Westfälische Wilhelms-Universität Münster
Applied Mathematics Münster

Interactive Simulations Using Parallel, Event-Driven Localized Reduced Basis

PDESoft 2014 - Heidelberg

Andreas Buhr
Christian Engwer
Mario Ohlberger

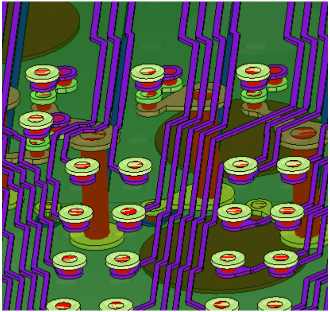
July 15, 2014

Outline

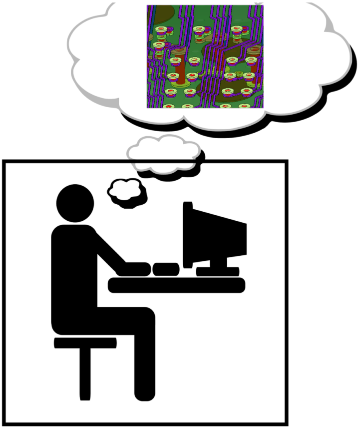
1. Time Harmonic Maxwell's Equations
2. Software Design
3. Model Order Reduction
4. Numerical Example
5. Outlook

Simulating a Chip Carrier in a Flip Chip Package

Simulating a Chip Carrier in a Flip Chip Package

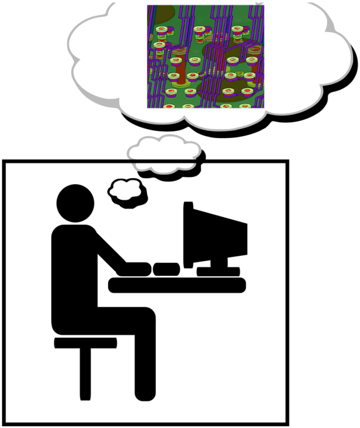


Non-Parametric Geometry Changes



Envision engineer working on design

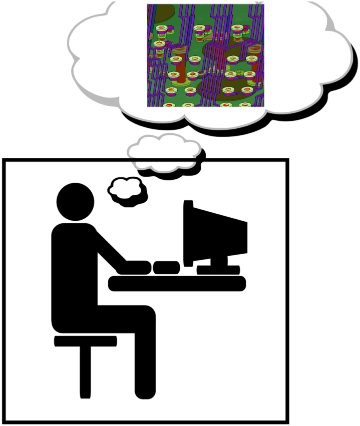
Non-Parametric Geometry Changes



Envision engineer working on design

Multi-query setting

Non-Parametric Geometry Changes



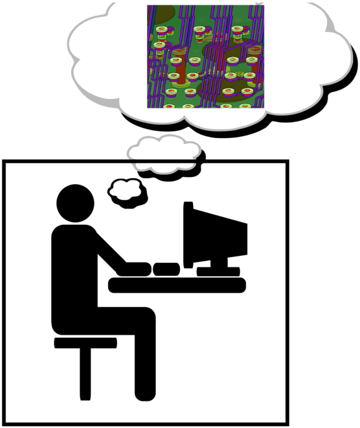
Envision engineer working on design

Multi-query setting

Properties of changes:

1. very localized

Non-Parametric Geometry Changes



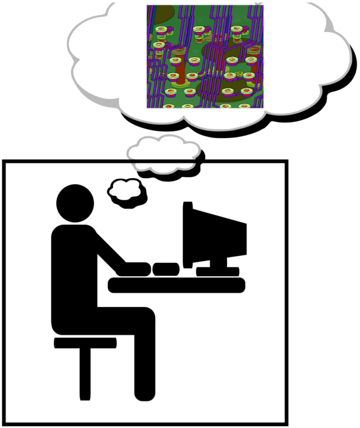
Envision engineer working on design

Multi-query setting

Properties of changes:

1. very localized
2. unforeseen

Non-Parametric Geometry Changes



Envision engineer working on design

Multi-query setting

Properties of changes:

1. very localized
2. unforeseen

Cluster often available.
Cloud always available.

Time Harmonic Maxwell's Equations

$$\nabla \times \frac{1}{\mu} \nabla \times E - \omega^2 \epsilon E = -i\omega j \qquad \text{in} \quad \Omega \tag{1}$$

Time Harmonic Maxwell's Equations

$$\nabla \times \frac{1}{\mu} \nabla \times E - \omega^2 \epsilon E = -i\omega j \qquad \text{in} \quad \Omega \tag{1}$$

- Simulation in a frequency range, e.g.

$$\omega \in [0, 10^{10}]$$

Time Harmonic Maxwell's Equations

$$\nabla \times \frac{1}{\mu} \nabla \times E - \omega^2 \epsilon E = -i\omega j \qquad \text{in} \quad \Omega \tag{1}$$

- Simulation in a frequency range, e.g.

$$\omega \in [0, 10^{10}]$$

- Dirichlet boundary:

$$E \times n = g \quad \text{on } \partial\Omega \quad (= 0 \text{ on most of } \partial\Omega)$$

Part I:

Software Design for Interactive Simulations

Software Design for Interactive Applications

Design of interactive applications is well understood:

- Event driven
- Signal/Slot based (like e.g. Qt / Boost.Signals)

Software Design for Interactive Applications

Design of interactive applications is well understood:

- Event driven
- Signal/Slot based (like e.g. Qt / Boost.Signals)

Technically:

- Slot is a function
- Signal is a list of function objects

Software Design for Interactive Applications

Design of interactive applications is well understood:

- Event driven
- Signal/Slot based (like e.g. Qt / Boost.Signals)

Technically:

- Slot is a function
- Signal is a list of function objects

Agenda

1. Signal/Slot in cluster
2. Implement FEM-Solver in terms of it

Signal/Slot in Cluster

HPX by Ste||ar Group at Louisiana State University[1]

HPX is
“A general purpose C++ runtime system for parallel and distributed applications of any scale”

[1]The STE||AR Group’ (stellar.cct.lsu.edu, github.com/STEllAR-GROUP/hpx)

Signal/Slot in Cluster

HPX by Ste||ar Group at Louisiana State University[1]

HPX is
“A general purpose C++ runtime system for parallel and distributed applications of any scale”

What we need:

- Objects in cluster
- Remote member function calls

[1] The STE||AR Group’ (stellar.cct.lsu.edu, github.com/STEllAR-GROUP/hpx)

Signal/Slot in Cluster

HPX by Ste||ar Group at Louisiana State University[1]

HPX is
“A general purpose C++ runtime system for parallel and distributed applications of any scale”

What we need:

- Objects in cluster
- Remote member function calls

→ We reimplemented what we need in Python.

[1] The STE||AR Group’ (stellar.cct.lsu.edu, github.com/STEllAR-GROUP/hpx)

Publishers and Subscribers

Objects can have publishers. Other objects can subscribe.
This models one-way data transport.

Publishers and Subscribers

Objects can have publishers. Other objects can subscribe.
This models one-way data transport.

Publisher interface:

- invalidate()
- publish(data)

Publishers and Subscribers

Objects can have publishers. Other objects can subscribe.
This models one-way data transport.

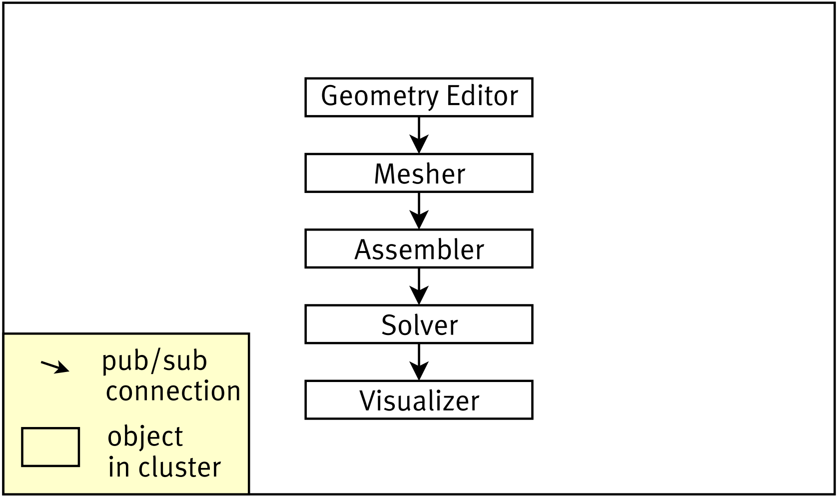
Publisher interface:

- invalidate()
- publish(data)

Subscriber interface:

- Subscriber(publisher_id, validation_callback, invalidation_callback)
- get_data()
- is_valid()

Event-Driven Finite Element Solver (simple)



Add Domain Decomposition

Without Domain Decomposition:

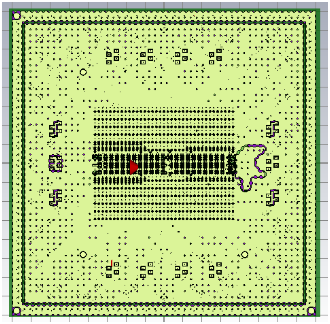
- Little parallelism
- Large amounts of data transferred

Add Domain Decomposition

Without Domain Decomposition:

- Little parallelism
- Large amounts of data transferred

Add Domain Decomposition

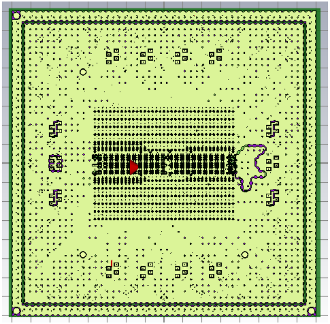


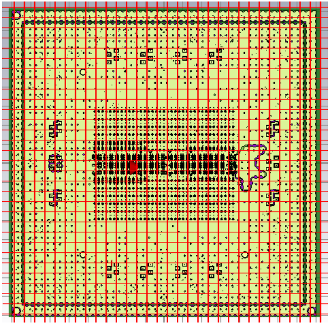
Add Domain Decomposition

Without Domain Decomposition:

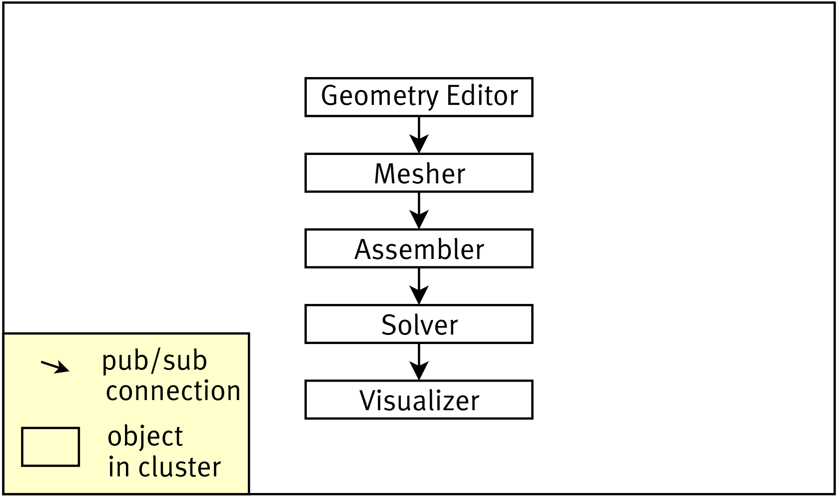
- Little parallelism
- Large amounts of data transferred

Add Domain Decomposition

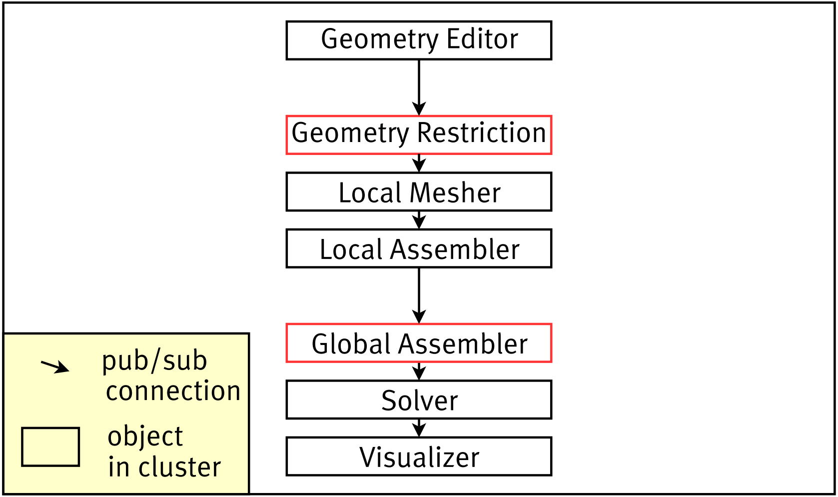




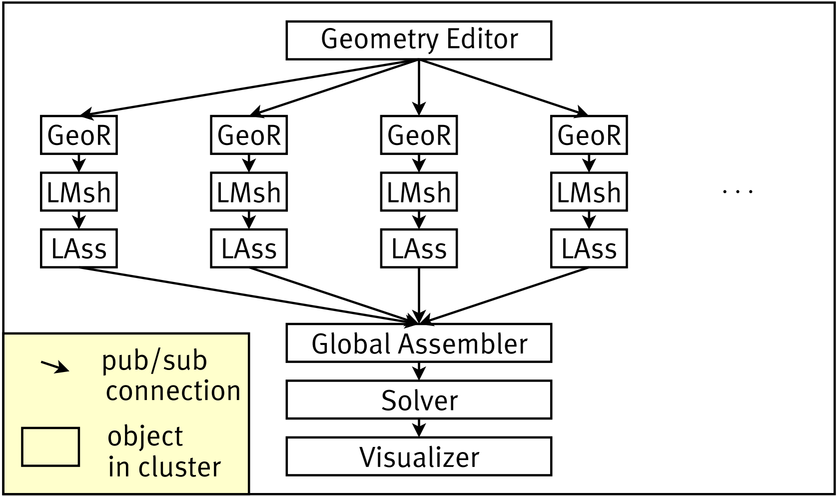
Event Driven Finite Element Solver



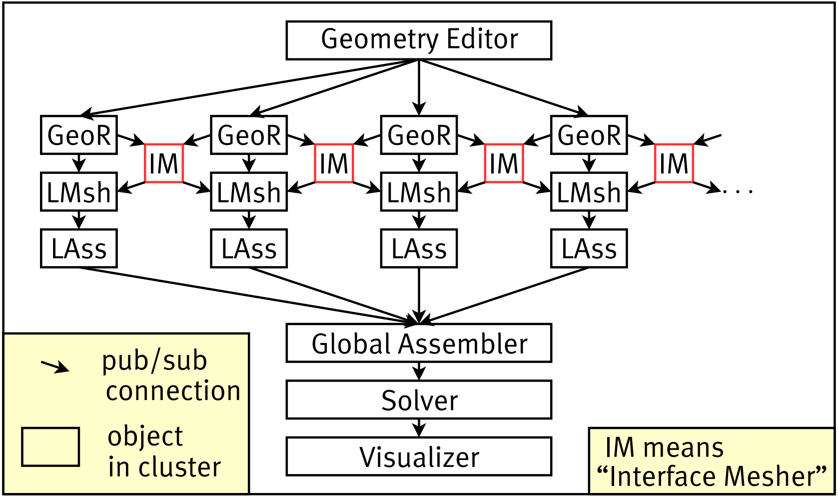
Event Driven Finite Element Solver



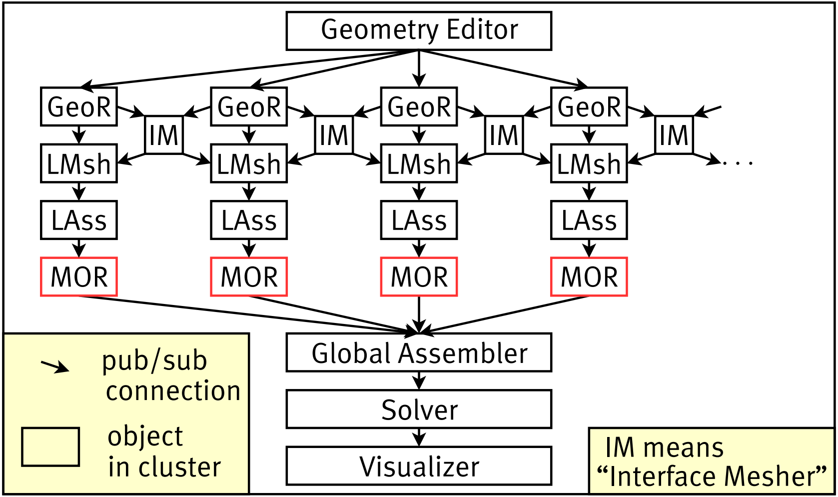
Event Driven Finite Element Solver



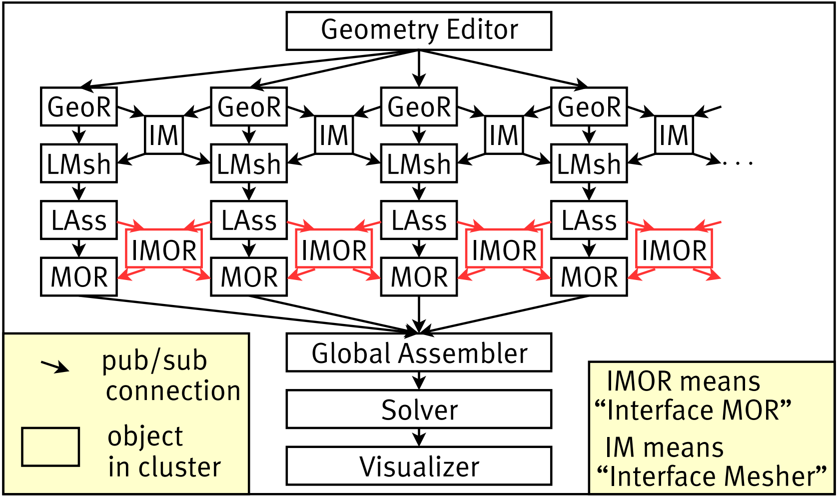
Event Driven Finite Element Solver



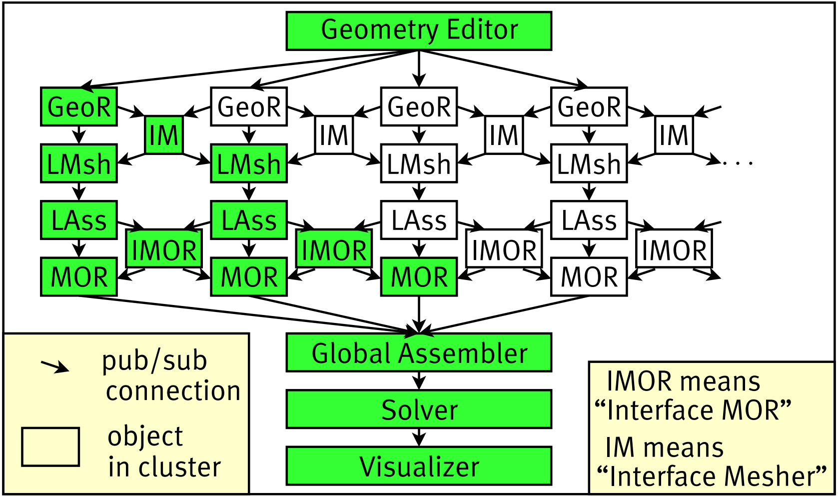
Event Driven Finite Element Solver



Event Driven Finite Element Solver

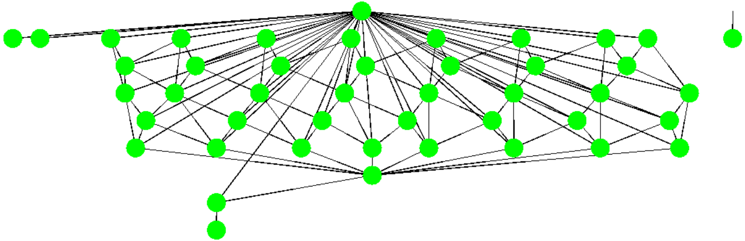


Event Driven Finite Element Solver



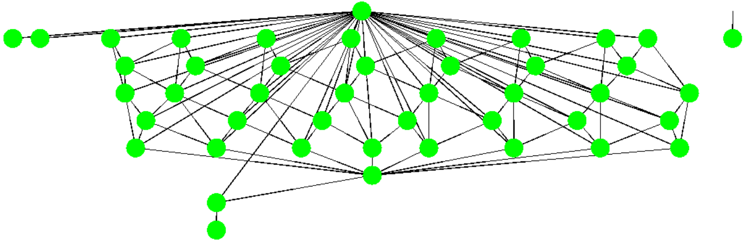
Real World Dependency Graphs

Dependency graph for 8 domains:



Real World Dependency Graphs

Dependency graph for 8 domains:



Dependency graph for 64 domains:

Dependency graph for more domains not shown

Part II:
Generating Subdomain Spaces

Standard Reduced Basis

Problem

Find u_ω in V such that

$$a_\omega(u_\omega, v) = f_\omega(v) \qquad \forall\, v \in V$$

Reduced Basis Approach

Construct subspace $\widetilde{V} \subset V$ with $\dim(\widetilde{V}) \ll \dim(V)$,
find $\widetilde{u}_\omega$ in $\widetilde{V}$:

$$a_\omega(\widetilde{u}_\omega, \widetilde{v}) = f_\omega(\widetilde{v}) \qquad \forall\, \widetilde{v} \in \widetilde{V}$$

Two Main Questions

The two main questions for this approach are:

- How to construct the reduced space $\widetilde{V}$?
- How to control the error $||u_\omega - \widetilde{u}_\omega||_V$?

For standard (not localized) RB, rich theory exists.

Localization by Grouping of Ansatzfunctions

Decomposition of Ansatz Space

The space V is the direct sum of subspaces:

$$V = \Big(\bigoplus_i V_{D_i}\Big) \oplus \Big(\bigoplus_j V_{I_j}\Big) \qquad V_{D_i} \subset V, \qquad V_{I_j} \subset V \tag{2}$$

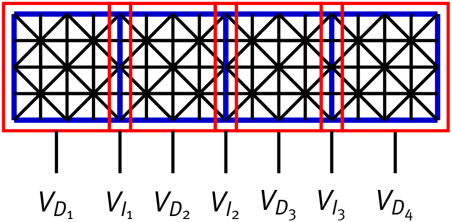
Localization by Grouping of Ansatzfunctions

Decomposition of Ansatz Space

The space V is the direct sum of subspaces:

$$V = \left(\bigoplus_i V_{D_i}\right) \oplus \left(\bigoplus_j V_{I_j}\right) \qquad V_{D_i} \subset V, \qquad V_{I_j} \subset V \tag{2}$$

Four Domain Example:



Localization by Grouping of Ansatzfunctions

Decomposition of Ansatz Space

The space V is the direct sum of subspaces:

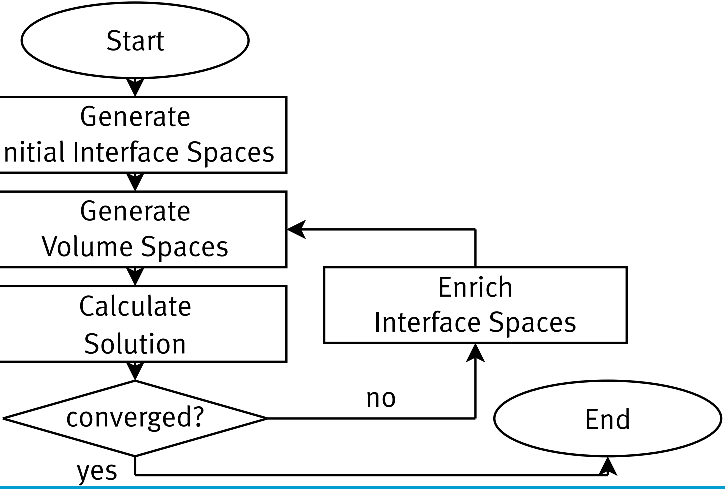
$$V = \Big(\bigoplus_i V_{D_i}\Big) \oplus \Big(\bigoplus_j V_{I_j}\Big) \qquad V_{D_i} \subset V, \qquad V_{I_j} \subset V \tag{3}$$

Localized Reduced Basis

For each V_{D_i} and V_{I_j} construct subspaces $\widetilde{V}_{D_i} \subset V_{D_i}$ and $\widetilde{V}_{I_j} \subset V_{I_j}$ with $\dim(\widetilde{V}_{D_i}) \ll \dim(V_{D_i})$ and $\dim(\widetilde{V}_{I_j}) \ll \dim(V_{I_j})$, find $\widetilde{u}_\omega$ in $\widetilde{V}_{LRB} := \big(\bigoplus_i \widetilde{V}_{D_i}\big) \oplus \big(\bigoplus_j \widetilde{V}_{I_j}\big) \subset V$:

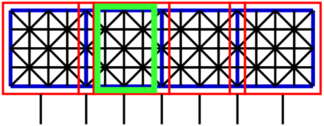
$$a_\omega(\widetilde{u}_\omega, \widetilde{v}) = f_\omega(\widetilde{v}) \qquad \forall\, \widetilde{v} \in \widetilde{V}_{LRB}$$

Enrichment Algorithm



Construction of Volume Subspaces

- easy, if interface subspaces are available
- construct space containing solutions for all interface conditions



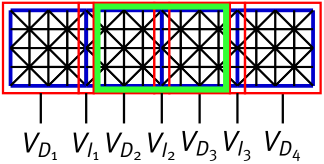
Construction of volume subspaces

E.g. for V_{D_2}:

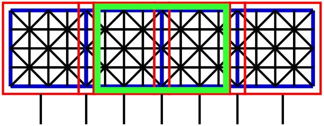
$$\widetilde{V}_{D_2} := \operatorname{span}(\{\psi \in V_{D_2},\ a_\omega(\psi + \varphi_1 + \varphi_2, v) = f_\omega(v) \quad \forall\, v \in V_{D_2},\ \varphi_1 \in \widetilde{V}_{I_1}, \varphi_2 \in \widetilde{V}_{I_2}\})$$

Construction of Interface Subspaces

- research topic
- right now: interface basis enriched by solving patch problems
- steered by localized error indicator



Construction of Interface Subspaces



- research topic
- right now: interface basis enriched by solving patch problems
- steered by localized error indicator

Projection Operator

P_{D_i}, P_{I_j} is projection to V_{D_i}, V_{I_j}:

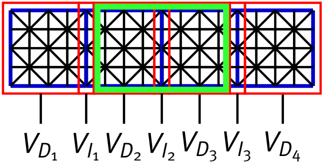
$$P_{D_i} : V \to V_{D_i} \qquad P_{I_j} : V \to V_{I_j}$$

defined by

$$v = \sum_i P_{D_i}(v) + \sum_j P_{I_j}(v) \quad \forall v \in V$$

Construction of Interface Subspaces

- research topic
- right now: interface basis enriched by solving patch problems
- steered by localized error indicator



Construction of interface subspaces

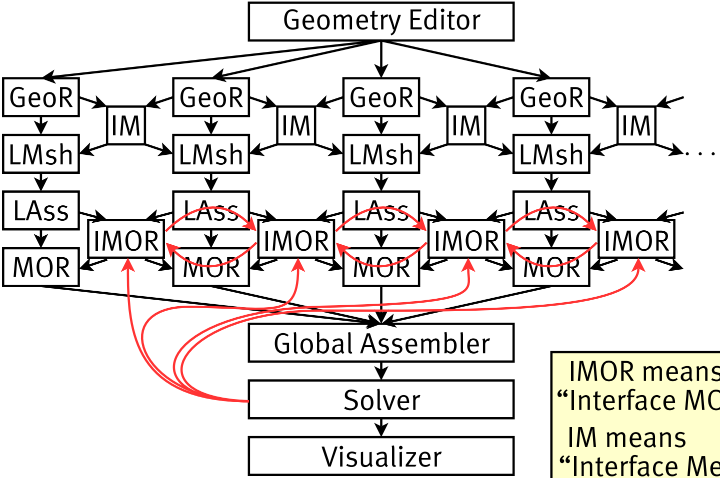
E.g. for V_{I_2}, solve in $V_{\text{patch},I_2} := V_{D_2} \oplus V_{I_2} \oplus V_{D_3}$

$$\widetilde{V}_{I_2,i+1} := \widetilde{V}_{I_2,i} \oplus \text{span}(P_{I_2}(\psi))$$

where ψ in V_{patch,I_2} is the solution of

$$a_\omega(\psi + P_{I_1}(\widetilde{u}_{\omega,i}) + P_{I_3}(\widetilde{u}_{\omega,i}), v) = f_\omega(v) \quad \forall\, v \in V_{\text{patch},I_2}$$

Additional Data Dependencies

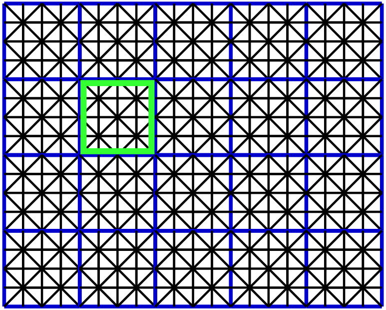


IMOR means “Interface MOR”

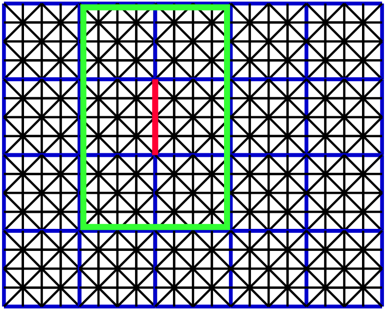
IM means “Interface Mesher”

Patches in 2D

for volume subspace



for interface subspace



Similar approaches in Literature

LRBMS, GMsFEM, PR-SCRBE:

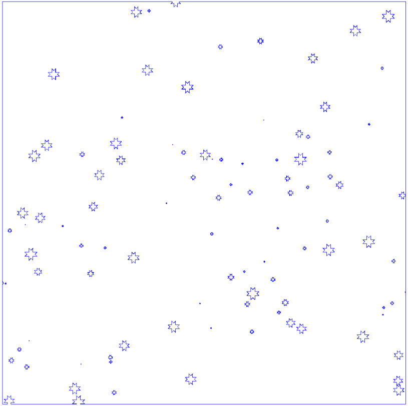
F. Albrecht, B. Haasdonk, M. Ohlberger, and S. Kaulmann.
The localized reduced basis multiscale method.
Proceedings of Algoritmy 2012, Conference on Scientific Computing, Vysoke Tatry, Podbanske, September 9-14, 2012, pages 393–403, 2012.

Yalchin Efendiev, Juan Galvis, and Thomas Y. Hou.
Generalized multiscale finite element methods (gmsfem).
January 2013.

Jens L Eftang and Anthony T Patera.
A port-reduced static condensation reduced basis element method for large component-synthesized structures: approximation and a posteriori error estimation.
Advanced Modeling and Simulation in Engineering Sciences, 1(1):3, 2014.

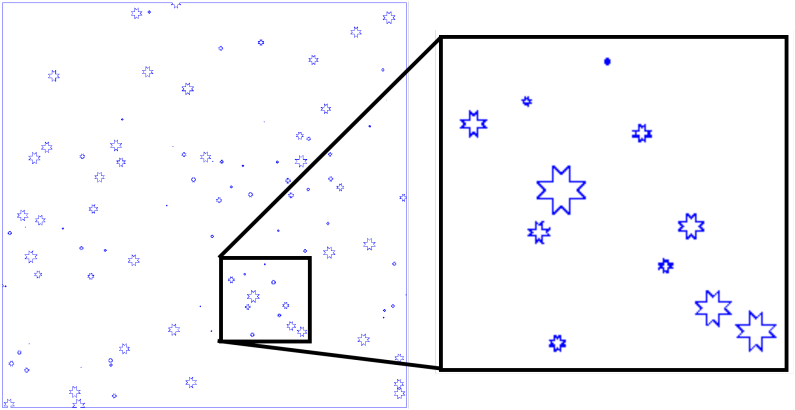
Numerical Example

test geometry: 2D metal box with random metal stars



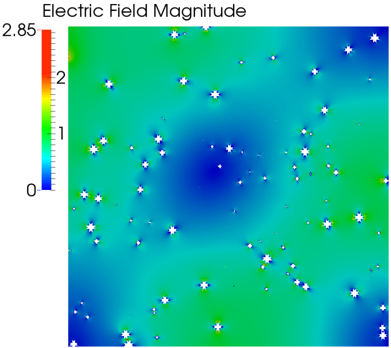
Numerical Example

test geometry: 2D metal box with random metal stars

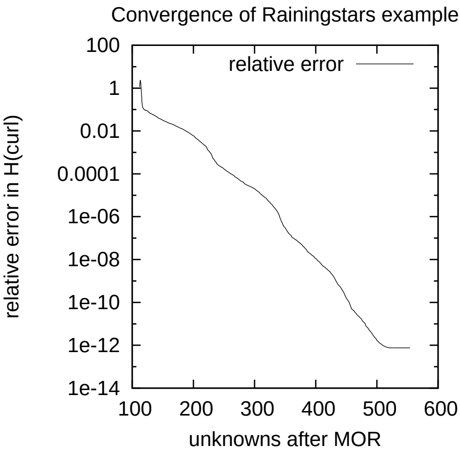


Numerical Example

test geometry: 2D metal box with random metal stars



Convergence with Iterative Enrichment



- 183 219 unknowns in full system
- 8x8 domain decomposition
- 112 internal interfaces
- 1e-5 at 313 dofs
- 1e-10 at 455 dofs

Outlook

We are planning to:

- implement rigorous a-posteriori error estimates[2]
- go for 3D
- evaluate more strategies for interface reduction

[2] K Smetana, A new certification framework for the port reduced static condensation reduced basis element method. Computer Methods in Applied Mechanics and Engineering

Acknowledgements

Many thanks to...

CST - Computer Simulation Technology AG[3]

... for sponsoring my research.

[3] www.cst.com