High-order isogeometric methods: Curse or blessing?

Matthias Möller[a]

Joint work with R. Tielen[a], J. Liu[a,⋆],
H. Schuttelaars[a], K. Vuik[a], D. Göddeke[b]

[a] Delft University of Technology, DIAM

[b] University of Stuttgart, IANS

INdAM: Geometric Challenges in Isogeometric Analysis

Rome, January 31, 2020

⋆CSC Scholarship

Outline

- Part 1: Solution **accuracy** [J. Liu]
  - interplay of approximation and round-off error
  - towards an a-posteriori *hp*-adaptation strategy
- Part 2: Solver **efficiency** [R. Tielen]
  - p-multigrid method with ILUT smoother
  - discussion of choices and numerical examples
- Conclusion and outlook

Part 1: Solution **accuracy**

Model problem #1

Poisson equation in bounded domain Ω with Lipschitz continuous boundary Γ with $f \in L_2(\Omega)$ and $h \in L_2(\Gamma_N)$:

$$
\begin{aligned}
-\Delta u &= f && \text{in } \Omega \\
u &= g && \text{on } \Gamma_D \\
\partial_n u &= h && \text{on } \Gamma_N
\end{aligned}
$$

If Ω is convex, $g = 0$, and $\Gamma_N = \emptyset$ then [Nečas 1967]

$$u \in H^2(\Omega) \quad \text{and} \quad \|u\|_{2,\Omega} \le c(\Omega)\|f\|_{0,\Omega}$$

Otherwise $u \in H^1_{g,D}(\Omega) := \left\{v \in H^1(\Omega) : v = w + g, w \in H^1_{0,D}(\Omega)\right\}$

A-priori error analysis

Weak form: Find $u \in H^1_{g,D}(\Omega)$ such that

$$(\nabla u, \nabla w) = (f, w) + \langle h, w\rangle_{\Gamma_N} \qquad \forall\, w \in H^1_{0,D}(\Omega)$$

Optimal approximation property of the FEM

$$\inf_{v_h \in V_h^{(p)}} \|u - v_h\|_{0,\Omega} = O(h^{p+1})$$

$$\inf_{v_h \in V_h^{(p)}} \|\nabla_h (u - v_h)\|_{0,\Omega} = O(h^{p})$$

A-priori error analysis

Weak form: Find $u \in H^1_{g,D}(\Omega)$ such that

$$(\nabla u, \nabla w) = (f, w) + \langle h, w \rangle_{\Gamma_N} \qquad \forall w \in H^1_{0,D}(\Omega)$$

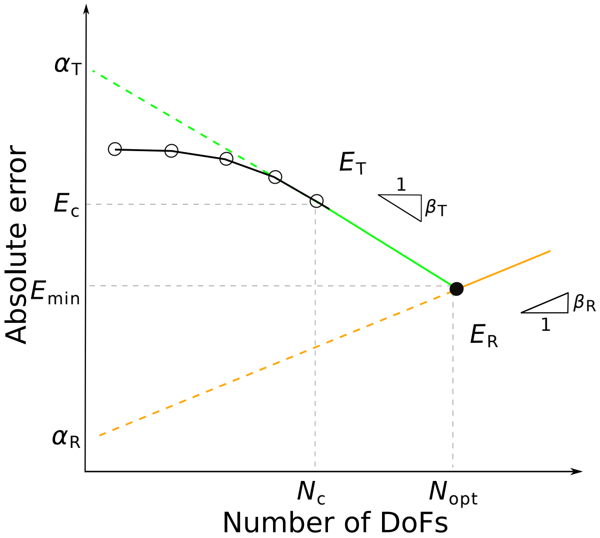
Optimal approximation property of the FEM

$$\inf_{v_h \in V_h^{(p)}} \| u - v_h \|_{0,\Omega} = O(h^{p+1})$$

$$\inf_{v_h \in V_h^{(p)}} \| \nabla_h (u - v_h) \|_{0,\Omega} = O(h^{p})$$

A word of caution: *asymptotic* convergence for $h \to 0$ is combated by round-off errors in practical computations w/ finite-precision arithmetic

Interplay of approximation and round-off errors



Interplay of approximation and round-off errors

Best *computable* solution u_h is obtained for*

$$N_{\text{opt}} = \left(\frac{\alpha_T \beta_T}{\alpha_R \beta_R}\right)^{\frac{1}{\beta_T + \beta_R}}$$

with smallest possible error

$$E_{\min} = \alpha_T \left(\frac{1}{N_{\text{opt}}}\right)^{\beta_T} + \alpha_R \left(\frac{1}{N_{\text{opt}}}\right)^{\beta_R}$$

* J. Liu, MM, H. Schuttelaars, arXiv: 1912.08004

Interplay of approximation and round-off errors

Best *computable* solution u_h is obtained for*

$$N_{\text{opt}} = \left(\frac{\alpha_T \beta_T}{\alpha_R \beta_R} \right)^{\frac{1}{\beta_T + \beta_R}}$$

with smallest possible error

$$E_{\min} = \alpha_T \left(\frac{1}{N_{\text{opt}}} \right)^{\beta_T} + \alpha_R \left(\frac{1}{N_{\text{opt}}} \right)^{\beta_R}$$

- How sensitive are $\alpha_T, \beta_T, \alpha_R, \beta_R$ to problem parameters?

* J. Liu, MM, H. Schuttelaars, arXiv: 1912.08004

Interplay of approximation and round-off errors

Best *computable* solution u_h is obtained for*

$$N_{\text{opt}} = \left(\frac{\alpha_T \beta_T}{\alpha_R \beta_R}\right)^{\frac{1}{\beta_T + \beta_R}}$$

with smallest possible error

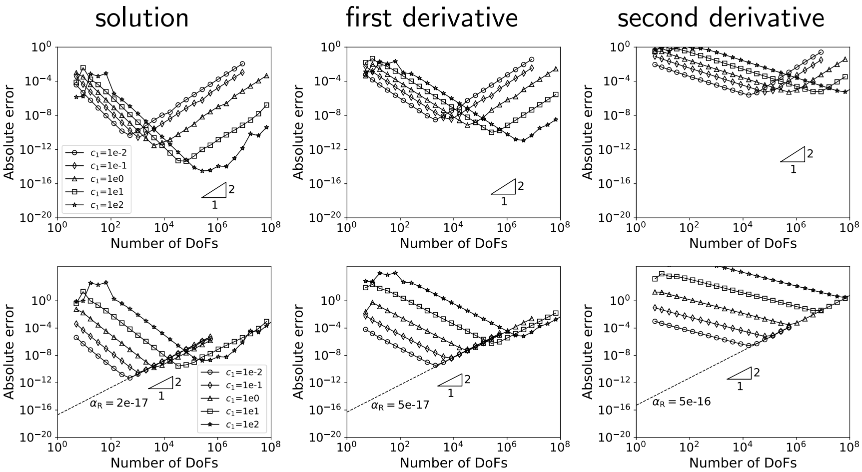
$$E_{\min} = \alpha_T \left(\frac{1}{N_{\text{opt}}}\right)^{\beta_T} + \alpha_R \left(\frac{1}{N_{\text{opt}}}\right)^{\beta_R}$$

- How sensitive are $\alpha_T, \beta_T, \alpha_R, \beta_R$ to problem parameters?
- Can we develop an a-posteriori *hp*–adaptation strategy?

---

* J. Liu, MM, H. Schuttelaars, arXiv: 1912.08004

P_2-FEM in 1d: $u(x) = (2\pi c_1)^{-2} \sin(2\pi c_1 x)$, $f(x) = \sin(2\pi c_1 x)$, $\Omega = (0, 1)$



Top row without scaling; bottom row with scaling $f/\|u\|$ and $u_h/\|u\|$

Analysis of further influence factors

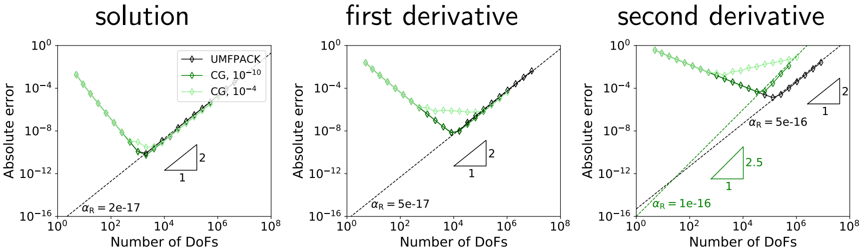
- Type of boundary conditions: *no influence*
- Imposition of Dirichlet boundary conditions: *no influence*
- Computer precision: α_R *changes,* β_R *remains constant*

All results (also using mixed FEM) were produced with deal.II code*

* J. Liu, MM, H. Schuttelaars, arXiv: 1912.08004

Analysis of further influence factors

- Type of boundary conditions: *no influence*
- Imposition of Dirichlet boundary conditions: *no influence*
- Computer precision: α_R *changes,* β_R *remains constant*
- Solution strategy: *moderate influence*



All results (also using mixed FEM) were produced with deal.II code*

*J. Liu, MM, H. Schuttelaars, arXiv: 1912.08004

A-posteriori *hp*-adaptation strategy

Input: initial geometry with mesh width h and approximation order p, tolerances for $E_{\min}$ and maximum mesh refinement steps

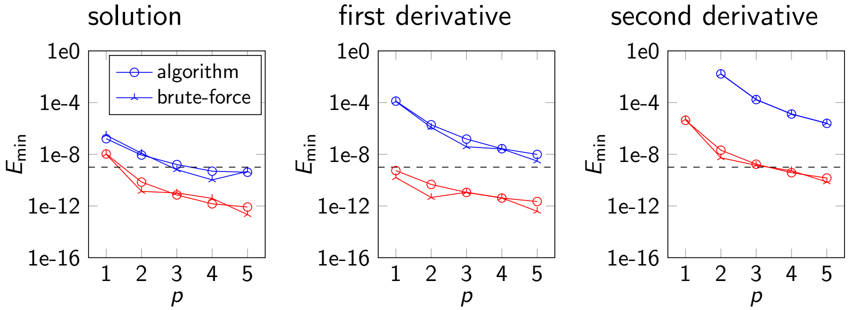
1. **Normalization**: compute u_h on coarse mesh and scale $f/\|u_h\|$
2. **Approximation error prediction**: compute u_h, $u_{h/2}$, ... on coarse meshes until asymptotic convergence rate is observed $\rightarrow \alpha_T, \beta_T$
3. **Round-off error prediction**: use lookup table from previous simulations *or* use manufactured solution that can be resolved exactly by P_p-FEM (possibly using lower precision) $\rightarrow \alpha_R, \beta_R$
4. **Effective error prediction**: compute N_{opt} and $E_{\min}$

Output: N_{opt} and $E_{\min}$. If the estimated error satisfies the required tolerance compute u_{opt} otherwise repeat procedure with $p := p + 1$ or switch to mixed FEM formulation

Model problem #2

Helmholtz equation:

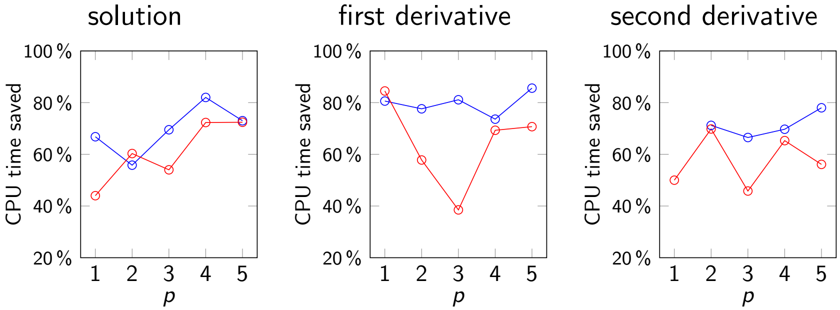
$$
\begin{aligned}
((0.01+x)(1.01-x)u_x)_x - (0.01i)u(x) &= 1.0 \quad \text{in } (0,1) \\
u(0) &= 0 \\
u_x(1) &= 0
\end{aligned}
$$



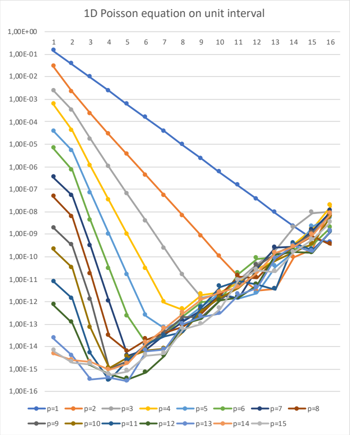
Model problem #2

Helmholtz equation:

$$((0.01+x)(1.01-x)u_x)_x - (0.01i)u(x) = 1.0 \quad \text{in } (0,1)$$
$$u(0) = 0$$
$$u_x(1) = 0$$



Is this of practical relevance?

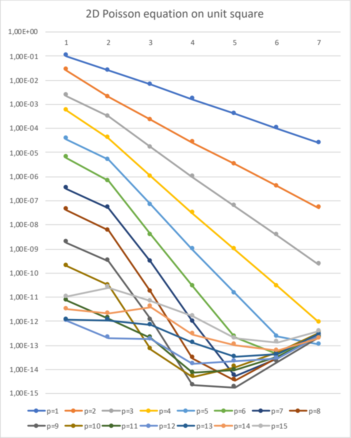


S_p^{p-1}-IGA solutions of model problem #1 with $\Omega = (0, 1)$

Yes …

- since high-order methods can improve the 'effective' accuracy of solutions by orders of magnitudes
- since *h*-refinement is only effective in a small range of refinements for (very) high-order methods and should therefore be used with care

Is this of practical relevance?



S_p^{p-1}-IGA solutions of model problem $\#1$ with $\Omega = (0,1)^2$

Yes …

- since high-order methods can improve the 'effective' accuracy of solutions by orders of magnitudes
- since *h*-refinement is only effective in a small range of refinements for (very) high-order methods and should therefore be used with care
- since the same phenomenon is observed already for moderately refined meshes in 2d (and 3d)

Part 2: Solver **efficiency**

Efficient solvers for IGA discretizations

h-**multigrid methods** enhanced with

- boundary corrected mass-Richardson smoother [Hofreither 2017]
- hybrid smoother [Sogn 2018]
- multiplicative Schwarz smoother [de la Riva 2018]
- ...

Preconditioners based on

- Schwarz methods [Beirão da Veiga 2012]
- Sylvester equation [Sangalli 2016]
- BPX for (T)HB [Bracco et al. 2019]
- ...

Basics of multigrid methods [Strang 2006]

Repeat until converged $\mathbf{u}_{fine}$ is reached

1. **Iterate** on $\mathbf{A}_{fine}\mathbf{u}_{fine} = \mathbf{f}_{fine}$ to reach $\tilde{\mathbf{u}}_{fine}$
2. **Restrict** the residual $\mathbf{r}_{fine} := \mathbf{f}_{fine} - \mathbf{A}_{fine}\tilde{\mathbf{u}}_{fine}$ to the coarse level by applying the restriction operator, i.e. $\mathbf{r}_{coarse} = \mathbf{I}_{fine}^{coarse}\mathbf{r}_{fine}$
3. **Solve** for the coarse level correction $\mathbf{A}_{coarse}\mathbf{E}_{coarse} = \mathbf{r}_{coarse}$
4. **Prolongate** $\mathbf{E}_{coarse}$ back to the fine level by $\mathbf{E}_{fine} = \mathbf{I}_{coarse}^{fine}\mathbf{E}_{coarse}$
5. **Add** the correction, i.e. $\hat{\mathbf{u}}_{fine} := \tilde{\mathbf{u}}_{fine} + \mathbf{E}_{fine}$
6. **Iterate** on $\mathbf{A}_{fine}\hat{\mathbf{u}}_{fine} = \mathbf{f}_{fine}$ to reach $\mathbf{u}_{fine}$

Step 3 calls the multigrid procedure recursively until a coarse level is reached, where the error equation can be solved 'exactly'.

Motivation for using p-multigrid methods

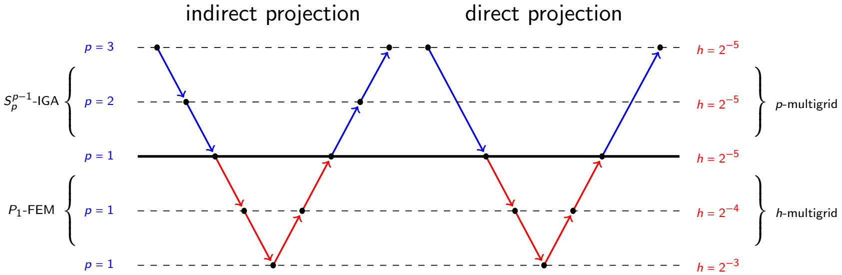
The linear system $\mathbf{A}_{h,p}\mathbf{u}_{h,p} = \mathbf{f}_{h,p}$

- becomes more difficult to solve for increasing p
- reduces to C^0-FEM for $p = 1$ (where h-multigrid works fine)

In contrast to h-multigrid methods

- the #DoFs does not reduce significantly on coarser p-levels
- the stencil reduces significantly on coarse p-levels
- the spaces are not nested, i.e. $(S_{h,p}^{p-1} \not\supset S_{h,p-1}^{p-2} \not\supset \dots)$

V-cycle p-multigrid variants



- ILUT or GS smoothing is applied at each level (•)
- LU decomposition is applied as direct coarse level solver

Prolongation and restriction

Prolongation in h

$\mathcal{I}_{2h,1}^{h,1}$ is linear interpolation

Restriction in h

$$\mathcal{I}_{h,1}^{2h,1} = \frac{1}{2}\left(\mathcal{I}_{2h,1}^{h,1}\right)^{\top}$$

Prolongation in p

$$\mathcal{I}_{h,p-1}^{h,p} := (\mathbf{M}_p^p)^{-1}\mathbf{M}_{p-1}^p$$

Restriction in p

$$\mathcal{I}_{h,p}^{h,p-1} := (\mathbf{M}_{p-1}^{p-1})^{-1}\mathbf{M}_p^{p-1}$$

Let ϕ_i^q denote the i^{th} basis function from $S_{h,q}^{q-1}$. Then define

$$(\mathbf{M}_q^r)_{(i,j)} := \int_{\hat{\Omega}_h} \phi_i^q(\boldsymbol{\xi})\ \phi_j^r(\boldsymbol{\xi})\ c(\boldsymbol{\xi})\ \mathrm{d}\hat{\Omega}$$

Replace $\mathbf{M}_q^q$ by its row-sum lumped counterpart ($\rightarrow$ diagonal matrix)

ILUT smoother [Saad 1994]

Setup: Incomplete LU factorization of $\mathbf{A}_{h,p} \approx \mathbf{L}_{h,p}\mathbf{U}_{h,p}$ thereby

1. dropping all elements lower than tolerance $\tau = 10^{-13}$
2. keeping only the N (= average number of non-zero entries in each row of $\mathbf{A}_{h,p}$) largest elements in each row

Application: perform $s = 1, \ldots, \nu$ smoothing steps

$$\begin{aligned} \mathbf{e}_{h,p}^{(s)} &= (\mathbf{L}_{h,p}\mathbf{U}_{h,p})^{-1}(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}_{h,p}^{(s)}) \\ \mathbf{u}_{h,p}^{(s+1)} &= \mathbf{u}_{h,p}^{(s)} + \mathbf{e}_{h,p}^{(s)} \end{aligned}$$

Model problem #1, revisited

Obtaining coarse level operators

- Galerkin projection $\mathbf{A}^G_{h,p-1} = \mathcal{I}^{h,p-1}_{h,p}\ \mathbf{A}_{h,p}\ \mathcal{I}^{h,p}_{h,p-1}$
- re-discretization of $\mathbf{A}_{h,p}$ on each level

Model problem #1, revisited

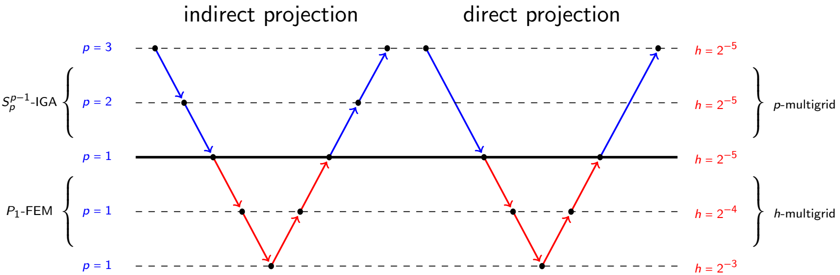
Obtaining coarse level operators

- Galerkin projection $\mathbf{A}^G_{h,p-1} = \mathcal{I}^{h,p-1}_{h,p} \; \mathbf{A}_{h,p} \; \mathcal{I}^{h,p}_{h,p-1}$
- re-discretization of $\mathbf{A}_{h,p}$ on each level

Poisson equation on quarter annulus with radii 1 and 2, $g = 0$, $\Gamma_N = \emptyset$, f such that $u(x,y) = -(x^2+y^2-1)(x^2+y^2-4)xy^2$

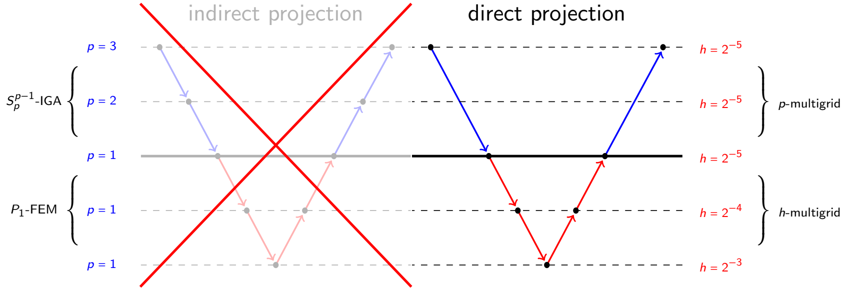
$p=2$	$\kappa(\mathbf{A}^G_{h,1})$	$\kappa(\mathbf{A}^{RD}_{h,1})$	$p=3$	$\kappa(\mathbf{A}^G_{h,2})$	$\kappa(\mathbf{A}^{RD}_{h,2})$
$h=2^{-4}$	$6.00 \cdot 10^7$	$9.78 \cdot 10^2$	$h=2^{-4}$	$7.00 \cdot 10^9$	$1.56 \cdot 10^3$
$h=2^{-5}$	$4.79 \cdot 10^9$	$4.19 \cdot 10^3$	$h=2^{-5}$	$6.15 \cdot 10^{10}$	$6.71 \cdot 10^3$
$h=2^{-6}$	$2.94 \cdot 10^{10}$	$1.76 \cdot 10^4$	$h=2^{-6}$	$4.99 \cdot 10^{11}$	$2.84 \cdot 10^4$
$h=2^{-7}$	$5.48 \cdot 10^{10}$	$7.28 \cdot 10^4$	$h=2^{-7}$	$7.58 \cdot 10^{12}$	$1.18 \cdot 10^5$

V-cycle p-multigrid variants, revisited



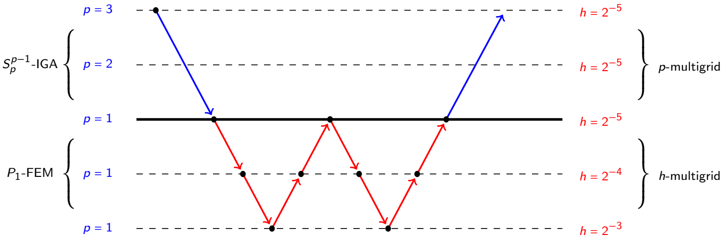
- Setup: Assembly of $\mathbf{A}_{h,p}$, $\mathcal{I}_{h,p}^{h,p-1}$, $\mathcal{I}_{h,p-1}^{h,p}$ each $O(N_{\text{dof}}p^{3d})$ flops
  ILUT factorization of $\mathbf{A}_{h,p}$ $O(N_{\text{dof}}p^{2d})$ flops
  Gauss-Seidel 'setup' $O(N_{\text{dof}})$ flops
- V-cycle: Application of smoother, rest/prol each $O(N_{\text{dof}}p^{d})$ flops

V-cycle p-multigrid variants, revisited



- Setup: Assembly of $\mathbf{A}_{h,p}$, $\mathcal{I}_{h,p}^{h,p-1}$, $\mathcal{I}_{h,p-1}^{h,p}$ each $O(N_{\text{dof}}p^{3d})$ flops
  ILUT factorization of $\mathbf{A}_{h,p}$ $O(N_{\text{dof}}p^{2d})$ flops
  Gauss-Seidel 'setup' $O(N_{\text{dof}})$ flops
- V-cycle: Application of smoother, rest/prol each $O(N_{\text{dof}}p^{d})$ flops
- Numerical tests show same V-cycle counts for both variants

The final V-cycle p-multigrid variant



- ILUT ($p > 1$) / GS smoothing ($p = 1$) is applied at each level (•)
- LU decomposition is applied as direct coarse level solver

Model problem #1: V-cycle counts

V-cycle p-multigrid as a solver

	$p = 2$		$p = 3$		$p = 4$		$p = 5$	
	ILUT*	GS	ILUT*	GS	ILUT*	GS	ILUT*	GS
$h = 2^{-6}$	4	30	3	62	3	176	3	491
$h = 2^{-7}$	4	29	3	61	3	172	3	499
$h = 2^{-8}$	5	30	3	60	3	163	3	473
$h = 2^{-9}$	5	32	3	61	3	163	3	452

V-cycle h-multigrid shows similar convergence behavior

*ILUT ($p > 1$), GS ($p = 1$)

Model problem #1: V-cycle counts

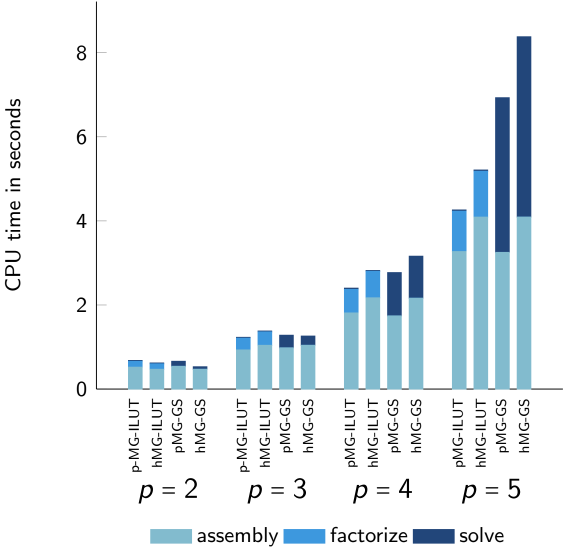
V-cycle p-multigrid as preconditioner in BiCGStab

	$p = 2$		$p = 3$		$p = 4$		$p = 5$	
	ILUT*	GS	ILUT*	GS	ILUT*	GS	ILUT*	GS
$h = 2^{-6}$	2	13	2	18	2	41	2	78
$h = 2^{-7}$	2	12	2	20	2	41	2	92
$h = 2^{-8}$	3	13	2	19	2	43	2	95
$h = 2^{-9}$	3	13	2	21	2	41	2	95

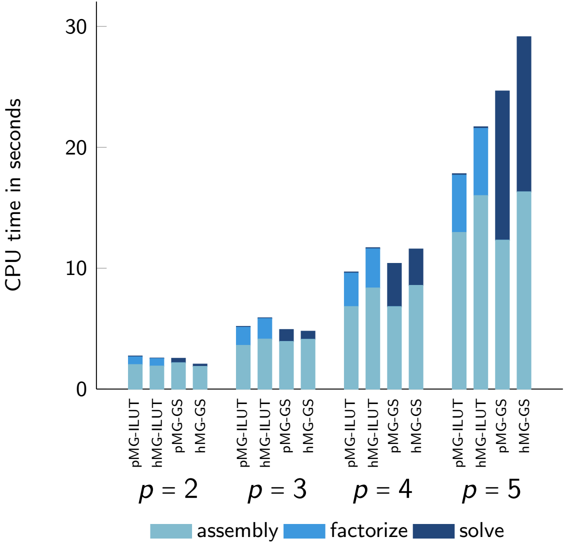
V-cycle h-multigrid shows similar convergence behavior

*ILUT ($p > 1$), GS ($p = 1$)

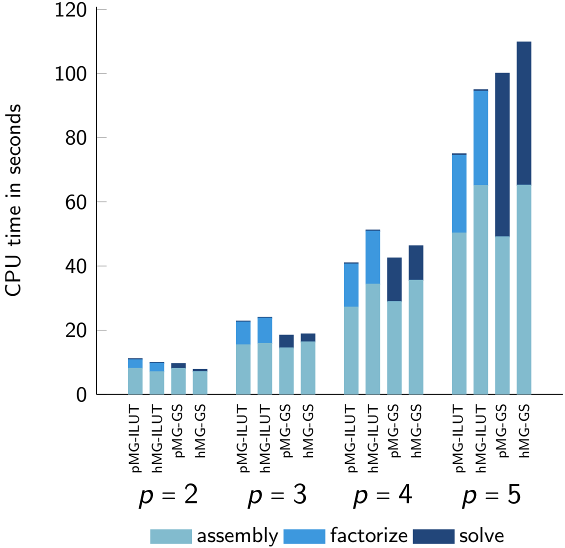
Model problem #1: CPU times for $h = 2^{-6}$



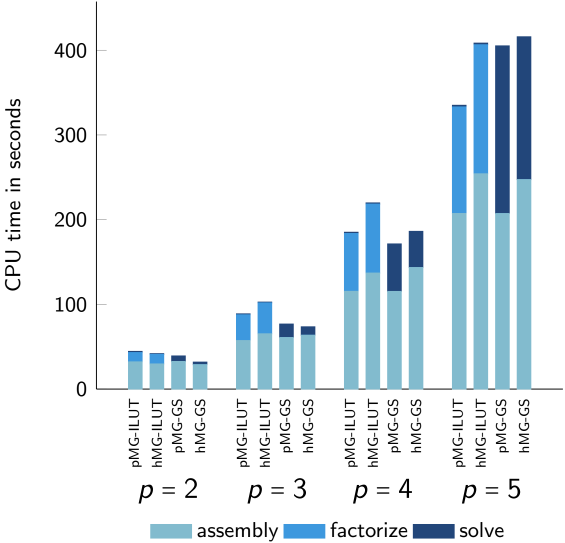
Model problem #1: CPU times for $h = 2^{-7}$



Model problem #1: CPU times for $h = 2^{-8}$



Model problem #1: CPU times for $h = 2^{-9}$



Model problem #3

Convection-diffusion-reaction equation in $\Omega = (0,1)^2$

$$
\begin{aligned}
-\nabla \cdot \left(\begin{bmatrix} 1.2 & -0.7 \\ -0.4 & 0.9 \end{bmatrix} \nabla u \right) + \begin{bmatrix} 0.4 \\ -0.2 \end{bmatrix} \cdot \nabla u + 0.3u = f \quad & \text{in } \Omega \\
u = 0 \quad & \text{on } \Gamma
\end{aligned}
$$

with f such that $u(x,y) = \sin(\pi x)\sin(\pi y)$

Model problem #3: V-cycle counts

V-cycle p-multigrid as a solver

	$p = 2$		$p = 3$		$p = 4$		$p = 5$	
	ILUT	GS	ILUT	GS	ILUT	GS	ILUT	GS
$h = 2^{-6}$	5	–	3	–	3	–	4	–
$h = 2^{-7}$	5	–	3	–	4	–	4	–
$h = 2^{-8}$	5	–	3	–	3	–	4	–
$h = 2^{-9}$	5	–	4	–	3	–	4	–

V-cycle h-multigrid shows similar convergence behavior

Model problem #3: V-cycle counts

V-cycle p-multigrid as preconditioner in BiCGStab

	$p = 2$		$p = 3$		$p = 4$		$p = 5$	
	ILUT	GS	ILUT	GS	ILUT	GS	ILUT	GS
$h = 2^{-6}$	2	7	2	13	2	29	2	65
$h = 2^{-7}$	2	8	2	13	2	29	2	70
$h = 2^{-8}$	2	7	2	12	2	29	2	64
$h = 2^{-9}$	2	7	2	14	2	28	2	72

V-cycle h-multigrid shows similar convergence behavior

Conclusion and outlook

1. **a-posteriori** *hp*-**adaptation strategy** to find (h, p) pair that ensures *computable* approximations with prescribed accuracy

2. *p*-**multigrid method with ILUT smoother** as efficient solver

Conclusion and outlook

1. **a-posteriori** *hp*-**adaptation strategy** to find (h, p) pair that ensures *computable* approximations with prescribed accuracy
   - integration as fully automated procedure in simulation code
   - further analysis of influence factors, i.e. iterative solvers
   - use of number formats that are less sensitive to round-off errors

2. *p*-**multigrid method with ILUT smoother** as efficient solver

Conclusion and outlook

1. **a-posteriori** ***hp*****-adaptation strategy** to find (h, p) pair that ensures *computable* approximations with prescribed accuracy
   - integration as fully automated procedure in simulation code
   - further analysis of influence factors, i.e. iterative solvers
   - use of number formats that are less sensitive to round-off errors

2. *p*-**multigrid method with ILUT smoother** as efficient solver
   - application to biharmonic equation and within NSE solver
   - extension to block-ILUT smoother for multi-patch IGA
   - optimization of assembly procedure in G+Smo

Conclusion and outlook

1. **a-posteriori** ***hp*****-adaptation strategy** to find (h, p) pair that ensures *computable* approximations with prescribed accuracy
   - integration as fully automated procedure in simulation code
   - further analysis of influence factors, i.e. iterative solvers
   - use of number formats that are less sensitive to round-off errors

2. ***p*****-multigrid method with ILUT smoother** as efficient solver
   - application to biharmonic equation and within NSE solver
   - extension to block-ILUT smoother for multi-patch IGA
   - optimization of assembly procedure in G+Smo

High-order methods, are they a curse or a blessing?

Conclusion and outlook

1. **a-posteriori** ***hp*****-adaptation strategy** to find (h, p) pair that ensures *computable* approximations with prescribed accuracy
   - integration as fully automated procedure in simulation code
   - further analysis of influence factors, i.e. iterative solvers
   - use of number formats that are less sensitive to round-off errors

2. *p*-**multigrid method with ILUT smoother** as efficient solver
   - application to biharmonic equation and within NSE solver
   - extension to block-ILUT smoother for multi-patch IGA
   - optimization of assembly procedure in G+Smo

High-order methods, are they a curse or a blessing? ... a challenge!

Thank you very much!