Méthodes de bases réduite pour la simulation d'EDP paramétrées : résultats de base, interpolation empirique, reconstruction par fouille de données

Yvon Maday

Laboratoire Jacques-Louis Lions - UPMC, Paris, France
IUF
Division of Applied Maths Brown University, Providence USA

Mathématiques et Grandes Dimensions

Introduction – models.

Partial differential equations

P.D.E.

These are relations — possibly nonlinear — between an unknown function and its various derivatives.

Some examples

Introduction – models.

Partial differential equations

P.D.E.
These are relations — possibly nonlinear — between an unknown function and its various derivatives.

Some examples

Introduction – models.

Partial differential equations

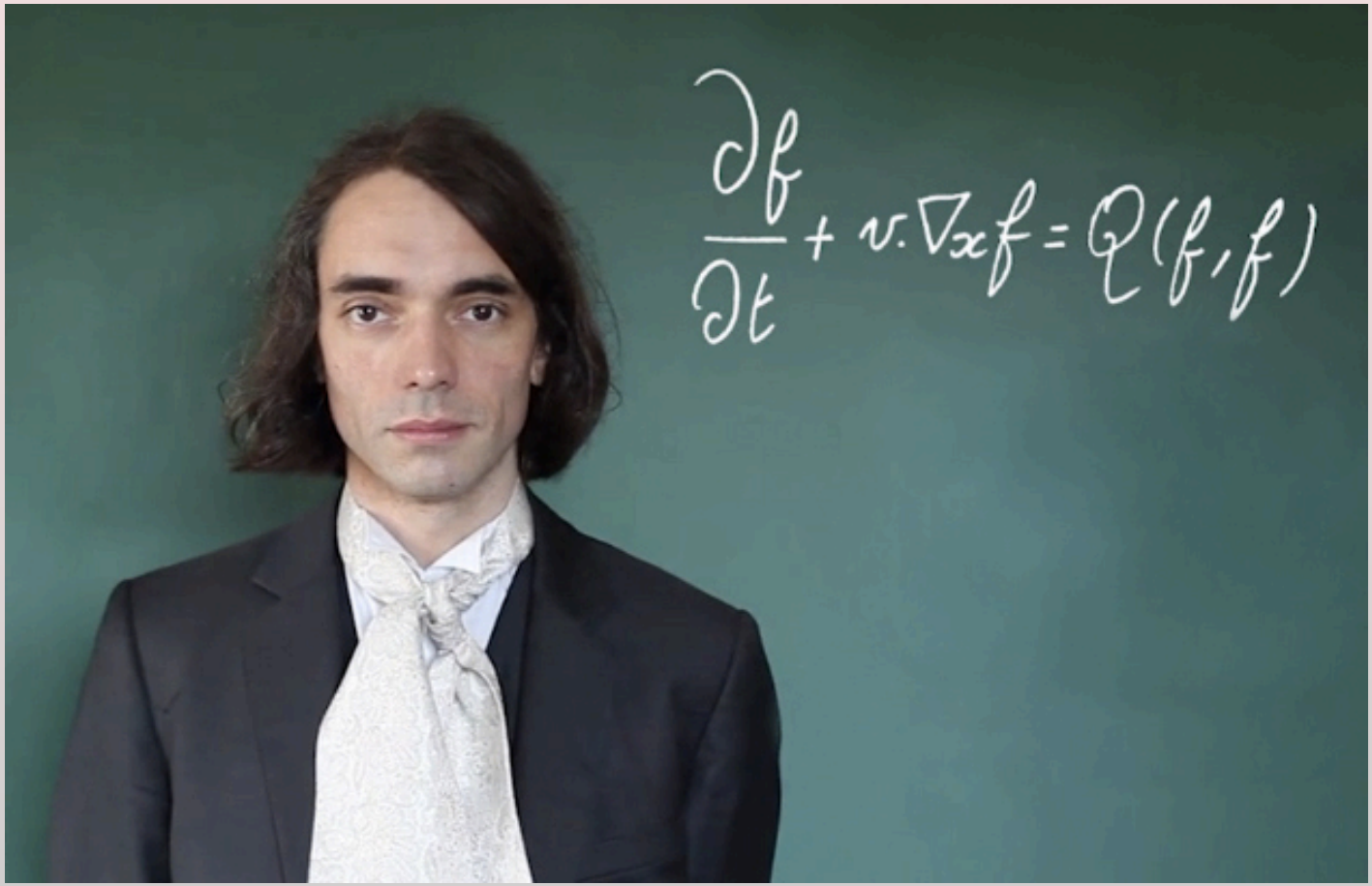
P.D.E.
These are relations — possibly nonlinear — between an unknown function and its various derivatives.

Some examples

Introduction – models.

Partial differential equations

Boltzmann equations



Introduction – models.

Partial differential equations

Navier-Stokes

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one ! !

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

Without ignoring the discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one ! !

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

Without ignoring the discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one!!

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

without getting the discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one!!

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

Without ignoring the discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one!!

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

Without ignoring the discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one!!

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

Without getting into discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one!!

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

Without getting into discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one!!

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

Without ignoring the discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one!!

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

without quoting the discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one!!

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

without quoting the discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one!!

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

without quoting the discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

Discretization methods for approximating the solution to PDE's are various, if you have one yourself, certainly yours is the best one!!

This can be

- Finite Element Method
- Finite Difference Method
- Finite Volume Methods
- Spectral Method
- Wavelets Method
- Mimetic Finite Difference Method
- Meshless Method

without quoting the discretization in time

all being interesting for some particular application

Introduction – discretization.

One shot approximation

As long as you will have to do a numerical simulation of a PDE, these methods (and of course new ones) will be useful

Introduction – discretization.

Multiple shots approximation.

But if you need to solve many times the same type of problem, there is room for invention ! !

Of course you are not solving the same problem, there will be slight differences.

This is the case for :

- Optimization
- Control
- Inverse Problem
- Randomness
- Unsteadiness
- Numerical Homogenization
- and more

Introduction – discretization.

Multiple shots approximation.

But if you need to solve many times the same type of problem, there is room for invention ! !

Of course you are not solving the same problem, there will be slight differences.

This is the case for :

- Optimization
- Control
- Inverse Problem
- Randomness
- Unsteadiness
- Numerical Homogenization
- and more

Introduction – discretization.

Multiple shots approximation.

But if you need to solve many times the same type of problem, there is room for invention ! !

Of course you are not solving the same problem, there will be slight differences.

This is the case for :

- Optimization
- Control
- Inverse Problem
- Randomness
- Unsteadiness
- Numerical Homogenization
- and more

Introduction – discretization.

Multiple shots approximation.

But if you need to solve many times the same type of problem, there is room for invention ! !

Of course you are not solving the same problem, there will be slight differences.

This is the case for :

- Optimization
- Control
- Inverse Problem
- Randomness
- Unsteadiness
- Numerical Homogenization
- and more

Introduction – discretization.

Multiple shots approximation.

But if you need to solve many times the same type of problem, there is room for invention ! !

Of course you are not solving the same problem, there will be slight differences.

This is the case for :

- Optimization
- Control
- Inverse Problem
- Randomness
- Unsteadiness
- Numerical Homogenization
- and more

Introduction – discretization.

Multiple shots approximation.

But if you need to solve many times the same type of problem, there is room for invention ! !

Of course you are not solving the same problem, there will be slight differences.

This is the case for :

- Optimization
- Control
- Inverse Problem
- Randomness
- Unsteadiness
- Numerical Homogenization
- and more

Introduction – discretization.

Multiple shots approximation.

But if you need to solve many times the same type of problem, there is room for invention ! !

Of course you are not solving the same problem, there will be slight differences.

This is the case for :

- Optimization
- Control
- Inverse Problem
- Randomness
- Unsteadiness
- Numerical Homogenization
- and more

Introduction – discretization.

Multiple shots approximation.

But if you need to solve many times the same type of problem, there is room for invention ! !

Of course you are not solving the same problem, there will be slight differences.

This is the case for :

- Optimization
- Control
- Inverse Problem
- Randomness
- Unsteadiness
- Numerical Homogenization
- and more

Introduction – discretization.

Multiple shots approximation.

But if you need to solve many times the same type of problem, there is room for invention ! !

Of course you are not solving the same problem, there will be slight differences.

This is the case for :

- Optimization
- Control
- Inverse Problem
- Randomness
- Unsteadiness
- Numerical Homogenization
- and more

Introduction – discretization.

Multiple shots approximation.

But if you need to solve many times the same type of problem, there is room for invention ! !

Of course you are not solving the same problem, there will be slight differences.

This is the case for :

- Optimization
- Control
- Inverse Problem
- Randomness
- Unsteadiness
- Numerical Homogenization
- and more

Introduction – discretization.

Multiple shots approximation.

Here using over and over the same method may be expensive, and most of the time there is not much parallelisation since the new simulation is often based on the previous one (except maybe for randomness)

There is room to take into account the fact that it is not a multipurpose approximation but a guided approximation ! !

Reduced Basis Approximations are associated to this context

Introduction – discretization.

Multiple shots approximation.

Here using over and over the same method may be expensive, and most of the time there is not much parallelisation since the new simulation is often based on the previous one (except maybe for randomness)

There is room to take into account the fact that it is not a multipurpose approximation but a guided approximation ! !

Reduced Basis Approximations are associated to this context

Introduction – discretization.

Multiple shots approximation.

Here using over and over the same method may be expensive, and most of the time there is not much parallelisation since the new simulation is often based on the previous one (except maybe for randomness)

There is room to take into account the fact that it is not a multipurpose approximation but a guided approximation ! !

Reduced Basis Approximations are associated to this context

Introduction – discretization.

Multiple shots approximation.

Here using over and over the same method may be expensive, and most of the time there is not much parallelisation since the new simulation is often based on the previous one (except maybe for randomness)

There is room to take into account the fact that it is not a multipurpose approximation but a guided approximation ! !

Reduced Basis Approximations are associated to this context

Introduction – discretization.

Multiple shots approximation.

Here using over and over the same method may be expensive, and most of the time there is not much parallelisation since the new simulation is often based on the previous one (except maybe for randomness)

There is room to take into account the fact that it is not a multipurpose approximation but a guided approximation ! !

Reduced Basis Approximations are associated to this context

Introduction – discretization.

Multiple shots approximation.

The parameters can be

- Constitutive coefficients
- Shape parameters
- Inverse problem (size, position, number)
- Randomness
- Time
- and more

This is the room for large dimensions

Introduction – discretization.

Multiple shots approximation.

The parameters can be

- Constitutive coefficients
- Shape parameters
- Inverse problem (size, position, number)
- Randomness
- Time
- and more

This is the room for large dimensions

Introduction – discretization.

Multiple shots approximation.

The parameters can be

- Constitutive coefficients
- Shape parameters
- Inverse problem (size, position, number)
- Randomness
- Time
- and more

This is the room for large dimensions

Introduction – discretization.

Multiple shots approximation.

The parameters can be

- Constitutive coefficients
- Shape parameters
- Inverse problem (size, position, number)
- Randomness
- Time
- and more

This is the room for large dimensions

Introduction – discretization.

Multiple shots approximation.

The parameters can be

- Constitutive coefficients
- Shape parameters
- Inverse problem (size, position, number)
- Randomness
- Time
- and more

This is the room for large dimensions

Introduction – discretization.

Multiple shots approximation.

The parameters can be

- Constitutive coefficients
- Shape parameters
- Inverse problem (size, position, number)
- Randomness
- Time
- and more

This is the room for large dimensions

Introduction – discretization.

Multiple shots approximation.

The parameters can be

- Constitutive coefficients
- Shape parameters
- Inverse problem (size, position, number)
- Randomness
- Time
- and more

This is the room for large dimensions

Parameter dependent solutions.

Set of all solutions

- Reduced Basis, POD and Reduced Order Methods for model and computational reduction for the approximation of parameter dependent PDE rely on the fact that the set of solutions (depending on the parameters) is a manifold with a simple structure.

- Stated in a mathematical way : ...
  It is relative to the smoothness of the set $\mathcal{S} = \{u(\mu), \mu \in \mathbb{P}\}$.
  This smoothness can be characterized by the notion of n-width following Kolmogorov

Parameter dependent solutions.

Set of all solutions

Definition

Let X be a normed linear space, $\mathcal{S}$ be a subset of X and X_n be a generic n-dimensional subspace of $\mathcal{X}$. The deviation of $\mathcal{S}$ from X_n is

$$E(\mathcal{S}; X_n) = \sup_{x \in \mathcal{S}} \inf_{y \in X_n} \|x - y\|_X.$$

The *Kolmogorov n-width* of $\mathcal{S}$ in X is given by

$$\begin{aligned} d_n(\mathcal{S}, X) &= \inf\{E(\mathcal{S}; X_n) : X_n \text{ an } n\text{-dimensional subspace of } X\} \\ &= \inf_{X_n} \sup_{x \in \mathcal{S}} \inf_{y \in X_n} \|x - y\|_X \ . \end{aligned} \tag{1}$$

The n-width of $\mathcal{S}$ thus measures the extent to which $\mathcal{S}$ may be approximated by a n-dimensional subspace of X.

Parameter dependent solutions.

Set of all solutions

Definition

Let X be a normed linear space, $\mathcal{S}$ be a subset of X and X_n be a generic n-dimensional subspace of $\mathcal{X}$. The deviation of $\mathcal{S}$ from X_n is

$$E(\mathcal{S}; X_n) = \sup_{x \in \mathcal{S}} \inf_{y \in X_n} \|x - y\|_X.$$

The *Kolmogorov n-width* of $\mathcal{S}$ in X is given by

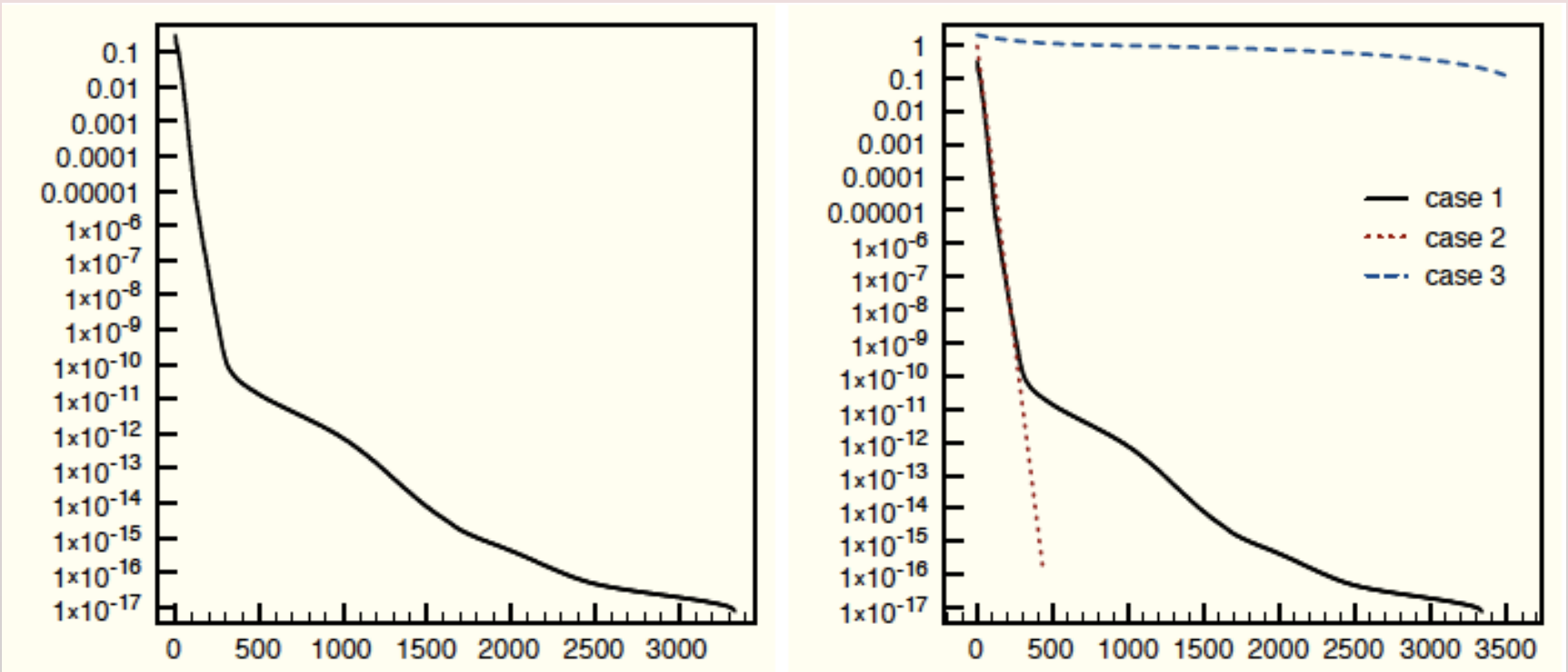
$$\begin{aligned} d_n(\mathcal{S}, X) &= \inf\{E(\mathcal{S}; X_n) : X_n \text{ an } n\text{-dimensional subspace of } X\} \\ &= \inf_{X_n} \sup_{x \in \mathcal{S}} \inf_{y \in X_n} \|x - y\|_X \ . \end{aligned} \tag{1}$$

The n-width of $\mathcal{S}$ thus measures the extent to which $\mathcal{S}$ may be approximated by a n-dimensional subspace of X.

Parameter dependent solutions.

Set of all solutions

A typical n-width.

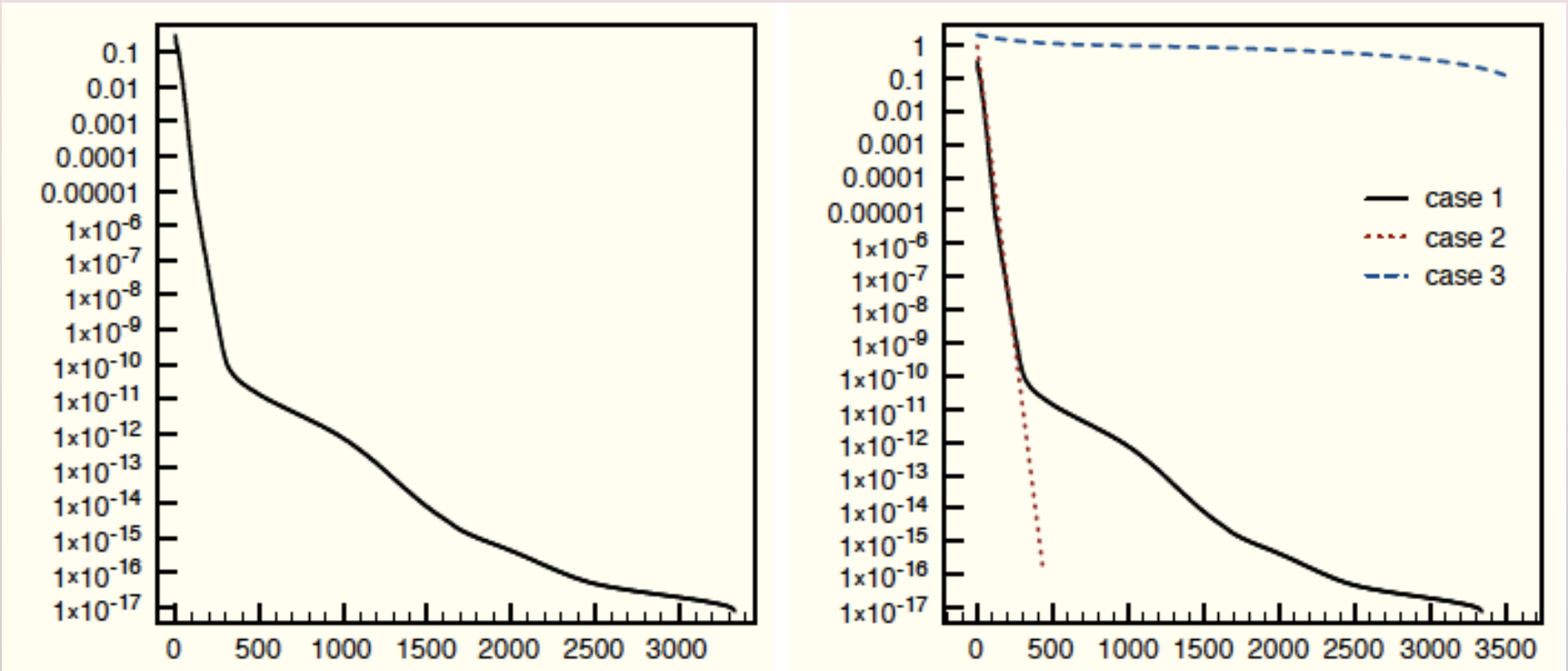


First test on a smaller problem . . .

Parameter dependent solutions.

Set of all solutions

A typical n-width.



First test on a smaller problem . . .

Parameter dependent solutions.

Reduced Approximation

The ability of being approximable by a small dimensional discrete space X_N does not mean that

- A pertinent choice is easy to find
- A pertinent discrete scheme is easy to find
- Such a discrete space is composed of solutions to problem (1) obtained for well chosen parameters μ_i, $i = 1, .., N$
- There is some idea of the accuracy that is obtained

For the first item, if you have no idea ... use POD (SVD) or even random

Parameter dependent solutions.

Reduced Approximation

The ability of being approximable by a small dimensional discrete space X_N does not mean that

- A pertinent choice is easy to find
- A pertinent discrete scheme is easy to find
- Such a discrete space is composed of solutions to problem (1) obtained for well chosen parameters μ_i, $i = 1, .., N$
- There is some idea of the accuracy that is obtained

For the first item, if you have no idea . . .use POD (SVD) or even random

Parameter dependent solutions.

Reduced Approximation

The ability of being approximable by a small dimensional discrete space X_N does not mean that

- A pertinent choice is easy to find
- A pertinent discrete scheme is easy to find
- Such a discrete space is composed of solutions to problem (1) obtained for well chosen parameters μ_i, $i = 1, .., N$
- There is some idea of the accuracy that is obtained

For the first item, if you have no idea . . .use POD (SVD) or even random

Parameter dependent solutions.

Reduced Approximation

The ability of being approximable by a small dimensional discrete space X_N does not mean that

- A pertinent choice is easy to find
- A pertinent discrete scheme is easy to find
- Such a discrete space is composed of solutions to problem (1) obtained for well chosen parameters μ_i, $i = 1, .., N$
- There is some idea of the accuracy that is obtained

For the first item, if you have no idea . . .use POD (SVD) or even random

Parameter dependent solutions.

Reduced Approximation

If you do not want to compute too many solutions . . .

Numerical Analysis allows to frame this type of approximation : error analysis, a priori and a posteriori

These a posteriori error analysis allow to construct a strategy for guiding the construction of X_N : the greedy algorithm provides such an opportunity. There are various implementations of such greedy procedure, depending on the degree of maturity the analysis of the RBM has got.

Parameter dependent solutions.

Reduced Approximation

If you do not want to compute too many solutions . . .

Numerical Analysis allows to frame this type of approximation : error analysis, a priori and a posteriori

These a posteriori error analysis allow to construct a strategy for guiding the construction of X_N : the greedy algorithm provides such an opportunity. There are various implementations of such greedy procedure, depending on the degree of maturity the analysis of the RBM has got.

Parameter dependent solutions.

Reduced Approximation

If you do not want to compute too many solutions . . .

Numerical Analysis allows to frame this type of approximation : error analysis, a priori and a posteriori
These a posteriori error analysis allow to construct a strategy for guiding the construction of X_N : the greedy algorithm provides such an opportunity. There are various implementations of such greedy procedure, depending on the degree of maturity the analysis of the RBM has got.

(Pure) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - P_{X_n} as the orthogonal projector onto X_n
  - $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- Thus at each step, the parameter μ_n is chosen in a greedy manner

(Pure) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - P_{X_n} as the orthogonal projector onto X_n
  - $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- Thus at each step, the parameter μ_n is chosen in a greedy manner

(Pure) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - P_{X_n} as the orthogonal projector onto X_n
  - $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- Thus at each step, the parameter μ_n is chosen in a greedy manner

(Pure) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - P_{X_n} as the orthogonal projector onto X_n
  - $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- Thus at each step, the parameter μ_n is chosen in a greedy manner

(Pure) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - P_{X_n} as the orthogonal projector onto X_n
  - $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- Thus at each step, the parameter μ_n is chosen in a greedy manner

(Pure) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - P_{X_n} as the orthogonal projector onto X_n
  - $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- Thus at each step, the parameter μ_n is chosen in a greedy manner

(Weak) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - error estimator $\widetilde{\varepsilon}(u(\mu))$ for all $\mu \in \mathbb{P}$ (e.g. a posteriori estimator for Galerkin reduced basis approximation onto X_n)
  - $\mu_{n+1} = \text{argmax}\{\widetilde{\varepsilon}(u(\mu)); \mu \in \mathbb{P}\}$
  - intead of $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- allows to construct (an approximation) of the only $u(\mu_m)$
- until $\widetilde{\varepsilon}(u(\mu_n))$ is small enough

Of course, a discrete finite element method (or spectral method) has to be applied to compute accurately each solution that are incorporated into the (reduced) basis set X_N.

(Weak) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - error estimator $\widetilde{\varepsilon}(u(\mu))$ for all $\mu \in \mathbb{P}$ (e.g. a posteriori estimator for Galerkin reduced basis approximation onto X_n)
  - $\mu_{n+1} = \text{argmax}\{\widetilde{\varepsilon}(u(\mu)); \mu \in \mathbb{P}\}$
  - intead of $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- allows to construct (an approximation) of the only $u(\mu_m)$
- until $\widetilde{\varepsilon}(u(\mu_n))$ is small enough

Of course, a discrete finite element method (or spectral method) has to be applied to compute accurately each solution that are incorporated into the (reduced) basis set X_N.

(Weak) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - error estimator $\widetilde{\varepsilon}(u(\mu))$ for all $\mu \in \mathbb{P}$ (e.g. a posteriori estimator for Galerkin reduced basis approximation onto X_n)
  - $\mu_{n+1} = \text{argmax}\{\widetilde{\varepsilon}(u(\mu)); \mu \in \mathbb{P}\}$
  - intead of $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- allows to construct (an approximation) of the only $u(\mu_m)$
- until $\widetilde{\varepsilon}(u(\mu_n))$ is small enough

Of course, a discrete finite element method (or spectral method) has to be applied to compute accurately each solution that are incorporated into the (reduced) basis set X_N.

(Weak) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - error estimator $\widetilde{\varepsilon}(u(\mu))$ for all $\mu \in \mathbb{P}$ (e.g. a posteriori estimator for Galerkin reduced basis approximation onto X_n)
  - $\mu_{n+1} = \text{argmax}\{\widetilde{\varepsilon}(u(\mu)); \mu \in \mathbb{P}\}$
  - intead of $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- allows to construct (an approximation) of the only $u(\mu_m)$
- until $\widetilde{\varepsilon}(u(\mu_n))$ is small enough

Of course, a discrete finite element method (or spectral method) has to be applied to compute accurately each solution that are incorporated into the (reduced) basis set X_N.

(Weak) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - error estimator $\widetilde{\varepsilon}(u(\mu))$ for all $\mu \in \mathbb{P}$ (e.g. a posteriori estimator for Galerkin reduced basis approximation onto X_n)
  - $\mu_{n+1} = \text{argmax}\{\widetilde{\varepsilon}(u(\mu)); \mu \in \mathbb{P}\}$
  - intead of $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- allows to construct (an approximation) of the only $u(\mu_m)$
- until $\widetilde{\varepsilon}(u(\mu_n))$ is small enough

Of course, a discrete finite element method (or spectral method) has to be applied to compute accurately each solution that are incorporated into the (reduced) basis set X_N.

(Weak) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - error estimator $\widetilde{\varepsilon}(u(\mu))$ for all $\mu \in \mathbb{P}$ (e.g. a posteriori estimator for Galerkin reduced basis approximation onto X_n)
  - $\mu_{n+1} = \text{argmax}\{\widetilde{\varepsilon}(u(\mu)); \mu \in \mathbb{P}\}$
  - intead of $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- allows to construct (an approximation) of the only $u(\mu_m)$
- until $\widetilde{\varepsilon}(u(\mu_n))$ is small enough

Of course, a discrete finite element method (or spectral method) has to be applied to compute accurately each solution that are recompressed into the (reduced) basis set X_N.

(Weak) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - error estimator $\widetilde{\varepsilon}(u(\mu))$ for all $\mu \in \mathbb{P}$ (e.g. a posteriori estimator for Galerkin reduced basis approximation onto X_n)
  - $\mu_{n+1} = \text{argmax}\{\widetilde{\varepsilon}(u(\mu)); \mu \in \mathbb{P}\}$
  - intead of $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- allows to construct (an approximation) of the only $u(\mu_m)$
- until $\widetilde{\varepsilon}(u(\mu_n))$ is small enough

Of course, a discrete finite element method (or spectral method) has to be added to compute accurately each solution that are incorporated into the (reduced) basis set X_N.

(Weak) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - error estimator $\widetilde{\varepsilon}(u(\mu))$ for all $\mu \in \mathbb{P}$ (e.g. a posteriori estimator for Galerkin reduced basis approximation onto X_n)
  - $\mu_{n+1} = \text{argmax}\{\widetilde{\varepsilon}(u(\mu)); \mu \in \mathbb{P}\}$
  - intead of $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- allows to construct (an approximation) of the only $u(\mu_m)$
- until $\widetilde{\varepsilon}(u(\mu_n))$ is small enough

Of course, a discrete finite element method (or spectral method) has to be added to compute accurately each solution that are incorporated into the (reduced) basis set X_N.

(Weak) greedy

- $\mu_1 = \text{argmax}\{\|u(\mu)\|_X; \mu \in \mathbb{P}\}$
- If $\mu_1, ..., \mu_n$ have been chosen, define
  - $X_n = \text{Span}\{u(\mu_1), ...u(\mu_n)\}$
  - error estimator $\widetilde{\varepsilon}(u(\mu))$ for all $\mu \in \mathbb{P}$ (e.g. a posteriori estimator for Galerkin reduced basis approximation onto X_n)
  - $\mu_{n+1} = \text{argmax}\{\widetilde{\varepsilon}(u(\mu)); \mu \in \mathbb{P}\}$
  - intead of $\mu_{n+1} = \text{argmax}\{\|u(\mu) - P_{X_n}(u(\mu))\|; \mu \in \mathbb{P}\}$
- allows to construct (an approximation) of the only $u(\mu_m)$
- until $\widetilde{\varepsilon}(u(\mu_n))$ is small enough

Of course, a discrete finite element method (or spectral method) has to be added to compute accurately each solution that are incorporated into the (reduced) basis set X_N.

The above method thus works fine for many example,

Notion of off-line/on-line computation

When the solution regularity with respect to the parameter is not so good, or when the dimension of the parameter grows, the actual size N of the reduced basis may turn out to be larger than desired. This has led to different adaptations of the method.

Remember that, by breaking the global framework to locally piecewise global approaches, the spectral element methods reveals superiority with respect to plain spectral method : a trade off between locality and globality is generally preferred as is demonstrated in e.g. [Cantwell, Sherwin, Kirby, Kelly] for approximation in spacial direction by spectral element/fourier analysis.

The above method thus works fine for many example,

Notion of off-line/on-line computation

When the solution regularity with respect to the parameter is not so good, or when the dimension of the parameter grows, the actual size N of the reduced basis may turn out to be larger than desired. This has led to different adaptations of the method.

Remember that, by breaking the global framework to locally piecewise global approaches, the spectral element methods reveals superiority with respect to plain spectral method : a trade off between locality and globality is generally preferred as is demonstrated in e.g. [Cantwell, Sherwin, Kirby, Kelly] for approximation in spacial direction by spectral element approximations.

The above method thus works fine for many example,

Notion of off-line/on-line computation

When the solution regularity with respect to the parameter is not so good, or when the dimension of the parameter grows, the actual size N of the reduced basis may turn out to be larger than desired. This has led to different adaptations of the method.

Remember that, by breaking the global framework to locally piecewise global approaches, the spectral element methods reveals superiority with respect to plain spectral method : a trade off between locality and globality is generally preferred as is demonstrated in e.g. [Cantwell, Sherwin, Kirby, Kelly] for approximation in spacial direction by spectral element approximations.

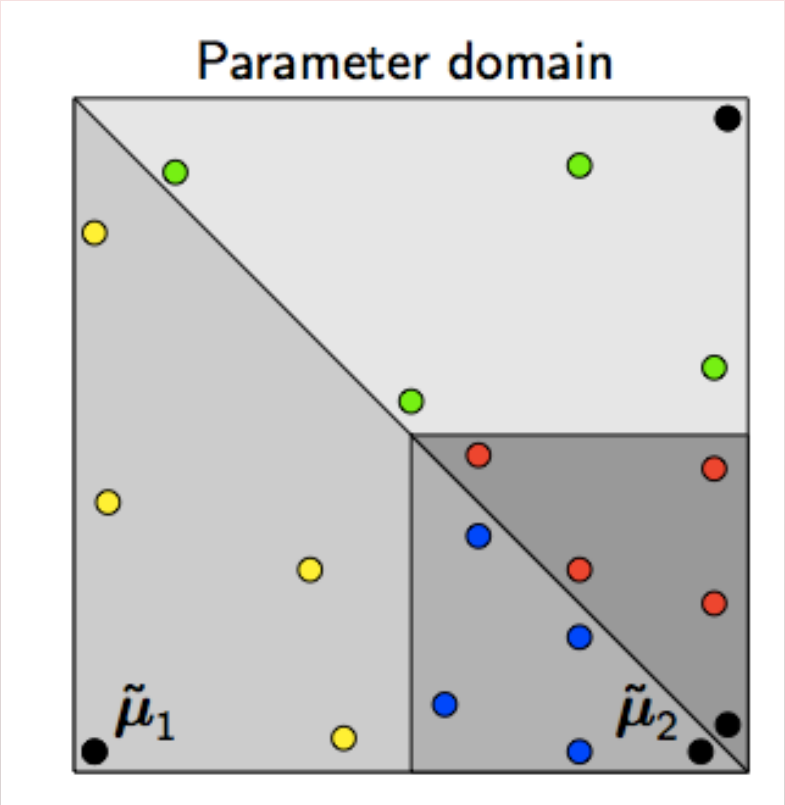
The above method thus works fine for many example,

Notion of off-line/on-line computation

When the solution regularity with respect to the parameter is not so good, or when the dimension of the parameter grows, the actual size N of the reduced basis may turn out to be larger than desired. This has led to different adaptations of the method.

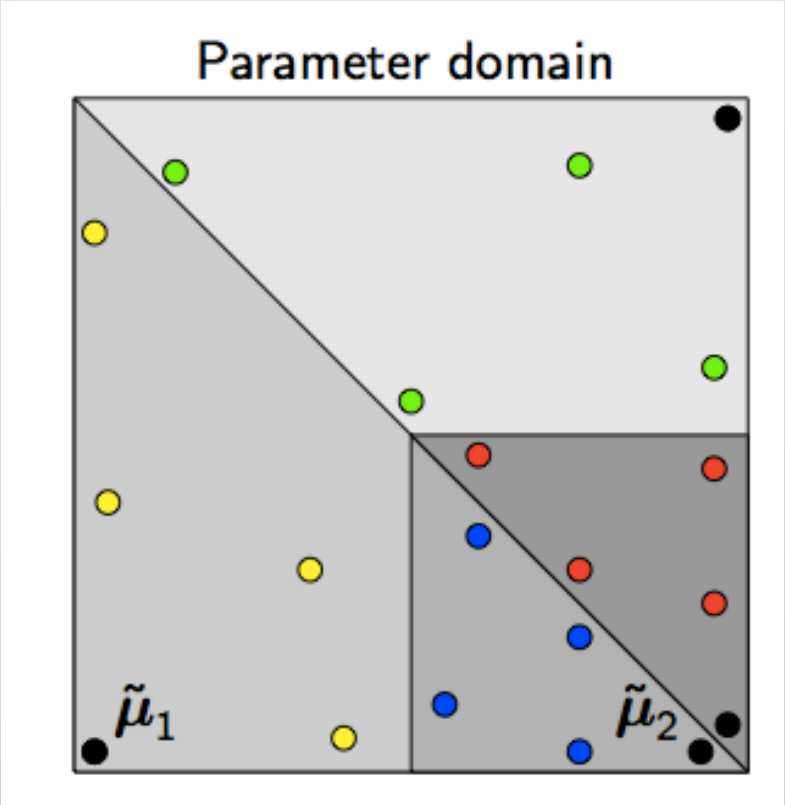
Remember that, by breaking the global framework to locally piecewise global approaches, the spectral element methods reveals superiority with respect to plain spectral method : a trade off between locality and globality is generally preferred as is demonstrated in e.g. [Cantwell, Sherwin, Kirby, Kelly] for approximation in spacial direction by spectral element approximations.

Following the lines drawn above, a first idea in this context has been presented in [Eftang, Patera, Rønquist] and also [Eftang, Stamm] where the parameter space is decomposed into cells where different reduced basis sets are assembled.


Parameter domain
$\tilde{\mu}_1$
$\tilde{\mu}_2$

This approach presents clear advantages in the size of the matricial system that appears in the on-line solution procedure, and corroborates the natural feeling that, in order to approximate the solution at a given parameter, primarily those solutions in the reduced basis corresponding to parameters that are close to the parameter we are interested in are to be involved in the linear approximation.

A drawback of the current approach however is that, in two adjacent parameter-subdomains, some of the parameters that are selected may be very close.



Due to the difference between the CPU time associated with exploiting a RB method in the on-line stage, and that required for constructing the basis elements in the off-line step, this leads to the idea that it might be interesting to be able to use, in one parameter subdomain, the parameters that are used in the adjacent ones.

Another drawback of all current greedy approach is to be unable to master the size of the discrete system that will be solved in the on-line procedure.

These remarks have motivated us, with Benjamin Stamm, to investigate the alternative discussed in the following.

Another feature that will be presented here is the detection of the important parameters, or combination of parameters. I.e. a non isotropic parameter-wise fine tuning of the sensitivity of the method to ...

Due to the difference between the CPU time associated with exploiting a RB method in the on-line stage, and that required for constructing the basis elements in the off-line step, this leads to the idea that it might be interesting to be able to use, in one parameter subdomain, the parameters that are used in the adjacent ones.

Another drawback of all current greedy approach is to be unable to master the size of the discrete system that will be solved in the on-line procedure.

These remarks have motivated us, with Benjamin Stamm, to investigate the alternative discussed in the following.

Another feature that will be presented here is the detection of the important parameters, or combination of parameters . . .i.e. a non isotropic learning process for the geometry of the manifold $\mathcal{S}$.

Due to the difference between the CPU time associated with exploiting a RB method in the on-line stage, and that required for constructing the basis elements in the off-line step, this leads to the idea that it might be interesting to be able to use, in one parameter subdomain, the parameters that are used in the adjacent ones.

Another drawback of all current greedy approach is to be unable to master the size of the discrete system that will be solved in the on-line procedure.

These remarks have motivated us, with Benjamin Stamm, to investigate the alternative discussed in the following.

Another feature that will be presented here is the detection of the important parameters, or combination of parameters . . .i.e. a non isotropic learning process for the geometry of the manifold $\mathcal{S}$.

Due to the difference between the CPU time associated with exploiting a RB method in the on-line stage, and that required for constructing the basis elements in the off-line step, this leads to the idea that it might be interesting to be able to use, in one parameter subdomain, the parameters that are used in the adjacent ones.

Another drawback of all current greedy approach is to be unable to master the size of the discrete system that will be solved in the on-line procedure.

These remarks have motivated us, with Benjamin Stamm, to investigate the alternative discussed in the following.

Another feature that will be presented here is the detection of the important parameters, or combination of parameters . . .i.e. a non isotropic learning process for the geometry of the manifold $\mathcal{S}$.

ClassicalGreedy

1. Choose (possibly randomly) $\mu^1 \in \Xi_{\texttt{trial}}$, set $\mathbb{S}_1 = \{\mu^1\}$, $\mathbb{W}_1 = \text{span}\{u(\mu^1)\}$ and $N = 1$, $\texttt{err} = \max_{\mu \in \Xi_{\texttt{trial}}} \eta(\mu; W_1)$.
2. While $\texttt{err} > \texttt{tol}$
3. Find $\mu^{N+1} = \text{argmax}_{\mu \in \Xi_{\texttt{trial}}} \eta(\mu; W_N)$, $\texttt{err} = \max_{\mu \in \Xi_{\texttt{trial}}} \eta(\mu; W_N)$.
4. Compute $u(\mu^{N+1})$, set $\mathbb{S}_{N+1} = \mathbb{S}_N \cup \{\mu^{N+1}\}$ and $\mathbb{W}_{N+1} = \text{span}\{\mathbb{W}_N, u(\mu^{N+1})\}$.
5. Set $N := N + 1$.
6. End while.

Algorithm : Classical greedy algorithm.

The introduction of the finite set $\Xi_{\texttt{trial}}$ is due to practical implementation. It should be large enough, but not too large
Actually the definition of this finite set can evolve during the algorithm.

Locally adaptive reduced basis method

Here, we do not impose a clear partition of the parameter space but rather collect a global set of sample points $\mathbb{S}$ (preliminary constructed — in an off-line stage — with a given tolerance requirement)

We choose a priori the size of the system we want to solve online by selecting an integer N, then when a reduced basis approximation is to be computed for a certain given parameter value $\mu \in \mathbb{P}$, we only use the N basis functions whose parameter values lie close to μ

Locally adaptive reduced basis method

Here, we do not impose a clear partition of the parameter space but rather collect a global set of sample points $\mathbb{S}$ (preliminary constructed — in an off-line stage — with a given tolerance requirement)

We choose a priori the size of the system we want to solve online by selecting an integer N, then when a reduced basis approximation is to be computed for a certain given parameter value $\mu \in \mathbb{P}$, we only use the N basis functions whose parameter values lie close to μ

Locally adaptive reduced basis method

Here, we do not impose a clear partition of the parameter space but rather collect a global set of sample points $\mathbb{S}$ (preliminary constructed — in an off-line stage — with a given tolerance requirement)

We choose a priori the size of the system we want to solve online by selecting an integer N, then when a reduced basis approximation is to be computed for a certain given parameter value $\mu \in \mathbb{P}$, we only use the N basis functions whose parameter values lie close to μ

Locally adaptive reduced basis method

Here, we do not impose a clear partition of the parameter space but rather collect a global set of sample points $\mathbb{S}$ (preliminary constructed — in an off-line stage — with a given tolerance requirement)

We choose a priori the size of the system we want to solve online by selecting an integer N, then when a reduced basis approximation is to be computed for a certain given parameter value $\mu \in \mathbb{P}$, we only use the N basis functions whose parameter values lie close to μ

These N basis functions whose parameter values lie close to μ are in a ball

$$B_\mu = \{\tilde{\mu} \in \mathbb{P} \,|\, d(\mu, \tilde{\mu}) \leq r(\mu)\}, \tag{2}$$

for a given semi-distance function $d(\cdot, \cdot)$ defined on the fly by learning the topology of the manifold of all solutions.

The radius $r(\mu)$ is computed in such a way that there are actually N basis functions in the ball. Therefore the local sample space is defined by

$$\mathbb{S}_\mu = B_\mu \cap \mathbb{S} = \{\tilde{\mu} \in \mathbb{S} \,|\, \tilde{\mu} \in B_\mu\}$$

with cardinality equal to N.

The local reduced basis approximation space shall be defined by $W_\mu = \text{span}\{u(\tilde{\mu}) \,|\, \tilde{\mu} \in \mathbb{S}_\mu\}$ and its associated Kolmogorov number by $d_N(\mathcal{M}) = W_\mu$.

These N basis functions whose parameter values lie close to μ are in a ball

$$B_\mu = \{\tilde{\mu} \in \mathbb{P} \,|\, d(\mu, \tilde{\mu}) \leq r(\mu)\}, \tag{2}$$

for a given semi-distance function $d(\cdot, \cdot)$ defined on the fly by learning the topology of the manifold of all solutions.

The radius $r(\mu)$ is computed in such a way that there are actually N basis functions in the ball. Therefore the local sample space is defined by

$$\mathbb{S}_\mu = B_\mu \cap \mathbb{S} = \{\tilde{\mu} \in \mathbb{S} \,|\, \tilde{\mu} \in B_\mu\}$$

with cardinality equal to N.

The local reduced basis approximation space shall be defined by $W_\mu = \text{span}\{u(\tilde{\mu}) \,|\, \tilde{\mu} \in \mathbb{S}_\mu\}$ and its associated local approximation by $P_\mu : \mathcal{W} \to \mathcal{W}_\mu$.

These N basis functions whose parameter values lie close to μ are in a ball

$$B_\mu = \{\tilde{\mu} \in \mathbb{P} \,|\, d(\mu, \tilde{\mu}) \leq r(\mu)\}, \tag{2}$$

for a given semi-distance function $d(\cdot, \cdot)$ defined on the fly by learning the topology of the manifold of all solutions.

The radius $r(\mu)$ is computed in such a way that there are actually N basis functions in the ball. Therefore the local sample space is defined by

$$\mathbb{S}_\mu = B_\mu \cap \mathbb{S} = \{\tilde{\mu} \in \mathbb{S} \,|\, \tilde{\mu} \in B_\mu\}$$

with cardinality equal to N.

The local reduced basis approximation space shall be defined by $\mathbb{W}_\mu = \text{span}\{u(\tilde{\mu}) \,|\, \tilde{\mu} \in \mathbb{S}_\mu\}$ and its associated local projection by $P_\mu : \mathbb{W} \to \mathbb{W}_\mu$.

These N basis functions whose parameter values lie close to μ are in a ball

$$B_\mu = \{\tilde{\mu} \in \mathbb{P} \,|\, d(\mu, \tilde{\mu}) \leq r(\mu)\}, \tag{2}$$

for a given semi-distance function $d(\cdot, \cdot)$ defined on the fly by learning the topology of the manifold of all solutions.

The radius $r(\mu)$ is computed in such a way that there are actually N basis functions in the ball. Therefore the local sample space is defined by

$$\mathbb{S}_\mu = B_\mu \cap \mathbb{S} = \{\tilde{\mu} \in \mathbb{S} \,|\, \tilde{\mu} \in B_\mu\}$$

with cardinality equal to N.

The local reduced basis approximation space shall be defined by $\mathbb{W}_\mu = \text{span}\{u(\tilde{\mu}) \,|\, \tilde{\mu} \in \mathbb{S}_\mu\}$ and its associated local projection by $P_\mu : \mathbb{W} \to \mathbb{W}_\mu$.

These N basis functions whose parameter values lie close to μ are in a ball

$$B_\mu = \{\tilde{\mu} \in \mathbb{P} \,|\, d(\mu, \tilde{\mu}) \leq r(\mu)\}, \tag{2}$$

for a given semi-distance function $d(\cdot, \cdot)$ defined on the fly by learning the topology of the manifold of all solutions.

The radius $r(\mu)$ is computed in such a way that there are actually N basis functions in the ball. Therefore the local sample space is defined by

$$\mathbb{S}_\mu = B_\mu \cap \mathbb{S} = \{\tilde{\mu} \in \mathbb{S} \,|\, \tilde{\mu} \in B_\mu\}$$

with cardinality equal to N.

The local reduced basis approximation space shall be defined by $\mathbb{W}_\mu = \text{span}\{u(\tilde{\mu}) \,|\, \tilde{\mu} \in \mathbb{S}_\mu\}$ and its associated local projection by $P_\mu : \mathbb{W} \to \mathbb{W}_\mu$.

Definition of the Hessian.

The goal is to define a Hessian matrix $\overline{H}(\mu)$ for each point $\mu \in \Xi_{\texttt{trial}}$ upon which the metric will be based on.

This Hessian is based on the reduced basis approximation : it is updated/constructed at each iteration in the algorithm.

Definition of the Hessian.

The goal is to define a Hessian matrix $\overline{H}(\mu)$ for each point $\mu \in \Xi_{\texttt{trial}}$ upon which the metric will be based on.

This Hessian is based on the reduced basis approximation : it is updated/constructed at each iteration in the algorithm.

On-line construction of the metric.

One can then perform an eigenvalue decomposition

$$H_{\mu_1\mu_2} = V \Lambda V^T$$

where V is an orthogonal and $(\Lambda)_{ii} = \lambda_i$ a diagonal matrix consisting of the eigenvalues λ_i. Consider the diagonal matrix

$$|\Lambda|_{ii} = \frac{|\lambda_i|}{\sqrt{\lambda_1^2 + \ldots + \lambda_P^2}}, \qquad i = 1, \ldots, P,$$

and the associated symmetric positive definite matrix $M_{\mu_1\mu_2} = V\,|\Lambda|\,V^T$ to define the semi-distance between μ_1 and μ_2 by

$$d(\mu_1, \mu_2) = \sqrt{(\mu_1 - \mu_2)^T M_{\mu_1\mu_2} (\mu_1 - \mu_2)}.$$

We focus now on the trial set $\Xi_{\texttt{trial}}$. It is aimed to keep its cardinality as small as possible, but large enough to capture the local geometry of the parametrized system.

Since not only the cardinality of $\Xi_{\texttt{trial}}$ but also the locations of its points matters we propose to use the above constructed metric to design problem adapted training sets of appropriate size which will increase with increasing number of basis functions selected.

Firstly, we construct at each iteration of the algorithm a new trial set $\Xi_{\texttt{trial}}$. The cardinality of the $\Xi_{\texttt{trial}}$ being variable and an increasing function of the inverse of the actual error (-estimation) $\texttt{err}$.

The new trial set $\Xi_{\texttt{trial}}$ is constructed such that edges of the corresponding unique Delaunay triangulation are uniform in the slightly modified metric.

We focus now on the trial set $\Xi_{\mathtt{trial}}$. It is aimed to keep its cardinality as small as possible, but large enough to capture the local geometry of the parametrized system.
Since not only the cardinality of $\Xi_{\mathtt{trial}}$ but also the locations of its points matters we propose to use the above constructed metric to design problem adapted training sets of appropriate size which will increase with increasing number of basis functions selected.

Firstly, we construct at each iteration of the algorithm a new trial set $\Xi_{\mathtt{trial}}$. The cardinality of the $\Xi_{\mathtt{trial}}$ being variable and an increasing function of the inverse of the actual error (-estimation) $\mathtt{err}$.
The new trial set $\Xi_{\mathtt{trial}}$ is constructed such that edges of the corresponding unique Delaunay triangulation are uniform in the slightly modified metric

We focus now on the trial set $\Xi_{\texttt{trial}}$. It is aimed to keep its cardinality as small as possible, but large enough to capture the local geometry of the parametrized system.
Since not only the cardinality of $\Xi_{\texttt{trial}}$ but also the locations of its points matters we propose to use the above constructed metric to design problem adapted training sets of appropriate size which will increase with increasing number of basis functions selected.

Firstly, we construct at each iteration of the algorithm a new trial set $\Xi_{\texttt{trial}}$. The cardinality of the $\Xi_{\texttt{trial}}$ being variable and an increasing function of the inverse of the actual error (-estimation) $\texttt{err}$.
The new trial set $\Xi_{\texttt{trial}}$ is constructed such that edges of the corresponding unique Delaunay triangulation are uniform in the slightly modified metric

We focus now on the trial set $\Xi_{\texttt{trial}}$. It is aimed to keep its cardinality as small as possible, but large enough to capture the local geometry of the parametrized system.
Since not only the cardinality of $\Xi_{\texttt{trial}}$ but also the locations of its points matters we propose to use the above constructed metric to design problem adapted training sets of appropriate size which will increase with increasing number of basis functions selected.

Firstly, we construct at each iteration of the algorithm a new trial set $\Xi_{\texttt{trial}}$. The cardinality of the $\Xi_{\texttt{trial}}$ being variable and an increasing function of the inverse of the actual error (-estimation) $\texttt{err}$.
The new trial set $\Xi_{\texttt{trial}}$ is constructed such that edges of the corresponding unique Delaunay triangulation are uniform in the slightly modified metric

Before we start with the numerical tests, an abstract description of the entire proposed local greedy algorithm is given in the following box :

`LocallyAdaptiveGreedy`

`========== Stage 1 ==========`

1. Perform a classical greedy algorithm to select $N+1$ basis functions.

`========== Stage 2 ==========`

2. Compute the error estimate $\eta(\mu, W_\mu)$ at each point $\mu \in \Xi_{\texttt{trial}}$.
3. Compute the metric function d.
4. Enrich the set of basis functions.
5. Create a new trial set $\Xi_{\texttt{trial}}$.
6. Go to `2.` until tolerance `tol` is achieved.

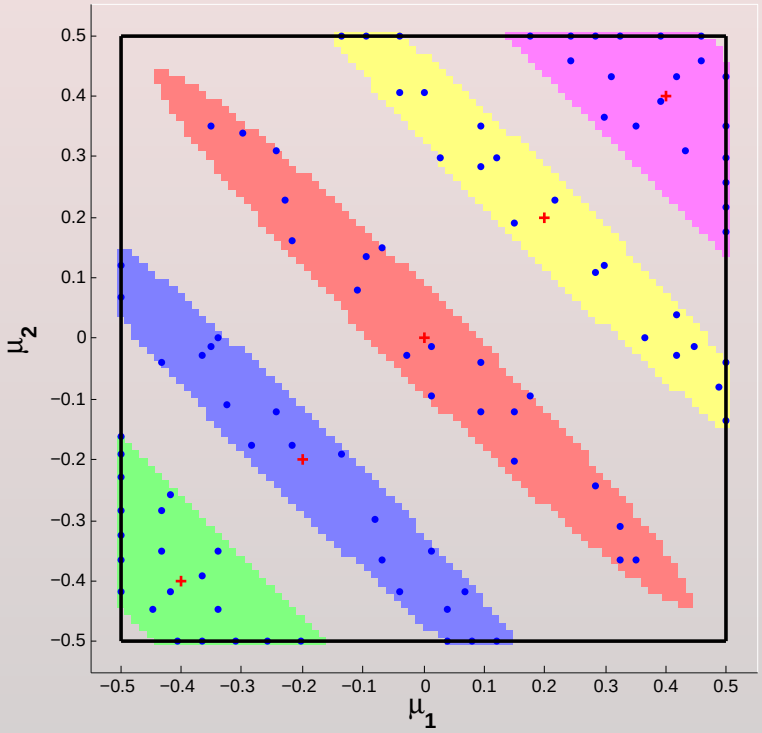
Algorithm : Locally adaptive greedy algorithm.

First with no adaptation of the trial set $\Xi_{\mathtt{trial}}$

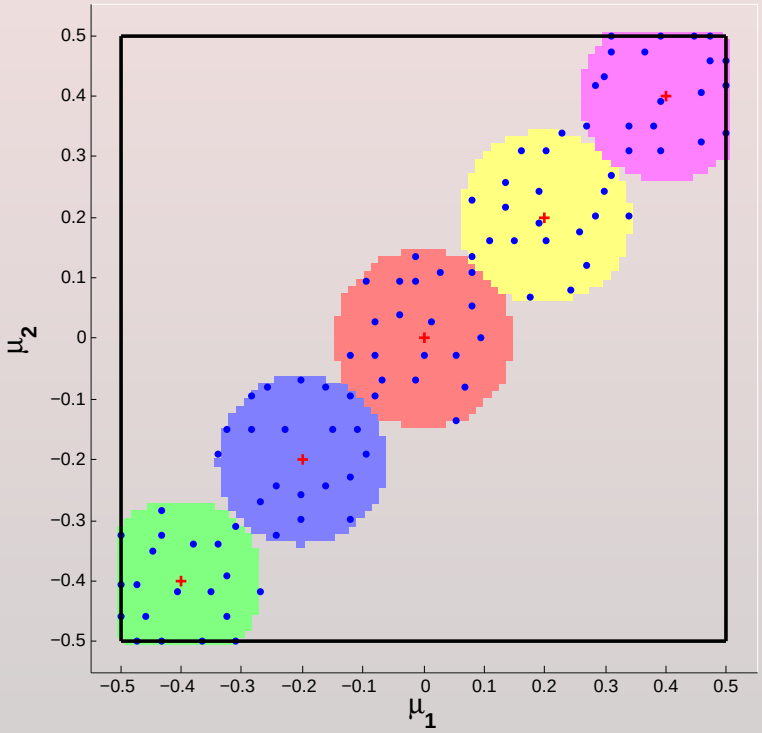
We start with presenting a numerical example to illustrate the benefit of the local anisotropic approximation spaces. Consider the function

$$f_1(\mathbf{x};\mu) = \exp\left[\frac{-(x_1-0.1(\mu_1-\mu_2))^2}{0.01} - \frac{(x_2-(\mu_1+\mu_2))^2}{0.01}\right],$$
$$\mathbf{x} \in \Omega = (-1,1)^2,\ \mu \in \mathbb{P} = [-0.5,0.5]^2$$

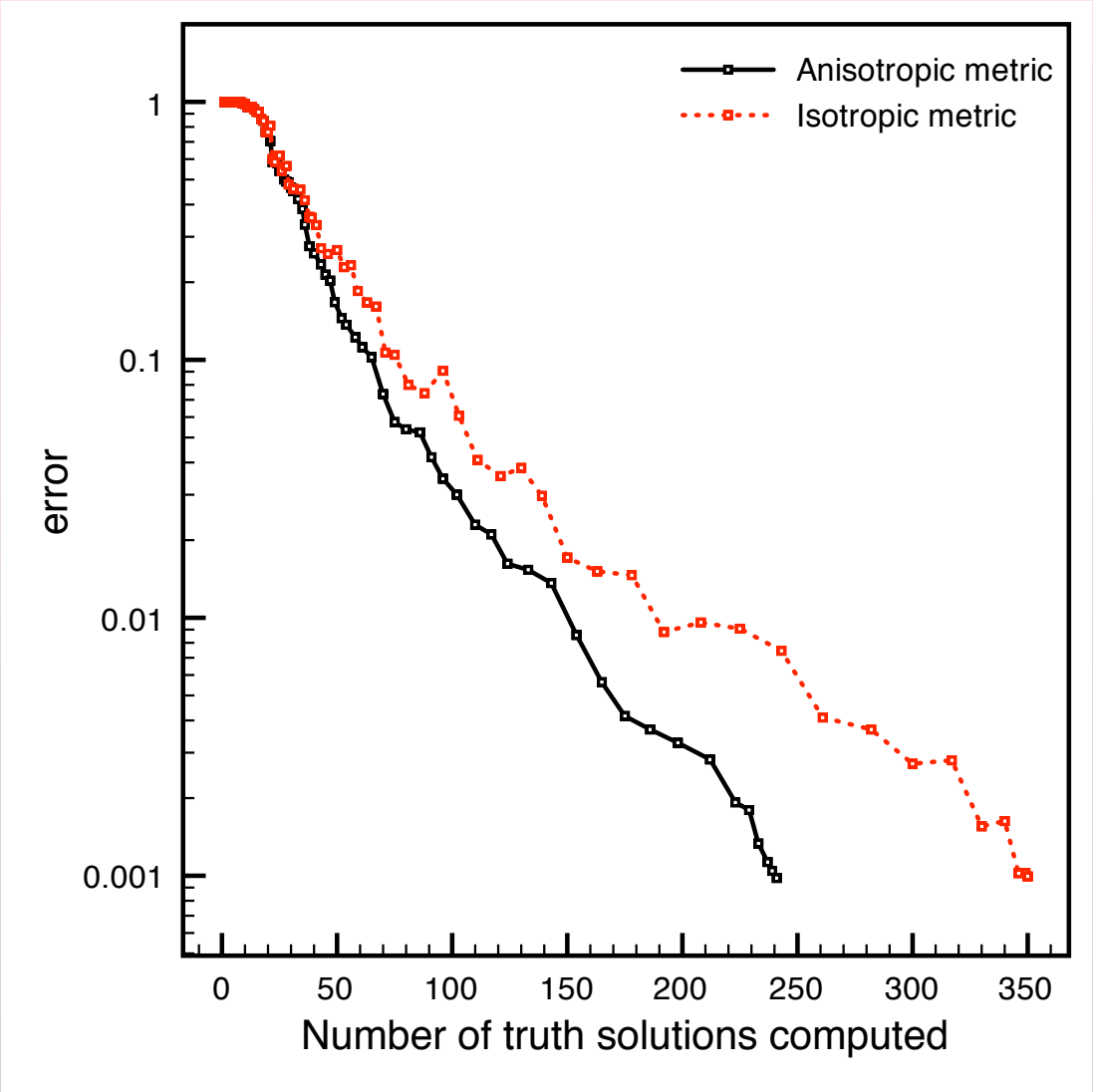
that exhibits a constant anisotropy of parameters over the whole parameter space. The dependency in the $(\mu_1+\mu_2)$-direction is ten times stronger than in the $(\mu_1-\mu_2)$-direction.



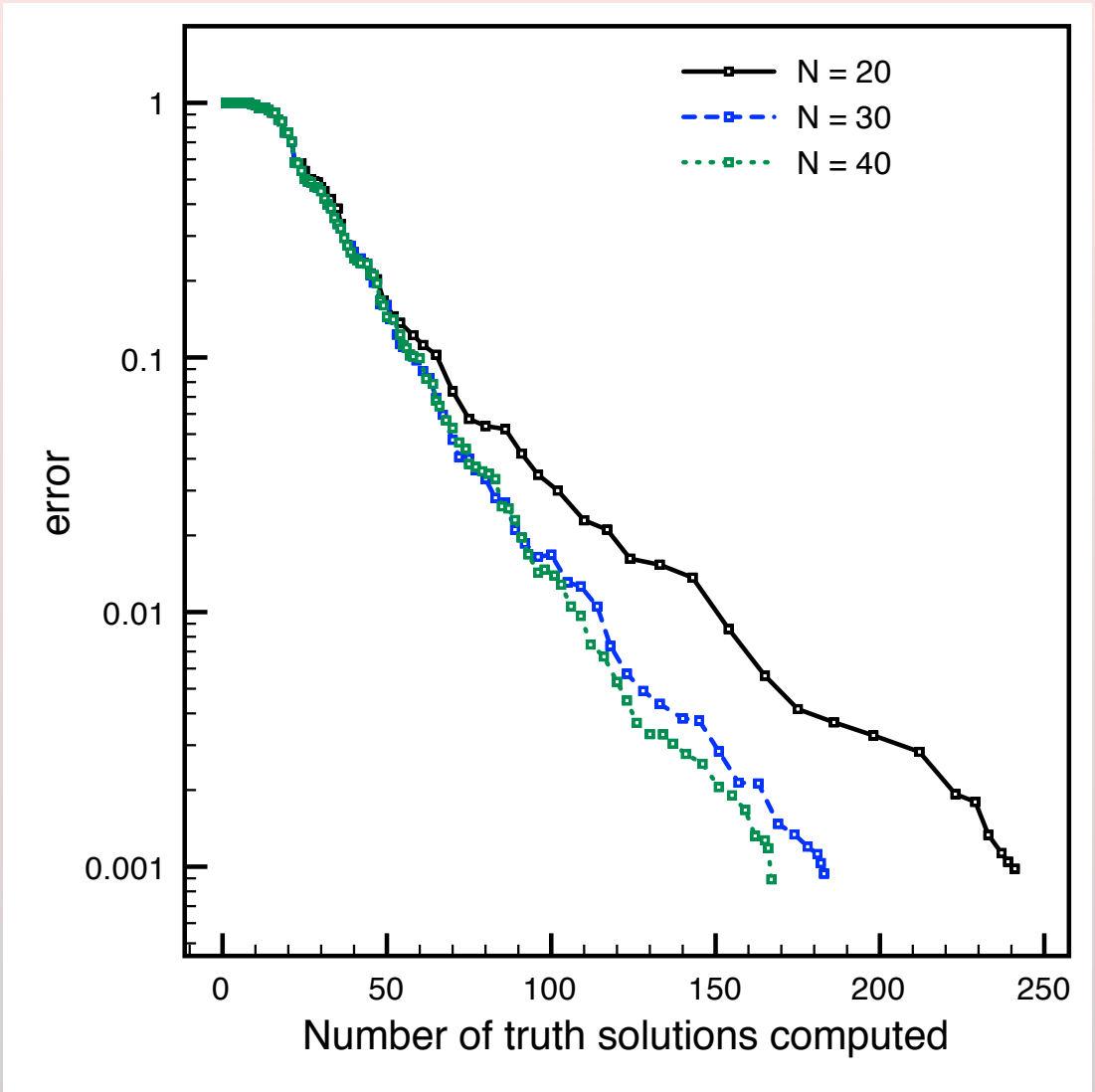
Local approximation spaces (anisotropic metric)



Local approximation spaces (isotropic metric)



Comparison of different approaches.

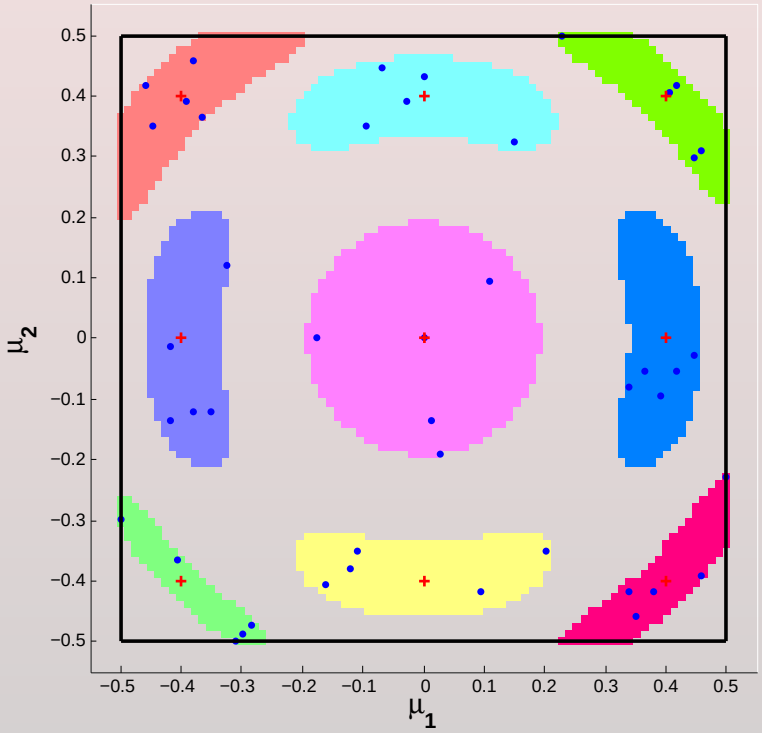


Comparison under variation of N

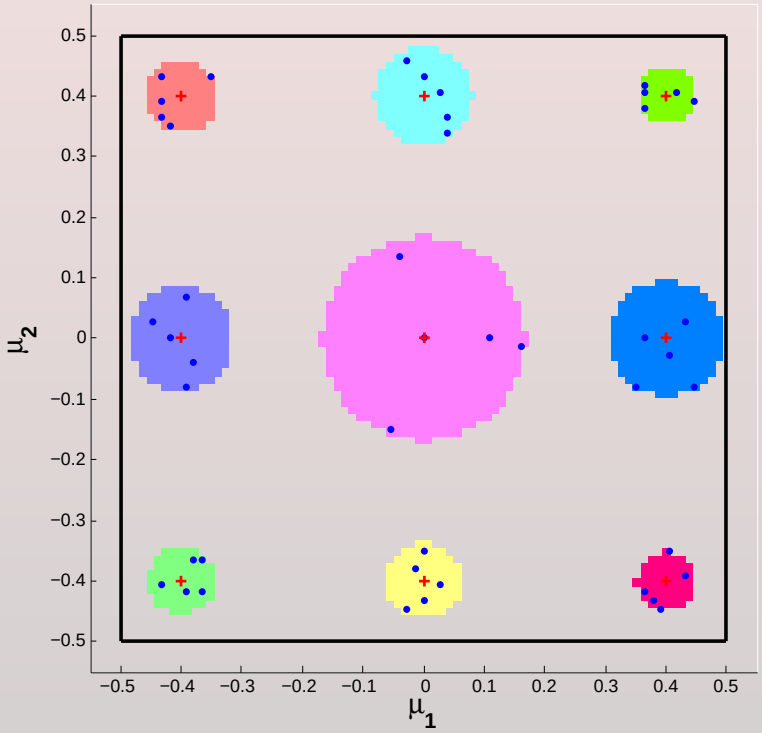
the next example

$$f_2(\mathbf{x};\mu) = \exp\left[-\frac{(x_1-(\mu_1^2+\mu_2^2))^2}{0.01} - \frac{(x_2-(\mu_1^2+\mu_2^2))^2}{0.01}\right],$$
$$\mathbf{x} \in \Omega = (-1,1)^2,\ \mu \in \mathbb{P} = [-0.5,0.5]^2$$

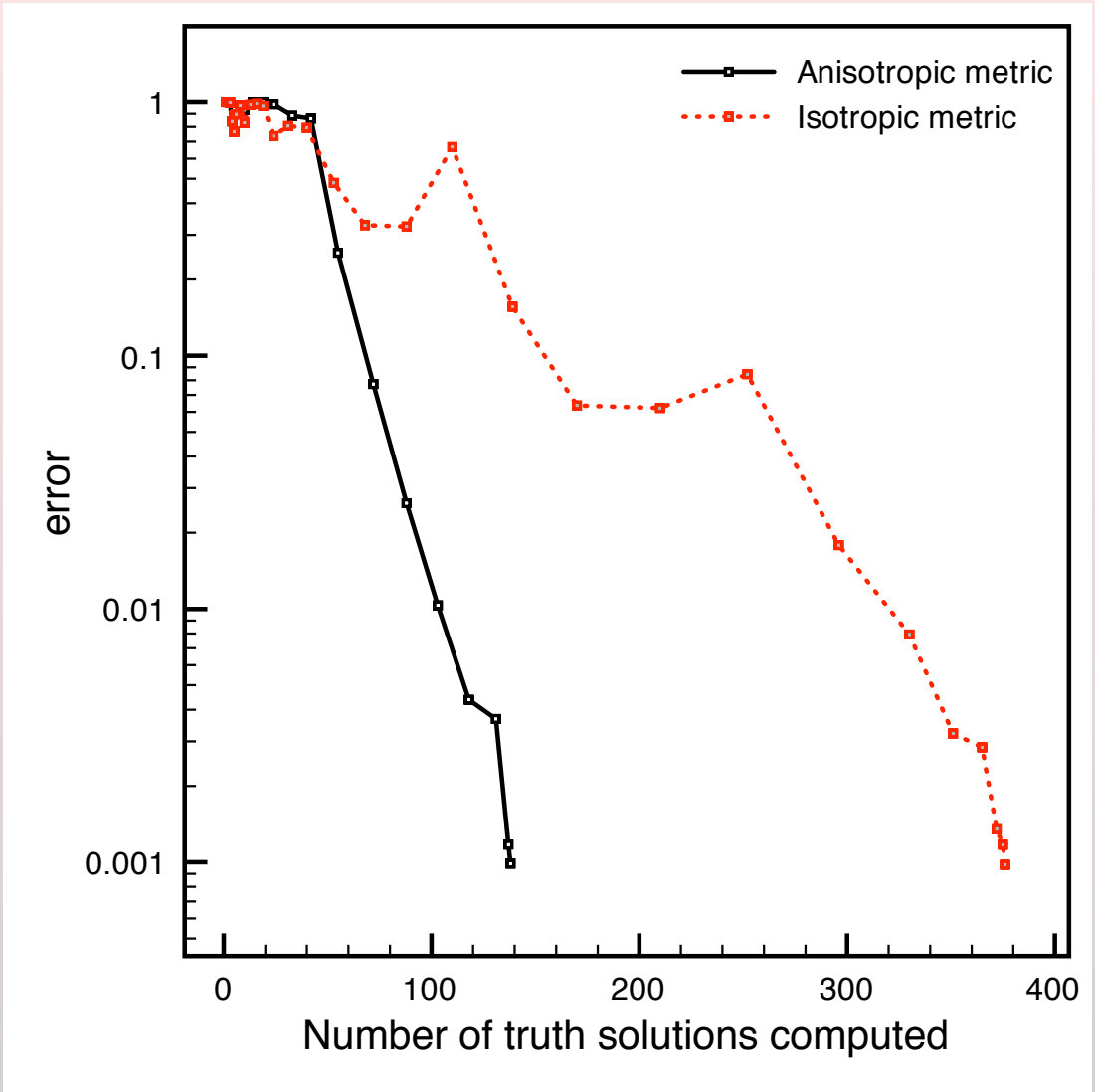
is more interesting. It presents a family of parametrized functions where the functions (as functions of $\mathbf{x}$) are constant along concentric circles around the origin in parameter space.



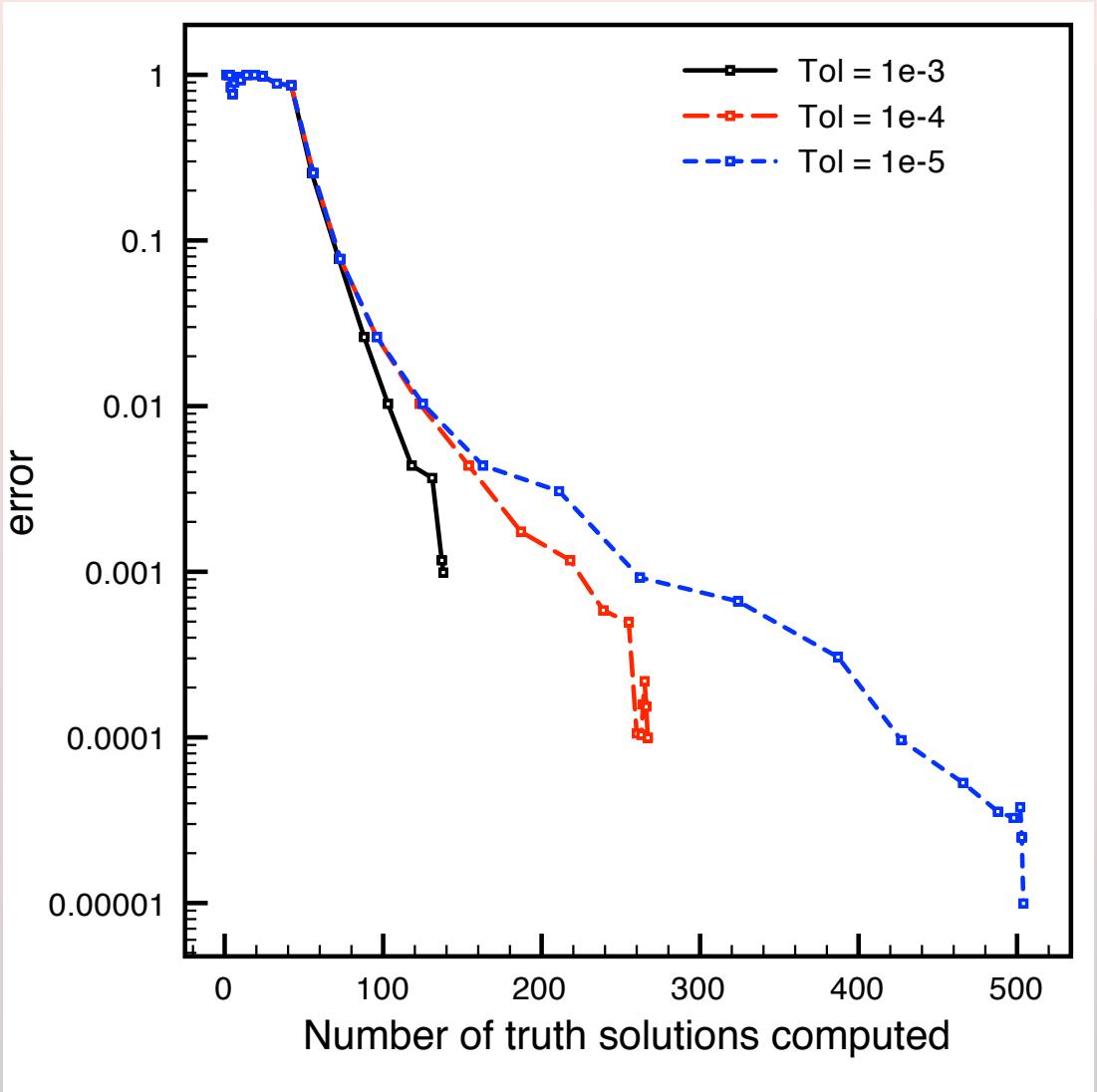
Local approximation spaces (anisotropic metric)



Local approximation spaces (isotropic metric)



Anisotropy vs. Isotropy

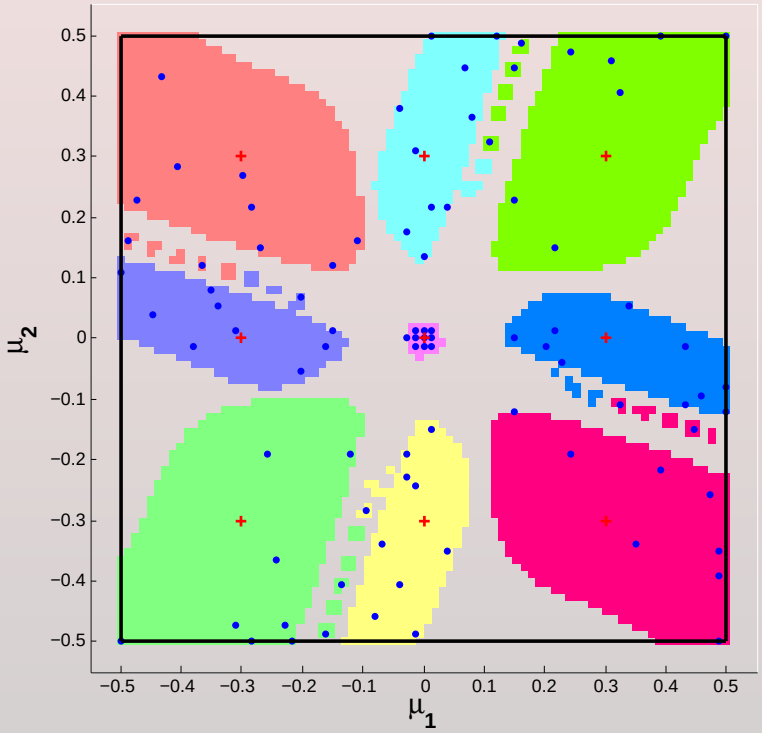


Different tolerances

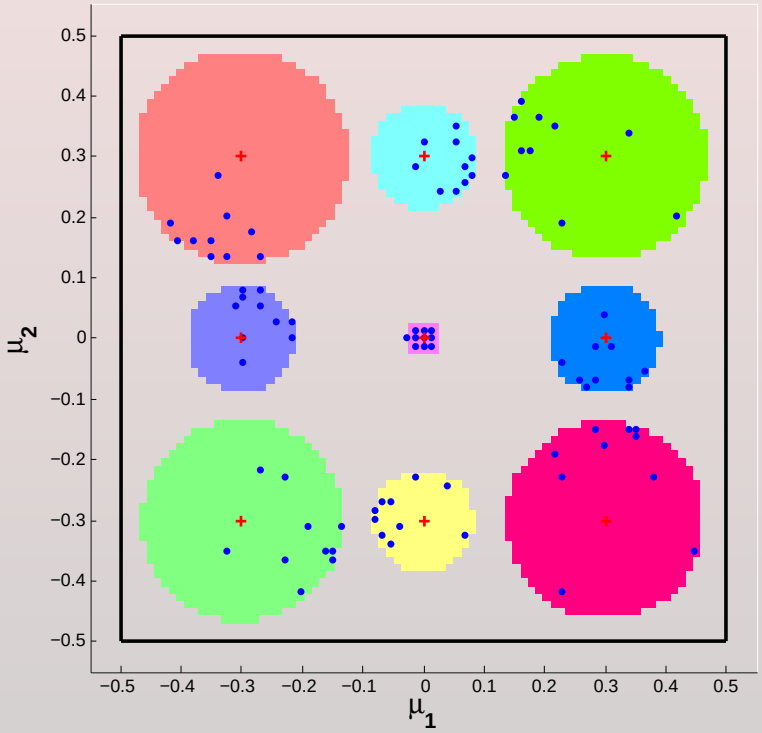
The next example

$$f_3(\mathbf{x};\mu) = \exp\left[-\frac{(x_1-(\mu_1+3\mu_2))^2}{0.1+5|\mu_1+3\mu_2|} - \frac{(x_2-(3\mu_1-\mu_2))^2}{0.1+5|3\mu_1-\mu_2|}\right],$$
$$\mathbf{x} \in \Omega = (-1,1)^2,\ \mu \in \mathbb{P} = [-0.5,0.5]^2$$

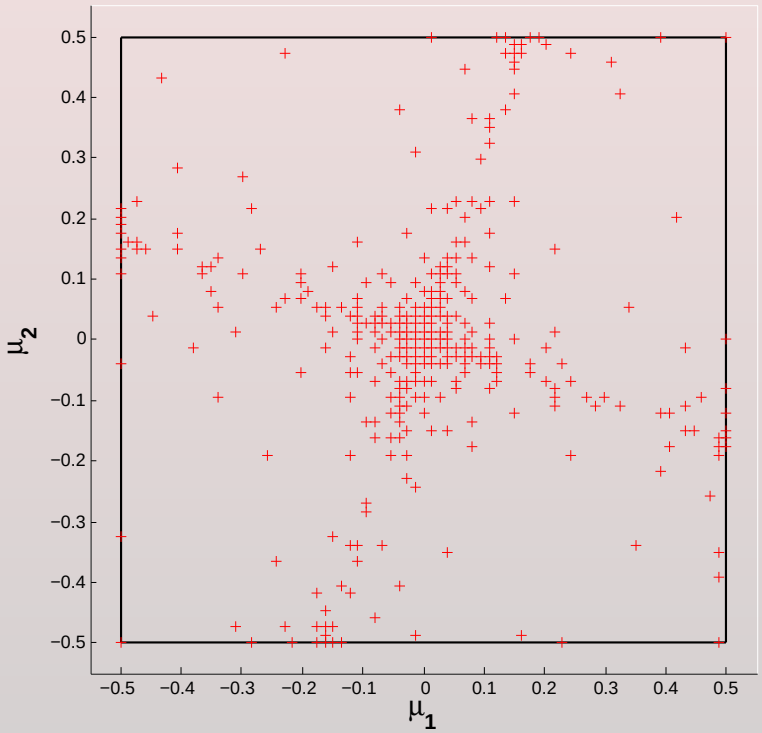
is interesting in the sense that it presents an almost singularity in parameter space at the origin.



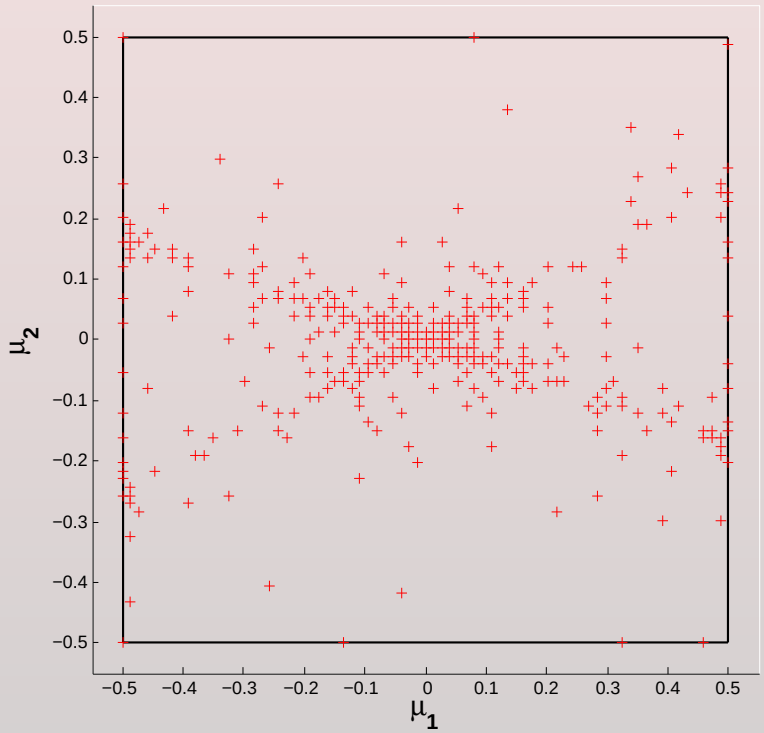
Local approximation spaces (anisotropic metric)



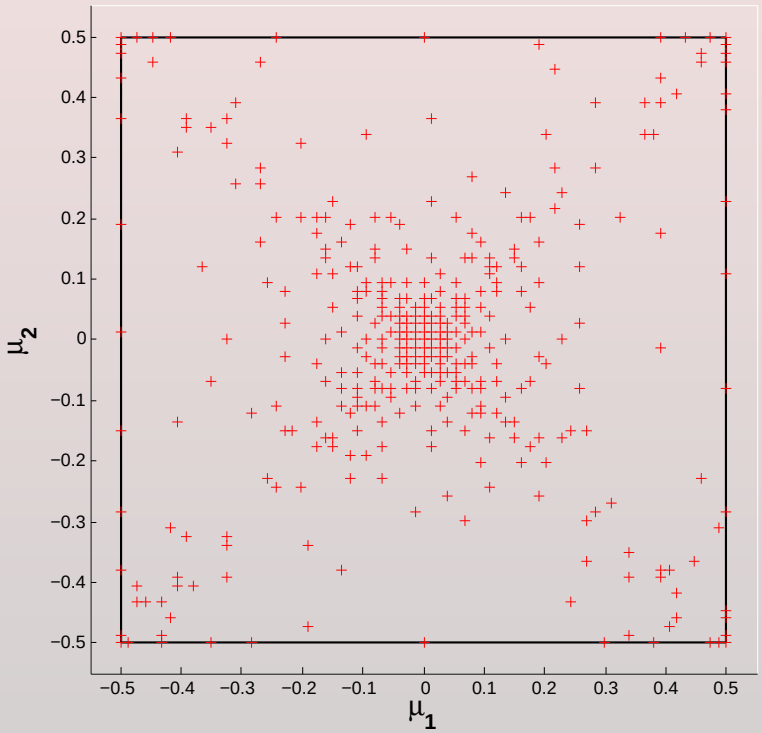
Local approximation spaces (isotropic metric)



Sample points



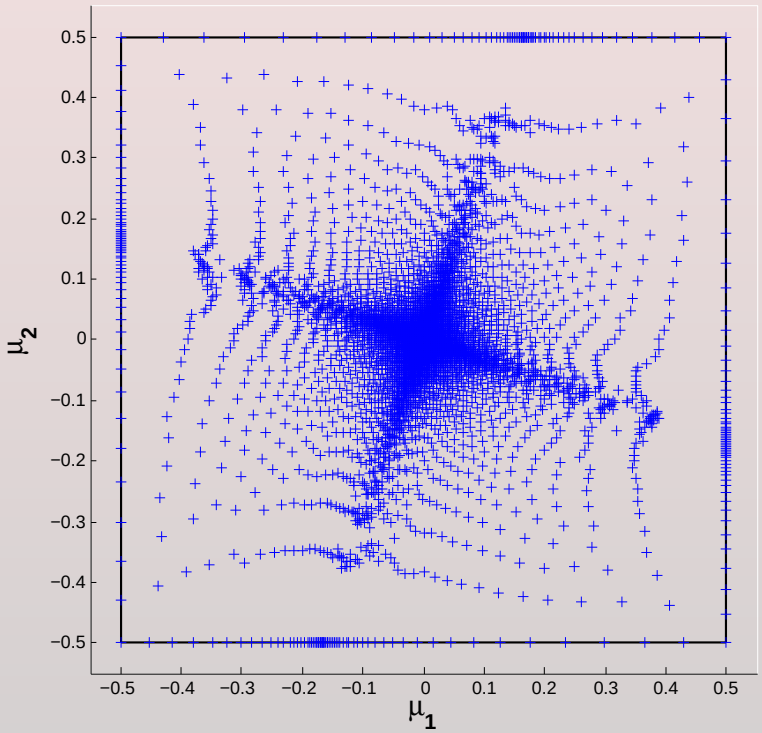
Sample points



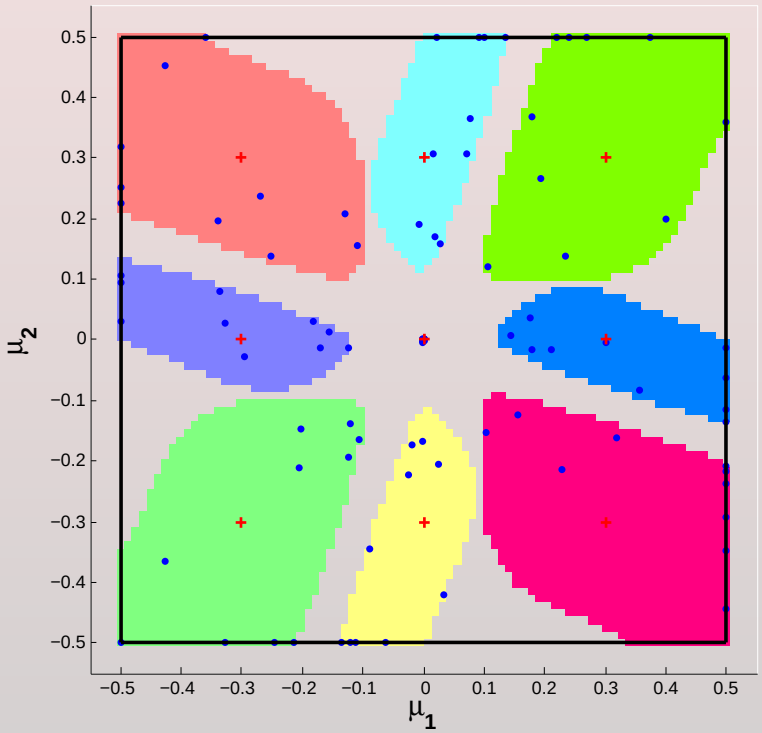
Sample points

Second with adaptation of the trial set $\Xi_{\mathtt{trial}}$

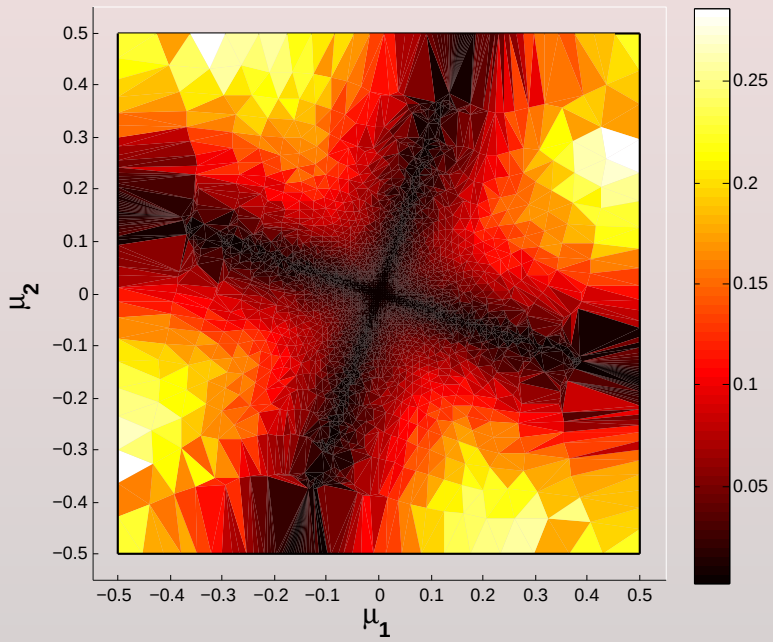
We present only the third case



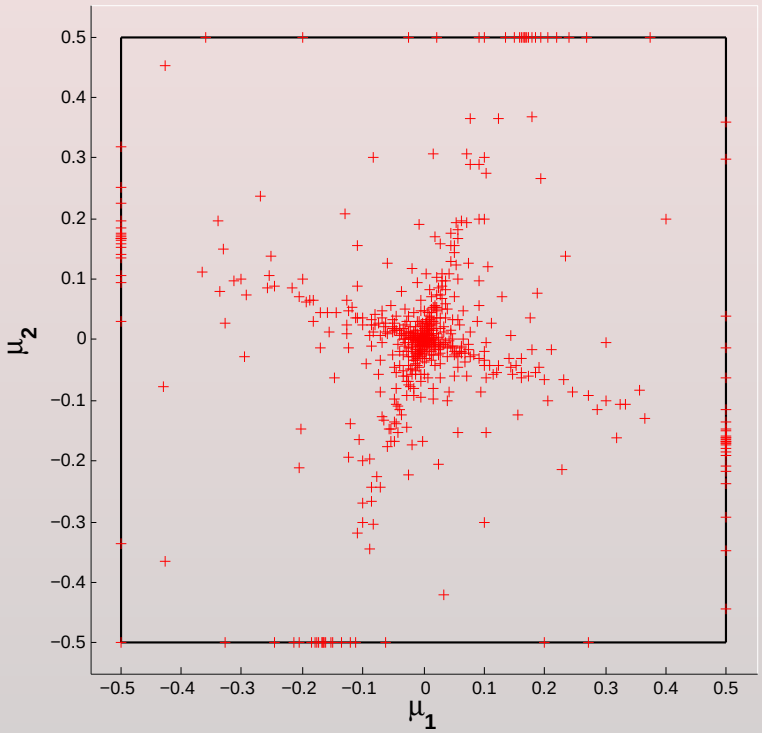
Trial points



Local approximation spaces


0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
μ2
−0.5 −0.4 −0.3 −0.2 −0.1 0 0.1 0.2 0.3 0.4 0.5
μ1
0.25
0.2
0.15
0.1
0.05

Radius



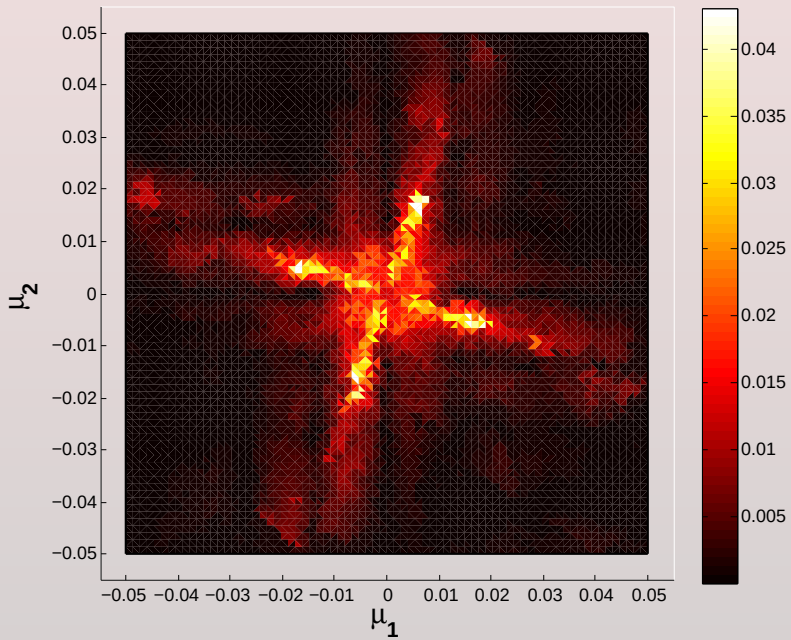
Sample points

In order to illustrate the benefit of using adaptive trial sets also in this case we consider a test sample of 75×75 uniformly distributed points in the region $[-0.05, 0.05]^2$ around the origin.

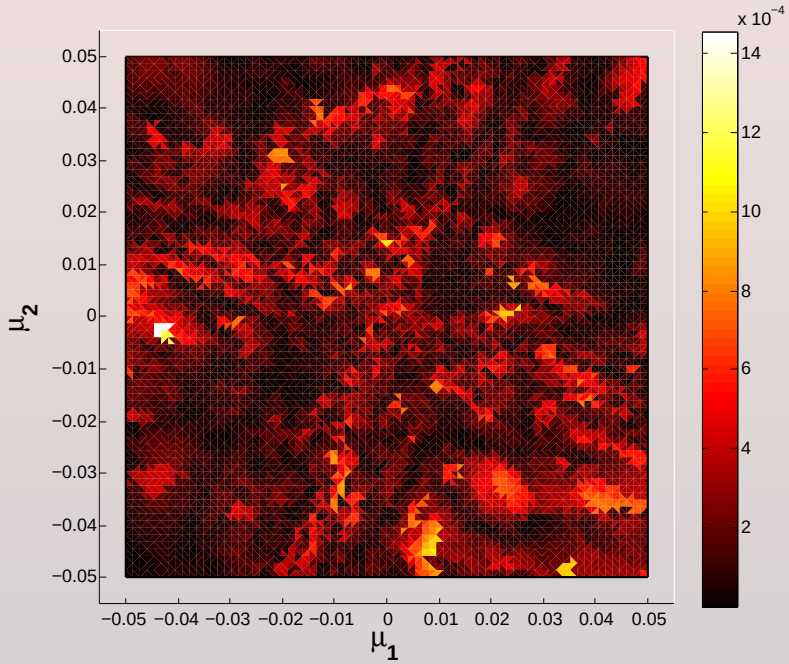
The following figure illustrates the error distribution using the online procedure generated using a fixed and an adaptive trial set. The maximum error is 0.043 resp. 0.00146. While the error tolerance is almost satisfied in the latter case, it is clearly not the case for the former approach.

In order to illustrate the benefit of using adaptive trial sets also in this case we consider a test sample of 75×75 uniformly distributed points in the region $[-0.05, 0.05]^2$ around the origin.

The following figure illustrates the error distribution using the online procedure generated using a fixed and an adaptive trial set. The maximum error is 0.043 resp. 0.00146. While the error tolerance is almost satisfied in the latter case, it is clearly not the case for the former approach.



Fixed trial set



Adaptive trial set

Further, the number of error evaluations is given in the following table :

Number of trial point evaluations during greedy using adapted trial sets :	215'888
Number of trial point evaluations during greedy using fixed trial sets :	241'875

Thus, the adaptive version uses still less error estimator evaluations and is more accurate in the region around the origin. Also, the error is more equally distributed.

Numerical Analysis

A priori Analysis

The convergence to zero of the Kolmogorov n-width $d_n(\mathcal{S}, X)$ is most of the times very fast.

In all cases (either exponential decay or faster [1], or polynomial decay [2]), the greedy procedure provides a sample of parameter values that allow for a suboptimal approximation with respect to the best possible choice hidden in the definition of the n-width.

1. results by Buffa, Maday, Patera, Prud'homme, Turinici
2. results by Binev, Cohen, Dahmen, DeVore, Petrova, and Wojtaszczyk

Numerical Analysis

A priori Analysis

The convergence to zero of the Kolmogorov n-width $d_n(\mathcal{S}, X)$ is most of the times very fast.

In all cases (either exponential decay or faster [1], or polynomial decay [2]), the greedy procedure provides a sample of parameter values that allow for a suboptimal approximation with respect to the best possible choice hidden in the definition of the n-width.

1. results by Buffa, Maday, Patera, Prud'homme, Turinici
2. results by Binev, Cohen, Dahmen, DeVore, Petrova, and Wojtaszczyk

Let us define by $\sigma_n(\mathcal{S}) = \|u(\mu_n) - P_{X_{n-1}}(u(\mu_n))\|_X$. The following allows to state a comparison between $\sigma_n(\mathcal{S})$ and $d_n(\mathcal{S}, X)$
More precisely

- in [Buffa, Maday, Patera, Prud'homme, Turinici], it is proven that

$$\sigma_n(\mathcal{S}) \leq Cn2^n d_n(\mathcal{S}, X)$$

showing that, provided that an exponential fast enough Kolmogorov $n-$width exists, then the greedy algorithm provides also an exponential approximation.
Since most of the time in the application a faster decay rate of the approximation is noticed $d_n(\mathcal{S}, X) \leq Ce^{-\gamma n^\beta}$, this results states that the decay rate obtained by the greedy algorithm is the same.

- this result has been improved in [Binev, Cohen, Dahmen, DeVore, Petrova, Wojtaszczyk] , where it is proven that if e.g. $d_n(\mathcal{S}, X) \leq Cn^{-\alpha}$, then $\sigma_n(\mathcal{S}) \leq C'n^{-\alpha}$

Let us define by $\sigma_n(\mathcal{S}) = \|u(\mu_n) - P_{X_{n-1}}(u(\mu_n))\|_X$. The following allows to state a comparison between $\sigma_n(\mathcal{S})$ and $d_n(\mathcal{S}, X)$
More precisely

- in [Buffa, Maday, Patera, Prud'homme, Turinici], it is proven that

$$\sigma_n(\mathcal{S}) \leq Cn2^n d_n(\mathcal{S}, X)$$

  showing that, provided that an exponential fast enough Kolmogorov $n-$width exists, then the greedy algorithm provides also an exponential approximation.
  Since most of the time in the application a faster decay rate of the approximation is noticed $d_n(\mathcal{S}, X) \leq Ce^{-\gamma n^\beta}$, this results states that the decay rate obtained by the greedy algorithm is the same.

- this result has been improved in [Binev, Cohen, Dahmen, DeVore, Petrova, Wojtaszczyk] , where it is proven that if e.g. $d_n(\mathcal{S}, X) \leq Cn^{-\alpha}$, then $\sigma_n(\mathcal{S}) \leq C'n^{-\alpha}$

Let us define by $\sigma_n(\mathcal{S}) = \|u(\mu_n) - P_{X_{n-1}}(u(\mu_n))\|_X$. The following allows to state a comparison between $\sigma_n(\mathcal{S})$ and $d_n(\mathcal{S}, X)$
More precisely

- in [Buffa, Maday, Patera, Prud'homme, Turinici], it is proven that

$$\sigma_n(\mathcal{S}) \leq Cn2^n d_n(\mathcal{S}, X)$$

  showing that, provided that an exponential fast enough Kolmogorov $n-$width exists, then the greedy algorithm provides also an exponential approximation.
  Since most of the time in the application a faster decay rate of the approximation is noticed $d_n(\mathcal{S}, X) \leq Ce^{-\gamma n^\beta}$, this results states that the decay rate obtained by the greedy algorithm is the same.
- this result has been improved in [Binev, Cohen, Dahmen, DeVore, Petrova, Wojtaszczyk] , where it is proven that if e.g. $d_n(\mathcal{S}, X) \leq Cn^{-\alpha}$, then $\sigma_n(\mathcal{S}) \leq C'n^{-\alpha}$

More recently even [DeVore, Petrova, Wojtaszczyk] proved e.g. that

$$\sigma_{2n}(\mathcal{S}) \leq C\sqrt{d_n(\mathcal{S}, X)}$$

providing the first direct comparison between the sequence $(\sigma_n(\mathcal{S}))_n$ and $(d_n(\mathcal{S}, X))_n$

They also extend it to the Banach (non Hilbertian) framework.

More recently even [DeVore, Petrova, Wojtaszczyk] proved e.g. that

$$\sigma_{2n}(\mathcal{S}) \leq C\sqrt{d_n(\mathcal{S}, X)}$$

providing the first direct comparison between the sequence $(\sigma_n(\mathcal{S}))_n$ and $(d_n(\mathcal{S}, X))_n$

They also extend it to the Banach (non Hilbertian) framework.

More recently even [DeVore, Petrova, Wojtaszczyk] proved e.g. that

$$\sigma_{2n}(\mathcal{S}) \leq C\sqrt{d_n(\mathcal{S}, X)}$$

providing the first direct comparison between the sequence $(\sigma_n(\mathcal{S}))_n$ and $(d_n(\mathcal{S}, X))_n$

They also extend it to the Banach (non Hilbertian) framework.

Parameter dependent solutions.

Basics of the RB method

Now the method . . .

Parameter dependent solutions.

Basics of the RB method

- Let X be a closed subspace of $H^1(\Omega)$.
  Find $u(\mu) \in X$ solution to the following variational problem :

$$\forall v \in X, \qquad a(u(\mu), v; \mu) = L(v; \mu) \tag{3}$$

  where a is a continuous, elliptic bilinear form over X depending on one or many parameters μ that belong to $\mathbb{P} \subset \mathbb{R}^M$.

- The reduced basis method relies on the hypothesis that, for any $\varepsilon > 0$, there exists a set of parameters $\mu_1, \mu_2, ..., \mu_N$ in $\mathbb{P}$, well chosen, such that

$$\forall \mu \in \mathbb{P}, \quad \exists \{\alpha_i(\mu)\}_i, \qquad \|u(\mu) - \sum_{i=1}^{N(\varepsilon)} \alpha_i(\mu) u(\mu_i)\|_X \leq \varepsilon.$$

Note that $\varepsilon = \varepsilon(N)$

Parameter dependent solutions.

Basics of the RB method

- Let X be a closed subspace of $H^1(\Omega)$.
  Find $u(\mu) \in X$ solution to the following variational problem :

$$\forall v \in X, \qquad a(u(\mu), v; \mu) = L(v; \mu) \tag{3}$$

  where a is a continuous, elliptic bilinear form over X depending on one or many parameters μ that belong to $\mathbb{P} \subset \mathbb{R}^M$.

- The reduced basis method relies on the hypothesis that, for any $\varepsilon > 0$, there exists a set of parameters $\mu_1, \mu_2, ..., \mu_N$ in $\mathbb{P}$, well chosen, such that

$$\forall \mu \in \mathbb{P}, \quad \exists \{\alpha_i(\mu)\}_i, \qquad \|u(\mu) - \sum_{i=1}^{N(\varepsilon)} \alpha_i(\mu) u(\mu_i)\|_X \leq \varepsilon.$$

  Note that $\varepsilon = \varepsilon(N)$

Parameter dependent solutions.

Reduced Approximation

- In a way or another, let X^N be given
- The reduced approximation consists in solving the Galerkin problem : Find $u^N(\mu) \in X^N$ such that :

$$\forall v^N \in X^N, \qquad a(u^N(\mu), v^N; \mu) = L(v^N) \tag{4}$$

- Cea's lemma states an upper bound

$$\|u(\mu) - u^N(\mu)\|_X \leq c \inf_{v \in X^N} \|u(\mu) - v\|_X$$

and thus, by using hypothesis (2), we get :

$$\|u(\mu) - u^N(\mu)\|_X \leq c\varepsilon(N). \tag{5}$$

Application to Maxwell.... with Y. Chen, J. Hesthaven, J. Rodriguez and X Zhu

We consider here a 2D problem on electromagnetism. In both cases, the electromagnetic waves are TM-polarized, that is, the electric and magnetic fields satisfy $E = (0, 0, E_z)$ and $H = (H_x, H_y, 0)$ in the Maxwell's equation. The problem is set on the domains

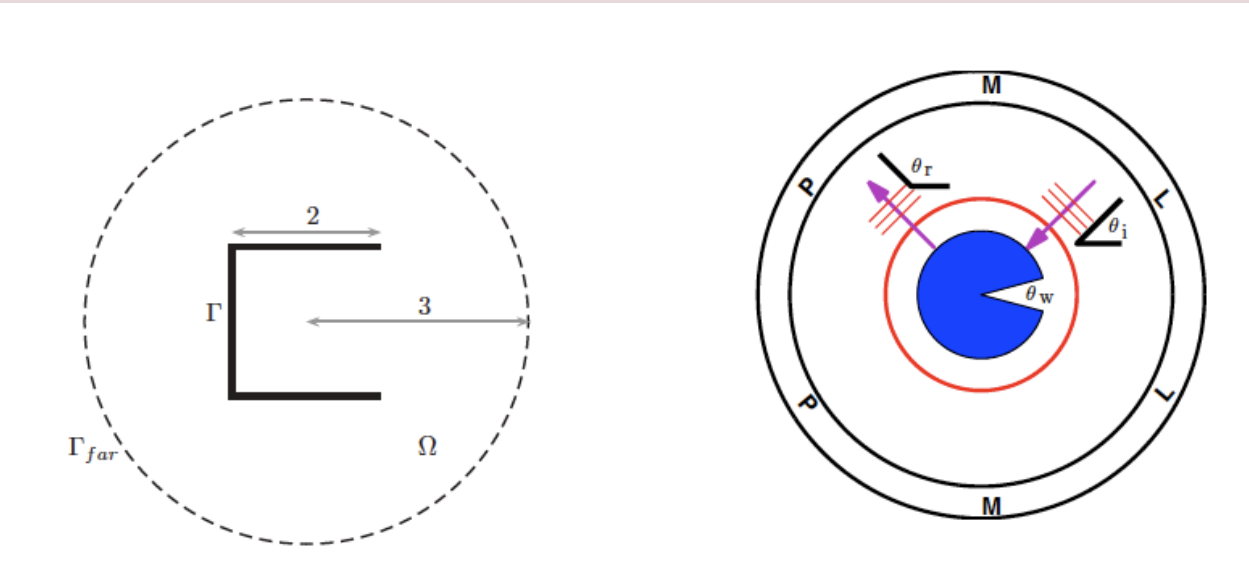


FIGURE: Geometries of the problems.

Application to Maxwell....

The coefficients that will determine the physics of the experiment are :

- the angle of incidence,
- the angle of measurement
- the angular frequency
- the angle of the wedge, leading to a modification of the geometry.

The output of interest is the radar cross section.

Application to Maxwell....

The equations of interest are

$$
\begin{aligned}
i\omega E_z &= \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} && \text{in } \Omega, \\
i\omega H_x &= -\frac{\partial E_z}{\partial y} && \text{in } \Omega, \\
i\omega H_y &= \frac{\partial E_z}{\partial x} && \text{in } \Omega, \\
E_z &= -E_{inc} && \text{on } \Gamma,
\end{aligned}
$$

Application to Maxwell....

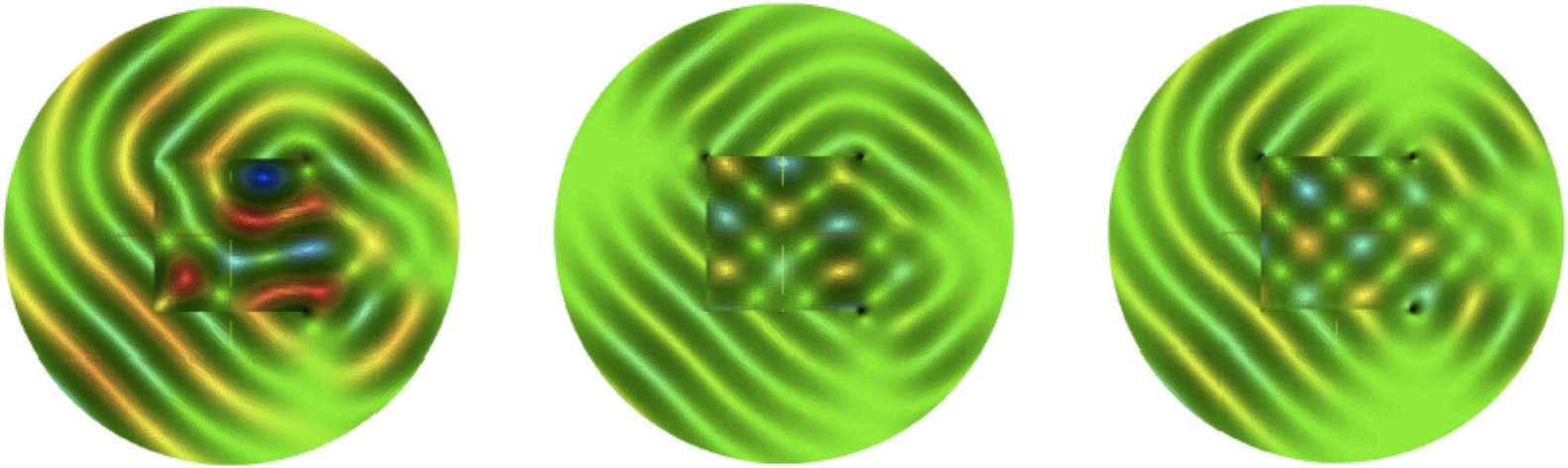


FIG. 3.1. *Solution of the problem on the open cavity (from left to right: E_z, H_x and H_y) for $\omega = 2\pi$ and $\theta = 5\pi/4$.*

Application to Maxwell....

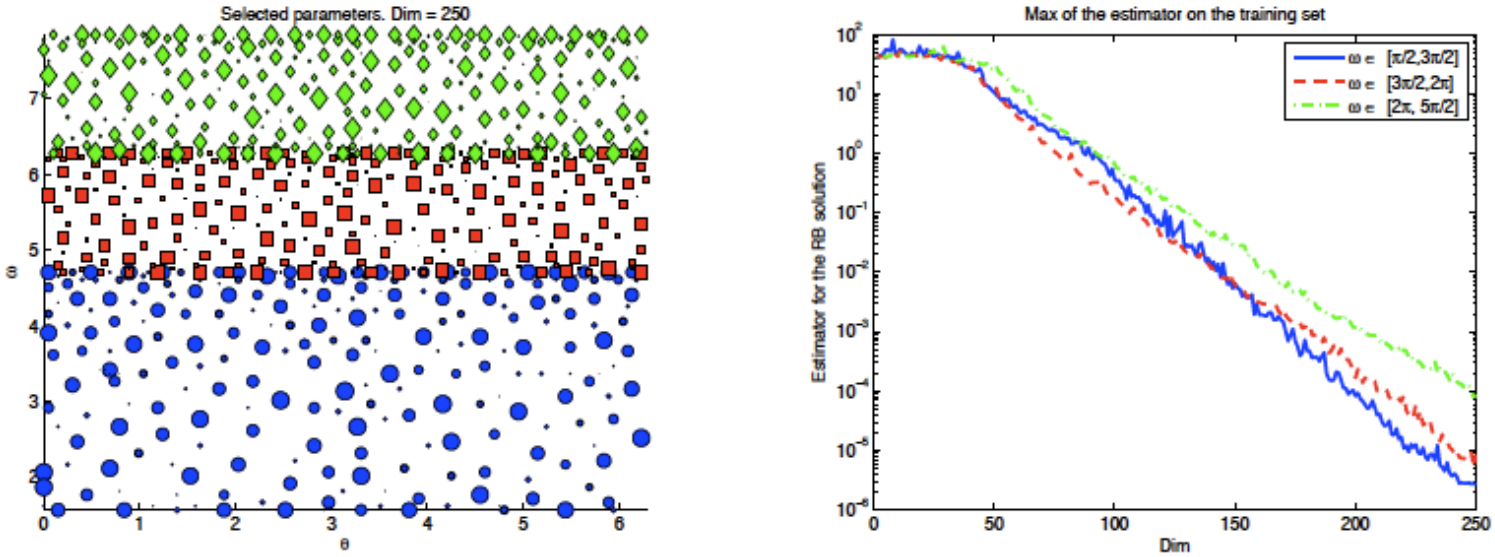


FIG. 3.5. *Construction of the reduced basis when* (θ, ω) *is in* $[0, 2\pi] \times [\pi/2, 3\pi/2]$ *(in blue),* $[0, 2\pi] \times [3\pi/2, 2\pi]$ *(in red) and* $[0, 2\pi] \times [2\pi, 5\pi/2]$ *(in green). On the left are the parameters selected by RBM (the larger the markers the earlier they have been chosen). On the right is the evolution of the maximum over the train sets* Ξ_i^{train}, $i \in \{1, 2, 3\}$ *of the a posteriori error estimator when the dimension of the reduced basis is increased.*

Application to Maxwell.... 1 parameter (angle)

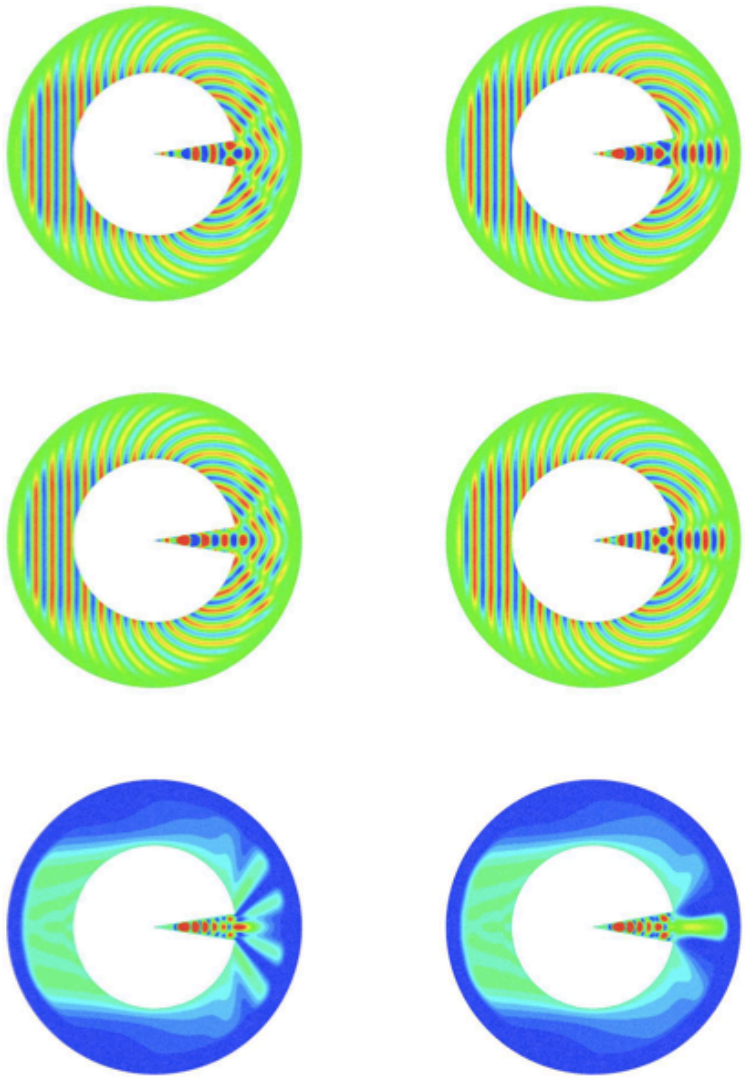


FIG. 3.13. *Real part (top), imaginary part (middle), and module (bottom) of the electric field with* $\theta_W = 18.5°$ *(left) and* $21.5°$ *(right).*

Application to Maxwell.... 1 parameter (angle)

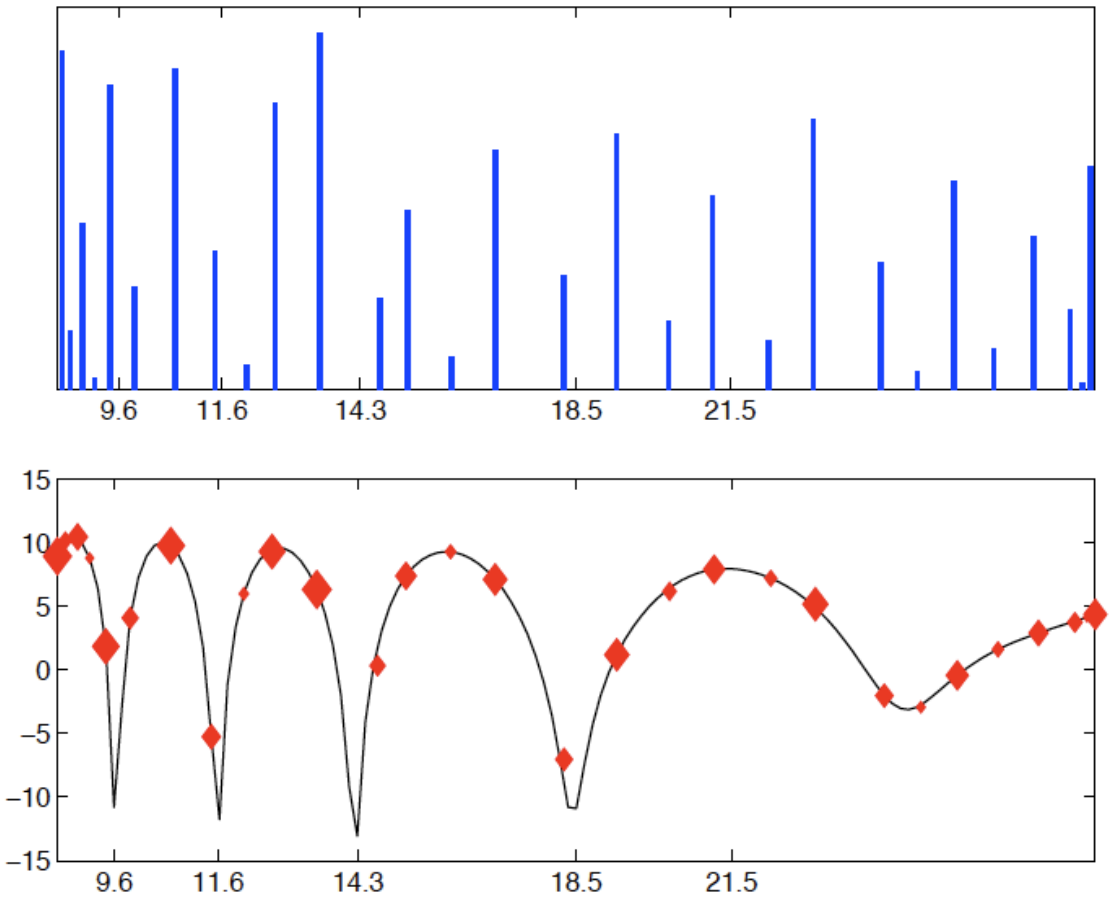


FIG. 3.14. *The 29 θ_W's the greedy algorithm of the RBM picks to build the RB space. Top: the higher the vertical line, the earlier that point was picked. Bottom: the points scattered on the monostatic scattering curve $10log_{10}(RCS(10\pi, \theta_W, 0, 0))$, the larger the marker, the earlier it is selected.*

Application to Maxwell.... 1 parameter (angle)

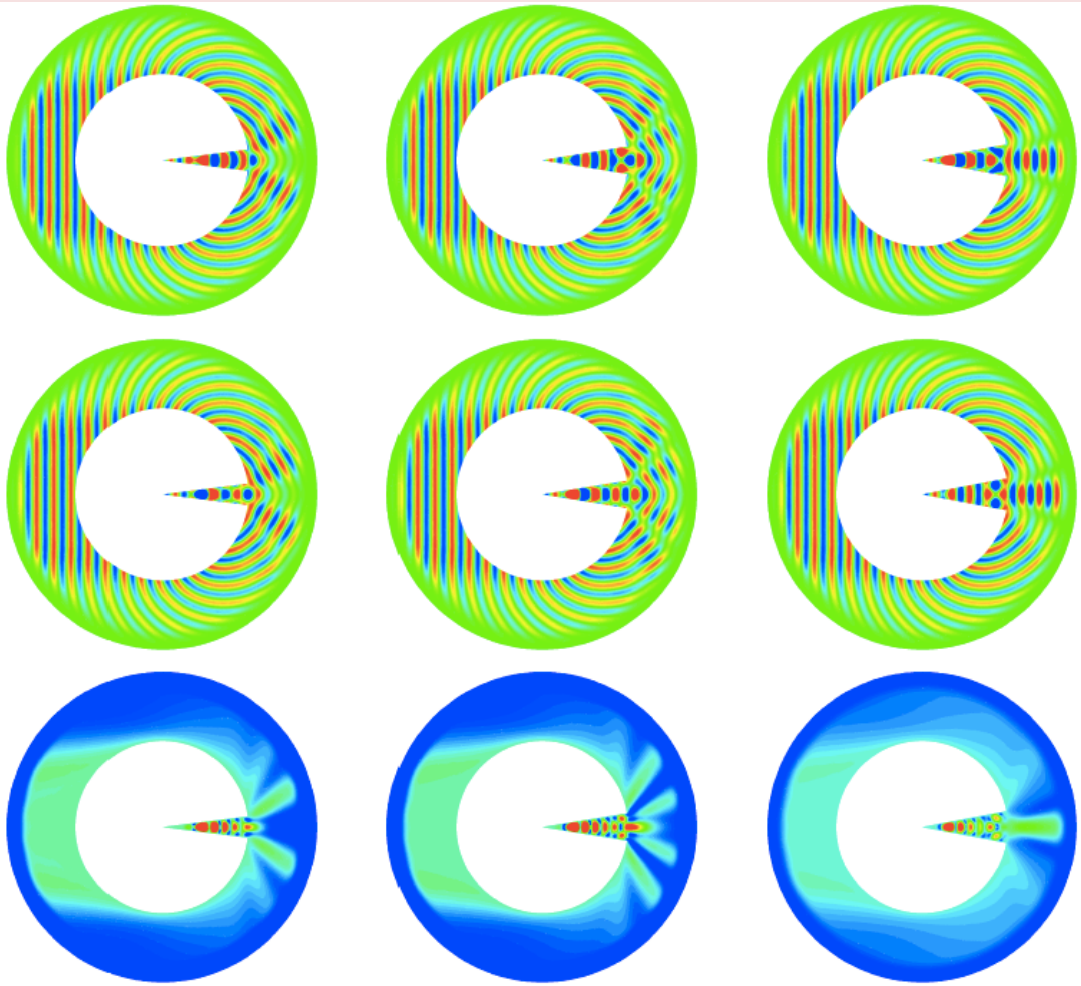


Figure 20: Real part (top), imaginary part (middle), and module (bottom) of the electric field with $\theta_W = 14.3°$ (left), $18.5°$ (middle) and $21.5°$ (right).

Application to Maxwell.... 1 parameter (angle)

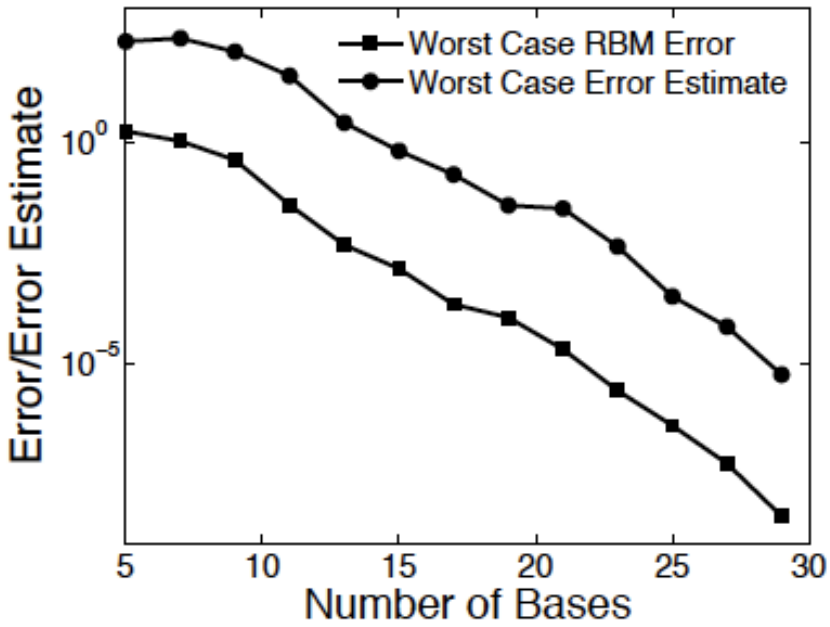
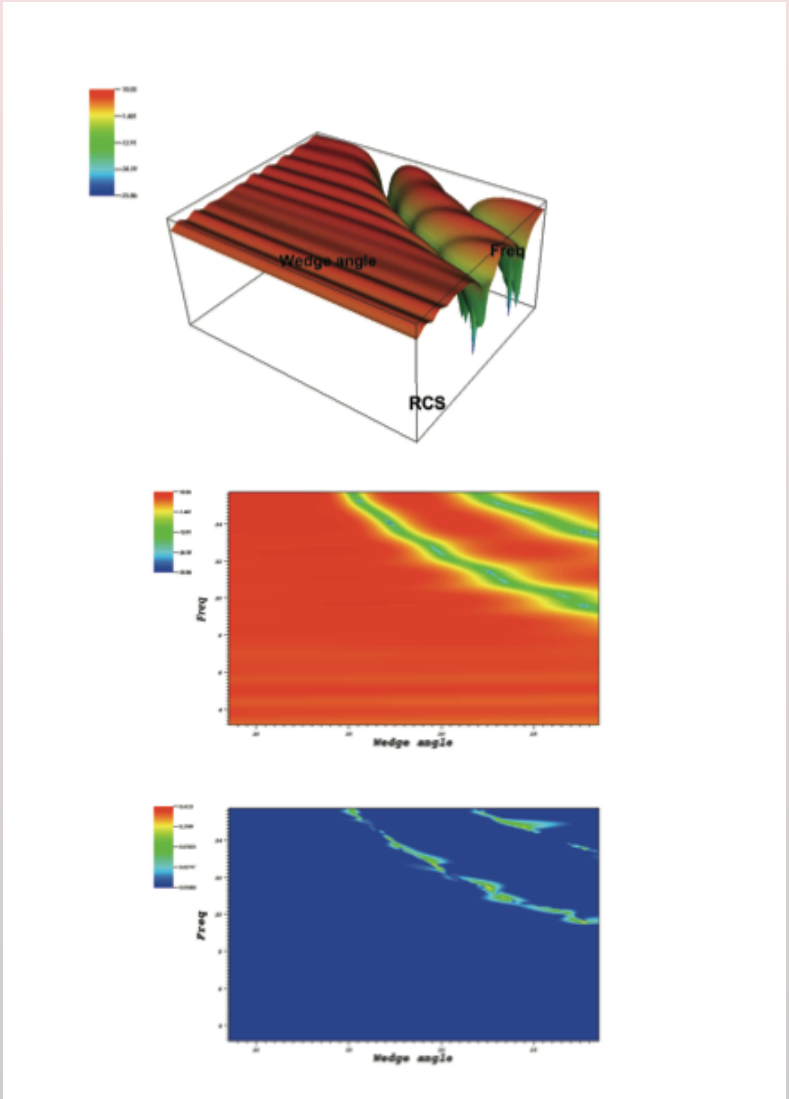


FIG. 3.15. *The worst case convergence history and the corresponding error estimate of the RBM for* 120 *randomly selected parameter values.*

Application to Maxwell.... 2 parameters (angle and frequency)



Numerical Analysis

A posteriori Analysis

There are many contributions here, they require the evaluation of ellipticity or inf sup constants

These are fundamental for an efficient offline construction of the RB

...but you can use POD ...or random

These are also necessary to certify the results ...but you can always do a truth approximation at the end ! !

Numerical Analysis

A posteriori Analysis

There are many contributions here, they require the evaluation of ellipticity or inf sup constants

These are fundamental for an efficient offline construction of the RB

. . .but you can use POD . . .or random

These are also necessary to certify the results . . .but you can always do a truth approximation at the end ! !

Numerical Analysis

A posteriori Analysis

There are many contributions here, they require the evaluation of ellipticity or inf sup constants

These are fundamental for an efficient offline construction of the RB

. . .but you can use POD . . .or random

These are also necessary to certify the results . . .but you can always do a truth approximation at the end ! !

Numerical Analysis

A posteriori Analysis

There are many contributions here, they require the evaluation of ellipticity or inf sup constants

These are fundamental for an efficient offline construction of the RB

. . .but you can use POD . . .or random

These are also necessary to certify the results . . .but you can always do a truth approximation at the end ! !

Numerical Analysis

A posteriori Analysis

There are many contributions here, they require the evaluation of ellipticity or inf sup constants

These are fundamental for an efficient offline construction of the RB . . .but you can use POD . . .or random

These are also necessary to certify the results . . .but you can always do a truth approximation at the end ! !

Numerical Analysis

A posteriori Analysis

There are many contributions here, they require the evaluation of ellipticity or inf sup constants

These are fundamental for an efficient offline construction of the RB ...but you can use POD ...or random

These are also necessary to certify the results ...but you can always do a truth approximation at the end ! !

Numerical Analysis

A posteriori Analysis

There are many contributions here, they require the evaluation of ellipticity or inf sup constants

These are fundamental for an efficient offline construction of the RB ...but you can use POD ...or random

These are also necessary to certify the results ...but you can always do a truth approximation at the end ! !

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods get their accuracy from the regularity of the solution. By using global basis the convergence is very fast.

Here the regularity (in space) is replaced by the small Kolmogorov width

Regularity (as a function of the parameter) implies small Kolmogorov width but is not equivalent

Small Kolmogorov width implies that the discrete space CAN BE COMPOSED of well selected solutions (greedy algorithm)

A posteriori estimators are fundamental in order to compute only useful solutions

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods get their accuracy from the regularity of the solution. By using global basis the convergence is very fast.

Here the regularity (in space) is replaced by the small Kolmogorov width

Regularity (as a function of the parameter) implies small Kolmogorov width but is not equivalent

Small Kolmogorov width implies that the discrete space CAN BE COMPOSED of well selected solutions (greedy algorithm)

A posteriori estimators are fundamental in order to control only the useful solutions

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods get their accuracy from the regularity of the solution. By using global basis the convergence is very fast.

Here the regularity (in space) is replaced by the small Kolmogorov width

Regularity (as a function of the parameter) implies small Kolmogorov width but is not equivalent

Small Kolmogorov width implies that the discrete space CAN BE COMPOSED of well selected solutions (greedy algorithm)

A posteriori estimators are fundamental in order to compute only the best solutions

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods get their accuracy from the regularity of the solution. By using global basis the convergence is very fast.

Here the regularity (in space) is replaced by the small Kolmogorov width

Regularity (as a function of the parameter) implies small Kolmogorov width but is not equivalent

Small Kolmogorov width implies that the discrete space CAN BE COMPOSED of well selected solutions (greedy algorithm)

A posteriori estimators : are fundamental in order to compute only useful solutions

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods get their accuracy from the regularity of the solution. By using global basis the convergence is very fast.

Here the regularity (in space) is replaced by the small Kolmogorov width

Regularity (as a function of the parameter) implies small Kolmogorov width but is not equivalent

Small Kolmogorov width implies that the discrete space CAN BE COMPOSED of well selected solutions (greedy algorithm)

A posteriori estimators : are fundamental in order to compute only THOSE solutions

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods get their accuracy from the regularity of the solution. By using global basis the convergence is very fast.

Here the regularity (in space) is replaced by the small Kolmogorov width

Regularity (as a function of the parameter) implies small Kolmogorov width but is not equivalent

Small Kolmogorov width implies that the discrete space CAN BE COMPOSED of well selected solutions (greedy algorithm)

A posteriori estimators : are fundamental in order to compute only THOSE solutions

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods get their accuracy from the regularity of the solution. By using global basis the convergence is very fast.

Here the regularity (in space) is replaced by the small Kolmogorov width

Regularity (as a function of the parameter) implies small Kolmogorov width but is not equivalent

Small Kolmogorov width implies that the discrete space CAN BE COMPOSED of well selected solutions (greedy algorithm)

A posteriori estimators : are fundamental in order to compute only THOSE solutions

Motivation

Among the most prominent applications in medical applications is the analysis of internal flows

- blood flows in arteries
- air flow in the lung

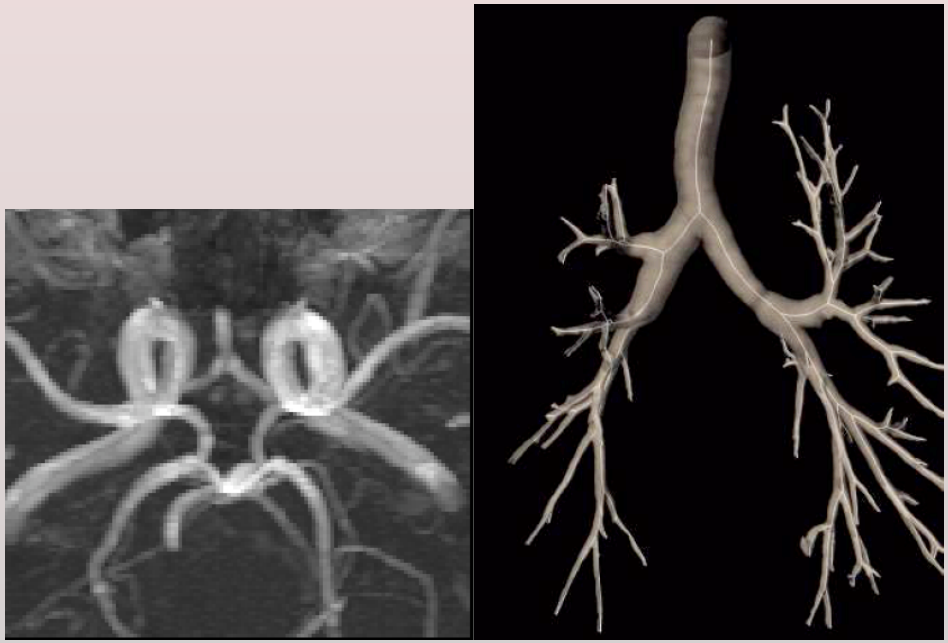


FIGURE: Reconstructed geometries of Willis complex (Thiriet) and of the upper part of the lung (Fetita-Prêteux)

Motivation

Among the most prominent applications in medical applications is the analysis of internal flows

- blood flows in arteries
- air flow in the lung

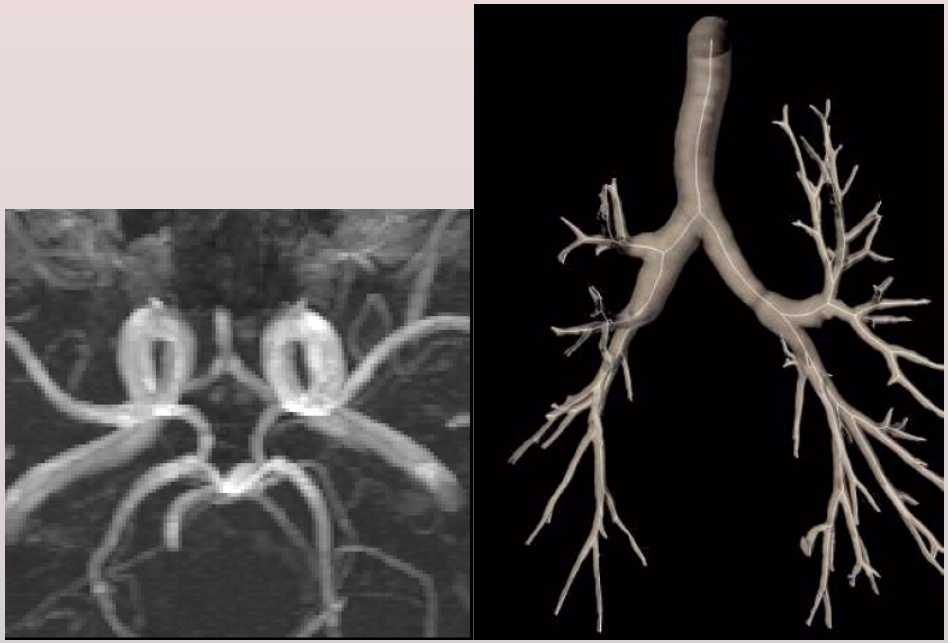


FIGURE: Reconstructed geometries of Willis complex (Thiriet) and of the upper part of the lung (Fetita-Prêteux)

Motivation

Among the most prominent applications in medical applications is the analysis of internal flows

- blood flows in arteries
- air flow in the lung

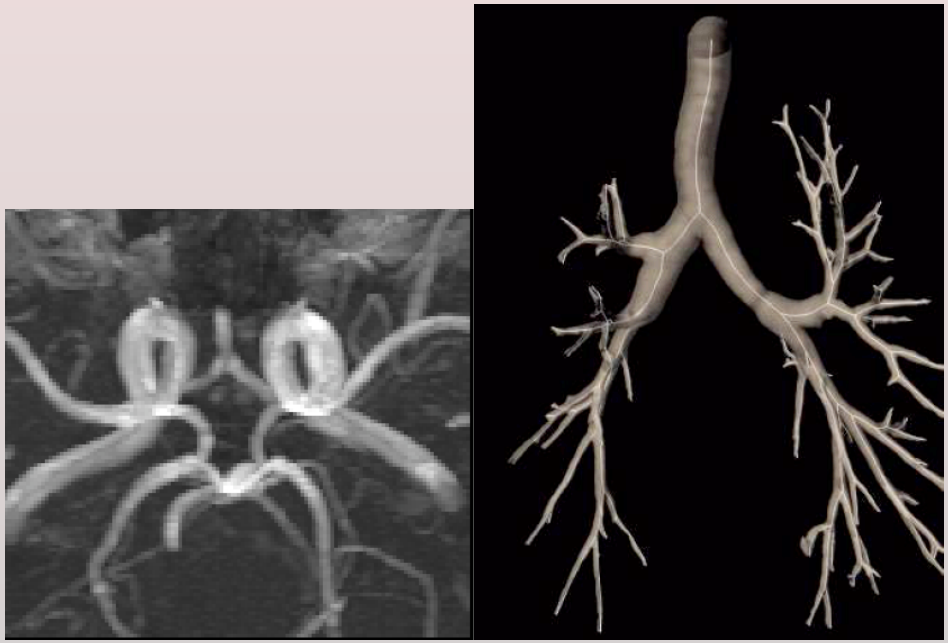


FIGURE: Reconstructed geometries of Willis complex (Thiriet) and of the upper part of the lung (Fetita-Prêteux)

Motivation

Among the most prominent applications in medical applications is the analysis of internal flows

- blood flows in arteries
- air flow in the lung

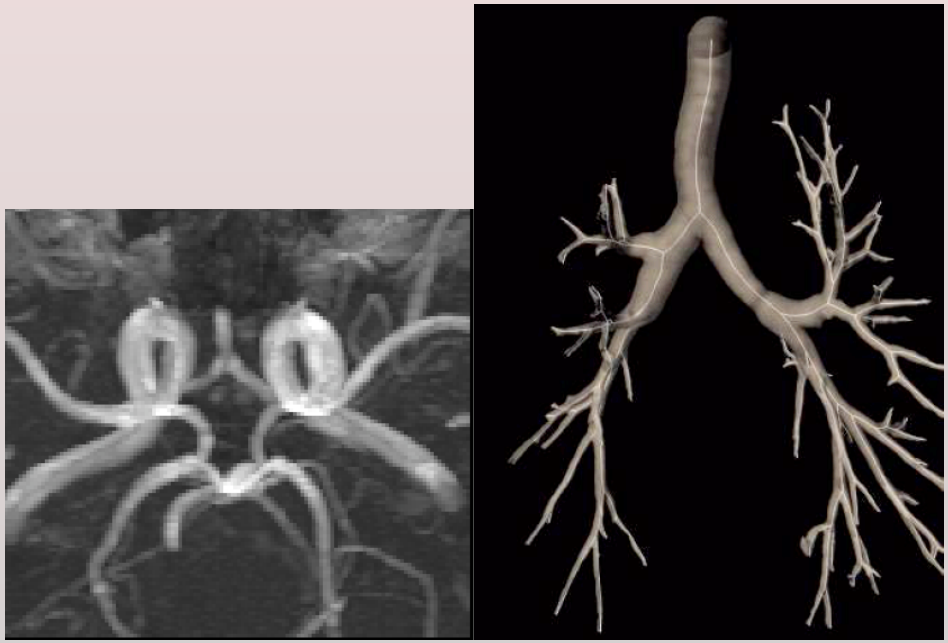


FIGURE: Reconstructed geometries of Willis complex (Thiriet) and of the upper part of the lung (Fetita-Prêteux)

Motivation

- In this range of applications, the challenge of the simulations comes more from the complexity of the geometry
- There is some repetitiveness or similarities in the behavior of the flow that allows for the definition of reduced model strategies.

Motivation

- In this range of applications, the challenge of the simulations comes more from the complexity of the geometry
- There is some repetitiveness or similarities in the behavior of the flow that allows for the definition of reduced model strategies.

Motivation

The reduced basis element method

- Reduced basis approximation
- domain decomposition

Motivation

The reduced basis element method

- Reduced basis approximation
- domain decomposition

Motivation

The reduced basis element method

- Reduced basis approximation

+

- domain decomposition

Motivation

The reduced basis element method

- Reduced basis approximation

+

- domain decomposition

Reduced element method

The domain of interest is first decomposed into several subdomains,

$$\overline{\Omega} = \cup_{k=1}^{K} \overline{\Omega_k^{bb}}$$

where each "**b**uilding **b**lock" Ω_k^{bb} is assumed to be the image of a reference $\hat{\Omega}$.

The mapping φ_k between $\hat{\Omega}$ and Ω_k^{bb} will be assume to be piecewise affine (and obviouly continuous) so that

$$\Omega_k^{bb} = \varphi_k[\hat{\Omega}]$$

Reduced element method

The domain of interest is first decomposed into several subdomains,

$$\overline{\Omega} = \cup_{k=1}^{K} \overline{\Omega_k^{bb}}$$

where each "**b**uilding **b**lock" Ω_k^{bb} is assumed to be the image of a reference $\hat{\Omega}$.

The mapping φ_k between $\hat{\Omega}$ and Ω_k^{bb} will be assume to be piecewise affine (and obviouly continuous) so that

$$\Omega_k^{bb} = \varphi_k[\hat{\Omega}]$$

Reduced element method

The domain of interest is first decomposed into several subdomains,

$$\overline{\Omega} = \cup_{k=1}^{K} \overline{\Omega_k^{bb}}$$

where each "**b**uilding **b**lock" Ω_k^{bb} is assumed to be the image of a reference $\hat{\Omega}$.

The mapping φ_k between $\hat{\Omega}$ and Ω_k^{bb} will be assume to be piecewise affine (and obviouly continuous) so that

$$\Omega_k^{bb} = \varphi_k[\hat{\Omega}]$$

Reduced element method

The domain of interest is first decomposed into several subdomains,

$$\overline{\Omega} = \cup_{k=1}^{K} \overline{\Omega_k^{bb}}$$

where each "**b**uilding **b**lock" Ω_k^{bb} is assumed to be the image of a reference $\hat{\Omega}$.

The mapping φ_k between $\hat{\Omega}$ and Ω_k^{bb} will be assume to be piecewise affine (and obviouly continuous) so that

$$\Omega_k^{bb} = \varphi_k[\hat{\Omega}]$$

Reduced element method

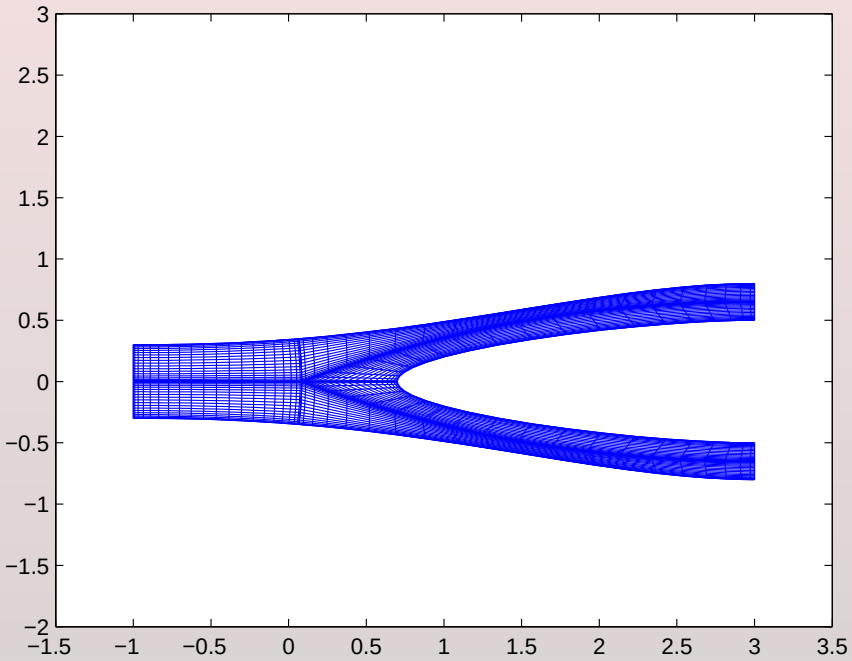


FIGURE: A first deformation of a building block.

Reduced element method

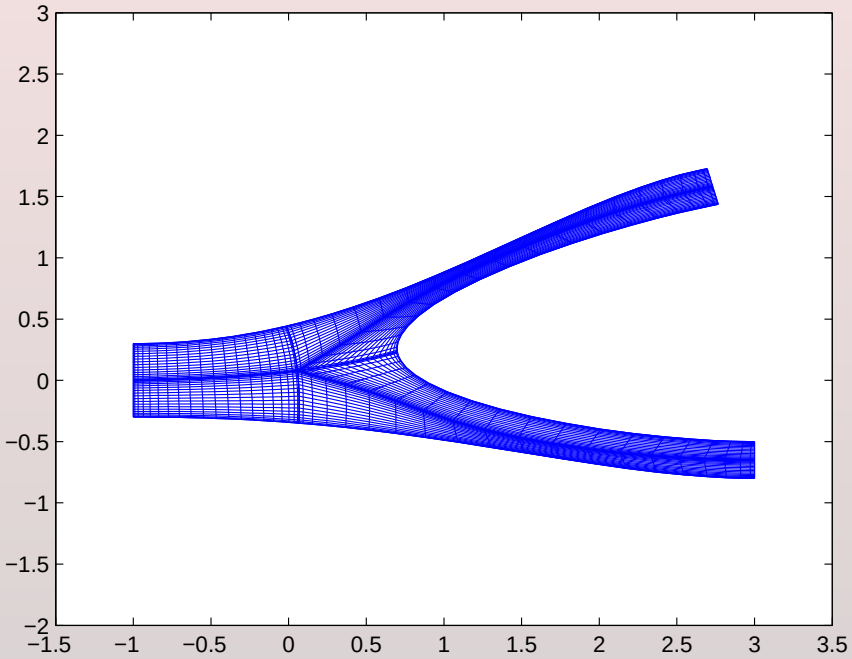


FIGURE: A second deformation of a building block.

Reduced element method

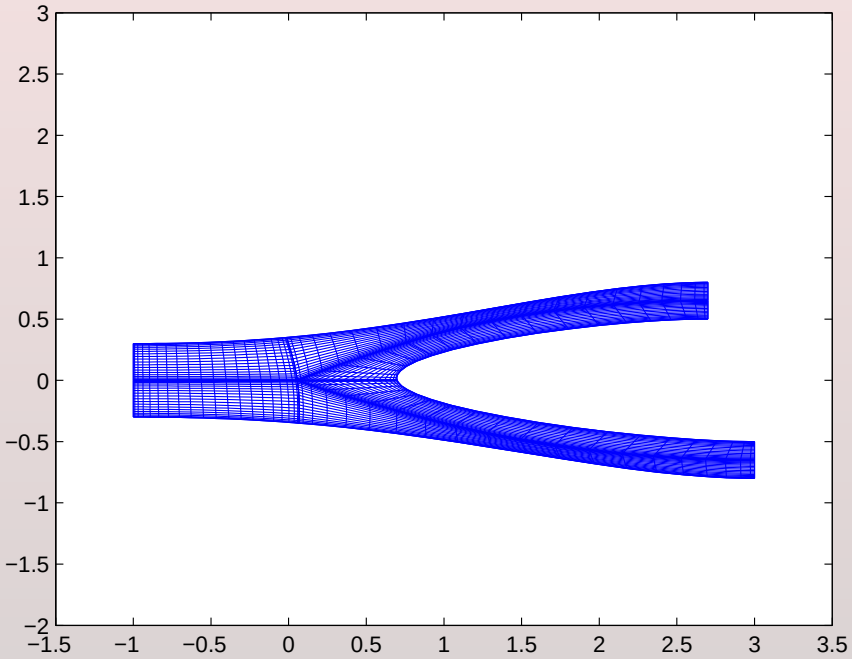


FIGURE: A third deformation of a building block.

Reduced element method

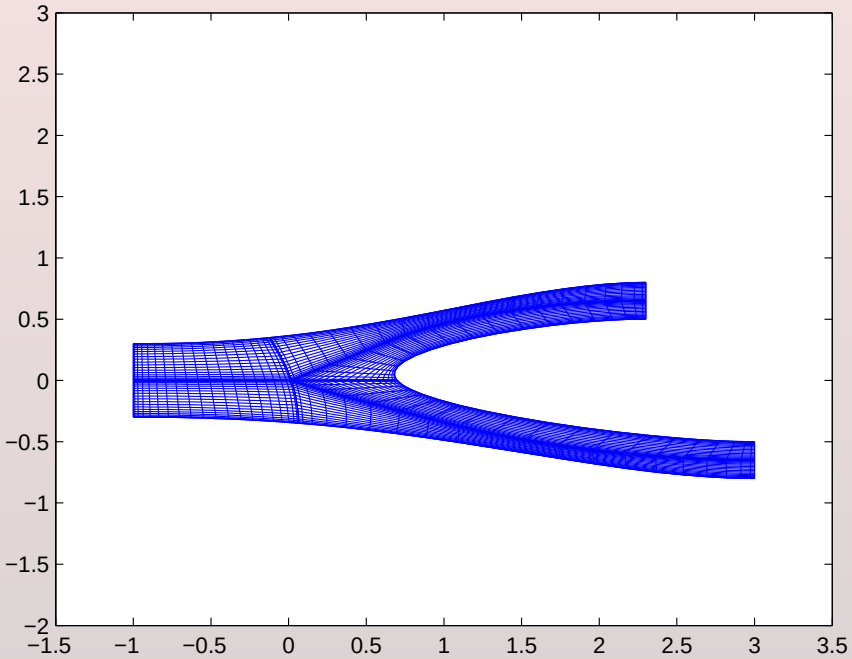


FIGURE: A fourth deformation of a building block.

Reduced element method

As a precomputation, the problem of interest is solved over various deformations of each reference building block and stored, after mapping, on the reference building block.

This gives basis functions $\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N$, supposed to be linearly independent

These basics solutions are mapped over each Ω_k^{bb} through φ_k.

The solution corresponding to an unknown, deformed geometry is then represented as a linear combination of these mapped solutions

$$X_N = \{ v_N \in L^2(\Omega) | \quad v_{N|\Omega_k^{bb}} \circ \varphi_k \in \text{span}\{\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N\} \} .$$

The discrete problem then reads : Find u_N in X_N such that

$$a(u_N, v_N) = f(v_N) \quad \forall v_N \in X_N$$

Reduced element method

As a precomputation, the problem of interest is solved over various deformations of each reference building block and stored, after mapping, on the reference building block.
This gives basis functions $\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N$, supposed to be linearly independent

These basics solutions are mapped over each Ω_k^{bb} through φ_k.
The solution corresponding to an unknown, deformed geometry is then represented as a linear combination of these mapped solutions

$$X_N = \{ v_N \in L^2(\Omega) | \quad v_{N|\Omega_k^{bb}} \circ \varphi_k \in \text{span}\{\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N\} \} .$$

The discrete problem then reads : Find u_N in X_N such that

$$a(u_N, v_N) = f(v_N), \quad \forall v_N \in X_N$$

Reduced element method

As a precomputation, the problem of interest is solved over various deformations of each reference building block and stored, after mapping, on the reference building block.
This gives basis functions $\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N$, supposed to be linearly independent
These basics solutions are mapped over each Ω_k^{bb} through φ_k.
The solution corresponding to an unknown, deformed geometry is then represented as a linear combination of these mapped solutions

$$X_N = \{ v_N \in L^2(\Omega) | \quad v_{N|\Omega_k^{bb}} \circ \varphi_k \in \text{span}\{\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N\} \} .$$

The discrete problem then reads : Find u_N in X_N such that

$$a(v_N, u_N) = f(v_N), \quad \forall v_N \in X_N$$

Reduced element method

As a precomputation, the problem of interest is solved over various deformations of each reference building block and stored, after mapping, on the reference building block.
This gives basis functions $\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N$, supposed to be linearly independent
These basics solutions are mapped over each Ω_k^{bb} through φ_k.
The solution corresponding to an unknown, deformed geometry is then represented as a linear combination of these mapped solutions

$$X_N = \{ v_N \in L^2(\Omega) | \quad v_{N|\Omega_k^{bb}} \circ \varphi_k \in \text{span}\{\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N\} \} .$$

The discrete problem then reads : Find u_N in X_N such that

$$a(u_N, v_N) = f(v_N), \quad \forall v_N \in X_N .$$

Reduced element method

As a precomputation, the problem of interest is solved over various deformations of each reference building block and stored, after mapping, on the reference building block.
This gives basis functions $\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N$, supposed to be linearly independent
These basics solutions are mapped over each Ω_k^{bb} through φ_k.
The solution corresponding to an unknown, deformed geometry is then represented as a linear combination of these mapped solutions

$$X_N = \{ v_N \in L^2(\Omega) | \quad v_{N|\Omega_k^{bb}} \circ \varphi_k \in \text{span}\{\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N\} \} \ .$$

The discrete problem then reads : Find u_N in X_N such that

$$a(u_N, v_N) = f(v_N), \quad \forall v_N \in X_N \ .$$

Reduced element method

As a precomputation, the problem of interest is solved over various deformations of each reference building block and stored, after mapping, on the reference building block.
This gives basis functions $\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N$, supposed to be linearly independent
These basics solutions are mapped over each Ω_k^{bb} through φ_k.
The solution corresponding to an unknown, deformed geometry is then represented as a linear combination of these mapped solutions

$$X_N = \{ v_N \in L^2(\Omega) | \quad v_{N|\Omega_k^{bb}} \circ \varphi_k \in \text{span}\{\hat{\zeta}_1, \hat{\zeta}_2, ..., \hat{\zeta}_N\} \} \ .$$

The discrete problem then reads : Find u_N in X_N such that

$$a(u_N, v_N) = f(v_N), \quad \forall v_N \in X_N \ .$$

Fluid flows

For flow problems the tranformations between the reference domain and the subdomains are more involved : The PIOLA Transform that allow the work with divergence free discrete spaces

$$\hat{u} = \mathcal{J}^{-1}(u \circ \Phi)|J|,$$

$\longrightarrow$ The velocity is computed independently of the pressure

Fluid flows

For flow problems the tranformations between the reference domain and the subdomains are more involved : The PIOLA Transform that allow the work with divergence free discrete spaces

$$\hat{u} = \mathcal{J}^{-1}(u \circ \Phi)|J|,$$

$\longrightarrow$ The velocity is computed independently of the pressure

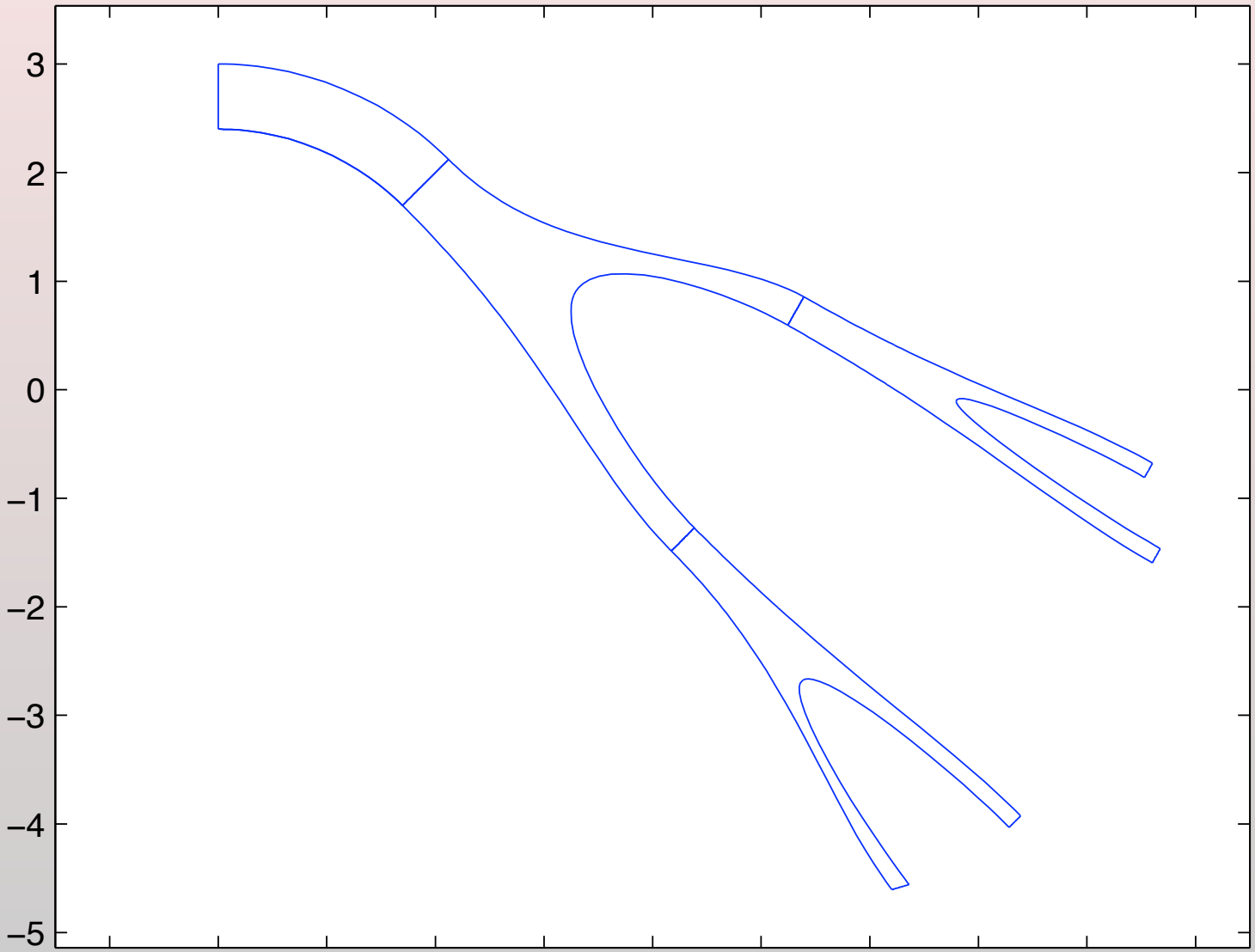
Fluid flows

For flow problems the tranformations between the reference domain and the subdomains are more involved : The PIOLA Transform that allow the work with divergence free discrete spaces

$$\hat{u} = \mathcal{J}^{-1}(u \circ \Phi)|J|,$$

$\longrightarrow$ The velocity is computed independently of the pressure

Fluid flows



Fluid flows

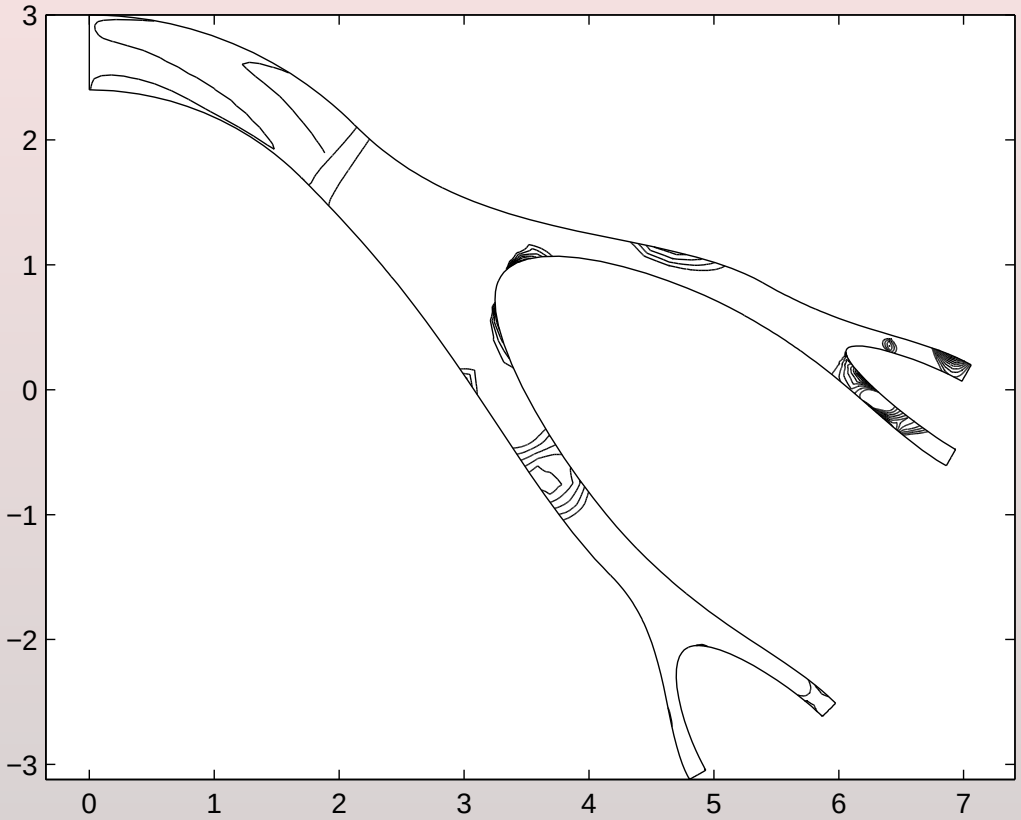


FIGURE: Error distribution for a new configuration $N_P = 15$, $N_B = 15$ error plot for the pressure max $=3.10^{-2}$, for the velocity error $\simeq 3.10^{-3}$.

Fluid flows

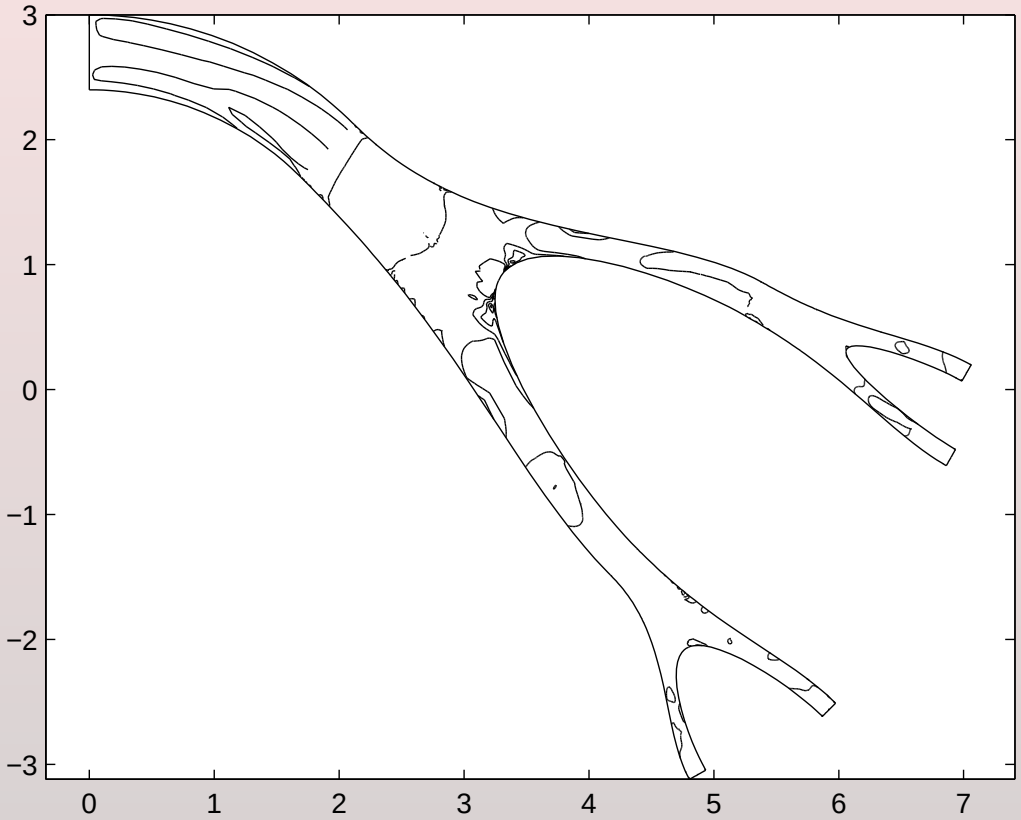


FIGURE: Error distribution for a new configuration $N_P = 15$, $N_B = 30$ error plot for the pressure max=6.10^{-3}, for the velocity error $\simeq 4.10^{-4}$, size problem < 340.

Fluid flows

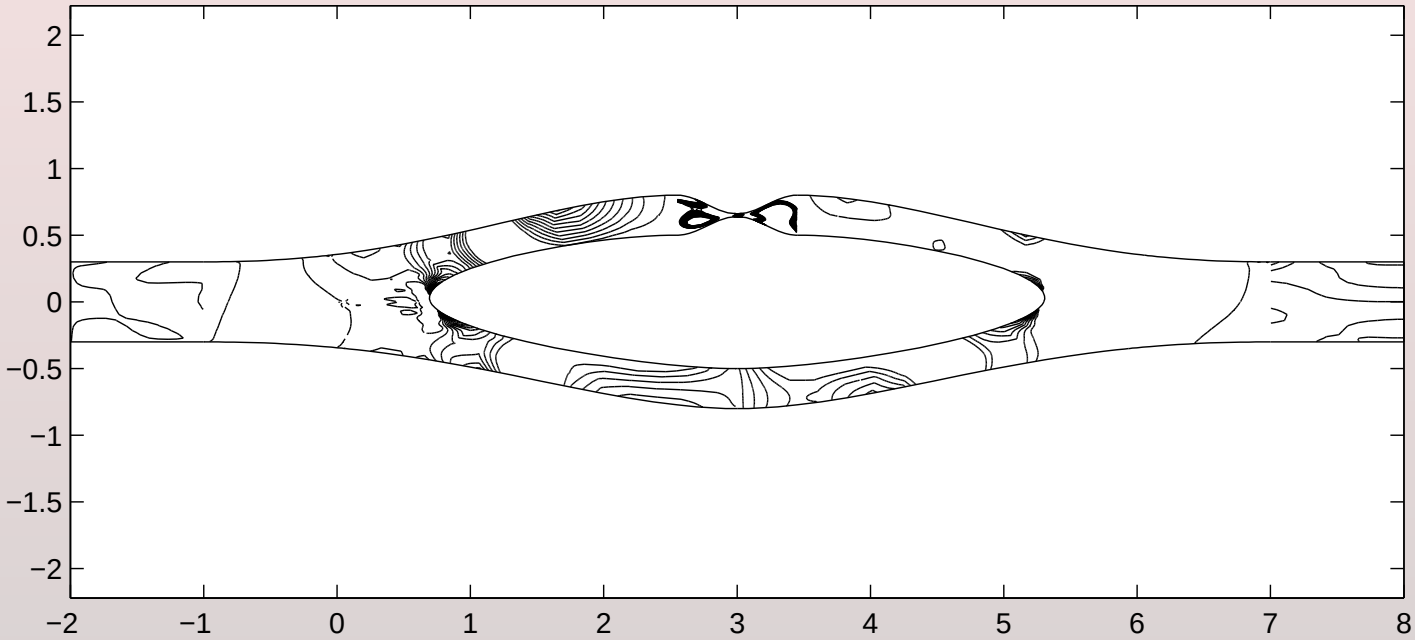


FIGURE: A stenosis problem with $N_P = 15$, $N_B = 15$.

Fluid flows

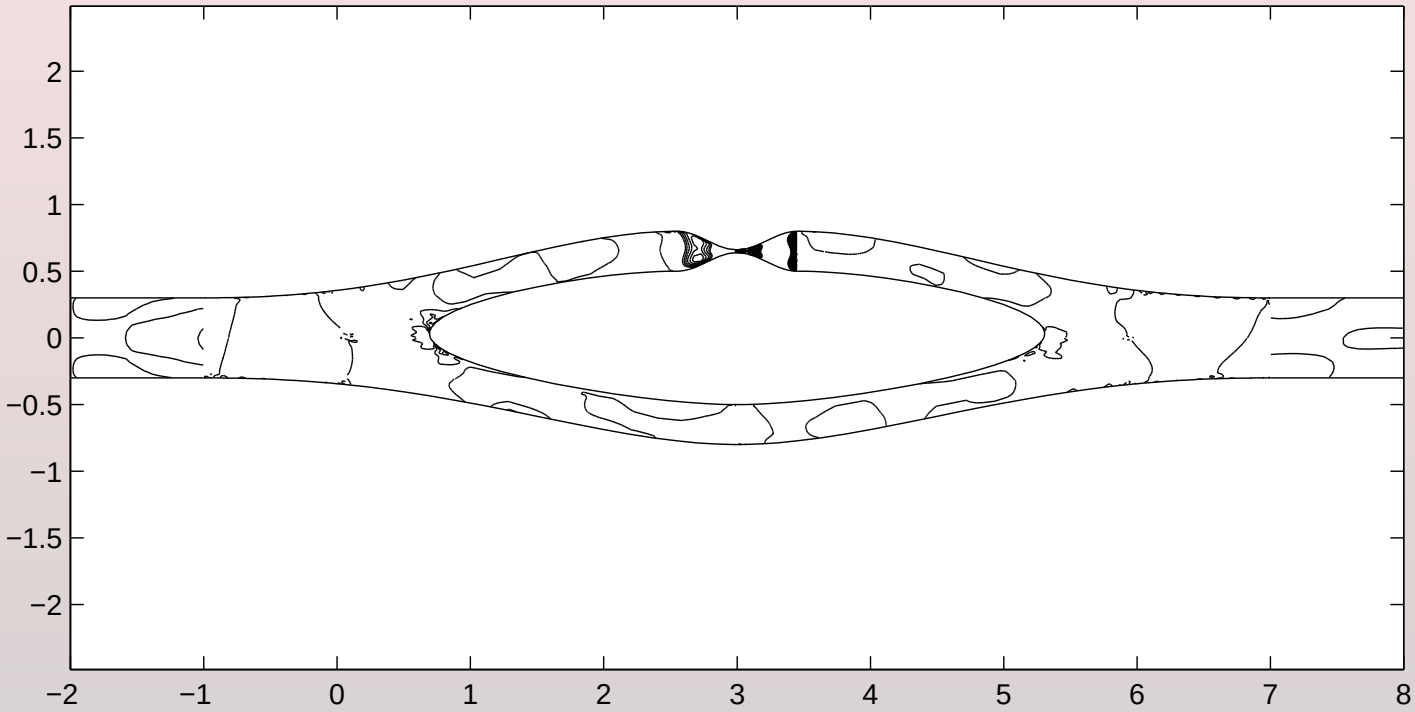


FIGURE: A stenosis problem with $N_P = 15$, $N_B = 30$.

Fluid flows

N	N_1	N_2	$\lvert u_N - u\rvert_{H^1}$	$\lvert\lvert p_N - p\rvert\rvert_{L^2}$
45	9	9	$9.3 \cdot 10^{-3}$	$3.3 \cdot 10$
55	11	11	$3.1 \cdot 10^{-3}$	$5.3 \cdot 10^{-1}$
65	13	13	$2.3 \cdot 10^{-3}$	$9.0 \cdot 10^{-2}$
75	15	15	$1.4 \cdot 10^{-3}$	$5.3 \cdot 10^{-2}$
105	15	30	$5.4 \cdot 10^{-4}$	$3.0 \cdot 10^{-2}$

TABLE: Steady Stokes solution on a multi-block bypass with three pipe blocks and two bifurcation blocks. Here, $N = 3N_1 + 2N_2$.

Important remarks....

- Note that contrarily to what happens in the parameter dependant problem
- The full problem over the global geometry is never constructed in the reduced element method
- This is a major achievement
- Note also that, more generally, the reduced basis functions have to be suitably prepared
- Finally, do not forget that off-line pre-computations have to be done in order to yield a reasonable approximation method which is sufficiently rapid for online computations.

Important remarks....

- Note that contrarily to what happens in the parameter dependant problem
- The full problem over the global geometry is never constructed in the reduced element method
- This is a major achievement
- Note also that, more generally, the reduced basis functions have to be suitably prepared
- Finally do not forget that off-line pre-computations have to be done involving your favorit approximation method, and that the approach is rapid for online computations.

Important remarks....

- Note that contrarily to what happens in the parameter dependant problem
- The full problem over the global geometry is never constructed in the reduced element method
- This is a major achievement
- Note also that, more generally, the reduced basis functions have to be suitably prepared
- Finally, do not forget that off-line pre-computations have to be done, involving your favorit approximation method, and that the approach is rapid for online computations.

Important remarks....

- Note that contrarily to what happens in the parameter dependant problem
- The full problem over the global geometry is never constructed in the reduced element method
- This is a major achievement

- Note also that, more generally, the reduced basis functions have to be suitably prepared

- Finally, do not forget that off-line pre-computations have to be done, involving your favorit approximation method, and that the approach is rapid for online computations.

Important remarks....

- Note that contrarily to what happens in the parameter dependant problem
- The full problem over the global geometry is never constructed in the reduced element method
- This is a major achievement

- Note also that, more generally, the reduced basis functions have to be suitably prepared

- Finally, do not forget that off-line pre-computations have to be done, involving your favorit approximation method, and that the approach is rapid for online computations.

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods and RB methods allow to get fast convergence rates (most of the times exponential)

Spectral element methods is generalized in this frame to the Reduced Basis Element Method that allows to combine the accuracy of the RB methods with the ease of DD (allowing to diminish the size of the parameter space . . .and thus the complexity of the solution)

What about collocation for non linear problems, i.e. interpolation ? ?

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods and RB methods allow to get fast convergence rates (most of the times exponential)

Spectral element methods is generalized in this frame to the Reduced Basis Element Method that allows to combine the accuracy of the RB methods with the ease of DD (allowing to diminish the size of the parameter space . . .and thus the complexity of the solution)

What about collocation for non linear problems, i.e. interpolation ? ?

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods and RB methods allow to get fast convergence rates (most of the times exponential)

Spectral element methods is generalized in this frame to the Reduced Basis Element Method that allows to combine the accuracy of the RB methods with the ease of DD (allowing to diminish the size of the parameter space . . .and thus the complexity of the solution)

What about collocation for non linear problems, i.e. interpolation ? ?

Partial Conclusion

Link with what you may know on spectral methods

Spectral methods and RB methods allow to get fast convergence rates (most of the times exponential)

Spectral element methods is generalized in this frame to the Reduced Basis Element Method that allows to combine the accuracy of the RB methods with the ease of DD (allowing to diminish the size of the parameter space . . .and thus the complexity of the solution)

What about collocation for non linear problems, i.e. interpolation ? ?

Background.

General issues about interpolation

Interpolation is a general and classical tool for approximation

Assume you are given a set $\varphi_1, \ldots, \varphi_n, \ldots$ of linearly independent functions, given a function f that you want to approximate, the problem is

- find a familly of scalars $\{\alpha_n^M\}_{1 \le n \le M}$ such that
- $f(\zeta_m) = \sum_{n=1}^{M} \alpha_n \varphi_n(\zeta_m)$

where the *interpolation points* ζ_m^M are suitably chosen

of course, the project is generally not limited to the approximation in the only

$$X_M = \mathrm{Span}\{\varphi_n,\ 1 \le n \le M\}$$

Background.

General issues about interpolation

Interpolation is a general and classical tool for approximation
Assume you are given a set $\varphi_1, \ldots, \varphi_n, \ldots$ of linearly independent functions, given a function f that you want to approximate, the problem is

- find a familly of scalars $\{\alpha_n^M\}_{1 \le n \le M}$ such that
- $f(\zeta_m) = \sum_{n=1}^{M} \alpha_n \varphi_n(\zeta_m)$

where the *interpolation points* ζ_m^M are suitably chosen
of course, the project is generally not limited to the approximation in the only

$$X_M = \mathrm{Span}\{\varphi_n, 1 \le n \le M\}$$

Background.

General issues about interpolation

Interpolation is a general and classical tool for approximation
Assume you are given a set $\varphi_1, \ldots, \varphi_n, \ldots$ of linearly independent functions, given a function f that you want to approximate, the problem is

- find a familly of scalars $\{\alpha_n^M\}_{1 \le n \le M}$ such that
- $f(\zeta_m^M) = \sum_{n=1}^{M} \alpha_n \varphi_n(\zeta_m^M)$

where the *interpolation points* ζ_m^M are suitably chosen
of course, the project is generally not limited to the approximation in the only

$$X_M = \text{Span}\{\varphi_n, \ 1 \le n \le M\}$$

Background.

General issues about interpolation

Interpolation is a general and classical tool for approximation
Assume you are given a set $\varphi_1, \ldots, \varphi_n, \ldots$ of linearly independent functions, given a function f that you want to approximate, the problem is

- find a familly of scalars $\{\alpha_n^M\}_{1 \le n \le M}$ such that
- $f(\zeta_m^M) = \sum_{n=1}^M \alpha_n^M \varphi_n(\zeta_m^M)$

where the *interpolation points* ζ_m^M are suitably chosen
of course, the project is generally not limited to the approximation in the only

$$X_M = \text{Span}\{\varphi_n,\ 1 \le n \le M\}$$

Background.

General issues about interpolation

Interpolation is a general and classical tool for approximation
Assume you are given a set $\varphi_1, \ldots, \varphi_n, \ldots$ of linearly independent functions, given a function f that you want to approximate, the problem is

- find a familly of scalars $\{\alpha_n^M\}_{1 \le n \le M}$ such that
- $f(\zeta_m^M) = \sum_{n=1}^M \alpha_n^M \varphi_n(\zeta_m^M)$

where the *interpolation points* ζ_m^M are suitably chosen

of course, the project is generally not limited to the approximation in the only

$$X_M = \text{Span}\{\varphi_n,\ 1 \le n \le M\}$$

Background.

General issues about interpolation

Interpolation is a general and classical tool for approximation
Assume you are given a set $\varphi_1, \ldots, \varphi_n, \ldots$ of linearly independent functions, given a function f that you want to approximate, the problem is

- find a familly of scalars $\{\alpha_n^M\}_{1 \le n \le M}$ such that
- $f(\zeta_m^M) = \sum_{n=1}^M \alpha_n^M \varphi_n(\zeta_m^M)$

where the *interpolation points* ζ_m^M are suitably chosen
of course, the project is generally not limited to the approximation in the only

$$X_M = \text{Span}\{\varphi_n,\ 1 \le n \le M\}$$

Background.

General issues about interpolation

Interpolation is a general and classical tool for approximation
Assume you are given a set $\varphi_1, \ldots, \varphi_n, \ldots$ of linearly independent functions, given a function f that you want to approximate, the problem is

- find a familly of scalars $\{\alpha_n^M\}_{1 \le n \le M}$ such that
- $f(\zeta_m^M) = \sum_{n=1}^M \alpha_n^M \varphi_n(\zeta_m^M)$

where the *interpolation points* ζ_m^M are suitably chosen
of course, the project is generally not limited to the approximation in the only

$$X_M = \text{Span}\{\varphi_n,\ 1 \le n \le M\}$$

Background.

General issues about interpolation

Among the classical questions raised by interpolation

- given a set of points, does the interpolant at these points exist ;
- is the interpolant unique ;
- how does the interpolation process compares with other approximations (in particular orthogonal projections) ;
- is there an optimal selection for the interpolation points ;
- is there a constructive optimal selection for the interpolation points ;

These question are covered in the polynomial case, though not completely and the answers are complex and rather recent.

Background.

General issues about interpolation

Among the classical questions raised by interpolation

- given a set of points, does the interpolant at these points exist ;
- is the interpolant unique ;
- how does the interpolation process compares with other approximations (in particular orthogonal projections) ;
- is there an optimal selection for the interpolation points ;
- is there a constructive optimal selection for the interpolation points ;

These question are covered in the polynomial case, though not completely and the answers are complex and rather recent.

Background.

General issues about interpolation

Among the classical questions raised by interpolation

- given a set of points, does the interpolant at these points exist ;
- is the interpolant unique ;
- how does the interpolation process compares with other approximations (in particular orthogonal projections) ;
- is there an optimal selection for the interpolation points ;
- is there a constructive optimal selection for the interpolation points ;

These question are covered in the polynomial case, though not completely and the answers are complex and rather recent.

Background.

General issues about interpolation

Among the classical questions raised by interpolation

- given a set of points, does the interpolant at these points exist ;
- is the interpolant unique ;
- how does the interpolation process compares with other approximations (in particular orthogonal projections) ;
- is there an optimal selection for the interpolation points ;
- is there a constructive optimal selection for the interpolation points ;

These question are covered in the polynomial case, though not completely and the answers are complex and rather recent.

Background.

General issues about interpolation

Among the classical questions raised by interpolation

- given a set of points, does the interpolant at these points exist ;
- is the interpolant unique ;
- how does the interpolation process compares with other approximations (in particular orthogonal projections) ;
- is there an optimal selection for the interpolation points ;
- is there a constructive optimal selection for the interpolation points ;

These question are covered in the polynomial case, though not completely and the answers are complex and rather recent.

Background.

General issues about interpolation

Among the classical questions raised by interpolation

- given a set of points, does the interpolant at these points exist ;
- is the interpolant unique ;
- how does the interpolation process compares with other approximations (in particular orthogonal projections) ;
- is there an optimal selection for the interpolation points ;
- is there a constructive optimal selection for the interpolation points ;

These question are covered in the polynomial case, though not completely and the answers are complex and rather recent

Background.

General issues about interpolation

Among the classical questions raised by interpolation

- given a set of points, does the interpolant at these points exist ;
- is the interpolant unique ;
- how does the interpolation process compares with other approximations (in particular orthogonal projections) ;
- is there an optimal selection for the interpolation points ;
- is there a constructive optimal selection for the interpolation points ;

These question are covered in the polynomial case, though not completely and the answers are complex and rather recent

Background.

General issues about interpolation

Among the classical questions raised by interpolation

- given a set of points, does the interpolant at these points exist ;
- is the interpolant unique ;
- how does the interpolation process compares with other approximations (in particular orthogonal projections) ;
- is there an optimal selection for the interpolation points ;
- is there a constructive optimal selection for the interpolation points ;

These question are covered in the polynomial case, though not completely and the answers are complex and rather recent

Approximation in a space of small n-width.

Interpolation

We are looking for a constructive way of approximating in $\mathcal{X}$, we assume that $\mathcal{X} \subset \mathcal{C}^0$

We propose a greedy approach both for constructing the interpolation points and the discrete spaces X_M,

our method is hierarchical

Approximation in a space of small n-width.

Interpolation

We are looking for a constructive way of approximating in $\mathcal{X}$, we assume that $\mathcal{X} \subset \mathcal{C}^0$

We propose a greedy approach both for constructing the interpolation points and the discrete spaces X_M,

our method is hierarchical

Approximation in a space of small n-width.

Interpolation

We are looking for a constructive way of approximating in $\mathcal{X}$, we assume that $\mathcal{X} \subset \mathcal{C}^0$

We propose a greedy approach both for constructing the interpolation points and the discrete spaces X_M,

our method is hierarchical

Approximation in a space of small n-width.

definition of the magic points

The first interpolating function is

$$\varphi_1 = \arg \max_{\Phi \in \mathcal{S}_1[X]} \|\Phi\|_{L^\infty(\Omega)}$$

The first interpolation point is

$$\zeta_1 = \arg \max_{x \in \Omega} |\varphi_1|$$

Allows to define the first interpolation operator $\mathcal{I}_1[\Phi] = \Phi(\zeta_1)\varphi_1(\cdot)/\varphi_1(\zeta_1)$

The second interpolating function is

$$\varphi_2 = \arg \max_{\Phi \in \mathcal{S}_1[X]} \|\Phi(\cdot) - \mathcal{I}_1[\Phi]\|_{L^\infty(\Omega)}$$

The second interpolation point is

$$\zeta_2 = \arg \max_{x \in \Omega} |\varphi_2(\cdot) - \mathcal{I}_1[\varphi_2]|$$

Approximation in a space of small n-width.

definition of the magic points

The first interpolating function is

$$\varphi_1 = \arg \max_{\Phi \in \mathcal{S}_1[X]} \|\Phi\|_{L^\infty(\Omega)}$$

The first interpolation point is

$$\zeta_1 = \arg \max_{x \in \Omega} |\varphi_1|$$

Allows to define the first interpolation operator $\mathcal{I}_1[\Phi] = \Phi(\zeta_1)\varphi_1(\cdot)/\varphi_1(\zeta_1)$

The second interpolating function is

$$\varphi_2 = \arg \max_{\Phi \in \mathcal{S}_1[X]} \|\Phi(\cdot) - \mathcal{I}_1[\Phi]\|_{L^\infty(\Omega)}$$

The second interpolation point is

$$\zeta_2 = \arg \max_{x \in \Omega} |\varphi_2(\cdot) - \mathcal{I}_1[\varphi_2]|$$

Approximation in a space of small n-width.

definition of the magic points

The first interpolating function is

$$\varphi_1 = \arg \max_{\Phi \in \mathcal{S}_1[X]} \|\Phi\|_{L^\infty(\Omega)}$$

The first interpolation point is

$$\zeta_1 = \arg \max_{x \in \Omega} |\varphi_1|$$

Allows to define the first interpolation operator $\mathcal{I}_1[\Phi] = \Phi(\zeta_1)\varphi_1(\cdot)/\varphi_1(\zeta_1)$

The second interpolating function is

$$\varphi_2 = \arg \max_{\Phi \in \mathcal{S}_1[X]} \|\Phi(\cdot) - \mathcal{I}_1[\Phi]\|_{L^\infty(\Omega)}$$

The second interpolation point is

$$\zeta_2 = \arg \max_{x \in \Omega} |\varphi_2(\cdot) - \mathcal{I}_1[\varphi_2]|$$

Approximation in a space of small n-width.

definition of the magic points

The first interpolating function is

$$\varphi_1 = \arg \max_{\Phi \in \mathcal{S}_1[X]} \|\Phi\|_{L^\infty(\Omega)}$$

The first interpolation point is

$$\zeta_1 = \arg \max_{x \in \Omega} |\varphi_1|$$

Allows to define the first interpolation operator $\mathcal{I}_1[\Phi] = \Phi(\zeta_1)\varphi_1(\cdot)/\varphi_1(\zeta_1)$
The second interpolating function is

$$\varphi_2 = \arg \max_{\Phi \in \mathcal{S}_1[X]} \|\Phi(\cdot) - \mathcal{I}_1[\Phi]\|_{L^\infty(\Omega)}$$

The second interpolation point is

$$\zeta_2 = \arg \max_{x \in \Omega} |\varphi_2(\cdot) - \mathcal{I}_1[\varphi_2]|$$

Approximation in a space of small n-width.

definition of the magic points

The first interpolating function is

$$\varphi_1 = \arg \max_{\Phi \in \mathcal{S}_1[X]} \|\Phi\|_{L^\infty(\Omega)}$$

The first interpolation point is

$$\zeta_1 = \arg \max_{x \in \Omega} |\varphi_1|$$

Allows to define the first interpolation operator $\mathcal{I}_1[\Phi] = \Phi(\zeta_1)\varphi_1(\cdot)/\varphi_1(\zeta_1)$
The second interpolating function is

$$\varphi_2 = \arg \max_{\Phi \in \mathcal{S}_1[X]} \|\Phi(\cdot) - \mathcal{I}_1[\Phi]\|_{L^\infty(\Omega)}$$

The second interpolation point is

$$\zeta_2 = \arg \max_{x \in \Omega} |\varphi_2(\cdot) - \mathcal{I}_1[\varphi_2]|$$

Approximation in a space of small n-width.

The recursion formula

We first solve the interpolation problem for $\alpha_j^{M-1}(\Phi), 1 \leq j \leq M-1$, from

$$\sum_{j=1}^{M-1} \varphi_j(\zeta_i)\alpha_j^{M-1}(\Phi) = \Phi(\zeta_i), \quad i = 1, \ldots, M-1 \ , \tag{6}$$

and set

$$\mathcal{I}_{M-1}[\Phi] = \sum_{j=1}^{M-1} \alpha_j^{M-1}(\Phi)\varphi_j \ , \tag{7}$$

and

$$\varepsilon_{M-1}(\Phi) = \|\Phi() - \mathcal{I}_{M-1}[\Phi()]\|_{L^\infty(\Omega)} \ , \tag{8}$$

for all $\Phi \in \mathcal{X}$;

Approximation in a space of small n-width.

The recursion formula

We first solve the interpolation problem for $\alpha_j^{M-1}(\Phi), 1 \leq j \leq M-1$, from

$$\sum_{j=1}^{M-1} \varphi_j(\zeta_i)\alpha_j^{M-1}(\Phi) = \Phi(\zeta_i), \quad i = 1, \ldots, M-1 \ , \tag{6}$$

and set

$$\mathcal{I}_{M-1}[\Phi] = \sum_{j=1}^{M-1} \alpha_j^{M-1}(\Phi)\varphi_j \ , \tag{7}$$

and

$$\varepsilon_{M-1}(\Phi) = \|\Phi() - \mathcal{I}_{M-1}[\Phi()]\|_{L^\infty(\Omega)} \ , \tag{8}$$

for all $\Phi \in \mathcal{X}$;

Approximation in a space of small n-width.

The recursion formula

We first solve the interpolation problem for $\alpha_j^{M-1}(\Phi), 1 \leq j \leq M-1$, from

$$\sum_{j=1}^{M-1} \varphi_j(\zeta_i)\alpha_j^{M-1}(\Phi) = \Phi(\zeta_i), \quad i = 1, \ldots, M-1 \ , \tag{6}$$

and set

$$\mathcal{I}_{M-1}[\Phi] = \sum_{j=1}^{M-1} \alpha_j^{M-1}(\Phi)\varphi_j \ , \tag{7}$$

and

$$\varepsilon_{M-1}(\Phi) = \|\Phi(\cdot) - \mathcal{I}_{M-1}[\Phi(\cdot)]\|_{L^\infty(\Omega)} \ , \tag{8}$$

for all $\Phi \in X$;

Approximation in a space of small n-width.

The recursion formula

We first solve the interpolation problem for $\alpha_j^{M-1}(\Phi), 1 \leq j \leq M-1$, from

$$\sum_{j=1}^{M-1} \varphi_j(\zeta_i)\alpha_j^{M-1}(\Phi) = \Phi(\zeta_i), \quad i = 1, \ldots, M-1 \ , \tag{6}$$

and set

$$\mathcal{I}_{M-1}[\Phi] = \sum_{j=1}^{M-1} \alpha_j^{M-1}(\Phi)\varphi_j \ , \tag{7}$$

and

$$\varepsilon_{M-1}(\Phi) = \|\Phi(\cdot) - \mathcal{I}_{M-1}[\Phi(\cdot)]\|_{L^\infty(\Omega)} \ , \tag{8}$$

for all $\Phi \in X$;

Approximation in a space of small n-width.

The recursion formula

We first solve the interpolation problem for $\alpha_j^{M-1}(\Phi), 1 \leq j \leq M-1$, from

$$\sum_{j=1}^{M-1} \varphi_j(\zeta_i)\alpha_j^{M-1}(\Phi) = \Phi(\zeta_i), \quad i = 1, \ldots, M-1 \ , \tag{6}$$

and set

$$\mathcal{I}_{M-1}[\Phi] = \sum_{j=1}^{M-1} \alpha_j^{M-1}(\Phi)\varphi_j \ , \tag{7}$$

and

$$\varepsilon_{M-1}(\Phi) = \|\Phi(\cdot) - \mathcal{I}_{M-1}[\Phi(\cdot)]\|_{L^\infty(\Omega)} \ , \tag{8}$$

for all $\Phi \in X$;

Approximation in a space of small n-width.

The recursion formula

We then define

$$\varphi_M = \arg \max_{\Phi \in \mathcal{S}_1[X]} \varepsilon_{M-1}(\Phi) \ , \tag{9}$$

and

$$\zeta_M = \arg \max_{x \in \Omega} \|\varphi_M(x) - \mathcal{I}_{M-1}[\varphi_M(x)]\|_{L^\infty(\Omega)} \ , \tag{10}$$

The procedure is well posed if X is of sufficently large dimension (for $M \leq M_{max} \leq \dim X$).

Approximation in a space of small n-width.

The recursion formula

We then define

$$\varphi_M = \arg \max_{\Phi \in \mathcal{S}_1[X]} \varepsilon_{M-1}(\Phi) \ , \tag{9}$$

and

$$\zeta_M = \arg \max_{x \in \Omega} \|\varphi_M(x) - \mathcal{I}_{M-1}[\varphi_M(x)]\|_{L^\infty(\Omega)} \ , \tag{10}$$

The procedure is well posed if X is of sufficently large dimension (for $M \leq M_{max} \leq \dim X$).

Approximation in a space of small n-width.

The recursion formula

We then define

$$\varphi_M = \arg \max_{\Phi \in \mathcal{S}_1[X]} \varepsilon_{M-1}(\Phi) \ , \tag{9}$$

and

$$\zeta_M = \arg \max_{x \in \Omega} \|\varphi_M(x) - \mathcal{I}_{M-1}[\varphi_M(x)]\|_{L^\infty(\Omega)} \ , \tag{10}$$

The procedure is well posed if X is of sufficently large dimension (for $M \leq M_{max} \leq \dim X$).

Approximation in a space of small n-width.

The recursion formula

We then define

$$\varphi_M = \arg \max_{\Phi \in \mathcal{S}_1[X]} \varepsilon_{M-1}(\Phi) \ , \tag{9}$$

and

$$\zeta_M = \arg \max_{x \in \Omega} \|\varphi_M(x) - \mathcal{I}_{M-1}[\varphi_M(x)]\|_{L^\infty(\Omega)} \ , \tag{10}$$

The procedure is well posed if X is of sufficently large dimension (for $M \leq M_{max} \leq \dim X$).

Approximation in a space of small n-width.

The Lebesgue constant

The error analysis of the interpolation procedure classically involves the Lebesgue constant $\Lambda_M = \sup_{x\in\Omega} \sum_{i=1}^{M} |h_i^M(x)|$, where the h_i^M is the associated Lagrange basis.

A (in practice very pessimistic) upper-bound for the Lebesgue constant is $2^M - 1$.

We remind also that the Lebesgue constant enters into the bound for the interpolation error as follows

Lemma

For any $u \in X$, the interpolation error satisfies

$$\|\Phi - \mathcal{I}_M\Phi\|_{L^\infty(\Omega)} \leq (1+\Lambda_M) \inf_{\psi_M \in span\{\varphi_i, 1\leq i\leq M\}} \|\Phi - \psi_M\|_{L^\infty(\Omega)}. \quad (11)$$

Approximation in a space of small n-width.

The Lebesgue constant

The error analysis of the interpolation procedure classically involves the Lebesgue constant $\Lambda_M = \sup_{x\in\Omega} \sum_{i=1}^{M} |h_i^M(x)|$, where the h_i^M is the associated Lagrange basis.

A (in practice very pessimistic) upper-bound for the Lebesgue constant is $2^M - 1$.

We remind also that the Lebesgue constant enters into the bound for the interpolation error as follows

Lemma

For any $u \in X$, the interpolation error satisfies

$$\|\Phi - \mathcal{I}_M\Phi\|_{L^\infty(\Omega)} \leq (1+\Lambda_M) \inf_{\psi_M \in span\{\varphi_i, 1\leq i\leq M\}} \|\Phi - \psi_M\|_{L^\infty(\Omega)}. \quad (11)$$

Approximation in a space of small n-width.

The Lebesgue constant

The error analysis of the interpolation procedure classically involves the Lebesgue constant $\Lambda_M = \sup_{x\in\Omega} \sum_{i=1}^{M} |h_i^M(x)|$, where the h_i^M is the associated Lagrange basis.
A (in practice very pessimistic) upper-bound for the Lebesgue constant is $2^M - 1$.
We remind also that the Lebesgue constant enters into the bound for the interpolation error as follows

Lemma

For any $u \in X$, the interpolation error satisfies

$$\|\Phi - \mathcal{I}_M\Phi\|_{L^\infty(\Omega)} \leq (1+\Lambda_M) \inf_{\psi_M \in span\{\varphi_i, 1\leq i\leq M\}} \|\Phi - \psi_M\|_{L^\infty(\Omega)}. \quad (11)$$

Approximation in a space of small n-width.

The Lebesgue constant

The error analysis of the interpolation procedure classically involves the Lebesgue constant $\Lambda_M = \sup_{x\in\Omega} \sum_{i=1}^{M} |h_i^M(x)|$, where the h_i^M is the associated Lagrange basis.
A (in practice very pessimistic) upper-bound for the Lebesgue constant is $2^M - 1$.
We remind also that the Lebesgue constant enters into the bound for the interpolation error as follows

Lemma

For any $u \in X$, the interpolation error satisfies

$$\|\Phi - \mathcal{I}_M\Phi\|_{L^\infty(\Omega)} \leq (1+\Lambda_M) \inf_{\psi_M \in span\{\varphi_i, 1\leq i\leq M\}} \|\Phi - \psi_M\|_{L^\infty(\Omega)}. \quad (11)$$

Approximation in a space of small n-width.

The Approximation of the greedy algorithm

We can also prove that

Theorem

Assume that there exists a sequence of finite dimensional spaces

$$X_1 \subset X_2 \subset \cdots \subset X_M \subset \cdots \subset X, \quad \dim X_M = M \tag{12}$$

such that there exists $c > 0$ *and* α *with*

$$\forall \Phi \in X, \inf_{\psi_M \in X_M} \|\Phi - \psi_M\|_X \leq c e^{-\alpha M} \tag{13}$$

then, if $\alpha > \log(4)$, *there exists* $\beta > 0$ *such that*

$$\|\Phi - \mathcal{I}_M \Phi\|_{L^\infty(\Omega)} \leq c e^{-\beta M}. \tag{14}$$

Interpolation error..... a posteriori

For an estimator on the error, let $M \leq M_{\max} - 1$, we define

$$\hat{\varepsilon}_M(\Phi) \equiv |\Phi(\zeta_{M+1}) - \mathcal{I}_M\Phi(\zeta_{M+1})|$$

Lemma : If $\Phi \in X_{M+1}$, then

$$\|\Phi(\,\cdot) - \mathcal{I}_M\Phi(\,\cdot)\|_{L^\infty(\Omega)} \leq \hat{\varepsilon}_M(\Phi)$$

Of course, in general $\Phi \notin X_{M+1}$

and hence our estimator $\hat{\varepsilon}_M(\Phi)$ is not a rigorous upper bound ;

however, if $\varepsilon_M(\Phi) \to 0$ very fast, we expect (and check) that the effectivity $\eta_M(\Phi) = \frac{\hat{\varepsilon}_M(\Phi)}{\varepsilon_M(\Phi)} \to 1$.

Furthermore, the estimator is very inexpensive – *one additional evaluation* of Φ.

Interpolation error..... a posteriori

For an estimator on the error, let $M \leq M_{\max} - 1$, we define

$$\hat{\varepsilon}_M(\Phi) \equiv |\Phi(\zeta_{M+1}) - \mathcal{I}_M\Phi(\zeta_{M+1})|$$

Lemma : If $\Phi \in X_{M+1}$, then

$$\|\Phi(\ \cdot) - \mathcal{I}_M\Phi(\ \cdot)\|_{L^\infty(\Omega)} \leq \hat{\varepsilon}_M(\Phi)$$

Of course, in general $\Phi \notin X_{M+1}$
and hence our estimator $\hat{\varepsilon}_M(\Phi)$ is not a rigorous upper bound ;

however, if $\varepsilon_M(\Phi) \longrightarrow 0$ very fast, we expect (and check) that the effectivity $\eta_M(\Phi) = \hat{\varepsilon}_M(\Phi) / \varepsilon_M(\Phi) \sim 1$.

Furthermore, the estimator is very inexpensive – *one additional evaluation* of Φ.

Interpolation error..... a posteriori

For an estimator on the error, let $M \leq M_{\max} - 1$, we define

$$\hat{\varepsilon}_M(\Phi) \equiv |\Phi(\zeta_{M+1}) - \mathcal{I}_M\Phi(\zeta_{M+1})|$$

Lemma : If $\Phi \in X_{M+1}$, then

$$\|\Phi(\,\cdot) - \mathcal{I}_M\Phi(\,\cdot)\|_{L^\infty(\Omega)} \leq \hat{\varepsilon}_M(\Phi)$$

Of course, in general $\Phi \notin X_{M+1}$
and hence our estimator $\hat{\varepsilon}_M(\Phi)$ is not a rigorous upper bound ;

however, if $\varepsilon_M(\Phi) \to 0$ very fast, we expect (and check) that the effectivity, $\eta_M(\Phi) \equiv \hat{\varepsilon}_M(\Phi)/\varepsilon_M(\Phi) \simeq 1$.

Furthermore, the estimator is very inexpensive – *one additional evaluation* of Φ.

Interpolation error..... a posteriori

For an estimator on the error, let $M \leq M_{\max} - 1$, we define

$$\hat{\varepsilon}_M(\Phi) \equiv |\Phi(\zeta_{M+1}) - \mathcal{I}_M\Phi(\zeta_{M+1})|$$

Lemma : If $\Phi \in X_{M+1}$, then

$$\|\Phi(\,\cdot) - \mathcal{I}_M\Phi(\,\cdot)\|_{L^\infty(\Omega)} \leq \hat{\varepsilon}_M(\Phi)$$

Of course, in general $\Phi \notin X_{M+1}$
and hence our estimator $\hat{\varepsilon}_M(\Phi)$ is not a rigorous upper bound ;

however, if $\varepsilon_M(\Phi) \longrightarrow 0$ very fast, we expect (and check) that the effectivity, $\eta_M(\Phi) \equiv \hat{\varepsilon}_M(\Phi)/\varepsilon_M(\Phi) \simeq 1$.

Furthermore, the estimator is very inexpensive – *one additional evaluation* of Φ.

Interpolation error..... a posteriori

For an estimator on the error, let $M \leq M_{\max} - 1$, we define

$$\hat{\varepsilon}_M(\Phi) \equiv |\Phi(\zeta_{M+1}) - \mathcal{I}_M\Phi(\zeta_{M+1})|$$

Lemma : If $\Phi \in X_{M+1}$, then

$$\|\Phi(\ \cdot) - \mathcal{I}_M\Phi(\ \cdot)\|_{L^\infty(\Omega)} \leq \hat{\varepsilon}_M(\Phi)$$

Of course, in general $\Phi \notin X_{M+1}$
and hence our estimator $\hat{\varepsilon}_M(\Phi)$ is not a rigorous upper bound ;

however, if $\varepsilon_M(\Phi) \to 0$ very fast, we expect (and check) that the effectivity, $\eta_M(\Phi) \equiv \hat{\varepsilon}_M(\Phi)/\varepsilon_M(\Phi) \simeq 1$.

Furthermore, the estimator is very inexpensive – *one additional evaluation of* Φ.

Interpolation error..... a posteriori

For an estimator on the error, let $M \leq M_{\max} - 1$, we define

$$\hat{\varepsilon}_M(\Phi) \equiv |\Phi(\zeta_{M+1}) - \mathcal{I}_M\Phi(\zeta_{M+1})|$$

Lemma : If $\Phi \in X_{M+1}$, then

$$\|\Phi(\ \cdot) - \mathcal{I}_M\Phi(\ \cdot)\|_{L^\infty(\Omega)} \leq \hat{\varepsilon}_M(\Phi)$$

Of course, in general $\Phi \notin X_{M+1}$
and hence our estimator $\hat{\varepsilon}_M(\Phi)$ is not a rigorous upper bound ;

however, if $\varepsilon_M(\Phi) \to 0$ very fast, we expect (and check) that the effectivity, $\eta_M(\Phi) \equiv \hat{\varepsilon}_M(\Phi)/\varepsilon_M(\Phi) \simeq 1$.

Furthermore, the estimator is very inexpensive – *one additional evaluation* of Φ.

Numerical results

We consider $\Phi(x) \equiv \Phi\left((x_1, x_2); (\mu_1, \mu_2)\right) \equiv \left((x_1 - \mu_1)^2 + (x_2 - \mu_2)^2\right)^{-1/2}$ for $x \in]0,1[^2$ and $\mu \in [-1, -0.01]^2$

M	$\varepsilon^*_{M,\max}$	$\overline{\rho}_M$	Λ_M	$\overline{\eta}_M$
8	8.30 E-02	0.68	1.76	0.17
16	4.20 E-03	0.67	2.63	0.1.
24	2.68 E-04	0.49	4.42	0.28
32	5.64 E-05	0.48	5.15	0.20
40	3.66 E-06	0.54	4.98	0.60
48	6.08 E-07	0.37	7.43	0.29

$\varepsilon^*_{M,\max}$ is the best fit error, $\overline{\rho}_M$ is the averaged ratio $\dfrac{\varepsilon}{\varepsilon^*(1+\Lambda)}$, Λ_M is the "Lebesgue" constant and $\overline{\eta}_M$ is the averaged effectivity index $\dfrac{\hat{\varepsilon}}{\varepsilon}$

Numerical results

We consider $\Phi(x) \equiv \Phi\left((x_1, x_2); (\mu_1, \mu_2)\right) \equiv \left((x_1 - \mu_1)^2 + (x_2 - \mu_2)^2\right)^{-1/2}$ for $x \in]0,1[^2$ and $\mu \in [-1, -0.01]^2$

M	$\varepsilon^*_{M,\max}$	$\overline{\rho}_M$	Λ_M	$\overline{\eta}_M$
8	8.30 E-02	0.68	1.76	0.17
16	4.20 E-03	0.67	2.63	0.1.
24	2.68 E-04	0.49	4.42	0.28
32	5.64 E-05	0.48	5.15	0.20
40	3.66 E-06	0.54	4.98	0.60
48	6.08 E-07	0.37	7.43	0.29

Note that we have here approximated the full set of $\Phi\left((x); \mu\right)$ with a few of them $\Phi\left((x); (\mu)\right) \simeq \sum_{i=1}^{M} \alpha_i \Phi\left((x); \mu^i\right)$

Numerical results

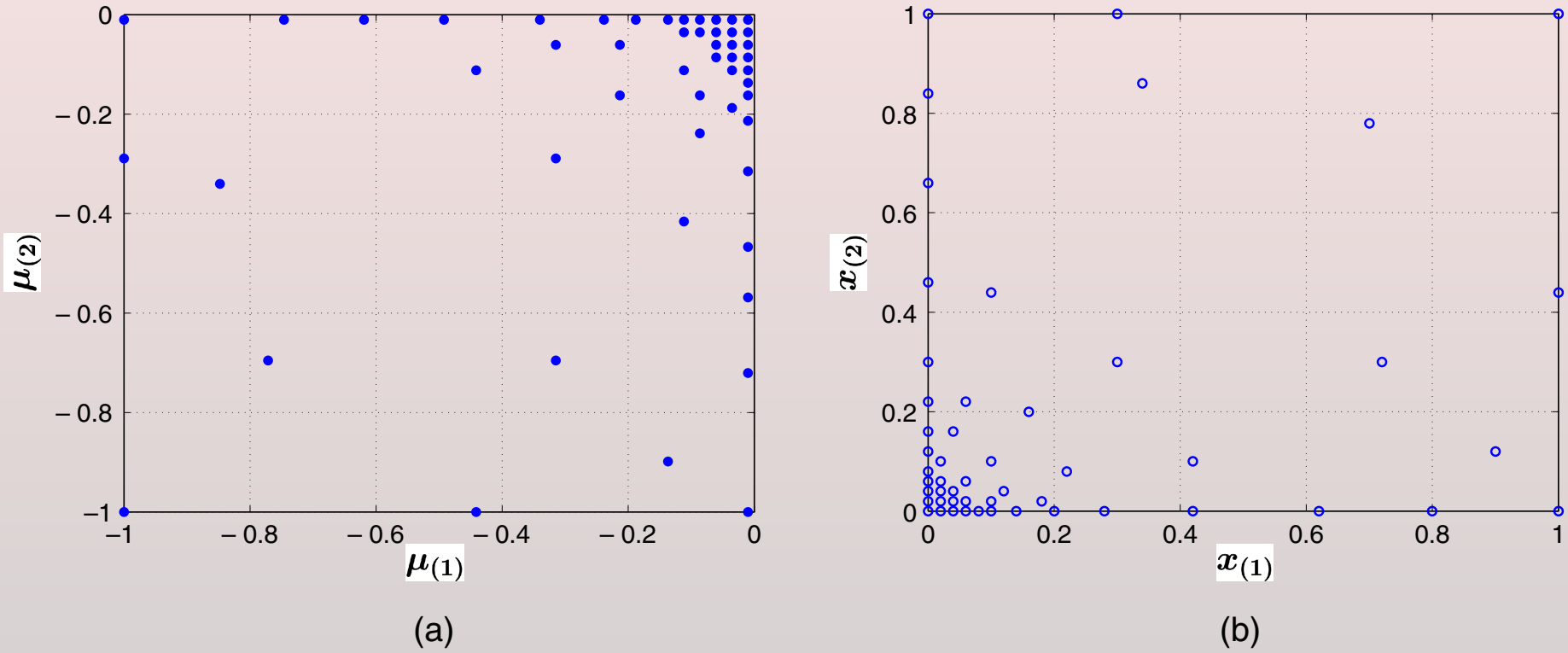


FIGURE: (a) Parameter sample set S_M^g, $M_{\max} = 51$, and (b) interpolation points x_m, $1 \leq m \leq M_{\max}$.

EIM 4 NLP

In the same way collocation methods have led to pseudo-spectral methods, Empirical Interpolation Methods have led to an easy implementation of nonlinear terms

GEIM

Moments - dictionary of linear forms

The interpolation process requires the evaluation of the functions at some points.

This can be replaced by any linear form $\sigma_i(.)$ instead of δ_{ζ_i}

Assume that you have a dictionary, a family, of such linear forms : relax the continuity assumption : X' instead of point evaluation in $\mathcal{C}^0$.

GEIM

Moments - dictionary of linear forms

The interpolation process requires the evaluation of the functions at some points.
This can be replaced by any linear form $\sigma_i(.)$ instead of δ_{ζ_i}
Assume that you have a dictionary, a family, of such linear forms : relax the continuity assumption : X' instead of point evaluation in $\mathcal{C}^0$.

GEIM

Moments - dictionary of linear forms

The interpolation process requires the evaluation of the functions at some points.
This can be replaced by any linear form $\sigma_i(.)$ instead of δ_{ζ_i}
Assume that you have a dictionary, a family, of such linear forms : relax the continuity assumption : X' instead of point evaluation in $\mathcal{C}^0$.

GEIM

Moments - dictionary of linear forms

The interpolation process requires the evaluation of the functions at some points.
This can be replaced by any linear form $\sigma_i(.)$ instead of δ_{ζ_i}
Assume that you have a dictionary, a family, of such linear forms : relax the continuity assumption : X' instead of point evaluation in $\mathcal{C}^0$.

GEIM

Moments - dictionary of linear forms

The interpolation process requires the evaluation of the functions at some points.
This can be replaced by any linear form $\sigma_i(.)$ instead of δ_{ζ_i}
Assume that you have a dictionary, a family, of such linear forms : relax the continuity assumption : X' instead of point evaluation in $\mathcal{C}^0$.

Problem at stake

The Laplace problem

- **Spatial domain :** $\Omega \subset \mathbb{R}^2$ and a non overlapping partition of it is : $\Omega = \Omega_1 \bigcup \Omega_2$.
- **Model :** the Laplace problem.
- **Hypothesis :** only Ω_1 is driving the phenomenon. The rest (Ω_2) is the environment.
- **Equation :**

$$-\Delta u = f, \ in\ \Omega$$

$$f = 1 + (\alpha \sin(x) + \beta \cos(\gamma \pi y))\chi_1(x, y)$$

α, β and γ are 3 parameters

Problem at stake

Aim

1. Reconstruction over Ω_2 with GEIM
2. Real-time simulation :
   - Ω_2 : Data acquisition + reconstruction with GEIM
   - Ω_1 : Implicit calculation with the boundary conditions provided by the interpolant of Ω_2.

Application to Laplace equation

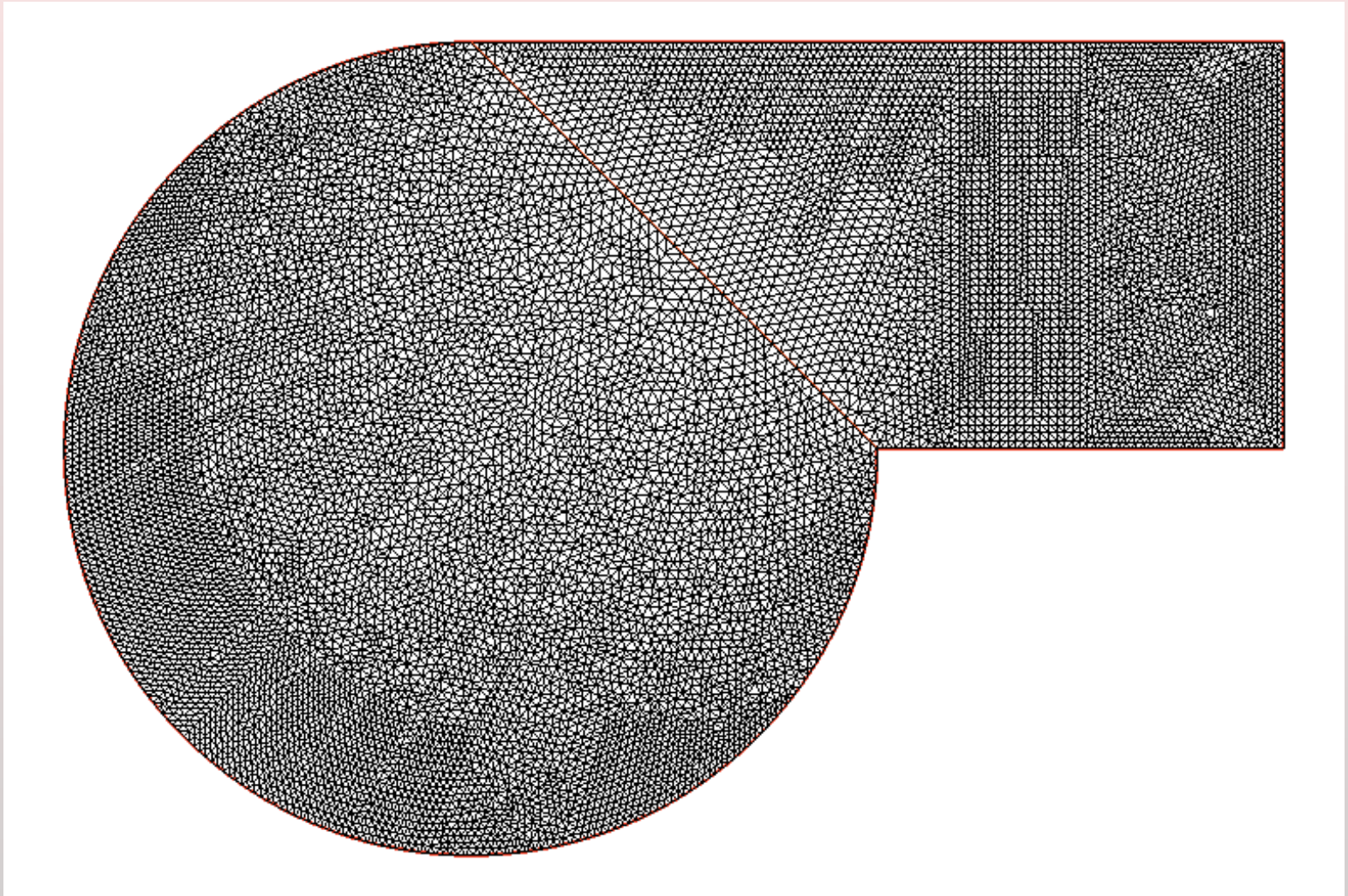


FIGURE: Ω_1 and Ω_2.

Application to Laplace equation

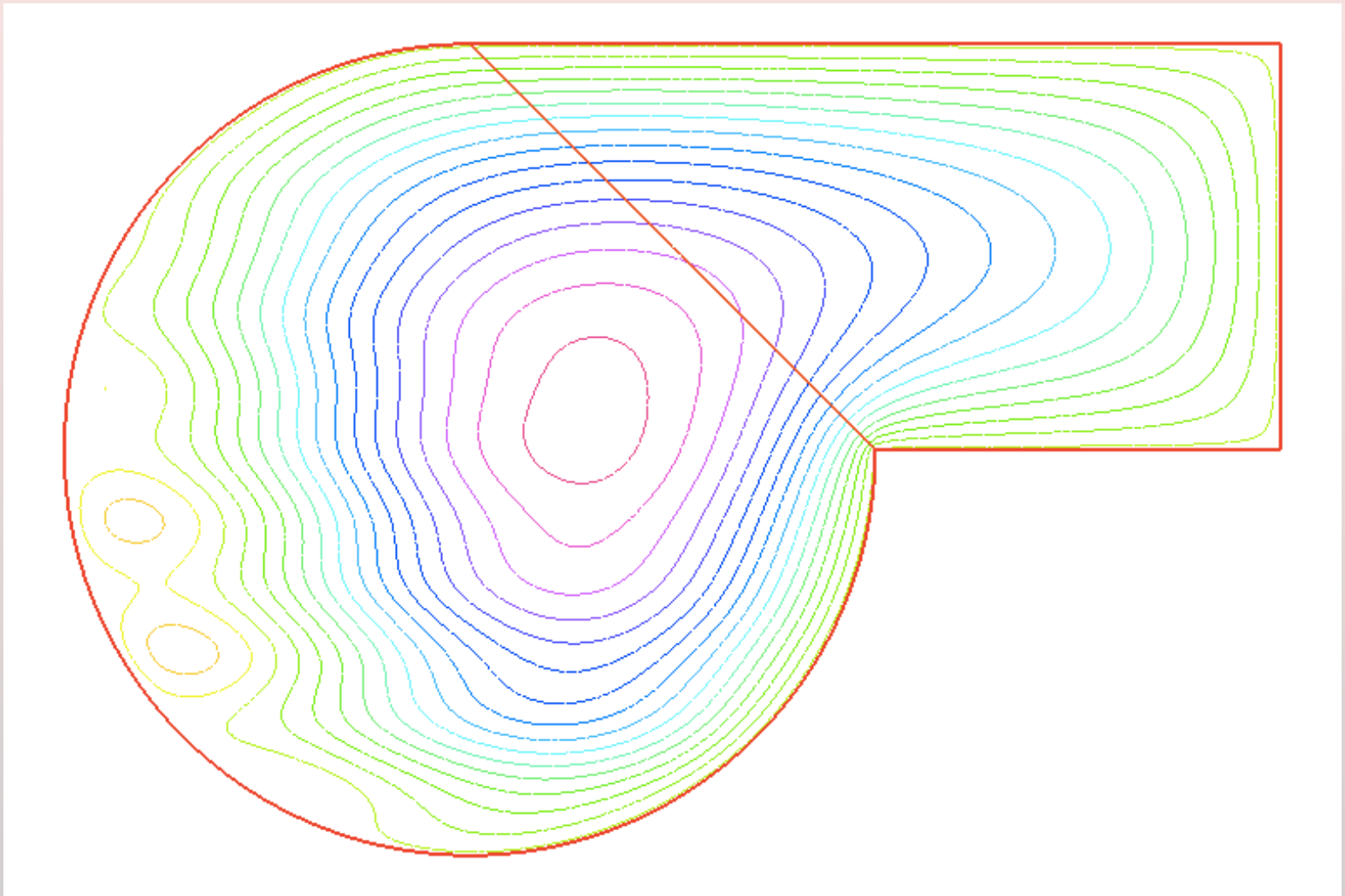


FIGURE: Solution over Ω for various parameters

Application to Laplace equation

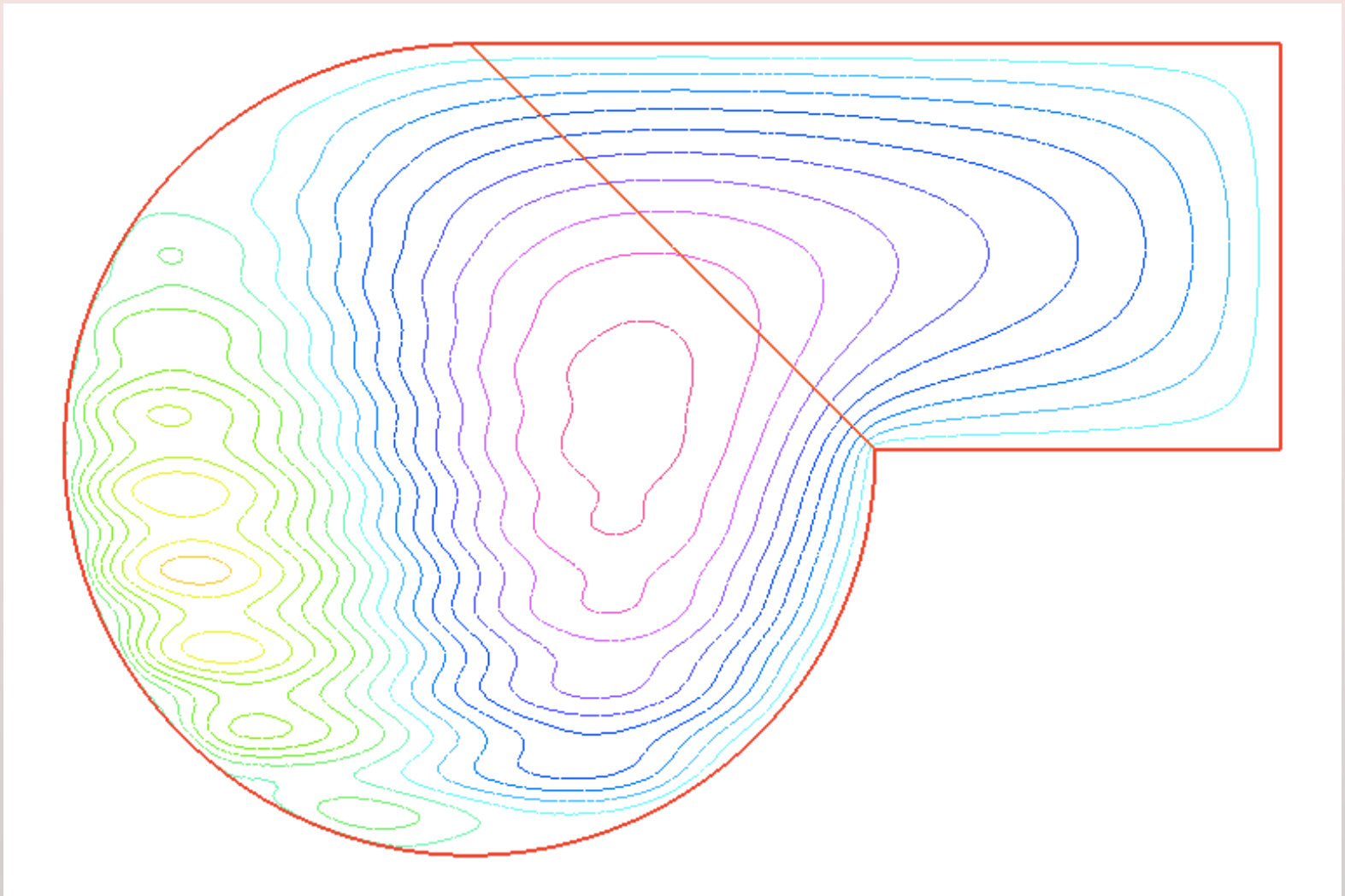


FIGURE: Solution over Ω for various parameters

Application to Laplace equation

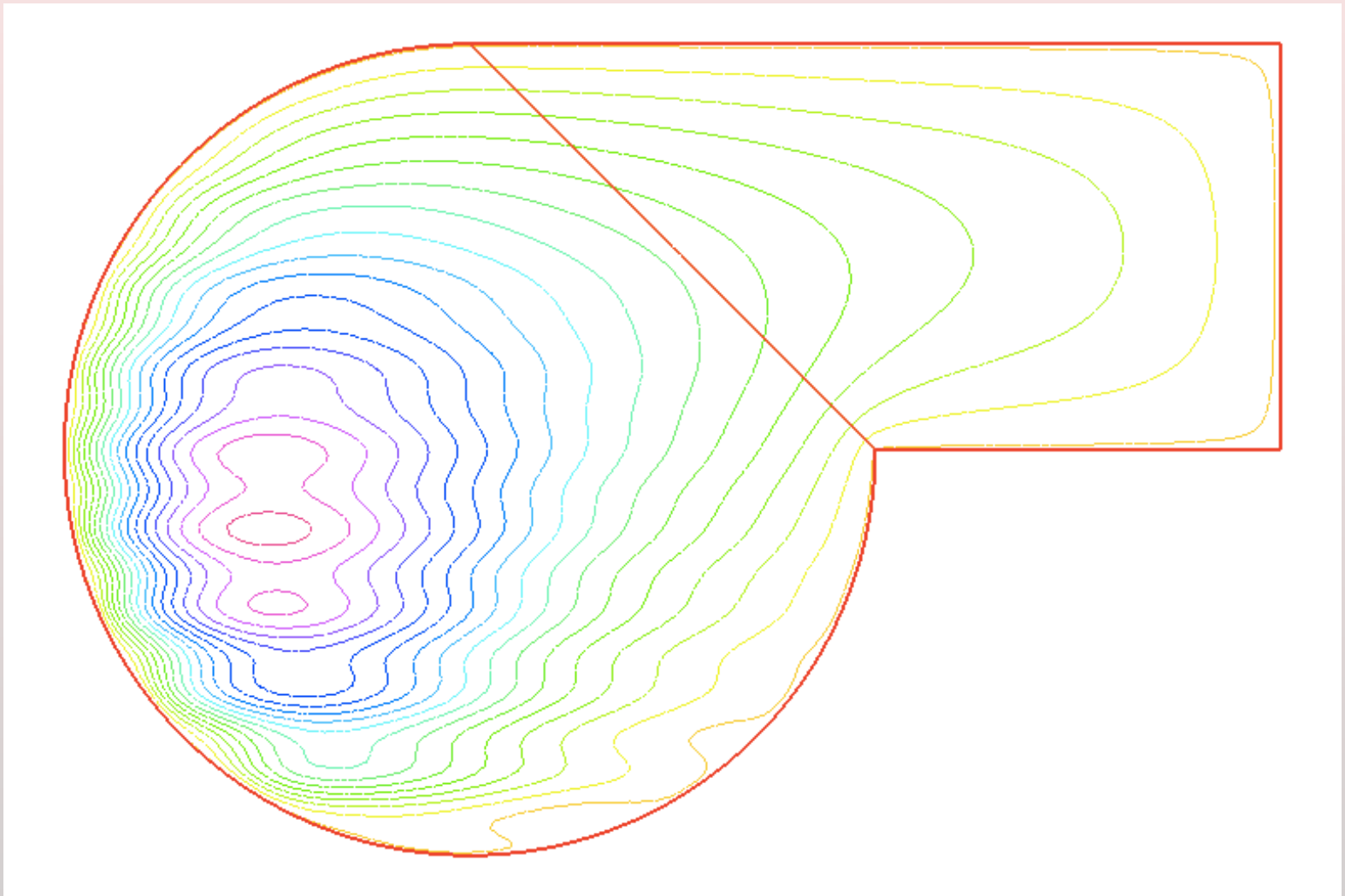


FIGURE: Solution over Ω for various parameters

1) Reconstruction of Ω_2 with GEIM

We shall use a reconstruction over the environment (Ω_2) by using data acquisition and GEIM.
i.e :

- **Offline :**
  - Computation of $u(.,\mu_1),\ldots,u(.,\mu_Q),\ Q \gg 1$ by an implicit method (finite elements)
  - Computation of the basis functions $\{q_i\}$ and selection of the appropriate captors (linear forms $\{\sigma_i\}$) by a Greedy process.
- **Online :** Given μ, computation of a general interpolant of $u(.,\mu)$ thanks to the data provided by the selected captors.

Solution over Ω_2 for various parameters

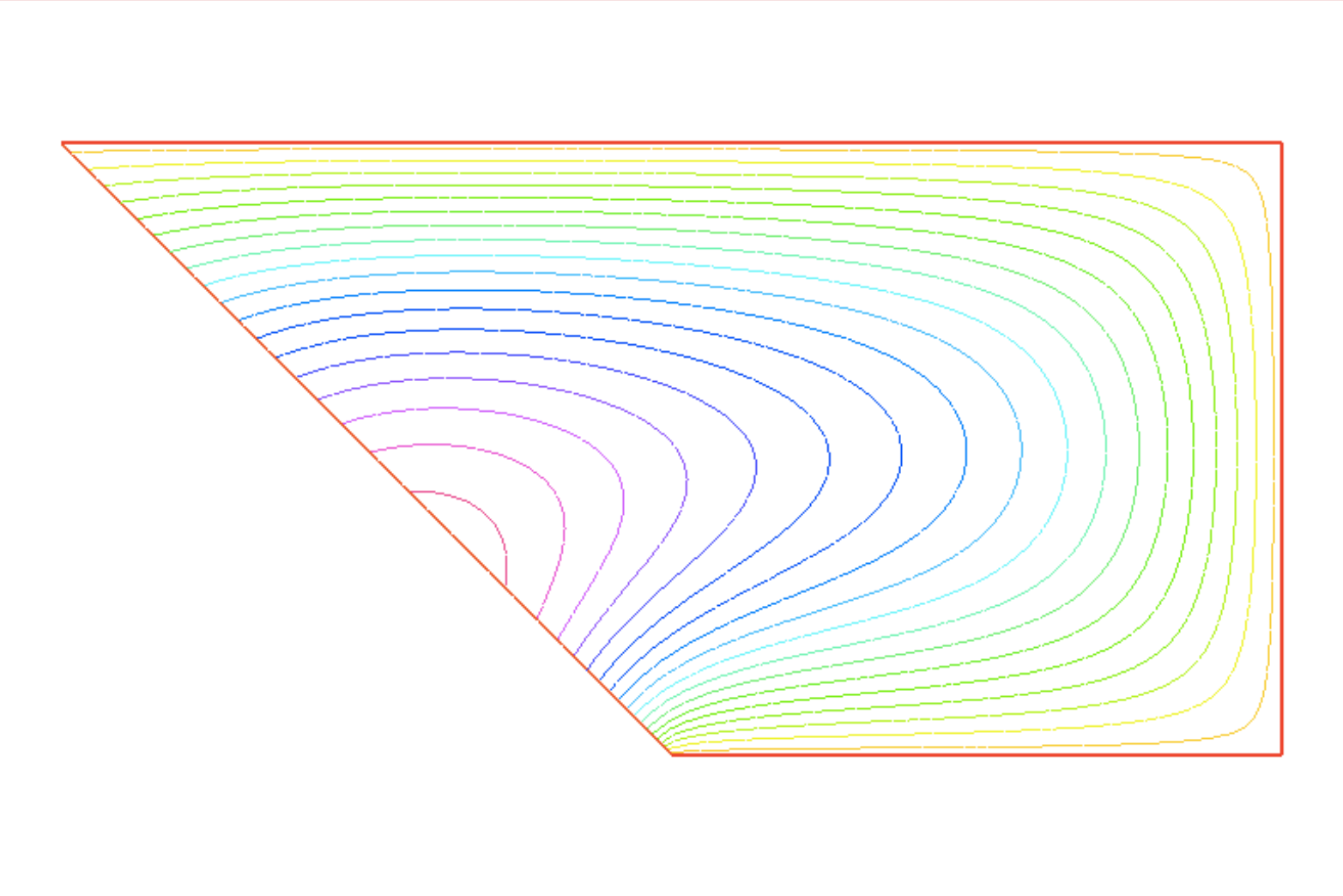


FIGURE: Solution 1.

Solution over Ω_2 for various parameters

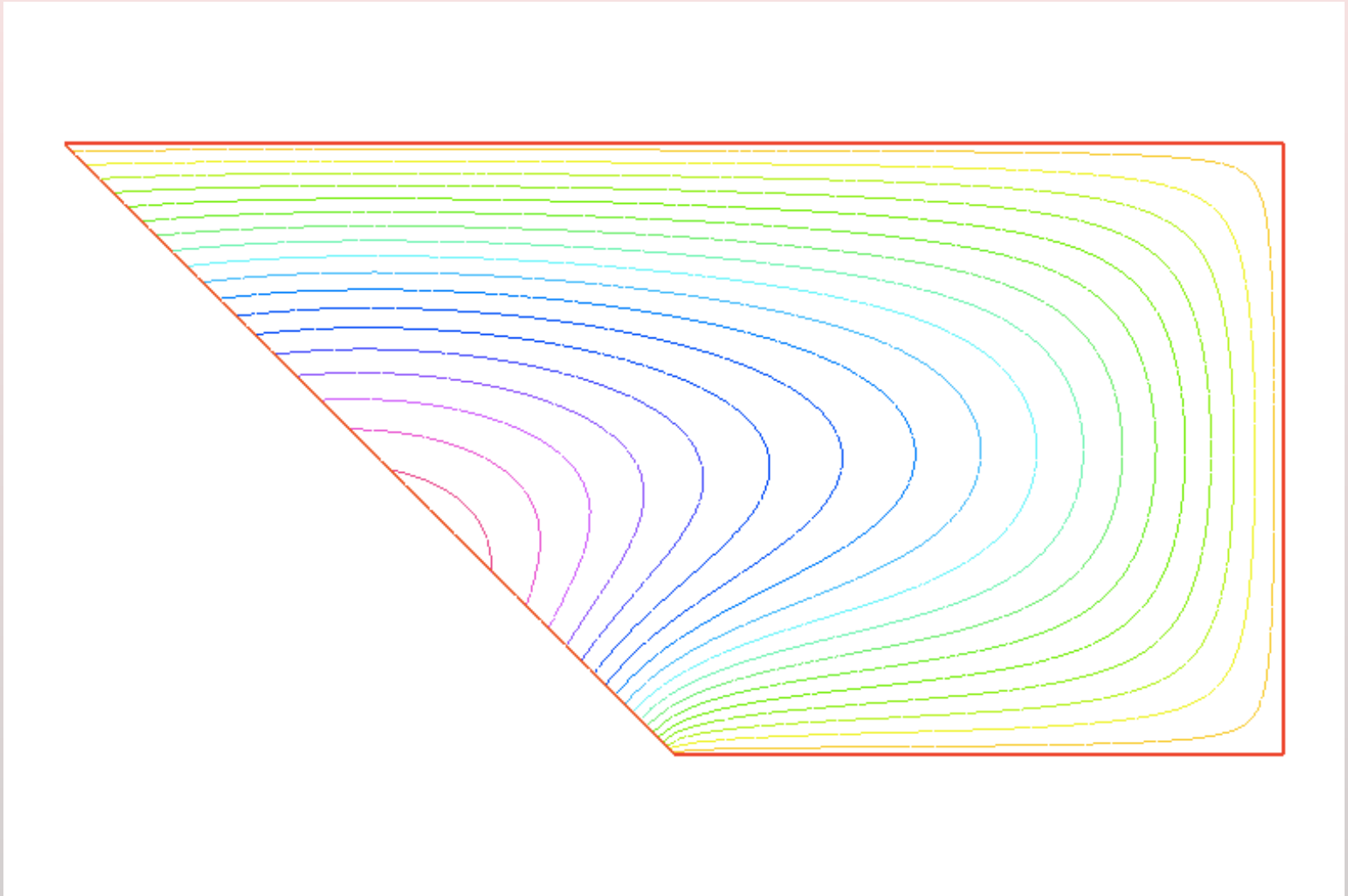


FIGURE: Solution 2.

Solution over Ω_2 for various parameters

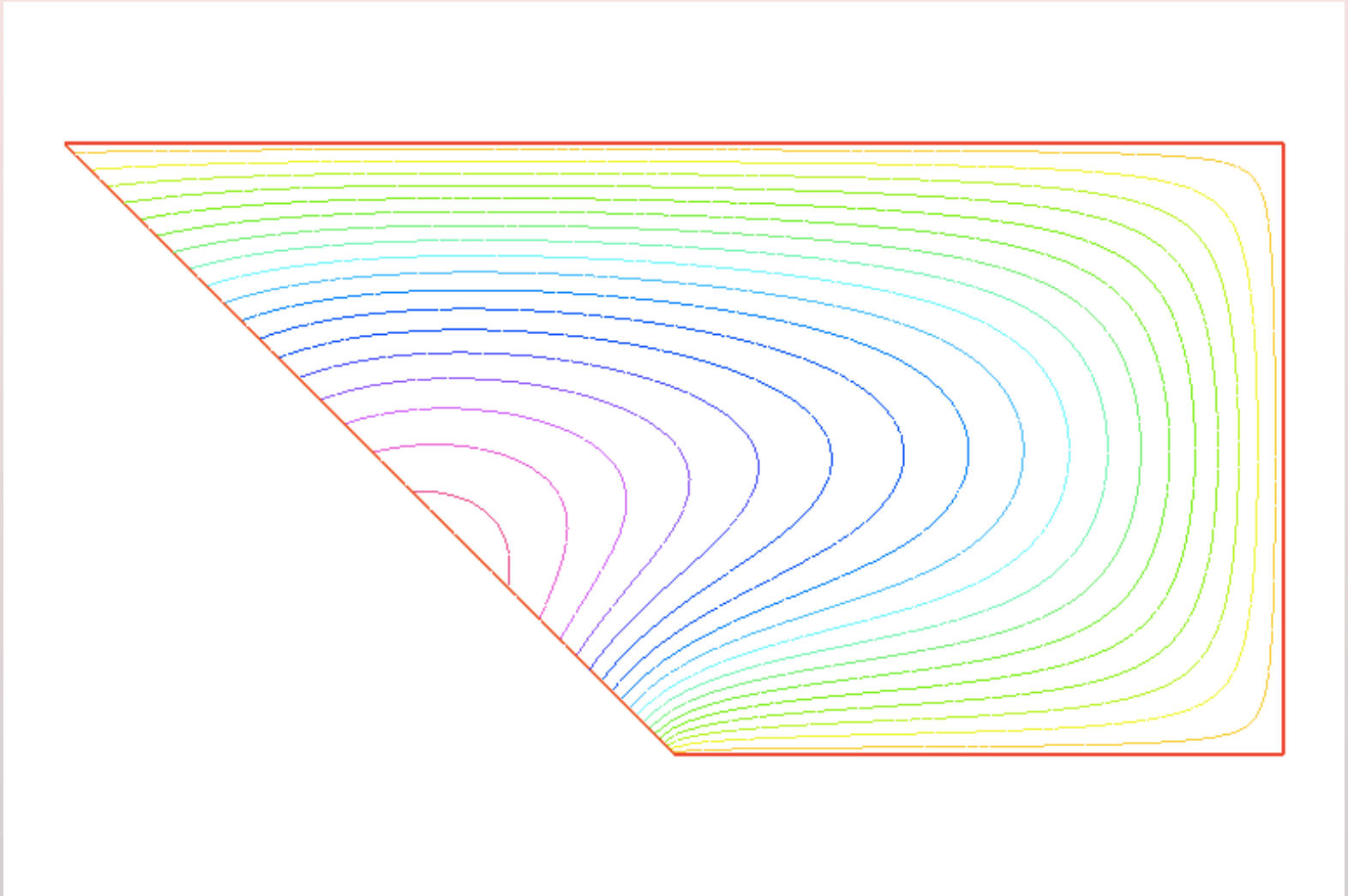


FIGURE: Solution 3.

1) Reconstruction over Ω_2 with GEIM

Reconstruction over Ω_2, via various moments via gaussian type

(L^2-error)^2 with 1 function/moment : $6\ 10^{-4}$
(L^2-error)^2 with 2 functions/moments : 110^{-7}
(L^2-error)^2 with 3 functions/moments : $8\ 10^{-9}$
(L^2-error)^2 with 4 functions/moments : $7\ 10^{-9}$
(L^2-error)^2 with 5 functions/moments : $2\ 10^{-21}$

1) Reconstruction over Ω_2 with GEIM

Reconstruction over Ω_2, via various moments via gaussian type

(L^2-error)^2 with 1 function/moment : 6 10^{-4}
(L^2-error)^2 with 2 functions/moments : 110^{-7}
(L^2-error)^2 with 3 functions/moments : 8 10^{-9}
(L^2-error)^2 with 4 functions/moments : 7 10^{-9}
(L^2-error)^2 with 5 functions/moments : 2 10^{-21}

1) Reconstruction over Ω_2 with GEIM

Reconstruction over Ω_2, via various moments via gaussian type

(L^2-error)^2 with 1 function/moment : 6 10^{-4}
(L^2-error)^2 with 2 functions/moments : 110^{-7}
(L^2-error)^2 with 3 functions/moments : 8 10^{-9}
(L^2-error)^2 with 4 functions/moments : 7 10^{-9}
(L^2-error)^2 with 5 functions/moments : 2 10^{-21}

1) Reconstruction over Ω_2 with GEIM

Reconstruction over Ω_2, via various moments via gaussian type

(L^2-error)^2 with 1 function/moment : 6 10^{-4}
(L^2-error)^2 with 2 functions/moments : 110^{-7}
(L^2-error)^2 with 3 functions/moments : 8 10^{-9}
(L^2-error)^2 with 4 functions/moments : 7 10^{-9}
(L^2-error)^2 with 5 functions/moments : 2 10^{-21}

1) Reconstruction over Ω_2 with GEIM

Reconstruction over Ω_2, via various moments via gaussian type

(L^2-error)^2 with 1 function/moment : 6 10^{-4}
(L^2-error)^2 with 2 functions/moments : 110^{-7}
(L^2-error)^2 with 3 functions/moments : 8 10^{-9}
(L^2-error)^2 with 4 functions/moments : 7 10^{-9}
(L^2-error)^2 with 5 functions/moments : 2 10^{-21}

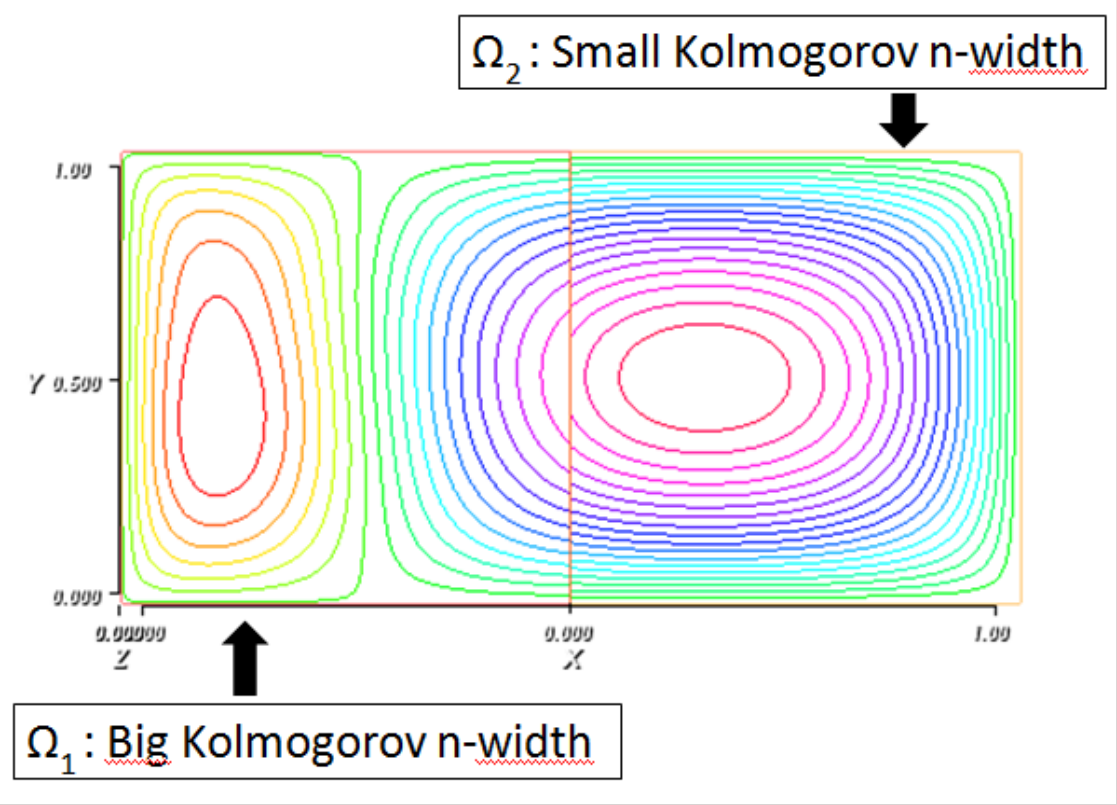
1) Reconstruction over Ω_2 with GEIM

Reconstruction over Ω_2, via various moments via gaussian type

$(L^2\text{-error})^2$ with 1 function/moment : $6\ 10^{-4}$
$(L^2\text{-error})^2$ with 2 functions/moments : 110^{-7}
$(L^2\text{-error})^2$ with 3 functions/moments : $8\ 10^{-9}$
$(L^2\text{-error})^2$ with 4 functions/moments : $7\ 10^{-9}$
$(L^2\text{-error})^2$ with 5 functions/moments : $2\ 10^{-21}$

1) Reconstruction over Ω_2 with GEIM

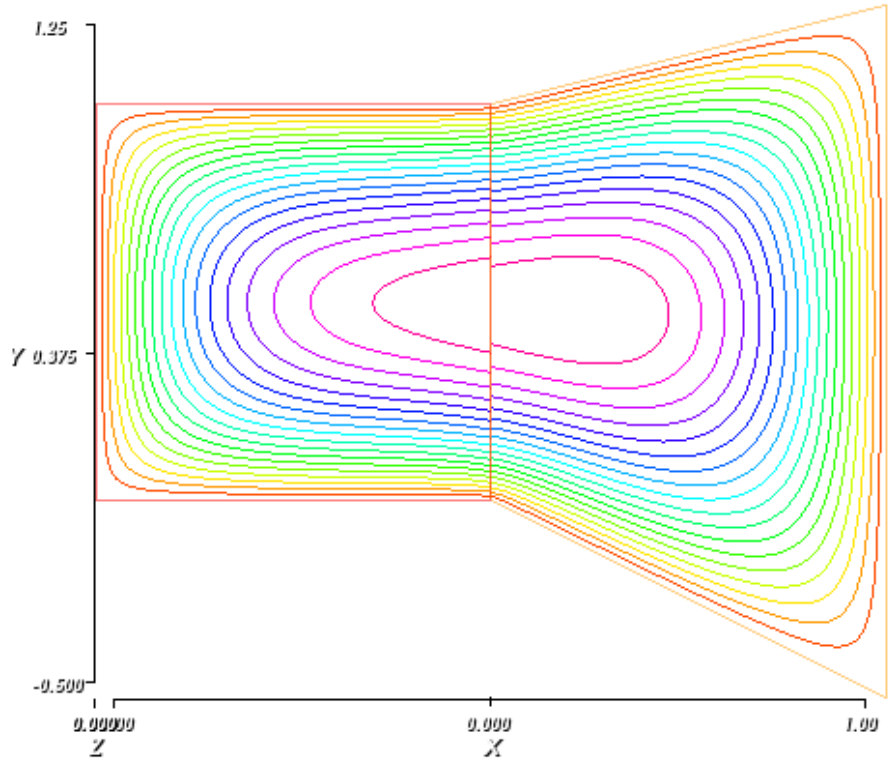
The geometry of Ω_2 is a further parameter



Parameters : everything that impacts the behavior of Ω_1, i.e. α, β, γ and the geometry of Ω_2.

1) Reconstruction over Ω_2 with GEIM

The geometry of Ω_2 is a further parameter



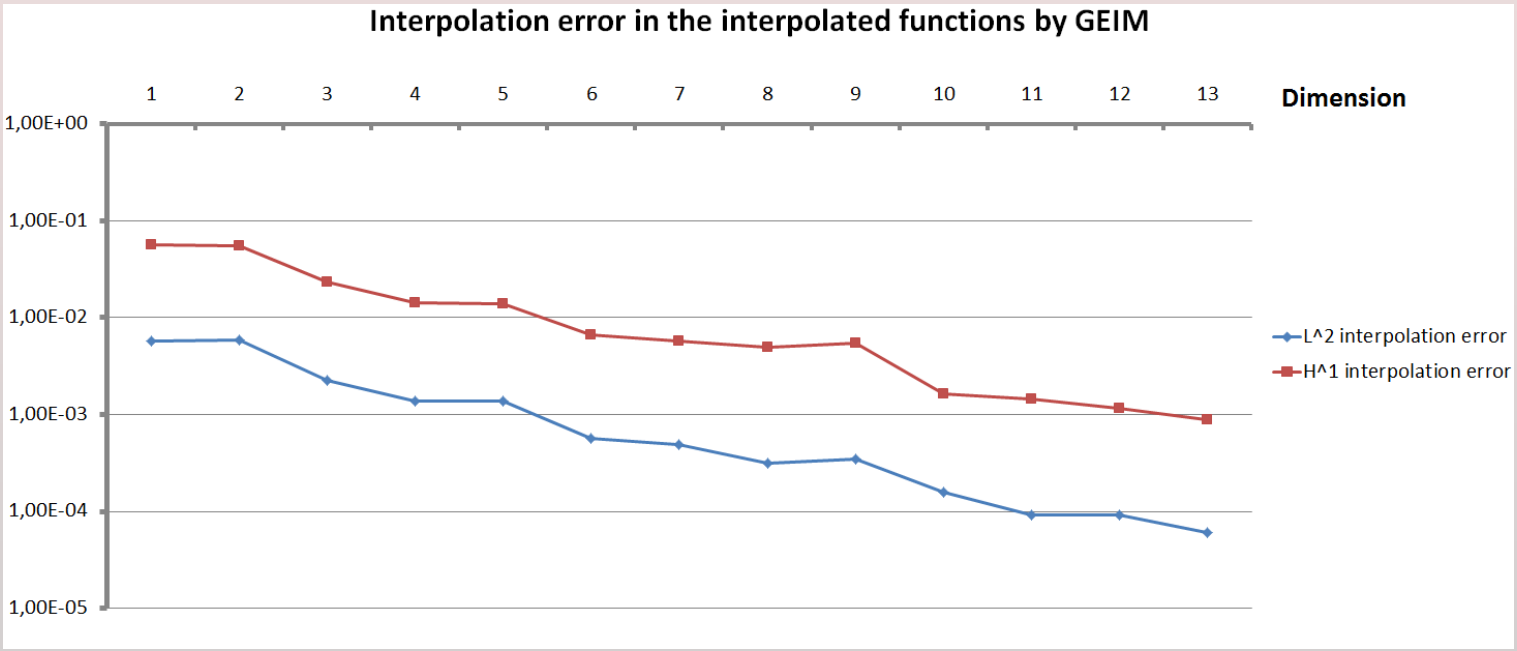
Parameters : everything that impacts the behavior of Ω_1, i.e. α, β, γ and the geometry of Ω_2.

1) Reconstruction of Ω_2 with GEIM

Interpolation error

We have reconstructed several solutions and computed for $1 \leq M \leq 13$:

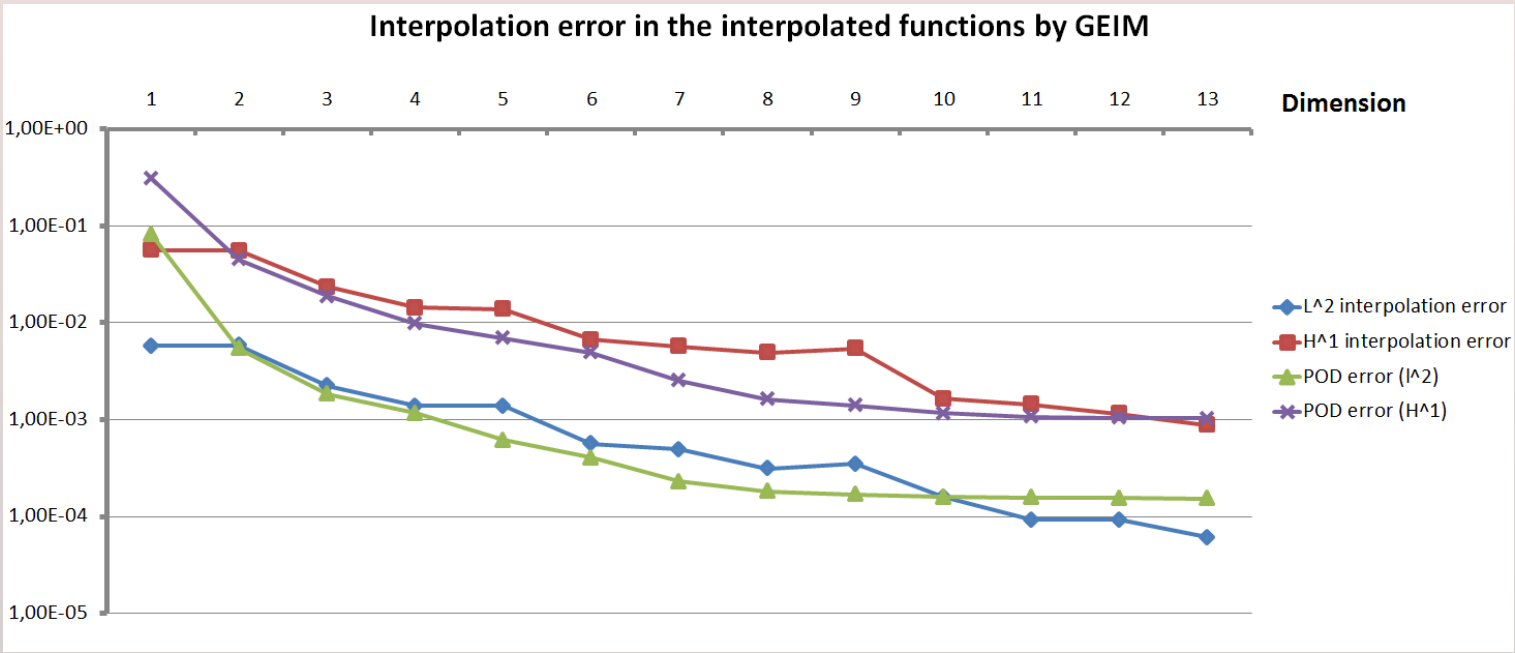
$$\max_{u \ \text{reconstructed}} \|u - \mathcal{I}_M[u]\|_* \ ; \ * = L^2(\Omega), \ H^1(\Omega)$$



1) Reconstruction of Ω_2 with GEIM

Comparison with POD

Our GEIM approach seems to have explored the set of solutions good enough to perform as well as POD.



1) Reconstruction of Ω_2 with GEIM

The FEM noise

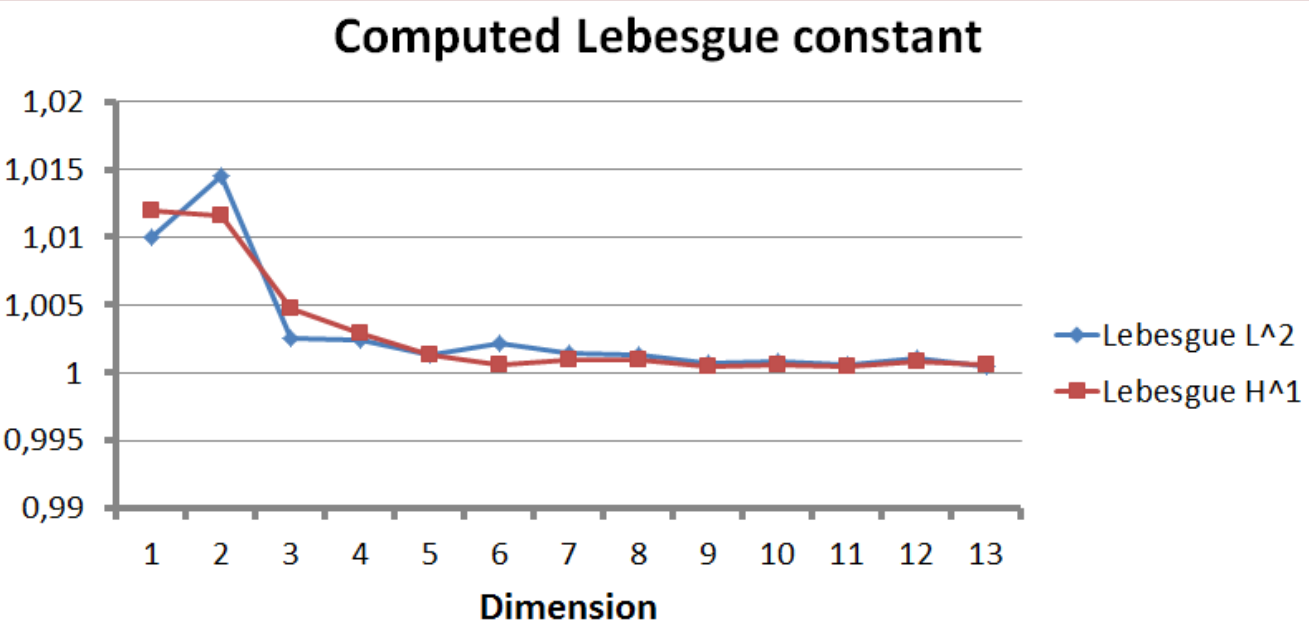
Going much beyond $M = 13$ is of no use since we reach the FEM precision ($\approx 10^{-4}$)

1) Reconstruction of Ω_2 with GEIM

The Lebesgue constant

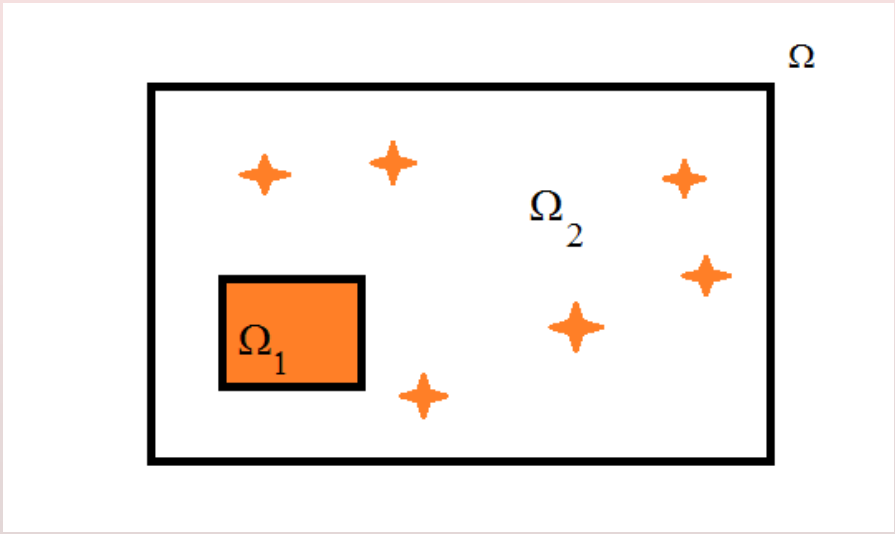
A computational estimation of Λ_M has been carried out :

$$\widetilde{\Lambda_M} = \max_{i \in [1,256]} \frac{\|\mathcal{I}_M[u_i]\|_{L^2(\Omega)}}{\|u_i\|_{L^2(\Omega)}}$$



2) Data acquisition coupled with simulation

Aim in the long run

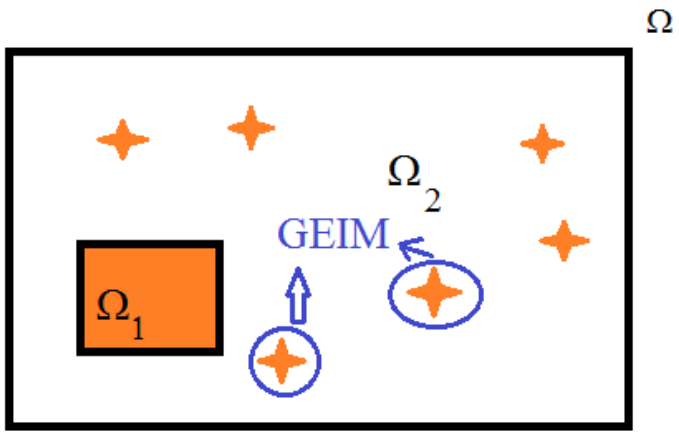


Imagine an experiment in a **huge** Ω and we wish a **real-time simulation** :

- Ω_1 : small subdomain but big Kolmogorov n-width. (e.g : wing, part of a nuclear core)
- Ω_2 : big subdomain but small Kolmogorov n-width (e.g. an aircraft, a nuclear core)

2) Data acquisition coupled with simulation

Aim in the long run

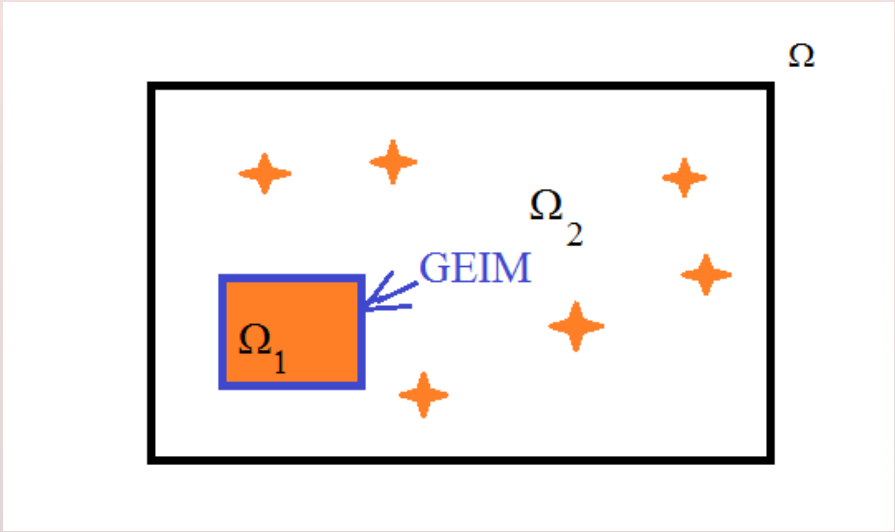


Imagine an experiment in a **huge** Ω and we wish a **real-time simulation** :

- Ω_1 : small but big Kolmogorov n-width. (e.g : a wing, a part of a nuclear core)
- Ω_2 : Computation of a fast approximation by GEIM

2) Data acquisition coupled with simulation

Aim in the long run



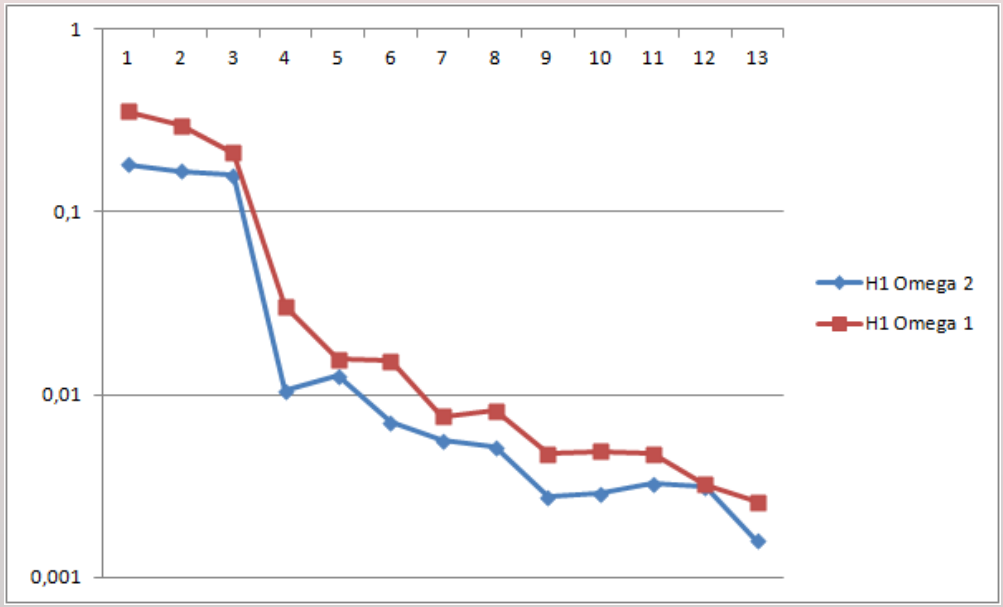
Imagine an experiment in a **huge** Ω and we wish a **real-time simulation** :

- Ω_1 : Computation by FEM.
- Ω_2 : Gives the appropriate boundary conditions to Ω_1.

2) Data acquisition coupled with simulation

Application to the Laplace problem

The interpolant in the environment Ω_2 can give the appropriate boundary conditions to the subdomain that has a big Kolmogorov n-width (Ω_1). We compute Ω_1 by FEM with the boundary conditions provided by the GEIM procedure.



3) Include uncertainty in the captors

Note that very few captors are used to recover a fair enough approximation of the solution.

There are generally more captors than required . . .at least if no noise in the datas.

The reconstruction is polluted with noise. . .from many sets of datas, we reconstruct a family of solutions, each polluted with noise.

An average allows to extract a better approximation together with average and standard deviations.

3) Include uncertainty in the captors

Note that very few captors are used to recover a fair enough approximation of the solution.

There are generally more captors than required . . .at least if no noise in the datas.

The reconstruction is polluted with noise. . .from many sets of datas, we reconstruct a family of solutions, each polluted with noise.

An average allows to extract a better approximation together with average and standard deviations

3) Include uncertainty in the captors

Note that very few captors are used to recover a fair enough approximation of the solution.

There are generally more captors than required . . .at least if no noise in the datas.

The reconstruction is polluted with noise. . .from many sets of datas, we reconstruct a family of solutions, each polluted with noise.

An average allows to extract a better approximation together with average and standard deviations

3) Include uncertainty in the captors

Note that very few captors are used to recover a fair enough approximation of the solution.

There are generally more captors than required . . .at least if no noise in the datas.

The reconstruction is polluted with noise. . .from many sets of datas, we reconstruct a family of solutions, each polluted with noise.

An average allows to extract a better approximation together with average and standard deviations

3) Include uncertainty in the captors

Note that very few captors are used to recover a fair enough approximation of the solution.

There are generally more captors than required . . .at least if no noise in the datas.

The reconstruction is polluted with noise. . .from many sets of datas, we reconstruct a family of solutions, each polluted with noise.

An average allows to extract a better approximation together with average and standard deviations

3) Include uncertainty in the captors

Note that very few captors are used to recover a fair enough approximation of the solution.

There are generally more captors than required . . .at least if no noise in the datas.

The reconstruction is polluted with noise. . .from many sets of datas, we reconstruct a family of solutions, each polluted with noise.

An average allows to extract a better approximation together with average and standard deviations

Conclusion and Future - RBM

- The reduced basis method is a powerfull method
- Combined with a posteriori error estimators it allows to derive *online* hyper fast reliable computations
- Many parameters can be used to define the reduced basis : geometry, time....
- Greedy (especially local version) is a way to minimize the number of RB functions that are computed and used at the end
- The application to highly non linear problems is at end thanks to the *magic points, for which there also exists a a posteriori estimation*
- Can be used as a coarse solver for preconditionner – either algebraic – or for the parareal algorithm (with Li Ping He)
- Industrial code : fully non intrusive with Rachida Chakir and Tigran A. Vlagandeyan
- You can also use accurate discrete solutions that might have been given to you ...

Conclusion and Future - RBM

- The reduced basis method is a powerfull method
- Combined with a posteriori error estimators it allows to derive *online* hyper fast reliable computations
- Many parameters can be used to define the reduced basis : geometry, time....
- Greedy (especially local version) is a way to minimize the number of RB functions that are computed and used at the end
- The application to highly non linear problems is at end thanks to the *magic points, for which there also exists a a posteriori estimation*
- Can be used as a coarse solver for preconditionner – either algebraic – or for the parareal algorithm (with Li Ping He)
- Industrial code ..... fully non intrusive with Rachida Chakir and *Tigran A. Nagapetyan*
- You can also use accurate discrete solutions that might have been given to you ....

Conclusion and Future - RBM

- The reduced basis method is a powerfull method
- Combined with a posteriori error estimators it allows to derive *online* hyper fast reliable computations
- Many parameters can be used to define the reduced basis : geometry, time....
- Greedy (especially local version) is a way to minimize the number of RB functions that are computed and used at the end
- The application to highly non linear problems is at end thanks to the *magic points, for which there also exists a a posteriori estimation*
- Can be used as a coarse solver for preconditionner – either algebraic – or for the parareal algorithm (with Li Ping He)
- Industrial code ... fully non intrusive with Rachida Chakir and *Tigran A. Nagapetyan*
- You can also use accurate discrete solutions that might have been given to you ....

Conclusion and Future - RBM

- The reduced basis method is a powerfull method
- Combined with a posteriori error estimators it allows to derive *online* hyper fast reliable computations
- Many parameters can be used to define the reduced basis : geometry, time....
- Greedy (especially local version) is a way to minimize the number of RB functions that are computed and used at the end
- The application to highly non linear problems is at end thanks to the *magic points, for which there also exists a a posteriori estimation*
- Can be used as a coarse solver for preconditionner – either algebraic – or for the parareal algorithm (with Li Ping He)
- Industrial code . . . fully non intrusive with Rachida Chakir and Tegran A. Wanganeyan
- You can also use accurate discrete solutions that might have been given to you . . .

Conclusion and Future - RBM

- The reduced basis method is a powerfull method
- Combined with a posteriori error estimators it allows to derive *online* hyper fast reliable computations
- Many parameters can be used to define the reduced basis : geometry, time....
- Greedy (especially local version) is a way to minimize the number of RB functions that are computed and used at the end
- The application to highly non linear problems is at end thanks to the *magic points, for which there also exists a a posteriori estimation*
- Can be used as a coarse solver for preconditionner – either algebraic – or for the parareal algorithm (with Li Ping He)
- Industrial code . . . fully non intrusive with Rachida Chakir and Tigran A. Nagapetyan
- You can also use accurate discrete solutions that might have been given to you . . .

Conclusion and Future - RBM

- The reduced basis method is a powerfull method
- Combined with a posteriori error estimators it allows to derive *online* hyper fast reliable computations
- Many parameters can be used to define the reduced basis : geometry, time....
- Greedy (especially local version) is a way to minimize the number of RB functions that are computed and used at the end
- The application to highly non linear problems is at end thanks to the *magic points, for which there also exists a a posteriori estimation*
- Can be used as a coarse solver for preconditionner – either algebraic – or for the parareal algorithm (with Li Ping He)
- Industrial code . . . fully non intrusive with Rachida Chakir and Tigran A. Nagapetyan
- You can also use accurate discrete solutions that might have been given to you . . .

Conclusion and Future - RBM

- The reduced basis method is a powerfull method
- Combined with a posteriori error estimators it allows to derive *online* hyper fast reliable computations
- Many parameters can be used to define the reduced basis : geometry, time....
- Greedy (especially local version) is a way to minimize the number of RB functions that are computed and used at the end
- The application to highly non linear problems is at end thanks to the *magic points, for which there also exists a a posteriori estimation*
- Can be used as a coarse solver for preconditionner – either algebraic – or for the parareal algorithm (with Li Ping He)
- Industrial code . . . fully non intrusive with Rachida Chakir and Tigran A. Nagapetyan
- You can also use accurate discrete solutions that might have been given to you . . .

Conclusion and Future - RBM

- The reduced basis method is a powerfull method
- Combined with a posteriori error estimators it allows to derive *online* hyper fast reliable computations
- Many parameters can be used to define the reduced basis : geometry, time....
- Greedy (especially local version) is a way to minimize the number of RB functions that are computed and used at the end
- The application to highly non linear problems is at end thanks to the *magic points, for which there also exists a a posteriori estimation*
- Can be used as a coarse solver for preconditionner – either algebraic – or for the parareal algorithm (with Li Ping He)
- Industrial code . . . fully non intrusive with Rachida Chakir and Tigran A. Nagapetyan
- You can also use accurate discrete solutions that might have been given to you . . .

Conclusion and Future - RBM

- The reduced basis method is a powerfull method
- Combined with a posteriori error estimators it allows to derive *online* hyper fast reliable computations
- Many parameters can be used to define the reduced basis : geometry, time....
- Greedy (especially local version) is a way to minimize the number of RB functions that are computed and used at the end
- The application to highly non linear problems is at end thanks to the *magic points, for which there also exists a a posteriori estimation*
- Can be used as a coarse solver for preconditionner – either algebraic – or for the parareal algorithm (with Li Ping He)
- Industrial code . . . fully non intrusive with Rachida Chakir and Tigran A. Nagapetyan
- You can also use accurate discrete solutions that might have been given to you . . .

Conclusion and Future - RBM

- The reduced basis method is a powerfull method
- Combined with a posteriori error estimators it allows to derive *online* hyper fast reliable computations
- Many parameters can be used to define the reduced basis : geometry, time....
- Greedy (especially local version) is a way to minimize the number of RB functions that are computed and used at the end
- The application to highly non linear problems is at end thanks to the *magic points, for which there also exists a a posteriori estimation*
- Can be used as a coarse solver for preconditionner – either algebraic – or for the parareal algorithm (with Li Ping He)
- Industrial code . . . fully non intrusive with Rachida Chakir and Tigran A. Nagapetyan
- You can also use accurate discrete solutions that might have been given to you . . .

Conclusion and Future - EIM

- The magic point approach has been introduced for collocation approach in reduced basis approximation methods
- It appears as a simple and flexible approach to the general problem of interpolation
- The comparison in case where we know the behavior illustrates the ability of the method.
- Of course we could optimize the Lebesgue constant, but our approach is hierarchical
- Many possible applications for approximation on various discrete spaces, and of course for numerical integration
- Has really allowed a breakthrough in reduced basis approximations (optimal complexity)
- There exist a posteriori estimators
- GEIM

Conclusion and Future - EIM

- The magic point approach has been introduced for collocation approach in reduced basis approximation methods
- It appears as a simple and flexible approach to the general problem of interpolation
- The comparison in case where we know the behavior illustrates the ability of the method.
- Of course we could optimize the Lebesgue constant, but our approach is hierarchical
- Many possible applications for approximation on various discrete spaces, and of course for numerical integration
- Has really allowed a breakthrough in reduced basis approximations (optimal complexity)
- There exist a posteriori estimators
- GEIM

Conclusion and Future - EIM

- The magic point approach has been introduced for collocation approach in reduced basis approximation methods
- It appears as a simple and flexible approach to the general problem of interpolation
- The comparison in case where we know the behavior illustrates the ability of the method.
- Of course we could optimize the Lebesgue constant, but our approach is hierarchical
- Many possible applications for approximation on various discrete spaces, and of course for numerical integration
- Has really allowed a breakthrough in reduced basis approximations (optimal complexity)
- There exist a posteriori estimators
- GEIM

Conclusion and Future - EIM

- The magic point approach has been introduced for collocation approach in reduced basis approximation methods
- It appears as a simple and flexible approach to the general problem of interpolation
- The comparison in case where we know the behavior illustrates the ability of the method.
- Of course we could optimize the Lebesgue constant, but our approach is hierarchical
- Many possible applications for approximation on various discrete spaces, and of course for numerical integration
- Has really allowed a breakthrough in reduced basis approximations (optimal complexity)
- There exist a posteriori estimators
- GEIM

Conclusion and Future - EIM

- The magic point approach has been introduced for collocation approach in reduced basis approximation methods
- It appears as a simple and flexible approach to the general problem of interpolation
- The comparison in case where we know the behavior illustrates the ability of the method.
- Of course we could optimize the Lebesgue constant, but our approach is hierarchical
- Many possible applications for approximation on various discrete spaces, and of course for numerical integration
- Has really allowed a breakthrough in reduced basis approximations (optimal complexity)
- There exist a posteriori estimators
- GEIM

Conclusion and Future - EIM

- The magic point approach has been introduced for collocation approach in reduced basis approximation methods
- It appears as a simple and flexible approach to the general problem of interpolation
- The comparison in case where we know the behavior illustrates the ability of the method.
- Of course we could optimize the Lebesgue constant, but our approach is hierarchical
- Many possible applications for approximation on various discrete spaces, and of course for numerical integration
- Has really allowed a breakthrough in reduced basis approximations (optimal complexity)
- There exist a posteriori estimators
- GEIM

Conclusion and Future - EIM

- The magic point approach has been introduced for collocation approach in reduced basis approximation methods
- It appears as a simple and flexible approach to the general problem of interpolation
- The comparison in case where we know the behavior illustrates the ability of the method.
- Of course we could optimize the Lebesgue constant, but our approach is hierarchical
- Many possible applications for approximation on various discrete spaces, and of course for numerical integration
- Has really allowed a breakthrough in reduced basis approximations (optimal complexity)
- There exist a posteriori estimators
- GEIM

Conclusion and Future - EIM

- The magic point approach has been introduced for collocation approach in reduced basis approximation methods
- It appears as a simple and flexible approach to the general problem of interpolation
- The comparison in case where we know the behavior illustrates the ability of the method.
- Of course we could optimize the Lebesgue constant, but our approach is hierarchical
- Many possible applications for approximation on various discrete spaces, and of course for numerical integration
- Has really allowed a breakthrough in reduced basis approximations (optimal complexity)
- There exist a posteriori estimators
- GEIM

Conclusion and Future - RBEM - NIRB

- The coupling between RBM and domain decomposition methods allows to tackle very large problems where the geometry is a parameter.
- Note that contrarily to what happens in the parameter dependant problem
- The full problem over the global geometry is never constructed in the reduced element method
- This is a major achievement
- NIRB gives a lot of flexibility with respect to the master code

Conclusion and Future - RBEM - NIRB

- The coupling between RBM and domain decomposition methods allows to tackle very large problems where the geometry is a parameter.
- Note that contrarily to what happens in the parameter dependant problem
- The full problem over the global geometry is never constructed in the reduced element method
- This is a major achievement
- NIRB gives a lot of flexibility with respect to the master code

Conclusion and Future - RBEM - NIRB

- The coupling between RBM and domain decomposition methods allows to tackle very large problems where the geometry is a parameter.
- Note that contrarily to what happens in the parameter dependant problem
- The full problem over the global geometry is never constructed in the reduced element method
- This is a major achievement
- NIRB gives a lot of flexibility with respect to the master code

Conclusion and Future - RBEM - NIRB

- The coupling between RBM and domain decomposition methods allows to tackle very large problems where the geometry is a parameter.
- Note that contrarily to what happens in the parameter dependant problem
- The full problem over the global geometry is never constructed in the reduced element method
- This is a major achievement
- NIRB gives a lot of flexibility with respect to the master code

Conclusion and Future - RBEM - NIRB

- The coupling between RBM and domain decomposition methods allows to tackle very large problems where the geometry is a parameter.
- Note that contrarily to what happens in the parameter dependant problem
- The full problem over the global geometry is never constructed in the reduced element method
- This is a major achievement
- NIRB gives a lot of flexibility with respect to the master code

Thanks...

... Thanks ....

Thanks...

... Thanks ....

We understand here that the reduce basis method is not an alternative to your favorite code but a companion technique that allows to enhance its features.

In the case we do not master the code, or if you do not want to get into the use of the magic points, you cannot implement the reduced basis method with an optimal complexity.

We understand here that the reduce basis method is not an alternative to your favorite code but a companion technique that allows to enhance its features.

In the case we do not master the code, or if you do not want to get into the use of the magic points, you cannot implement the reduced basis method with an optimal complexity.

What do we actually seek ? ? ?

We seek the best approximation of $u(\mu)$ in the space spanned by the $u_h(\mu_i)$

and the best approximation is the orthogonal projection of $u(\mu)$ in the finite element space spanned by the $u_h(\mu_i)$

which is more easy to handle with an orthonormal basis

We seek the best approximation of $u(\mu)$ in the space spanned by the $u_h(\mu_i)$

and the best approximation is the orthogonal projection of $u(\mu)$ in the finite element space spanned by the $u_h(\mu_i)$

which is more easy to handle with an orthonormal basis

We seek the best approximation of $u(\mu)$ in the space spanned by the $u_h(\mu_i)$

and the best approximation is the orthogonal projection of $u(\mu)$ in the finite element space spanned by the $u_h(\mu_i)$

which is more easy to handle with an orthonormal basis

- Let us consider the following eigenvalue problem : find $\xi \in X_h^N$ and $\lambda \in \mathbb{R}$ such that

$$\forall v \in X_h^N, \qquad \int_\Omega \nabla \xi \nabla v = \lambda \int_\Omega \xi v.$$

- We obtain a set of eigenvalue, that we rank in increasing order λ_i^{BR} and associated eigenvectors ξ_i^{BR} that constitute an orthonormal basis of the space, X_h^N both in L^2 and in H^1 ! ! .

- Let us consider the following eigenvalue problem : find $\xi \in X_h^N$ and $\lambda \in \mathbb{R}$ such that

$$\forall v \in X_h^N, \qquad \int_\Omega \nabla \xi \nabla v = \lambda \int_\Omega \xi v.$$

- We obtain a set of eigenvalue, that we rank in increasing order λ_i^{BR} and associated eigenvectors ξ_i^{BR} that constitute an orthonormal basis of the space, X_h^N both in L^2 and in H^1 ! ! .

- Let $u_N^{hP}(\mu)$ be the best approximation of $u_h(\mu)$ in X_N, it is defined as

$$u_N^{hP}(\mu) = \sum_{i=1}^{N} \beta_i^h(\mu)\xi_i^{BR} \qquad \text{where } \beta_i^h(\mu) = \int_\Omega u_h(\mu)\xi_i^{BR}$$

- The reduced basis method is an alternative procedure to solve this best approximation without knowing $u_h(\mu)$

- We look for

$$u_h^N(\mu) = \sum_{i=1}^{N} \gamma_i^h(\mu)\xi_i^{BR},$$

where the $\gamma_i^h(\mu)$ are substitute to the optimal coefficients $\beta_i^h(\mu)$.

- The difficulty of the reduced basis method is to implement it with a complexity that is linked to N (the dimension of the reduced basis space) and not $\mathcal{N}$ (the dimension of the finite element space X_h)
- Black box approaches have to be used, and this involves modifications inside the original finite element code
- this leads to the difference of "*offline*" and "*online*" computations

- The difficulty of the reduced basis method is to implement it with a complexity that is linked to N (the dimension of the reduced basis space) and not $\mathcal{N}$ (the dimension of the finite element space X_h)
- Black box approaches have to be used, and this involves modifications inside the original finite element code
- this leads to the difference of "*offline*" and "*online*" computations

- The difficulty of the reduced basis method is to implement it with a complexity that is linked to N (the dimension of the reduced basis space) and not $\mathcal{N}$ (the dimension of the finite element space X_h)
- Black box approaches have to be used, and this involves modifications inside the original finite element code
- this leads to the difference of "*offline*" and "*online*" computations

Let us assume that we do not want — or even it is impossible — to modify the finite element code.

- The computation of $u_H(\mu)$, for $H >> h$, being cheaper that the computation of $u_h(\mu)$, we propose to replace the $\beta_i^h(\mu)$ by $\beta_i^H(\mu) = \int_\Omega u_H(\mu)\xi_i^{BR}$ and then construct

$$u_N^{Hh}(\mu) = \sum_{i=1}^{N} \beta_i^H(\mu)\xi_i^{BR}.$$

- We remark that

$$|\beta_i^h(\mu) - \beta_i^H(\mu)| \leq \|u_h(\mu) - u_H(\mu)\|_{0,\Omega}$$

and this is Aubin Nietsche's trick

$$\|u(\mu) - u_H(\mu)\|_{0,\Omega} \leq cH\|u(\mu) - u_H(\mu)\|_X \leq cH^2$$

- It is then possible to get $\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq c(N) + cH \leq cH$ that is asymptotically similar to (7) provided that we choose $h \sim H^2$

- The computation of $u_H(\mu)$, for $H >> h$, being cheaper that the computation of $u_h(\mu)$, we propose to replace the $\beta_i^h(\mu)$ by $\beta_i^H(\mu) = \int_\Omega u_H(\mu)\xi_i^{BR}$ and then construct

$$u_N^{Hh}(\mu) = \sum_{i=1}^{N} \beta_i^H(\mu)\xi_i^{BR}.$$

- We remark that

$$|\beta_i^h(\mu) - \beta_i^H(\mu)| \leq \|u_h(\mu) - u_H(\mu)\|_{0,\Omega}$$

and this is Aubin Nietsche's trick

$$\|u(\mu) - u_H(\mu)\|_{0,\Omega} \leq cH\|u(\mu) - u_H(\mu)\|_X \leq cH^2$$

- It is then possible to get : $\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq \varepsilon(N) + c_5 h + c_6 H^2$ that is asymptotically similar to (7) provided that we choose $h \sim H^2$.

- The computation of $u_H(\mu)$, for $H >> h$, being cheaper that the computation of $u_h(\mu)$, we propose to replace the $\beta_i^h(\mu)$ by $\beta_i^H(\mu) = \int_\Omega u_H(\mu)\xi_i^{BR}$ and then construct

$$u_N^{Hh}(\mu) = \sum_{i=1}^{N} \beta_i^H(\mu)\xi_i^{BR}.$$

- We remark that

$$|\beta_i^h(\mu) - \beta_i^H(\mu)| \leq \|u_h(\mu) - u_H(\mu)\|_{0,\Omega}$$

and this is Aubin Nietsche's trick

$$\|u(\mu) - u_H(\mu)\|_{0,\Omega} \leq cH\|u(\mu) - u_H(\mu)\|_X \leq cH^2$$

- It is then possible to get : $\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq \varepsilon(N) + c_5 h + c_6 H^2$ that is asymptotically similar to (7) provided that we choose $h \sim H^2$.

- The computation of $u_H(\mu)$, for $H >> h$, being cheaper that the computation of $u_h(\mu)$, we propose to replace the $\beta_i^h(\mu)$ by $\beta_i^H(\mu) = \int_\Omega u_H(\mu)\xi_i^{BR}$ and then construct

$$u_N^{Hh}(\mu) = \sum_{i=1}^{N} \beta_i^H(\mu)\xi_i^{BR}.$$

- We remark that

$$|\beta_i^h(\mu) - \beta_i^H(\mu)| \leq \|u_h(\mu) - u_H(\mu)\|_{0,\Omega}$$

and this is Aubin Nietsche's trick

$$\|u(\mu) - u_H(\mu)\|_{0,\Omega} \leq cH\|u(\mu) - u_H(\mu)\|_X \leq cH^2$$

- It is then possible to get : $\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq \varepsilon(N) + c_5 h + c_6 H^2$ that is asymptotically similar to (7) provided that we choose $h \sim H^2$.

- In the case of $\mathbb{P}_2$ finite elements, inequatity (7) becomes

$$\|u(\mu) - u_h^N(\mu)\|_X \leq \varepsilon(N) + c_2 h^2. \tag{15}$$

- Let Φ_i^{BR} be the dual functions associated with ξ_i^{BR} be such that

$$\forall v \in X, \qquad a(\Phi_i^{BR}, v; \mu) = \int_\Omega \xi_i^{BR} v$$

- Then

$$\begin{aligned} \beta_i^h(\mu) - \beta_i^H(\mu) &= \int_\Omega \xi_i^{BR}(u_h(\mu) - u_H(\mu)) \\ &= a(u_h(\mu) - u_H(\mu), \Phi_i^{BR}; \mu) \end{aligned}$$

- Since $X_H \subset X_h$, then by definition of $u_h(\mu)$ and $u_H(\mu)$, we get

$$\forall w_H \in X_H, \qquad a(u_h(\mu) - u_H(\mu), w_H; \mu) = 0$$

- Which leads to

$$|\beta_i^h(\mu) - \beta_i^H| \leq c\|u_h(\mu) - u_H(\mu)\|_X \|\Phi_i^{BR} - w_H; \mu)\|_X \leq cH^4$$

and thus

$$\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq \varepsilon(N) + c_7 h^2 + c_8 H^4.$$

that is asymptotically similar to (8) provided that we choose again $h \sim H^2$.

Important remark

- In the case of $\mathbb{P}_1-$ finite element we thus have

$$\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq \varepsilon(N) + c_5 h + c_6 H^2$$

- While in the case of $\mathbb{P}_2-$ finite element we get

$$\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq \varepsilon(N) + c_7 h^2 + c_8 H^4.$$

The constants c_6 et c_8 depends on N

Important remark

- In the case of $\mathbb{P}_1-$ finite element we thus have

$$\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq \varepsilon(N) + c_5 h + c_6 H^2$$

- While in the case of $\mathbb{P}_2-$ finite element we get

$$\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq \varepsilon(N) + c_7 h^2 + c_8 H^4.$$

The constants c_6 et c_8 depends on N

Important remark

- In the case of $\mathbb{P}_1-$ finite element we thus have

$$\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq \varepsilon(N) + c_5 h + c_6 H^2$$

- While in the case of $\mathbb{P}_2-$ finite element we get

$$\|u(\mu) - u_N^{Hh}(\mu)\|_X \leq \varepsilon(N) + c_7 h^2 + c_8 H^4.$$

The constants c_6 et c_8 depends on N

- Let us consider the following problem : find $u \in H^1(\Omega)$ such that

$$\begin{cases} -\Delta u + u^3 = sin(x)sin(y) & \text{dans } \Omega = [0,1]^2 \setminus (]\frac{1}{2},1[^2) \\ \beta u + \frac{\partial u}{\partial n} = y(1-y) & \text{sur } \Gamma_F = \{(1,y), y \in [0,\frac{1}{2}]\} \\ u = y^2 & \text{sur } \Gamma_2 = \{(x,1), x \in [0,1]\} \\ u = \eta\ xy(1-y)(1-x) & \text{sur } \Gamma_D = \partial\Omega \setminus \Gamma_F \end{cases}$$

  In this example, the set of parameters is $\mu = (\beta, \eta) \in \mathcal{D} = [1,37] \times [1,100]$.

- Let

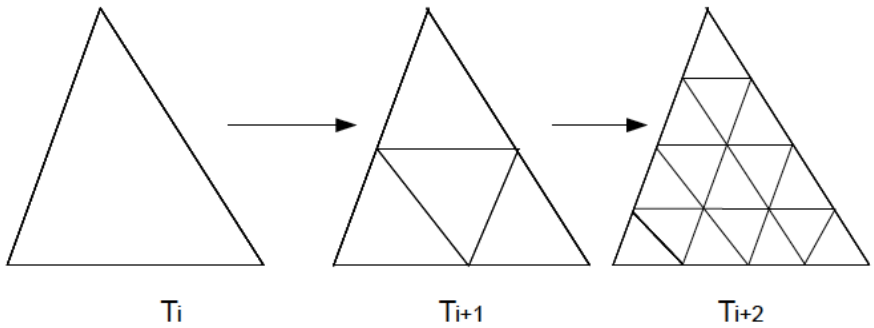
$$\mu_{H_i} = \text{argmax}\{||u(\mu) - u_N^{hH_i}(\mu)||_{1,\Omega}, \mu \in \mathcal{D}\}$$

  and

$$\mu_h = \text{argmax}\{||u(\mu) - u^h(\mu)||_{1,\Omega}, \mu \in \mathcal{D}\}$$

Two grids method

The problem we consider is in 2 dimensions.



In the case where $X_h = \{v \in \mathcal{C}^0(\overline{\Omega}),\ v_{|T} \in \mathbb{P}_1(T),\ T \in \mathcal{T}_4\}$

- $||u(\mu_h) - u_h(\mu_h)||_{1,\Omega} = 3.3 \times 10^{-2}$

TABLE:

Error for the problem in a L-shape domain when $X_h = \{v \in \mathcal{C}^0(\overline{\Omega}),\ v_{|T} \in \mathbb{P}_1(T),\ T \in \mathcal{T}_4\}$

		$\|\|u(\mu_{MH_i}) - u_N^{hH_i}(\mu_{MH_i})\|\|_{1,\Omega}$	$\|\|u(\mu_{Mh}) - u_N^{hR}(\mu_{Mh})\|\|_{1,\Omega}$	$\|\|u(\mu_{MH_i}) - u_{H_i}(\mu_{MH_i})\|\|_{1,\Omega}$
$N = 5$	$\mathcal{T}_{H_0}$	0.132447	0.191438	0.48848
	$\mathcal{T}_{H_1}$	0.159983	0.191438	0.282155
	$\mathcal{T}_{H_2}$	0.182003	0.191438	0.148391
	$\mathcal{T}_{H_3}$	0.189419	0.191438	0.0727888
$N = 10$	$\mathcal{T}_{H_0}$	0.353764	0.0361391	0.48848
	$\mathcal{T}_{H_1}$	0.0682768	0.0361391	0.282155
	$\mathcal{T}_{H_2}$	0.0381123	0.0361391	0.148391
	$\mathcal{T}_{H_3}$	0.035279	0.0361391	0.0727888
$N = 15$	$\mathcal{T}_{H_0}$	0.473683	0.0338436	0.48848
	$\mathcal{T}_{H_1}$	0.141902	0.0338436	0.282155
	$\mathcal{T}_{H_2}$	0.0389836	0.0338436	0.148391
	$\mathcal{T}_{H_3}$	0.033873	0.0338436	0.0727888
$N = 20$	$\mathcal{T}_{H_0}$	0.561913	0.0334947	0.48848
	$\mathcal{T}_{H_1}$	0.197724	0.0334947	0.282155
	$\mathcal{T}_{H_2}$	0.0483812	0.0334947	0.148391
	$\mathcal{T}_{H_3}$	0.0338948	0.0334947	0.0727888

In the case where $X_h = \{v \in \mathcal{C}^0(\overline{\Omega}),\ v_{|T} \in \mathbb{P}_1(T),\ T \in \mathcal{T}_4\}$

- $||u(\mu_h) - u_h(\mu_h)||_{1,\Omega} = 3.3 \times 10^{-2}$

TABLE:

Error for the problem in a L-shape domain when $X_h = \{v \in \mathcal{C}^0(\overline{\Omega}),\ v_{|T} \in \mathbb{P}_1(T),\ T \in \mathcal{T}_4\}$

		$\|\|u(\mu_{MH_i}) - u_N^{hH_i}(\mu_{MH_i})\|\|_{1,\Omega}$	$\|\|u(\mu_{Mh}) - u_N^{hR}(\mu_{Mh})\|\|_{1,\Omega}$	$\|\|u(\mu_{MH_i}) - u_{H_i}(\mu_{MH_i})\|\|_{1,\Omega}$
$N = 5$	$\mathcal{T}_{H_0}$	0.132447	0.191438	0.48848
	$\mathcal{T}_{H_1}$	0.159983	0.191438	0.282155
	$\mathcal{T}_{H_2}$	0.182003	0.191438	0.148391
	$\mathcal{T}_{H_3}$	0.189419	0.191438	0.0727888
$N = 10$	$\mathcal{T}_{H_0}$	0.353764	0.0361391	0.48848
	$\mathcal{T}_{H_1}$	0.0682768	0.0361391	0.282155
	$\mathcal{T}_{H_2}$	0.0381123	0.0361391	0.148391
	$\mathcal{T}_{H_3}$	0.035279	0.0361391	0.0727888
$N = 15$	$\mathcal{T}_{H_0}$	0.473683	0.0338436	0.48848
	$\mathcal{T}_{H_1}$	0.141902	0.0338436	0.282155
	$\mathcal{T}_{H_2}$	0.0389836	0.0338436	0.148391
	$\mathcal{T}_{H_3}$	0.033873	0.0338436	0.0727888
$N = 20$	$\mathcal{T}_{H_0}$	0.561913	0.0334947	0.48848
	$\mathcal{T}_{H_1}$	0.197724	0.0334947	0.282155
	$\mathcal{T}_{H_2}$	0.0483812	0.0334947	0.148391
	$\mathcal{T}_{H_3}$	0.0338948	0.0334947	0.0727888

Two grids method

We are in interested in solving the following convection dominated problem : find $u \in H^1(\Omega)$ such that

$$\begin{array}{ll} -(0.01)\Delta u + v \cdot \nabla u = 0 & \text{in } \Omega = [0,1]^2 \\ u = x^2 & \text{on } \Gamma_1 = \{(1,y), y \in [0,1]\} \\ u = y^2 & \text{on } \Gamma_2 = \{(x,1), x \in [0,1]\} \\ u = 0 & \text{on } \Gamma_3 = \partial\Omega \setminus (\Gamma_1 \cup \Gamma_2). \end{array}$$

where v is such as $v = (\cos\ \mu,\ \sin\ \mu)$.

$\longrightarrow$ Here, the varying parameter is the angle of the convection flux $\mu \in [0, \frac{\pi}{2}]$.

Two grids method

Table: Error with $X_h = \{v \in \mathcal{C}^0(\overline{\Omega}),\ v_{|T} \in \mathbb{P}_2(T),\ T \in \mathcal{T}_{H_2}\}$

$$\|u_{ref}(\mu_p) - u_h(\mu_p)\|_{1,\Omega} = 4.09 \times 10^{-3}$$

N	i	$\|\|u_{ref}(\mu_p) - \tilde{u}^N_{h,H_i}(\mu_p)\|\|_{1,\Omega}$ with post-processing	$\|\|u_{ref}(\mu_p) - u^N_{h,H_i}(\mu_p)\|\|_{1,\Omega}$ without post-processing	$\|\|u_{ref}(\mu_p) - u_{H_i}(\mu_p)\|\|_{1,\Omega}$
$\|\|u_{ref}(\mu_p) - u^N_{h,h}(\mu_p)\|\|_{1,\Omega} = 1.65 \times 10^{-2}$				
5	0	1.64×10^{-2}	1.59×10^{-2}	5.86×10^{-2}
	1	1.65×10^{-2}	1.65×10^{-2}	1.62×10^{-2}
$\|\|u_{ref}(\mu_p) - u^N_{h,h}(\mu_p)\|\|_{1,\Omega} = 4.10 \times 10^{-3}$				
5	0	4.10×10^{-3}	1.23×10^{-2}	5.86×10^{-2}
	1	4.10×10^{-3}	4.19×10^{-3}	1.62×10^{-2}
$\|\|u_{ref}(\mu_p) - u^N_{h,h}(\mu_p)\|\|_{1,\Omega} = 4.10 \times 10^{-3}$				
5	0	4.10×10^{-3}	1.28×10^{-2}	5.86×10^{-2}
	1	4.10×10^{-3}	4.21×10^{-3}	1.62×10^{-2}

Two grids method

Table: Error with $X_h = \{v \in \mathcal{C}^0(\overline{\Omega}),\ v_{|T} \in \mathbb{P}_3(T),\ T \in \mathcal{T}_{H_2}\}$

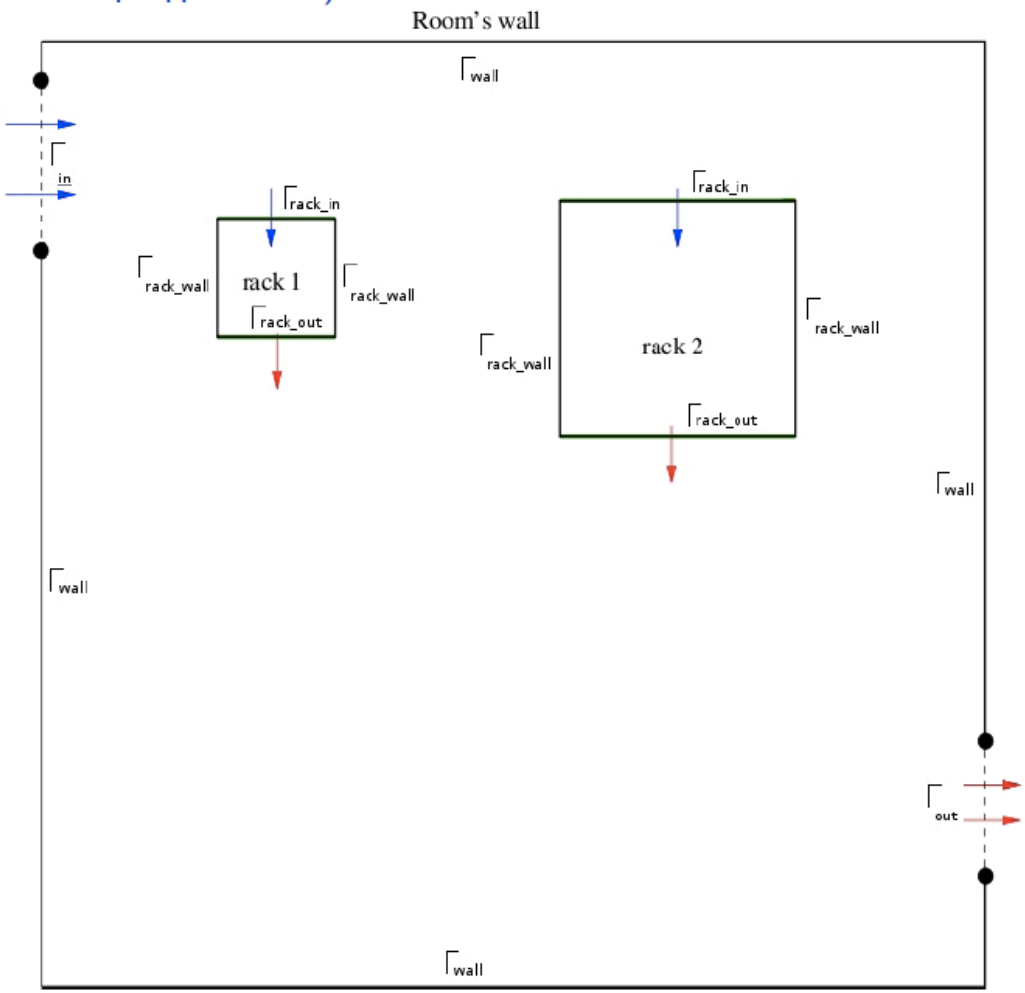
$$\|u_{ref}(\mu_p) - u_h(\mu_p)\|_{1,\Omega} = 1.31 \times 10^{-4}$$

N	i	$\|\|u_{ref}(\mu_p) - \tilde{u}^N_{h,H_i}(\mu_p)\|\|_{1,\Omega}$ with post-processing	$\|\|u_{ref}(\mu_p) - u^N_{h,H_i}(\mu_p)\|\|_{1,\Omega}$ without post-processing	$\|\|u_{ref}(\mu_p) - u_{H_i}(\mu_p)\|\|_{1,\Omega}$
$\|\|u_{ref}(\mu_p) - u^N_{h,h}(\mu_p)\|\|_{1,\Omega} = 1.60 \times 10^{-2}$				
5	0	1.60×10^{-2}	1.60×10^{-2}	7.13×10^{-3}
	1	1.60×10^{-2}	1.60×10^{-2}	1.01×10^{-3}
$\|\|u_{ref}(\mu_p) - u^N_{h,h}(\mu_p)\|\|_{1,\Omega} = 1.31 \times 10^{-4}$				
5	0	1.31×10^{-4}	2.77×10^{-4}	7.13×10^{-3}
	1	1.31×10^{-4}	1.32×10^{-4}	1.01×10^{-3}
$\|\|u_{ref}(\mu_p) - u^N_{h,h}(\mu_p)\|\|_{1,\Omega} = 1.31 \times 10^{-4}$				
5	0	1.31×10^{-4}	4.99×10^{-4}	7.13×10^{-3}
	1	1.31×10^{-4}	1.32×10^{-4}	1.01×10^{-3}

A more complex problem...

Model : Incompressible steady Navier Stokes stationnaire + Heat equation

(Boussinesq's approximation)



$V_{in} \in [0.5; 2]$,
$\theta_{in} \in [288; 292]$,
$V_{rack} \in [0.1; 0.4]$,
$\theta_{rack} \in [295; 315]$.

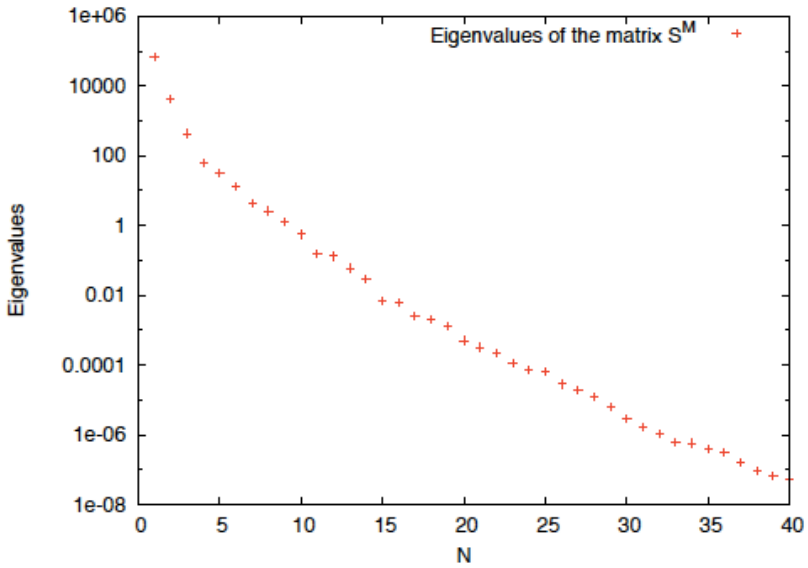
A more complex problem...

Sampling to extract the reduced basis

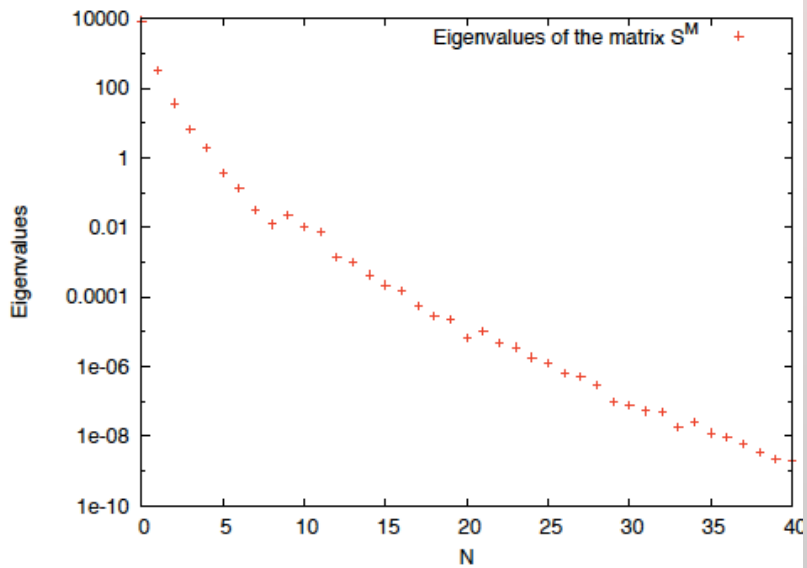
θ_{in}	V_{in}	V_{rack}	θ_{rack}
288	0.5	0.1	295
292	1	0.2	300
	2	0.3	305
		0.4	310
			315

Computations of 120 snapshots using a $\mathbb{P}_2 - \mathbb{P}_1$ F.E steady Navier-Stokes solver within Freefem++ on a reference mesh.

Figure: Values of the N largest eigenvalues of the matrix $S^{\mathcal{M}}$

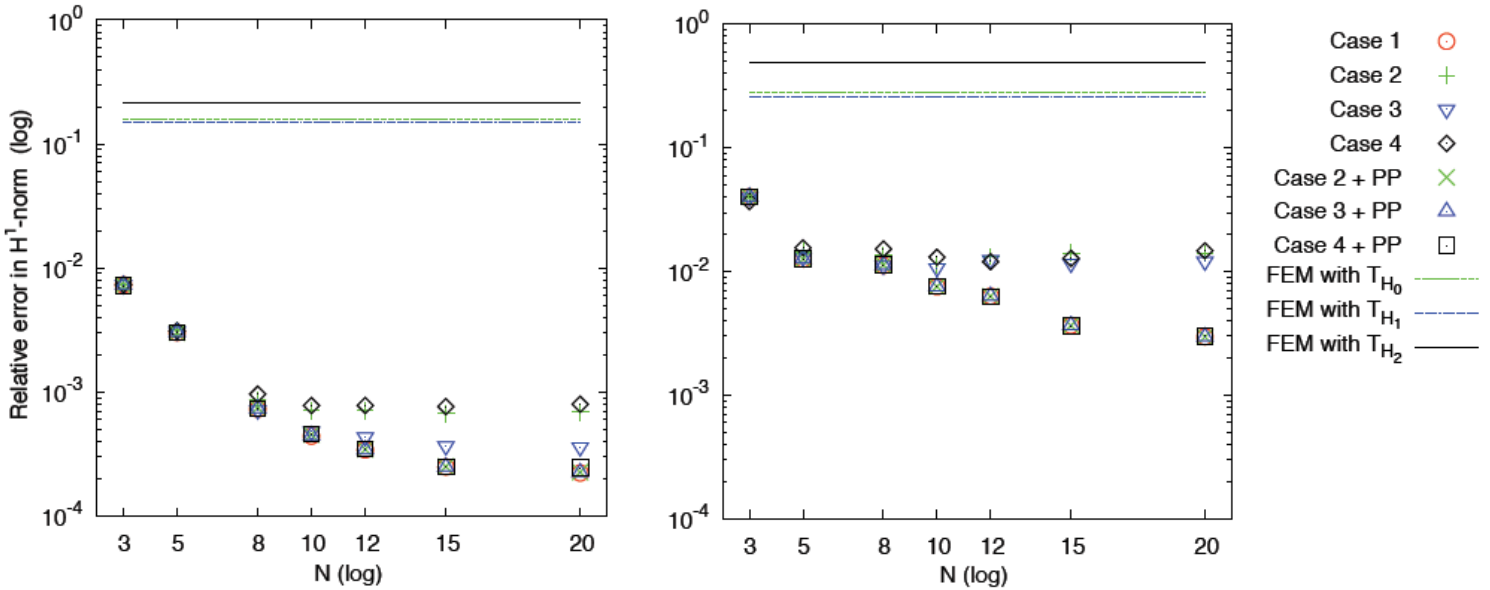


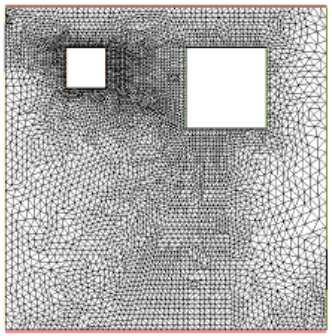
Temperature



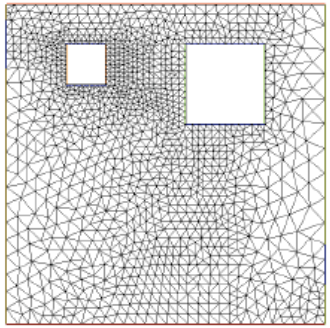
Velocity

A more complex problem...

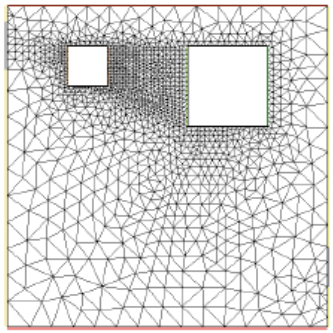




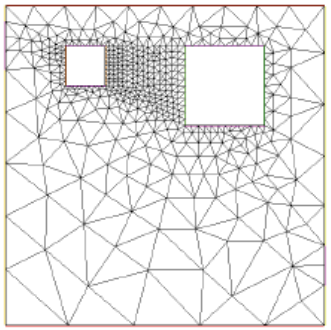
reference mesh
Ndof = 20213
Case 1



coarse embedded mesh
Ndof = 5148
Case 2



coarse non embedded mesh
Ndof = 5088
Case 3

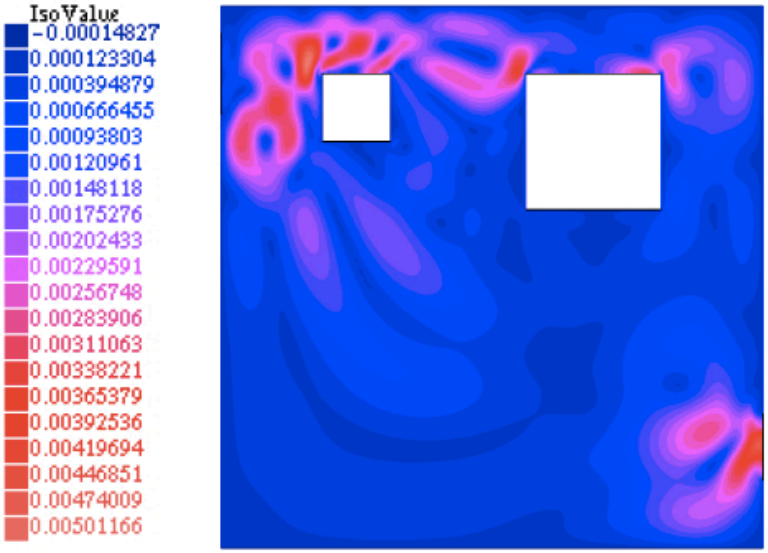


very coarse non embedded mesh
Ndof = 1871
Case 4

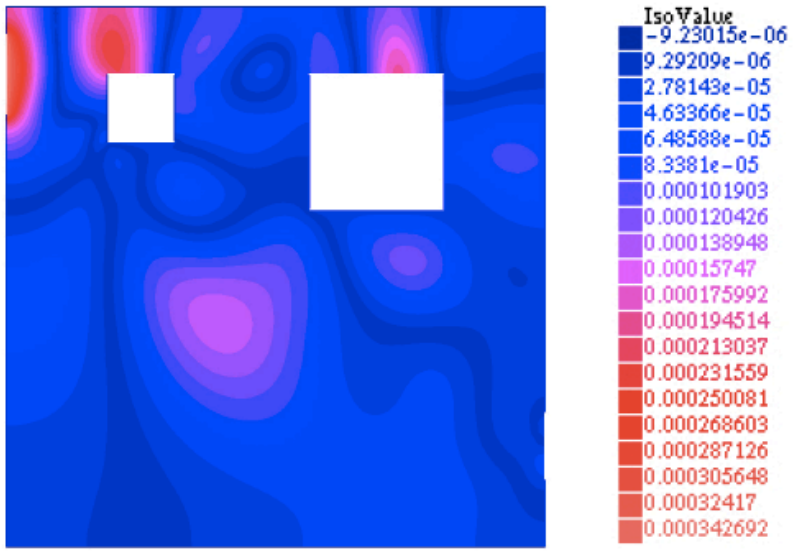
A more complex problem...

Relative error plot between the reference F.E. and the NIRB solutions

(case 4 + p.p. with $N = 15$)



Velocity magnetude



Tempertaure

Mean value of the online's stage with post-processing executions's time - $N = 15$

Reference FEM	NIRB - case 2	NIRB - case 3	NIRB - case 4	
200 sec	52 sec	52 sec	17 sec	Temperature
	53 sec	53 sec	18 sec	Velocity
	54 sec	54 sec	19 sec	Both

Anoter complex problem...

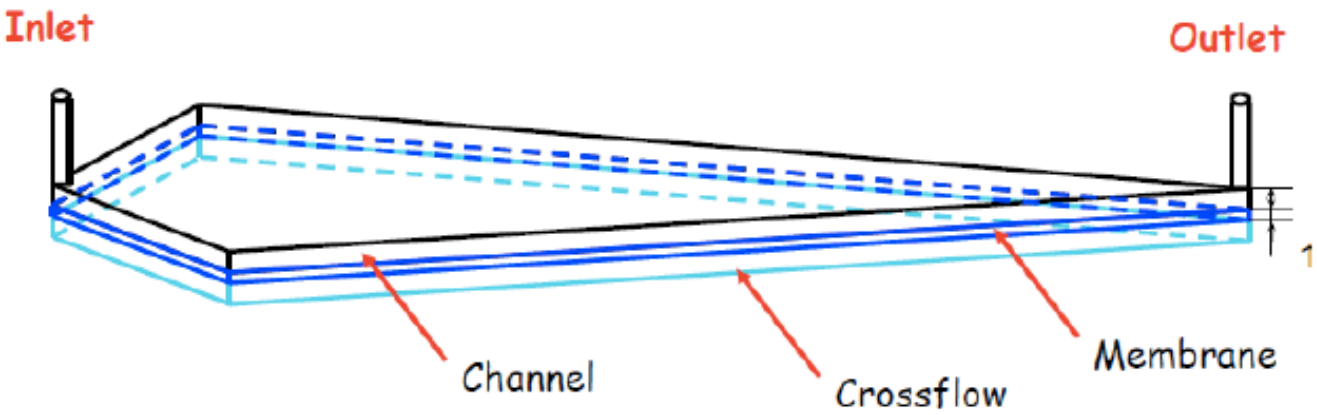
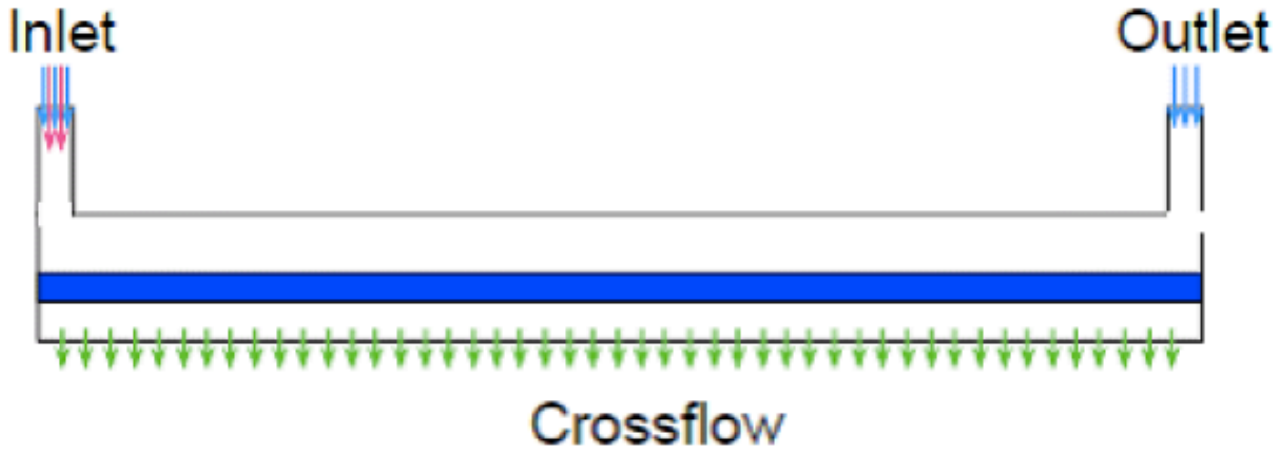


Figure 1: Sketch of 3D channel of fractionation device.



Another complex problem...

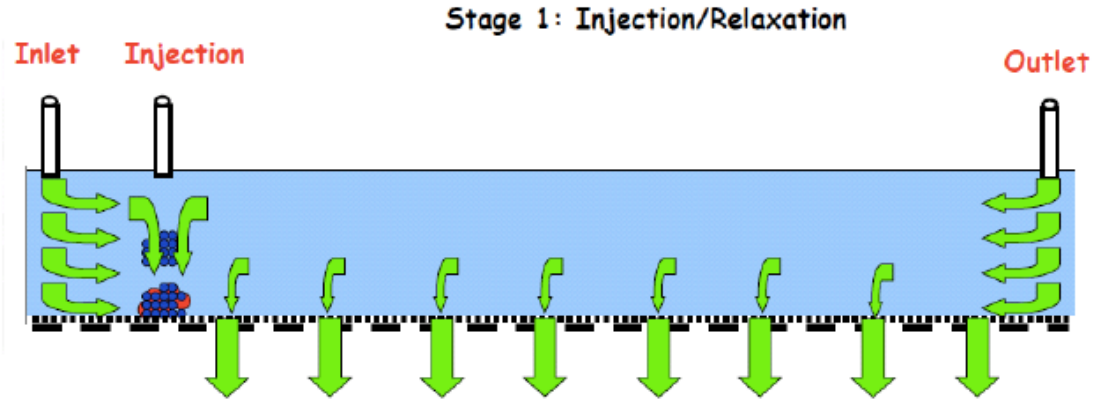


Figure 3: Sketch of the focusing–injection stage of AFFFF. Courtesy of Wyatt Europe GmbH.

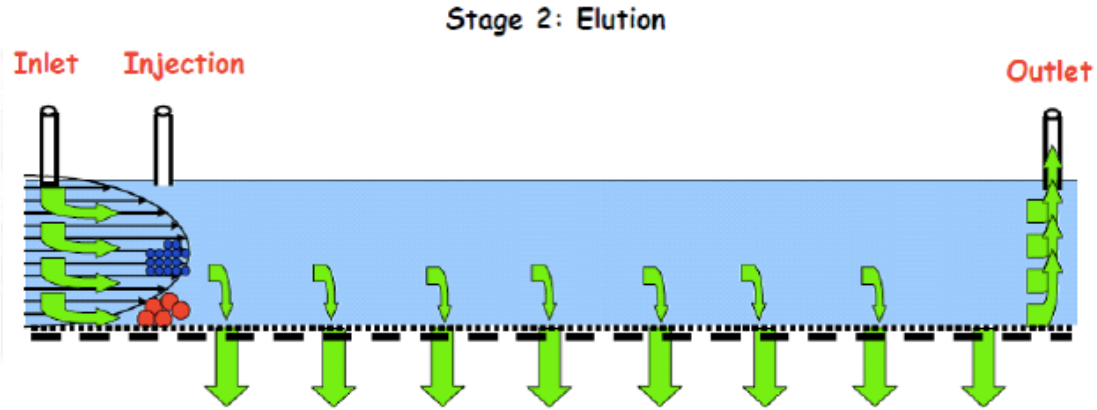


Figure 4: Sketch of the Elution Stage of AFFFF. Courtesy of Wyatt Europe GmbH.

Another complex problem...