ANOTHER $hp3D$, ORIENTATION EMBEDDED SHAPE FUNCTIONS FOR ELEMENTS OF ALL SHAPES, AND HOW TO CODE THE PRIMAL DPG METHOD

L. Demkowicz
F. Fuentes, B. Keith and S. Nagaraj
ICES, The University of Texas at Austin

ICERM Workshop on *Robust Discretization and Fast Solvers for Computable Multi-Physics Models*
Brown University, May 11-15, 2014

Motto:

The difference between 2D and 3D computations is analogous to those that separate boys from men.

Act One

One more $hp3D$ code...

hp3d * overview

The code is designed for solving 3D coupled multi-physics problems:

GMP provides *MBG* description with globally C^0-compatible parameterizations.

hp-Mesh stores all topological (i.e., vertex, edge, face, and interior) nodes and supports variable order of approximation; an element is defined as an assembly of nodes.

Data struct. includes only two object arrays.

- `ELEMS` initial mesh elements only.
- `NODES` a family of nodal trees.

FEs support for exact sequence elements; shape functions with embedded orientations.

Refinements support for isotropic and anisotropic refinements for elements of all shapes (i.e., tet, hexa, prism, and pyramid).

Constrained Approx. allows one-level hanging nodes.

*Developed with Paolo Gatto (now at Brown) and Kyongjoo Kim (now at Sandia).

Mesh topology

hp-mesh structure includes all topological entities

- For multi-physics problem, one may need to support *different energy spaces following the exact sequence*.
- All topological nodes are defined with *orientations*.

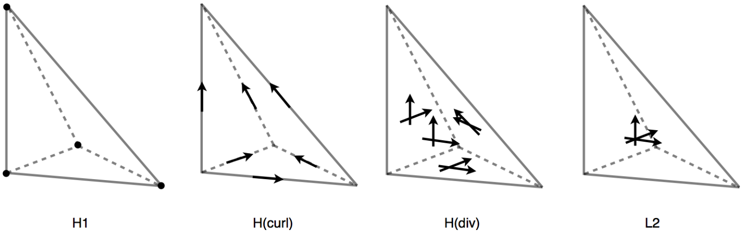


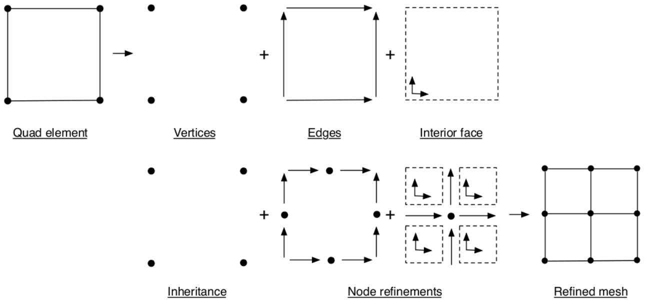
Figure: Different discretizations associate *DOF* with different topological nodes.

Mesh topology

hp-mesh structure includes all topological entities

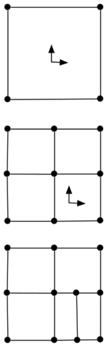
- For multi-physics problem, one may need to support *different energy spaces following the exact sequence*.
- All topological nodes are defined with *orientations*.
- Mesh refinement translates into *an orchestrated node breaking procedure*.

$$\text{Element} = \text{Vertices} + \text{Edges} + \text{Faces} + \text{Interior}$$

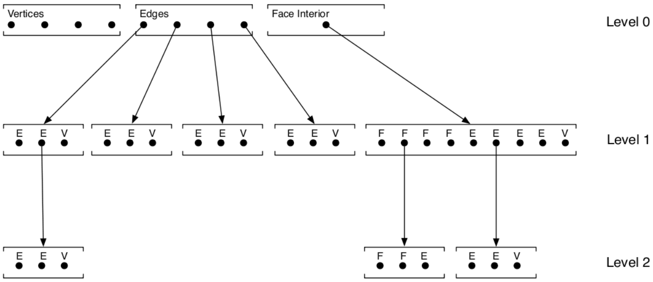


Nodal trees

- Mesh refinements are *explicitly recorded in nodal trees*.
- *No need for additional search tables* (e.g., hash-table) to keep track of dynamically changed element-to-node connectivities.
- Generic algorithms are developed to reconstruct element-to-node connectivities.



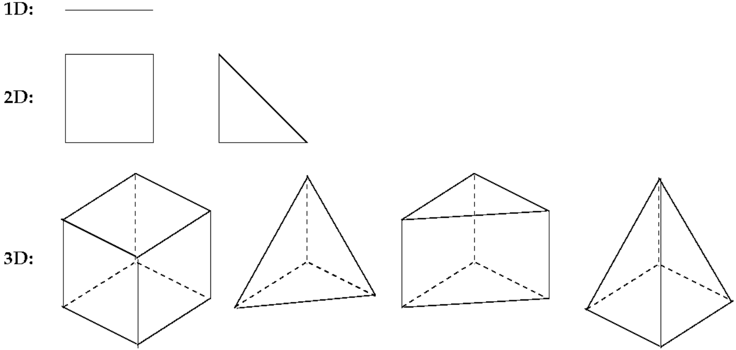
Mesh adaptation



Node trees

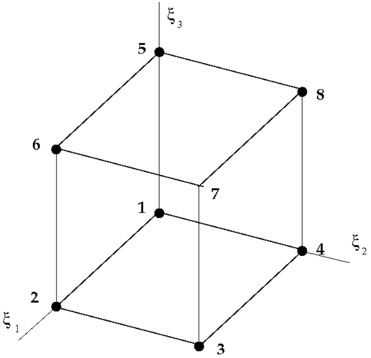
Shape functions for elements of all shapes
and the exact sequence spaces

Elements of “all shapes”



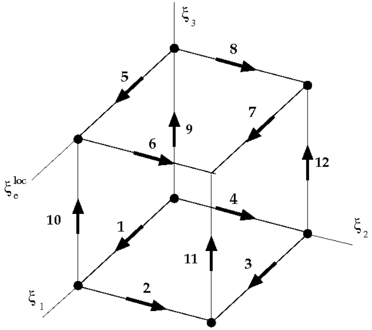
Element system of coordinates, node enumeration and local orientations

Hexahedral element vertices:



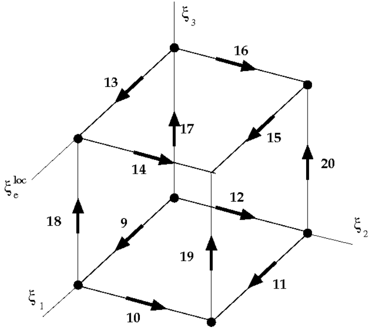
Element system of coordinates, node enumeration and local orientations

Hexahedral element edges. Enumeration and local orientation (parametrization):



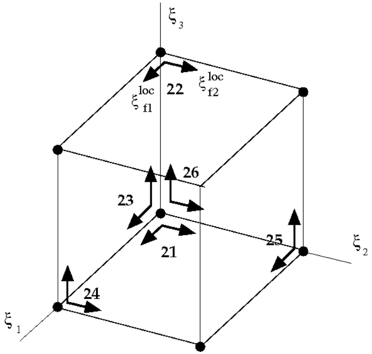
Element system of coordinates, node enumeration and local orientations

Hexahedral element edge nodes enumeration :



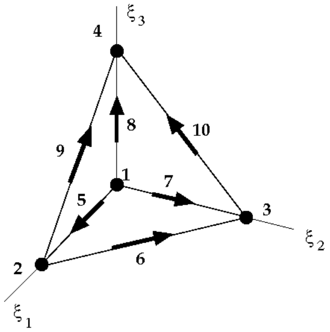
Element system of coordinates, node enumeration and local orientations

Hexahedral element face nodes enumeration and orientation (parametrization):



Element system of coordinates, node enumeration and local orientations

Tetrahedral element vertex and edge nodes enumeration and orientation:



and so on... (Show `module/element_data.F`)

Finite element spaces form exact sequences of the first type †

segment: $\mathcal{P}^p \xrightarrow{\partial} \mathcal{P}^{p-1}$

quad: $\mathcal{P}^p \otimes \mathcal{P}^q \xrightarrow{\boldsymbol{\nabla}} (\mathcal{P}^{p-1} \otimes \mathcal{P}^q) \times (\mathcal{P}^p \otimes \mathcal{P}^{q-1}) \xrightarrow{\text{curl}} \mathcal{P}^{p-1} \otimes \mathcal{P}^{q-1}$

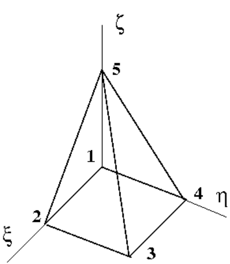
triangle: $\mathcal{P}^p \xrightarrow{\boldsymbol{\nabla}} \mathcal{P}^{p-1} \oplus \mathcal{R}^p \xrightarrow{\text{curl}} \mathcal{P}^{p-1}$ where:
$\mathcal{R}^p : \{E \in \tilde{\mathcal{P}}^p \ : \ x \cdot E(x) = 0 \quad \forall x\}$

hexa: $\mathcal{P}^p \otimes \mathcal{P}^q \otimes \mathcal{P}^r \xrightarrow{\boldsymbol{\nabla}} (\mathcal{P}^{p-1} \otimes \mathcal{P}^q \otimes \mathcal{P}^r) \times (\mathcal{P}^p \otimes \mathcal{P}^{q-1} \otimes \mathcal{P}^r) \times (\mathcal{P}^p \otimes \mathcal{P}^q \otimes \mathcal{P}^{r-1})$
$\xrightarrow{\boldsymbol{\nabla}\times} (\mathcal{P}^p \otimes \mathcal{P}^{q-1} \otimes \mathcal{P}^{r-1}) \times (\mathcal{P}^{p-1} \otimes \mathcal{P}^q \otimes \mathcal{P}^{r-1}) \times (\mathcal{P}^{p-1} \otimes \mathcal{P}^{q-1} \otimes \mathcal{P}^r)$
$\xrightarrow{\boldsymbol{\nabla}\circ} \mathcal{P}^{p-1} \otimes \mathcal{P}^{q-1} \otimes \mathcal{P}^{r-1}$

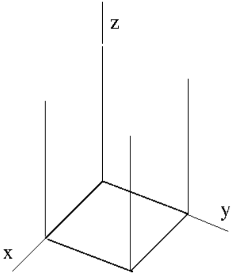
and so on...

† Refers to Nédélec's elements of the first type. The drop in polynomial degree is always one.

For the pyramid the spaces are far from trivial [‡]



$$x = \frac{\xi}{1+\zeta} \qquad \xi = \frac{x}{1-z}$$
$$y = \frac{\eta}{1+\zeta} \qquad \eta = \frac{y}{1-z}$$
$$z = \frac{\zeta}{1+\zeta} \qquad \zeta = \frac{z}{1-z}$$



$$\begin{aligned}
W_p &= \{u \in \mathcal{Q}_p^{p,p,p} : \nabla u \in \mathcal{Q}_p^{p-1,p,p-1} \times \mathcal{Q}_p^{p,p-1,p-1} \times \mathcal{Q}_{p+1}^{p,p,p-1}\} \\
Q_p &= \{E \in \mathcal{Q}_{p+1}^{p-1,p,p-1} \times \mathcal{Q}_{p+1}^{p,p-1,p-1} \times \mathcal{Q}_{p+1}^{p,p,p-2} : \\
&\quad \nabla \times E \in \mathcal{Q}_{p+2}^{p,p-1,p-1} \times \mathcal{Q}_{p+2}^{p-1,p,p-1} \times \mathcal{Q}_{p+2}^{p-1,p-1,p}\} \\
V_p &= \{V \in \mathcal{Q}_{p+2}^{p,p-1,p-1} \times \mathcal{Q}_{p+2}^{p-1,p,p-1} \times \mathcal{Q}_{p+2}^{p-1,p-1,p} : \\
&\quad \nabla \cdot V \in \mathcal{Q}_{p+3}^{p-1,p-1,p}\} \\
Y_p &= \mathcal{Q}_{p+3}^{p-1,p-1,p}
\end{aligned}$$

where

$$\mathcal{Q}_k^{pqr} := \{\frac{x^i y^j z^l}{(1+z)^k} : 0 \le i \le p,\ 0 \le j \le q,\ 0 \le l \le r\}$$

[‡] Nigam, N. and Phillips, J., "High-order Conforming Finite Elements on Pyramids", *IMA Journal of Numerical Analysis*, 32(2): 448-483, 2012.

A bit of history: Ciarlet's approach.

Degrees of freedom come first:

$$\psi_j : \mathcal{X}(K) \supset X(K) \to \mathbb{R}, \quad j = 1, \ldots, n$$

where $\mathcal{X}(K)$ is continuosly embedded in the energy space, $X(K)$ is the space of FE shape functions, d.o.f. functionals are continuous, $n = \text{dim} = X(K)$, and $\psi_j|_{X(K)}$ are linearly independent (unisolvency condition).
Shape functions come next (dual basis):

$$\phi_i \in X(K), \quad \langle \psi_j, \phi_i \rangle = \delta_{ij}$$

Global conformity (continuity) is enforced by a proper selection of d.o.f.. *The approach implies standard assembly procedure.* Shape functions are precomputed as linear combinations of monomials.

A bit of history: Szabo's approach.

Shape functions come first:

$$\phi_i \in X(K), i = 1, \ldots, n = \text{dim} X(K)$$

Global conformity (continuity) is enforced by classifying the shape functions into vertex, edge, face and interior "modes" (Szabo's terminology). *The assembly procedure is non-standard* as it requires accounting for edge and face orientations (Szabo's sign factors). Shape functions are computed on the fly using recursive formulas.
No d.o.f. are introduced.

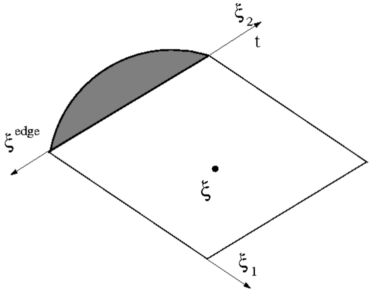
Topological character of shape functions

H^1	vertex, edge, face, interior functions
$H(\text{curl})$	edge, face, interior functions
$H(\text{div})$	face, interior functions
L^2	interior functions

Orientation embedded shape functions [§]. Edge shape functions are owned by the edge.

Switch from edge coordinate ξ_{edge} to local edge coordinate t:

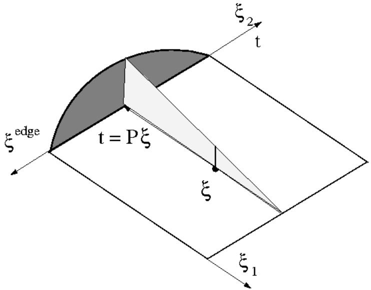
$$\hat{u}(t) = u(\xi_{\text{edge}}), \quad t = 1 - \xi_{\text{edge}}$$



[§] P. Gatto, L. Demkowicz, "Construction of H^1-Conforming Hierarchical Shape Functions for Elements of All Shapes and Transfinite Interpolation", *Finite Elements in Analysis and Design*, 46: 474-486, 2010.

Orientation embedded shape functions . Edge shape functions are owned by the edge.

Project ξ onto the edge:



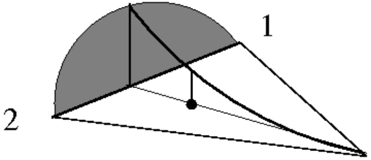
pick up the value at $t = P\xi$, and blend with a blending function $\psi(\xi)$:

$$u(\xi) := \hat{u}(t)\psi(\xi)$$

transform $\rightarrow$ project $\rightarrow$ blend

(Show the code.)

In a triangle we need to blend with a higher order polynomial (Dubinger's construction)



It is convenient to work with edge affine coordinates λ_1, λ_2:

$$\phi(\xi) = \hat{u}_n(\underbrace{\frac{\lambda_1}{\lambda_1 + \lambda_2}}_{\mu_1}, \underbrace{\frac{\lambda_2}{\lambda_1 + \lambda_2}}_{\mu_2}) \underbrace{(\lambda_1 + \lambda_2)^n}_{\text{blending function}}$$

where $\hat{u}(\mu_1, \mu_2)$ is one of possible representation of the edge shape function in terms of edge affine coordinates μ_1, μ_2.

This leads to the concept of scaled polynomials [§]

Scaled polynomials: $u_n \in \mathcal{P}^n(-1,1)$,

$$\tilde{u}_n(x;t) := u_n(\frac{x}{t})t^n$$

Scaling produces a *homogeneous* polynomials of (total) degree n in (x,t).
Scaled Legendre Polynomials are evaluated using the standard recursion formula:

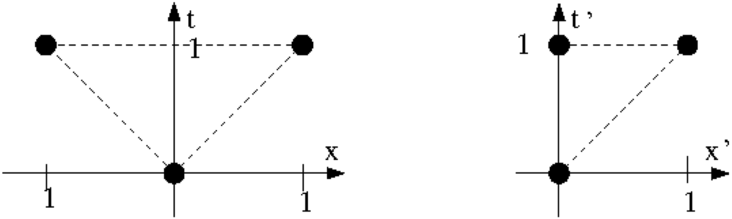
$$n\tilde{P}_n(x;t) = (2n-1)x\tilde{P}_{n-1}(x;t) - (n-1)t^2\tilde{P}_{n-2}(x;t)\,.$$

Shifted scaled Legendre polynomials are invariant (by construction) with affine transformations:

transform to the symmetric triangle $\rightarrow$ scale $\rightarrow$ backtransform (shift)

[§] J. Schoeberl, J. Zaglmayr, "High order Nédélec Elements with Local Complete Sequence Property", *International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 24(2): 374-384, 2005.

Shifted scaled polynomials



$$\tilde{P}^s_n(x';t') := \tilde{P}_n(2x'-t';t')$$

Shifted Scaled Lobatto Polynomials are integrals of shifted scaled Legendre polynomials

$$\tilde{L}^s_{n+1}(x';t') := \int_0^{x'} \tilde{P}^s_n(s';t')\,ds'$$

and are computed using recursion.

Edge functions revisited.

Homogenization of polynomials:

$$\begin{aligned}[\hat{u}]_n(\lambda_1,\lambda_2) &:= \hat{u}_n(\lambda_1,\lambda_2;\lambda_1+\lambda_2) \\ &= \hat{u}_n\left(\tfrac{\lambda_1}{\lambda_1+\lambda_2},\tfrac{\lambda_2}{\lambda_1+\lambda_2}\right)(\lambda_1+\lambda_2)^n\end{aligned}$$

Edge H^1 shape function again:

$$\phi = [\hat{u}]_n(\lambda_1,\lambda_2)$$

Fastforward to edge $H(\text{curl})$ shape function:

$$E = [\hat{u}]_n(\lambda_1,\lambda_2)\ \underbrace{(\lambda_1\boldsymbol{\nabla}\lambda_2 - \lambda_2\boldsymbol{\nabla}\lambda_1)}_{\text{Whitney edge shape function}}$$

Critical property:

$$\boldsymbol{\nabla}\times E = (n+2)[\hat{u}]_n(\lambda_1,\lambda_2)\boldsymbol{\nabla}\lambda_1\times\boldsymbol{\nabla}\lambda_2$$

(no derivatives of u_n needed !).

An analogous property for H(div) shape functions

Face shape function:

$$V = [\hat{u}]_n(\lambda_1, \lambda_2, \lambda_3)\,(\lambda_1 \boldsymbol{\nabla}\lambda_2 \times \boldsymbol{\nabla}\lambda_3 + \lambda_3 \boldsymbol{\nabla}\lambda_1 \times \boldsymbol{\nabla}\lambda_2 + \lambda_2 \boldsymbol{\nabla}\lambda_3 \times \boldsymbol{\nabla}\lambda_1)$$

Divergence of V:

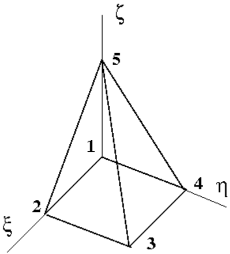
$$\boldsymbol{\nabla} \circ V = (n+3)[\hat{u}]_n(\lambda_1, \lambda_2, \lambda_3)\,\underbrace{[\boldsymbol{\nabla}\lambda_1, \boldsymbol{\nabla}\lambda_1, \boldsymbol{\nabla}\lambda_3]}_{\text{mixed product}}$$

requires no derivatives of u_n.
Orientations are accounted for by swapping affine coordinates (Show the code).

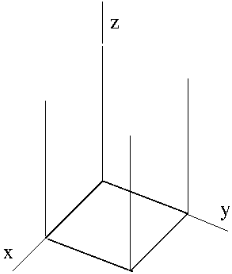
Tones of details skipped

In the incoming ICES Report:
Construction of H^1 face and element bubbles,
Construction of $H(\text{curl})$ face and element bubbles,
Construction of $H(\text{div})$ bubbles,
Construction of shape functions for the prism (tensor product of a triangle and interval), etc.

Tricky pyramid



$$x = \frac{\xi}{1+\zeta} \qquad \xi = \frac{x}{1-z}$$
$$y = \frac{\eta}{1+\zeta} \qquad \eta = \frac{y}{1-z}$$
$$z = \frac{\zeta}{1+\zeta} \qquad \zeta = \frac{z}{1-z}$$



Vertex shape functions:

$$\begin{aligned}
\lambda_1 &= \frac{(1-x)(1-y)}{1+z} &&= \frac{(1-\xi-\zeta)(1-\eta-\zeta)}{1-\zeta}\\
\lambda_2 &= \frac{x(1-y)}{1+z} &&= \frac{\xi(1-\eta-\zeta)}{1-\zeta}\\
\lambda_3 &= \frac{xy}{1+z} &&= \frac{\xi\eta}{1-\zeta}\\
\lambda_4 &= \frac{(1-x)y}{1+z} &&= \frac{(1-\xi-\zeta)\eta}{1-\zeta}\\
\lambda_5 &= \frac{z}{1+z} &&= \zeta
\end{aligned}$$

Tricky pyramid

Some of the shape functions can be constructed using the formulas for the tetrahedron with the affine coordinates replaced with vertex shape functions. For instance, for vertical edge 15 (common to two triangular faces),

H^1 edge modes:

$$\phi = [\hat{u}]_n(\lambda_1, \lambda_5)$$

H(curl) modes:

$$E = [\hat{u}]_n(\lambda_1 \boldsymbol{\nabla} \lambda_5 - \lambda_5 \boldsymbol{\nabla} \lambda_1)$$

For edge 12, however, the construction is different as it needs to reconcile the triangular vertical face and horizontal square base face shape functions.

Tricky pyramid; a stumbling block

For all but one case, we were able to construct the edge and face modes as products of properly extended higher order shape functions defined on the edge or face and the lowest order edge or shape function. This is a unique property of Nédélec spaces of first type.

The stumbling block was the $H(\text{div})$ face 125 modes.

Pull back maps (Piola transforms) and the original space

$$\begin{pmatrix} V_\xi \\ V_\eta \\ V_\zeta \end{pmatrix} = (1+z)^2 \begin{pmatrix} 1+z & 0 & -x \\ 0 & 1+z & -y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}$$

$$\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = -\frac{1}{(1+z)^3} \begin{pmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & -(1+z) \end{pmatrix} \begin{pmatrix} V_\xi \\ V_\eta \\ V_\zeta \end{pmatrix}$$

$$\{V \in \mathcal{Q}_{k+2}^{k,k-1,k-1} \times \mathcal{Q}_{k+2}^{k-1,k,k-1} \times \mathcal{Q}_{k+2}^{k-1,k-1,k} : \boldsymbol{\nabla} \cdot V \in \mathcal{Q}_{k+3}^{k-1,k-1,k-1}\}$$

Lowest order mode[¶]

$$V = \frac{1}{(1+z)^3}\begin{pmatrix} 0 \\ 1-y \\ -\frac{1}{2}z \end{pmatrix} \qquad \nabla \cdot V = -\frac{3}{2(1+z)^4}\,.$$

Higher order mode, first attempt:

$$V = \tfrac{1}{(1+z)^3}\begin{pmatrix} 0 \\ (1-y)\,u_{k-1}(\tfrac{x}{1+z},\tfrac{z}{1+z}) \\ -\frac{C}{2}z \end{pmatrix},$$

$$\nabla \cdot V = -\tfrac{1}{(1+z)^4}\left((u_{k-1}(\tfrac{x}{1+z},\tfrac{z}{1+z}) - C)(1+z) + \tfrac{3C}{2}\right)$$

where $C = u_{k-1}(0,1)$.

[¶] At a first glance, presence of non-zero z-component seems to be unnecessary. With vanishing V_z, the shape function would still have had the right vanishing properties in the x, y, z space. It is only upon computing the divergence that we learn that the third component is necessary to fit the function into the right space.

Our solution: add a bubble

The main source of the trouble seems to come from the fact that the $H(\text{div})$ space is a bit "too tied". The problem can be solved by adding the bubble

$$V = \frac{1}{(1+z)^3}\begin{pmatrix} 0 \\ 0 \\ z \end{pmatrix} \qquad \nabla \cdot V = \frac{1-2z}{(1+z)^4}$$

or, in the finite pyramid frame,

$$V = -\zeta \begin{pmatrix} \frac{\xi}{1-\zeta} \\ \frac{\eta}{1-\zeta} \\ -1 \end{pmatrix} \qquad \nabla \cdot V = 1 - 2\frac{\zeta}{1-\zeta}\,.$$

The spaces are extended accordingly for the higher order element,

$$\{V \in \mathcal{Q}_{k+2}^{k,k-1,k-1} \times \mathcal{Q}_{k+2}^{k-1,k,k-1} \times \mathcal{Q}_{k+2}^{k-1,k-1,k} \;:\; \nabla \cdot V \in \mathcal{Q}_{k+3}^{k-1,k-1,k}\}\,.$$

Higher order modes

$$V = \frac{u_{k-1}\left(\frac{x}{1+z}, \frac{z}{1+z}\right)}{(1+z)^3} \begin{pmatrix} 0 \\ 1-y \\ 0 \end{pmatrix}$$

or, in the finite pyramid frame,

$$V = u(\xi, \zeta) \begin{pmatrix} 0 \\ \frac{1-\eta-\zeta}{1-\zeta} \\ 0 \end{pmatrix} \qquad \text{div } V = -\frac{u(\xi, \zeta)}{1-\zeta}\,.$$

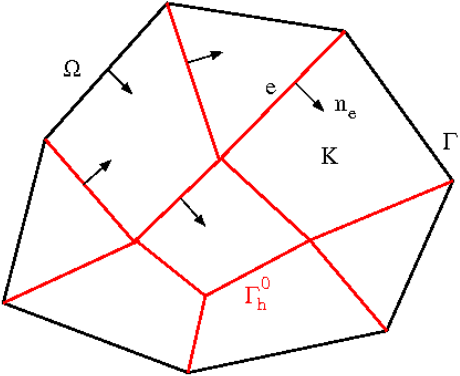
A summary

- A logically coherent construction for elements of all shapes and the whole exact sequence.
- Small number of primitives (scaled polynomials) and a hierarchical logic allowing for quick modifications.
- Stand alone package in Fortran 90 (6k lines).
- Extended element-to-nodes connectivity needed, standard assembly procedure.
- Partial verification (reproduction of polynomials) done.
- Anyone interested in helping with verification ?

Primal DPG Method

Primal DPG method

Standard assumptions: $\Omega \subset \mathbb{R}^N$ Lipschitz domain,



Elements: K
Faces (Edges): e
Skeleton: $\Gamma_h = \bigcup_K \partial K$
Internal skeleton: $\Gamma_h^0 = \Gamma_h - \Gamma$

Primal DPG method

Given $f \in L^2(\Omega)$, consider the model problem,

$$\begin{cases} \quad u &= u_0 \quad \text{on } \Gamma := \partial\Omega \\ -\Delta u &= f \quad \text{in } \Omega \end{cases}$$

Multiply the PDE with a test function v, integrate over each element K, integrate by parts and sum up over all elements,

$$\sum_K \int_K \nabla u \cdot \nabla v + \sum_K \int_{\partial K} \frac{\partial u}{\partial n} v = \sum_K \int f v$$

The boundary term represents jumps,

$$\sum_K \int_{\partial K} \frac{\partial u}{\partial n} v = \sum_{e \subset \Gamma_h^0} \int_e \frac{\partial u}{\partial n_e} [v] + \sum_{e \subset \Gamma} \int_e \frac{\partial u}{\partial n_e} v$$

Primal DPG method

This leads to the variational problem:

$$\begin{cases} u \in H^1(\Omega), u = u_0 \text{ on } \Gamma,\ \hat{t} \in H^{-1/2}(\Gamma_h) \\ (\nabla u, \nabla_h v) - \langle \hat{t}, v \rangle_{\Gamma_h} = (f, v) \quad v \in H^1(\Omega_h) \end{cases}$$

where

$$H^{-1/2}(\Gamma_h) = \text{trace of } H(\text{div}, \Omega) \text{ on } \Gamma_h$$

equipped with the quotient norm.

Theorem [‖]

The variational problem above is well posed with a mesh independent inf-sup constant γ.

[‖] L. Demkowicz and J. Gopalakrishnan. "A primal DPG method without a first order reformulation", *Comput. Math. Appl.*, 66(6):1058–1064, 2013

DPG is a minimum residual method **

$$\left\{ \begin{array}{l} u \in U \\ b(u,v) = l(v) \quad v \in V \end{array} \right. \quad \Leftrightarrow \quad \begin{array}{l} Bu = l \quad B : U \to V' \\ \langle Bu, v \rangle = b(u,v) \end{array}$$

**

- J.H. Bramble, R.D. Lazarov, J.E. Pasciak, "A Least-squares Approach Based on a Discrete Minus One Inner Product for First Order Systems" *Math. Comp*, **66**, 935-955, 1997.
- L.D., J. Gopalakrishnan. "A Class of Discontinuous Petrov-Galerkin Methods. Part II: Optimal Test Functions." *Numer. Meth. Part. D. E.*, **27**, 70-105, 2011.

DPG is a minimum residual method **

$$\left\{\begin{array}{ll} u \in U & \\ b(u,v) = l(v) & v \in V \end{array}\right. \quad \Leftrightarrow \quad \begin{array}{l} Bu = l \quad B : U \to V' \\ \langle Bu, v\rangle = b(u,v) \end{array}$$

- **Minimum residual method:** $U_h \subset U$,

$$\frac{1}{2}\|Bu_h - l\|_{V'}^2 \to \min_{u_h \in U_h}$$

**

- J.H. Bramble, R.D. Lazarov, J.E. Pasciak, "A Least-squares Approach Based on a Discrete Minus One Inner Product for First Order Systems" *Math. Comp*, **66**, 935-955, 1997.
- L.D., J. Gopalakrishnan. "A Class of Discontinuous Petrov-Galerkin Methods. Part II: Optimal Test Functions." *Numer. Meth. Part. D. E.*, **27**, 70-105, 2011.

DPG is a minimum residual method **

$$\begin{cases} u \in U \\ b(u,v) = l(v) \quad v \in V \end{cases} \Leftrightarrow \begin{array}{l} Bu = l \quad B : U \to V' \\ \langle Bu, v \rangle = b(u,v) \end{array}$$

- **Minimum residual method:** $U_h \subset U$,

$$\frac{1}{2}\|Bu_h - l\|^2_{V'} \to \min_{u_h \in U_h}$$

- **Riesz operator:**

$$R_V : V \to V', \ \langle R_V v, \delta v \rangle = (v, \delta v)_V$$

is an *isometry*, $\|R_V v\|_{V'} = \|v\|_V$.

**

- J.H. Bramble, R.D. Lazarov, J.E. Pasciak, "A Least-squares Approach Based on a Discrete Minus One Inner Product for First Order Systems" *Math. Comp*, **66**, 935-955, 1997.
- L.D., J. Gopalakrishnan. "A Class of Discontinuous Petrov-Galerkin Methods. Part II: Optimal Test Functions." *Numer. Meth. Part. D. E.*, **27**, 70-105, 2011.

DPG is a minimum residual method **

$$\begin{cases} u \in U \\ b(u,v) = l(v) \quad v \in V \end{cases} \quad \Leftrightarrow \quad \begin{array}{l} Bu = l \quad B : U \to V' \\ \langle Bu, v \rangle = b(u,v) \end{array}$$

- **Minimum residual method:** $U_h \subset U$,

$$\frac{1}{2}\|Bu_h - l\|^2_{V'} \to \min_{u_h \in U_h}$$

- **Riesz operator:**

$$R_V : V \to V', \ \langle R_V v, \delta v \rangle = (v, \delta v)_V$$

  is an *isometry*, $\|R_V v\|_{V'} = \|v\|_V$.
- **Minimum residual method reformulated:**

$$\frac{1}{2}\|Bu_h - l\|^2_{V'} = \frac{1}{2}\|R_V^{-1}(Bu_h - l)\|^2_V \to \min_{u_h \in U_h}$$

**

- J.H. Bramble, R.D. Lazarov, J.E. Pasciak, "A Least-squares Approach Based on a Discrete Minus One Inner Product for First Order Systems" *Math. Comp*, **66**, 935-955, 1997.
- L.D., J. Gopalakrishnan. "A Class of Discontinuous Petrov-Galerkin Methods. Part II: Optimal Test Functions." *Numer. Meth. Part. D. E.*, **27**, 70-105, 2011.

(Practical) DPG is an easy mixed method[††]

Introducing,

$$\big(\underbrace{R_V^{-1}(Bu_h - l)}_{=:\psi(\text{error representation function})}, R_V^{-1} B\delta u_h\big)_V = 0 \quad \delta u_h \in U_h$$

[††]W. Dahmen, Ch. Huang, Ch. Schwab, and G. Welper. "Adaptive Petrov Galerkin methods for first order transport equations", *SIAM J. Num. Anal.* 50(5): 242-2445, 2012

(Practical) DPG is an easy mixed method[††]

Introducing,

$$\Big(\underbrace{R_V^{-1}(Bu_h - l)}_{=:\psi\text{(error representation function)}}, R_V^{-1} B\delta u_h \Big)_V = 0 \quad \delta u_h \in U_h$$

or

$$\begin{cases} \psi = R_V^{-1}(Bu_h - l) \\ (\psi, R_V^{-1} B\delta u_h)_V = 0 \quad \delta u_h \in U_h \end{cases}$$

[††] W. Dahmen, Ch. Huang, Ch. Schwab, and G. Welper. "Adaptive Petrov Galerkin methods for first order transport equations", *SIAM J. Num. Anal.* 50(5): 242-2445, 2012

(Practical) DPG is an easy mixed method[††]

Introducing,

$$\Big(\underbrace{R_V^{-1}(Bu_h - l)}_{=:\psi\text{(error representation function)}}, R_V^{-1} B\delta u_h\Big)_V = 0 \quad \delta u_h \in U_h$$

or

$$\begin{cases} (\psi, \delta v)_V - b(u_h, \delta v) & = -l(\delta v) \quad \forall \delta v \in V \\ b(\delta u_h, \psi) & = 0 \qquad\quad\; \forall \delta u_h \in U_h \end{cases}$$

[††] W. Dahmen, Ch. Huang, Ch. Schwab, and G. Welper. "Adaptive Petrov Galerkin methods for first order transport equations", *SIAM J. Num. Anal.* 50(5): 242-2445, 2012

(Practical) DPG is an easy mixed method[††]

Introducing,

$$(\underbrace{R_V^{-1}(Bu_h - l)}_{=:\psi(\text{error representation function})}, R_V^{-1}B\delta u_h)_V = 0 \quad \delta u_h \in U_h$$

or

$$\begin{cases} (\psi, \delta v)_V - b(u_h, \delta v) & = -l(\delta v) \quad \forall \delta v \in V \\ b(\delta u_h, \psi) & = 0 \qquad\quad \forall \delta u_h \in U_h \end{cases}$$

In practice, error representation function ψ must be approximated within an "enriched test space" $\psi_h \in \tilde{V}_h \subset V$

$$\begin{cases} (\psi_h, \delta v_h)_V - b(u_h, \delta v_h) & = -l(\delta v_h) \quad \forall \delta v_h \in \tilde{V}_h \\ b(\delta u_h, \psi_h) & = 0 \qquad\qquad \forall \delta u_h \in U_h \end{cases}$$

[††] W. Dahmen, Ch. Huang, Ch. Schwab, and G. Welper. "Adaptive Petrov Galerkin methods for first order transport equations", *SIAM J. Num. Anal.* 50(5): 242-2445, 2012

Brezzi's Conditions

- inf-sup in the kernel - trivially satisfied
- LBB inf-sup condition:

$$\sup_{\tilde{v}_h \in \tilde{V}_h} \frac{|b(u_h, \tilde{v}_h)|}{\|\tilde{v}_h\|_V} \geq \gamma_h \|u_h\|_U ,$$

  is "easy" to satisfy if we take "large" *enriched* space $\tilde{V}_h \subset V$.

The proof relies on the construction of an appropriate Fortin operator [‡‡] .The Fortin constant affects the ultimate discrete inf-sup constant γ_h that no longer is equal to γ.

[‡‡] J. Gopalakrishnan and W. Qiu. "An analysis of the practical DPG method", *Math. Comp.*, 2012.

Brezzi's Conditions

- inf-sup in the kernel - trivially satisfied
- LBB inf-sup condition:

$$\sup_{\tilde{v}_h \in \tilde{V}_h} \frac{|b(u_h, \tilde{v}_h)|}{\|\tilde{v}_h\|_V} \geq \gamma_h \|u_h\|_U \,,$$

  is "easy" to satisfy if we take "large" *enriched* space $\tilde{V}_h \subset V$.

The proof relies on the construction of an appropriate Fortin operator [‡‡] .The Fortin constant affects the ultimate discrete inf-sup constant γ_h that no longer is equal to γ. The approximate mixed problem can be viewed as the discretization of two infinite-dimensional mixed problems solved for:

u_h, ψ (a-posteriori error estimation), and
$u, \psi = 0$ (a-priori estimates).

See Jay's talk.

[‡‡] J. Gopalakrishnan and W. Qiu. "An analysis of the practical DPG method", *Math. Comp.*, 2012.

Main point: ψ is Condensed Out Elementwise

$$u_h = \sum_{i=1}^{N} u_i e_i, \quad \psi_h \approx \sum_{j=1}^{M} \psi_j g_j, \quad M >> N$$

Matrix form:

$$\begin{pmatrix} \boldsymbol{G} & -\boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\psi} \\ \boldsymbol{u} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{b} \\ \boldsymbol{0} \end{pmatrix}$$

where

$$\underbrace{G_{ij}}_{\text{Gram matrix}} = (g_i, g_j), \qquad \underbrace{B_{ij}}_{\text{expanded stiffness matrix}} = b(e_i, g_j)\,.$$

For broken test space, $\boldsymbol{G}$ is element block-diagonal, and $\boldsymbol{\psi}_h$ is eliminated *elemenwise*,

$$\underbrace{\boldsymbol{B}^T\boldsymbol{G}^{-1}\boldsymbol{B}}_{\text{DPG stiffness matrix}} \quad \boldsymbol{u}_h = \underbrace{\boldsymbol{B}^T\boldsymbol{G}^{-1}\boldsymbol{b}}_{\text{DPG load vector}} \quad .$$

DPG can be viewed as a preconditioned least squares method.

Primal DPG Formulation

Group unknown (watch for the overloaded symbol):

$$u_h := (\underbrace{u_h}_{\text{field}}, \underbrace{\hat{t}_h}_{\text{flux}})$$

Mixed system:

$$\begin{pmatrix} \boldsymbol{G} & -\boldsymbol{B}_1 & -\boldsymbol{B}_2 \\ \boldsymbol{B}_1^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B}_2^T & \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\psi} \\ \boldsymbol{u} \\ \hat{\boldsymbol{t}} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{b} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix}$$

where $\boldsymbol{B}_1, \boldsymbol{B}_2$ correspond to $(\boldsymbol{\nabla} u_h, \boldsymbol{\nabla}_h v_h)$ and $-\langle \hat{t}_h, v_h \rangle$, resp.
Eliminate $\boldsymbol{\psi}$ to get the DPG system:

$$\begin{pmatrix} \boldsymbol{B}_1^T \boldsymbol{G}^{-1} \boldsymbol{B}_1 & \boldsymbol{B}_1^T \boldsymbol{G}^{-1} \boldsymbol{B}_2 \\ \boldsymbol{B}_2^T \boldsymbol{G}^{-1} \boldsymbol{B}_1 & \boldsymbol{B}_2^T \boldsymbol{G}^{-1} \boldsymbol{B}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \hat{\boldsymbol{t}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{B}_1^T \boldsymbol{G}^{-1} \boldsymbol{b} \\ \boldsymbol{B}_2^T \boldsymbol{G}^{-1} \boldsymbol{b} \end{pmatrix}$$

Primal DPG method

Neglecting the error steming from the approximation of optimal test functions (computation of residual), we have,

$$\left(\|u - u_h\|^2_{H^1(\Omega)} \quad + \|\hat{t} - \hat{t}_h\|^2_{H^{-1/2}(\Gamma_h)}\right)^{1/2}$$
$$\leq \frac{1}{\gamma} \underbrace{\inf_{w_h, r_h} \left(\|u - w_h\|^2_{H^1(\Omega)} + \|\hat{t} - r_h\|^2_{H^{-1/2}(\Gamma_h)}\right)^{1/2}}_{\text{best approximation error}}$$

Additionally,

$$\left(\|u - u_h\|^2_{H^1(\Omega)} \quad + \|\hat{t} - \hat{t}_h\|^2_{H^{-1/2}(\Gamma_h)}\right)^{1/2}$$
$$\leq \frac{1}{\gamma} \underbrace{\sup_{v \in H^1(\Omega_h)} \frac{|(\boldsymbol{\nabla} u_h, \boldsymbol{\nabla}_h v) - \langle \hat{t}_h, v \rangle_{\Gamma_h}|}{\|v\|_{H^1(\Omega_h)}}}_{\text{computable residual}}$$

FE discretization

Hexahedral meshes

H^1 element for field u_h:

$$\mathcal{P}^p \otimes \mathcal{P}^p \otimes \mathcal{P}^p,$$

Trace of $H(\text{div})$ element:

$$(\mathcal{P}^p \otimes \mathcal{P}^{p-1} \otimes \mathcal{P}^{p-1}) \times (\mathcal{P}^{p-1} \otimes \mathcal{P}^p \otimes \mathcal{P}^{p-1}) \times (\mathcal{P}^{p-1} \otimes \mathcal{P}^{p-1} \otimes \mathcal{P}^p)$$

for flux $\hat{t}_h$, and the enriched element:

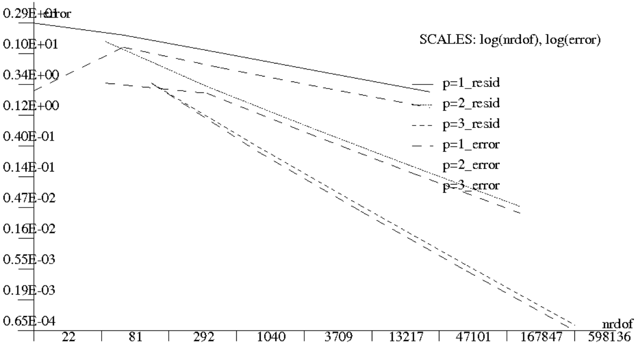
$$\mathcal{P}^{p+\Delta p} \otimes \mathcal{P}^{p+\Delta p} \otimes \mathcal{P}^{p+\Delta p},$$

for test function v_h.

In reported experiments: $p = 1, 2, 3$, $\Delta p = 2$.

Smooth solution, uniform refinements

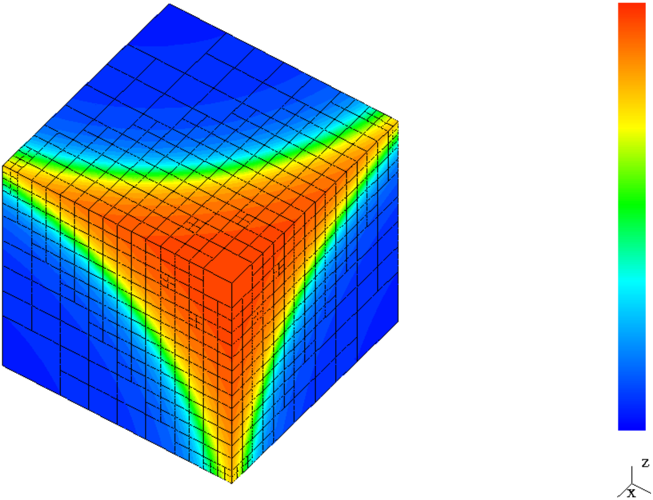
Rectangular domain $\Omega = (0,1) \times (0,2) \times (0,1)$,
Smooth solution: $u = \sin \pi x \sin \pi y \sin \pi z$
Boundary condition: $u = 0$.



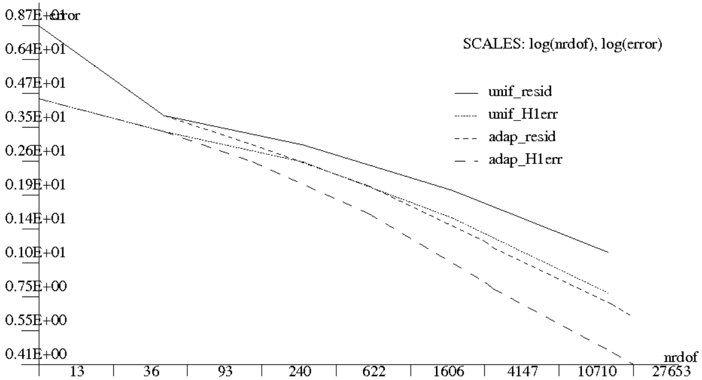
Residual versus H^1 error.

Manufactured shock solution

BC: $u = u_0$.

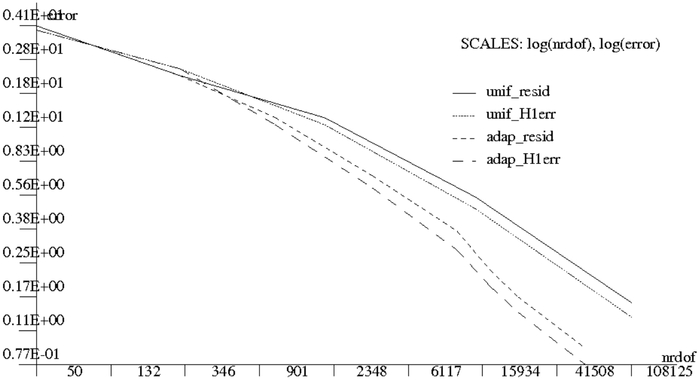


Shock solution, uniform and h-adaptive refinements, $p = 1$



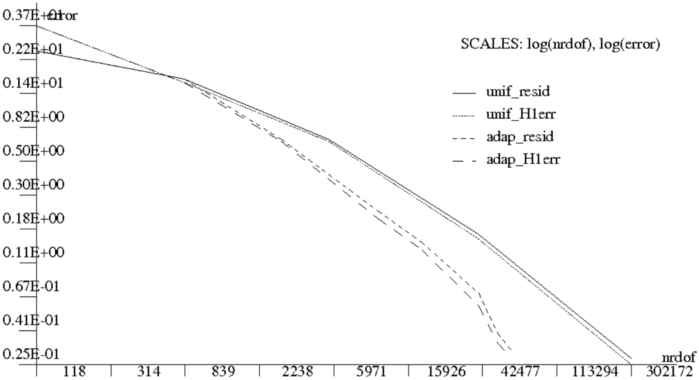
Convergence history for the residual and H^1 error

Shock solution, uniform and h-adaptive refinements, $p = 2$



Convergence history for the residual and H^1 error

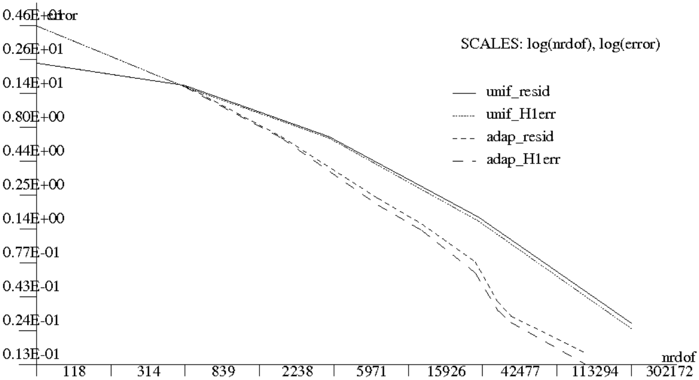
Shock solution, uniform and h-adaptive refinements, $p = 3$



Convergence history for the residual and H^1 error

Shock solution, $p = 3$, Mixed BC

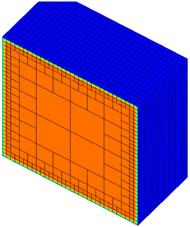
Mixed BC: trace: bottom, top, flux: sides.

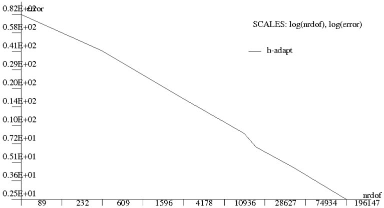


Convergence history for the residual and H^1 error

Reaction-dominated difusion, $p = 2$.

$$\begin{cases} u &= 0 \quad \text{on } \Gamma \\ -\epsilon^2 \Delta u + u &= 1 \quad \text{in } \Omega \end{cases}$$

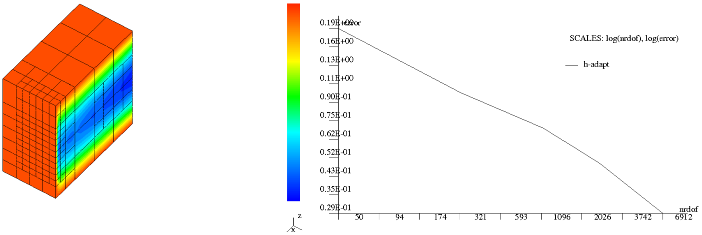




$\epsilon = 0.01$, left: solution after 7 iterations, right: convergence history

Convection-dominated difusion, $p = 2$.

$$\begin{cases} -\epsilon^2 \Delta u - u & = \sin \pi y \, \sin \pi z \qquad \text{at } x = 0 \\ u & = 0 \qquad \text{on the rest of } \Gamma \\ -\epsilon^2 \Delta u + \frac{\partial u}{\partial x} & = 0 \qquad \text{in } \Omega \end{cases}$$



$\epsilon = 0.01$, left: solution after 5 iterations, right: convergence history

Concluding remarks

- Incorporation of orientations into the element shape functions routine dramatically simplifies assembly procedure and constrained approximation.
- Coding DPG within a code supporting the exact sequence elements is very straightforward.
- With ultraweak variational formulation there is no need to construct 3D conforming discretizations - DPG naturally extends to polyhedral elements.

Thank you