



WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER



APPLIED
MATHEMATICS
MÜNSTER

Interactive Simulations Using the Localized Reduced Basis Method

Reduced Basis Summer School 2014 - Münster



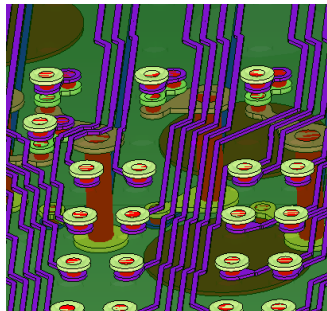
Outline

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

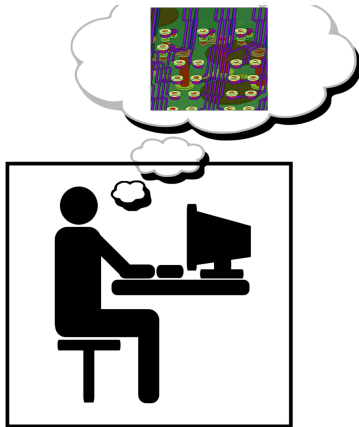
Simulating a Chip Carrier in a Flip Chip Package



Simulating a Chip Carrier in a Flip Chip Package

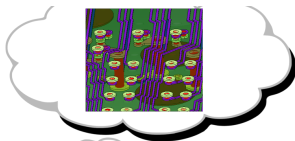


Non-Parametric Geometry Changes



Envision engineer working on design

Non-Parametric Geometry Changes

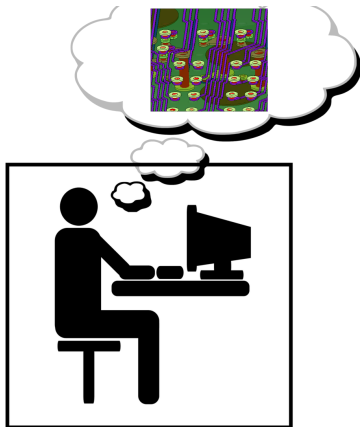


Envision engineer working on design

Multi-query setting



Non-Parametric Geometry Changes



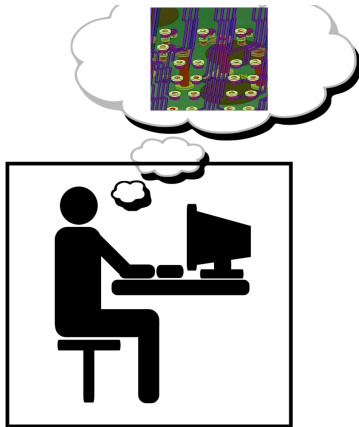
Envision engineer working on design

Multi-query setting

Properties of changes:

1. very localized

Non-Parametric Geometry Changes



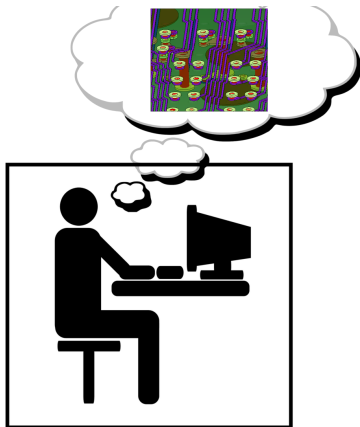
Envision engineer working on design

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

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

Time Harmonic Maxwell's Equations

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

- Simulation in a frequency range, e.g.

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

Time Harmonic Maxwell's Equations

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

- ▶ 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 University¹



HPX is

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

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

Signal/Slot in Cluster

HPX by Ste||ar Group at Louisiana State University¹



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

¹The STE||AR Group' (stellar.cct.lsu.edu, github.com/STELLAR-GROUP/hpx)

Signal/Slot in Cluster

HPX by Ste||ar Group at Louisiana State University¹



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.

¹The STE||AR Group' (stellar.cct.lsu.edu, github.com/STELLAR-GROUP/hpx)

What I did, call it pyX

I implemented my own parallel runtime system, inspired by HPX, but much simpler, using Python3 and the networking library ZeroMQ.

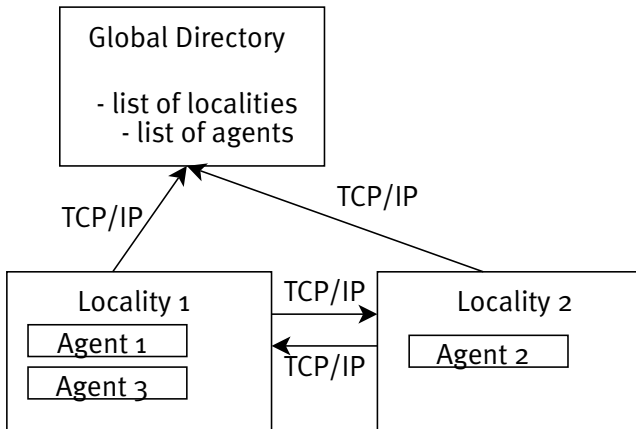


Remote Member Function Calls

```
cluster_client.call_method(priority,agent_number,function,arguments)
```

The caller does not need to know where the recipient is.

Remote Member Function Calls



Publishers and Subscribers

Agents can have publishers. Other agents can subscribe.
This models one-way data transport.

Publishers and Subscribers

Agents can have publishers. Other agents can subscribe.
This models one-way data transport.

Publisher interface:

- ▶ `invalidate()`
- ▶ `publish(data)`

Publishers and Subscribers

Agents can have publishers. Other agents 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()`

pyX - Example

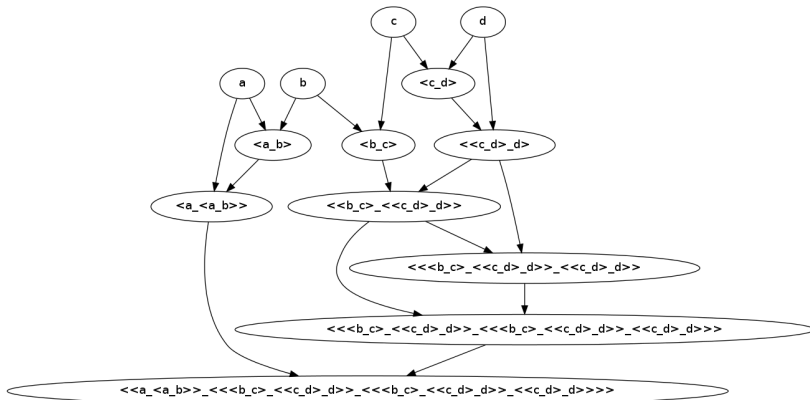
Let's consider an extremely simple toy problem:

We calculate from the four values a, b, c, d the expression

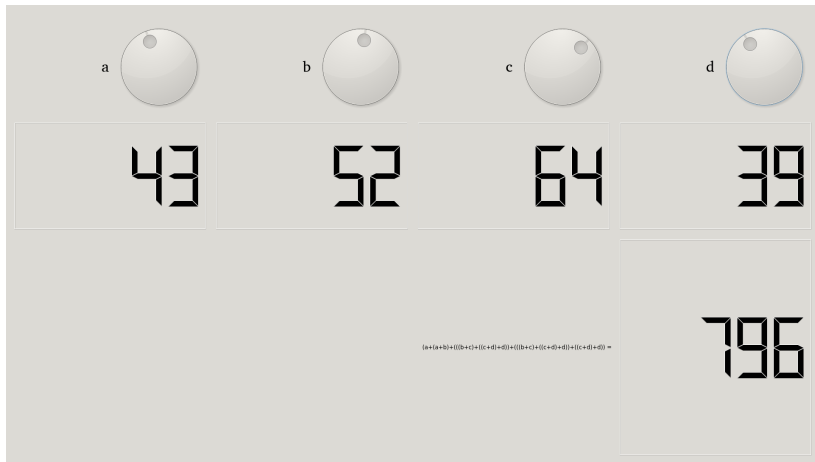
$$((a+(a+b))+(((b+c)+((c+d)+d))+(((b+c)+((c+d)+d))+((c+d)+d))))$$

Now imagine $+$ is an expensive operation, taking a second. We create an “AdderAgent”, which subscribes to two values and publishes a third one.

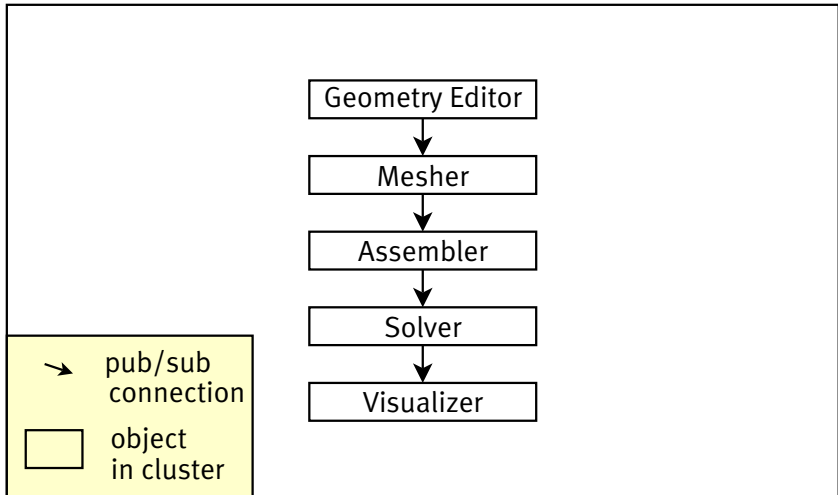
Dependency Graph



Live Demo



Event-Driven Finite Element Solver (simple)



Add Domain Decomposition

Without Domain Decomposition:

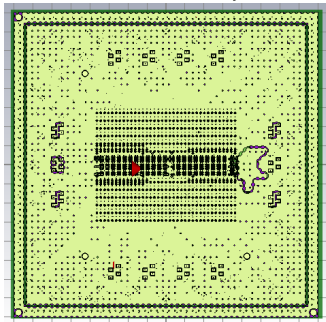
- ▶ Little parallelism
- ▶ Large amounts of data transferred

Add Domain Decomposition

Without Domain Decomposition:

- ▶ Little parallelism
- ▶ Large amounts of data transferred

Add Domain Decomposition

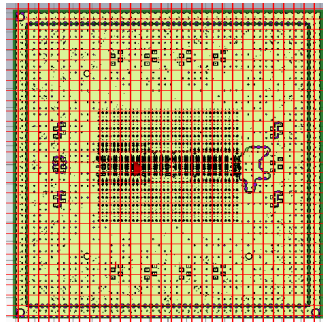
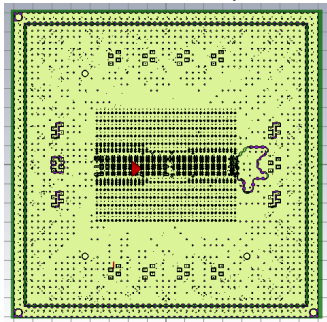


Add Domain Decomposition

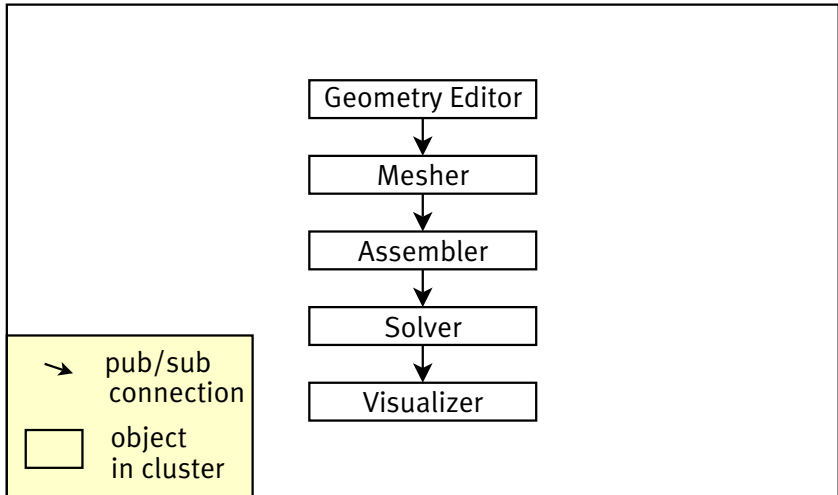
Without Domain Decomposition:

- ▶ Little parallelism
- ▶ Large amounts of data transferred

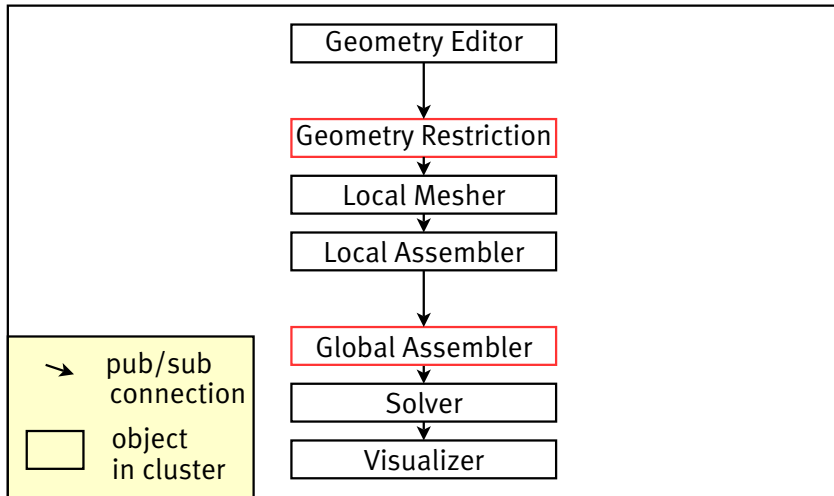
Add Domain Decomposition



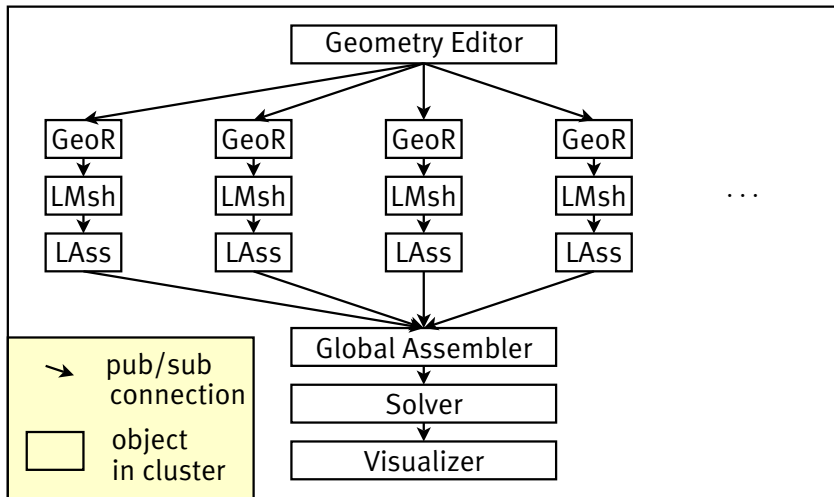
Event Driven Finite Element Solver



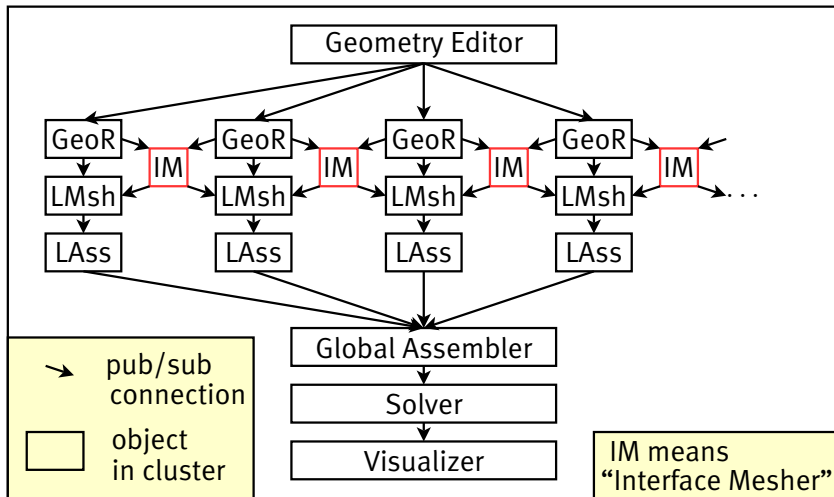
Event Driven Finite Element Solver



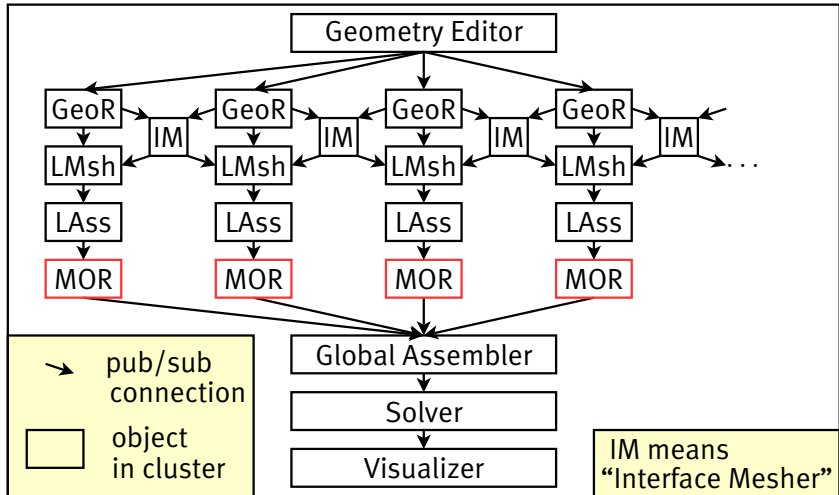
Event Driven Finite Element Solver



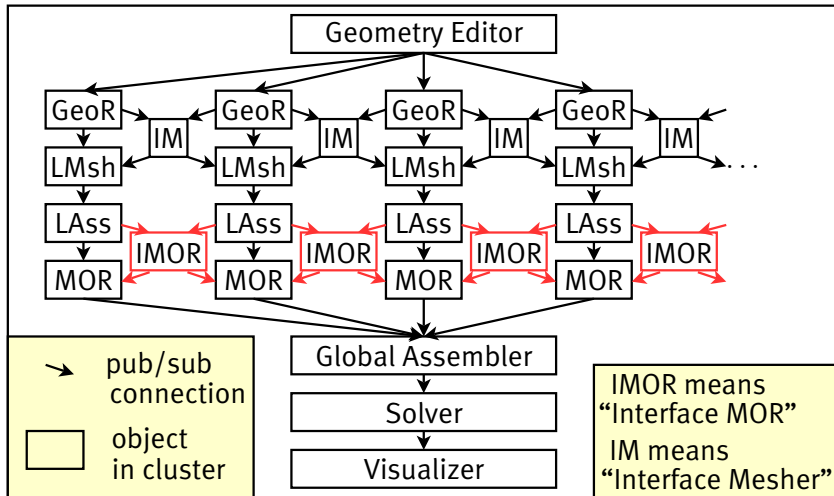
Event Driven Finite Element Solver



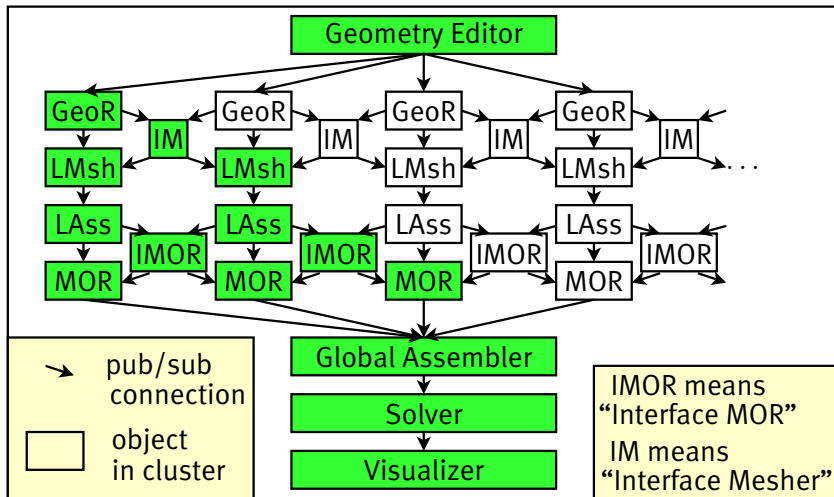
Event Driven Finite Element Solver



Event Driven Finite Element Solver

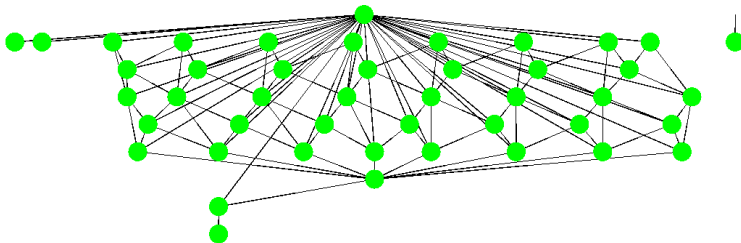


Event Driven Finite Element Solver



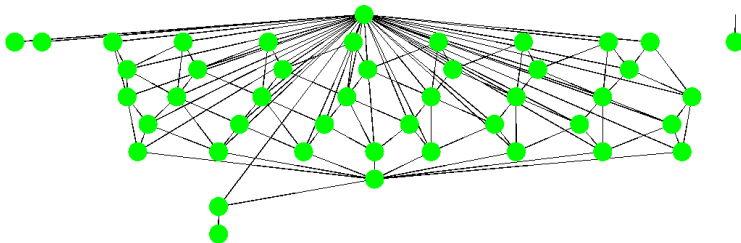
Real World Dependency Graphs

Dependency graph for 8 domains:



Real World Dependency Graphs

Dependency graph for 8 domains:



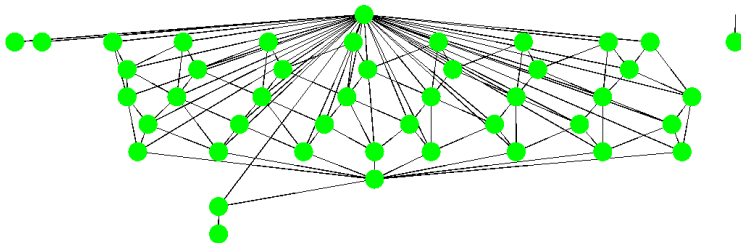
Dependency graph for 64 domains:



Dependency graph for more domains not shown

Real World Dependency Graphs

Dependency graph for 8 domains:



Dependency graph for 64 domains:



Dependency graph for more domains not shown → Live-Demo



Part II: Generating Subdomain Spaces

Standard Reduced Basis

Problem

Find u_ω in V such that

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

Reduced Basis Approach

Construct subspace $\tilde{V} \subset V$ with $\dim(\tilde{V}) \ll \dim(V)$,
find \tilde{u}_ω in \tilde{V} :

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

Two Main Questions

The two main questions for this approach are:

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

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:

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

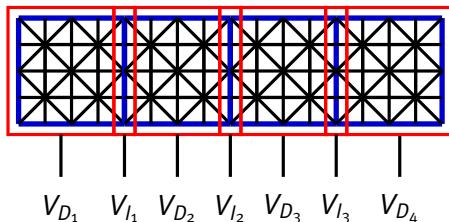
Localization by Grouping of Ansatzfunctions

Decomposition of Ansatz Space

The space V is the direct sum of subspaces:

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

Four Domain Example:



Localization by Grouping of Ansatzfunctions

Decomposition of Ansatz Space

The space V is the direct sum of subspaces:

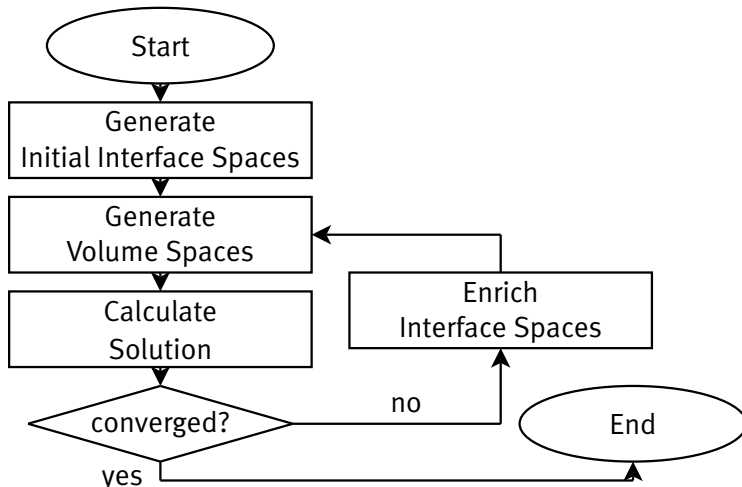
$$V = \left(\bigoplus_i V_{D_i} \right) \oplus \left(\bigoplus_j V_{I_j} \right) \quad V_{D_i} \subset V, \quad V_{I_j} \subset V \quad (3)$$

Localized Reduced Basis

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

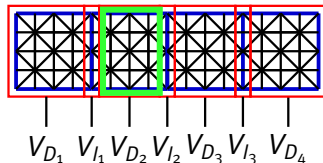
$$a_\omega(\tilde{u}_\omega, \tilde{v}) = f_\omega(\tilde{v}) \quad \forall \tilde{v} \in \tilde{V}_{LRB}$$

Enrichment Algorithm



Construction of Volume Subspaces

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



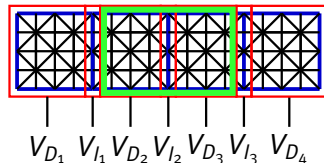
Construction of volume subspaces

E.g. for V_{D_2} :

$$\begin{aligned} \tilde{V}_{D_2} := \text{span}(\{ & \psi \in V_{D_2}, \\ & a_\omega(\psi + \varphi_1 + \varphi_2, v) = f_\omega(v) \quad \forall v \in V_{D_2}, \\ & \varphi_1 \in \tilde{V}_{I_1}, \varphi_2 \in \tilde{V}_{I_2} \}) \end{aligned}$$

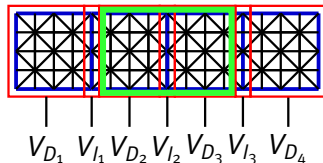
Construction of Interface Subspaces

- ▶ 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} :

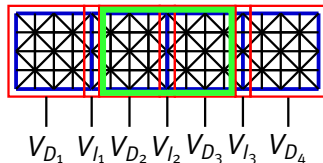
$$P_{D_i} : V \rightarrow V_{D_i} \quad P_{I_j} : V \rightarrow V_{I_j}$$

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

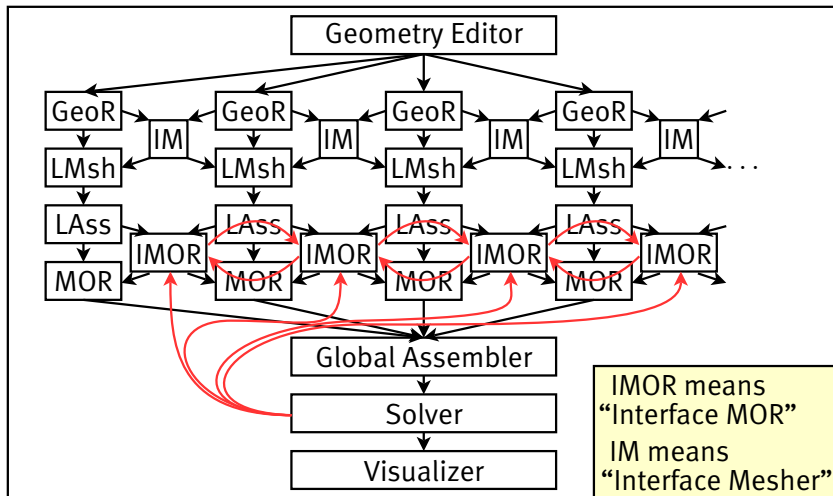
E.g. for V_{I_2} , solve in $V_{\text{patch}, I_2} := V_{D_2} \oplus V_{I_2} \oplus V_{D_3}$

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

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

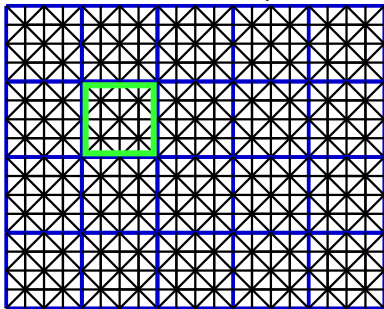
$$a_\omega(\psi + P_{I_1}(\tilde{u}_{\omega, i}) + P_{I_3}(\tilde{u}_{\omega, i}), v) = f_\omega(v) \quad \forall v \in V_{\text{patch}, I_2}$$

Additional Data Dependencies

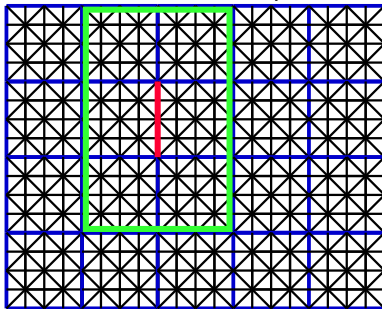


Patches in 2D

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.

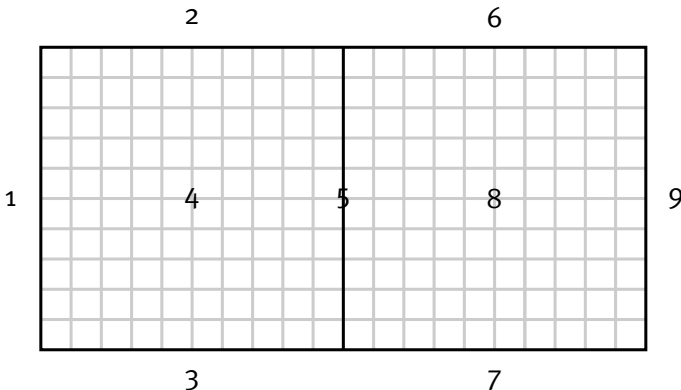
A port-reduced static condensation reduced basis element method for large component-synthesized structures: approximation and a posteriori error estimation.

Advanced Modeling and Simulation in Engineering Sciences, 1(1):3, 2014.

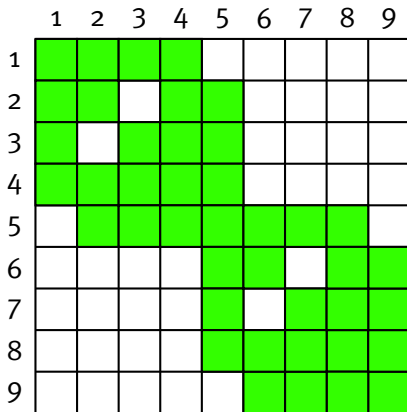
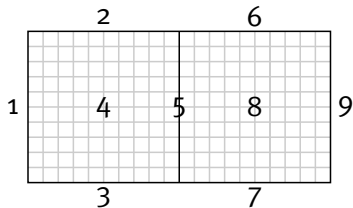
Restrictable Operators

- ▶ You have to restrict your (global) operators to local patches.
- ▶ Save operator as matrix of matrices to easily build patch operators.

DOF Classification for Two Domain Example

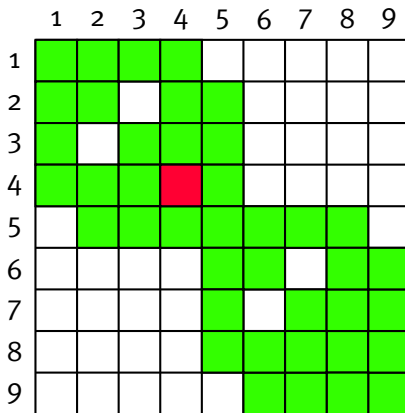
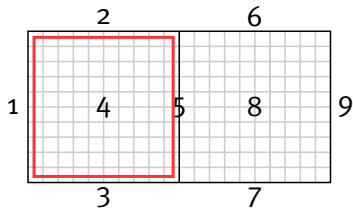


Structure of Sparse Matrix

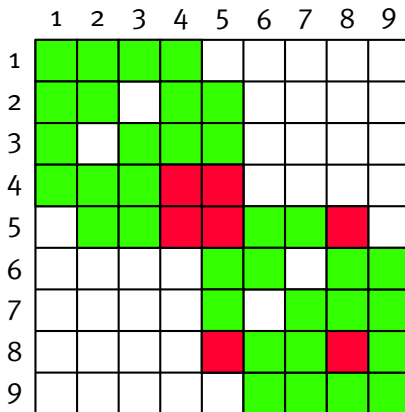
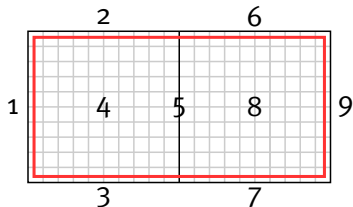


Save it as sparse matrix of sparse matrices.

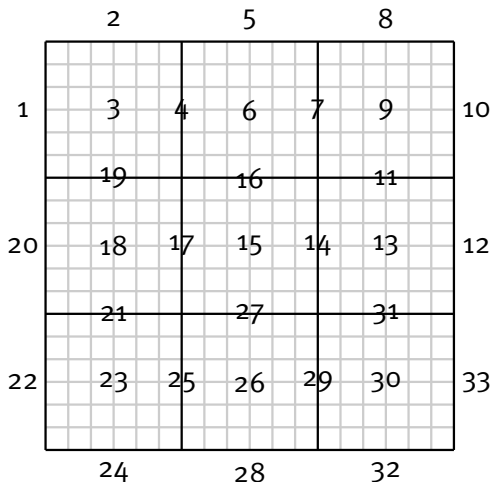
Restriction to Cell 4 with Dirichlet = 0 Boundary



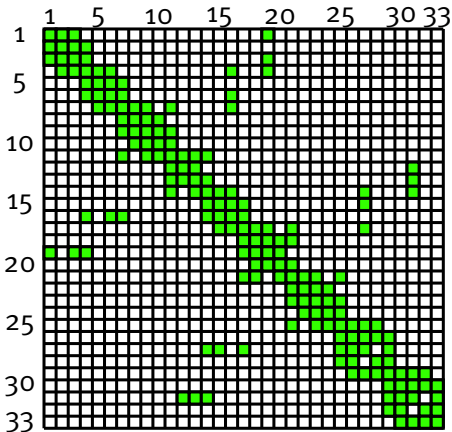
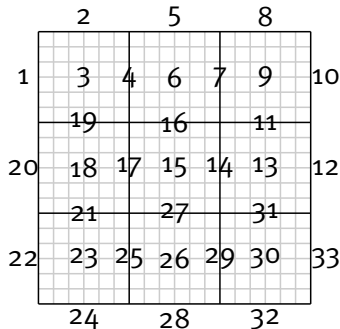
Restriction to Cell 4 & 8 with Dirichlet = 0 Boundary



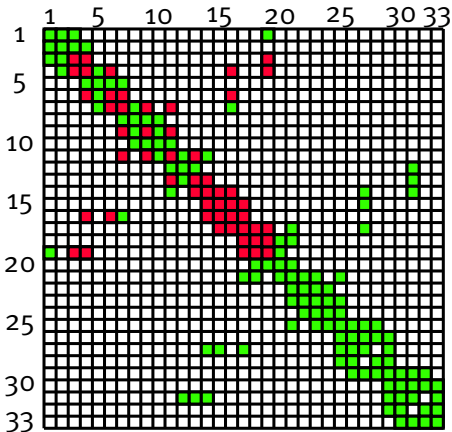
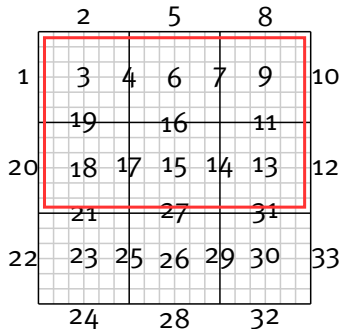
A More Interesting Example



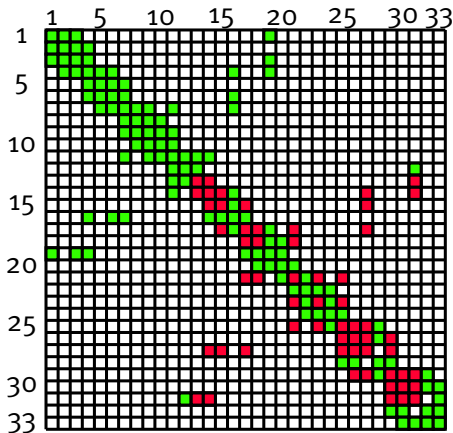
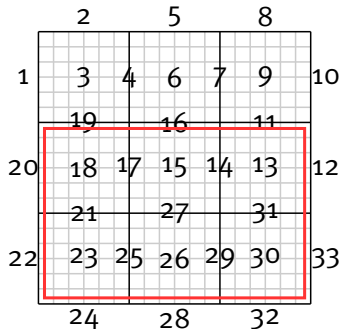
A More Interesting Example



A More Interesting Example

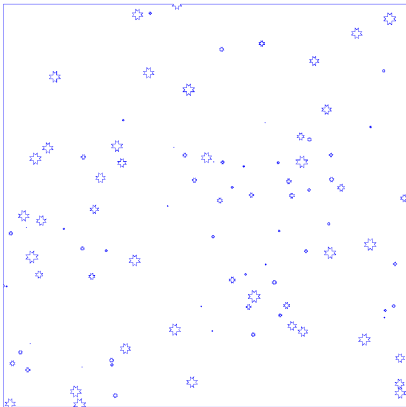


A More Interesting Example



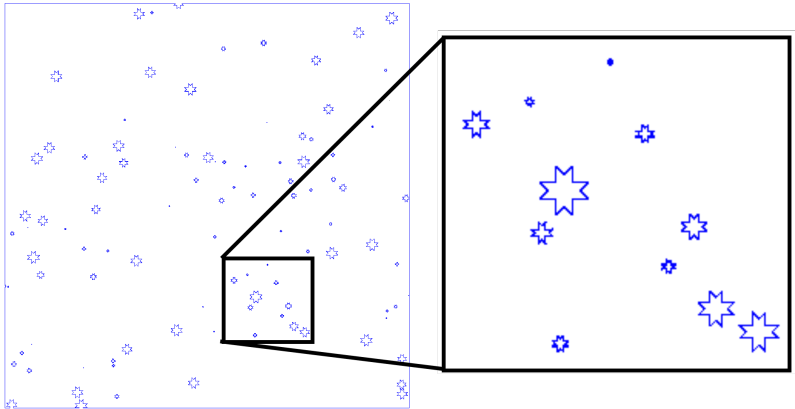
Numerical Example

test geometry: 2D metal box with random metal stars



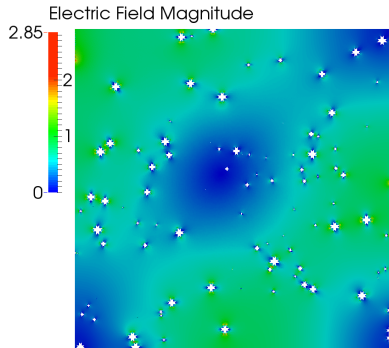
Numerical Example

test geometry: 2D metal box with random metal stars



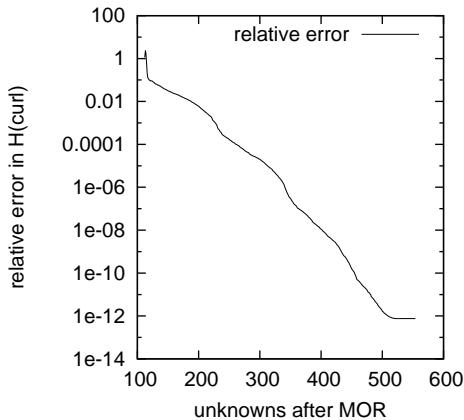
Numerical Example

test geometry: 2D metal box with random metal stars



Convergence with Iterative Enrichment

Convergence of Rainingstars example



- ▶ 183 219 unknowns in full system
- ▶ 8x8 domain decomposition
- ▶ 112 internal interfaces
- ▶ 10^{-5} at 313 dofs
- ▶ 10^{-10} at 455 dofs

Summary

- ▶ A parallel and event-driven runtime system was implemented

Summary

- ▶ A parallel and event-driven runtime system was implemented
- ▶ A localized Reduced Basis method was implemented within that system.

Summary

- ▶ A parallel and event-driven runtime system was implemented
- ▶ A localized Reduced Basis method was implemented within that system.
- ▶ Operators are saved as sparse matrices of sparse matrices to ease forming of patch operators.

Summary

- ▶ A parallel and event-driven runtime system was implemented
- ▶ A localized Reduced Basis method was implemented within that system.
- ▶ Operators are saved as sparse matrices of sparse matrices to ease forming of patch operators.
- ▶ The localized basis enrichment tested yielded exponential convergence.

Outlook

We are planning to:

- ▶ implement rigorous a-posteriori error estimates²
- ▶ go for 3D
- ▶ evaluate more strategies for interface reduction

²K 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 AG³

... for sponsoring my research.

³www.cst.com