



Efficient Discontinuous Galerkin Methods for solving acoustic problems

Nicolas Chevaugeon, Jean-François Remacle, Xavier Gallez, Paul Ploumans,
Stéphane Caro

► To cite this version:

Nicolas Chevaugeon, Jean-François Remacle, Xavier Gallez, Paul Ploumans, Stéphane Caro. Efficient Discontinuous Galerkin Methods for solving acoustic problems. 11th AIAA/CEAS Aeroacoustics Conference, May 2005, Monterey, California, United States. 10.2514/6.2005-2823 . hal-03924279

HAL Id: hal-03924279

<https://hal.science/hal-03924279>

Submitted on 5 Jan 2023

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

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

Efficient Discontinuous Galerkin Methods for solving acoustic problems

Nicolas Chevaugeron* and Jean-François Remacle†

Université Catholique de Louvain, Louvain-la-Neuve, 1348, Belgium

Xavier Gallez‡, Paul Ploumhans§ and Stéphane Caro¶

Free Field Technologies, Louvain-la-Neuve, 1348, Belgium

Contents

I	Introduction	2
II	Mathematical model	2
A	Finite and infinite elements, frequency-domain approach	2
1	Governing equation	2
2	Free field boundary condition	3
3	Coupling with acoustic ducts	3
4	Spatial discretization	3
5	Solver	4
B	RK-DGM approach	4
1	Governing equations	4
2	Spatial discretization	5
3	Implementation of the space Operator	5
4	Optimization of the computation of surface fluxes	7
5	Convergence properties	8
6	Time discretization	10
7	Boundary conditions	10
III	Validation	12
A	Test case	12
B	Results	12
IV	Large-scale applications	15
A	Parallel implementation of the DG method	15
B	Test case 1	15
C	Test case 2	17
V	Conclusions	18
VI	Acknowledgments	18

*Post-doctoral fellow, Civil Engineering Department, 1 pl. du Levant, B-1348, Louvain-la-Neuve, Belgium / chevaugeron@gce.ucl.ac.be

†Associate Professor, Civil Engineering Department, 1 pl. du Levant, B-1348, Louvain-la-Neuve, Belgium / remacle@gce.ucl.ac.be

‡Aeroacoustics research manager, 16 pl de l'Université, B-1348, Louvain-la-Neuve, Belgium / xavier.gallez@fft.be

§Product manager, 16 pl de l'Université, B-1348, Louvain-la-Neuve, Belgium / paul.ploumhans@fft.be

¶Aeroacoustics specialist, 16 pl de l'Université, B-1348, Louvain-la-Neuve, Belgium / stephane.caro@fft.be

I. Introduction

For being the major source of noise pollution in aero-engines, fan forward and rearward noise is of primary importance in the aeronautics industry. With a view to enabling acoustic-based design of engines' nacelle and bypass ducts, Free Field Technologies has developed a tool, called ACTRAN/TM, over the past five years. Briefly, ACTRAN/TM computes the propagation of harmonic acoustic disturbances in non-uniform, homentropic mean flow and their radiation to the far field. Using Myers' formulation of acoustic boundary conditions,^{1,2} the effect of acoustic liners in the presence of flow is accurately accounted for. Strong coupling with duct acoustic modes allows one to both specify incident modes and compute the amplitude of reflected ones. The algebraic system of equations resulting from the finite and infinite element discretization of the acoustic potential equation is solved with a parallel, out-of-core, direct solver.

While ACTRAN/TM is widely being used in the industry, for example for nacelle liner optimization, the constant need for higher and higher values of the Helmholtz number in numerical simulations and the increasing interest in fan rearward and turbine noise pose new challenges. For example, a three-dimensional nacelle problem with flow at $kr = 30$ currently requires several Gigabytes of RAM and hundreds of gigabytes of disk space. The limited efficiency and scalability of direct solvers do not make them good candidates for addressing problems where the Helmholtz number is as high as 50, especially on distributed memory computers. Moreover, propagation in rotational, non-isothermal mean flows can not be addressed with an acoustic potential equation.

The above aspects of industrial applications motivate the combination of the time-domain linearized Euler equations (LEE) and the quadrature-free Runge-Kutta Discontinuous Galerkin Method (RK-DGM). An advantageous feature of the DGM is that it allows the use of high-order spatial discretization schemes in a natural fashion. While the number of floating-point operations is high in the DGM, a quadrature-free formulation allows efficient algorithms. By organizing data in a matrix fashion and noting that element matrices differ from one element to the next by only a coefficient, the most demanding operations are written in matrix-matrix form and taken care of by BLAS3 functions. Capitalizing on the work of reputed groups,³⁻⁷ and conducting their own research in collaboration with major research institutes, UCL and Free Field Technologies recently embarked on implementing the RK-DGM for LEE, focusing on large-scale, industrial applications.

This paper presents some early results obtained with our code, focusing on performance, efficiency, data structures and parallel scalability. Our work on time-domain impedance boundary conditions and non-reflecting boundary conditions will be the topic of forthcoming papers.

The paper is structured as follows. In Section II, we recall the frequency-domain convected wave equation and the time-domain linearized Euler equations. Numerical discretization schemes (finite/infinite elements and RK-DGM) and their properties are presented, with emphasis on the implementation of the RK-DGM scheme. Our RK-DGM code is validated against ACTRAN/TM for a two-dimensional case in Section III. In Section IV, the RK-DGM code is applied to a large three-dimensional application. Conclusions are presented in Section V.

II. Mathematical model

A. Finite and infinite elements, frequency-domain approach

1. Governing equation

The propagation of acoustic waves in a non-rotational, homentropic mean flow is governed by the convected wave equation:

$$\frac{D_0}{Dt} \left(\frac{\rho_0}{c_0^2} \frac{D_0 \phi'}{Dt} \right) - \nabla \cdot (\rho_0 \nabla \phi') - \rho' \nabla \cdot \mathbf{v}_0 = 0. \quad (1)$$

In (1), ϕ' denotes the acoustic potential, D_0/Dt is the material derivative along mean flow streamlines :

$$\frac{D_0}{Dt} = \frac{\partial}{\partial t} + \mathbf{v}_0 \cdot \nabla, \quad (2)$$

while variables with subscript "0" and variables with superscript "'" result from the splitting of field variables

into a steady, mean flow part and an acoustic part:

$$\begin{cases} \rho = \rho_0 + \rho' \\ p = p_0 + p' \\ \mathbf{v} = \mathbf{v}_0 + \mathbf{v}' \end{cases} . \quad (3)$$

For time-harmonic excitations, (1) is written in the frequency domain by introducing the frequency-domain acoustic potential, $\tilde{\phi}$, such that

$$\phi(\mathbf{x}, t) = \text{Re} \left(\tilde{\phi}(\mathbf{x}, t) e^{i\omega t} \right) , \quad (4)$$

where $\omega = 2\pi f$ in the circular frequency. The frequency domain acoustic pressure, \tilde{p} , is related to $\tilde{\phi}$ through

$$\tilde{p} = -\rho_0 \left(i\omega \tilde{\phi} + \mathbf{v}_0 \cdot \nabla \tilde{\phi} \right) . \quad (5)$$

2. Free field boundary condition

The acoustic field satisfies the Sommerfeld boundary condition at large distances from acoustic sources:

$$\lim_{r \leftarrow \infty} \left[r \left(\frac{\partial \tilde{p}}{\partial r} - ik\tilde{p} \right) \right] = 0 , \quad (6)$$

where $k = \omega/c_0$ is the wavenumber.

3. Coupling with acoustic ducts

At the fan face, the computational domain is coupled to a semi-infinite duct, where the acoustic potential is expressed in terms of duct modes. For annular ducts, we have that

$$\begin{aligned} \tilde{\phi}(r, \theta, z) = & \sum_{m,n} (J_m(k_{rmn}r) + CY_m(k_{rmn}r)) \\ & e^{im\theta} (A_{mn}^+ e^{-ik_{zmn}^+ z} + A_{mn}^- e^{-ik_{zmn}^- z}) , \end{aligned} \quad (7)$$

where J_m and Y_m are Bessel and Neumann's functions of order m , respectively, k_{rmn} is the radial wavenumber, and k_{zmn}^+ and k_{zmn}^- are longitudinal wavenumbers associated to the so-called *incident* and *reflected* modes, respectively. The values of k_{rmn} and C are such that (7) satisfies the hard wall boundary condition, $\tilde{v}_n = 0$.

4. Spatial discretization

The unbounded computational domain of interest, Ω , is divided into an inner region and a outer region denoted Ω_i and Ω_o , respectively (see Figure 5a). The inner region is discretized using linear or quadratic finite elements, while the acoustic potential in the outer region is approximated by infinite elements, thereby enforcing the Sommerfeld boundary condition and allowing the reconstruction of the acoustic solution in the far field. To do so, the flow is uniform in the outer region. The infinite element method implemented in ACTRAN/TM is an extension of a variable order Legendre polynomial formulation whose numerical performances have been extensively studied.^{8,9} The description of this extension for the convected case is referenced in.¹⁰ To enforce the strong coupling between the finite element solution and the acoustic duct mode solution at the fan face, (7) is projected on the finite element shape functions. Let us note that the reflected modes amplitudes are a priori unknown, just like finite element nodal values.

5. Solver

The finite and infinite element discretizations and the coupling with duct acoustic modes lead to a complex algebraic sparse system of equations,

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (8)$$

solved by a direct solver. For large problems, the computational domain is partitioned and (8) solved in parallel using out-of-core capabilities.¹¹

B. RK-DGM approach

In the case of RK-DGM, we solve the linearized Euler equations (LEE) in the time domain. Starting from the LEE equations, we develop briefly the spatial discretization of the equation using the Discontinuous Galerkin method (DGM) stressing out the consequences of the quadrature free version of the scheme in terms of practical implementation. Convergence results of the scheme are also shown for one-dimensional problems, as well as the stability condition for explicit multi-step spatial resolution. Finally, we briefly describe the boundary conditions for imposing duct modes and open-boundary conditions in time domain.

1. Governing equations

For homentropic flows, the isentropic linearized Euler equations reduce to:¹²

$$\frac{\partial p'}{\partial t} + \nabla \cdot (\rho_0 c_0^2 \mathbf{v}' + \mathbf{v}_0 p') = 0 \quad (9)$$

$$\frac{\partial \mathbf{v}'}{\partial t} + \nabla \cdot \left(\mathbf{v}' \mathbf{v}_0 + \frac{1}{\rho_0} \mathbf{I} p' \right) = -\frac{1}{\rho_0 c_0^2} \left(\mathbf{v}_0 \cdot \nabla \mathbf{v}_0 + \frac{c_0^2}{\rho_0} \nabla \rho_0 \right) p' + (\nabla \cdot \mathbf{v}_0) \mathbf{v}' - \mathbf{v}' \cdot \nabla \mathbf{v}_0. \quad (10)$$

To simplify the development of DG methods, and write it in a general way, we introduce the array of fields variables, \mathbf{u}' , the array of flux vectors, \mathbf{F} , and the array of source terms, \mathbf{s} , such as :

$$\mathbf{u}' = \begin{pmatrix} p' \\ v'_x \\ v'_y \\ v'_z \end{pmatrix}, \quad (11)$$

$$\mathbf{F} = \begin{pmatrix} \frac{1}{c_0^2} v_{0x} p' + \rho_0 v'_x & \frac{1}{c_0^2} v_{0y} p' + \rho_0 v'_y & \frac{1}{c_0^2} v_{0z} p' + \rho_0 v'_z \\ v'_x v_{0x} + \frac{p'}{\rho_0} & v'_x v_{0y} & v'_x v_{0z} \\ v'_y v_{0x} & v'_y v_{0y} + \frac{p'}{\rho_0} & v'_y v_{0z} \\ v'_z v_{0x} & v'_z v_{0y} & v'_z v_{0z} + \frac{p'}{\rho_0} \end{pmatrix}, \quad (12)$$

$$\mathbf{s} = -\frac{1}{\rho_0 c_0^2} \begin{pmatrix} 0 \\ \frac{1}{\rho_0} c_0^2 \rho_{0,x} p' + (\mathbf{v}_0 \nabla \mathbf{v}_0)_x \\ \frac{1}{\rho_0} c_0^2 \rho_{0,y} p' + (\mathbf{v}_0 \nabla \mathbf{v}_0)_y \\ \frac{1}{\rho_0} c_0^2 \rho_{0,z} p' + (\mathbf{v}_0 \nabla \mathbf{v}_0)_z \end{pmatrix}. \quad (13)$$

Equation (10) can now be rewritten in compact form :

$$\frac{\partial \mathbf{u}'}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{s}. \quad (14)$$

2. Spatial discretization

To obtain the DG formulation, one first multiplies equation (14) with a test function, $\hat{\mathbf{u}}$, and integrates over the domain, Ω . The divergence theorem is then applied to obtain the following variational formulation

$$\begin{aligned} \int_{\Omega} \partial_t \mathbf{u}' \cdot \hat{\mathbf{u}} \, dv + \int_{\Omega} \mathbf{F}_x(\mathbf{u}') \cdot \partial_x \hat{\mathbf{u}} \, dv + \int_{\Omega} \mathbf{F}_y(\mathbf{u}') \cdot \partial_y \hat{\mathbf{u}} \, dv \\ + \int_{\Omega} \mathbf{F}_z(\mathbf{u}') \cdot \partial_z \hat{\mathbf{u}} \, dv - \int_{\partial\Omega} f \hat{\mathbf{u}} \, ds = \int_{\Omega} \mathbf{s} \cdot \hat{\mathbf{u}} \, dv, \quad \forall \hat{\mathbf{u}}, \end{aligned} \quad (15)$$

where $\mathbf{f}(\mathbf{u}') = \mathbf{F}\mathbf{u}' \cdot \vec{n}$ is the normal trace of the fluxes.

The physical domain Ω is discretized into a collection of \mathcal{N}_e elements,

$$\mathcal{T} = \bigcup_{e=1}^{\mathcal{N}_e} e, \quad (16)$$

called *mesh*. In each element e of \mathcal{T} , each component of \mathbf{u}' is discretized using polynomials. It is common, in the finite element world, to distinguish reference coordinates, ξ, η, ζ , from real coordinates x, y, z . We use piecewise continuous approximations on each element:

$$\mathbf{u}'(\xi, \eta, \zeta) = \sum_{k=1}^d N_k(\xi, \eta, \zeta) \mathbf{u}'_k{}^e. \quad (17)$$

We see that, in (17), the coefficients $\mathbf{u}'_k{}^e$ of the interpolation are associated to the element. The interpolation is then discontinuous at inter-element boundaries, contrary to classical finite element methods. The interface term in 15 appears for each interface between elements, and we need to compute for each of them the normal trace of the flux across this interface. The flux is not uniquely determined there, and the choice of an appropriate numerical flux is at the heart of the properties of the method. For each interface in the mesh, we define a unique normal to it, \mathbf{n} , and we note \mathbf{u}'_L and \mathbf{u}'_R the field variables on the left and right sides of this face, respectively, with \mathbf{n} going from the left to the right. The numerical flux, $f(\mathbf{u}'_L, \mathbf{u}'_R)$, is computed using a Riemann solver. The Riemann solver that we use is constructed by computing characteristics of the left and right fields, and only the upwind quantities are kept to compute the normal flux.

3. Implementation of the space Operator

The problem (14) that we aim to solve is a linear hyperbolic PDE with non-constant coefficients if the mean flow defined by v_0 , ρ_0 and c_0 is not uniform. The following very important assumption is done for treating non linearities. If $g(\mathbf{u}')$ is a function of the unknown \mathbf{u}' , we apply the following rule to compute g :

$$g(\mathbf{u}') = g\left(\sum_{k=1}^d N_k \mathbf{u}'_k{}^e\right) \simeq \sum_{k=1}^d N_k g(\mathbf{u}'_k{}^e). \quad (18)$$

Where d is the number of shape function per elements.

This assumption permits to derive the quadrature free DGM,⁵ an approximation that permits to achieve a much higher efficiency than the direct integration of each term by means of summation over quadrature points.

The idea behind assumption (18) is that any function is approximated on the “same grid as \mathbf{u}' ”. For example, if $g = \mathbf{u}'^2$ and if $\mathbf{u}'^e = \sum_k N_k \mathbf{u}'_k{}^e$ is in \mathbb{P}^p , then $(\mathbf{u}'^e)^2 = (\sum_k N_k \mathbf{u}'_k{}^e)^2$ is in \mathbb{P}^{2p} . With assumption (18), $\mathbf{u}'^2 \simeq \sum_k N_k (\mathbf{u}'_k{}^e)^2 \in \mathbb{P}^p$. Equation (18) is, of course, only exact for linear functions F .

In this work, we use Lagrange shape functions for interpolating \mathbf{u}' . This choice is certainly not the only one available. Orthogonal shape functions,¹³ for example, have specific interest. Here, we have d lagrange points in each element e . Denoting the value of a given function f at node k of element e by f_k^e , we therefore have that^a

$$f(\mathbf{u}') = \sum_k f_k^e N_k = f_k^e N_k. \quad (19)$$

^aStarting here, we use the Einstein summation rule for repeated indices.

The interpolation in one element being disconnected from the interpolation in the neighboring elements, it is possible to write (15) for each element e of \mathcal{T} as

$$\begin{aligned} \partial_t \mathbf{u}'_k \int_e N_k N_j dv &= \mathbf{F}_x(\mathbf{u}'_k) \int_e N_k \partial_x N_j dv \\ &+ \mathbf{F}_y(\mathbf{u}'_k) \int_e N_k \partial_y N_j dv \\ &+ \mathbf{F}_z(\mathbf{u}'_k) \int_e N_k \partial_z N_j dv \\ &- \sum_{l=1}^{n_e} \mathbf{f}(\mathbf{u}'_L, \mathbf{u}'_R)_k^{e,l} \cdot \vec{n} \int_{\partial e_l} N_k N_j ds = 0 \end{aligned} \quad (20)$$

$$+ \mathbf{s}(\mathbf{u}'_k) \int_e N_k N_j dv, \forall i, j \quad (21)$$

where we have decomposed the boundary ∂e of element e into n_e parts ∂e_l corresponding, for three-dimensional problems, to the 4 triangular faces of the tetrahedron. In (20), $\mathbf{f}(\mathbf{u}'_L, \mathbf{u}'_R)_k^{e,l} \cdot \vec{n}$ is the numerical flux computed using the Riemann solver between the field on the left and right of the face.

We further assume that the Jacobian matrix,

$$J = \begin{bmatrix} \partial_\xi x & \partial_\eta x & \partial_\zeta x \\ \partial_\xi y & \partial_\eta y & \partial_\zeta y \\ \partial_\xi z & \partial_\eta z & \partial_\zeta z \end{bmatrix} \quad (22)$$

is constant for any element e . This is true if all edges of the mesh are straight sided. We note $\|e\| = \det J$. Some important DGM operators appear in (20). We define two mass matrix operators, one relative to element e ,

$$M_{ij}^e = \int_e N_i N_j dx dy dz = \int_e N_i N_j d\xi d\eta d\zeta \|e\| = M_{ij} \|e\|, \quad (23)$$

and one relative to element boundaries ∂e_k ,

$$M_{ij}^{\partial e_l} = \int_{\partial e_l} N_i N_j dx dy dz = \int_{\partial e_l} N_i N_j d\xi d\eta d\zeta \|\partial e_l\| = M_{ij}^l \|\partial e_k\|. \quad (24)$$

M_{ij} and M_{ij}^l are constant matrices, independent of e , and of size $d \times d$. We finally define three derivatives operators,

$$D_{ij}^\xi = \int_e N_i \partial_\xi N_j d\xi d\eta d\zeta, \quad D_{ij}^\eta = \int_e N_i \partial_\eta N_j d\xi d\eta d\zeta, \quad D_{ij}^\zeta = \int_e N_i \partial_\zeta N_j d\xi d\eta d\zeta. \quad (25)$$

These operators are square matrices of size $d \times d$. They are all independent of e .

The choice of a specific family of shape functions makes the structure of some DGM operators computationally interesting. For Lagrange shape functions, boundary operators M_{ij}^e have lots of zeros. In case of orthogonal polynomials, on the other hand, derivative operators are upper triangular matrices. Note that

$$D_{ij}^x = \int_e N_i \partial_x N_j dx dy dz = \|e\| \left(D_{ij}^\xi J_{11}^{-1} + D_{ij}^\eta J_{12}^{-1} + D_{ij}^\zeta J_{13}^{-1} \right). \quad (26)$$

Thanks to the previous definitions (22) to (25), the DGM formulation is written, for element e , as

$$\begin{aligned} \partial_t \mathbf{u}'_k M_{kj} &= (\mathbf{F}_\xi \mathbf{u}')_k^e D_{kj}^\xi + (\mathbf{F}_\eta \mathbf{u}')_k^e D_{kj}^\eta + (\mathbf{F}_\zeta \mathbf{u}')_k^e D_{kj}^\zeta \\ &- \frac{1}{\|e\|} \sum_{l=1}^{n_e} \|\partial e_l\| f_k^{e,l} M_{kj}^l + \mathbf{s}_k^e M_{kj} \end{aligned} \quad (27)$$

In (27), we have computed fluxes in the reference system of coordinates i.e.

$$\begin{aligned}
(\mathbf{F}_\xi(\mathbf{u}'))_k^e &= (\mathbf{F}_x(\mathbf{u}'))_k^e J_{11}^{-1} + (\mathbf{F}_y(\mathbf{u}'))_k^e J_{21}^{-1} + (\mathbf{F}_z(\mathbf{u}'))_k^e J_{31}^{-1}, \\
(\mathbf{F}_\eta(\mathbf{u}'))_k^e &= (\mathbf{F}_x(\mathbf{u}'))_k^e J_{12}^{-1} + (\mathbf{F}_y(\mathbf{u}'))_k^e J_{22}^{-1} + (\mathbf{F}_z(\mathbf{u}'))_k^e J_{32}^{-1}, \\
(\mathbf{F}_\zeta(\mathbf{u}'))_k^e &= (\mathbf{F}_x(\mathbf{u}'))_k^e J_{13}^{-1} + (\mathbf{F}_y(\mathbf{u}'))_k^e J_{23}^{-1} + (\mathbf{F}_z(\mathbf{u}'))_k^e J_{33}^{-1}.
\end{aligned}$$

Formulation (27) is a quadrature-free version of the DGM. For sufficiently large p , most of the computation time is spent in the computation of the derivative operator. An efficient way to implement (27) on a computer is to use Basic Linear Algebra Subroutines (BLAS). In particular, (27) involves three large matrix-matrix multiplications. Indeed, since all elements have the same matrices D^ξ , D^η , D^ζ , one can first compute the fluxes in each element frame of reference for all interpolation points and all elements, using equation 28. The fluxes computed this way can be stored in a matrix with d lines and $4\mathcal{N}_e$ columns, where 4 is the number of fields in \mathbf{u}' for our acoustic case. The derivative operator for every element can then be written by 3 matrix-matrix multiplications involving matrices of size sufficiently large to take advantage of Level 3 BLAS (BLAS3) subroutines.

Typically, for the DGM of order $p = 4$ on hexahedra, we have $d = (p+1)^3 = 125$ Lagrange shape functions for each element and the matrixes of problem (27) have a size 125×125 . The bigger the elementary matrices, the higher the sustained processor performance, and the more we need to rely on the quality of the BLAS implementation. Figure 1 shows the Gflops measured on a full computation of one time step for different p and BLAS implementations, on a 2.4GHZ Intel Xeon processor. All the codes were implemented using uBlas, a template library for linear algebra, part of the boost library, which provides optional bindings to BLAS packages. For raw uBlas, we reach a peak performance for $p = 7$ at less than 0.7 Gflops. Using bindings to the netlib version of the BLAS library, an performance of 1 Gflops is reached for $p = 8$. A much higher performance is obtained using The ATLAS open source optimized BLAS library,¹⁴ with a maximum measured performance of 2.5 Gflops for $p = 9$. The best results on Intel processors were obtained using the MKL implementation distributed by Intel. The choice of the BLAS implementation is therefore very important to take full advantage of the matrix structure of the code.

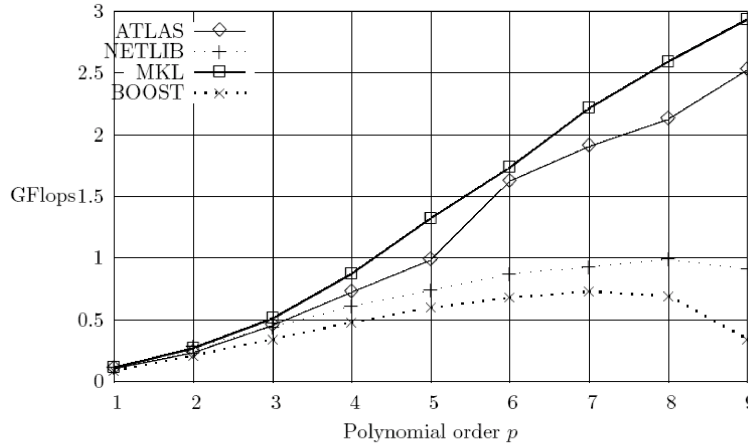


Figure 1. Performance of the DG Code for several p and BLAS implementations

4. Optimization of the computation of surface fluxes

As the computation of the volume fluxes is optimized, the computation of the fluxes on the elements' faces can, in certain circumstances, become non negligible and must be optimized also. Let us first write the surface flux of (27) in the following form:

$$\mathbf{f}(\mathbf{u}^*) \cdot \mathbf{n} = \underbrace{n_i \mathbf{F}_i \cdot \mathbf{T}^{-1} \cdot \mathbf{R}_{S, \text{left}} \cdot \mathbf{T} \cdot \mathbf{u}'_L}_{F_{S,L}} + \underbrace{n_i \mathbf{F}_i \cdot \mathbf{T}^{-1} \cdot \mathbf{R}_{S,R} \cdot \mathbf{T} \cdot \mathbf{u}'_R}_{F_{S,R}} \quad (28)$$

where

- the subscript ”*” denotes the restriction of the field variables, \mathbf{u}' , on the a face;
- \mathbf{T} is a transformation matrix that maps the vector field variables onto their normal and tangential components:

$$\mathbf{w}'^* = \mathbf{T} \cdot \mathbf{u}'^* \quad (29)$$

with $\mathbf{w}'^* = (\rho', v'_n, v'_t, v'_s)$;

- $\mathbf{R}_{S,left}$ and $\mathbf{R}_{S,right}$ are the Riemann solver matrices, such that

$$\mathbf{w}'^* = \mathbf{R}_{S,left} \cdot \mathbf{w}'_{left} + \mathbf{R}_{S,right} \cdot \mathbf{w}'_{right} ; \quad (30)$$

- $\mathbf{F}_{S,left}$ and $\mathbf{F}_{S,right}$ are called *surface flux matrices*

Since $\mathbf{F}_{S,left}$ and $\mathbf{F}_{S,right}$ depend only on the steady mean flow and the geometry, they can be cached, that is computed, stored, and repeatedly used over the time-steps of the simulation. Figure 2, show the memory requirement for each element for different with or without caching, compared to the overall memory requirement for different orders of p . Full caching of the surface flux caching, almost double the memory requirement, and should be used only for sufficiently small problems, or only at low order, when the cpu time for the surface term is still big compare to the volumes terms.

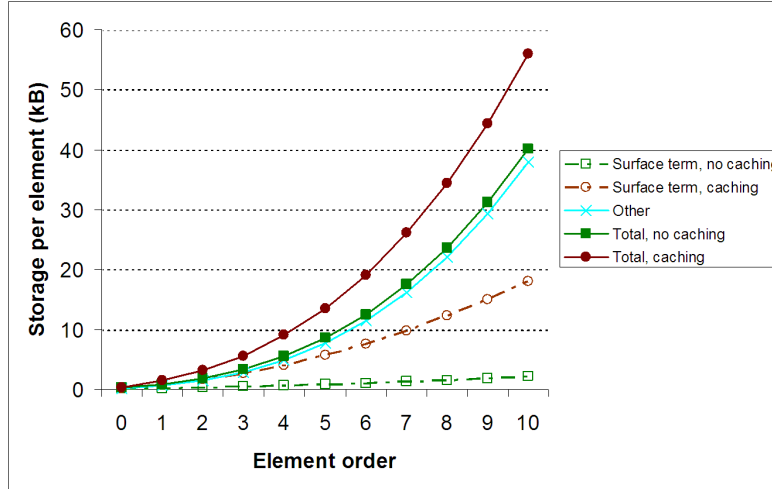


Figure 2. Amount of memory required, per tetrahedral element, for different caching schemes. Caching face data can increase the amount of memory by up to 70%.

5. Convergence properties

In the context of the present numerical study, the term “wave number” in the following refers to the number of complete wave cycles that exist in one meter of linear space. If h denotes the mesh size and k the dimensional wave number, we define a non-dimensional wave number as $k_h = kh$. The non-dimensional wave number is interpreted as a wave number where the length measure is taken as the mesh size, h . For example, a non-dimensional wave number of $kh = 1/5$ corresponds to a wave length of $5h$, i.e. the size of 5 elements. The important issue we aim to address here is the following. Considering a mesh of spacing h and polynomials of order p , what are scales that are correctly approximated by our DGM discretization ? In other words, which wave numbers are accurately approximated ?

The diffusive and dispersive properties of high order discontinuous Galerkin method have been studied extensively by Hu and Atkins¹⁵ and, more recently, by Ainsworth¹⁶. In¹⁶, Mark Ainsworth shows that there

exists three separate regimes (convergence rates) of error that correspond to ranges of non-dimensional wave numbers. meters of linear space.

The first range of wave numbers is the one of *resolved wave numbers*. It corresponds to relatively small values of k_h i.e. values of k_h that are such that $2p + 1 > (2\pi kh) + \mathcal{O}(kh)^{1/3}$. Resolved numerical wave numbers are super-convergent i.e. the relative error (the difference between the exact wave number and the numerical one) converges very rapidly :

$$\begin{aligned} \text{dispersion error} : & \quad \mathcal{O}((hk)^{2p+3}) \\ \text{dissipation error} : & \quad \mathcal{O}((hk)^{2p+2}). \end{aligned} \quad (31)$$

Typically, the imaginary part of k_h that corresponds to diffusion (or instabilities) is the one that dominates in the resolved range. In DGM, resolved waves are computed with a super-accuracy. With such a super-convergence rate, p should be chosen to be (one of) the smallest p that falls into the super convergence range. All higher p will lead to negligible gains in the absolute accuracy but will lead to substantial increase in computational cost.

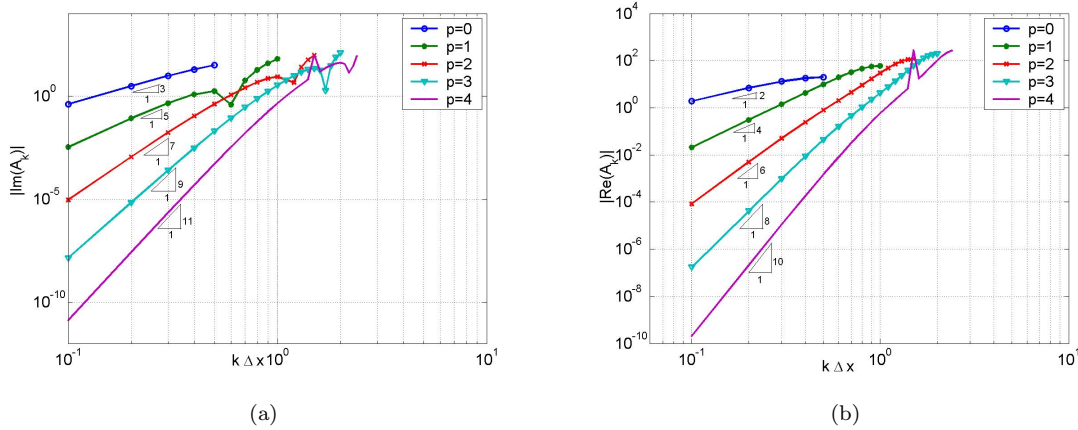


Figure 3. Plots of the error on the imaginary part (a) and real part (b) as a function of k_h for different values of p .

At the opposite extreme, for large wave numbers, when k_h is much larger than p , (more precisely when $2p + 1 < (2\pi kh) - \mathcal{O}(kh)^{1/3}$), there is a high relative error (high damping i.e. big imaginary part of k_h and typically $\mathcal{O}(100\%)$ of error on the real part of k_h). If the RK-DGM has to convect a wave corresponding to a large k_h , our RK time stepping scheme damps the wave very quickly. This is the feature of the DGM that gives it a key advantage over spectral methods, that is the DGM scheme is auto-adaptive: it self-filters unresolved modes while producing super-accuracy on resolved ones. The numerical dissipation of the DGM schemes acts like an adaptive filter.

The transition between those two extremes occurs when the order, p , lies in the following narrow range: $(2\pi kh) - \mathcal{O}(kh)^{1/3} < 2p + 1 < (2\pi kh) + \mathcal{O}(kh)^{1/3}$. In this region, the error is of order of 100% but decreases at an algebraic rate $(p)^{-1/3}$.

Figure 3 shows the DG numerical error in the real and imaginary parts of the numerical wave numbers as a function of k_h . We see that the super-convergence is observed below a certain wave number that depends on p . Here, we can decide that we aim to limit the error to 10^{-4} . Therefore, Figure 3 shows that it is necessary to use at least 10 elements per wavelength for $p = 2$, 6 elements per wavelength for $p = 4$ and 2 elements per wavelength for $p = 6$. As it was previously stated, the number of resolved modes grows roughly like $2p + 1$. With 10 elements per wavelength, we have 1 resolved mode for $p = 2$, 10/6 resolved modes for $p = 4$ and 5 resolved modes for $p = 6$. In theory, we should have 9/5 more resolved modes for $p = 4$ than for $p = 2$ and 13/5 more resolved modes for $p = 6$ than for $p = 2$.

Figure 4 shows the DGM eigenmodes in the space domain relative to several resolved and non resolved wave numbers. It is very interesting to see that resolved modes are smooth functions without inter-element jumps. Non-resolved modes exhibit jumps that become larger as k_h increases. The numerical damping of the DGM is known to be contained within the inter-element jumps.

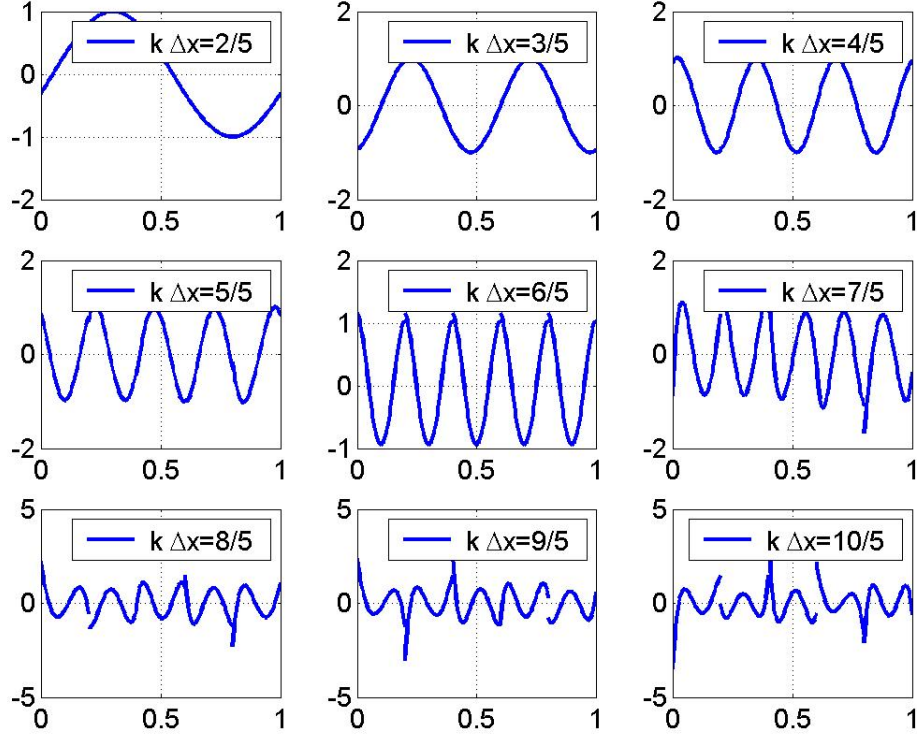


Figure 4. Representation of nine eigenvectors of \mathbf{A} . The mesh was equally spaced with $N = 5$, $p = 4$ and $\mu = 1$. The modes that correspond to small wave numbers are clearly well resolved. Modes corresponding to higher wave numbers are less resolved and inter-element jumps are appearing.

6. Time discretization

The time stepping scheme used here is a Runge-Kutta (RK) of order $p + 1$. The combination of RK of order $p + 1$ with DG at order p can be proven¹⁷ to be stable under the CFL condition

$$\Delta t < \frac{h}{c(2p + 1)}, \quad (32)$$

where h is the mesh size of element e and $c = \max(c_0 + \|\mathbf{v}_0\|)$ is the maximum wave velocity in e . Note that this bound is weak one when p is high. In other words, the RK(2p+1) stability range grows faster than the spectral radius of the DGM amplification matrix divided by $2p + 1$. Table 6 compares, in the 1D case, the spectral radius of the amplification matrix $S(\mathbf{A}_p)$ of the DGM(p) divided by $c(2p + 1)$ with the corresponding stability limit of RK(p+1). Of course, stability requires that, if we choose the proposed CFL condition (32), $S(\mathbf{A}_p)/(c(2p + 1)) \leq RK(p + 1)$. In the table, we see that the proposed CFL condition is sharp at low orders and becomes sub-optimal when p increases.

7. Boundary conditions

While open boundary conditions for acoustic problems in the time domain are still an open problem, an efficient and quite simple approach is the application of a one-dimensional characteristic boundary condition combined with a layer of dissipative elements to damp any remaining reflections. This is the approach we want to develop for our RK-DGM kernel. For the time being, the damping layer is not fully implemented yet, but according to our experience, the one dimensional characteristic variables as we re-develop them in the following, do not produce excessive reflections in the domain of interest. In the framework of our formulation,

p	$\frac{S(\mathbf{A}_p)}{c(2p+1)}$	$RK(p+1)$	p	$\frac{S(\mathbf{A}_p)}{c(2p+1)}$	$RK(p+1)$
0	2.00	2.00	7	4.10	4.31
1	2.00	2.00	8	4.42	4.70
2	2.36	2.57	9	4.72	5.06
3	2.74	2.78	10	5.02	5.45
4	3.10	3.21	11	5.32	5.82
5	3.44	3.55	12	5.62	6.20
6	3.78	3.95	13	5.90	6.57

Table 1. $S(\mathbf{A}_p)$ is the spectral radius of the amplification matrix of the Discontinuous Galerkin scheme for a given polynomial order p . $RK(p+1)$ is the norm of the maximal eigenvalue .

boundary condition are applied in a weak form by computing the boundary flux at each interpolation point. For the open boundary condition, we compute at each point the characteristics of the flow, and the characteristics that enter the domain are set to zero before computing the flux back in field variables. Let \mathbf{n} be the outward-pointing normal to the boundary, and \mathbf{s}, \mathbf{t} two vectors normal to \mathbf{n} , such that $(\mathbf{n}, \mathbf{s}, \mathbf{t})$ form an orthonormal basis on the boundary face. If we define \mathbf{R} and \mathbf{R}^{-1} by

$$\mathbf{R} = \frac{1}{2} \begin{pmatrix} \rho_0 c_0 & 0 & 0 & 0 \\ n_x & -n_x & t_x & s_x \\ n_y & -n_y & t_y & s_y \\ n_z & -n_z & t_z & s_z \end{pmatrix} \quad (33)$$

and

$$\mathbf{R}^{-1} = \frac{1}{2} \begin{pmatrix} \frac{1}{\rho_0 c_0} & n_x & n_y & n_z \\ \frac{1}{\rho_0 c_0} & -n_x & -n_y & -n_z \\ 0 & -t_x & -t_y & -t_z \\ 0 & -s_x & -s_y & -s_z \end{pmatrix}, \quad (34)$$

computing the characteristic variables c requires the multiplication of \mathbf{R}^{-1} with \mathbf{u} :

$$c = \begin{pmatrix} \frac{1}{\rho_0 c_0} p' + v'_n \\ \frac{1}{\rho_0 c_0} p' - v'_n \\ v'_t \\ v'_s \end{pmatrix} = \mathbf{R}^{-1} \mathbf{u}. \quad (35)$$

Characteristic fluxes eliminating the ingoing mode can then be set to zero by further multiplying by the following matrix:

$$\mathbf{K} = \begin{pmatrix} c_0 + \mathbf{v}_0 \mathbf{n} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & S \mathbf{v}_0 \mathbf{n} & 0 \\ 0 & 0 & 0 & S \mathbf{v}_0 \mathbf{n} \end{pmatrix}, \quad (36)$$

where \mathbf{S} is equal to 0 if $\mathbf{n} \cdot \mathbf{v}_0 < 0$ (the characteristics come from outside) and to 1 otherwise. Finally, we have that

$$\mathbf{F}_n = \mathbf{R} \mathbf{K} \mathbf{R}^{-1} \begin{pmatrix} p' \\ v'_x \\ v'_y \\ v'_z \end{pmatrix}. \quad (37)$$

To model the noise of the fan in engine application, we need to be able to couple duct modes representing the wave before the fan, and the domain where we solve the LEE in time domain. The approach we choose here, is to impose characteristics variable going inside the domain so that the imposed pressure fits the one from the duct, using the same characteristic approach (this is consistent with the approach presented in Section 3).

III. Validation

A. Test case

To validate our RK-DGM code, we consider the propagation of an acoustic mode in a 2D planar nacelle-like geometry, as depicted in Figure 5a. The geometry is actually a slice of a 3D nacelle that will be considered in section IV. The inner and outer radius of the fan are 0.14 m and 0.68 m , respectively. Since only characteristic non-reflecting boundary conditions were implemented at the time of this writing, a single incident plane wave is imposed at the fan face, at a frequency $f = 3\text{ kHz}$. The mean flow, depicted in Figure 5b is incompressible and such that $M = 0.2$ at the boundary while $M = 0.3$ at the fan face. The highest value of the Mach number, $M = 0.36$, is attained near the bottom lip.

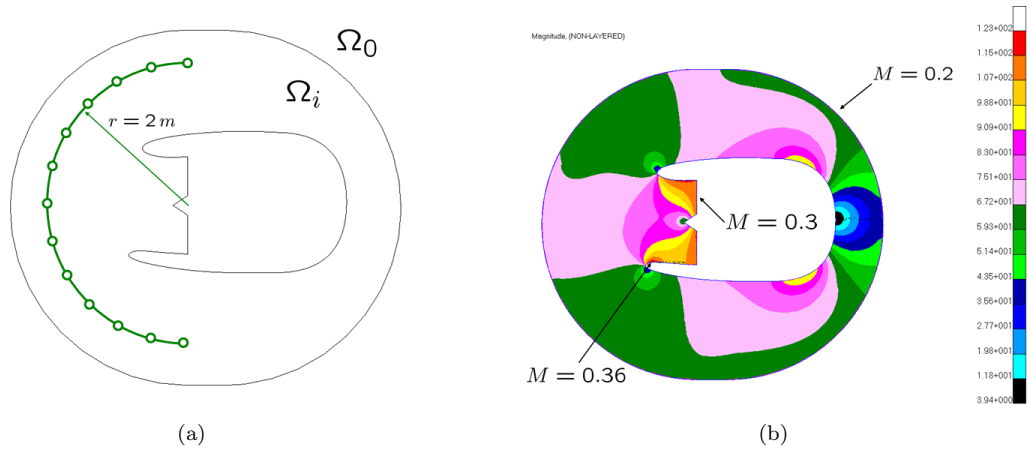


Figure 5. Test case for the 2D plane wave propagation. (a) Geometry and series of sampling points. The sampling points are located on a half-circle centered about the middle of the fan face with radius of 2 m . (b) Magnitude of the mean flow velocity. The maximum Mach number is 0.36 .

B. Results

The mesh used in the ACTRAN/TM simulation consists of 95,000 quadratic triangles and 642 infinite elements of order 40, for a total of 240,000 nodes (also equal to the number of degrees-of-freedom). In the RK-DGM simulation, a mesh with 9600 elements of order 6 was used, leading to around 1,000,000 degrees-of-freedom.

Figures 6 and 7 show the acoustic pressure in the computational domain obtained with ACTRAN/TM and our RK-DGM code. The acoustic field consists of two high-pressure zones located on both sides of the horizontal mid-plane. Note that the bottom lip of the nacelle, slightly longer than the top one, reflects some acoustic energy up front of the nacelle. The highest pressure values are observed in the region of highest Mach number.

Results compare rather well, if it weren't that the RK-DGM code seems to slightly overestimate the acoustic pressure. This is also observed in Figure 8, where SPL on the series of sampling points depicted in Figure 5 is shown. Here, the SPL curves are slightly shifted with respect to one another by a few degrees. We attribute this to the fact that, in the RK-DGM code, boundaries are approximated by straight lines while quadratic polynomials are used in ACTRAN/TM. Finally, no wave reflection is observed near the external boundary.

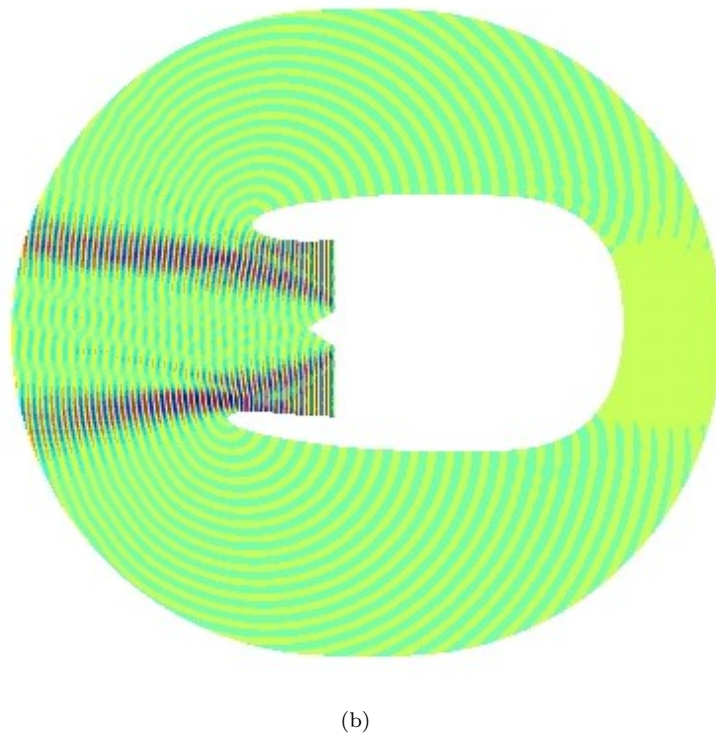
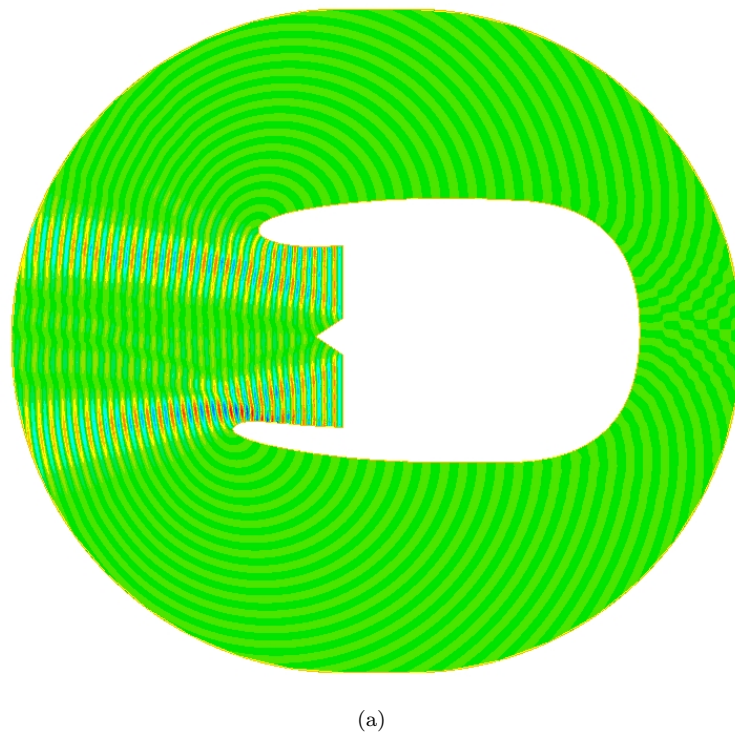


Figure 6. Instantaneous acoustic pressure for the 2D planar wave propagation problem. Results obtained with (a) ACTRAN/TM and (b) our RK-DGM code.

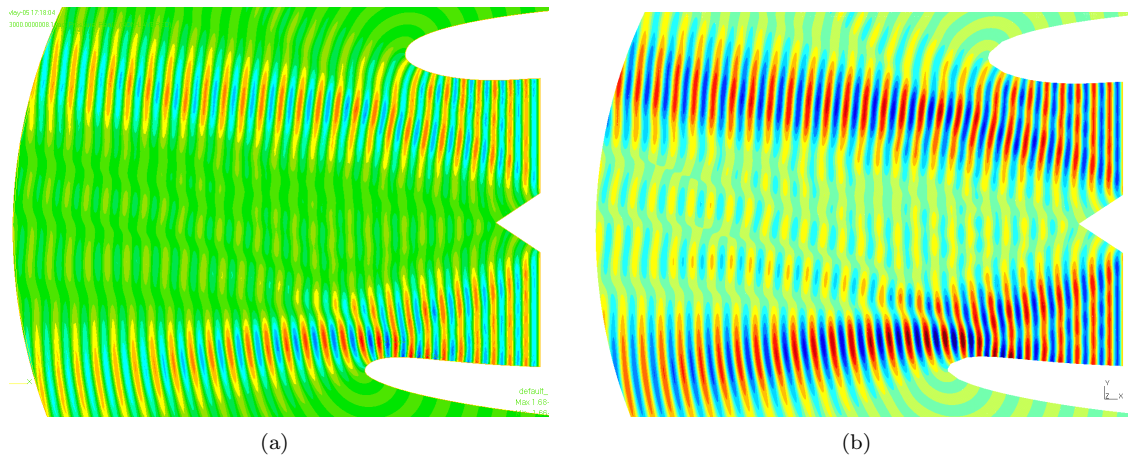


Figure 7. Instantaneous acoustic pressure for the 2D planar wave propagation problem. Results obtained with (a) ACTRAN/TM and (b) our RK-DGM code.

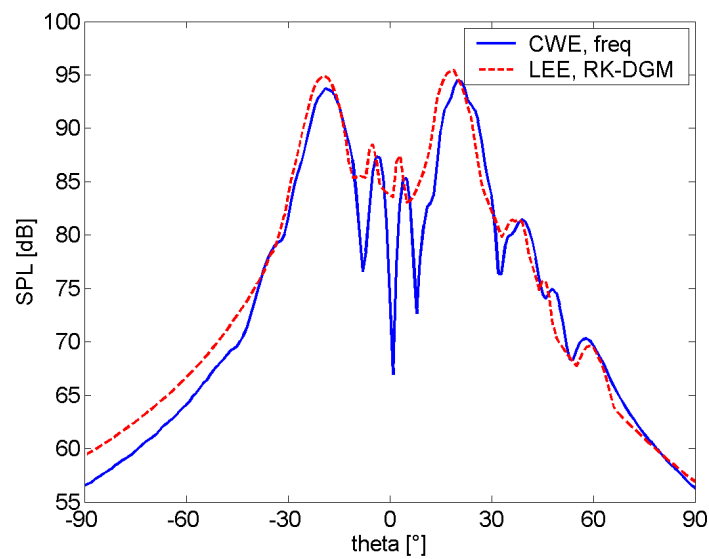


Figure 8. Amplitude of the acoustic pressure at the sampling points shown in Figure 5. The solid lines were obtained with ACTRAN/TM and the dashed line obtained with our RK-DGM code.

IV. Large-scale applications

A. Parallel implementation of the DG method

The parallel implementation of the DG method relies on the AOMD software package¹⁸ for the initial partitioning and load-balancing of the mesh. When the preprocessing phase is accomplished, each processor owns the degrees of freedom relative to the elements lying on its partition. The interface between two processors consists of a surface mesh which exists on both processors. Each element of this interface mesh knows on which processor and where it can find the two volume elements connected to itself. The coupling between elements for the explicit DG scheme is only done through the computation of the interface fluxes. No other coupling between elements appears in the mass matrix. Therefore, the parallelization of the scheme is rather simple, the only thing that needs to be done is to communicate left and right values of the field at interpolation points on each interface element of the partition boundary, prior to the computation of the fluxes on those interface elements. Such communications were implemented using the MPI interface. This compactness of the stencil is one of the nice features of the explicit DGM that make this method a good candidate for efficient parallelization. Since the amount of required communications is rather small when compared to the overall computations, a good parallel efficiency can be expected. To measure the scalability of the code, the following problem was setup. A cubic domain is divided in $40 \times 40 \times 40$ hexahedral elements, and we measure the time to solve one time step of the propagation of an acoustic pulse, with $p = 6$, where p is the polynomial order of the approximation. This problem has a total number of 69 millions of degrees of freedom ($40^3 * (p + 1)^3 * 4$). The time to solve one time step with one processor was $t_1 = 1194$ seconds on a 3.2 GHz Intel Xeon processor. Figure 9 shows the parallel speed-up, $4 * t_4 / t_n$, where n is the number of processors and t_4 the CPU time on 4 processors. Since the cluster we used is build on 2 processors nodes, and that we wanted to scale the results with a computation which actually use the communication through the Myrinet connections between nodes, we chose to scale the CPU time with respect to t_4 instead of t_1 . As shown on the figure, an almost linear scalability curve is obtained up to 64 processors, with a speed-up of 45.3 for 64 processors. Scaling with respect to t_1 gives a speed-up of 34.22 on 64 processors. While probably marginally improvable, we consider this results good enough for now.

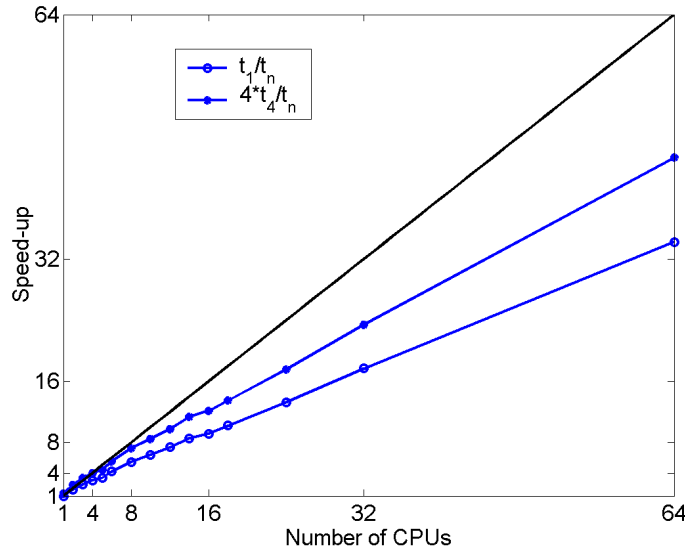


Figure 9. Plot of the parallel speed-up of our RK-DGM code, with a number of processors ranging from 1 to 64.

B. Test case 1

In this first test case, we study the in-duct propagation of an acoustic mode in a three-dimensional configuration consisting of the inside of an idealized, axi-symmetrical engine duct, with a conical tip to the fan.

The duct has a non-dimensional length of 2 and a maximum diameter of 1, sound speed and density are set to 1, and we consider the case with no mean flow. kr is set to $2\pi * 10 = 68.28$, and a mode of azimuthal order 2 and radial order 2 is imposed on the left side of the domain. The computational domain is limited to one quarter of the engine duct. The mesh contains 60,000 order elements and the polynomial order is set to 4, for a total of 8.4 millions of degrees-of-freedom. The simulation was run on 16 processors (Figure 10 depicts the mesh partitioning). Each partition contains the same number of elements and the number of interface elements is limited. The simulation was run until the acoustic wave reached the right side of the engine duct, at $t = 2$. Figure 11 shows maps of the pressure, on the left (a) at $t = 1$, on the right (b) at the end of the simulation. In both cases, isovalues are plotted on cross-sections located at $z = 0$, $y = 0$, $x = 0$, $x = 0.25$, $x = 0.5$ and $x = 0.75$. Results show that the wave propagates at the correct speed, with low dissipation. No side effect appears at the crossing of partition boundaries, which validates the parallel implementation.

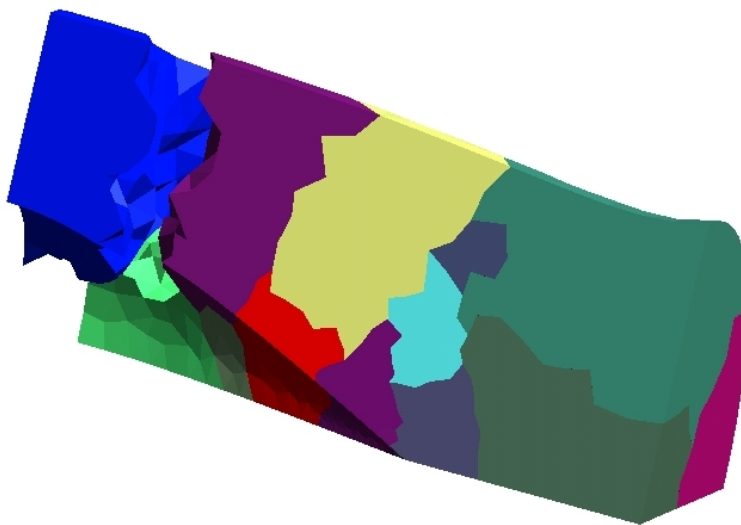


Figure 10. Partition of the mesh on 16 processors, with one color per processor

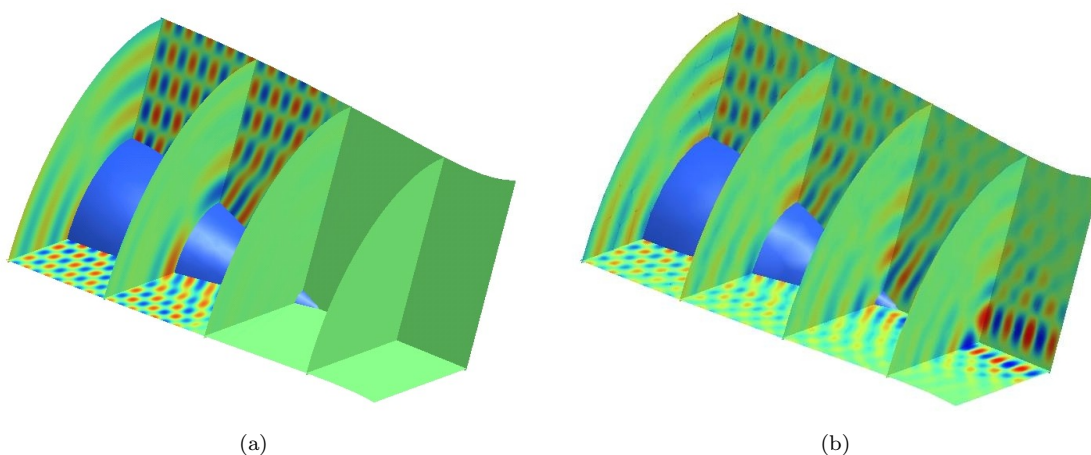


Figure 11. Pressure cut maps for the duct case, at $t = 1$ (a) and $t = 2$ (b)

The computation took 4 hours of wall-clock time, and 10,000 time step were needed, with a time step of $\Delta t = 2 \cdot 10^{-4}$.

Analysis of the mesh and the time step show that a better computation time could be obtained. The

maximum size of the element of the mesh is set so that we get a maximum of half a wave length per element. If all the elements of the mesh were of this same size, the time step would be equal to $4 \cot 10^{-3}$, so that the computation time would be reduced by a factor of 20. Unfortunately, to capture sharp features of the geometry, such as the tip of the cone, much smaller elements are needed. We are currently developing a strategy consisting in reducing the polynomial order on such small elements, where the accuracy is too high, thereby increasing the limit on the time step imposed by those elements. The first results are promising.

C. Test case 2

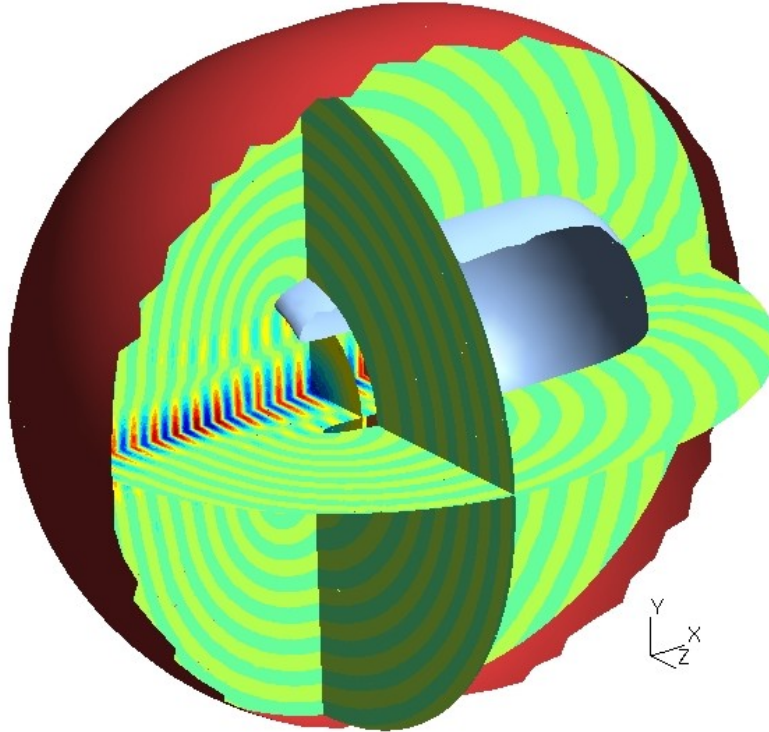


Figure 12. Pressure map for the 3d Nacelle with radial mode 1, azimuthal mode 0, $f = 1500$ Hz and a mean flow at $M = 0.3$ at the fan face.

The second test case deals with the propagation of acoustic modes upstream of a full three-dimensional nacelle. The geometry of the nacelle is actually the same as the one we took a slice from for the two-dimensional validation (see Section III). The mesh consists of 124,000 tetrahedra of order $p = 6$, for a total of 62 millions degrees of freedoms. 32 3.2 GHz Xeon processors were used, and the simulation ran for 2 days. In Figure 12, we present the solution for a mean flow of Mach number $M = 0.3$ at the fan face, an azimuthal mode order set to 1, a radial order equal to 0, and a frequency of 1,500 Hz ($kr = 18$). We show the map of the solution on cutting planes located at $y = 0$, $z = 0$ and $x = 3.3$. The pressure wave is well captured, the compression of the wave length due to the mean flow is clearly visible when comparing the wave lengths in front of the engine to wave length at the back. No effect of numerical reflection of the wave on the free boundary are visible.

In Figure 13, we plot the solution for a mode of azimuthal order 2 and radial order 2, a frequency of 2,000 Hz ($kr = 24$) with no mean flow. Here again, results look good. No jump between elements are visible even at high zooming factor, which is a very good visual clue of the convergence of the results. On the plane located at $x = 3.3$, the shape of the radial order 2 mode is clearly visible. The cut off of the pressure amplitude on the x -axis, directly in front of the engine corresponds to the results observed in two dimensions.

The three-dimensional results are so far rather encouraging. Nevertheless, more simulations and verifications are required to fully validate our implementation.

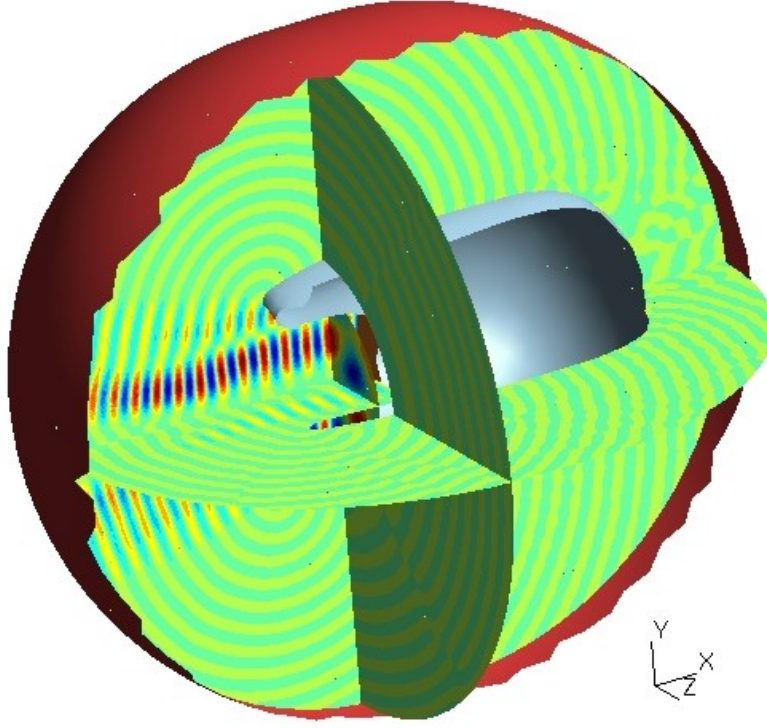


Figure 13. Pressure map for the 3d Nacelle with radial mode 2 azimuthal mode 2, frequency of 2000Hz and no mean flow.

V. Conclusions

In this paper we gave a description of the ongoing development of a discontinuous Galerkin kernel to solve large-scale aero-acoustics problems. An analysis of the efficiency of the method shows that the highest computational performance is obtained with meshes consisting of a small number of high-order elements, provided a good implementation of the quadrature-free version of the scheme be used. The compactness of the DG scheme allows for a relatively easy implementation for such a high-order method, even in the parallel case, with good scalability properties. We consider that this work as shown that the RK-DG method was a good candidate to complement the more classical frequency domain approach implemented in ACTRAN/TM, in particular for high value of the Helmholtz number. If some elements of validation have been proposed here, by direct comparison with an established solver, more testing is still needed to use the DGM kernel in an industrial context. Further developments include better boundary conditions and improving CPU time by the adaptive polynomial order approach mentioned in Section IV.

VI. Acknowledgments

We gratefully acknowledge the European Community for supporting this work through the MESSIAEN project. Support also came in the form of top computing facilities. In this connection, we would like to thank the CENAERO research center (www.cenaero.be)

References

- ¹Myers, K. K., "On the acoustic boundary condition in the presence of flow," *J. Sound Vib.*, Vol. 71, 1980, pp. 429–434.
- ²Eversman, W., "The boundary condition at an impedance wall in a non-uniform duct with potential mean flow," *J. Sound Vib.*, Vol. 246, No. 1, 2001, pp. 63–69.
- ³Atkins, H. L. and Shu, C. W., "Quadrature-free implementation of discontinuous Galerkin methods for hyperbolic equations," *AIAA Journal*, Vol. 36, No. 5, 1998.

- ⁴Atkins, H. L., "Continuous development of the discontinuous Galerkin method for aeroacoustic applications," *Proceeding of the 3rd AIAA/CEAS Aeroacoustics Conference*, No. AIAA Paper 1997-1581, Atlanta, GA, 1997.
- ⁵Lockard, D. and Atkins, H., "Efficient implementations of the quadrature-free discontinuous Galerkin method," *Proceeding of 14th AIAA CFD conference*, AIAA, 1999.
- ⁶Cockburn, B. and Shu, C.-H., "TVD Runge-Kutta local projection Discontinuous Galerkin finite element method for conservation laws," *Mathematics of Computations*, Vol. 52(186), 1989, pp. 411–435.
- ⁷Atkins, H. L., "Application of a Generalized Quadrature-Free Discontinuous Galerkin Method in aeroacoustics," *Proceeding of the 9th AIAA/CEAS Aeroacoustics Conference*, No. AIAA Paper 2003-3120, Hilton Head, SC, 2003.
- ⁸Astley, R. J. and Coyette, J.-P., "Conditioning of infinite element schemes for wave problems," *Commun. Numer. Meth. Engng*, Vol. 17, 2001, pp. 31–41.
- ⁹Astley, R. J. and Coyette, J.-P., "The performance of spheroidal infinite elements," *Int. J. Numer. Meth. Engng*, Vol. 52, 2001, pp. 1379–1396.
- ¹⁰Astley, R. J., Hamilton, J. A., Baker, N., and Kitchen, A. H., "Modelling tone propagation from turbofan inlets - The effect of extended lip liners," 2002, AIAA Paper 2002-2449.
- ¹¹Ploumhans, P., Meerbergen, K., Knapen, T., Gallez, X., Lielens, G., and Coyette, J.-P., "Development and Validation of a Parallel Out-of-core Propagation and Radiation Code With Validation on a Turbofan Application," *Proceedings of the Eighteenth International Congress on Acoustics*, ICA, 2004.
- ¹²Rienstra, S. W. and Hirschberg, A., *An Introduction to Acoustics*, 2002.
- ¹³Remacle, J.-F., Flaherty, J., and Shephard, M., "An Adaptive Discontinuous Galerkin Technique with an Orthogonal Basis Applied to Compressible Flow Problems," *SIAM Review*, Vol. 45, 2003, pp. 53–72.
- ¹⁴"ATLAS (Automatically Tuned Linear Algebra Software)," .
- ¹⁵Hu, F. and Atkins, H., "Eigensolution Analysis of the Discontinuous Galerkin Method with Nonuniform Grids I One Space Dimension," *Journal of Computational Physics*, Vol. 182, No. 2, 2002, pp. 516–545.
- ¹⁶Ainsworth, M., "Dispersive and dissipative behavior of high order Discontinuous Galerkin finite element methods," *Journal of Computational Physics*, Vol. 198, No. 1, 2004, pp. 106–130.
- ¹⁷Cockburn, B. and Shu, C.-W., "Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, Vol. 16, 2001, pp. 173–261.
- ¹⁸Remacle, J.-F., Flaherty, J. E., and Shephard, M. S., "Parallel Algorithm Oriented Mesh Database," *Tenth International Meshing Roundtable*, 2001.