High-order locally A-stable implicit schemes for linear ODEs

Hélène Barucq, Marc Duruflé, Mamadou N'Ddiaye

High-order locally A-stable implicit schemes for linear ODEs

Hélène Barucq[1,3], Marc Duruflé[1,2] and Mamadou N'Diaye[1,3]

December 19, 2019

Abstract

Accurate simulations of wave propagation in complex media like Earth subsurface can be performed with a reasonable computational burden by using hybrid meshes stuffing fine and coarse cells. Locally implicit time discretizations are then of great interest. They indeed allow using unconditionally stable schemes in the regions of computational domain covered by small cells. The receivable values of the time step are then increased which reduces the computational costs while limiting the dispersion effects. In this work we construct a method that combines optimized explicit schemes and implicit schemes to form locally implicit schemes for linear ODEs, including in particular semi-discretized wave problems that are considered herein for numerical experiments. Both the explicit and implicit schemes used are one-step methods constructed using their stability function. The stability function of the explicit schemes are computed by maximizing the time step that can be chosen. The implicit schemes used are unconditionally stable. The performance assessment we provide shows a very good level of accuracy for locally implicit schemes. It also shows that locally implicit scheme is a good compromise between purely explicit and purely implicit schemes in terms of computational time and memory usage.

1 Introduction

Simulation of wave propagation in complex domain is improved by using high-order space approximations (for example [1, 2, 3, 4, 5]) possibly coupled locally with low orders associated with locally refined meshes adapted to complex geometries. After space discretization, the obtained ODE can be discretized either with explicit or implicit time schemes. Explicit time schemes [6, 7] are very popular for providing integration process both cheap in memory and highly scalable. However, for stability purposes the time step is restricted by the CFL (Courant-Friedrichs-Lewy) condition which depends on the size of the smallest elements of the mesh and the order of the space discretization method. As a consequence, even few small elements can make

[1]Inria Centre de Recherche Bordeaux Sud-Ouest, Magique-3D E2S UPPA, 200 avenue de la vieille Tour, 33 405 Talence, FRANCE. email: helene.barucq@inria.fr, marc.durufle@inria.fr, mamadou.ndiaye@inria.fr

[2]Université de Bordeaux, Institut de Mathématiques de Bordeaux, Bordeaux, France. email: marc.durufle@inria.fr

[3]Université de Pau et des Pays de l'Adour, Laboratoire de Mathématiques et leurs Applications, Magique-3D Inria, avenue de l'université, 64012 Pau, France. email: helene.barucq@univ-pau.fr, mamadou.ndiaye@univ-pau.fr

the value of the global time step [6, 8]) so small that the computational cost becomes prohibitive.

Some research focusses on the construction of explicit schemes with a maximal CFL number [9, 10, 11, 12, 13, 14]. In these works, the obtained schemes are called optimal in the sense that they allow the use of a large time step compared to the classical explicit schemes. Other works, like in [15, 16, 17, 18], deal with the development of locally explicit time stepping methods in which we can choose different time steps while solving the problem. The underlying idea consists of dividing the computational domain into different zones distinguishing themselves from the size of the mesh elements. By this way, each zone has its own CFL and the time step is chosen zone by zone. This process allows to take small time steps precisely where the smallest elements are located and reduces by this way the overall computational time.

If you compare explicit schemes with implicit ones, [8, 7, 19] the latter are known to have better stability properties. In particular, some implicit schemes are unconditionally stable like those presented in [19, 20]. This means that there is no stability constraint for those schemes and the only constraint on the time step depends on the accuracy level. However, for 3D realistic problems, using only implicit methods is not always feasible since they are extremely memory consuming [21]. There is thus a clear interest in combining unconditionally stable schemes in a small region where an explicit scheme should require a very small time step, but not in the whole domain for large 3D like problems.

Then taking advantage of the good stability properties of implicit schemes with less memory consuming explicit schemes should be a good compromise to perform simulations of realistic problems.

In [22], a second-order locally implicit method was proposed for Maxwell's equations. The construction of the time method is based on the splitting of the mesh grid into coarse and fine areas. It is shown in [23] that this procedure does not change the second-order convergence of the obtained scheme. In [24] a locally implicit time integration scheme was developed for Maxwell's equations. Its construction is based on the Crank-Nicolson scheme (equivalent to the second order Padé scheme for linear ODE [25]) and the Runge-Kutta (RK) implicit mid-point rule (see [8]). The authors use the discontinuous Galerkin (DG) method (with central or upwind fluxes) with locally implicit schemes to discretize the Maxwell's equations and prove the stability of the fully discrete system under a CFL condition and get convergence results of order two as well.

Following [16] and [22], we propose the construction of locally implicit schemes with an arbitrary order of accuracy for linear ODEs. The general rules for their development share same ideas as in [16] except that we use implicit schemes in the region covered with a refined mesh. We validate the developed scheme by solving the acoustic wave equation with a hybridizable discontinuous Galerkin (HDG) formulation [21, 26, 27].

The remaining of this paper is structured as follows. First we give a general setting of the construction of time integration schemes using a stability function. Then we introduce unconditionally implicit and optimized explicit schemes in the next section. In section 3, we present the methodology we use to develop high-order locally implicit schemes. In the follow up section, we discuss the splitting technique used to divide the computational domain into coarse and fine regions. In section 6 numerical results for both 2-D and 3-D cases are displayed.

2 Construction of a numerical time integration schemes from a stability function

2.1 Stability function of a time scheme

In this section, we settle preliminary statements of time integration schemes for linear ODEs that we consider in this paper.

Let $\Omega \subset \mathbb{R}^n$ be an open set representing the computational domain and $X(t) \in \Omega$, $t \in [0, T]$, $T \in \mathbb{R}^+$ be the solution of the following ODE

$$\begin{cases} M_h \dfrac{dX(t)}{dt} + K_h X(t) = F(t), \\ X(0) = X_0. \end{cases} \tag{1}$$

The initial problem (1) is a standard one obtained after spatial discretization of a partial differential equation (PDE), where M_h is the mass matrix and K_h is the stiffness matrix. h denotes the mesh size, $F(t)$ is a source term obtained after discretizing the continuous source term in space and X_0 is the initial condition. In Chapter 3 of [21], it is detailed how this ODE is obtained in the case of wave equations. Let $t_0 < t_1 < \cdots < t_{N-1} < t_N$, $N \in \mathbb{N}$ be a uniform grid of the time interval $[0, T]$:

$$t_n = n\Delta t$$

where Δt is the time step. The analytical solution to (1) is given by

$$X(t_{n+1}) = \mathrm{e}^{\Delta t A} \left(X(t_n) + \int_0^{\Delta t} \mathrm{e}^{-uA} M_h^{-1} F(n\Delta t + u)\, \mathrm{d}u \right), \tag{2}$$

where $A = -M_h^{-1} K_h$.

The numerical solution can then be constructed by approximating the exponential, i.e. find R such that

$$\mathrm{e}^{\Delta t A} \approx R(\Delta t A). \tag{3}$$

Herein R is a rational function where both the numerator and denominator are polynomials of $\Delta t A$. The numerical process considered here aims at computing a sequence X_n, which is an accurate approximation of the analytical solution $X(t_n)$. The corresponding numerical scheme reads as:

$$X_{n+1} = R(\Delta t A) X_n + \widetilde{\phi}_n \tag{4}$$

where $\widetilde{\phi}_n$ is given as an approximation of the following quantity:

$$\widetilde{\phi}_n \approx R(\Delta t A) \int_0^{\Delta t} \mathrm{e}^{-uA} M_h^{-1} F(n\Delta t + u)\, \mathrm{d}u$$

The function R in (3) is the stability function of the corresponding numerical scheme.

Implicit schemes are known to have better stability properties than explicit ones [7, 8, 20]. In a previous work, we have developed high-order A-stable implicit one-step schemes. The construction of these schemes is based on the definition of their stability function which has the following form:

$$R(z) = \frac{N(z)}{D(z)}, \ \forall z \in \mathbb{C}^-,$$

where $N(z)$ and $D(z)$ are polynomials of z. The corresponding numerical scheme is implicit when the degree of D is greater than one otherwise the numerical scheme is explicit. We propose to construct locally implicit schemes involving implicit unconditionally stable Padé and Linear-SDIRK schemes which are in [25]. In the next section, we briefly introduce the construction of optimized CFL number explicit schemes used in this paper.

2.2 Optimized explicit time schemes for linear ODEs

Many efforts have been made on the construction of higher-order optimal CFL number explicit schemes. In [10, 11], the authors propose the modified equation technique to obtain high-order time schemes with an optimal CFL number for linear second order ODEs (i.e. ODEs of the type $y'' = f(y)$ where f is linear). In the proceedings [14], we propose optimized high-order explicit Runge-Kutta (ERK) Nyström methods for non-linear second-order ODEs. Other works addressed the optimization of the stability domain of ERK schemes [8, 13, 9, 12]. All those schemes are constructed for both linear and non-linear ODEs. The number of non-linear conditions to be satisfied to obtain the ERK coefficients [7] becomes very large, especially for higher orders. For this reason, most optimized ERK schemes available in the literature are at most of order 4. In this sub-section, we present a technique to build high-order optimized ERK schemes for linear ODEs. The construction is based on the computation of a stability function that leads to a maximal CFL number for the corresponding scheme.

Let us denote by $R_s^\ell(z)$, $s, \ell \in \mathbb{N}$, the stability function of a $m = s+\ell$ stages Linear-ERK schemes. It is defined by:

$$R_s^\ell(z) = 1 + z + \frac{z^2}{2!} + \cdots + \frac{z^s}{s!} + \alpha_{s+1} z^{s+1} + \cdots + \alpha_{s+\ell} z^{s+\ell}, \tag{5}$$

where $\alpha_i \in \mathbb{R}$, $i = s+1, \cdots, s+\ell$ are free parameters. The optimized Linear-ERK scheme with $m = s+\ell$ stages (denoted as ERK $s-\ell$) is given as

$$X_{n+1} = R_s^\ell(\Delta t A) X_n \tag{6}$$

We note that this explicit scheme has a order s by definition (for a linear ODE with $F(t) = 0$).

The parameters $\alpha_i \in \mathbb{R}$, $i = s+1, \cdots, s+l$ can be chosen such that they maximize the CFL number with the methodology described in [13]. The first step of this methodology consists in providing the spectrum of the linear operator $A = -M_h^{-1} K_h$ of the considered ODE (1). The linear operator depends on the solved equation, the physical environment, the mesh, the chosen FEM formulation (which can be continuous or discontinuous), the physical material... Herein, we have chosen to conduct the optimization for the HDG formulation for acoustic wave equation in homogenous material. We expect that the spectrum of matrix A in other configurations and wave-like equations is similar. In the Figure 1, we display the spectrum obtained for a 2-D regular mesh made of quadrilateral elements covering $[-4, 4]^2$ and $\mathbb{Q}_3$ HDG approximation (with Neumann boundary conditions). This is the typical spectrum obtained for a mesh containing elements of the same size. From this spectrum, we have defined a set noted $Cabane$, that includes this typical spectrum (see Figure 1).

The upper part of the envelope $Cabane$ is defined as a junction of three curves in the complex plane

$$Cabane^+ := \{it,\ t \in [0,1]\} \oplus \{i - t,\ t \in [0,1]\} \oplus \left\{ t - 2 + i\frac{t}{10}(14 - 4t),\ t \in [0,1] \right\} \tag{7}$$

The lower part of $Cabane$ is obtained by symmetry of its profile with respect to the real axis. The optimization problem then reads

$$\underset{\alpha_1, \alpha_2, \cdots, \alpha_\ell}{\arg\max}\ \mathbf{cfl}(\alpha_1, \alpha_2, \cdots, \alpha_\ell) \tag{8}$$

where cfl is the CFL number defined as

$$\mathbf{cfl}(\alpha_1, \alpha_2, \cdots, \alpha_\ell) = \max\{\Delta t \text{ such that } \Delta t \times Cabane \subset \mathcal{S}(\alpha_1, \alpha_2, \cdots, \alpha_\ell)\} \tag{9}$$

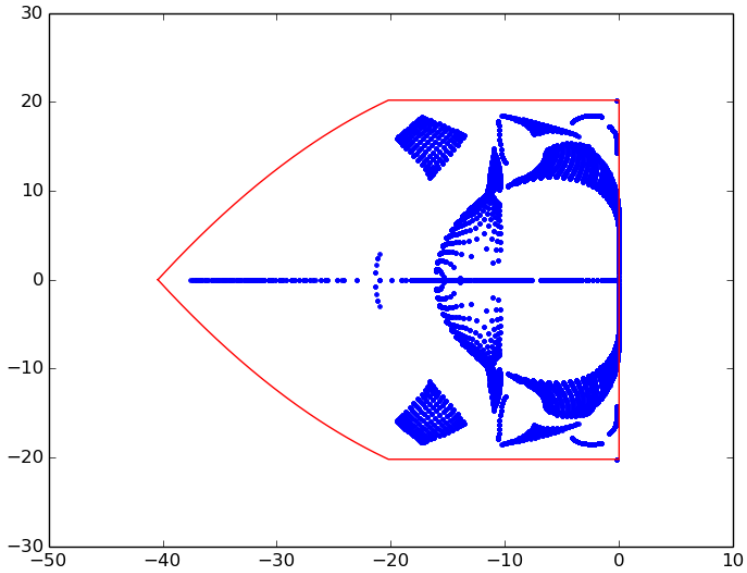


Figure 1: Spectrum of $A = -M_h^{-1}K_h$ (blue points are eigenvalues of A) in the complex plane included in a set $\alpha \times Cabane$ (inside the red polygon) obtained when solving the acoustic wave equation with HDG formulation and $\mathbb{Q}_3$ polynomials. The computational domain is a square $[-4,4]^2$ of 15×15 elements. Neumann condition is set on the boundary of the square.

where $\mathcal{S}(\alpha_1, \alpha_2, \cdots, \alpha_\ell)$ is the stability region of the corresponding Linear-ERK scheme given by:

$$\mathcal{S}(\alpha_1, \alpha_2, \cdots, \alpha_\ell) = \{z \in \mathbb{C},\ |R_s^\ell(z)| \leq 1\}.$$

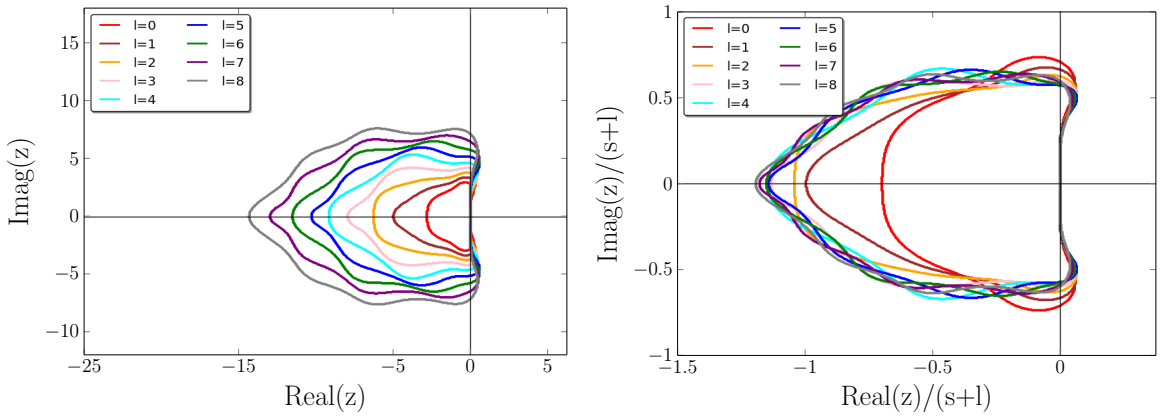
If the spectrum of the operator A (denoted by $sp(A)$) has exactly the same shape as the one defined by $Cabane$, the final time step should satisfy

$$\Delta t \leq \frac{\textbf{cfl}(\alpha_1, \alpha_2, \cdots, \alpha_\ell)}{\max_{\lambda \in sp(A)} |Im(\lambda)|} \tag{10}$$

to get a stable solution. Unfortunately, this is not usually the case and the maximal time step is given as

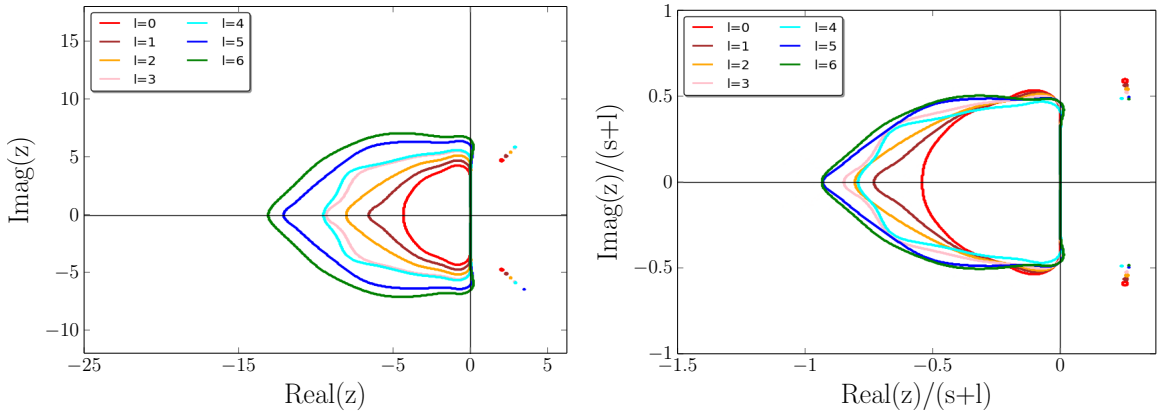
$$\Delta t = \max \{\Delta t \text{ such that } \lambda \Delta t \in \mathcal{S}(\alpha_1, \alpha_2, \cdots, \alpha_\ell)\ \forall \lambda \in sp(A)\}. \tag{11}$$

Note that in [13] the authors use the true spectrum of the operator and look for the maximal time step Δt that satisfies the stability constraint. After optimization the obtained value Δt represents the maximal time step such that $z = \lambda \Delta t \in \mathcal{S}$ for all $\lambda \in sp(A)$. Here, we consider a normalized envelope $Cabane$, and for the operator A the maximal time step is obtained from (11). The stability domains of the different schemes of order 4 and 8 are shown in the Figure 2. We show in the Figure 2(b), 2(d) the stability region taking into account the number of stages for each scheme. For this reason we have chosen $Real(z)/(s+l)$ and $Imag(z)/(s+l)$ in the x and y coordinate respectively. We observe that by increasing the number of additional stages l, the stability region tends to the desired shape $Cabane$. The optimal coefficients and CFL numbers obtained for Linear ERK schemes of order 2, 4, 6 and 8 are given in chapter 7 of [21].



(a) Stability region

(b) Stability region with the complexity



(c) Stability region

(d) Stability region with the complexity

Figure 2: Stability region of the Linear-ERK schemes of order 4 (top) and 8 (bottom) for different values of ℓ.

3 Construction of locally implicit time schemes

For the construction of locally implicit time schemes, we have adapted the procedure described in [16] for optimized explicit schemes. Let us note $A = -M_h^{-1}K_h$ and $y'(t) = \dfrac{dX(t)}{dt}$ in (1). We consider the following ODE

$$y'(t) = Ay(t) + F(t). \tag{12}$$

where A is a linear operator and y is the unknown vector. We consider the following splitting

$$y(t) = y^{(c)}(t) + y^{(f)}(t) \tag{13}$$

with

$$y^{(c)} = (I - P)y(t), \tag{14a}$$

$$y^{(f)} = Py(t), \tag{14b}$$

where $y^{(c)}(t)$ corresponds to the solution in the coarse part of the mesh and $y^{(f)}(t)$ corresponds to the solution in the fine part. P is a diagonal matrix with entries $P_{i,i} = 1$ if i corresponds to a degree of freedom in the fine area of the mesh and 0 elsewhere. After integration of (12) over the interval $[t_n, t_n + \xi\Delta t]$ where ξ is real between 0 and 1, we obtain the following expression of $y(t_n + \xi\Delta t)$:

$$\begin{aligned} y(t_n + \xi\Delta t) = y(t_n) + \underbrace{\int_{t_n}^{t_n+\xi\Delta t} A(I-P)y(t)dt + \overbrace{\int_{t_n}^{t_n+\xi\Delta t} (I-P)F(t)dt}^{\text{source term}}}_{\text{Coarse part}} \\ + \underbrace{\int_{t_n}^{t_n+\xi\Delta t} APy(t)dt + \overbrace{\int_{t_n}^{t_n+\xi\Delta t} PF(t)dt}^{\text{source term}}}_{\text{Fine part}} \end{aligned} \tag{15}$$

In the coarse part we apply an explicit scheme while an implicit scheme will be used in the fine part of the mesh.

3.1 Treatment of the coarse part

To treat the coarse part we first approximate the integral term containing $y(t)$ using an s-point quadrature method. Let c_i be quadrature points and b_i the corresponding weights. We obtain

$$\int_{t_n}^{t_n+\xi\Delta t} A(I-P)y(t)dt \approx \xi\Delta t A(I-P)\sum_{i=1}^{s} b_i\, y(t_n + c_i\xi\Delta t). \tag{16}$$

We replace $y(t_n + c_i\xi\Delta t)$ by its Taylor expansion of order $r-1$ to get

$$\int_{t_n}^{t_n+\xi\Delta t} A(I-P)y(t)dt \approx \xi\Delta t A(I-P)\sum_{i=1}^{s} b_i \sum_{j=0}^{r-1} \frac{(c_i\xi\Delta t)^j}{j!} y^{(j)}(t_n). \tag{17}$$

Then, we require that the quadrature method is of order $r-1$ at least. So we can use

$$\sum_{i=1}^{s} b_i c_i^j = \frac{1}{j+1}, \quad j = 0, \cdots, r-1 \tag{18}$$

we then obtain

$$\int_{t_n}^{t_n+\xi\Delta t} A(I-P)y(t)dt \approx \xi\Delta t A(I-P)\sum_{j=0}^{r-1}\frac{(\xi\Delta t)^j}{(j+1)!}y^{(j)}(t_n). \tag{19}$$

To handle the derivative of y in the previous sum, we differentiate the ODE (12) $r-1$ times and replace it into the equation to obtain

$$\begin{aligned}&\int_{t_n}^{t_n+\xi\Delta t} A(I-P)y(t)dt\\ &\approx \xi\Delta t A(I-P)\sum_{j=0}^{r-1}\frac{(\xi\Delta t)^j}{(j+1)!}\left(A^j y(t_n)+\sum_{\ell=1}^{j}A^{j-\ell}F^{(\ell-1)}(t_n)\right).\end{aligned}$$

From here, to optimize the CFL number of the explicit scheme, we can add extra terms beyond the term of the Taylor expansion of order $r-1$ without changing the order of accuracy. In this paper, we use the following expression for the stability function $R(z)$ of the chosen explicit scheme ($m+1=s+\ell$):

$$R(z)=\alpha_0+\alpha_1 z+\alpha_r z^r+\cdots+\alpha_{m+1}z^{m+1},\ m\geq r.$$

In order to coincide with explicit optimized schemes, we add extra-terms to obtain:

$$\begin{aligned}\int_{t_n}^{t_n+\xi\Delta t} A(I-P)y(t)dt &\approx \xi\Delta t A(I-P)\left(\sum_{j=0}^{r-1}\frac{(\xi\Delta t)^j}{(j+1)!}A^j y(t_n)+\sum_{j=r}^{m}\alpha_{j+1}(\xi\Delta t)^j A^j y(t_n)\right)\\ &+\xi\Delta t A(I-P)\left(\sum_{j=0}^{r-1}\frac{(\xi\Delta t)^j}{(j+1)!}\sum_{\ell=1}^{j}A^{j-\ell}F^{(\ell-1)}(t_n)+\sum_{j=r}^{m}\alpha_{j+1}(\xi\Delta t)^j\sum_{\ell=1}^{j}A^{j-\ell}F^{(\ell-1)}(t_n)\right)\end{aligned} \tag{20}$$

that can be written as

$$\begin{aligned}\int_{t_n}^{t_n+\xi\Delta t} A(I-P)y(t)dt \ &\approx \xi\Delta t A(I-P)\left(\sum_{j=0}^{m}\alpha_{j+1}(\xi\Delta t)^j A^j y(t_n)\right)\\ &+\xi\Delta t A(I-P)\left(\sum_{j=0}^{m}\alpha_{j+1}(\xi\Delta t)^j\sum_{\ell=1}^{j}A^{j-\ell}F^{(\ell-1)}(t_n)\right)\end{aligned} \tag{21}$$

since

$$\alpha_j=\frac{1}{j!},\ j\leq r.$$

This is due to the fact that the chosen explicit scheme is of order r. We then replace F by its interpolation with quadrature points c_i (denoted by Q):

$$F(t_n+\xi\Delta t)\ \approx\ Q(t_n+\xi\Delta t). \tag{22}$$

Then the interpolation polynomial Q reads

$$Q(t_n+\xi\Delta t)=\widetilde{Q}(\xi)=\sum_{i=1}^{s}F_i\,\widetilde{\varphi}_i(\xi), \tag{23}$$

with

$$F_i=F(t_n+c_i\Delta t)$$

and the basis functions $\widetilde{\varphi}_i$ valued between 0 and 1 are given by

$$\widetilde{\varphi}_i(\xi) = \frac{\prod_{j\neq i} \xi - c_j}{\prod_{j\neq i} c_i - c_j}.$$

We introduce the following notation

$$\widetilde{w}_j \;=\; A^j y(t_n) + \sum_{\ell=1}^{j} A^{j-\ell} Q^{(\ell-1)}(t_n), \tag{24}$$

to end up with a simplified expression

$$\int_{t_n}^{t_n+\xi\Delta t} A(I-P)y(t)dt \;\approx\; \xi\Delta t A(I-P)\left(\sum_{j=0}^{m} \alpha_{j+1}(\xi\Delta t)^j \widetilde{w}_j\right) \tag{25}$$

We then use the following relation

$$Q^{(\ell)}(t_n) = \frac{1}{\Delta t^\ell}\,\widetilde{Q}^{(\ell)}(0).$$

to compute $Q^{(\ell-1)}(t_n)$ as a linear combination of F_i:

$$Q^{(\ell-1)}(t_n) = \sum_{i=1}^{s} \frac{\widetilde{\varphi}_i^{(\ell-1)}(0)}{(\Delta t)^{\ell-1}} F_i.$$

The coefficients

$$D_{i,\ell} = \frac{\widetilde{\varphi}_i^{(\ell)}(0)}{(\Delta t)^\ell}$$

can be pre-computed prior to time iterations. In practice, we will compute and store the vectors

$$\zeta_j = \alpha_{j+1} A(I-P)\widetilde{w}_j. \tag{26}$$

An optimal way to do that is provided by the Algorithm 1.

Algorithm 1 Computation of ζ_j (26)

Coefficients $D_{i,l}$ are pre-computed as
$D_{i,\ell} = \dfrac{\widetilde{\varphi}_i^{(\ell)}(0)}{(\Delta t)^\ell}$
for $i = 1 \ldots s$ **do**
  compute $F_i = F(t_n + c_i \Delta t)$
end for
$w = y_n$
for $j = 0 \ldots m$ **do**
  compute $z = A\,(I-P)\,w$ and $z_p = A\,P\,w$
  $\zeta_j = \alpha_{j+1}\, z$
  compute $Q^{(j)} = \sum_{i=1}^{s} D_{i,j} F_i$
  $w = z + z_p + Q^{(j)}$
end for

Now we treat the second-integral in (15) corresponding to the source term in the coarse part. It is approximated as follows:

$$\int_{t_n}^{t_n+\xi\Delta t} (I-P)F(t)dt \approx (I-P)\xi\Delta t \sum_{i=1}^{s} b_i Q(t_n + c_i \xi \Delta t). \tag{27}$$

For degrees of freedom (dofs) that are far from the fine region (i.e. dofs belonging to elements that are not adjacent or inside the fine part of the mesh), the corresponding row of AP is zero, and $P_{i,i} = 0$. As a result the contribution of fine part in (15) can be dropped. The vector y_{n+1} (with $\xi = 1$) is computed as follows for these degrees of freedom:

$$y_{n+1} = y_n + \Delta t A(I-P)\sum_{j=0}^{m} \alpha_{j+1}\Delta t^j \widetilde{w}_j + \Delta t \sum_{i=1}^{s} b_i F_i$$

or equivalently

$$y_{n+1} = y_n + \Delta t \sum_{j=0}^{m} \Delta t^j \zeta_j + \Delta t \sum_{i=1}^{s} b_i F_i \tag{28}$$

We see that this expression is a linear combination of the vectors F_i and ζ_j computed in the Algorithm 1.

Remark If the fine part is void (i.e. $P = 0$), the obtained explicit scheme can be written in the usual form given in [21]:

$$y_{n+1} = \sum_{j=0}^{m+1} \alpha_j \Delta t^j A^j y_n + \Delta t \sum_{k=0}^{m} \Delta t^k A^k \sum_{i=1}^{s} \omega_i^k F_i$$

where

$$\omega_i^k = \begin{cases} \displaystyle\sum_{\ell=1}^{m+1-k} \alpha_{k+\ell}\, \widetilde{\varphi}_i^{(\ell-1)}(0), \text{ if } k > 0 \\ b_i, \text{ if } k = 0 \end{cases}$$

The result is obtained by introducing $k = j + 1 - \ell$.

Remark In formula (28), we can observe that the coarse part is advanced by using Hörner's algorithm (instead of the stable algorithm given in [21]). This algorithm is necessary to advance the fine part efficiently. That is the main reason why we preferred the optimized explicit schemes with a small number of additional stages, such that the Hörner's algorithm should not deteriorate the accuracy.

3.2 Treatment of the coarse part combined with the fine part

We recall that Q (23) denotes the polynomial that interpolates the source function F. We introduce $\hat{Q}$ as the anti-derivative of Q. Then the integral of the source term in the coarse region (that will be in contact with the fine region) can be approximated as follows:

$$\int_{t_n}^{t_n+\xi\Delta t} (I-P)F(t)dt \approx (I-P)\left(\hat{Q}(t_n+\xi\Delta t) - \hat{Q}(t_n)\right). \tag{29}$$

Using the equation (29) and the approximation (25) in the coarse region, the equation (15) becomes:

$$\begin{aligned} y(t_n+\xi\Delta t) \approx y_n + A(I-P)\sum_{j=0}^{m} \alpha_{j+1}(\xi\Delta t)^{j+1}\widetilde{w}_j + (I-P)\left(\hat{Q}(t_n+\xi\Delta t) - \hat{Q}(t_n)\right) \\ + \int_{t_n}^{t_n+\xi\Delta t} APy(t) + PF(t)dt \end{aligned}$$

We introduce the variable $\tau = \xi \Delta t$ to obtain

$$y(t_n+\tau) = y_n + A(I-P)\sum_{j=0}^{m} \alpha_{j+1}\,\tau^{j+1}\widetilde{w}_j + (I-P)\left(\hat{Q}(t_n+\tau) - \hat{Q}(t_n)\right) + \int_{t_n}^{t_n+\tau} APy(t) + PF(t)dt \tag{30}$$

Let $\widetilde{y}$ be defined as

$$\widetilde{y}(\tau) = y(t_n+\tau)$$

Like in [16], we differentiate (30) with respect to τ to obtain the following ODE:

$$\frac{d\widetilde{y}(\tau)}{d\tau} = \overbrace{A(I-P)\sum_{j=0}^{m}(j+1)\alpha_{j+1}\,\tau^{j}\widetilde{w}_j + (I-P)Q(t_n+\tau) + PF(t_n+\tau)}^{\text{updated source term}} + AP\widetilde{y}(\tau) \tag{31}$$

The equation (31) represents the ODE that has to be solved in the fine region with an updated source term. This updated source term is known since its comes from the explicit scheme used in the coarse region and the source term set in the fine region. We have made the choice of approximating F with Q in this term as well. With this approach, we obtain the following ODE:

$$\frac{d\widetilde{y}(\tau)}{d\tau} = AP\widetilde{y}(\tau) + \widetilde{F}(\tau) \tag{32}$$

where

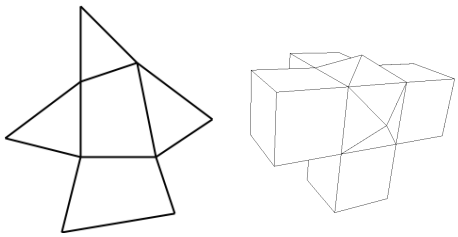
$$\widetilde{F}(\tau) = \sum_{j=0}^{m}(j+1)\,\tau^{j}\,\zeta_j + Q(t_n+\tau) = \sum_{j=0}^{m}(j+1)\,\tau^{j}\,\zeta_j + \sum_{i=1}^{s}\widetilde{\varphi}_i\left(\frac{\tau}{\Delta t}\right)F_i \tag{33}$$

The ODE (32) can be solved using either explicit schemes with small time step as in [16] or implicit schemes as we propose here. We use A-stable implicit schemes developed in [25] to solve (32), to obtain locally implicit schemes. By this way, the most stiff part of the computational domain becomes a CFL-free region since we use A-stable time integration schemes. This allows increasing the admissible values of time step which contributes to reduce the computational costs.

Remark The ODE (32) is solved for close degrees of freedom, i.e. degrees of freedom that belong to an element inside the fine region or adjacent to the fine region. These close degrees of freedom correspond to non-null rows of operator AP. We use Algorithm 2 to compute y_{n+1} from y_n. In Algorithm 2, the tasks 1 and 2 can be performed in parallel, but the cost of these two tasks are very different since the task 1 consists of a linear combination while task 2 will involve the solution of several linear systems.

Algorithm 2 Computation of y_{n+1}

compute vectors F_i and ζ_j with algorithm 1
Task 1 : compute y_{n+1} for far degrees of freedom with formula (28)
Task 2 : compute y_{n+1} for close degrees of freedom by solving the ODE (32) with an implicit scheme



(a) 2D (b) 3D

Figure 3: Example of 2D and 3D small meshes used to evaluate the local time step of each element.

Remark The ODE (32) holds true for far degrees of freedom. For these degrees of freedom, it reduces fo $d\widetilde{y}(\tau)/d\tau = \widetilde{F}(\tau)$ since the correspoding rows of AP are null. This ODE can be integrated exactly, and we obtain the expression (28) for y_{n+1}.

Remark For HDG formulation, the linear system to be solved involves degrees of freedom associated with the unknown λ for edges (faces in 3-D) of the fine region only. Then the unknowns u and v (for acoustics) are reconstructed in the close region (fine region + adjacent elements).

4 Splitting based on the local time step

Let $\Omega \subset \mathbb{R}^n$ be an open set that represents the computational domain. The domain Ω is assumed to be meshed with elements K_i such that

$$\Omega = \bigcup K_i.$$

We consider the following semi-discrete ODE for a sub-mesh Ω_i of the computational domain Ω:

$$\begin{cases} M_h \dfrac{dy(t)}{dt} + K_h y(t) = F(t) \quad t \in (0, T] \\ y(0) = y_0 \end{cases} \tag{34}$$

The sub-mesh Ω_i comprises the element K_i and its adjacent elements (elements that share an edge in 2D (a face in 3D) with the element K_i). In the Figure 3, we have represented examples of a sub-mesh Ω_i in 2D and 3D in the case of an hybrid mesh. In (34), M_h is the local mass matrix and K_h is the local stiffness matrix. We denote by $A_i = M_h^{-1} K_h$ the product of the inverse of the local mass matrix and the local stiffness matrix of the sub-mesh Ω_i. The local time step Δt_i is then given as

$$\Delta t_i = \max \left\{ \Delta t \text{ such that } |R_s^\ell(\lambda \Delta t)| \leq 1, \ \forall \lambda \in sp(A_i) \right\}$$

This local time step is expensive to compute, since we need to compute all the eigenvalues of A_i and then compute the maximum. We preferred to apply the algorithm 3 to evaluate an approximation of Δt_i:
In this algorithm , we find the eigenvalues λ with the largest modulus $|R_s^\ell(\lambda \Delta t_{\text{nom}})|$, by using Slepc software [28]. This software allows the user to provide its own function that compares two eigenvalues (through function EPSSetEigenvalueComparison).

Once the local time steps Δt_i are approximated, the splitting between fine and coarse region is done by comparing these local time steps to a reference time step

Algorithm 3 Computation of Δt_i

Choose a nominal time step Δt_{nom} large enough.
Find the largest eigenvalue $\lambda_{\max}$ of A_i. Eigenvalues are sorted by increasing modulus $|R_s^\ell(\lambda \Delta t_{\text{nom}})|$.
Apply the bisection method to find $\Delta t_i \in [0, \Delta t_{\text{nom}}]$ such that $|R(\lambda_{\max} \Delta t_i)| = 1$.

Δt_{ref}:

$$\begin{cases} \text{if } \Delta t_i \leq \Delta t_{ref},\ K_i \in \Omega^{fine} = \text{ fine region} \\ \text{if } \Delta t_i > \Delta t_{ref},\ K_i \in \Omega^{coarse} = \text{ coarse region} \end{cases} \tag{35}$$

5 Accuracy of the locally implicit methods

In this section, we aim at showing that the locally implicit methods obtained from the combination of the Linear-ERK schemes of order r with implicit schemes of the same order (Padé or Linear-SDIRK schemes), are of order r. This result follows from the result obtained for the locally time stepping method presented in [16].

By construction, the algorithm used to advance far degrees of freedom with formula (28) is of order r, i.e. the consistency error of the formula is in $O(\Delta t^{r+1})$. The algorithm used to advance close degrees of freedom (an implicit scheme of order r) will also provide a local error in $O(\Delta t^{r+1})$. So it seems clear that the global scheme has a local error in $O(\Delta t^{r+1})$. As a result, if the global scheme is stable, the global error should be in $O(\Delta t^r)$. We do not have any proof for the stability, but we are rather convinced that the CFL of the global scheme will be controlled by the CFL of the coarse part since the ODE of the close region is advanced with an A-stable scheme.

6 Numerical results

This section provides numerical results in order to validate the locally implicit schemes implemented and to evaluate the efficiency of the schemes. The schemes have been implemented in the finite element C++ code Montjoie [5] (`http://montjoie.gforge.inria.fr/`) for HDG formulation.

We first consider the acoustic wave equation. The scalar field u and vectorial field v depend on the space $\mathbf{x}$ and the time t and are solutions to the following boundary value problem:

$$\begin{cases} \rho\, \partial_t u - \operatorname{div} v = f, \quad \forall (x,t) \in \Omega \times \mathbb{R}^+ \\ \mu^{-1} \partial_t v - \nabla u = 0, \quad \forall (x,t) \in \Omega \times \mathbb{R}^+ \\ u(x,0) = \partial_t u(x,0) = 0, \quad \forall x \in \Omega \quad \text{(null initial conditions)} \\ u = f_D, \quad x \in \Gamma_D \quad \text{(Dirichlet condition)} \\ \mu \partial_n u = f_N, \quad x \in \Gamma_N \quad \text{(Neumann condition)} \\ \mu \partial_n u + \sqrt{\rho \mu}\, \partial_t u = f_A, \quad x \in \Gamma_A \quad \text{(Absorbing condition)} \end{cases} \tag{36}$$

where Ω is the computational domain. Γ_D, Γ_N and Γ_A are the boundaries associated respectively with Dirichlet, Neumann and absorbing boundary condition. n is the normal vector outgoing to the considered boundary, ρ and μ are physical indexes, which are piecewise constant. f_D, f_N and f_A are given source functions. Hybrid Discontinuous Galerkin formulation is used for the space discretization of these

equations (see [26]). Schur complement technique is performed such that the linear system to be solved has a reduced size (see details in [21]). This linear system is factorized and solved by calling Mumps routines [29]. For implicit schemes, we can use Padé schemes, denoted as Padé r where r is the order. We can also use Linear-Sdirk schemes, denoted as LSDIRK $r - \ell$ where r is the order and ℓ the number of additional stages. For these schemes, the total number of stages is equal to $s + \ell$ where $s = r - 1$.

6.1 Time convergence

In this sub-section, we consider a square $[-5, 5]^2$ with homogeneous indexes $\rho = \mu = 1$. The inner square $[-1, 1]^2$ is refined (see figure 4) in order to have a fine resolution where the source is not null. This inner square will correspond to the fine region (i.e. implicit part) where an implicit scheme will be used. We choose the following source

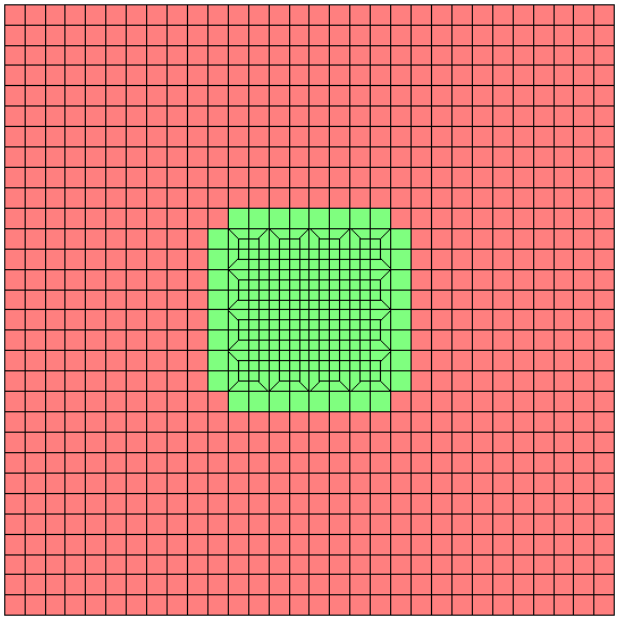


Figure 4: Mesh used for time convergence (Green zone : fine region, Red zone : coarse region).

term (gaussian in space)

$$f(x, y, t) = \beta \exp(-\alpha(x^2 + y^2))h(t)$$

where

$$r_0 = 1, \quad \alpha = \frac{\log(10^6)}{r_0^2}, \quad \beta = \sqrt{\frac{\alpha}{\pi}}$$

The time pulse $h(t)$ is chosen as a derivative of a gaussian

$$h(t) = -(f_0 t - 1) \exp\left(-(f_0 t - 1)^2 \pi^2\right)$$

where f_0 is the the central frequency (we choose $f_0 = 1$). Homogeneous Dirichlet conditions are set on the boundary of the square. The solution obtained for $t = 4$ and $t = 20$ is plotted in figure 5. In this sub-section, we study the convergence

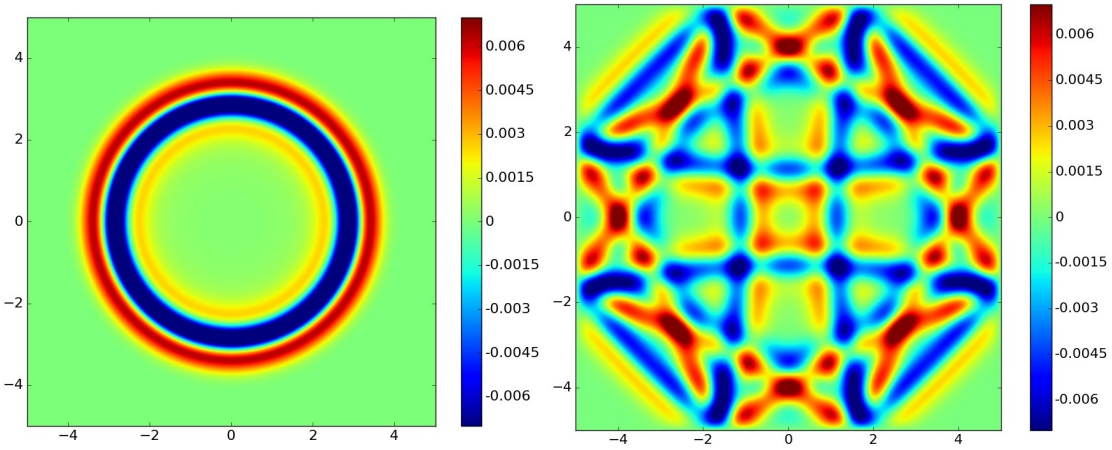


Figure 5: Solution for the square at time $t = 4$ and $t = 20$.

in time only of the local implicit scheme. As a result, the space discretization is fixed with the mesh displayed in figure 4 and $\mathbb{Q}_{12}$ space approximation such that the space error is close to 10^{-12} (due to round-off errors in double precision). The reference solution is computed with an explicit eighth-order Runge-Kutta scheme ($\Delta t = 0.001$), the relative error is computed for the final time $t = 20$. In figure 6, 7

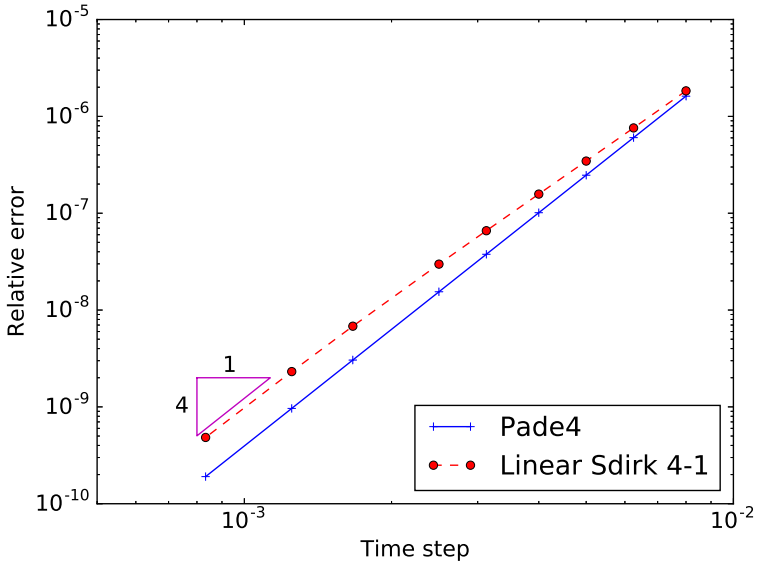


Figure 6: Relative L^2 error versus Δt for ERK 4-2 and Padé 4 (or Linear SDIRK 4-1).

and 8, we can observe the convergence in time of local implicit schemes. Linear ERK schemes are used for the explicit part (respectively ERK 4-2, ERK 6-2 and ERK 8-2). Padé schemes or Linear SDIRK schemes with one additional stage are used for the implicit part. For fourth-order schemes (cf. figure 6, we observe a nice convergence in $O(\Delta t^4)$ with either Padé scheme or Linear SDIRK for the implicit part. For sixth-order schemes (cf. figure 7), we obtain a convergence in $O(\Delta t^6)$ when Padé scheme is used. The stagnation below 10^{-12} is due to round off errors (double precision is used). When Linear Sdirk scheme is used, the convergence seems slower, the relative error

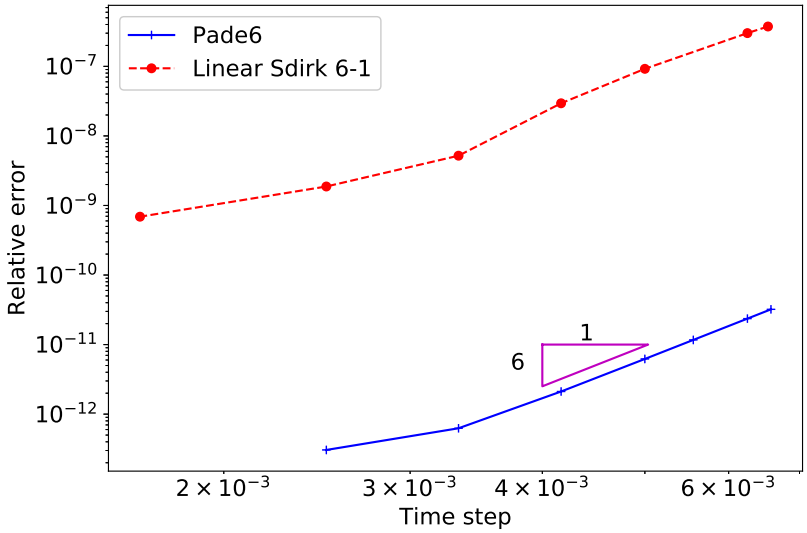


Figure 7: Relative L^2 error versus Δt for ERK 6-2 and Padé 6 (or Linear SDIRK 6-1).

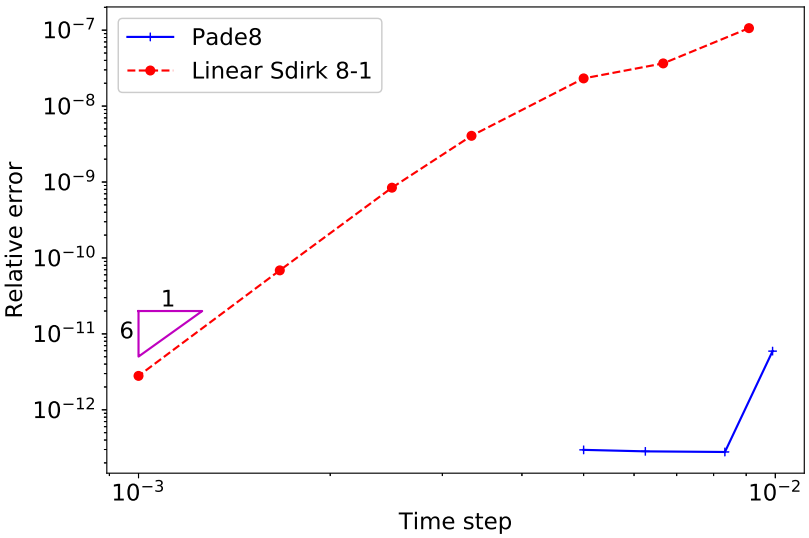


Figure 8: Relative L^2 error versus Δt for ERK 8-2 and Padé 8 (or Linear SDIRK 8-1).

is much larger than with Padé schemes. For eighth-order, Padé scheme will probably provide a nice convergence in $O(\Delta t^8)$ but it cannot be observed since the error is already in 10^{-12} for the maximal time step. Linear Sdirk scheme provides a reduced convergence (here we measure a sixth-order convergence in figure 8).

6.2 Space-time convergence

In this sub-section, the convergence both in space and time is studied We consider the same case than in the previous section. Only the mesh size will vary, an example

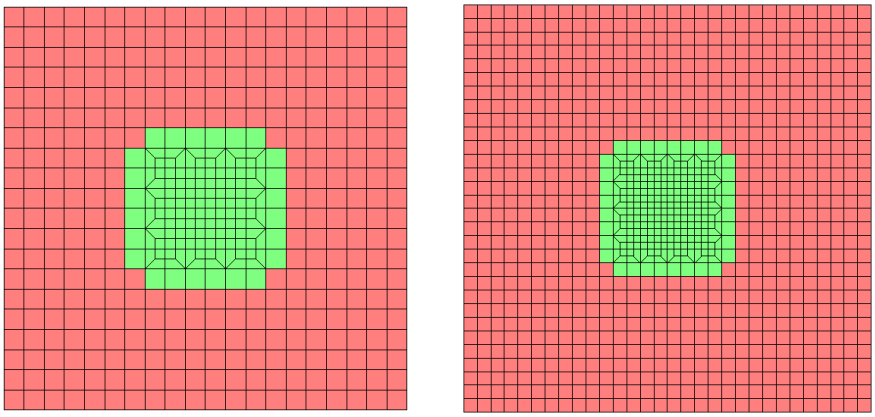


Figure 9: Example of two meshes used for space-time convergence.

of two meshes is given in figure 9. The reference solution is computed with $\mathbb{Q}_{12}$ approximation and 40 elements in x-direction with an explicit eighth-order Runge-Kutta scheme ($\Delta t = 0.001$).

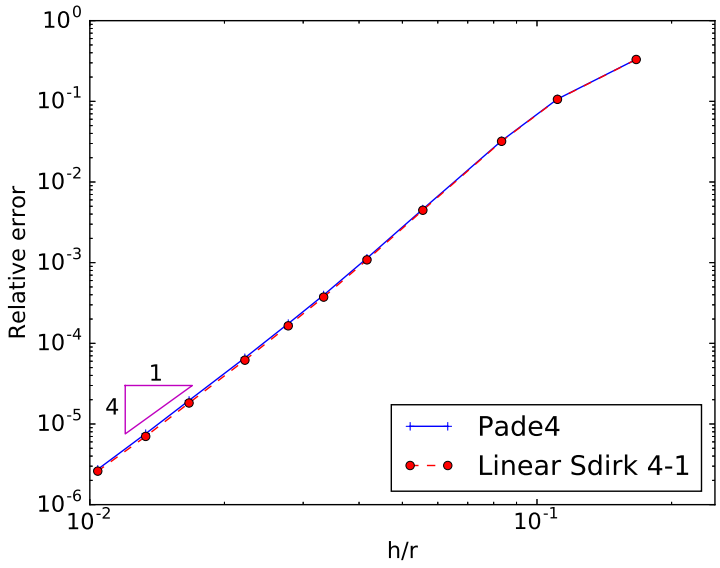


Figure 10: Space-time convergence with $\mathbb{Q}_3$, ERK4-2 and Pade 4 (or Linear SDIRK 4-1).

We consider $\mathbb{Q}_3, \mathbb{Q}_5$ and $\mathbb{Q}_7$ approximation in order to obtain a convergence in $O(h^4), O(h^6)$ and $O(h^8)$ where h is the mesh size. The time step is chosen close to the maximal time step, namely we choose

$$\Delta t = \frac{h}{3.5}, \quad \Delta t = \frac{h}{10}, \quad \Delta t = \frac{h}{12.5},$$

for respectively $\mathbb{Q}_3, \mathbb{Q}_5$ and $\mathbb{Q}_7$ space approximation. In the table 1, the measured maximal time step for this case is given for the locally implicit schemes. This maximal time step is found manually by checking that the solution is stable until $T = 10000$. It is compared with the time step measured for a regular square (with only

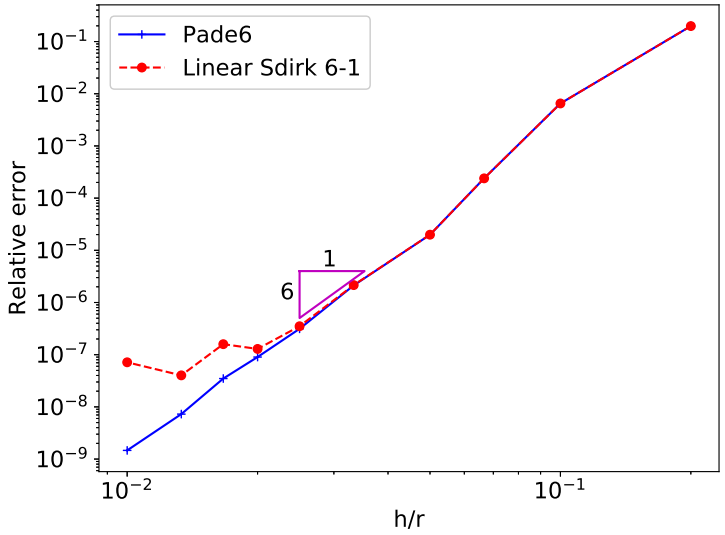


Figure 11: Space-time convergence with $\mathbb{Q}_5$, ERK6-2 and Pade 6 (or Linear SDIRK 6-1).

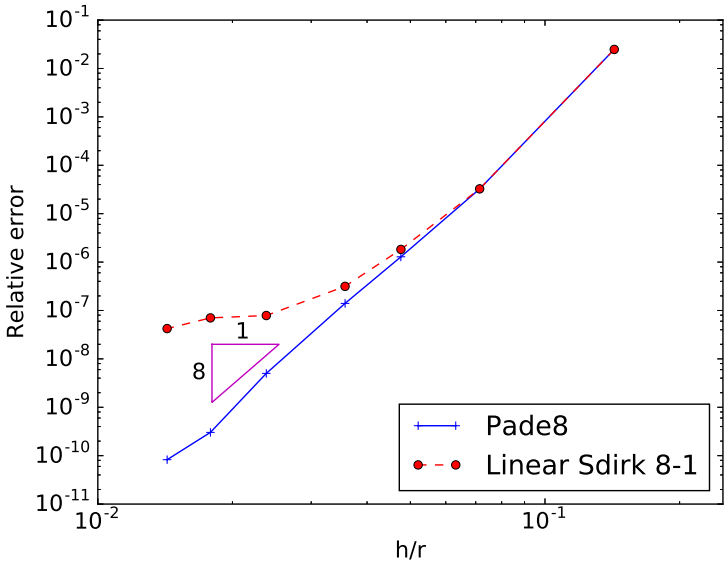


Figure 12: Space-time convergence with $\mathbb{Q}_7$, ERK8-2 and Pade 8 (or Linear SDIRK 8-1).

ERK scheme), in order to check if the local implicit scheme does not deteriorate the stability condition due to the coarse part of the mesh. It is observed that the maximal time step is slightly smaller with locally implicit schemes from order 6. In this table, the maximal time step computed by Slepc (see section 4) is also given to show that this approximation is fairly good, sometimes this procedure gives exactly the maximal time step (here for order 4 and 8). In figures 10, 11 and 12, we plot the relative L^2 error versus the mesh size h for fourth-order, sixth-order and eighth-order schemes. When a local implicit scheme is used with Padé scheme for the implicit part, we observe a nice convergence in $O(h^{r+1})$ as expected (where r is the order of

Table 1: Maximal time step for the square will local refinement for local implicit schemes, without refinement for explicit scheme. Last column gives the estimate computed with Slepc.

Order	h	Local LSDIRK	Local Padé	Explicit	Slepc
4 ($\mathbb{Q}_3$)	0.25	0.0727	0.0727	0.0727	0.0727
6 ($\mathbb{Q}_3$)	0.5	0.0512	0.0516	0.0543	0.0567
8 ($\mathbb{Q}_7$)	0.5	0.0410	0.0418	0.0428	0.0428

space approximation). When Linear Sdirk scheme is used as implicit, we observe a nice convergence for fourth-order scheme but a deteriorated convergence for higher order schemes. This deterioration could be due to spurious oscillations that have been previously observed in [25]. Nevertheless this issue appears for small values of h and the relative L^2 error is satisfactory.

6.3 2-D case

In this section, we consider the scattering by a magnetron (see figure 13). The

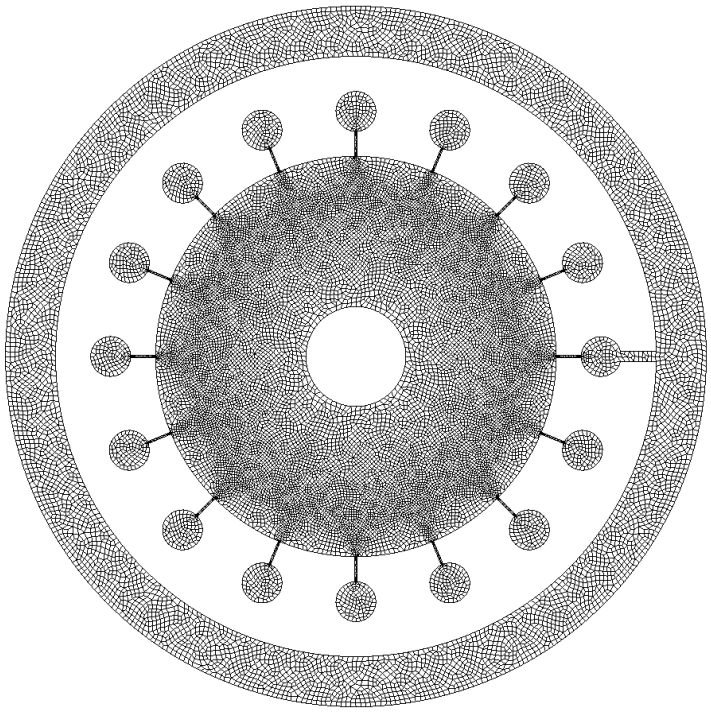


Figure 13: Mesh used for the simulations for the scattering of a magnetron ($\mathbb{Q}_8$ approximation).

medium is homogeneous ($\rho = \mu = 1$), there are 16 circular cavities of radius 4.0. The domain has a diameter of 140 wavelengths, since the radius of the outer circle is equal to 70, and the central frequency of the source is equal to 1.0. A homogeneous Neumann boundary condition is set on all the boundaries except at the external circle. On this outer circle, an inhomogeneous absorbing boundary condition is set to simulate the propagation of an incident plave wave. The time source is a sine func-

tion modulated by a gaussian

$$h(t) = \sin(2\pi t)\exp(-0.1(t-t_0)^2)$$

t_0 is chosen such that the gaussian is equal to 10^{-6} at $t = 0$:

$$t_0 = \sqrt{10\ \log(10^6)}$$

The source term f_A is given as

$$f_A(x,t) = (1 - u \cdot n)h'(t - u \cdot x - 70)$$

where $u = (-1, 0)$ is the orientation of the plane wave and n the outgoing normale. The solution obtained is plotted in figure 14 for different times. The reference solution is computed with eighth-order Padé scheme and $\Delta t = 0.01$. Among fourth-order time schemes, we compare the following schemes

- Classical RK4 scheme (ERK 4-0)
- ERK 4-2 with local implicit LSDIRK 4-1
- ERK 4-2 with local implicit Padé 4
- LSDIRK 4-1
- Padé 4

Local implicit schemes use the splitting computed as described in section 4 with $\Delta t_{\text{ref}} = 0.032$. The obtained coarse and fine regions are displayed in figure 15. The results obtained for fourth-order schemes are summarized in table 2. For this case,

Table 2: Comparison of different fourth-order time schemes for the magnetron.

Time Scheme	Time step	L^2 error	Computation time	Memory
Explicit RK4	$4.57 \cdot 10^{-4}$	$8.12 \cdot 10^{-10}$	9h20min	745 Mo
Local LSdirk 4-1	0.025	$1.17 \cdot 10^{-3}$	53min31s	1.8 Go
Local Padé 4	0.025	$1.36 \cdot 10^{-3}$	39min30s	2.3 Go
LSDIRK 4-1	0.04	$2.645 \cdot 10^{-3}$	68min57s	3.2 Go
Padé 4	0.0333	$3.454 \cdot 10^{-3}$	37min	4.8 Go

we observe that the explicit scheme (ERK 4-0) is cheap in memory (745 Mo) but is expensive in computational time (more than 9 hours). On the other hand, the implicit fourth-order Padé scheme is the fastest (37 minutes) but requires much more memory (4.8 Go). Locally implicit schemes achieve a compromise between these two schemes, since the storage requirement is lower than for implicit schemes and the computational time smaller than for explicit schemes. Among eighth-order time schemes, we compare the following schemes:

- ERK 8-2 with local implicit LSDIRK 8-1
- ERK 8-2 with local implicit Padé 8
- LSDIRK 8-1
- Padé 8

Here, local implicit schemes use the same splitting as for fourth order-schemes (see figure 15). The results obtained for these schemes are summarized in table 3. Because of the use of eighth-order schemes, the local implicit schemes are much more accurate compared to fourth order schemes. Again the implicit Padé scheme is the fastest and local implicit schemes use less memory. For this case, we observe that the error obtained for local LSDIRK scheme is much higher than for local Padé scheme. This is probably due to the issue of convergence we have observed for the square.

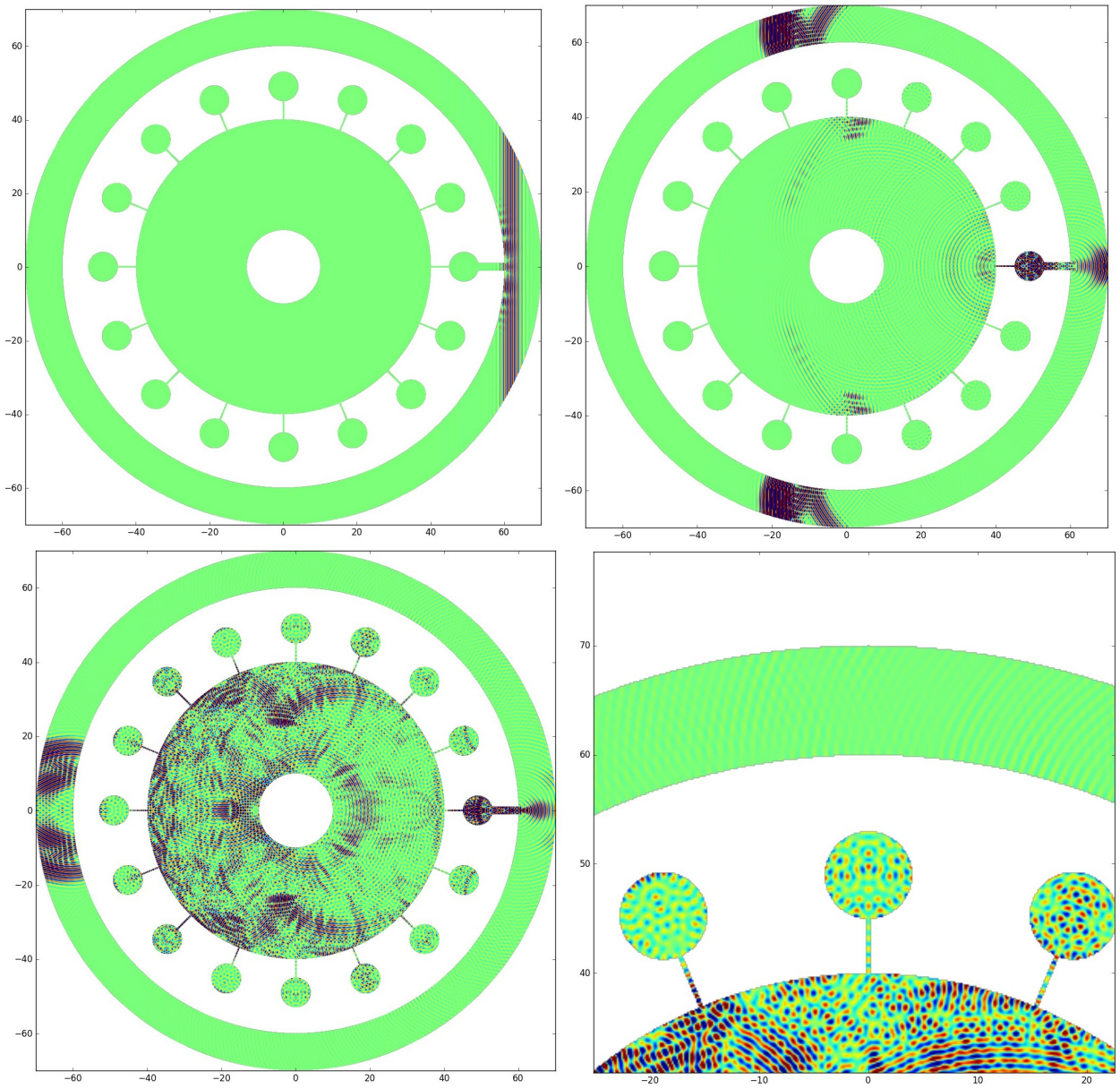


Figure 14: Solution for the magnetron for $t = 20, 100, 200$. On right bottom, zoom of the solution near circular cavities.

Table 3: Comparison of different eighth-order time schemes for the magnetron.

Time Scheme	Time step	L^2 error	Computation time	Memory
Local LSDIRK 8-1	$1/30$	$1.92 \cdot 10^{-6}$	82min7s	2.1 Go
Local Padé 8	$1/30$	$2.25 \cdot 10^{-9}$	59min20s	3.1 Go
LSdirk 8-1	$1/6$	$2.01 \cdot 10^{-3}$	34min2s	3.2 Go
Padé 8	$1/4$	$2.02 \cdot 10^{-3}$	10min31s	7.9 Go

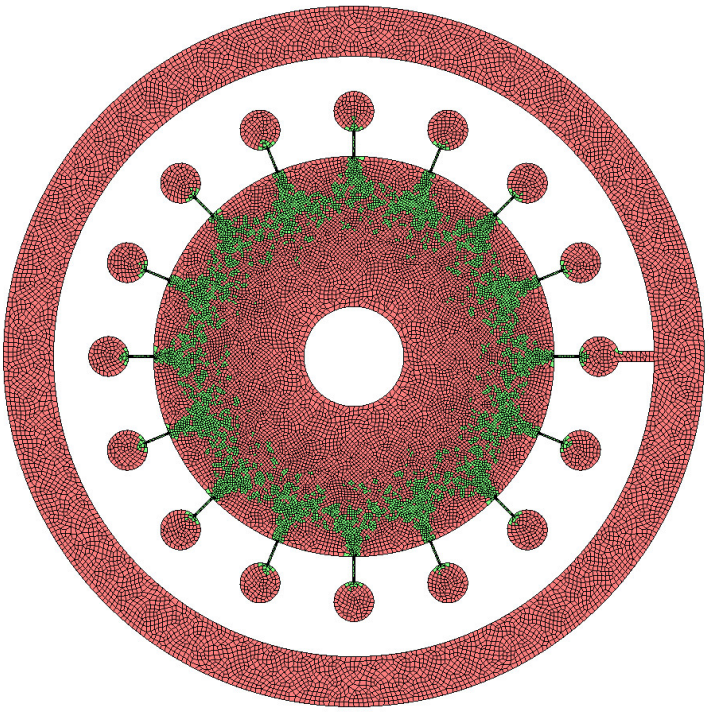


Figure 15: Splitting of the magnetron mesh into the fine part (green) and coarse part (red).

6.4 3-D case

In this section, we consider the scattering by a network of small spheres (see figure 16). These spheres are placed regularly in a parallelepiped (5 spheres in x-axis, 5

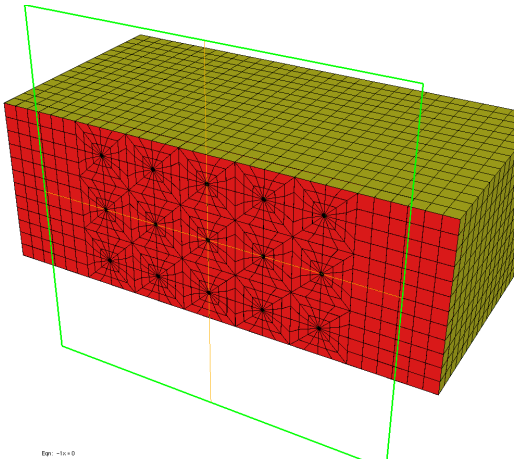


Figure 16: 3-D mesh used for the scattering by a network of small spheres. ($\mathbb{Q}_4$ approximation)

in y-axis and 3 in z-axis). The distance between each sphere is equal to 2.0 whereas the radius of each sphere is equal to 0.07. This network of spheres is placed in the parallelepiped box $[-8, 8] \times [-8, 8] \times [-3, 3]$. The physical indexes inside the spheres

are given by

$$\rho = 0.1,\ \mu = 0.8$$

whereas the surrounding media is characterized by $\rho = \mu = 1$. Inhomogeneous Dirichlet condition is set on the bottom plane $z = -3$ whereas homogeneous absorbing boundary condition is set on other boundaries. In figure 16, the hexahedral mesh used for the simulations is displayed. $\mathbb{Q}_4$ polynomial approximation is used with this mesh. On the plane $z = -3$, we have taken the following inhomogeneous Dirichlet condition

$$f_D = g(x, y)h(t)$$

where the space source g is a gaussian centered at the origin

$$g(x, y) = \beta \exp(-\alpha(x^2 + y^2))$$

with

$$r_0 = 5, \quad \alpha = \frac{\log(10^6)}{r_0^2}, \quad \beta = \sqrt{\frac{\alpha}{\pi}}$$

and h is a sine function modulated by a gaussian

$$h(t) = \exp(-2(t - t_0)^2) \sin(2\pi t)$$

t_0 is chosen such that the gaussian is equal to 10^{-6} at $t = 0$:

$$t_0 = \sqrt{\frac{\log(10^6)}{2}}$$

The solution obtained for $t = 6$ and $t = 12$ is represented in figure 17. We focus here on the following fourth-order time schemes:

- Classical RK4 scheme
- ERK 4-0 with local implicit LSDIRK 4-1
- LSDIRK 4-1

For the first scheme which is purely explicit, we set Δt close to the CFL. For the second time scheme, we have obtained the splitting given in figure 18 between the coarse and fine mesh for $\Delta t_{\text{ref}} = 0.01$. For the third time scheme, the time step is chosen such that we obtain 0.1% of L^2 error for the final time $T = 20$. The computational time and memory required for the three schemes to compute the solution until the final time $T = 20$ are given in table 4. The L^2 error has been computed

Table 4: Comparison of different fourth-order time schemes for the scattering by a network of spheres.

Time scheme	Time step	L^2 error	Computation Time	Memory
Explicit RK4	$2.22 \cdot 10^{-4}$	-	9h47	3.3 Go
Locally implicit	0.01	$1.47 \cdot 10^{-5}$	2h19	62.8 Go
LSDIRK 4-1	0.05	$1.06 \cdot 10^{-3}$	48min	108 Go

for the final time and by taking the solution given with the first scheme (explicit) as reference. It can be observed that the second scheme (locally implicit) achieves a compromise between computational time (faster than a purely explicit scheme) and memory usage (less expensive than a purely implicit scheme).

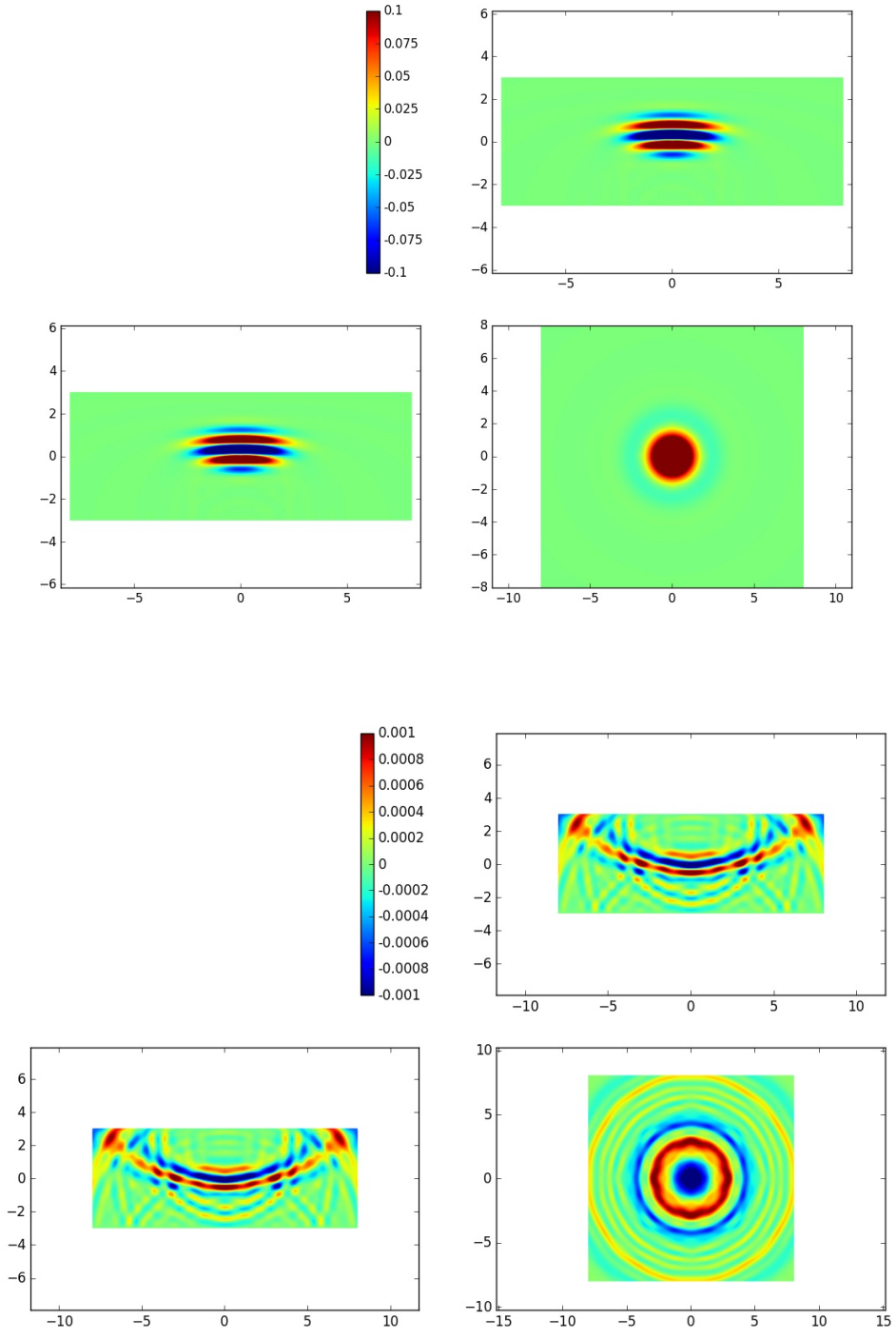


Figure 17: Solution obtained for the scattering by a network of spheres. Solution for $t = 6$ (top) and $t = 12$ (bottom).

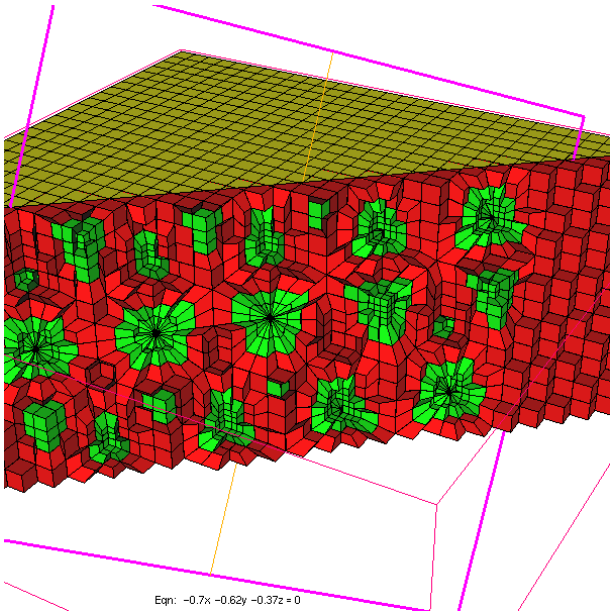


Figure 18: Splitting into coarse (red) and fine region (green).

7 Conclusion

In this paper, we have presented locally implicit schemes for linear ODEs with an arbitrary order. These schemes couple ERK (Explicit Runge Kutta) schemes with different type of one-step implicit schemes. Herein, we have considered either Padé schemes or Linear SDIRK schemes for the implicit time integration. When Padé schemes are used for the implicit part, we have observed that the global scheme converges with the expected order (i.e. in $O(\Delta t^p)$ for the scheme of order p). We also observed that the CFL is almost the same as the CFL computed for the coarse region. The fully discrete scheme in space and time exhibits also a nice convergence in $O(h^p + \Delta t^p)$ when Hybridizable Discontinuous Galerkin approximation is used in space. When Linear SDIRK schemes are used for the implicit part, the convergence is nice only for fourth-order schemes, but it becomes progressively bad for higher orders. This deterioration could be due to spurious oscillations that have been previously observed in [25]. In spite of this issue of convergence, the accuracy obtained with these schemes is still satisfactory and besides they require less memory usage than the locally implicit methods formed using the Padé schemes.

The 2-D and 3-D numerical results for the wave equation presented in this paper show that explicit schemes are highly constrained by the stability condition, i.e. the time-step has to be chosen such that the CFL condition is satisfied. For this reason even though the optimized explicit schemes give an accurate solution, they are very expensive in terms of computational time which make them less efficient for the tested cases. On the other hand globally implicit integration with unconditionally stable scheme has no CFL condition to satisfy, but they are limited by an accuracy level, i.e. the time-step has still to be chosen to reach the targeted accuracy level. We observed that they require less computational time compared to explicit schemes, but they consume a large amount of memory when direct solvers are used. In a middle of explicit and implicit schemes, the locally implicit schemes we have developed achieve a compromise between implicit and explicit schemes. In this case, the time step is chosen by the user in order to have a reduced computational time and memory usage.

As a prospect, we think that it would be interesting to couple local time stepping strategy with locally implicit method in order to use an implicit method for very fine regions.

Acknowledgements

This work has been supported by the Inria-TOTAL strategic action DIP (dip.inria.fr). Experiments presented in this work were carried out using the PlaFRIM experimental platform.

References

[1] G. Cohen and S. Fauqueux, "Mixed finite elements with mass-lumping for the transient wave equation," *Journal of Computational Acoustics*, vol. 8, pp. 171–188, 2000.

[2] G. Cohen and S. Pernet, *Finite element and discontinuous Galerkin methods for transient wave equations*. Springer, 2017.

[3] J. S. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods algorithms, analysis, and applications*. Springer, 2008.

[4] P. Monk, *Finite element methods for Maxwell's equations*. Oxford Science Publications, 2003.

[5] M. Duruflé, *Intégration numérique et éléments finis d'ordre élevé appliqués aux équations de Maxwell en regime harmonique*. PhD thesis, Université Paris Dauphine, 2006.

[6] E. Hairer, S. P. Norsett, and G. Wanner, *Solving ordinary differential equations I nonstiff problem*. Springer, 2008.

[7] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations Second Edition*. John Willey & Sons Ltd, 2008.

[8] E. Hairer and G. Wanner, *Solving ordinary differential equations II stiff and differential-algebraic problem*. Springer, 2010.

[9] J. L. Mead and R. A. Renaut, "Optimal Runge-Kutta methods for first order pseudospectral operators," *Journal of Computational Physics*, pp. 404–419, 1999.

[10] J.-C. Gilbert and P. Joly, "Higher order time stepping for second order hyperbolic problems and optimal CFL conditions," *Computational Methods in Applied Sciences*, vol. 16, pp. 67–93, 2008.

[11] P. Joly and J. Rodrìguez, "Optimized higher order time discretization of second order hyperbolic problems: Construction and numerical study," *Journal of Computational and Applied Mathematics*, vol. 234, pp. 1953–1961, 2010.

[12] M. Parsani, D. I. Ketcheson, and W. Deconinck, "Optimized explicit Runge-Kutta schemes for the spectral difference method applied to wave propagation problems," *SIAM Journal on Scientific Computing*, vol. 35, pp. A957–A986, 2013.

[13] D. I. Ketcheson and A. J. Ahmadia, "Optimal stability polynomials for numerical integration of initial value problems," *Communications in Applied Mathematics And Computational Science*, vol. 7, pp. 247–271, 2012.

[14] M. Duruflé and M. N'diaye, "Optimized high-order explicit Runge-Kutta-Nyström schemes," in *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016* (M. Bittencourt, N. Dumont, and J. S. Hesthaven, eds.), Springer, 2016.

[15] J. Diaz and M. Grote, "Energy conserving explicit local time stepping for second-order wave equations," *SIAM Journal on Scientific Computing*, vol. 31, pp. 1985–2014, 2009.

[16] M. Mehlin, T. Mitkova, and M. Grote, "Runge-Kutta-based explicit local time-stepping methods for wave propagation," *SIAM Journal on Scientific Computing*, vol. 37, pp. A747–A775, 2015.

[17] S. Piperno, "Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems," *ESAIM Mathematical Modelling and Numerical Analysis*, vol. 40, pp. 815–841, 2006.

[18] J. Rodrìguez, *Raffinement de maillage spatio-temporel pour les équations de l'élastodynamique*. PhD thesis, Université Paris Dauphine, 2004.

[19] L. M. Skvortsov, "Diagonally implicit Runge-Kutta methods for stiff problems," *Computational Mathematics and Computational Physics*, vol. 46, pp. 2110–2123, 2006.

[20] B. L. Ehle, "A-stable methods and Padé approximations to the exponential," *SIAM Journal on Numerical Analysis*, vol. 4, pp. 671–680, 1973.

[21] 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.

[22] S. Descombes, S. Lanteri, and L. Moya, "Locally implicit discontinuous Galerkin time domain method for electromagnetic wave propagation in dispersive media applied to numerical dosimetry in biological tissues," *SIAM Journal on Scientific Computing*, vol. 38, pp. A2611–A2633, 2016.

[23] S. Descombes, S. Lanteri, and L. Moya, "Locally implicit time integration strategies in a discontinuous Galerkin method for Maxwell's equations," tech. rep., INRIA, 2012.

[24] A. Sturm, *Locally implicit time integration for linear Maxwell's equations*. PhD thesis, Karlsruhe Institute of Technology, 2017.

[25] H. Barucq, M. Duruflé, and M. N'Diaye, "High-order Padé and singly diagonally Runge-Kutta schemes for linear ODEs, application to wave propagation problems," *Numerical Methods for Partial Differential Equations*, pp. 1–39, 2017.

[26] M. Kronbichler, S. Schoeder, C. Müller, and W. A. Wall, "Comparison of implicit and explicit hybridizable discontinuous Galerkin methods for the acoustics wave equation," *International Journal for Numerical Methods in Engineering*, vol. 106, pp. 712–739, 2016.

[27] N. Nguyen, J. Peraire, and B. Cockburn, "High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics," *Journal of Computational Physics*, vol. 230, pp. 3695–3718, 2011.

[28] V. Hernandez, J. E. Roman, and V. Vidal, "Slepc: A scalable and flexible toolkit for the solution of eigenvalue problems," *ACM Trans. Math. Software*, vol. 31, pp. 351–362, 2005.

[29] P. Amestoy, I. Duff, J. Koster, and J.-Y. L'Excellent, "A fully asynchronous multifrontal solver using distributed dynamic scheduling," *SIAM Journal on Matrix Analysis and Applications*, vol. 23, pp. 15–41, 2001.