Coupling of discontinuous Galerkin and pseudo-spectral methods for time-dependent acoustic problems

Rose-Cloé Meyer, Hadrien Bériot, Gwenael Gabard, Axel Modave

Coupling of discontinuous Galerkin and pseudo-spectral methods for time-dependent acoustic problems*

Rose-Cloé Meyer[1,2], Hadrien Bériot[3], Gwénaël Gabard[4], and Axel Modave[1,⋆]

[1]POEMS, CNRS, Inria, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France
[2]Siemens Industry Software SAS, 107 Avenue de la République, 92320 Châtillon, France
[3]Siemens Industry Software N.V., Interleuvenlaan 68, 3001 Leuven, Belgium
[4]LAUM, UMR 6613, Institut d'Acoustique - Graduate School (IA-GS), CNRS, Le Mans Université, 72000 Le Mans, France
⋆Corresponding author: A. Modave, axel.modave@ensta-paris.fr

Abstract

Many realistic problems in computational acoustics involve complex geometries and sound propagation over large domains, which requires accurate and efficient numerical schemes. It is difficult to meet these requirements with a single numerical method. Pseudo-spectral (PS) methods are very efficient, but are limited to rectangular shaped domains. In contrast, the nodal discontinuous Galerkin (DG) method can be easily applied to complex geometries, but can become expensive for large problems.

In this paper, we study a coupling strategy between the PS and DG methods to efficiently solve time-domain acoustic wave problems. The idea is to combine the strengths of these two methods: the PS method is used on the part of the domain without geometric constraints, while the DG method is used around the PS region to accurately represent the geometry. This combination allows for the rapid and accurate simulations of large-scale acoustic problems with complex geometries, but the coupling and the parameter selection require great care.

The coupling is achieved by introducing an overlap between the PS and DG regions. The solutions are interpolated on the overlaps, which allows the use of unstructured finite element meshes. A standard explicit Runge–Kutta time-stepping scheme is used with the DG scheme, while implicit schemes can be used with the PS scheme due to the peculiar structure of this scheme. We present one- and two-dimensional results to validate the coupling technique. To guide future implementations of this method, we extensively study the influence of different numerical parameters on the accuracy of the schemes and the coupling strategy.

1 Introduction

The present paper is concerned with the simulations in the time domain of large-scale acoustic problems with complex geometries. The numerical methods must be carefully chosen to minimize the numerical dispersion error, in order to obtain accurate results when the waves propagate over long distances. This must be achieved while using an accurate representation of the complex geometric features of the computational domain. Addressing these different challenges with a single numerical scheme is difficult. Therefore, in this paper, we develop a coupling between

the pseudo-spectral (PS) method and the discontinuous Galerkin (DG) method to exploit their respective strengths.

On the one hand, the nodal Discontinuous Galerkin (DG) method yields high-order numerical approximations in time and space, and explicit time integration models [16]. It can easily handle complex geometries by using unstructured finite-element meshes. In addition, much of the computation in the DG method is easily parallelized, especially on GPUs [11, 24, 32, 33]. It has been applied to various wave propagation problems, including room acoustics [52], aeroacoustics [1, 47] and seismic waves [26, 46]. However, the DG method remains computationally expensive for large-scale problems, in terms of runtime and number of degrees of freedom, even when using optimized formulations such as the quadrature-free approach [30] or a matrix-free method [25].

On the other hand, the spectral methods have been extensively studied for solving time-domain problems [6, 17, 45], including analysis of the convergence and stability of the methods. In particular, the pseudo-spectral (PS) method uses the Fourier transform to efficiently compute the spatial derivatives of the numerical solution. The PS method has been used for wave propagation simulations: for the solution of Maxwell's equations [22, 28], or for acoustic wave propagation [18, 21]. The PS method has the advantage of requiring a very small number of points per wavelength to provide accurate results. In addition, the orthogonality of the basis leads to a diagonal system, which significantly reduces the computational cost. This method also benefits from efficient implementations of discrete Fourier transforms, including on GPUs [14, 20].

However, a limitation of the PS method is that the solution should be periodic, otherwise the Gibbs phenomenon can occur, see [2, 17, 49]. One way to overcome this limitation is to use a different basis, such as Chebyshev polynomials, see [2, 23, 40, 44]. But in this case, the basis is not orthogonal and the computational cost is higher. Another strategy is the domain extension or data flipping, see [37–39]. The Fourier continuation is another approach that involves a polynomial interpolation of the solution near the boundaries of the domain using, for example, a Gram polynomial basis [3, 4]. However, this method can also break the orthogonality of the basis functions. Finally, it is possible to use filters in physical space or Fourier space to recover the convergence of the solution [12, 13, 51]. In our case, we chose to use Gaussian windowing, which enforces the periodicity of the PS solution by setting the solution to zero near the boundaries, see for example [19, 36].

Another problem with the PS method is that the domain should be rectangular to take advantage of the base orthogonality. This severely limits the application of this method to complex geometries and requires coupling with another scheme to represent the geometric features.

Various strategies for coupling spectral and other discretization schemes have been studied. For example, a hybrid method has been developed by coupling finite-difference schemes with a spectral method [31, 41, 42]. This involves a decomposition of the domain into rectangular regions in which the solution is represented by a basis of cosine functions (i.e. modes of an acoustic cavity). The results show a significant reduction in computational cost. For instance, in [31], the runtime was reduced by three orders of magnitude compared to FDTD solvers for a realistic room acoustic case. However, the use of a finite difference grid can be too restrictive for the geometry, thus introducing an additional numerical error. Another idea is to couple the DG and PS methods. This has been done for the linearized Euler equations [10, 35, 36] with the requirement of a conformal overlap between the two methods. It also includes a Gaussian window and a low-pass filter to reduce numerical instabilities.

In this paper, we study the coupling of the DG method with high-order nodal polynomial basis functions and a PS method with trigonometric basis functions. Due to the periodicity of

the basis functions in the PS scheme, we use a windowing technique, which acts as a PML, to bring the solution to zero at the boundaries of the PS domain. More details on the various aspects of the proposed coupling method are given in the following sections. In contrast to [10, 35, 36], a novelty of the present approach is to allow unstructured meshes in the overlap, which greatly simplifies the coupling with unstructured meshes for the DG method. An additional contribution of the paper is a detailed analysis of the numerical parameters for the coupling is presented to help in the selection and understanding of the influence of these parameters. Finally, an implicit time-stepping scheme is also introduced for the PS method to improve the stability limit of the numerical model. Due to the orthogonality of the basis, there is no additional computational cost compared to explicit time schemes.

The remainder of the paper is organized as follows. The DG method, the spectral method and the coupling strategy are described in Sections 2, 3 and 4, respectively. The performance of the methods and the influence of the different parameters on the accuracy of the solution are studied in detail in one dimension in Section 5. Finally, in Section 6, we validate the coupling method in two dimensions and present an example of application.

2 Discontinuous Galerkin (DG) method

To introduce the method, we consider a time-domain acoustic wave problem defined on a bounded domain $\Omega \subset \mathbb{R}^{\mathsf{d}}$, with the spatial dimension $\mathsf{d} = 1$, 2 or 3. The problem is written with the pressure-velocity system defined on Ω,

$$\begin{cases} \dfrac{\partial p}{\partial t} + \rho c^2 \, \nabla \cdot \mathbf{v} = 0, \\ \rho \, \dfrac{\partial \mathbf{v}}{\partial t} + \nabla p = 0, \end{cases} \qquad \text{for } \mathbf{x} \in \Omega \text{ and } t > 0, \tag{2.1}$$

and an absorbing boundary condition prescribed on the boundary $\partial\Omega$,

$$p - \rho c \, \mathbf{n} \cdot \mathbf{v} = 0, \qquad \text{for } \mathbf{x} \in \partial\Omega \text{ and } t > 0, \tag{2.2}$$

where $p(\mathbf{x}, t)$ is the pressure field, $\mathbf{v}(\mathbf{x}, t)$ is the particle velocity field, $\mathbf{n}(\mathbf{x})$ is the outward unit normal on $\partial\Omega$, ρ is the density, and c is the sound speed. To facilitate the presentation, we assume that ρ and c are constant. The problem is completed with initial conditions: $p(\mathbf{x}, 0) = p_0(\mathbf{x})$ and $\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x})$.

The computational domain is represented by a conformal mesh made of segments, triangles or tetrahedra, depending on the spatial dimension d. In the following, an element is denoted by K and a face of an element is denoted by F.

In each element K, the fields are approximated by polynomials of maximum degree p. In the DG method, the numerical fields can be discontinuous at the interface between the elements. They

satisfy, for each element K, the following variational formulation, see e.g. [16]:

$$\left|\begin{array}{l}\text{Find } p^K \in \mathbb{P}^{\mathsf{p}}(K) \text{ and } \mathbf{v}^K \in [\mathbb{P}^{\mathsf{p}}(K)]^{\mathsf{d}} \text{ such that} \\ \quad \displaystyle\int_K \frac{\partial p^K}{\partial t} q^K \,\mathrm{d}\mathbf{x} + \rho c^2 \int_K (\nabla \cdot \mathbf{v}^K)\, q^K \,\mathrm{d}\mathbf{x} + \rho c^2 \int_{\partial K} \mathbf{n} \cdot (\widehat{\mathbf{v}} - \mathbf{v}^K)\, q^K \,\mathrm{d}\mathbf{x} = 0, \\ \quad \rho \displaystyle\int_K \frac{\partial \mathbf{v}^K}{\partial t} \cdot \mathbf{w}^K \,\mathrm{d}\mathbf{x} + \int_K (\nabla p^K) \cdot \mathbf{w}^K \,\mathrm{d}\mathbf{x} + \int_{\partial K} (\widehat{p} - p^K)\, \mathbf{n} \cdot \mathbf{w}^K \,\mathrm{d}\mathbf{x} = 0, \\ \text{for all } q^K \in \mathbb{P}^{\mathsf{p}}(K) \text{ and } \mathbf{w}^K \in [\mathbb{P}^{\mathsf{p}}(K)]^{\mathsf{d}},\end{array}\right. \tag{2.3}$$

where $\mathbf{n} \cdot \widehat{\mathbf{v}}$ and $\widehat{p}$ are the numerical fluxes. The fluxes weakly enforce the continuity of the fields at the interfaces between the elements, and they impose the boundary condition on the boundary of the domain Ω. Here, we consider upwind fluxes, which are widely used for wave propagation problems, see e.g. [16, 27, 32]. On each face F of each element K, we write

$$\mathbf{n} \cdot (\widehat{\mathbf{v}} - \mathbf{v}^K) := \begin{cases} (\rho c\, \mathbf{n} \cdot [\![\mathbf{v}]\!] - [\![p]\!])/(2\rho c) & \text{if } F \notin \partial\Omega, \\ (p^K - \rho c\, \mathbf{n} \cdot \mathbf{v}^K)/(2\rho c) & \text{if } F \in \partial\Omega, \end{cases}$$

$$(\widehat{p} - p^K) := \begin{cases} ([\![p]\!] - \rho c\, \mathbf{n} \cdot [\![\mathbf{v}]\!])/2 & \text{if } F \notin \partial\Omega, \\ -(p^K - \rho c\, \mathbf{n} \cdot \mathbf{v}^K)/2 & \text{if } F \in \partial\Omega, \end{cases}$$

where $[\![p]\!] := p^{K'} - p^K$ and $[\![\mathbf{v}]\!] := \mathbf{v}^{K'} - \mathbf{v}^K$ are the jumps at the interface between K and the neighboring element K' sharing the face F.

We consider the DG method using nodal basis functions, see e.g. [15, 16]. On each element K, the numerical fields can be written as

$$p^K(t, \mathbf{x}) = \sum_{i=1}^{N_n} p_i^K(t)\, \ell_i^K(\mathbf{x}), \tag{2.4}$$

$$\mathbf{v}^K(t, \mathbf{x}) = \sum_{i=1}^{N_n} \mathbf{v}_i^K(t)\, \ell_i^K(\mathbf{x}), \tag{2.5}$$

where $\{\ell_i^K(\mathbf{x})\}_i$ are the Lagrange polynomials associated to N_n nodes defined on the element K, and $\{p_i^K(t)\}_i$ and $\{v_i^K(t)\}_i$ are the values of the fields at these nodes. In practice, the polynomials and the nodes are defined on a reference element, and a mapping between the reference element and the physical element K is used. The number of nodes per element, i.e. N_n, depends on the polynomial order p and the dimension d. The position of the reference nodes are defined by using the optimization procedure described in [53].

The semi-discrete scheme is obtained by using equations (2.4)–(2.5) in the variational formulation (2.3), and by using the Lagrange polynomials as test functions. In this work, the time stepping of the DG solution is performed with a low-storage explicit 4^{th} order Runge–Kutta (ERK4) scheme [7, 16]. Since this scheme is conditionally stable, the time step Δt must verify a Courant-Friedrichs-Lewy (CFL) stability condition, see e.g. [9, 16, 48].

We refer e.g. to [16, 24, 32, 33] for a generalization of the DG scheme to more than one dimension, and for efficient parallel implementations.

3 Pseudo-spectral (PS) method

In this section, we describe a PS method (Section 3.1) combined with implicit Runge–Kutta time-stepping schemes (Section 3.2) for solving the pressure-velocity system. The PS method is *a priori* only suitable for periodic solutions. To overcome this problem, we use a windowing strategy (Section 3.3), which can be interpreted as the use of perfectly matched layers.

3.1 Pseudo-spectral discretization in space

In the PS methods, the numerical solution is represented by a sum of functions. While complex exponential Fourier modes are frequently considered, we use real sine and cosine Fourier modes, because the solutions of the considered problems are real. Using real numbers instead of complex numbers reduces memory usage by a factor two. Thanks to this *modal representation* of the solution, the discrete problem can be written as a system of uncoupled equations associated with the modes. The numerical solution can also be rewritten with a *nodal representation* by using the Inverse Discrete Cosine Transform (IDCT) and the Inverse Discrete Sine Transform (IDST) in space, see e.g. [43]. The discrete unknowns then correspond to the values of the fields at the nodes of a regular spatial grid of the domain.

In the proposed approach, cosine modes are used for the pressure field. For the Cartesian components of the velocity field, the modes are obtained by differentiating the cosine functions, resulting in sine modes with multiplicative coefficients. In the following, we derive the semidiscrete schemes in one and two dimensions.

Spatial discretization in one dimension We consider the acoustic wave system (2.1) on the domain $\Omega = [0, L]$. The approximations of the unknown fields are written as

$$p(x,t) = \sum_{i=0}^{N_m-1} p_i(t)\,\varphi_i(x) \quad \text{and} \quad v(x,t) = \sum_{i=0}^{N_m-1} v_i(t)\,\partial_x\varphi_i(x), \tag{3.1}$$

with the *modal coefficients* $p_i(t)$ and $v_i(t)$ and the modes $\varphi_i(x) = \cos(i\pi x/L)$ and $\partial_x\varphi_i(x) = -(i\pi/L)\sin(i\pi x/L)$, where N_m is the number of modes. These functions strongly enforce the boundary condition $v(x,t) = 0$ at the boundaries of the domain.

By injecting the numerical fields (3.1) into equation (2.1), and by using orthogonality properties of the modes, we obtain the system of ordinary differential equations

$$\begin{cases} \dfrac{\mathrm{d}p_i}{\mathrm{d}t} - \rho\omega_i^2 v_i = 0, & \text{for } i = 0 \dots N_m - 1, \\ \rho\dfrac{\mathrm{d}v_i}{\mathrm{d}t} + p_i = 0, & \text{for } i = 1 \dots N_m - 1, \\ \dfrac{\mathrm{d}v_0}{\mathrm{d}t} = 0. \end{cases} \tag{3.2}$$

with $\omega_i = c\pi i/L$. The key property is that the discrete unknowns corresponding to different modes are decoupled. For each mode i, the unknown vector

$$\mathbf{y}_i = \begin{pmatrix} p_i \\ v_i \end{pmatrix}$$

is governed by the system

$$\frac{\mathrm{d}\mathbf{y}_i}{\mathrm{d}t} = \mathbf{F}_i \mathbf{y}_i, \quad \text{with } \mathbf{F}_i = \begin{cases} \begin{pmatrix} 0 & \rho\omega_i^2 \\ -\rho^{-1} & 0 \end{pmatrix}, & \text{if } i \neq 0, \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, & \text{if } i = 0. \end{cases} \tag{3.3}$$

Interestingly, this semi-discrete scheme can also be obtained by using the semi-discrete fields (3.1) in the DG formulation (2.3) with the cosine/sine modes as test functions. The boundary terms of the DG formulation are cancelled by applying the appropriate boundary conditions.

In the pseudo-spectral methods, the differentiation terms are frequently computed with Fourier and inverse Fourier transforms, see e.g. [28]. The unknown fields can be written as

$$p(x,t) = \sum_{i=0}^{N_m-1} \tilde{p}_i(t)\, e^{\jmath i\pi x/L} \quad \text{and} \quad v(x,t) = \sum_{i=0}^{N_m-1} \jmath(i\pi/L)\, \tilde{v}_i(t)\, e^{\jmath i\pi x/L}$$

with the imaginary unit $\jmath = \sqrt{-1}$ and the Fourier coefficients $\tilde{p}_i(t)$ and $\tilde{v}_i(t)$. Using these discrete fields in the pressure-velocity system (2.1) and taking advantage of orthogonality properties, we also obtain the uncoupled systems (3.2) with the unknowns $\tilde{p}_i(t)$ and $\tilde{v}_i(t)$ instead of $p_i(t)$ and $v_i(t)$.

Spatial discretization in two dimensions. We consider now the rectangular domain $[0, L_x] \times [0, L_y]$. The unknowns p, v_x and v_y are approximated with trigonometric modes:

$$\begin{aligned}
p(\mathbf{x}) &= \sum_{i_x=0}^{N_x-1} \sum_{i_y=0}^{N_y-1} p_{i_x,i_y} \varphi_{i_x,i_y}(\mathbf{x}), & \varphi_{i_x,i_y}(\mathbf{x}) &= \cos\left(\frac{i_x\pi x}{L_x}\right) \cos\left(\frac{i_y\pi y}{L_y}\right), \\
v_x(\mathbf{x}) &= \sum_{i_x=0}^{N_x-1} \sum_{i_y=0}^{N_y-1} v_{x,i_x,i_y} \partial_x\varphi_{i_x,i_y}(\mathbf{x}), & \partial_x\varphi_{i_x,i_y}(\mathbf{x}) &= -\frac{i_x\pi}{L_x} \sin\left(\frac{i_x\pi x}{L_x}\right) \cos\left(\frac{i_y\pi y}{L_y}\right), \\
v_y(\mathbf{x}) &= \sum_{i_x=0}^{N_x-1} \sum_{i_y=0}^{N_y-1} v_{y,i_x,i_y} \partial_y\varphi_{i_x,i_y}(\mathbf{x}), & \partial_y\varphi_{i_x,i_y}(\mathbf{x}) &= -\frac{i_y\pi}{L_y} \cos\left(\frac{i_x\pi x}{L_x}\right) \sin\left(\frac{i_y\pi y}{L_y}\right).
\end{aligned}$$

Using these discrete fields in the acoustic wave system, and taking advantage of the orthogonality of the modes, we obtain the system of ordinary differential equations

$$\begin{cases}
\dfrac{\mathrm{d}p_{i_x,i_y}}{\mathrm{d}t} - \omega_{i_x}^2 v_{x,i_x,i_y} - \omega_{i_y}^2 v_{y,i_x,i_y} = 0, & \text{for } i_x = 0 \ldots N_x - 1, i_y = 0 \ldots N_y - 1, \\
\rho \dfrac{\mathrm{d}v_{x,i_x,i_y}}{\mathrm{d}t} + p_{i_x,i_y} = 0, & \text{for } i_x = 1 \ldots N_x - 1, i_y = 0 \ldots N_y - 1, \\
\dfrac{\mathrm{d}v_{x,0,i_y}}{\mathrm{d}t} = 0, & \text{for } i_y = 1 \ldots N_y - 1, \\
\rho \dfrac{\mathrm{d}v_{y,i_x,i_y}}{\mathrm{d}t} + p_{i_x,i_y} = 0, & \text{for } i_x = 0 \ldots N_x - 1, i_y = 1 \ldots N_y - 1, \\
\dfrac{\mathrm{d}v_{y,i_x,0}}{\mathrm{d}t} = 0, & \text{for } i_x = 1 \ldots N_x - 1,
\end{cases}$$

with the coefficients $\omega_{i_x} = c\pi i_x/L_x$ and $\omega_{i_y} = c\pi i_y/L_y$. For each mode (i_x, i_y), the equations can be rewritten in matrix form by introducing the unknown vector

$$\mathbf{y}_{i_x,i_y} = \begin{pmatrix} p_{i_x,i_y} \\ v_{x,i_x,i_y} \\ v_{y,i_x,i_y} \end{pmatrix}.$$

The unknown vector is governed by the system

$$\frac{\mathrm{d}\mathbf{y}_{i_x,i_y}}{\mathrm{d}t} = \mathbf{F}_{i_x,i_y}\mathbf{y}_{i_x,i_y}, \quad \text{with } \mathbf{F}_{i_x,i_y} = \begin{cases} \begin{pmatrix} 0 & \rho\omega_{i_x}^2 & \rho\omega_{i_y}^2 \\ -\rho^{-1} & 0 & 0 \\ -\rho^{-1} & 0 & 0 \end{pmatrix}, & \text{if } i_x \neq 0 \text{ and } i_y \neq 0, \\ \begin{pmatrix} 0 & 0 & \rho\omega_{i_y}^2 \\ 0 & 0 & 0 \\ -\rho^{-1} & 0 & 0 \end{pmatrix}, & \text{if } i_x = 0 \text{ and } i_y \neq 0, \\ \begin{pmatrix} 0 & \rho\omega_{i_x}^2 & 0 \\ -\rho^{-1} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & \text{if } i_x \neq 0 \text{ and } i_y = 0, \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & \text{if } i_x = 0 \text{ and } i_y = 0. \end{cases} \tag{3.4}$$

3.2 Runge–Kutta time stepping schemes

We use the specific form of the PS system to derive efficient implementations of implicit Runge–Kutta (IRK) time-stepping schemes. Implicit schemes are interesting because they can be unconditionally stable, i.e. there is no constraint on the time step, but they usually require solving a linear system at each time step. Fortunately, the PS system is formed of many small uncoupled systems of size $\mathsf{d}+1$ (where d is the spatial dimension). The matrices of these systems (denoted $\mathbf{F}_i$ and $\mathbf{F}_{i_x,i_y}$ above) can be easily inverted analytically.

We have derived explicit expressions of the update schemes by applying the standard IRK methods to the semi-discrete systems (3.3) and (3.4). There is no intermediate steps, and then no additional storage for intermediate solutions that are generally required with these multi-level schemes. For the one-dimensional case (equation (3.3)), we obtain the following update schemes for the solutions corresponding to mode $i \neq 0$ with several classical IRK methods:

- The implicit Euler method, which is a 1st-order implicit Runge–Kutta (IRK1) method, gives:

$$\begin{cases} p_i^{n+1} = \dfrac{1}{1+C_i}\left(p_i^n + \Delta t\, \rho\, \omega_i^2 v_i^n\right), \\ v_i^{n+1} = \dfrac{1}{1+C_i}\left(v_i^n - \dfrac{\Delta t}{\rho} p_i^n\right), \end{cases}$$

  with $C_i = \Delta t^2 \omega_i^2$.

- The trapezoidal rule, which is a 2^{nd}-order implicit Runge–Kutta (IRK2) method, gives:

$$\begin{cases} p_i^{n+1} = \dfrac{1}{1+C_i}\left((1-C_i)\,p_i^n + \Delta t\,\rho\,\omega_i^2 v_i^n\right), \\ v_i^{n+1} = \dfrac{1}{1+C_i}\left((1-C_i)\,v_i^n - \dfrac{\Delta t}{\rho} p_i^n\right), \end{cases}$$

  with $C_i = \Delta t^2 \omega_i^2/4$.

- The Gauss-Legendre method [5, 34], which is a 4^{th}-order implicit Runge–Kutta (IRK4) method, gives:

$$\begin{cases} p_i^{n+1} = \dfrac{1}{(1+C_i+C_i^2)}\left(\left(1-5C_i+C_i^2\right)p_i^n + \Delta t\,\rho\,\omega_i^2\,(1-C_i)\,v_i^n\right), \\ v_i^{n+1} = \dfrac{1}{(1+C_i+C_i^2)}\left(\left(1-5C_i+C_i^2\right)v_i^n - \dfrac{\Delta t}{\rho}\,(1-C_i)\,p_i^n\right), \end{cases}$$

  with $C_i = \Delta t^2 \omega_i^2/12$.

Details of the derivation of these schemes are given in Appendix A.

Similar update formulas can be obtained for the two- and three-dimensional cases and for the explicit Runge–Kutta (ERK) methods. By comparing the number of floating-point operations per time step per mode, we have observed that the computational cost of the IRK and ERK update formulas are very close.

3.3 Windowing

The PS method described above is *a priori* only suitable for problems with periodic solutions, since the trigonometric basis functions are periodic. Without special treatment, non-periodic solutions are degraded by the Gibbs phenomenon.

Several approaches can be considered to overcome the Gibbs phenomenon. Chebyshev polynomial basis functions [2, 23, 40, 44] can be used instead of the trigonometric functions, or an approach similar to the Fourier continuation method [3, 4, 50] can be implemented to represent non-periodic solutions. Unfortunately, with these approaches, the solutions corresponding to different modes are generally coupled, then the problem cannot be solved mode by mode, and we lose a crucial feature required for fast solution procedures. Strategies based on domain extension and data flipping can be explored, see e.g. [37–39], but this requires the use of a much larger computational domain. Filters have also been used to improve the solution, see e.g. [12, 13, 51].

In this work, we consider a windowing strategy. At each time step, the discrete fields are computed at the nodes of a regular spatial grid (this is the *nodal representation* of the solution)

Algorithm 1 Solution procedure with the PS method and the windowing strategy

Set the nodal values of the initial solution: p_{nod} and v_{nod}.
for $t = 0, \ldots, t_{\max}$ (time loop) **do**
  Compute the modal coefficients: $p_{\text{mod}} = \text{DCT}(p_{\text{nod}})$ and $v_{\text{mod}} = \text{DST}(v_{\text{nod}})$.
  Update the modal coefficients of the PS solution by using the time-stepping scheme.
  Compute the nodal values: $p_{\text{nod}} = \text{IDCT}(p_{\text{mod}})$ and $v_{\text{nod}} = \text{IDST}(v_{\text{mod}})$.
  Multiply the nodal values by the window function.
end for

by using inverse discrete cosine/sine transforms (i.e. IDCT and IDST), and the nodal values are multiplied by a Gaussian window function that attenuates the waves near the domain boundaries [19]. The solution is equal to zero at the boundaries, and the solution is periodic. This can be interpreted as a perfectly matched layer (PML) technique [29]. Finally, the *modal representation* of the solution (equation (3.1)) is recovered by using discrete cosine/sine transforms (i.e. DCT and DST). The complete solution procedure is summarized in Algorithm 1.

In the one-dimensional case, the regular spatial grid corresponds to N_m evenly spaced points. We consider the following window function

$$W_l = \begin{cases} e^{-\alpha_{\text{win}} \ln(10) \left(\frac{l - N_{\text{win}}}{N_{\text{win}}} \right)^{2\beta_{\text{win}}}}, & \text{for } l = 1, \ldots, N_{\text{win}}, \\ 1, & \text{for } l = N_{\text{win}} + 1, \ldots, N_m - N_{\text{win}}, \\ e^{-\alpha_{\text{win}} \ln(10) \left(\frac{l - (N_m - N_{\text{win}})}{N_{\text{win}}} \right)^{2\beta_{\text{win}}}}, & \text{for } l = N_m - N_{\text{win}} + 1, \ldots, N_m. \end{cases}$$

where N_{win} is the number of points in each windowed part of the function, and α_{win} and β_{win} are

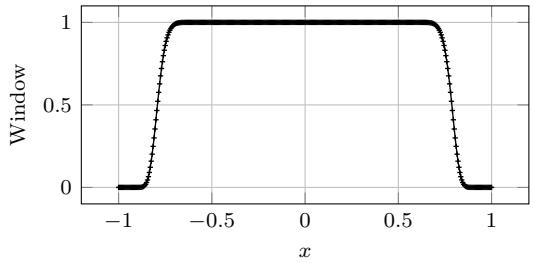
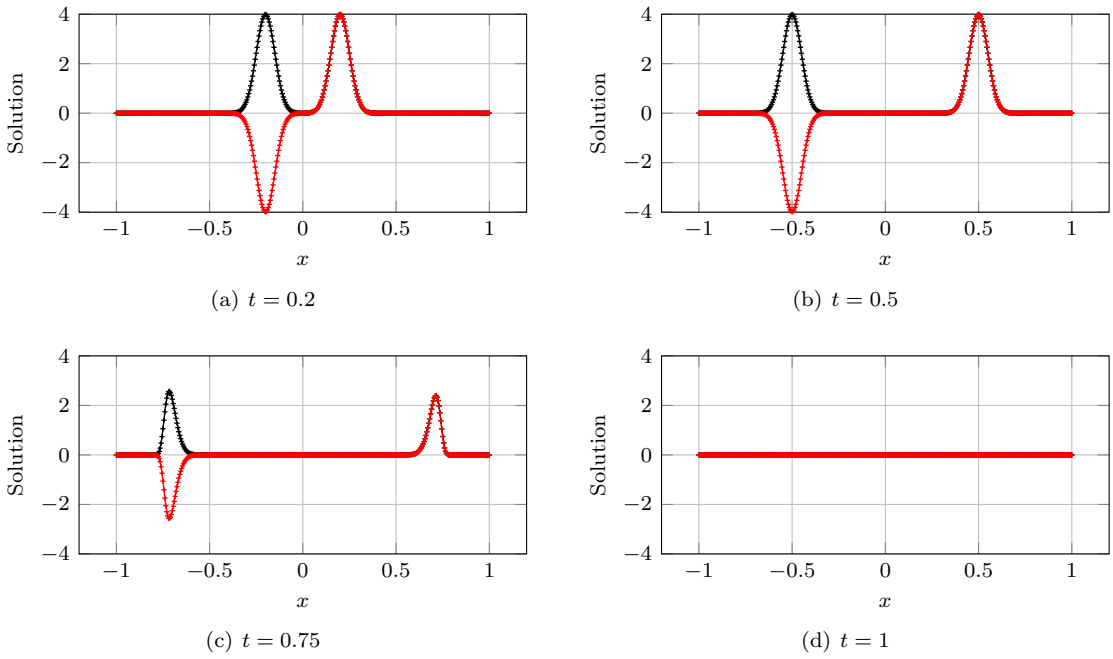


Figure 1: Example of a window function with $N_m = 400$ and $N_{\text{win}} = 80$.



(a) $t = 0.2$ (b) $t = 0.5$

(c) $t = 0.75$ (d) $t = 1$

Figure 2: Propagation of a 1D wave with the PS method and the windowing technique. The pressure p and the velocity v are plotted in black and red, respectively.

window coefficients. We use $\alpha_{\text{win}} = (N_m - 3)/14$ and $\beta_{\text{win}} = 3$. To illustrate the method, the acoustic system (2.1) is solved on the domain $\Omega = [-1, 1]$ with the initial fields

$$p(x, 0) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)} \quad \text{and} \quad v(x, 0) = 0,$$

with $\sigma = 0.1$. The window function is represented in Figure 1, and the solution is plotted at different times in Figure 2. We observe that the wave is absorbed near the boundaries of the domain.

4 Coupling of the DG and PS methods

In our approach, the computational domain is divided into large boxes (e.g. segments in 1D and rectangles in 2D) surrounded by unstructured finite element meshes to match the geometry. The PS scheme is used in the large boxes, and the DG scheme is used on the unstructured meshes. The idea is to make the PS regions as large as possible in order to reduce the computational cost of the DG regions.

The coupling between the methods is achieved thanks to an overlap between the DG regions and the PS regions. Figure 3 shows a PS region embedded between two DG regions in one dimension. We distinguish two overlaps: data is transferred from the PS region to the DG region in the $L_{\text{ov,DG}}$ overlap, and from the DG region to the PS region in the $L_{\text{ov,PS}}$ overlap.

The acoustic problem in the DG domain is marched in time using the ERK4 scheme, while the IRK4 scheme is used in the PS region. After each time step, the local solution of each region must be updated by using the local solutions of the neighboring regions. In the overlap close to the boundary of the region, the local solution is simply replaced by the local solution of the neighboring region, as shown in Figure 3. The procedure is detailed in Algorithm 2.

For the communications from a PS region to a DG region, the solution in the PS region is simply evaluated at the nodes of the DG elements in the overlap $L_{\text{ov,DG}}$. In one dimension, they are computed using equation (3.1). This approach is easily used with nodal finite elements.

For the communications from a DG region to a PS region, the DG solution is evaluated at the nodes of the PS regular spatial grid in the overlap $L_{\text{ov,PS}}$. This is a bit more involved, since this requires a mapping between the nodes of the PS spatial grid and the elements of the finite element mesh, which is unstructured. Then, for each node of the PS spatial grid, the DG solution is computed by using the local representation (2.4)-(2.5). Once the PS solution is updated on the

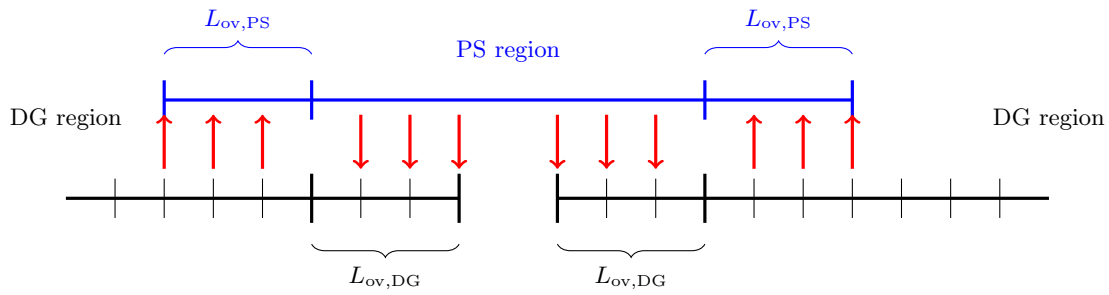


Figure 3: Partition of the domain into the DG and PS regions in one dimension. The red arrow indicated the data transfers.

Algorithm 2 Solution procedure with the coupled DG and PS methods

Set the nodal values of the initial PS solutions.
Set the initial DG solutions.
for $t = 1, \ldots, t_{\max}$ (time loop) **do**
  Compute the modal coefficients of the PS solutions (with DCT and DST).
  Update the modal coefficients of the PS solutions with IRK4.
  Update the DG solutions with ERK4.
  Compute and transfer the data on the overlaps.
  Compute the nodal values of the PS solutions (with IDCT and IDST).
  Multiply the nodal values of the PS solutions by the window function.
end for

regular grid, the modal coefficients are recomputed using the DCT and DST.

This coupling strategy introduces several parameters that need to be selected. First, the element sizes in the DG and PS regions must be specified. In one dimension, the same spatial step can be used in both DG and PS regions, which simplifies the implementation. However, the method must be able to perform well with unstructured finite element methods in order to deal with two and three-dimensional cases. Second, different time steps could be used in the DG and PS regions. When different time steps are used in neighboring regions, the solution is not communicated until it is updated in the neighboring regions. In the DG regions, the time step must verify a CFL stability condition because the ERK scheme is conditionally stable, while in the PS region there is no time step constraint. Finally, the sizes of the two overlaps are important parameters. The influence of these parameters is studied in detail in the next section.

5 Numerical studies and analysis of the parameters in 1D

In this section, we first study the DG and PS methods without coupling in a 1D configuration, then we study the coupling described in the previous section. The influence of the different parameters is systematically evaluated.

We consider a simple dimensionless reference benchmark with the parameters $c = 1$ and $\rho = 1$. The reference solution corresponds to a Gaussian pulse traveling from left to right through the domain $\Omega = [-1, 2]$,

$$p(x,t) = v(x,t) = e^{-(x-x_0-t)^2/\sigma^2}$$

with $\sigma = 0.05$ and $x_0 = -0.5$. This solution is used to initialize the fields at $t = 0$. The pulse starts in the middle of the region $[-1, 0]$ and reaches the middle of the regions $[0, 1]$ and $[1, 2]$ at $t = 1$ and $t = 2$, respectively. At $t = 3$, the pulse has left the domain if the boundary on the right-hand side is non-reflecting.

5.1 Numerical study of the DG and PS methods without coupling

We first examine the methods without coupling using the reference benchmark. For the PS method, the left and right windowing zones are $[-1, -0.8]$ and $[1.8, 2]$, respectively. Note that the Gaussian pulse has not yet reached the right windowing zone at $t = 2$, and it must have left the domain at $t = 3$. By default, we consider the ERK4 time-stepping scheme for the DG method and the IRK4 time-stepping scheme for the PS method. The element size for the DG scheme is

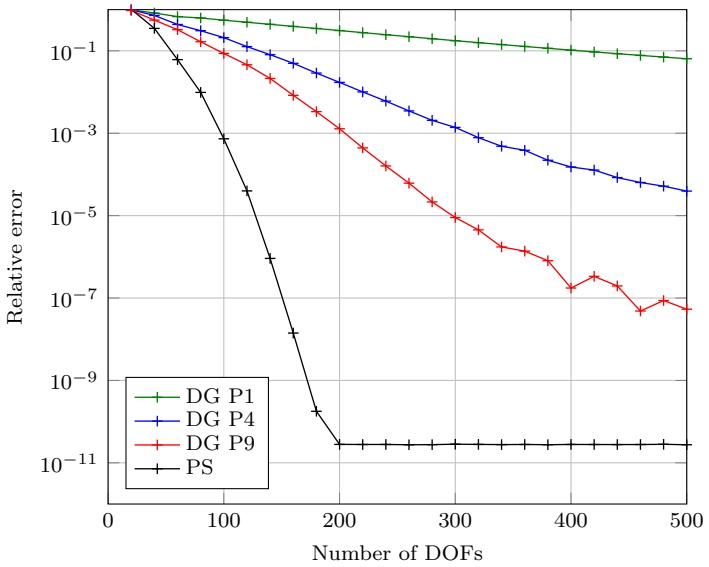


Figure 4: Numerical errors of the DG and PS methods as functions of the number of degrees of freedom (DOFs) for the 1D reference benchmark.

denoted by Δx_{DG}, and that for the PS method is given by $\Delta x_{\mathrm{PS}} = L/N_m$, where L is the size of the domain and N_m and the number of modes.

Influence of the spatial discretization

The accuracy of the spatial discretization schemes is studied by computing the error at the time $t = 2$. To avoid the influence of the time-stepping scheme on the accuracy, the methods are tested with a very small time step, $\Delta t = 10^{-4}$.

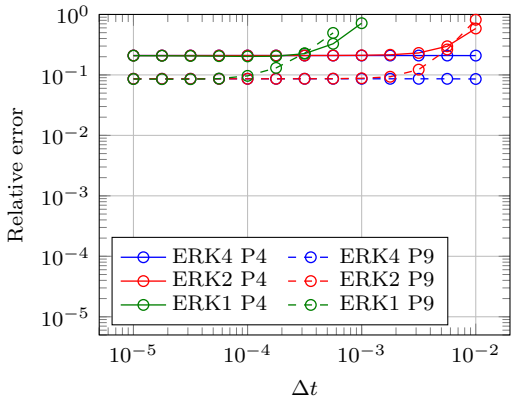
The relative errors are plotted as a function of the number of degrees of freedom (DOFs) in Figure 4. The L^2 norm of the error on the pressure field is computed with a Gaussian integration for the DG method, and with a trapezoidal rule for the PS method. The number of DOFs corresponds to the number of modes for the PS method, and to the number of nodes for the DG method. We have tested P1, P4 and P9 elements, corresponding to $\mathsf{p} = 1$, 4 and 9, respectively.

With the DG method, the numerical error is systematically reduced as the polynomial order is increased, for any given number of DOFs. The expected order of convergence $\mathcal{O}((\Delta x_{\mathrm{DG}})^{\mathsf{p}+1})$ is recovered (results not shown for brevity), see [16].

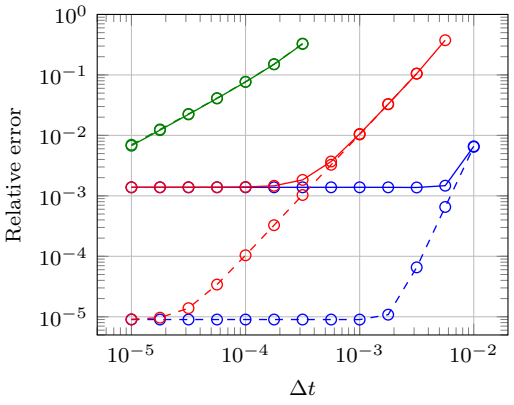
With the PS method, the error decreases rapidly as we increase the number of modes until it reaches machine accuracy at about 200 modes. This result is expected since the PS method has exponential convergence [45]. We have also verified that there is a similar convergence with the PS method when using exponential functions as basis functions. In fact, there is little difference between the two sets of basis functions for the PS method. For the remainder of this article, the PS method is considered using only the trigonometric basis functions.

Influence of the time discretization

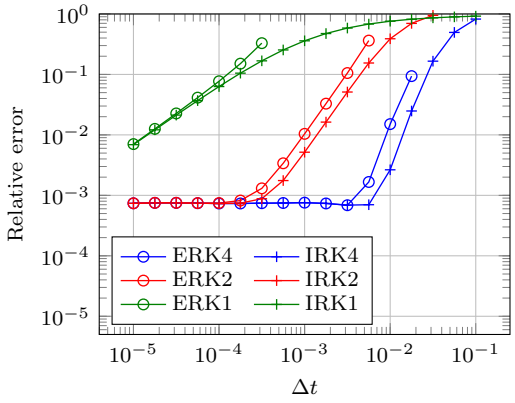
We now compare the different Runge–Kutta time-stepping schemes, and we study the influence of the time step Δt. In Figure 5, the relative error is plotted as a function of the time step Δt for



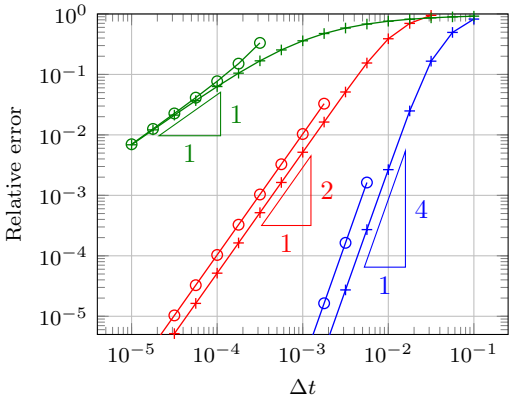
(a) DG method with 100 DOFs



(b) DG method with 300 DOFs



(c) PS method with 100 DOFs



(d) PS method with 300 DOFs

Figure 5: Error obtained with the DG *(top)* and spectral *(bottom)* methods with different time-stepping schemes as a function of the time step Δt for the 1D reference benchmark. For both methods, we have used either 100 DOFs *(left)* or 300 DOFs *(right)*.

the different methods and the different time-stepping schemes.

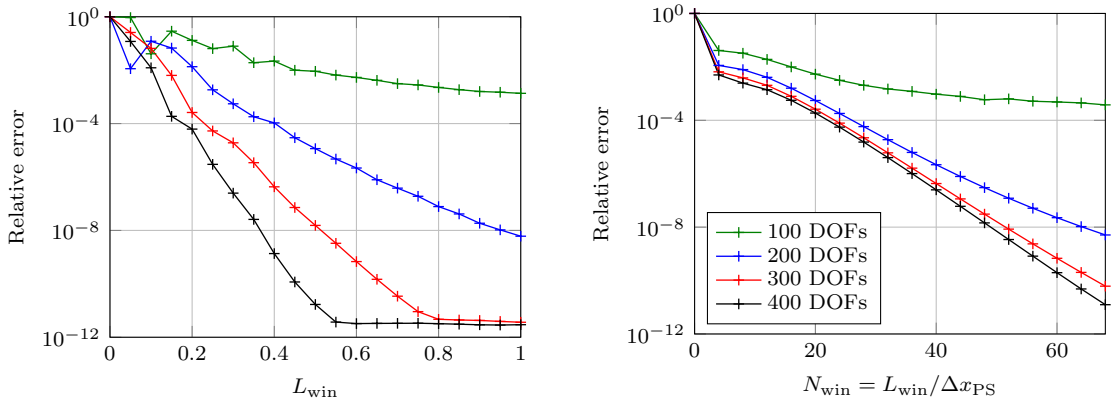
With the DG method, the error quickly reaches a plateau as we decrease Δt. This plateau corresponds to the spatial resolution, which then dominates the overall numerical error. The level of the plateau is lowered by tacking a larger number of DOFs or a higher polynomial degree.

The plateaus are much lower with the PS method, and we observe the expected convergence rates of the different RK time-marching schemes. For any given value of Δt, the ERK4 and IRK4 schemes achieve better accuracy than the RK2 or RK1 schemes at similar computational cost. In addition, the implicit methods have a slightly better accuracy than the explicit methods for the same time step Δt.

Influence of the windowing for the PS method

To study the windowing, the simulation is run up to $t = 3$. At this final time, the traveling pulse should have left the domain completely. Therefore, the pulse should be absorbed by the right window, ideally without any reflection.

The relative error is estimated by computing the norm of the solution in the domain at the



(a) Error *vs* length of the window L_{win}

(b) Error *vs* number of points in the window N_{win}

Figure 6: Error obtained with the PS method as a function of the size of the window on the right of the domain. These results are obtained with $N_{\text{win}} = 100$, 200, 300 and 400 modes, and with the IRK4 scheme for $\Delta t = 10^{-4}$.

final time, divided by that at the initial time. This relative error should be close to 1 if the pulse is reflected, and close to 0 if it is completely absorbed. It could be interpreted as the size of a reflection coefficient.

In Figure 6, the error is plotted as a function of the size of the window on the right, for several numbers of modes. The size of the window can be defined in terms of the length L_{win} and in terms of the number of points in the right window, i.e. $N_{\text{win}} = \text{ceil}(L_{\text{win}}/\Delta x_{\text{PS}})$. We observe that the error decreases as we increase the length of L_{win}, and the decay is faster with a larger number of modes (Figure 6(a)). In fact, the amount of spurious reflections can be controlled by the number of points inside the window. We observe in Figure 6(b) that the relative error is close to 10^{-4} with $N_{\text{win}} = 20$ points in all cases, except for the simulation with the smaller number of DOFs. In the latter case, the global error is likely dominated by the spatial discretization of the PS scheme.

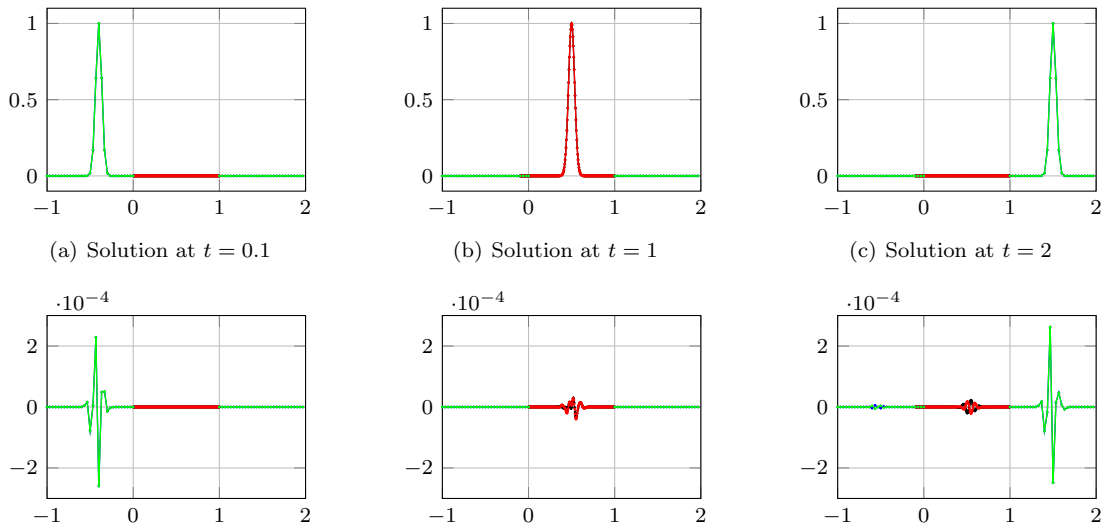
5.2 Numerical study of the DG/PS coupling

To study the DG/PS coupling, the computational domain $[-1, 2]$ is partitioned into one PS region $[-L_{\text{ov,PS}}, 1 + L_{\text{ov,PS}}]$ placed between two DG regions $[-1, L_{\text{ov,DG}}]$ and $[1 - L_{\text{ov,DG}}, 2]$, with small overlapping zones of size $L_{\text{ov,PS}}$ and $L_{\text{ov,DG}}$. The Gaussian pulse starts in the left DG region, it traverses the PS region, and it is located in the right DG region at the end of the simulation. Snapshots of the DG and PS solutions and errors are shown in Figure 7 at different times.

The DG solutions are updated with ERK4, the PS solution is updated with IRK4, and data is exchanged in the overlapping zones, as described in Section 4. In the DG regions, the degree of the polynomial basis functions is $P = 4$, and the time step is given by the empirical rule $\Delta t_{\text{DG}} = 0.5/(2P + 1)\Delta x_{\text{DG}}$ [9]. In the PS region, the size of the window L_{win} is always equal to the size of the overlap $L_{\text{ov,PS}}$, and the time step is $\Delta t_{\text{PS}} = \Delta t_{\text{DG}}$ by default.

Validation of the coupling and parameter selection

To assess the accuracy of the coupling method and to select the parameters, a number of simulations are reported in Table 1. Each simulation targets a given level for the overall numerical error.



(a) Solution at $t = 0.1$ (b) Solution at $t = 1$ (c) Solution at $t = 2$

(d) Error at $t = 0.1$ (e) Error at $t = 1$ (f) Error at $t = 2$

Figure 7: Snapshots of the solution and the corresponding error at different times. Green curves correspond to the DG regions, and red curves correspond to the PS region. The parameters correspond to the target error 10^{-4} given in Table 1.

First, the element size in the DG regions (Δx_{DG}) and the distance between two points in the PS region (Δx_{PS}) are adjusted so that the numerical error in each region is close to the target error when the coupling is disabled (i.e. the exact solution is prescribed at the interface) and $\Delta t_{\mathrm{DG}} = 10^{-4}$. The second and third columns in Table 1 correspond to the relative errors obtained in this way.

The fourth column in Table 1 gives the numerical error with the coupling strategy and either $\Delta t_{\mathrm{DG}} = 10^{-4}$ or, in parentheses, the time step given by the empirical rule above. Any increase in numerical error in the coupled model can be fully attributed to the coupling method. We observe that for all the error targets, the error for the coupling method is greater than the error when using either the DG or PS method alone, but it remains of the same order of magnitude. This indicates that the proposed coupling does not deteriorate significantly the global accuracy of the simulation.

Influence of the overlaps

In this section, we study how the size of the overlaps can impact the accuracy of the solution.

Table 1: Parameters and numerical errors for the coupling method with different error targets.

TE	Error DG	Error PS	Error DG/PS coupling	Δx_{DG}	Δx_{PS}	K	N	$K_{\mathrm{ov,DG}}$	$N_{\mathrm{ov,PS}}$
10^{-2}	$9.5\ 10^{-3}$	$8.2\ 10^{-3}$	$2.0\ 10^{-2}$ $(1.6\ 10^{-2})$	0.080	0.025	15	56	2	9
10^{-3}	$9.3\ 10^{-4}$	$8.1\ 10^{-4}$	$2.5\ 10^{-3}$ $(1.9\ 10^{-3})$	0.055	0.018	22	77	3	12
10^{-4}	$9.2\ 10^{-5}$	$8.4\ 10^{-5}$	$3.8\ 10^{-4}$ $(2.7\ 10^{-4})$	0.037	0.014	32	99	5	15
10^{-5}	$9.4\ 10^{-6}$	$9.9\ 10^{-6}$	$2.7\ 10^{-5}$ $(2.1\ 10^{-5})$	0.023	0.012	52	121	8	18
10^{-6}	$9.8\ 10^{-7}$	$6.7\ 10^{-7}$	$3.1\ 10^{-6}$ $(3.4\ 10^{-6})$	0.015	0.009	82	147	13	22
10^{-7}	$9.9\ 10^{-8}$	$8.0\ 10^{-8}$	$2.6\ 10^{-7}$ $(2.0\ 10^{-7})$	0.009	0.008	130	169	21	25
10^{-8}	$9.8\ 10^{-9}$	$6.2\ 10^{-9}$	$3.6\ 10^{-8}$ $(3.9\ 10^{-8})$	0.006	0.007	208	196	34	29

First, we study the influence of the DG overlap with a fixed PS overlap. Figure 8 shows the numerical error for different numbers of DG elements in the overlap. In all the cases, using more than 5 elements in the DG overlap does not significantly improve the accuracy. However, reducing the number of elements below 5 systematically increases the error. This threshold of 5 elements follows directly from the choice of the time integration scheme. Indeed, the low-storage RK4 scheme [7, 8] involves five stages at each time step. At each stage, the DG model transfers data from each DG element to its neighbors. Therefore, at each time step, information is transmitted through five DG elements. The results in Figure 8 indicate that the DG overlap should be sufficiently wide so that the waves cannot completely cross the overlap during a single time step. To verify this interpretation, we have checked (results not shown here) that, when using the RK1 and RK2 schemes, increasing the width of the DG overlap beyond 1 and 2 elements, respectively, does not improve the accuracy.

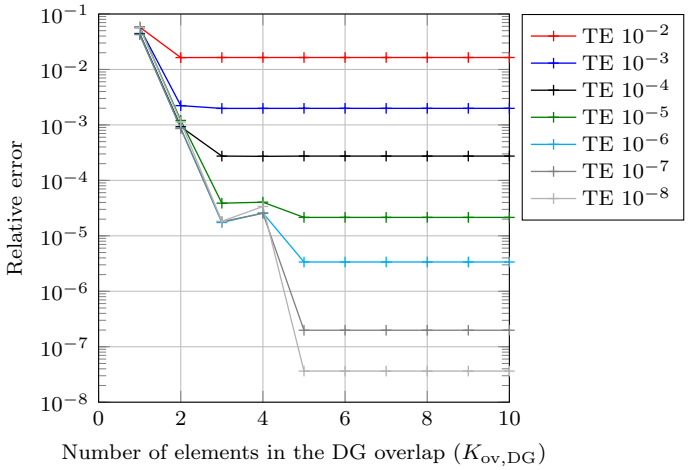
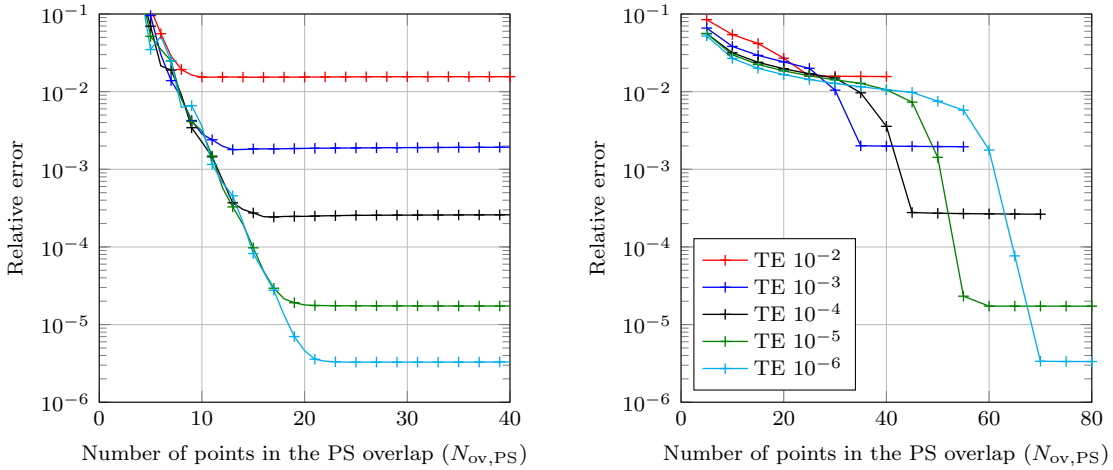


Figure 8: Error as a function of the number of elements in the DG overlap with the parameters given in Table 1.



(a) Case with windowing

(b) Case without windowing

Figure 9: Error as a function of the number of points in the PS overlap with the parameters given in Table 1.

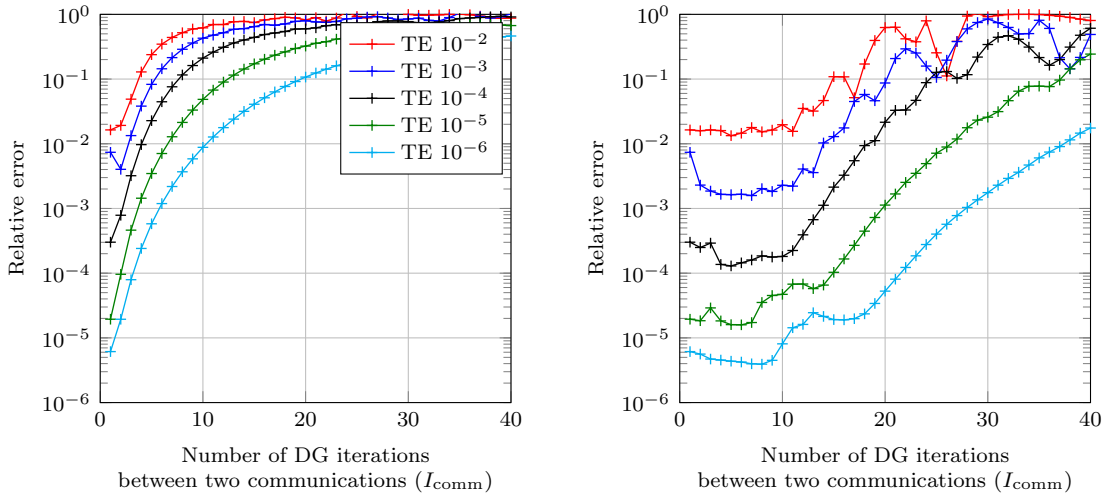
We now investigate how the windowing technique and the size of the PS overlap can impact the accuracy of the solution. Figure 9 shows the numerical error as a function of the number of points in the PS overlap, obtained with and without windowing. When the PS overlap is thin, i.e. with a few points, the relative error is high in all the cases. The error decreases as the number of points in the overlap increases, and the decay is much faster with the windowing technique. With many points, the relative errors reach the same levels with and without windowing. This can be expected because, with a thick PS overlap, the PS solution is replaced on large zones close to the boundaries of the PS region at each iteration. This reduces the influence of the treatment at the boundary of the PS zone.

Influence of the communications at the DG/PS interface and the PS time step

We now consider cases where the time steps in the DG and PS regions are not equal, and where the data exchange does not occur at every time step. The DG time step (Δt_{DG}) must verify a stability condition, while the PS time step (Δt_{PS}) can be much larger thanks to the use of an implicit time stepping scheme.

First, we increase the PS time step according to $\Delta t_{\mathrm{PS}} = I_{\mathrm{comm}} \Delta t_{\mathrm{DG}}$ for a given DG time step, where I_{comm} is a positive integer. The communication between the regions is performed after each PS time step. Therefore, the DG solution in the DG overlap is updated at every I_{comm} DG time step. In Figure 10(a), it is clear that the error increases rapidly with I_{comm}.

In Figure 10(b), the DG and PS time steps are equal, but the communications are performed at every I_{comm} time step. Therefore, the methods could run independently with communication only at certain steps. We observe that the error does not increase as much as in Figure 10(a). Since the only difference between the two figures is the PS time step, this means that the error in the left figure is mainly due to the time stepping scheme used in the PS region. We also observe that the effect of I_{comm} on the results is not significant, up to a certain value of I_{comm}. The simulation time between two communications (i.e. $I_{\mathrm{comm}} \Delta t_{\mathrm{DG}}$) cannot be larger than the simulation time to cross



(a) Case $\Delta t_{\mathrm{PS}} = I_{\mathrm{comm}} \Delta t_{\mathrm{DG}}$

(b) Case $\Delta t_{\mathrm{PS}} = \Delta t_{\mathrm{DG}}$

Figure 10: Error as a function of the number of DG iterations between two communications (I_{comm}) with the parameters given in Table 1.

the smallest overlap (i.e. $\min(L_{\text{ov,PS}}, L_{\text{ov,DG}})/c$). When $I_{\text{comm}}\Delta t_{\text{DG}} > \min(L_{\text{ov,PS}}, L_{\text{ov,DG}})/c$, the relative error is close to one in all the cases (results not shown for brevity).

6 Numerical studies in two dimensions

6.1 Validation of the PS method

To validate the PS implementation in two dimensions with the IRK4 scheme, we consider a dimensionless reference benchmark that represents the propagation of an acoustic pulse in the square domain $\Omega = [-2, 2] \times [-2, 2]$ with $\rho = 1$ and $c = 1$. The windowing of the PS solution is applied to the boundary of this domain with $L_{\text{win}} = 2.5$. The initial condition is a Gaussian pressure pulse located at the center of the domain,

$$p(\mathbf{x}, 0) = e^{-\|\mathbf{x}\|^2/(2\sigma^2)} \quad \text{and} \quad \mathbf{v}(\mathbf{x}, 0) = \mathbf{0},$$

with $\sigma = 0.05$. We use the same number of modes on the two directions $N_x = N_y = N_m$ for the PS method. Figure 11 shows the numerical solution at different times.

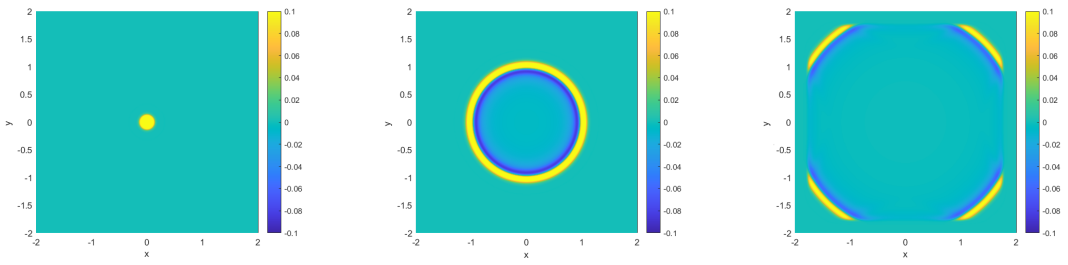
The solution of the PS method at $t = 1$ is compared with a reference numerical solution obtained with many modes. Note that at $t = 1$ the acoustic pulse has not yet reached the outer boundary of the computational domain. The influence of the number N_m of modes in each direction on the L^2 error is plotted in Figure 12(a). As in one dimension, the error decreases rapidly as we increase the number of modes. In addition, the performance of the IRK4 scheme is examined by varying the time step Δt, see Figure 12(b). As expected, the IRK4 scheme exhibits a fourth-order convergence rate.

6.2 Validation and analysis of the coupling method

In this section, we present results for the coupling method in two dimensions, and we compare them with results obtained with the DG method. We consider again the computational domain $\Omega = [-2, 2] \times [-2, 2]$ with $\rho = 1$ and $c = 1$. A plane wave propagates through the domain along the direction $\mathbf{d} = (1, 1)/\sqrt{2}$. The reference solution is given by

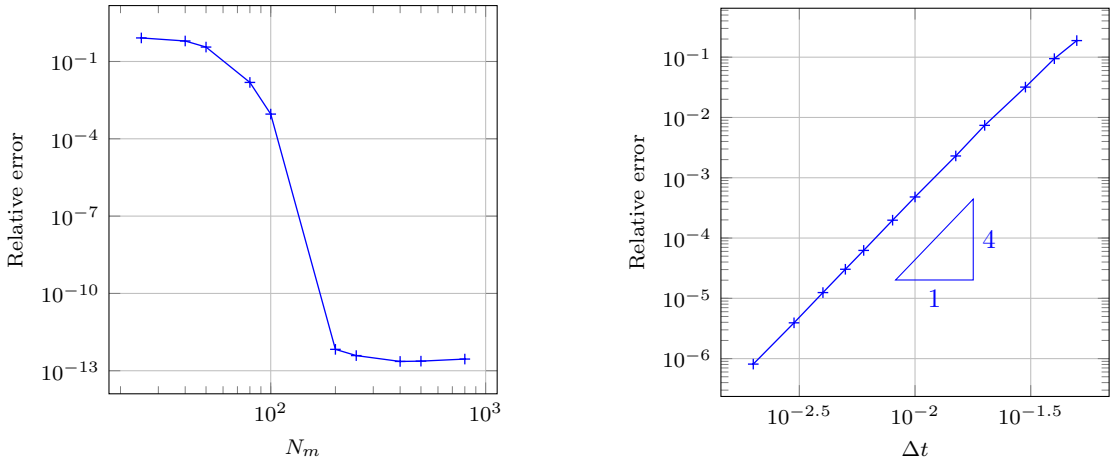
$$p(\mathbf{x}, t) = e^{-(\mathbf{d}\cdot\mathbf{x} - x_0 - t)^2/\sigma^2} \quad \text{and} \quad \mathbf{v}(\mathbf{x}, t) = p\mathbf{d},$$

with $\sigma = 0.5$ and $x_0 = -4$. It is used to define the initial fields.



(a) $t = 0$ (b) $t = 1$ (c) $t = 2$

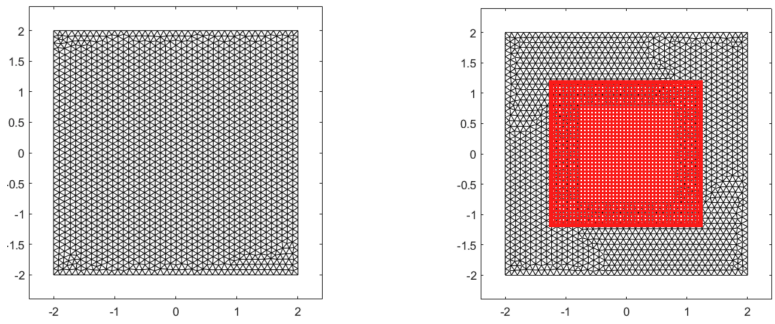
Figure 11: Pressure at different times in two dimensions.



(a) Spatial resolution

(b) Temporal resolution

Figure 12: Error given in terms of spatial and time discretization.



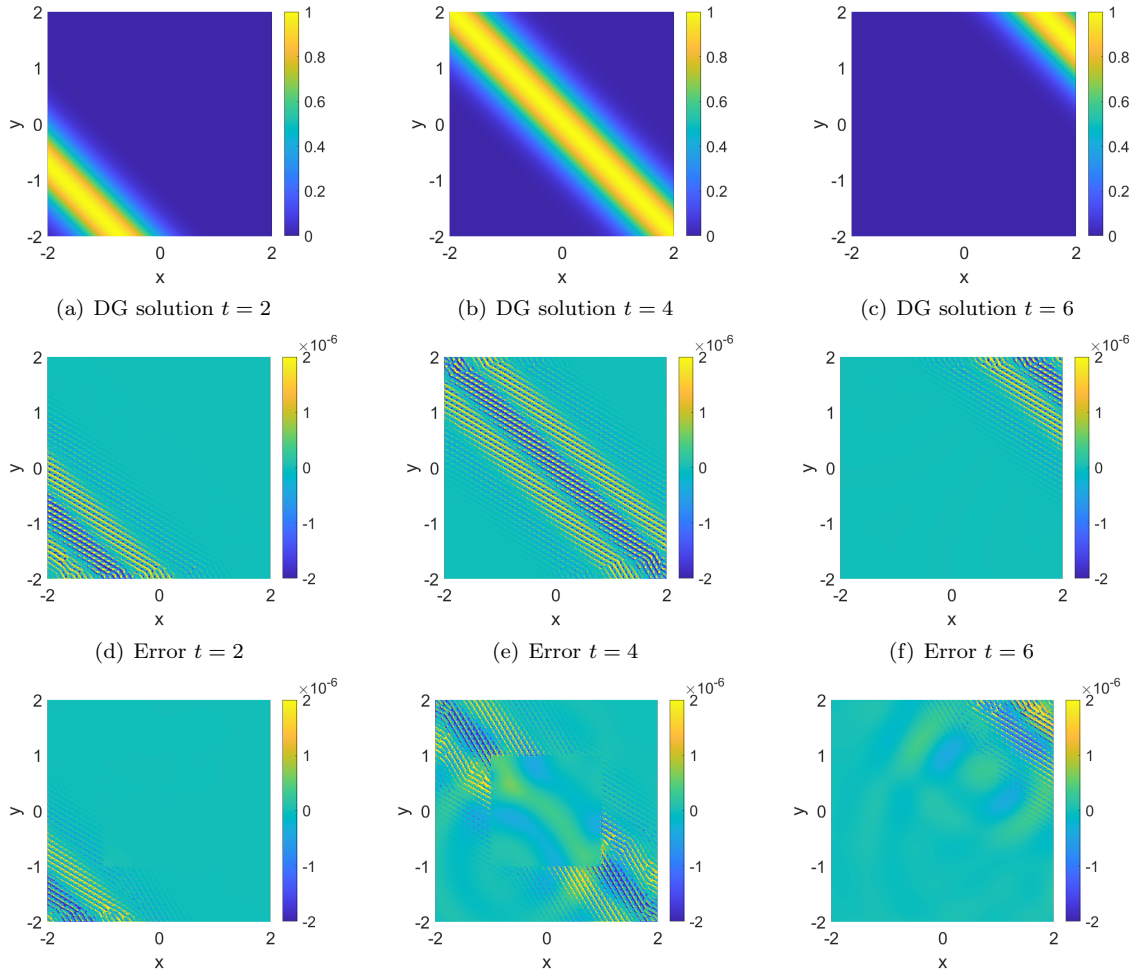
(a) DG mesh

(b) Hybrid mesh

Figure 13: Examples of meshes used for the comparison between the DG method and the coupling DG/SP method. DG elements are shown in black. The spectral domain is shown in red.

For the DG simulations, the entire domain is meshed with P3 elements, see Figure 13(a), and the non-homogeneous Robin condition (2.2) is prescribed at the boundaries of the mesh. For the DG/PS simulations, we define an interior PS region $\Omega_{\text{PS}} = [-1 - L_{\text{ov,PS}}, 1 + L_{\text{ov,PS}}]^2$ inside Ω with $L_{\text{ov,PS}} = 0.3$, see Figure 13(b). The same number of modes is used in both directions, $N_x = N_y = N_m$ with $\Delta x_{\text{PS}} = \Delta y_{\text{PS}} = 10^{-2}$. The windowing technique is used at the boundary of this region. The PS region is surrounded by a DG region Ω_{DG} with P3 elements. The DG region corresponds to Ω from which the square $[-1 + L_{\text{ov,DG}}, 1 - L_{\text{ov,DG}}]^2$ is removed, with $L_{\text{ov,DG}} = 0.2$. A non-homogeneous Robin boundary condition is used at the exterior boundary of Ω_{DG}. The same time step is used in all the regions. Here, we use the empirical rule $\Delta t = 2/3 \min_n(\Delta r_n)$, where $\min(\Delta r_n)$ is the minimal distance between two finite element nodes in the mesh, see [16].

The solutions and numerical errors are shown in Figure 14 for the DG method and the coupling method. Only the DG solutions are shown in Figure 14 because the solutions from both methods are very close and cannot be visually distinguished. The error in the DG method follows the plane wave. For the coupling method, we observe that the error is lower in the PS region than in the DG region, see Figure 14(h). In addition, for long times, we observe small reflections due to the interface between the two regions, see Figure 14(i).



(a) DG solution $t = 2$ (b) DG solution $t = 4$ (c) DG solution $t = 6$

(d) Error $t = 2$ (e) Error $t = 4$ (f) Error $t = 6$

(g) Error $t = 2$ (h) Error $t = 4$ (i) Error $t = 6$

Figure 14: Top: DG results at different times. Middle: Difference between the DG results and the exact solution. Bottom: Difference between the hybrid method and the exact solution.

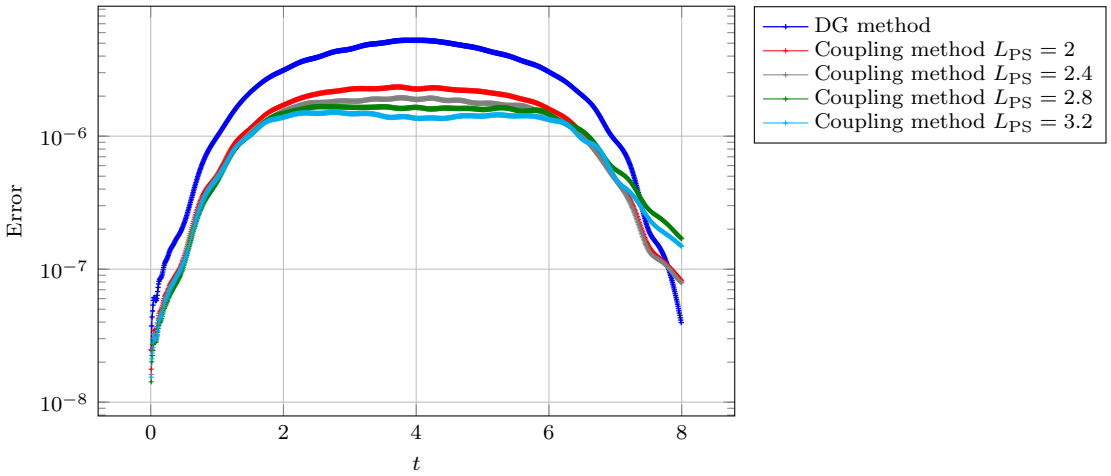


Figure 15: Error as a function of time for the DG and coupling methods.

Next, we gradually increase the size of the PS region without modifying the domain Ω, thereby decreasing the size of the DG region. We consider square PS regions of sizes $L_{\text{PS}} = 2$, 2.4, 2.8, and 3.2. In these cases, the number of degrees of freedom in the DG region represents respectively 32%, 24%, 18% and 12% of the total number of degrees of freedom, respectively.

Figure 15 shows the error as a function of time for the DG method and the coupling method with different sizes of the PS region. As expected, the numerical errors increase as the plane wave enters the domain and then decrease as the wave leaves the domain. In addition, we observe that the DG error is larger than the coupling method error in all cases and for most of the simulation time. This is consistent with the previous results, e.g. Figure 14(h), which shows a smaller error in the PS domain. At the end of the simulation ($t = 8$), the error of the coupling method is slightly higher than the DG error. This may be due to the small spurious reflections generated at the interface between the PS and DG regions. Varying the size L_{PS} of the PS region does not significantly change the numerical error over the duration of the simulation, indicating that the accuracy of the coupling method can be maintained.

6.3 Example of application

We present an example with a more complex geometry, shown in Figure 16(a). A Gaussian pulse propagates through the mesh, starting at position $\mathbf{x}_0 = (0.75, 0.25)$. We consider the parameters $\rho = 1$ and $c = 1$, and a homogeneous Dirichlet condition is imposed at the boundary of the domain.

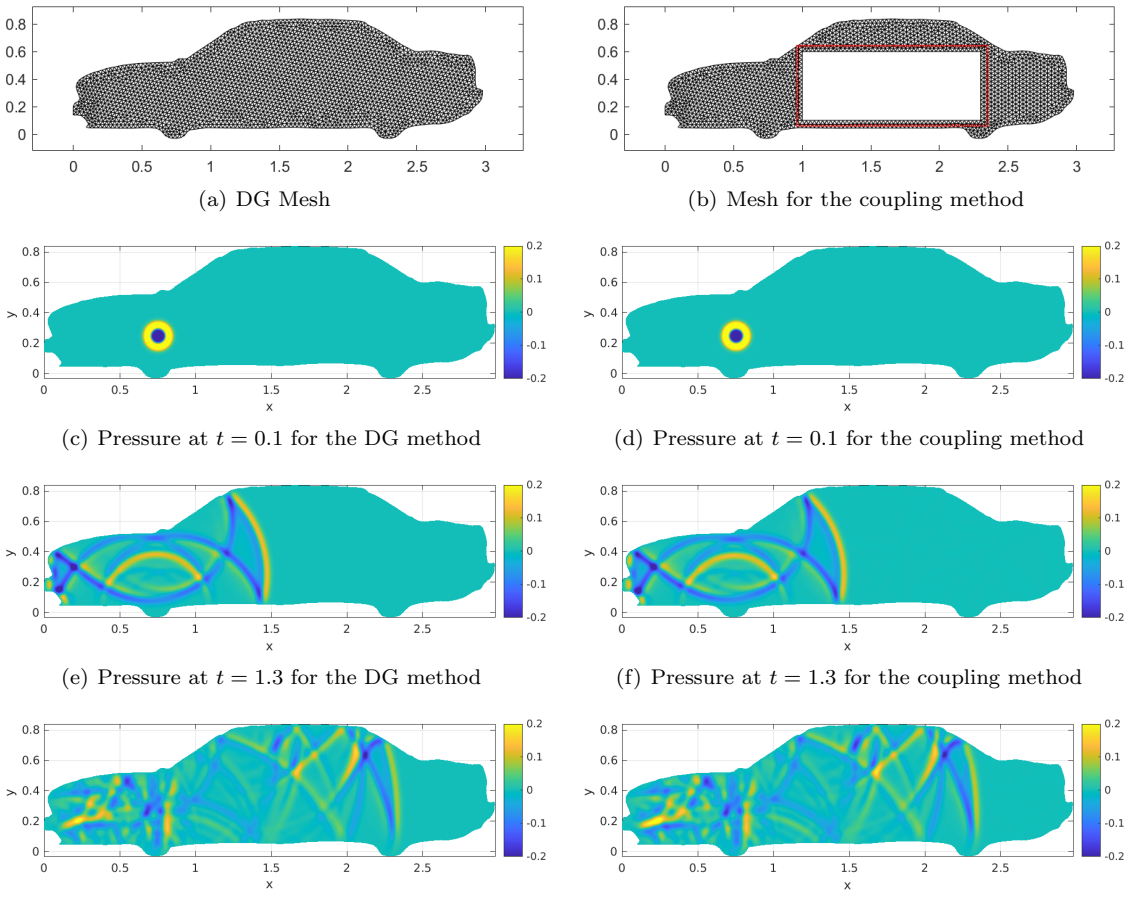
The solution is computed using the DG method alone and the coupling strategy. Figure 16(a) shows an example of a triangular mesh for the DG model covering the whole domain (in practice, a finer mesh was used). For the coupling method, the PS region is a rectangle covering a large, central part of the domain, see Figure 16(b). The unstructured mesh used in the DG region is able to represent the gap between the PS region and the complex boundary of the domain. The DG model consists of 65350 P3 elements, while in the coupling model, there are 41659 P3 DG elements and 50000 modes in the PS region.

The solutions obtained with the DG method and the DG/SP coupling method are shown in Figure 16. The two methods give almost identical results. This shows that the coupling method does not introduce significant scattering at the interface between the PS and DG regions.

7 Summary and perspectives

We have presented and extensively studied a coupling strategy of PS and DG methods to solve the wave equation in the time domain. The DG method uses high-order nodal polynomial shape functions, and the PS method uses trigonometric modes. The motivation for this coupling strategy is to combine the efficiency of the PS method for large-scale simulations with the flexibility of the DG method for complex geometries. However, this approach introduces several parameters that need to be carefully studied.

The exchange of information between the PS and DG regions is performed by interpolating the solutions over the overlaps between the two regions. The proposed method can handle non-conforming meshes, i.e. the PS nodes do not have to coincide with the DG nodes. Moreover, by exploiting the peculiar structure of the spectral system, we have developed efficient Runge–Kutta implicit time integration schemes with no additional computational cost. This gives more flexibility in the choice of time step for the spectral method.



(a) DG Mesh

(b) Mesh for the coupling method

(c) Pressure at $t = 0.1$ for the DG method

(d) Pressure at $t = 0.1$ for the coupling method

(e) Pressure at $t = 1.3$ for the DG method

(f) Pressure at $t = 1.3$ for the coupling method

(g) Pressure at $t = 3$ for the DG method

(h) Pressure at $t = 3$ for the coupling method

Figure 16: Mesh and results for the DG method *(left)* and the coupling method *(right)*.

A detailed parametric study of the method has been performed in one dimension. The accuracy of the coupling method is mainly influenced by the width of the overlaps. The width of the DG overlap is directly related to the time-integration scheme (e.g., five elements for the fourth-order, low-memory scheme used in the present work). It was also shown that one can use different time steps in the two methods, although increasing the PS time step should be limited due to the time discretization error. Synchronization between the PS and DG regions can be performed every few time steps.

The coupling method has also been implemented and validated in two dimensions using unstructured triangular meshes in the DG method. The coupling of the methods does not significantly increase the numerical error.

The main perspective of this work is the development of optimized 3D codes, which are of paramount importance to test and validate the approach on real cases. Indeed, the performance of the resulting solvers could be compared with DG solvers to evaluate the speedup provided by the coupling method. Implementation techniques have already been proposed to develop efficient PS and DG solvers, including on GPU clusters, as discussed in the introduction.

Another important perspective is to further develop the versatility of the approach. We think that multiple PS regions could be used and arranged differently to cover a larger region of the

computational domain and further reduce the number of DG degrees of freedom. This may require the introduction of a coupling technique between adjacent PS regions. The PS region could also be used up to the boundary of the domain, provided that a periodic boundary condition or a non-reflecting boundary condition has to be imposed, which increases the range of applications.

Acknowledgments. This work was supported in part by France Relance and Siemens Industry Software SAS (WavesDG-GPU project, 15/01/2022-14/01/2024).

A Runge–Kutta time stepping schemes

In this appendix, we derive Implicit Runge–Kutta (IRK) schemes for the general linear system

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{F}\mathbf{y},$$

with a constant square matrix $\mathbf{F}$ and an unknown vector $\mathbf{y}$. A s-stage IRK scheme is given by

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \Delta t \sum_{i=1}^{s} b_i \mathbf{k}_i,$$

with

$$\mathbf{k}_i = \mathbf{F}\Big(\mathbf{y}^n + \Delta t \sum_{j=1}^{s} a_{ij}\mathbf{k}_j\Big), \quad \text{for } i = 1 \dots s,$$

where a_{ij} and b_i are constant coefficients. The coefficients of several standard schemes are given in Table 2.

Table 2: Butcher tables for several IRK methods.

(a) Notation

$\mathbf{c}$	$\mathbf{A}$
	$\mathbf{b}^\top$

(b) Implicit Euler

1	1
	1

(c) Trapezoidal rule

0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

(d) Gauss–Legendre

$\frac{1}{2} - \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4} - \frac{\sqrt{3}}{6}$
$\frac{1}{2} + \frac{\sqrt{3}}{6}$	$\frac{1}{4} + \frac{\sqrt{3}}{6}$	$\frac{1}{4}$
	$\frac{1}{2}$	$\frac{1}{2}$

The implicit Euler method (Table 2b) is a first-order scheme (IRK1). We have

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{F}(\mathbf{y}^n + \Delta t\, \mathbf{k}_1), \\ \mathbf{y}^{n+1} &= \mathbf{y}^n + \Delta t\, \mathbf{k}_1, \end{aligned}$$

which leads to the update formula

$$\mathbf{y}^{n+1} = (\mathbf{I} - \Delta t\, \mathbf{F})^{-1}\, \mathbf{y}^n.$$

The trapezoidal rule (Table 2c) gives a second-order scheme (IRK2). We have

$$\begin{aligned}\mathbf{k}_1 &= \mathbf{F}\mathbf{y}^n,\\ \mathbf{k}_2 &= \mathbf{F}\left(\mathbf{y}^n + \tfrac{\Delta t}{2}\mathbf{k}_1 + \tfrac{\Delta t}{2}\mathbf{k}_2\right),\\ \mathbf{y}^{n+1} &= \mathbf{y}^n + \tfrac{\Delta t}{2}\mathbf{k}_1 + \tfrac{\Delta t}{2}\mathbf{k}_2.\end{aligned}$$

By deriving explicit expressions for $\mathbf{k}_1$ and $\mathbf{k}_2$ and replacing in the last equation, we obtain the update formula

$$\mathbf{y}^{n+1} = \left(\mathbf{I} - \tfrac{\Delta t}{2}\mathbf{F}\right)^{-1}\left(\mathbf{I} + \tfrac{\Delta t}{2}\mathbf{F}\right)\mathbf{y}^n.$$

The Gauss–Legendre method is a two-step fourth-order scheme (IRK4). We have

$$\begin{aligned}\mathbf{k}_1 &= \mathbf{F}\left(\mathbf{y}^n + \Delta t\, a_{11}\,\mathbf{k}_1 + \Delta t\, a_{12}\,\mathbf{k}_2\right),\\ \mathbf{k}_2 &= \mathbf{F}\left(\mathbf{y}^n + \Delta t\, a_{21}\,\mathbf{k}_1 + \Delta t\, a_{22}\,\mathbf{k}_2\right),\\ \mathbf{y}^{n+1} &= \mathbf{y}^n + \tfrac{\Delta t}{2}\mathbf{k}_1 + \tfrac{\Delta t}{2}\mathbf{k}_2.\end{aligned}$$

Combining these equations and using the coefficients in Table 2d give

$$\begin{aligned}\mathbf{k}_1 &= \left(\mathbf{I} + \Delta t\tfrac{\sqrt{3}}{6}\mathbf{F}\right)^{-1}\left(\mathbf{I} - \Delta t\tfrac{\sqrt{3}}{6}\mathbf{F}\right)\mathbf{k}_2,\\ \mathbf{k}_2 &= \left((\mathbf{I} - \tfrac{\Delta t}{4}\mathbf{F}) - \Delta t\left(\tfrac{1}{4} + \tfrac{\sqrt{3}}{6}\right)\mathbf{F}\left(\mathbf{I} + \Delta t\tfrac{\sqrt{3}}{6}\mathbf{F}\right)^{-1}\left(\mathbf{I} - \Delta t\tfrac{\sqrt{3}}{6}\mathbf{F}\right)\right)^{-1}\mathbf{F}\mathbf{y}^n,\\ \mathbf{y}^{n+1} &= \mathbf{y}^n + \Delta t\left(\mathbf{I} + \Delta t\tfrac{\sqrt{3}}{6}\mathbf{F}\right)^{-1}\mathbf{k}_2.\end{aligned}$$

We finally obtain:

$$\begin{aligned}\mathbf{y}^{n+1} = \mathbf{y}^n + \Delta t\left(\mathbf{I} + \Delta t\tfrac{\sqrt{3}}{6}\mathbf{F}\right)^{-1}&\\ \left((\mathbf{I} - \tfrac{\Delta t}{4}\mathbf{F}) - \Delta t\left(\tfrac{1}{4} + \tfrac{\sqrt{3}}{6}\right)\mathbf{F}\left(\mathbf{I} + \Delta t\tfrac{\sqrt{3}}{6}\mathbf{F}\right)^{-1}\left(\mathbf{I} - \Delta t\tfrac{\sqrt{3}}{6}\mathbf{F}\right)\right)^{-1}&\mathbf{F}\mathbf{y}^n.\end{aligned}$$

References

[1] M. Bernacki and S. Piperno. A dissipation-free time-domain discontinuous Galerkin method applied to three-dimensional linearized Euler equations around a steady-state non-uniform inviscid flow. *Journal of Computational Acoustics*, 14(04):445–467, Dec. 2006.

[2] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[3] O. P. Bruno and J. Paul. Two-dimensional Fourier continuation and applications. *SIAM Journal on Scientific Computing*, 44(2):A964–A992, 2022.

[4] O. P. Bruno, Y. Han, and M. M. Pohlman. Accurate, high-order representation of complex three-dimensional surfaces via Fourier continuation analysis. *Journal of computational Physics*, 227(2):1094–1125, 2007.

[5] J. C. Butcher. Implicit Runge–Kutta processes. *Mathematics of computation*, 18(85):50–64, 1964.

[6] C. Canuto, M. Hussaini, A. Quarteroni, and T. Zang. Spectral methods. *Encyclopedia of Computational Mechanics*, 2004.

[7] M. H. Carpenter and C. A. Kennedy. Fourth-order 2n-storage Runge-Kutta schemes. Tech-

nical report, NASA Langley Research Center Hampton, VA, United States, 1994.

[8] D. Cavaglieri and T. Bewley. Low-storage implicit/explicit Runge–Kutta schemes for the simulation of stiff high-dimensional ODE systems. *Journal of Computational Physics*, 286: 172–193, 2015.

[9] B. Cockburn and C.-W. Shu. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of scientific computing*, 16:173–261, 2001.

[10] M. Cosnefroy and M. Hornikx. Hybrid Fourier pseudospectral/discontinuous Galerkin time-domain method for urban sound propagation in a moving atmosphere. *Proceedings of Euronoise 2021*, 2021.

[11] M. Fuhry, A. Giuliani, and L. Krivodonova. Discontinuous Galerkin methods on graphics processing units for nonlinear hyperbolic conservation laws. *International Journal for Numerical Methods in Fluids*, 76(12):982–1003, 2014.

[12] D. Gottlieb and C.-W. Shu. On the Gibbs phenomenon and its resolution. *SIAM review*, 39 (4):644–668, 1997.

[13] D. Gottlieb, C.-W. Shu, A. Solomonoff, and H. Vandeven. On the Gibbs phenomenon I: Recovering exponential accuracy from the Fourier partial sum of a nonperiodic analytic function. *Journal of Computational and Applied Mathematics*, 43(1-2):81–98, 1992.

[14] N. K. Govindaraju, B. Lloyd, Y. Dotsenko, B. Smith, and J. Manferdelli. High performance discrete Fourier transforms on graphics processors. In *SC'08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, pages 1–12. Ieee, 2008.

[15] J. S. Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids: I. Time-domain solution of Maxwell's equations. *Journal of Computational Physics*, 181(1): 186–221, 2002.

[16] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[17] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral methods for time-dependent problems*, volume 21. Cambridge University Press, 2007.

[18] M. Hornikx, R. Waxler, and J. Forssén. The extended Fourier pseudospectral time-domain method for atmospheric sound propagation. *The Journal of the Acoustical Society of America*, 128(4):1632–1646, 2010.

[19] M. Hornikx, W. De Roeck, and W. Desmet. A multi-domain Fourier pseudospectral time-domain method for the linearized Euler equations. *Journal of Computational Physics*, 231 (14):4759–4774, 2012.

[20] M. Hornikx, T. Krijnen, and L. van Harten. openPSTD: The open source pseudospectral time-domain method for acoustic propagation. *Computer Physics Communications*, 203:298–308, 2016.

[21] X. Huang and X. Zhang. A Fourier pseudospectral method for some computational aeroacoustics problems. *International Journal of Aeroacoustics*, 5(3):279–294, 2006.

[22] Q. Huo Liu and G. Zhao. Review of PSTD methods for transient electromagnetics. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 17(3): 299–323, 2004.

[23] D. Huybrechs. On the Fourier extension of nonperiodic functions. *SIAM Journal on Numerical Analysis*, 47(6):4326–4355, 2010.

[24] A. Klöckner, T. Warburton, J. Bridge, and J. S. Hesthaven. Nodal discontinuous Galerkin methods on graphics processors. *Journal of Computational Physics*, 228(21):7863–7882, 2009.

[25] M. Kronbichler and K. Kormann. Fast matrix-free evaluation of discontinuous Galerkin finite element operators. *ACM Trans. Math. Softw.*, 45(3):29:1–29:40, Aug. 2019.

[26] L. Lambrecht, A. Lamert, W. Friederich, T. Möller, and M. S. Boxberg. A nodal discontinuous Galerkin approach to 3-D viscoelastic wave propagation in complex geological media.

Geophysical Journal International, 212(3):1570–1587, Mar. 2018.

[27] R. J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

[28] Q. H. Liu. The PSTD algorithm: A time-domain method requiring only two cells per wavelength. *Microwave and optical technology letters*, 15(3):158–165, 1997.

[29] Q. H. Liu. PML and PSTD algorithm for arbitrary lossy anisotropic media. *IEEE microwave and guided wave letters*, 9(2):48–50, 1999.

[30] D. Lockard and H. Atkins. Efficient implementations of the quadrature-free discontinuous Galerkin method. In *14th Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, Aug. 1999. doi: 10.2514/6.1999-3309.

[31] R. Mehra, N. Raghuvanshi, L. Savioja, M. C. Lin, and D. Manocha. An efficient GPU-based time domain solver for the acoustic wave equation. *Applied Acoustics*, 73(2):83–94, 2012.

[32] A. Modave, A. St-Cyr, W. A. Mulder, and T. Warburton. A nodal discontinuous Galerkin method for reverse-time migration on GPU clusters. *Geophysical Journal International*, 203 (2):1419–1435, 2015.

[33] A. Modave, A. St-Cyr, and T. Warburton. GPU performance analysis of a nodal discontinuous Galerkin method for acoustic and elastic models. *Computers & Geosciences*, 91:64–76, 2016.

[34] M. N'diaye. *On the study and development of high-order time integration schemes for ODEs applied to acoustic and electromagnetic wave propagation problems*. PhD thesis, Université de Pau et des pays de l'Adour, 2017.

[35] R. Pagán Muñoz and M. Hornikx. A hybrid PSTD/DG method to solve the linearized Euler equations: optimization and accuracy. In *22nd AIAA/CEAS Aeroacoustics Conference*, page 2718, 2016.

[36] R. Pagán Muñoz and M. Hornikx. Hybrid Fourier pseudospectral/discontinuous Galerkin time-domain method for wave propagation. *Journal of Computational Physics*, 348:416–432, 2017.

[37] C. Pan. Design of a windowless digital filter using FFT algorithm. Technical report, Stanford Linear Accelerator Center, 1993.

[38] C. Pan. Gibbs phenomenon suppression and optimal windowing for attenuation and Q measurements. Technical report, Stanford Linear Accelerator Center, 1993.

[39] C. Pan. Gibbs phenomenon removal and digital filtering directly through the fast Fourier transform. *IEEE Transactions on Signal Processing*, 49(2):444–448, 2001.

[40] R. B. Platte and A. Gelb. A hybrid Fourier–Chebyshev method for partial differential equations. *Journal of Scientific Computing*, 39:244–264, 2009.

[41] N. Raghuvanshi, B. Lloyd, N. Govindaraju, and M. C. Lin. Efficient numerical acoustic simulation on graphics processors using adaptive rectangular decomposition. In *Proceedings of the EAA Symposium on Auralization*, 2009.

[42] N. Raghuvanshi, R. Narain, and M. C. Lin. Efficient and accurate sound propagation using adaptive rectangular decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):789–801, 2009.

[43] K. R. Rao and P. Yip. *Discrete cosine transform: algorithms, advantages, applications*. Academic press, 2014.

[44] J. Shen. Efficient spectral-Galerkin method II. Direct solvers of second-and fourth-order equations using Chebyshev polynomials. *SIAM Journal on Scientific Computing*, 16(1):74–87, 1995.

[45] J. Shen, T. Tang, and L.-L. Wang. *Spectral methods: algorithms, analysis and applications*, volume 41. Springer Science & Business Media, 2011.

[46] S. Terrana, J. P. Vilotte, and L. Guillot. A spectral hybridizable discontinuous Galerkin method for elastic–acoustic wave propagation. *Geophysical Journal International*, 213(1):

574–602, Apr. 2018.

[47] I. Toulopoulos and J. A. Ekaterinaris. High-order discontinuous Galerkin discretizations for computational aeroacoustics in complex domains. *AIAA Journal*, 44(3):502–511, 2006.

[48] T. Toulorge and W. Desmet. CFL conditions for Runge–Kutta discontinuous Galerkin methods on triangular grids. *Journal of Computational Physics*, 230(12):4657–4678, 2011.

[49] L. N. Trefethen. *Spectral Methods in MATLAB*. Society for Industrial and Applied Mathematics, 2000.

[50] K. van der Sande, D. Appelö, and N. Albin. Fourier continuation discontinuous Galerkin methods for linear hyperbolic problems. *Communications on Applied Mathematics and Computation*, 5(4):1385–1405, 2023.

[51] H. Vandeven. Family of spectral filters for discontinuous problems. *Journal of Scientific Computing*, 6:159–192, 1991.

[52] H. Wang, I. Sihar, R. Pagán Muñoz, and M. Hornikx. Room acoustics modelling in the time-domain with the nodal discontinuous Galerkin method. *The Journal of the Acoustical Society of America*, 145(4):2650–2663, Apr. 2019.

[53] T. Warburton. An explicit construction of interpolation nodes on the simplex. *Journal of engineering mathematics*, 56:247–262, 2006.