^{n_{\mathcal{L}}}\left(A_x^k\mathbf{R}\otimes A_y^k\mathbf{S}\right) = \mathcal{G}_h - \sum_{k=1}^{n_{\mathcal{L}}}\sum_{i=1}^{m-1}\alpha_w^i\left(A_x^k\mathbf{F}_w^i\otimes A_y^k\mathbf{G}_w^i\right) \tag{32}$$

This nonlinear system is solved within a fixed-point strategy. Thus, in order to compute $\mathbf{R}$, we choose to fix $\mathbf{S}$ and we project Eq. (32) onto the vector $\mathbf{S}$. This gives the following problem, corresponding to Eq. (18) in continuous form:

$$\sum_{k=1}^{n_{\mathcal{L}}}\gamma_k^1 A_x^k\mathbf{R} = \sum_{j=1}^{n_{\mathcal{G}}}\gamma_j^2\mathbf{f}_x^j - \sum_{k=1}^{n_{\mathcal{L}}}\sum_{i=1}^{m-1}\gamma_{i,k}^3\alpha_w^i A_x^k\mathbf{F}_w^i \tag{33}$$

with

$$\gamma_k^1 = {}^t\mathbf{S}A_y^k\mathbf{S}\in\mathbb{R} \qquad \gamma_j^2 = {}^t\mathbf{S}\mathbf{f}_y^j\in\mathbb{R} \qquad \gamma_{i,k}^3 = {}^t\mathbf{S}A_y^k\mathbf{G}_w^i\in\mathbb{R} \tag{34}$$

Similarly, in order to compute $\mathbf{S}$, we set $\mathbf{R}$ at the value just computed in Eq. (33) and we project Eq. (32) onto the vector $\mathbf{R}$. This gives the following problem, corresponding to Eq. (17) in continuous form

$$\sum_{k=1}^{n_{\mathcal{L}}}\eta_k^1 A_y^k\mathbf{S} = \sum_{j=1}^{n_{\mathcal{G}}}\eta_j^2\mathbf{f}_y^j - \sum_{k=1}^{n_{\mathcal{L}}}\sum_{i=1}^{m-1}\eta_{i,k}^3\alpha_w^i A_y^k\mathbf{G}_w^i \tag{35}$$

with

$$\eta_k^1 = {}^t\mathbf{R}A_x^k\mathbf{R}\in\mathbb{R} \qquad \eta_j^2 = {}^t\mathbf{R}\mathbf{f}_x^j\in\mathbb{R} \qquad \eta_{i,k}^3 = {}^t\mathbf{R}A_x^k\mathbf{F}_w^i\in\mathbb{R} \tag{36}$$

Problems (33) and (35) are solved iteratively. The fixed-point procedure stops when the kth iteration satisfies:

$$\|(\mathbf{R}\otimes\mathbf{S})_k - (\mathbf{R}\otimes\mathbf{S})_{k-1}\| \leq \epsilon \tag{37}$$

where $\|\cdot\|$ is the L^2 norm and ϵ is a parameter chosen by the user. The new $\mathbf{F}_w^m$ and $\mathbf{G}_w^m$ are then given by the next normalization:

$$\mathbf{F}_w^m = \frac{\mathbf{R}}{\|\mathbf{R}\|} \quad \mathbf{G}^m = \frac{\mathbf{S}}{\|\mathbf{S}\|} \tag{38}$$

A.3. Projection Step

Assuming the $\mathbf{F}_w^i$ and $\mathbf{G}_w^i$ functions are known, α_w^i $(1\leq i\leq m)$ has to be computed. For this purpose, we project Eq. (32) onto $\mathbf{F}_w^j\mathbf{G}_w^j$. Thus the following linear problem, whose size is m, corresponding to Eq. (14) in continuous form, is obtained

$$\mathbb{H}\alpha_\mathbf{w} = \mathbf{J} \qquad \text{with}\,{}^t\alpha_\mathbf{w} = \{\alpha_\mathbf{w}^1,\cdots,\alpha_\mathbf{w}^m\} \tag{39}$$

where the components of $\mathbb{H}$ and $\mathbf{J}$ are defined by

$$\mathbb{H}_{ij} = \sum_{k=1}^{n_\mathcal{L}} {}^t\mathbf{F}_w^j \mathbb{A}_x^k \mathbf{F}_w^i \cdot {}^t\mathbf{G}_w^j \mathbb{A}_y^k \mathbf{G}_w^i \qquad \text{and} \qquad J_j = \sum_{k=1}^{n_\mathcal{G}} {}^t\mathbf{F}_w^j \mathbf{f}_x^k \cdot {}^t\mathbf{G}_w^j \mathbf{f}_y^k \tag{40}$$

A.4. Check Convergence

In order to estimate the convergence of the algorithm, a computation is performed of the residual Res of Eq. (12) defined by

$$\text{Res} = \sum_{k=1}^{n_\mathcal{L}} \sum_{i=1}^{m} \alpha_w^i \left(\left[\mathbb{A}_x^k \right] \mathbf{F}_w^i \otimes \left[\mathbb{A}_y^k \right] \mathbf{G}_w^i \right) - \sum_{k=1}^{n_\mathcal{G}} \mathbf{f}_x^k \otimes \mathbf{f}_y^k \tag{41}$$

When the L^2 norm of this residual becomes lower than a coefficient ϵ set by the user, the algorithm is considered to be at convergence, and the solution of the problem is expressed ass

$$w_h = \sum_{i=1}^{m} \alpha_w^i \mathbf{F}_w^i \otimes \mathbf{G}_w^i \tag{42}$$