Efficient p-multigrid solvers for Isogeometric Analysis

Mark Looije

TU Delft
Supervised by Roel Tielen and Kees Vuik

April 22, 2021

Motivation

Isogeometric Analysis (IgA) is an extension to the finite element method (FEM).

Motivation

Isogeometric Analysis (IgA) is an extension to the finite element method (FEM).

Many common solvers for linear systems do not perform well for IgA discretizations.

Program

Isogeometric Analysis
Multigrid
Smoothers
Numerical results
Where next?

IgA: Variational form

Start with a PDE

$$\begin{cases} -\Delta u & = f \text{ in } \Omega, \\ u & = 0 \text{ on } \partial\Omega. \end{cases}$$

IgA: Variational form

Start with a PDE

$$\begin{cases} -\Delta u & = f \text{ in } \Omega, \\ u & = 0 \text{ on } \partial\Omega. \end{cases}$$

Multiply with test function $v \in V$, integrate over Ω, integration by parts, Gauss divergence theorem

$$-\int_\Omega \Delta u v \, d\Omega = \int_\Omega f v \, d\Omega,$$

$$-\int_\Omega \text{div}(\nabla u v) \, d\Omega + \int_\Omega \nabla u \nabla v \, d\Omega = \int_\Omega f v \, d\Omega,$$

$$-\int_{\partial\Omega} \nabla u v \cdot n \, d\Gamma + \int_\Omega \nabla u \nabla v \, d\Omega = \int_\Omega f v \, d\Omega,$$

$$\int_\Omega \nabla u \nabla v \, d\Omega = \int_\Omega f v \, d\Omega,$$

$$a(u, v) = \langle f, v \rangle .$$

IgA: Matrix Equation

$a(u, v) = \langle f, v \rangle$ must hold for every $v \in V$.

IgA: Matrix Equation

$a(u, v) = \langle f, v \rangle$ must hold for every $v \in V$.
Replace V by final dimensional subspace V_h, with $\phi_1, ..., \phi_n$ a basis for V_h.

IgA: Matrix Equation

$a(u, v) = \langle f, v \rangle$ must hold for every $v \in V$.

Replace V by final dimensional subspace V_h, with $\phi_1, ..., \phi_n$ a basis for V_h.

Then the numerical approximation is given by

$$u_h = \sum_{i=1}^{n} u_i \phi_i.$$

IgA: Matrix Equation

$a(u, v) = \langle f, v \rangle$ must hold for every $v \in V$.
Replace V by final dimensional subspace V_h, with $\phi_1, ..., \phi_n$ a basis for V_h.
Then the numerical approximation is given by

$$u_h = \sum_{i=1}^{n} u_i \phi_i.$$

Inserting this into the variational form we see

$$a(u_h, \phi_i) = \sum_{i=1}^{n} u_i \; a(\phi_i, \phi_j) = \langle f, \phi_i \rangle \quad \text{for } j = 1, ..., n.$$

IgA: Matrix Equation

$a(u, v) = \langle f, v\rangle$ must hold for every $v \in V$.
Replace V by final dimensional subspace V_h, with $\phi_1, ..., \phi_n$ a basis for V_h.
Then the numerical approximation is given by

$$u_h = \sum_{i=1}^{n} u_i \phi_i.$$

Inserting this into the variational form we see

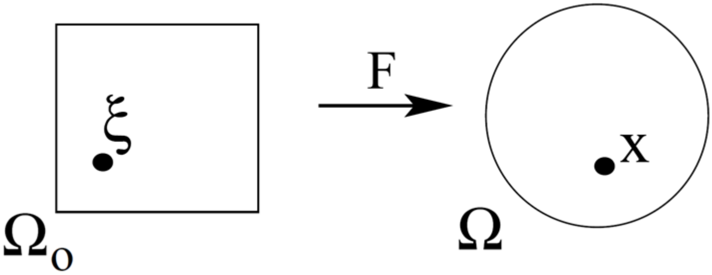
$$a(u_h, \phi_i) = \sum_{i=1}^{n} u_i \; a(\phi_i, \phi_j) = \langle f, \phi_i\rangle \;\; \text{for } j = 1, ..., n.$$

Leading to the matrix equation

$$A\mathbf{u} = \mathbf{f}, \text{ where}$$

$$A_{i,j} = a(\phi_i, \phi_j), \;\; f_i = \langle f, \phi_i\rangle \quad i, j = 1, ..., n.$$

IgA: Geometry



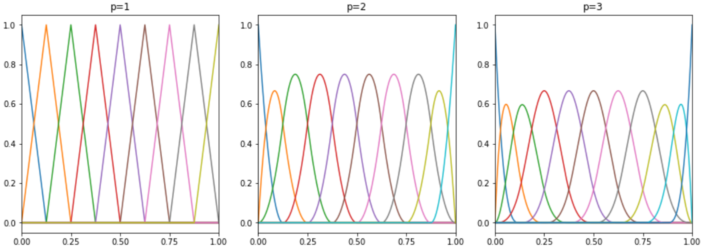
IgA: Geometry

IgA: B-Spline basis functions

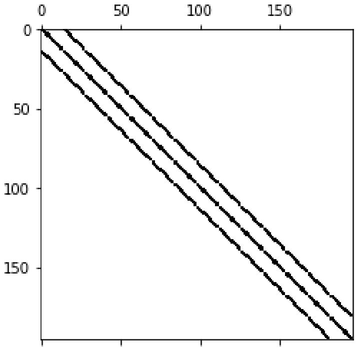
Cox-de Boor formula

$$\phi_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{else}, \end{cases}$$

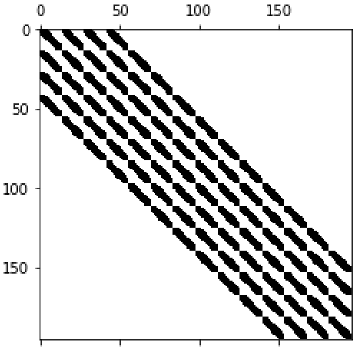
$$\phi_{i,p}(\xi) = \begin{cases} \frac{\xi-\xi_i}{\xi_{i+p}-\xi_i}\phi_{i,p-1}(\xi) + \frac{\xi_{i+p+1}-\xi}{\xi_{i+p+1}-\xi_{i+1}}\phi_{i+1,p-1}(\xi) & \text{if well defined,} \\ 0 & \text{else.} \end{cases}$$



IgA: Support



(a) $p = 1, h = 1/16$



(b) $p = 3, h = 1/16$

Sparisty pattern of various system matrices.

IgA: Recap

Vectors $\mathbf{u}$ and $\mathbf{f}$ consist of coefficients to basis functions.

Non-zero structure of the system matrix A.

MG: Multigrid

Goal: determine $\mathbf{u}$ in $A\mathbf{u} = \mathbf{f}$.

MG: Multigrid

Goal: determine $\mathbf{u}$ in $A\mathbf{u} = \mathbf{f}$.
In a multigrid method we work both on $A\mathbf{u} = \mathbf{f}$, as well as on a connected problem $\tilde{A}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}$.

MG: Multigrid

Goal: determine $\mathbf{u}$ in $A\mathbf{u} = \mathbf{f}$.
In a multigrid method we work both on $A\mathbf{u} = \mathbf{f}$, as well as on a connected problem $\tilde{A}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}$.

Two-grid cycle:

1. Relax ν_1 times on $A\mathbf{u} = \mathbf{f}$ with initial guess $\mathbf{v}$.
2. Compute the residual $\mathbf{r} = \mathbf{f} - A\mathbf{v}$.
3. Restrict the residual $\tilde{\mathbf{r}} = \mathcal{I}_R\mathbf{r}$.
4. Solve $\tilde{A}\tilde{\mathbf{e}} = \tilde{\mathbf{r}}$.
5. Prolongate the error $\mathbf{e} = \mathcal{I}_P\tilde{\mathbf{e}}$.
6. Update the guess $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{e}$.
7. Relax ν_2 times on $A\mathbf{u} = \mathbf{f}$ with initial guess $\mathbf{v}$.

MG: p-multigrid

Motivation: We know most solving methods perform badly for higher values of p.

MG: p-multigrid

Motivation: We know most solving methods perform badly for higher values of p.

Coarser level is based on a lower order discretization.
In fact, we go straight to level $p = 1$.

MG: p-multigrid

Motivation: We know most solving methods perform badly for higher values of p.

Coarser level is based on a lower order discretization.
In fact, we go straight to level $p = 1$.

What does this mean for $\tilde{A}$, $\tilde{\mathbf{v}}$?

MG: L_2-projection

Intergrid operators are based on an L_2-projection.

MG: L_2-projection

Intergrid operators are based on an L_2-projection.
The prolongation- and restriction operator are given by

$$\mathcal{I}_1^p = (M_p)^{-1} P_1^p,$$

$$\mathcal{I}_p^1 = (M_1)^{-1} P_p^1.$$

MG: L_2-projection

Intergrid operators are based on an L_2-projection.
The prolongation- and restriction operator are given by

$$\mathcal{I}_1^p = (M_p)^{-1} P_1^p,$$

$$\mathcal{I}_p^1 = (M_1)^{-1} P_p^1.$$

Here the mass matrices M and transfer matrices P are defined as

$$(M_p)_{i,j} = \int_\Omega \phi_{i,p}\phi_{j,p} \, \mathrm{d}\Omega, \qquad (P_1^p)_{i,j} = \int_\Omega \phi_{i,p}\phi_{j,1} \, \mathrm{d}\Omega,$$

$$(M_1)_{i,j} = \int_\Omega \phi_{i,1}\phi_{j,1} \, \mathrm{d}\Omega, \qquad (P_p^1)_{i,j} = \int_\Omega \phi_{i,1}\phi_{j,p} \, \mathrm{d}\Omega.$$

MG: smoothers

We are searching for methods where convergence is independent of the order of discretization p.

MG: smoothers

We are searching for methods where convergence is independent of the order of discretization p.

Using for example Gauss Seidel as a smoother on the fine grid does not work.

MG: smoothers

We are searching for methods where convergence is independent of the order of discretization p.

Using for example Gauss Seidel as a smoother on the fine grid does not work.

The method chosen is an ILUT-smoother, based on an incomplete LU factorization.

MG: ILUT smoother

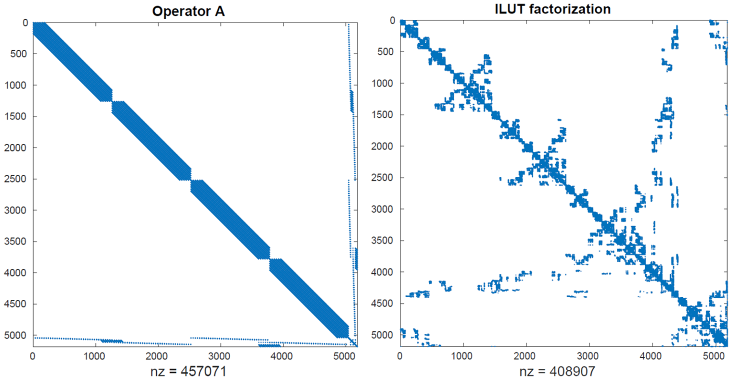


Figure: Sparsity pattern of A and its ILUT factorization.
Image from "A block ILUT smoother for multipatch geometries in Isogeometric Analysis" by R. Tielen, M. Möller and K. Vuik

MG: Recap

We want a method where convergence is independent of p.
This is achieved by a p-multigrid method, with coarsening based on a lower order discretization.
As a smoother on the fine level we can use an ILUT smoother.

NR: Problem description

2-dimensional homogeneous Poisson equation.

$$\begin{cases} -\Delta u = f, & \text{on } [0,1]^2 \\ u = 0, & \text{on the boundary.} \end{cases}$$

With right hand side $f(x) = 2\pi^2 \sin(\pi x)\sin(\pi y)$ and initial guess the zero vector $u^0 \equiv 0$.

The primary thing we are interested in for our multigrid program is to see how many steps it takes to converge. For this we use the stopping criterium

$$\frac{||\mathbf{r}^k||}{||\mathbf{r}^0||} < 10^{-8}.$$

where $\mathbf{r}^k$ and $\mathbf{r}^0$ are the residuals after k and 0 steps respectively.

NR: *h*-multigrid

	p=1	p=2	p=3	p=4	p=5	p=6		
Ndof=5	4	8	33	XX	XX	XX		2-dim Poisson problem, IgA discretization
Ndof=9	6	8	28	116	730	XX		h-multigrid, smoother = GS
Ndof=17	8	9	25	74	311	840		steps till convergence
Ndof=33	8	9	24	72	228	720		
Ndof=65								

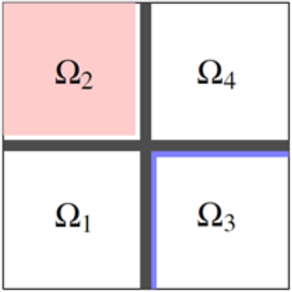
NR: p-multigrid + Gauss Seidel

	p=1	p=2	p=3	p=4	p=5	p=6		
Nel=4	-	9	30	113	388	708		2-dim Poisson problem, IgA discretization
Nel=8	-	8	22	84	347	1042		p-multigrid, smoother = GS
Nel=16	-	7	23	68	254	793		steps till convergence
Nel=32	-	5	21	63	204	774		
Nel=64	-							

NR: p-multigrid + ILUT smoother

	p=1	p=2	p=3	p=4	p=5	p=6		
Nel=4	-	2	2	2	2	3		2-dim Poisson problem, IgA discretization
Nel=8	-	3	2	2	2	2		p-multigrid, smoother = ILUT
Nel=16	-	3	3	3	3	3		steps till convergence
Nel=32	-	3	3	3	3	3		
Nel=64	-							

Next: Multipatch



$$\begin{bmatrix} A_{11} & 0 & 0 & 0 & A_{\Gamma 1} \\ 0 & A_{22} & 0 & 0 & A_{\Gamma 2} \\ 0 & 0 & A_{33} & 0 & A_{\Gamma 3} \\ 0 & 0 & 0 & A_{44} & A_{\Gamma 4} \\ A_{1\Gamma} & A_{2\Gamma} & A_{3\Gamma} & A_{4\Gamma} & A_{\Gamma\Gamma} \end{bmatrix}$$

Next: Block Structure

$$A = \begin{bmatrix} A_{11} & & 0 & A_{\Gamma 1} \\ & \ddots & & \vdots \\ 0 & & A_{KK} & A_{\Gamma K} \\ A_{1\Gamma} & \dots & A_{K\Gamma} & A_{\Gamma\Gamma} \end{bmatrix}.$$

Next: Block Structure

$$A = \begin{bmatrix} A_{11} & & 0 & A_{\Gamma 1} \\ & \ddots & & \vdots \\ 0 & & A_{KK} & A_{\Gamma K} \\ A_{1\Gamma} & \dots & A_{K\Gamma} & A_{\Gamma\Gamma} \end{bmatrix}.$$

$$A = LU = \begin{bmatrix} L_1 & & & \\ & \ddots & & \\ & & L_K & \\ B_1 & \dots & B_K & I \end{bmatrix} \begin{bmatrix} U_1 & & & C_1 \\ & \ddots & & \vdots \\ & & U_K & C_k \\ & & & S \end{bmatrix}.$$

Next: Block ILUT

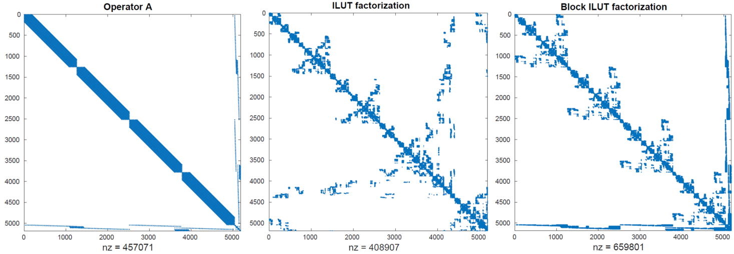


Figure: Sparsity pattern of A, its global ILUT factorization and its block ILUT factorization.
Image from "A block ILUT smoother for multipatch geometries in Isogeometric Analysis" by R. Tielen, M. Möller and K. Vuik.

Efficient p-multigrid solvers for Isogeometric Analysis

Mark Looije

TU Delft
Supervised by Roel Tielen and Kees Vuik

April 22, 2021