



HAL
open science

SEM-DG Approximation for elasto-acoustics

Hélène Barucq, Henri Calandra, Aurélien Citrain, Julien Diaz, Christian Gout

► **To cite this version:**

Hélène Barucq, Henri Calandra, Aurélien Citrain, Julien Diaz, Christian Gout. SEM-DG Approximation for elasto-acoustics. MATHIAS 2018 Computational Science Engineering & Data Science by TOTAL, Oct 2018, Serris, France. hal-01872812v2

HAL Id: hal-01872812

<https://hal.science/hal-01872812v2>

Submitted on 29 Oct 2018

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.

SEM-DG Approximation for elasto-acoustics

Hélène Barucq¹, Henri Calandra², Aurélien Citrain^{3,1}, Julien Diaz¹ and Christian Gout³

¹ Team project Magique.3D, INRIA, E2S UPPA, CNRS, Pau, France.

² TOTAL SA, CSTJF, Pau, France.

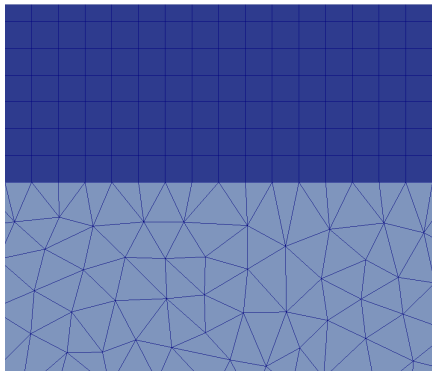
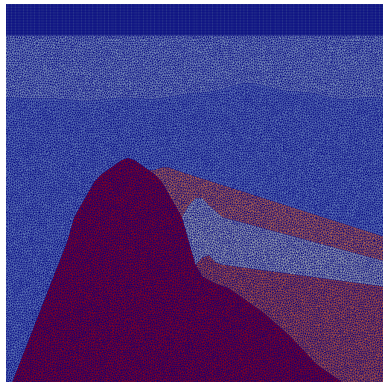
³ INSA Rouen-Normandie Université, LMI EA 3226, 76000, Rouen.

MATHIAS 2018 October 22-24



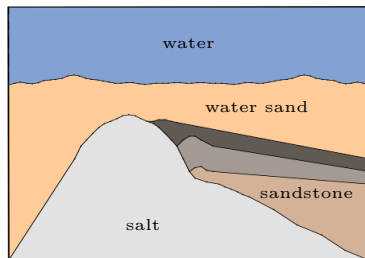
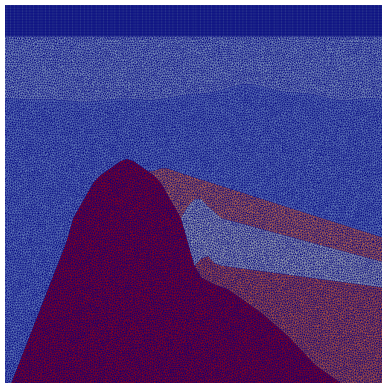
The authors thank the M2NUM project which is co-financed by the European Union with the European regional development fund (ERDF, HN0002137) and by the Normandie Regional Council.

Why using hybrid meshes?



- Useful when the use of unstructured grid is non-sense (e.g. medium with a layer of water).
- Well suited for the coupling of numerical methods in order to reduce the computational cost and improve the accuracy.

Why using hybrid meshes?



- Useful when the use of unstructured grid is non-sense (e.g. medium with a layer of water).
- Well suited for the coupling of numerical methods in order to reduce the computational cost and improve the accuracy.

$x \in \Omega \subset \mathbb{R}^d$, $t \in [0, T]$, $T > 0$:

$$\begin{cases} \rho(x) \frac{\partial v}{\partial t}(x, t) &= \nabla \cdot \underline{\underline{\sigma}}(x, t), \\ \frac{\partial \underline{\underline{\sigma}}}{\partial t}(x, t) &= \underline{\underline{C}}(x) \underline{\underline{\epsilon}}(v(x, t)). \end{cases}$$

With:

- $\rho(x)$ the density,
- $\underline{\underline{C}}(x)$ the elasticity tensor,
- $\underline{\underline{\epsilon}}(x, t)$ the deformation tensor,
- $v(x, t)$, the wavespeed,
- $\underline{\underline{\sigma}}(x, t)$ the strain tensor.

Software written in **Fortran** for wave propagation simulation in the **time domain**

Features

Simulation:

- on various types of meshes (**unstructured triangles and tetrahedra**),
 - on **heterogeneous media (acoustic, elastic and elasto-acoustic)**.
-
- **Discontinuous Galerkin (DG)** based on **unstructured triangles and unstructured tetrahedra**,
 - with various time-schemes : **Runge-Kutta (2 or 4), Leap-Frog**,
 - with **multi-order** computation(**p-adaptivity**)...

- 1 Numerical Methods
- 2 Comparison DG/SEM on structured quadrangle mesh
- 3 DG/SEM coupling
- 4 3D extension

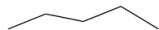
1 Numerical Methods

- Discontinuous Galerkin Method (DG)
- Spectral Element Method (SEM)
- Advantages of each method

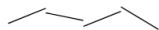
Use discontinuous functions :



mesh

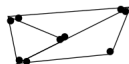


continuous

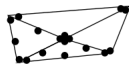


discontinuous

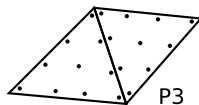
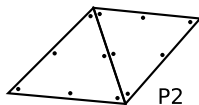
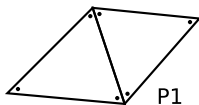
h adaptivity :



p adaptivity :

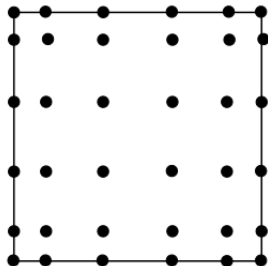


Degrees of freedom necessary on each cell :



General principle

- Finite Element Method (FEM) discretization + Gauss-Lobatto quadrature,
- Gauss-Lobatto points as degrees of freedom (gives us exponential convergence on L^2 -norm).



- $\int f(x) dx \approx \sum_{j=1}^{N+1} \omega_j f(\xi_j),$
- $\varphi_i(\xi_j) = \delta_{ij}.$

DG

- Element per element computation (*hp*-adaptivity).
- Time discretization quasi explicit (block diagonal mass matrix).
- Simple to parallelize.
- Robust to brutal changes of physics and geometry

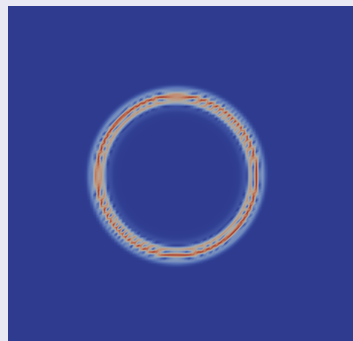
SEM

- Couples the flexibility of FEM with the accuracy of the pseudo-spectral method.
- Simplifies the mass and stiffness matrices (mass matrix diagonal).

2 Comparison DG/SEM on structured quadrangle mesh

- Description of the test cases
- Comparative tables

Physical parameters



P wavespeed	1000 m.s^{-1}
Density	1 kg.m^{-3}

Second order **Ricker Source** in P wave
($f_{peak} = 10\text{Hz}$)

General context

- **Acoustic homogeneous** medium.
- Four different meshes : **10000 cells**, **22500 cells**, **90000 cells**, **250000 cells**.
- CFL computed using **power iteration** method.
- **Leap-Frog** time scheme.
- **Eight threads** parallel execution with **OpenMP**.

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0$)	3.18e-3	2e-1	5.13	629
SEM	4.9e-3	5e-2	0.80	409

Figure: DG not penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0$)	2.12e-3	7e-1	18.11	943
SEM	3.26e-3	4e-2	3.54	613

Figure: DG not penalized and SEM comparison on the 20000 cells case

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0$)	3.18e-3	2e-1	5.13	629
SEM	4.9e-3	5e-2	0.80	409

Figure: DG not penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0$)	2.12e-3	7e-1	18.11	943
SEM	3.26e-3	4e-2	3.54	613

Figure: DG not penalized and SEM comparison on the 20000 cells case

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0$)	3.18e-3	2e-1	5.13	629
SEM	4.9e-3	5e-2	0.80	409

Figure: DG not penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0$)	2.12e-3	7e-1	18.11	943
SEM	3.26e-3	4e-2	3.54	613

Figure: DG not penalized and SEM comparison on the 20000 cells case

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	4.9e-3	5e-2	0.80	409

Figure: DG penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	3.26e-3	4e-2	3.54	613

Figure: DG penalized and SEM comparison on the 20000 cells case

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	4.9e-3	5e-2	0.80	409

Figure: DG penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	3.26e-3	4e-2	3.54	613

Figure: DG penalized and SEM comparison on the 20000 cells case

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	4.9e-3	5e-2	0.80	409

Figure: DG penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	3.26e-3	4e-2	3.54	613

Figure: DG penalized and SEM comparison on the 20000 cells case

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	4.9e-3	5e-2	0.80	409

Figure: DG penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	3.26e-3	4e-2	3.54	613

Figure: DG penalized and SEM comparison on the 20000 cells case

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	2e-3	3e-2	2.12	1000

Figure: DG penalized and SEM comparison using the same CFL on a 10000 thousands cells mesh

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG ($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	1.33e-3	2e-2	8.67	1502

Figure: DG penalized and SEM comparison using the same CFL on a 20000 thousands cells mesh

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	2e-3	3e-2	2.12	1000

Figure: DG penalized and SEM comparison using the same CFL on a 10000 thousands cells mesh

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG ($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	1.33e-3	2e-2	8.67	1502

Figure: DG penalized and SEM comparison using the same CFL on a 20000 thousands cells mesh

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	2e-3	3e-2	2.12	1000

Figure: DG penalized and SEM comparison using the same CFL on a 10000 thousands cells mesh

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG ($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	1.33e-3	2e-2	8.67	1502

Figure: DG penalized and SEM comparison using the same CFL on a 20000 thousands cells mesh

DG

- Element per element computation (*hp*-adaptivity).
- Time discretization quasi explicit (block diagonal mass matrix).
- Simple to parallelize.
- Robust to brutal changes of physics and geometry

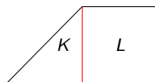
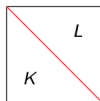
SEM

- Couples the flexibility of FEM with the accuracy of the pseudo-spectral method.
- Simplifies the mass and stiffness matrices (mass matrix diagonal).
- **Reduces the computational costs on structured quadrangle cells in comparison with DG**

- 3 DG/SEM coupling
 - Hybrid meshes structures
 - Variational formulation

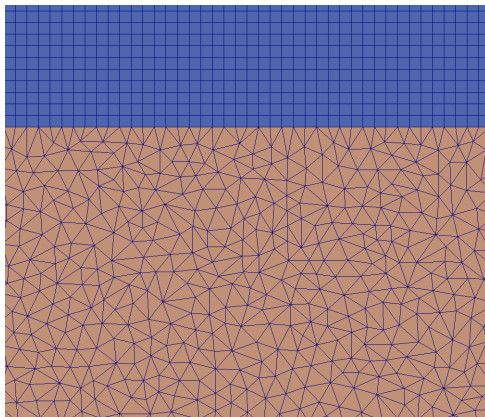
- Aim at coupling P_k and Q_k structures.
- Need to extend or split some structures (e.g. neighbour indexes).
- Define new face matrices:

$$M_{ij}^{K,L} = \int_{K \cap L} \phi_i^K \phi_j^L, \quad M_{ij}^{K,L} = \int_{K \cap L} \psi_i^K \psi_j^L, \quad M_{ij}^{K,L} = \int_{K \cap L} \phi_i^K \psi_j^L.$$



Global context

- Domain in two parts : $\Omega_{h,1}$ (**structured quadrangles + SEM**), $\Omega_{h,2}$ (**unstructured triangles + DG**).



SEM variational formulation :

$$\begin{cases} \int_{\Omega_{h,1}} \rho \partial_t v_1 \cdot w_1 = - \int_{\Omega_{h,1}} \sigma_1 \cdot \nabla w_1 + \int_{\Gamma_{out,1}} (\sigma_1 \mathbf{n}_1) \cdot w_1, \\ \int_{\Omega_{h,1}} \partial_t \sigma_1 : \xi_1 = - \int_{\Omega_{h,1}} (\nabla(C\xi_1)) \cdot v_1 + \int_{\Gamma_{out,1}} (C\xi_1 \mathbf{n}_1) \cdot v_1. \end{cases}$$

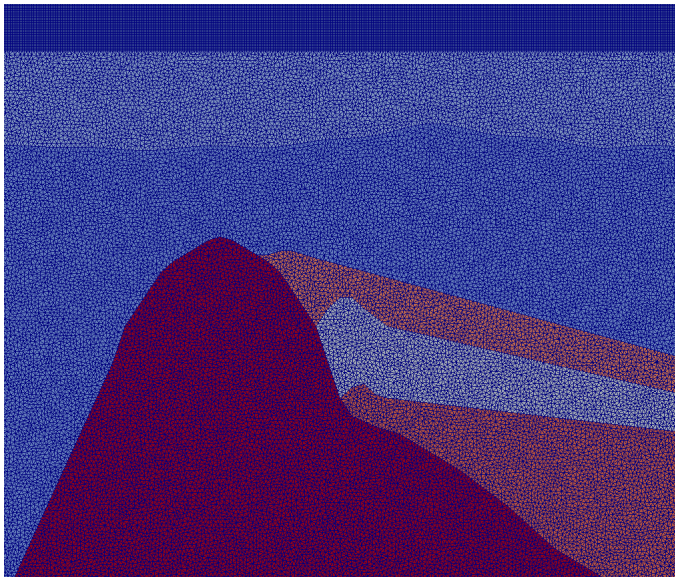
DG variational formulation :

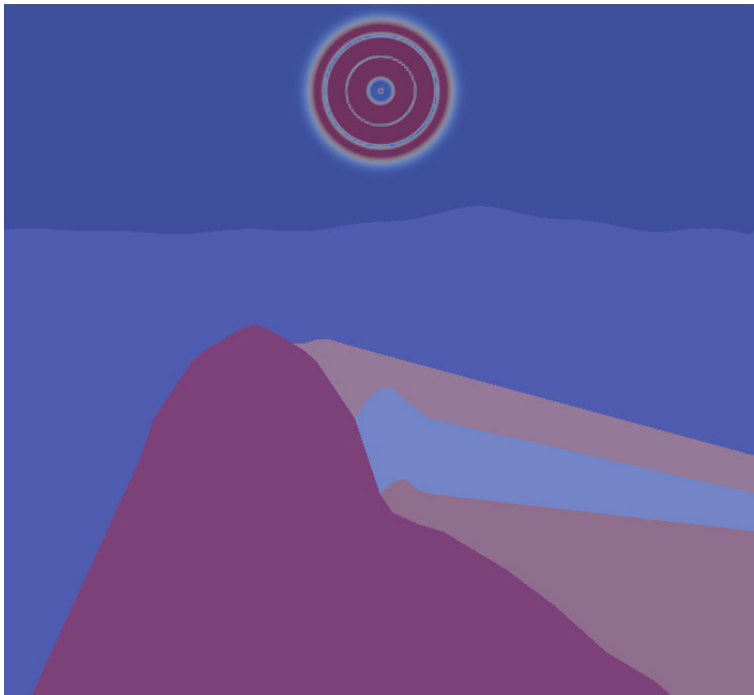
$$\begin{cases} \int_{\Omega_{h,2}} \rho \partial_t v_2 \cdot w_2 = - \int_{\Omega_{h,2}} \sigma_2 \cdot \nabla w_2 + \int_{\Gamma_{out,2}} (\sigma_2 \mathbf{n}_2) \cdot w_2 + \int_{\Gamma_{int}} \{ \{ \sigma_2 \} \} [[w_2]] \cdot \mathbf{n}_2, \\ \int_{\Omega_{h,2}} \partial_t \sigma_2 : \xi_2 = - \int_{\Omega_{h,2}} (\nabla(C\xi_2)) \cdot v_2 + \int_{\Gamma_{out,2}} (C\xi_2 \mathbf{n}_2) \cdot v_2 + \int_{\Gamma_{int}} \{ \{ v_2 \} \} [[C\xi_2]] \cdot \mathbf{n}_2. \end{cases}$$

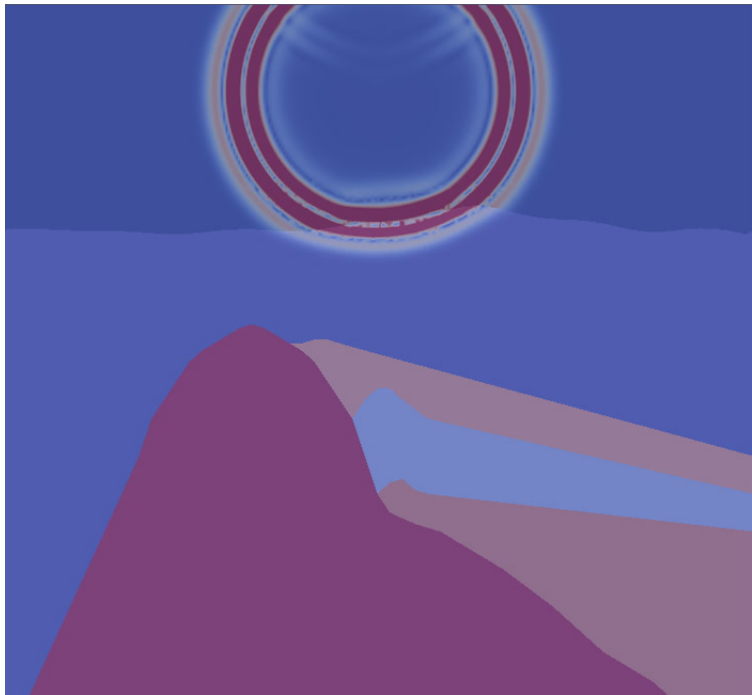
$$\left\{ \begin{aligned}
 & \int_{\Omega_{h,1}} \rho \partial_t v_1 \cdot w_1 + \int_{\Omega_{h,2}} \rho \partial_t v_2 \cdot w_2 = - \int_{\Omega_{h,1}} \sigma_1 \cdot \nabla w_1 - \int_{\Omega_{h,2}} \sigma_2 \cdot \nabla w_2 \\
 & + \int_{\Gamma_{out,1}} (\sigma_1 \mathbf{n}_1) \cdot w_1 + \int_{\Gamma_{out,2}} (\sigma_2 \mathbf{n}_2) \cdot w_2 + \int_{\Gamma_{int}} \{ \{ \sigma_2 \} \} [[w_2]] \cdot \mathbf{n}_2 \\
 & + \int_{\Gamma_{1/2}} [[\sigma w]] \cdot \mathbf{n}, \\
 & \int_{\Omega_{h,1}} \partial_t \sigma_1 : \xi_1 + \int_{\Omega_{h,2}} \partial_t \sigma_2 : \xi_2 = - \int_{\Omega_{h,1}} (\nabla(C\xi_1)) \cdot v_1 - \int_{\Omega_{h,2}} (\nabla(C\xi_2)) \cdot v_2 \\
 & + \int_{\Gamma_{out,1}} (C\xi_1 \mathbf{n}_1) \cdot v_1 + \int_{\Gamma_{out,2}} (C\xi_2 \mathbf{n}_2) \cdot v_2 + \int_{\Gamma_{int}} \{ \{ \sigma_2 \} \} [[w_2]] \cdot \mathbf{n}_2 \\
 & + \int_{\Gamma_{1/2}} [[(C\xi)v]] \cdot \mathbf{n}.
 \end{aligned} \right.$$

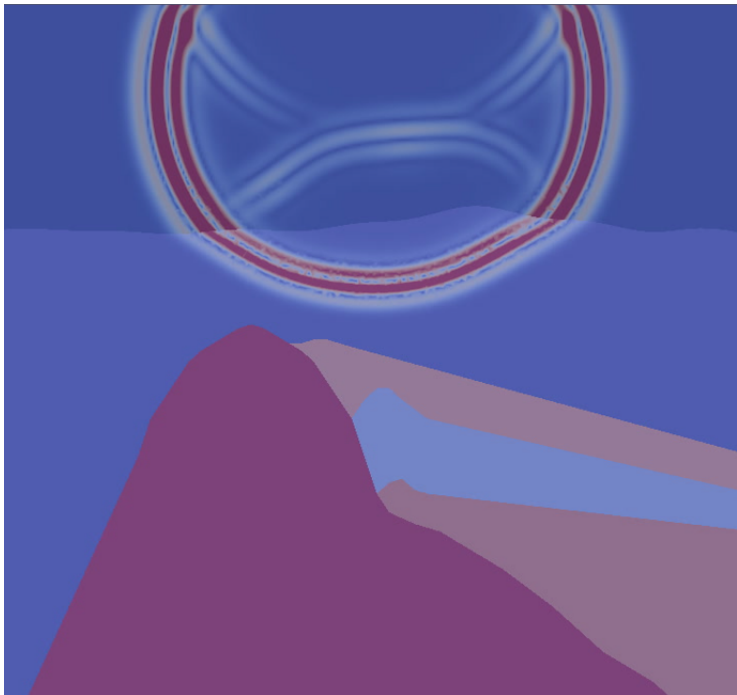
$$\left\{ \begin{aligned}
 & \int_{\Omega_{h,1}} \rho \partial_t v_1 \cdot w_1 + \int_{\Omega_{h,2}} \rho \partial_t v_2 \cdot w_2 = - \int_{\Omega_{h,1}} \sigma_1 \cdot \nabla w_1 - \int_{\Omega_{h,2}} \sigma_2 \cdot \nabla w_2 \\
 & + \int_{\Gamma_{out,1}} (\sigma_1 \mathbf{n}_1) \cdot w_1 + \int_{\Gamma_{out,2}} (\sigma_2 \mathbf{n}_2) \cdot w_2 + \int_{\Gamma_{int}} \{ \{ \sigma_2 \} \} [[w_2]] \cdot \mathbf{n}_2 \\
 & + \int_{\Gamma_{1/2}} \{ \{ \sigma \} \} [[w]] \cdot \mathbf{n} + [[\sigma]] \{ \{ w \} \} \cdot \mathbf{n}, \\
 & \int_{\Omega_{h,1}} \partial_t \sigma_1 : \xi_1 + \int_{\Omega_{h,2}} \partial_t \sigma_2 : \xi_2 = - \int_{\Omega_{h,1}} (\nabla(C\xi_1)) \cdot v_1 - \int_{\Omega_{h,2}} (\nabla(C\xi_2)) \cdot v_2 \\
 & + \int_{\Gamma_{out,1}} (C\xi_1 \mathbf{n}_1) \cdot v_1 + \int_{\Gamma_{out,2}} (C\xi_2 \mathbf{n}_2) \cdot v_2 + \int_{\Gamma_{int}} \{ \{ \sigma_2 \} \} [[w_2]] \cdot \mathbf{n}_2 \\
 & + \int_{\Gamma_{1/2}} [[C\xi]] \{ \{ v \} \} \cdot \mathbf{n} + \{ \{ C\xi \} \} [[v]] \cdot \mathbf{n}.
 \end{aligned} \right.$$

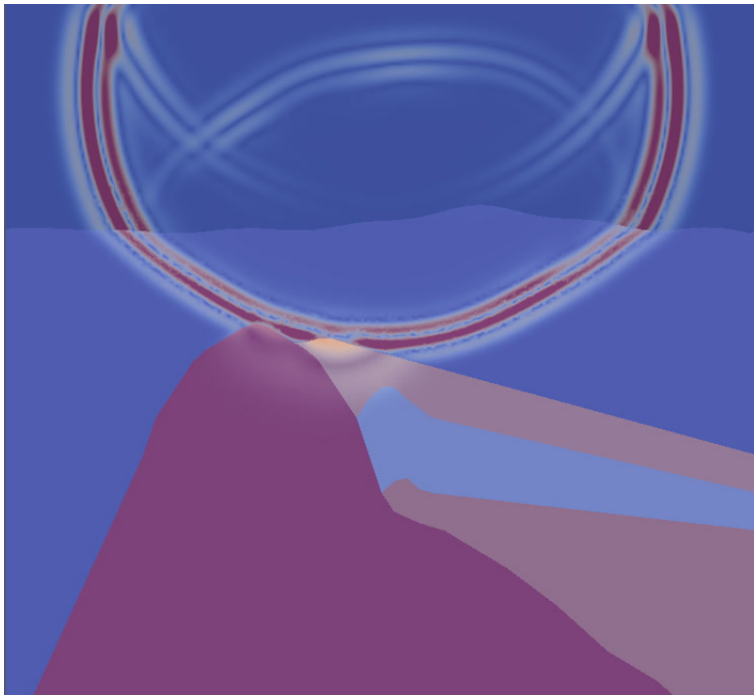
$$\left\{ \begin{aligned}
 & \int_{\Omega_{h,1}} \rho \partial_t v_1 \cdot w_1 + \int_{\Omega_{h,2}} \rho \partial_t v_2 \cdot w_2 = - \int_{\Omega_{h,1}} \sigma_1 \cdot \nabla w_1 - \int_{\Omega_{h,2}} \sigma_2 \cdot \nabla w_2 \\
 & + \int_{\Gamma_{out,1}} (\sigma_1 \mathbf{n}_1) \cdot w_1 + \int_{\Gamma_{out,2}} (\sigma_2 \mathbf{n}_2) \cdot w_2 + \int_{\Gamma_{int}} \{ \{ \sigma_2 \} \} [[w_2]] \cdot \mathbf{n}_2 \\
 & + \int_{\Gamma_{1/2}} \{ \{ \sigma \} \} [[w]] \cdot \mathbf{n} \quad \cancel{+ [[\sigma]] \{ \{ w \} \} \cdot \mathbf{n}}, \\
 & \int_{\Omega_{h,1}} \partial_t \sigma_1 : \xi_1 + \int_{\Omega_{h,2}} \partial_t \sigma_2 : \xi_2 = - \int_{\Omega_{h,1}} (\nabla(C\xi_1)) \cdot v_1 - \int_{\Omega_{h,2}} (\nabla(C\xi_2)) \cdot v_2 \\
 & + \int_{\Gamma_{out,1}} (C\xi_1 \mathbf{n}_1) \cdot v_1 + \int_{\Gamma_{out,2}} (C\xi_2 \mathbf{n}_2) \cdot v_2 + \int_{\Gamma_{int}} \{ \{ \sigma_2 \} \} [[w_2]] \cdot \mathbf{n}_2 \\
 & + \int_{\Gamma_{1/2}} [[C\xi]] \{ \{ v \} \} \cdot \mathbf{n} \quad \cancel{+ \{ \{ C\xi \} \} [[v]] \cdot \mathbf{n}}.
 \end{aligned} \right.$$

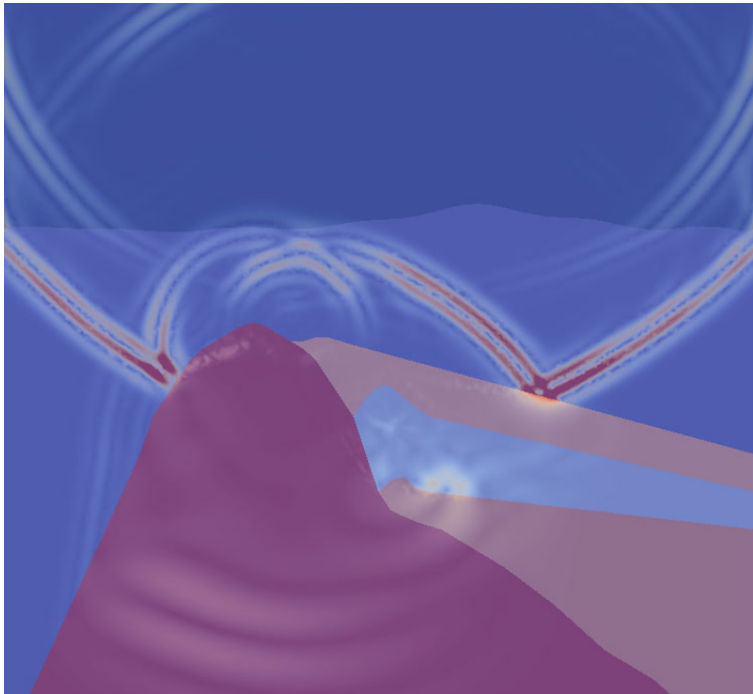












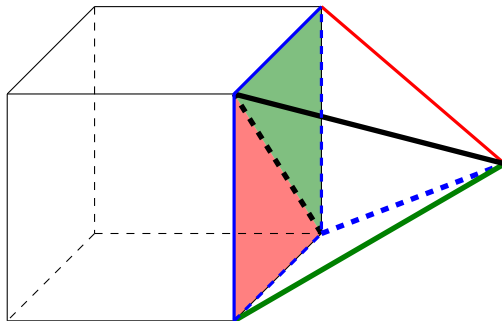


Figure: Hexa/Tet boundary configuration

- Only deal with a simple case of 3D hybrid meshes : one hexahedron has only two tetrahedra as neighbour.
- Extend SEM in 3D (basis functions...).
- Require introducing a new matrix which handles the rotation cases between two elements.

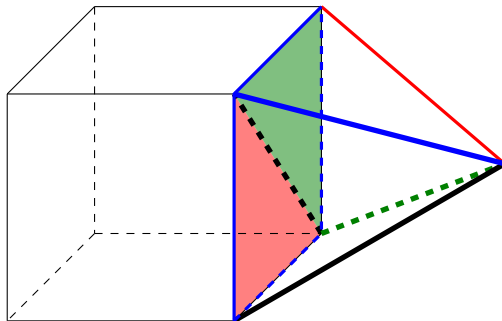


Figure: Hexa/Tet boundary configuration

- Only deal with a simple case of 3D hybrid meshes : one hexahedron has only two tetrahedra as neighbour.
- Extend SEM in 3D (basis functions...).
- Require introducing a new matrix which handles the rotation cases between two elements.

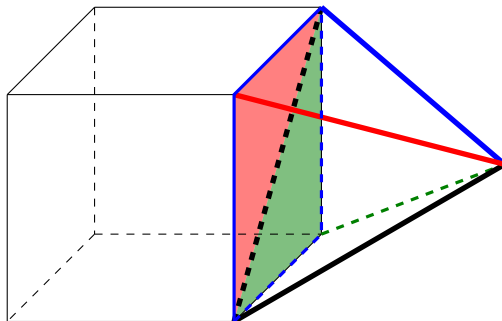


Figure: Hexa/Tet boundary configuration

- Only deal with a simple case of 3D hybrid meshes : one hexahedron has only two tetrahedra as neighbour.
- Extend SEM in 3D (basis functions...).
- Require introducing a new matrix which handles the rotation cases between two elements.

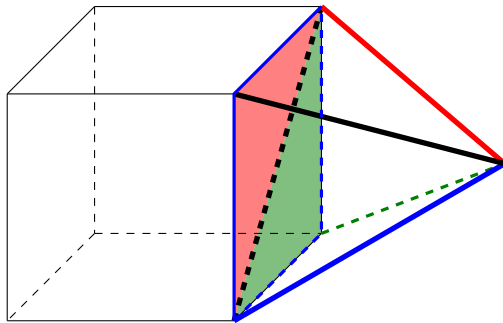
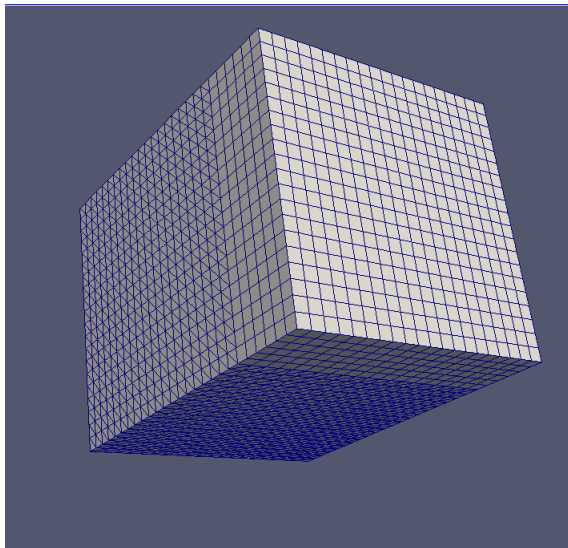
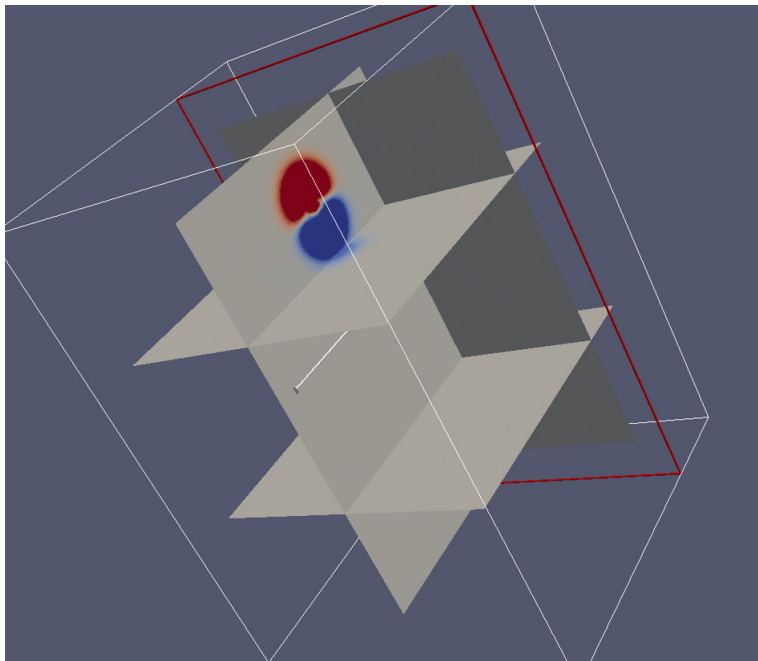
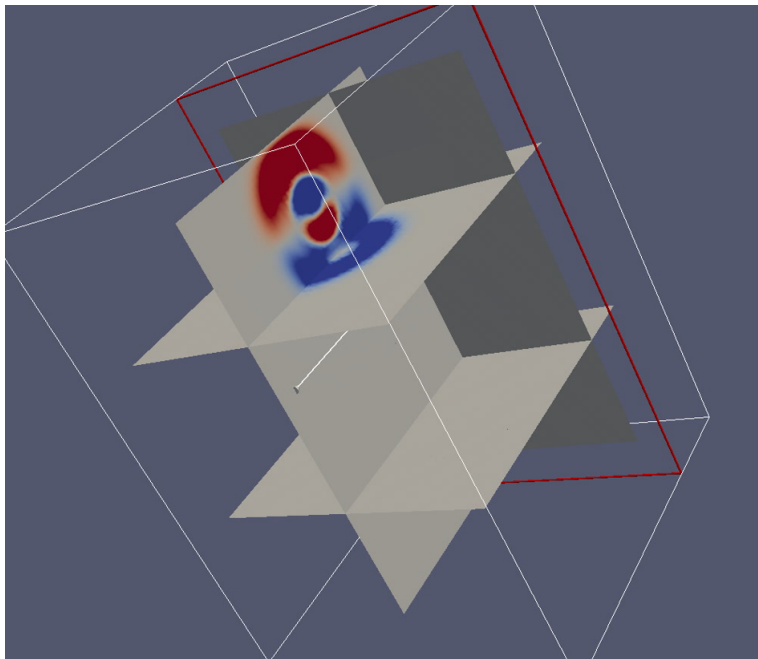


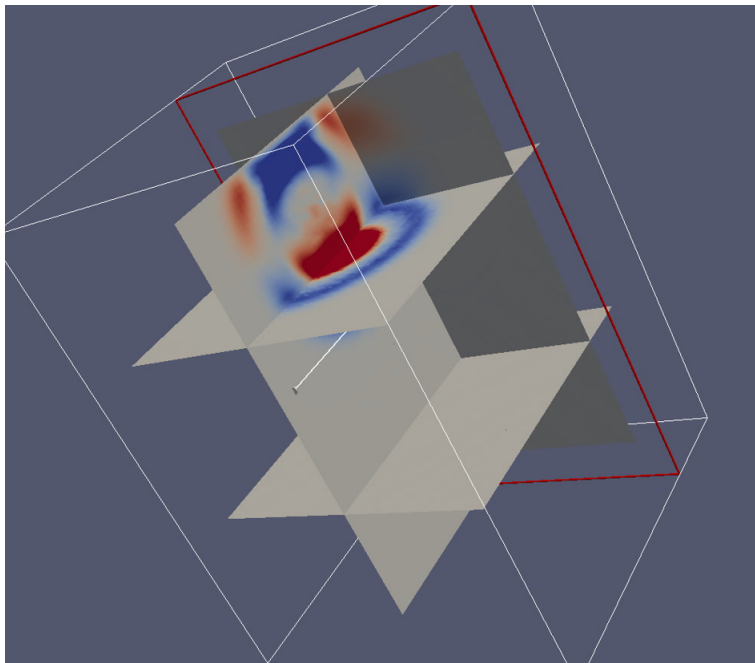
Figure: Hexa/Tet boundary configuration

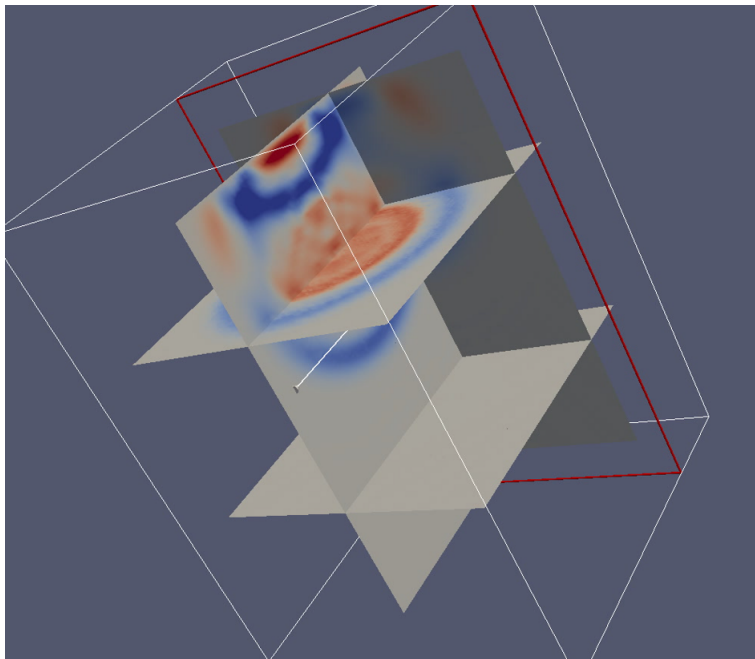
- Only deal with a simple case of 3D hybrid meshes : one hexahedron has only two tetrahedra as neighbour.
- Require introducing a new matrix which handles the rotation cases between two elements.

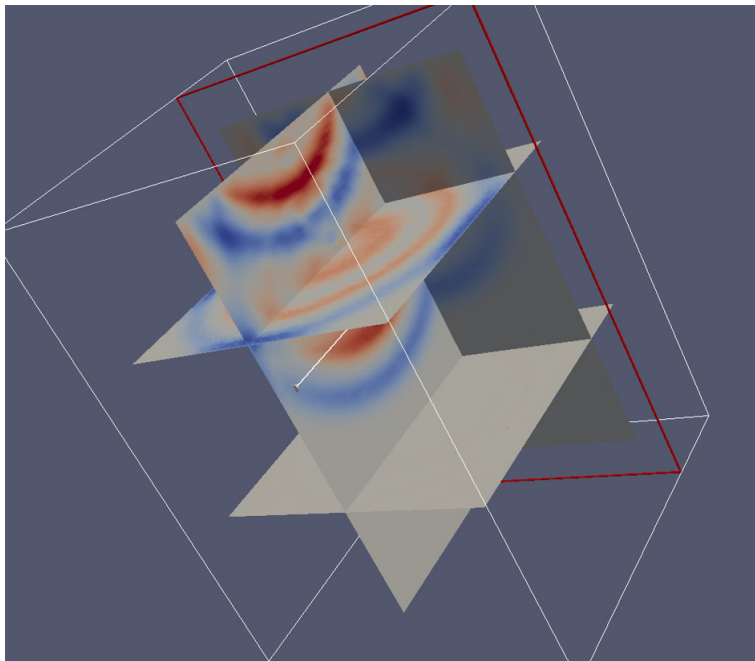


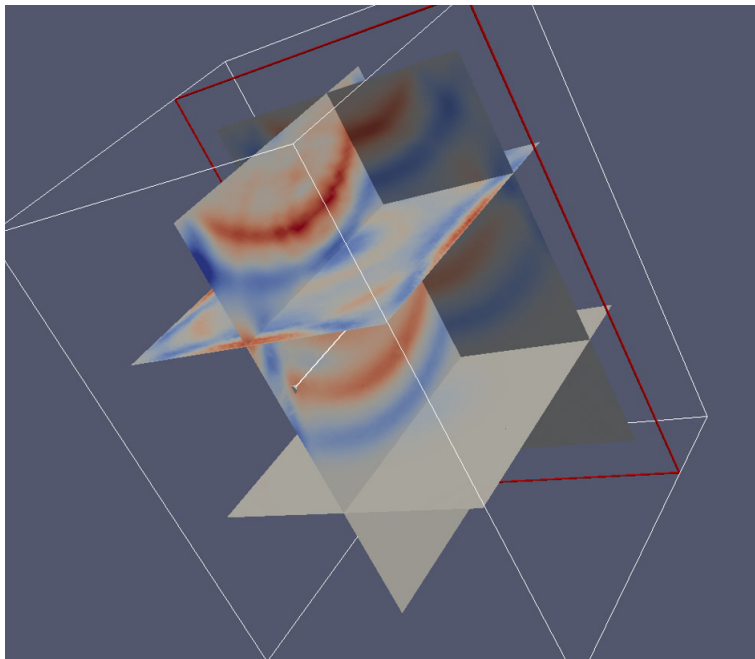


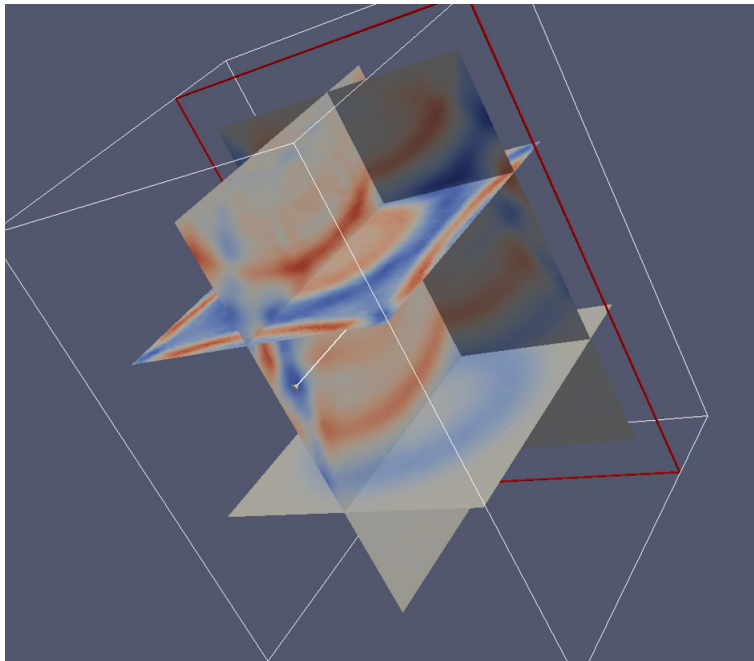












Conclusion

- 1 SEM is more efficient on structured quadrangle mesh than DG
- 2 Build a variational formulation for DG/SEM coupling and find a CFL condition that ensures stability
- 3 Show the utility of using hybrid meshes and method coupling (reduce computational cost,...)

Perspectives

- Implement DG/SEM coupling on the code (2D) ✓
- Develop DG/SEM coupling in 3D ✓
- Develop PML in the hexahedral part
- Add a local time-stepping scheme

Thank you for your attention !

Questions?

Error-order graphic

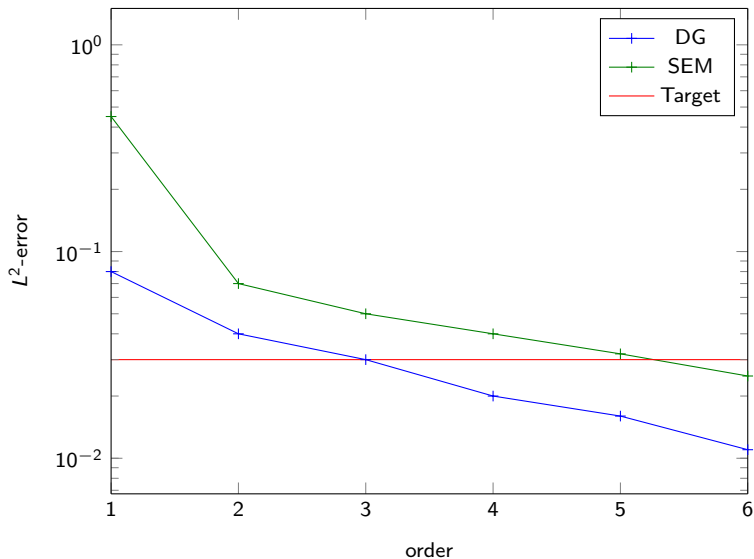


Figure: L^2 -error comparison on a 10000 cells mesh

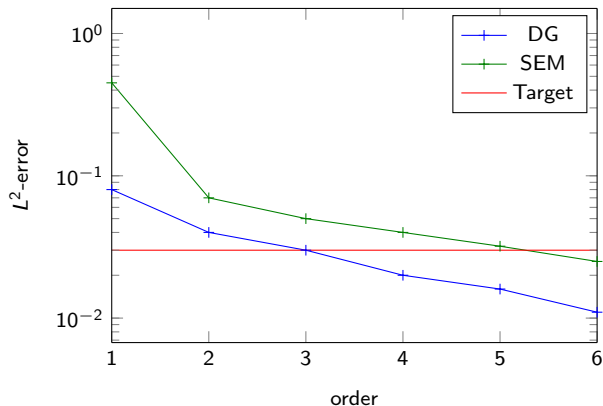


Figure: L^2 -error comparison on a 10000 cells mesh

	CFL	L2-error	CPU-time	Nb of time steps
DG	2e-3	3e-2	7.93	1000
SEM(order five)	2.13e-3	3e-2	9.06	943

Figure: SEM and DG comparison with fixed error