Isogeometric Analysis and Schwarz Additive DD, DD22, 2013

Michel Bercovier

Hebrew University of Jerusalem

December 7, 2013

(Joint work with Ilya Soloveichick)

DD, IGA and Schwarz

- Solid Modelling (CSG)
- Schwarz Additive Domain Decomposition and Isogeometric Analysis
- Boundary Conditions
- 1D numerical results
- 2D examples and application to local zooming
- 3D heat and elasticity examples
- Parallelisation

DD, IGA and Schwarz

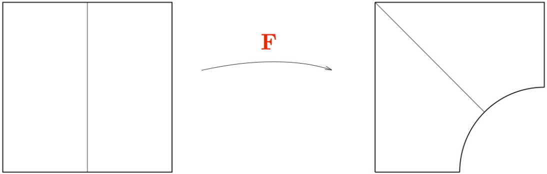
- Solid Modelling (CSG)
- Schwarz Additive Domain Decomposition and Isogeometric Analysis
- Boundary Conditions
- 1D numerical results
- 2D examples and application to local zooming
- 3D heat and elasticity examples
- Parallelisation
- Current Research and Conclusion

All computations were done using GeoPDES 1.2

C. de Falco, A. Reali, and R. Vazquez. GeoPDEs: a research tool for Isogeometric Analysis of PDEs. *Advances in Software Engineering*,40 (2011),1020-1034.

Isogeometric Mapping

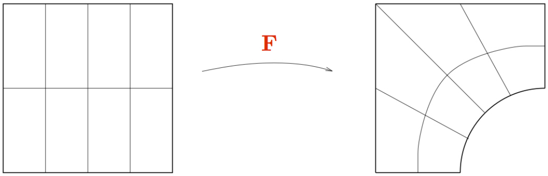
parametric domain $\widehat{\Omega}$ physical domain Ω



$\{B_i\}_{i=1,\ldots,N_0}$ push forward of $\{B_i\}_{i=1,\ldots,N_0}$

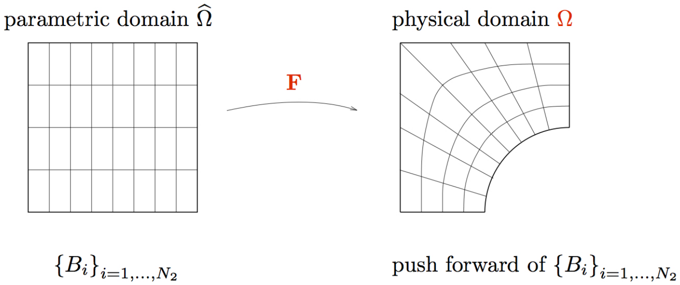
Isogeometric Mapping

parametric domain $\widehat{\Omega}$ physical domain Ω

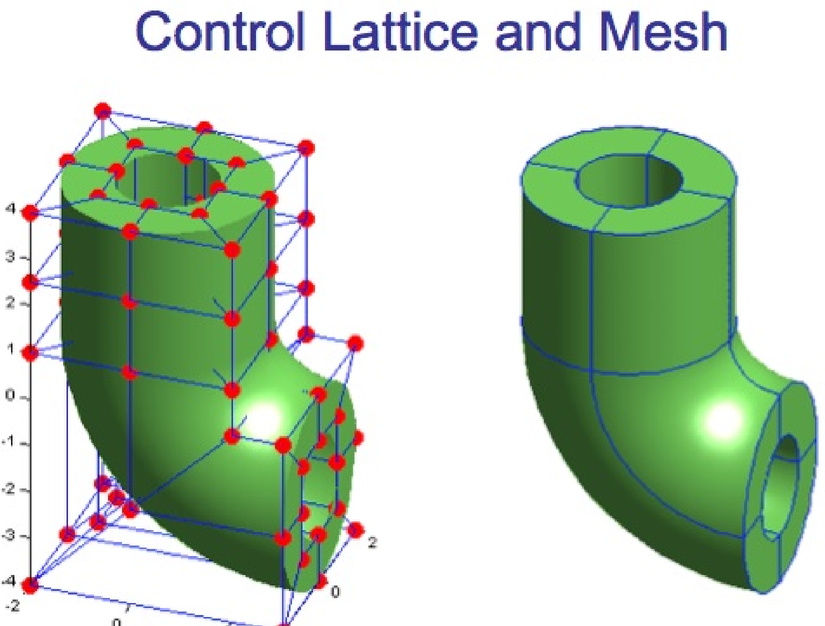


$\{B_i\}_{i=1,\dots,N_1}$ push forward of $\{B_i\}_{i=1,\dots,N_1}$

Isogeometric Mapping a

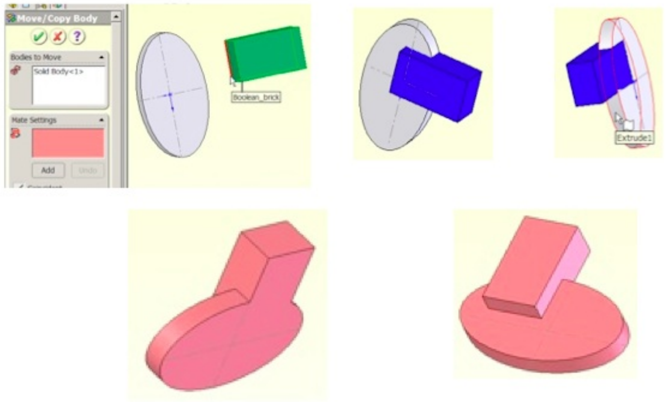


Example of Tensor Product Domain



Solid Modelling

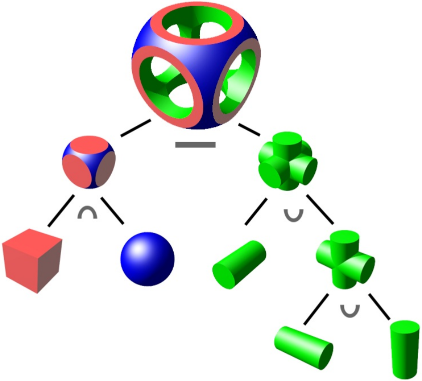
- Constructive Solid Geometry (CSG) relies on boolean operations of primitives



Example of Boolean Union

Solid Modelling

- Constructive Solid Geometry (CSG) relies on boolean operations of primitives.

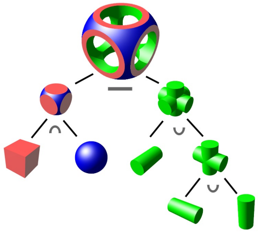


More Complex Construct: Union and Substraction

IGA-alone

CSG is only one of, but an important one, tool used by designers.

To define the isogeometric mapping for these complex structures one needs the latest results like



W. Wang, Y. Zhang, L. Liu and T.J.R. Hughes: Solid T-spline Construction from Boundary Triangulations with Arbitrary Genus Topology, ICES 2012

Research on DD and IGA is growing, see the conference by L. Beirao da Vega this morning and the present Mini Symposium !

IGA-alone

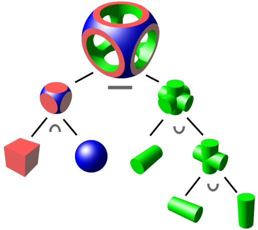
CSG is only one of, but an important one, tool used by designers.

To define the isogeometric mapping for these complex structures one needs the latest results like

Another alternative:

- Using Domain Decomposition

Methods for CSG defined solids.



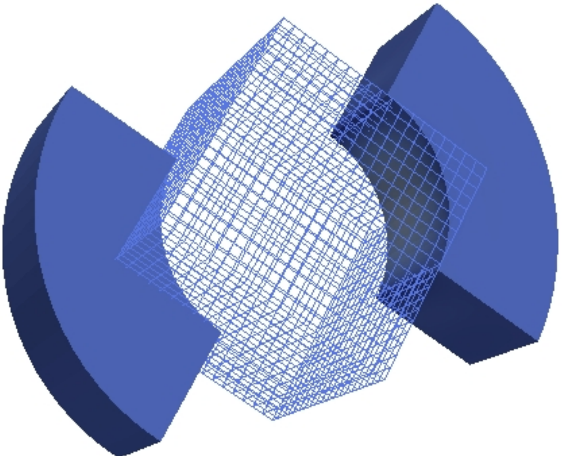
W. Wang, Y. Zhang, L. Liu and T.J.R. Hughes: Solid T-spline Construction from Boundary Triangulations with Arbitrary Genus Topology, ICES 2012

L. Beirao da Veiga, D. Cho, L. Pavarino, and S. Scacchi. Overlapping Schwarz methods for Isogeometric Analysis. SIAM J. N. A., 2012

Research on DD and IGA is growing, see the conference by L. Beirao da Vega this morning and the present Mini Symposium !

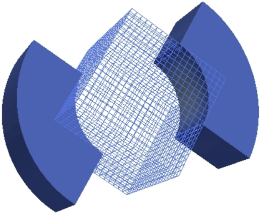
Domain Decomposition and IGA

WHAT PROPERTIES DO WE DEMAND?



WHAT PROPERTIES DO WE DEMAND?

- Non-matching meshes

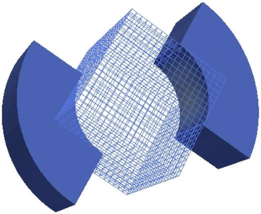


- Easy parallelisation

Domain Decomposition and IGA

WHAT PROPERTIES DO WE DEMAND?

- Non-matching meshes

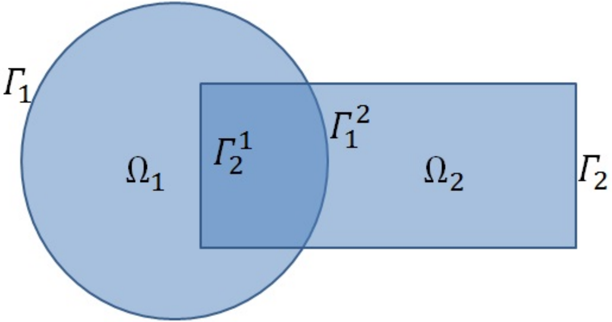


- Easy parallelisation

Schwarz Additive Domain Decomposition fits

DD and IGA

Consider the equation $\Delta u = 0, u|_\Omega = 0$ on the domain given by the logo of Domain Decomposition Organization, which is the union of a circle and an overlapping rectangle: $\Omega = \Omega_1 \cup \Omega_2$.
The boundary of Ω is $\Gamma = \Gamma_1 \cup \Gamma_2$, where Γ_1 (resp. Γ_2) is the boundary of $\Omega_1 \backslash \Omega_2$ (resp. $\Omega_2 \backslash \Omega_1$).



Classic Reminder

Define a bilinear form $a(\cdot,\cdot) : \mathcal{H}^1(\Omega) \times \mathcal{H}^1(\Omega) \to \mathbb{R}$ and a functional $L : \mathcal{H}^1(\Omega) \to \mathbb{R}$ as:

$$a(w,u) = \int_\Omega \mathbf{grad} w \cdot \mathbf{grad} u \mathbf{d}\Omega, \tag{1}$$

and

$$L(w) = \int_\Omega wf\mathbf{d}\Omega + \int_{\Gamma_N} wh\mathbf{d}\Gamma. \tag{2}$$

Now the weak form reads as:

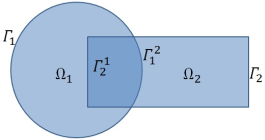
$$a(w,u) = L(w), \tag{3}$$

DD continuous case

Since we assume the domain boundaries to be piecewise smooth and the solutions belong to $\mathcal{H}^1(\Omega_j)$ spaces, the Trace theorem provides us with the definition of the trace operator:

$$P_i : \mathcal{H}^1(\Omega_j) \to \mathcal{L}^2(\Gamma_i^j)$$

$$u_j \to u_j|_{\Gamma_i^j}.$$



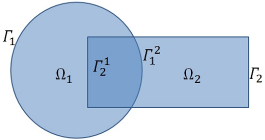
DD continuous case

There exist an extension operators :

$$E_i : \mathcal{H}^1(\Gamma_i^j) \to \mathcal{H}^1(\Omega_i)$$

$$v_i \to u_i \text{ such that } u_i|_{\Gamma_i^j} = v_i, u_i|_{\Gamma_i} = g|_{\Gamma_i}.$$

Here we assume that the conditions $u_i|_{\Gamma_i^j} = v_i, u_i|_{\Gamma_i} = g|_{\Gamma_i}$ define some continuous $\mathcal{H}^1(\Omega_i)$ function.
The continuous version of the ASDDM follows.



DD continuous case

Given initial guesses $u_i^0 \in \mathcal{H}^1(\Omega_i), i = 1, 2$, such that $u_i^0|_{\Gamma_i^j} = g|_{\Gamma_i^j}, i, j = 1, 2$

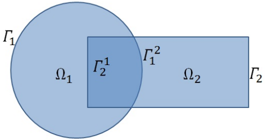
While convergence conditions are not met:

Find $u_i^n \in \mathcal{H}^1(\Omega_i)$ *such that* $u_i^n|_{\Gamma_i} = g|_{\Gamma_i}$ and

$$a_i(u_i^n - E_i P_i u_j^{n-1}, v_i) = L_i(v_i) - a_i(E_i P_i u_j^{n-1}, v_i)$$

for any $v_i \in \mathcal{H}_0^1(\Omega_i), i, j = 1, 2, i \neq j.$

ASDDM algorithm



DD and IGA, finite dimensional case

Given $V_i \subset H_0^1(\Omega_i) = [\Phi_{(i,j)}(\underline{x}) = B_j(F_i^{-1}(\underline{x}));]$, set $u_1^0 = 0, u_2^0 = 0$;

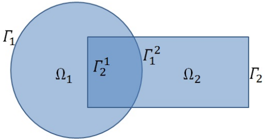
Given $u_i^0 \in V_i, i = 1, 2$, such that $u_i^0 l_{\Gamma_i^j} = g|_{\Gamma_j}, i = 1, 2,$
for $i = 1, 2; j = 1, 2, i \neq j$
While convergence conditions are not met:

Find $u_i^n \in V_i$ such that $u_i^n|_{\Gamma_i} = g|_{\Gamma_i}$, and

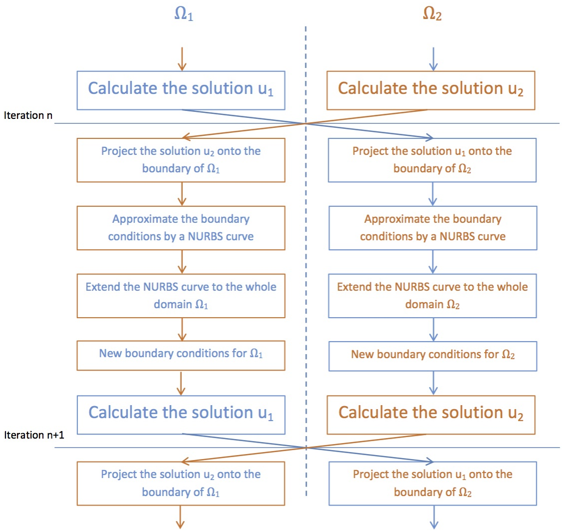
$$a_i(u_i^n - E_i P_i u_j^{n-1}, v_i) = L_i(v_i) - a_i(E_i P_i u_j^{n-1}, v_i)$$

for any $v_i \in V_i$.

ASDDM algorithm



DD and IGA, algorithm



DD and IGA

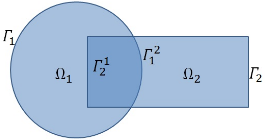
We test the convergence on

$$||u^m_{1,\Omega_1\cap\Omega_2} - u^{m-1}_{1,\Omega_1\cap\Omega_2}|| \text{ and } ||u^m_{2,\Omega_1\cap\Omega_2} - u^{m-1}_{2,\Omega_1\cap\Omega_2}||$$

There is no notion of a global approximate solution, on the overlap. We define the global solution by choosing the subdomain iterative solution on each subdomain and any weighted average within the overlap:

$$u^n = \chi_1 v^n + \chi_2 w^n,$$

where $\chi_1 = 1$ on $\Omega_1 \backslash (\Omega_1 \cap \Omega_2)$, $\chi_2 = 1$ on $\Omega_2 \backslash (\Omega_1 \cap \Omega_2)$ and $\chi_1 + \chi_2 = 1$ on $\Omega_1 \cap \Omega_2$.
This approach extends easily to multi-patched domain decomposition.

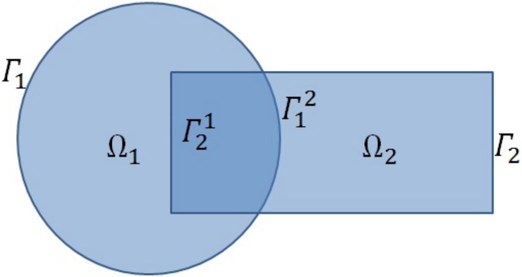


Overlapping Schwarz and BC

Construction of the Trace Operator

We tested two approaches.

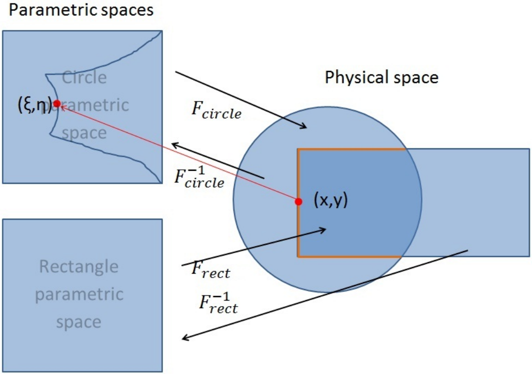
- Computation in the parametric space
- Computation in the physical space



Parametric space approach

Pre-Image computation

Overlapping boundaries are trimming curves in the parameteric space.



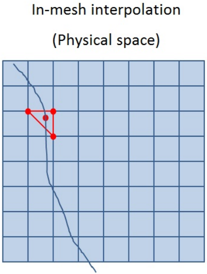
Given any point (x, y) on the boundary Γ^i_j of Ω_j : compute its pre-image coordinates (ξ, η) in the parametric space $\widehat{\Omega}_i$.

Evaluate the solution u^m_i at that given (ξ, η) point at each iteration.

Physical space approach

Second approach : work only in the physical space.

The solution u_i in the domain Ω_i is tabulated at some mesh coordinates $(\mathbf{X}, \mathbf{Y})$ (say, integration points).
For any point (x, y) at which this solution is to be evaluated (the integration points of the boundary of the sub-domain Ω_j) we interpolate by linear or cubic polynomials.



This does not affect the rate of the iterative convergence significantly.

Non-homogeneous Dirichlet BC in IGA

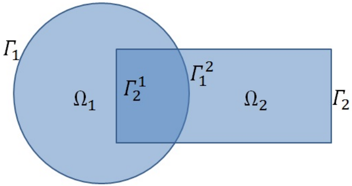
In both approaches we have a discrete collection of values on the boundary Γ_i^j

We still have to convert this information into boundary B-spline degrees of freedom.

We considered two methods:

Least-square approximation of the Dirichlet BC

Quasi Interpolation



There are also other approaches to impose the Dirichlet boundary conditions such as local least-squares (Govindjee et All.) or Nitsche's method (Harari)

Non-homogeneous Dirichlet BC in IGA

Least-squares approximation of the Dirichlet BC

Find such coefficients $\{\widetilde{q}_i\}_{i=N_h+1}^{N_h+N_h^b}$ that minimize the following integral:

$$\min_{\{q_i\}_{i=N_h+1}^{N_h+N_h^b}} \int_{\Gamma_D} (g(\mathbf{x}) - \sum_{N_h+1}^{N_h+N_h^b} q_i \phi_i(\mathbf{x}))^2 \mathbf{d}\Gamma.$$

In practice we will compute the integral using numerical formula, such as Gaussian quadrature, hence **x** will be computed at a predetermined collection of values , in the parametric domain.

Non-homogeneous Dirichlet BC in IGA

Quasi interpolation of the Dirichlet BC

Consider a point-wise approximation of $g(\mathbf{x})$.

Assume that the "intersectting" boundary Γ corresponds to one side of the parametric domain with N degrees of freedom.

Given N points on this boundary, solve the system of linear equations:

$$\sum_{j=1}^{N} q_j \phi_j(x_i, y_i) = g(x_i, y_i), i = 1 \dots N.$$

Applying the same algorithm to all the sides of the domain where the Dirichlet BC are imposed, gives the value of the corresponding degrees of freedom.

Quasi interpolation of the Dirichlet BC

Take $\{(x_i, y_i)\}_{i=1}^{N}$ as the images of the centers of the knot spans $\Rightarrow$ they are uniformly distributed in the parametric space.

These points may be chosen in different ways (e.g., uniform chord length).

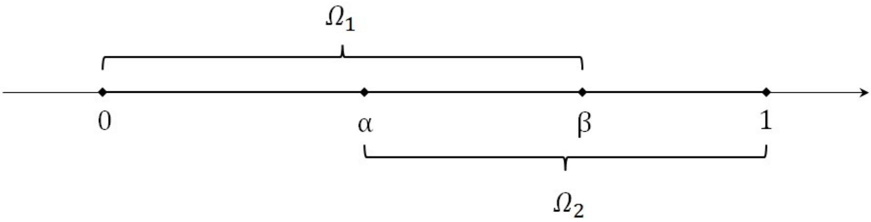
One of the open questions we are interested in is to choose these points optimally, given the geometry.

Dirichlet BC, 1D example

1D Domain Decomposition example :
two overlapping domains, non-matching grid.

$$\Delta u = -1,$$

$$u(0) = u(1) = 0.$$



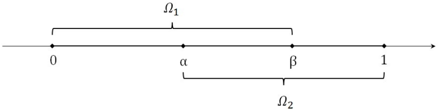
Dirichlet BC, 1D example: Hypotheses and Notations

Uniform open knot vectors : $\Xi_1 = [0, \ldots \beta]$ and $\Xi_2 = [\alpha, \ldots 1]$;
B-splines of degree p_1 and p_2 on the subdomains Ω_1 and Ω_2, respectively.;N_{h1} and N_{h2}, the resp. numbers of degrees of freedom
The mapping F is the identity mapping,(parametric space and physical space are the same)

Let v (resp. w) be the solution on Ω_1 (resp. Ω_2) .Formally we discretize:

$$v_{xx}^n = f \text{ on } \Omega_1, v^n(0) = 0, v^n(\beta) = w^{n-1}(\beta). \tag{4}$$

$$w_{xx}^n = f \text{ on } \Omega_2, w^n(0) = 0, w^n(\alpha) = v^{n-1}(\alpha). \tag{5}$$

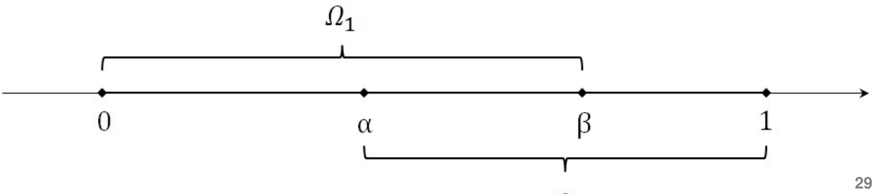


1D example: Construction of the Iteration Matrix

The basis functions are numbered by the order of the knot vectors. Only the first and last basis functions are interpolatory, and all the others vanish at the boundary $t = 0$ and $t = 1$.

Consider subdomain Ω_1. The first (resp. last)basis function satisfies $\phi_1(0) = 1$, (resp. $\phi_{N_{h1}}(\beta) = 1$.)
The approximation operators A_i are the identities.

Given the solution $\widetilde{w}^{n-1} = \sum_{i=1}^{N_{h2}} \psi_i w_i^{n-1}$ at boundary point β we project it on the second subdomain:
$v^n_{N_{h1}} = P_1(\widetilde{w}^{n-1}(\beta))$, where P_1 is the trace operator.
The same reasoning applies on Ω_2.

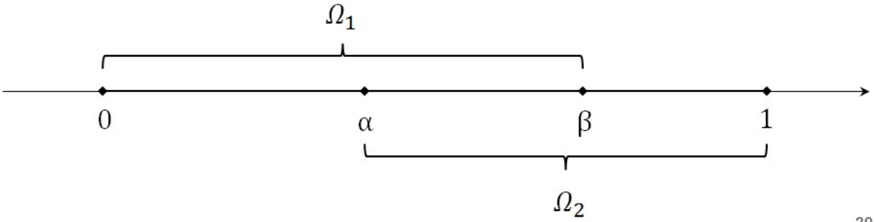


1D example,Construction of the Iteration Matrix

Denote the vector of degrees of freedom of the solution $\widetilde{w}$ (resp. $\widetilde{v}$) as **w**, (resp. **v**.)

We can now build a matrix representation of the iterations .
The degree of B-splines on Ω_2 is p_2 thus there are no more than $p_2 + 1$ non-zero basis functions ψ_i at the point β.
We may use different trace operators in order to project the solution on the subdomain Ω_2 onto the boundary Γ_1^2 of the subdomain Ω_1.

We will consider both cases: the exact trace operator and the interpolation trace operator



1D example, Iteration Matrix , Pre-Image

For this trivial trace operator, the value $v^n_{N_{h1}}$ we get is:
$v^n_{N_{h1}} = P_1(\widetilde{w}^{n-1}(\beta))$, where P_1 is the trace operator.
The operator matrix for the boundary of the Ω_1 subdomain is:

$$v^n_{N_{h1}} = P_1(\widetilde{w}^{n-1}(\beta)) = P_1^e \cdot \begin{pmatrix} 0 \\ \vdots \\ w_i^{n-1} \\ w_{i+1}^{n-1} \\ \vdots \\ w_{i+p_2}^{n-1} \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & \psi_i(\beta) & \psi_{i+1}(\beta) & \dots & \psi_{i+p_2}(\beta) & \dots & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \vdots \\ w_i^{n-1} \\ w_{i+1}^{n-1} \\ \vdots \\ w_{i+p_2}^{n-1} \\ \vdots \\ 0 \end{pmatrix}$$

where $P_1^e \in \mathbb{R}^{N_{h1} \times N_{h2}}$.

1D example, Iteration Matrix , Pre-Image

Denote by A_1 the stiffness matrix for the first subdomain. We partition the sets of indices of basis functions $\mathcal{J} = \{1, 2, \ldots, N_{hj}\}, j = 1, 2$ into two subsets. The subset $\mathcal{I} \subset \mathcal{J}$ of the inner degrees of freedom and $\mathcal{B} \subset \mathcal{J}$ of the boundary degrees of freedom.
In the one dimensional case $\mathcal{B} = \{1, N_{hj}\}$ and
$\mathcal{I} = \mathcal{J} \backslash \{1, N_{hj}\} = \{2, 3, \ldots, N_{hj} - 1\}, j = 1, 2.$

The restriction of the stiffness matrix corresponding to the inner degrees of freedom is $A_1(\mathcal{I}, \mathcal{I})$.

When we impose the Dirichlet boundary conditions on some of the degrees of freedom, we have to substract the corresponding values from the degrees of freedom of the basis functions $\{\phi\}_{i=N_{h1}-k}^{N_{h1}}$ which support intersects with the support of $\phi_{N_{h1}}$. This corresponds to the *thickness of the interface* in standard IGA-DD methods ...

1D example, Iteration Matrix , Pre-Image

Now the discretized equation for the subdomain Ω_1 can be rewritten as:

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & & A_1(\mathcal{I},\mathcal{I}) & & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \cdot \mathbf{v}^n = \begin{pmatrix} 0 \\ \mathbf{f_1} \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & \dots & 0 \\ & -A_1(\mathcal{I},\mathcal{B}) & \\ 0 & \dots & 1 \end{pmatrix} \cdot$$
$$\cdot \begin{pmatrix} 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & \psi_i(\beta) & \psi_{i+1}(\beta) & \dots & \psi_{i+p_2}(\beta) & \dots & 0 \end{pmatrix} \cdot \mathbf{w}^{n-1}, \tag{6}$$

where the vector $\mathbf{f_1}$ corresponds to the inner degrees of freedom of the first subdomain.

1D example, Iteration Matrix , Pre-Image

Exactly the same reasoning can be applied to the second subdomain Ω_2 to get the following discretized equation:

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & & A_2(\mathcal{I},\mathcal{I}) & & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \cdot \mathbf{w}^n = \begin{pmatrix} 0 \\ \mathbf{f_2} \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & \dots & 0 \\ & -A_2(\mathcal{I},\mathcal{B}) & \\ 0 & \dots & 1 \end{pmatrix} \cdot$$
$$\cdot \begin{pmatrix} 0 & \dots & \phi_j(\alpha) & \phi_{j+1}(\alpha) & \dots & \phi_{j+p_1}(\alpha) & \dots & 0 \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix} \cdot \mathbf{v}^{n-1}. \tag{7}$$

We regroup all the definitions and steps :

$$P = \left(\begin{array}{ccccccc|ccccccc} & & & & & & & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ & & & & & & & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ & & & O & & & & \vdots & & \vdots & & \vdots & & \vdots \\ & & & & & & & \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots \\ & & & & & & & 0 & \dots & \psi_i(\beta) & \dots & \psi_{i+p_2}(\beta) & \dots & 0 \\ 0 & \dots & \phi_j(\alpha) & \dots & \phi_{j+p_1}(\alpha) & \dots & 0 & & & & & & & \\ \vdots & & \vdots & & \vdots & & \vdots & & & & & & & \\ \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & & & & O & & & \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 & & & & & & & \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 & & & & & & & \end{array} \right) \tag{8}$$
$$= \begin{pmatrix} O & \tilde{P}_1 \\ \tilde{P}_2 & O \end{pmatrix},$$

with $P_1^e \in \mathbb{R}^{N_{h1} \times N_{h2}}$, $P_2^e \in \mathbb{R}^{N_{h2} \times N_{h1}}$.

1D example, Iteration Matrix , Pre-Image

For the stiffness matrices we have:

$$A = \begin{pmatrix} 1 & \dots & 0 & & & \\ 0 & A_1(\mathcal{I},\mathcal{I}) & 0 & & O & \\ 0 & \dots & 1 & & & \\ & & & 1 & \dots & 0 \\ & O & & 0 & A_2(\mathcal{I},\mathcal{I}) & 0 \\ & & & 0 & \dots & 1 \end{pmatrix} = \begin{pmatrix} \tilde{A}_1 & O \\ O & \tilde{A}_2 \end{pmatrix},$$

with $\tilde{A}_1 \in \mathbb{R}^{N_{h1} \times N_{h1}}$, $\tilde{A}_2 \in \mathbb{R}^{N_{h2} \times N_{h2}}$,

$$f = \begin{pmatrix} 0 \\ \mathbf{f_1} \\ 0 \\ 0 \\ \mathbf{f_2} \\ 0 \end{pmatrix},$$

$$A_{dir} = \begin{pmatrix} 1 & \dots & 0 & & & \\ & -A_1(\mathcal{I},\mathcal{B}) & & & O & \\ 0 & \dots & 1 & & & \\ & & & 1 & \dots & 0 \\ & O & & & -A_2(\mathcal{I},\mathcal{B}) & \\ & & & 0 & \dots & 1 \end{pmatrix} = \begin{pmatrix} \widetilde{A}_{dir1} & O \\ O & \widetilde{A}_{dir2} \end{pmatrix},$$

with $\widetilde{A}_{dir1} \in \mathbb{R}^{N_{h1} \times N_{h1}}$, $\widetilde{A}_{dir2} \in \mathbb{R}^{N_{h2} \times N_{h2}}$,

1D example, Iteration Matrix , Pre-Image

Let us denote the vector of degrees of freedom of the "whole" approximate solution $\widetilde{u}^n$ as $\mathbf{u}^n = \begin{pmatrix} \mathbf{v}^n \\ \mathbf{w}^n \end{pmatrix}$. Now, we can unite the two equations (6) and (7) into a single matrix equation

$$A \cdot \mathbf{u}^n = f + A_{dir} \cdot P \cdot \mathbf{u}^{n-1}, \tag{9}$$

The iterative scheme is given by:

$$\begin{aligned} \begin{pmatrix} \tilde{A}_1 & O \\ O & \tilde{A}_2 \end{pmatrix} \cdot \mathbf{u}^n = f + \begin{pmatrix} \widetilde{A}_{dir1} & O \\ O & \widetilde{A}_{dir2} \end{pmatrix} \cdot \begin{pmatrix} O & P_1^e \\ P_2^e & O \end{pmatrix} \cdot \mathbf{u}^{n-1} = \\ f + \begin{pmatrix} O & \widetilde{A}_{dir1} P_1^e \\ \widetilde{A}_{dir2} P_2^e & O \end{pmatrix} \cdot \mathbf{u}^{n-1}, \end{aligned} \tag{10}$$

1D example, Iteration Matrix ,Physical Space

Here we have a simple linear operator.
The algorithm works as was explained above: consider the first subdomain $\Omega_1 = [0, \beta)$. In order to construct a linear interpolation of the solution $\widetilde{w}^{n-1}$ at the point $\eta = \beta$ we need to find the knot span $[\eta_i, \eta_{i+1})$ which contains β. We take the values of the function $\widetilde{w}^{n-1}$ at the ends of this chosen span interval $\widetilde{w}^{n-1}(\eta_i)$ and $\widetilde{w}^{n-1}(\eta_{i+1})$ and their weighted sum:

$$v^n_{N_{h1}} = \frac{\beta - \eta_i}{\eta_{i+1} - \eta_i} \widetilde{w}^{n-1}(\eta_i) + \frac{\eta_{i+1} - \beta}{\eta_{i+1} - \eta_i} \widetilde{w}^{n-1}(\eta_{i+1}).$$

We see that the linear interpolation trace operator is, actually, a convex sum of the two exact interpolation operators corresponding to the points η_i and η_{i+1}:

$$P^l_1 = \frac{\beta - \eta_i}{\eta_{i+1} - \eta_i} P^e(\eta_i) + \frac{\eta_{i+1} - \beta}{\eta_{i+1} - \eta_i} P^e(\eta_{i+1}). \tag{11}$$

Consequently, the matrix of this interpolation trace operator can be obtained as the convex sum of the matrices of the exact trace operators at the points η_i and η_{i+1}.

$$\begin{aligned} P^l_1 = {} & \frac{\beta - \eta_i}{\eta_{i+1} - \eta_i} \begin{pmatrix} 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & \psi_i(\eta_i) & \psi_{i+1}(\eta_i) & \dots & \psi_{i+p_2}(\eta_i) & \dots & 0 \end{pmatrix} \\ & + \frac{\eta_{i+1} - \beta}{\eta_{i+1} - \eta_i} \begin{pmatrix} 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & \psi_{i+1}(\eta_{i+1}) & \psi_{i+2}(\eta_{i+1}) & \dots & \psi_{i+p_2+1}(\eta_{i+1}) & \dots & 0 \end{pmatrix}. \end{aligned} \tag{12}$$

1D example, Iteration Matrix ,Physical Space

The iterative scheme is now:

$$\begin{pmatrix} \tilde{A}_1 & O \\ O & \tilde{A}_2 \end{pmatrix} \cdot \mathbf{u}^n = f + \begin{pmatrix} \widetilde{A}_{dir1} & O \\ O & \widetilde{A}_{dir2} \end{pmatrix} \cdot \begin{pmatrix} O & P_1^l \\ P_2^l & O \end{pmatrix} \cdot \mathbf{u}^{n-1} = \\ f + \begin{pmatrix} O & \widetilde{A}_{dir1} P_1^l \\ \widetilde{A}_{dir2} P_2^l & O \end{pmatrix} \cdot \mathbf{u}^{n-1}, \tag{13}$$

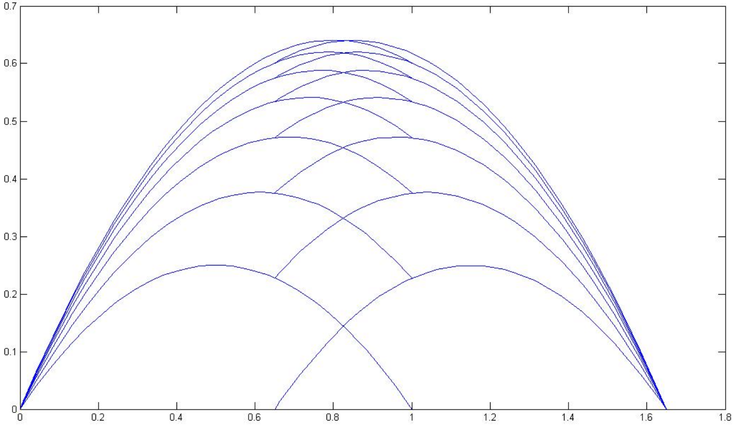
where

$$P_2^l = \frac{\alpha - \xi_j}{\xi_{j+1} - \xi_j} P^e(\xi_j) + \frac{\xi_{j+1} - \alpha}{\xi_{j+1} - \xi_j} P^e(\xi_{j+1}). \tag{14}$$

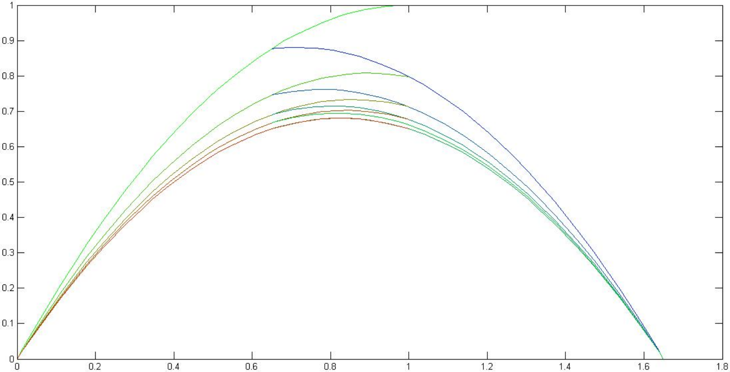
and α belongs to the knot span $[\xi_j, \xi_{j+1})$

Dirichlet BC, 1D example

1D example using the parametric space approach



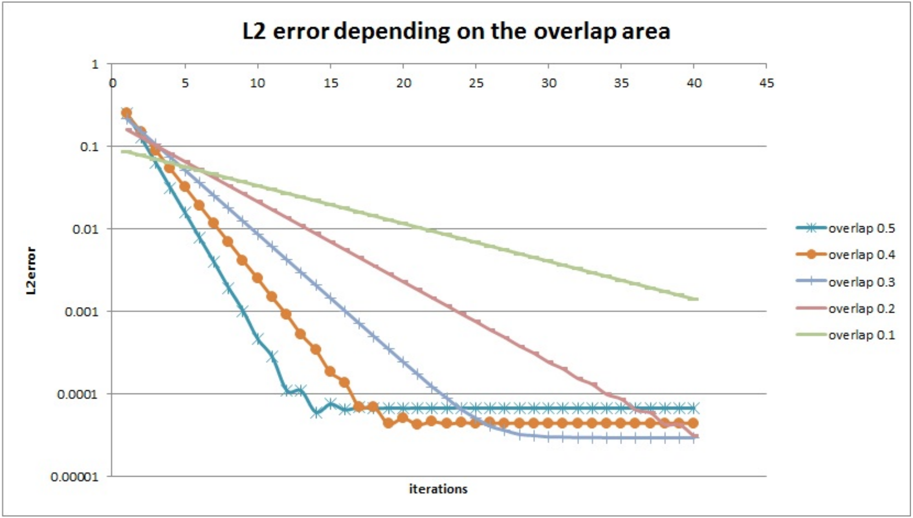
Dirichlet BC, 1D example



Convergence of the one-dimensional SADDM with non-zero initial guesses. $\widetilde{v}^0(\beta) = \beta$ and $\widetilde{w}^0(\alpha) = \alpha$

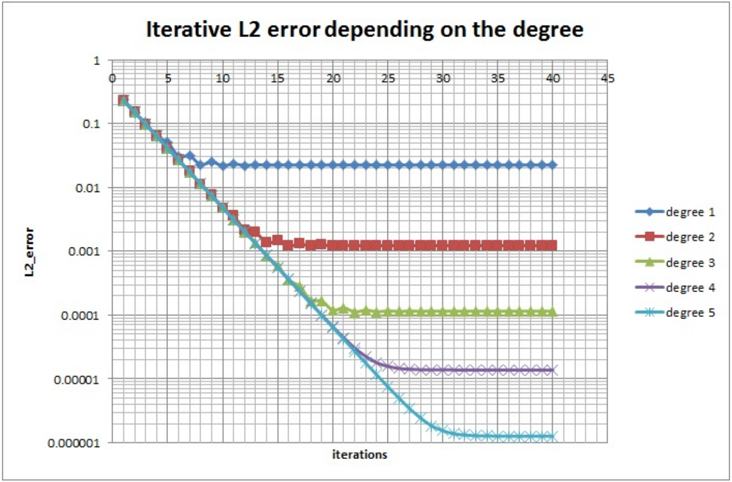
1D example and Overlapping

Iteration speed vs % of overlapping of domains.



DD iteration and degree of approximation

The impact of the degree *p* of the B-splines on the iterative convergence.



Numerical Results

2D numerical results:

- Zooming DD convergence vs. analytical solutions
- Zooming DD convergence in a singular case
- Zooming: approximation of singular derivative

Analytical example, zoom and convergence

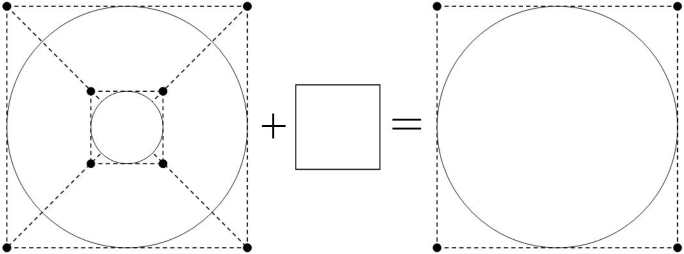
Let Ω be a circle of radius 3, $u = sin(x^2 + y^2 - 9)$ is the solution of

$$-\Delta u = sin(x^2 + y^2 - 9) - 4cos(x^2 + y^2 - 9),$$

$$u|_{\partial\Omega=0}.$$

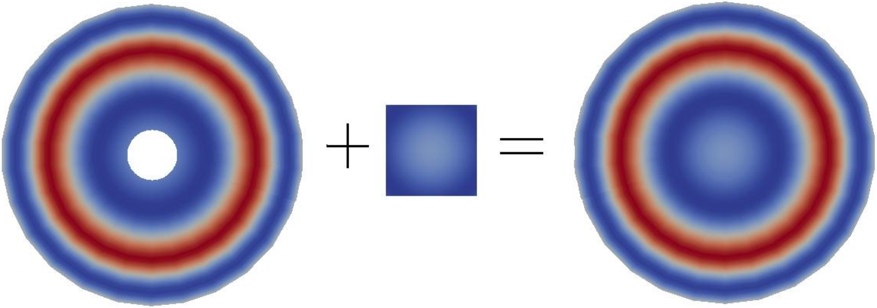
Apply zooming by considering Ω as a union of an annulus and a square. $\Omega = \Omega_{annulus} \cup \Omega_{square}$.

F. Hecht, A. Lozinski and O. Pironneau, *Numerical Zoom and the Schwarz Algorithm*, D. D. Methods in Science and Eng. XVIII, Springer Verlag 2009, 63-74.



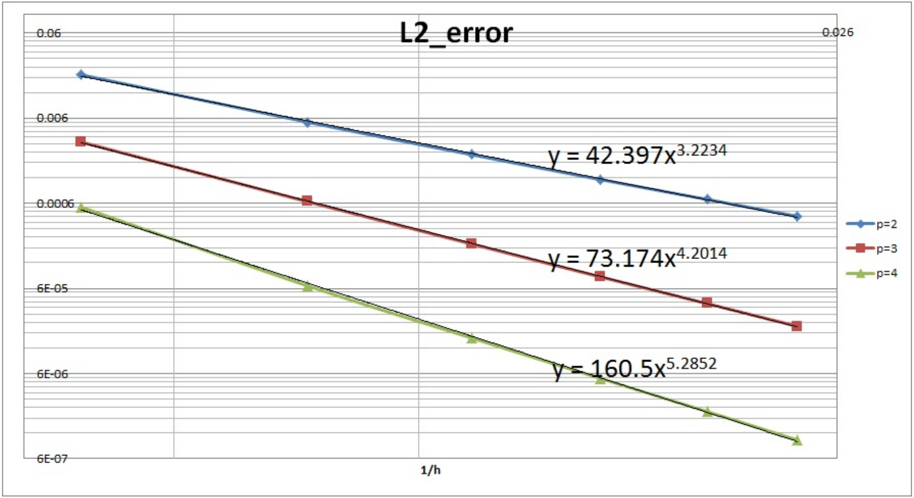
Analytical example, zoom and convergence

The numerical solution:



Analytical example, zoom and convergence

L2-error for different degrees p of the B-splines as a function of the mesh size.



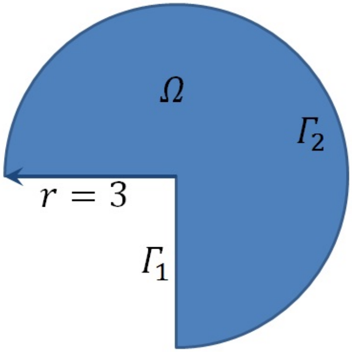
Example of zoom: singularity at the corner

The problem:

$$\triangle u = 0 \text{ in } \Omega,$$

$$u = 0 \text{ on } \Gamma_1 \subset \Gamma_\Omega = \partial\Omega;\ u = \theta(\alpha - \theta) \text{ on } \Gamma_2 = \Gamma_\Omega \backslash \Gamma_1, \alpha = \frac{3\pi}{2},$$

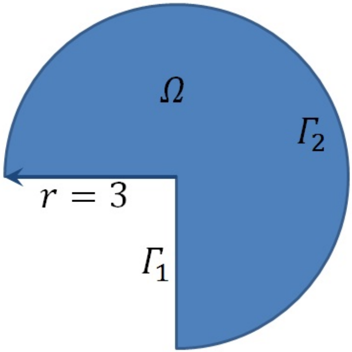
$$\Omega = \{(\rho, \theta) \in \mathbb{R} | 0 \leq \rho \leq 3; -\frac{\pi}{2} \leq \theta \leq \pi\}.$$



Example of zoom: singularity at the corner

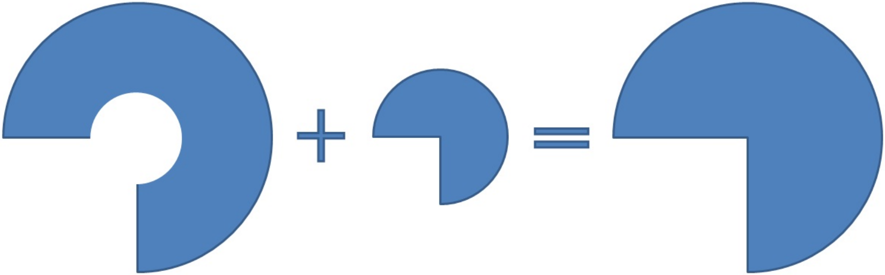
The exact solution is given by the series:

$$u_{ex}(\rho,\theta) = \frac{9}{\pi} \sum_{n=1,3,5,\ldots} \frac{1}{n^3} \left(\frac{\rho}{r}\right)^{\frac{2n}{3}} \sin\left(\frac{2n\theta}{3}\right).$$



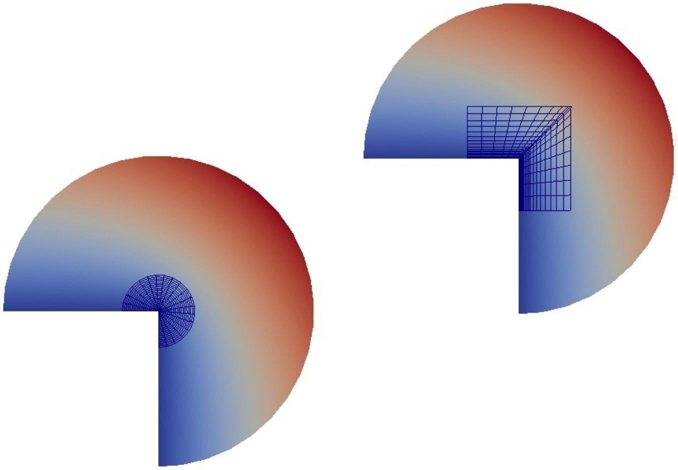
Example of zoom: singularity at the corner

Domain Decomposition: $\Omega = \Omega_{out} \cup \Omega_{zoom}$,



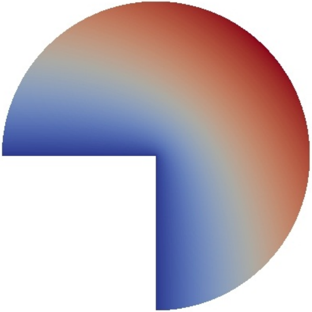
Example of zoom: singularity at the corner

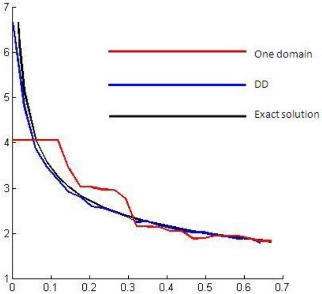
Domain Decomposition: $\Omega = \Omega_{out} \cup \Omega_{zoom}$,



Example of zoom: singularity at the corner

Projection and non-homogeneous Dirichlet BC.
In this example we used the exact projection method and we imposed the Dirichlet boundary conditions by the least-squares method.

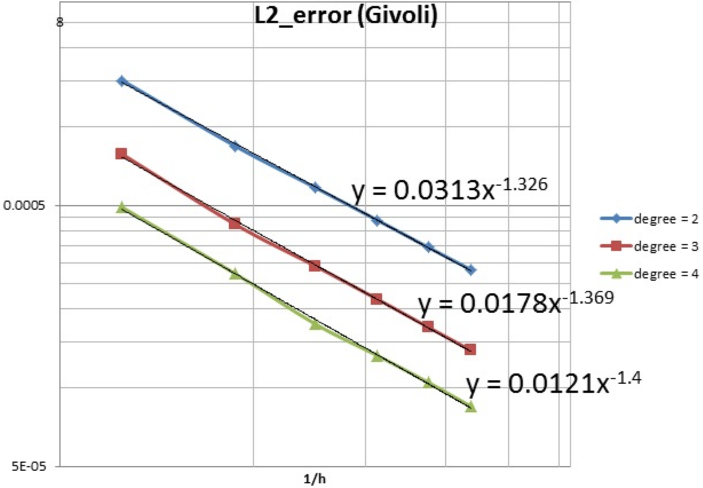




Exact solution is $r^{1/3}$, and our result coincides with this value.

Example of zoom: singularity at the corner

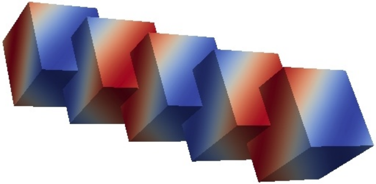
The solution being in $H^{1+\epsilon}(\Omega)$ and not in $H^2(\Omega)$, we get a convergence rate that does not improve with p

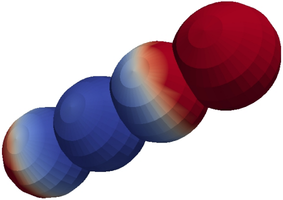


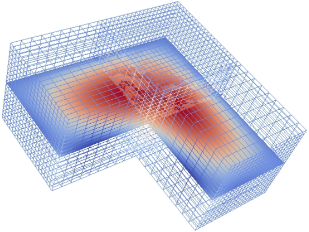
3D examples:

- Heat and elasticity problems
- Parallelisation

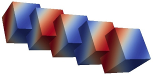
3D examples

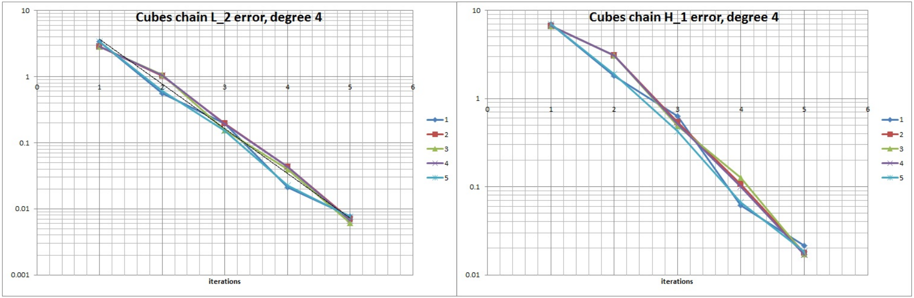






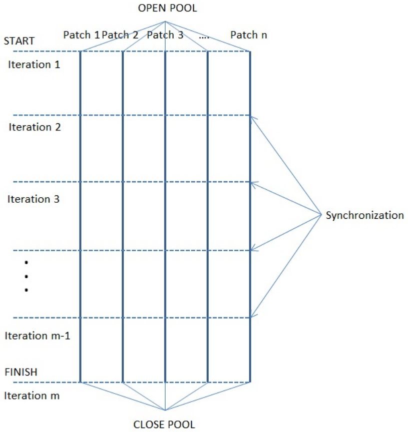
Chain of cubes with the analytical solution $sin(x + y + z)$.





3D and Parallelisation

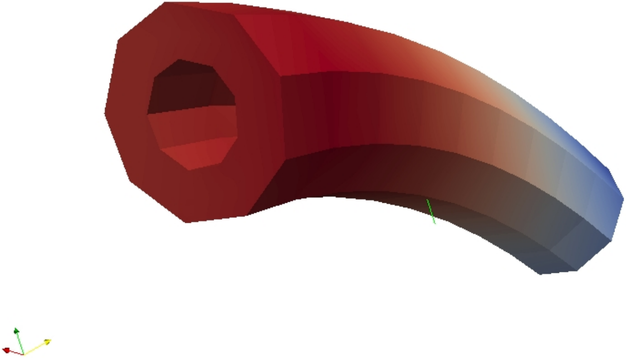
In order to solve "real 3D" examples we implemented a parallel version of the code using MatLab (not easy for shared data).
Our solution:
use unrelated variables to perform the computations on each domain at every iteration and synchronize the solutions between the iterations.



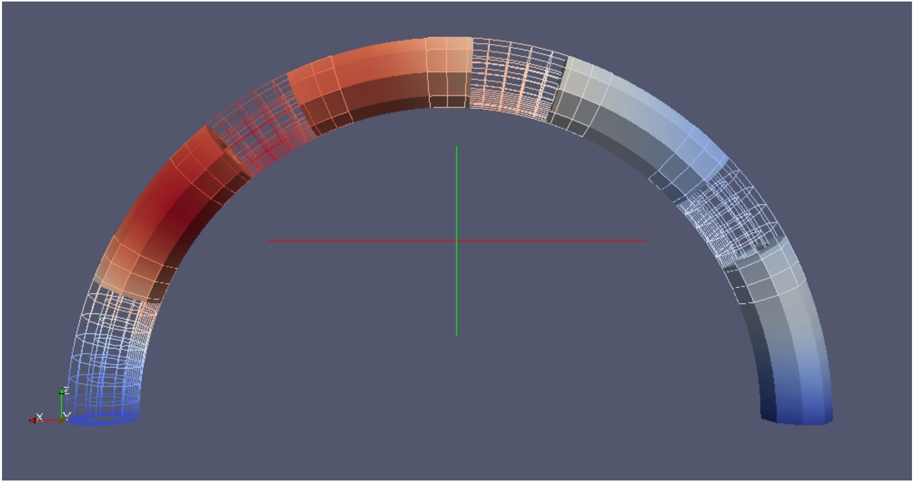
3D and Parallelisation

180^o hollow pipe, defined by 8 overlapping domains, elasticity model under uniform field force.

"Unit" Patch :

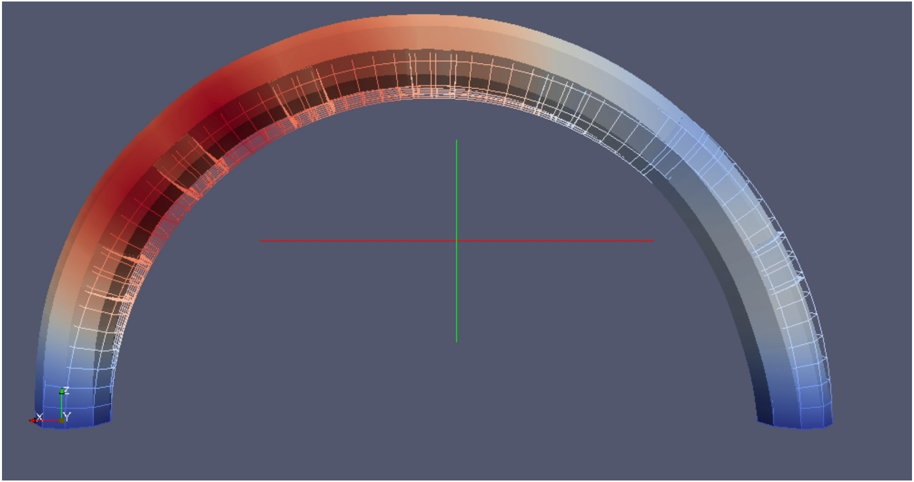


180^o hollow pipe, defined by 8 overlapping domains, elasticity model under uniform field force.



3D and Parallelisation

180^o hollow pipe, defined by 5 overlapping domains, elasticity model under uniform field force.



Final Remarks I

DD on unmatching grids/mapping provides a powerful and natural tool for IGA.

Parallelisation is easy and effective : on a quad core with 8 threads we have an acceleration factor of 4, using GEOPDEs.

To complete it we need to study:

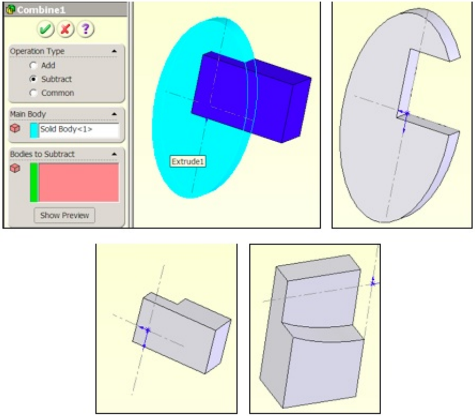
-Dirichlet BC on trimmed surface patches.

-Preconditioning for very large problems.

-Incompressibility etc.

Final Remarks I I

But Boolean operations are not limited of union and intersections, there also subtractions....



Hence we need to extend our method to Chimera type algorithm....

THANKS

Thank you for your attention!