A Discontinuous Galerkin Method for Computational AeroAcoustics

Christophe Peyret and Phil Delorme
Introduction
Discontinuous Galerkin Method
Discontinuous Galerkin Method

• It is «Expensive» but...
Discontinuous Galerkin Method

• It is «Expensive» but...
• Variational Formulation
Discontinuous Galerkin Method

- It is «Expensive» but...
- Variational Formulation
- Any kind of Meshes
Discontinuous Galerkin Method

- It is «Expensive» but...
- Variational Formulation
- Any kind of Meshes
- Local Order Adaptation «p adapt»
Discontinuous Galerkin Method

- It is «Expensive» but...
- Variational Formulation
- Any kind of Meshes
- Local Order Adaptation «p adapt»
- Local Mesh Adaptation «h adapt»
Discontinuous Galerkin Method

- It is «Expensive» but...
- Variational Formulation
- Any kind of Meshes
- Local Order Adaptation «p adapt»
- Local Mesh Adaptation «h adapt»
- Local Time Step Adaptation «t adapt»
Discontinuous Galerkin Method

- It is «Expensive» but...
- Variational Formulation
- Any kind of Meshes
- Local Order Adaptation «p adapt»
- Local Mesh Adaptation «h adapt»
- Local Time Step Adaptation «t adapt»
- Local Physical Model Adaptation «phi adapt»
Discontinuous Galerkin Method

- It is «Expensive» but...
- Variational Formulation
- Any kind of Meshes
- Local Order Adaptation «p adapt»
- Local Mesh Adaptation «h adapt»
- Local Time Step Adaptation «t adapt»
- Local Physical Model Adaptation «phi adapt»
- Adapted Functional Basis
FEM Resolution of Galbrun’s With No Flow

\[ G(\xi) = \rho_0 \, D_t^2 \xi - \nabla (\rho_0 a_0^2 \, \nabla \cdot \xi) - T \, \nabla \xi \cdot \nabla p_0 + \nabla \cdot \xi \, \nabla p_0 = 0 \]

\[ H_1 \quad \text{H}_{\text{div}} \]
FEM Resolution of Galbrun's With No Flow

\[ G(\xi) = \rho_0 \ D_t^2 \xi - \nabla (\rho_0 a_0^2 \ \nabla \cdot \xi) - T \ \nabla \xi \cdot \nabla p_0 + \nabla \cdot \xi \ \nabla p_0 = 0 \]
FEM Resolution of Galbrun’s With No Flow

\[ G(\xi) = \rho_0 \, D^2_t \xi - \nabla (\rho_0 a_0^2 \, \nabla \cdot \xi) - T \, \nabla \xi \cdot \nabla p_0 + \nabla \cdot \xi \, \nabla p_0 = 0 \]

**H\(_1\)**

![Graph](https://via.placeholder.com/150)

**H\(_{\text{div}}\)**

![Graph](https://via.placeholder.com/150)
Physical Modeling and Mathematical formulation
(Pierre-Alain Mazet - Phil Delorme)
Physical Modeling and Mathematical formulation
(Pierre-Alain Mazet - Phil Delorme)

Linearized Euler’s Equations
Physical Modeling and Mathematical formulation
(Pierre-Alain Mazet - Phil Delorme)

Linearized Euler’s Equations

\[ \varphi = \begin{pmatrix} u_1 \\ v_1 \\ a_0 \rho_1 \rho_0 \end{pmatrix} \]
Physical Modeling and Mathematical formulation
(Pierre-Alain Mazet - Phil Delorme)

Linearized Euler’s Equations

Symmetric Friedrich System

\[ \frac{\partial \varphi}{\partial t} + A_i \frac{\partial \varphi}{\partial i} + B \varphi = 0 \]

Matrix \( A_i \frac{\partial}{\partial i} \) is symmetric

\[ \varphi = \begin{pmatrix} u_1 \\ v_1 \\ a_0 \rho_1/\rho_0 \end{pmatrix} \]
Physical Modeling and Mathematical formulation
(Pierre-Alain Mazet - Phil Delorme)

Linearized Euler's Equations

\[ \frac{\partial \varphi}{\partial t} + A_i \frac{\partial}{\partial x_i} \varphi + B \varphi = 0 \]

Symmetric Friedrich System
Matrix $A_i \frac{\partial}{\partial x_i}$ is symmetric

Variational formulation
\[ \{ \varphi_h \in W_k(\omega_h) \mid \forall \psi_h \in W_k(\omega_h) ; L(\varphi_h, \psi_h) = 0 \} \]
Physical Modeling and Mathematical formulation
(Pierre-Alain Mazet - Phil Delorme)

Linearized Euler’s Equations

Symetric Friedrich System
\[ \partial_t \varphi + A_i \partial_i \varphi + B \varphi = 0 \]
Matrix \( A_i \partial_i \) is symetric

Variational formulation
\[ \varphi_h \in W^k(\omega_h) \mid \forall \psi_h \in W^k(\omega_h) ; L(\varphi_h, \psi_h) = 0 \]

\[ \varphi = \begin{pmatrix} u_1 \\ v_1 \\ a_0 \rho_1/\rho_0 \end{pmatrix} \]

\[ A_{in_i} \text{ is diagonalizable} \]
\[ A_{in_i} = [A_{in_i}]^+ + [A_{in_i}]^- \]
Physical Modeling and Mathematical formulation
(Pierre-Alain Mazet - Phil Delorme)

Linearized Euler’s Equations

\[ \phi_h = \Omega \psi_h. \partial_t \phi_h + \Omega A_i \partial_i \phi_h + B \phi = 0 \]

Symmetric Friedrich System
Matrix \( A_i \partial_i \) is symmetric

Variational formulation

\[ \phi_h \in W^k(\omega_h) \mid \forall \psi_h \in W^k(\omega_h) ; \mathcal{L}(\phi_h, \psi_h) = 0 \]

Variational formulation

\[ \mathcal{L}(\phi_h, \psi_h) = \int_\Omega \psi_h \cdot \partial_t \phi_h + \int_\Omega \psi_h \cdot A_i \partial_i \phi_h + \int_\Omega \psi_h \cdot B \phi_h \]

\[ + \int_{\partial \omega_h \cap \partial \Omega} \psi_h \cdot [A_i n_i]^- (\phi_h^0 - \phi_h^i) + \int_{\partial \omega_h \cap \partial \Omega} \psi_h \cdot (M \phi_h - g) - \int_\Omega \psi_h \cdot g \]

\[ \phi = \begin{pmatrix} u_1 \\ v_1 \\ a_0 \rho_1 / \rho_0 \end{pmatrix} \]

\( A_i n_i \) is diagonalizable

\( A_i n_i = [A_i n_i]^+ + [A_i n_i]^- \)
Physical Modeling and Mathematical formulation
(Pierre-Alain Mazet - Phil Delorme)

**Linearized Euler’s Equations**

\[ \varphi = \begin{pmatrix} u_1 \\ v_1 \\ a_0 \rho_1 / \rho_0 \end{pmatrix} \]

**Symmetric Friedrich System**

\[ \partial_t \varphi + A_i \partial_i \varphi + B \varphi = 0 \]

Matrix \( A_i \partial_i \) is symmetric

**Variational formulation**

\[ \begin{cases} \varphi_h \in W^k(\omega_h) \mid \forall \psi_h \in W^k(\omega_h) ; \mathcal{L}(\varphi_h, \psi_h) = 0 \end{cases} \]

\[ \mathcal{L}(\varphi_h, \psi_h) = \int_\Omega \psi_h \cdot \partial_t \varphi_h + \int_\Omega \psi_h \cdot A_i \partial_i \varphi_h + \int_\Omega \psi_h \cdot B \varphi_h \]

\[ + \int_{\partial \omega_h \cap \partial \Omega} \psi_h \cdot [A_i \cdot n_i]^- (\varphi^0_h - \varphi^i_h) + \int_{\partial \omega_h \cap \partial \Omega} \psi_h \cdot (M \varphi_h - g) - \int_\Omega \psi_h \cdot g \]

**Fully Upwind Scheme**
Discretization and Computation
Discretization and Computation

\[ Y^T \times (\mathcal{M} \partial_t X + S \times X) = Y^T \times V \]
Discretization and Computation

\[ Y^T \times (\mathcal{M} \partial_t X + S \times X) = Y^T \times V \]

- Time dependant problem:

\[ \partial_t \varphi^k = \mathcal{M}^{-1} \times (b(t) - S \times \varphi^k) = f(t_k, \varphi^k) \]
Discretization and Computation

\[ Y^T \times (\mathcal{M} \, \partial_t \, X + S \times X) = Y^T \times V \]

- Time dependant problem:

\[ \partial_t \varphi^k = \mathcal{M}^{-1} \times (b(t) - S \times \varphi^k) = f(t_k, \varphi^k) \]

Using a second order Runge-Kutta

\[ \begin{align*}
\delta_1 &= \Delta t \, f(t_k, \varphi^k) \\
\delta_2 &= \Delta t \, f(t_k + \Delta t/2, \varphi^k + \delta_1/2) \\
\Rightarrow \varphi^{k+1} &= \varphi^k + \delta_1 + \delta_2 + O(\Delta t^3)
\end{align*} \]
Discretization and Computation

\[ Y^T \times (M \partial_t X + S \times X) = Y^T \times V \]

- Time dependant problem:

\[ \partial_t \varphi^k = M^{-1} \times (b(t) - S \times \varphi^k) = f(t_k, \varphi^k) \]

using a second order Runge-Kutta

\[ \delta_1 = \Delta t \, f(t_k, \varphi^k) \]
\[ \delta_2 = \Delta t \, f(t_k + \Delta t/2, \varphi^k + \delta_1/2) \]

\[ \Rightarrow \varphi^{k+1} = \varphi^k + \delta_1 + \delta_2 + O(\Delta t^3) \]

- Harmonic dependant problem:

\[ (j \omega M + S) \times \varphi^k = b(t) = f(t_k, \varphi^k) \]
Discretization and Computation

\[ Y^T \times (M \partial_t X + S \times X) = Y^T \times V \]

- Time dependant problem:

\[ \partial_t \varphi^k = M^{-1} \times (b(t) - S \times \varphi^k) = f(t_k, \varphi^k) \]

using a second order Runge-Kutta

\[ \delta_1 = \Delta t \; f(t_k, \varphi^k) \]
\[ \delta_2 = \Delta t \; f(t_k + \Delta t/2, \varphi^k + \delta_1/2) \]

\[ \Rightarrow \varphi^{k+1} = \varphi^k + \delta_1 + \delta_2 + O(\Delta t^3) \]

- Harmonic dependant problem:

\[ (j\omega M + S) \times \varphi^k = b(t) = f(t_k, \varphi^k) \]

Resolution of a complex linear sparse system
«p» refinement in DG
Remeshing Tool

Original nv=67108  nt=134212

Adapted nv=9796   nt=19588
Optimal hp DGM CAA Meshes

Optimal Mesh: 3729 triangles

CFD Mesh: 17024 triangles

lundi 7 décembre 2009
Order Adaptation

\[ f = 2 \text{ kHz} \]
\[ f = 2 \text{ kHz} \]
CAA Computation

\[ f = 2 \text{ kHz} \]
CAA Computation

$f = 2 \text{ kHz}$

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>iter</td>
<td>1456</td>
</tr>
<tr>
<td>$dt$</td>
<td>4.46 $\mu$s</td>
</tr>
<tr>
<td>$t$</td>
<td>6504 $\mu$s</td>
</tr>
<tr>
<td>CPU</td>
<td>580 s</td>
</tr>
<tr>
<td>RAM</td>
<td>300 MO</td>
</tr>
</tbody>
</table>

CAA Computation

lundi 7 décembre 2009
CAA Computation

\[ f = 2 \text{ kHz} \]

\[ \begin{array}{|c|c|}
\hline
\text{iter} & 1456 \\
\hline
\text{dt} & 4.46 \mu s \\
\hline
\text{t} & 6504 \mu s \\
\hline
\text{CPU} & 580'' \\
\hline
\text{RAM} & 300 \text{ MO} \\
\hline
\end{array} \]

\[ \begin{array}{|c|c|}
\hline
\text{iter} & 1200 \\
\hline
\text{dt} & 5.42 \mu s \\
\hline
\text{t} & 6507 \mu s \\
\hline
\text{CPU} & 195'' \\
\hline
\text{RAM} & 200 \text{ MO} \\
\hline
\end{array} \]
Mesh Adaptation Application
Adaptation Loop

Auteurs : Frédéric Alauzet, Pascal Frey

\[(\mathcal{T}_0, \mathcal{S}_0^0)\]

Boucle interne

Interpolation solution
\[\mathcal{S}_{i,j+1}^0\]

Génération maillage
\[\mathcal{T}_{i,j+1}\]

Calcul métrique
\[\mathcal{M}_{i,j} = \bigcap_{k=1}^{m} \mathcal{M}_{i,j}^k\]

\[\mathcal{T}_{i,j}, \mathcal{S}_{i,j}\]

\[\mathcal{T}_{i+1}, \mathcal{S}_i, \mathcal{T}_i\]

\[\mathcal{T}_{i+1}, \mathcal{S}_i, \mathcal{T}_i\]

\[\mathcal{T}_{i+1}, \mathcal{S}_i, \mathcal{T}_i\]
Mesh Adaptation
Industrial Application
Acoustic propagation: exhaust
Acoustic propagation: exhaust
Acoustic propagation : exhaust
Acoustic propagation : exhaust
Acoustic propagation: exhaust
Acoustic propagation : exhaust

Aitec A12 : mode (0,1) f=1250 Hz
Acoustic propagation: exhaust

Aitec A12: mode (0,1) f=1250 Hz

Aitec A12: mode (1,1) f=1250 Hz

Time: 0.009168 s

lundi 7 décembre 2009
Acoustic propagation: exhaust

3D complex Geometry

Aitec A12: mode (0,1) f=1250 Hz

Aitec A12: mode (1,1) f=1250 Hz

Time: 0.009168 s

lundi 7 décembre 2009
Acoustic propagation: exhaust

3D complex Geometry
Liners
Acoustic propagation: exhaust

- 3D complex Geometry
- Liners
- Analytic Sources
How to Integrate CAA DG Solver with other CAA Solvers?

Mach Framework: «CAA Simulation Hub»
Objective:
Objective:

🌞 Full Aeroacoustic Computation (PDE based)
Objective:

Full Aeroacoustic Computation (PDE based)

Means:
Mach framework «CAA Simulation Hub»

Objective:

ках Full Aeroacoustic Computation (PDE based)

Means:

ках Coupling DG and FD numerical methods on hybrid structured/unstructured meshes for an efficient CAA solver
Objective:

🌈 Full Aeroacoustic Computation (PDE based)

Means:

🌈 Coupling DG and FD numerical methods on hybrid structured/unstructured meshes for an efficient CAA solver
🌈 Coupling CFD and CAA solvers (hybridation of physical model)
First Step Achieved (Collaboration PS3A+ELCI)
Parallel Multi-Coupled DG LEE Computations
First Step Achieved (Collaboration PS3A+ELCI)
Parallel Multi-Coupled DG LEE Computations

Time: 0.005755 s

mpirun --prefix /usr/local/openmpi-1.3-v11-i64/
-hostfile -peyret/hostfile
-n 2 space64 -f Test/Couplage4/centre.def -fast -noscreen : 
-n 2 space64 -f Test/Couplage4/bas.def -fast -noscreen :
-n 2 space64 -f Test/Couplage4/droite.def -fast -noscreen :
-n 2 space64 -f Test/Couplage4/haut.def -fast -noscreen :
-n 2 space64 -f Test/Couplage4/gauche.def -fast -noscreen

lundi 7 décembre 2009
Second Step in Progress (PHD Leger+Piperno)
Parallel Multi-Coupled DG-FD LEE Computations

Time: 0.002869
Second Step in Progress (PHD Leger+Piperno)
Parallel Multi-Coupled DG-FD LEE Computations

Time: 0.004426

lundi 7 décembre 2009
The End