Perspectives of discontinuous Galerkin method for the numerical solution of computational fluid dynamics problems

Vít Dolejší

Charles University Prague
Faculty of Mathematics and Physics

Programy a algoritmy numerické matematiky 17
8. - 13. června 2014
Dolní Maxov

Perspectives of discontinuous Galerkin method for the numerical solution of computational fluid dynamics problems

Vít Dolejší

Charles University Prague
Faculty of Mathematics and Physics

Programy a algoritmy numerické matematiky 17
8. - 13. června 2014
Dolní Maxov

Perspectives of discontinuous Galerkin method for the numerical solution of computational fluid dynamics problems

Vít Dolejší

Charles University Prague
Faculty of Mathematics and Physics

Programy a algoritmy numerické matematiky 17
8. - 13. června 2014
Dolní Maxov

Introduction

Computational fluid dynamics (CFD)

- WIKI: CFD uses numerical methods to solve and analyze problems involving fluid flows,
- turbomachinery, aeroacoustic, aeronautics, car industry, biomedical, electronics, environmental protection,
- problems are described by partial differential equations (PDEs).

The main goal

compute the quantities of interest
under the given accuracy
within the shortest possible time.

Introduction

Computational fluid dynamics (CFD)

- WIKI: CFD uses numerical methods to solve and analyze problems involving fluid flows,
- turbomachinery, aeroacoustic, aeronautics, car industry, biomedical, electronics, environmental protection,
- problems are described by partial differential equations (PDEs).

The main goal

compute the quantities of interest
under the given accuracy
within the shortest possible time.

Introduction

Computational fluid dynamics (CFD)

- WIKI: CFD uses numerical methods to solve and analyze problems involving fluid flows,
- turbomachinery, aeroacoustic, aeronautics, car industry, biomedical, electronics, environmental protection,
- problems are described by partial differential equations (PDEs).

The main goal

compute the quantities of interest
under the given accuracy
within the shortest possible time.

Introduction

Computational fluid dynamics (CFD)

- WIKI: CFD uses numerical methods to solve and analyze problems involving fluid flows,
- turbomachinery, aeroacoustic, aeronautics, car industry, biomedical, electronics, environmental protection,
- problems are described by partial differential equations (PDEs).

The main goal

compute the quantities of interest
under the given accuracy
within the shortest possible time.

Introduction

Computational fluid dynamics (CFD)

- WIKI: CFD uses numerical methods to solve and analyze problems involving fluid flows,
- turbomachinery, aeroacoustic, aeronautics, car industry, biomedical, electronics, environmental protection,
- problems are described by partial differential equations (PDEs).

The main goal

compute the quantities of interest
under the given accuracy
within the shortest possible time.

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: h, hp variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: *h*, *hp* variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: *h*, *hp* variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: *h*, *hp* variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: *h*, *hp* variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: h, hp variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: h, hp variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: h, hp variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: h, hp variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: h, hp variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: h, hp variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

How to achieve the main goal

- suitable discretization:
  numerical method & domain partition $\mathscr{T}_h$, τ_k,
- efficient solution of arising algebraic systems
- accurate and efficient error estimates.

Requirements

- numerical method: accurate, robust, parallelizable
- algebraic solver: iterative, stopping criteria
- a posterior error analysis: accurate and fast approximation of error, algebraic error
- mesh adaptation: h, hp variants, anisotropy
- time step adaptation: explicit/implicit, order adaptation

Software for CFD

Google searching

- “CFD software finite element method” – 276 000 links
- “CFD software finite volume method” – 161 000 links
- “CFD software spectral method” – 168 000 links
- “CFD software discontinuous Galerkin method” – 25 000 links

www.cfd-online.com – list of CFD codes (public and commercial)

- finite element methods (FEM) – 50%
- finite volume methods (FVM) – 40%
- spectral methods (SM) – 10%
- discontinuous Galerkin methods (DGM) – 0%(?)

Software for CFD

Google searching

- "CFD software finite element method" – 276 000 links
- "CFD software finite volume method" – 161 000 links
- "CFD software spectral method" – 168 000 links
- "CFD software discontinuous Galerkin method" – 25 000 links

www.cfd-online.com – list of CFD codes (public and commercial)

- finite element methods (FEM) – 50%
- finite volume methods (FVM) – 40%
- spectral methods (SM) – 10%
- discontinuous Galerkin methods (DGM) – 0%(?)

Software for CFD

Google searching

- "CFD software finite element method" – 276 000 links
- "CFD software finite volume method" – 161 000 links
- "CFD software spectral method" – 168 000 links
- "CFD software discontinuous Galerkin method" – 25 000 links

www.cfd-online.com – list of CFD codes (public and commercial)

- finite element methods (FEM) – 50%
- finite volume methods (FVM) – 40%
- spectral methods (SM) – 10%
- discontinuous Galerkin methods (DGM) – 0%(?)

Software for CFD

Google searching

- "CFD software finite element method" – 276 000 links
- "CFD software finite volume method" – 161 000 links
- "CFD software spectral method" – 168 000 links
- "CFD software discontinuous Galerkin method" – 25 000 links

www.cfd-online.com – list of CFD codes (public and commercial)

- finite element methods (FEM) – 50%
- finite volume methods (FVM) – 40%
- spectral methods (SM) – 10%
- discontinuous Galerkin methods (DGM) – 0%(?)

Software for CFD

Google searching

- "CFD software finite element method" – 276 000 links
- "CFD software finite volume method" – 161 000 links
- "CFD software spectral method" – 168 000 links
- "CFD software discontinuous Galerkin method" – 25 000 links

www.cfd-online.com – list of CFD codes (public and commercial)

- finite element methods (FEM) – 50%
- finite volume methods (FVM) – 40%
- spectral methods (SM) – 10%
- discontinuous Galerkin methods (DGM) – 0%(?)

Software for CFD

Google searching

- "CFD software finite element method" – 276 000 links
- "CFD software finite volume method" – 161 000 links
- "CFD software spectral method" – 168 000 links
- "CFD software discontinuous Galerkin method" – 25 000 links

www.cfd-online.com – list of CFD codes (public and commercial)

- finite element methods (FEM) – 50%
- finite volume methods (FVM) – 40%
- spectral methods (SM) – 10%
- discontinuous Galerkin methods (DGM) – 0%(?)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2^{nd} order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2^{nd} order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2^{nd} order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2^{nd} order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2[nd] order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2$^{\mathrm{nd}}$ order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2$^{\text{nd}}$ order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2$^{\mathrm{nd}}$ order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2$^{\text{nd}}$ order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2^{nd} order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2^{nd} order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2^nd order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise polynomial discontinuous approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FEM and FVM

FEM – piecewise polynomial continuous approximation

- high order of accuracy supported by theory
- efficient for problems with continuous solutions

FVM – piecewise constant discontinuous approximation

- low order of accuracy (commercial codes at most 2$^{\text{nd}}$ order)
- works for problems with discontinuities or steep gradients

discontinuous Galerkin method

- piecewise <u>polynomial discontinuous</u> approximation
- a little complicated formulation and theoretical analysis
- higher freedom (adaptation, parallelization, etc.)

Comparison of FVM, FEM and DGM

1D convection-diffusion problem

$$-\varepsilon u'' + u' = 1 \quad \text{in } \Omega := (0,1), \qquad u(0) = u(1) = 0,$$

$0 < \varepsilon \ll 1$ convection dominanting problem

$\Rightarrow$ boundary layer at $x = 1$ proportional to ε.

Numerical tests

- FEM with continuous P_1 approximation with $h = 1/32$,
- FVM piecewise constant approximation with $h = 1/32$,
- DGM using P_4 approximation with $h = 1/8$.

$\Rightarrow$ 32 degrees of freedom for all cases.

Comparison of FVM, FEM and DGM

1D convection-diffusion problem

$$-\varepsilon u'' + u' = 1 \quad \text{in } \Omega := (0,1), \quad u(0) = u(1) = 0,$$

$0 < \varepsilon \ll 1$ convection dominanting problem

$\Rightarrow$ boundary layer at $x = 1$ proportional to ε.

Numerical tests

- FEM with continuous P_1 approximation with $h = 1/32$,
- FVM piecewise constant approximation with $h = 1/32$,
- DGM using P_4 approximation with $h = 1/8$.

$\Rightarrow$ 32 degrees of freedom for all cases.

Comparison of FVM, FEM and DGM

1D convection-diffusion problem

$$-\varepsilon u'' + u' = 1 \quad \text{in } \Omega := (0,1), \qquad u(0) = u(1) = 0,$$

$0 < \varepsilon \ll 1$ convection dominanting problem

$\Rightarrow$ boundary layer at $x = 1$ proportional to ε.

Numerical tests

- FEM with continuous P_1 approximation with $h = 1/32$,
- FVM piecewise constant approximation with $h = 1/32$,
- DGM using P_4 approximation with $h = 1/8$.

$\Rightarrow$ 32 degrees of freedom for all cases.

Comparison of FVM, FEM and DGM

1D convection-diffusion problem

$$-\varepsilon u'' + u' = 1 \quad \text{in } \Omega := (0,1), \qquad u(0) = u(1) = 0,$$

$0 < \varepsilon \ll 1$ convection dominanting problem

$\Rightarrow$ boundary layer at $x = 1$ proportional to ε.

Numerical tests

- FEM with continuous P_1 approximation with $h = 1/32$,
- FVM piecewise constant approximation with $h = 1/32$,
- DGM using P_4 approximation with $h = 1/8$.

$\Rightarrow$ 32 degrees of freedom for all cases.

Comparison of FVM, FEM and DGM

1D convection-diffusion problem

$$-\varepsilon u'' + u' = 1 \text{ in } \Omega := (0,1), \quad u(0) = u(1) = 0,$$

$0 < \varepsilon \ll 1$ convection dominanting problem

$\Rightarrow$ boundary layer at $x = 1$ proportional to ε.

Numerical tests

- FEM with continuous P_1 approximation with $h = 1/32$,
- FVM piecewise constant approximation with $h = 1/32$,
- DGM using P_4 approximation with $h = 1/8$.

$\Rightarrow$ 32 degrees of freedom for all cases.

Comparison of FVM, FEM and DGM

1D convection-diffusion problem

$$-\varepsilon u'' + u' = 1 \quad \text{in } \Omega := (0,1), \qquad u(0) = u(1) = 0,$$

$0 < \varepsilon \ll 1$ convection dominanting problem

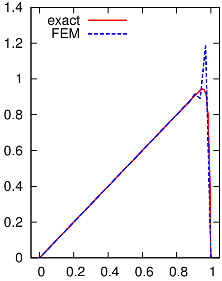
$\Rightarrow$ boundary layer at $x = 1$ proportional to ε.

Numerical tests

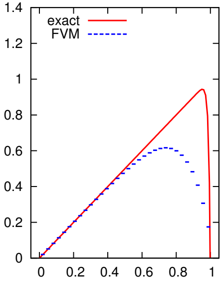
- FEM with continuous P_1 approximation with $h = 1/32$,
- FVM piecewise constant approximation with $h = 1/32$,
- DGM using P_4 approximation with $h = 1/8$.

$\Rightarrow$ 32 degrees of freedom for all cases.

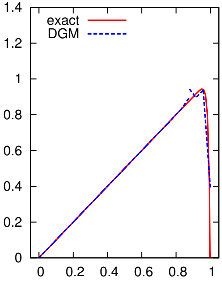
Comparison of FVM, FEM and DGM with $\varepsilon = 10^{-2}$



FEM

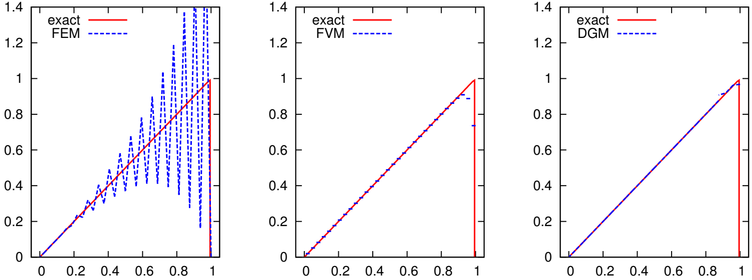


FVM



DGM

Comparison of FVM, FEM and DGM with $\varepsilon = 10^{-3}$



FEM FVM DGM

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx C h^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$
$p =$ polyn. degree, $\mathrm{DOF} \approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx C h^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$

$p =$ polyn. degree, DOF $\approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx Ch^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$

$p =$ polyn. degree, DOF $\approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx C h^{p+1}$, $h \approx (\#\mathcal{T}_h)^{1/d}$, $d = 2, 3$

$p =$ polyn. degree, $\mathrm{DOF} \approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathcal{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \le \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx C h^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$

$p =$ polyn. degree, DOF $\approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \le \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx C h^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$

$p =$ polyn. degree, DOF $\approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx C h^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$

$p =$ polyn. degree, DOF $\approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx C h^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$

$p =$ polyn. degree, $\mathrm{DOF} \approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx C h^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$

$p =$ polyn. degree, $\qquad \mathrm{DOF} \approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx C h^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$
$p =$ polyn. degree, DOF $\approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

$\|u - u_h\| \approx Ch^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$

$p =$ polyn. degree, DOF $\approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Why higher order approximation?

the aim is $\|u - u_h\| \leq \omega$ & DOF minimal

Theoretical as well as experimental results

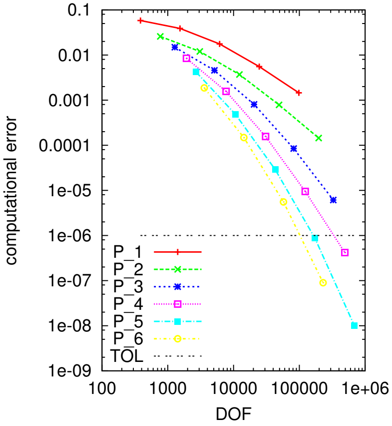
$\|u - u_h\| \approx C h^{p+1}$, $h \approx (\#\mathscr{T}_h)^{1/d}$, $d = 2, 3$
$p =$ polyn. degree, DOF $\approx \frac{1}{d!}(p+1)\dots(p+d)\#\mathscr{T}_h$

number of DOF necessary to fulfil $\|u - u_h\| \leq \omega$:

d	ω	P_1	P_2	P_3	P_4	P_5	P_6
2	1.00E-02	301	130	101	95	98	105
	1.00E-03	3001	601	317	238	211	202
	1.00E-04	30001	2785	1001	598	453	390
3	1.00E-02	3001	601	317	238	211	202
	1.00E-03	94869	6001	1779	947	665	541
	1.00E-04	3000001	60001	10001	3768	2101	1451

Practical demonstration of the efficiency

- scalar linear convection-diffusion equation solved by DGM



1. Introduction

2. DGM for problems of CFD

3. Solution strategies

4. Mesh adaptation

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \qquad (1)$$

- $\mathbf{w} : \Omega \times (0,T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N$, $s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$, $s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$$S_h^p := \{v;\ v \in L^2(\Omega),\ v|_K \in P_p(K)\ \forall K \in \mathscr{T}_h\},\ \mathbf{S}_{hp} := (S_h^p)^N.$$

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \tag{1}$$

- $\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N$, $s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$, $s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$$S_h^p := \{v;\ v \in L^2(\Omega),\ v|_K \in P_p(K)\ \forall K \in \mathscr{T}_h\},\ \mathbf{S}_{hp} := (S_h^p)^N.$$

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s}\mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s}\mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \qquad (1)$$

- $\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N, \; s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N, \; s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$$S_h^p := \{v; \; v \in L^2(\Omega), \; v|_K \in P_p(K) \; \forall K \in \mathscr{T}_h\}, \; \mathbf{S}_{hp} := (S_h^p)^N.$$

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \qquad (1)$$

- $\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N, \;\; s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N, \;\; s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$$S_h^p := \{v;\; v \in L^2(\Omega),\; v|_K \in P_p(K)\; \forall K \in \mathscr{T}_h\}, \; \mathbf{S}_{hp} := (S_h^p)^N.$$

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \qquad (1)$$

- $\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N, \ s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N, \ s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$$S_h^p := \{v; \ v \in L^2(\Omega), \ v|_K \in P_p(K) \ \forall K \in \mathscr{T}_h\}, \ \mathbf{S}_{hp} := (S_h^p)^N.$$

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \qquad (1)$$

- $\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N$, $s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$, $s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$$S_h^p := \{v;\ v \in L^2(\Omega),\ v|_K \in P_p(K)\ \forall K \in \mathscr{T}_h\},\ \mathbf{S}_{hp} := (S_h^p)^N.$$

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \qquad (1)$$

- $\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N, \ s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N, \ s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$$S_h^p := \{v;\ v \in L^2(\Omega),\ v|_K \in P_p(K)\ \forall K \in \mathscr{T}_h\},\ \mathbf{S}_{hp} := (S_h^p)^N.$$

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \qquad (1)$$

- $\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N$, $s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$, $s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$$S_h^p := \{v;\ v \in L^2(\Omega),\ v|_K \in P_p(K)\ \forall K \in \mathscr{T}_h\},\ \mathbf{S}_{hp} := (S_h^p)^N.$$

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \qquad (1)$$

- $\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N, \ s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N, \ s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$S_h^p := \{v;\ v \in L^2(\Omega),\ v|_K \in P_p(K)\ \forall K \in \mathscr{T}_h\}$, $\mathbf{S}_{hp} := (S_h^p)^N$.

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \qquad (1)$$

- $\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N, \ s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N, \ s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$S_h^p := \{v;\ v \in L^2(\Omega),\ v|_K \in P_p(K)\ \forall K \in \mathscr{T}_h\},\ \mathbf{S}_{hp} := (S_h^p)^N.$

Abstract CFD problem

System of convection-diffusion equations

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{f}_s(\mathbf{w}) = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \mathbf{R}_s(\mathbf{w}, \nabla \mathbf{w}), \qquad (1)$$

- $\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^N$,
- convective terms $\mathbf{f}_s : \mathbb{R}^N \to \mathbb{R}^N, \ s = 1, \dots, d$,
- diffusive terms $\mathbf{R}_s : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N, \ s = 1, \dots, d$,
- closing relations, e.g., from thermodynamic, BC + IC.

compressible Navier-Stokes equations, shallow water equations, etc.

Space of piecewise polynomial functions

$S_h^p := \{v;\ v \in L^2(\Omega),\ v|_K \in P_p(K)\ \forall K \in \mathscr{T}_h\},\ \mathbf{S}_{hp} := (S_h^p)^N.$

DG discretization

Space-time discretization, $w(t_k) \approx w_h^k \in \mathbf{S}_{hp}$

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \; k = 1, 2, \ldots$$

$$\mathbf{c}_h(\mathbf{w}_h, \varphi_h) = \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x$$

$$- \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \rangle n_s [\varphi_h] \, \mathrm{d}S + \eta \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \varphi_h) \rangle n_s [\mathbf{w}_h] \, \mathrm{d}S$$

$$- \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{f}_s(\mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \mathbf{H}(\mathbf{w}_h|_\Gamma^{(L)}, \mathbf{w}_h|_\Gamma^{(R)}, \mathbf{n}_\Gamma)[\varphi_h]_\Gamma \, \mathrm{d}S$$

$$+ \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sigma [\mathbf{w}_h] \, [\varphi_h] \, \mathrm{d}S$$

DG discretization

Space-time discretization, $w(t_k) \approx w_h^k \in \mathbf{S}_{hp}$

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \;\; k = 1, 2, \ldots$$

$$\mathbf{c}_h(\mathbf{w}_h, \varphi_h) = \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s}\, \mathrm{d}x$$

$$- \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \rangle n_s [\varphi_h]\, \mathrm{d}S + \eta \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \varphi_h) \rangle n_s [\mathbf{w}_h]\, \mathrm{d}S$$

$$- \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{f}_s(\mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s}\, \mathrm{d}x + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \mathbf{H}(\mathbf{w}_h|_\Gamma^{(L)}, \mathbf{w}_h|_\Gamma^{(R)}, \mathbf{n}_\Gamma)[\varphi_h]_\Gamma\, \mathrm{d}S$$

$$+ \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sigma [\mathbf{w}_h]\, [\varphi_h]\, \mathrm{d}S$$

DG discretization

Space-time discretization, $w(t_k) \approx w_h^k \in \mathbf{S}_{hp}$

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \; k = 1, 2, \ldots$$

$$\begin{aligned}
\mathbf{c}_h(\mathbf{w}_h, \varphi_h) = & \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \,\mathrm{d}x \\
& - \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \rangle n_s [\varphi_h] \,\mathrm{d}S + \eta \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \varphi_h) \rangle n_s [\mathbf{w}_h] \,\mathrm{d}S \\
& - \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{f}_s(\mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \,\mathrm{d}x + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \mathbf{H}(\mathbf{w}_h|_\Gamma^{(L)}, \mathbf{w}_h|_\Gamma^{(R)}, \mathbf{n}_\Gamma)[\varphi_h]_\Gamma \,\mathrm{d}S \\
& + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sigma [\mathbf{w}_h] \, [\varphi_h] \,\mathrm{d}S
\end{aligned}$$

DG discretization

Space-time discretization, $w(t_k) \approx w_h^k \in \mathbf{S}_{hp}$

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \; k = 1, 2, \ldots$$

$$\begin{aligned}
\mathbf{c}_h(\mathbf{w}_h, \varphi_h) = & \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x \\
& - \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \rangle n_s [\varphi_h] \, \mathrm{d}S + \eta \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \varphi_h) \rangle n_s [\mathbf{w}_h] \, \mathrm{d}S \\
& - \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{f}_s(\mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \mathbf{H}(\mathbf{w}_h|_\Gamma^{(L)}, \mathbf{w}_h|_\Gamma^{(R)}, \mathbf{n}_\Gamma) [\varphi_h]_\Gamma \, \mathrm{d}S \\
& + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sigma [\mathbf{w}_h] \, [\varphi_h] \, \mathrm{d}S
\end{aligned}$$

DG discretization

Space-time discretization, $w(t_k) \approx w_h^k \in \mathbf{S}_{hp}$

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \; k = 1, 2, \ldots$$

$$\begin{aligned}
\mathbf{c}_h(\mathbf{w}_h, \varphi_h) = & \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x \\
& - \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \rangle n_s [\varphi_h] \, \mathrm{d}S + \eta \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \varphi_h) \rangle n_s [\mathbf{w}_h] \, \mathrm{d}S \\
& - \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{f}_s(\mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \mathbf{H}(\mathbf{w}_h|_\Gamma^{(L)}, \mathbf{w}_h|_\Gamma^{(R)}, \mathbf{n}_\Gamma) [\varphi_h]_\Gamma \, \mathrm{d}S \\
& + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sigma [\mathbf{w}_h] \, [\varphi_h] \, \mathrm{d}S
\end{aligned}$$

DG discretization

Space-time discretization, $w(t_k) \approx w_h^k \in \mathbf{S}_{hp}$

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, 2, \dots$$

$$\begin{aligned}
\mathbf{c}_h(\mathbf{w}_h, \varphi_h) = & \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x \\
& - \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \rangle n_s [\varphi_h] \, \mathrm{d}S + \eta \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \varphi_h) \rangle n_s [\mathbf{w}_h] \, \mathrm{d}S \\
& - \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{f}_s(\mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \mathbf{H}(\mathbf{w}_h|_\Gamma^{(L)}, \mathbf{w}_h|_\Gamma^{(R)}, \mathbf{n}_\Gamma) [\varphi_h]_\Gamma \, \mathrm{d}S \\
& + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sigma [\mathbf{w}_h] \, [\varphi_h] \, \mathrm{d}S
\end{aligned}$$

DG discretization

Space-time discretization, $w(t_k) \approx w_h^k \in \mathbf{S}_{hp}$

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, 2, \dots$$

$$\mathbf{c}_h(\mathbf{w}_h, \varphi_h) = \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x$$

$$- \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \rangle n_s [\varphi_h] \, \mathrm{d}S + \eta \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \varphi_h) \rangle n_s [\mathbf{w}_h] \, \mathrm{d}S$$

$$- \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{f}_s(\mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \mathbf{H}(\mathbf{w}_h|_\Gamma^{(L)}, \mathbf{w}_h|_\Gamma^{(R)}, \mathbf{n}_\Gamma)[\varphi_h]_\Gamma \, \mathrm{d}S$$

$$+ \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sigma [\mathbf{w}_h] \, [\varphi_h] \, \mathrm{d}S$$

finite volume method

DG discretization

Space-time discretization, $w(t_k) \approx w_h^k \in \mathbf{S}_{hp}$

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \; k = 1, 2, \ldots$$

$$\begin{aligned}
\mathbf{c}_h(\mathbf{w}_h, \varphi_h) = & \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \,\mathrm{d}x \\
& - \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \rangle n_s [\varphi_h] \,\mathrm{d}S + \eta \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \varphi_h) \rangle n_s [\mathbf{w}_h] \,\mathrm{d}S \\
& - \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{f}_s(\mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \,\mathrm{d}x + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \mathbf{H}(\mathbf{w}_h|_\Gamma^{(L)}, \mathbf{w}_h|_\Gamma^{(R)}, \mathbf{n}_\Gamma)[\varphi_h]_\Gamma \,\mathrm{d}S \\
& + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sigma [\mathbf{w}_h] \, [\varphi_h] \,\mathrm{d}S
\end{aligned}$$

finite element method

DG discretization

Space-time discretization, $w(t_k) \approx w_h^k \in \mathbf{S}_{hp}$

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, 2, \ldots$$

$$\begin{aligned}
\mathbf{c}_h(\mathbf{w}_h, \varphi_h) = & \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x \\
& - \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \rangle n_s [\varphi_h] \, \mathrm{d}S + \eta \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \varphi_h) \rangle n_s [\mathbf{w}_h] \, \mathrm{d}S \\
& - \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{f}_s(\mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \mathbf{H}(\mathbf{w}_h|_\Gamma^{(L)}, \mathbf{w}_h|_\Gamma^{(R)}, \mathbf{n}_\Gamma) [\varphi_h]_\Gamma \, \mathrm{d}S \\
& + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sigma [\mathbf{w}_h] [\varphi_h] \, \mathrm{d}S
\end{aligned}$$

only DGM

DG discretization

Space-time discretization, $w(t_k) \approx w_h^k \in \mathbf{S}_{hp}$

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, 2, \ldots$$

$$\mathbf{c}_h(\mathbf{w}_h, \varphi_h) = \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x$$

$$- \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \mathbf{w}_h) \rangle n_s [\varphi_h] \, \mathrm{d}S + \eta \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sum_{s=1}^{d} \langle \mathbf{R}_s(\mathbf{w}_h, \nabla \varphi_h) \rangle n_s [\mathbf{w}_h] \, \mathrm{d}S$$

$$- \sum_{K \in \mathscr{T}_h} \int_K \sum_{s=1}^{d} \mathbf{f}_s(\mathbf{w}_h) \frac{\partial \varphi_h}{\partial x_s} \, \mathrm{d}x + \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \mathbf{H}(\mathbf{w}_h|_\Gamma^{(L)}, \mathbf{w}_h|_\Gamma^{(R)}, \mathbf{n}_\Gamma) [\varphi_h]_\Gamma \, \mathrm{d}S$$

$$+ \sum_{\Gamma \in \mathscr{F}_h} \int_\Gamma \sigma [\mathbf{w}_h] \, [\varphi_h] \, \mathrm{d}S$$

Numerical analysis

Asymptotical order of convergence

Poisson problem & scalar nonlinear convection-diffusion equation

$$|u - u_h|_{H^1(\Omega)} = O(h^p), \qquad \|u - u_h\|_{L^2(\Omega)} = O(h^{p+1})$$

the same order of convergence as for FEM

Number of degrees of freedom (DOF)

let $d = 2$, $M = \#\mathscr{T}_h$

p	1	2	3	4	5	6
FEM	$\frac{1}{2}M$	$2M$	$\frac{9}{2}$	$8M$	$\frac{25}{2}M$	$18M$
DGM	$3M$	$6M$	$10M$	$15M$	$21M$	$28M$
ratio= $\frac{\mathrm{DGM}}{\mathrm{FEM}}$	6	3	2.22	1.875	1.68	1.55

DGM is more expensive than FEM, the ratio is smaller for larger p

Numerical analysis

Asymptotical order of convergence

Poisson problem & scalar nonlinear convection-diffusion equation

$$|u - u_h|_{H^1(\Omega)} = O(h^p), \qquad \|u - u_h\|_{L^2(\Omega)} = O(h^{p+1})$$

the same order of convergence as for FEM

Number of degrees of freedom (DOF)

let $d = 2$, $M = \#\mathscr{T}_h$

p	1	2	3	4	5	6
FEM	$\frac{1}{2}M$	$2M$	$\frac{9}{2}$	$8M$	$\frac{25}{2}M$	$18M$
DGM	$3M$	$6M$	$10M$	$15M$	$21M$	$28M$
ratio$= \frac{\text{DGM}}{\text{FEM}}$	6	3	2.22	1.875	1.68	1.55

DGM is more expensive than FEM, the ratio is smaller for larger p

Numerical analysis

Asymptotical order of convergence

Poisson problem & scalar nonlinear convection-diffusion equation

$$|u - u_h|_{H^1(\Omega)} = O(h^p), \qquad \|u - u_h\|_{L^2(\Omega)} = O(h^{p+1})$$

the same order of convergence as for FEM

Number of degrees of freedom (DOF)

let $d = 2$, $M = \#\mathscr{T}_h$

p	1	2	3	4	5	6
FEM	$\frac{1}{2}M$	$2M$	$\frac{9}{2}$	$8M$	$\frac{25}{2}M$	$18M$
DGM	$3M$	$6M$	$10M$	$15M$	$21M$	$28M$
ratio= $\frac{\mathrm{DGM}}{\mathrm{FEM}}$	6	3	2.22	1.875	1.68	1.55

DGM is more expensive than FEM, the ratio is smaller for larger p

Numerical analysis

Asymptotical order of convergence

Poisson problem & scalar nonlinear convection-diffusion equation

$$|u - u_h|_{H^1(\Omega)} = O(h^p), \qquad \|u - u_h\|_{L^2(\Omega)} = O(h^{p+1})$$

the same order of convergence as for FEM

Number of degrees of freedom (DOF)

let $d = 2$, $M = \#\mathscr{T}_h$

p	1	2	3	4	5	6
FEM	$\frac{1}{2}M$	$2M$	$\frac{9}{2}$	$8M$	$\frac{25}{2}M$	$18M$
DGM	$3M$	$6M$	$10M$	$15M$	$21M$	$28M$
ratio$=\frac{\mathrm{DGM}}{\mathrm{FEM}}$	6	3	2.22	1.875	1.68	1.55

DGM is more expensive than FEM, the ratio is smaller for larger p

Numerical analysis

Asymptotical order of convergence

Poisson problem & scalar nonlinear convection-diffusion equation

$$|u - u_h|_{H^1(\Omega)} = O(h^p), \qquad \|u - u_h\|_{L^2(\Omega)} = O(h^{p+1})$$

the same order of convergence as for FEM

Number of degrees of freedom (DOF)

let $d = 2$, $M = \#\mathscr{T}_h$

p	1	2	3	4	5	6
FEM	$\frac{1}{2}M$	$2M$	$\frac{9}{2}$	$8M$	$\frac{25}{2}M$	$18M$
DGM	$3M$	$6M$	$10M$	$15M$	$21M$	$28M$
ratio= $\frac{\mathrm{DGM}}{\mathrm{FEM}}$	6	3	2.22	1.875	1.68	1.55

DGM is more expensive than FEM, the ratio is smaller for larger p

Numerical analysis

Asymptotical order of convergence

Poisson problem & scalar nonlinear convection-diffusion equation

$$|u - u_h|_{H^1(\Omega)} = O(h^p), \qquad \|u - u_h\|_{L^2(\Omega)} = O(h^{p+1})$$

the same order of convergence as for FEM

Number of degrees of freedom (DOF)

let $d = 2$, $M = \#\mathscr{T}_h$

p	1	2	3	4	5	6
FEM	$\frac{1}{2}M$	$2M$	$\frac{9}{2}$	$8M$	$\frac{25}{2}M$	$18M$
DGM	$3M$	$6M$	$10M$	$15M$	$21M$	$28M$
ratio$=\frac{\mathrm{DGM}}{\mathrm{FEM}}$	6	3	2.22	1.875	1.68	1.55

DGM is more expensive than FEM, the ratio is smaller for larger p

Numerical analysis

Asymptotical order of convergence

Poisson problem & scalar nonlinear convection-diffusion equation

$$|u - u_h|_{H^1(\Omega)} = O(h^p), \qquad \|u - u_h\|_{L^2(\Omega)} = O(h^{p+1})$$

the same order of convergence as for FEM

Number of degrees of freedom (DOF)

let $d = 2$, $M = \#\mathscr{T}_h$

p	1	2	3	4	5	6
FEM	$\frac{1}{2}M$	$2M$	$\frac{9}{2}$	$8M$	$\frac{25}{2}M$	$18M$
DGM	$3M$	$6M$	$10M$	$15M$	$21M$	$28M$
ratio $= \frac{\mathrm{DGM}}{\mathrm{FEM}}$	6	3	2.22	1.875	1.68	1.55

DGM is more expensive than FEM, the ratio is smaller for larger p

Numerical analysis

Asymptotical order of convergence

Poisson problem & scalar nonlinear convection-diffusion equation

$$|u - u_h|_{H^1(\Omega)} = O(h^p), \qquad \|u - u_h\|_{L^2(\Omega)} = O(h^{p+1})$$

the same order of convergence as for FEM

Number of degrees of freedom (DOF)

let $d = 2$, $M = \#\mathscr{T}_h$

p	1	2	3	4	5	6
FEM	$\frac{1}{2}M$	$2M$	$\frac{9}{2}$	$8M$	$\frac{25}{2}M$	$18M$
DGM	$3M$	$6M$	$10M$	$15M$	$21M$	$28M$
ratio= $\frac{\mathrm{DGM}}{\mathrm{FEM}}$	6	3	2.22	1.875	1.68	1.55

DGM is more expensive than FEM, the ratio is smaller for larger p

Numerical analysis

Asymptotical order of convergence

Poisson problem & scalar nonlinear convection-diffusion equation

$$|u - u_h|_{H^1(\Omega)} = O(h^p), \qquad \|u - u_h\|_{L^2(\Omega)} = O(h^{p+1})$$

the same order of convergence as for FEM

Number of degrees of freedom (DOF)

let $d = 2$, $M = \#\mathscr{T}_h$

p	1	2	3	4	5	6
FEM	$\frac{1}{2}M$	$2M$	$\frac{9}{2}$	$8M$	$\frac{25}{2}M$	$18M$
DGM	$3M$	$6M$	$10M$	$15M$	$21M$	$28M$
ratio$=\frac{\mathrm{DGM}}{\mathrm{FEM}}$	6	3	2.22	1.875	1.68	1.55

DGM is more expensive than FEM, the ratio is smaller for larger p

Algebraic representation

Space-time discretization

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, \dots \quad (2)$$

- system of nonlinear algebraic equations for each k, let $\mathbf{w}_h^k \in \mathbf{S}_{hp} \leftrightarrow \xi_k \in \mathbb{R}^{\mathrm{DOF}}$, $\mathrm{DOF} = \dim \mathbf{S}_{hp}$, (2) is equivalent to

$$\frac{1}{\tau_k}\mathbb{M}\xi_k + \mathbf{F}(\xi_k) = \frac{1}{\tau_k}\mathbb{M}\xi_{k-1}, \ k = 1, 2, \dots$$

- Newton-like method uses $\mathbf{F}(\xi_k) \approx \mathbb{C}(\xi_k)\xi_k - \mathbf{b}(\xi_k)$,
- $\mathbb{M}$ and $\mathbb{C}$ sparse block-structure matrices

Algebraic representation

Space-time discretization

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, \dots \quad (2)$$

- system of nonlinear algebraic equations for each k,
  let $\mathbf{w}_h^k \in \mathbf{S}_{hp} \leftrightarrow \xi_k \in \mathbb{R}^{\mathrm{DOF}}$, $\mathrm{DOF} = \dim \mathbf{S}_{hp}$, (2) is equivalent to

$$\frac{1}{\tau_k}\mathbb{M}\xi_k + \mathbf{F}(\xi_k) = \frac{1}{\tau_k}\mathbb{M}\xi_{k-1}, \ k = 1, 2, \dots$$

- Newton-like method uses $\mathbf{F}(\xi_k) \approx \mathbb{C}(\xi_k)\xi_k - \mathbf{b}(\xi_k)$,
- $\mathbb{M}$ and $\mathbb{C}$ sparse block-structure matrices

Algebraic representation

Space-time discretization

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, \dots \quad (2)$$

- system of nonlinear algebraic equations for each k, let $\mathbf{w}_h^k \in \mathbf{S}_{hp} \leftrightarrow \boldsymbol{\xi}_k \in \mathbb{R}^{\mathrm{DOF}}$, $\mathrm{DOF} = \dim \mathbf{S}_{hp}$, (2) is equivalent to

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ k = 1, 2, \dots$$

- Newton-like method uses $\mathbf{F}(\boldsymbol{\xi}_k) \approx \mathbb{C}(\boldsymbol{\xi}_k)\boldsymbol{\xi}_k - \mathbf{b}(\boldsymbol{\xi}_k)$,
- $\mathbb{M}$ and $\mathbb{C}$ sparse block-structure matrices

Algebraic representation

Space-time discretization

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, \dots \quad (2)$$

- system of nonlinear algebraic equations for each k, let $\mathbf{w}_h^k \in \mathbf{S}_{hp} \leftrightarrow \boldsymbol{\xi}_k \in \mathbb{R}^{\mathrm{DOF}}$, $\mathrm{DOF} = \dim \mathbf{S}_{hp}$, (2) is equivalent to

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ k = 1, 2, \dots$$

- Newton-like method uses $\mathbf{F}(\xi_k) \approx \mathbb{C}(\xi_k)\xi_k - \mathbf{b}(\xi_k)$,
- $\mathbb{M}$ and $\mathbb{C}$ sparse block-structure matrices

Algebraic representation

Space-time discretization

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, \dots \quad (2)$$

- system of nonlinear algebraic equations for each k, let $\mathbf{w}_h^k \in \mathbf{S}_{hp} \leftrightarrow \boldsymbol{\xi}_k \in \mathbb{R}^{\mathrm{DOF}}$, $\mathrm{DOF} = \dim \mathbf{S}_{hp}$, (2) is equivalent to

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ k = 1, 2, \dots$$

- Newton-like method uses $\mathbf{F}(\boldsymbol{\xi}_k) \approx \mathbb{C}(\boldsymbol{\xi}_k)\boldsymbol{\xi}_k - \mathbf{b}(\boldsymbol{\xi}_k)$,
- $\mathbb{M}$ and $\mathbb{C}$ sparse block-structure matrices

Algebraic representation

Space-time discretization

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, \dots \quad (2)$$

- system of nonlinear algebraic equations for each k,
  let $\mathbf{w}_h^k \in \mathbf{S}_{hp} \leftrightarrow \boldsymbol{\xi}_k \in \mathbb{R}^{\mathrm{DOF}}$, $\mathrm{DOF} = \dim \mathbf{S}_{hp}$, (2) is equivalent to

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ k = 1, 2, \dots$$

- Newton-like method uses $\mathbf{F}(\xi_k) \approx \mathbb{C}(\xi_k)\xi_k - \mathbf{b}(\xi_k)$,
- $\mathbb{M}$ and $\mathbb{C}$ sparse block-structure matrices

Algebraic representation

Space-time discretization

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, \dots \quad (2)$$

- system of nonlinear algebraic equations for each k, let $\mathbf{w}_h^k \in \mathbf{S}_{hp} \leftrightarrow \boldsymbol{\xi}_k \in \mathbb{R}^{\mathrm{DOF}}$, $\mathrm{DOF} = \dim \mathbf{S}_{hp}$, (2) is equivalent to

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ k = 1, 2, \dots$$

- Newton-like method uses $\mathbf{F}(\xi_k) \approx \mathbb{C}(\xi_k)\xi_k - \mathbf{b}(\xi_k)$,
- $\mathbb{M}$ and $\mathbb{C}$ sparse block-structure matrices

Algebraic representation

Space-time discretization

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, \dots \quad (2)$$

- system of nonlinear algebraic equations for each k, let $\mathbf{w}_h^k \in \mathbf{S}_{hp} \leftrightarrow \boldsymbol{\xi}_k \in \mathbb{R}^{\mathrm{DOF}}$, $\mathrm{DOF} = \dim \mathbf{S}_{hp}$, (2) is equivalent to

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ k = 1, 2, \dots$$

- Newton-like method uses $\mathbf{F}(\boldsymbol{\xi}_k) \approx \mathbb{C}(\boldsymbol{\xi}_k)\boldsymbol{\xi}_k - \mathbf{b}(\boldsymbol{\xi}_k)$,
- $\mathbb{M}$ and $\mathbb{C}$ sparse block-structure matrices

Algebraic representation

Space-time discretization

$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp}, \ k = 1, \dots \quad (2)$$

- system of nonlinear algebraic equations for each k, let $\mathbf{w}_h^k \in \mathbf{S}_{hp} \leftrightarrow \boldsymbol{\xi}_k \in \mathbb{R}^{\mathrm{DOF}}$, $\mathrm{DOF} = \dim \mathbf{S}_{hp}$, (2) is equivalent to

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ k = 1, 2, \dots$$

- Newton-like method uses $\mathbf{F}(\boldsymbol{\xi}_k) \approx \mathbb{C}(\boldsymbol{\xi}_k)\boldsymbol{\xi}_k - \mathbf{b}(\boldsymbol{\xi}_k)$,
- $\mathbb{M}$ and $\mathbb{C}$ sparse block-structure matrices

Algebraic representation

Space-time discretization

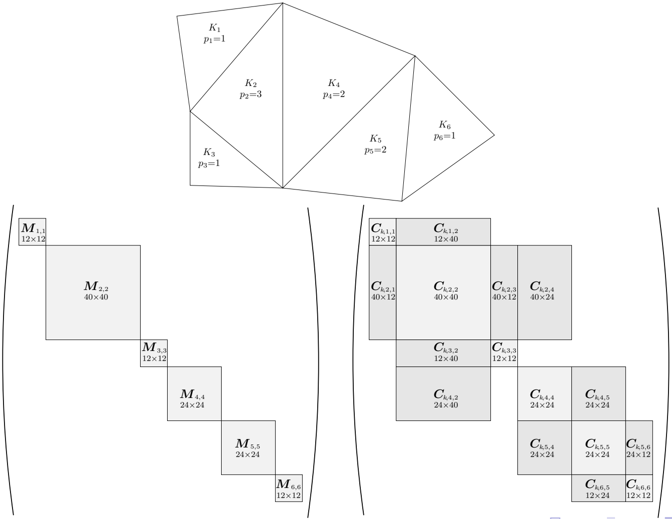
$$\frac{1}{\tau_k}\left(\mathbf{w}_h^k - \mathbf{w}_h^{k-1}, \varphi_h\right) + \mathbf{c}_h(\mathbf{w}_h^k, \varphi_h) = 0 \quad \forall \varphi_h \in \mathbf{S}_{hp},\ k = 1, \dots \quad (2)$$

- system of nonlinear algebraic equations for each k, let $\mathbf{w}_h^k \in \mathbf{S}_{hp} \leftrightarrow \boldsymbol{\xi}_k \in \mathbb{R}^{\mathrm{DOF}}$, $\mathrm{DOF} = \dim \mathbf{S}_{hp}$, (2) is equivalent to

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1},\ k = 1, 2, \dots$$

- Newton-like method uses $\mathbf{F}(\boldsymbol{\xi}_k) \approx \mathbb{C}(\boldsymbol{\xi}_k)\boldsymbol{\xi}_k - \mathbf{b}(\boldsymbol{\xi}_k)$,
- $\mathbb{M}$ and $\mathbb{C}$ sparse block-structure matrices

Block structure of $\mathbb{M}$ and $\mathbb{C}$



Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \;\; k = 1, 2, \ldots$$

solved iteratively by a Newton-like method: $\xi_k^1, \xi_k^2, \ldots$

$$\xi_k^{l+1} := \xi_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\xi_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\xi_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ \ k = 1, 2, \dots$$

solved iteratively by a Newton-like method: $\xi_k^1, \xi_k^2, \dots$

$$\xi_k^{l+1} := \xi_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\xi_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\xi_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ k = 1, 2, \dots$$

solved iteratively by a Newton-like method: ξ_k^1, ξ_k^2, ...

$$\xi_k^{l+1} := \xi_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\xi_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\xi_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \quad k = 1, 2, \ldots$$

solved iteratively by a Newton-like method: $\xi_k^1, \xi_k^2, \ldots$

$$\xi_k^{l+1} := \xi_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\xi_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\xi_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \quad k = 1, 2, \ldots$$

solved iteratively by a Newton-like method: $\boldsymbol{\xi}_k^1, \boldsymbol{\xi}_k^2, \ldots$

$$\boldsymbol{\xi}_k^{l+1} := \boldsymbol{\xi}_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\boldsymbol{\xi}_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\boldsymbol{\xi}_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ k = 1, 2, \dots$$

solved iteratively by a Newton-like method: $\xi_k^1, \xi_k^2, \dots$

$$\xi_k^{l+1} := \xi_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\xi_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\xi_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ \ k = 1, 2, \dots$$

solved iteratively by a Newton-like method: $\boldsymbol{\xi}_k^1, \boldsymbol{\xi}_k^2, \dots$

$$\boldsymbol{\xi}_k^{l+1} := \boldsymbol{\xi}_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\boldsymbol{\xi}_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\boldsymbol{\xi}_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \; k = 1, 2, \ldots$$

solved iteratively by a Newton-like method: $\boldsymbol{\xi}_k^1$, $\boldsymbol{\xi}_k^2$, ...

$$\boldsymbol{\xi}_k^{l+1} := \boldsymbol{\xi}_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\boldsymbol{\xi}_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\boldsymbol{\xi}_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \;\; k = 1, 2, \dots$$

solved iteratively by a Newton-like method: $\xi_k^1, \xi_k^2, \dots$

$$\xi_k^{l+1} := \xi_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\xi_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\xi_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \; k = 1, 2, \dots$$

solved iteratively by a Newton-like method: $\xi_k^1, \xi_k^2, \dots$

$$\xi_k^{l+1} := \xi_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\xi_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\xi_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \;\; k = 1, 2, \ldots$$

solved iteratively by a Newton-like method: $\xi_k^1, \xi_k^2, \ldots$

$$\xi_k^{l+1} := \xi_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\xi_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\xi_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \; k = 1, 2, \dots$$

solved iteratively by a Newton-like method: $\boldsymbol{\xi}_k^1, \boldsymbol{\xi}_k^2, \dots$

$$\boldsymbol{\xi}_k^{l+1} := \boldsymbol{\xi}_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\boldsymbol{\xi}_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\boldsymbol{\xi}_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ k = 1, 2, \dots$$

solved iteratively by a Newton-like method: ξ_k^1, ξ_k^2, ...

$$\xi_k^{l+1} := \xi_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\xi_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\xi_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Iterative solvers

Nonlinear algebraic problem

$$\frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_k + \mathbf{F}(\boldsymbol{\xi}_k) = \frac{1}{\tau_k}\mathbb{M}\boldsymbol{\xi}_{k-1}, \ \ k = 1, 2, \dots$$

solved iteratively by a Newton-like method: $\boldsymbol{\xi}_k^1, \boldsymbol{\xi}_k^2, \dots$

$$\boldsymbol{\xi}_k^{l+1} := \boldsymbol{\xi}_k^l + \mathbf{d}^l, \qquad \tilde{\mathbb{C}}(\boldsymbol{\xi}_k^{l-1})\mathbf{d}^l = \tilde{\mathbf{b}}(\boldsymbol{\xi}_k^{l-1})$$

Linear algebraic problems

- suitable to solved iteratively (for large systems), GMRES,
- block ILU(0) preconditioning, only $4\#\mathscr{T}_h$ block operations,
- suitable stopping criterion, related to
  - nonlinear algebraic stopping criterion
  - discretization error

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,
  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}$, $l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation
  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,
  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}$, $l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation
  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,

  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}, \ l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation

  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,
  $S_h^{p-l} := \{v;\, v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}, \ l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation
  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,

  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}$, $l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation

  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,

  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}$, $l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation

  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,
  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}$, $l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation
  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
    - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
    - restriction of the residuum (defect) on $\mathscr{T}_H$,
    - solution a smaller problem on $\mathscr{T}_H$,
    - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,
  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}$, $l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation
  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,
  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}$, $l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation
  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,

  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}, \ l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation

  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,

  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}, \ l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation

  [talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,
  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}, \ l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation

[talk of A. Živčák]

Multigrid method

Linear multigrid

- h-multigrid is based on hierarchy of grids $\mathscr{T}_h$, $h \in \{h_0, h_1, \dots\}$
- let $\mathscr{T}_h$ and $\mathscr{T}_H$ be two grids, $H > h$,
  - smoothing: few linear algebraic iterations on $\mathscr{T}_h$,
  - restriction of the residuum (defect) on $\mathscr{T}_H$,
  - solution a smaller problem on $\mathscr{T}_H$,
  - prolongation on $\mathscr{T}_h$ and further smoothing.

DGM perfectly suits for p-multigrid

- hierarchy of grids $\mathscr{T}_h$ is replaced by hierarchy of spaces S_h^p,

  $S_h^{p-l} := \{v; v \in L^2(\Omega), v|_K \in P_{p-l}(K) \forall K \in \mathscr{T}_h\}, \; l = 0, 1, \dots$
- L^2-orthogonal basis $\Rightarrow$ simple restriction and prolongation

  [talk of A. Živčák]

Parallelization

Solution of complex problems requires parallelization

- suitable discretization technique
- suitable algebraic solvers

Parallelization of DGM

- DGM perfectly suits for parallelization
- DGM allows to distribute efficiently the computation and required memory among the processors
- interchange of data among processors is minimal
- evaluation of one block-row requires data from neighbouring elements only from adjacent faces

Parallelization

Solution of complex problems requires parallelization

- suitable discretization technique
- suitable algebraic solvers

Parallelization of DGM

- DGM perfectly suits for parallelization
- DGM allows to distribute efficiently the computation and required memory among the processors
- interchange of data among processors is minimal
- evaluation of one block-row requires data from neighbouring elements only from adjacent faces

Parallelization

Solution of complex problems requires parallelization

- suitable discretization technique
- suitable algebraic solvers

Parallelization of DGM

- DGM perfectly suits for parallelization
- DGM allows to distribute efficiently the computation and required memory among the processors
- interchange of data among processors is minimal
- evaluation of one block-row requires data from neighbouring elements only from adjacent faces

Parallelization

Solution of complex problems requires parallelization

- suitable discretization technique
- suitable algebraic solvers

Parallelization of DGM

- DGM perfectly suits for parallelization
- DGM allows to distribute efficiently the computation and required memory among the processors
- interchange of data among processors is minimal
- evaluation of one block-row requires data from neighbouring elements only from adjacent faces

Parallelization

Solution of complex problems requires parallelization

- suitable discretization technique
- suitable algebraic solvers

Parallelization of DGM

- DGM perfectly suits for parallelization
- DGM allows to distribute efficiently the computation and required memory among the processors
- interchange of data among processors is minimal
- evaluation of one block-row requires data from neighbouring elements only from adjacent faces

Parallelization

Solution of complex problems requires parallelization

- suitable discretization technique
- suitable algebraic solvers

Parallelization of DGM

- DGM perfectly suits for parallelization
- DGM allows to distribute efficiently the computation and required memory among the processors
- interchange of data among processors is minimal
- evaluation of one block-row requires data from neighbouring elements only from adjacent faces

Parallelization

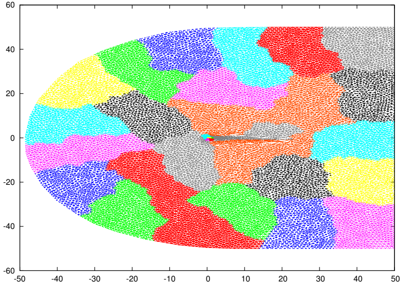
Solution of complex problems requires parallelization

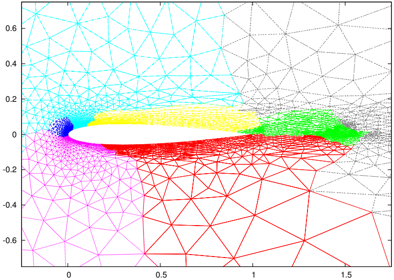
- suitable discretization technique
- suitable algebraic solvers

Parallelization of DGM

- DGM perfectly suits for parallelization
- DGM allows to distribute efficiently the computation and required memory among the processors
- interchange of data among processors is minimal
- evaluation of one block-row requires data from neighbouring elements only from adjacent faces

Mesh decomposition [V. Šíp, Master thesis 2011]





mesh decomposition among 32 processors

Efficiency of parallelization [V. Šíp, Master thesis 2011]

mesh	$\#\mathscr{T}$	P_1 DOF	P_3 DOF
$\mathscr{T}_{h1}$	6 876	82 512	275 040
$\mathscr{T}_{h2}$	29 040	348 480	1 161 600

π-number of processors, (1,2,4,8,16,32),
$S_\pi = T_1/T_\pi$ - acceleration,
$E_\pi = S_\pi/\pi$ - efficiency,
ideal case: $S_\pi = \pi$ & $E_\pi = 1$

		$\mathscr{T}_{h1}$			$\mathscr{T}_{h2}$		
	π	T_π	S_π	E_π	T_π	S_π	E_π
P_1	1	102.43 s	1.00	1.00	539.73 s	1.00	1.00
	2	50.75 s	2.02	1.01	258.16 s	2.09	1.05
	4	25.90 s	3.95	0.99	131.49 s	4.10	1.03
	8	15.07 s	6.80	0.85	65.80 s	8.20	1.03
	16	9.79 s	10.46	0.65	35.21 s	15.33	0.96
	32	7.77 s	13.19	0.41	26.75 s	20.18	0.63
P_3	1	1278.27 s	1.00	1.00	5988.53 s	1.00	1.00
	2	651.42 s	1.96	0.98	3026.76 s	1.98	0.99
	4	310.50 s	4.12	1.03	1526.50 s	3.92	0.98
	8	156.58 s	8.16	1.02	760.79 s	7.87	0.98
	16	87.28 s	14.65	0.92	395.66 s	15.13	0.94
	32	50.07 s	25.53	0.80	206.64 s	28.98	0.91

Efficiency of parallelization [V. Šíp, Master thesis 2011]

mesh	$\#\mathscr{T}$	P_1 DOF	P_3 DOF
$\mathscr{T}_{h1}$	6 876	82 512	275 040
$\mathscr{T}_{h2}$	29 040	348 480	1 161 600

π-number of processors, (1,2,4,8,16,32),
$S_\pi = T_1/T_\pi$ - acceleration,
$E_\pi = S_\pi/\pi$ - efficiency,
ideal case: $S_\pi = \pi$ & $E_\pi = 1$

		$\mathscr{T}_{h1}$			$\mathscr{T}_{h2}$		
	π	T_π	S_π	E_π	T_π	S_π	E_π
P_1	1	102.43 s	1.00	1.00	539.73 s	1.00	1.00
	2	50.75 s	2.02	1.01	258.16 s	2.09	1.05
	4	25.90 s	3.95	0.99	131.49 s	4.10	1.03
	8	15.07 s	6.80	0.85	65.80 s	8.20	1.03
	16	9.79 s	10.46	0.65	35.21 s	15.33	0.96
	32	7.77 s	13.19	0.41	26.75 s	20.18	0.63
P_3	1	1278.27 s	1.00	1.00	5988.53 s	1.00	1.00
	2	651.42 s	1.96	0.98	3026.76 s	1.98	0.99
	4	310.50 s	4.12	1.03	1526.50 s	3.92	0.98
	8	156.58 s	8.16	1.02	760.79 s	7.87	0.98
	16	87.28 s	14.65	0.92	395.66 s	15.13	0.94
	32	50.07 s	25.53	0.80	206.64 s	28.98	0.91

Efficiency of parallelization [V. Šíp, Master thesis 2011]

mesh	$\#\mathscr{T}$	P_1 DOF	P_3 DOF
$\mathscr{T}_{h1}$	6 876	82 512	275 040
$\mathscr{T}_{h2}$	29 040	348 480	1 161 600

π-number of processors, (1,2,4,8,16,32),
$S_\pi = T_1/T_\pi$ - acceleration,
$E_\pi = S_\pi/\pi$ - efficiency,
ideal case: $S_\pi = \pi$ & $E_\pi = 1$

		$\mathscr{T}_{h1}$			$\mathscr{T}_{h2}$		
	π	T_π	S_π	E_π	T_π	S_π	E_π
P_1	1	102.43 s	1.00	1.00	539.73 s	1.00	1.00
	2	50.75 s	2.02	1.01	258.16 s	2.09	1.05
	4	25.90 s	3.95	0.99	131.49 s	4.10	1.03
	8	15.07 s	6.80	0.85	65.80 s	8.20	1.03
	16	9.79 s	10.46	0.65	35.21 s	15.33	0.96
	32	7.77 s	13.19	0.41	26.75 s	20.18	0.63
P_3	1	1278.27 s	1.00	1.00	5988.53 s	1.00	1.00
	2	651.42 s	1.96	0.98	3026.76 s	1.98	0.99
	4	310.50 s	4.12	1.03	1526.50 s	3.92	0.98
	8	156.58 s	8.16	1.02	760.79 s	7.87	0.98
	16	87.28 s	14.65	0.92	395.66 s	15.13	0.94
	32	50.07 s	25.53	0.80	206.64 s	28.98	0.91

Efficiency of parallelization [V. Šíp, Master thesis 2011]

mesh	$\#\mathscr{T}$	P_1 DOF	P_3 DOF
$\mathscr{T}_{h1}$	6 876	82 512	275 040
$\mathscr{T}_{h2}$	29 040	348 480	1 161 600

π-number of processors, (1,2,4,8,16,32),
$S_\pi = T_1/T_\pi$ - acceleration,
$E_\pi = S_\pi/\pi$ - efficiency,
ideal case: $S_\pi = \pi$ & $E_\pi = 1$

		$\mathscr{T}_{h1}$			$\mathscr{T}_{h2}$		
	π	T_π	S_π	E_π	T_π	S_π	E_π
P_1	1	102.43 s	1.00	1.00	539.73 s	1.00	1.00
	2	50.75 s	2.02	1.01	258.16 s	2.09	1.05
	4	25.90 s	3.95	0.99	131.49 s	4.10	1.03
	8	15.07 s	6.80	0.85	65.80 s	8.20	1.03
	16	9.79 s	10.46	0.65	35.21 s	15.33	0.96
	32	7.77 s	13.19	0.41	26.75 s	20.18	0.63
P_3	1	1278.27 s	1.00	1.00	5988.53 s	1.00	1.00
	2	651.42 s	1.96	0.98	3026.76 s	1.98	0.99
	4	310.50 s	4.12	1.03	1526.50 s	3.92	0.98
	8	156.58 s	8.16	1.02	760.79 s	7.87	0.98
	16	87.28 s	14.65	0.92	395.66 s	15.13	0.94
	32	50.07 s	25.53	0.80	206.64 s	28.98	0.91

Efficiency of parallelization [V. Šíp, Master thesis 2011]

mesh	$\#\mathscr{T}$	P_1 DOF	P_3 DOF
$\mathscr{T}_{h1}$	6 876	82 512	275 040
$\mathscr{T}_{h2}$	29 040	348 480	1 161 600

π-number of processors, (1,2,4,8,16,32),
$S_\pi = T_1/T_\pi$ - acceleration,
$E_\pi = S_\pi/\pi$ - efficiency,
ideal case: $S_\pi = \pi$ & $E_\pi = 1$

		$\mathscr{T}_{h1}$			$\mathscr{T}_{h2}$		
	π	T_π	S_π	E_π	T_π	S_π	E_π
P_1	1	102.43 s	1.00	1.00	539.73 s	1.00	1.00
	2	50.75 s	2.02	1.01	258.16 s	2.09	1.05
	4	25.90 s	3.95	0.99	131.49 s	4.10	1.03
	8	15.07 s	6.80	0.85	65.80 s	8.20	1.03
	16	9.79 s	10.46	0.65	35.21 s	15.33	0.96
	32	7.77 s	13.19	0.41	26.75 s	20.18	0.63
P_3	1	1278.27 s	1.00	1.00	5988.53 s	1.00	1.00
	2	651.42 s	1.96	0.98	3026.76 s	1.98	0.99
	4	310.50 s	4.12	1.03	1526.50 s	3.92	0.98
	8	156.58 s	8.16	1.02	760.79 s	7.87	0.98
	16	87.28 s	14.65	0.92	395.66 s	15.13	0.94
	32	50.07 s	25.53	0.80	206.64 s	28.98	0.91

Efficiency of parallelization [V. Šíp, Master thesis 2011]

mesh	$\#\mathscr{T}$	P_1 DOF	P_3 DOF
$\mathscr{T}_{h1}$	6 876	82 512	275 040
$\mathscr{T}_{h2}$	29 040	348 480	1 161 600

π-number of processors, (1,2,4,8,16,32),
$S_\pi = T_1/T_\pi$ - acceleration,
$E_\pi = S_\pi/\pi$ - efficiency,
ideal case: $S_\pi = \pi$ & $E_\pi = 1$

		$\mathscr{T}_{h1}$			$\mathscr{T}_{h2}$		
	π	T_π	S_π	E_π	T_π	S_π	E_π
P_1	1	102.43 s	1.00	1.00	539.73 s	1.00	1.00
	2	50.75 s	2.02	1.01	258.16 s	2.09	1.05
	4	25.90 s	3.95	0.99	131.49 s	4.10	1.03
	8	15.07 s	6.80	0.85	65.80 s	8.20	1.03
	16	9.79 s	10.46	0.65	35.21 s	15.33	0.96
	32	7.77 s	13.19	0.41	26.75 s	20.18	0.63
P_3	1	1278.27 s	1.00	1.00	5988.53 s	1.00	1.00
	2	651.42 s	1.96	0.98	3026.76 s	1.98	0.99
	4	310.50 s	4.12	1.03	1526.50 s	3.92	0.98
	8	156.58 s	8.16	1.02	760.79 s	7.87	0.98
	16	87.28 s	14.65	0.92	395.66 s	15.13	0.94
	32	50.07 s	25.53	0.80	206.64 s	28.98	0.91

Efficiency of parallelization [V. Šíp, Master thesis 2011]

mesh	$\#\mathscr{T}$	P_1 DOF	P_3 DOF
$\mathscr{T}_{h1}$	6 876	82 512	275 040
$\mathscr{T}_{h2}$	29 040	348 480	1 161 600

π-number of processors, (1,2,4,8,16,32),
$S_\pi = T_1/T_\pi$ - acceleration,
$E_\pi = S_\pi/\pi$ - efficiency,
ideal case: $S_\pi = \pi$ & $E_\pi = 1$

		$\mathscr{T}_{h1}$			$\mathscr{T}_{h2}$		
	π	T_π	S_π	E_π	T_π	S_π	E_π
P_1	1	102.43 s	1.00	1.00	539.73 s	1.00	1.00
	2	50.75 s	2.02	1.01	258.16 s	2.09	1.05
	4	25.90 s	3.95	0.99	131.49 s	4.10	1.03
	8	15.07 s	6.80	0.85	65.80 s	8.20	1.03
	16	9.79 s	10.46	0.65	35.21 s	15.33	0.96
	32	7.77 s	13.19	0.41	26.75 s	20.18	0.63
P_3	1	1278.27 s	1.00	1.00	5988.53 s	1.00	1.00
	2	651.42 s	1.96	0.98	3026.76 s	1.98	0.99
	4	310.50 s	4.12	1.03	1526.50 s	3.92	0.98
	8	156.58 s	8.16	1.02	760.79 s	7.87	0.98
	16	87.28 s	14.65	0.92	395.66 s	15.13	0.94
	32	50.07 s	25.53	0.80	206.64 s	28.98	0.91

Efficiency of parallelization [V. Šíp, Master thesis 2011]

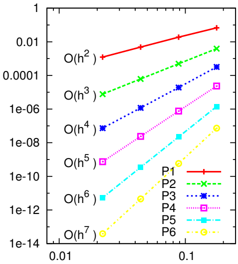
mesh	$\#\mathscr{T}$	P_1 DOF	P_3 DOF
$\mathscr{T}_{h1}$	6 876	82 512	275 040
$\mathscr{T}_{h2}$	29 040	348 480	1 161 600

π-number of processors, (1,2,4,8,16,32),
$S_\pi = T_1/T_\pi$ - acceleration,
$E_\pi = S_\pi/\pi$ - efficiency,
ideal case: $S_\pi = \pi$ & $E_\pi = 1$

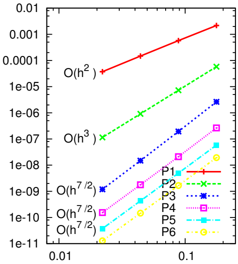
		$\mathscr{T}_{h1}$			$\mathscr{T}_{h2}$		
	π	T_π	S_π	E_π	T_π	S_π	E_π
P_1	1	102.43 s	1.00	1.00	539.73 s	1.00	1.00
	2	50.75 s	2.02	1.01	258.16 s	2.09	1.05
	4	25.90 s	3.95	0.99	131.49 s	4.10	1.03
	8	15.07 s	6.80	0.85	65.80 s	8.20	1.03
	16	9.79 s	10.46	0.65	35.21 s	15.33	0.96
	32	7.77 s	13.19	0.41	26.75 s	20.18	0.63
P_3	1	1278.27 s	1.00	1.00	5988.53 s	1.00	1.00
	2	651.42 s	1.96	0.98	3026.76 s	1.98	0.99
	4	310.50 s	4.12	1.03	1526.50 s	3.92	0.98
	8	156.58 s	8.16	1.02	760.79 s	7.87	0.98
	16	87.28 s	14.65	0.92	395.66 s	15.13	0.94
	32	50.07 s	25.53	0.80	206.64 s	28.98	0.91

Orders of convergence

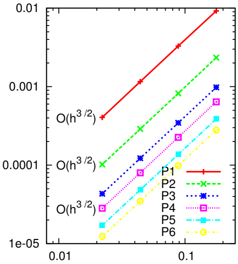
- higher order schemes are efficient
- BUT only if the exact solution is sufficiently regular
- $\|u - u_h\|_{L^2} = O(h^{\mu})$, $\mu = \min(p+1, s)$, $u_h \in S_{hp}$, $u \in H^s(\Omega)$



$u \in C^{\infty}(\Omega)$



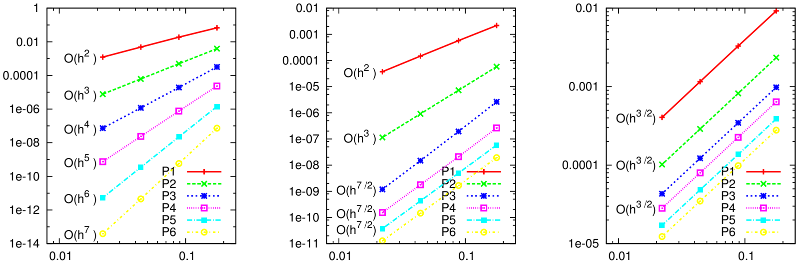
$u \in H^{3.5}(\Omega)$



$u \in H^{1.5}(\Omega)$

Orders of convergence

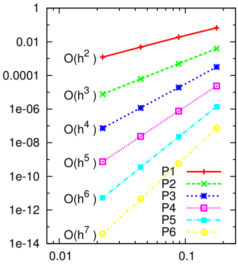
- higher order schemes are efficient
- BUT only if the exact solution is sufficiently regular
- $\|u - u_h\|_{L^2} = O(h^{\mu})$, $\mu = \min(p+1, s)$, $u_h \in S_{hp}$, $u \in H^s(\Omega)$



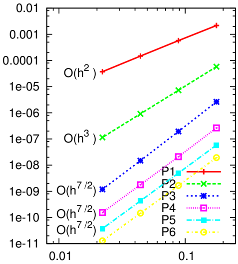
$u \in C^{\infty}(\Omega)$ $u \in H^{3.5}(\Omega)$ $u \in H^{1.5}(\Omega)$

Orders of convergence

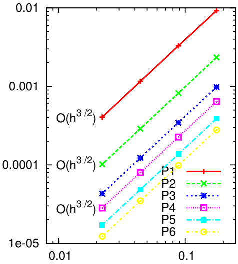
- higher order schemes are efficient
- BUT only if the exact solution is sufficiently regular
- $\|u - u_h\|_{L^2} = O(h^\mu)$, $\mu = \min(p+1, s)$, $u_h \in S_{hp}$, $u \in H^s(\Omega)$



$u \in C^\infty(\Omega)$



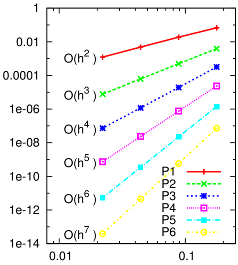
$u \in H^{3.5}(\Omega)$



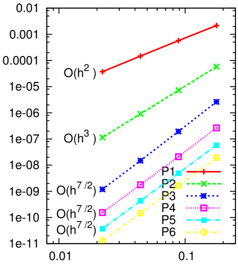
$u \in H^{1.5}(\Omega)$

Orders of convergence

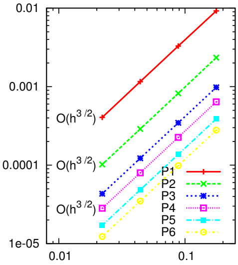
- higher order schemes are efficient
- BUT only if the exact solution is sufficiently regular
- $\|u - u_h\|_{L^2} = O(h^{\mu})$, $\mu = \min(p+1, s)$, $u_h \in S_{hp}$, $u \in H^s(\Omega)$



$u \in C^{\infty}(\Omega)$



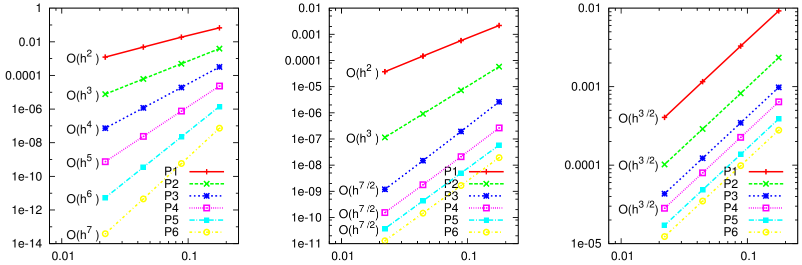
$u \in H^{3.5}(\Omega)$



$u \in H^{1.5}(\Omega)$

Orders of convergence

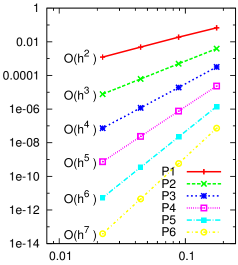
- higher order schemes are efficient
- BUT only if the exact solution is sufficiently regular
- $\|u - u_h\|_{L^2} = O(h^{\mu})$, $\mu = \min(p+1, s)$, $u_h \in S_{hp}$, $u \in H^s(\Omega)$



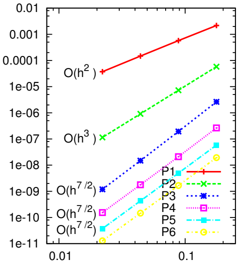
$u \in C^{\infty}(\Omega)$ $\quad$ $u \in H^{3.5}(\Omega)$ $\quad$ $u \in H^{1.5}(\Omega)$

Orders of convergence

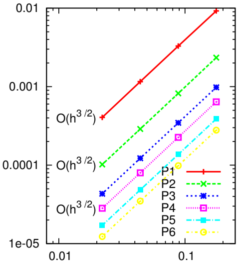
- higher order schemes are efficient
- BUT only if the exact solution is sufficiently regular
- $\|u - u_h\|_{L^2} = O(h^\mu),\ \mu = \min(p+1, s),\ u_h \in S_{hp},\ u \in H^s(\Omega)$



$u \in C^\infty(\Omega)$



$u \in H^{3.5}(\Omega)$



$u \in H^{1.5}(\Omega)$

hp-DGM

Idea of *hp*-methods

- *p*-refinement in subdomains where solution is regular,
- *h*-refinement in subdomains where solution is not regular,

Technical aspects

- marking of elements for refinement (based on error estimates)
- decide if p or h adaptation
- many (more or less heuristic) techniques
  [Demkowicz, Georgoulis, Houston, Melenk, Schwab, Šolín, Süli, ...]

Implementation aspects of *hp*-DGM

- basis of S_h^p: local and discontinuous
  $\Longrightarrow$ NO implementation obstacle

hp-DGM

Idea of *hp*-methods

- *p*-refinement in subdomains where solution is regular,
- *h*-refinement in subdomains where solution is not regular,

Technical aspects

- marking of elements for refinement (based on error estimates)
- decide if p or h adaptation
- many (more or less heuristic) techniques
  [Demkowicz, Georgoulis, Houston, Melenk, Schwab, Šolín, Süli, ...]

Implementation aspects of *hp*-DGM

- basis of S_h^p: local and discontinuous
  $\Longrightarrow$ NO implementation obstacle

hp-DGM

Idea of *hp*-methods

- *p*-refinement in subdomains where solution is regular,
- *h*-refinement in subdomains where solution is not regular,

Technical aspects

- marking of elements for refinement (based on error estimates)
- decide if *p* or *h* adaptation
- many (more or less heuristic) techniques
  [Demkowicz, Georgoulis, Houston, Melenk, Schwab, Šolín, Süli, ...]

Implementation aspects of *hp*-DGM

- basis of S_h^p: local and discontinuous
  $\Longrightarrow$ NO implementation obstacle

hp-DGM

Idea of *hp*-methods

- *p*-refinement in subdomains where solution is regular,
- *h*-refinement in subdomains where solution is not regular,

Technical aspects

- marking of elements for refinement (based on error estimates)
- decide if *p* or *h* adaptation
- many (more or less heuristic) techniques
  [Demkowicz, Georgoulis, Houston, Melenk, Schwab, Šolín, Süli, ...]

Implementation aspects of *hp*-DGM

- basis of S_h^p: local and discontinuous
  $\Longrightarrow$ NO implementation obstacle

hp-DGM

Idea of *hp*-methods

- p-refinement in subdomains where solution is regular,
- h-refinement in subdomains where solution is not regular,

Technical aspects

- marking of elements for refinement (based on error estimates)
- decide if p or h adaptation
- many (more or less heuristic) techniques
  [Demkowicz, Georgoulis, Houston, Melenk, Schwab, Šolín, Süli, ...]

Implementation aspects of *hp*-DGM

- basis of S_h^p: local and discontinuous
  $\Longrightarrow$ NO implementation obstacle

hp-DGM

Idea of *hp*-methods

- *p*-refinement in subdomains where solution is regular,
- *h*-refinement in subdomains where solution is not regular,

Technical aspects

- marking of elements for refinement (based on error estimates)
- decide if p or h adaptation
- many (more or less heuristic) techniques
  [Demkowicz, Georgoulis, Houston, Melenk, Schwab, Šolín, Süli, . . .]

Implementation aspects of *hp*-DGM

- basis of S_h^p: local and discontinuous
  $\Longrightarrow$ NO implementation obstacle

hp-DGM

Idea of *hp*-methods

- p-refinement in subdomains where solution is regular,
- h-refinement in subdomains where solution is not regular,

Technical aspects

- marking of elements for refinement (based on error estimates)
- decide if p or h adaptation
- many (more or less heuristic) techniques
  [Demkowicz, Georgoulis, Houston, Melenk, Schwab, Šolín, Süli, . . .]

Implementation aspects of *hp*-DGM

- basis of S_h^p: local and discontinuous
  $\Longrightarrow$ NO implementation obstacle

hp-DGM

Idea of *hp*-methods

- *p*-refinement in subdomains where solution is regular,
- *h*-refinement in subdomains where solution is not regular,

Technical aspects

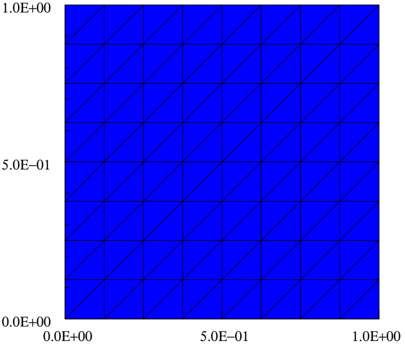
- marking of elements for refinement (based on error estimates)
- decide if *p* or *h* adaptation
- many (more or less heuristic) techniques
  [Demkowicz, Georgoulis, Houston, Melenk, Schwab, Šolín, Süli, ...]

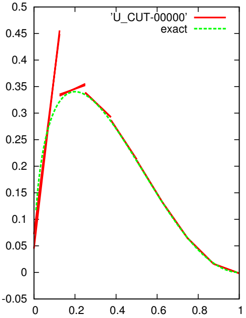
Implementation aspects of *hp*-DGM

- basis of S_h^p: local and discontinuous
  $\Longrightarrow$ NO implementation obstacle

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

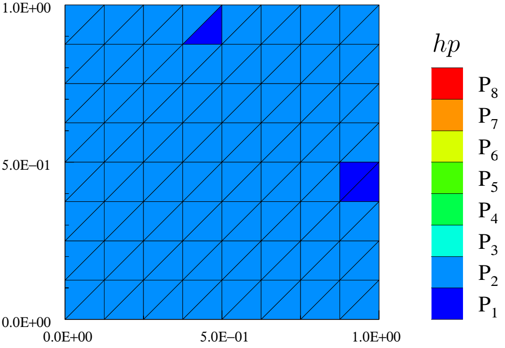


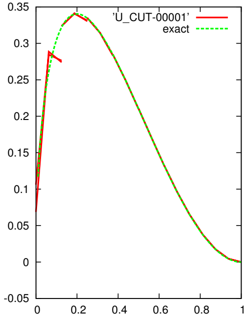


adaptation level = 00

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

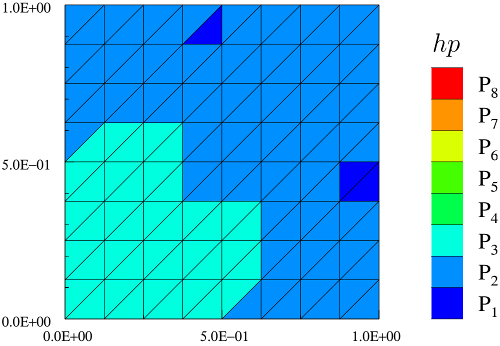


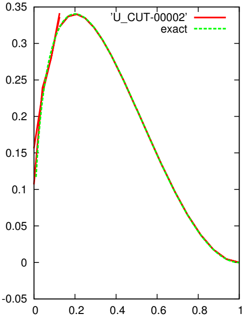


adaptation level = 01

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

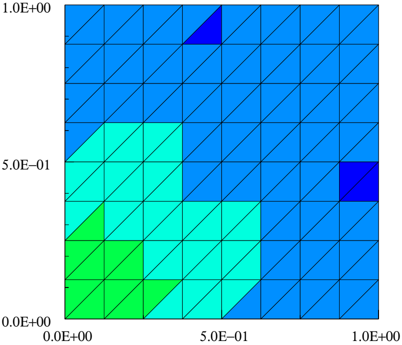


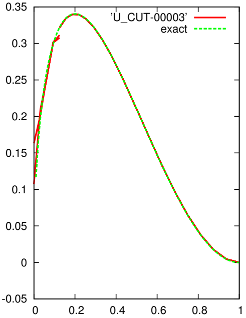


adaptation level = 02

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

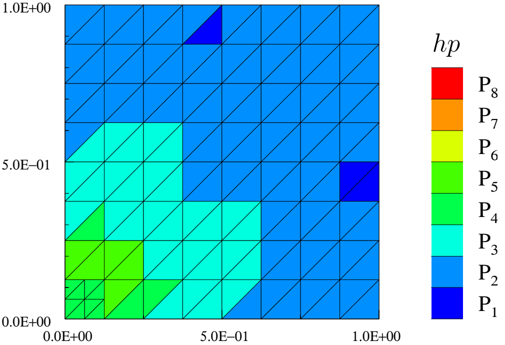


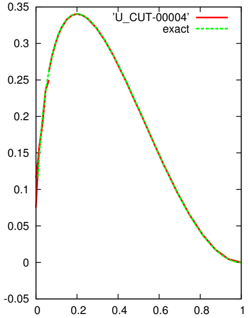


adaptation level = 03

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

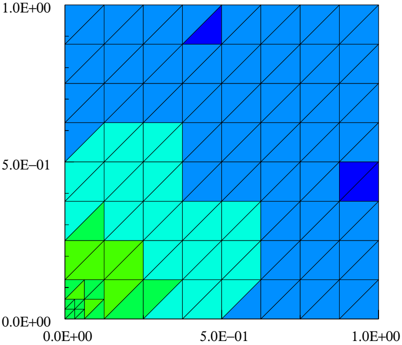


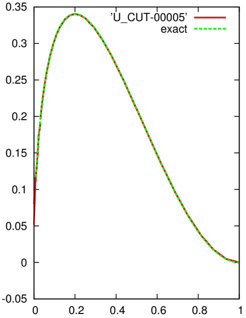


adaptation level $= 04$

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

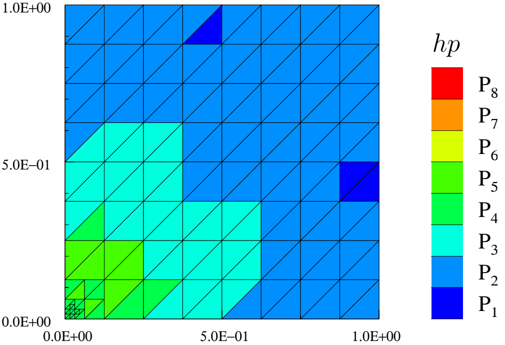


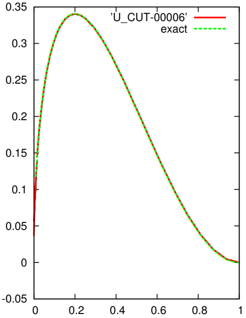


adaptation level = 05

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

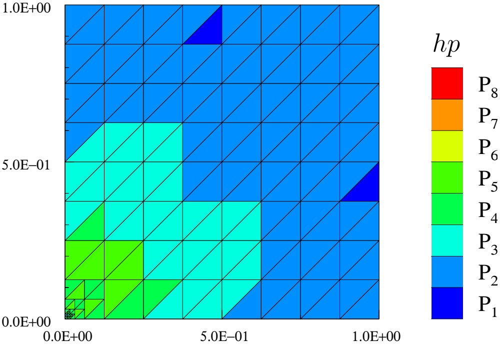


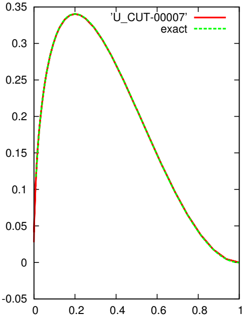


adaptation level = 06

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

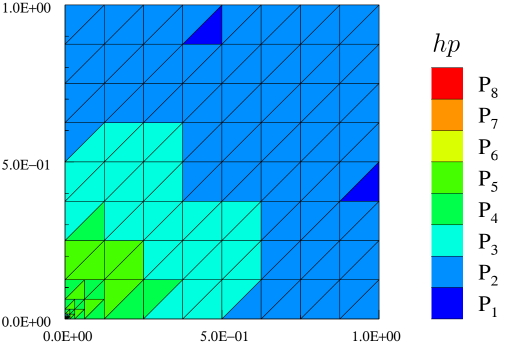


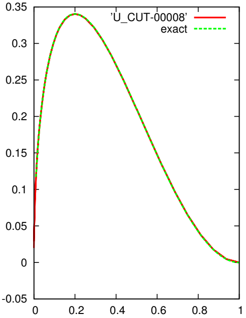


adaptation level = 07

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

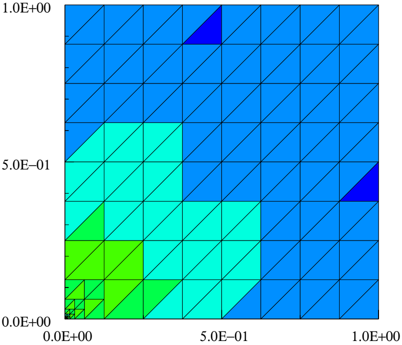


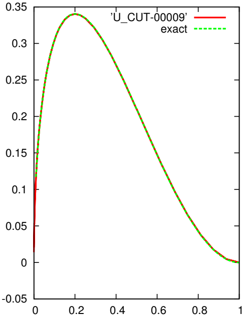


adaptation level = 08

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

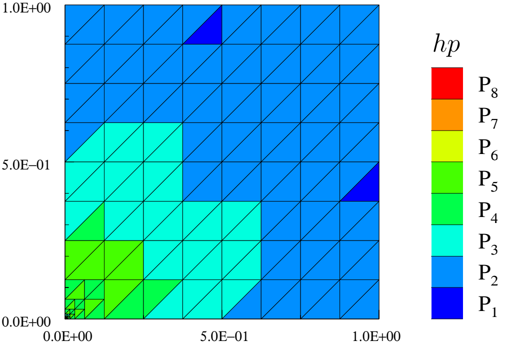


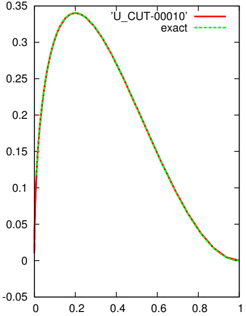


adaptation level = 09

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity

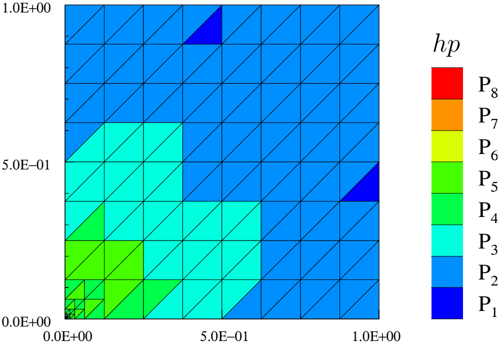


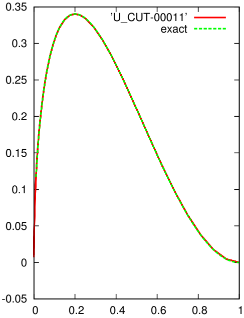


adaptation level = 10

Performance of the *hp*-DGM

nonlinear convection-diffusion equation with a corner singularity





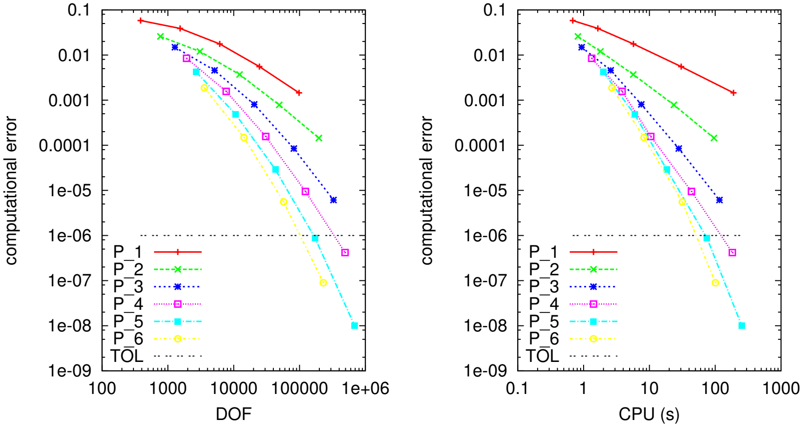
adaptation level $= 11$

Practical demonstration of the efficiency of *hp*-DGM

- scalar linear convection-diffusion equation solved by DGM

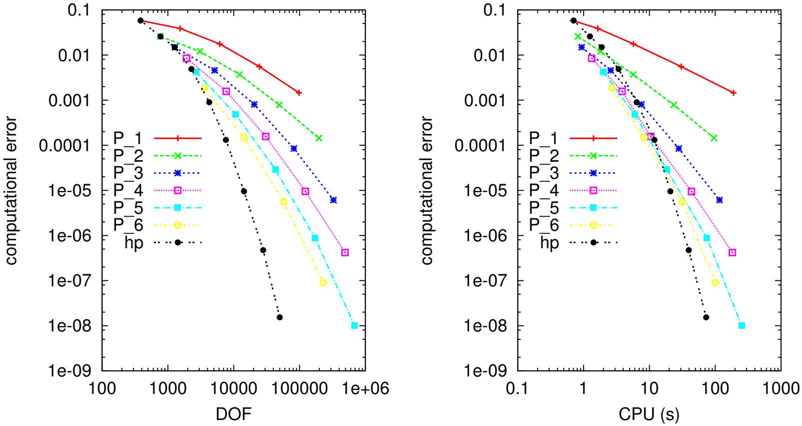
Practical demonstration of the efficiency of *hp*-DGM

- scalar linear convection-diffusion equation solved by DGM



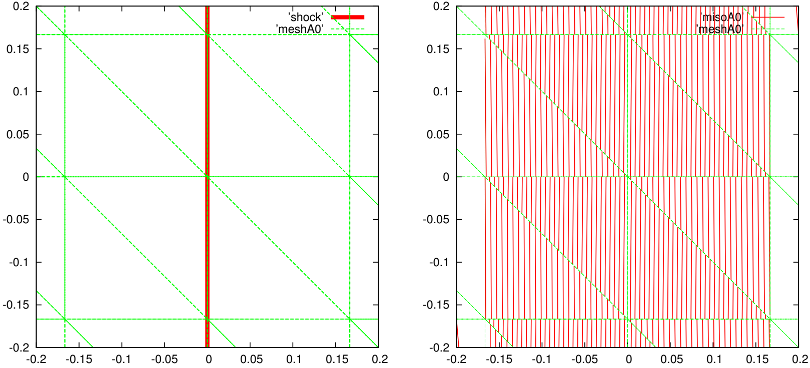
Practical demonstration of the efficiency of *hp*-DGM

- scalar linear convection-diffusion equation solved by DGM



h-refinement for interior and boundary layers problems

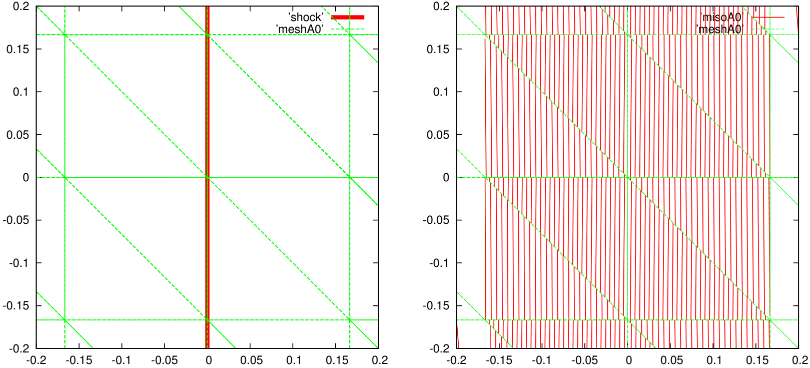
- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient



adaptation level = 0

h-refinement for interior and boundary layers problems

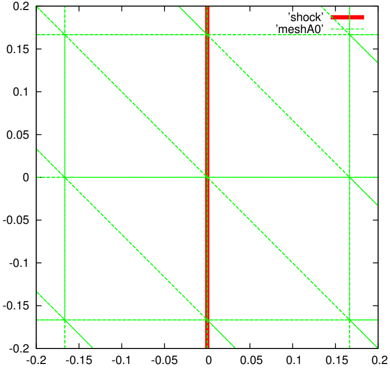
- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

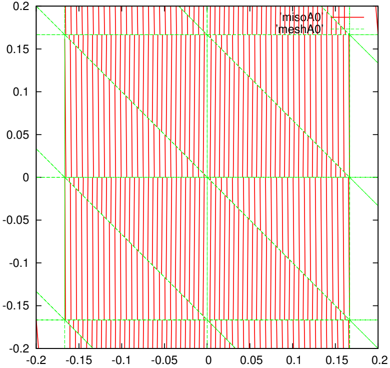


adaptation level $= 0$

h-refinement for interior and boundary layers problems

- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

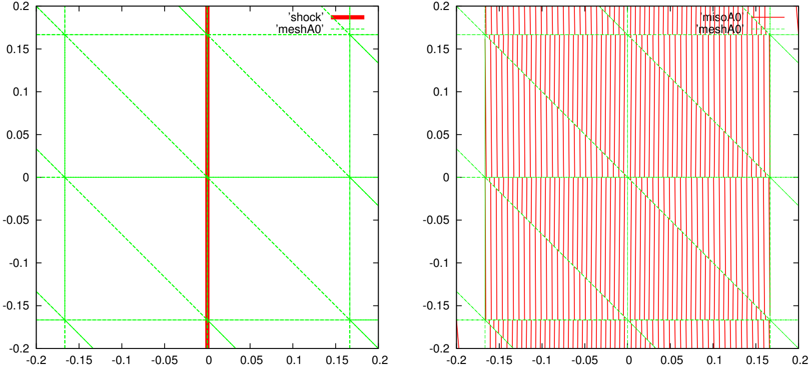




adaptation level $= 0$

h-refinement for interior and boundary layers problems

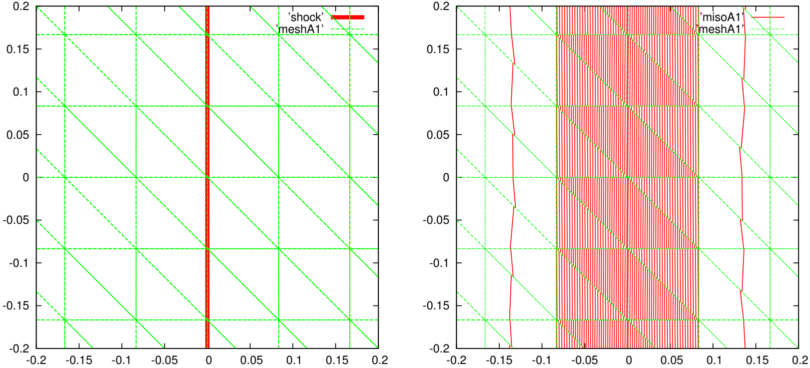
- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient



adaptation level $= 0$

h-refinement for interior and boundary layers problems

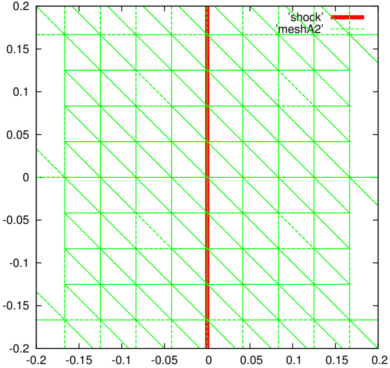
- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

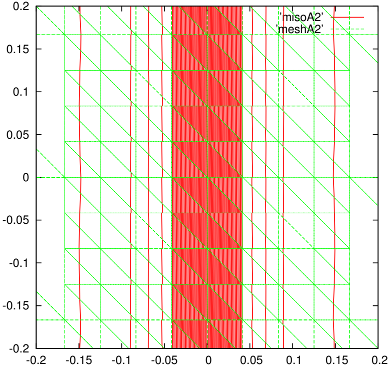


adaptation level $= 1$

h-refinement for interior and boundary layers problems

- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

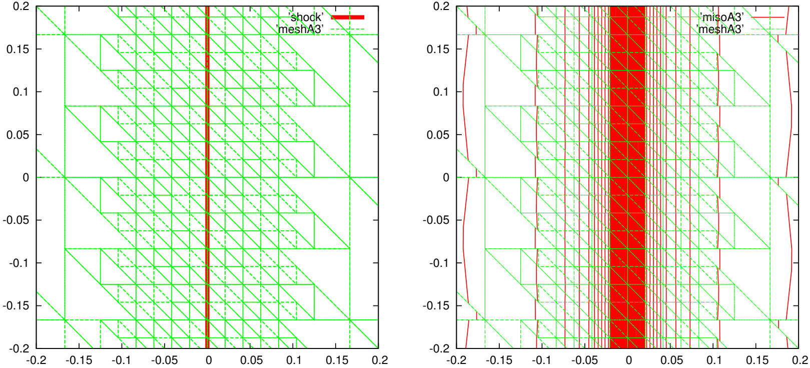




adaptation level = 2

h-refinement for interior and boundary layers problems

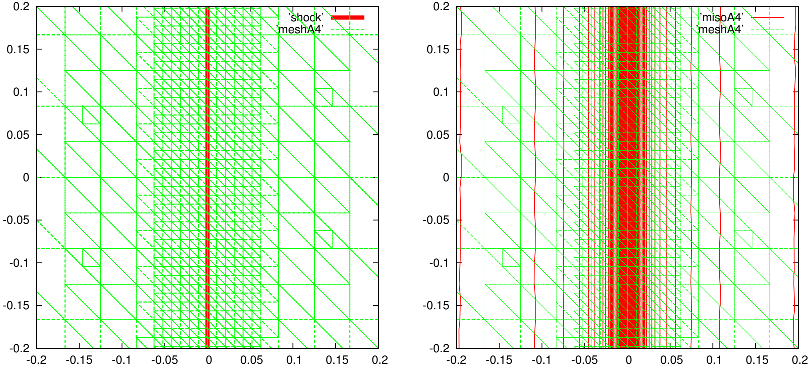
- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient



adaptation level = 3

h-refinement for interior and boundary layers problems

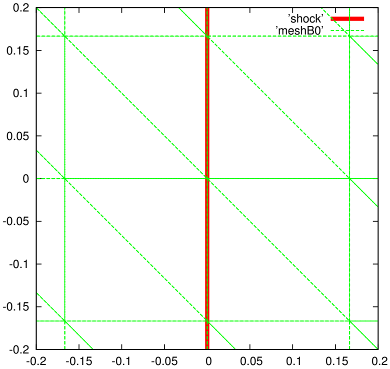
- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

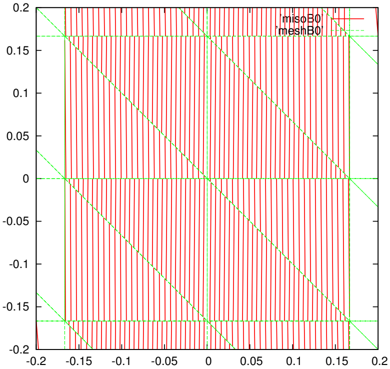


adaptation level = 4

h-refinement for interior and boundary layers problems

- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

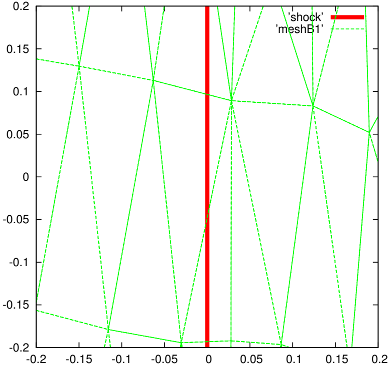


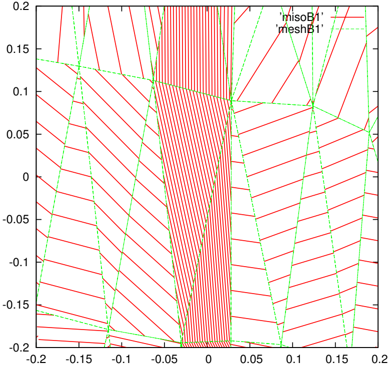


AMA adaptation level = 0

h-refinement for interior and boundary layers problems

- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

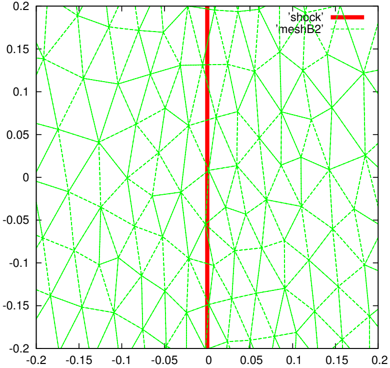


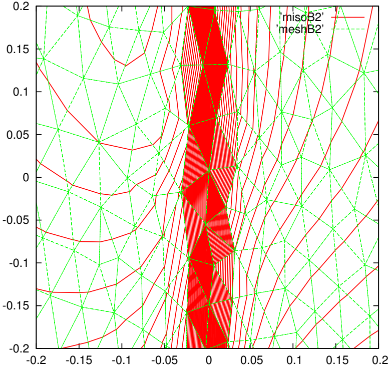


AMA adaptation level = 1

h-refinement for interior and boundary layers problems

- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

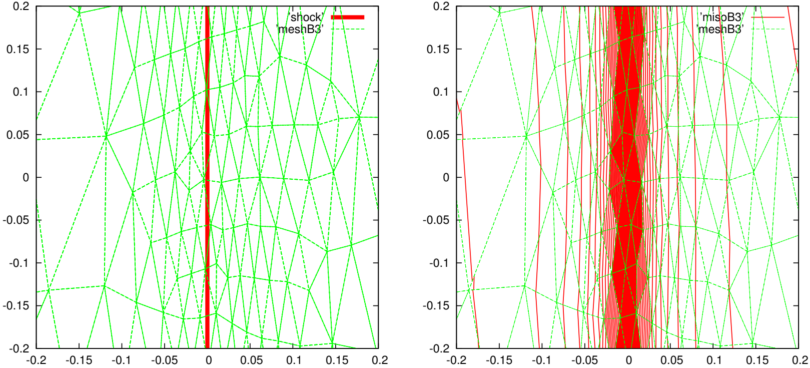




AMA adaptation level = 2

h-refinement for interior and boundary layers problems

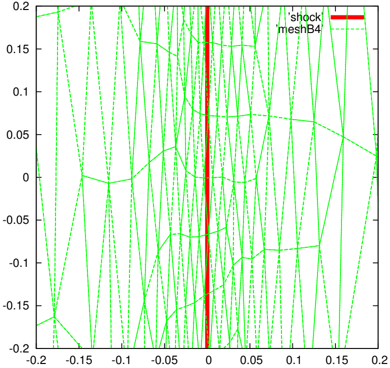
- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

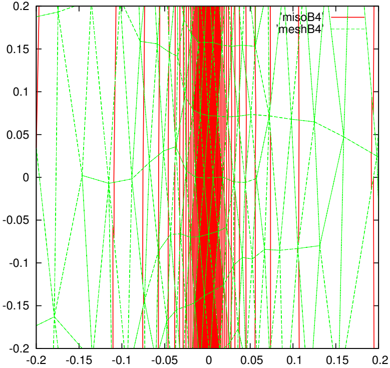


AMA adaptation level = 3

h-refinement for interior and boundary layers problems

- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

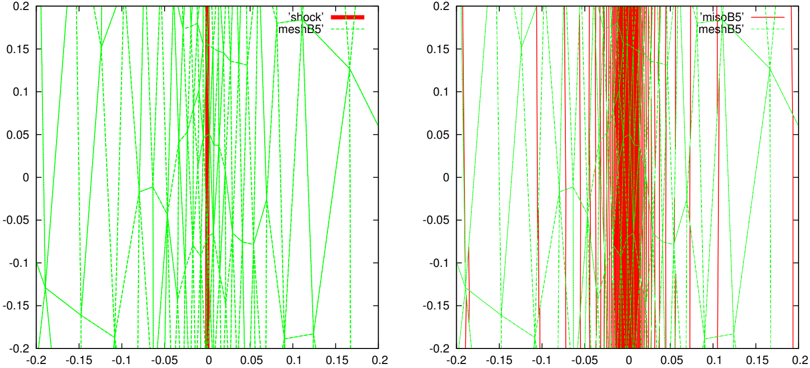




AMA adaptation level = 4

h-refinement for interior and boundary layers problems

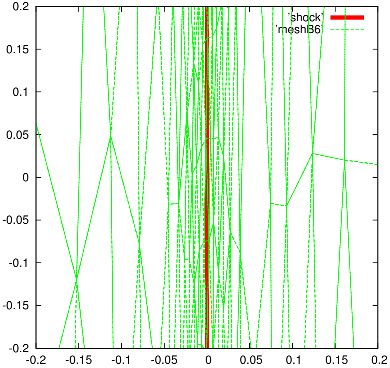
- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

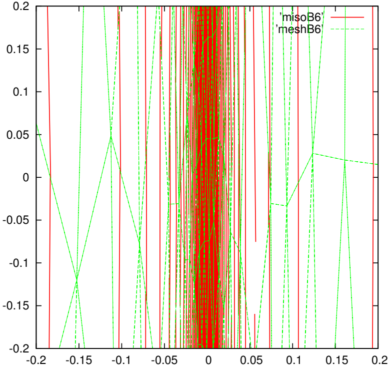


AMA adaptation level = 5

h-refinement for interior and boundary layers problems

- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient

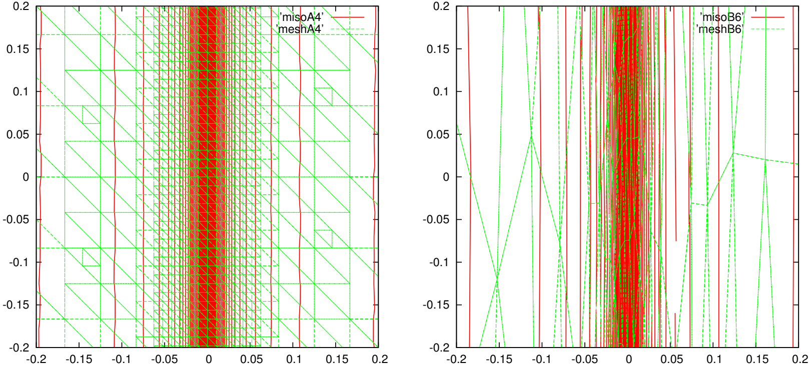




AMA adaptation level = 6

h-refinement for interior and boundary layers problems

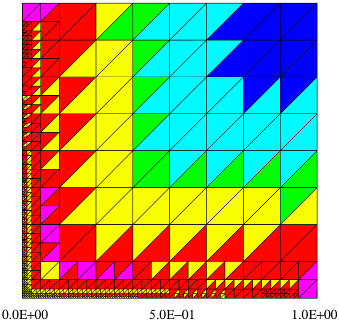
- isotropic *h*-refinement – not efficient
- anisotropic mesh adaptation – more efficient



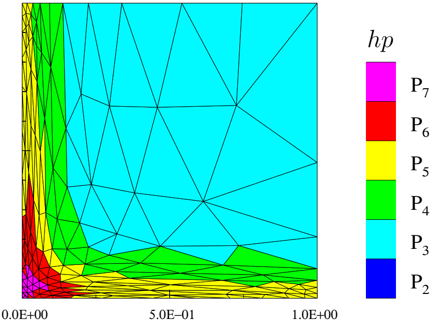
comparison: isotropic (5649 elems) – anisotropic (432 elems)

Anisotropic *hp*-DGM

hp-DGM & anisotropic mesh adaptation



isotropic *hp*-mesh



anisotropic *hp*-mesh

Anisotropic *hp*-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ "optimal" S_{hp}.

Anisotropic *hp*-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ "optimal" S_{hp}.

Anisotropic *hp*-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ "optimal" S_{hp}.

Anisotropic *hp*-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ "optimal" S_{hp}.

Anisotropic *hp*-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ “optimal” S_{hp}.

Anisotropic *hp*-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ "optimal" S_{hp}.

Anisotropic *hp*-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\mathrm{dim} S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ “optimal” S_{hp}.

Anisotropic *hp*-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ "optimal" S_{hp}.

Anisotropic hp-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ “optimal” S_{hp}.

Anisotropic *hp*-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ "optimal" S_{hp}.

Anisotropic *hp*-DGM

Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for x $\Longrightarrow$ "optimal" S_{hp}.

Anisotropic *hp*-DGM

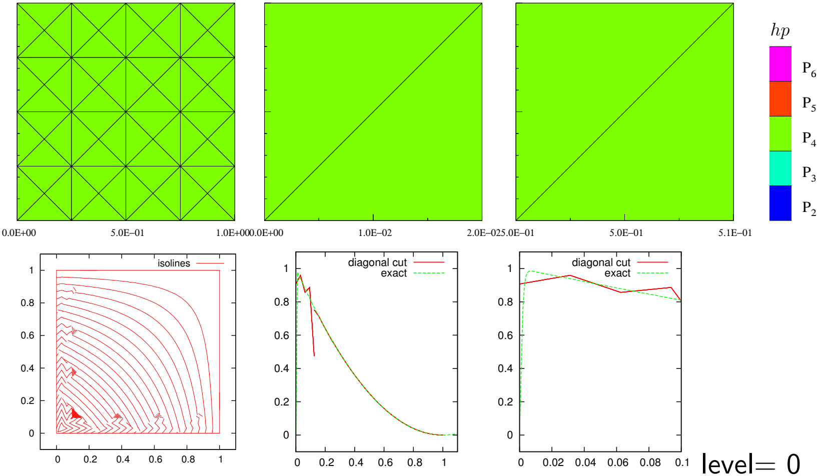
Basic idea [D. APNUM 2014]

- let $u : \Omega \to \mathbb{R}$, $q > 1$ and $\omega > 0$ be given,
- we seek mesh $\mathscr{T}_h$ and the corresponding space S_{hp} such that
  - $\|u - \Pi_{hp} u\|_{L^q(\Omega)} \leq \omega$,
  - $\dim S_{hp}$ is minimal,
- Π_{hp} is the L^2-orthogonal projection into S_{hp}.

Practical realization

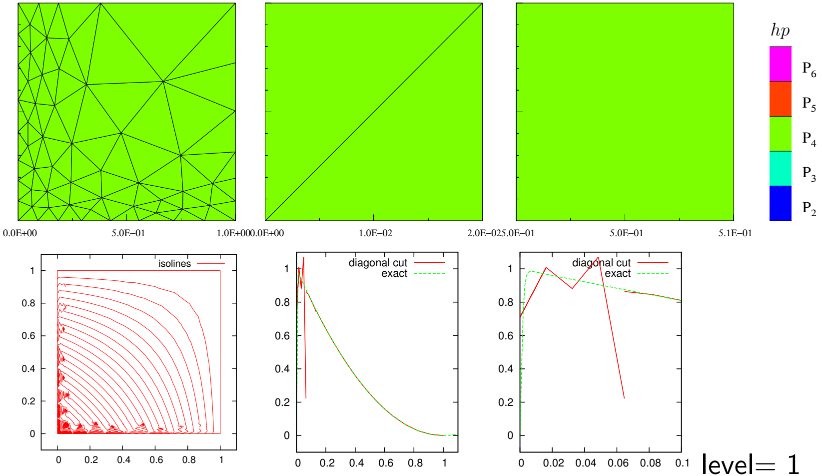
- for each $x \in \Omega$ and $p \in \mathbb{N}$, we seek optimal shape of $K_{x,p}$ minimizing $\|u - \Pi_{hp} u\|_{L^q(K)}$,
- from $\|u - \Pi_{hp} u\|^q_{L^q(K)} \leq \omega^q \frac{|K|}{|\Omega|}$, we set size of $K_{x,p}$
- we set optimal p for $x \implies$ "optimal" S_{hp}.

Boundary layers with $\varepsilon = 10^{-3}$ – performance



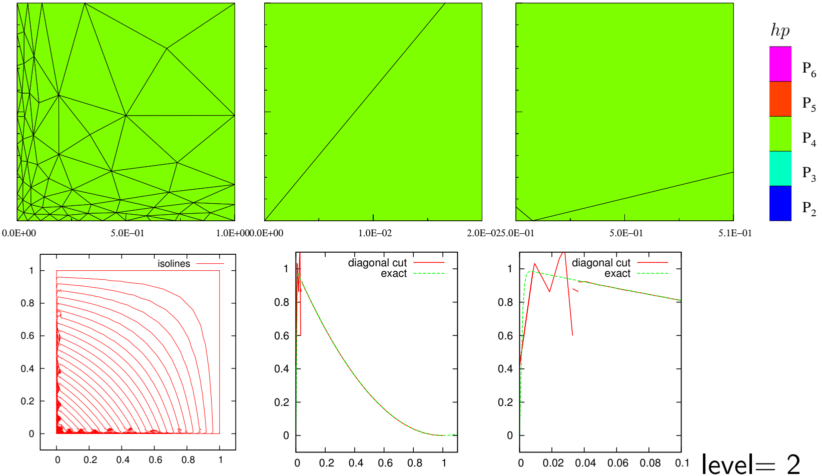
level= 0

Boundary layers with $\varepsilon = 10^{-3}$ – performance



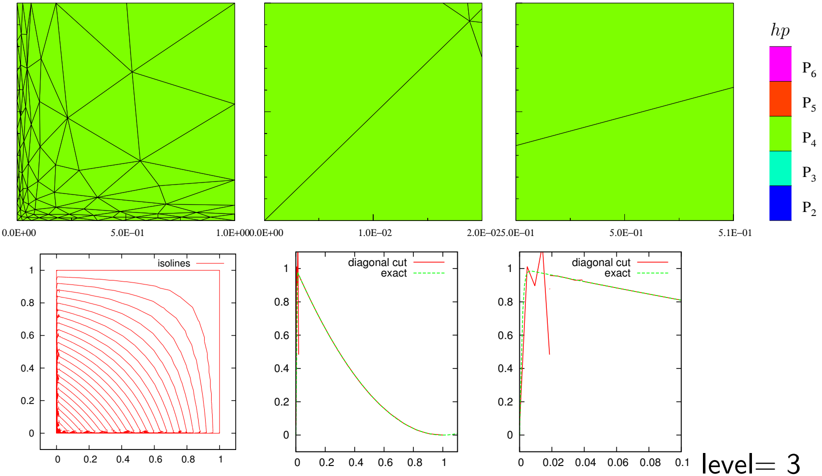
level= 1

Boundary layers with $\varepsilon = 10^{-3}$ – performance



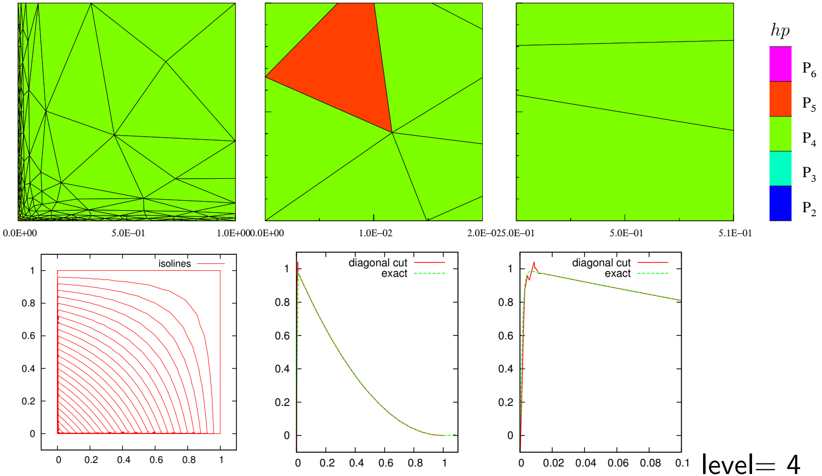
level= 2

Boundary layers with $\varepsilon = 10^{-3}$ – performance



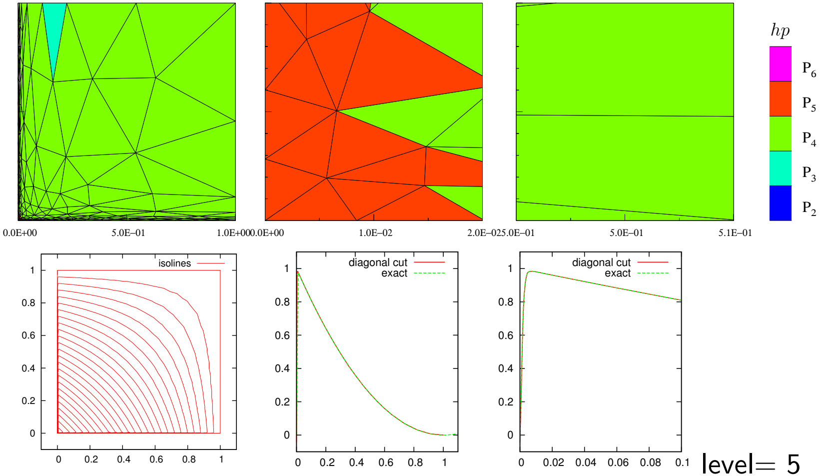
level$= 3$

Boundary layers with $\varepsilon = 10^{-3}$ – performance



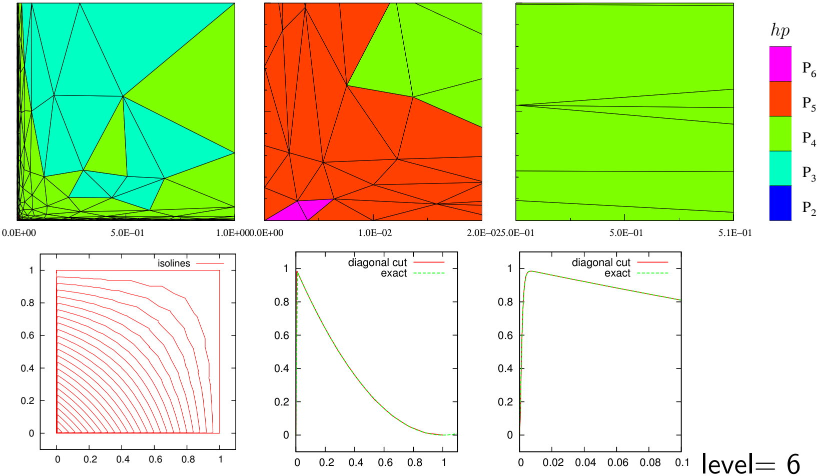
level= 4

Boundary layers with $\varepsilon = 10^{-3}$ – performance



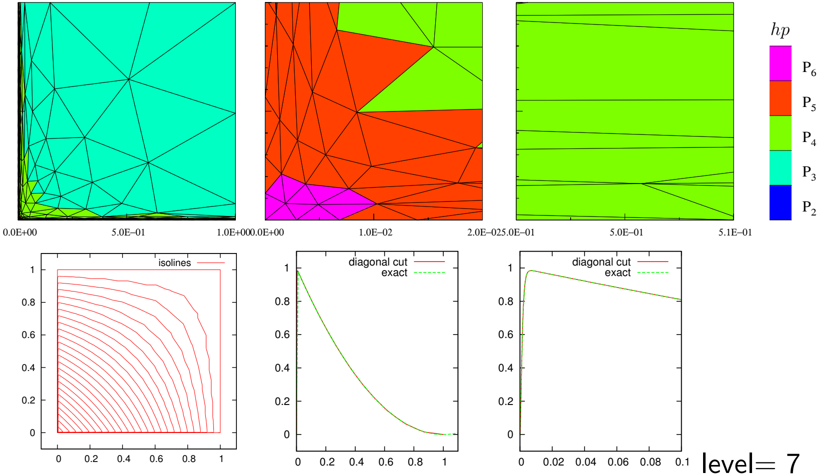
level$= 5$

Boundary layers with $\varepsilon = 10^{-3}$ – performance



level$= 6$

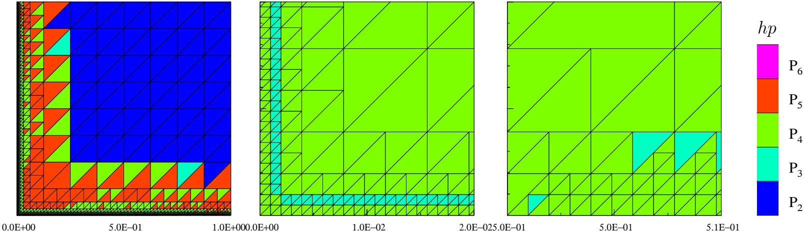
Boundary layers with $\varepsilon = 10^{-3}$ – performance



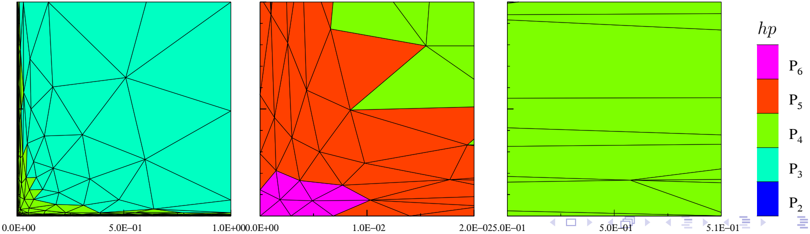
level= 7

Boundary layers with $\varepsilon = 10^{-3}$ – comparison

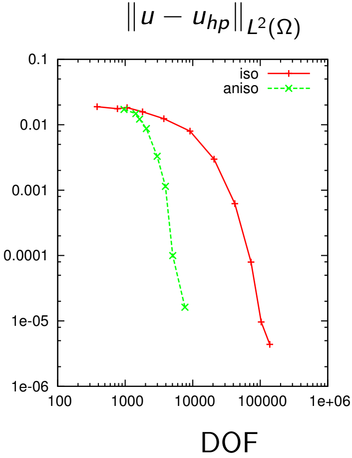
isotropic: total — zoom 50 of origin — zoom 100 of boundary layer

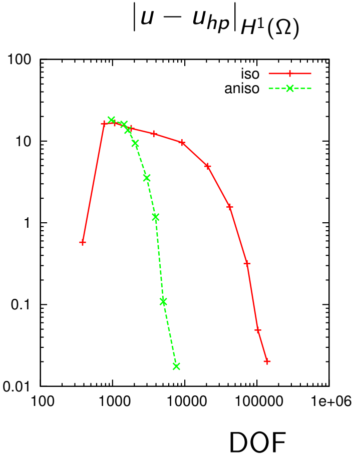


anisotropic



Boundary layers with $\varepsilon = 10^{-3}$ – comparison





Anisotropic *hp*-DGM for compressible flow simulation

Viscous compressible flow around NACA 0012

$M_{\rm inlet} = 0.5$, $\alpha = 3^\circ$, $\mathrm{Re} = 5\,000$,

Rerefence values of aerodynamical coefficients

$c_D = 0.057701, \quad c_L = 0.052737, \quad c_M = -0.022987$

The goal: compute aerodynamical coefficients

with tolerances: 1% for c_D, 5% for c_L, 2% for c_M

Adaptive techniques

- isotropic *hp*-refinement
- P_1 and P_2 anisotropic mesh refinement
- anisotropic *hp*-refinement

Anisotropic *hp*-DGM for compressible flow simulation

Viscous compressible flow around NACA 0012

$M_{\text{inlet}} = 0.5$, $\alpha = 3^\circ$, $\text{Re} = 5\,000$,

Rerefence values of aerodynamical coefficients

$c_D = 0.057701, \quad c_L = 0.052737, \quad c_M = -0.022987$

The goal: compute aerodynamical coefficients

with tolerances: 1% for c_D, 5% for c_L, 2% for c_M

Adaptive techniques

- isotropic *hp*-refinement
- P_1 and P_2 anisotropic mesh refinement
- anisotropic *hp*-refinement

Anisotropic *hp*-DGM for compressible flow simulation

Viscous compressible flow around NACA 0012

$M_{\mathrm{inlet}} = 0.5$, $\alpha = 3^\circ$, $\mathrm{Re} = 5\,000$,

Rerefence values of aerodynamical coefficients

$c_D = 0.057701$, $c_L = 0.052737$, $c_M = -0.022987$

The goal: compute aerodynamical coefficients

with tolerances: 1% for c_D, 5% for c_L, 2% for c_M

Adaptive techniques

- isotropic *hp*-refinement
- P_1 and P_2 anisotropic mesh refinement
- anisotropic *hp*-refinement

Anisotropic *hp*-DGM for compressible flow simulation

Viscous compressible flow around NACA 0012

$M_{\text{inlet}} = 0.5$, $\alpha = 3^\circ$, $\text{Re} = 5\,000$,

Rerefence values of aerodynamical coefficients

$c_D = 0.057701, \quad c_L = 0.052737, \quad c_M = -0.022987$

The goal: compute aerodynamical coefficients

with tolerances: 1% for c_D, 5% for c_L, 2% for c_M

Adaptive techniques

- isotropic *hp*-refinement
- P_1 and P_2 anisotropic mesh refinement
- anisotropic *hp*-refinement

Anisotropic *hp*-DGM for compressible flow simulation

Viscous compressible flow around NACA 0012

$M_{\mathrm{inlet}} = 0.5$, $\alpha = 3^\circ$, $\mathrm{Re} = 5\,000$,

Rerefence values of aerodynamical coefficients

$c_D = 0.057701$, $c_L = 0.052737$, $c_M = -0.022987$

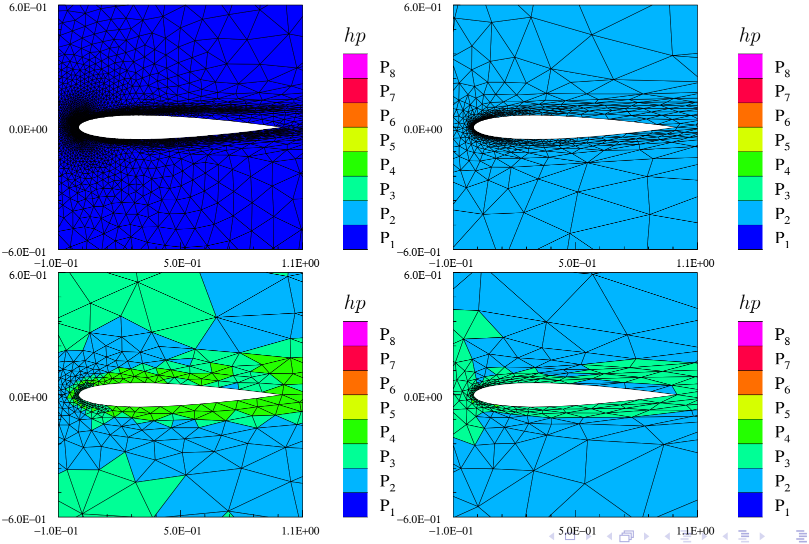
The goal: compute aerodynamical coefficients

with tolerances: 1% for c_D, 5% for c_L, 2% for c_M

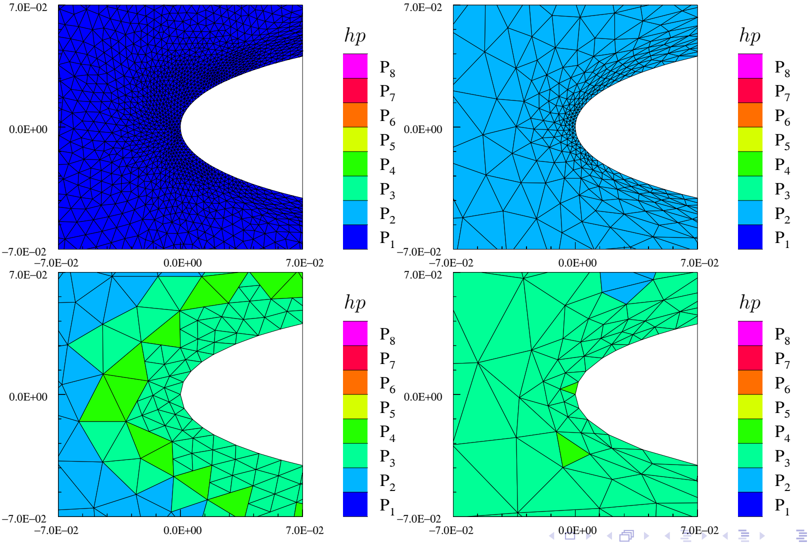
Adaptive techniques

- isotropic *hp*-refinement
- P_1 and P_2 anisotropic mesh refinement
- anisotropic *hp*-refinement

Comparison of adaptations: *hp*-grids



Comparison of adaptations: *hp*-grids



Comparison of DOF and CPU

method	$\#\mathscr{T}_h$	DOF	CPU(s)
AMA P_1	7968	95616	266
AMA P_2	1618	38832	107
iso *hp*	885	31488	91
hp-AMA	595	18488	80

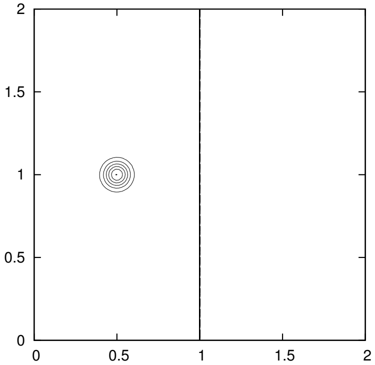
Anisotropic *hp* looks nice but NO guaranty of the error

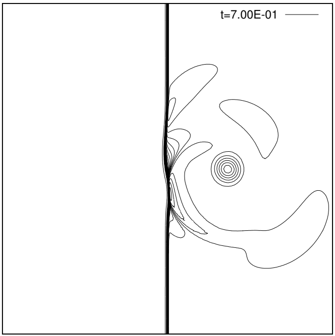
Comparison of DOF and CPU

method	$\#\mathscr{T}_h$	DOF	CPU(s)
AMA P_1	7968	95616	266
AMA P_2	1618	38832	107
iso *hp*	885	31488	91
hp-AMA	595	18488	80

Anisotropic *hp* looks nice but NO guaranty of the error

Viscous shock-vortex interaction - preliminary results





Viscous shock-vortex interaction – adaptive refinement

A posteriori error estimates

State of the art

- many theoretical works for scalar linear problems,
- nonlinear problems: significantly smaller number of results
- most results estimates of $\|\mathbf{w} - \mathbf{w}_h\|$ in a suitable norm
- It is really our goal?

A posteriori error estimates of the quantity of interest

- quantity of interest: a linear functional $J(\mathbf{w})$,
- estimate $J(\mathbf{w}) - J(\mathbf{w}_h)$
- theory [Becker, Rannacher, Suli, Giles, Hartmann, ...],
- for anisotropic *hp* adaptation completly open

A posteriori error estimates

State of the art

- many theoretical works for scalar linear problems,
- nonlinear problems: significantly smaller number of results
- most results estimates of $\|\mathbf{w} - \mathbf{w}_h\|$ in a suitable norm
- It is really our goal?

A posteriori error estimates of the quantity of interest

- quantity of interest: a linear functional $J(\mathbf{w})$,
- estimate $J(\mathbf{w}) - J(\mathbf{w}_h)$
- theory [Becker, Rannacher, Suli, Giles, Hartmann, ...],
- for anisotropic *hp* adaptation completly open

A posteriori error estimates

State of the art

- many theoretical works for scalar linear problems,
- nonlinear problems: significantly smaller number of results
- most results estimates of $\|\mathbf{w} - \mathbf{w}_h\|$ in a suitable norm
- It is really our goal?

A posteriori error estimates of the quantity of interest

- quantity of interest: a linear functional $J(\mathbf{w})$,
- estimate $J(\mathbf{w}) - J(\mathbf{w}_h)$
- theory [Becker, Rannacher, Suli, Giles, Hartmann, . . .],
- for anisotropic *hp* adaptation completly open

A posteriori error estimates

State of the art

- many theoretical works for scalar linear problems,
- nonlinear problems: significantly smaller number of results
- most results estimates of $\|\mathbf{w} - \mathbf{w}_h\|$ in a suitable norm
- It is really our goal?

A posteriori error estimates of the quantity of interest

- quantity of interest: a linear functional $J(\mathbf{w})$,
- estimate $J(\mathbf{w}) - J(\mathbf{w}_h)$
- theory [Becker, Rannacher, Suli, Giles, Hartmann, . . .],
- for anisotropic *hp* adaptation completly open

A posteriori error estimates

State of the art

- many theoretical works for scalar linear problems,
- nonlinear problems: significantly smaller number of results
- most results estimates of $\|\mathbf{w} - \mathbf{w}_h\|$ in a suitable norm
- It is really our goal?

A posteriori error estimates of the quantity of interest

- quantity of interest: a linear functional $J(\mathbf{w})$,
- estimate $J(\mathbf{w}) - J(\mathbf{w}_h)$
- theory [Becker, Rannacher, Suli, Giles, Hartmann, ...],
- for anisotropic *hp* adaptation completly open

A posteriori error estimates

State of the art

- many theoretical works for scalar linear problems,
- nonlinear problems: significantly smaller number of results
- most results estimates of $\|\mathbf{w} - \mathbf{w}_h\|$ in a suitable norm
- It is really our goal?

A posteriori error estimates of the quantity of interest

- quantity of interest: a linear functional $J(\mathbf{w})$,
- estimate $J(\mathbf{w}) - J(\mathbf{w}_h)$
- theory [Becker, Rannacher, Suli, Giles, Hartmann, . . .],
- for anisotropic *hp* adaptation completly open

A posteriori error estimates

State of the art

- many theoretical works for scalar linear problems,
- nonlinear problems: significantly smaller number of results
- most results estimates of $\|\mathbf{w} - \mathbf{w}_h\|$ in a suitable norm
- It is really our goal?

A posteriori error estimates of the quantity of interest

- quantity of interest: a linear functional $J(\mathbf{w})$,
- estimate $J(\mathbf{w}) - J(\mathbf{w}_h)$
- theory [Becker, Rannacher, Suli, Giles, Hartmann, . . .],
- for anisotropic *hp* adaptation completly open

A posteriori error estimates

State of the art

- many theoretical works for scalar linear problems,
- nonlinear problems: significantly smaller number of results
- most results estimates of $\|\mathbf{w} - \mathbf{w}_h\|$ in a suitable norm
- It is really our goal?

A posteriori error estimates of the quantity of interest

- quantity of interest: a linear functional $J(\mathbf{w})$,
- estimate $J(\mathbf{w}) - J(\mathbf{w}_h)$
- theory [Becker, Rannacher, Suli, Giles, Hartmann, . . .],
- for anisotropic *hp* adaptation completly open

A posteriori error estimates

State of the art

- many theoretical works for scalar linear problems,
- nonlinear problems: significantly smaller number of results
- most results estimates of $\|\mathbf{w} - \mathbf{w}_h\|$ in a suitable norm
- It is really our goal?

A posteriori error estimates of the quantity of interest

- quantity of interest: a linear functional $J(\mathbf{w})$,
- estimate $J(\mathbf{w}) - J(\mathbf{w}_h)$
- theory [Becker, Rannacher, Suli, Giles, Hartmann, . . .],
- for anisotropic *hp* adaptation completly open

A posteriori error estimates

State of the art

- many theoretical works for scalar linear problems,
- nonlinear problems: significantly smaller number of results
- most results estimates of $\|\mathbf{w} - \mathbf{w}_h\|$ in a suitable norm
- It is really our goal?

A posteriori error estimates of the quantity of interest

- quantity of interest: a linear functional $J(\mathbf{w})$,
- estimate $J(\mathbf{w}) - J(\mathbf{w}_h)$
- theory [Becker, Rannacher, Suli, Giles, Hartmann, . . .],
- for anisotropic *hp* adaptation completly open

Conclusion

Is DGM perspective for CFD problems?

YES, if we employ all benifits of DGM which allows:

- high polynomial degrees of approximation
- anisotropic *hp*-grids
- *p*-multigrid techniques
- efficient paralelization

Conclusion

Is DGM perspective for CFD problems?

YES, if we employ all benifits of DGM which allows:

- high polynomial degrees of approximation
- anisotropic *hp*-grids
- *p*-multigrid techniques
- efficient paralelization

Conclusion

Is DGM perspective for CFD problems?

YES, if we employ all benifits of DGM which allows:

- high polynomial degrees of approximation
- anisotropic *hp*-grids
- *p*-multigrid techniques
- efficient paralelization

Conclusion

Is DGM perspective for CFD problems?

YES, if we employ all benifits of DGM which allows:

- high polynomial degrees of approximation
- anisotropic *hp*-grids
- *p*-multigrid techniques
- efficient paralelization