Discontinuous Galerkin Methods

Part 1: Discretisation and efficient implementation

K.Hillewaert

Cenaero
Excellence in Simulation Technologies for Aeronautics

Cemracs Summer School, Marseille, July 20th 2012

Outline

DGM/IP methods

Framework : governing equations

Consider a generic set of N convection-diffusion-reaction equations

$$\mathcal{L}_m(\tilde{u}) = \frac{\partial \tilde{u}_m}{\partial t} + \nabla \cdot \vec{f}_m(\tilde{u}) + \nabla \cdot \vec{d}_m(\tilde{u}, \nabla \tilde{u}) + S_m(\tilde{u}, \nabla \tilde{u}) = 0$$

where

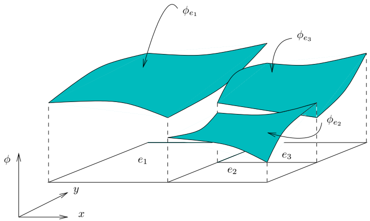
- $\tilde{u} \in (\mathbb{R}(\Omega))^N$ the state vector
- $\vec{f}$ the convective flux vector
- $\vec{d}$ the diffusive flux vector
- S the source term

with the first order expansion of $\vec{d}$

$$\vec{d}^k_m = \mathbf{D}^{kl}_{mn} \frac{\partial \tilde{u}_n}{\partial x^l} + \mathcal{O}((\nabla \tilde{u})^2)$$

DGM/IP methods

Framework : basic ingredients



Approximation $u \approx \tilde{u}$ on $\mathcal{E} = \cup e \approx \Omega$ is

- regular (polynomial, harmonic functions, waves, ...) on each element

$$u|_e \in (\mathcal{P}(e))^N$$

- not $\mathcal{C}_0$ continuous $\leftrightarrow$ standard FEM

$$u \in (\Phi(\mathcal{E}))^N = \cup (\mathcal{P}(e))^N$$

Galerkin formulation

$$a(u, v) \hat{=} \int_\Omega v_m \cdot \mathcal{L}_m(u) dV = 0 \ , \ \forall v \in \Phi$$

DGM/IP methods

Framework : Galerkin variational formulation

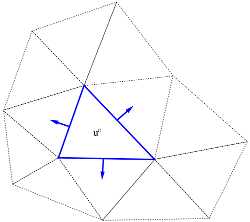
Take generic conservation equation

$$\frac{\partial \tilde{u}_m}{\partial t} + \nabla \cdot \vec{g}_m = 0$$

Naive Galerkin :

$$\int_\Omega v_m \frac{\partial u_m}{\partial t} dV + \int_\Omega v_m \nabla \cdot \vec{g}_m dV = 0 \ , \ \ \forall v \in \Phi$$

$$= \sum_e \int_e v_m \frac{\partial u_m}{\partial t} dV + \sum_e \left(- \int_e \nabla v_m \cdot \vec{g}_m dV + \oint_{\partial e} v_m \vec{g}_m \cdot \vec{n} dS \right)$$

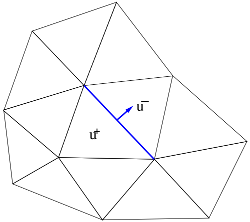


DGM/IP methods

Framework : Galerkin variational formulation

Take generic conservation equation

$$\frac{\partial \tilde{u}_m}{\partial t} + \nabla \cdot \vec{g}_m = 0$$



Naive Galerkin :

$$\int_\Omega v_m \frac{\partial u_m}{\partial t} dV + \int_\Omega v_m \nabla \cdot \vec{g}_m dV = 0 \ , \ \ \forall v \in \Phi$$

$$= \sum_e \int_e v_m \frac{\partial u_m}{\partial t} dV + \sum_e \left(- \int_e \nabla v_m \cdot \vec{g}_m dV + \oint_{\partial e} v_m \vec{g}_m \cdot \vec{n} dS \right)$$

Define operators on boundary (*trace*) wrt normal $\vec{n}^+ = \vec{n} = -\vec{n}^-$

$$[\![a]\!] = a^+ \vec{n}^+ + a^- \vec{n}^-$$

$$[\![\vec{g}]\!] = \vec{g}^+ \cdot \vec{n}^+ + \vec{g}^- \cdot \vec{n}^-$$

$$\{\!\{a\}\!\} = (a^+ + a^-)/2$$

Then we continue

$$\sum_e \int_e v_m \frac{\partial u_m}{\partial t} dV - \sum_e \int_e \nabla v_m \cdot \vec{g}_m dV + \sum_f \int_f [\![v_m]\!] \{\!\{\vec{g}_m\}\!\} + [\![\vec{g}_m]\!] \{\!\{v_m\}\!\} dS$$

DGM/IP methods

Framework : interface fluxes

The DGM discretisation is then defined as

$$\sum_e v_m \frac{\partial u_m}{\partial t} - \sum_e \int_e \nabla v_m \cdot \vec{g}_m dV + \sum_f \int_f \gamma_m(\tilde{u}^+, \tilde{u}^-, v^+, v^-, \vec{n}) dS = 0 \ , \ \ \forall v \in \Phi$$

Requirements for γ

- stability
- consistent as $u^+ = u^- = \tilde{u}$

$$\begin{aligned}\lim_{h \to 0} \int_f \vec{g}_m^* dS &= \int_f [[v_m \vec{g}_m(\tilde{u})]].\, dS \\ &= \int_f [[v_m]] \{\{\vec{g}_m(\tilde{u})\}\} + \{\{v_m\}\} [[\vec{g}_m(\tilde{u})]] \, dS \\ &= \int_f [[v_m]] \, \vec{g}_m(\tilde{u}) dS\end{aligned}$$

- conservative : let $W_m = 1 \ \ \forall x \in e \ , \ \ W_m = 0 \ \ \forall x \notin e$

$$a(W_m, u_m) = -\oint_e \gamma_m(u^+, u^-, 1, 0, \vec{n}) dS \Rightarrow \gamma_m(u^+, u^-, 1, 0, \vec{n}) = -\gamma_m(u^-, u^+, 1, 0, -\vec{n})$$

DGM/IP methods

Framework : local reinterpretation

Global formulation

$$\sum_e \int_e v_m \frac{\partial u_m}{\partial t} dV - \sum_e \int_e \nabla v_m \cdot \vec{g}_m dV + \sum_f \int_f \gamma_m(\tilde{u}^+, \tilde{u}^-, v^+, v^-, \vec{n}) dS = 0 \ , \ \ \forall v \in \Phi$$

Choose basis for Φ composed of locally supported v^e and expand

$$u = \sum_e u^e$$

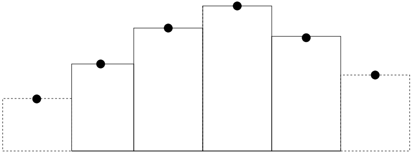
then the formulation reduces to elementwise FEM problems coupled by internal bc

$$\int_e v_m^e \frac{\partial u_m^e}{\partial t} dV - \int_e \nabla v_m^e \cdot \vec{g}_m dV + \int_{\partial e} \gamma_m(\tilde{u}^e, \tilde{u}^*, v^+, 0, \vec{n}) dS = 0 \ , \ \ \forall v^e \in (\Phi)_e$$

- internal bc provide guiding principle for choosing γ
- locally structured problem
- no global operations needed (in particular inversion of mass matrix)
- highly dense blocked matrix structure

DGM/IP methods

Convective terms : finite volume methods



Godunov scheme

- solution constant per element
- elementwise flux balance

$$V^e \frac{\partial u^e}{\partial t} + \oint_{\partial e} \mathcal{H}(u^e, u^*, \vec{n}) dS$$

- interface flux ~ local Riemann problem
  - consistency : $\mathcal{H}(u, u, n) = \vec{f}(u) \cdot \vec{n}$
  - conservation : $\mathcal{H}(u^-, u^+, -\vec{n}) = -\mathcal{H}(u^+, u^-, \vec{n})$
  - stability
  - entropy satisfying solutions

DGM/IP methods

Convective terms :

Unstructured maximum principle : *Local Extrema Diminishing* / positivity

Scalar problem

- E-flux

$$\frac{\mathcal{H}(u^+, u^-, \vec{n}) - \vec{f}(u) \cdot \vec{n}}{u^- - u^+} \leq 0 \ , \ \ \forall u \in [u^+, u^-]$$

- monotone fluxes

$$\frac{\partial \mathcal{H}}{\partial u^+} \geq 0 \qquad \frac{\partial \mathcal{H}}{\partial u^-} \leq 0 \forall u \in [u^+, u^-]$$

- upwind fluxes

$$\mathcal{H}(u^+, u^-, \vec{n}) = \mathtt{max}(0, (\vec{f} \cdot \vec{n})_u) u^+ + \mathtt{min}(0, (\vec{f} \cdot \vec{n})_u) u^- +$$

System of equations

- (approximate) Riemann solvers
- monotone fluxes

DGM/IP methods

Convective terms : Local Extrema Diminishing (LED)

If we can rewrite the FVM scheme as

$$\frac{du^e}{dt} = \sum_{ef} C_e (u_f - u_e)$$

with all $C_{ef} \geq 0$ then we can choose Δt such that the following is a convex combination

$$\begin{aligned}
\frac{du^e}{dt} &= \frac{1}{V^e} \sum_f \mathcal{H}(u^e, u^f, \vec{n}) \\
&= \frac{1}{V^e} \sum_f H(u^e, u^f, \vec{n}) - \vec{f}(u) \cdot \vec{n} \\
&= \frac{1}{V^e} \sum_f \frac{H(u^e, u^f, \vec{n}) - \vec{f}(u) \cdot \vec{n}}{u^f - u^e} (u^f - u^e) \\
&= \frac{1}{V^e} \sum_f \frac{\partial \mathcal{H}}{\partial u^-} (u^f - u^e)
\end{aligned}$$

and hence the scheme is *local extrema diminishing (LED)*

DGM/IP methods

Convective terms : finite volume reinterpreted as DGM

DGM formulation

$$\sum_e \int_e v_m \frac{\partial u_m}{\partial t} dV - \sum_e \int_e \nabla v_m \cdot \vec{g}_m dV + \sum_f \int_f \gamma_m(\tilde{u}^+, \tilde{u}^-, v^+, v^-, \vec{n}) dS = 0$$

Choose piecewise constant function space

$$\begin{aligned} v^e &= 1 \quad \forall x \in e \\ v^e &= 0 \quad \forall x \notin e \end{aligned}$$

Then

$$\begin{aligned} V^e \frac{du^e}{dt} + \oint_{\partial e} \gamma(u^e, u^*, 1, 0, \vec{n}) dS &= 0 \ , \ \forall e \\ V^e \frac{du^e}{dt} + \oint_{\partial e} \mathcal{H}(u^e, u^*, \vec{n}) dS &= 0 \end{aligned}$$

Generalisation

$$\sum_e \int_e v_m \cdot \frac{\partial u_m}{\partial t} dV - \sum_e \int_e \nabla v_m \cdot \vec{f}_m dV + \sum_f \int_f [[v_m]] \, \vec{n} \mathcal{H}_m(u^+, u^-, \vec{n}) dS = 0 \ , \ \forall v \in \Phi$$

DGM/IP methods

Convective terms : energy stability of HO version

LED in FVM is weakened to energy stability for DGM

Plug in $v = u$

$$\sum_e \int_e u \frac{\partial u}{\partial t} dV = \sum_e \int_e \nabla u \cdot \vec{f}(u) dV - \sum_f \int_f [\![u]\!] \cdot \vec{n} \mathcal{H}(u^+, u^-, \vec{n}) dS$$

$$\Downarrow \vec{g}(u) = \int^u \vec{f}(u) du$$

$$\frac{\partial}{\partial t} \sum_e \int_e \frac{u^2}{2} dV = \sum_e \int_e \nabla \cdot \vec{g}(u) dV - \sum_f \int_f [\![u]\!] \cdot \vec{n} \mathcal{H}(u^+, u^-, \vec{n}) dS$$

$$= - \sum_f \int_f \left([\![u]\!] \cdot \vec{n} \mathcal{H}(u^+, u^-, \vec{n}) - [\![\vec{g}(u)]\!] \right) dS$$

$$\Downarrow \text{midpoint rule}$$

$$= - \sum_f \int_f \left(u^+ - u^- \right) \left(\mathcal{H}(u^+, u^-, \vec{n}) - \vec{f}(u^*) \cdot \vec{n} \right) dS \ , \ \ u^* \in [u^+, u^-]$$

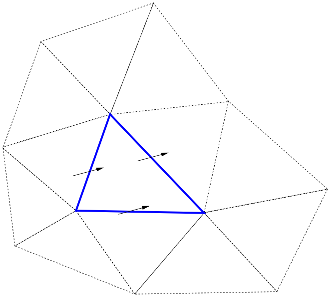
$$\Downarrow \text{E-flux} (\mathcal{H}(u^+, u^-, \vec{n}) - \vec{f}(u) \cdot \vec{n})(u^- - u^+) \leq 0$$

$$\leq 0$$

and a local elementwise entropy inequality (*Jiang* [JS94])

DGM/IP methods

Convective terms : local FEM reinterpretation



For each element e find $u^e \in \Phi(e)$

$$\int_e v^e_m \frac{\partial u^e_m}{\partial t} dV - \int_e \nabla v^e_m \cdot f_m(u^e) dV + \sum_{f \in e} \int_f v^e_m \mathcal{H}_m(u, u^*, \vec{n}) dS = 0 \ , \ \forall v^e_m \in \Phi(e)$$

then we find

- Galerkin FEM problem for each element e
- flux boundary conditions ensure "Dirichlet"-like coupling to the neighbours
- choice of $\mathcal{H}$ ensures stability of the bc
- if $\mathcal{H}$ is upwind flux imposes correct characteristics to/from external state $u^* = u^-$.

DGM/IP methods

Functional analysis : Lax-Milgram theorem

V is a Hilbert space

- Complete vector space
  - $x, y \in V \Rightarrow x + y \in V$
  - $x \in V \Rightarrow \alpha x \in V$
  - any Cauchy sequence converges in V
- Inner product $(.,.)$
  - $(u, v) = (v, u)$
  - $(u + w, v) = (u, v) + (w, v)$
  - $(u, u) \geq 0$

$a(.,.)$ is a *continuous* and *coercive bilinear* form $V \times V \to \mathbb{R}$

- $\exists c_1 > 0 : |a(u, v)| \leq c_1 ||u|| \cdot ||v|| \quad \forall u, v \in V$
- $\exists c_2 > 0 : a(u, u) \geq c_2 ||u||^2 \quad \forall u \in V$
- $a(u + v, w) = a(u, w) + a(v, w)$, $a(u, v + w) = a(u, v) + a(u, w)$

$< f, . >$ is a continuous linear form $V \to \mathbb{R}$

$$\exists c > 0 \ : \ |< f, u >| \ \leq \ c||u||$$

Then the problem

$$a(u, v) =< f, v > , \ \forall v \in V$$

has a unique solution $u \in V$

DGM/IP methods

Functional analysis : Lax-Milgram - illlustration for $\mathbb{R}^n$

Lax-Milgram is sufficient but not necessary condition for solvability (not applicable to convective DGM)

Eg. apply Lax-Milgram to solve for $\mathbf{A}\mathbf{x} = b$, $\mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{x}, \mathbf{b} \in \mathbb{R}^n$

- define inner product $(\mathbf{x}, \mathbf{y}) = \mathbf{y}^T \mathbf{x}$
- define bilinear form $a(\mathbf{x}, \mathbf{y}) = \mathbf{y}^T \mathbf{A}\mathbf{x}$
  - continuity implies $\mathbf{A}$ is *bounded*

$$|\mathbf{y}^T \mathbf{A}\mathbf{x}| = |\sum_i y_i^l \lambda_i \mathbf{l}_i^T \cdot \mathbf{x}| \leq \sum_i |\lambda_i y_i^l x_i^r| \leq |\lambda|_{max} ||x|| ||y||$$

  - coercivity implies $\mathbf{A}$ is *positive definite*

$$\mathbf{x}^T \mathbf{A}\mathbf{x} = \sum_i x_i^l \lambda_i \mathbf{r}_i \cdot \mathbf{x} = \sum_i \lambda_i x_i^l x_i^r \geq \lambda_{min} ||x||^2$$

- define a linear form $f(\mathbf{x}) = \mathbf{x}^T \cdot \mathbf{a}$, continuity implies $\mathbf{a}$ is *finite* : $|f(\mathbf{x})| \leq ||\mathbf{a}|| ||\mathbf{x}||$
- hence $\mathbf{y}^T \mathbf{A}\mathbf{x} = \mathbf{y}^T \cdot \mathbf{a}\ \forall \mathbf{y} \in \mathbb{R}^n$ has a unique solution

DGM/IP methods
Functional analysis : Broken Sobolev spaces

The broken Sobolev space $H^s(\mathcal{E})$ defined by its

- elements

$$H^s(\mathcal{E}) = \{v \in L^2(\Omega) \ : \ v|_e \in H^s(E) \ , \ \forall E \in \mathcal{E}\}$$

- broken norm

$$||u||_{H^s(\mathcal{E})} = \sum_{E \in \mathcal{E}} ||u||_{H^s(E)}$$

- broken inner product

$$(u,v)_{H^s(\mathcal{E})} = \sum_{E \in \mathcal{E}} (u,v)_{H^s(E)}$$

2nd order PDE : use $H^1(\mathcal{E})$

- natural norm :

$$||u||_{H^1()} = \sum_e ||u||_{H^1(e)} = \sum_e (|u|^2_{0,e} + |u|^2_{1,e})$$

- DG energy norm :

$$||u||_{DG} = \sum_e |\nabla u|^2_e + \sum_f |\,[\![u]\!]\,|^2_{0,f}$$

DGM/IP methods

Interior penalty methods : Naive Galerkin for elliptic/parabolic equations

Elliptic problem

$$
\begin{aligned}
\nabla \cdot \mu \nabla \tilde{u} &= f \\
u &= u^* \quad \forall x \in \Gamma_D \\
\partial_n u &= g \quad \forall x \in \Gamma_N
\end{aligned}
$$

Naive DG approach $\forall v \in \Phi$

$$
\begin{aligned}
a(u,v) &= \sum_e \int_e \nabla v \cdot \mu \nabla u \, dV - \sum_f \int_f [[v \mu \nabla u]] \, dS \\
&= \sum_e \int_e \nabla v \cdot \mu \nabla u \, dV - \sum_f \int_f [[v]] \cdot \{\{\nabla u\}\} + \cancel{[[\mu \nabla u]] \cdot \{\{v\}\}} \, dS \\
&= \sum_e \int_e \nabla v \cdot \mu \nabla u \, dV - \sum_f \int_f [[v]] \cdot \{\{\nabla u\}\} \, dS \forall v \in \Phi
\end{aligned}
$$

$$
\Rightarrow \exists v \in \Phi : a(v,v) \not> 0
$$

Conclusions :

- – is not coercive and hence unique solution is not guaranteed
- \+ however DG allows consistent stabilisation using solution jumps

DGM/IP methods

Interior penalty methods : Baumann-Oden (BO)

Bilinear form compensates consistent interface term

$$a(u,v) = \sum_e \int_e \nabla v \cdot \nabla u \, dV - \sum_f \int_f \left([\![v]\!] \cdot \{\!\{\nabla u\}\!\} - [\![u]\!] \cdot \{\!\{\nabla v\}\!\} \right) dS \ , \ \ u, v \in V_h$$

Coercivity ?

$$a(v,v) = \sum_e \int_e |\nabla v|^2 \, dV - \sum_f \int_f \left([\![v]\!] \cdot \{\!\{\nabla v\}\!\} - [\![v]\!] \cdot \{\!\{\nabla v\}\!\} \right) dS \quad = \sum_e \int_e |\nabla v|^2 \, dV \ , \ \ \forall u, v \in V_h$$

Conclusions

- \+ very natural way for stabilisation
- – not stable for pure diffusion (constant functions) since only larger than seminorm
- – formulation is not symmetric
  - non-symmetric Krylov iterator (`BiCG/GMRES`) instead of `CG`
  - convergence of stationary methods (`Jacobi/GS/SOR/...`)

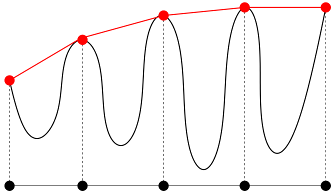
DGM/IP methods

Interior penalty methods : boundary penalty methods

Nitsche 71

Elliptic problem with rough Dirichlet bc

$$
\begin{aligned}
\nabla \cdot \mu \nabla u &= 0 \quad \forall x \in \Omega \\
u &= g \quad \forall x \in \partial\Omega
\end{aligned}
$$



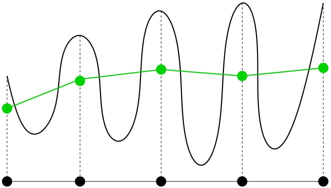
DGM/IP methods

Interior penalty methods : boundary penalty methods

Nitsche 71

Elliptic problem with rough Dirichlet bc

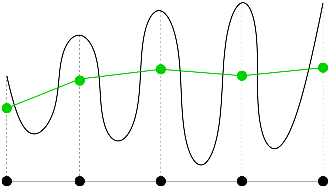
$$
\begin{aligned}
\nabla \cdot \mu \nabla u &= 0 \quad \forall x \in \Omega \\
u &= g \quad \forall x \in \partial\Omega
\end{aligned}
$$



DGM/IP methods

Interior penalty methods : boundary penalty methods

Nitsche 71



Elliptic problem with rough Dirichlet bc

$$\begin{aligned} \nabla \cdot \mu \nabla u = 0 \quad \forall x \in \Omega \\ u = g \quad \forall x \in \partial\Omega \end{aligned}$$

Penalty bc
+ consistency term - conditional stability ifo $\sigma \sim \mu C/h$

$$\int_\Omega \nabla u \cdot \mu \nabla v dV + \int_{\partial\Omega} \sigma (u-g) v dS - \int_{\partial\Omega} v \mu \nabla u \cdot \vec{n} dS = 0$$

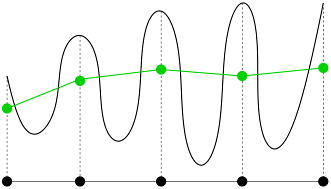
DGM/IP methods

Interior penalty methods : boundary penalty methods

Nitsche 71

Elliptic problem with rough Dirichlet bc

$$\begin{aligned} \nabla \cdot \mu \nabla u = 0 \quad \forall x \in \Omega \\ u = g \quad \forall x \in \partial\Omega \end{aligned}$$



Penalty bc
+ consistency term - conditional stability ifo $\sigma \sim \mu C/h$

$$\int_\Omega \nabla u \cdot \mu \nabla v dV + \int_{\partial\Omega} \sigma (u-g) v dS - \int_{\partial\Omega} v \mu \nabla u \cdot \vec{n} dS = 0$$

Symmetrizing variant - conditional stability ifo $\sigma \sim C/h$

$$\int_\Omega \nabla u \cdot \mu \nabla v dV + \int_{\partial\Omega} \sigma (u-g) v dS - \int_{\partial\Omega} (v \mu \nabla u + (u-g) \mu \nabla v) \cdot \vec{n} dS = 0$$

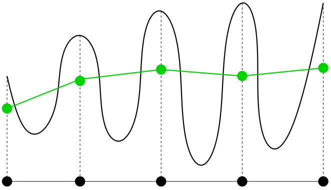
DGM/IP methods

Interior penalty methods : boundary penalty methods

Nitsche 71

Elliptic problem with rough Dirichlet bc

$$\nabla \cdot \mu \nabla u = 0 \quad \forall x \in \Omega$$
$$u = g \quad \forall x \in \partial\Omega$$



Penalty bc
+ consistency term - conditional stability ifo $\sigma \sim \mu C/h$

$$\int_\Omega \nabla u \cdot \mu \nabla v dV + \int_{\partial\Omega} \sigma (u-g) v dS - \int_{\partial\Omega} v \mu \nabla u \cdot \vec{n} dS = 0$$

Symmetrizing variant - conditional stability ifo $\sigma \sim C/h$

$$\int_\Omega \nabla u \cdot \mu \nabla v dV + \int_{\partial\Omega} \sigma (u-g) v dS - \int_{\partial\Omega} (v \mu \nabla u + (u-g) \mu \nabla v) \cdot \vec{n} dS = 0$$

Antisymmetric variant - stability for all $\sigma > 0$

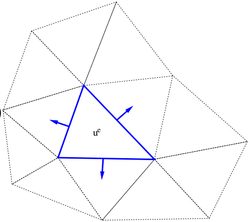
$$\int_\Omega \nabla u \mu \cdot \nabla v dV + \int_{\partial\Omega} \sigma (u-g) v dS - \int_{\partial\Omega} (v \mu \nabla u - \mu (u-g) \nabla v) \cdot n dS = 0$$

DGM/IP methods

Interior penalty methods : Interior Penalty Method - local view point

Local problem : for each element e find $u^e \in \Phi(e)$

$$\int_e \nabla v^e \cdot \nabla u^e dV + \int_{\partial e} \sigma v^e (u^e - u^o) dS$$
$$- \int_{\partial e} v^e \nabla u^e + \theta (u^e - u^*) \nabla v^e \vec{n} dS = 0 \ , \ \forall v^e \in \Phi(e)$$



with Nitsche penalties for coupling boundary conditions
Global problem : find $u \in \Phi$

$$\sum_e \int_e \nabla v \cdot \mu \nabla u dV + \sum_f \sigma \int_f [[u]] \, [[v]] \, dS$$
$$- \sum_f \int_f [[v]] \, \{\{\mu \nabla u\}\} + \cancel{[[\mu \nabla u]] \, \{\{v\}\}} dS$$
$$- \theta \sum_f \int_f [[u]] \, \{\{\mu \nabla v\}\} \, dS$$

DGM/IP methods

Interior penalty methods : properties

$$\sum_e \int_e \nabla v \cdot \mu \nabla u dV + \sum_f \sigma \int_f [[u]] \, [[v]] \, dS$$
$$- \sum_f \int_f [[v]] \, \{\{\mu \nabla u\}\} + \cancel{[[\mu \nabla u]] \, \{\{v\}\}} dS$$
$$- \theta \sum_f \int_f [[u]] \, \{\{\mu \nabla v\}\} \, dS$$

- *theta* = 1 *Non-Symmetric Interior Penalty (SIP)* - symmetric, conditionnally stable ($\sigma > \sigma_c$)
- *theta* = −1 *Incomplete Interior Penalty (NIP)* - antisymmetric, marginally stable ($\sigma > 0$)

Description Rivière [Riv08]
Relation to lifting based methods *Arnold et al. [ABCM02]*
Question : how do we choose σ for SIP

DGM/IP methods

Interior penalty methods : Coercivity of SIP [Sha05]

$$
\begin{aligned}
a(v,v) &= \sum_e \int_e |\nabla v|^2 \, dV - 2\sum_f \int_f \{\!\{\nabla v\}\!\} [\![v]\!] \, dS + \sum_f \sigma_f \int_f [\![v]\!]^2 \, dS > C_1 ||v||^2 \quad ? \\
&\geq \sum_e \int_e |\nabla v|^2 \, dV - \sum_f \frac{1}{\epsilon_F} \int_f \{\!\{\nabla v\}\!\}^2 \, dS + \sum_f (\sigma_f - \epsilon_f) \int_f [\![v]\!]^2 \, dS \\
&\geq \sum_e \int_e |\nabla v|^2 \, dV - \sum_f \frac{1}{4\epsilon_F} \int_f |\nabla v^+|^2 + |\nabla v^-|^2 + 2\nabla v^- \cdot \nabla v^+ \, dS + \dots \\
&\geq \sum_e \int_e |\nabla v|^2 \, dV - \sum_f \frac{1}{2\epsilon_F} \int_f |\nabla v^+|^2 + |\nabla v^-|^2 \, dS - \sum_{f \in \Gamma} \frac{1}{\epsilon_F} \int_f |\nabla v^-|^2 \, dS + \dots \\
&\geq \sum_e \left(1 - \sum_{f \in e} \frac{c^*_{f,e}}{\epsilon_f} \right) \int_e |\nabla v|^2 \, dV + \sum_f \int_f (\sigma_f - \epsilon_f) [v]^2 \, dS
\end{aligned}
$$

$$
\begin{aligned}
c^*_{f,e} &= c_{\mathfrak{f},\mathfrak{e}} \frac{\mathcal{A}(f)}{\mathcal{V}(e)} \quad , \quad \forall f \in \Gamma \\
&= \frac{c_{\mathfrak{f},\mathfrak{e}}}{2} \frac{\mathcal{A}(f)}{\mathcal{V}(e)} \quad , \quad \forall f \notin \Gamma
\end{aligned}
$$

DGM/IP methods

Interior penalty methods : Trace inequality constants

$e/\ f$	edge	triangle	quadrilateral
triangle*	$(p+1)(p+2)/2$	-	-
tetrahedron*	-	$(p+1)(p+3)/3$	-
quadrilateral†	$(p+1)^2$	-	-
hexahedron†	-	-	$(p+1)^2$
wedge†	-	$(p+1)^2$	$(p+1)(p+2)/2$
pyramid†	-	$1.05(p+1)(2p+3)/3$	$(p+1)(p+3)/3$

$$\int_f u^2 dS \leq c_{\mathfrak{e},\mathfrak{f}}(p) \cdot \frac{\mathcal{A}(f)}{\mathcal{V}(e)} \int_e u^2 dV \ , \ \forall u \in \Phi_p$$

Hillewaert & Remacle, submitted to Sinum

DGM/IP methods

Interior penalty methods : Coercivity of SIP - alternatives for σ

Choose ϵ_f and σ_f such that

$$a(v,v) \geq \sum_e \left(1 - \sum_{f \in e} \frac{c^*_{f,e}}{\epsilon_f}\right) \int_e |\nabla v|^2 dV + \sum_f \int_f (\sigma_f - \epsilon_f)[v]^2 dS$$

DGM/IP methods

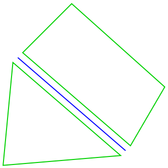
Interior penalty methods : Coercivity of SIP - alternatives for σ

Choose ϵ_f and σ_f such that

$$a(v,v) \geq \sum_e \left(1 - \sum_{f \in e} \frac{c^*_{f,e}}{\epsilon_f} \right) \int_e |\nabla v|^2 dV + \sum_f \int_f (\sigma_f - \epsilon_f) [v]^2 dS$$

Generalisation of Shahbazi (05)

$$\begin{aligned}
\sigma_f &> \epsilon_f \\
\epsilon_f &> \max_{e \ni f} \left(\sum_{f' \in e} c^*_{f',e} \right) = \max_{e \ni f} \left(\frac{1}{\mathcal{V}(e)} \sum_{f \in e} c_{f',e} \mathcal{A}(f') \right)
\end{aligned}$$



DGM/IP methods

Interior penalty methods : Coercivity of SIP - alternatives for σ

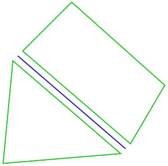
Choose ϵ_f and σ_f such that

$$a(v,v) \geq \sum_e \left(1 - \sum_{f\in e} \frac{c^*_{f,e}}{\epsilon_f}\right) \int_e |\nabla v|^2 dV + \sum_f \int_f (\sigma_f - \epsilon_f)[v]^2 dS$$

Generalisation of Shahbazi (05)

$$\sigma_f > \epsilon_f$$

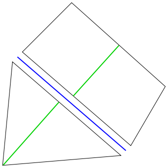
$$\epsilon_f > \max_{e\ni f}\left(\sum_{f'\in e} c^*_{f',e}\right) = \max_{e\ni f}\left(\frac{1}{\mathcal{V}(e)} \sum_{f\in e} c_{\mathfrak{f}',\mathfrak{e}} \mathcal{A}(f')\right)$$



Anisotropic definition

$$\sigma_f > \epsilon_f$$

$$\epsilon_f > \max_{e\ni f}\left(n c^*_{f,e}\right) = \max_{e\ni f}\left(n c_{\mathfrak{f},\mathfrak{e}} \frac{\mathcal{A}(f)}{\mathcal{V}(e)}\right)$$



DGM/IP methods

Interior penalty methods : verification

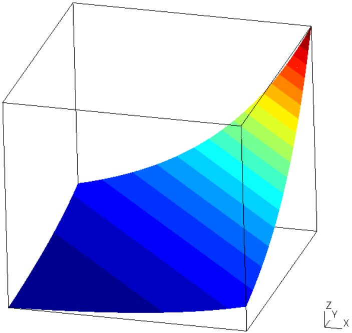
- manufactured solution

$$\Delta u = -\Delta f \ , \ \forall x \in \Omega$$
$$u = f \ , \ \forall x \in \Gamma$$
$$f = \prod_{i=1}^{d} e^{x_i}$$

- define

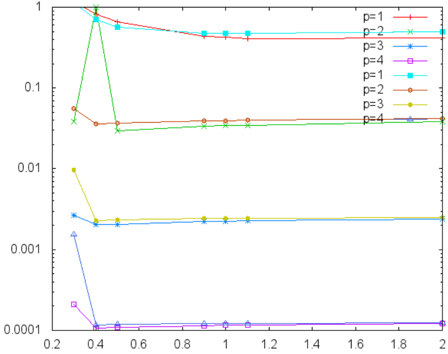
$$\sigma_f = \alpha \sigma_f^*$$

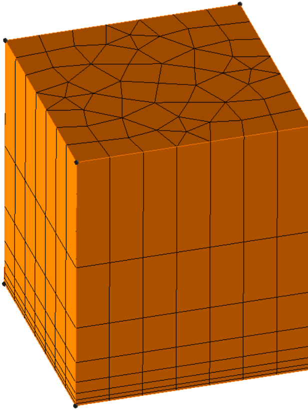
- plot
  L_2-norm of the error as a function of α
- single-precision
  direct solver ← conditioning



DGM/IP methods

Interior penalty methods : verification





DGM/IP methods

Interior penalty methods : extension to systems

Scalar penalty

$$\sum_e \int_e \frac{\partial v_m}{\partial x_k} \cdot \mathbf{D}^{kl}_{mn} \frac{\partial u_n}{\partial x_l}\, dV - \sum_f \int_f [[v_m]]^k \left\{\!\!\left\{ \mathbf{D}^{kl}_{mn} \cdot \frac{\partial u_n}{\partial x^l} \right\}\!\!\right\} dS$$
$$- \theta \sum_f \int_f [[u_n]]^k \left\{\!\!\left\{ \mathbf{D}^{kl}_{nm} \cdot \frac{\partial v_m}{\partial x^l} \right\}\!\!\right\} dS$$
$$+ \sum_f \sigma \int_f [[u_m]] \cdot [[v_m]]\, dS = 0$$

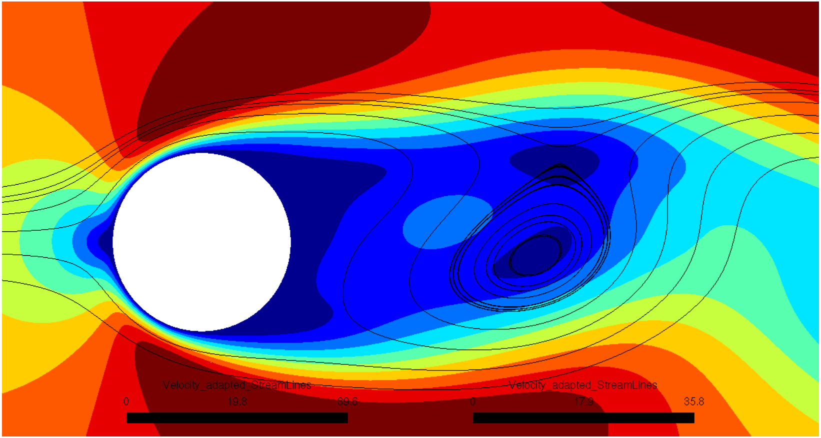
Matrix penalty

$$\sum_e \int_e \frac{\partial v_m}{\partial x_k} \cdot \mathbf{D}^{kl}_{mn} \frac{\partial u_n}{\partial x_l}\, dV - \sum_f \int_f [[v_m]]^k \left\{\!\!\left\{ \mathbf{D}^{kl}_{mn} \cdot \frac{\partial u_n}{\partial x^l} \right\}\!\!\right\} dS$$
$$- \theta \sum_f \int_f [[u_n]]^k \left\{\!\!\left\{ \mathbf{D}^{kl}_{nm} \cdot \frac{\partial v_m}{\partial x^l} \right\}\!\!\right\} dS$$
$$+ \sum_f \frac{\sigma^*}{2} \int_f \left([[v_m]]^l \left(\mathbf{D}^{kl}_{mn} + \mathbf{D}^{kl}_{nm} \right) [[u_n]]^k \right) dS = 0$$

DGM/IP methods

Interpolation and quadrature : need for curved meshes and mappings

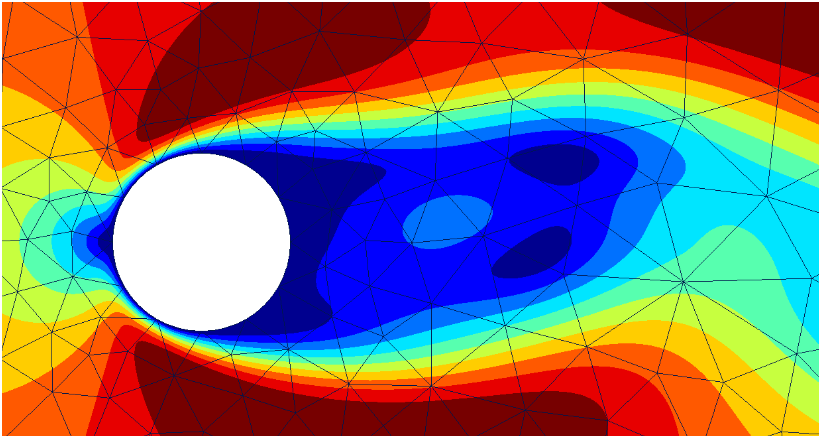
Example : von Karman street from cylinder Re=100, DGM(4)



DGM/IP methods

Interpolation and quadrature : need for curved meshes and mappings

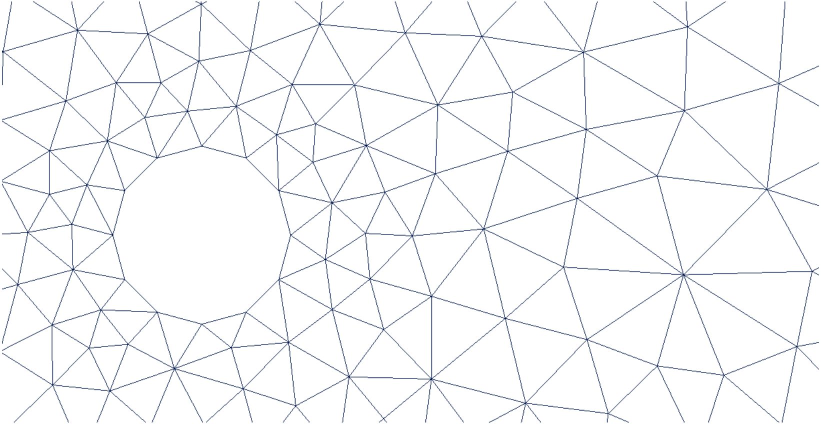
Example : von Karman street from cylinder Re=100, DGM(4)



DGM/IP methods

Interpolation and quadrature : need for curved meshes and mappings

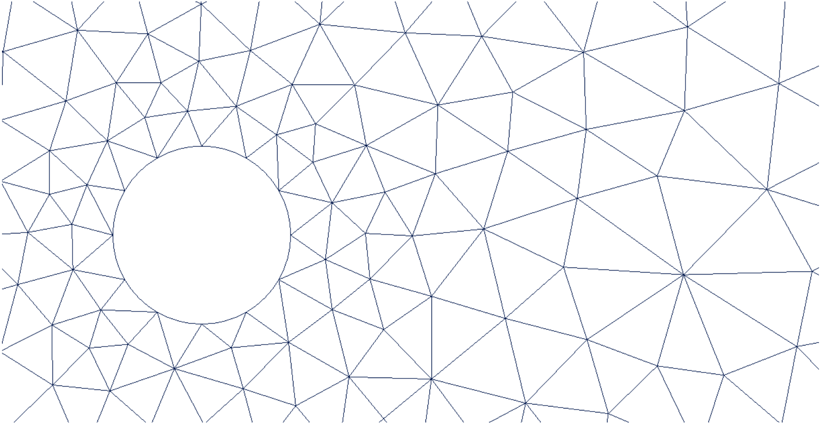
Example : von Karman street from cylinder Re=100, DGM(4)



DGM/IP methods

Interpolation and quadrature : need for curved meshes and mappings

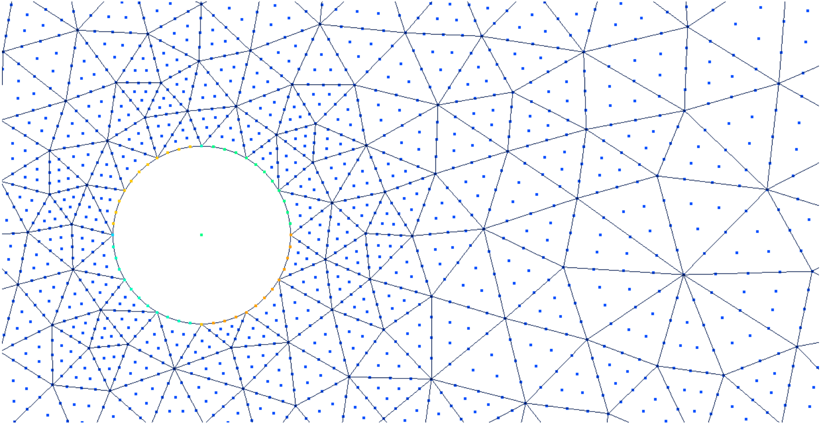
Example : von Karman street from cylinder Re=100, DGM(4)



DGM/IP methods

Interpolation and quadrature : need for curved meshes and mappings

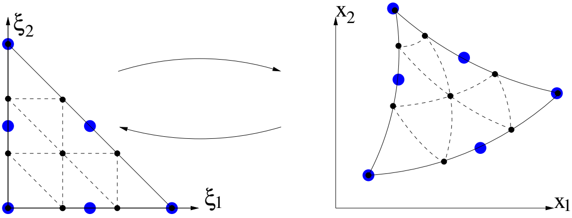
Example : von Karman street from cylinder Re=100, DGM(4)



Pre- and postprocessing tools are (not enough) subject of research
Gmsh (http ://www.geuz.org/gmsh)

DGM/IP methods

Interpolation and quadrature : Taxonomy of base functions



DGM : basis ϕ_i for Φ is supported on a single element $\rightarrow$ total freedom

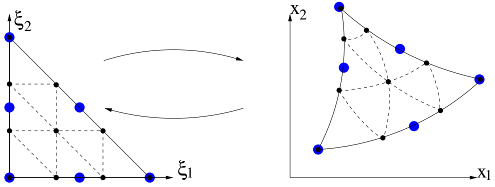
- *modal* : easy/well-conditioned base irrespective of geometry
  - monomials $1, \xi, \xi^2, \ldots$
  - orthogonal polynomials, eg. Legendre in 1D $\mathcal{P}^n(\xi)$
  - fundamental solutions : eg. plane waves
  - ...
- *nodal* : control points associated to the geometry
  - Lagrangian (equidistant, optimised, spectral elements)
  - Splines

Coordinate system

- Parametric
- Cartesian

DGM/IP methods

Interpolation and quadrature : Standard choice : parametric Lagrangian interpolation



Solutions and coordinates expanded in parametric coordinates $\boldsymbol{\xi}$

$$u_m = \sum_{i=1}^{N_\phi} \mathbf{u}_{im} \phi_i(\boldsymbol{\xi}) \qquad x^k = \sum_{i=1}^{N_\psi} \mathbf{x}_i^k \psi_i(\boldsymbol{\xi})$$

Jacobian $\mathbf{J}$ of $\mathbf{x}$ wrt $\boldsymbol{\xi}$

$$\mathbf{J}_{kl} = \frac{\partial x^k}{\partial \xi^l} = \sum_{i=1}^{N_c} \frac{\partial \psi_i}{\partial \xi^l} \qquad \left(\mathbf{J}^{-1}\right)_{kl} = \frac{\partial \xi^k}{\partial x^l}$$

Classical Gauss-Legendre quadrature $\mathcal{O}(2p+1)$

$$\int_V \nabla \phi_i (\mathbf{f}_c(u) + \mathbf{f}_d(u, \nabla u)) dV \approx \sum_{q=1}^{N_q} w_q \left(\frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}_{kl}^{-1} \left(f_c^l(u) + f_d^l(u, \nabla u) \right) |\mathbf{J}| \right)_{\xi_q}$$

DGM/IP methods

Interpolation and quadrature : Lagrangian boundary closures

Suppose

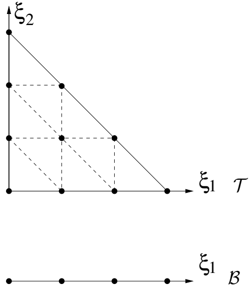
- $\Lambda(\psi_i, \xi_i) = \{\lambda_i\}$
- suppose $span\{\psi_k\}_f = span\{\psi_k^f\}$
- $\mathbf{V}^f$ is invertible, with $\mathbf{V}_{ij}^f = \psi_i^f(\xi_j^f)$

For any λ_i

$$\lambda_i\big|_f = \sum_j \beta_{ij}\psi_j^f = \sum_{j \in \Xi^f} \beta_{ij}\lambda_j^f$$

$$\lambda_i(\xi_k) = 0$$

$$= \sum_j \beta_{ij}\lambda_j^f(\xi_k) = \beta_{ik} \ , \ \ \forall \xi_k \in \Xi^f$$



and hence

$$\lambda_k\big|_f = 0 \ , \ \ \forall \xi_k \notin f$$

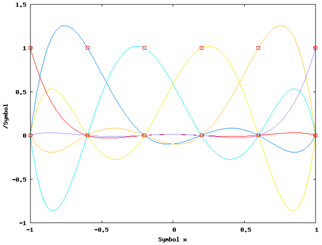
$\Rightarrow$ whatever the basis ψ_i, Lagrangian elements with complete boundary spaces will result interpolations on the boundary that only depend on the interpolation nodes on that same boundary

- $\mathcal{C}^0$ continuity (mesh generation)
- efficient assembly

DGM/IP methods

Interpolation and quadrature : computation

Lagrangian interpolants λ_i based on points ξ_i
and whatever set of basis functions $\psi_i : \Phi = \text{span}(\psi_i)$



$$\lambda_i \in \Phi : \lambda_i\left(\xi_j\right) = \delta_{ij}$$

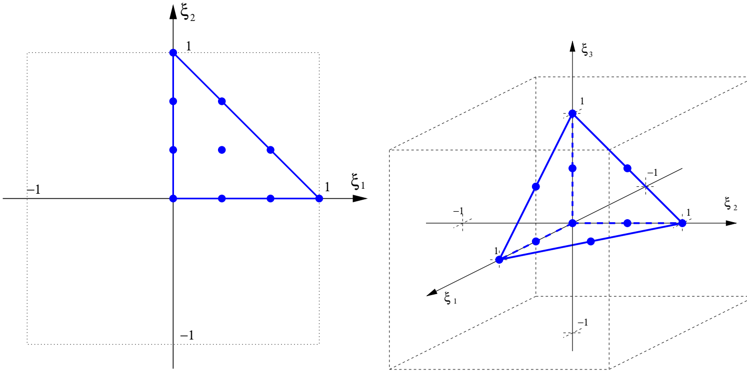
$$\lambda_i = \sum_j \mathbf{A}_{ij} \psi_j$$

$$\underbrace{\begin{bmatrix} \lambda_1(\xi_1) & \lambda_1(\xi_2) & \dots & \lambda_1(\xi_n) \\ \lambda_2(\xi_1) & \lambda_2(\xi_2) & \dots & \lambda_2(\xi_n) \\ \dots & & & \\ \lambda_n(\xi_1) & \lambda_n(\xi_2) & \dots & \lambda_n(\xi_n) \end{bmatrix}}_{I} = \mathbf{A} \cdot \underbrace{\begin{bmatrix} \psi_1(\xi_1) & \psi_1(\xi_2) & \dots & \psi_1(\xi_n) \\ \psi_2(\xi_1) & \psi_2(\xi_2) & \dots & \psi_2(\xi_n) \\ \dots & & & \\ \psi_n(\xi_1) & \psi_n(\xi_2) & \dots & \psi_n(\xi_n) \end{bmatrix}}_{\mathbf{V}}$$

$$\Rightarrow \mathbf{A} = \mathbf{V}^{-1}$$

DGM/IP methods

Interpolation and quadrature : simplex templates

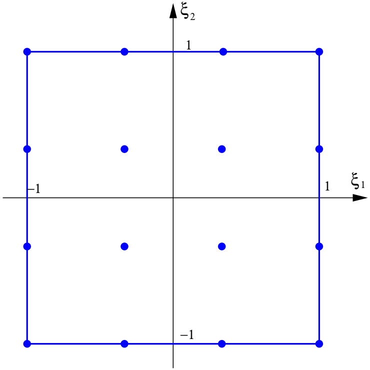


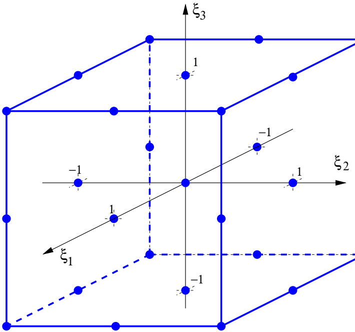
Functional space $\mathbb{P}^d_p = \mathrm{span}\{\prod_{i=1}^d \xi_i^{p_i} : 0 \leq \sum_{i=1}^p p_i \leq p\}$

Compendium of quadrature rules in *Solin 2004* [SSD04]

DGM/IP methods

Interpolation and quadrature :tensor product element templates





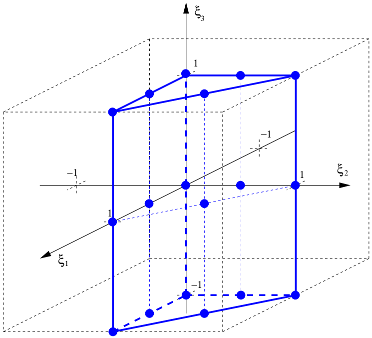
Functional space $\mathbb{Q}_p^d = \mathrm{span}\{\prod_{i=1}^d \xi_i^{p_i} : 0 \leq p_i \leq p\}$

Quadrature rules : tensor product of 1D Gauss-Legendre

Caveat : optimised quadrature rules (Solin) often apply to Pascal space

DGM/IP methods

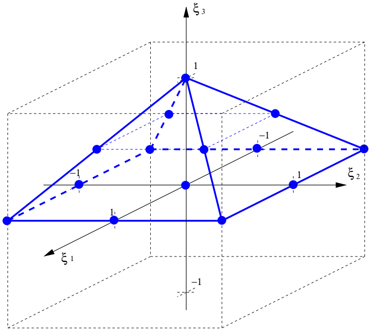
Interpolation and quadrature : prisms



Functional space and quadrature : Tensor product of triangle and lines

DGM/IP methods

Interpolation and quadrature :transition element templates



Bergot et al. [BCD10] boundary compliant functional space

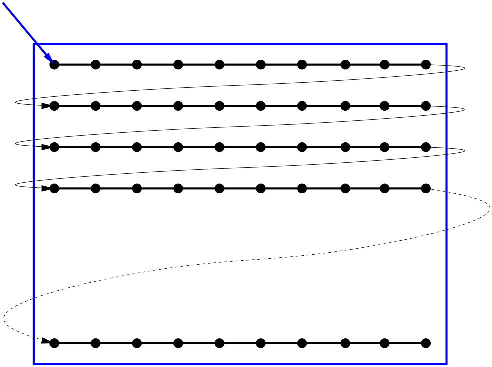
$$
\begin{aligned}
\Phi^e &= \operatorname{span}\left\{\psi_{ijk}, 0 \leq i,j \leq p \ , \ 0 \leq k \leq p - \mu_{ij}\right\} \\
\psi_{ijk} &= \mathcal{P}_i\left(\frac{\xi_1}{1-\xi_3}\right)\ \mathcal{P}_j\left(\frac{\xi_2}{1-\xi_3}\right)\ (1-\xi_3)^{\mu_{ij}}\ \mathcal{P}_k^{2\mu_{ij}+2,0}\left(2\xi_3 - 1\right) \\
\mu_{ij} &= \max\left(i,j\right)
\end{aligned}
$$

Quadrature rules based on degenerated hex

Outline

Practical implementation

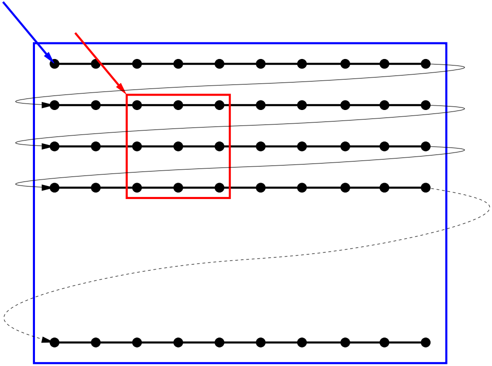
Computational kernels : matrix and vector proxies



$$\mathbf{A} \in \mathbb{R}^{m \times n} = (\mathbf{a}, n, m, lda)$$
$$\mathbf{A}_{ij} = *(\mathbf{a} + i * n + j)$$
$$\mathbf{A}_{ij} = *(\mathbf{a} + i * lda + j)$$

Practical implementation

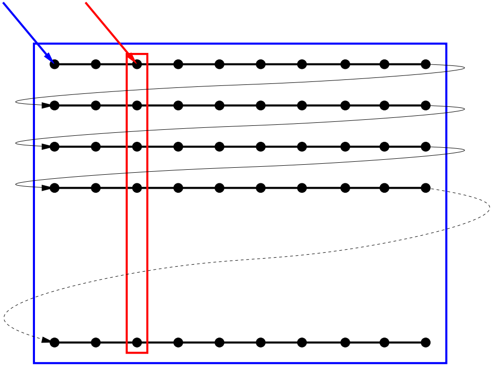
Computational kernels : matrix and vector proxies



$$\mathbf{B} \in \mathbb{R}^{p \times q} = (\mathbf{b}, p, q, lda)$$
$$\mathbf{b} = \mathbf{a} + i_b * lda + j_b$$
$$\mathbf{B}_{ij} = *(\mathbf{b} + i * lda + j) = *(\mathbf{a} + (i + i_b) * n + (j + j_b))$$

Practical implementation

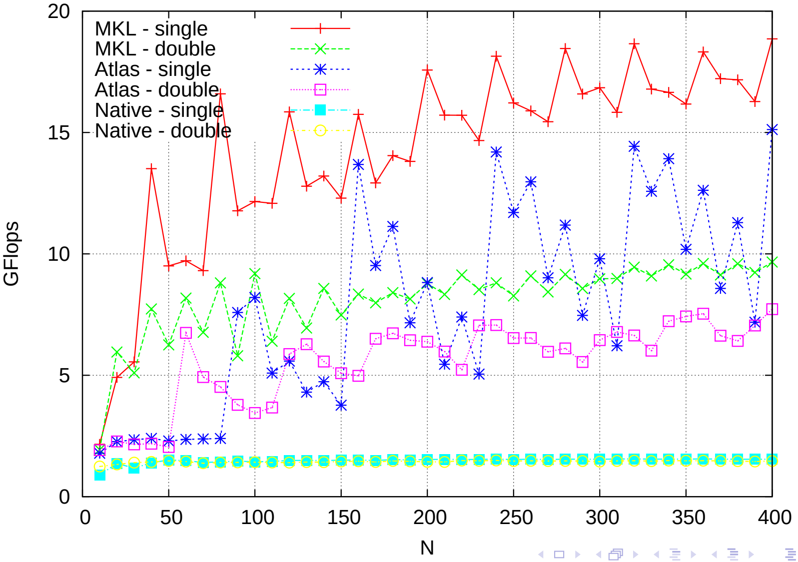
Computational kernels : matrix and vector proxies



$$\mathbf{b} \in \mathbb{R}^p = (\mathbf{b}, p, lda)$$
$$\mathbf{b} = \mathbf{a} + i_b * lda$$
$$\mathbf{b}_i = *(\mathbf{b} + i * lda) = *(\mathbf{a} + (i + i_b) * n$$

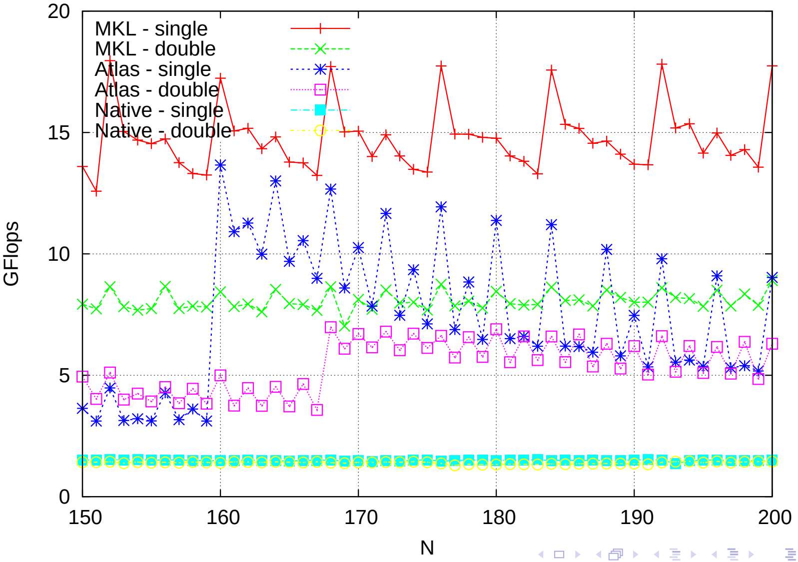
Practical implementation

Computational kernels : BLAS GEMM



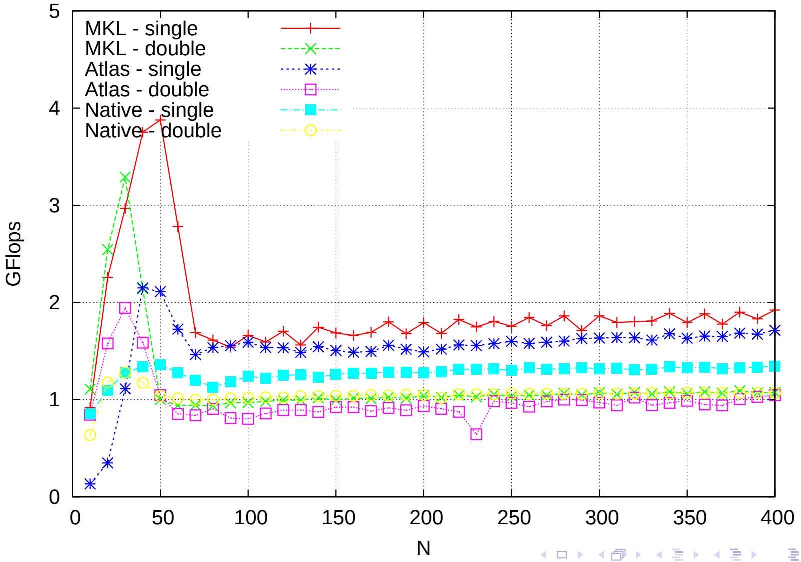
Practical implementation

Computational kernels : BLAS GEMM



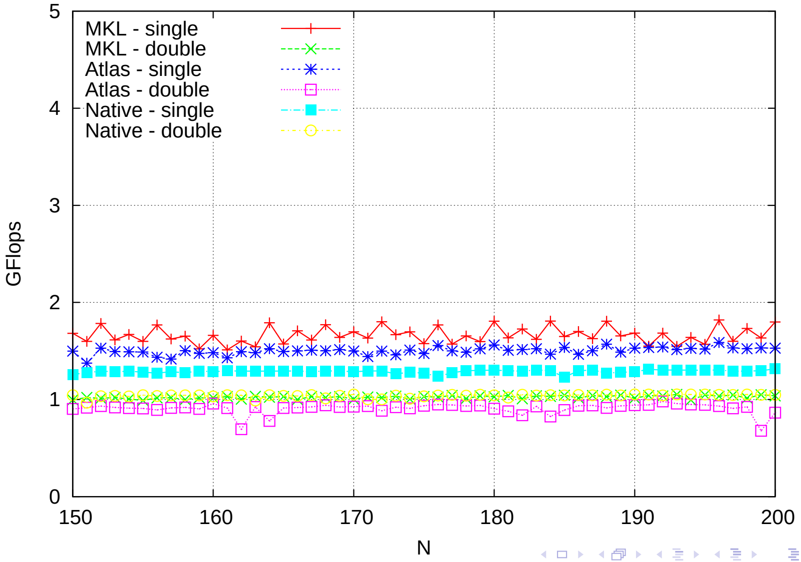
Practical implementation

Computational kernels : BLAS GEMV



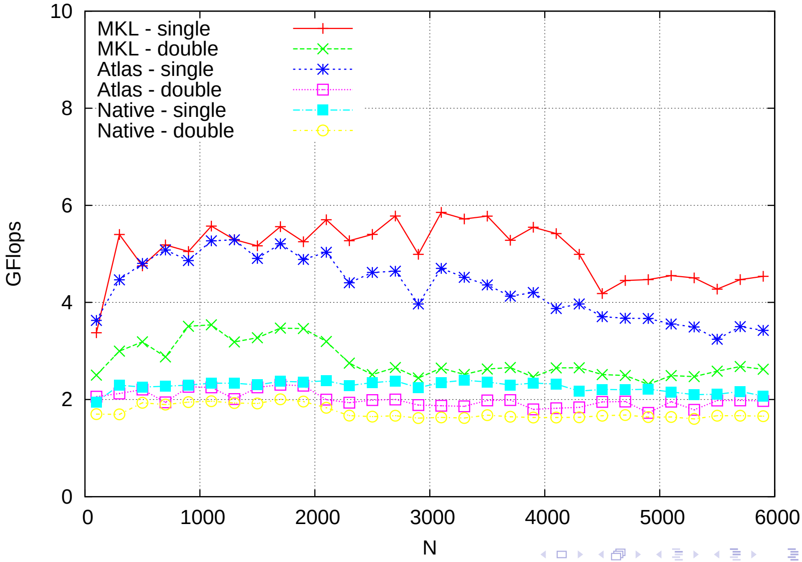
Practical implementation

Computational kernels : BLAS GEMV



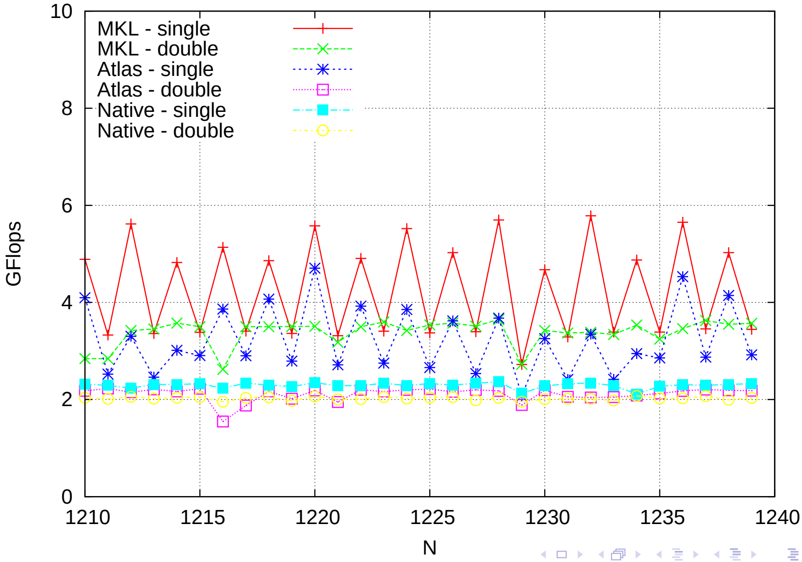
Practical implementation

Computational kernels : BLAS AXPY



Practical implementation

Computational kernels : BLAS AXPY



Practical implementation

Practical implementation : lessons

- peak flop rate : multiple of clock-speed due to inherent vectorisation (*SIMD*)
  - AMD, Intel $\leq$ Harpertown : SSE4 - 4 double, 8 single
  - Intel Sandy Bridge : AVC - 8 double, 16 single
  - BG/P 4 double (fma)
  - BG/Q 8 double (fma)

  requires efficient pipelining (data alignment and cache)
- efficiency increases with BLAS level ~ cache effects and pipelining effects - work to memory $\sim n^l/n$
- data packing effects clearly visible in efficiency → interest for padding
- efficiency depends very much on library

Practical quadrature

Full quadrature

eg. volume terms

Classical Gauss-Legendre quadrature $\mathcal{O}(2p+1)$

$$\mathbf{r}_{im} \leftarrow \mathbf{r}_{im} + \int_V \nabla\phi_{im}(\vec{f}_m(u) + \vec{d}_m(u,\nabla u))dV \approx \mathbf{r}_{im} + \sum_{q=1}^{N_q} w_q \left(\frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}^{-1}_{kl} \left(f^l_c(u) + f^l_d(u,\nabla u) \right) |\mathbf{J}| \right)_{\xi_q}$$

efficient implementation : split up in *parametric* and *physical* steps :

1. Collocation

$$u_m(\xi_q) = \sum_{i=1}^{N_\phi} \phi_i(\xi_q)\mathbf{u}_{im}$$

$$\left(\frac{\partial u_m}{\partial \xi^l} \right)_{\xi_q} = \sum_{i=1}^{N_\phi} \mathbf{u}_{im} \left(\frac{\partial \phi_i}{\partial \xi^l} \right)_{\xi_q}$$

2. Evaluation of geometry and physics

$$\left(\frac{\partial u_m}{\partial x^k} \right)_{\xi_q} = \sum_{l=1}^{d} \left(\mathbf{J}^{-1}_{lk} \frac{\partial u_m}{\partial \xi^l} \right)_{\xi_q}$$

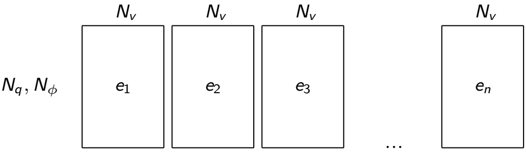
$$\mathfrak{f}^k_{qm} = |\mathbf{J}|\mathbf{J}^{-1}_{kl} \left(f^k_{c,m}(u) + f^k_{d,m}(u,\nabla u) \right)_{\xi_q}$$

3. Flux redistribution

$$\mathbf{r}_{im} \leftarrow \mathbf{r}_{im} + \sum_{q=1}^{N_q} w_q \left(\frac{\partial \phi_i}{\partial \xi^k} \right)_{\xi_q} \mathfrak{f}^k_{qm}$$

Practical implementation

Practical quadrature :Matrix operations in parametric



Solution collocation

$$u_m(\xi_q) = \sum_{i=1}^{N_\phi} \phi_i(\xi_q)\mathbf{u}_{im} \qquad \mathfrak{u}_{qm} = \sum_{i=1}^{N_\phi} \mathbf{C}_{qi}\mathbf{u}_{im}$$

Gradient collocation

$$\left(\frac{\partial u_m}{\partial \xi^l}\right)_{\xi_q} = \sum_{i=1}^{N_\phi} \mathbf{u}_{im}\left(\frac{\partial \phi_i}{\partial \xi^l}\right)_{\xi_q} \qquad \mathfrak{g}^l_{qm} = \sum_{i=1}^{N_\phi} \mathfrak{G}^l_{qi}\mathbf{u}_{im}$$

Flux redistribution

$$\mathbf{r}_{im} + = \sum_{q=1}^{N_q} w_q \left(\frac{\partial \phi_i}{\partial \xi^k}\right)_{\xi_q} \mathfrak{f}^k_{qm} \qquad \mathbf{r}_{im} + = \sum_k \sum_q \mathfrak{R}^k_{iq}\mathfrak{f}^k_{qm}$$

Premultiplication with the mass matrix

$$\mathbf{r}^*_{im} + = \mathfrak{M}_{ij}\mathbf{r}_{jm}$$

Practical implementation

Implicit solver : Damped inexact Newton

Backward-Euler with one Newton solve

$$\mathbf{r}^{\star}_{im} = \left(\phi_i, \frac{u^n_m - u^{n-1}_m}{\Delta\tau^n} + \mathcal{L}_m\left(u^n\right)\right) = 0 \ , \ \forall m, \ \forall \phi_i \in \Phi$$

$$\mathbf{A}^{\star} \cdot \Delta\mathbf{u}^n = -\mathbf{r}^{\star}$$

$$\mathbf{A}^{\star} = \frac{\partial \mathbf{r}^{\star}}{\partial \mathbf{u}} = \frac{\mathbf{M}}{\Delta\tau^n} + \mathbf{A}$$

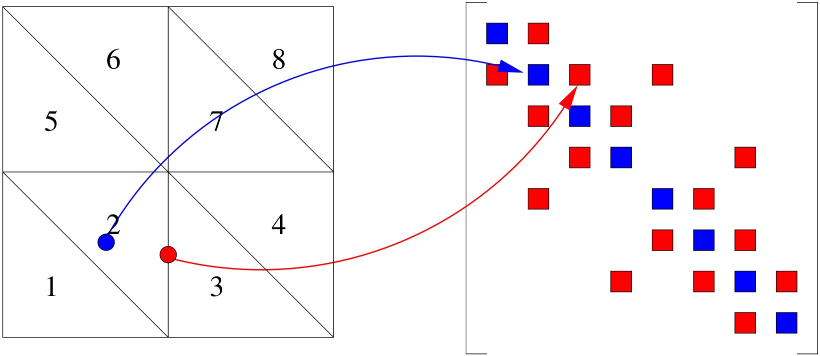
Strategy for global CFL

$$\Delta\tau^n = CFL^n \frac{\Delta x}{u \cdot (2p+1)}$$

$$CFL^n = CFL^0 \cdot \left(\frac{||\mathbf{r}^o||_2}{||\mathbf{r}^{n-1}||_2}\right)^{\alpha}$$

Options

- direct solvers (Gauss)
- Matrix iterative solvers - Krylov subspace



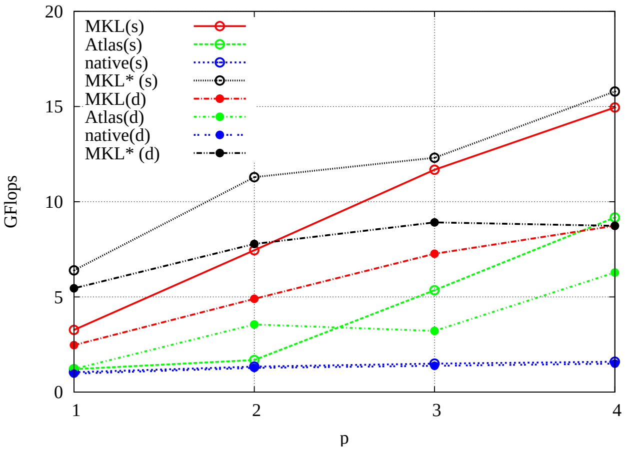
block CSR structure

+ very low indexing overhead allowing simple and flexible datastructure
+ computations recastable in dense gemm, inversion and gemv
    - datastructure can be deallocated/allocated on the fly
    - internal renumbering independent of the mesh
- large block size $N_\phi N_v \sim p^3 N_v$ (eg. DGM(4) hex, Navier-Stokes : 625) → memory bottle-neck

Implicit solver

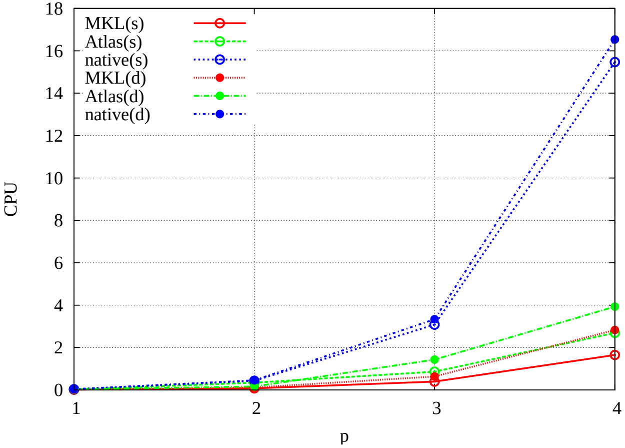
Matrix operations efficiency

inversion



Implicit solver
Matrix operations efficiency

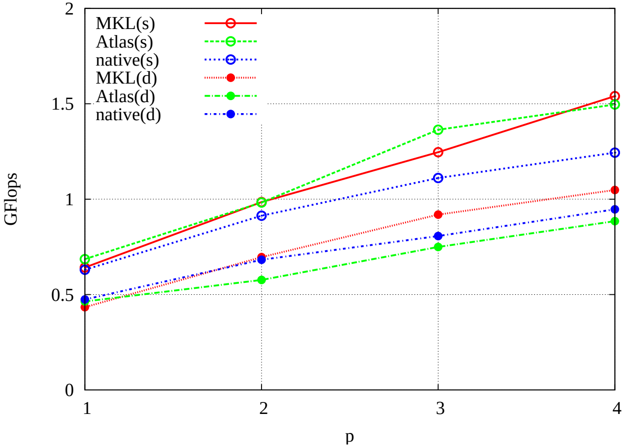
inversion



Implicit solver

Matrix operations efficiency

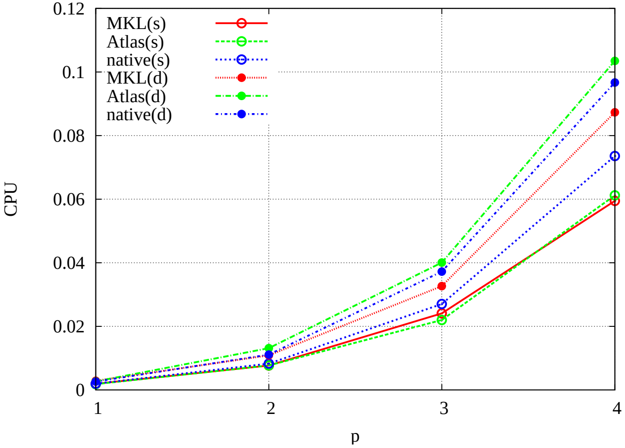
matrix-vector product



Implicit solver

Matrix operations efficiency

matrix-vector product



Practical implementation

Implicit solver : Krylov subspace methods (see Saad00 [Saa00])

Look for correction $\Delta u^n \in \mathcal{K}_n\left(\mathbf{A}^\star, \mathbf{r}^\star\right)$

$$\mathcal{K}_n\left(\mathbf{A}^\star, \mathbf{r}^\star\right) = \mathrm{span}\{\mathbf{r}^\star, \mathbf{A}^\star \cdot \mathbf{r}^\star, \ldots, \left(\mathbf{A}^\star\right)^n \cdot \mathbf{r}^\star\}$$

Needs

- operator to provide $\mathbf{A}^* \cdot p$ for generic p matrix-free Krylov

$$\mathbf{A}^\star \cdot \mathbf{p} \approx \frac{\mathbf{r}^\star\left(\mathbf{u}^n + \epsilon\ \mathbf{p}\right) - \mathbf{r}^\star\left(\mathbf{u}^n\right)}{\epsilon}$$
$$\epsilon = \sqrt{\mu}\frac{||u||}{||p||}$$

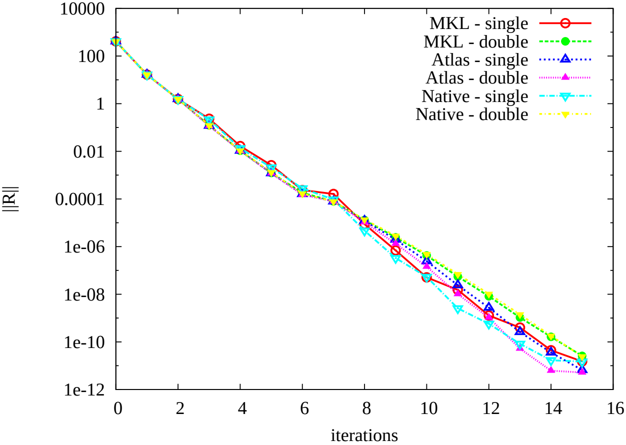
- vector internal product (parallellisation) (Gramm-Schmidt)
- preconditioning : pick iterative method $\mathbf{P} \sim A^-1$

$$\mathbf{A}^\star \cdot \mathbf{P} \cdot \mathbf{x} = -\mathbf{r}^\star$$
$$\Delta \mathbf{u}^n = \mathbf{P} \cdot \mathbf{x} \sim \left(\mathbf{A}^\star\right)^{-1} \cdot \mathbf{x}$$

  matrix preconditioners (BILU, BJacobi), hp-multigrid

Practical implementation

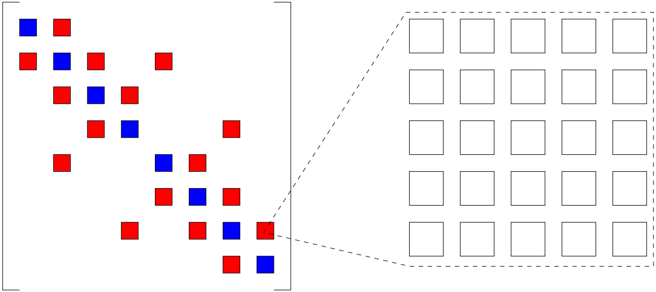
Implicit solver : single precision preconditioner



- still solve a double precision problem
- preconditioning only requires approximate solution
- halves the memory
- up to twice as efficient

Practical implementation

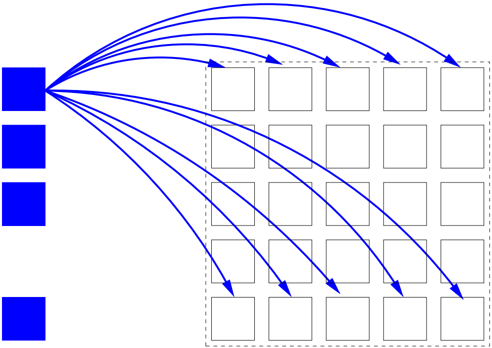
Efficient Jacobian assembly : volume Jacobian assembly (naive)



$$\mathbf{r}_{im} = \int_V \nabla\phi \vec{f}_c dV = \sum_q w_q \left(|\mathbf{J}| \frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}^{-1}_{kl} f^l_m(u) \right)_{\xi_q} \Rightarrow \frac{\partial \mathbf{r}_{im}}{\partial \mathbf{u}_{jn}} = \sum_q \left(w_q \frac{\partial \phi_i}{\partial \xi^k} \phi_j \right)_{\xi_q} \left(|\mathbf{J}| \mathbf{J}^{-1}_{kl} \frac{\partial f^l_m}{\partial u_n}(u) \right)_{\xi_q}$$

Practical implementation

Efficient Jacobian assembly : volume Jacobian assembly (contiguous)

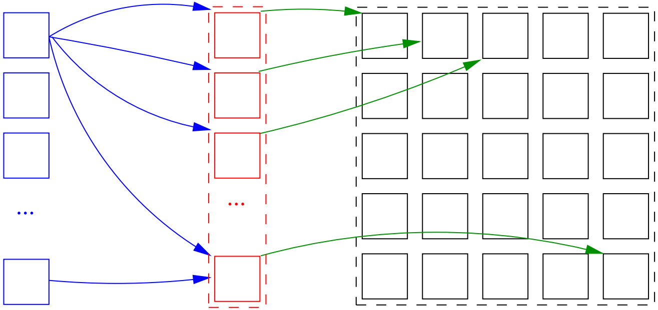


$$\mathbf{r}_{im} = \int_V \nabla \phi \vec{f}_c dV = \sum_q w_q \left(|\mathbf{J}| \frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}^{-1}_{kl} f^l_m(u) \right)_{\xi_q} \Rightarrow \frac{\partial \mathbf{r}_{im}}{\partial \mathbf{u}_{jn}} = \sum_q \left(w_q \frac{\partial \phi_i}{\partial \xi^k} \phi_j \right)_{\xi_q} \left(|\mathbf{J}| \mathbf{J}^{-1}_{kl} \frac{\partial f^l_m}{\partial u_n}(u) \right)_{\xi_q}$$

- subblock (right) is proxy → not contiguous in memory → addition is done row per row

Practical implementation

Efficient Jacobian assembly : volume Jacobian assembly (optimized)

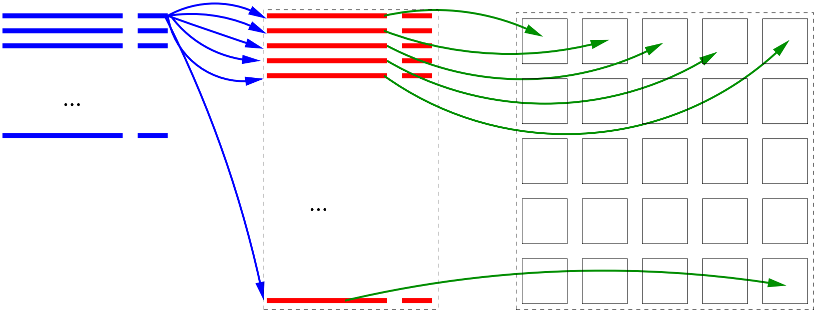


$$\mathbf{r}_{im} = \int_V \nabla \phi \vec{f}_c dV = \sum_q w_q \left(|\mathbf{J}| \frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}^{-1}_{kl} f^l_m(u) \right)_{\xi_q} \Rightarrow \frac{\partial \mathbf{r}_{im}}{\partial \mathbf{u}_{jn}} = \sum_q \left(w_q \frac{\partial \phi_i}{\partial \xi^k} \phi_j \right)_{\xi_q} \left(|\mathbf{J}| \mathbf{J}^{-1}_{kl} \frac{\partial f^l_m}{\partial u_n}(u) \right)_{\xi_q}$$

- subblock (right) is proxy → not contiguous in memory
- intermediate blocks contiguous → single contiguous sum for N_q assembly steps (blue) + single copy (green)

Practical implementation

Efficient Jacobian assembly : volume Jacobian assembly



$$\mathbf{r}_{im} = \int_V \nabla\phi \vec{f}_c dV = \sum_q w_q \left(|\mathbf{J}| \frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}^{-1}_{kl} f^l_m(u) \right)_{\xi_q} \Rightarrow \frac{\partial \mathbf{r}_{im}}{\partial \mathbf{u}_{jn}} = \sum_q \left(w_q \frac{\partial \phi_i}{\partial \xi^k} \phi_j \right)_{\xi_q} \left(|\mathbf{J}|\mathbf{J}^{-1}_{kl} \frac{\partial f^l_m}{\partial u_n}(u) \right)_{\xi_q}$$

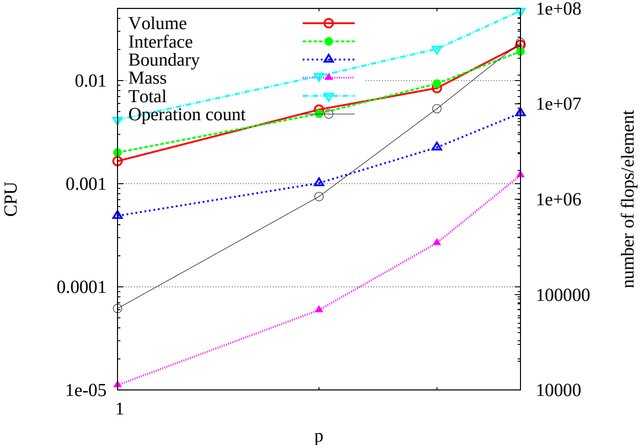
- subblock (right) is proxy → not contiguous in memory
- intermediate blocks contiguous → single contiguous sum for N_q assembly steps (blue) + single copy (green)
- padding increases flop efficiency in the assembly sums (blue)

Similar optimisations for interface terms etc. See [Hil10] for details.

Efficient Jacobian assembly

Evolution of assembly cost

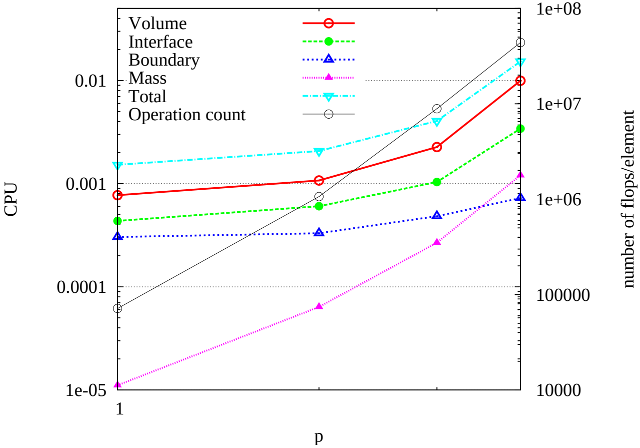
MKL - naive



Efficient Jacobian assembly

Evolution of assembly cost

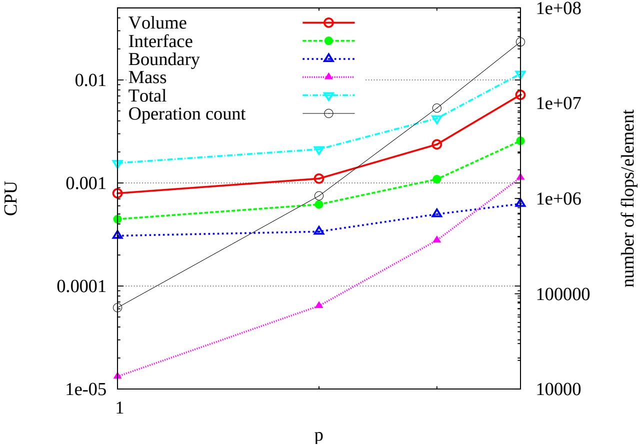
MKL - contiguous



Efficient Jacobian assembly

Evolution of assembly cost

MKL - optimized

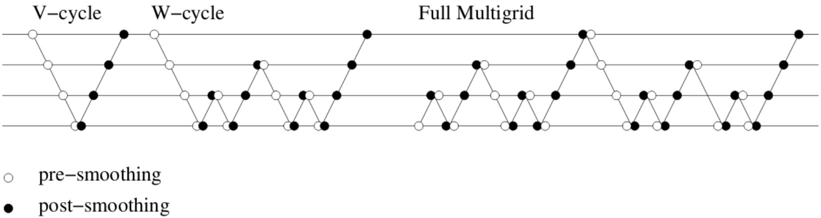


Computational complexity $\neq$ computational effort

Outline

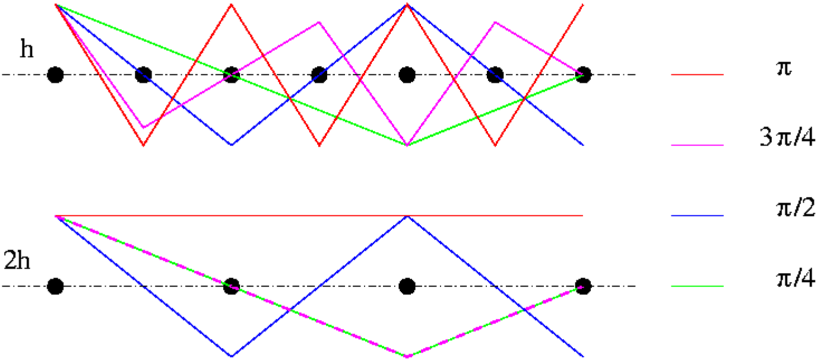
hp-multigrid

Basics : multilevel methods



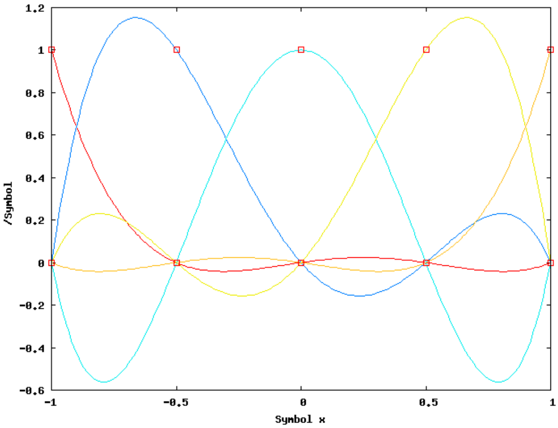
hp-multigrid

Basics : h-Multigrid → element size coarsening



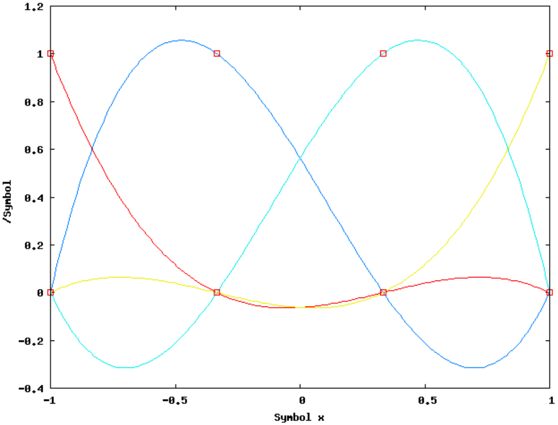
hp-multigrid

Basics : p-Multigrid → element order coarsening [FOLD05]



hp-multigrid

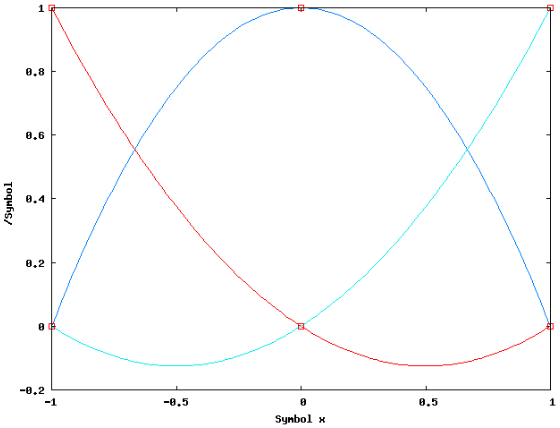
Basics : p-Multigrid → element order coarsening [FOLD05]



hp-multigrid

Basics : p-Multigrid → element order coarsening

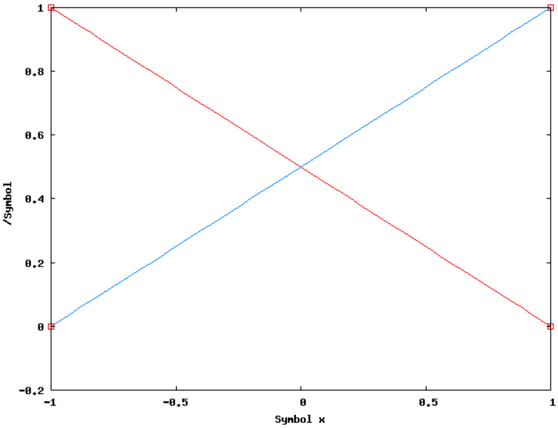
[FOLD05]



hp-multigrid

Basics : p-Multigrid → element order coarsening

[FOLD05]



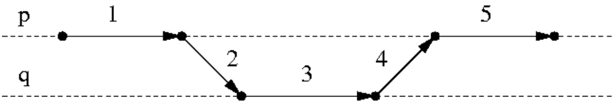
hp-multigrid

Basics : DGM - variational FAS

[HCGR06, HRC+06]

Define fine (p) and coarse level (q) by either grid- or order-coarsening

$$u^p \in \Phi^p$$
$$u^q \in \Phi^q$$



FAS cycle :

1. iterate on the fine level : $u^p \rightarrow u^{p'}$
2. restrict the fine level solution : $u^{q'} = \mathcal{T}^{qp}\left(u^{p'}\right)$
3. solve variational formulation of the defect correction equation

$$\left(\mathcal{L}\left(u^q\right) - \mathcal{L}\left(u^{q'}\right) + \mathcal{L}\left(u^{p'}\right), \phi_j^q\right) = 0 \quad \forall \phi_j^q \in \Phi^q$$

4. prolongate the correction $u^p := u^{p'} + \mathcal{T}^{pq}\left(u^q - u^{q'}\right)$
5. iterate on the finest level to smooth the error

hp-multigrid

Transfer operators : Solution transfer between levels a and b [HCGR06, HRC+06]

Both restriction and prolongation : use Galerkin projection

$u^a \in \Phi^a \rightarrow u^b \in \Phi^b$

$$
\begin{aligned}
u^a_m &= \mathbf{u}^a_{im}\phi^a_i \ , \ \phi^a_i \in \Phi^a \\
\mathcal{T}^{ba} u^a_m = u^b_m &= \mathbf{u}^b_{jm}\phi^b_j \ , \ \phi^b_j \in \Phi^b
\end{aligned}
$$

L_2 projection $\Phi^a \rightarrow \Phi^b$

$$
\left(\phi^b_k, \phi^b_j\right)\mathbf{u}^b_{jm} = \left(\phi^b_k, \phi^a_i\right)\mathbf{u}^a_{im}, \ \ \forall \phi^b_k \in \Phi^b
$$

Solution transfer matrix $\mathbf{T}^{ba}$

$$
\mathbf{u}^b = \mathbf{T}^{ba} \cdot \mathbf{u}^a = \left(\mathbf{M}^{bb}\right)^{-1} \cdot \mathbf{M}^{ba} \cdot \mathbf{u}^a
$$

hp-multigrid

Transfer operators : Residual transfer

[HCGR06, HRC+06]

Expand $\mathcal{L}(u^p)$ in Φ^p using L_2 projection

$$\mathcal{L}_m(u^p) \approx \sum_i \mathbf{l}^p_{im} \phi^p_i$$

$$\sum_i \mathbf{l}^p_{im} \left(\phi^p_i, \phi^p_j\right) \approx \left(\phi^p_j, \mathcal{L}(u^p)\right) = \mathbf{r}^p_{jm}$$

then weigh with ϕ^q_j

$$\mathbf{r}^{q\prime}_{im} = \left(\phi^q_i, \mathcal{L}_m(u^p)\right) \approx \left(\phi^q_i, \sum \mathbf{l}^p_{im} \phi^p_i\right)$$

residual transfer matrix $\tilde{\mathbf{T}}^{qp}$

$$\begin{aligned} \mathbf{r}^{q\prime} &= \mathbf{M}^{qp} \cdot \left(\mathbf{M}^{pp}\right)^{-1} \mathbf{r}^p \\ &= \tilde{\mathbf{T}}^{qp} \mathbf{r}^p = \left(\mathbf{T}^{pq}\right)^T \mathbf{r}^p \end{aligned}$$

hp-multigrid

Transfer operators : equivalence of DCGA and GCGA [HCGR06, HRC+06]

$\mathcal{L}$ is linear :

$$\mathbf{A}^p \cdot \mathbf{u}^p = \mathbf{s}^p$$
$$\mathbf{A}^p_{ij} = \left(\phi^p_i, \mathcal{L}\left(\phi^p_j\right)\right)$$

$\mathbf{\Phi}^q \in \mathbf{\Phi}^p$

$$\phi^q_i = \alpha^{qp}_{ij} \cdot \phi^p_j \ , \ \forall \phi^q_i \in \mathbf{\Phi}^q$$
$$\alpha^{qp} = \mathbf{M}^{qp} \cdot \left(\mathbf{M}^{pp}\right)^{-1} = \tilde{\mathbf{T}}^{qp} = \left(\mathbf{T}^{pq}\right)^T$$

then we can compute the *Coarse Grid Approximation (CGA)* of $\mathbf{A}^q$

$$\mathbf{A}^q_{ij} = \left(\phi^q_i, \mathcal{L}\left(\phi^q_j\right)\right) = \alpha^{qp}_{ik} \cdot \left(\phi^p_k, \mathcal{L}(\phi^p_l)\right) \cdot \alpha^{qp}_{jl}$$
$$\mathbf{A}^q = \alpha^{qp}\mathbf{A}p\left(\alpha^{qp}\right)^T = \tilde{\mathbf{T}}^{qp} \cdot \mathbf{A}^p \cdot \mathbf{T}^{pq}$$

Using Galerkin projection and variational FAS, we get *optimal* Galerkin CGA from *simple/standard* discretisation CGA

hp-multigrid

Transfer operators : Galerkin CGA and error propagation

- restriction of residual after correction is exactly 0

$$\tilde{\mathbf{T}}^{qp}\left(\mathbf{A}^p\cdot\left(\mathbf{u}^{p\prime}+\mathbf{T}^{pq}\cdot\left(\mathbf{u}^q-\mathbf{u}^{q\prime}\right)\right)-\mathbf{s}^p\right)=0$$

- the error after coarse grid correction depends only on the smooth part of the initial error

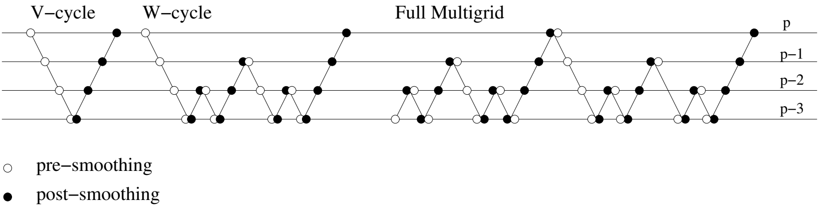
$$\begin{aligned}
\mathbf{A}^p\mathbf{e}^p &= \mathbf{A}^p\mathbf{u}^p-\mathbf{s}^p=\mathbf{r}^p\\
\mathbf{e}^{p\prime}_S &= \left(\mathbf{T}^{pq}\cdot\left(\tilde{\mathbf{T}}^{qp}\mathbf{T}^{pq}\right)^{-1}\cdot\tilde{\mathbf{T}}^{qp}\right)\cdot\mathbf{e}^{p\prime}\in\mathrm{ran}\left(\mathbf{T}^{qp}\right)\\
\mathbf{e}^{p\prime}_R &= \left(\mathbf{I}p-\mathbf{T}^{pq}\cdot\left(\tilde{\mathbf{T}}^{qp}\mathbf{T}^{pq}\right)^{-1}\cdot\tilde{\mathbf{T}}^{qp}\right)\cdot\mathbf{e}^{p\prime}\in\ker\left(\tilde{\mathbf{T}}^{pq}\right)
\end{aligned}$$

then

$$\mathbf{e}^p=\left(\mathbf{I}p-\mathbf{T}^{pq}\cdot\left(\mathbf{A}^q\right)^{-1}\cdot\tilde{\mathbf{T}}^{qp}\cdot\mathbf{A}^p\right)\cdot\mathbf{e}^{p\prime}_R$$

hp-multigrid

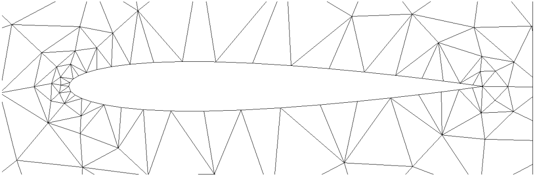
Performance for convective problems : Strategy

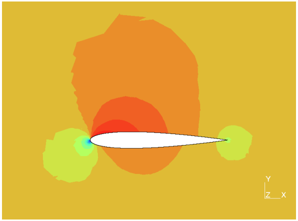


Schemes

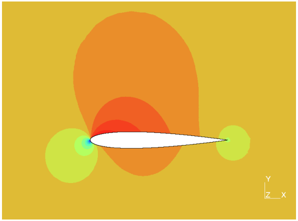
- 4step explicit Runge-Kutta on finest levels
- 10 pre- and postsmoothing steps
- hybrid cycles : Newton step at coarsest level

hp-multigrid : Performance for convective problems
NACA0012





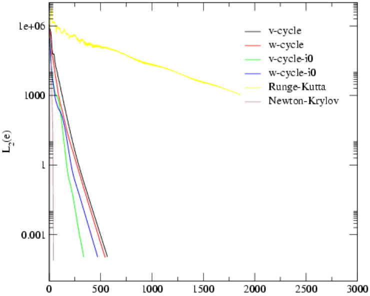
(f) p=2



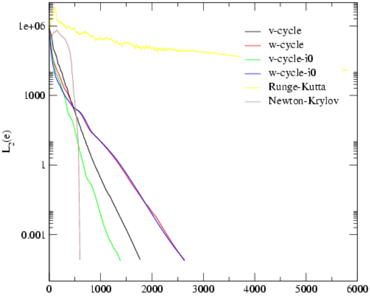
(g) p=4

hp-multigrid : Performance for convective problems

CPU comparison with Newton-Krylov - coarse mesh



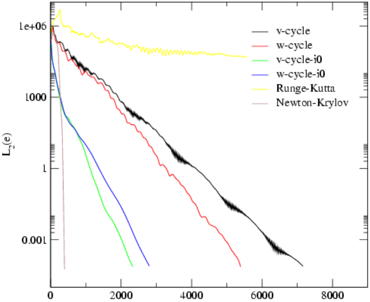
(h) p=2



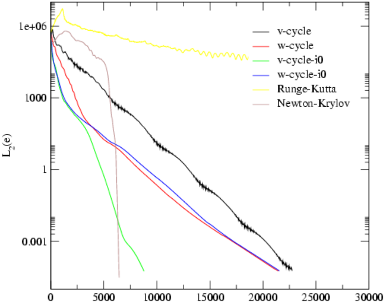
(i) p=4

hp-multigrid : Performance for convective problems

CPU comparison with Newton-Krylov - fine mesh



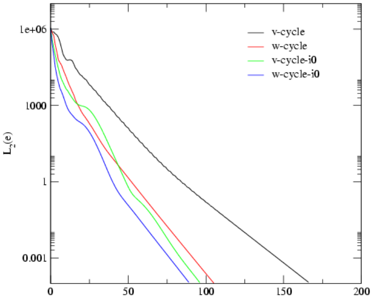
(j) p=2



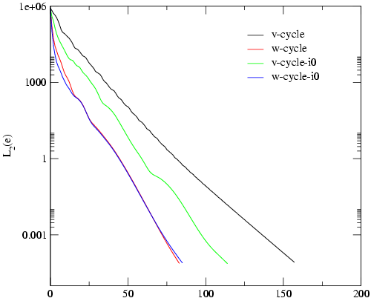
(k) p=4

hp-multigrid

Performance for convective problems : Cycling strategies - coarse mesh



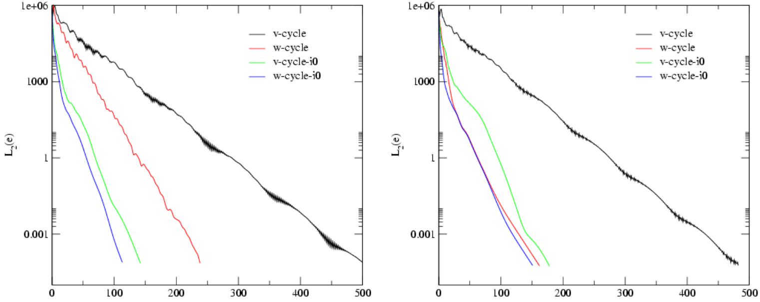
(l) p=2



(m) p=4

hp-multigrid

Performance for convective problems : Cycling strategies - fine mesh



(n) p=2 (o) p=4

hp-multigrid : Concluding remarks

Concluding remarks

Conclusions

- easy multigrid implementation for DG
- optimal transfer operators
- hybrid p-multigrid approach promising for inviscid flows (hybrid cycle)
- very easy to use for nested initial iterations

Further work

- h-Multigrid implementations on unstructured meshes
  - agglomeration multigrid *Tesini 2008* [Tes08]
  - nested meshes
  - independent meshes
  - directional coarsening
- design of smoothers for viscous flows *VdVegt, JCP [vdVR12a, vdVR12b]*, ?

References I

D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini.
Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems.
SIAM J. Num. Anal, 39 :1749–1779, 2002.

M. Bergot, G. Cohen, and M. Duruflé.
High-Order Finite Elements for Hybrid Meshes Using New Nodal Pyramidal Elements.
Journal of Scientific Computing, 42 :345–381, 2010.

K.J. Fidkowski, T.A. Oliver, J. Lu, and D.L. Darmofal.
p-Multigrid solution of high-order discontinuous Galerkin discretizations of the Navier-Stokes equations.
Journal of Computational Physics, 207 :92–113, 2005.

K. Hillewaert, N. Chevaugeon, P. Geuzaine, and J.-F. Remacle.
Hierarchic multigrid iteration strategy for the discontinuous Galerkin solution of the steady Euler equations.
International Journal for Numerical Methods in Fluids, 51(9-10) :1157 – 1176, 2006.

K. Hillewaert.
ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications - Results of a collaborative research project funded by the European Union, 2006-2009, volume 113 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, chapter Chapter 2 : Exploiting data locality in the DGM discretisation for optimal efficiency, pages 11–24.
Springer, 2010.

References II

K. Hillewaert, J.-F. Remacle, N. Cheveaugeon, P.-E. Bernard, and P. Geuzaine.
Analysis of a hybrid p-multigrid method for the Discontinuous Galerkin Discretisation of the Euler Equations.
In P. Wesseling, E. Oñate, and J. Périaux, editors, *ECCOMAS CFD Conference*, 2006.

G. Jiang and C.-W. Shu.
On a cell entropy inequality for discontinuous Galerkin methods.
Mathematics anc Computation, 62(206) :531–538, April 1994.

B. Rivière.
Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations - Theory and Implementation.
Frontiers in applied mechanics. SIAM, 2008.

Y. Saad.
Iterative methods for Sparse Linear Systems.
2nd edition, 2000.
http ://www.stanford.edu/class/cme324/saad.pdf.

K. Shahbazi.
An explicit expression for the penalty parameter of the interior penalty method (Short note).
Journal of Computational Physics, 205 :401–407, 2005.

P. Solin, K. Segeth, and I. Dolezel.
Higher Order Finite Element Methods.
Studies in Advanced Mathematics. Chapman and Hall/CRC, 2004.

References III

P. Tesini.
An h-Multigrid Approach for High-Order Discontinuous Galerkin Methods.
PhD thesis, Universita degli studi di Bergamo - Dipartimento di Ingegneria Industriale, 2008.

J.J.W van der Vegt and S. Rhebergen.
hp-multigrid as smoother algorithm for higher order discontinuous galerkin discretizations of advection dominated flows. part i. multilevel analysis.
Journal of Computational Physics (in press), 2012.

J.J.W van der Vegt and S. Rhebergen.
hp-multigrid as smoother algorithm for higher order discontinuous galerkin discretizations of advection dominated flows. part ii. optimization of the runge-kutta smoother.
Journal of Computational Physics (in press), 2012.