# High order ADER-IPDG methods for the unsteady advection-diffusion equation

Michel Bergmann, Afaf Bouharguane, Angelo Iollo, Alexis Tardieu

▶ **To cite this version:**

Michel Bergmann, Afaf Bouharguane, Angelo Iollo, Alexis Tardieu. High order ADER-IPDG methods for the unsteady advection-diffusion equation. 2024. hal-04032238v2

**HAL Id: hal-04032238**

**https://inria.hal.science/hal-04032238v2**

Preprint submitted on 15 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

# High order ADER-IPDG methods for the unsteady advection-diffusion equation

Michel Bergmann[2,3], Afaf Bouharguane[1,2,3], Angelo Iollo[1,2,3] and Alexis Tardieu[1,2,3*]

[1]Université de Bordeaux, Bordeaux, France.
[2]Centre Inria de l'Université de Bordeaux, Memphis team, Talence, France.
[3]Institut de Mathématiques de Bordeaux, UMR CNRS 5251, Talence, France.

*Corresponding author(s). E-mail(s): alexis.tardieu@u-bordeaux.fr;
Contributing authors: michel.bergmann@u-bordeaux.fr; afaf.bouharguane@u-bordeaux.fr; angelo.iollo@u-bordeaux.fr;

**Abstract**

We present a high-order Galerkin method in both space and time for the one-dimensional unsteady linear advection-diffusion equation. Three Interior Penalty Discontinuous Galerkin (IPDG) schemes are detailed for the space discretization, while the time integration is performed at the same order of accuracy thanks to an Arbitrary high order DERivatives (ADER) method. The orders of convergence of the three ADER-IPDG methods are carefully examined through numerical illustrations, showing that the approach is consistent, accurate and efficient. The numerical results indicate that the symmetric version of IPDG is typically more accurate and more efficient compared to the other approaches.

**Keywords:** Advection-diffusion, Galerkin, ADER approach, IPDG, high-order schemes, empirical convergence rates.

# 1 Introduction

The numerical integration of the unsteady advection-diffusion equation has been widely studied by using a large variety of schemes. Finite Differences (FD), Finite Volumes (FV) or Finite Elements (FE) methods are the most common space discretization techniques, each with advantages and drawbacks especially for reaching a high order accuracy with compact stencils. The time integration is generally achieved thanks to Runge-Kutta (RK) techniques. A compromise between the upwind FV schemes and the FE methods is the Residual Distribution Scheme which can be coupled to a RK integration in time [1, 2]. Halfway between the high accuracy of continous polynomial basis offered by classical FE methods and the local formulation of a FV scheme over a space-time cell, the Discontinous Galerkin (DG) schemes [3–5] represent the numerical solution by a high degree polynomial in each cell.

Thanks to its local space-time high-order formulation, the Arbitrary high-order DERivatives-FV (ADER-FV) scheme [6, 7] proved accurate and efficient on a large variety of problems. In addition, the Multidimensional Optimal Order Detection (MOOD) paradigm has been developed in the last decade. It allows to detect the optimal order to ensure the preservation of the physical properties of the solution, giving birth to high order ADER-MOOD schemes for hyperbolic systems of conservation laws [8, 9]. On the other hand, the ADER-DG combination is particularly suitable to reach a desirable accuracy at the interface between overset grids or at level jumps with hierarchical meshes like quadtrees or octrees. In this sense, the work presented in [10] and [11] has shown promising results respectively for the advection-diffusion equation and the incompressible Navier-Stokes equations using a second or third order ADER-FV scheme on Chimera meshes. In our ongoing research, the plan is to set an additional step to extend that work to improve accuracy at quadtrees or octrees level jumps (see for example [12]) in hierarchical meshes. However, to attain a higher order precision at level jumps with a compact stencil for improved parallelism, we switch from the ADER-FV to the ADER-DG framework.

The Interior Penalty Discontinuous Galerkin (IPDG) scheme is a combination of the DG framework and the Interior Penalty (IP) methods, the latter being detailed in [13, 14] and applied to a wide range of equations [13, 15]. As in the DG framework, the solution is approximated within each cell using a set of basis functions. Since the solution is possibly discontinuous across the edges of the cells, the IPDG method introduces a penalty term at the interfaces between the cells to ensure the continuity of the solution and its gradient.

The novelty of the approach presented in this paper is to combine the IPDG method to handle the diffusion term and the ADER method to perform the space-time integration. This can result in a method that is accurate, efficient, and can deal with a wide range of advection-diffusion problems. More precisely, this article proposes a comparison between the classical ADER-DG scheme used in [9, 16–22], which is equivalent to the IIPG method, and its symmetric and anti-symmetric variants, named respectively SIPG and NIPG methods.

Section 2 introduces the important notations as well as the space-time structure used in this paper. In section 3, we present three ADER-IPDG prediction-correction schemes carefully describing the numerical fluxes chosen at interfaces between cells. In section 4, we validate our implementations by numerical illustrations. Section 5 is devoted to some conclusions and perspectives of our work.

# 2 Preliminary definitions and notations

## 2.1 The advection-diffusion equation

Let us consider a function $u : \Omega \times [0, T_f] \to \mathbb{R}$, where $\Omega := [x^-, x^+] \subset \mathbb{R}$, as well as the constants $a \in \mathbb{R}$ and $\lambda \in \mathbb{R}^+$ standing for the advection velocity and the diffusion coefficient respectively. On the same space-time domain as the unknown $u$, we denote by $S$ the source term and $F := F_a + F_d$ the flux function that is expressed as the sum of its advective and diffusive components, respectively defined as

$$F_a(u) := au, \quad F_d(u) := -\lambda \partial_x u. \tag{1}$$

Thus, the advection-diffusion equation completed with an initial condition as well as Dirichlet boundary conditions reads

$$\begin{cases} \partial_t u + \partial_x F(u) = S & \text{on } \Omega \times [0, T_f], \\ u(x, 0) = u_0(x) & \text{on } \Omega, \\ u(x^\pm, t) = g_\pm(t) & \text{on } \partial\Omega \times [0, T_f]. \end{cases} \tag{2}$$

The space and time domains $\Omega$ and $[0, T_f]$ are discretized respectively into $N_x$ uniform intervals $\omega_i$ of size $\Delta x$ and $N_t$ uniform intervals $T^n$ of size $\Delta t$ such that

$$\Omega := \bigcup_{i=1}^{N_x} \omega_i, \qquad \omega_i := [x_{i-1}, x_i], \qquad x_i := x^- + i\Delta x, \tag{3a}$$

$$[0, T_f] := \bigcup_{n=0}^{N_t - 1} T^n, \qquad T^n := [t^n, t^{n+1}], \qquad t^n := n\Delta t. \tag{3b}$$

On a given space-time cell $c_i^n := \omega_i \times T^n$, we consider the usual change of variables

$$x(\xi) := x_i + \xi \Delta x, \qquad \xi \in \hat{\omega} := [0, 1], \tag{4a}$$

$$t(\tau) := t^n + \tau \Delta t, \qquad \tau \in \hat{T} := [0, 1], \tag{4b}$$

which allows to rewrite equation (2) with respect to the reference variables $\xi \in \hat{\omega}$ and $\tau \in \hat{T}$ instead of the physical variables $x \in \omega_i$ and $t \in T^n$. By

using the chain rule for the derivative of composite functions, and adding a hat on quantities that are defined in the reference coordinates, we can define the following advective and diffusive fluxes

$$\hat{F}_a(\hat{u}) := a\hat{u}, \quad \hat{F}_d(\hat{u}) := -\frac{\lambda}{\Delta x}\partial_\xi \hat{u}, \tag{5}$$

and we still have $\hat{F} := \hat{F}_a + \hat{F}_d$. As a consequence, the advection-diffusion defined on $c_i^n$ can be expressed in the reference variables on $\hat{c} := \hat{\omega} \times \hat{T}$ as follows

$$\partial_\tau \hat{u} + \left(\frac{\Delta t}{\Delta x}\right) \partial_\xi \hat{F}(\hat{u}) = \Delta t \hat{S}. \tag{6}$$

The main advantage of working on this formulation at this early stage (see [10]) is that the space and time discretizations can be entirely developed in the reference space-time cell $\hat{c}$, which avoids making change of variables in integrals as it is generally done for classical finite elements methods.

## 2.2 Reference space-time structure

Generally speaking, Galerkin methods are based on a high degree piecewise polynomial representation of the numerical solution on the space domain $\Omega$, while the high order in time is reached by using RK schemes for instance. In the particular case of the DG framework, the solution may be discontinuous across interfaces which implies that the constraints on the localization of the degrees of freedom disappear. As a consequence, a remarkable interest in combining the DG discretization in space with the ADER approach in time is the possibility to build a space-time structure as a tensor product of the polynomial basis with respect to the space and time variables. In this paper, we use the 1D Lagrange polynomials $\theta_1, \ldots, \theta_K \in \mathbb{P}_{K-1}$ (the vector space of polynomials of degree $d \leq K-1$) passing through the 1D Gauss-Legendre quadrature points denoted by $p_1, \ldots, p_K$. They write

$$\theta_s(p) := \prod_{k \neq s} \left(\frac{p - p_k}{p_s - p_k}\right), \quad p = \xi \in \hat{\omega},\ \tau \in \hat{T}. \tag{7}$$

The 2D space-time Lagrange basis functions $\Theta_1, \ldots, \Theta_L$ (with $L := K^2$) passing through the 2D quadrature points $\pi_1, \ldots, \pi_L$ are then defined as products of the 1D polynomials introduced in (7)

$$\Theta_\ell(\xi, \tau) := \theta_s(\xi)\theta_{s'}(\tau), \quad \ell := \mathbf{m}(s, s'), \tag{8}$$

where the mapping $\mathbf{m}$ between local 1D indices $(s, s')$ and the 2D global index $\ell$ is defined as

$$\mathbf{m} : \begin{cases} \{1, \ldots, K\}^2 & \to \{1, \ldots, L\} \\ (s, s') & \mapsto \ell := K(s'-1) + s. \end{cases} \tag{9}$$

For instance, the $\ell$-th 2D quadrature point is defined by $\pi_\ell := (p_s, p_{s'}) \in \hat{c}$ where we have taken $\ell := \mathbf{m}(s, s')$. The case $K = 3$ is illustrated in figure 1.

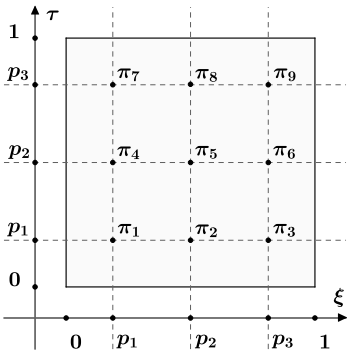


**Fig. 1** 2D space-time quadrature points $\pi_\ell$ obtained as tensor products of the space and time 1D quadrature points $p_k$.

This structure is particularly convenient as its extension from 1D to 2D or 3D problems in space is straightforward for structured grids. Indeed, the 2D or 3D basis functions in space can be defined and implemented as the products of the 1D basis functions with respect to the two or three space variables, the time variable being treated exactly in the same way as for the 1D case. This remark on structured grids stands not only for regular meshes but also for quadtree or octree grids, which are the target of this work.

## 2.3 Local scalar products and matrix definitions

Let us introduce the following notations corresponding to various scalar products defined by integrals

$$\langle f, g \rangle := \int_{\hat{\omega}} \int_{\hat{T}} f(\xi, \tau) g(\xi, \tau) \mathrm{d}\xi \mathrm{d}\tau, \tag{10a}$$

$$[f, g]_\tau := \int_{\hat{\omega}} f(\xi, \tau) g(\xi, \tau) \mathrm{d}\xi, \tag{10b}$$

$$\{f, g\}_\xi := \int_{\hat{T}} f(\xi, \tau) g(\xi, \tau) \mathrm{d}\tau, \tag{10c}$$

$$[f, g] := \int_{\hat{\omega}} f(p) g(p) \mathrm{d}p, \quad p = \xi, \tau. \tag{10d}$$

This allows us to define some classical FE matrices below.

- The space-time matrices belonging to the space $\mathcal{M}_L(\mathbb{R})$

$$\underline{M}_{k,\ell} := \langle \Theta_k, \Theta_\ell \rangle, \qquad \underline{M}^{\tau}_{k,\ell} := [\Theta_k, \Theta_\ell]_\tau, \tag{11a}$$

$$\underline{K}^{\tau}_{k,\ell} := \langle \partial_\tau \Theta_k, \Theta_\ell \rangle, \qquad \underline{K}^{\xi}_{k,\ell} := \langle \partial_\xi \Theta_k, \Theta_\ell \rangle. \tag{11b}$$

- The mixed matrices belonging to the space $\mathcal{M}_{K,L}(\mathbb{R})$

$$\underline{N}^{\tau}_{k,\ell} := [\theta_k, \Theta_\ell]_\tau, \qquad \underline{P}_{k,\ell} := \langle \theta_k, \Theta_\ell \rangle, \tag{12a}$$

$$\underline{C}_{k,\ell} := \langle \partial_\xi \theta_k, \Theta_\ell \rangle, \qquad \underline{E}_{k,\ell} := \langle \partial_\xi \theta_k, \partial_\xi \Theta_\ell \rangle, \tag{12b}$$

$$\underline{Q}^{p,q}_{k,\ell} := [\theta_k(p), \Theta_\ell(q,\cdot)], \qquad \underline{R}^{p,q}_{k,\ell} := [\theta_k(p), \partial_\xi \Theta_\ell(q,\cdot)], \tag{12c}$$

$$\underline{W}^{p,q}_{k,\ell} := [\partial_\xi \theta_k(p), \Theta_\ell(q,\cdot)]. \tag{12d}$$

- The space matrix belonging to the space $\mathcal{M}_K(\mathbb{R})$

$$\underline{N}_{k,\ell} := [\theta_k, \theta_\ell]. \tag{13a}$$

For the sake of clarity, for instance, the notation $\langle f, g \rangle$ denotes the space-time integral of the quantity $f(\xi,\tau)g(\xi,\tau)$ when both $f$ and $g$ depend on two variables, but it also holds for the quantity $f(\xi)g(\xi,\tau)$ if the function $f$ only depends on $\xi$ (we make the same comment on $g$).

The use of all these quantities will simplify the notations and help us to compactly write numerical schemes where all the contributions between neighbouring cells are easy to identify.

# 3 The ADER-IPDG scheme

## 3.1 The ADER-DG prediction-correction scheme

The ADER approach to build a high-order scheme in time is combined to a DG method in space where the solution is composed of piecewise polynomials per cell at each iteration in time. The ADER technique has been used for the resolution of a wide range of problems, especially in the framework of compressible flows [16–18, 21–23], even with a diffusion term [19, 24, 25].

Initially, the ADER discretization was performed by using a Cauchy-Kowalewski procedure [26, 27], whether it was combined with a FV or a DG scheme in space. In its current form, the ADER-DG prediction-correction scheme is composed of two stages that we describe below in the particular case of a linear unsteady advection-diffusion problem.

***Predictor step***

For any function $\hat{\phi} = \hat{q}, \hat{F}, \hat{S}$ defined on the reference space-time cell $\hat{c}$, let us introduce its polynomial expansions with respect to the space-time Lagrange basis functions (8). By linearity, the expansion of any derivative is

straightforward

$$\hat{\phi}(\xi,\tau) := \sum_{\ell=1}^{L} \hat{\phi}_\ell \Theta_\ell(\xi,\tau), \quad \partial_p \hat{\phi}(\xi,\tau) := \sum_{\ell=1}^{L} \hat{\phi}_\ell \partial_p \Theta_\ell(\xi,\tau), \quad p = \xi,\tau. \tag{14}$$

In the prediction stage, we multiply equation (6) by a test function $\Theta_k(\xi,\tau)$ and integrate by parts in time to determine the 2D local space-time Galerkin predictor $\hat{q}(\xi,\tau)$ starting from a 1D polynomial $\hat{u}^0(\xi,0)$ at fictious time $\tau = 0$ (see for instance [22]). Using expansions (14), this procedure writes

$$\langle \Theta_k, \partial_\tau \hat{q} \rangle + \left(\frac{\Delta t}{\Delta x}\right) \langle \Theta_k, \partial_\xi \hat{F}(\hat{q}) \rangle = \Delta t \langle \Theta_k, \hat{S} \rangle, \tag{15}$$

$$\Leftrightarrow \sum_{\ell=1}^{L} \left( [\Theta_k, \Theta_\ell]_1 - \langle \partial_\tau \Theta_k, \Theta_\ell \rangle \right) \hat{q}_\ell + \left(\frac{\Delta t}{\Delta x}\right) \sum_{\ell=1}^{L} \langle \Theta_k, \partial_\xi \Theta_\ell \rangle \hat{F}_\ell = \sum_{\ell=1}^{K} [\Theta_k, \theta_\ell]_0 \hat{u}_\ell^0 + \Delta t \sum_{\ell=1}^{L} \langle \Theta_k, \Theta_\ell \rangle \hat{S}_\ell, \tag{16}$$

$$\Leftrightarrow \left( \underline{M}^1 - \underline{K}^\tau \right) \hat{\mathbf{q}} + \left(\frac{\Delta t}{\Delta x}\right) (\underline{K}^\xi)^T \hat{\mathbf{F}} = (\underline{N}^0)^T \hat{\mathbf{u}}^0 + \Delta t \underline{M} \hat{\mathbf{S}}, \tag{17}$$

where we reused the notations (11) and $\mathbf{\Phi} := [\phi_1, \ldots, \phi_\star]^T \in \mathbb{R}^\star$ for $\star = K, L$ is the vector containing the Degrees Of Freedom (DOF) of the 1D or 2D function $\phi$ with respect to the corresponding Lagrange basis. In particular, $\hat{\mathbf{u}}^0 \in \mathbb{R}^K$ contains the DOF of the numerical solution on a physical cell $\omega_i$ at time $t^n$.

Introducing the matrices and the vector

$$\underline{A} := \underline{M}^1 - \underline{K}^\tau, \quad \underline{B} := \left(\frac{\Delta t}{\Delta x}\right) (\underline{K}^\xi)^T, \quad \hat{\mathbf{c}} := (\underline{N}^0)^T \hat{\mathbf{u}}^0 + \Delta t \underline{M} \hat{\mathbf{S}}, \tag{18}$$

the prediction step consists in solving the equation

$$\underline{A} \hat{\mathbf{q}} + \underline{B} \hat{\mathbf{F}} = \hat{\mathbf{c}}. \tag{19}$$

When the physical flux is nonlinear, this system can be solved for $\hat{\mathbf{q}}$ with a Picard fixed-point algorithm for example. As the advective and diffusive fluxes given in (5) are linear with respect to the solution or its derivative, one can express the DOF of the total flux $\hat{F}(\hat{q})$ as a linear combination of the DOF of the predictor $\hat{q}$

$$\hat{F} = a\hat{q} - \frac{\lambda}{\Delta x} \partial_\xi \hat{q} \quad \Leftrightarrow \quad \hat{\mathbf{F}} = \left( a\underline{I} - \frac{\lambda}{\Delta x} \underline{G} \right) \hat{\mathbf{q}} =: \underline{H} \hat{\mathbf{q}}, \tag{20}$$

where the gradient matrix is defined by $\underline{G}_{k,\ell} := \partial_\xi \Theta_\ell(\pi_k)$ and $\underline{I} \in \mathcal{M}_L(\mathbb{R})$ stands for the identity matrix. Indeed, we exploit the fact that the physical

flux $\hat{F}$ is linear, so we use the following expansion for its evaluation at the $k$-th space-time quadrature point (i.e. the $k$-th DOF of the vector $\hat{\mathbf{F}}$)

$$\hat{F}(\hat{q}(\pi_k)) = \sum_{\ell=1}^{L} \left( a\Theta_\ell(\pi_k) - \frac{\lambda}{\Delta x}\partial_\xi \Theta_\ell(\pi_k) \right) \hat{q}_\ell, \tag{21}$$

as presented in eq. (14). In particular, we observe that the matrix $\underline{B}$ represents the divergence operator (through matrix $\underline{K}^\xi$, which is computed with quadratures), while the matrix $\underline{G}$ represents the gradient operator (through a pointwise evaluation). This implies that the second order derivatives of the basis functions never need to be computed.

Finally, the space-time Galerkin predictor is completely determined by solving the linear system of size $L$

$$(\underline{A} + \underline{BH})\, \hat{\mathbf{q}} = \hat{\mathbf{c}}. \tag{22}$$

The resolution of this small local linear system is not expensive, especially if one uses a uniform mesh and constant values for $a$ and $\lambda$. Indeed, the matrix $(\underline{A} + \underline{BH})$ does not depend on the space-time cell $c_i^n$ on which the prediction step is solved, so it can be pre-computed once to lower the total computational cost. We also remark that this prediction stage is embarassingly parallel because no communications are needed between the discretization cells.

***Corrector step***

Within the DG framework, the solution is represented by piecewise polynomials of arbitrary degree $(K-1)$ in space, i.e., each cell contains $K$ local DOF relative to $\hat{u}$, defined on the reference space-time cell $\hat{c}$. Let us introduce its polynomial expansion with respect to the space Lagrange basis functions (7)

$$\hat{u}(\xi, \tau) := \sum_{\ell=1}^{K} \hat{u}_\ell(\tau)\theta_\ell(\xi), \tag{23}$$

the DOF $\hat{u}_\ell(\tau)$ being updated at each iteration.

In the correction stage, we multiply equation (6) by a space test function $\hat{\psi} \equiv \theta_k(\xi)$ and integrate by parts in both time and space to couple the DOF between neighbouring cells, and we replace the solution $\hat{u}$ by the predictor $\hat{q}$ in the fluxes. In the following, we denote by $\tilde{\phi}$ the numerical flux of the quantity $\hat{\phi}$ standing for the physical fluxes $\hat{F}(\hat{q})$ or the solution itself $\hat{q}$. This step then reads

$$\begin{aligned} &\langle \hat{\psi}, \partial_\tau \hat{u} \rangle + \left( \frac{\Delta t}{\Delta x} \right) \langle \hat{\psi}, \partial_\xi \hat{F}(\hat{q}) \rangle = \Delta t \langle \hat{\psi}, \hat{S} \rangle, \\ \Leftrightarrow\ &[\hat{\psi}, \hat{u}(\cdot, 1)] = [\hat{\psi}, \hat{u}(\cdot, 0)] + \left( \frac{\Delta t}{\Delta x} \right) \langle \partial_\xi \hat{\psi}, \hat{F}(\hat{q}) \rangle + \Delta t \langle \hat{\psi}, \hat{S} \rangle \end{aligned}$$

$$- \left(\frac{\Delta t}{\Delta x}\right) \left( \{\hat{\psi}, \tilde{F}(\tilde{q})\}_1 - \{\hat{\psi}, \tilde{F}(\tilde{q})\}_0 \right), \qquad (24)$$

where we keep in mind that $\hat{F} = \hat{F}_a + \hat{F}_d$. In the following, the terms $\langle \partial_\xi \hat{\psi}, \hat{F}(\hat{q}) \rangle$ and $\{\hat{\psi}, \tilde{F}(\tilde{q})\}_\xi$ will be refered to as volume and interface terms respectively. The three different possibilities to solve this correction stage in this paper are developed in the next section dedicated to IPDG methods applied to time-dependent problems.

In figure 2, we summarize the ADER procedure. The prediction stage consists in determining a local continuous space-time Galerkin predictor from the knowledge of the initial data, which is a polynomial of high degree in the DG approach. The correction stage recombines the information between neighbouring cells through the computation of numerical fluxes.

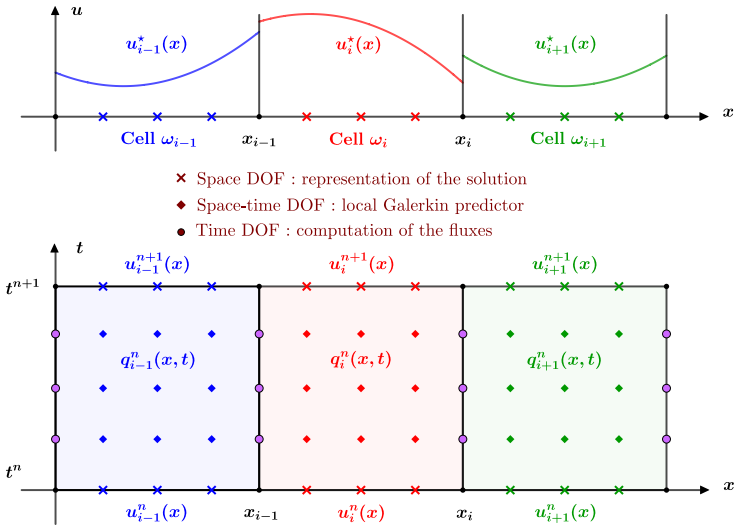


**Fig. 2** Top: representation of the solution by discontinuous piecewise polynomials. Bottom: illustration of the ADER-DG prediction-correction scheme.

The choice of Lagrange polynomials passing through the Gauss-Legendre quadrature points as nodal basis functions allows to reduce the global computational cost. Indeed, all the integrals (10) are computed by directly using the values of the unknowns at the Gauss-Legendre nodes at no additional cost.

## 3.2 A family of IPDG methods

In the previous variational formulation (24), let us break the total flux into its advective and diffusive components and replace them by their expressions in the volume terms. Since the solution may be discontinuous across cell interfaces, there is not a unique definition for the solution and its derivative at a

given interface between two neighbouring cells. Thus, we add the subscript $i$ to the quantities that refer to the cell $\omega_i$ and are uniquely defined. Now, for all interfaces, we introduce the discrete parameter $\varepsilon \in \{-1, 0, +1\}$ and consider the modified variational formulation below

$$\begin{aligned}\left[\hat{\psi}_i, \hat{u}_i(\cdot,1)\right] = \left[\hat{\psi}_i, \hat{u}_i(\cdot,0)\right] + \Delta t \left\langle \hat{\psi}_i, \hat{S}_i \right\rangle \\ + \left(\frac{\Delta t}{\Delta x}\right) \left\langle \partial_\xi \hat{\psi}_i, a\hat{q}_i \right\rangle - \left(\frac{\Delta t}{\Delta x^2}\right) \left\langle \partial_\xi \hat{\psi}_i, \lambda \partial_\xi \hat{q}_i \right\rangle \\ - \left(\frac{\Delta t}{\Delta x}\right) \left( \left\{ \hat{\psi}_i, \tilde{F}_a\left(\tilde{q}\right) \right\}_1 - \left\{ \hat{\psi}_i, \tilde{F}_a\left(\tilde{q}\right) \right\}_0 \right) \\ - \left(\frac{\Delta t}{\Delta x}\right) \left( \left\{ \hat{\psi}_i, \tilde{F}_d\left(\tilde{q}\right) \right\}_1 - \left\{ \hat{\psi}_i, \tilde{F}_d\left(\tilde{q}\right) \right\}_0 \right) \\ + \varepsilon \left(\frac{\Delta t}{\Delta x}\right) \left( \left\{ \tilde{q}, \hat{F}_d\left(\hat{\psi}_i\right) \right\}_1 - \left\{ \tilde{q}, \hat{F}_d\left(\hat{\psi}_i\right) \right\}_0 \right). \end{aligned} \tag{25}$$

Clearly, the case $\varepsilon = 0$ corresponds to the previous equation (24); the case $\varepsilon = +1$ creates an anti-symmetry with respect to $\tilde{q}$ and $\hat{\psi}$ on the diffusive interface terms; finally, the case $\varepsilon = -1$ provides a total symmetry on these diffusive interface terms. The interface terms involving the advective flux are already symmetric by construction. This manipulation leads to the definition of the three different IPDG methods studied in this paper, as shown in table 1.

| Parameter $\varepsilon$ | IPDG method | Abbreviation |
|---|---|---|
| $-1$ | Symmetric Interior Penalty Galerkin | SIPG |
| 0 | Incomplete Interior Penalty Galerkin | IIPG |
| $+1$ | Nonsymmetric Interior Penalty Galerkin | NIPG |

**Table 1** Three IPDG methods depending on the value of the parameter $\varepsilon$.

Many details on these three IPDG methods can be found in [13] for theoretical or implementation aspects, with Euler or RK schemes for the time integration. A presentation of some classical IP techniques can be found in [5, 14]. We draw the reader's attention to the fact that the ADER-IIPG scheme actually corresponds to the classical ADER-DG scheme used in [17, 18, 22] for example.

Independently of the IPDG method used, the quantities $\hat{q}$ and $\hat{F}(\hat{q})$ need to be replaced by a consistent and conservative numerical flux which is detailed in the next section.

### 3.3 Cell average, jump and numerical flux

At the interface $x_i$ separating the two cells $\omega_i$ and $\omega_{i+1}$, we define the limits of any function $\phi$ from the left and the right sides at a given time $t$ as

$$\phi_i^-(t) := \lim_{\delta \to 0^+} \phi(x_i - \delta, t), \quad \phi_i^+(t) := \lim_{\delta \to 0^+} \phi(x_i + \delta, t). \tag{26}$$

Thus the average and the jump of the quantity $\phi$ across an interior interface $x_i$ for $i = 1, \ldots, N_x - 1$ read respectively

$$\langle \phi(t) \rangle_i := \frac{1}{2} \left[ \phi_i^-(t) + \phi_i^+(t) \right], \tag{27}$$

$$[\![ \phi(t) ]\!]_i := \phi_i^-(t) - \phi_i^+(t). \tag{28}$$

The numerical fluxes are generally designed on the same basis which is a linear combination of the average of the physical flux and the jump of the solution, such as the Lax-Friedrichs or the Rusanov numerical fluxes. In this work, we use a Rusanov-like numerical flux both for the advective and the diffusive components at an interface $e := \omega_i \cap \omega_{i\pm1}$.

- For interior interfaces

$$\tilde{F}_a\left(\tilde{q}_e(\tau)\right) := \langle \hat{F}_a\left(\hat{q}(\tau)\right) \rangle_i + \sigma_a [\![ \hat{q}(\tau) ]\!]_i, \tag{29}$$

$$\tilde{F}_d\left(\tilde{q}_e(\tau)\right) := \langle \hat{F}_d\left(\hat{q}(\tau)\right) \rangle_i + \sigma_d [\![ \hat{q}(\tau) ]\!]_i, \tag{30}$$

$$\tilde{q}_e(\tau) := \frac{1}{2} [\![ \hat{q}(\tau) ]\!]_i \vec{n}. \tag{31}$$

- For Dirichlet boundary interfaces

$$\tilde{F}_a\left(\tilde{q}_e(\tau)\right) := \frac{1}{2} \left[ \hat{F}\left(\hat{q}^{\mp}(\tau)\right) + \hat{F}\left(g_{\pm}(\tau)\right) \right] + \sigma_a \left( \hat{q}^{\mp}(\tau) - g_{\pm}(\tau) \right) \vec{n}, \tag{32}$$

$$\tilde{F}_d\left(\tilde{q}_e(\tau)\right) := \hat{F}_d\left(\hat{q}^{\mp}(\tau)\right) + \sigma_d \left( \hat{q}^{\mp}(\tau) - g_{\pm}(\tau) \right) \vec{n}, \tag{33}$$

$$\tilde{q}_e(\tau) := \hat{q}^{\mp}(\tau) - g_{\pm}(\tau). \tag{34}$$

When we consider the cell $\omega_i$, the outward unit vector $\vec{n}$ in 1D comes down to $-1$ (respectively $+1$) on the left (respectively right) interface of the cell. Moreover, recalling that $(K-1)$ is the degree of the polynomials used in the ADER-IPDG prediction-correction scheme, the coefficients $\sigma_a, \sigma_d > 0$ are defined as

$$\sigma_a := \frac{|a|}{2}, \quad \sigma_d := \mu \frac{\lambda}{2\Delta x}, \quad \mu := \gamma (K-1)^2, \tag{35}$$

where $\gamma > 0$ is a penalization parameter for the diffusive flux (see [14] for discussions on the choice for $\mu$). Finally, replacing the numerical fluxes $\tilde{q}$ and $\tilde{F}$ in variational formulation (25) by their expressions (29) and (32) allows to rewrite explicitly the complete scheme according to the different unknowns as well as the boundary conditions.

This choice of numerical fluxes for the advective and diffusive compenents comes with the definition of a suitable CFL condition. A synthesis of the approaches proposed in [19–21, 24] leads us to compute the time step using the following CFL restriction

$$\Delta t := \alpha(k)\frac{\Delta x_\star^2}{2\lambda + |a|\Delta x_\star}, \quad \Delta x_\star := \frac{\Delta x}{2k+1}. \tag{36}$$

Contrary to FV schemes, the time step for DG schemes strongly depends on the degree $k$ of the approximation polynomials. It is possible to apply an additional stabilization through the parameter $\alpha(k)$ which may depend on $k$, as explored in [5] and reported in [26].

## 3.4 Construction of the local linear systems

In the particular case of the advection-diffusion equation, the physical flux is linear with respect to the unknown and its derivative. Consequently, the numerical scheme can be cast into the form of a linear combination of the unknown DOF on each cell of the mesh. Starting from the variational formulation (25), the case of interior or boundary cells needs to be treated separately (see appendix A for the computations). To simplify the final expression of the numerical schemes, we introduce the notations

$$\underline{Z}_\star := \left(\frac{1}{\Delta x}\right)\underline{Z}_\star^a - \left(\frac{1}{\Delta x^2}\right)\underline{Z}_\star^d, \qquad \star = L, C, R, \tag{37}$$

$$\mathbf{b}_\star := \left(\frac{1}{\Delta x}\right)\mathbf{b}_\star^a - \left(\frac{1}{\Delta x^2}\right)\mathbf{b}_\star^d, \qquad \star = L, R, \tag{38}$$

the former quantity standing for matrices and the latter for vectors. The letters $L, C, R$ refer respectively to the left, center and right positions.

*Interior cells.* The ADER-IPDG scheme for interior cells reads

$$\underline{N}\mathbf{u}_i^{n+1} = \underline{N}\mathbf{u}_i^n + \Delta t\left(\underline{Z}_L\mathbf{q}_{i-1}^n + \underline{Z}_C\mathbf{q}_i^n + \underline{Z}_R\mathbf{q}_{i+1}^n\right) + \Delta t\underline{P}\mathbf{S}_i^n, \tag{39}$$

where the advective and diffusive local space-time matrices are defined as

$$\underline{Z}_L^a := a^+\underline{Q}^{0,1}, \tag{40}$$

$$\underline{Z}_C^a := a\underline{C} - a^+\underline{Q}^{1,1} - a^-\underline{Q}^{0,0}, \tag{41}$$

$$\underline{Z}_R^a := a^-\underline{Q}^{1,0}, \tag{42}$$

$$\underline{Z}_L^d := \frac{\lambda}{2}\left[\underline{R}^{0,1} + \varepsilon\underline{W}^{0,1} - \mu\underline{Q}^{0,1}\right], \tag{43}$$

$$\underline{Z}_C^d := \frac{\lambda}{2}\left[2\underline{E} + \underline{R}^{0,0} - \underline{R}^{1,1} - \varepsilon\left(\underline{W}^{0,0} - \underline{W}^{1,1}\right) + \mu\left(\underline{Q}^{0,0} + \underline{Q}^{1,1}\right)\right], \tag{44}$$

$$\underline{Z}_R^d := \frac{\lambda}{2}\left[-\underline{R}^{1,0} - \varepsilon\underline{W}^{1,0} - \mu\underline{Q}^{1,0}\right]. \tag{45}$$

*Left boundary cell.* The ADER-IPDG scheme for the left cell reads

$$\underline{N}\mathbf{u}_1^{n+1} = \underline{N}\mathbf{u}_1^n + \Delta t\left(\underline{Z}_C\mathbf{q}_1^n + \underline{Z}_R\mathbf{q}_2^n\right) + \Delta t\underline{P}\mathbf{S}_1^n + \Delta t\mathbf{b}_L^n, \tag{46}$$

where the local space-time matrices are exactly the same as for interior cells, except for $\underline{Z}_C^d$ in which a coefficient 2 has to be added in front of the matrices $\underline{R}^{0,0}$ and $\underline{W}^{0,0}$. The boundary conditions for the advective and diffusive components are

$$(\mathbf{b}_L^a)^n := a^+\mathbf{d}^0, \quad (\mathbf{b}_L^d)^n := \frac{\lambda}{2}\left(-\mu\mathbf{d}^0 + 2\varepsilon\mathbf{e}^0\right). \tag{47}$$

*Right boundary cell.* The ADER-IPDG scheme for the right cell reads

$$\underline{N}\mathbf{u}_{N_x}^{n+1} = \underline{N}\mathbf{u}_{N_x}^n + \Delta t\left(\underline{Z}_L\mathbf{q}_{N_x-1}^n + \underline{Z}_C\mathbf{q}_{N_x}^n\right) + \Delta t\underline{P}\mathbf{S}_{N_x}^n + \Delta t\mathbf{b}_R^n, \tag{48}$$

where, again, all the local matrices are the same as for interior cells, except for $\underline{Z}_C^d$ in which a coefficient 2 needs to be added in front of the matrices $\underline{R}^{1,1}$ and $\underline{W}^{1,1}$. The advective and diffusive components for the boundary conditions are

$$(\mathbf{b}_R^a)^n := a^-\mathbf{d}^1, \quad (\mathbf{b}_R^d)^n := -\frac{\lambda}{2}\left(\mu\mathbf{d}^1 + 2\varepsilon\mathbf{e}^1\right). \tag{49}$$

The remark on the computational cost that we made on the prediction step also holds for this correction stage, regardless the IPDG method chosen. Indeed, as we consider that $a$ and $\lambda$ are constants and we use a uniform mesh, all the matrices involved are constant and can be pre-computed only once. Moreover, this linear system to be solved for $\mathbf{u}_i^n$ is of small size $K$ and the matrix $\underline{N}$ is diagonal, which is a consequence of our choice of an orthogonal Lagrange polynomial basis. On the other hand, and contrary to FV methods that need a large stencil to reconstruct the information with a high precision, this DG scheme is compact (it is enough to know the DOF of the cell $\omega_i$ and of its two immediate neighbours) and thus the parallelization is simple and efficient.

# 4 Numerical validation of the high-order ADER-IPDG scheme

In this section, we present some numerical results on two different test cases in one dimension. The first one is a classical test: a gaussian that remains far from the boundaries. The second one is a bit more challenging: a smooth step initially located out of the domain and which crosses the boundary. Finally, for the third example, we propose a two dimensional extension.

## 4.1 First example in 1D

In this section, we present a manufactured test case. We take an advection velocity $a = 1$ and a diffusion coefficient $\lambda = 1$ on a computational domain

$\Omega := [0, 3]$. The Dirichlet boundary conditions and the initial condition are imposed by using the values of the exact solution at the boundaries of $\Omega$ or at time $t = 0$ respectively, and the final time is $T_f = 1$. In all our simulations, we take $\gamma = 10$ in the diffusive numerical flux. We enforce the source term $S := \partial_t u + \partial_x F(u)$ corresponding to the exact solution below

$$u(x,t) := u_0(x - at)e^{-\lambda t}, \quad u_0(x) := e^{-30(x-1)^2}. \tag{50}$$

For instance, the results obtained using an ADER-SIPG method on a space mesh composed of $N_x = 30$ cells with space-time $\mathbb{P}_4$ polynomials are presented in figure 3, where the dots represent the values of the numerical solution at each Gauss-Legendre quadrature point on every cell of the mesh.

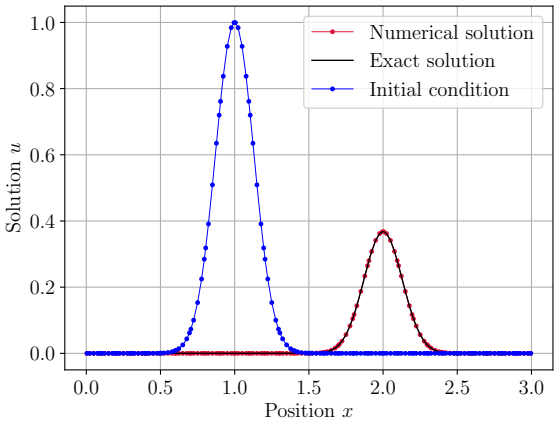


**Fig. 3** Test case 1: numerical and exact solutions obtained by the ADER-SIPG scheme with $\mathbb{P}_4$ polynomials on $N_x = 30$ cells.

To validate numerically the theoretical orders of convergence of the global ADER-IPDG schemes, we need to examine the convergence rates of the three methods presented.

### *Convergence study*

On the same test case based on manufactured solution (50), we now perform a study of the mesh convergence of the SIPG, IIPG and NIPG methods. It is well known that the order of convergence of the DG scheme for pure advection is always $(k + 1)$ for $\mathbb{P}_k$ polynomials. However, we know from [13] that, in the $L^2$ norm, the orders of convergence of the different IPDG methods for the pure diffusion problem are supposed to be

- For IIPG and NIPG: order $(k + 1)$ for odd $k$ and order $k$ for even $k$.
- For SIPG: order $(k + 1)$ no matter if $k$ is odd or even.

As a consequence, for the complete advection-diffusion equation, the resulting order of convergence of a given scheme is the minimum of the orders of convergence for the pure advection or pure diffusion problems, which comes down to the one for pure diffusion as it is the most restrictive. For more details on this theoretical result, the reader may refer to [14].

We thus compare the three IPDG methods by considering a series of meshes composed of $N_x := 16 \times 2^m$ cells for $m \in \{0, \dots, 3\}$, with $\mathbb{P}_k$ polynomials for all $k \in \{1, \dots, 4\}$. The resulting mesh sizes range from $3 \times 16^{-1}$ to $3 \times 128^{-1}$. The numerical orders of convergence are presented on figure 4.

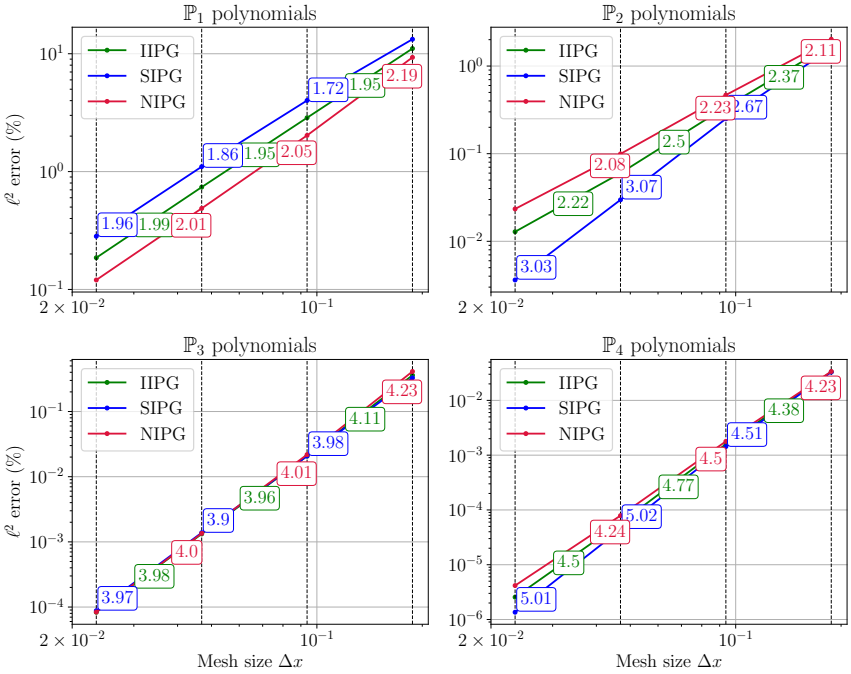


**Fig. 4** Test case 1: orders of convergence of the three ADER-IPDG methods.

From the theory, we expect the order of convergence for an advection-diffusion problem to always be $(k+1)$ for $\mathbb{P}_k$ polynomials when using the SIPG scheme. On the contrary, this order may depend on the parity of $k$ with the IIPG or NIPG methods. In our results, the numerical orders of convergence are 2 and 4 respectively for $\mathbb{P}_1$ and $\mathbb{P}_3$ polynomials for all the methods. For a $\mathbb{P}_2$ representation of the solution, the SIPG method is of order 3 as expected, while the two others are slightly better than order 2. On the other hand, for $\mathbb{P}_4$ polynomials, if the fifth order is reached by the SIPG approach, its advantage over the NIPG and especially the IIPG methods is less evident in this case. The convergence study is thus globally compatible with the theoretical results.

Now, the performances of the different ADER-IPDG methods need to be further analyzed, especially in terms of execution time to reach a certain precision and with a convergence study.

### *Performances*

For $\mathbb{P}_k$ polynomials for $k \in \{1, \dots, 4\}$, we propose to determine the number of cells $N_x$ required so that the $\ell^2$ relative error is less than $5\times10^{-2}\%$ or $5\times10^{-3}\%$ respectively. This study allows to compare the different $\mathbb{P}_k$ approximations regarding the total number $N_{\text{DOF}} := (k+1)N_x$ of DOF on the whole mesh as well as the execution time (on 6 threads) in order to reach a given precision. The results are presented in tables 2 and 3 respectively.

| Poly. | IP method | $N_x$ | $N_{\text{DOF}}$ | Time (s) |
|---|---|---|---|---|
| $\mathbb{P}_1$ | SIPG | 306 | 612 | 144 |
| | IIPG | 247 | 494 | 73 |
| | NIPG | 199 | 398 | 48 |
| idrule | SIPG | 54 | 162 | 11 |
| $\mathbb{P}_2$ | IIPG | 70 | 210 | 24 |
| | NIPG | 89 | 267 | 64 |
| $\mathbb{P}_3$ | SIPG | 26 | 104 | 10 |
| | IIPG | 26 | 104 | 11 |
| | NIPG | 27 | 108 | 13 |
| $\mathbb{P}_4$ | SIPG | 16 | 80 | 11 |
| | IIPG | 16 | 80 | 13 |
| | NIPG | 16 | 80 | 17 |

**Table 2** Test case 1: total number of DOF and execution time of the three IPDG methods to reach a precision of $5 \times 10^{-2}\%$.

| Poly. | IP method | $N_x$ | $N_{\text{DOF}}$ | Time (s) |
|---|---|---|---|---|
| $\mathbb{P}_2$ | SIPG | 116 | 348 | 109 |
| | IIPG | 202 | 606 | 594 |
| | NIPG | 275 | 825 | 1,623 |
| $\mathbb{P}_3$ | SIPG | 47 | 188 | 56 |
| | IIPG | 46 | 184 | 56 |
| | NIPG | 47 | 188 | 71 |
| $\mathbb{P}_4$ | SIPG | 25 | 125 | 51 |
| | IIPG | 26 | 130 | 63 |
| | NIPG | 26 | 130 | 68 |

**Table 3** Test case 1: total number of DOF and execution time of the three IPDG methods to reach a precision of $5 \times 10^{-3}\%$.

A fair remark is that when a smaller relative error is required, the schemes of higher orders seem preferable for their speed and low memory footprint regarding the total number of DOF, which is shown in table 3 for $\mathbb{P}_3$ and especially $\mathbb{P}_4$ polynomials. More precisely, the execution times show that when requiring a relative error of $5 \times 10^{-2}\%$, $\mathbb{P}_1$ schemes are not competitive while higher degree approximations remain comparable. If we ask for an error divided by a factor 10, the same comment can be made on $\mathbb{P}_2$ approximations of the solution. In addition, the SIPG variant appears to be generally more competitive compared to the other IP methods regarding the execution time needed to reach a certain precision. It also always needs less DOF, which is an advantage in terms of memory footprint.

## 4.2 Second example in 1D

Here, we propose another manufactured test case that reproduces the same physical behaviours, with an advective velocity $a = 1$ and a diffusion coefficient $\lambda = 1$. The computational domain is $\Omega := [0, 1]$ and the final time of the simulation is $T_f = 1$. We take $\gamma = 10$ in the diffusive numerical flux. The Dirichlet boundary conditions, the initial condition and the source term $S$ are imposed by using the values of the exact solution below

$$u(x,t) := u_0(x - at)e^{-\lambda t}, \quad u_0(x) := 1 - \tanh\left(10(x + 0.4)\right). \tag{51}$$

The results of the ADER-SIPG scheme with a space mesh composed of $N_x = 20$ and $\mathbb{P}_4$ polynomials are presentd in figure 5, where the dots represent the values of the numerical solution at the Gauss-Legendre quadrature points in every cell.

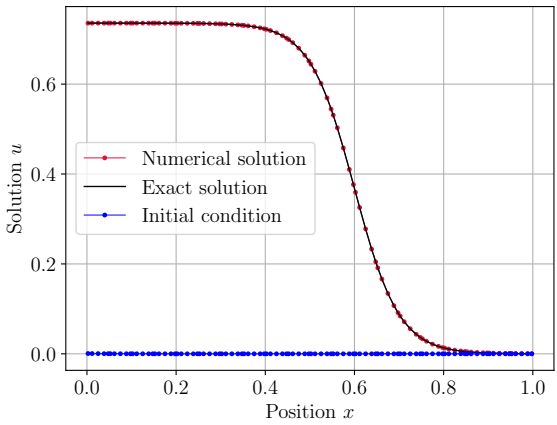


**Fig. 5** Test case 2: numerical and exact solutions obtained by the ADER-SIPG scheme with $\mathbb{P}_4$ polynomials on $N_x = 20$ cells.

A similar convergence study is performed on this test case. The goal is to determine if the boundary conditions (especially on the left boundary of the domain) deteriorate the global order of accuracy of the different ADER-IPDG schemes or if the method remains consistent to the expected precision.

***Convergence study***

On the same test case based on manufactured solution (51), we carry out a convergence study of the SIPG, IIPG and NIPG methods, keeping in mind that, in the $L^2$ norm, the orders of convergence of the different IPDG methods for the complete advection-diffusion problem should be $(k+1)$ for all of them except for IIPG and NIPG with even $k$ for which the order is supposed to be $k$.

In order to compare the three IPDG methods, we consider a series of meshes composed of $N_x := 8 \times 2^m$ cells for $m \in \{0, \dots, 3\}$, with $\mathbb{P}_k$ polynomials for all $k \in \{1, \dots, 4\}$. The resulting mesh sizes now range from $2 \times 16^{-1}$ to $2 \times 128^{-1}$, which is comparable to the sizes of the cells for the first test case as the domain $\Omega$ is three times smaller here. Again, the numerical orders of convergence are shown in Figure 6.

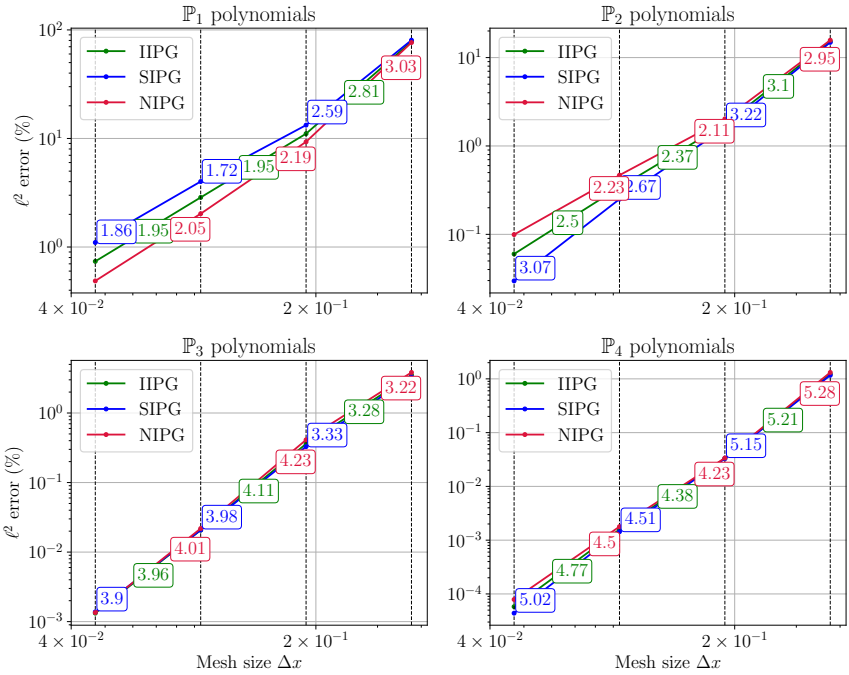


**Fig. 6** Test case 2: orders of convergence of the three ADER-IPDG methods.

The conclusion here is the same as for the first test case: the numerical orders of convergence are coherent with those expected. More precisely, the

SIPG method is always of optimal order $(k+1)$ when the mesh is refined. The IIPG and NIPG methods depend on the parity of $k$, which is well documented (see [13, 24, 28]). Taking finer meshes may allow to better show the advantage of the SIPG method compared to the other two in the case of $\mathbb{P}_4$ polynomials, in relation to their respective orders of convergence.

## 4.3 Third example: 2D case

This section is dedicated to the two dimensional (2D) version of the first manufactured test case. As we use a cartesian mesh, writing both the predictor and the corrector step of the ADER-IPDG scheme is a straightforward extension of the 1D study presented in this paper.

We take the advection velocities $a = b = 1$ with respect to the $x$ and $y$ directions, and a diffusion coefficient $\lambda = 1$ on a computational domain $\Omega := [0,3]^2$. The final time is $T_f = 1$ and we keep the value $\gamma = 10$ in the diffusive numerical flux. The Dirichlet boundary conditions, the initial condition and the source term imposed are those corresponding to the following exact solution

$$u(x,y,t) := u_0(x-at, y-bt)e^{-\lambda t}, \quad u_0(x,y) := e^{-30[(x-1)^2+(y-1)^2]}. \tag{52}$$

A convergence study is carried out on this 2D example in order to show that the 2D extension remains coherent with respect to the theoretical results announced in 1D.

***Convergence study***

On test case (52), we perform a study of the mesh convergence of the same three ADER-IPDG methods. We recall that the order of convergence of the SIPG scheme is expected to be still $(k+1)$ for $\mathbb{P}_k$ polynomials, while it should be $k$ or $(k+1)$ if $k$ is even or odd respectively.

We thus compare the three IPDG variants by considering a series of meshes composed of $N_x = N_y := 32 \times 2^{m-k+1}$ cells for $m \in \{0, \dots, 3\}$, with $\mathbb{P}_k$ polynomials for all $k \in \{1, \dots, 4\}$. This choice of shifting the range of meshes used for higher degree polynomials allows to reduce the computational cost of the study. In this sense, we also fix the duration of the simulation to $T_f = 0.01$. The numerical orders of convergence are shown on figure 7, where we have defined $h = \Delta x = \Delta y$.

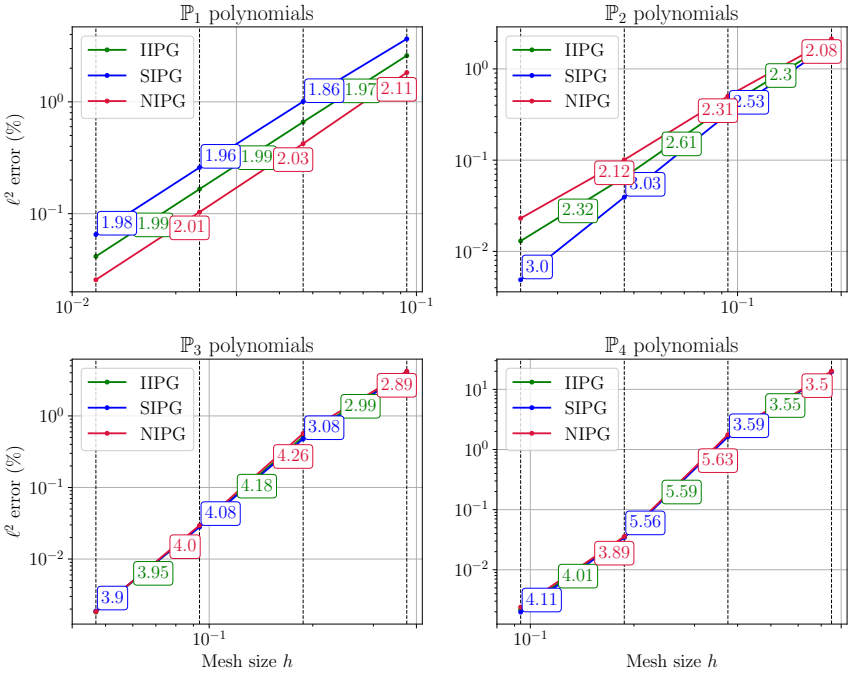


**Fig. 7** Test case 3: orders of convergence of the three ADER-IPDG methods.

For the $\mathbb{P}_1$ and $\mathbb{P}_3$ approximations, the three ADER-IPDG methods provide the expected order of accuracy, that is to say 2 and 4 respectively. The same observation can be made concerning the $\mathbb{P}_2$ schemes: the numerical convergence rates are coherent with the theoretical expectations. In the case of $\mathbb{P}_4$ polynomials, no significant difference between the SIPG method and the other two can be noticed yet. We expect that by using finer meshes, one can recover the results obtained in 1D. Indeed, in this 2D convergence study, the finer mesh is around $h = 10^{-1}$: at this level of refinement, the 1D curves for $\mathbb{P}_4$ approximations did not reach the asymptotic behaviour yet.

As we did for the 1D version of this test case, we now present a study of the performance of the three ADER-IPDG schemes.

### *Performances*

For $\mathbb{P}_k$ polynomials for $k \in \{1, \dots, 4\}$, we want to determine the total number $N^2$ of cells (i.e. $N_x = N_y = N$ cells in each space direction) to reach an $\ell^2$ relative error of 0.1% or 0.05% respectively. We do not ask for the same errors as in the 1D case because of the increased computational cost in 2D. Thus, we propose to compare the different $\mathbb{P}_k$ approximations regarding the total number $N_{\text{DOF}} := (k+1)^2 N^2$ of DOF and the execution time (again, on 6 threads) to reach a prescribed precision. The results are shown in tables 4 and 5 respectively.

| Poly. | IP method | $N^2$ | $N_{\text{DOF}}$ | Time (s) |
|---|---|---|---|---|
| $\mathbb{P}_1$ | SIPG | $207^2$ | 171,396 | 539 |
| | IIPG | $165^2$ | 108,900 | 238 |
| | NIPG | $131^2$ | 68,644 | 100 |
| $\mathbb{P}_2$ | SIPG | $48^2$ | 20,736 | 68 |
| | IIPG | $54^2$ | 26,244 | 107 |
| | NIPG | $65^2$ | 38,025 | 273 |
| $\mathbb{P}_3$ | SIPG | $23^2$ | 8,464 | 76 |
| | IIPG | $24^2$ | 9,216 | 96 |
| | NIPG | $24^2$ | 9,216 | 88 |
| $\mathbb{P}_4$ | SIPG | $16^2$ | 6,400 | 271 |
| | IIPG | $16^2$ | 6,400 | 297 |
| | NIPG | $16^2$ | 6,400 | 326 |

**Table 4** Test case 3: total number of DOF and execution time of the three IPDG methods to reach a precision of 0.1%.

| Poly. | IP method | $N^2$ | $N_{\text{DOF}}$ | Time (s) |
|---|---|---|---|---|
| $\mathbb{P}_2$ | SIPG | $60^2$ | 32,400 | 158 |
| | IIPG | $72^2$ | 46,656 | 372 |
| | NIPG | $89^2$ | 71,289 | 1,026 |
| $\mathbb{P}_3$ | SIPG | $28^2$ | 12,544 | 149 |
| | IIPG | $28^2$ | 12,544 | 170 |
| | NIPG | $29^2$ | 13,456 | 247 |
| $\mathbb{P}_4$ | SIPG | $16^2$ | 6,400 | 271 |
| | IIPG | $16^2$ | 6,400 | 297 |
| | NIPG | $16^2$ | 6,400 | 326 |

**Table 5** Test case 3: total number of DOF and execution time of the three IPDG methods to reach a precision of 0.05%.

From the two tables, the numerical results show that the SIPG method is almost always faster than its two variants to reach a given precision. In this 2D framework, it also has to be noticed that higher order schemes provide a reasonable advantage regarding the memory footprint and the execution time. From table 4, the $\mathbb{P}_4$ schemes may not seem competitive, but table 5 shows that they even allow to reach an error divided by a factor 2 (0.05% instead of 0.1%) without refining the mesh. It is due to the high order of the method that provides a strongly decreasing error when the mesh is refined. Thus, the 2D performance study finally joins the analysis made for the 1D case.

# 5 Conclusion and perspectives

In this paper, we proposed an original association of the IPDG framework with an ADER technique for the time integration to solve the unsteady linear advection-diffusion equation. This ADER-IPDG combination allowed to reach an arbitrary high order of convergence on smooth test cases and showed that higher order schemes are competitive to meet certain requirements regarding both the desired precision and the execution time with a compact stencil. Our numerical studies suggest that the symmetric version, SIPG, is preferable in most of the cases: it is always of optimal order of convergence, and its execution is always faster to reach a desired precision.

In [10] and [11], an ADER-FV scheme was developed with $\mathbb{P}_2$ polynomials for the advection-diffusion equation. As a consequence, the contribution from the diffusive part of the total flux was evaluated by using a relaxation equation to approximate the gradient of the solution. Although this approach granted the global consistency and accuracy of the ADER-FV scheme, the relaxation parameter deteriorated the CFL condition in the limit of mesh refinement. Our different high order ADER-IPDG methods overcome this difficulty at no additional cost.

The results obtained in this paper constitute a first step towards the solution of more challenging problems like the incompressible Navier-Stokes equations. In this sense, the compactness and precision of the IPDG schemes presented combined with the adaptivity of the ADER procedure to the deforming cells of Chimera meshes show promising perspectives.

# Appendix A Additional Definitions

Let us recall that the test functions considered in (25) are the space basis functions $\hat{\psi} \equiv \theta_k(\xi)$. In order to present all the necessary computations in a legible way, we first define the quantities

$$(D_i^\tau)_k := [\theta_k, \hat{u}_i(\cdot,\tau)]\,, \qquad (J_i)_k := \left\langle \theta_k, \hat{S}_i \right\rangle, \tag{A1a}$$

$$(G_i^a)_k := \langle \partial_\xi \theta_k, a\hat{q}_i \rangle\,, \qquad (G_i^d)_k := \langle \partial_\xi \theta_k, \lambda \partial_\xi \hat{q}_i \rangle\,, \tag{A1b}$$

$$(H_i^\xi)_k^a := \left\{ \theta_k, \tilde{F}_a\left(\tilde{q}\right) \right\}_\xi, \qquad (H_i^\xi)_k^d := \left\{ \theta_k, \tilde{F}_d\left(\tilde{q}\right) \right\}_\xi, \tag{A1c}$$

$$(\bar{H}_i^\xi)_k^d := \left\{ \tilde{q}, \hat{F}_d\left(\theta_k\right) \right\}_\xi. \tag{A1d}$$

The ADER-IPDG variational formulation (25) written for all test functions $\theta_k$ for $k \in \{1, \ldots, K\}$ now reads

$$\begin{aligned} \mathbf{D}_i^1 = \mathbf{D}_i^0 &+ \left(\frac{\Delta t}{\Delta x}\right) \mathbf{G}_i^a - \left(\frac{\Delta t}{\Delta x^2}\right) \mathbf{G}_i^d + \Delta t \mathbf{J}_i \\ &- \left(\frac{\Delta t}{\Delta x}\right) \left( (\mathbf{H}_i^1)^a - (\mathbf{H}_i^0)^a \right) \end{aligned}$$

$$- \left(\frac{\Delta t}{\Delta x}\right) \left((\mathbf{H}_i^1)^d - (\mathbf{H}_i^0)^d\right)$$
$$+ \varepsilon \left(\frac{\Delta t}{\Delta x}\right) \left((\bar{\mathbf{H}}_i^1)^d - (\bar{\mathbf{H}}_i^0)^d\right). \tag{A2}$$

The goal now is to express the different terms $\mathbf{D}_i, \mathbf{G}_i, \mathbf{H}_i, \bar{\mathbf{H}}_i$ as functions of the DOF of the Galerkin predictors on cells $\omega_j$ for $j \in \{i-1, i, i+1\}$ denoted by $\mathbf{q}_j$ in order to build the local linear systems to be solved. Let us reuse the polynomial expansions (14)−(23), the numerical fluxes (29)−(32) as well as the matrices (11)−(12)−(13). In the following, the quantities $\mathbf{H}_i$ and $\bar{\mathbf{H}}_i$ will be refered to as *original terms* and *extra terms* respectively.

*The term of DOF.*

$$(D_i^\tau)_k = \sum_{\ell=1}^{K} [\theta_k, \theta_\ell]\, \hat{u}_\ell^i(\tau) \quad \Leftrightarrow \quad \mathbf{D}_i^\tau = \underline{N}\mathbf{u}_i^{n+\tau}. \tag{A3}$$

*The advective volume term.*

$$(G_i^a)_k = a \sum_{\ell=1}^{L} [\partial_\xi \theta_k, \Theta_\ell]\, \hat{q}_\ell^i \quad \Leftrightarrow \quad \mathbf{G}_i^a = a\underline{C}\mathbf{q}_i^n. \tag{A4}$$

*The diffusive volume term.*

$$(G_i^d)_k = \lambda \sum_{\ell=1}^{L} [\partial_\xi \theta_k, \partial_\xi \Theta_\ell]\, \hat{u}_\ell^i \quad \Leftrightarrow \quad \mathbf{G}_i^d = \lambda\underline{E}\mathbf{q}_i^n. \tag{A5}$$

*The source term.*

$$(J_i)_k = \sum_{\ell=1}^{L} [\theta_k, \Theta_\ell]\, \hat{S}_\ell^i \quad \Leftrightarrow \quad \mathbf{J}_i = \underline{P}\mathbf{S}_i^n. \tag{A6}$$

We now introduce the notations $\mathbf{d}_k^\xi := [\theta_k(\xi), g_\pm(\cdot)]$ and $\mathbf{e}_k^\xi := [\partial_\xi \theta_k(\xi), g_\pm(\cdot)]$, where $g_\pm$ is the Dirichlet boundary condition. For any real number $z$, we define its positive and negative parts respectively by $z^+ := \frac{1}{2}(|z| + z) \geq 0$ and $z^- := \frac{1}{2}(|z| - z) \geq 0$. Then, we can compute the interface terms on a given cell.

*The left advective original term.*

- For interior interfaces

$$(H_i^0)_k^a = \left[\theta_k(0), \frac{1}{2}\left(\hat{F}_a\left(\hat{q}_{i-1}(1,\cdot)\right) + \hat{F}_a\left(\hat{q}_i(0,\cdot)\right)\right) + \sigma_a\left(\hat{q}_{i-1}(1,\cdot) - \hat{q}_i(0,\cdot)\right)\right],$$
$$\Leftrightarrow (\mathbf{H}_i^0)^a = \left(a^+\underline{Q}^{0,1}\right)\mathbf{q}_{i-1}^n + \left(-a^-\underline{Q}^{0,0}\right)\mathbf{q}_i^n. \tag{A7}$$

- For a left Dirichlet boundary interface

$$(H_0^0)_k^a = \left[\theta_k(0), \frac{1}{2}\left(\hat{F}_a\left(g_-(\cdot)\right) + \hat{F}_a\left(\hat{q}_1(0,\cdot)\right)\right) + \sigma_a\left(g_-(\cdot) - \hat{q}_1(0,\cdot)\right)\right],$$
$$\Leftrightarrow (\mathbf{H}_0^0)^a = a^+\mathbf{d}^0 + \left(-a^-\underline{Q}^{0,0}\right)\mathbf{q}_1^n. \tag{A8}$$

*The right advective original term.*

- For interior interfaces

$$(H_i^1)_k^a = \left[\theta_k(1), \frac{1}{2}\left(\hat{F}_a\left(\hat{q}_i(1,\cdot)\right) + \hat{F}_a\left(\hat{q}_{i+1}(0,\cdot)\right)\right) + \sigma_a\left(\hat{q}_i(1,\cdot) - \hat{q}_{i+1}(0,\cdot)\right)\right],$$
$$\Leftrightarrow (\mathbf{H}_i^1)^a = \left(a^+\underline{Q}^{1,1}\right)\mathbf{q}_i^n + \left(-a^-\underline{Q}^{1,0}\right)\mathbf{q}_{i+1}^n. \tag{A9}$$

- For a right Dirichlet boundary interface

$$(H_{N_x}^1)_k^a = \left[\theta_k(1), \frac{1}{2}\left(\hat{F}_a\left(\hat{q}_{N_x}(1,\cdot)\right) + \hat{F}_a\left(g_+(\cdot)\right)\right) + \sigma_a\left(\hat{q}_{N_x}(1,\cdot) - g_+(\cdot)\right)\right],$$
$$\Leftrightarrow (\mathbf{H}_{N_x}^1)^a = \left(a^+\underline{Q}^{1,1}\right)\mathbf{q}_{N_x}^n - a^-\mathbf{d}^1. \tag{A10}$$

*The left diffusive original term.*

- For interior interfaces

$$(H_i^0)_k^d = \left[\theta_k(0), \frac{1}{2}\left(\hat{F}_d\left(\hat{q}_{i-1}(1,\cdot)\right) + \hat{F}_d\left(\hat{q}_i(0,\cdot)\right)\right) + \sigma_d\left(\hat{q}_{i-1}(1,\cdot) - \hat{q}_i(0,\cdot)\right)\right],$$
$$\Leftrightarrow (\mathbf{H}_i^0)^d = \frac{\lambda}{2\Delta x}\left[\left(-\underline{R}^{0,1} + \mu\underline{Q}^{0,1}\right)\mathbf{q}_{i-1}^n + \left(-\underline{R}^{0,0} - \mu\underline{Q}^{0,0}\right)\mathbf{q}_i^n\right]. \tag{A11}$$

- For a left Dirichlet boundary interface

$$(H_1^0)_k^d = \left[\theta_k(0), \hat{F}_d\left(\hat{q}_1(0,\cdot)\right) + \sigma_d\left(g_-(\cdot) - \hat{q}_1(0,\cdot)\right)\right],$$
$$\Leftrightarrow (\mathbf{H}_1^0)^d = \frac{\lambda}{2\Delta x}\left[\mu\mathbf{d}^0 + \left(-2\underline{R}^{0,0} - \mu\underline{Q}^{0,0}\right)\mathbf{q}_1^n\right]. \tag{A12}$$

*The right diffusive original term.*

- For interior interfaces

$$(H_i^1)_k^d = \left[\theta_k(1), \frac{1}{2}\left(\hat{F}_d\left(\hat{q}_i(1,\cdot)\right) + \hat{F}_d\left(\hat{q}_{i+1}(0,\cdot)\right)\right) + \sigma_d\left(\hat{q}_i(1,\cdot) - \hat{q}_{i+1}(0,\cdot)\right)\right],$$
$$\Leftrightarrow (\mathbf{H}_i^1)^d = \frac{\lambda}{2\Delta x}\left[\left(-\underline{R}^{1,1} + \mu\underline{Q}^{1,1}\right)\mathbf{q}_i^n + \left(-\underline{R}^{1,0} - \mu\underline{Q}^{1,0}\right)\mathbf{q}_{i+1}^n\right]. \tag{A13}$$

- For a right Dirichlet boundary interface

$$(H_{N_x}^1)_k^d = \left[\theta_k(1), \hat{F}_d\left(\hat{q}_{N_x}(1,\cdot)\right) + \sigma_d\left(\hat{q}_{N_x}(1,\cdot) - g_+(\cdot)\right)\right],$$

$$\Leftrightarrow (\mathbf{H}^1_{N_x})^d = \frac{\lambda}{2\Delta x}\left[\left(-2\underline{R}^{1,1} + \mu\underline{Q}^{1,1}\right)\mathbf{q}^n_{N_x} - \mu\mathbf{d}^1\right]. \tag{A14}$$

*The left diffusive extra term.*

- For interior interfaces

$$\begin{aligned}(\bar{H}^0_i)^d_k &= \left[-\frac{\lambda}{\Delta x}\partial_\xi\theta_k(0), \frac{1}{2}\left(\hat{q}_i(0,\cdot) - \hat{q}_{i-1}(1,\cdot)\right)\right],\\ \Leftrightarrow (\bar{\mathbf{H}}^0_i)^d &= \frac{\lambda}{2\Delta x}\left[\left(\underline{W}^{0,1}\right)\mathbf{q}^n_{i-1} + \left(-\underline{W}^{0,0}\right)\mathbf{q}^n_i\right].\end{aligned} \tag{A15}$$

- For a left Dirichlet boundary interface

$$\begin{aligned}(\bar{H}^0_1)^d_k &= \left[-\frac{\lambda}{\Delta x}\partial_\xi\theta_k(0), \left(\hat{q}_1(0,\cdot) - g_-(\cdot)\right)\right],\\ \Leftrightarrow (\bar{\mathbf{H}}^0_1)^d &= \frac{\lambda}{2\Delta x}\left[2\mathbf{e}^0 + \left(-2\underline{W}^{0,0}\right)\mathbf{q}^n_1\right].\end{aligned} \tag{A16}$$

*The right diffusive extra term.*

- For interior interfaces

$$\begin{aligned}(\bar{H}^1_i)^d_k &= \left[-\frac{\lambda}{\Delta x}\partial_\xi\theta_k(1), \frac{1}{2}\left(\hat{q}_i(1,\cdot) - \hat{q}_{i+1}(0,\cdot)\right)\right],\\ \Leftrightarrow (\bar{\mathbf{H}}^1_i)^d &= \frac{\lambda}{2\Delta x}\left[\left(-\underline{W}^{1,1}\right)\mathbf{q}^n_i + \left(\underline{W}^{1,0}\right)\mathbf{q}^n_{i+1}\right].\end{aligned} \tag{A17}$$

- For a right Dirichlet boundary interface

$$\begin{aligned}(\bar{H}^1_{N_x})^d_k &= \left[-\frac{\lambda}{\Delta x}\partial_\xi\theta_k(1), \left(\hat{q}_{N_x}(1,\cdot) - g^+(\cdot)\right)\right],\\ \Leftrightarrow (\bar{\mathbf{H}}^1_{N_x})^d &= \frac{\lambda}{2\Delta x}\left[\left(-2\underline{W}^{1,1}\right)\mathbf{q}^n_{N_x} + 2\mathbf{e}^1\right].\end{aligned} \tag{A18}$$

Substituting these different quantities that have been expressed in terms of the unknown DOF in each cell provides a complete and fully explicit ADER-IPDG scheme.

# Declarations

On behalf of all authors, the corresponding author states that there is no conflict of interest.

# References

[1] Abgrall, R.: Residual distribution schemes: Current status and future trends. Computers & Fluids **35**(7), 641–669 (2006). https://doi.org/10.1016/j.compfluid.2005.01.007

[2] Ricchiuto, M., Abgrall, R.: Explicit Runge–Kutta residual distribution schemes for time dependent problems: Second order case. Journal of Computational Physics **229**(16), 5653–5691 (2010). https://doi.org/10.1016/j.jcp.2010.04.002

[3] Cockburn, B.: Discontinuous Galerkin methods. ZAMM **83**(11), 731–754 (2003). https://doi.org/10.1002/zamm.200310088

[4] Cockburn, B., Karniadakis, G.E., Shu, C.-W., Griebel, M., Keyes, D.E., Nieminen, R.M., Roose, D., Schlick, T. (eds.): Discontinuous Galerkin Methods: Theory, Computation and Applications. Lecture Notes in Computational Science and Engineering, vol. 11. Springer, Berlin, Heidelberg (2000). https://doi.org/10.1007/978-3-642-59721-3. https://link.springer.com/10.1007/978-3-642-59721-3

[5] Cockburn, B., Shu, C.-W.: Runge–kutta discontinuous galerkin methods for convection-dominated problems. Journal of scientific computing **16**, 173–261 (2001)

[6] Titarev, V.A., Toro, E.F.: Ader: Arbitrary high order godunov approach. Journal of Scientific Computing **17**(1/4), 609–618 (2002). https://doi.org/10.1023/A:1015126814947

[7] Toro, E.F., Montecinos, G.I.: Advection-Diffusion-Reaction Equations: Hyperbolization and High-Order ADER Discretizations. SIAM Journal on Scientific Computing **36**(5), 2423–2457 (2014). https://doi.org/10.1137/130937469

[8] Loubère, R., Dumbser, M., Diot, S.: A New Family of High Order Unstructured MOOD and ADER Finite Volume Schemes for Multidimensional Systems of Hyperbolic Conservation Laws. Communications in Computational Physics **16**(3), 718–763 (2014). https://doi.org/10.4208/cicp.181113.140314a

[9] Boscheri, W., Loubère, R.: High Order Accurate Direct Arbitrary-Lagrangian-Eulerian ADER-MOOD Finite Volume Schemes for Non-Conservative Hyperbolic Systems with Stiff Source Terms. Communications in Computational Physics **21**(1), 271–312 (2017). https://doi.org/10.4208/cicp.OA-2015-0024

[10] Bergmann, M., Carlino, M.G., Iollo, A.: Second Order ADER Scheme for

Unsteady Advection-Diffusion on Moving Overset Grids with a Compact Transmission Condition. SIAM Journal on Scientific Computing **44**(1), 524–553 (2022). https://doi.org/10.1137/21M1393911

[11] Bergmann, M., Carlino, M.G., Iollo, A., Telib, H.: ADER scheme for incompressible Navier-Stokes equations on overset grids with a compact transmission condition. Journal of Computational Physics **467**, 111414 (2022). https://doi.org/10.1016/j.jcp.2022.111414

[12] Bergmann, M., Fondanèche, A., Iollo, A.: An Eulerian finite-volume approach of fluid-structure interaction problems on quadtree meshes. Journal of Computational Physics **471**, 111647 (2022). https://doi.org/10.1016/j.jcp.2022.111647

[13] Rivière, B.: Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation. Frontiers in applied mathematics. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA (2008). OCLC: ocn226292048

[14] Hartmann, R.: Numerical analysis of higher order discontinuous galerkin finite element methods (2008)

[15] Baldassari, C.: Modélisation et simulation numérique pour la migration terrestre par équation d'ondes. Theses, Université de Pau et des Pays de l'Adour (December 2009). https://theses.hal.science/tel-00472810

[16] Fambri, F.: Discontinuous Galerkin Methods for Compressible and Incompressible Flows on Space–Time Adaptive Meshes: Toward a Novel Family of Efficient Numerical Methods for Fluid Dynamics. Archives of Computational Methods in Engineering **27**(1), 199–283 (2020). https://doi.org/10.1007/s11831-018-09308-6

[17] Busto, S., Dumbser, M., Escalante, C., Favrie, N., Gavrilyuk, S.: On High Order ADER Discontinuous Galerkin Schemes for First Order Hyperbolic Reformulations of Nonlinear Dispersive Systems. Journal of Scientific Computing **87**(2), 48 (2021). https://doi.org/10.1007/s10915-021-01429-8

[18] Dumbser, M., Munz, C.-D.: Building Blocks for Arbitrary High Order Discontinuous Galerkin Schemes. Journal of Scientific Computing **27**(1-3), 215–230 (2006). https://doi.org/10.1007/s10915-005-9025-0

[19] Fambri, F., Dumbser, M., Zanotti, O.: Space–time adaptive ADER-DG schemes for dissipative flows: Compressible Navier–Stokes and resistive MHD equations. Computer Physics Communications **220**, 297–318 (2017). https://doi.org/10.1016/j.cpc.2017.08.001

[20] Boscheri, W., Dumbser, M.: Arbitrary-Lagrangian–Eulerian Discontinuous Galerkin schemes with a posteriori subcell finite volume limiting on moving unstructured meshes. Journal of Computational Physics **346**, 449–479 (2017). https://doi.org/10.1016/j.jcp.2017.06.022

[21] Fambri, F., Dumbser, M.: Semi-implicit discontinuous Galerkin methods for the incompressible Navier–Stokes equations on adaptive staggered Cartesian grids. Computer Methods in Applied Mechanics and Engineering **324**, 170–203 (2017). https://doi.org/10.1016/j.cma.2017.06.003

[22] Dumbser, M., Balsara, D.S., Toro, E.F., Munz, C.-D.: A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. Journal of Computational Physics **227**(18), 8209–8253 (2008). https://doi.org/10.1016/j.jcp.2008.05.025

[23] Dumbser, M., Fambri, F., Tavelli, M., Bader, M., Weinzierl, T.: Efficient Implementation of ADER Discontinuous Galerkin Schemes for a Scalable Hyperbolic PDE Engine. Axioms **7**(3), 63 (2018). https://doi.org/10.3390/axioms7030063

[24] Hidalgo, A., Dumbser, M.: ADER Schemes for Nonlinear Systems of Stiff Advection–Diffusion–Reaction Equations. Journal of Scientific Computing **48**(1-3), 173–189 (2011). https://doi.org/10.1007/s10915-010-9426-6

[25] Busto, S., Toro, E.F., Vázquez-Cendón, M.E.: Design and analysis of ADER-type schemes for model advection–diffusion–reaction equations. Journal of Computational Physics **327**, 553–575 (2016). https://doi.org/10.1016/j.jcp.2016.09.043

[26] Gassner, G., Dumbser, M., Hindenlang, F., Munz, C.-D.: Explicit one-step time discretizations for discontinuous Galerkin and finite volume schemes based on local predictors. Journal of Computational Physics **230**(11), 4232–4247 (2011). https://doi.org/10.1016/j.jcp.2010.10.024

[27] Zahran, Y.H.: Central ADER schemes for hyperbolic conservation laws. Journal of Mathematical Analysis and Applications **346**(1), 120–140 (2008). https://doi.org/10.1016/j.jmaa.2008.05.032

[28] Liu, H., Yan, J.: The Direct Discontinuous Galerkin (DDG) Methods for Diffusion Problems. SIAM Journal on Numerical Analysis **47**(1), 675–698 (2009). https://doi.org/10.1137/080720255