Numerical Integration and High Order Finite Element Method Applied to Maxwell's Equations

M. Durufle, G Cohen

INRIA, project POEMS

25th april 2007

Bibliography and motivation

- Y. Maday, E. Ronquist, Spectral Methods
- N. Tordjman, mass lumping for wave equation (triangles/quadrilaterals)
- Cohen, Monk, mass lumping for Maxwell's equations (hexahedra)
- S. Fauqueux, mixed spectral elements for wave and elastic equations (hexahedra)
- S. Pernet, Discontinuous Galerkin methods for Maxwell's equations (hexahedra)

Introduction

- Apply techniques of “mass lumping” and “mixed formulation”, which are efficient in temporal domain
  - Application of these techniques to Helmholtz and time-harmonic Maxwell equations
  - Gain in storage and time, by using these techniques in frequential domain
- Choose an efficient preconditioning technique to solve linear systems issued from these equations
- Apply the developped algorithms to evaluate accurately radar cross sections of electromagnetic targets

Introduction

- Apply techniques of “mass lumping” and “mixed formulation”, which are efficient in temporal domain
  - Application of these techniques to Helmholtz and time-harmonic Maxwell equations
  - Gain in storage and time, by using these techniques in frequential domain

- Choose an efficient preconditioning technique to solve linear systems issued from these equations

- Apply the developped algorithms to evaluate accurately radar cross sections of electromagnetic targets

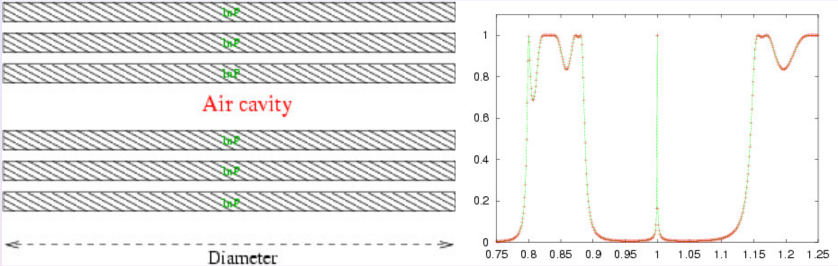
Introduction

- Apply techniques of “mass lumping” and “mixed formulation”, which are efficient in temporal domain
    - Application of these techniques to Helmholtz and time-harmonic Maxwell equations
    - Gain in storage and time, by using these techniques in frequential domain
- Choose an efficient preconditioning technique to solve linear systems issued from these equations
- Apply the developped algorithms to evaluate accurately radar cross sections of electromagnetic targets

Outline

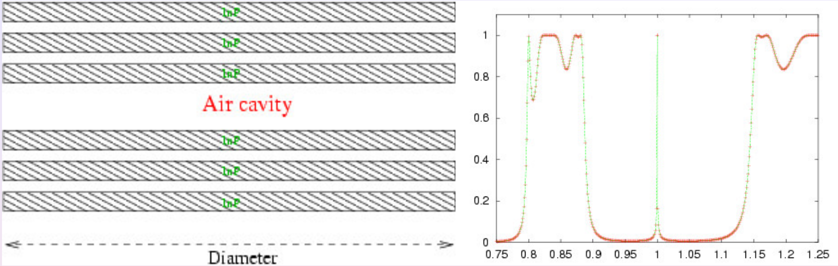
1. Resolution of Helmholtz equation
   - Interest to use high order methods
   - Efficient matrix-vector product on hexahedral meshes
   - Efficient iterative solver and preconditioning
2. Time-harmonic Maxwell equations
   - Spurious modes for Nedelec's second family
   - Spurious modes for Discontinuous Galerkin method
   - Efficient matrix-vector product for Nedelec's first family
   - Efficient iterative resolution
3. Time-domain Maxwell equations
   - Description of DG method
   - Numerical Results

A test case : an optical filter



At right, transmission coefficient according to the frequency

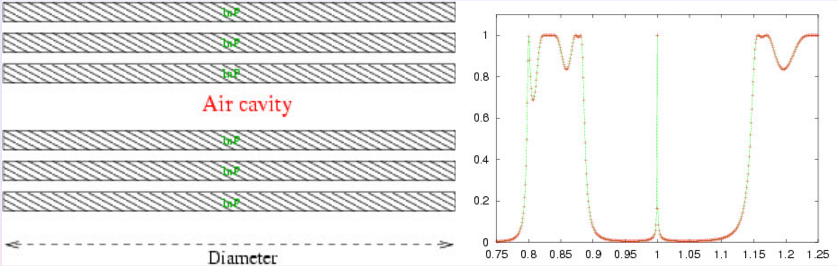
- Frequency F = 1.0 is a resonant frequency of the device
- Enlightment of the device by a gaussian beam.
- PML around the computational domain.



At right, transmission coefficient according to the frequency

- Frequency F = 1.0 is a resonant frequency of the device
- Enlightment of the device by a gaussian beam.
- PML around the computational domain.

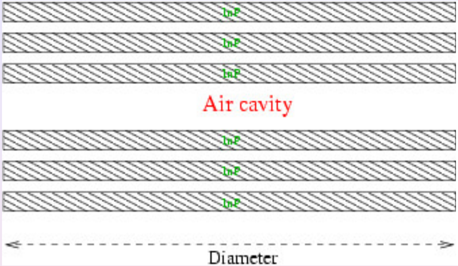
A test case : an optical filter

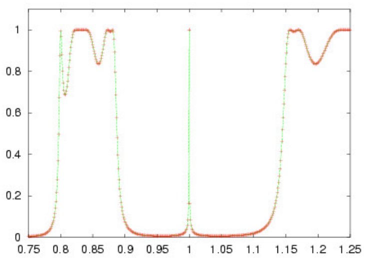


At right, transmission coefficient according to the frequency

- Frequency F = 1.0 is a resonant frequency of the device
- Enlightment of the device by a gaussian beam.
- PML around the computational domain.

A test case : an optical filter

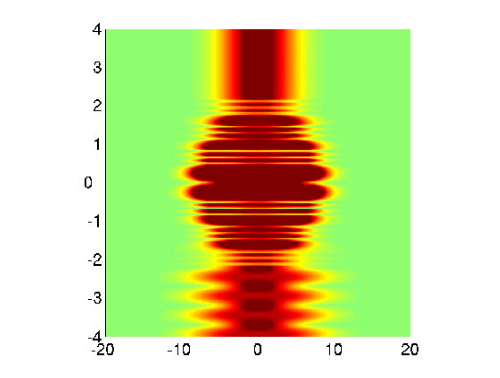




At right, transmission coefficient according to the frequency

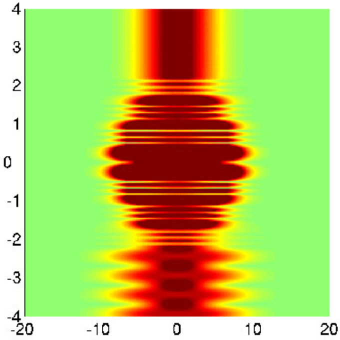
- Frequency F = 1.0 is a resonant frequency of the device
- Enlightment of the device by a gaussian beam.
- PML around the computational domain.

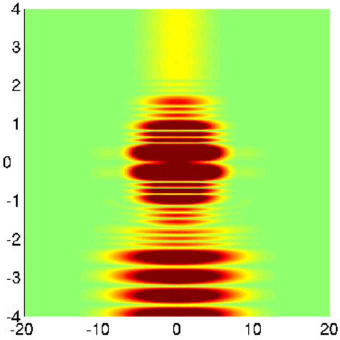
Advantage to use high order method



Numerical solution for $\mathbf{Q_5}$ with 10 points by wavelength

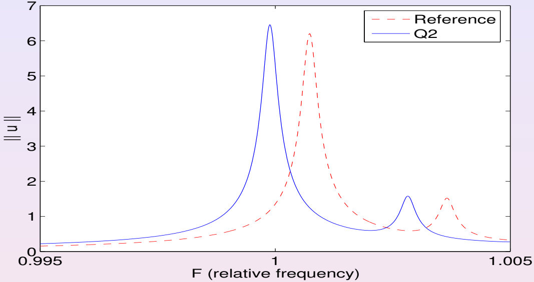
Advantage to use high order method





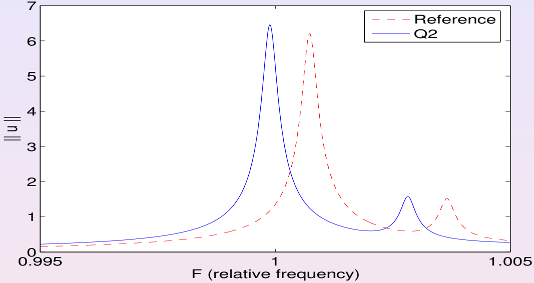
At right, numerical solution for $\mathbf{Q_2}$ with 10 points by wavelength

Advantage to use high order method



Norm of the solution at the ouput, according to the frequency

Advantage to use high order method



Norm of the solution at the ouput, according to the frequency
Which order is optimal to reach an error less than 10% ?

Order	2	3	4	5	6	7
Nb dofs	453 000	69 800	52 000	33 200	47 700	42 200

Helmholtz equation

$$-\rho\,\omega^2\,u \;-\; \mathrm{div}(\mu\,\nabla u) \;=\; f \quad \in \Omega$$

Use of finite element method leads to the following linear system :

$$(-\omega^2 D_h \;+\; K_h)\;U_h \;=\; F_h$$

Mass matrix $D_h \;=\; \displaystyle\int_\Omega \rho\,\varphi_i^{GL}\,\varphi_j^{GL}\,dx$

Stiffness matrix $K_h \;=\; \displaystyle\int_\Omega \mu\,\nabla\varphi_i^{GL}\cdot\nabla\varphi_j^{GL}\,dx$

Our aim is to develop an efficient iterative solver for an high order of approximation r. We need then a fast matrix-vector product $(-\omega^2 D_h \;+\; K_h)\;U_h$

Helmholtz equation

$$-\rho\,\omega^2\,u \;-\; \mathrm{div}(\mu\,\nabla u) \;=\; f \quad \in \Omega$$

Use of finite element method leads to the following linear system :

$$(-\omega^2 D_h \,+\, K_h)\,U_h \;=\; F_h$$

Mass matrix $D_h \;=\; \int_\Omega \rho\,\varphi_i^{GL}\,\varphi_j^{GL}\,dx$

Stiffness matrix $K_h \;=\; \int_\Omega \mu\,\nabla\varphi_i^{GL}\cdot\nabla\varphi_j^{GL}\,dx$

Our aim is to develop an efficient iterative solver for an high order of approximation r. We need then a fast matrix-vector product $(-\omega^2 D_h \,+\, K_h)\,U_h$

Helmholtz equation

$$-\rho\,\omega^2\,u - \mathrm{div}(\mu\,\nabla u) = f \quad \in \Omega$$

Use of finite element method leads to the following linear system :

$$(-\omega^2 D_h + K_h)\,U_h = F_h$$

Mass matrix $D_h = \displaystyle\int_\Omega \rho\,\varphi_i^{GL}\,\varphi_j^{GL}\,dx$

Stiffness matrix $K_h = \displaystyle\int_\Omega \mu\,\nabla\varphi_i^{GL}\cdot\nabla\varphi_j^{GL}\,dx$

Our aim is to develop an efficient iterative solver for an high order of approximation r. We need then a fast matrix-vector product $(-\omega^2 D_h + K_h)\,U_h$

Helmholtz equation

$$-\rho\,\omega^2\,u \;-\; \mathrm{div}(\mu\,\nabla u) \;=\; f \quad \in \Omega$$

Use of finite element method leads to the following linear system :

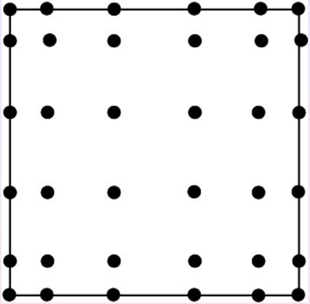
$$(-\omega^2 D_h \;+\; K_h)\,U_h \;=\; F_h$$

Mass matrix $D_h \;=\; \displaystyle\int_\Omega \rho\,\varphi_i^{GL}\,\varphi_j^{GL}\,dx$

Stiffness matrix $K_h \;=\; \displaystyle\int_\Omega \mu\,\nabla\varphi_i^{GL}\cdot\nabla\varphi_j^{GL}\,dx$

Our aim is to develop an efficient iterative solver for an high order of approximation r. We need then a fast matrix-vector product $(-\omega^2 D_h \;+\; K_h)\,U_h$

Use of Gauss-Lobatto points



Gauss-Lobatto points for Q_5

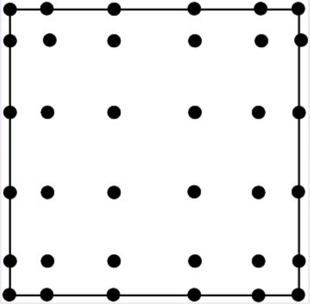
on the unit square $\hat{K}$

Use of these points both for interpolation and numerical quadrature leads to a diagonal mass matrix D_h and a fast matrix-vector product for $K_h U_h$

See the thesis of S. Fauqueux, 2003

These points permit a fast matrix-vector product

Use of Gauss-Lobatto points



Gauss-Lobatto points for Q_5

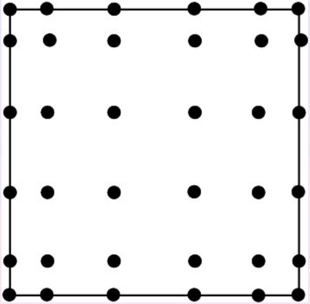
on the unit square $\hat{K}$

Use of these points both for interpolation and numerical quadrature leads to a diagonal mass matrix D_h and a fast matrix-vector product for $K_h\ U_h$

See the thesis of S. Fauqueux, 2003

These points permit a fast matrix-vector product

Use of Gauss-Lobatto points



Gauss-Lobatto points for Q_5

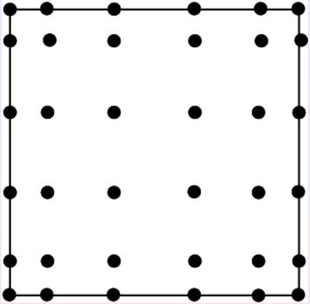
on the unit square $\hat{K}$

Use of these points both for interpolation and numerical quadrature leads to a diagonal mass matrix D_h and a fast matrix-vector product for $K_h\, U_h$

See the thesis of S. Fauqueux, 2003

These points permit a fast matrix-vector product

Use of Gauss-Lobatto points



Gauss-Lobatto points for Q_5

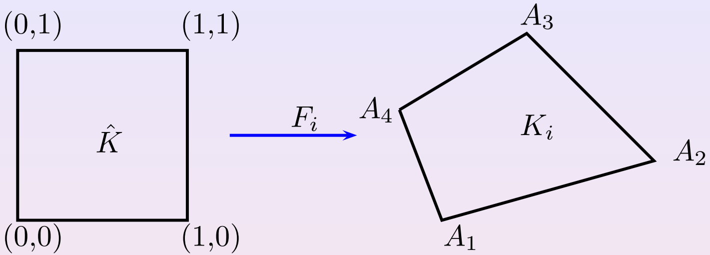
on the unit square $\hat{K}$

Use of these points both for interpolation and numerical quadrature leads to a diagonal mass matrix D_h and a fast matrix-vector product for $K_h\, U_h$

See the thesis of S. Fauqueux, 2003

These points permit a fast matrix-vector product

Elementary matrices



The transformation F_i

Elementary matrices

$$(D_h)_{i,j} = \int_{\hat{K}} \rho J_i \, \hat{\varphi}_i^{GL} \, \hat{\varphi}_j^{GL} \, d\hat{x}$$

$$(K_h)_{i,j} = \int_{\hat{K}} \mu \, J_i \, DF_i^{-1} \, DF_i^{*-1} \, \hat{\nabla} \, \hat{\varphi}_i^{GL} \cdot \hat{\nabla} \hat{\varphi}_j^{GL} \, d\hat{x}$$

Elementary matrices

$$(D_h)_{i,j} \;=\; \int_{\hat{K}} \rho J_i \, \hat{\varphi}_i^{GL} \, \hat{\varphi}_j^{GL} \, d\hat{x}$$

$$(K_h)_{i,j} \;=\; \int_{\hat{K}} \mu \, J_i \, DF_i^{-1} \, DF_i^{*-1} \, \hat{\nabla} \, \hat{\varphi}_i^{GL} \cdot \hat{\nabla} \hat{\varphi}_j^{GL} \, d\hat{x}$$

- Use of quadrature formulas (ω_k^X, ξ_k^X) on the unit square
  - X can be equal to GL (Gauss-Lobatto quadrature)
  - X can be equal to G (Gauss quadrature)

Elementary matrices

$$(D_h)_{i,j} = \int_{\hat{K}} \rho J_i \, \hat{\varphi}_i^{GL} \, \hat{\varphi}_j^{GL} \, d\hat{x}$$

$$(K_h)_{i,j} = \int_{\hat{K}} \mu \, J_i \, DF_i^{-1} \, DF_i^{*-1} \, \hat{\nabla} \, \hat{\varphi}_i^{GL} \cdot \hat{\nabla} \hat{\varphi}_j^{GL} \, d\hat{x}$$

- Use of quadrature formulas (ω_k^X, ξ_k^X) on the unit square
- Diagonal matrix

$$(A_h)_{k,k} = \rho \, J_i(\xi_k^X) \, \omega_k^X$$

- Bloc-diagonal matrix

$$(B_h)_{k,k} = \mu \, J_i \, DF_i^{-1} \, DF_i^{*-1} (\xi_k^X) \, \omega_k^X$$

Fast matrix vector product with any points

Let us introduce the two following matrices, independant of the geometry :

$$\hat{C}_{i,j} = \hat{\varphi}_i^{GL}(\xi_j^X) \qquad \hat{R}_{i,j} = \hat{\nabla}\hat{\varphi}_i^X(\xi_j^X)$$

Thus, we have : $D_h = \hat{C}\, A_h \hat{C}^* \qquad K_h = \hat{C}\hat{R}\, B_h \hat{R}^* \hat{C}^*$

Fast matrix vector product with any points

Let us introduce the two following matrices, independant of the geometry :

$$\hat{C}_{i,j} = \hat{\varphi}_i^{GL}(\xi_j^X) \qquad \hat{R}_{i,j} = \hat{\nabla}\hat{\varphi}_i^X(\xi_j^X)$$

Thus, we have : $D_h = \hat{C}\, A_h \hat{C}^* \qquad K_h = \hat{C}\hat{R}\, B_h \hat{R}^* \hat{C}^*$

Fast matrix vector product with any points

Let us introduce the two following matrices, independant of the geometry :

$$\hat{C}_{i,j} = \hat{\varphi}_i^{GL}(\xi_j^X) \qquad \hat{R}_{i,j} = \hat{\nabla}\hat{\varphi}_i^X(\xi_j^X)$$

Thus, we have : $D_h = \hat{C}\, A_h \hat{C}^* \qquad K_h = \hat{C}\hat{R}\, B_h \hat{R}^* \hat{C}^*$

r is the order of approximation

If $\hat{C}$ and $\hat{R}$ are stored as full matrices

- Complexity of $\hat{C}\, U$: $2\,(r+1)^6$ operations in 3-D
- Complexity of $\hat{R}\, U$: $6\,(r+1)^6$ operations in 3-D

Complexity of standard matrix vector product : $2\,(r+1)^6$ operations in 3-D

Fast matrix vector product with any points

Let us introduce the two following matrices, independant of the geometry :

$$\hat{C}_{i,j} = \hat{\varphi}_i^{GL}(\xi_j^X) \qquad \hat{R}_{i,j} = \hat{\nabla}\hat{\varphi}_i^X(\xi_j^X)$$

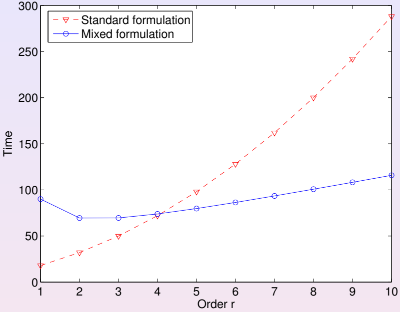
Thus, we have : $\quad D_h = \hat{C}\, A_h \hat{C}^* \qquad K_h = \hat{C}\hat{R}\, B_h \hat{R}^* \hat{C}^*$

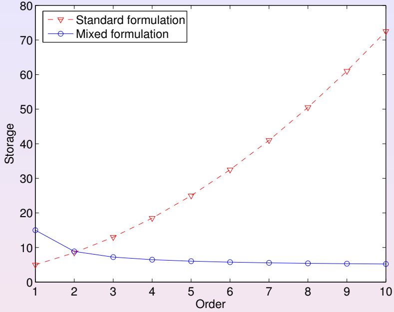
For hexahedral elements (tensorization), we have

- Complexity of $\hat{C}\, U$: $6\,(r+1)^4$ operations in 3-D
- Complexity of $\hat{R}\, U$: $6\,(r+1)^4$ operations in 3-D
- Complexity of $A_h\, U$ and $B_h\, V$: $16\,(r+1)^3$ operations in 3-D

- If we use Gauss-Lobatto points to integrate : $\hat{C} = I$
  In this case : "equivalence theorem" of S. Fauqueux
- Same storage for Gauss or GL points (A_h and B_h)
- MV product two times slower with Gauss integration

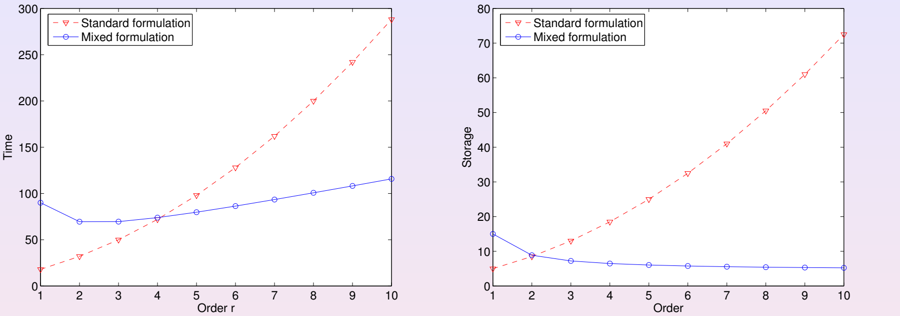
Matrix vector-product faster than standard methods ?





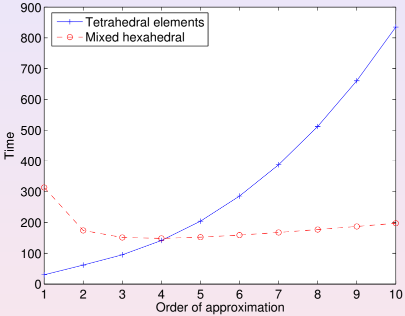
3-D comparison between the classical matrix-vector algorithm and the fast algorithm (mixed formulation), in 3-D.
At left, time according to the order of approximation, at right storage.

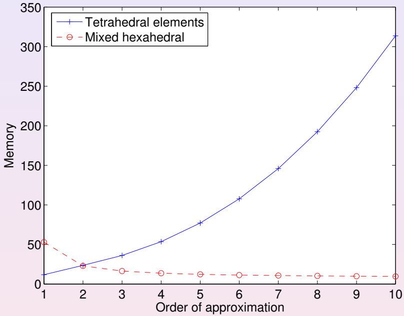
Matrix vector-product faster than standard methods ?



3-D comparison between the classical matrix-vector algorithm and the fast algorithm (mixed formulation), in 3-D.
At left, time according to the order of approximation, at right storage.
Gain in time for $r \geq 4$, gain in storage for $r \geq 2$.

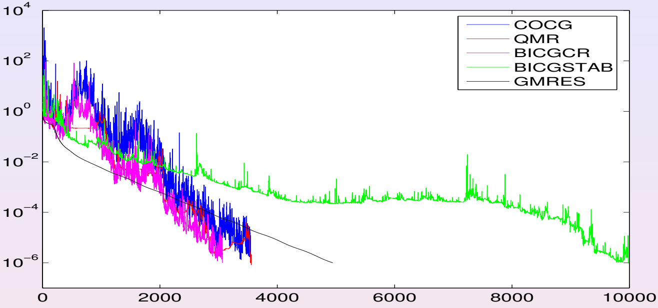
Matrix vector-product faster than standard methods ?





Comparison between hexahedral and tetrahedral elements, for time computation (at left) and storage (at right)

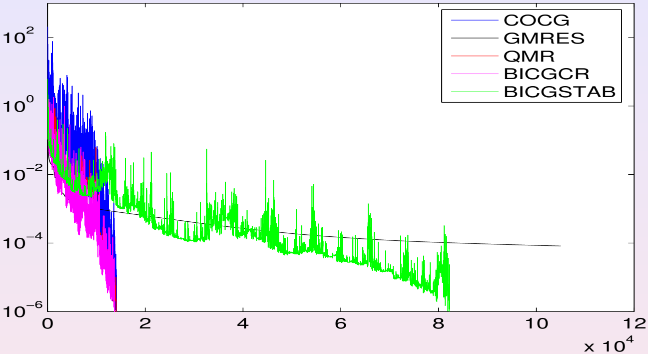
Iterative methods used



Evolution of the residual norm for the scattering of a perfectly conductor disc (Dirichlet condition).

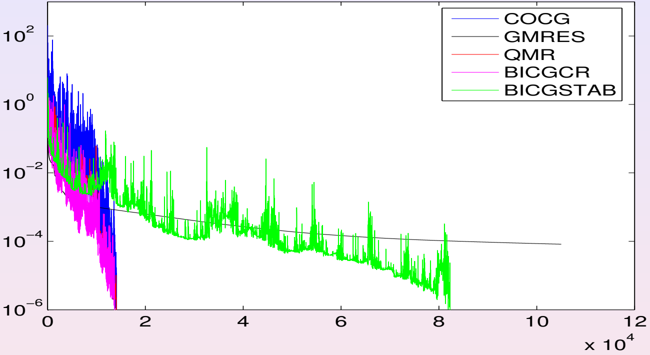
- GMRES, BICGSTAB and QMR for complex unsymmetric matrices
- COCG, BICGCR for complex symmetric matrices

Iterative methods used



Evolution of the residual norm for the scattering of a dielectric disc ($\rho = 4$).

Iterative methods used



- We choose to use BICGCR for all future experiments
- Need of preconditioning techniques to have less iterations

Preconditioning used

- Incomplete factorization with threshold on the damped Helmholtz equation :

$$-k^2(\alpha + i\beta)u - \Delta u = 0$$

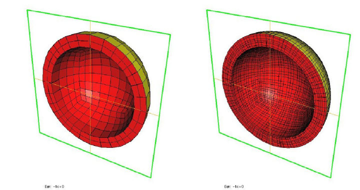
  - see Y. Saad, Iterative methods for sparse linear systems

Preconditioning used

- Incomplete factorization with threshold on the damped Helmholtz equation :

$$-k^2(\alpha + i\beta)u - \Delta u = 0$$

  - see Y. Saad, Iterative methods for sparse linear systems
  - We use a Q_1 subdivided mesh to compute matrix



At left, initial mesh Q_3, at right, subdivided mesh Q_1

Preconditioning used

- Incomplete factorization with threshold on the damped Helmholtz equation :

$$-k^2(\alpha + i\beta)u - \Delta u = 0$$

  - see Y. Saad, Iterative methods for sparse linear systems
- Multigrid method on the damped Helmholtz equation
  - see Y. A. Erlangga and al, Report of Delft University Technology, 2004

Preconditioning used

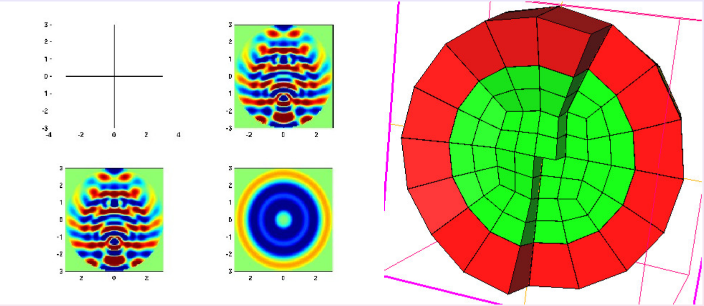
- Incomplete factorization with threshold on the damped Helmholtz equation :

$$-k^2(\alpha + i\beta)u - \Delta u = 0$$

  - see Y. Saad, Iterative methods for sparse linear systems
- Multigrid method on the damped Helmholtz equation
  - see Y. A. Erlangga and al, Report of Delft University Technology, 2004

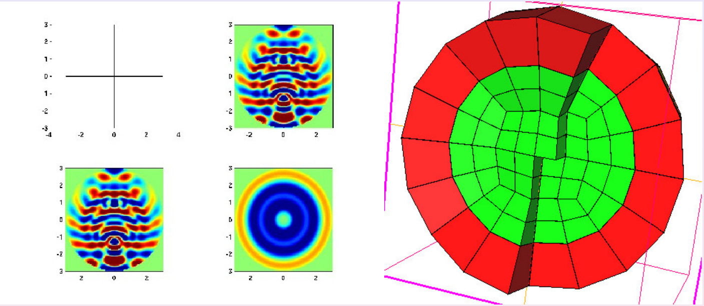
- Without damping, both preconditioners **does not lead** to convergence.
- A good choice of parameter is $\alpha = 1, \ \beta = 0.5$

Scattering by a dielectric sphere



- Dielectric sphere of radius 2 and with $\rho = 4 \quad \omega = 2\pi$
- First order absorbing boundary condition on a sphere of radius 3

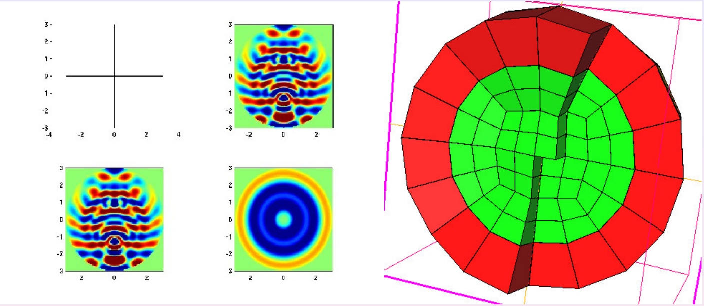
Scattering by a dielectric sphere



Number of dofs to reach less than 5 % L^2 error

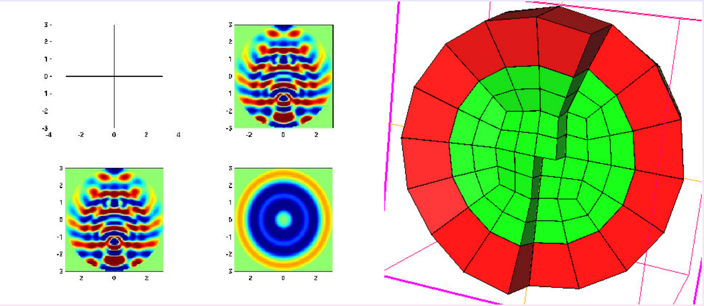
Finite element	structured $\mathbf{Q}_2$	struct $\mathbf{Q}_4$	struct $\mathbf{Q}_6$	n.s. $\mathbf{Q}_4$	n.s. $\mathbf{P}_4$
Number of dofs	220 000	85 000	78 000	243 000	180 000

Scattering by a dielectric sphere



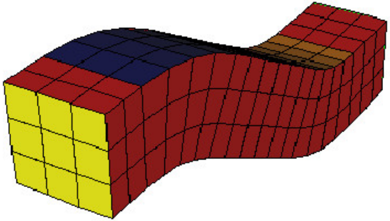
Finite element	structured $\mathbf{Q}_4$	non-structured $\mathbf{Q}_4$	non-structured $\mathbf{P}_4$
No preconditioning	708 s	5 795 s	1 597 s
ILUT(0.01)	91 s	534 s	363 s
Multigrid	185 s	729 s	695 s

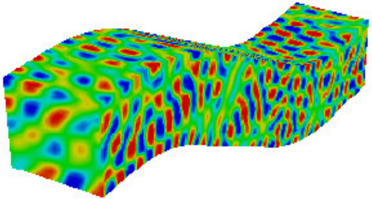
Scattering by a dielectric sphere



Finite element	structured $\mathbf{Q}_4$	non-structured $\mathbf{Q}_4$	non-structured $\mathbf{P}_4$
No preconditioning	34 Mo	99 Mo	136 Mo
ILUT(0.01)	137 Mo	420 Mo	507 Mo
Multigrid	50 Mo	143 Mo	327 Mo

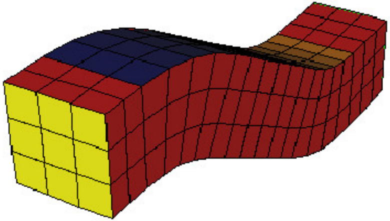
Scattering by a cobra cavity

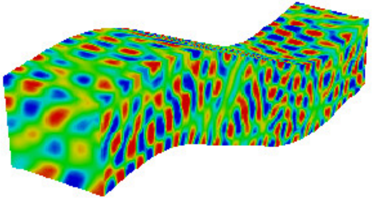




- Cobra cavity of length 20, and depth 4
- First order absorbing boundary condition on the yellow face

Scattering by a cobra cavity

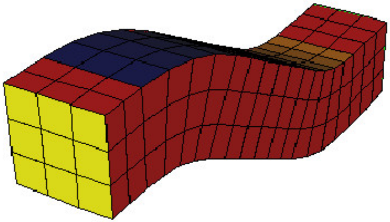


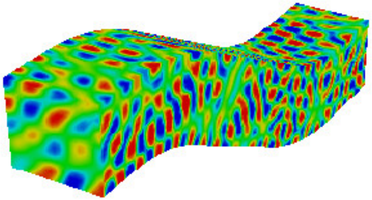


Number of dofs to reach less than 5 % L^2 error

Order	struct $\mathbf{Q}_4$	struct $\mathbf{Q}_6$	struct $\mathbf{Q}_8$	n.s. $\mathbf{Q}_4$	n.s. $\mathbf{Q}_6$	n.s. $\mathbf{P}_4$
Nb dofs	330 000	185 000	95 600	567,000	466 000	360 000

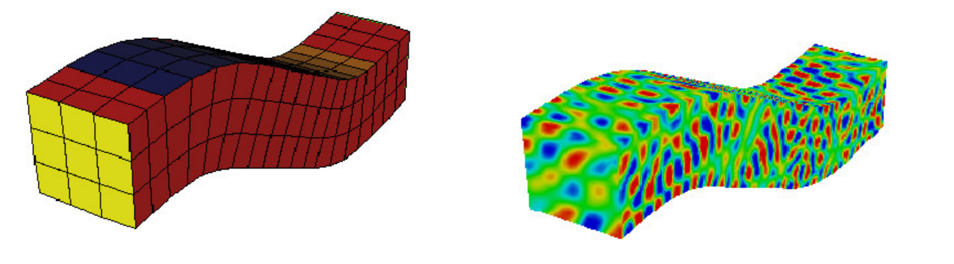
Scattering by a cobra cavity





Finite element	structured $\mathbf{Q}_8$	non-structured $\mathbf{Q}_6$	non-structured $\mathbf{P}_4$
No preconditioning	9 860 s	NC	NC
ILUT(0.01)	1 021 s	13 766 s	8 036 s
Two-grid	1 082 s	6 821 s	14 016 s

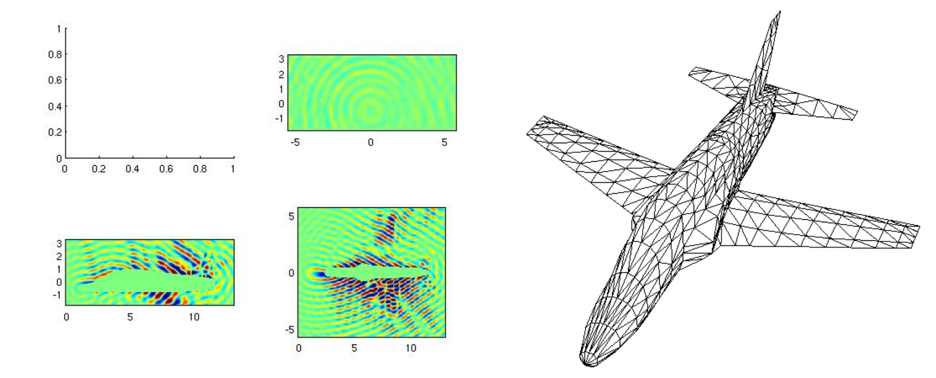
Scattering by a cobra cavity



Finite element	structured $\mathbf{Q}_8$	non-structured $\mathbf{Q}_6$	non-structured $\mathbf{P}_4$
No preconditioning	9 860 s	NC	NC
ILUT(0.01)	1 021 s	13 766 s	8 036 s
Two-grid	1 082 s	6 821 s	14 016 s

Finite element	structured $\mathbf{Q}_8$	non-structured $\mathbf{Q}_6$	non-structured $\mathbf{P}_4$
No preconditioning	32 Mo	162 Mo	251 Mo
ILUT(0.01)	150 Mo	1 250 Mo	1 400 Mo
Two-grid	60 Mo	283 Mo	710 Mo

Scattering by a plane



- Real part of the diffracted for an oblique incident plane wave
- Q4, 7.2 million of dofs
- 650 iterations and 7 hours with multigrid preconditioning

Outline

1. Resolution of Helmholtz equation
   - Interest to use high order methods
   - Efficient matrix-vector product on hexahedral meshes
   - Efficient iterative solver and preconditioning
2. Time-harmonic Maxwell equations
   - Spurious modes for Nedelec's second family
   - Spurious modes for Discontinuous Galerkin method
   - Efficient matrix-vector product for Nedelec's first family
   - Efficient iterative resolution
3. Time-domain Maxwell equations
   - Description of DG method
   - Numerical Results

Nedelec's second family on hexahedrals

Time-harmonic Maxwell's equations :

$$-\omega^2 \, \varepsilon \, \vec{E}(x) \; + \; \mathrm{curl}\Big(\frac{1}{\mu(x)} \mathrm{curl}(\vec{E}(x))\Big) \; = \; 0$$

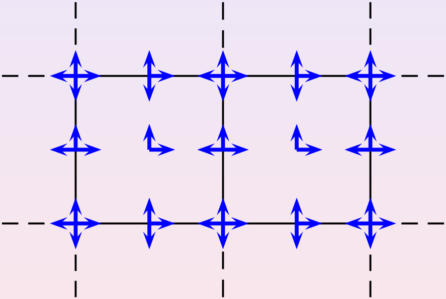
Space of approximation

$$V_h \; = \; \{ \, \vec{u} \in \; \mathrm{H(curl,}\Omega) \text{ such as } DF_i^* \, \vec{u} \circ F_i \; \in (Q_r)^3 \, \}$$

Nedelec's second family on hexahedrals

Time-harmonic Maxwell's equations :

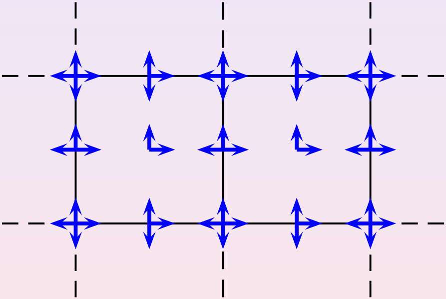
$$-\omega^2 \varepsilon \vec{E}(x) + \text{curl}\left(\frac{1}{\mu(x)} \text{curl}(\vec{E}(x))\right) = 0$$



Nedelec's second family on hexahedrals

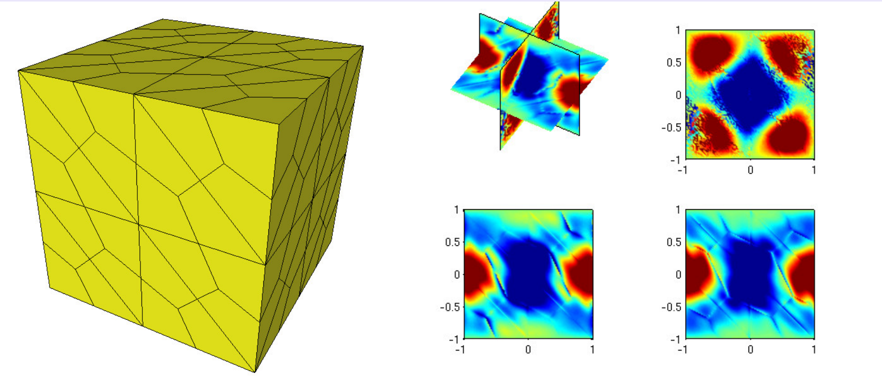
Time-harmonic Maxwell's equations :

$$-\omega^2 \, \varepsilon \, \vec{E}(x) \; + \; \mathrm{curl}\left(\frac{1}{\mu(x)} \mathrm{curl}(\vec{E}(x)) \right) \; = \; 0$$



- Mass lumping and factorization of stiffness matrix
- Low-storage and fast matrix-vector product

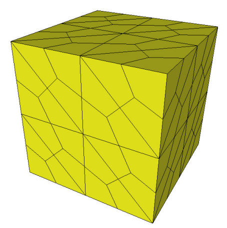
The unwanted oscillations



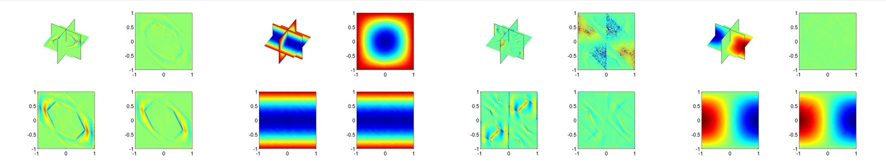
Dipole source on a cubic cavity. Left, mesh used for the simulations . Right, numerical solution with $\mathbf{Q_3}$ finite edge elements with mass-lumping.

Eigenmodes with the second family

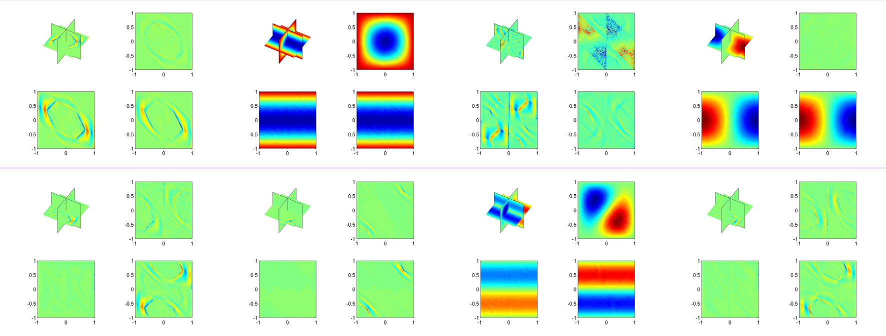
Mesh used for the simulations ($\mathbf{Q_3}$)



Eigenmodes with the second family



Eigenmodes with the second family



Two types of penalization

Mixed formulation of Maxwell equations

$$
\begin{aligned}
-\omega \int_\Omega E \cdot \varphi + \int_\Omega H \cdot \mathrm{rot}(\varphi) - i\alpha \sum_e \sum_{\text{face}} \int_{\Gamma_e} [E \cdot n][\varphi \cdot n] &= \int_\Omega f \cdot \varphi \\
-\omega \int_\Omega H \cdot \varphi + \int_\Omega \mathrm{rot}(E) \cdot \varphi - i\delta \sum_e \sum_{\text{face}} \int_{\Gamma_e} [H \times n] \cdot [\varphi \times n] &= 0
\end{aligned}
$$

Approximation space for H

$$
W_h = \{ \vec{u} \in L^2(\Omega) \text{ so that } DF_i^* \, \vec{u} \circ F_i \in (Q_r)^3 \}
$$

- Equivalence with second-order formulation ($\alpha = \delta = 0$)
- Dissipative terms of penalization
- Penalization in α does not need of a mixed formulation

Two types of penalization

Mixed formulation of Maxwell equations

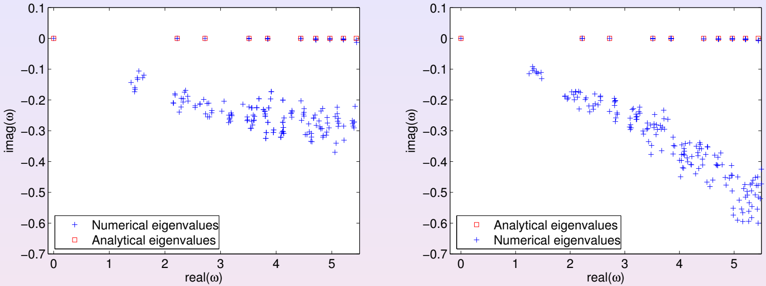
$$\begin{aligned} -\omega \int_\Omega E\cdot\varphi + \int_\Omega H\cdot \mathrm{rot}(\varphi) - i\alpha \sum_{e} \sum_{\text{face}} \int_{\Gamma_e} [E\cdot n][\varphi\cdot n] &= \int_\Omega f\cdot\varphi \\ -\omega \int_\Omega H\cdot\varphi + \int_\Omega \mathrm{rot}(E)\cdot\varphi - i\delta \sum_{e} \sum_{\text{face}} \int_{\Gamma_e} [H\times n]\cdot[\varphi\times n] &= 0 \end{aligned}$$

Approximation space for H

$$W_h = \{ \vec{u} \in L^2(\Omega) \text{ so that } DF_i^* \, \vec{u} \circ F_i \in (Q_r)^3 \}$$

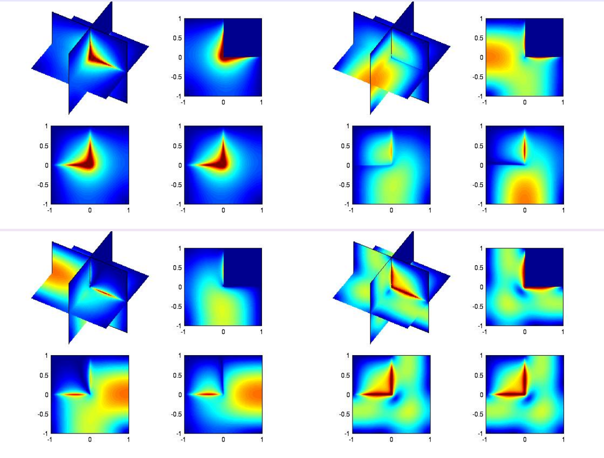
- Equivalence with second-order formulation ($\alpha = \delta = 0$)
- Dissipative terms of penalization
- Penalization in α does not need of a mixed formulation

Effects of penalization



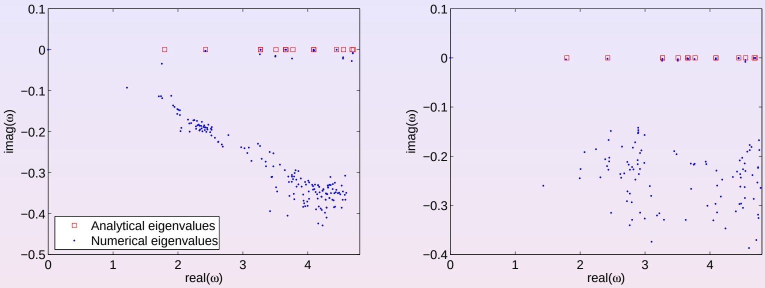
- Case of the cubic cavity meshed with slip tetrahedrals
- At left $\alpha = 0.1$, at right $\alpha = 0.5$

Effects of penalization



Four modes of the Fichera corner

Effects of penalization



- Case of the Fichera corner
- At left $\alpha = 0.5$, at right $\delta = 0.5$

- Both penalizations efficient for regular domains
- Delta-penalization more robust for singular domains

Discontinuous Galerkin method

$$
\begin{aligned}
-\omega \int_{K_i} \varepsilon \vec{E} \cdot \vec{\varphi} \; - \int_{K_i} H \, \nabla \times \vec{\varphi} \; - \int_{\partial K_i} \{H\} \, \vec{\varphi} \times \vec{\nu} &= 0 \\
-\omega \int_{K_i} \mu H \, \psi \; - \int_{K_i} \nabla \times \vec{E} \, \psi \; - \frac{1}{2} \int_{\partial K_i} [\vec{E}] \times \vec{\nu} \, \psi &= 0
\end{aligned}
$$

Let us notice that

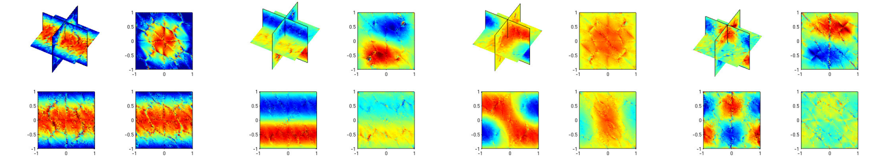
$$
\begin{aligned}
\{H\} &= \tfrac{1}{2}(H_i + H_j) \\
[\,\vec{E}] &= (\vec{E}_i - \vec{E}_j)
\end{aligned}
\tag{1}
$$

Discontinuous Galerkin method

$$
\begin{aligned}
-\omega \int_{K_i} \varepsilon \vec{E} \cdot \vec{\varphi} \; - \int_{K_i} H \, \nabla \times \vec{\varphi} \; - \int_{\partial K_i} \{H\} \, \vec{\varphi} \times \vec{\nu} \; &= \; 0 \\
-\omega \int_{K_i} \mu H \, \psi \; - \int_{K_i} \nabla \times \vec{E} \, \psi \; - \frac{1}{2} \int_{\partial K_i} [\vec{E}] \times \vec{\nu} \, \psi \; &= \; 0
\end{aligned}
$$

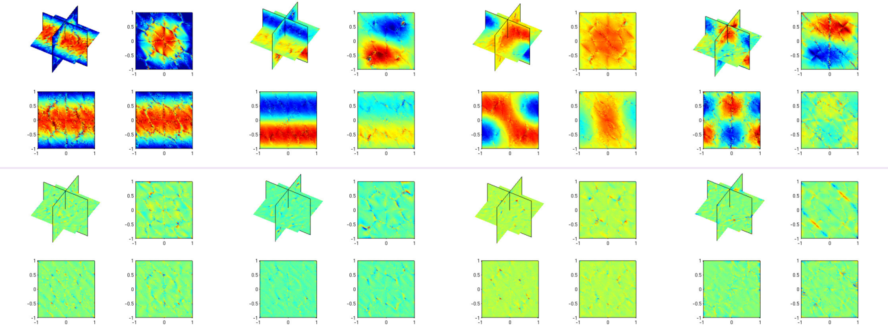
- Unknowns in $L^2 \Rightarrow$ Gauss points instead of GL points
- Mass lumping and fast matrix vector product
- Thesis of S. Pernet, in time-domain

Eigenmodes in DG method (3-D)



- Constant number of spurious for regular meshes
- Increasing number of spurious modes, otherwise

Eigenmodes in DG method (3-D)



- Constant number of spurious for regular meshes
- Increasing number of spurious modes, otherwise

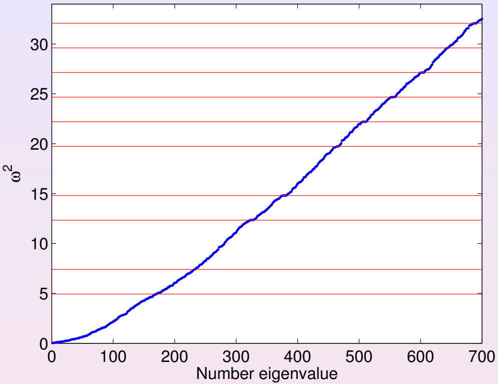
Penalization terms, eigenvalues

To the first equation in E, we add :

$$-i\omega\,\alpha \int_{\partial K_i} [\mathbf{E} \times \mathbf{n}] \cdot \varphi \times \mathbf{n}\, dx$$

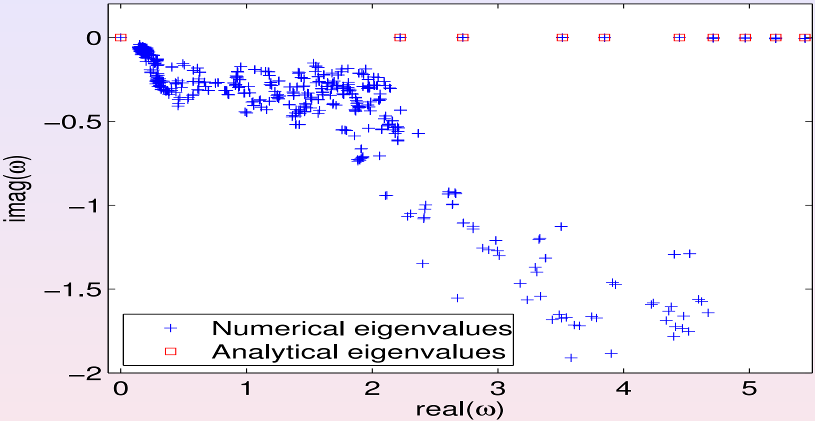
We take $\alpha \, = \, 0.5$

Penalization terms, eigenvalues



- Eigenvalues, if no penalization is used $\alpha = 0$
- Blue points are numeric eigenvalues, red lines analytic eigenvalues.

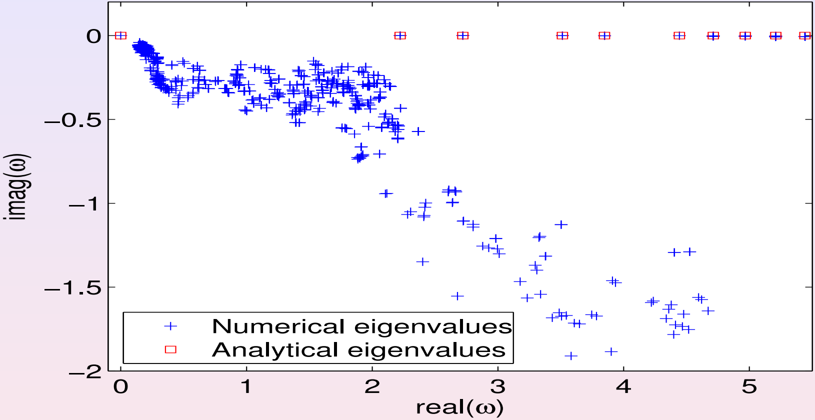
Penalization terms, eigenvalues



Eigenvalues if penalization is used $\alpha = 0.5$

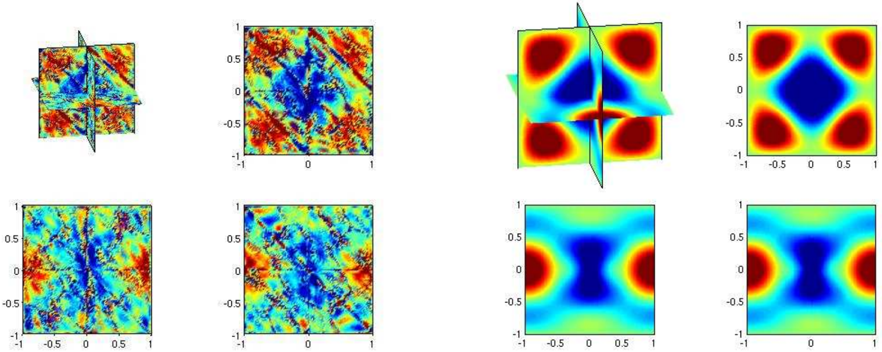
Blue points are numeric eigenvalues, red squares analytic eigenvalues.

Penalization terms, eigenvalues



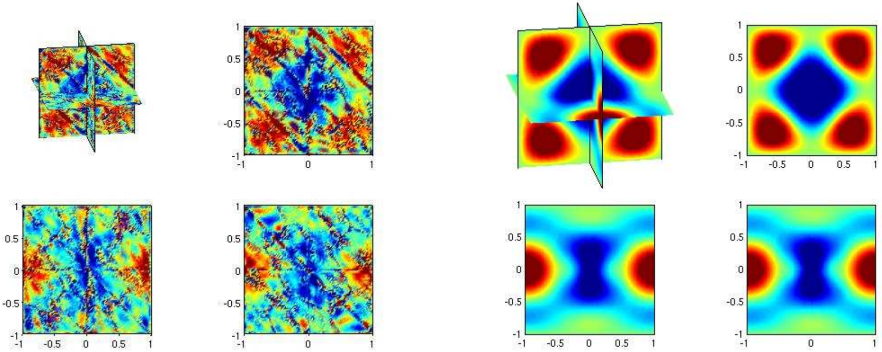
- Penalization terms reject ALL spurious modes in complex plane
- Persistance of some spurious mode near 0

Effects of penalization



At left, numerical solution with $\alpha = 0$, at right with $\alpha = 0.5$

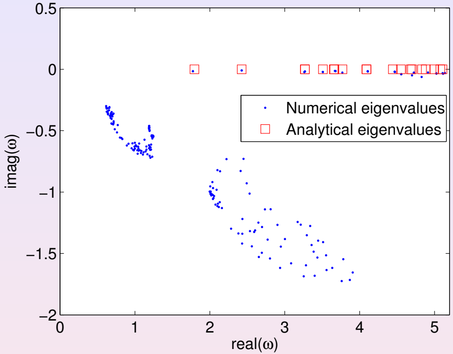
Effects of penalization



At left, numerical solution with $\alpha = 0$, at right with $\alpha = 0.5$

- Fine solution on split meshes
- Negligible overcost in computational time

Effects of penalization



Eigenvalues for the Fichera corner, on split tetrahedral mesh. 4

- Good approximation of singular eigenvalues
- No need to add penalization terms in 2-D

Nedelec's first family on hexahedra

Space of approximation

$$V_h = \{ \vec{u} \in \mathrm{H(curl,}\Omega) \text{ so that } DF_i^t \, \vec{u} \circ F_i \in Q_{r-1,r,r} \times Q_{r,r-1,r} \times Q_{r,r,r-1} \}$$

Basis functions

$$\hat{\vec{\varphi}}^1_{i,j,k}(\hat{x},\hat{y},\hat{z}) = \hat{\psi}^G_i(\hat{x}) \, \hat{\psi}^{GL}_j(\hat{y}) \, \hat{\psi}^{GL}_k(\hat{z}) \, \vec{e_x} \quad 1 \le i \le r \quad 1 \le j,k \le r+1$$

$$\hat{\vec{\varphi}}^2_{j,i,k}(\hat{x},\hat{y},\hat{z}) = \hat{\psi}^{GL}_j(\hat{x}) \, \hat{\psi}^G_i(\hat{y}) \, \hat{\psi}^{GL}_k(\hat{z}) \, \vec{e_y} \quad 1 \le i \le r \quad 1 \le j,k \le r+1$$

$$\hat{\vec{\varphi}}^3_{k,j,i}(\hat{x},\hat{y},\hat{z}) = \hat{\psi}^{GL}_k(\hat{x}) \, \hat{\psi}^{GL}_j(\hat{y}) \, \hat{\psi}^G_i(\hat{x}) \, \vec{e_z} \quad 1 \le i \le r \quad 1 \le j,k \le r+1$$

ψ^G_i, ψ^{GL}_i lagragian functions linked respectively with Gauss points and Gauss-Lobatto points.
See. G. Cohen, P. Monk, Gauss points mass lumping

Nedelec's first family on hexahedra

Space of approximation

$$V_h = \{ \vec{u} \in \ \mathrm{H(curl,\Omega)} \ \text{so that } DF_i^t \, \vec{u} \circ F_i \in Q_{r-1,r,r} \times Q_{r,r-1,r} \times Q_{r,r,r-1} \}$$

Basis functions

$$\hat{\vec{\varphi}}^1_{i,j,k}(\hat{x},\hat{y},\hat{z}) = \hat{\psi}^G_i(\hat{x}) \, \hat{\psi}^{GL}_j(\hat{y}) \, \hat{\psi}^{GL}_k(\hat{z}) \, \vec{e_x} \quad 1 \le i \le r \quad 1 \le j,k \le r+1$$

$$\hat{\vec{\varphi}}^2_{j,i,k}(\hat{x},\hat{y},\hat{z}) = \hat{\psi}^{GL}_j(\hat{x}) \, \hat{\psi}^{G}_i(\hat{y}) \, \hat{\psi}^{GL}_k(\hat{z}) \, \vec{e_y} \quad 1 \le i \le r \quad 1 \le j,k \le r+1$$

$$\hat{\vec{\varphi}}^3_{k,j,i}(\hat{x},\hat{y},\hat{z}) = \hat{\psi}^{GL}_k(\hat{x}) \, \hat{\psi}^{GL}_j(\hat{y}) \, \hat{\psi}^{G}_i(\hat{x}) \, \vec{e_z} \quad 1 \le i \le r \quad 1 \le j,k \le r+1$$

ψ_i^G, ψ_i^{GL} lagragian functions linked respectively with Gauss points and Gauss-Lobatto points.

See. G. Cohen, P. Monk, Gauss points mass lumping

Nedelec's first family on hexahedra

Space of approximation

$$V_h = \{ \vec{u} \in \ \mathrm{H(curl,}\Omega) \text{ so that } DF_i^t \, \vec{u} \circ F_i \in Q_{r-1,r,r} \times Q_{r,r-1,r} \times Q_{r,r,r-1} \}$$

Basis functions

$$\hat{\vec{\varphi}}^1_{i,j,k}(\hat{x},\hat{y},\hat{z}) = \hat{\psi}^G_i(\hat{x}) \, \hat{\psi}^{GL}_j(\hat{y}) \, \hat{\psi}^{GL}_k(\hat{z}) \, \vec{e_x} \quad 1 \le i \le r \quad 1 \le j,k \le r+1$$

$$\hat{\vec{\varphi}}^2_{j,i,k}(\hat{x},\hat{y},\hat{z}) = \hat{\psi}^{GL}_j(\hat{x}) \, \hat{\psi}^{G}_i(\hat{y}) \, \hat{\psi}^{GL}_k(\hat{z}) \, \vec{e_y} \quad 1 \le i \le r \quad 1 \le j,k \le r+1$$

$$\hat{\vec{\varphi}}^3_{k,j,i}(\hat{x},\hat{y},\hat{z}) = \hat{\psi}^{GL}_k(\hat{x}) \, \hat{\psi}^{GL}_j(\hat{y}) \, \hat{\psi}^{G}_i(\hat{x}) \, \vec{e_z} \quad 1 \le i \le r \quad 1 \le j,k \le r+1$$

ψ_i^G, ψ_i^{GL} lagragian functions linked respectively with Gauss points and Gauss-Lobatto points.
See. G. Cohen, P. Monk, Gauss points mass lumping

Elementary matrices

Mass matrix :

$$(M_h)_{i,j} = \int_{\hat{K}} J_i \, DF_i^{-1} \, \varepsilon \, DF_i^{*-1} \, \hat{\varphi}_i \cdot \hat{\varphi}_k \, d\hat{x}$$

Stiffness matrix :

$$(K_h)_{i,j} = \int_{\hat{K}} \frac{1}{J_i} \, DF_i^t \, \mu^{-1} \, DF_i \, \hat{\nabla} \times \hat{\varphi}_i \cdot \hat{\nabla} \times \hat{\varphi}_k \, d\hat{x}$$

- Use of Gauss-Lobatto quadrature $(\omega_k^{GL}, \xi_k^{GL})$
- Block-diagonal matrix

$$(A_h)_{k,k} = \left[J_i \, DF_i^{-1} \, \varepsilon \, DF_i^{*-1} \right] (\xi_k^{GL}) \omega_k^{GL}$$

- Block-diagonal matrix

$$(B_h)_{k,k} = \left[\frac{1}{J_i} DF_i^t \mu^{-1} \, DF_i \right] (\xi_k^{GL}) \omega_k^{GL}$$

Elementary matrices

Mass matrix :

$$(M_h)_{i,j} \;=\; \int_{\hat{K}} J_i \, DF_i^{-1} \, \varepsilon \, DF_i^{*-1} \, \hat{\varphi}_i \cdot \hat{\varphi}_k \, d\hat{x}$$

Stiffness matrix :

$$(K_h)_{i,j} \;=\; \int_{\hat{K}} \frac{1}{J_i} \, DF_i^t \, \mu^{-1} \, DF_i \, \hat{\nabla} \times \hat{\varphi}_i \cdot \hat{\nabla} \times \hat{\varphi}_k \, d\hat{x}$$

- Use of Gauss-Lobatto quadrature (ω_k^{GL}, ξ_k^{GL})
- Block-diagonal matrix

$$(A_h)_{k,k} = \left[J_i \, DF_i^{-1} \, \varepsilon \, DF_i^{*-1} \right] (\xi_k^{GL}) \omega_k^{GL}$$

- Block-diagonal matrix

$$(B_h)_{k,k} = \left[\frac{1}{J_i} DF_i^t \mu^{-1} \, DF_i \right] (\xi_k^{GL}) \omega_k^{GL}$$

Elementary matrices

Mass matrix :

$$(M_h)_{i,j} \;=\; \int_{\hat{K}} J_i\, DF_i^{-1}\, \varepsilon\, DF_i^{*-1}\, \hat{\varphi}_i \cdot \hat{\varphi}_k\, d\hat{x}$$

Stiffness matrix :

$$(K_h)_{i,j} \;=\; \int_{\hat{K}} \frac{1}{J_i}\, DF_i^t\, \mu^{-1}\, DF_i\, \hat{\nabla} \times \hat{\varphi}_i \cdot \hat{\nabla} \times \hat{\varphi}_k\, d\hat{x}$$

- Use of Gauss-Lobatto quadrature $(\omega_k^{GL}, \xi_k^{GL})$
- Block-diagonal matrix

$$(A_h)_{k,k} = \left[J_i\, DF_i^{-1}\, \varepsilon\, DF_i^{*-1} \right] (\xi_k^{GL}) \omega_k^{GL}$$

- Block-diagonal matrix

$$(B_h)_{k,k} = \left[\frac{1}{J_i} DF_i^t \mu^{-1}\, DF_i \right] (\xi_k^{GL}) \omega_k^{GL}$$

Fast matrix vector product

Let us introduce the two following matrices, independant of the geometry :

$$\hat{C}_{i,j} = \hat{\varphi}_i(\xi_j^{GL}) \qquad \hat{R}_{i,j} = \hat{\nabla} \times \hat{\varphi}_i^{GL}(\xi_j^{GL})$$

Then, we have : $M_h = \hat{C}\, A_h \hat{C}^* \qquad K_h = \hat{C} \hat{R}\, B_h \hat{R}^* \hat{C}^*$

- Complexity of $\hat{C}\, U$: $6\,(r+1)^4$ operations in 3-D
- Complexity of $\hat{R}\, U$: $12\,(r+1)^4$ operations in 3-D
- Complexity of $A_h\, U + B_h\, U$: $30\,(r+1)^3$ operations

Complexity of standard matrix vector product $18r^3\,(r+1)^3$

- Matrix-vector product 67% slower by using exact integration

Fast matrix vector product

Let us introduce the two following matrices, independant of the geometry :

$$\hat{C}_{i,j} = \hat{\varphi}_i(\xi_j^{GL}) \qquad \hat{R}_{i,j} = \hat{\nabla} \times \hat{\varphi}_i^{GL}(\xi_j^{GL})$$

Then, we have : $M_h = \hat{C}\, A_h \hat{C}^* \qquad K_h = \hat{C} \hat{R}\, B_h \hat{R}^* \hat{C}^*$

- Complexity of $\hat{C}\, U$: $6\,(r+1)^4$ operations in 3-D
- Complexity of $\hat{R}\, U$: $12\,(r+1)^4$ operations in 3-D
- Complexity of $A_h\, U + B_h\, U$: $30\,(r+1)^3$ operations

Complexity of standard matrix vector product $18 r^3\,(r+1)^3$

- Matrix-vector product 67% slower by using exact integration

Fast matrix vector product

Let us introduce the two following matrices, independant of the geometry :

$$\hat{C}_{i,j} = \hat{\varphi}_i(\xi_j^{GL}) \qquad \hat{R}_{i,j} = \hat{\nabla} \times \hat{\varphi}_i^{GL}(\xi_j^{GL})$$

Then, we have : $M_h = \hat{C}\, A_h \hat{C}^* \qquad K_h = \hat{C} \hat{R}\, B_h \hat{R}^* \hat{C}^*$

- Complexity of $\hat{C}\, U$: $6\,(r+1)^4$ operations in 3-D
- Complexity of $\hat{R}\, U$: $12\,(r+1)^4$ operations in 3-D
- Complexity of $A_h\, U + B_h\, U$: $30\,(r+1)^3$ operations

Complexity of standard matrix vector product $18 r^3\,(r+1)^3$

- Matrix-vector product 67% slower by using exact integration

Fast matrix vector product

Let us introduce the two following matrices, independant of the geometry :

$$\hat{C}_{i,j} = \hat{\varphi}_i(\xi_j^{GL}) \qquad \hat{R}_{i,j} = \hat{\nabla} \times \hat{\varphi}_i^{GL}(\xi_j^{GL})$$

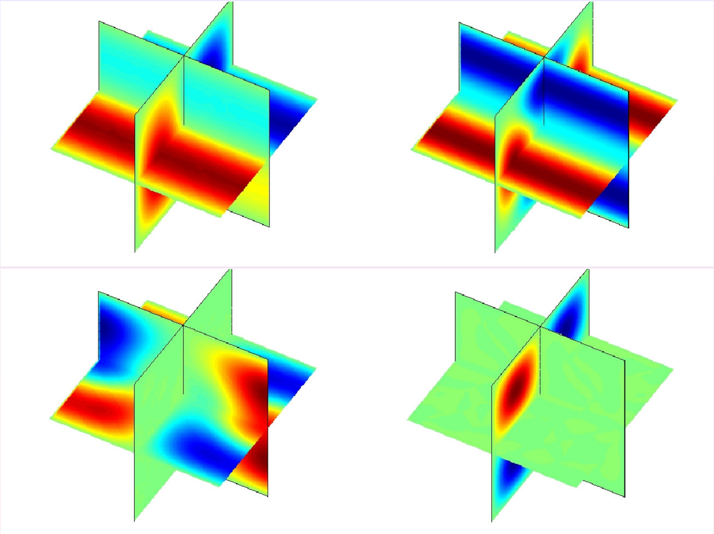
Then, we have : $M_h = \hat{C}\, A_h \hat{C}^* \qquad K_h = \hat{C}\hat{R}\, B_h \hat{R}^* \hat{C}^*$

- Complexity of $\hat{C}\, U$: $6\,(r+1)^4$ operations in 3-D
- Complexity of $\hat{R}\, U$: $12\,(r+1)^4$ operations in 3-D
- Complexity of $A_h\, U + B_h\, U$: $30\,(r+1)^3$ operations

Complexity of standard matrix vector product $18 r^3\,(r+1)^3$

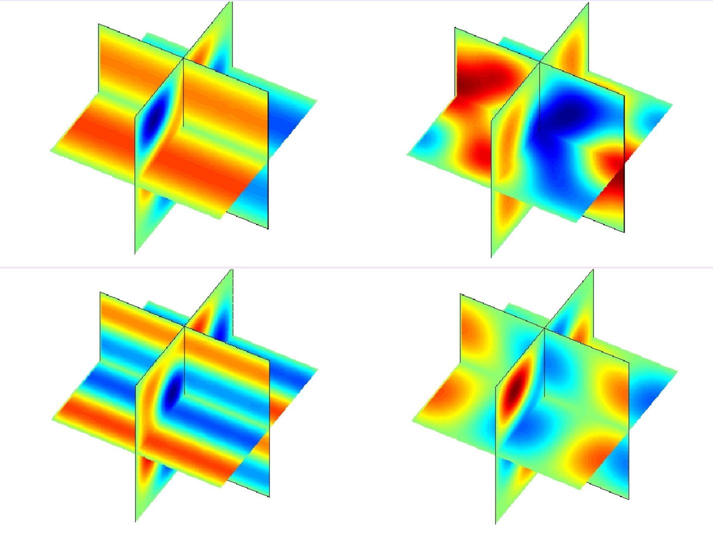
- Matrix-vector product 67% slower by using exact integration

Spurious free method



- Approximate integration leads to a spurious-free method

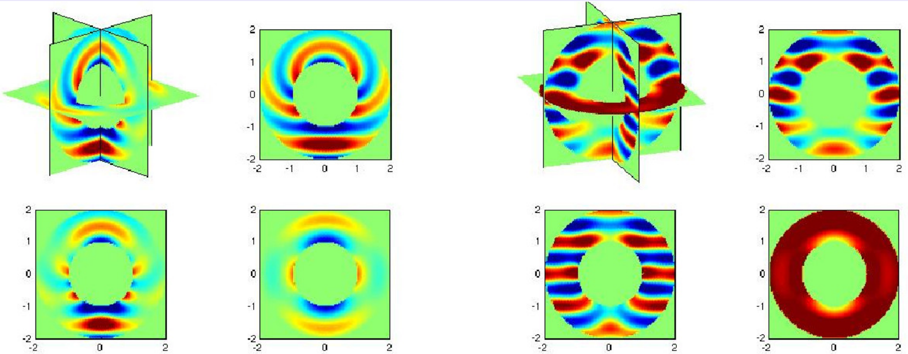
Spurious free method



- Approximate integration leads to a spurious-free method

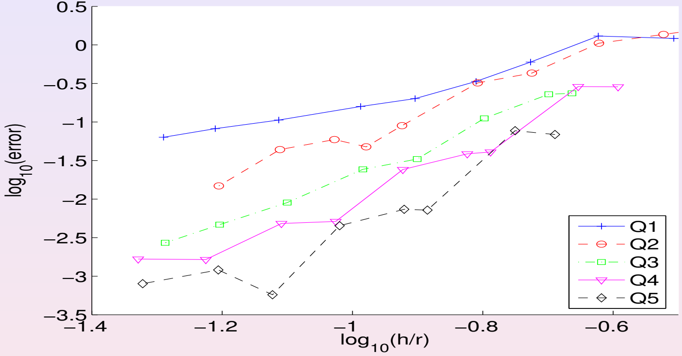
Convergence of the method

Scattering by a perfectly conductor sphere $E \times n = 0$



Convergence of the method

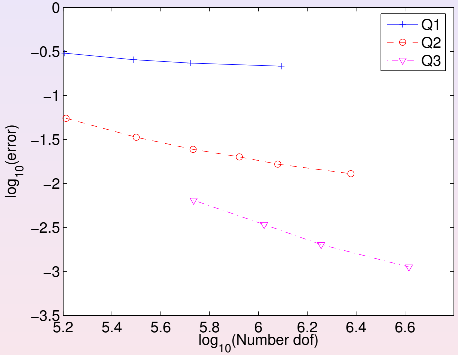
Convergence of Nedelec's first family on regular meshes



- Optimal convergence $O(h^r)$ in H(curl,Ω) norm

Convergence of the method

Convergence on tetrahedral meshes split in hexahedra



- Loss of one order, convergence $O(h^{r-1})$ in H(curl,Ω) norm

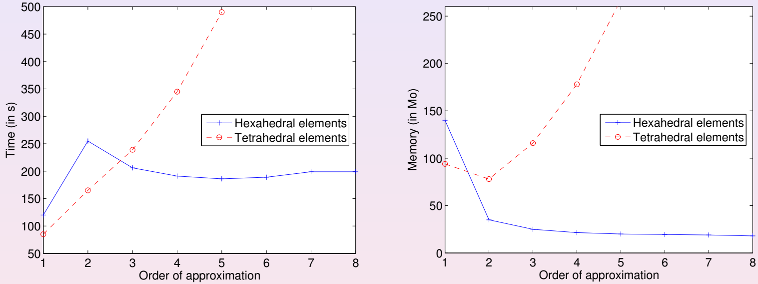
Is the matrix-vector product fast ?

Comparison between standard formulation and discrete factorization

Order	1	2	3	4	5
Time, standard formulation	55s	127s	224s	380s	631
Time, discrete factorization	244s	128s	106s	97s	96s
Storage, standard formulation	18 Mo	50 Mo	105 Mo	187 Mo	308 Mo
Storage, discrete factorization	23 Mo	9.9 Mo	6.9 Mo	5.7 Mo	5.0 Mo

Is the matrix-vector product fast ?

Comparison between tetrahedral and hexahedral elements



At left, time computation for a thousand iterations of COCG
At right, storage for mesh and matrices

Comparison DG method vs first family

- Both methods are spectrally correct
- Both methods have a fast MV product
- DG needs more dof, because DG $\mathbf{Q}_3$ is less accurate than Family1 $\mathbf{Q}_4$
- DG needs more storage for direct solvers (about 4 times than first family)
- DG can deal easily non-conforming meshes
- DDM methods are faster with DG

Comparison DG method vs first family

- Both methods are spectrally correct
- Both methods have a fast MV product
- DG needs more dof, because DG $\mathbf{Q_3}$ is less accurate than Family1 $\mathbf{Q_4}$
- DG needs more storage for direct solvers (about 4 times than first family)
- DG can deal easily non-conforming meshes
- DDM methods are faster with DG

Comparison DG method vs first family

- Both methods are spectrally correct
- Both methods have a fast MV product
- DG needs more dof, because DG $\mathbf{Q_3}$ is less accurate than Family1 $\mathbf{Q_4}$
- DG needs more storage for direct solvers (about 4 times than first family)
- DG can deal easily non-conforming meshes
- DDM methods are faster with DG

Preconditioning used

- Incomplete factorization with threshold on the damped Maxwell equation :

$$-k^2(\alpha + i\beta)\varepsilon E - \nabla \times (\frac{1}{\mu}\nabla \times E) = 0$$

  - ILUT threshold ≥ 0.05 in order to have a low storage

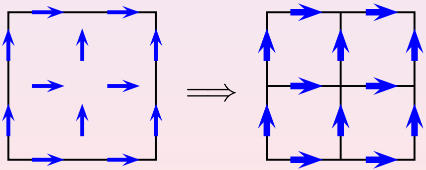
- Without damping, both preconditioners **does not lead** to

Preconditioning used

- Incomplete factorization with threshold on the damped Maxwell equation :

$$-k^2(\alpha + i\beta)\varepsilon\, E \;-\; \nabla \times (\frac{1}{\mu}\nabla \times E) \;=\; 0$$

  - ILUT threshold ≥ 0.05 in order to have a low storage
  - Use of a Q_1 subdivided mesh to compute matrix



Preconditioning used

- Incomplete factorization with threshold on the damped Maxwell equation :

$$-k^2(\alpha + i\beta)\varepsilon\, E \;-\; \nabla \times (\frac{1}{\mu}\nabla \times E) \;=\; 0$$

- Multigrid method on the damped Maxwell equation
  - Use of the $\mathbf{Q_1}$ mesh to do the multigrid iteration

- Without damping, both preconditioners **does not lead** to convergence.

- A good choice of parameter is $\alpha = 0.7$, $\beta = 0.35$

Transparent condition

Silver-Muller condition is a first-order ABC :

$$E \times n + n \times H \times n = 0$$

- Use of a transparent condition based on integral representation formulas :

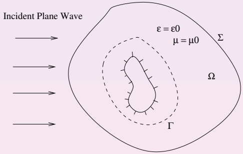
$$E^{pot}(x) = \int_\Gamma ik\,(G(x,y) + \frac{1}{k^2}\nabla_y\nabla_y G(x,y))\,(n\times H)(y)\,dy + \int_\Gamma (n\times E)(y)\times\nabla_y G(x,y)\,dy$$

new boundary condition $E \times n + n \times H \times n = E^{pot} \times n + n \times H^{pot} \times n$

Transparent condition

Silver-Muller condition is a first-order ABC :

$$\color{red}{E} \times n + n \times H \times n = 0$$

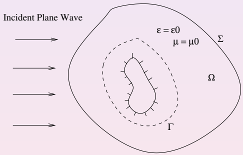


- Needs of a virtual boundary Γ
- GMRES iterations to solve linear system

Transparent condition

Silver-Muller condition is a first-order ABC :

$$E \times n + n \times H \times n = 0$$



- Needs of a virtual boundary Γ
- GMRES iterations to solve linear system

- C. Hazard, M. Lenoir, On the solution of time-harmonic scattering problems for Maxwell's equations

Radar cross section

Computation of far field of the electromagnetic objects by the formula

$$\sigma(\mathbf{u}) = \frac{k^2}{4\pi} \int_\Sigma e^{ik\mathbf{u}\cdot\mathbf{OM}} \left[\mathbf{u} \times (\mathbf{n} \times \mathbf{H}) + (u \otimes u - I)(\mathbf{E} \times \mathbf{n}) \right] dM$$

- Bistatic RCS : the vector of observation $\mathbf{u}$ varies
- Monostatic RCS : the wave vector $\mathbf{k}$ varies and $\mathbf{u} = \mathbf{k}$

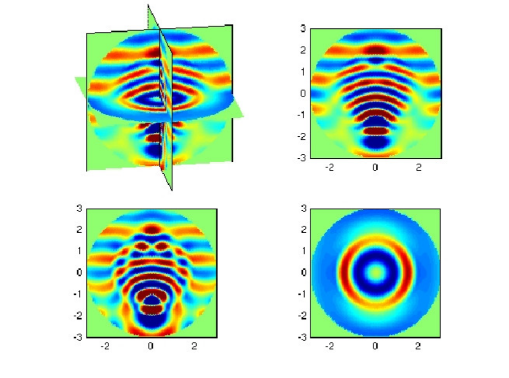
Radar cross section

Computation of far field of the electromagnetic objects by the formula

$$\sigma(\mathbf{u}) = \frac{k^2}{4\pi} \int_\Sigma e^{ik\mathbf{u}\cdot\mathbf{OM}} \left[\mathbf{u} \times (\mathbf{n} \times \mathbf{H}) + (u \otimes u - I)(\mathbf{E} \times \mathbf{n}) \right] dM$$

- Bistatic RCS : the vector of observation $\mathbf{u}$ varies
- Monostatic RCS : the wave vector $\mathbf{k}$ varies and $\mathbf{u} = \mathbf{k}$

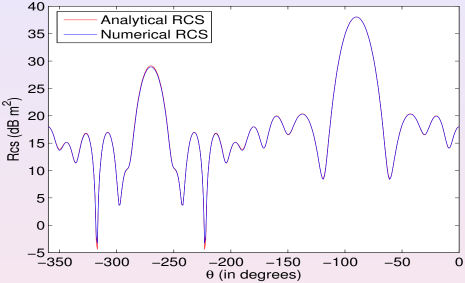
Scattering by a dielectric sphere



- Sphere of radius 2 with $\varepsilon = 3.5$ $\mu = 1$
- Outside boundary on a sphere of radius 3.

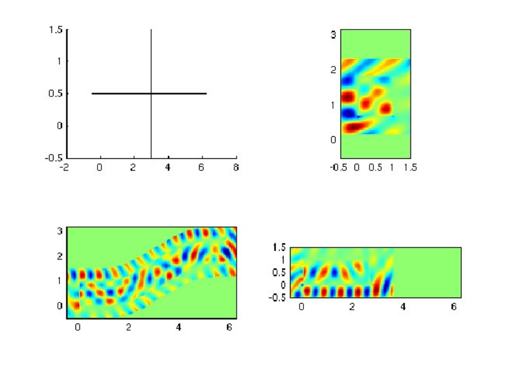
Scattering by a dielectric sphere

How many dofs/time to reach an error less than 0.5 dB



Finite Element	Q_2	Q_4	Q_6	Q_8
Nb dofs	940 000	88 000	230 000	88 000
No preconditioning	19 486 s	894 s	4 401 s	1 484 s
ILUT(0.05)	-	189 s	1 035 s	307 s
Two-grid	4 4344 s	488 s	1 095 s	952 s

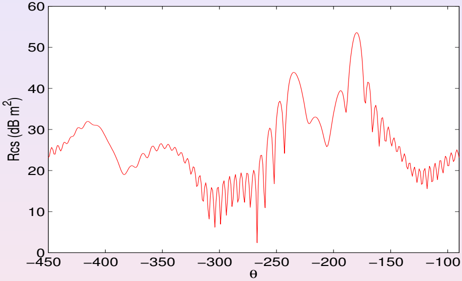
Scattering by a cobra cavity



- Cobra cavity of length 10, and depth 2
- Outside boundary at a distance of 1

Scattering by a cobra cavity

How many dofs/time to reach an error less than 0.5 dB



Finite Element	Q_4	Q_6
Nb dofs	412 000	187 000
No preconditioning	14 039 s	12 096 s
ILUT(0.05)	2 247 s	846 s
Two-grid	9 294 s	10 500 s

Outline

Discontinuous Galerkin Method

Let $\Omega = \bigcup_{i=1}^{N_e} K_i$. Find $\vec{E}(.,t) \in [L^2(\Omega)]^3,\ \vec{H}(.,t) \in [L^2(\Omega)]^3$ s.t.

$$\frac{\partial}{\partial t}\int_{K_i} \underline{\underline{\varepsilon}}\, \vec{E}_{K_i} \cdot \vec{\varphi}_{K_i}\, dx - \int_{K_i} \nabla \wedge \vec{H}_{K_i} \cdot \vec{\varphi}_{K_i}\, dx$$

$$+ \int_{K_i} \underline{\underline{\sigma}}\, \vec{E}_{K_i} \cdot \vec{\varphi}_{K_i}\, dx + \int_{K_i} \vec{J} \cdot \vec{\varphi}_{K_i}\, dx =$$

$$\int_{\partial K_i} \alpha [\vec{n}_{K_i} \wedge (\vec{E} \wedge \vec{n}_{K_i})]_{\partial K_i}^{K_i} \cdot \vec{\varphi}_{K_i}\, d\sigma + \int_{\partial K_i} \beta [\vec{H} \wedge \vec{n}_{K_i}]_{\partial K_i}^{K_i} \cdot \vec{\varphi}_{K_i}\, d\sigma,$$

$\forall \vec{\varphi}_{K_i} \in H(curl, K_i)$

$$\frac{\partial}{\partial t}\int_{K_i} \underline{\underline{\mu}}\, \vec{H}_{K_i} \cdot \vec{\psi}_{K_i}\, dx + \int_{K_i} \nabla \wedge \vec{E}_{K_i} \cdot \vec{\psi}_{K_i}\, dx =$$

$$\int_{\partial K_i} \gamma [\vec{E} \wedge \vec{n}_{K_i}]_{\partial K_i}^{K_i} \cdot \vec{\psi}_{K_i}\, d\sigma + \int_{\partial K_i} \delta [\vec{n}_{K_i} \wedge (\vec{H} \wedge \vec{n}_{K_i})]_{\partial K_i}^{K_i} \cdot \vec{\psi}_{K_i}\, d\sigma,$$

$$\forall \vec{\psi}_{K_i} \in H(curl, K_i)$$

+ metallic boundary condition on $\Gamma_b = \partial\Omega$ and initial conditions,

where $\vec{E}_{K_i} = \vec{E}_{|K_i}$, $\vec{H}_{K_i} = \vec{H}_{|K_i}$, $\vec{\varphi}_{K_i} = \vec{\varphi}_{|K_i}$, $\vec{\psi}_{K_i} = \vec{\varphi}_{|K_i}$ and α, β, γ, δ real constant parameters.

Discontinuous Galerkin Methods for Time-Domain

$$\frac{\partial}{\partial t}\int_{K_i} \underline{\underline{\mu}}\, \vec{H}_{K_i} \cdot \vec{\psi}_{K_i}\, dx + \int_{K_i} \nabla \wedge \vec{E}_{K_i} \cdot \vec{\psi}_{K_i}\, dx =$$

$$\int_{\partial K_i} \gamma [\vec{E} \wedge \vec{n}_{K_i}]_{\partial K_i}^{K_i} \cdot \vec{\psi}_{K_i}\, d\sigma + \int_{\partial K_i} \delta [\vec{n}_{K_i} \wedge (\vec{H} \wedge \vec{n}_{K_i})]_{\partial K_i}^{K_i} \cdot \vec{\psi}_{K_i}\, d\sigma,$$

$$\forall \vec{\psi}_{K_i} \in H(curl, K_i)$$

+ metallic boundary condition on $\Gamma_b = \partial\Omega$ and initial conditions,

where $\vec{E}_{K_i} = \vec{E}_{|K_i}$, $\vec{H}_{K_i} = \vec{H}_{|K_i}$, $\vec{\varphi}_{K_i} = \vec{\varphi}_{|K_i}$, $\vec{\psi}_{K_i} = \vec{\varphi}_{|K_i}$ and α, β, γ, δ real constant parameters.

$$\frac{\partial}{\partial t}\int_{K_i} \underline{\underline{\mu}}\, \vec{H}_{K_i} \cdot \vec{\psi}_{K_i}\, dx + \int_{K_i} \nabla \wedge \vec{E}_{K_i} \cdot \vec{\psi}_{K_i}\, dx =$$

$$\int_{\partial K_i} \gamma [\vec{E} \wedge \vec{n}_{K_i}]_{\partial K_i}^{K_i} \cdot \vec{\psi}_{K_i}\, d\sigma + \int_{\partial K_i} \delta [\vec{n}_{K_i} \wedge (\vec{H} \wedge \vec{n}_{K_i})]_{\partial K_i}^{K_i} \cdot \vec{\psi}_{K_i}\, d\sigma,$$

$$\forall \vec{\psi}_{K_i} \in H(curl, K_i)$$

+ metallic boundary condition on $\Gamma_b = \partial\Omega$ and initial conditions,

where $\vec{E}_{K_i} = \vec{E}_{|K_i}$, $\vec{H}_{K_i} = \vec{H}_{|K_i}$, $\vec{\varphi}_{K_i} = \vec{\varphi}_{|K_i}$, $\vec{\psi}_{K_i} = \vec{\varphi}_{|K_i}$ and α, β, γ, δ real constant parameters.

Discrete Energy

$$\mathcal{E}_{K_i}(t) = \sum_{K_i \subset \Omega} \left\{ \int_{K_i} (\underline{\underline{\epsilon}} \vec{E}_{K_i}) \cdot \vec{E}_{K_i} dx + \int_{K_i} (\underline{\underline{\mu}} \vec{H}_{K_i}) \cdot \vec{H}_{K_i} dx \right\}$$

1. $-\beta = \gamma = \frac{1}{2}$, $\alpha \geq 0$ and $\delta \geq 0 \Longrightarrow$

$$\boxed{\begin{aligned} \frac{\partial \mathcal{E}}{\partial t}(t) = & \sum_{\Gamma \in \mathcal{F}_i, \Gamma = K_i \cap K_j} \{-\alpha \| [\vec{n}_{K_i} \wedge (\vec{E} \wedge \vec{n}_{K_i}) - \delta \| [\vec{n}_{K_i} \wedge (\vec{H} \wedge \vec{n}_{K_i})] \|_\Gamma^2 \} \\ & \sum_{\Gamma \in \Gamma_b, \Gamma \subset K_i} \{-\alpha \| \vec{n}_{K_i} \wedge (\vec{E}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 - \delta \| \vec{n}_{K_i} \wedge (\vec{H}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 \} \end{aligned}}$$

$\Longrightarrow$ Decreasing energy: Dissipative scheme.

2. $-\beta = \gamma = \frac{1}{2}$, $\alpha = 0$ et $\delta = 0 \Longrightarrow \boxed{\frac{\partial}{\partial t}\mathcal{E}(t) = 0}$

$\Longrightarrow$ Energy conservation: Conservative scheme.

Discrete Energy

$$\mathcal{E}_{K_i}(t) = \sum_{K_i \subset \Omega} \left\{ \int_{K_i} (\underline{\underline{\epsilon}} \vec{E}_{K_i}) \cdot \vec{E}_{K_i} dx + \int_{K_i} (\underline{\underline{\mu}} \vec{H}_{K_i}) \cdot \vec{H}_{K_i} dx \right\}$$

1. $-\beta = \gamma = \frac{1}{2}$, $\alpha \geq 0$ and $\delta \geq 0 \Longrightarrow$

$$\boxed{\begin{aligned} \frac{\partial \mathcal{E}}{\partial t}(t) = & \sum_{\Gamma \in \mathcal{F}_i, \Gamma = K_i \cap K_j} \{-\alpha \| [\vec{n}_{K_i} \wedge (\vec{E} \wedge \vec{n}_{K_i}) - \delta \| [\vec{n}_{K_i} \wedge (\vec{H} \wedge \vec{n}_{K_i})] \|_\Gamma^2\} \\ & \sum_{\Gamma \in \Gamma_b, \Gamma \subset K_i} \{-\alpha \| \vec{n}_{K_i} \wedge (\vec{E}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 - \delta \| \vec{n}_{K_i} \wedge (\vec{H}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2\} \end{aligned}}$$

$\Longrightarrow$ Decreasing energy: Dissipative scheme.

2. $-\beta = \gamma = \frac{1}{2}$, $\alpha = 0$ et $\delta = 0 \Longrightarrow \boxed{\frac{\partial}{\partial t}\mathcal{E}(t) = 0}$

$\Longrightarrow$ Energy conservation: Conservative scheme.

Discrete Energy

$$\mathcal{E}_{K_i}(t) = \sum_{K_i \subset \Omega} \left\{ \int_{K_i} (\underline{\underline{\epsilon}} \vec{E}_{K_i}) \cdot \vec{E}_{K_i} dx + \int_{K_i} (\underline{\underline{\mu}} \vec{H}_{K_i}) \cdot \vec{H}_{K_i} dx \right\}$$

1. $-\beta = \gamma = \frac{1}{2}$, $\alpha \geq 0$ and $\delta \geq 0 \Longrightarrow$

$$\boxed{\begin{aligned} \frac{\partial \mathcal{E}}{\partial t}(t) = & \sum_{\Gamma \in \mathcal{F}_i, \Gamma = K_i \cap K_j} \{ -\alpha \| [\vec{n}_{K_i} \wedge (\vec{E} \wedge \vec{n}_{K_i}) - \delta \| [\vec{n}_{K_i} \wedge (\vec{H} \wedge \vec{n}_{K_i})] \|_\Gamma^2 \} \\ & \sum_{\Gamma \in \Gamma_b, \Gamma \subset K_i} \{ -\alpha \| \vec{n}_{K_i} \wedge (\vec{E}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 - \delta \| \vec{n}_{K_i} \wedge (\vec{H}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 \} \end{aligned}}$$

$\Longrightarrow$ Decreasing energy: Dissipative scheme.

2. $-\beta = \gamma = \frac{1}{2}$, $\alpha = 0$ et $\delta = 0 \Longrightarrow \boxed{\frac{\partial}{\partial t}\mathcal{E}(t) = 0}$

$\Longrightarrow$ Energy conservation: Conservative scheme.

Discrete Energy

$$\mathcal{E}_{K_i}(t) = \sum_{K_i \subset \Omega} \left\{ \int_{K_i} (\underline{\underline{\epsilon}} \vec{E}_{K_i}) \cdot \vec{E}_{K_i} dx + \int_{K_i} (\underline{\underline{\mu}} \vec{H}_{K_i}) \cdot \vec{H}_{K_i} dx \right\}$$

1. $-\beta = \gamma = \frac{1}{2}$, $\alpha \geq 0$ and $\delta \geq 0 \Longrightarrow$

$$\boxed{\begin{aligned}\frac{\partial \mathcal{E}}{\partial t}(t) = & \sum_{\Gamma \in \mathcal{F}_i, \Gamma = K_i \cap K_j} \{-\alpha \| [\vec{n}_{K_i} \wedge (\vec{E} \wedge \vec{n}_{K_i}) - \delta \| [\vec{n}_{K_i} \wedge (\vec{H} \wedge \vec{n}_{K_i})] \|_\Gamma^2 \} \\ & \sum_{\Gamma \in \Gamma_b, \Gamma \subset K_i} \{-\alpha \| \vec{n}_{K_i} \wedge (\vec{E}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 - \delta \| \vec{n}_{K_i} \wedge (\vec{H}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 \}\end{aligned}}$$

$\Longrightarrow$ Decreasing energy: Dissipative scheme.

2. $-\beta = \gamma = \frac{1}{2}$, $\alpha = 0$ et $\delta = 0 \Longrightarrow \boxed{\frac{\partial}{\partial t}\mathcal{E}(t) = 0}$

$\Longrightarrow$ Energy conservation: Conservative scheme.

Discrete Energy

$$\mathcal{E}_{K_i}(t) = \sum_{K_i \subset \Omega} \left\{ \int_{K_i} (\underline{\underline{\epsilon}} \vec{E}_{K_i}) \cdot \vec{E}_{K_i} dx + \int_{K_i} (\underline{\underline{\mu}} \vec{H}_{K_i}) \cdot \vec{H}_{K_i} dx \right\}$$

1. $-\beta = \gamma = \frac{1}{2}$, $\alpha \geq 0$ and $\delta \geq 0 \Longrightarrow$

$$\boxed{\begin{aligned} \frac{\partial \mathcal{E}}{\partial t}(t) = & \sum_{\Gamma \in \mathcal{F}_i, \Gamma = K_i \cap K_j} \{-\alpha \| [\vec{n}_{K_i} \wedge (\vec{E} \wedge \vec{n}_{K_i}) - \delta \| [\vec{n}_{K_i} \wedge (\vec{H} \wedge \vec{n}_{K_i})] \|_\Gamma^2 \} \\ & \sum_{\Gamma \in \Gamma_b, \Gamma \subset K_i} \{-\alpha \| \vec{n}_{K_i} \wedge (\vec{E}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 - \delta \| \vec{n}_{K_i} \wedge (\vec{H}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 \} \end{aligned}}$$

$\Longrightarrow$ Decreasing energy: Dissipative scheme.

2. $-\beta = \gamma = \frac{1}{2}$, $\alpha = 0$ et $\delta = 0 \Longrightarrow \boxed{\frac{\partial}{\partial t} \mathcal{E}(t) = 0}$

$\Longrightarrow$ Energy conservation: Conservative scheme.

Discrete Energy

$$\mathcal{E}_{K_i}(t) = \sum_{K_i \subset \Omega} \left\{ \int_{K_i} (\underline{\underline{\epsilon}} \vec{E}_{K_i}) \cdot \vec{E}_{K_i} dx + \int_{K_i} (\underline{\underline{\mu}} \vec{H}_{K_i}) \cdot \vec{H}_{K_i} dx \right\}$$

1. $-\beta = \gamma = \frac{1}{2}$, $\alpha \geq 0$ and $\delta \geq 0 \Longrightarrow$

$$\boxed{\begin{array}{l} \dfrac{\partial \mathcal{E}}{\partial t}(t) = \displaystyle\sum_{\Gamma \in \mathcal{F}_i, \Gamma = K_i \cap K_j} \{ -\alpha \| [\vec{n}_{K_i} \wedge (\vec{E} \wedge \vec{n}_{K_i}) - \delta \| [\vec{n}_{K_i} \wedge (\vec{H} \wedge \vec{n}_{K_i})] \|_\Gamma^2 \} \\ \displaystyle\sum_{\Gamma \in \Gamma_b, \Gamma \subset K_i} \{ -\alpha \| \vec{n}_{K_i} \wedge (\vec{E}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 - \delta \| \vec{n}_{K_i} \wedge (\vec{H}_{K_i} \wedge \vec{n}_{K_i}) \|_\Gamma^2 \} \end{array}}$$

$\Longrightarrow$ Decreasing energy: Dissipative scheme.

2. $-\beta = \gamma = \frac{1}{2}$, $\alpha = 0$ et $\delta = 0 \Longrightarrow \boxed{\dfrac{\partial}{\partial t} \mathcal{E}(t) = 0}$

$\Longrightarrow$ Energy conservation: Conservative scheme.

Discrete Formulation (Gauss Points)

$$B_\varepsilon \frac{\mathbf{E^{n+1}} - \mathbf{E^n}}{\Delta t} + R_h \, \mathbf{H^{n+1/2}} + B_\sigma \frac{\mathbf{E^{n+1}} + \mathbf{E^n}}{2}$$

$$+ \alpha \, D_h \, \mathbf{E^n} + \beta \, S_h \, \mathbf{H^{n+1/2}} + \mathbf{J^n} = 0,$$

$$B_\mu \frac{\mathbf{H^{n+1/2}} - \mathbf{H^{n-1/2}}}{\Delta t} + R_h \, \mathbf{E^n} + \gamma \, S_h^* \, \mathbf{E^n} + \delta \, D_h^* \, \mathbf{H^{n-1/2}} = 0,$$

Discrete Formulation (Gauss Points)

$$B_\varepsilon \, \frac{\mathbf{E^{n+1}} - \mathbf{E^n}}{\Delta t} + R_h \, \mathbf{H^{n+1/2}} + B_\sigma \, \frac{\mathbf{E^{n+1}} + \mathbf{E^n}}{2}$$

$$+ \alpha \, D_h \, \mathbf{E^n} + \beta \, S_h \, \mathbf{H^{n+1/2}} + \mathbf{J^n} = 0,$$

$$B_\mu \, \frac{\mathbf{H^{n+1/2}} - \mathbf{H^{n-1/2}}}{\Delta t} + R_h \, \mathbf{E^n} + \gamma \, S_h^* \, \mathbf{E^n} + \delta \, D_h^* \, \mathbf{H^{n-1/2}} = 0,$$

Discrete Formulation (Gauss Points)

$$B_\varepsilon \, \frac{\mathbf{E^{n+1}} - \mathbf{E^{n}}}{\Delta t} + R_h \, \mathbf{H^{n+1/2}} + B_\sigma \, \frac{\mathbf{E^{n+1}} + \mathbf{E^{n}}}{2}$$
$$+ \alpha \, D_h \, \mathbf{E^{n}} + \beta \, S_h \, \mathbf{H^{n+1/2}} + \mathbf{J^{n}} = 0,$$

$$B_\mu \, \frac{\mathbf{H^{n+1/2}} - \mathbf{H^{n-1/2}}}{\Delta t} + R_h \, \mathbf{E^{n}} + \gamma \, S_h^* \, \mathbf{E^{n}} + \delta \, D_h^* \, \mathbf{H^{n-1/2}} = 0,$$

Main Features of this Approximation

- B_ε, B_σ, B_μ: 3×3 block-diagonal symmetric mass matrices,
- R_h: very sparse matrix which needs no storage,
- S_h, S_h^*: jump block-diagonal symmetric matrices which need no storage,
- D_h, $D_h{}^*$: jump block-diagonal symmetric matrices which must be stored.

$\longrightarrow$ The dissipative terms induce a (reasonable) additonal storage.

Main Features of this Approximation

- B_ε, B_σ, B_μ: 3×3 block-diagonal symmetric mass matrices,
- R_h: very sparse matrix which needs no storage,
- S_h, S_h^*: jump block-diagonal symmetric matrices which need no storage,
- D_h, $D_h{}^*$: jump block-diagonal symmetric matrices which must be stored.

$\longrightarrow$ The dissipative terms induce a (reasonable) additonal storage.

Main Features of this Approximation

- B_ε, B_σ, B_μ: 3×3 block-diagonal symmetric mass matrices,
- R_h: very sparse matrix which needs no storage,
- S_h, S_h^*: jump block-diagonal symmetric matrices which need no storage,
- D_h, $D_h{}^*$: jump block-diagonal symmetric matrices which must be stored.

$\longrightarrow$ The dissipative terms induce a (reasonable) additonal storage.

Main Features of this Approximation

- B_ε, B_σ, B_μ: 3×3 block-diagonal symmetric mass matrices,
- R_h: very sparse matrix which needs no storage,
- S_h, S_h^*: jump block-diagonal symmetric matrices which need no storage,
- D_h, $D_h{}^*$: jump block-diagonal symmetric matrices which must be stored.

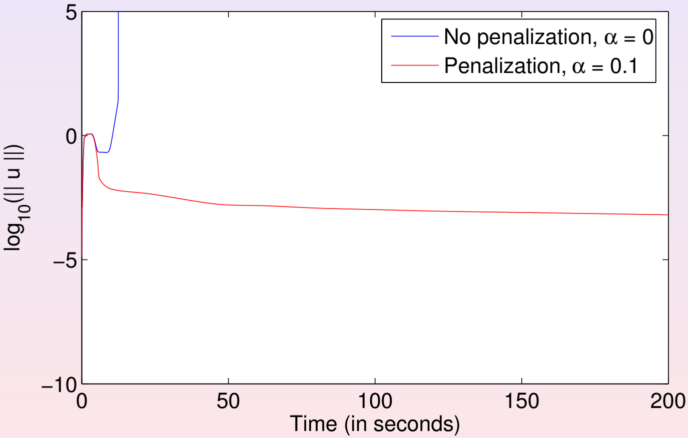
$\longrightarrow$ The dissipative terms induce a (reasonable) additonal storage.

Main Features of this Approximation

- B_ε, B_σ, B_μ: 3×3 block-diagonal symmetric mass matrices,
- R_h: very sparse matrix which needs no storage,
- S_h, S_h^*: jump block-diagonal symmetric matrices which need no storage,
- D_h, $D_h{}^*$: jump block-diagonal symmetric matrices which must be stored.

$\longrightarrow$ The dissipative terms induce a (reasonable) additonal storage.

Another Feature of Numerical Dissipation: PML Stabilization



Dielectric spherical torus

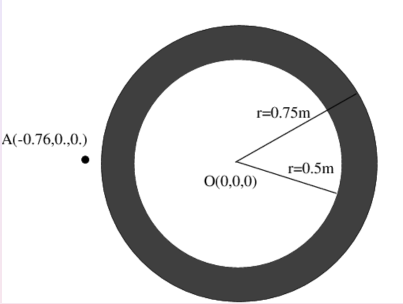
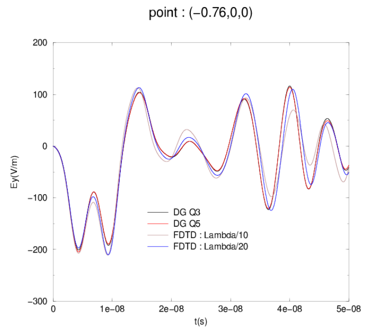


Figure: Configuration of the experiment

Dielectric spherical torus



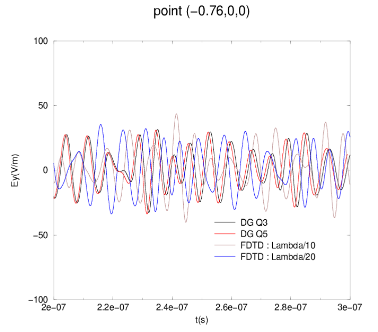


Figure: E_y component of the electric field at a point of the domain after propagation across 10λ *(left)* and 120λ *(right)*.

Dielectric spherical torus

- CPU time: FETD (Q_3) : 300 s, FDTD (20pts/λ) : 1100 s.
- Storage FDTD (20pts/λ)/FETD (Q_3) = 10.

Dielectric spherical torus

- CPU time: FETD (Q_3) : 300 s, FDTD (20pts/λ) : 1100 s.
- Storage FDTD (20pts/λ)/FETD (Q_3) = 10.

Numerical Examples

Airplane

- Frequency: 0.75 Ghz (30λ).
- Mesh: 78 000 elements, 30 000 000 DOF (Q_4).
- Storage: 1.2 Go.
- CPU time (30λ): 30 h on a monoprocessor Linux system, 2Go Ram, 3.2 GHz.

Airplane

- Frequency: 0.75 Ghz (30λ).
- Mesh: 78 000 elements, 30 000 000 DOF (Q_4).
- Storage: 1.2 Go.
- CPU time (30λ): 30 h on a monoprocessor Linux system, 2Go Ram, 3.2 GHz.

Airplane

- Frequency: 0.75 Ghz (30λ).
- Mesh: 78 000 elements, 30 000 000 DOF (Q_4).
- Storage: 1.2 Go.
- CPU time (30λ): 30 h on a monoprocessor Linux system, 2Go Ram, 3.2 GHz.

Airplane

- Frequency: 0.75 Ghz (30λ).
- Mesh: 78 000 elements, 30 000 000 DOF (Q_4).
- Storage: 1.2 Go.
- CPU time (30λ): 30 h on a monoprocessor Linux system, 2Go Ram, 3.2 GHz.

Numerical Examples

Airplane

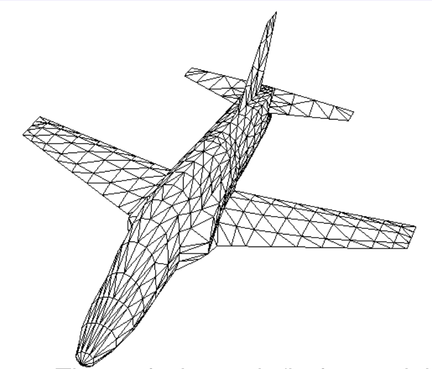


Figure: The surfacic mesh (before splitting)

Airplane

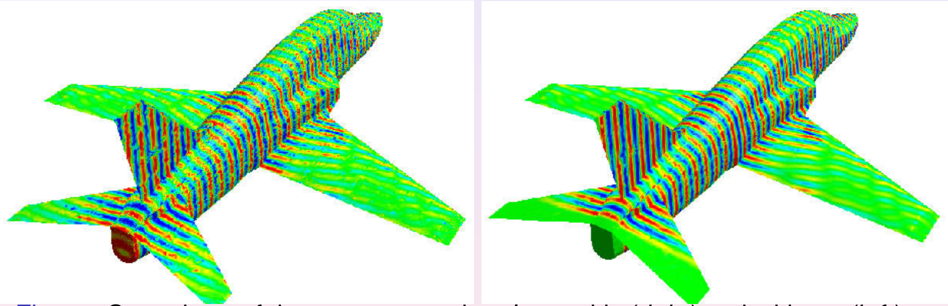


Figure: Snapshots of the currents on the plane with *(right)* and without *(left)* dissipation