# High-order Finite Element Methods on Hybrid Meshes Including Pyramids. Application to Wave Equation and Helmholtz Equation.

M. Duruflé

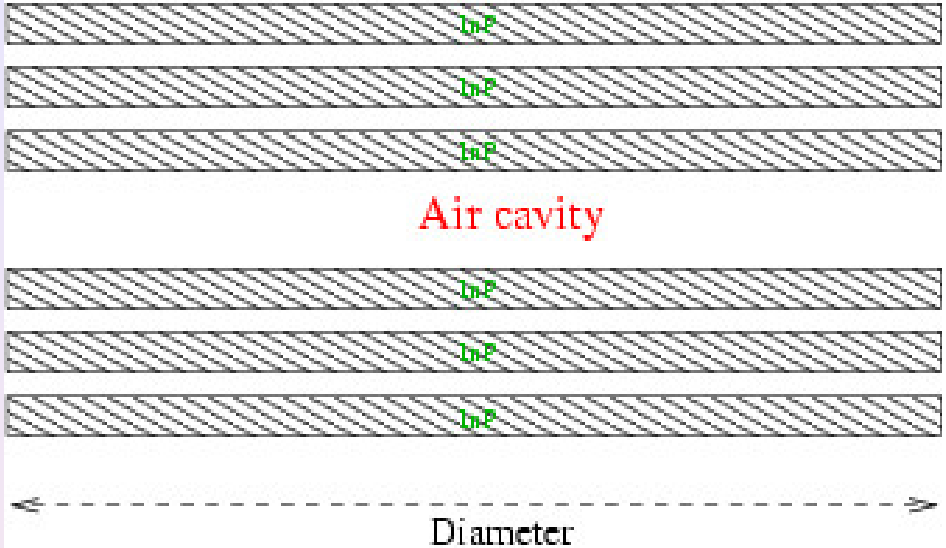
IMB, Bacchus

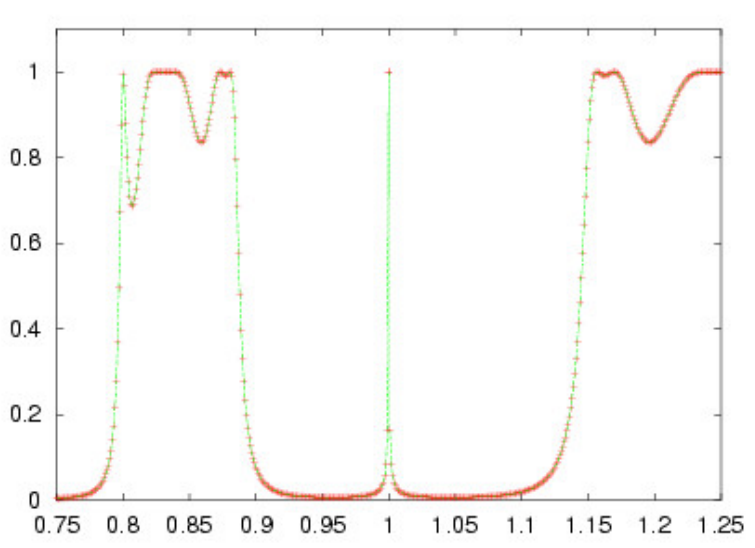
14th January 2010

# Bibliography and motivation

- S. Fauqueux, mixed spectral elements for wave and elastic equations (hexahedra)
- S. Pernet, Discontinuous Galerkin methods for Maxwell's equations (hexahedra)
- G.E. Karniadakis, S. Sherwin, T. Warburton, continuous and discontinuous finite elements on tetrahedra/prisms/pyramids by considering "degenerated" cube
- Bedrosian, Early work on pyramids, nodal basis functions for order 1 and 2
- Nigam, Philips, Recent work on finite element spaces for pyramids, infinite pyramid is the reference element

# A test case : an optical filter

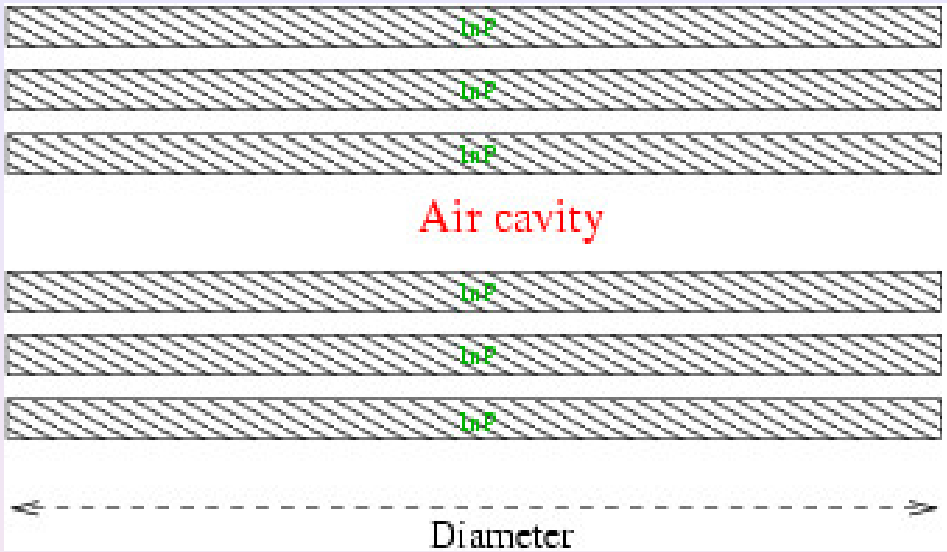


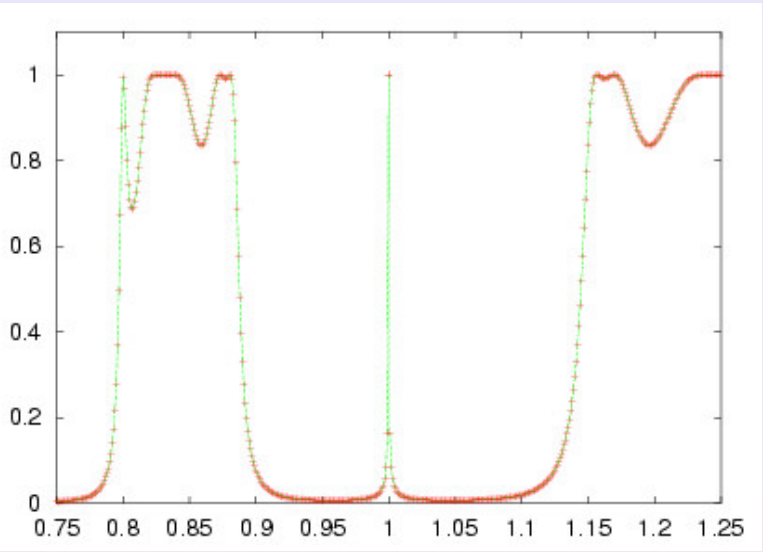




On the right, transmission coefficient versus frequency

- 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

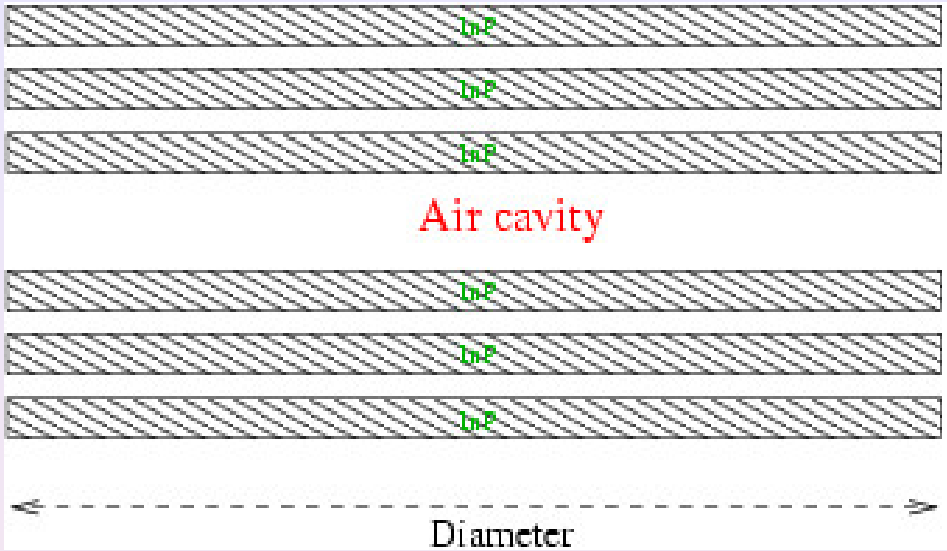


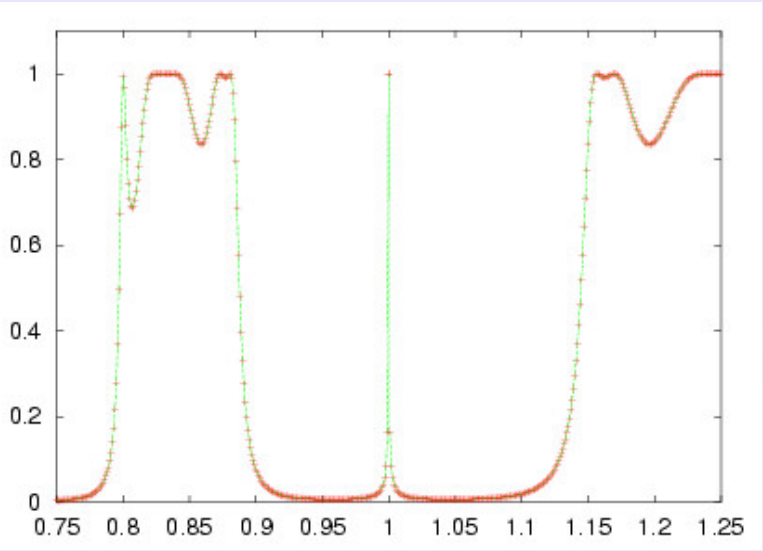




On the right, transmission coefficient versus frequency

- 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

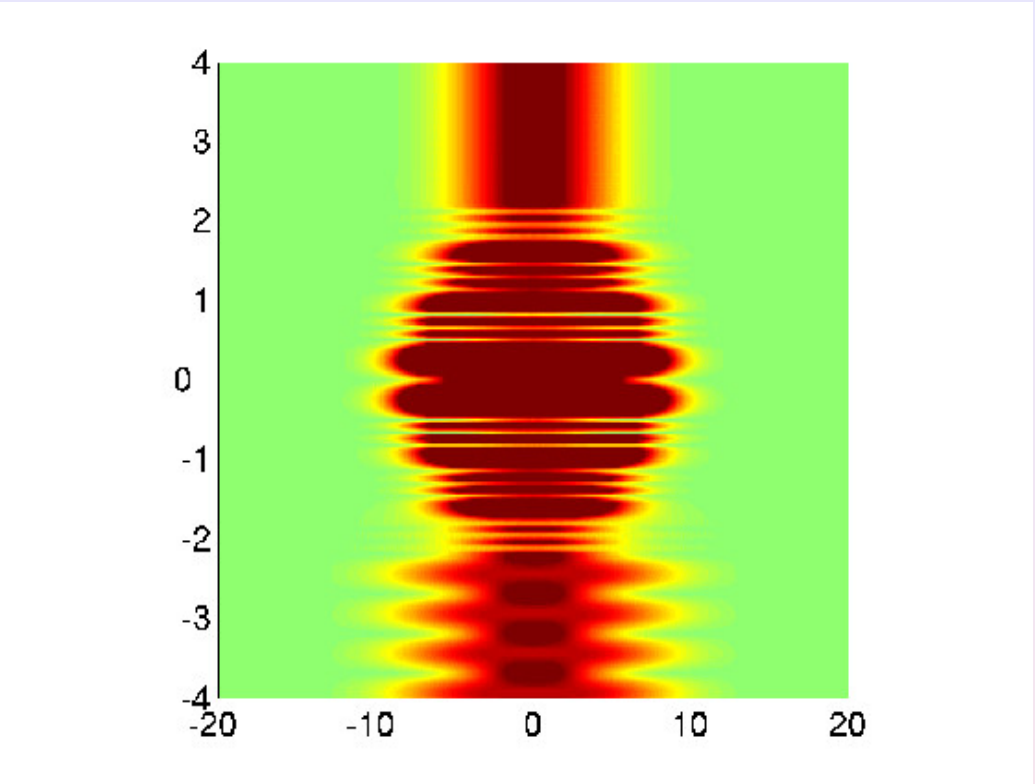



On the right, transmission coefficient versus frequency

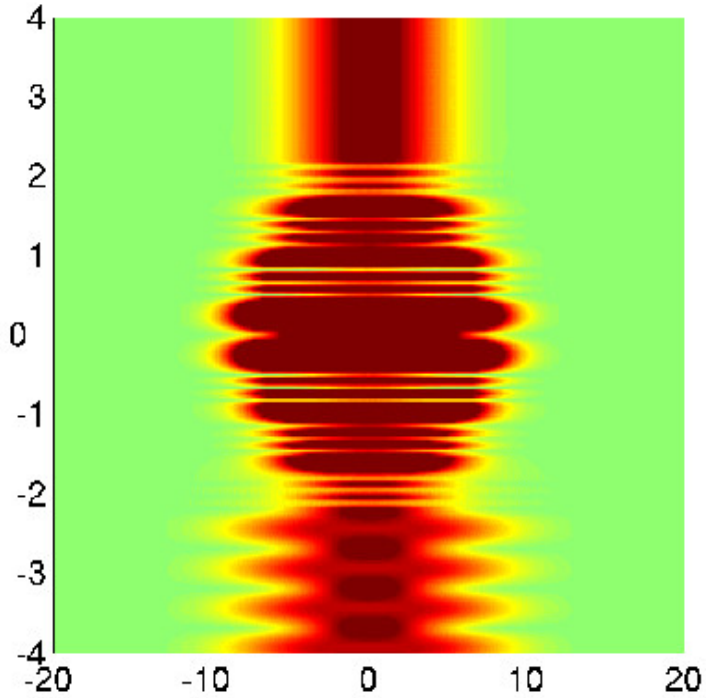
- Frequency F = 1.0 is a resonant frequency of the device
- Enlightment of the device by a gaussian beam.
- PML around the computational domain.

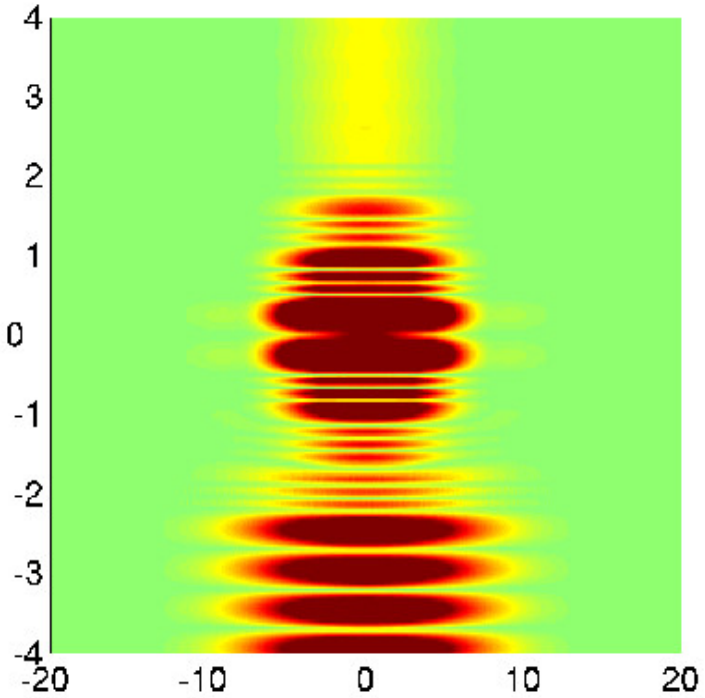
# Advantage to use high order method




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

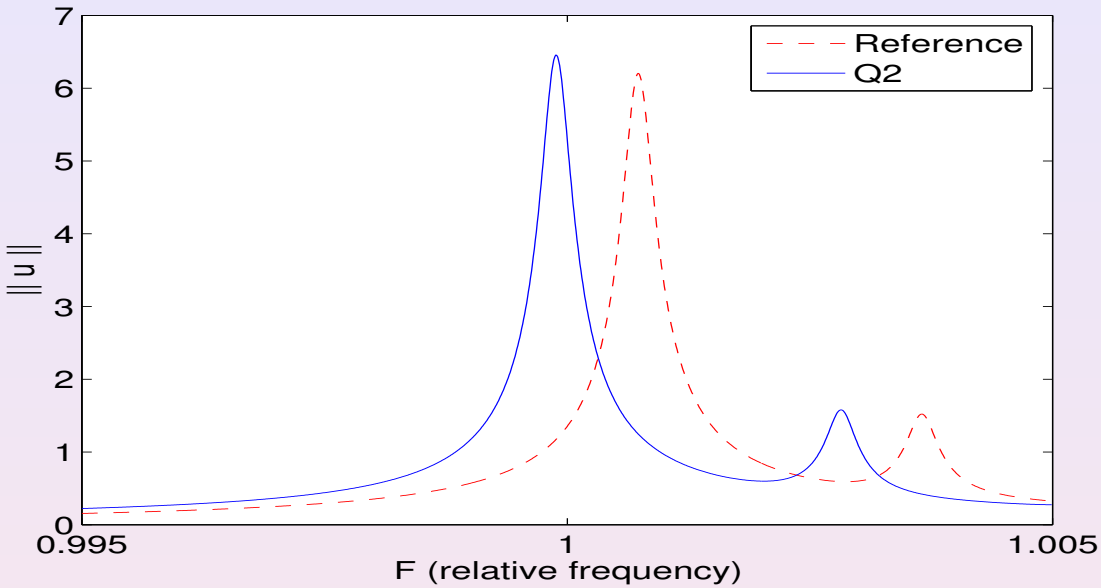
# Advantage to use high order method



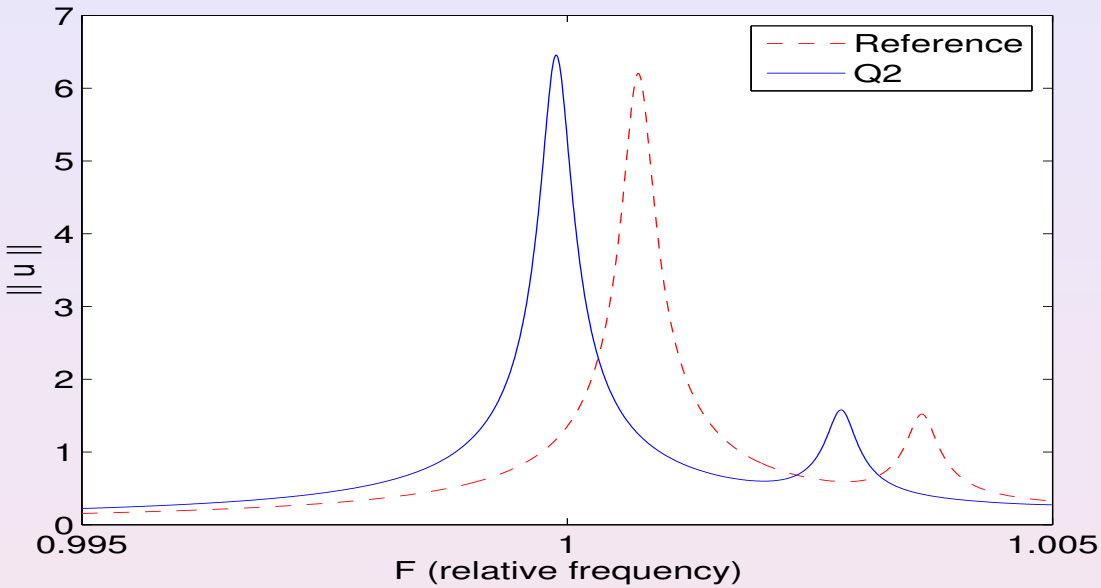

On the right, numerical solution for $\mathbf{Q_2}$ with 10 points by wavelength

# Advantage to use high order method




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

# Advantage to use high order method




Norm of the solution on 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

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

# 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 :

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

# Helmholtz equation

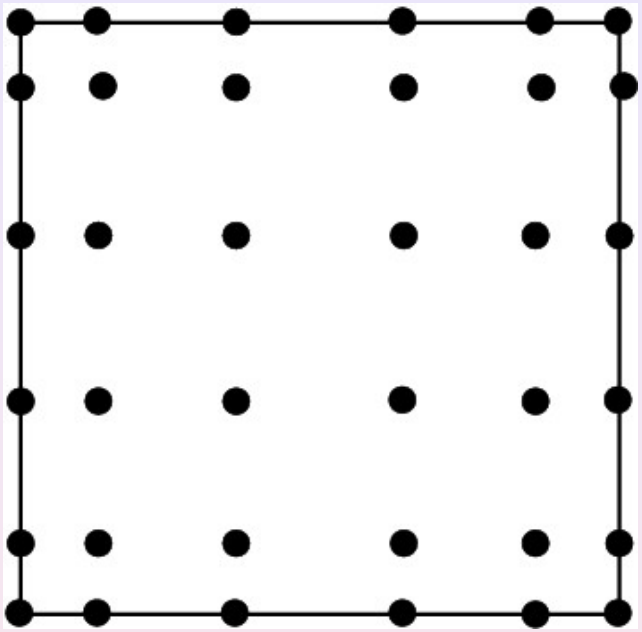
$$-\rho\,\omega^2\,u\;-\;\mathrm{div}(\mu\,\nabla u)\;=\;f\quad\in\Omega$$

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

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

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

# Elementary matrices



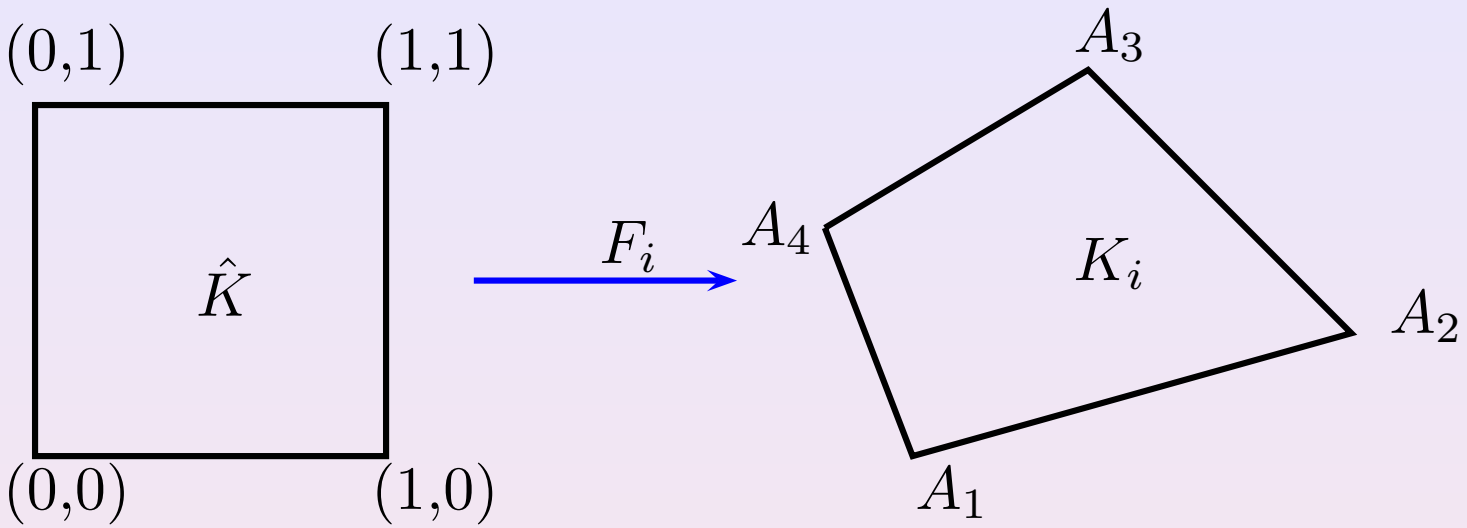
Gauss-Lobatto points for $Q_5$

on the unit square $\hat{K}$

Tensorized basis functions :

$$\hat{\varphi}_i = \hat{\varphi}_{i_1}(\hat{x})\,\hat{\varphi}_{i_2}(\hat{y})$$

# Elementary matrices




The transformation $F_i$

$DF_i$, $J_i$ jacobian matrix and determinant

# Elementary matrices

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

Use of quadrature formulas $(\omega_m^X, \xi_m^X)$ on the unit square/cube

- $X$ can be equal to $GL$ (Gauss-Lobatto quadrature, faster)
- $X$ can be equal to $G$ (Gauss quadrature, more accurate)

# Elementary matrices

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

Use of quadrature formulas $(\omega_m^X, \xi_m^X)$ on the unit square/cube

- $X$ can be equal to $GL$ (Gauss-Lobatto quadrature, faster)
- $X$ can be equal to $G$ (Gauss quadrature, more accurate)

$$(D_h)_{i,j} = \sum_m \omega_m^X \, \rho J_i \, \hat{\varphi}_i^{GL}(\xi_m^X) \, \hat{\varphi}_j^{GL}(\xi_m^X) \, d\hat{x}$$

Matrix-vector product $D_h U$ can be split into three steps :

$$v_m = \sum_j \hat{\varphi}_j^{GL}(\xi_m^X) u_j$$

$$w_m = \omega_m \rho J_i(\xi_m) v_m$$

$$y_i = \sum_m \hat{\varphi}_i^{GL}(\xi_m^X) w_m$$

# Elementary matrices

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

Use of quadrature formulas $(\omega_m^X, \xi_m^X)$ on the unit square/cube

- $X$ can be equal to $GL$ (Gauss-Lobatto quadrature, faster)
- $X$ can be equal to $G$ (Gauss quadrature, more accurate)

$$(D_h)_{i,j} = \sum_m \omega_m^X \, \rho J_i \, \hat{\varphi}_i^{GL}(\xi_m^X) \, \hat{\varphi}_j^{GL}(\xi_m^X) \, d\hat{x}$$

Underlying factorization

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

$$(A_h)_m = \omega_m \rho J_i(\xi_m)$$

$$D_h = \hat{C} A_h \hat{C}^*$$

$\Rightarrow$ only storage of $\omega_m \rho J_i(\xi_m)$

# Elementary matrices

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

Use of quadrature formulas $(\omega_m^X, \xi_m^X)$ on the unit square/cube

- $X$ can be equal to $GL$ (Gauss-Lobatto quadrature, faster)
- $X$ can be equal to $G$ (Gauss quadrature, more accurate)

Product $Y = \hat{C}U$ is split into three steps :

$$v_{i_1,j_2,j_3} = \sum_{j_1} \hat{\varphi}_{j_1}^{GL}(\xi_{i_1}^X) u_{j_1,j_2,j_3}$$

$$w_{i_1,i_2,j_3} = \sum_{j_2} \hat{\varphi}_{j_2}^{GL}(\xi_{i_2}^X) v_{i_1,j_2,j_3}$$

$$y_{i_1,i_2,i_3} = \sum_{j_3} \hat{\varphi}_{j_3}^{GL}(\xi_{i_3}^X) w_{i_1,i_2,j_3}$$

# Fast matrix vector product with any points

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

# Fast matrix vector product with any points

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

Matrix-vector product $K_h U$ can be split into three steps

$$v_m = \sum_j \hat{\nabla}\hat{\varphi}_j^{GL}(\xi_m^X) u_j$$

$$w_m = \omega_m J_i \, DF_i^{-1} \mu DF_i^{*-1} v_m$$

$$y_i = \sum_q \hat{\nabla}\hat{\varphi}_i^{GL}(\xi_m^X) w_m$$

# Fast matrix vector product with any points

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

Underlying factorization

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

$$(B_h)_m = \omega_m J_i \, DF_i^{-1} \mu DF_i^{*-1}$$

$$K_h = \hat{S} B_h \hat{S}^*$$

$\Rightarrow$ only storage of $J_i \, DF_i^{-1} \mu DF_i^{*-1}$

# Fast matrix vector product with any points

By using the matrices

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

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

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

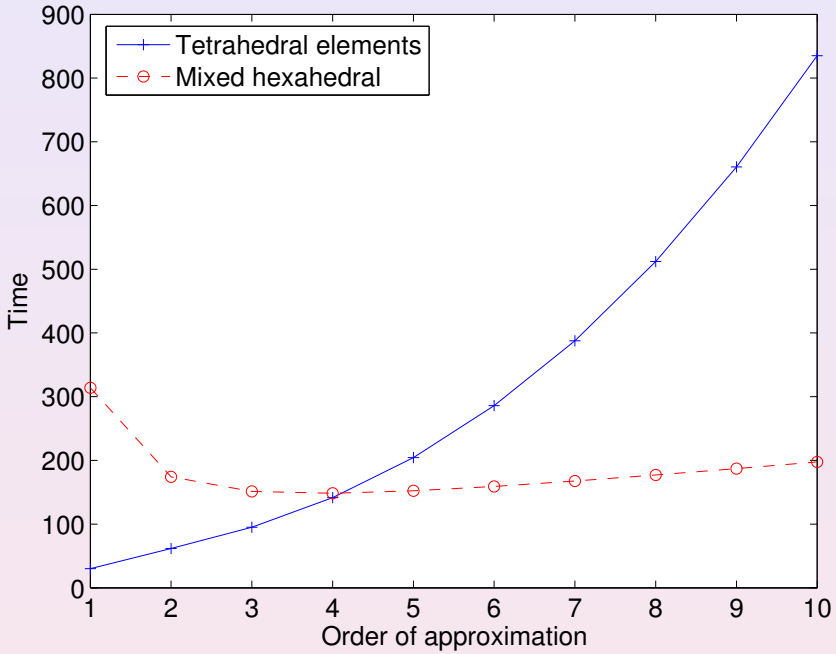
we have $\hat{S} = \hat{R}\hat{C}$
final matrix : $\hat{C}(-\omega^2 A_h + \hat{R}B_h\hat{R}^*)\hat{C}^*$

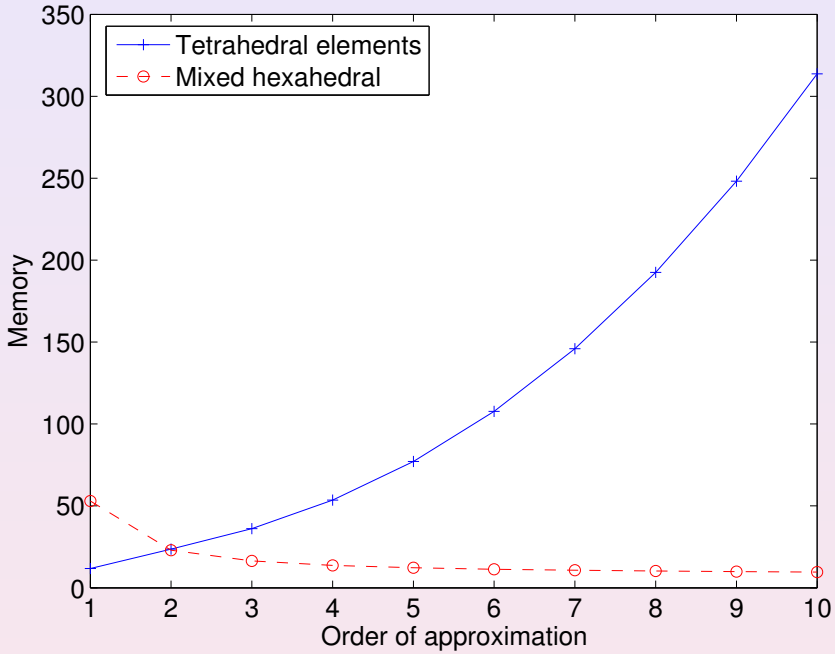
# Fast matrix vector product with any points

$r$ is the order of approximation
For hexahedral elements (tensorization), complexity in $O(r^4)$
For tetrahedral elements (no tensorization), complexity in $O(r^6)$

- If we use Gauss-Lobatto points to integrate : $\hat{C} = I$
- Same storage for Gauss or GL points
- Matrix-vector product slower with Gauss integration
- Loss of one order for GL points in 3-D

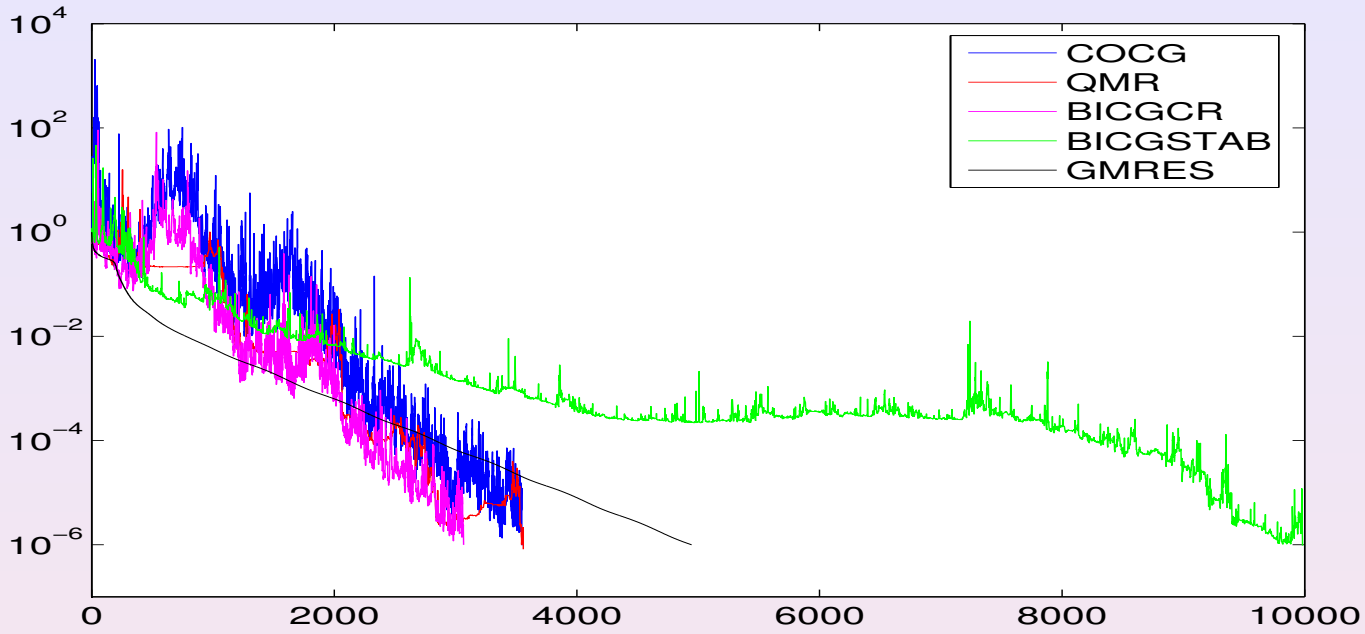
# Matrix vector product faster than with tetrahedral ?




Comparison between hexahedral and tetrahedral elements, for computational time (left plot) and storage (right plot)

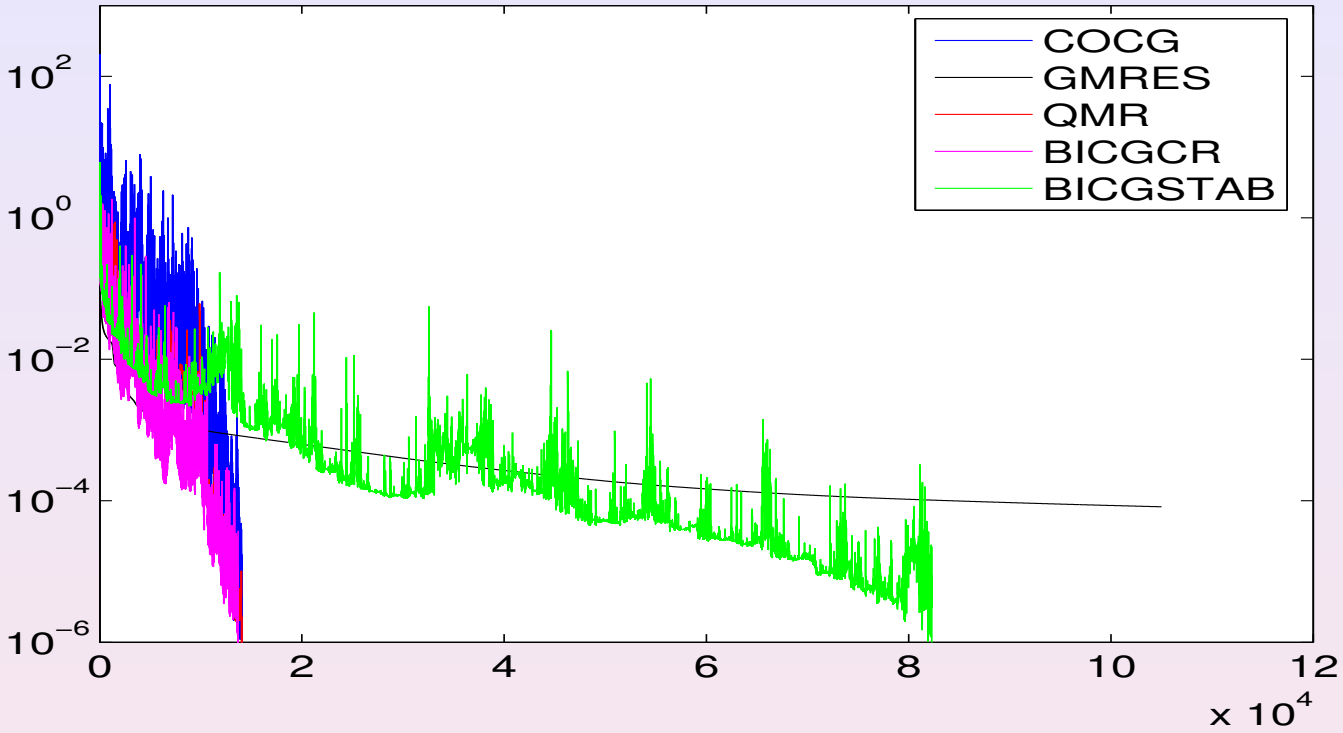
# Iterative methods used




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

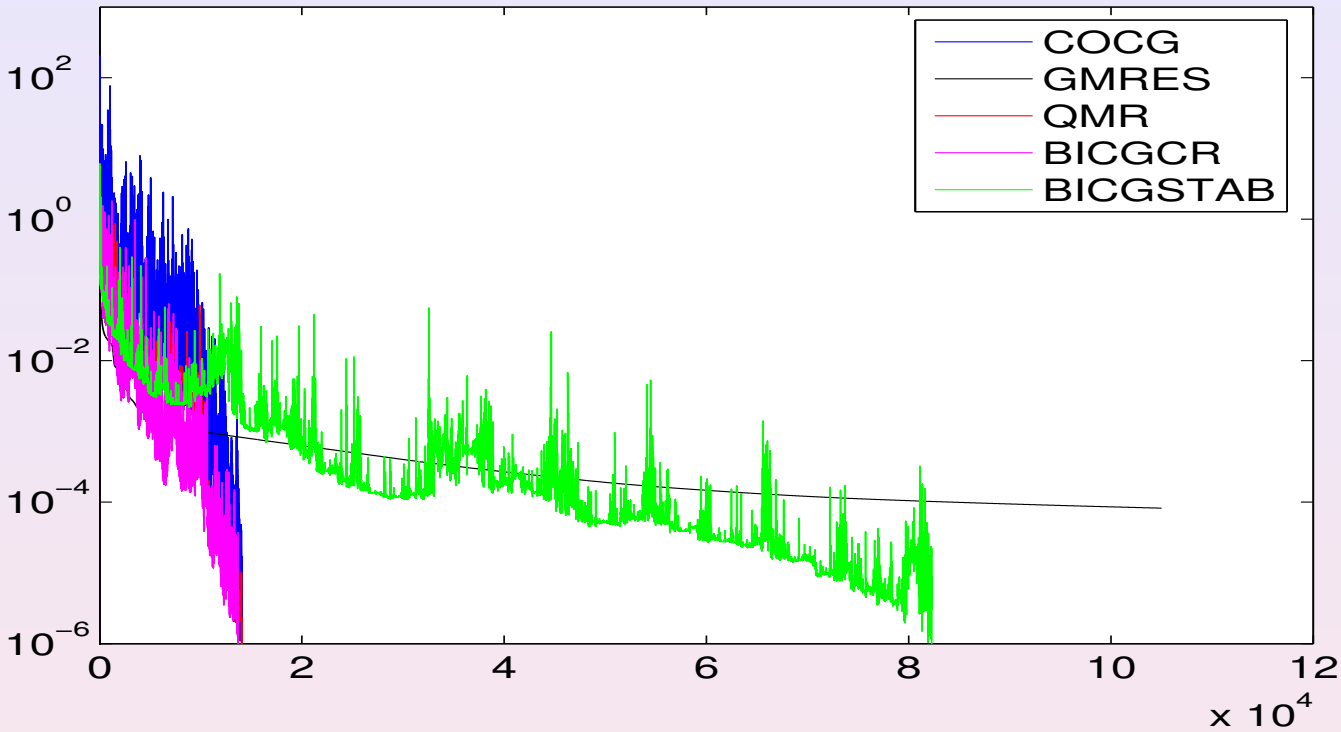
- GMRES, BICGSTAB and QMR for complex unsymmetric matrices
- COCG, BICGCR for complex symmetric matrices

# 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 (faster) or COCG (less storage)
- Need of preconditioning techniques to have less iterations

# Preconditioning used

- Incomplete factorization with threshold on the damped Helmholtz equation :

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

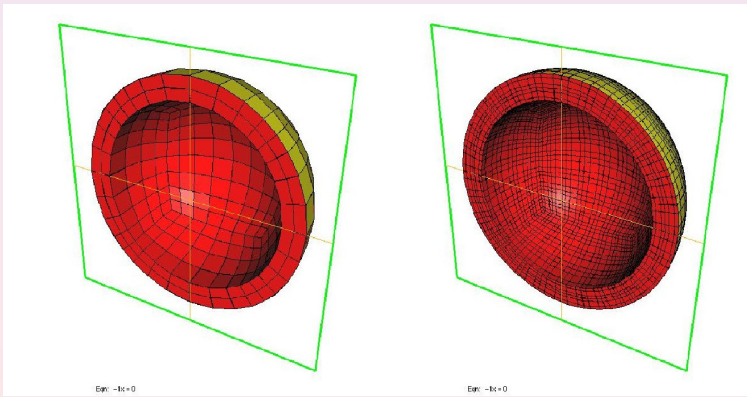
  - see Y. Saad, Iterative methods for sparse linear systems

# 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



On the left, initial mesh $Q_3$, on the 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, (Phd at Delft)

# Preconditioning used

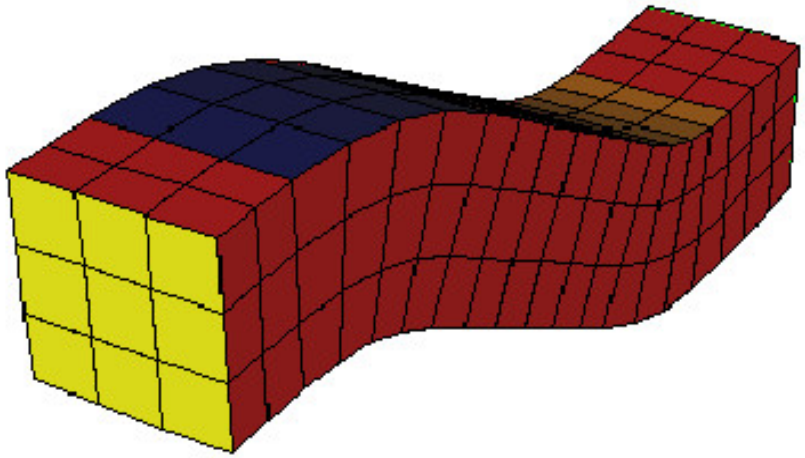
- Incomplete factorization with threshold on the damped Helmholtz equation :

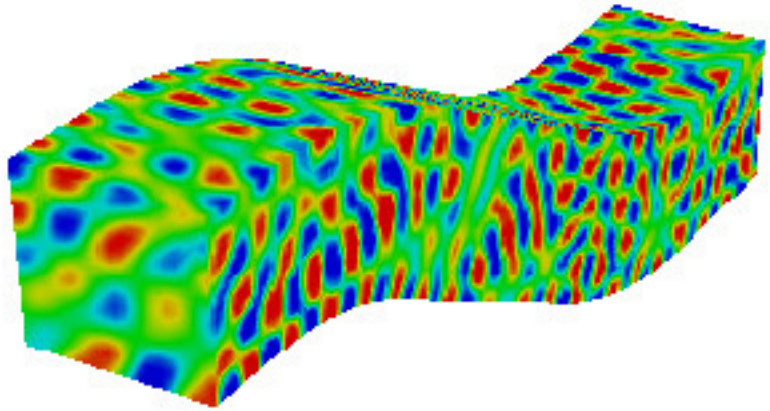
$$-k^2(\alpha + i\beta)u - \Delta u = 0$$

  - see Y. Saad, Iterative methods for sparse linear systems
- Multigrid method on the damped Helmholtz equation
  - see Y. A. Erlangga and al, (Phd at Delft)

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

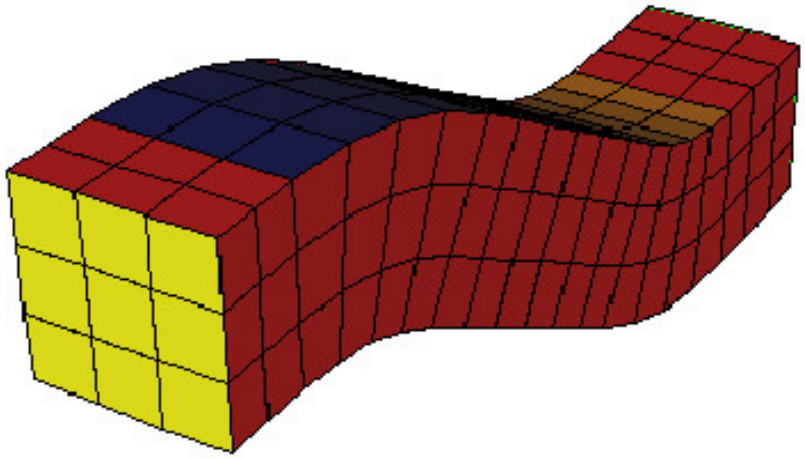
# Scattering by a cobra cavity

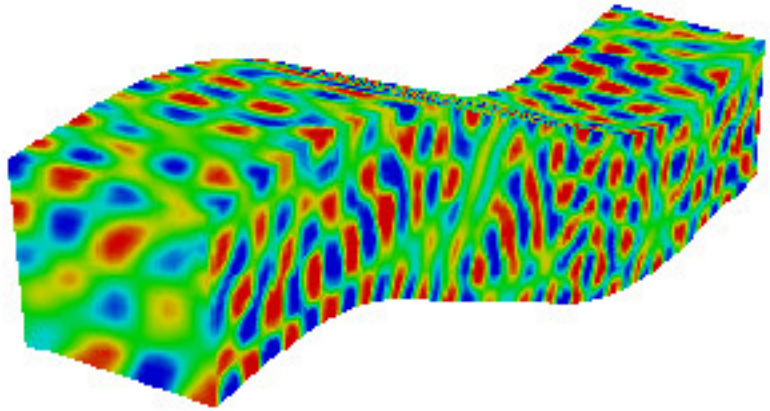




- Cobra cavity of length 20, and depth 4
- First order absorbing boundary condition on the yellow face

# Scattering by a cobra cavity

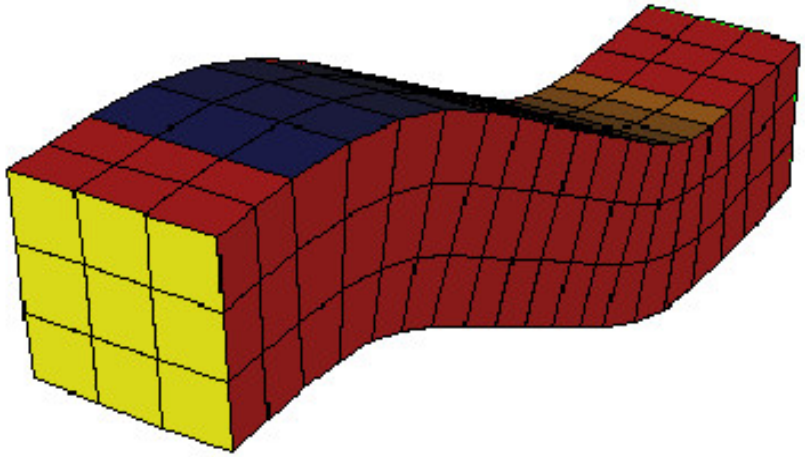


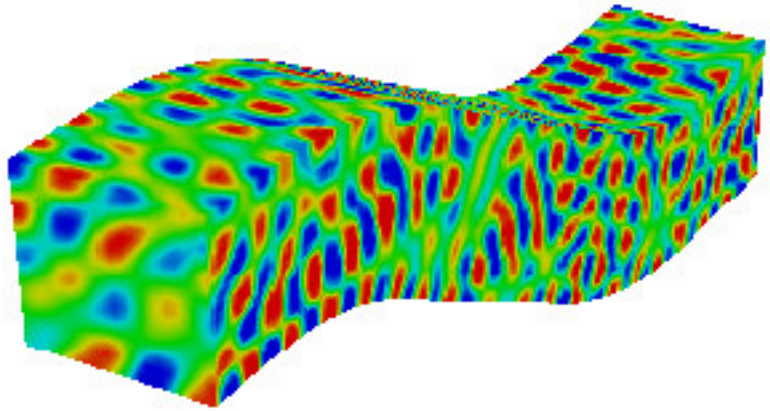


Number of dofs to reach less than 5 % $L^2$ error

| 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 |

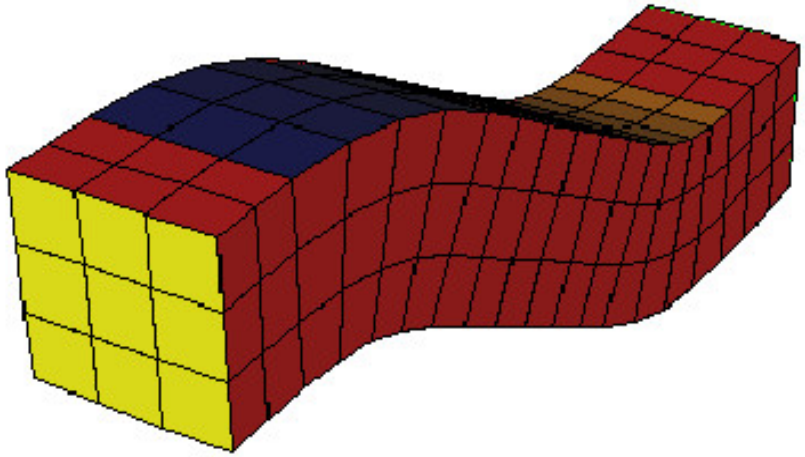
# Scattering by a cobra cavity

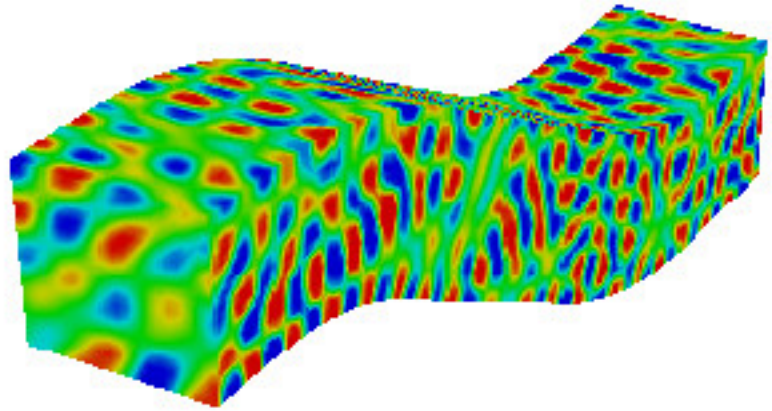




| Finite element | structured $\mathbf{Q_8}$ | non-structured $\mathbf{Q_6}$ | non-structured $\mathbf{P_4}$ |
|---|---|---|---|
| No preconditioning | 9 860 s | NC | NC |
| ILUT(0.01) | 1 021 s | 13 766 s | 8 036 s |
| Two-grid | 1 082 s | 6 821 s | 14 016 s |

# Scattering by a cobra cavity

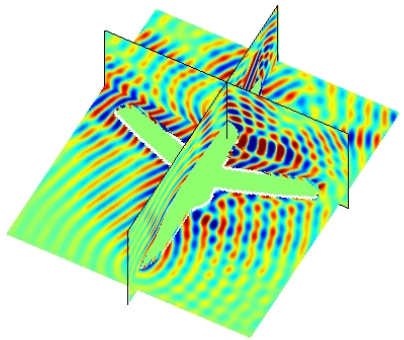


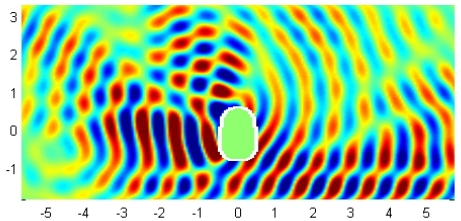


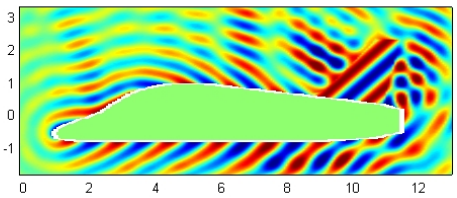
| Finite element | structured $Q_8$ | non-structured $Q_6$ | non-structured $P_4$ |
|---|---|---|---|
| No preconditioning | 9 860 s | NC | NC |
| ILUT(0.01) | 1 021 s | 13 766 s | 8 036 s |
| Two-grid | 1 082 s | 6 821 s | 14 016 s |

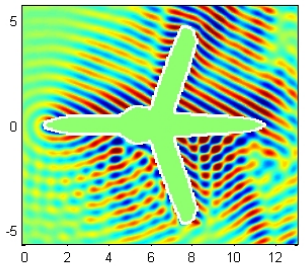
| Finite element | structured $Q_8$ | non-structured $Q_6$ | non-structured $P_4$ |
|---|---|---|---|
| No preconditioning | 32 Mo | 162 Mo | 251 Mo |
| ILUT(0.01) | 150 Mo | 1 250 Mo | 1 400 Mo |
| Two-grid | 60 Mo | 283 Mo | 710 Mo |

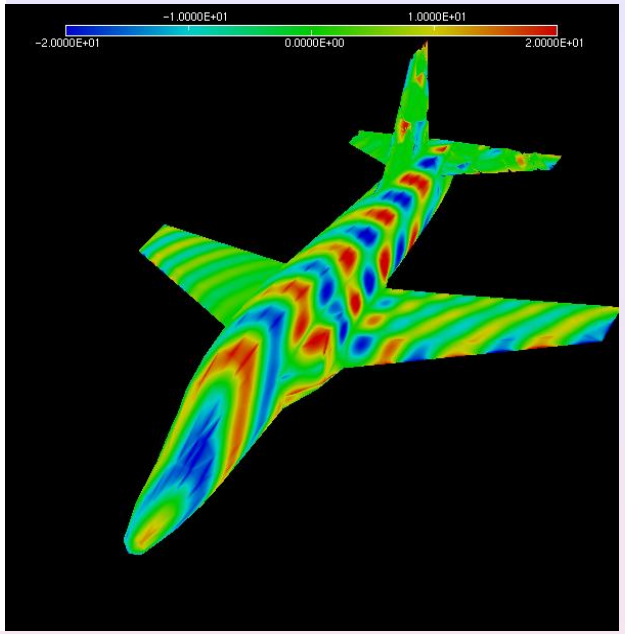
# Scattering by a plane





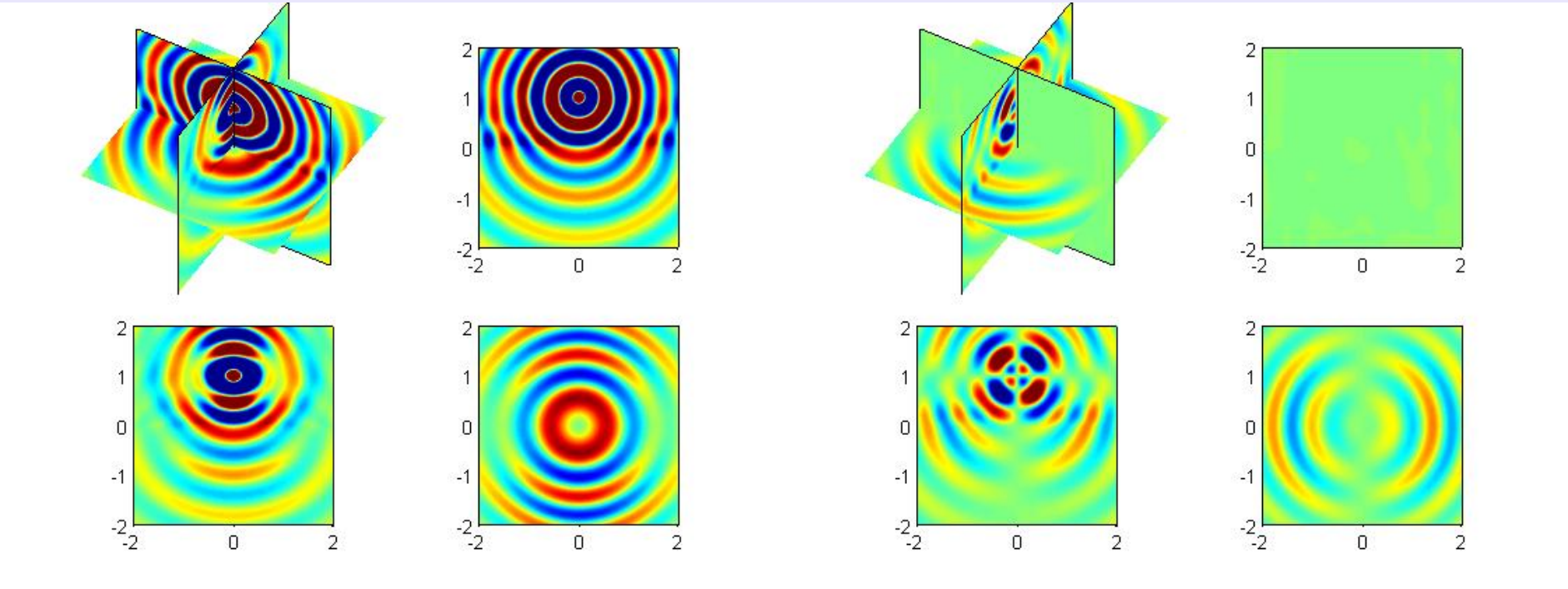




- Real part of the diffracted for an oblique incident plane wave
- Q4, 7.2 million of dofs (2 Go memory)
- 280 iterations and 2 hours with multigrid preconditioning
- More than 20 000 iterations without preconditioning

# Time-harmonic elastic equation



- Displacement components $u_x$ and $u_z$
- Q7, 4.3 million of dofs (with PML layers)
- 170 iterations and 50mn with multigrid preconditioning
- More than 20 000 iterations without preconditioning

# Why do we choose $Q_r$ for hexahedra ?

- $P_r$ optimal finite element space for tetrahedra
- Is $Q_r$ the right choice for hexahedra ? Why not choosing $P_r$ (with DG method for instance) ?
- If transformation $F_i$ is affine (i.e. paralleloid ), $P_r$ is the optimal finite element space
- If transformation $F_i$ is trilinear (general hexahedron ), $Q_r$ is the optimal finite element space

# Why do we choose $Q_r$ for hexahedra ?

- $P_r$ optimal finite element space for tetrahedra
- Is $Q_r$ the right choice for hexahedra ? Why not choosing $P_r$ (with DG method for instance) ?

- If transformation $F_i$ is affine (i.e. paralleloid ), $P_r$ is the optimal finite element space
- If transformation $F_i$ is trilinear (general hexahedron ), $Q_r$ is the optimal finite element space

# Condition of optimality

We define the finite element space with real element $K_i$ :

$$V_h = \{u \in H^1(\Omega) \text{ such that } u|_{K_i} \in V_F^r\}$$

$V_F^r$ : finite element space for the real element

We define the finite element space with reference element $\hat{K}$ :

$$V_h = \{u \in H^1(\Omega) \text{ such that } u|_{K_i} \circ F_i \in \hat{V}^r\}$$

$\hat{V}^r$ : finite element space for the reference element

Condition of optimality :

$$V_F^r \supset P_r$$

For hexahedra, we can prove :

$$V_F^r \supset P_r \Leftrightarrow \hat{V}^r \supset Q_r$$

# Idea of the reason of optimality of $Q_r$

We consider a monomial of $P_r$ :

$$x^m y^n z^p, \qquad m+n+p \leq r$$

and we write its expression on reference element with transformation $F_i$ :

$$x = a_1 + a_2\hat{x} + a_3\hat{y} + a_4\hat{z} + a_5\hat{x}\hat{z} + a_6\hat{y}\hat{z} + a_7\hat{x}\hat{y} + a_8\hat{x}\hat{y}\hat{z}$$

$$y = b_1 + b_2\hat{x} + b_3\hat{y} + b_4\hat{z} + b_5\hat{x}\hat{z} + b_6\hat{y}\hat{z} + b_7\hat{x}\hat{y} + b_8\hat{x}\hat{y}\hat{z}$$

$$z = c_1 + c_2\hat{x} + c_3\hat{y} + c_4\hat{z} + c_5\hat{x}z + c_6\hat{y}\hat{z} + c_7\hat{x}\hat{y} + c_8\hat{x}\hat{y}\hat{z}$$

By expanding $x^m y^n z^p$, the higher-degree term is equal to :

$$a_8^m b_8^n c_8^p \hat{x}^{m+n+p} \hat{y}^{m+n+p} \hat{z}^{m+n+p}$$

Hence we have obtained for $m+n+p = r$ the higher-degree term of $Q_r$.

# Influence of quadrature

Then, thanks to Bramble-Hilbert lemma, it is easy to get :

$$||u - u_h||_1 \leq C h^r ||u||_{r+1}$$

However, in our case, integrals are not evaluated exactly, we have to consider Strang Lemma :

$$||u - u_h||_1 \leq C \inf_{v_h \in V_h} \{ ||u - v_h||_1 + \sup_{w_h \in V_h} \frac{|a(v_h, w_h) - a_h(v_h, w_h)|}{||w_h||_1} \}$$

Here we choose $v_h = \Pi_r u$ with the projector on real space, so that

$$v_h \in P_r \Rightarrow \hat{v}_h \in Q_r$$

$$\nabla v_h \in P_{r-1} \Rightarrow \widehat{\nabla v_h} \in Q_{r-1}$$

# Quadrature error

Thus, we are able to prove that :

$$(D - D_h)(v_h, w_h) = (D - D_h)(v_h - \pi_{r-1} v_h, w_h - \pi_0 w_h)$$

for Gauss rules, since $J_i \in Q_2$ (so $J_i \, \pi_{r-1} v_h \, w_h \in Q_{2r+1}$)

$$\Leftrightarrow |(D - D_h)(v_h, w_h)| \leq C h^{r+1} ||v_h||_r ||w_h||_1$$

for Gauss-Lobatto rules

$$(D - D_h)(v_h, w_h) = (D - D_h)(v_h - \pi_{r-3} v_h, w_h - \pi_0 w_h)$$

$$\Leftrightarrow |(D - D_h)(v_h, w_h)| \leq C h^{r-1} ||v_h||_{r-2} ||w_h||_1$$

Therefore, we have a loss of order for Gauss-Lobatto rules

# Quadrature error

For stiffness term, we have :

$$(K - K_h)(v_h, w_h) = (D - D_h)(\nabla v_h - \pi_r \nabla v_h, \nabla w_h)$$

for Gauss rules, since $J_i DF_i^{*-1} \hat{\nabla} \hat{w} h \in Q_{r+1}$

$$\Leftrightarrow |(K - K_h)(v_h, w_h)| \leq C h^{r+1} ||v_h||_{r+1} ||w_h||_1$$

for Gauss-Lobatto rules

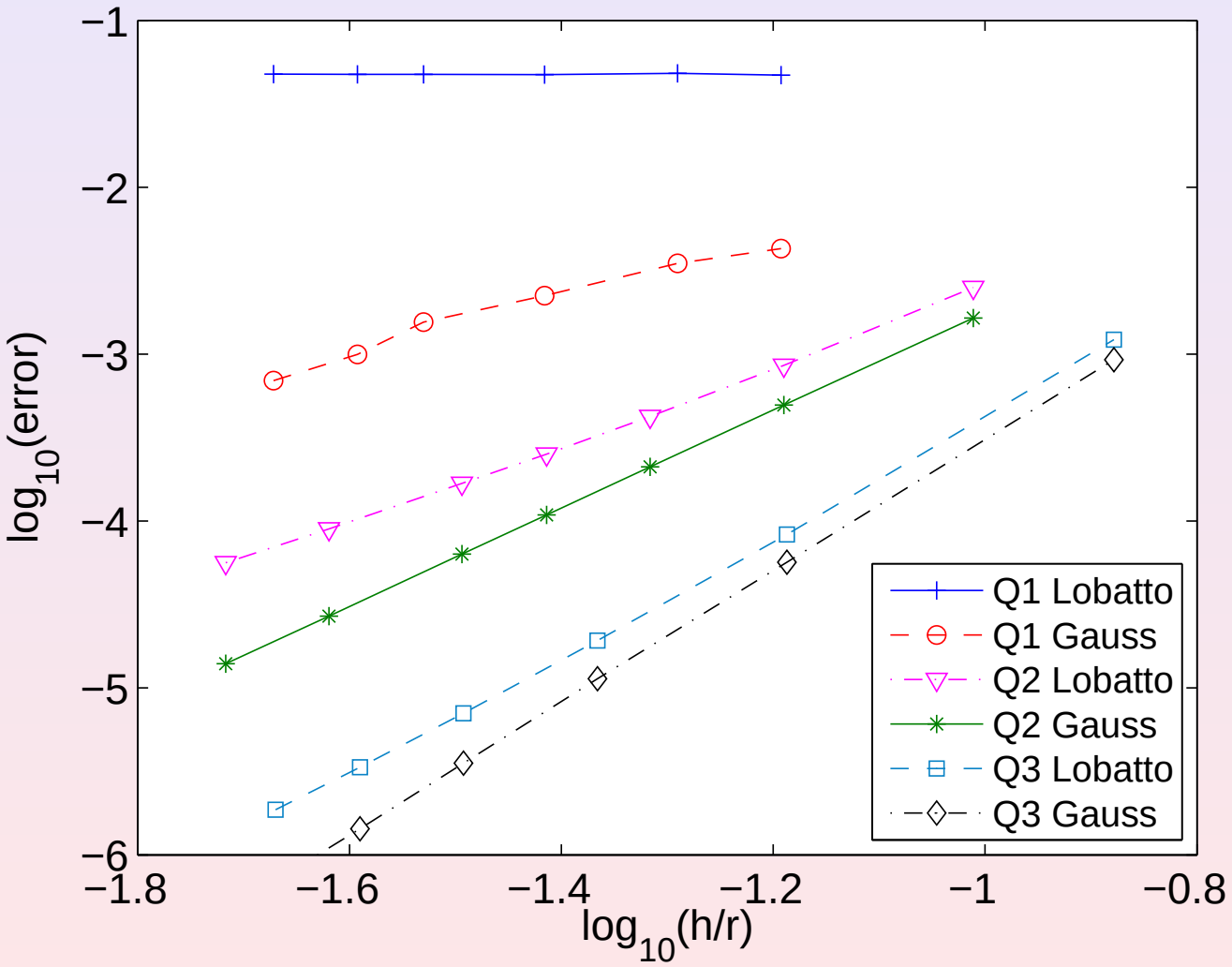
$$(K - K_h)(v_h, w_h) = (D - D_h)(\nabla v_h - \pi_{r-2} \nabla v_h, \nabla w_h)$$

$$\Leftrightarrow |(D - D_h)(v_h, w_h)| \leq C h^{r-1} ||v_h||_{r-1} ||w_h||_1$$

Therefore, we have a loss of order for Gauss-Lobatto rules

# Quadrature error

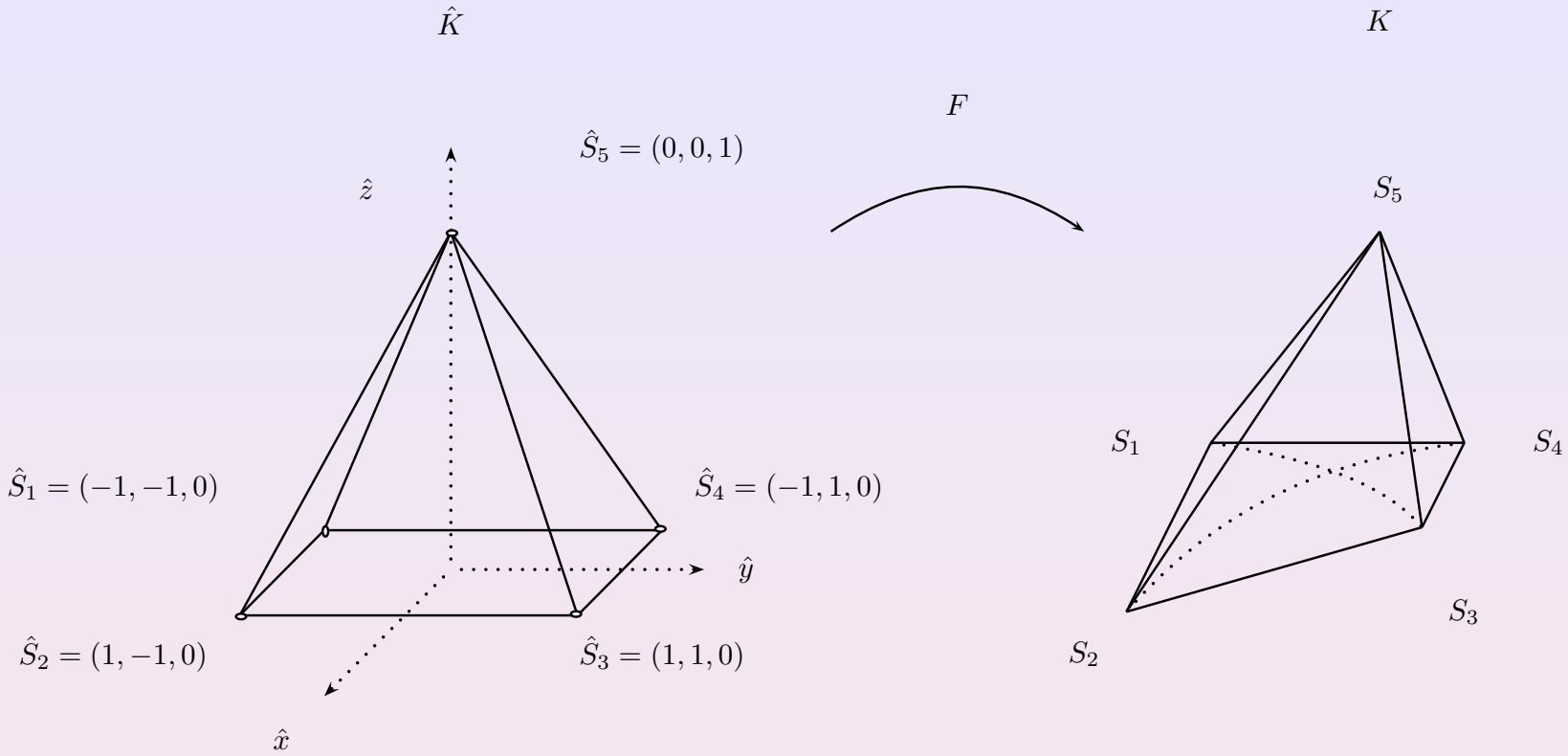
Illustration of this effect for a cube meshed with split tetrahedra



# Why pyramids ?

- Automatic generation of high-quality hexahedral meshes is difficult
- “Solution of split tetrahedra” is not interesting
- Some mesh tools are able to produce meshes with a high ratio of hexahedra and some remaining pyramids/tets/prisms.
- Pyramids elements not as well known as other elements.

# Two main approaches




Simplest expression of $F_i$ (Bedrosian) :

$$F_i(\hat{x},\hat{y},\hat{z}) = A + B\hat{x} + C\hat{y} + D\hat{z} + \frac{\hat{x}\hat{y}}{4(1-\hat{z})}(S_1 + S_3 - S_2 - S_4)$$

# Two main approaches

- Use of rational fractions to define $F_i$
  - Early work of Bedrosian with explicit first and second order basis functions
  - Work of Sherwin, Karniadakis, Warburton : h-p Basis functions obtained by considering a degenerated cube (coincidence with Bedrosian functions for r = 1)
  - Recent work of Nigam, Phillips with a reference infinite pyramid (but same basis functions as Bedrosian for r = 1)
- Use of piecewise polynomial to define $F_i$ (polynomial on each sub-tetrahedron)
  - Work of Wieners, with first and second order basis functions
  - Work of Knabner and Summ, with an analysis of this transformation
  - Work of Bluck and Walker, with a proposition of high order basis functions

# Optimal finite element space

Same approach than for hexahedra : We consider a monomial of $P_r$ :

$$x^m, \qquad m \leq r$$

$$(a + b\hat{x} + c\hat{y} + d\hat{z} + \alpha(\frac{\hat{x}\hat{y}}{1-\hat{z}}))^m$$

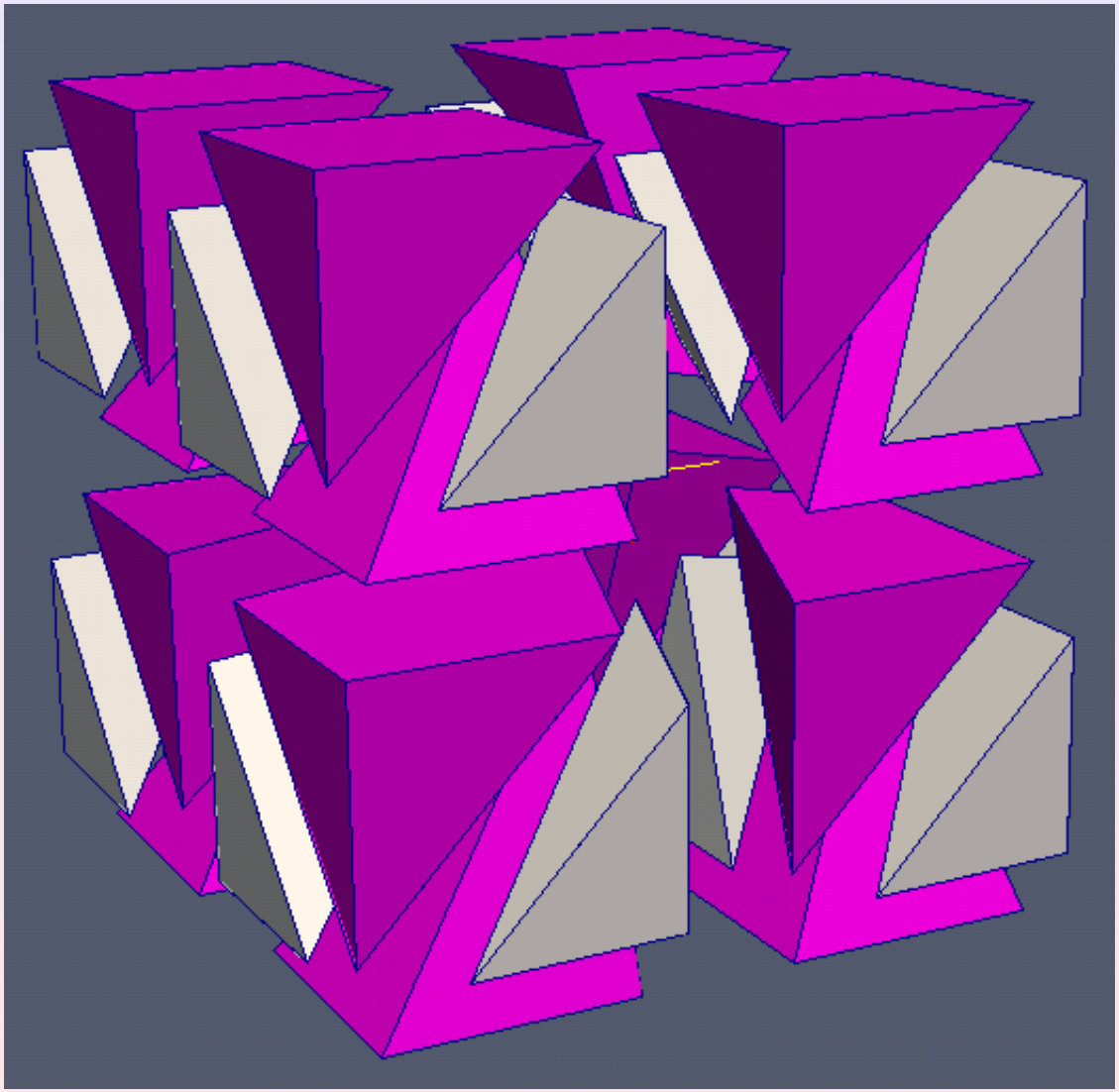
$$\sum_k C_m^k (a + b\hat{x} + c\hat{y})^k (d\hat{z})^k \alpha^{m-k} (\frac{\hat{x}\hat{y}}{1-\hat{z}})^{m-k}$$

After some calculations, you can show that the optimal finite element space is

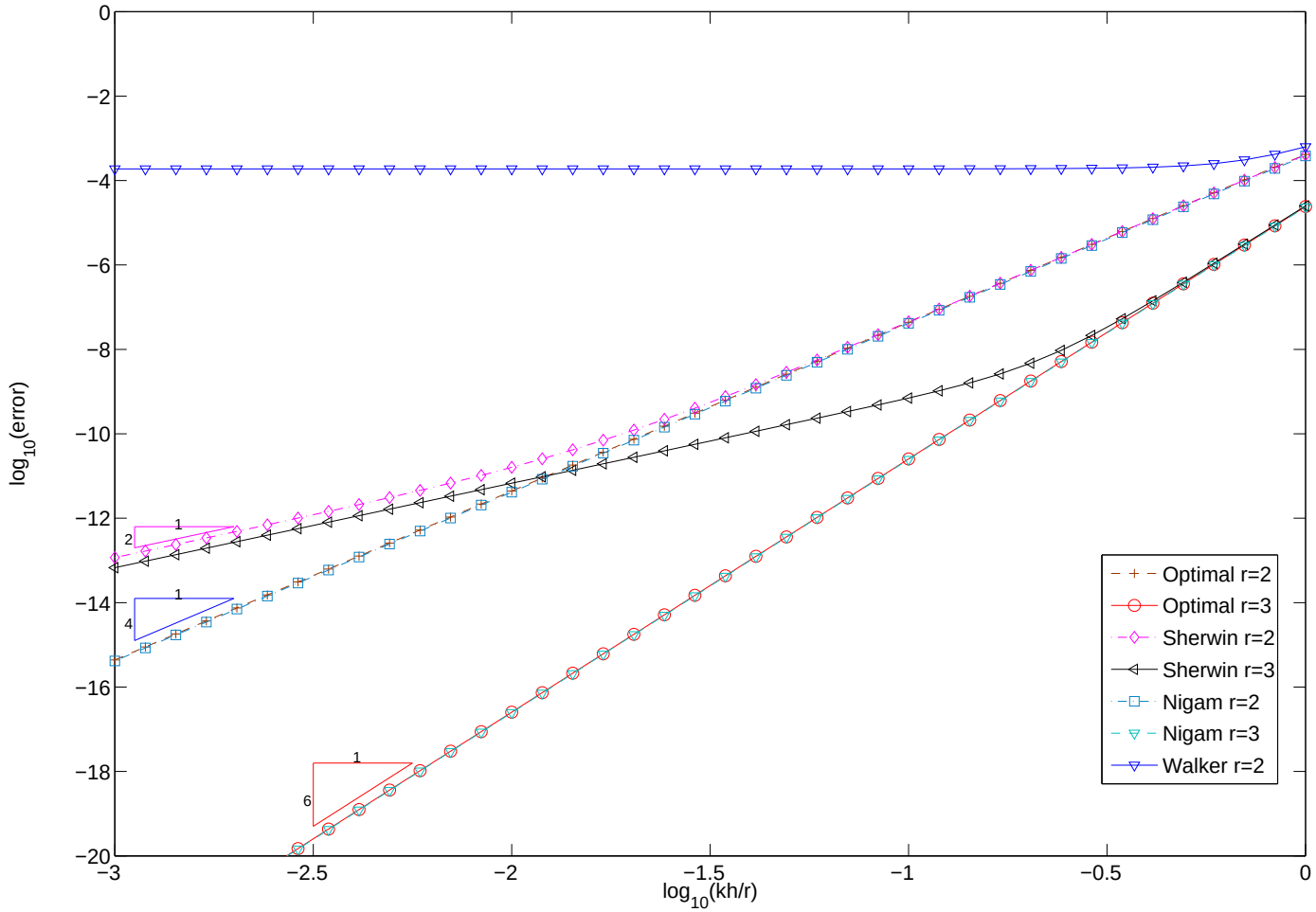
$$\hat{V}^r = P_r \oplus \sum_{k=0}^{r-1} (\frac{\hat{x}\hat{y}}{1-\hat{z}})^{r-k} P_k(\hat{x}, \hat{y})$$

# Numerical comparison between different methods

We perform a dispersion analysis on the following hybrid mesh :



# Numerical comparison between different methods



# Numerical comparison between different methods

- We obtained same finite element space as Demkowicz/Zaglmayr
- We obtained a smaller finite element space than Nigam/Phillips
- We proposed modifications of basis functions of Sherwin/Karniadakis/Warburton so that they span the optimal finite element space
- Alternative approach using piecewise polynomial (by splitting pyramid in two or four tets) is not consistent for non-affine pyramids

# Influence of quadrature

Then, thanks to Bramble-Hilbert lemma, it is easy to get :

$$||u - u_h||_1 \leq C h^r ||u||_{r+1}$$

However, in our case, integrals are not evaluated exactly, we have to consider Strang Lemma :

$$||u - u_h||_1 \leq C \inf_{v_h \in V_h} \{ ||u - v_h||_1 + \sup_{w_h \in V_h} \frac{a(v_h, w_h) - a_h(v_h, w_h)|}{||w_h||_1} \}$$

Here we choose $v_h = \Pi_r u$ with the projector on real space, so that

$$v_h \in P_r \Rightarrow \hat{v}_h \in \hat{V}^r$$

$$\nabla v_h \in P_{r-1} \Rightarrow \widehat{\nabla v_h} \in \hat{V}^{r-1}$$

# Quadrature error

By expressing integrals on the cube, we are able to prove that :

$$(D - D_h)(v_h, w_h) = (D - D_h)(v_h - \pi_r v_h, w_h - \pi_0 w_h)$$

for Gauss-Jacobi rules (because of $(1 - \hat{z})^2$ weight), since $J_i \in Q_1$ (so $J_i \, \pi_r v_h \, w_h \in Q_{2r+1}$)

$$\Leftrightarrow |(D - D_h)(v_h, w_h)| \leq C h^{r+2} ||v_h||_{r+1} ||w_h||_1$$

We could use Gauss-Jacobi-Lobatto rules for mass term without loss of accuracy

# Quadrature error

For stiffness term, we have :

$$(K - K_h)(v_h, w_h) = (D - D_h)(\nabla v_h - \pi_r \nabla v_h, \nabla w_h)$$

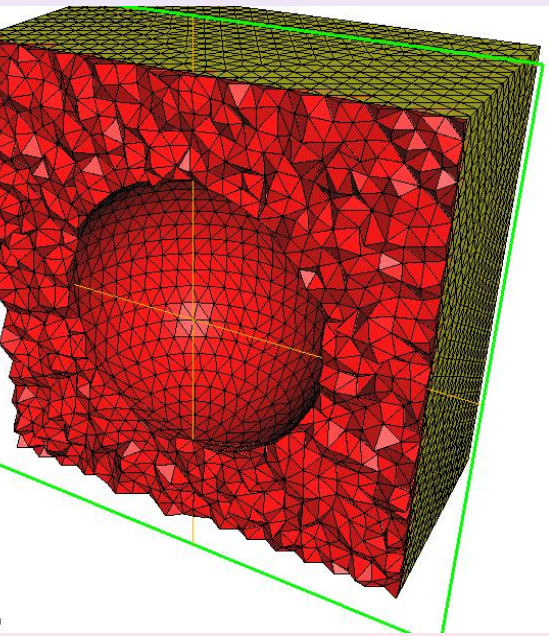
for Gauss rules, since $J_i DF_i^{*-1} \hat{\nabla} \hat{w} h \in Q_{r+1}$

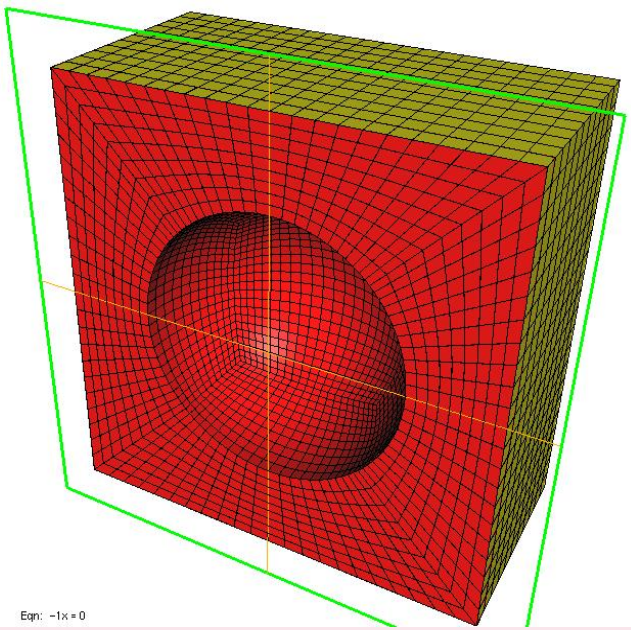
$$\Leftrightarrow |(K - K_h)(v_h, w_h)| \leq C h^{r+1} ||v_h||_{r+1} ||w_h||_1$$

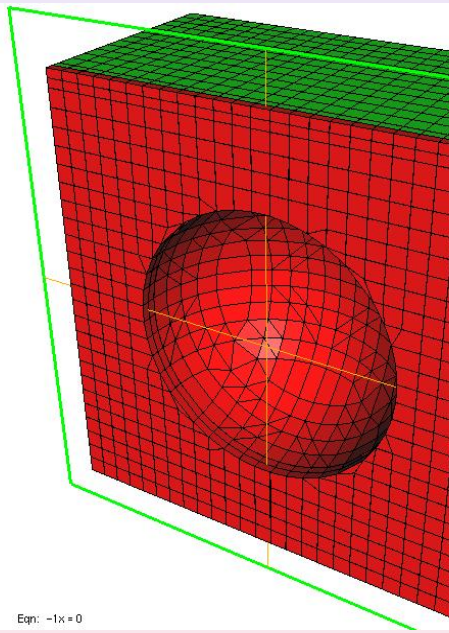
we can't use Gauss-Jacobi-Lobatto rules without loss of an order

# Numerical results for Helmholtz equation

Comparison of three kind of meshes :

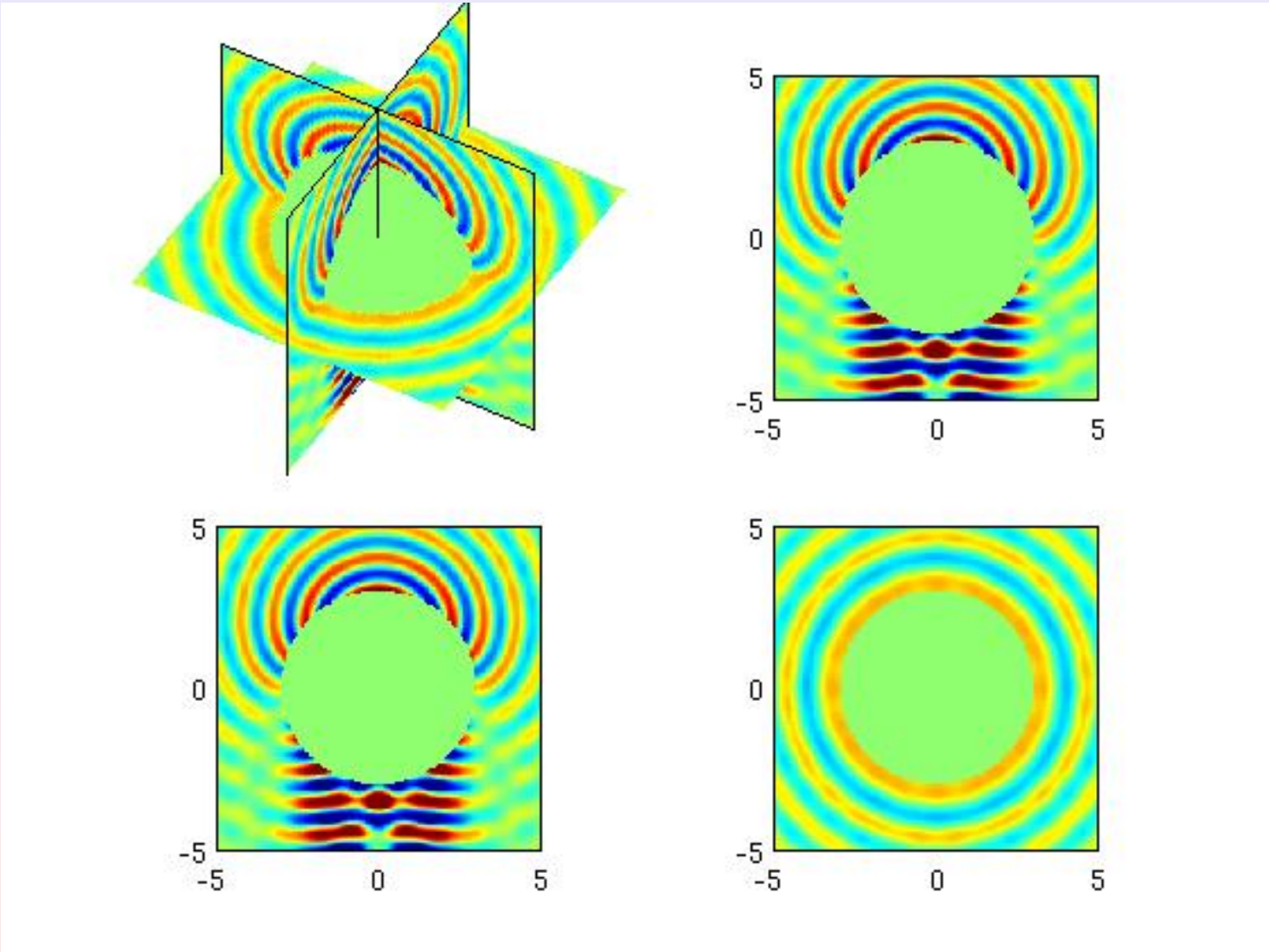




# Numerical results for Helmholtz equation

For the scattering of a sphere :



# Numerical results for Helmholtz equation

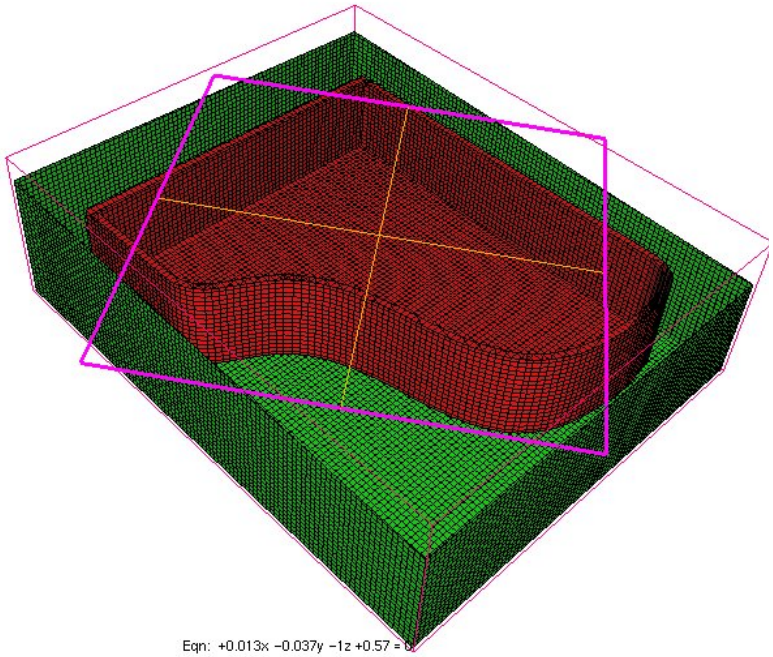
For order 4 for tets, and order 5 for hex/hybrid, less than 2 % error :

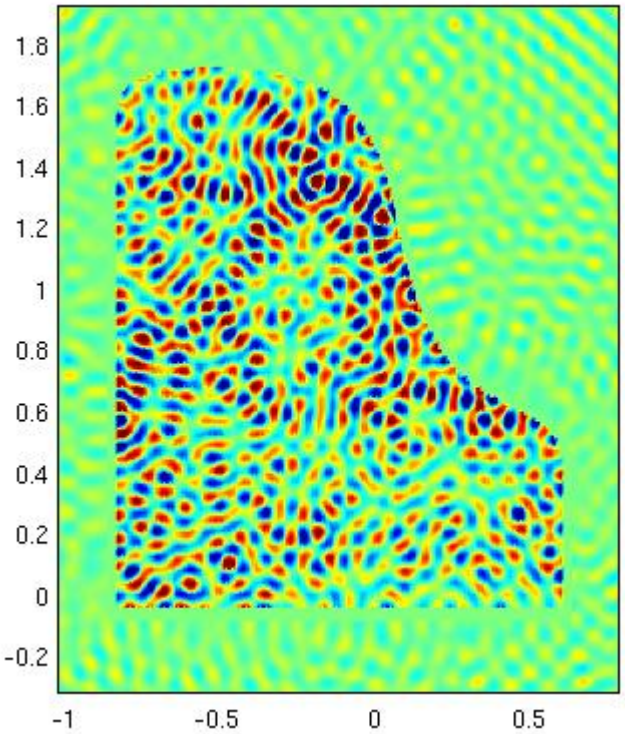
| Mesh type | Tetra | Hexa | Split Tetra | Hybrid |
|---|---|---|---|---|
| number of dofs | 339 000 | 315 000 | 520 000 | 266 000 |
| multigrid prec. | 119 iter (587s) | 130 iter (152s) | 93 iter (266s) | 128 iter (161s) |

# Numerical results for wave equation

Use of Discontinuous Galerkin formulation for solving wave equation :

$$\frac{\partial^2 u}{\partial t^2} - \Delta u = 0$$





# Numerical results for wave equation

Efficiency of different kind of meshes for the piano-shaped cavity with a third order approximation on a fine mesh :

| Mesh type | Tetra | Split Tetra | Hybrid |
|---|---|---|---|
| Obtained accuracy | 5.7% | 9.4 % | 6.3% |
| Number of dofs | 16.9 millions | 49.3 millions | 14.9 millions |
| Time step (freq = 14) | $\Delta t = 0.0004$ | $\Delta t = 0.0002$ | $\Delta t = 0.0005$ |
| Computational time | 4.3 days | 12.3 days | 1.2 day |

Computational time obtained by summing computational time for each processor without cost of communications.