Time implicit high-order discontinuous galerkin method with reduced evaluation cost

Florent Renac, Claude Marmignon, Frédéric Coquel

TIME IMPLICIT HIGH-ORDER DISCONTINUOUS GALERKIN METHOD WITH REDUCED EVALUATION COST

FLORENT RENAC[*], CLAUDE MARMIGNON[†], AND FRÉDÉRIC COQUEL[‡]

Abstract. An efficient and robust time integration procedure for a high-order discontinuous Galerkin method is introduced for solving nonlinear second-order partial differential equations. The time discretization is based on an explicit formulation for the hyperbolic term and an implicit formulation for the parabolic term. The procedure uses an iterative algorithm with reduced evaluation cost. The size of the linear system to be solved is greatly reduced thanks to partial uncoupling in space between low-order and high-order degrees of freedom. Numerical examples are presented for the nonlinear convection-diffusion equation in one and two dimensions including steady and unsteady flow problems. The performance of the present method is investigated in terms of CPU time and compared to a fully implicit method. A Von Neumann stability analysis is carried out in order to determine the stability and damping properties of the method. Besides a fairly reduced CPU effort, numerical results demonstrate better convergence properties of the present algorithm when compared to the fully implicit method.

Key words. discontinuous Galerkin method, nonlinear convection-diffusion equation, implicit-explicit time discretization, pseudo-time integration

AMS subject classifications. 65N30, 65N12

1. Introduction. Discontinuous Galerkin (DG) methods are high-order finite element discretizations and were introduced in the early 1970s for the numerical simulation of the first-order hyperbolic neutron transport equation [34, 40]. The method was later extended to nonlinear convection dominated flow problems with the use of a Runge-Kutta method for the time integration [12, 14]. In recent years, there has been a strong interest for these techniques in the field of computational fluid dynamics which has led to the introduction of discretization schemes for parabolic and purely elliptic equations. See for example [4, 5, 8, 9, 13, 19, 20, 24, 28, 37, 43] and references cited therein. For more details, the reader is referred to the analysis of existing discretizations in an unified framework developed by Arnold *et al.* [3] and to the overview of recent progress in DG methods for compressible flows [32]. The success of these methods lies in their high-order of accuracy and flexibility thanks to their high degree of locality. The stencil of most DG methods is compact and independent of the space discretization order. This means that the evaluation of the residual in a discretization element involves only this element and its immediate neighbours. The order of the numerical scheme depends on the degree of the approximated piecewise polynomials which can be easily increased. These properties make the DG method well suited to algorithm parallelization, *hp*-refinement, *hp*-multigrid, unstructured meshes, the application of boundary conditions, etc.

However, the association of DG methods to an explicit time discretization leads to a strong restriction on the time step due to the so-called Courant-Friedrichs-Levy (CFL) condition for stability of the numerical scheme [2, 10]. Numerical experiments indicate that this condition becomes more restrictive as the space discretization order increases [14]. Moreover, the discretization of parabolic terms requires an additional stability constraint, the Von Neumann condition, which becomes more restrictive than

---

[*]CFD and Aeroacoustics Department DSNA, ONERA, 92320 Chatillon Cedex, France (`florent.renac@onera.fr`).

[†]CFD and Aeroacoustics Department DSNA, ONERA, 92320 Chatillon Cedex, France.

[‡]CMAP, Ecole Polytechnique, 91120 Palaiseau, France.

the CFL condition for flow regions with small cell Reynolds number as in boundary layers. In high Reynolds number flows, the large variations in element size required to resolve all the spatial scales make the use of explicit time integration techniques prohibitive.

Implicit solvers are thus needed with this type of flows. However, the shortcoming of these methods is the extremely high computational cost induced by the large number of degrees of freedom (DOF) per element. The application of an implicit time discretization technique to a nonlinear system of equations leads to an algebraic system to be solved at each time step. The size of this system is equal to the total number of unknowns of the discrete problem and may become very large for practical applications (e.g. the resolution of RANS equations coupled with turbulent transport equations on three-dimensional grids). Being able to solve these large systems at a reasonable computational cost is therefore essential for the efficiency of the DG methods and adapted solutions have been proposed.

For large size problems, the use of direct solution techniques remains too expensive and iterative methods must be employed. Several authors have applied Newton GMRES and implicit Runge-Kutta methods to the resolution of compressible Euler, Navier-Stokes and RANS equations in the context of DG discretizations [6, 7, 22, 30, 39]. Various preconditioned iterative algorithms have been introduced showing the potential benefits of taking advantage of either the block structure of the Jacobian matrix or a factorization using reordering [18, 25, 38, 39]. In [29] a block diagonal preconditioner for linear problems based upon a domain decomposition method with static condensation is seen to achieve an optimal convergence where the convergence rate of the implicit iterative solver is independent of the number of DOF. Static condensation was also successfully applied to a direct solver in [42]. Multigrid approaches where convergence acceleration is achieved through the use of coarse levels constructed by reducing the polynomial degree (p-multigrid) or using coarser grids with fewer elements (h-multigrid) have also been proposed [18, 25, 36]. Yasue *et al.* [44] used a pointwise relaxation implicit scheme associated to the resolution of a simplified implicit system. The results showed a reduction of the number of iterations and CPU time needed for convergence of the algorithm for compressible Euler and Navier-Stokes equations in three-dimensions. The method was however limited to second-order space discretization schemes. Dolejsì *et al.* developed a semi-implicit method where the linear terms in the discrete equations are treated implicitly and the nonlinear terms are treated with an explicit extrapolation [21, 22]. We note that all these techniques consider the resolution of a large size nonlinear problem for the total number of unknowns.

The purpose of this article is to introduce a robust and fast time integration method adapted to an implicit formulation of a DG discretization of nonlinear viscous terms of partial differential equations (PDE) while convective terms are treated explicitly. We use the BR2 scheme from Bassi and Rebay [5] which guaranties an optimal order of convergence of the L_2-norm of the error between the numerical approximation and the underlying analytical solution [3]. The stencil of this method is also compact which allows the use of a sparse implicit matrix and hence reduces memory requirement and operation count. We use an admissible change of variable which follows from the classical notion of entropy solutions in the context of first-order and second-order PDE. Entropy solutions have been found to play a fundamental role in the mathematical analysis of PDE [1, 35], the analysis of complex physical systems [11, 17], and the numerical approximation of PDE [27]. We also refer to the text-books

by Dafermos [15] or Godlewski and Raviart [26]. For the resolution of the discrete problem at each time step, we propose to use a simplified implicit system based on a reduced communication between elements which share a common interface. This assumption follows from the natural hierarchy of the basis of the discrete function space. The numerical solution is the sum of modes associated to different polynomials spanning the basis. The communication between adjacent elements is assumed to be dominated by the coupling between DOF associated to the lowest order modes only. This condition is essential to simplify the implicit problem and reduce its size. After solving the implicit problem for the low-order DOF, a local reconstruction is used for evaluating the higher order DOF. By local, we mean a reconstruction based on the stencil of the DG discretization. This property is essential to keep the locality of the DG method. The simplified algorithm is here introduced for the resolution of steady problems with a time marching technique and using an iterative solver. An application to an unsteady flow problem with a pseudo-time stepping method is also presented.

The paper is organized as follows. Section 2 presents the numerical approach. After introducing the nonlinear convection-diffusion equation and entropy solution in § 2.1, the DG discretization is formulated in § 2.2. Then, the time implicit discretization and its simplification are introduced in § 2.3 and 2.4, and the computational cost of the methods is analyzed in § 2.5. Section 3 is devoted to a Von Neumann analysis of the numerical scheme. The overall performance of the method is assessed by several numerical experiments in § 4. Results are obtained for a one-dimensional (1D) problem (section 4.1) and two-dimensional (2D) steady and unsteady flow problems (sections 4.2 and 4.3). Finally, the conclusions of this research are summarized in § 5.

2. Model problem and discretization.

2.1. Nonlinear convection-diffusion equation and entropy solution. The discussion in this paper focuses on the discretization of nonlinear scalar convection-diffusion equation with a DG method. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain where d is the space dimension and consider the following problem

$$
\begin{align}
\nabla \cdot (\mathbf{c}(\mathbf{x})u) - \nabla \cdot (\mathbf{B}(\mathbf{x}, u)\nabla u) &= s(\mathbf{x}) \quad \text{in } \Omega, \tag{2.1a} \\
u &= u_D \quad \text{on } \Gamma_D, \tag{2.1b} \\
\nabla u \cdot \mathbf{n} &= g_N \quad \text{on } \Gamma_N, \tag{2.1c}
\end{align}
$$

which describes the conservation of the scalar quantity $u(\mathbf{x})$ convected in a velocity field $\mathbf{c}(\mathbf{x}) \in \mathbb{R}^d$ and subject to diffusion. The nonlinear diffusivity tensor $\mathbf{B}(\mathbf{x}, u)$ is here assumed to be strictly positive definite and we will specify its special form later on. The forcing function $s(\mathbf{x}) \in L^2(\Omega)$ represents production of the quantity u. The boundary of the domain $\partial \Omega$ is partitioned into nonoverlapping boundaries as $\partial \Omega = \Gamma_D \cup \Gamma_N$ where Γ_D and Γ_N denote Dirichlet and Neumann boundaries and we assume that $u_D \in L^2(\Gamma_D)$ and $g_N \in L^2(\Gamma_N)$. By $\mathbf{n}$ we denote the unit outward normal vector to $\partial \Omega$.

The aim of the present work is to derive a fast time marching method for the numerical resolution of problem (2.1) with a DG space discretization. The quantity $u = u(\mathbf{x}, t)$ is now assumed to be a function of space and time and a pseudo-time derivative is added to equation (2.1a) thus leading to

$$
u_t + \nabla \cdot (\mathbf{c}(\mathbf{x})u) - \nabla \cdot (\mathbf{B}(\mathbf{x}, u)\nabla u) = s(\mathbf{x}) \quad \text{in } \Omega \times (0, \infty) \tag{2.2}
$$

with boundary conditions (2.1b,c) and the initial condition $u(\mathbf{x}, 0) = u_0(\mathbf{x})$ in Ω, where $u_0 \in H^1(\Omega)$. We consider entropy solutions of this problem and assume that the diffusivity tensor is of the form $\mathbf{B}(\mathbf{x}, u) = \mathcal{U}_{uu}(u)\mathbf{C}(\mathbf{x})$ where for any given $\mathbf{x} \in \mathbb{R}^d$ $\mathbf{C}(\mathbf{x}) \in \mathbb{R}^{d \times d}$ is a positive definite matrix and $\mathcal{U}(u)$ is a strictly convex scalar function. The function $\mathcal{U}$ is said to be an entropy function for the problem (2.2) [27]. As pointed out in the introduction, entropy solutions are motivated by physical problems where conservative variables are consistent with entropy variables (see references cited in the introduction). Introducing the change of variable $v = \mathcal{U}_u(u)$ such that $\mathcal{U}_{uu} u_v = 1$ leads to the following problem for v:

$$u(v)_t + \nabla \cdot (\mathbf{c}(\mathbf{x})u(v)) - \nabla \cdot (\mathbf{C}(\mathbf{x})\nabla v) = s(\mathbf{x}) \quad \text{in } \Omega \times (0, \infty) \qquad (2.3)$$

with boundary and initial conditions

$$
\begin{aligned}
v &= \mathcal{U}_u(u_D) & \text{on } \Gamma_D, & \qquad (2.4\text{a}) \\
u_v(v)\nabla v \cdot \mathbf{n} &= g_N & \text{on } \Gamma_N, & \qquad (2.4\text{b}) \\
v(\mathbf{x}, 0) &= \mathcal{U}_u(u_0(\mathbf{x})) & \text{in } \Omega. & \qquad (2.4\text{c})
\end{aligned}
$$

The nonlinearity in the second-order derivative of the diffusion term has been shifted to the first-order derivatives in time and space.

2.2. Discontinuous Galerkin formulation. The DG method consists in defining a discretized weak formulation of the problem (2.3) subject to initial and boundary conditions (2.4). Using classical notations (see for instance [3]), the domain Ω is partitioned into a shape-regular mesh $\Omega_h = \bigcup_{j=1}^N \Omega_j$ consisting of nonoverlapping and nonempty elements Ω_j of characteristic size $h := \min\{\text{diam}\,\Omega_j, 1 \leq j \leq N\}$ where $\text{diam}\,\Omega_j$ is the diameter of biggest ball included in Ω_j. We define the set of interior and boundary faces as

$$\mathcal{E}_i = \{e \in \mathcal{E}_h : e \cap \partial\Omega_h = \emptyset\}, \quad \mathcal{E}_b = \{e \in \mathcal{E}_h : e \in \partial\Omega_h\},$$

where $\mathcal{E}_h = \mathcal{E}_i \cup \mathcal{E}_b$ denotes the union of all open $(d-1)$-dimensional faces in Ω_h. The set of boundary faces may be further divided into $\mathcal{E}_b = \mathcal{E}_D \cup \mathcal{E}_N$ consisting of faces of Γ_D and Γ_N, respectively.

We consider the broken space $L^2(\Omega_h) = \{\varphi \in L^2(\Omega) : \varphi|_{\Omega_j} \in L^2(\Omega_j), 1 \leq j \leq N\}$ and look for approximate solutions in the function space of discontinuous polynomials

$$\mathcal{V}_h^p = \{\varphi \in L^2(\Omega_h) : \varphi|_{\Omega_j} \circ F_j \in \mathcal{Q}_p(\kappa), 1 \leq j \leq N\}, \qquad (2.5)$$

where $\mathcal{Q}_p(\kappa)$ denotes the space of functions formed by tensor products of polynomials of degree at most p on the master element $\kappa = I^d$ where $I = [-1, 1]$: $\mathcal{Q}_p(\kappa) = \bigotimes_{i=0}^d \mathcal{P}_p(I)$. Each physical element Ω_j is the image of κ through the mapping F_j. The numerical solution of equation (2.3) can be written as

$$v_h(\mathbf{x}, t) = \sum_{l=1}^{N_p} \phi^l(\mathbf{x}) V_j^l(t) \quad \forall \mathbf{x} \in \Omega_j, 1 \leq j \leq N, \forall t \geq 0, \qquad (2.6)$$

where V_j^l are the DOF in the element Ω_j. The subset $(\phi^1, \ldots, \phi^{N_p})$ constitutes a hierarchical and orthogonal modal basis of $\mathcal{V}_h^p$ and $N_p = \Pi_{i=1}^d (p+i)/i$ is its dimension. In this work, we will use Legendre polynomials as basis in one dimension and the Dubiner basis [23] in two dimensions.

The face-based formulation of the BR2 discretization [5] of equation (2.3) reads: find v_h in $\mathcal{V}_h^p$ such that for all ϕ in $\mathcal{V}_h^p$ we have

$$\int_{\Omega_h} \phi u(v_h)_t d\mathbf{x} + \mathcal{L}_c(v_h, \phi) + \mathcal{L}_v(v_h, \phi) + \mathcal{L}_s(\phi) = 0 \qquad (2.7)$$

with the initial condition

$$\int_{\Omega_h} \phi v_h(\mathbf{x}, 0) d\mathbf{x} = \int_{\Omega_h} \phi \mathcal{U}_u(u_0(\mathbf{x})) d\mathbf{x}.$$

Hereafter, we will use the following operators $\{\phi\} = (\phi^+ + \phi^-)/2$ and $[\![\phi]\!] = \phi^+ - \phi^-$ which denote the average and jump operators. Here, ϕ^+ and ϕ^- are the traces of ϕ on an interface e taken from within the interior of the element Ω_j^+ and the interior of the neighbouring element Ω_j^-, respectively (see Figure 2.1).

The discretization of the hyperbolic term in (2.7) is nonlinear in its first argument and reads

$$\mathcal{L}_c(v_h, \phi) = -\int_{\Omega_h} u(v_h)\mathbf{c} \cdot \nabla\phi d\mathbf{x} + \sum_{e \in \mathcal{E}_i} \int_e [\![\phi]\!]\hat{f}_c dS + \int_{\mathcal{E}_N} \phi u(v_h^+)\mathbf{c} \cdot \mathbf{n} dS, \quad (2.8)$$

where $\mathbf{n}$ denotes the unit outward normal vector to an element Ω_j. We use a standard upwind discretization flux as numerical convective flux:

$$\hat{f}_c = \{u(v_h)\mathbf{c}\} \cdot \mathbf{n} + \frac{\alpha}{2}[\![v_h]\!],$$

where the numerical constant α is the biggest eigenvalue of the convective flux Jacobian: $\alpha = \max\{|u_v(v)\mathbf{c} \cdot \mathbf{n}| : v = v_h^{\pm}\}$.

The bilinear and linear forms in (2.7) are defined by

$$\mathcal{L}_v(v_h, \phi) = \int_{\Omega_h} \mathbf{C}(\nabla_h v_h + \mathbf{R_h}) \cdot \nabla\phi d\mathbf{x} - \sum_{e \in \mathcal{E}_i} \int_e [\![\phi]\!]\{\mathbf{C}(\nabla_h v_h + \mathbf{r_h^e})\} \cdot \mathbf{n} dS$$

$$- \int_{\mathcal{E}_D} \phi\mathbf{C}(\nabla_h v_h^+ + \mathbf{r_h^{e+}}) \cdot \mathbf{n} dS - \int_{\mathcal{E}_N} \phi(g_N/u_v(v_h))\mathbf{C} dS, \qquad (2.9)$$

and

$$\mathcal{L}_s(\phi) = -\int_{\Omega_h} \phi s(\mathbf{x})d\mathbf{x} + \int_{\mathcal{E}_D} \phi u_D \mathbf{c} \cdot \mathbf{n}dS. \tag{2.10}$$

The so-called global lifting operator $\mathbf{R_h}$ is defined as the sum of local lifting operators $\mathbf{r_h^e}$: $\mathbf{R_h} = \sum_{e \in \mathcal{E}_i \cup \mathcal{E}_b} \mathbf{r_h^e}$, where $\mathbf{r_h^e}$ has support on elements adjacent to e and is solution of the following problem: for all ϕ in $\mathcal{V}_h^p$ we have

$$\int_{\Omega_j^+ \cup \Omega_j^-} \phi \mathbf{r_h^e} d\mathbf{x} = \begin{cases} -\int_e \{\phi\} [\![v_h]\!] \mathbf{n}dS & \text{if } e \in \mathcal{E}_i, \\ -\int_e \{\phi\}(v_h^+ - \mathcal{U}_u(u_D))\mathbf{n}dS & \text{if } e \in \mathcal{E}_D, \\ 0 & \text{if } e \in \mathcal{E}_N. \end{cases}$$

2.3. Time discretization. The semi-discrete equation (2.7) is advanced in time by means of an explicit treatment for the hyperbolic term (2.8) and an implicit treatment for the parabolic term (2.9). A backward Euler scheme is used for the time implicit integration. The temporal derivative in (2.7) may be approximated by

$$\int_{\Omega_h} \phi^k u(v_h)_t d\mathbf{x} \simeq \int_{\Omega_h} \phi^k u_{v_h}^{(n)} \frac{v_h^{(n+1)} - v_h^{(n)}}{\Delta t} d\mathbf{x},$$

where by $u_{v_h}^{(n)} > 0$ we denote either $u_v(v_h^{(n)})$ with $v_h^{(n)} = v_h(\mathbf{x}, n\Delta t)$ and Δt the time step or an approximation of it, say for instance $u_{v_h}^{(n)} = u_v(v_h(\mathbf{x}_j, n\Delta t))$ where $\mathbf{x}_j$ is the centre of the element Ω_j. The discrete problem for (2.3) reduces to the following linear system at each time step

$$\mathbf{A}(\mathbf{V}^{(n+1)} - \mathbf{V}^{(n)}) = -\mathbf{R}(\mathbf{V}^{(n)}) \tag{2.11}$$

with $\mathbf{V}^{(n)}$ the vector of DOF, $\mathbf{R}(\mathbf{V}^{(n)}) = \mathbf{L_c}(\mathbf{V}^{(n)}) + \mathbf{L_v}\mathbf{V}^{(n)} + \mathbf{L_s}$ the residual vector defined by the sum of discrete forms (2.8) to (2.10). The so-called implicit matrix $\mathbf{A}$ is defined by

$$\mathbf{A} = \frac{1}{\Delta t}\mathbf{M}^{(n)} + \mathbf{L_v}, \tag{2.12}$$

where $\mathbf{M}^{(n)} = \text{diag}(\mathbf{M_1}^{(n)}, \dots, \mathbf{M_N}^{(n)})$ denotes the mass matrix whose entries for each block are

$$(\mathbf{M_j}^{(n)})_{kl} = \int_{\Omega_j} \phi^k \phi^l u_v(v_h^{(n)})d\mathbf{x}, \quad 1 \le j \le N, \quad 1 \le k,l \le N_p. \tag{2.13}$$

Here, the matrix $\mathbf{M}^{(n)}$ is either diagonal if $u_{v_h}^{(n)} = u_v(v_h(\mathbf{x}_j, t^{(n)}))$ or block diagonal if $u_{v_h}^{(n)} = u_v(v_h^{(n)})$.
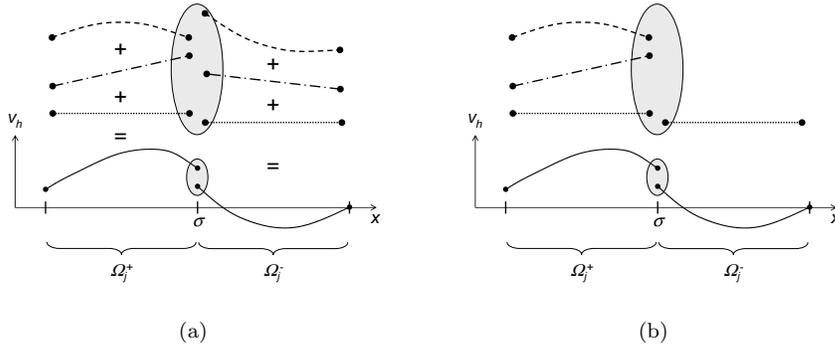
(a)   (b)

FIG. 2.2. *1D example of the implicit procedure simplification: in (a) the elementwise numerical solution $v_h \in \mathcal{V}_h^2$ (continuous lines) is the sum of a constant mode (dotted lines), a linear mode (dash-dotted lines) and a quadratic mode (dashed lines). The coupling between modes in elements adjacent to a common interface e is illustrated by the grey zones; in (b) the coupling between every mode in Ω_j^+ and linear and quadratic modes in Ω_j^- is neglected.*

2.4. Simplification of the implicit procedure. The matrix $\mathbf{A}$ is an unsymmetric real matrix of size $N_{DOF} = N \times N_p$ since $\mathbf{L_v}$ is and we use a preconditioned biconjugate gradient (BCG) method for solving the linear system (2.11) at each time step. The inversion cost of $\mathbf{A}$ may become prohibitive in practical applications where N is large and where accuracy is looked for (see § 2.5).

We aim at simplifying the linear system resolution by modifying the implicit problem (2.11). This is done by reducing the coupling between modes in adjacent elements. For the sake of clarity, the method is now introduced for a 1D example on the segment $\Omega_h = \cup_{j=1}^N [(j-1)h, jh]$ where $h = 1/N$. The application of this method to multidimensional problems is straightforward. The numerical solution (2.6) is a sum of modes as illustrated in Figure 2.2(a) for a numerical solution v_h in $\mathcal{V}_h^2$. The lowest order mode is the constant mode $\phi^0(\mathbf{x})V_j^0(t)$ where V_j^0 corresponds to the mean value of the numerical solution in Ω_j^+. Higher order modes correspond to the linear mode $\phi^1(\mathbf{x})V_j^1(t)$ and quadratic mode $\phi^2(\mathbf{x})V_j^2(t)$.

Every mode in Ω_j^+ is linked to all modes in the adjacent element Ω_j^- through the numerical flux and the lifting operators. Here, we assume that the coupling between low-order modes dominates and we choose to neglect coupling with higher order modes in neighbouring elements. For that purpose, let $0 \leq p_s \leq p$ be an integer that separates low-order modes $0 \leq q \leq p_s$ and high-order modes $p_s < q \leq p$. We neglect the coupling between modes in Ω_j^+ and high-order modes in Ω_j^-. This is illustrated in Figure 2.2(b) for $p_s = 0$.

The implicit matrix (2.12) has a tridiagonal block structure that may be written in the compact form as $\mathbf{A} = (\mathbf{A}_{j-1}|\mathbf{A}_j|\mathbf{A}_{j+1})$, where $\mathbf{A}_j$ and $\mathbf{A}_{j\pm1}$ are diagonal and off-diagonal blocks of size $N_p \times N_p$. Each block may be further decomposed into

$$\mathbf{A}_i = \begin{pmatrix} \tilde{\mathbf{A}}_i & \mathbf{B}_i \\ \mathbf{C}_i & \hat{\mathbf{A}}_i \end{pmatrix}, \quad j-1 \leq i \leq j+1,$$

where $\tilde{\mathbf{A}}_i \in \mathbb{R}^{N_{p_s} \times N_{p_s}}$, $\hat{\mathbf{A}}_i \in \mathbb{R}^{(N_p - N_{p_s}) \times (N_p - N_{p_s})}$ and $N_{p_s} = \Pi_{i=1}^d (p_s + i)/i$ denotes the number of low-order modes. Uncoupling of high-order modes $p_s < q \leq p$ in

neighbouring elements amounts to replace the implicit matrix $\mathbf{A}$ by the following matrix

$$\mathbf{A}^\star = \left(\begin{array}{cc|cc|cc} \tilde{\mathbf{A}}_{j-1} & 0 & \tilde{\mathbf{A}}_j & \mathbf{B}_j & \tilde{\mathbf{A}}_{j+1} & 0 \\ \mathbf{C}_{j-1} & 0 & \mathbf{C}_j & \hat{\mathbf{A}}_j & \mathbf{C}_{j+1} & 0 \end{array} \right).$$

The linear system (2.11) at a physical time step is then replaced by a two-stage algorithm. First, an implicit problem for the low-order modes is solved by expliciting terms accounting for higher order modes:

$$\tilde{\mathbf{A}} \Delta \tilde{\mathbf{V}}^{(n+1)} = -\tilde{\mathbf{R}}(\mathbf{V}^{(n)}) - \mathbf{B} \Delta \hat{\mathbf{V}}^{(n)}, \tag{2.14}$$

where $\tilde{\mathbf{A}} = (\tilde{\mathbf{A}}_{j-1}|\tilde{\mathbf{A}}_j|\tilde{\mathbf{A}}_{j+1})$ is a square matrix of size $N \times N_{p_s}$, $\Delta \tilde{\mathbf{V}}^{(n+1)} = \tilde{\mathbf{V}}^{(n+1)} - \tilde{\mathbf{V}}^{(n)}$ and $\Delta \hat{\mathbf{V}}^{(n)} = \hat{\mathbf{V}}^{(n)} - \hat{\mathbf{V}}^{(n-1)}$ where $\tilde{\mathbf{V}}$ and $\hat{\mathbf{V}}$ contain low-order and high-order modes, respectively. By $\tilde{\mathbf{R}}(\mathbf{V}^{(n)})$ we denote the residuals associated to low-order modes and we set $\mathbf{B} = \mathrm{diag}(\mathbf{B}_1, \ldots, \mathbf{B}_N)$.

Second, higher order modes are reconstructed from the low-order modes and precedent time step solution. We note $\hat{\mathbf{A}} = (\hat{\mathbf{A}}_{j-1}|\hat{\mathbf{A}}_j|\hat{\mathbf{A}}_{j+1})$ the square matrix of size $N \times (N_p - N_{p_s})$ and let $\hat{\mathbf{A}} = \hat{\mathbf{L}} + \hat{\mathbf{D}} + \hat{\mathbf{U}}$ be its decomposition into strictly lower, diagonal and strictly upper parts, respectively. The reconstruction of higher modes reads as follows

$$(\hat{\mathbf{L}} + \hat{\mathbf{D}}) \Delta \hat{\mathbf{V}}^{(n+1)} = -\hat{\mathbf{R}}(\mathbf{V}^{(n)}) - \mathbf{C} \Delta \tilde{\mathbf{V}}^{(n+1)} - \hat{\mathbf{U}} \Delta \hat{\mathbf{V}}^{(n)}, \tag{2.15}$$

where $\hat{\mathbf{R}}(\mathbf{V}^{(n)})$ corresponds to the vector of residuals associated to high-order modes and $\mathbf{C} = \mathrm{diag}(\mathbf{C}_1, \ldots, \mathbf{C}_N)$. Hence, the linear system to be solved reads

$$\boldsymbol{\mathcal{A}} \Delta \mathbf{V}^{(n+1)} = -\mathbf{R}(\mathbf{V}^{(n)}) - \boldsymbol{\mathcal{B}} \Delta \mathbf{V}^{(n)}, \tag{2.16}$$

where the matrices $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{N_{DOF} \times N_{DOF}}$ are defined by

$$\boldsymbol{\mathcal{A}} = \begin{pmatrix} \tilde{\mathbf{A}} & 0 \\ \mathbf{C} & \hat{\mathbf{L}} + \hat{\mathbf{D}} \end{pmatrix}, \quad \boldsymbol{\mathcal{B}} = \begin{pmatrix} 0 & \mathbf{B} \\ 0 & \hat{\mathbf{U}} \end{pmatrix}. \tag{2.17}$$

Note that $\boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{B}} = \mathbf{A}$, one therefore recovers the full implicit problem (2.11) when $\Delta \mathbf{V}^{(n+1)} = \Delta \mathbf{V}^{(n)}$. This guaranties the same accuracy of both numerical schemes at convergence.

In the following, the full implicit method (2.11) will be referred to as the FULL method, and the simplified implicit method (2.16) will be referred to as the SIMPp_s method where p_s denotes the integer that separates the low- and high-order modes. We now compare the properties and performances of both methods in the next sections.

2.5. Computational cost. To quantify the relative cost in floating point operations (flops) of both methods, we evaluate the asymptotic operation count of the matrix inversion at each time step with both algorithms. We use a BCG method with diagonal preconditioning. The matrix inversion depends on the problem dimension and the representation of the sparse matrix. Here, we use a modified sparse row format for the 1D computations and a dense block format for the 2D computations [41]. The number of flops for basic sparse linear algebra operations are

Asymptotic reduction factor $(N_p/N_{p_s})^2$ in matrix inversion cost for large N by using the SIMP method.

	p	1	2	2	3	3	3	4	4	4	4
	p_s	0	0	1	0	1	2	0	1	2	3
$(N_p/N_{p_s})^2$	1D	4	9	2.25	16	4	1.78	25	6.25	2.78	1.56
	2D	9	36	4	100	11.1	2.78	225	25	6.25	2.25

- $\mathcal{O}(\beta N_p^2 N)$ for a matrix-vector product with $\beta = 6$ in 1D and $\beta = 14$ in 2D,
- $\mathcal{O}(N_p N)$ for a diagonal preconditioning,
- $\mathcal{O}(2\beta N_p^2 N)$ for a diagonal preconditioned BCG step,
- $\mathcal{O}(N(N_p - N_{p_s})^2)$ for a triangular back-solve for the resolution of equation (2.15).

The BCG method would converge in $N_p N$ steps in exact arithmetic. In our experiments, we observed that the method required about $\alpha N_p N$ steps with $\alpha \leq 1$ to reach the required convergence level. The factor α is here assumed to be constant for the sake of simplicity.

The FULL method requires the inversion of the full matrix of size $N \times N_p$ which requires $\mathcal{O}(2\alpha\beta N_p^2 N^2)$ flops. The first stage of the SIMP method consists in the inversion of a matrix of size $N \times N_{p_s}$ which hence requires $\mathcal{O}(2\alpha\beta N_{p_s}^2 N^2)$ flops. The second stage is the local reconstruction of the high-order modes. This procedure is elementwise and uses operations on N_p-by-N_p matrices and N_p-by-1 vectors. The leading term of operation counts is $\mathcal{O}(2N N_p(N_p - N_{p_s}))$ and comes from the construction of the right-hand side of equation (2.15) and its resolution.

We conclude that the computational cost of the SIMP method is less expensive than the FULL method by a factor $(N_p/N_{p_s})^2$ for large N. This means that the saving in flops becomes significant when the SIMP method is based on a uncoupling with a reduced number of low-order modes, i.e. when p_s is quite lower than p. Moreover, we note that the relative part of high-order modes increases as the space dimension d increases for large p and p_s: $N_p/N_{p_s} \simeq (p/p_s)^d$ thus meaning that the separation between low- and high-order modes may be increased for multidimensional problems for a same flop saving (see Table 2.1).

3. Von Neumann analysis.

3.1. Amplification matrix. A Von Neumann analysis is here performed in order to derive practical stability constraints and to analyze dispersion and dissipation errors of the numerical schemes. For the sake of simplicity, we consider the 1D linear convection-diffusion equation

$$v_t + cv_x - \nu v_{xx} = 0, \quad \text{in} (0,1] \times (0,+\infty) \tag{3.1}$$

with $\nu > 0$, $c > 0$, and periodic boundary condition $v(0,t) = v(1,t)$, for all $t \geq 0$. Equation (3.1) admits Fourier modes of the form $v(x,t) = e^{i(kx-\omega t)}$ as exact solutions where k is the prescribed real wave number and ω is the corresponding complex frequency satisfying the dispersion relation $\omega = kc - i\nu k^2$ with $i^2 = -1$. The exact amplification factor reads

$$\tilde{G} = \frac{v(x,t+\Delta t)}{v(x,t)} = |\tilde{G}|e^{-i\tilde{\Phi}}, \tag{3.2}$$

where $|\tilde{G}|$ and $\tilde{\Phi}$ denote the amplitude and phase of $\tilde{G}$, respectively.

A similar decomposition is performed for the numerical solution v_h. The domain is first divided into regular elements of size $h = 1/N$. The exact amplification factor (3.2) may be rewritten as $\tilde{G} = e^{-D\theta^2} e^{-i\sigma\theta}$ where $\sigma = c\Delta t/h$ is the Courant Number, $D = \nu\Delta t/h^2$ is the Von Neumann number and $\theta = kh$ is the reduced wavenumber. Hence the exact amplitude and phase become $|\tilde{G}| = e^{-D\theta^2}$ and $\tilde{\Phi} = \sigma\theta$, respectively.

Then, the numerical solution is sought in $\mathcal{V}_h^1$. The vector of DOF in an element Ω_j is assumed to be a Fourier mode $\mathbf{V}_j^{(n)} = \boldsymbol{\Xi}^{(n)} e^{ij\theta}$ with $\theta \in (-\pi, \pi]$. The amplification matrix for the FULL method is hence $\boldsymbol{\Xi}^{(n+1)} = \mathbf{G}_f \boldsymbol{\Xi}^{(n)}$ with

$$\mathbf{G}_f = \mathbf{I} - \langle \mathbf{A} \rangle^{-1} \langle \mathbf{R} \rangle \tag{3.3}$$

where $\mathbf{G}_f$ is in $\mathbb{C}^{N_p \times N_p}$ and $\langle \mathbf{A} \rangle = \mathbf{A}_{j-1} e^{-i\theta} + \mathbf{A}_j + \mathbf{A}_{j+1} e^{i\theta}$ represents the Fourier transform of the implicit matrix $\mathbf{A}$ in the element Ω_j.

The SIMP method (2.16) is a two-stage algorithm which may be written as $(\boldsymbol{\Xi}^{(n+1)}, \boldsymbol{\Xi}^{(n)})^\top = \mathbf{G}_s (\boldsymbol{\Xi}^{(n)}, \boldsymbol{\Xi}^{(n-1)})^\top$ where the exponent $\top$ denotes the transpose operator and the amplification matrix $\mathbf{G}_s$ in $\mathbb{C}^{2N_p \times 2N_p}$ is defined by

$$\mathbf{G}_s = \begin{pmatrix} (\mathbf{I} - \langle \mathcal{A} \rangle^{-1}(\langle \mathbf{R} \rangle + \langle \mathcal{B} \rangle)) & \langle \mathcal{A} \rangle^{-1} \langle \mathcal{B} \rangle \\ \mathbf{I} & 0 \end{pmatrix}. \tag{3.4}$$

We note that the amplification matrices are nonsingular and can be diagonalized as $\mathbf{G} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{-1}$ with $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_i)$ the diagonal matrix containing the eigenvalues and $\mathbf{Q}$ the matrix of right eigenvectors. Hence equations (3.3) and (3.4) reduce to the following form $\boldsymbol{\Pi}^{(n+1)} = \boldsymbol{\Lambda}\boldsymbol{\Pi}^{(n)}$ with $\boldsymbol{\Pi} = \mathbf{Q}^{-1}\boldsymbol{\Xi}$ and the stability properties of the numerical scheme follow from an eigenvalue problem for $\mathbf{G}$.

The condition for stability that any error should remain uniformly bounded for $n \to \infty$ at fixed time step Δt can be written as

$$\rho(\mathbf{G}) \leq 1, \quad \forall \theta \in (-\pi, \pi], \tag{3.5}$$

where $\rho(\mathbf{G})$ denotes the spectral radius of the amplification matrix (see for example [31]). Moreover, the eigenvalue analysis of amplification matrices will bring information on the spectral analysis of numerical errors for each scheme. Introducing the notation $\lambda = |\lambda| e^{i\Phi}$, the error in amplitude is called the dissipation error and is defined by $\epsilon_D = |\lambda| - |\tilde{G}|$. Likewise, the dispersion error is measured by $\epsilon_\phi = |\Phi/\tilde{\Phi}|$.

3.2. Spectral properties. The eigenspectra of amplification matrices for numerical schemes of the FULL and SIMP0 methods are plotted in Figure 3.1 for $p = 1$ and $p_s = 0$. We note that the SIMP0 method here reduces to a one-level scheme since the matrices $\mathbf{B}$ and $\mathbf{C}$ in equation (2.17) are zero. However, we observed that results obtained for $p = 1$ are similar to results obtained for higher polynomial degrees thus justifying this analysis. Results are obtained for a cell Reynolds number $Re_h = ch/\nu = 0.1$ and three different values of the Courant number. According to the condition (3.5), both methods are stable for $\sigma = 10$ and $\sigma = 20$ and are unstable for $\sigma = 30$.

A computation of theoretical constraints on the Courant and Von-Neumann numbers for which the methods are stable is given in Table 3.1 as a function of the cell
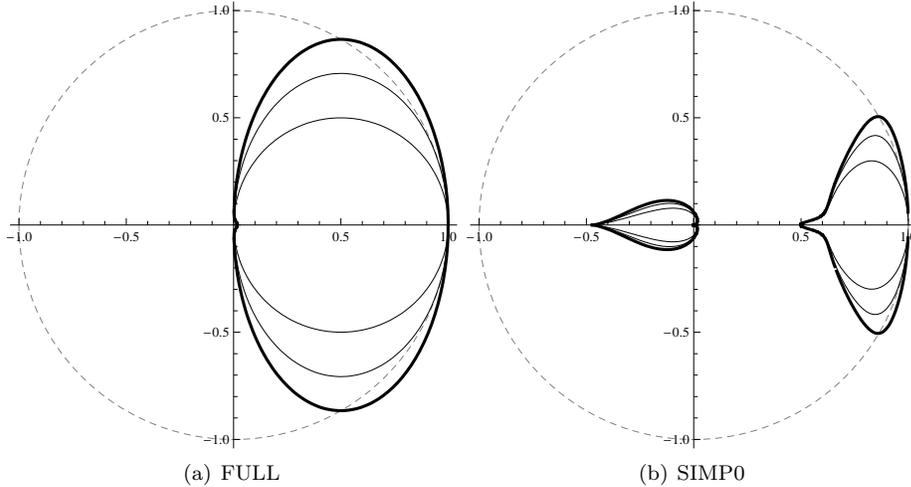
(a) FULL   (b) SIMP0

FIG. 3.1. *Amplification matrix eigenspectra in the complex plane for $p = 1$, $Re_h = 0.1$ and $\sigma = 10$ (thin lines), $\sigma = 20$ (normal lines) and $\sigma = 30$ (thick lines). The Von-Neumann number is set at $D = \sigma/Re_h$. The unit circle $\mathcal{C}(0,1)$ in dashed grey denotes the stability domain.*

TABLE 3.1
Stability limits for the FULL and SIMP0 methods with $p = 1$.

FULL	$Re_h = 0.01$	$\sigma \leq 200.0$	$D \leq 2 \times 10^4$
	$Re_h = 0.1$	$\sigma \leq 20.0$	$D \leq 2 \times 10^2$
	$Re_h = 1$	$\sigma \leq 2$	$D \leq 2$
	$Re_h = 5$	$\sigma \leq 0.4$	$D \leq 8 \times 10^{-2}$
	$Re_h = 7$	unstable	unstable
SIMP0	$Re_h = 0.01$	$\sigma \leq 200.0$	$D \leq 2 \times 10^4$
	$Re_h = 0.1$	$\sigma \leq 20.0$	$D \leq 2 \times 10^2$
	$Re_h = 1$	$\sigma \leq 2$	$D \leq 2$
	$Re_h = 5$	$\sigma \leq 0.4$	$D \leq 8 \times 10^{-2}$
	$Re_h = 7$	unstable	unstable

Reynolds number. As expected the stability becomes restrictive for weakly viscous flow regimes. Moreover, results of Figure 3.1 and Table 3.1 indicate that there is no modification of the stability constraints due to the SIMP method.

Necessary conditions for stability of both FULL and SIMP0 methods for $p = 1$ are derived in Appendix A and read

$$\sigma Re_h \leq 2 \quad \text{and} \quad Re_h \leq 6. \tag{3.6}$$

It is remarkable to note that every constraint in Table 3.1 is in agreement with above results making equation (3.6) a more tractable condition for stability of the numerical schemes. Setting $Re_h = 6$ in the first condition, one recovers a classical CFL condition $\sigma \leq 1/3$. Rewriting equation (3.6) as a function of nondimensional time steps leads to $\sigma \leq \min\{(2D)^{1/2}, 6D\}$ and justify that both methods are designed for viscous flows since there is no limitation on the viscous time step. The stability conditions (3.6) should also be compared to the case of pure advection $\nu = 0$ for

which the numerical scheme is stable under the prohibitive condition $\sigma \leq \mathcal{O}(h^{1/2})$ [10]. Moreover, numerical experiments have shown that for polynomials of degree p a Runge-Kutta method is necessary for an explicit time integration and that the CFL condition behaves as $\sigma(p) \simeq \sigma(0)/(2p+1)$ [14]. The addition of viscosity therefore stabilizes the numerical scheme with an implicit time integration and present numerical experiments tend to show that there is no p-dependency of the stability condition (see § 4).

However, both eigenspectra in Figure 3.1 present important differences. The distributions of eigenvalues seem not to be affected by the simplification in the neighbourhood of $\theta = 0$ where the stability condition (3.5) is violated, but are strongly modified for higher values of θ. Moreover, some eigenvalues for the SIMP method have negative real parts and imaginary parts present lower amplitudes compared to the FULL method. An in-depth analysis of the eigenspectra is given in Figure 3.2 where the dispersion and dissipation errors are plotted as a function of the reduced wavenumber. We consider stable numerical schemes with $\sigma = 10$ and two different cell Reynolds number values. The dispersion error ϵ_ϕ is a decreasing function of θ and vanishes for short wavelengths, but it is strongly amplified in the limit of long wavelengths for the SIMP scheme and for the eigenvalue presenting negative real part in Figure 3.1. This latter effect is amplified as the Reynolds number is increased for the SIMP method, while it is independent of Re_h for the FULL method. The dissipation error ϵ_D slightly increases with Re_h for the SIMP method in the limit of long wavelengths and does not tend to zero for short wavelengths. These characteristics will alter the damping properties of the SIMP scheme and its efficacy in reducing the error after a few iterations.

4. Numerical experiments. In this section we present three numerical experiments in 1D and 2D to illustrate the performances of the simplification of the implicit time discretization derived in § 2.4.

4.1. One-dimensional steady convection-diffusion equation. In this example, we let $\Omega = [0, 1]$ and consider the following problem

$$\frac{\partial u}{\partial x} - \frac{\partial^2 B(u)}{\partial x^2} = s(x) \quad \text{in } (0, 1) \tag{4.1}$$

with homogeneous Dirichlet boundary conditions $u(0) = u(1) = 0$ and where $B(u) = \nu(e^u - 1)$ with $\nu > 0$. The forcing function s is chosen so that the exact solution is $u(x) = \exp\left(\sin(2\pi x)/2\pi\right) - 1$. The problem (4.1) is solved by using the method introduced in § 2. We first investigate the asymptotic accuracy of the DG discretization on a sequence of successively finer meshes. The domain is discretized with an uniform grid $\Omega_h = \cup_{j=1}^N [(j-1)h, jh]$ with $h = 1/N$. Figure 4.1 displays the L^2-norm of the error $e_h \equiv u - u_h$ as a function of grid spacing for polynomial degrees $1 \leq p \leq 4$ and for the FULL and SIMP0 methods. Both schemes achieve optimal convergence $\|e_h\|_2 \sim h^{p+1}$ as h tends to zero in agreement with the theoretical study of Arnold et al. [3]. Likewise, error levels are identical between both methods indicating that the implicit simplification does not alter the error of the numerical scheme at convergence. We recall that setting $\Delta \mathbf{V}^{(n+1)} = \Delta \mathbf{V}^{(n)} = 0$ in the numerical schemes (2.11) and (2.16) of the FULL and SIMP methods, the numerical error satisfies the same condition $\mathbf{R}(\mathbf{V}^{(n)}) = 0$. The errors at convergence are therefore identical.

Next, We study the rate of convergence to steady state of the numerical solution. In Figures 4.2 and 4.3, the L^2-norm of the error is plotted as a function of the time
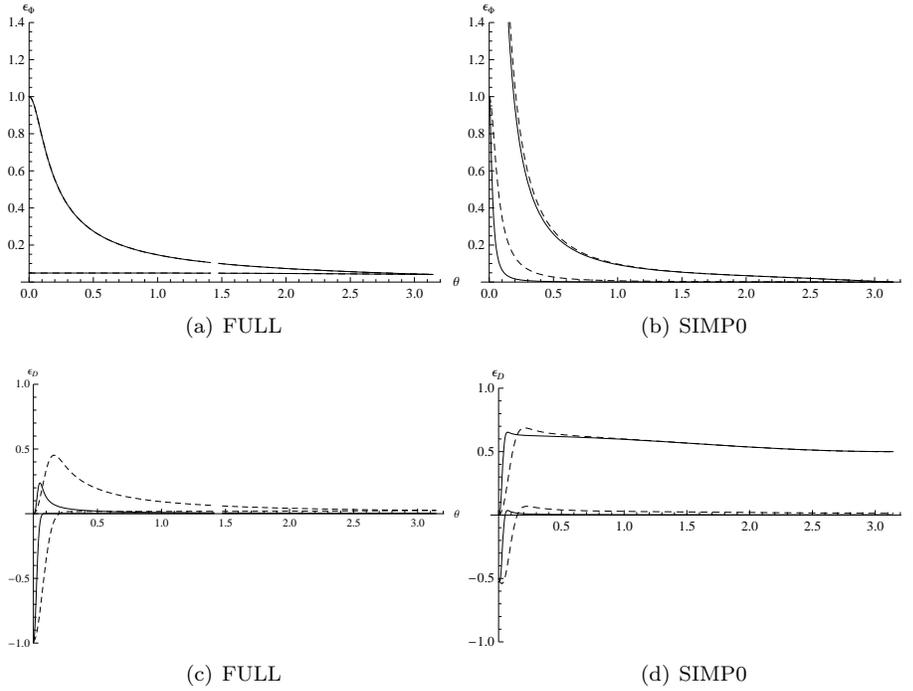
(a) FULL

(b) SIMP0

(c) FULL

(d) SIMP0

FIG. 3.2. *Dispersion error (top) and dissipation error (bottom) for $p = 1$, $\sigma = 10$, $Re_h = 0.01$ (continuous lines) and $Re_h = 0.1$ (dashed lines).*
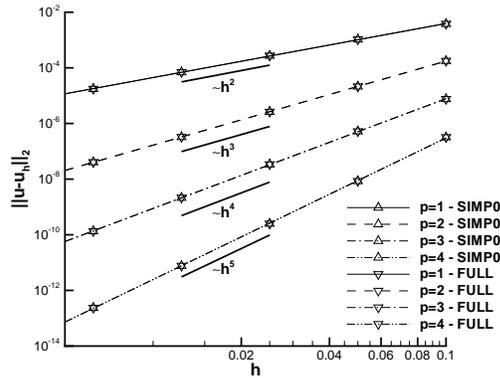

FIG. 4.1. *1D steady convection-diffusion problem: convergence of the FULL and SIMP DG methods under mesh refinement. The thick lines indicate power laws $\|e_h\|_2 \sim h^{p+1}$ for $1 \leq p \leq 4$.*

iteration (n) and of the CPU time. Figure 4.2 analyzes the effects of the polynomial degree for a fixed Reynolds number $Re_h = 0.1$ and a fixed grid, while mesh size effects with a fixed polynomial degree $p = 3$ are illustrated in Figure 4.3. The parabolic term of equation (4.1) has been linearized by using entropy solutions (see § 2.1) so the off-diagonal blocks of the implicit matrix are constructed only once at the beginning of
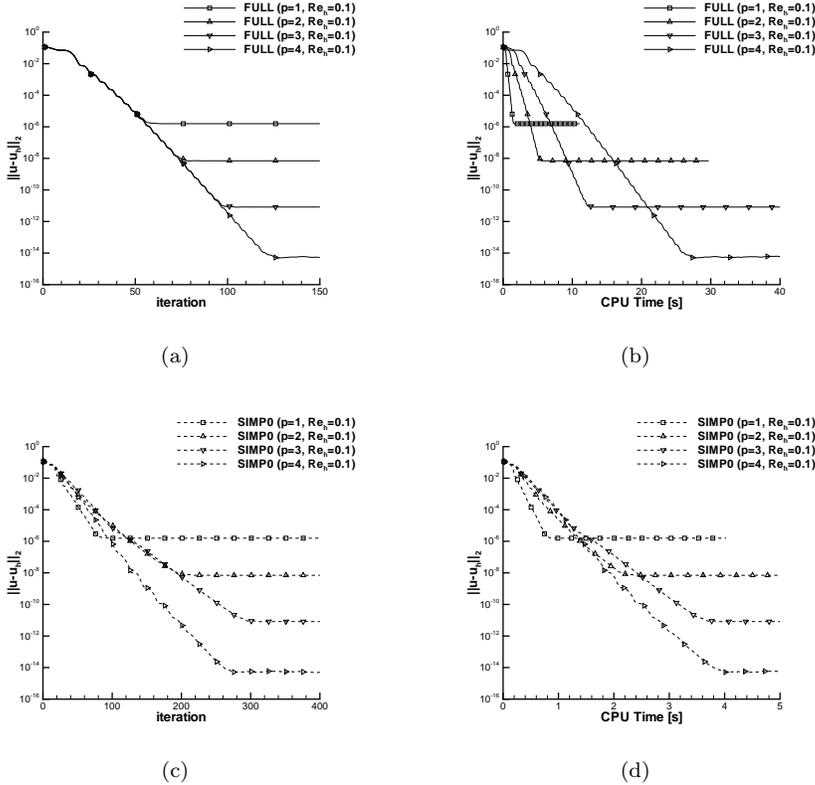
FIG. 4.2. *1D steady convection-diffusion problem: evolutions of the L^2-norm of the error e_h as a function of the number of iterations (left) and CPU time (right). Results are shown for numerical parameters $Re_h = 0.1$, $\sigma = 20$ and $D = 200$ and for the FULL method (top) and SIMP0 method (bottom).*

the computation. The diagonal blocks are evaluated at each time step to account for the time dependant mass matrix (2.13). The CPU time is initialized after the construction of off-diagonal blocks of the implicit matrix and evaluates the total time for the numerical resolution of the equation. Results with the simplified method are obtained for $p_s = 0$ and are compared to results obtained with the FULL method. In Figure 4.2, we take the highest time step allowed by the stability analysis in § 3. In Figure 4.3, the nondimensional time steps σ and D are chosen so that the physical time step remains constant for the sequence of meshes. These choices ensure that a fair comparison between CPU time for all configurations can be made. We note that the necessary conditions (3.6) were seen to hold for every polynomial degree during the numerical experiments.

As illustrated in Figures 4.2(a) and 4.3(a), the rate of convergence of the FULL method is independent of the polynomial degree and of the number of discretization elements. This property is different from precedent observations where a block diagonal preconditioner was needed to obtain an optimal rate of convergence (i.e. independent of the total number of unknowns of the problem) for a linear symmetric elliptic problem solved with a conjugate gradient method [29]. The SIMP method keeps the h-independency but is seen to be slightly dependent on p as shown in Fig-
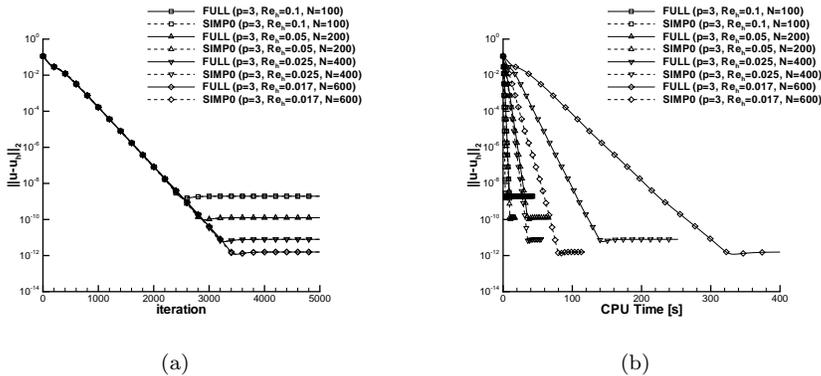
(a) (b)

FIG. 4.3. *1D steady convection-diffusion problem: evolutions of the L^2-norm of the error as a function of the number of iterations (left) and CPU time (right). Results are shown for the FULL method (continuous lines) and for the SIMP0 method (dashed lines) while keeping the time step constant $\Delta t = 2 \times 10^{-3}$ and a fixed polynomial degree $p = 3$ (see Table 4.1 for details on numerical parameters).*

ures 4.3(a) and 4.2(c), respectively. A comparison between Figures 4.2(a) and 4.2(c) indicates that, for a fixed polynomial degree p, the convergence to steady state solution needs more iterations for the SIMP0 method than for the FULL method. This lower rate of convergence per iteration may be attributed to the low damping properties of the SIMP0 method for a wide range of wavelength components of the error highlighted in § 3.2 for $p = 1$. In contrast, the SIMP0 method needs less CPU time to reach convergence (compare Figures 4.2(b,d)) and is less dependent on the size of the problem as seen in Figures 4.2(d) and 4.3(b). As noted in § 2.5, the SIMP0 algorithm requires fewer operations per time step than the FULL algorithm. The computationally more expensive operation is the matrix inversion and is indeed independent of the polynomial degree when $p_s = 0$.

These opposite effects promote the efficiency of the SIMP method for high polynomial degrees. In order to evaluate this efficiency, we define the speedup as the ratio between the CPU time needed to reach convergence with the FULL method and the CPU time for convergence with the SIMP method:

$$speedup = \frac{\text{CPU time(FULL)}}{\text{CPU time(SIMP)}}.$$

Table 4.1 indicates the speedups observed during numerical experiments for the 1D problem (4.1). Results for a fixed cell Reynolds number $Re_h = 0.1$ come from Figure 4.2 and results for a fixed polynomial degree $p = 3$ come from Figure 4.3. The saving in CPU time increases with the polynomial degree which may be associated to the increasing reduction in operation counts (see § 2.5 and Table 2.1). The speedup also increases with the number of discretization elements N and we conclude that the SIMP method results in significant cost savings for this application.

4.2. Two-dimensional steady convection-diffusion equation. In this second example, we consider the following problem in the square domain $\Omega = [0, 1]^2$ with homogeneous Dirichlet boundary conditions:

TABLE 4.1
1D steady convection-diffusion problem: speedups observed from experiments.

p	p_s	N	Re_h	σ	D	*speedup*
1	0	400	0.1	20	200	1.9
2	0	400	0.1	20	200	2.2
3	0	400	0.1	20	200	3.3
4	0	400	0.1	20	200	6.9
3	0	100	0.1	0.2	2.0	2.8
3	0	200	0.05	0.4	8.0	3.5
3	0	400	0.025	0.8	32.0	4.0
3	0	600	0.017	1.2	72.0	4.2

$$\nabla \cdot (\mathbf{c}u) - \nabla \cdot (\mathbf{B}(u)\nabla u) \;=\; s(\mathbf{x}) \quad \text{in } (0,1)^2, \qquad (4.2a)$$

$$u(x,y) \;=\; 0 \qquad \text{on } \partial\Omega, \qquad (4.2b)$$

with $\mathbf{B}(u) = \nu(e^u - 1)\mathbf{I}$, $\nu > 0$ and $\mathbf{c} = (1,1)^\top$. We choose s such that the exact solution of (4.2) reads

$$u(x,y) = \exp\left(xy\left(1 - e^{-\frac{1-x}{\nu}}\right)\left(1 - e^{-\frac{1-y}{\nu}}\right)\right) - 1.$$

We use partitions of Ω built from an uniform grid of m-by-m squares split into a total of $N = 2m^2$ isosceles triangles. The evolution of the L^2-norm of the error e_h is plotted as a function of iteration and CPU time in Figures 4.4 and 4.5 where the effects of polynomial degree and mesh size are investigated, respectively. Table 4.2 summarizes the speedup values observed during these experiments. Results are similar to those obtained with the 1D problem (see § 4.1). The convergence rate per iteration of the FULL method is independent of N and p (Figures 4.4(a) and 4.5(a)), while it slightly increases with p but is independent of N for the SIMP method (Figures 4.4(c) and 4.5(a)). As seen on Figures 4.4(a,c), the SIMP method needs more iterations to reach convergence than the FULL method. The application of the SIMP method with $p_s = 0$ for $1 \le p \le 3$ results in significant CPU time savings (compare Figures 4.4(b,d) and see Table 4.2). However, the SIMP method is seen to be unstable for $p = 4$ and $p_s = 0$. Keeping modes up to $p_s = 1$ and $p_s = 2$ as low-order modes is here necessary to reach a faster convergence than the FULL method (see results in Table 4.2). We attribute this effect to the communication improvement between DOF in adjacent elements as p_s is increased. Neglecting the coupling between the lowest order mode $p_s = 0$ and higher order modes slows down the convergence of the computation to steady state and may leads to an unstable numerical scheme. Keeping more low-order modes with $p_s = 1$ and $p_s = 2$ results in a lower saving in operation count of the SIMP algorithm. However, we note that the convergence properties are also improved as illustrated in Figure 4.4(c) and result in an increasing speedup of the SIMP method (see Table 4.2).

The overall efficiency of the SIMP method is therefore an increasing function of the polynomial degree. Furthermore, we observe in Figure 4.5(b) and Table 4.2 that the efficiency also increases with the number of elements, thereby confirming that the method is well suited to large-scale problems.

(a)

(b)

(c)

(d)

Fig. 4.4. *2D steady convection-diffusion problem: evolutions of the L^2-norm of the error as a function of the number of iterations (left) and CPU time (right). Results are shown for numerical parameters $Re_h = 0.1$, $\sigma = 10$ and $D = 100$ and for the FULL method (top) and SIMP method (bottom).*

(a)

(b)

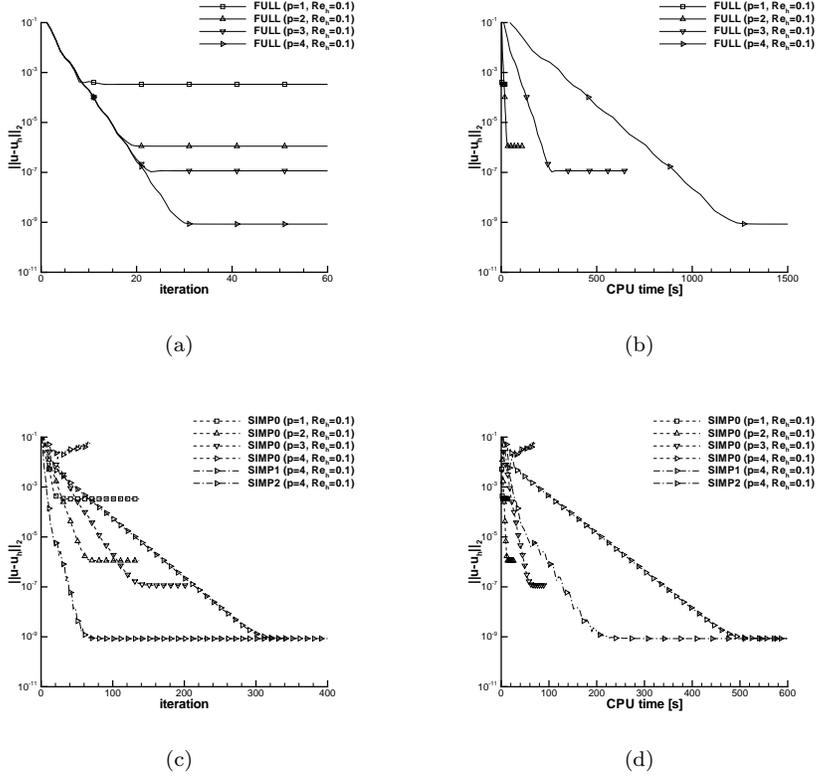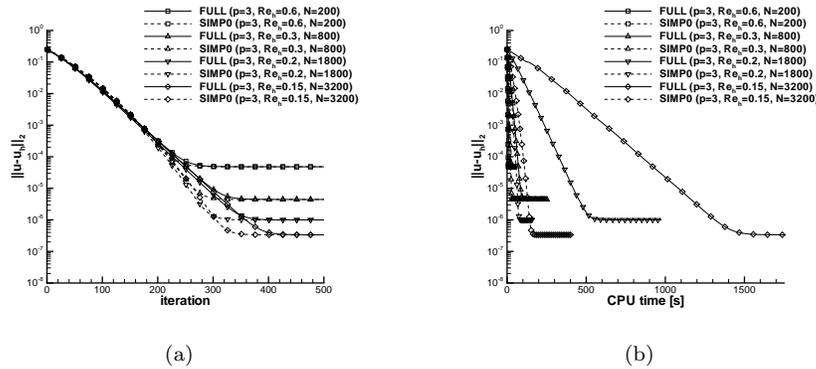Fig. 4.5. *2D steady convection-diffusion problem: evolutions of the L^2-norm of the error as a function of the number of iterations (left) and CPU time (right). Results are shown for the FULL method (continuous lines) and SIMP0 method (dashed lines) while keeping the time step constant $\Delta t = 6 \times 10^{-3}$ and a fixed polynomial degree $p = 3$ (see Table 4.2 for details on numerical parameters).*

p	p_s	N	Re_h	σ	D	*speedup*
1	0	3200	0.1	10	100	1.1
2	0	3200	0.1	10	100	2.6
3	0	3200	0.1	10	100	4.0
4	1	3200	0.1	10	100	2.3
4	2	3200	0.1	10	100	4.8
3	0	200	0.6	0.1	0.2	2.0
3	0	800	0.3	0.2	0.7	3.0
3	0	1800	0.2	0.3	2.0	6.9
3	0	3200	0.15	0.4	2.7	9.4

4.3. Two-dimensional unsteady convection-diffusion equation.

4.3.1. Problem formulation. In this third example, we aim at solving an unsteady problem with the SIMP method. Hence, we consider the following problem with periodic boundary conditions in $\Omega = [0,1]^2$:

$$
\begin{aligned}
u_t + \nabla \cdot (\mathbf{c}u) - \nabla \cdot (\mathbf{B}(u,\mathbf{x})\nabla u) &= s(\mathbf{x},t) \quad \text{in } \Omega \times (0,T], & \text{(4.3a)} \\
u(x,y,0) &= u_0(\mathbf{x}) \quad \text{in } \Omega, & \text{(4.3b)} \\
u(0,y,t) &= u(1,y,t) \quad \forall y \in [0,1], t \in (0,T], & \text{(4.3c)} \\
u(x,0,t) &= u(x,1,t) \quad \forall x \in [0,1], t \in (0,T], & \text{(4.3d)}
\end{aligned}
$$

with axial advection $\mathbf{c} = (c,0)^\top$ and a nonlinear diffusion tensor of the form $\mathbf{B}(u,\mathbf{x}) = \nu(e^u - 1)\mathbf{C}(\mathbf{x})$ with $\nu > 0$ and $\mathbf{C}(\mathbf{x}) = (3 + \cos \pi y)/4\mathbf{I}$, r denotes the distance from the centre of the square domain Ω. The initial condition is a Gaussian pulse $u_0(\mathbf{x}) = e^{-r^2/R^2}/4$ with $R = 1/7$ and we choose s such that the exact solution of (4.3) reads

$$
u(x,y,t) = \exp\left(e^{\frac{(x-\frac{1}{2}+cT)^2+(y-\frac{1}{2})^2}{R^2} - 4\pi\nu t}\right) - 1.
$$

4.3.2. Pseudo-time stepping method. Since we are interested with an unsteady problem, we use the present method in conjunction with subiterations to advance equation (4.3a) in a time-accurate manner. We use entropy solutions $v = \mathcal{U}_u(u)$ introduced in § 2.1 and add a pseudo-time derivative to equation (4.3a), one finally obtains

$$
v_\tau + u(v)_t + \nabla \cdot (\mathbf{c}u(v)) - \nabla \cdot (\mathbf{C}(\mathbf{x})\nabla v) = s(\mathbf{x},t), \tag{4.4}
$$

where τ denotes the pseudo-time. The space-time discretization introduced in § 2 leads to the following linear system

$$
\mathbf{A}\Delta\mathbf{V}^{(n+1,m+1)} = -\mathbf{R}(\mathbf{V}^{(n+1,m)}) - \frac{1}{\Delta t}\left(\mathbf{U}(v_h^{(n+1,m)}) - \mathbf{U}(v_h^{(n)})\right), \tag{4.5}
$$

where $(n+1,m)$ denotes the iterate on pseudo time step (m) between the physical times steps (n) and $(n+1)$. At steady-state, we have $\mathbf{V}^{(n+1)} = \lim_{m\to\infty}\mathbf{V}^{(n+1,m)}$.

TABLE 4.3

2D unsteady convection-diffusion problem: convergence of the FULL and SIMP methods under time refinement and orders of accuracy. Computations were obtained with $T = 2.5$ and $Re_h = 0.1$.

parameters			FULL method		SIMP method		
p	σ	D	$\|u - u_h\|_{L2}$	order	p_s	$\|u - u_h\|_{L2}$	order
1	10.0	100.0	$0.34570E-02$	—	0	$0.34568E-02$	—
1	5.0	50.0	$0.15807E-02$	1.13	0	$0.15806E-02$	1.13
1	2.5	25.0	$0.69140E-03$	1.19	0	$0.69137E-03$	1.19
1	1.25	12.5	$0.27143E-03$	1.35	0	$0.27144E-03$	1.35
2	10.0	100.0	$0.34570E-02$	—	0	$0.34567E-02$	—
2	5.0	50.0	$0.15807E-02$	1.13	0	$0.15806E-02$	1.13
2	2.5	25.0	$0.69142E-03$	1.19	0	$0.69147E-03$	1.19
2	1.25	12.5	$0.27147E-03$	1.35	0	$0.27174E-03$	1.35
3	10.0	100.0	$0.34570E-02$	—	0	$0.34578E-02$	—
3	5.0	50.0	$0.15807E-02$	1.13	0	$0.15810E-02$	1.13
3	2.5	25.0	$0.69142E-03$	1.19	0	$0.69173E-03$	1.19
3	1.25	12.5	$0.27147E-03$	1.35	0	$0.27171E-03$	1.35

By $\mathbf{U}(v_h^{(n+1,m)})$, we denote the vector whose components are the projections of $u(v_h^{(n+1,m)})$ in the discrete space $\mathcal{V}_h^p$. The implicit matrix $\mathbf{A}$ is now defined by

$$\mathbf{A} = \frac{1}{\Delta\tau}\mathbf{N} + \frac{1}{\Delta t}\mathbf{M}^{(n+1,m)} + \mathbf{L_v}, \tag{4.6}$$

where $\mathbf{N} = \mathrm{diag}(\mathbf{N_1}, \ldots, \mathbf{N_N})$ denotes the diagonal mass matrix whose entries for each block are $(\mathbf{N_j})_{kl} = \int_{\Omega_j} \phi^k \phi^l d\mathbf{x}$. Throughout this study, we use $\Delta\tau/\Delta t = 100$ and stop the subiterations when the L^2-norm of the right-hand side of equation (4.5) is below than $\epsilon = 10^{-4}$.

4.3.3. Convergence acceleration. As a validation of the pseudo-time stepping integration, we first investigate the asymptotic accuracy of the time discretization for both methods under time step refinement. The accuracy under space refinement has been validated in § 4.1. To isolate the temporal convergence behaviour as much as possible, effects of spatial discretization errors are minimized by using a sufficiently refined mesh. Table 4.3 presents the L^2-norms of the error as a function of the nondimensional time step σ and of the polynomial degree as well as the observed order of accuracy. Results are shown for the FULL and SIMP methods. The observed orders of accuracy are in good agreement with their theoretical values. We note that the error is the same for both methods and this property was seen to hold for all experiments presented throughout this study so that the SIMP method may be used for the resolution of unsteady flow problems.

Table 4.4 presents the speedup as a function of the polynomial degree for a fixed Courant number and four cell Reynolds number values. Since we are interested in time-accurate solutions, the computations are carried out at a unit Courant number $\sigma = 1$. Therefore, as the time-step is reduced, the spatial mesh is refined correspondingly. We observe that the speedup is an increasing function of p and N in agreement with previous results. However, results show moderate CPU time savings for low polynomial degrees and coarse meshes. We note that the cell Reynolds number has a strong impact on the efficiency of the method. The speedup indeed increases for

TABLE 4.4
2D unsteady convection-diffusion problem: speedups observed from computations with $T = 2.5$, $\sigma = 1$, $D = \sigma/Re_h$ with different values of the cell Reynolds number.

p	N	\multicolumn{2}{c}{$Re_h = 2 \times 10^{-3}$}		\multicolumn{2}{c}{$Re_h = 10^{-2}$}		\multicolumn{2}{c}{$Re_h = 10^{-1}$}		\multicolumn{2}{c}{$Re_h = 1$}
		p_s	speedup	p_s	speedup	p_s	speedup	p_s	speedup
1	200	0	2.1	0	1.5	0	1.6	0	1.6
1	800	0	1.8	0	1.4	0	1.3	0	1.2
1	1800	0	1.7	0	1.2	0	1.3	0	1.1
1	3200	0	1.5	0	1.2	0	1.2	0	1.1
2	200	0	1.6	0	1.2	0	1.2	1	1.3
2	800	0	1.4	0	1.2	0	1.3	1	1.4
2	1800	0	1.6	0	1.3	0	1.2	1	1.3
2	3200	0	2.5	0	1.9	0	1.7	1	1.4
3	200	0	2.2	1	1.7	1	1.6	1	1.5
3	800	0	2.7	1	1.8	1	1.8	1	1.4
3	1800	0	5.2	1	3.7	1	2.8	1	1.9
3	3200	0	11.1	1	4.0	1	2.8	1	1.8
4	200	2	1.4	2	1.3	2	1.6	2	1.7
4	800	2	4.5	2	4.4	2	5.6	2	3.9
4	1800	2	7.1	2	7.4	2	7.3	2	9.4
4	3200	2	5.6	2	13.3	2	5.7	2	3.9

diffusion dominated problems $Re_h \ll 1$ and achieves very high values for high polynomial degrees $p \geq 3$. In contrast, the efficiency of the method decreases when the convection becomes significant. In particular, for polynomial degrees $p = 2$ and $p = 3$ and $Re_h = 1$ the method SIMP0 converges slower than the FULL method and we need to use a SIMP1 method leading to lower speedup than those observed for lower Re_h values. We attribute this effect to the worsening of the damping properties of the SIMP numerical scheme when increasing Re_h as highlighted in Figure 3.2(d). We expect that this effect is amplified as the polynomial degree is increased. Moreover, the SIMP1 method achieves higher speedups than the SIMP0 method for $p = 3$ and $Re_h = 0.1$ or $Re_h = 0.01$. We did not observe this effect in experiments on the steady flow problems (see Tables 4.1 and 4.2) where the SIMP0 method was seen to be faster for $p = 3$, $Re_h = 0.1$ and $\sigma = 10$. These results indicate that the SIMP method needs lower p_s values for higher time steps.

5. Concluding remarks. An efficient and robust time integration procedure has been successfully applied to nonlinear convection-diffusion equations. We have used an explicit-implicit time discretization in which the convective term is treated explicitly and the diffusive term implicitly. The method aims at simplifying the implicit problem based on an uncoupling between DOF in the discretization element and high-order DOF in adjacent elements. At each time step, this simplification allows the decomposition of the numerical resolution of the underlying algebraic system into two stages: first an implicit problem is solved for the low-order modes only, then higher-order modes are locally reconstructed. This method keeps the locality of the DG discretization and simplifies the task of solving the algebraic system. In terms of CPU time, numerical experiments demonstrate an acceleration of convergence of the numerical solution with an iterative algorithm. The method becomes significantly faster for large number of elements and high polynomial degree. Likewise, this method

is well suited to steady and unsteady flow problems by using a pseudo-time stepping method. Cell Reynolds number effects are assessed and indicate a better efficiency of the method when applied to diffusion dominated flow problems.

These results suggest the use of this method in diffusion dominated flow regions such as boundary layers. This would allow for a significant reduction in computational cost in these regions, where the Von-Neumann condition becomes the most restrictive on the time step. We note that a coupled use of FULL and SIMP methods in different flow regions is easy to implement. The extension of the method to multidimensional problems and to systems of conservation laws is straightforward but its efficiency must yet be analyzed. In particular, it should be clarified to what extent the method can be generalized to convection dominated flows. Moreover, even though the method has been introduced in the context of a BR2 discretization it can be easily applied to other viscous term discretizations [4, 13, 28, 37]. The method has already been successfully applied to the BR1 discretization of a linear elliptic equation [16]. Finally, a theoretical analysis of the coupling between DOF in adjacent elements must be carried out and may help in the construction of convergence acceleration algorithms.

Appendix. In this appendix, we prove the necessary conditions (3.6) for stability of the FULL and SIMP0 methods with polynomials of degree $p = 1$.

Let $sp(\mathbf{G}_f) = \{\lambda_{1,f}, \lambda_{2,f}\}$ be the spectrum of the amplification matrix (3.3) for the FULL scheme. A general analysis of spectral radii of both amplification matrices is a tedious task. Instead, a Taylor series of eigenvalues up to second-order in the neighbourhood of $\theta = 0$ gives

$$
\begin{aligned}
\lambda_{1,f} &= 1 + i\sigma\theta - D\theta^2 + \mathcal{O}(\theta^3), \\
\lambda_{2,f} &= \frac{1 + 6\sigma}{1 + 36D} - \frac{3i\sigma\theta}{1 + 36D} - \frac{(\sigma + 3D(6\sigma - 1))\theta^2}{(1 + 36D)^2} + \mathcal{O}(\theta^3).
\end{aligned}
$$

According to equation (3.5), the stability condition $|\lambda_{1,f}| \leq 1$ on the first eigenvalue imposes $\partial^2|\lambda_{1,f}|/\partial\theta^2|_{\theta=0} = 2(\sigma^2 - 2D) \leq 0$. A necessary condition for the second eigenvalue reads $|\lambda_{2,f}|_{\theta=0} = (1 + 6\sigma)/(1 + 36D) \leq 1$. Hence, a necessary condition for stability of the FULL method may be written as

$$
\sigma Re_h \leq 2 \quad \text{and} \quad Re_h \leq 6.
$$

As pointed out in § 3.2, the SIMP0 method for $p = 1$ reduces to a one-level scheme. A Taylor series of the amplification matrix (3.4) eigenvalues $sp(\mathbf{G}_s) = \{\lambda_{1,s}, \lambda_{2,s}\}$ in the neighbourhood of $\theta = 0$ reads

$$
\begin{aligned}
\lambda_{1,s} &= 1 + i\sigma\theta - D\theta^2 + \mathcal{O}(\theta^3), \\
\lambda_{2,s} &= \frac{1 - 12D + 6\sigma}{1 + 24D} - \frac{3i\sigma\theta}{1 + 24D} - \frac{(3D(1 + 12D) - \sigma(1 + 6D))\theta^2}{(1 + 24D)^2} + \mathcal{O}(\theta^3),
\end{aligned}
$$

and the stability conditions on both eigenvalues read $2(\sigma^2 - 2D) \leq 0$ and $(1 - 12D + 6\sigma)/(1 + 24D) \leq 1$. Hence, one recovers the same necessary conditions (3.6) as those derived for the FULL method.

REFERENCES

[1] H. W. ALT AND S. LUCKHAUS, *Quasilinear elliptic-parabolic differential equations*, Math. Z., 183 (1983), pp. 311–341.

[2] J.-B. APOUNG KAMGA AND B. DESPRÉS, *CFL condition and boundary conditions for DGM approximation of convection-diffusion*, SIAM J. Numer. Anal., 44 (2006), pp. 2245–2269.

[3] D. N. ARNOLD, F. BREZZI, B. COCKBURN AND L. D. MARINI, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal., 39 (2002), pp. 1749–1779.

[4] F. BASSI AND S. REBAY, *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations*, J. Comput. Phys., 131 (1997), pp. 267–279.

[5] F. BASSI, S. REBAY, G. MARIOTTI, S. PEDINOTTI AND M. SAVINI, *A High-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows*, In proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, R. Decuypere, G. Dibelius (eds.), Antwerpen, Belgium, 1997.

[6] F. BASSI, A. CRIVELLINI, S. REBAY AND M. SAVINI, *Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and $k - \omega$ turbulence model equations*, Comput. Fluids, 34 (2005), pp. 507–540.

[7] F. BASSI AND S. REBAY, *Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations*, Int. J. Numer. Methods Fluids, 40 (2002), pp. 197–202.

[8] C. E. BAUMANN AND J. T. ODEN, *A discontinuous hp finite element method for the Euler and Navier-Stokes equations*, Int. J. Numer. Methods Fluids, 31 (1999), pp. 79–95.

[9] M. BORREL AND J. RYAN, *A new discontinuous Galerkin method for the Navier-Stokes equations*, Lecture Notes in Comp. Sci. and Eng., 76 (2011), pp. 373–381.

[10] G. CHAVENT AND B. COCKBURN, *The local projection $P^0 P^1$-discontinuous Galerkin finite element method for scalar conservative laws*, M2AN Math. Model. Anal. Numer., 23 (1989), pp. 565–592.

[11] I. CHOQUET, P. DEGOND AND B. LUCQUIN-DESREUX, *A hierarchy of diffusion models for partially ionized plasmas*, Discrete and Continuous Dynamical Systems - Series B, 8 (2007), pp. 735–772.

[12] B. COCKBURN AND C. W. SHU, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: general framework*, Math. Comp., 52 (1989), pp. 411–435.

[13] B. COCKBURN AND C. W. SHU, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal., 35 (1998), pp. 2440–2463.

[14] B. COCKBURN AND C. W. SHU, *Runge-Kutta discontinuous Galerkin methods for convection-dominated problems*, J. Sci. Computing, 16 (2001), pp. 173–261.

[15] C. M. DAFERMOS, *Hyperbolic Conservation Laws in Continuum Physics, 3rd ed.*, Springer, 2010.

[16] T. DAIRAY, *Développement et évaluation d'une méthode implicite à coût réduit appliquée aux schémas de type Galerkin discontinu*, MSc Thesis, ONERA, 2010.

[17] P. DEGOND, S. GÉNIEYS AND A. JÜNGEL, *Symmetrization and entropy inequality for general diffusion equations*, C. R. Acad. Sci., 325 (1997), pp. 963–968.

[18] L. T. DIOSADY AND D. L. DARMOFAL, *Preconditioning methods for discontinuous Galerkin solutions of the Navier-Stokes equations*, J. Comput. Phys., 228 (2009), pp. 3917–3935.

[19] V. DOLEJŠÌ, *On the discontinuous Galerkin method for the numerical solution of the Navier-Stokes equations*, Int. J. Numer. Methods Fluids, 45 (2004), pp. 1083–1106.

[20] V. DOLEJŠÌ, M. FEISTAUER AND C. SCHWAB, *On discontinuous Galerkin methods for nonlinear convection-diffusion problems and compressible flow*, Math. Bohem., 127 (2002), pp. 163–179.

[21] V. DOLEJŠÌ, *Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows*, Commun. Comput. Phys., 4 (2008), pp. 231–274.

[22] V. DOLEJŠÌ, M. HOLÍK AND J. HOZMAN, *Efficient solution strategy for the semi-implicit discontinuous Galerkin discretization of the Navier-Stokes equations*, J. Comput. Phys., In Press, 2010.

[23] M. DUBINER, *Spectral methods on triangles and other domains*, SIAM J. Sci. Comput., 6 (1991), pp. 345–390.

[24] A. ERN, A. F. STEPHANSEN AND P. ZUNINO, *A Discontinuous Galerkin method with weighted averages for advection–diffusion equations with locally small and anisotropic diffusivity*, IMA J. Numer. Anal., 29 (2009), pp. 235–256.

[25] K. J. FIDKOWSKI, T. A. OLIVER, J. LU AND D. L. DARMOFAL, *p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations*, J. Comput. Phys., 207 (2005), pp. 92–113.

[26] E. Godlewski and P.-A. Raviart, *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, Springer, 1986.

[27] A. Harten, *On the symmetric form of systems of conservation laws with entropy*, J. Comput. Phys., 49 (1983), pp. 151–164.

[28] R. Hartmann and P. Houston, *Symmetric interior penalty DG methods for the compressible Navier-Stokes equations. I: Method formulation*, Int. J. Numer. Anal. Model., 3 (2006), pp. 1–20.

[29] R. Hartmann, M. Lukacova-Medvidova and F. Prill, *Efficient preconditioning for the discontinuous Galerkin finite element method by low-order elements*, Appl. Numer. Math. 59 (2009), pp. 1737–1753.

[30] R. Hartmann, J. Held, T. Leicht, and F. Prill, *Discontinuous Galerkin methods for computational aerodynamics - 3D adaptive flow simulation with the DLR PADGE code*, Aerospace Science and Technology, 14 (2010), pp. 512–519.

[31] C. Hirsch, *Numerical computation of internal and external flows: fundamentals of computational fluid dynamics*, John Wiley & Sons, 2007.

[32] N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven and K. Sorensen (eds.), *ADIGMA - A european initiative on the development of adaptive higher-order variational methods for aerospace applications*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, 113 (2010), Springer Verlag.

[33] B. Landmann, M. Kessler, S. Wagner and E. Krämer, *A parallel high-order discontinuous Galerkin code for laminar and turbulent flows*, Comp. Fluids, 37 (2008), pp. 427–438.

[34] P. Lesaint and P.-A. Raviart, *On a finite element method for solving the neutron transport equation, in Mathematical Aspects of Finite Elements in Partial Differential Equations*, de Boor ed., Academic Press, New York, 1974, pp. 89–123.

[35] A. Majda and R. L. Pego, *Stable viscosity matrices for systems of conservation laws*, Journal of Differential Equations, 56 (1985), pp. 229–262.

[36] C. R. Nastase and D. J. Mavriplis, *High-order discontinuous Galerkin methods using an hp-multigrid approach*, J. Comput. Phys., 213 (2006), pp. 330–357.

[37] J. Peraire and P. O. Person, *The compact discontinuous Galerkin (CDG) method for elliptic problems*, SIAM J. Sci. Comput., 30 (2008), pp. 1806–1824.

[38] P. O. Person and J. Peraire, *Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations*, SIAM J. Sci. Comput., 30 (2008), pp. 2709–2733.

[39] P. Rasetarinera and M. Y. Hussaini, *An efficient implicit discontinuous spectral Galerkin method*, J. Comput. Phys., 172 (2001), pp. 718–738.

[40] W. H. Reed and T. R. Hill, *Triangular mesh methods for the neutron transport equation*, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, NM, 1973.

[41] Y. Saad, *Iterative methods for sparse linear systems, Second ed.*, Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2003.

[42] S. J. Sherwin, R. M. Kirby, J. Peiró, R. L. Taylor and O. C. Zienkiewicz, *On 2D elliptic discontinuous Galerkin methods*, Int. J. Numer. Methods Fluids, 65 (2006), pp. 752–784.

[43] B. van Leer, M. Loy and M. van Raalte, *A discontinuous Galerkin method for diffusion based on recovery*, in Proceedings of the 18th AIAA CFD Conference, Miami, FL, 2007, AIAA-2007-4083.

[44] K. Yasue, M. Furudate, N. Ohnishi and K. Sawada, *Implicit discontiunous Galerkin method for RANS simulation utilizing pointwise relaxation algorithm*, Commun. Comput. Phys., 7 (2010), pp. 510–533.