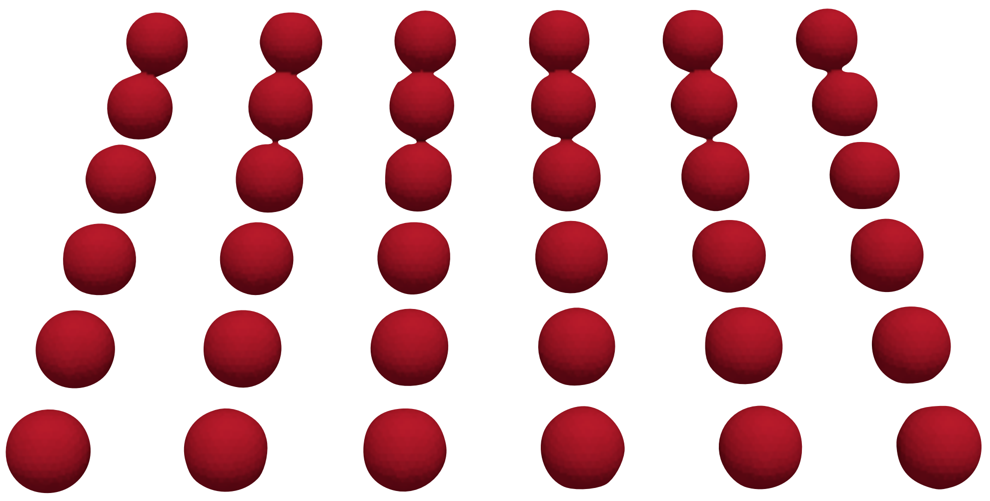
Battling bottlenecks:

Overcoming the computational complexity of reduced basis methods for high-d parameter spaces

Jan S Hesthaven
EPFL-SB-MATHICSE-MCSS
jan.Hesthaven@epfl.ch



w/ B. Stamm (Paris VI), S. Zhang (CUHK)

Funded by AFOSR/OSD

Problems of interest

We consider the need to model physical systems of the form

$$\mathcal{L}(\mathbf{x},\mu)u(\mathbf{x},\mu) = f(\mathbf{x},\mu) \qquad \mathbf{x} \in \Omega$$

$$u(\mathbf{x},\mu) = g(\mathbf{x},\mu) \qquad \mathbf{x} \in \partial\Omega$$

where the solutions are implicitly parameterized by

$$\mu \in \mathcal{D} \subset \mathrm{R}^M$$

We focus on problems with special characteristics

- ✓ Real-time or near real time need
- ✓ Many query problems
- ✓ In situ needs/deployed system

Model reduction

What we need is an accurate way to evaluate the solution at new parameter values at reduced complexity.

input: parameter value $\boldsymbol{\mu} \in \mathcal{D}$

PDE solver

$$\mathcal{L}_h(u_h(\boldsymbol{\mu}); \boldsymbol{\mu}) = 0$$

output: $s_h(\boldsymbol{\mu}) = l(u_h(\boldsymbol{\mu}); \boldsymbol{\mu})$

RBM 101

Let us define:

The exact solution: Find $u(\mu) \in X$ such that

$$a(u, \mu, v) = f(\mu, v), \quad \forall v \in X$$

The truth solution: Find $u_h(\mu) \in X_h$ such that

$$a_h(u_h, \mu, v_h) = f_h(\mu, v_h), \quad \forall v_h \in X_h \qquad \dim(X_h) = \mathcal{N}$$

The RB solution: Find $u_{RB}(\mu) \in X_N$ such that

$$a_h(u_{RB}, \mu, v_N) = f_h(\mu, v_N), \forall v_N \in X_N \qquad \dim(X_N) = N$$

We always assume that $\mathcal{N} \gg N$

RBM 101

Solving for the truth is expensive - but we need to be able to trust the RB solution

$$\|u(\mu) - u_{RB}(\mu)\| \leq \|u(\mu) - u_h(\mu)\| + \|u_h(\mu) - u_{RB}(\mu)\|$$

We assume that

$$\|u(\mu) - u_h(\mu)\| \leq \varepsilon$$

This is <u>your favorite solver</u> and it is assumed it can be as accurate as you desire - the truth

So if we can bound we achieve two things

✓ Certify the accuracy of the reduced basis method
✓ Use this estimate to build the basis

RBM 101

Let us define the residual in the dual norm

$$R_N(\mu, v) := f(v, \mu) - a(u_{RB}, \mu, v), \quad \forall v \in X$$

$$\varepsilon_N(\mu) := \sup_{v \in X} \frac{|R_N(\mu, v)|}{\|v\|_X}$$

and require stability as

$$\beta(\mu) := \inf_{v \in X} \sup_{w \in X} \frac{|a(v, \mu, w)|}{\|v\|_X \|w\|_X}$$

$$0 < \beta_{LB}(\mu) \leq \beta(\mu), \quad \forall \mu \in \mathcal{D}$$

then the error is obtained as

$$\boxed{\Delta_N(\mu) := \frac{\varepsilon_N(\mu)}{\beta_{LB}(\mu)}}$$

RBM 101

Defining the effectivity

$$\eta_N(\mu) := \frac{\Delta_N(\mu)}{\|u_h - u_{RB}\|_X}$$

One proves

$$1 \leq \eta_N(\mu), \quad \forall \mu \in \mathcal{D}$$

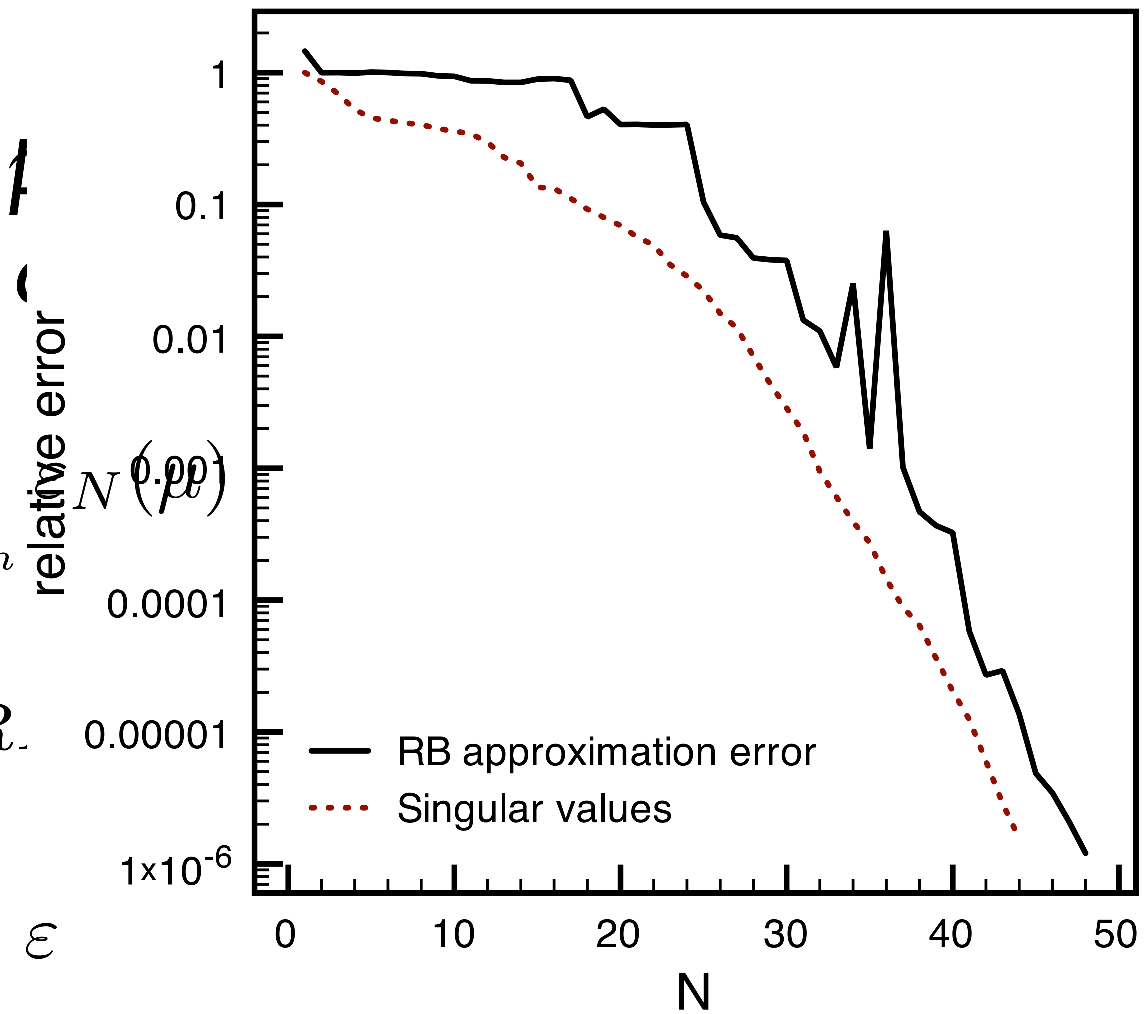
So if the basis is known, we can estimate the error when using the reduced model

... but we have still to construct the reduced basis

RBM 101

We use the error estimator to construct the reduced basis in a *greedy* approach.

1. *Define a (fine)training set in*
2. *Choose a member randomly*
3. *Define* $u_{RB} = u_h(\mu_1)$
   a. *Find* $\mu_{i+1} = \arg \sup_{\mu \in \Pi_{train}} \varepsilon_N(\mu)$
   b. *Compute* $u_h(\mu_{i+1})$
   c. *Orthonormalize wrt* u_{R}
   d. *Add new solution basis*
4. *Continue until* $\sup_{\mu \in \Pi_{train}} \varepsilon_N \leq \varepsilon$



Resulting in

$$u_{RM}(\mu) = \sum_{i=1}^{N} u_N^i(\mu)\xi_i$$

RBM 101

Speed relies on the affine assumption

$$a(u,\mu,v) = \sum_{k=1}^{Q_a} \Theta_k(\mu) a_k(u,v)$$

$$f(\mu,v) = \sum_{k=1}^{Q_f} \Theta_k^f(\mu) f_k(v)$$

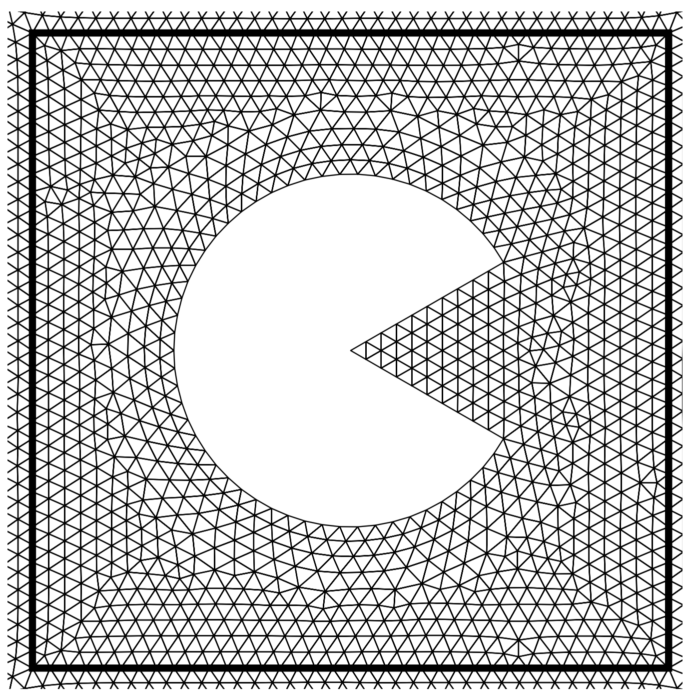
This pushes majority of work off-line, e.g.

$$\sum_{i=1}^{N} \left[\sum_{k=1}^{Q_a} \Theta_k(\mu) a_k(\xi_i,\xi_j) \right] u_N^j(\mu) = \sum_{k=1}^{Q_f} \Theta_k^f(\mu) f_k(\xi_j), \quad j \in [1,\ldots,N]$$

All operations are now independent of $\mathcal{N}$
and depends solely on N and Q

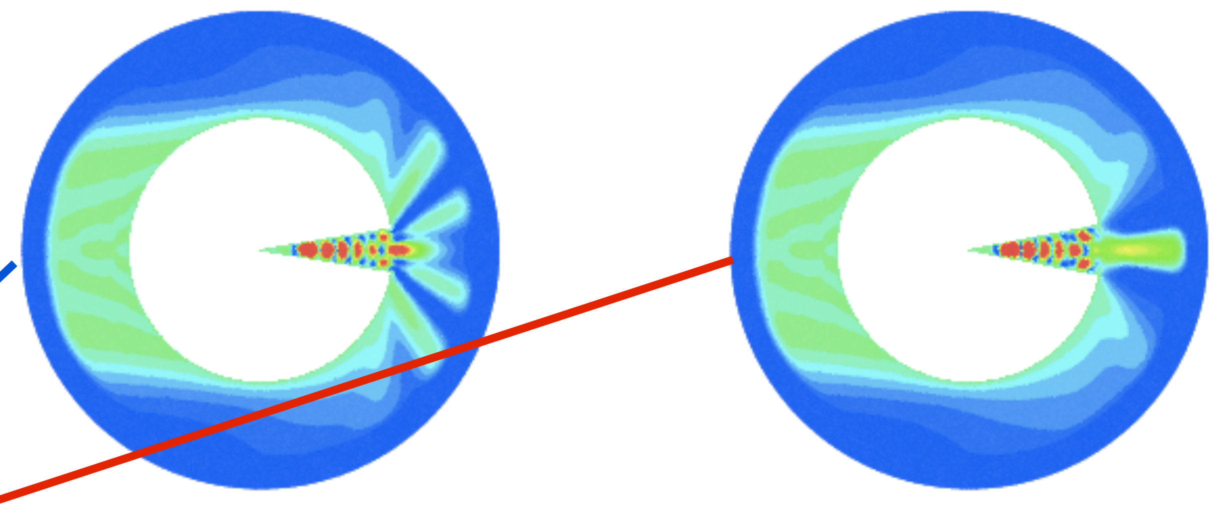
Also possible for error estimator

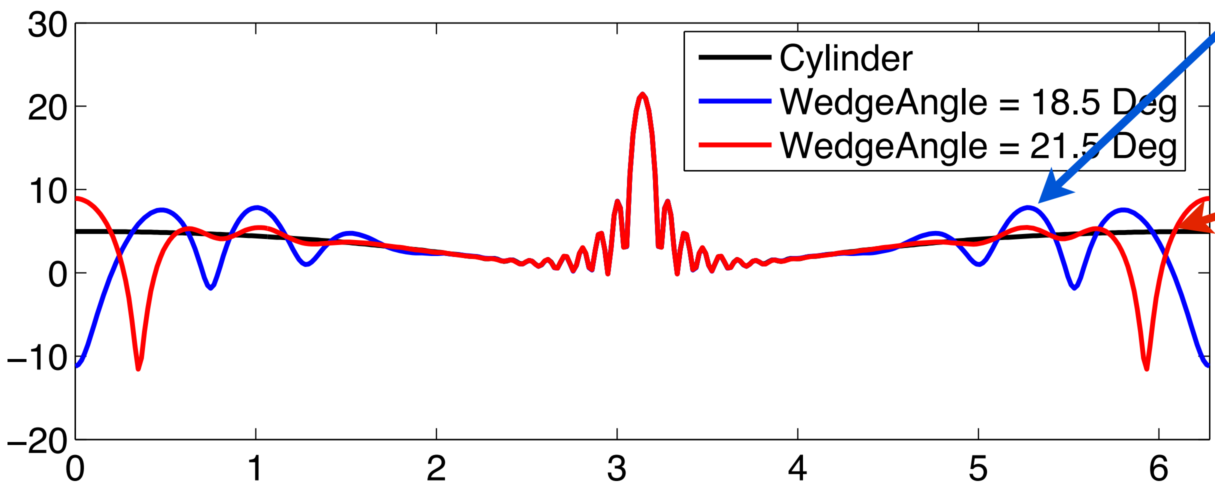
One example - 2D Pacman problem



Scattering by 2D PEC Pacman

Backscatter depends very sensitively on cutout angle and frequency.



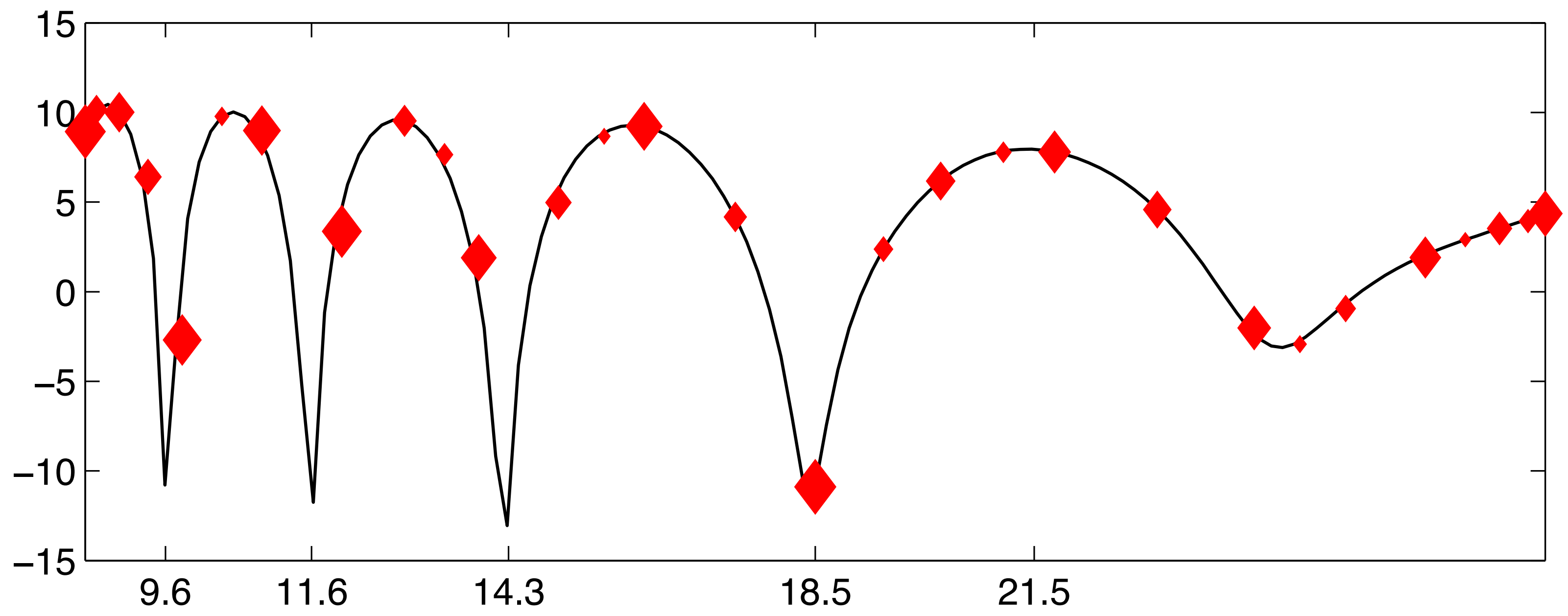


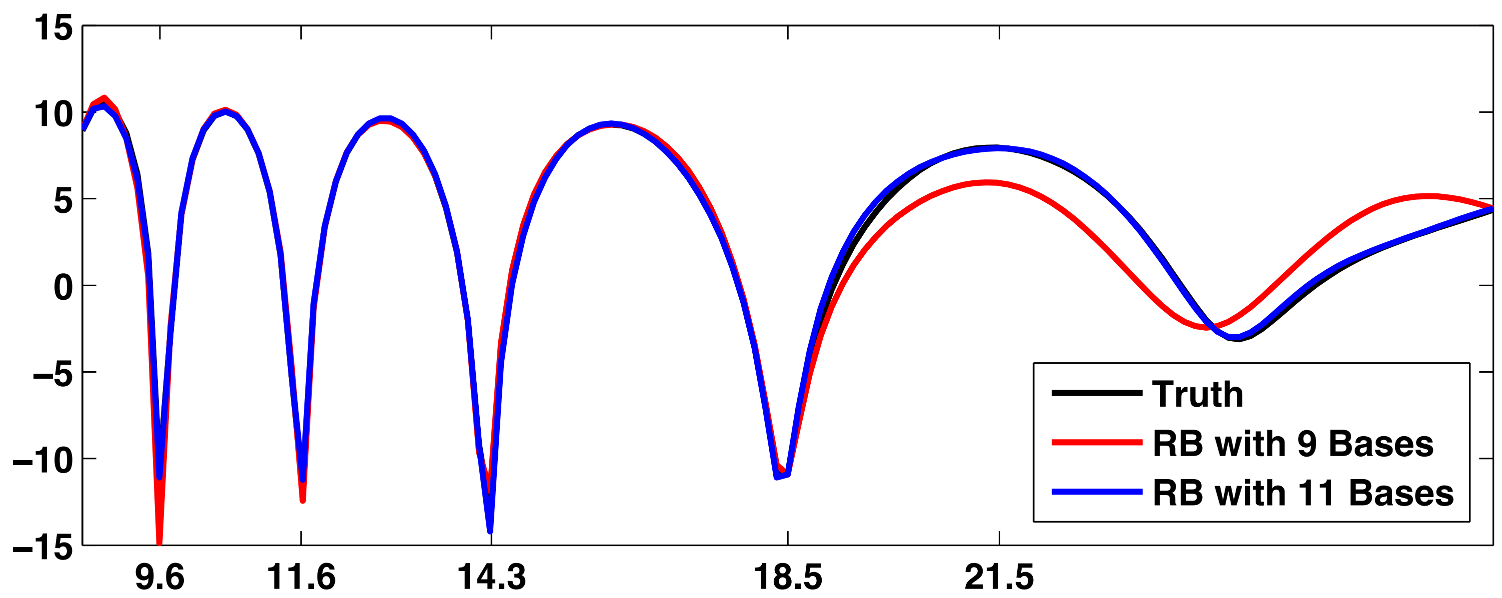
TM polarization

Difference in scattering is clear in fields

2D Pacman problem

Greedy approach selects critical angles early in the selection process

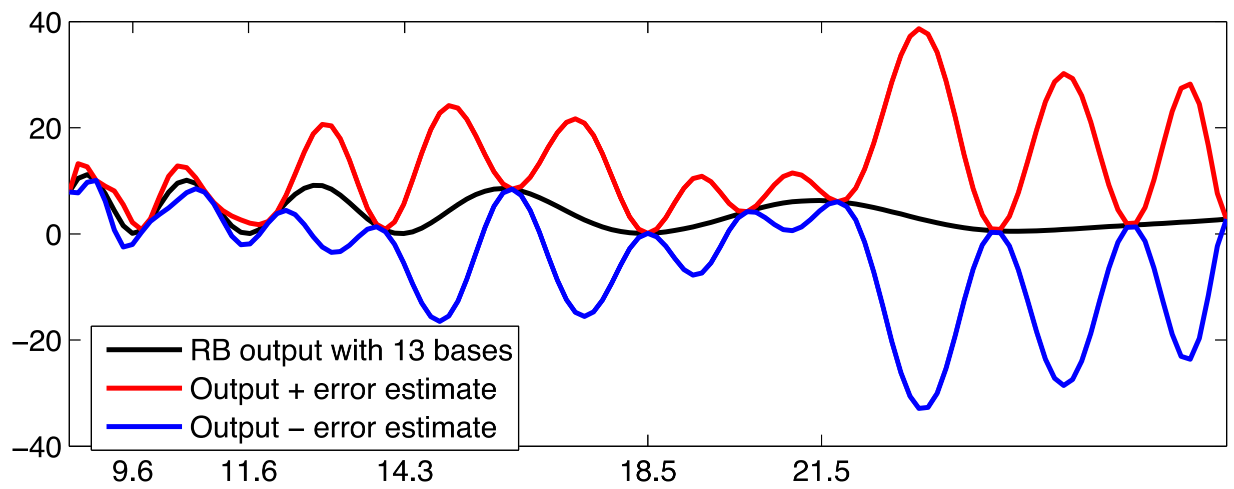




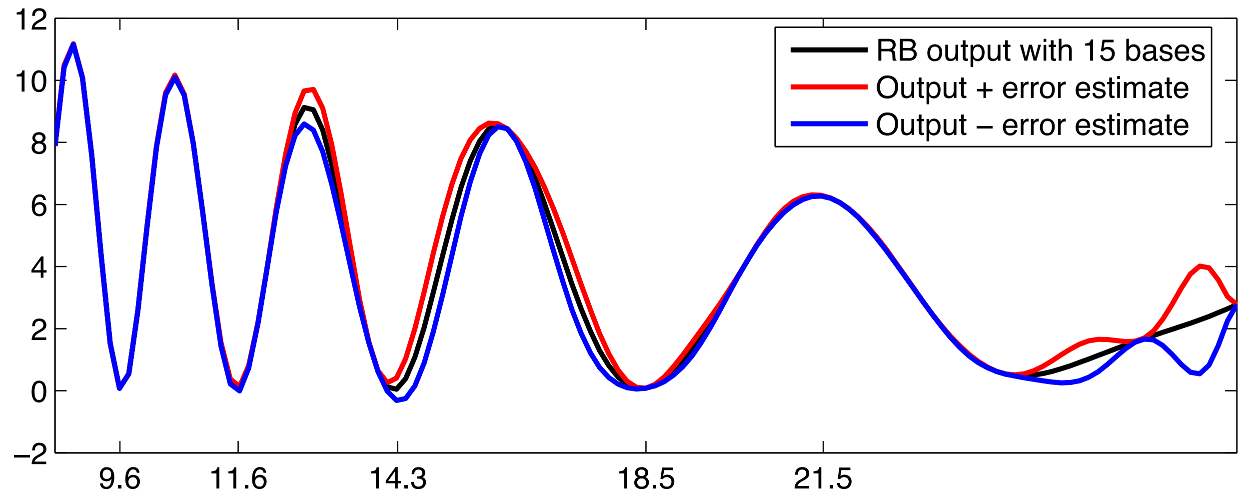
Convergence of output with O(10) basis elements

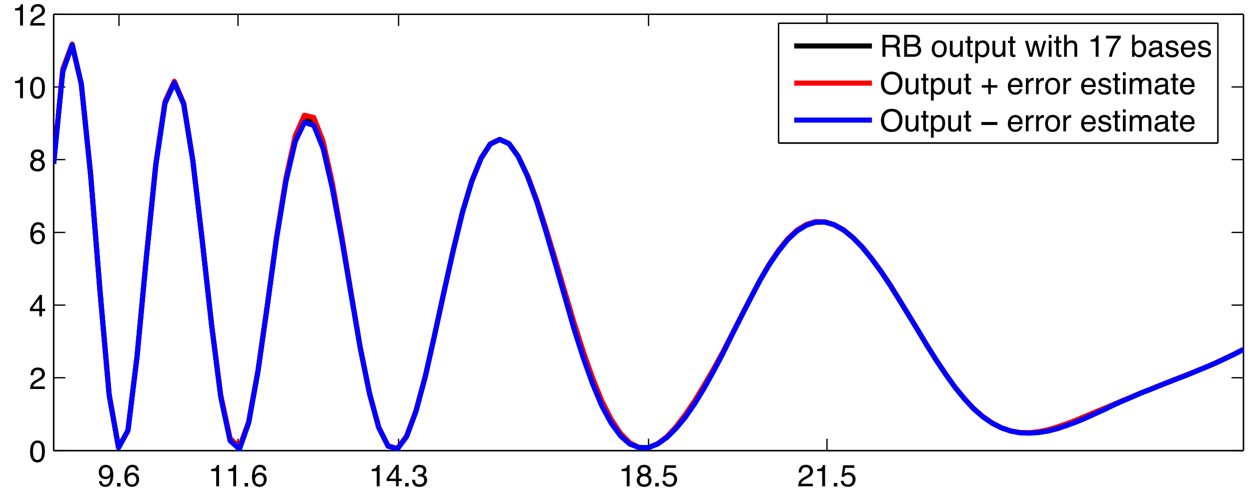
Output of interest - backscatter

2D Pacman problem

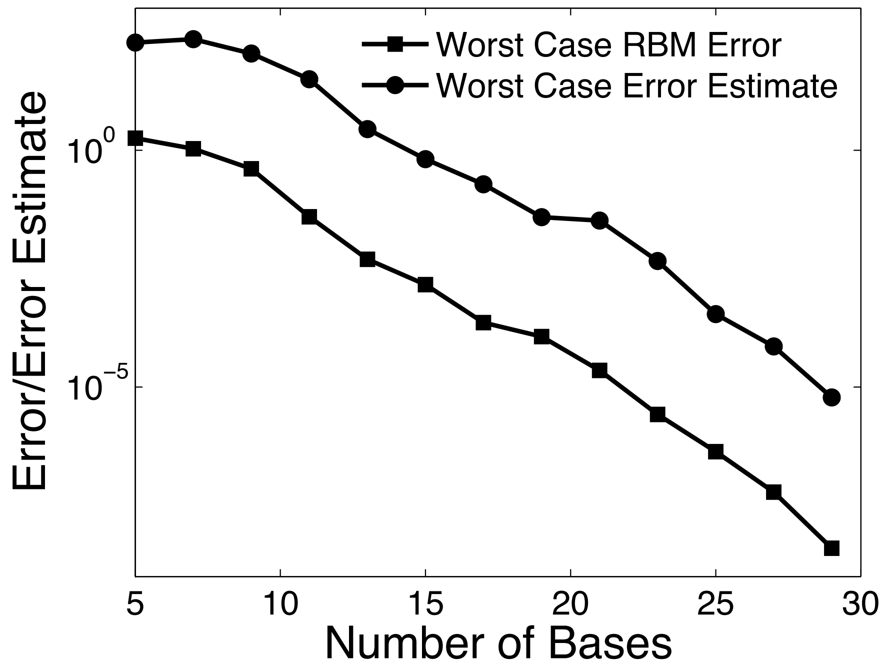


Convergence of error bounds over full parameter range





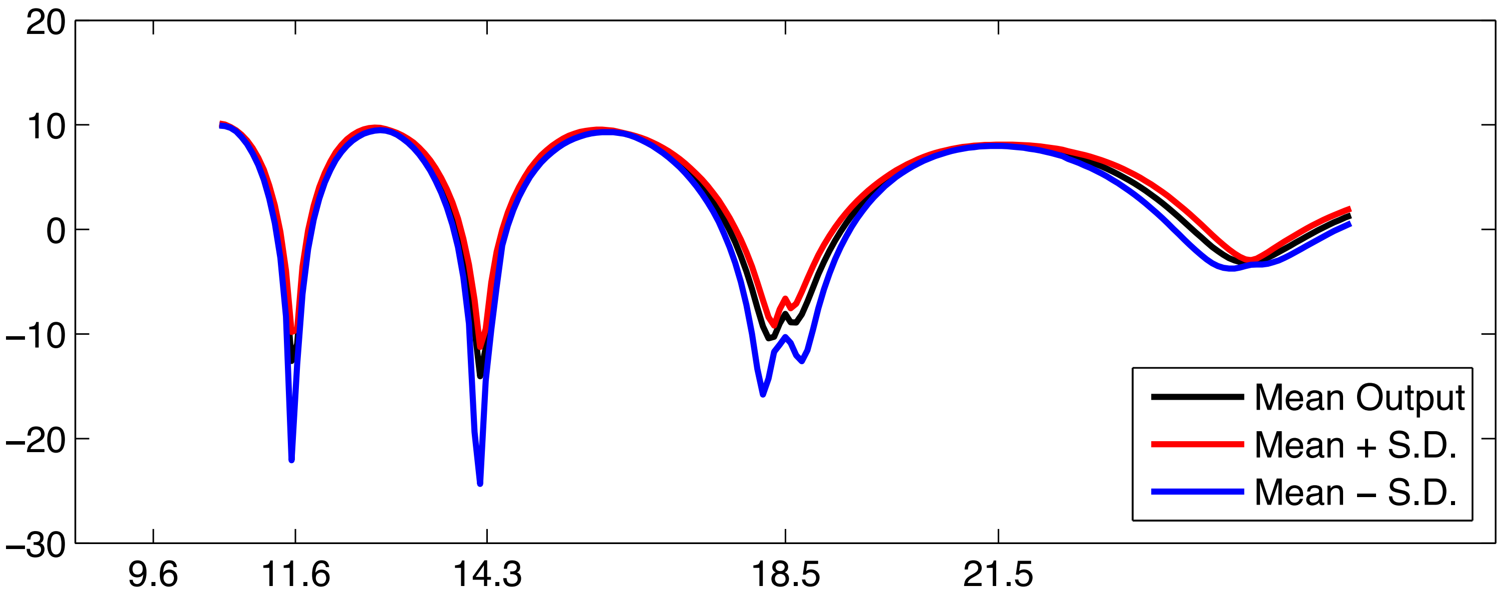
Exponential convergence of predicted error estimator and real error over large training set

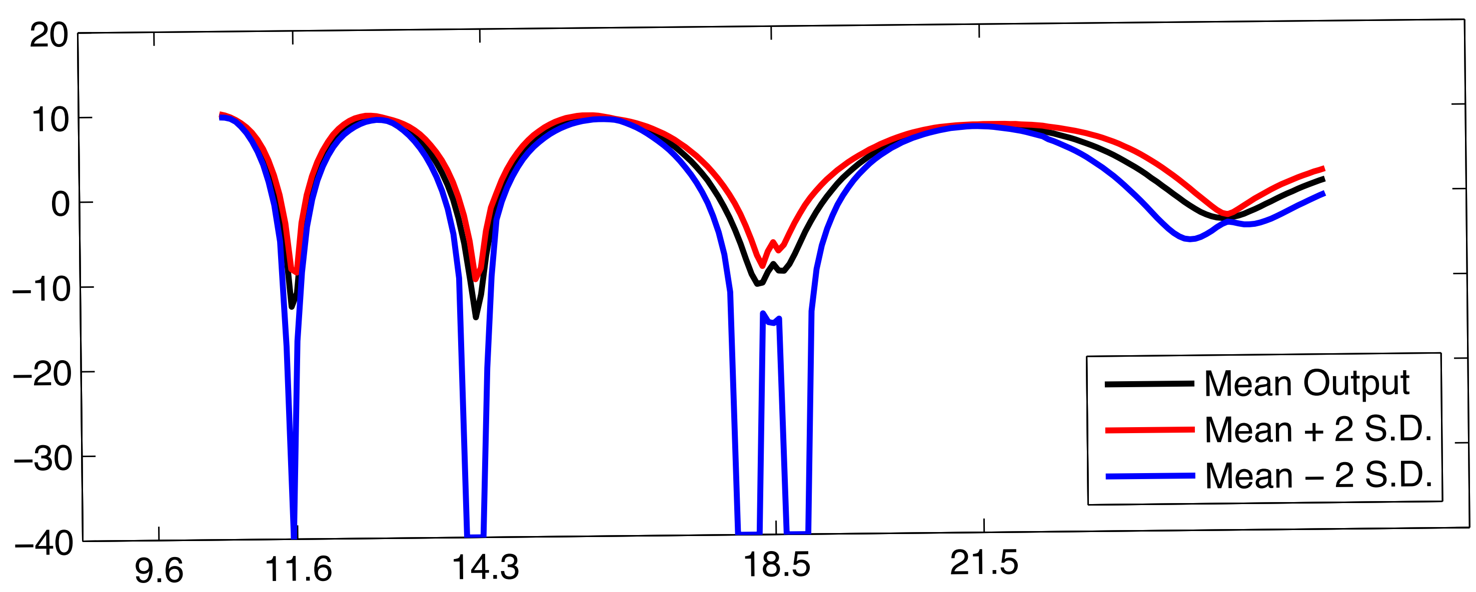


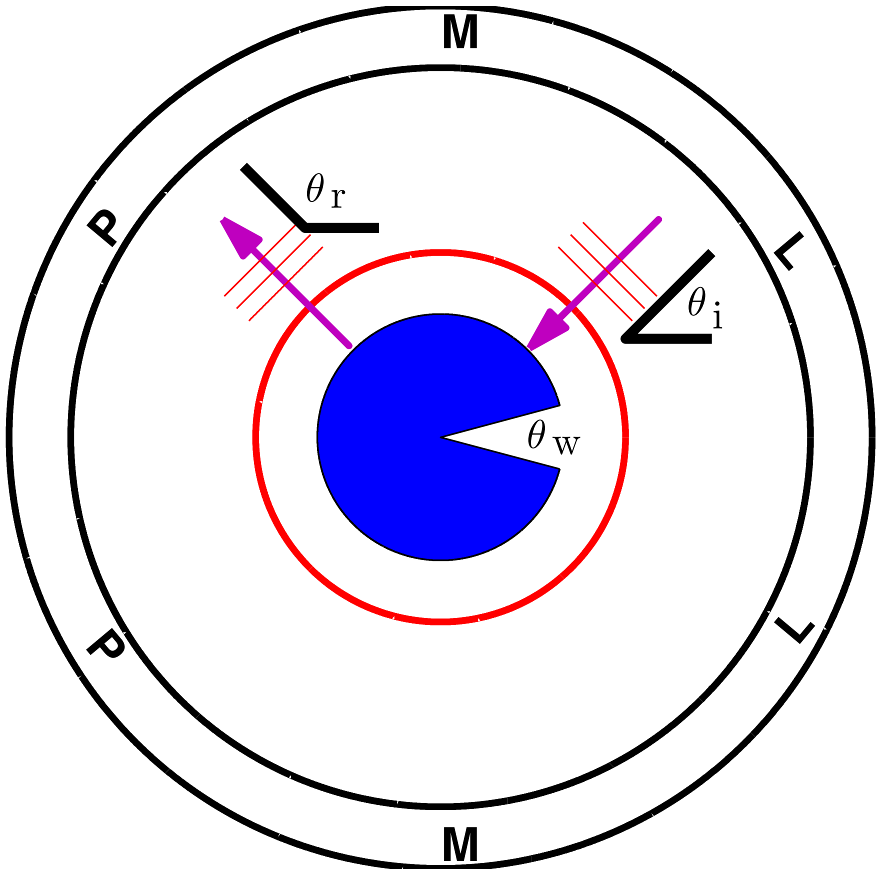
Note: Linear scale, not db scale

2D Pacman prototype for UQ

Fast evaluation over parameter space allows for rapid uncertainty quantification



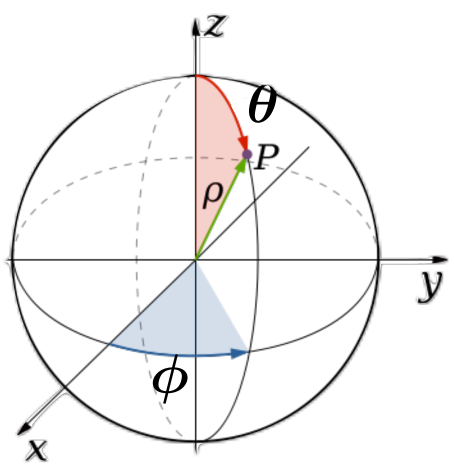


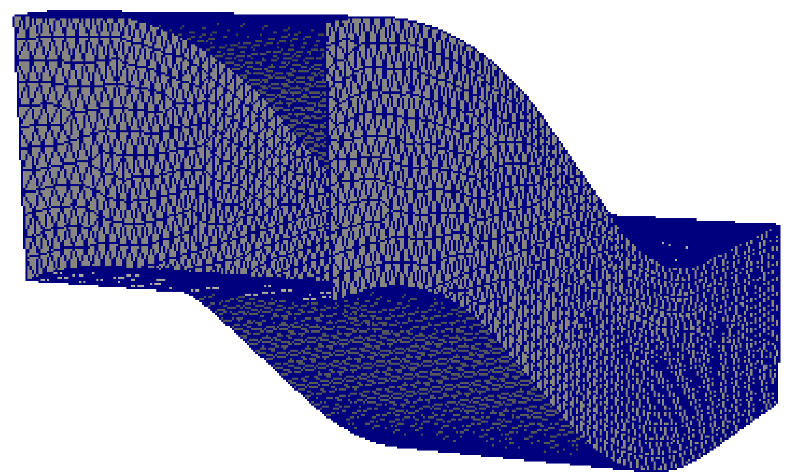


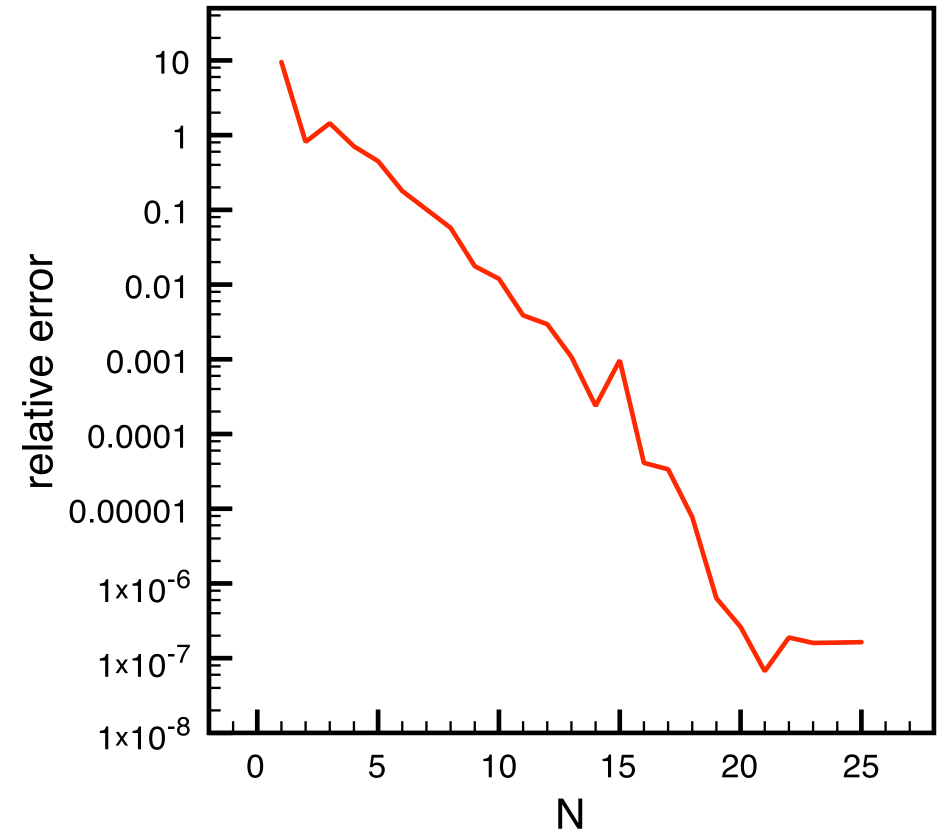
Uniformly distributed 5% randomness in gap angle

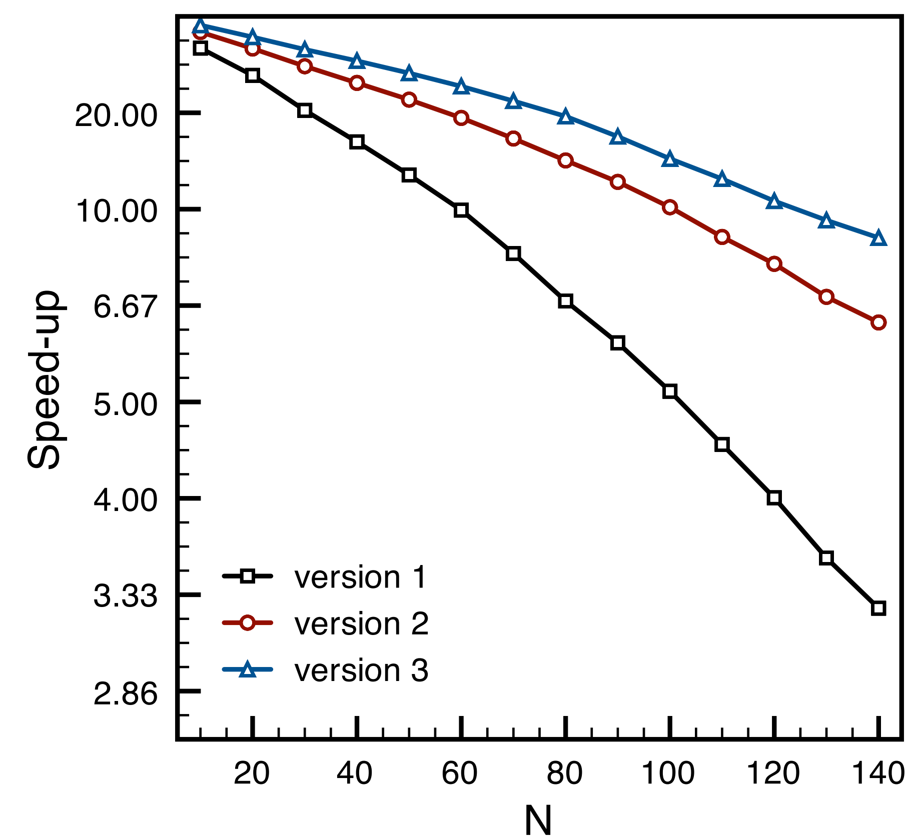
Scattering example

1 parameter, $\mu = k$ with $\mathcal{D} = [1, 25.5]$
$(\theta, \phi) = (\frac{\pi}{6}, 0)$ fixed

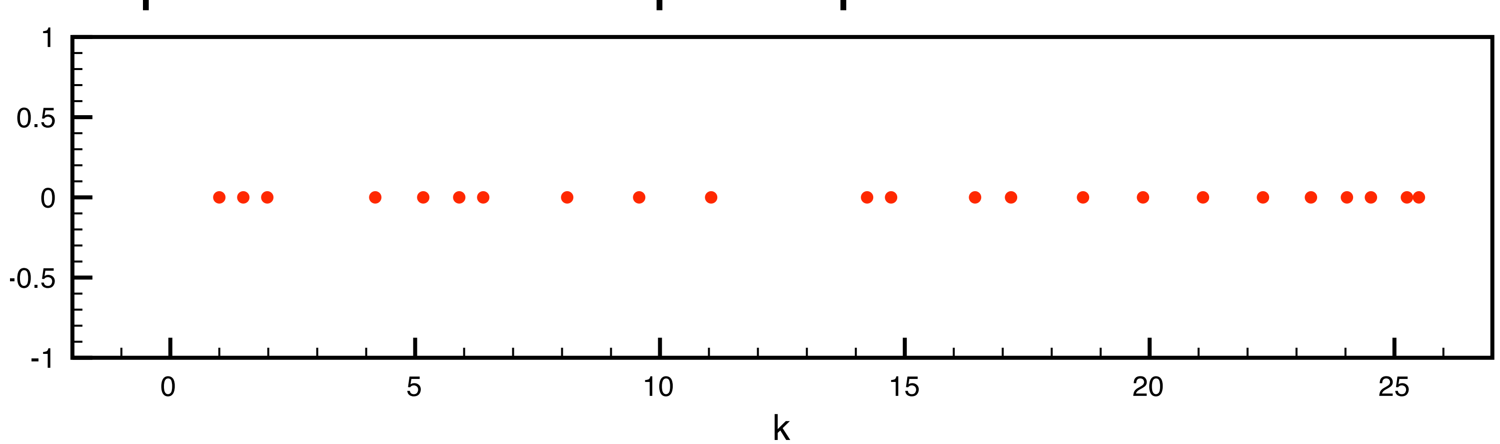






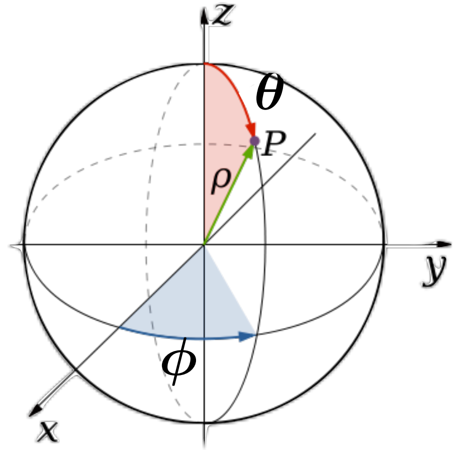


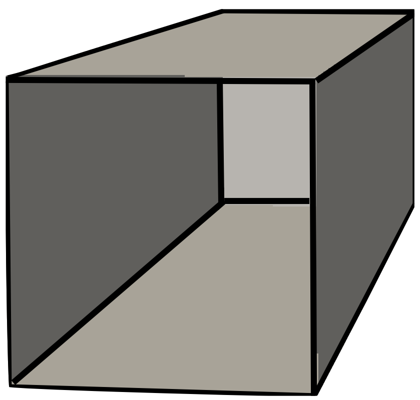
Repartition of 23 first picked parameters:

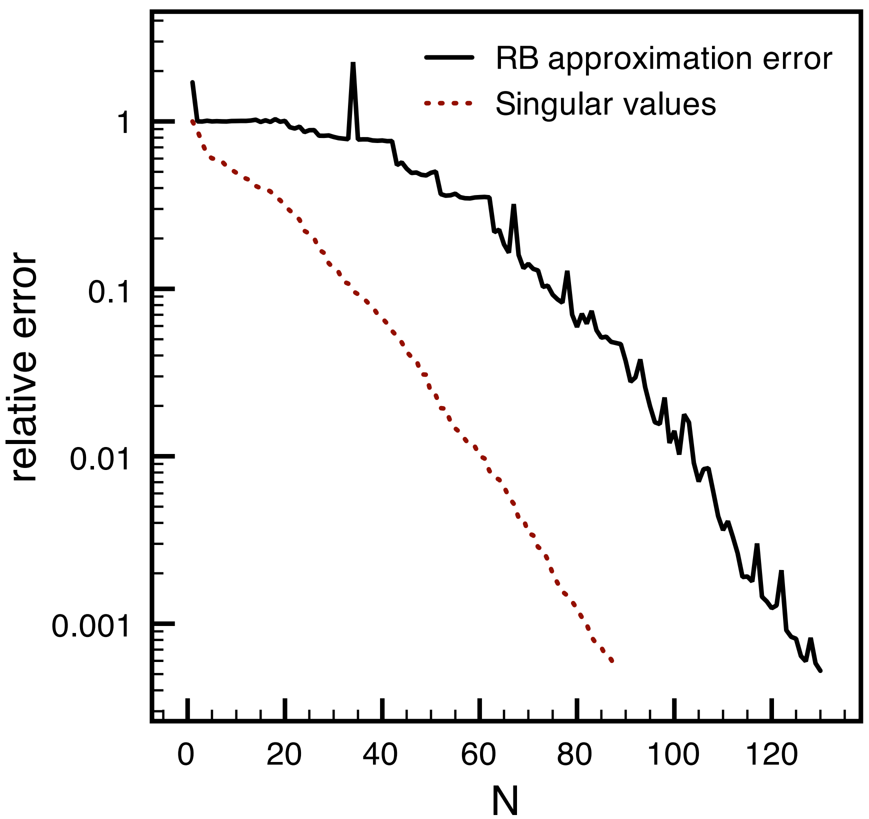


Scattering example

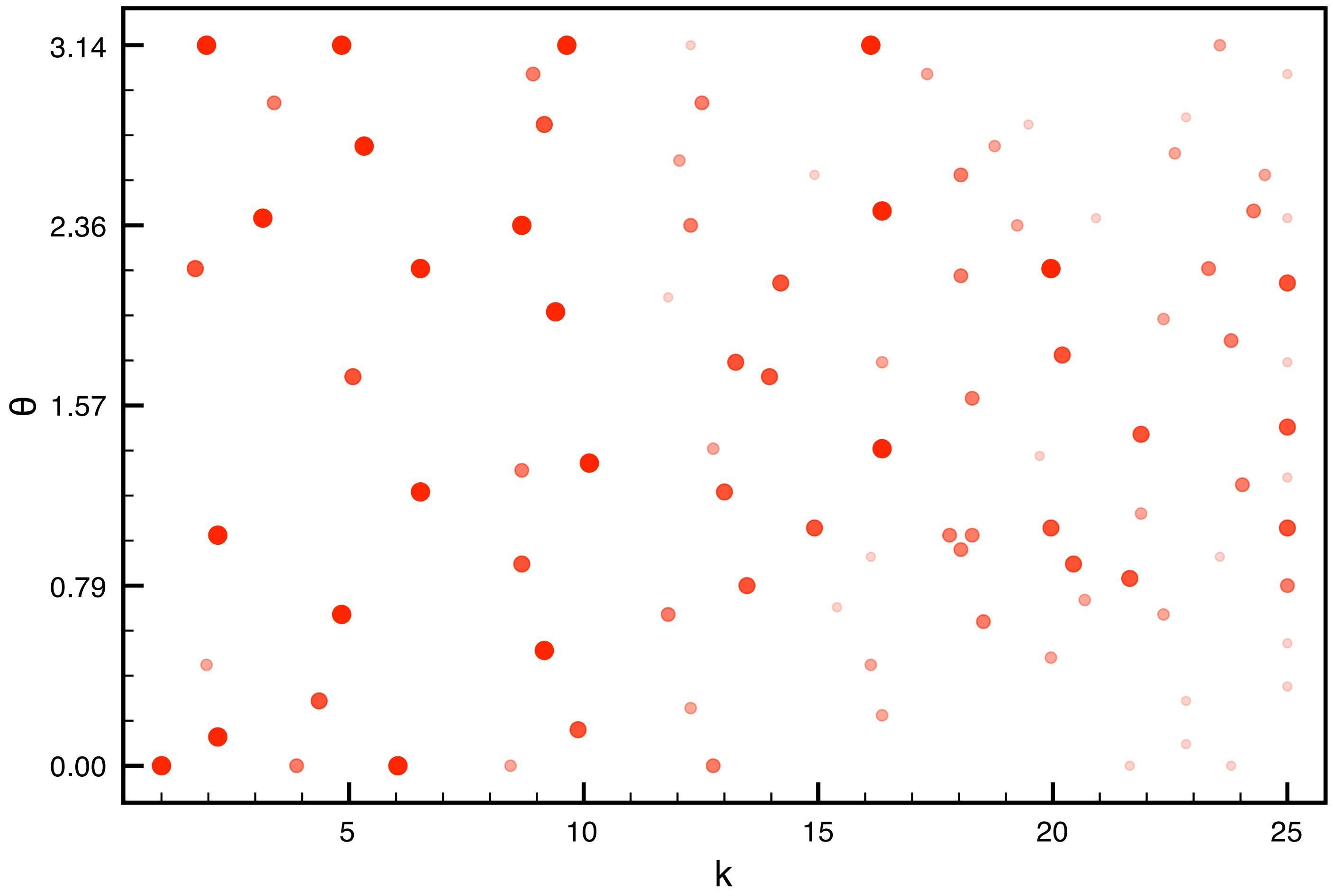
2 parameters, $\boldsymbol{\mu} = (k, \theta)$ with $\mathcal{D} = [1, 25] \times [0, \pi]$
$\phi = 0$ fixed

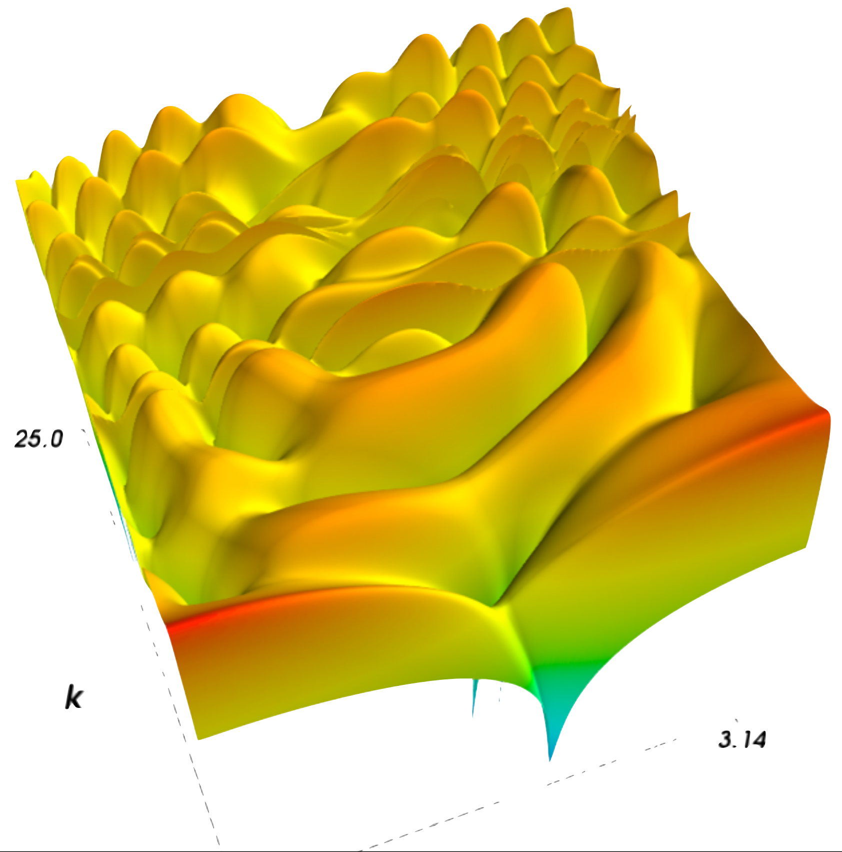






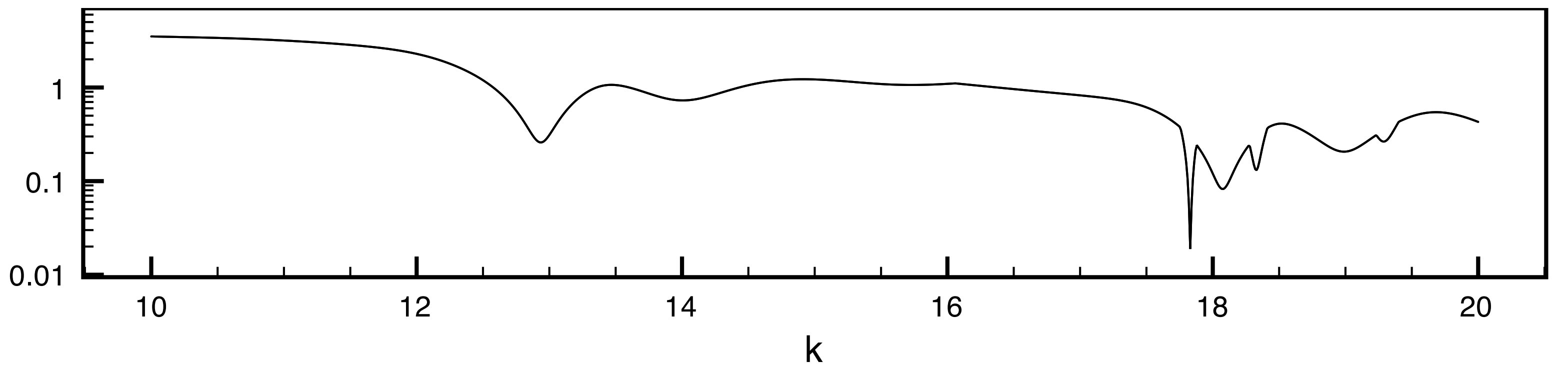
Picked parameters:

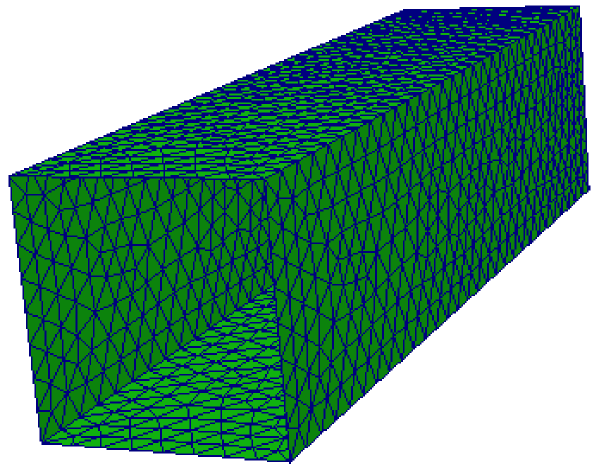




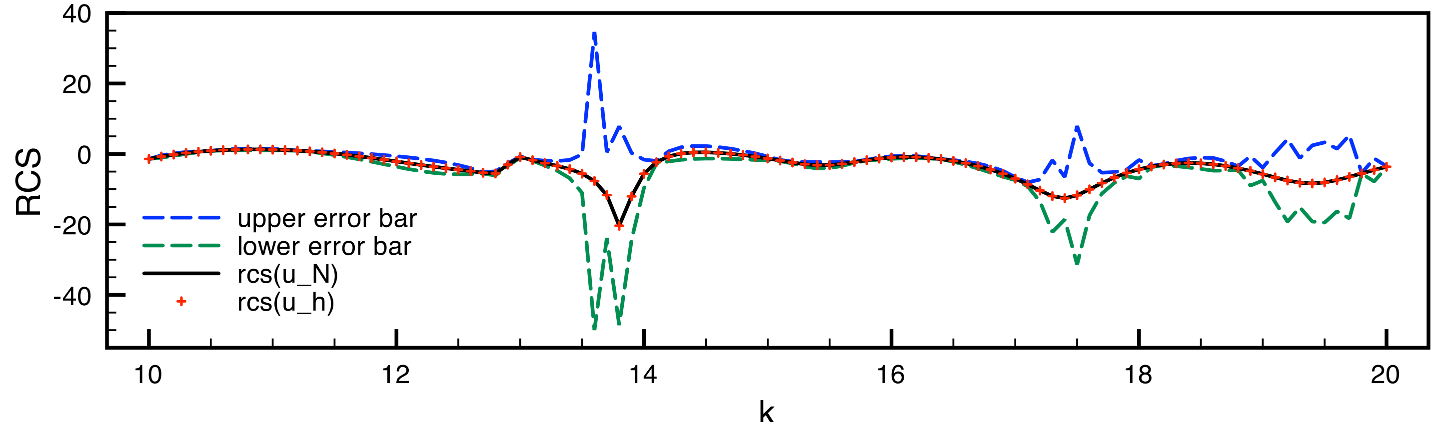
Scattering example

Stability parameter

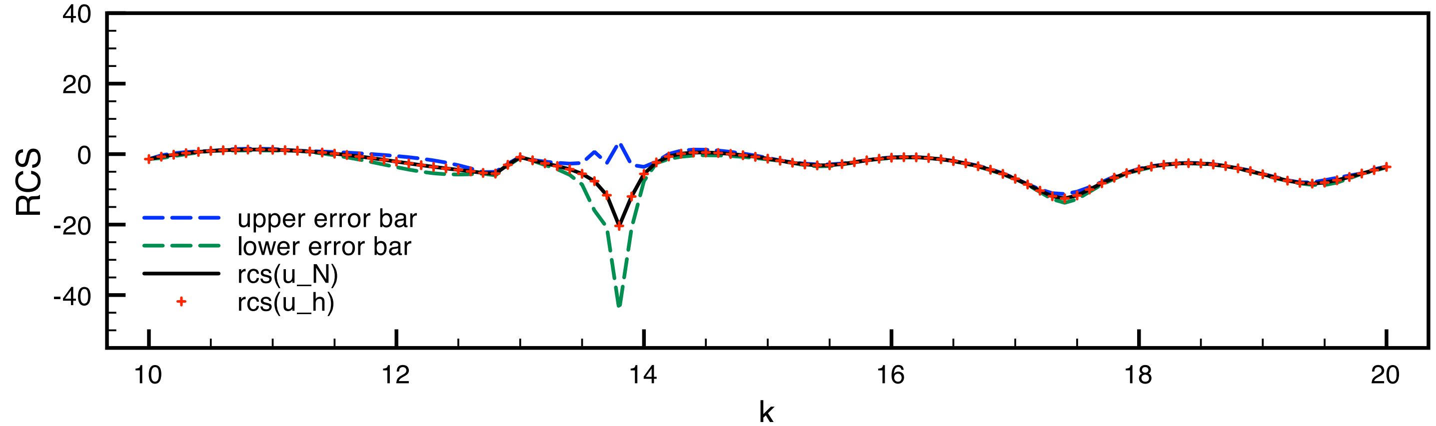




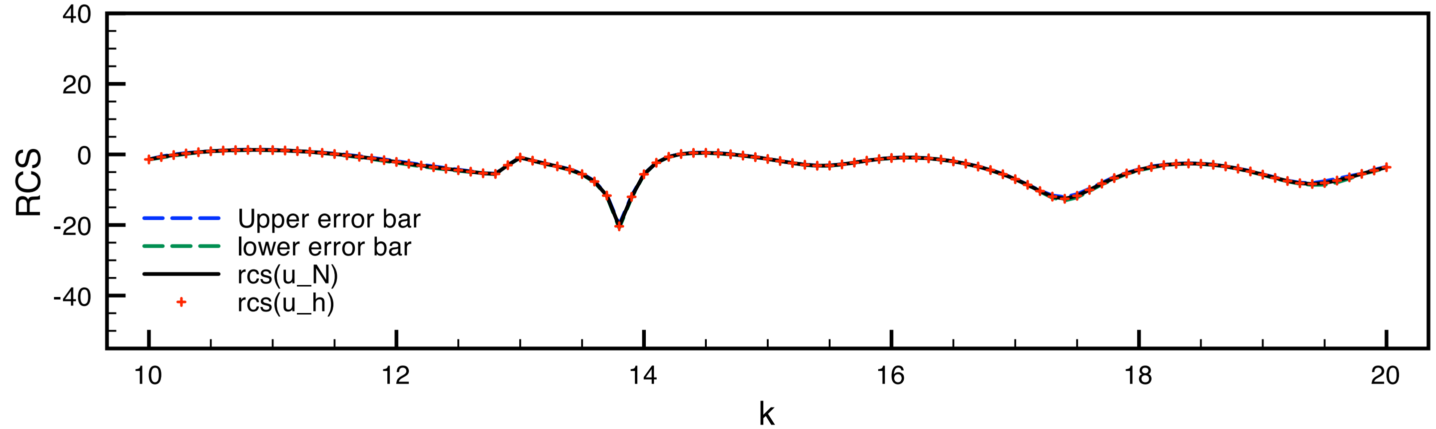
N=21



N=22



N=23



Bottlenecks

So where are the bottlenecks ?

In the **online** stage -

- Large number of terms in the affine expansion

$$a(u,\mu,v) = \sum_{k=1}^{Q_a} \Theta_k(\mu) a_k(u,v)$$

- Large number of terms in basis

$$N_d \propto N_1^{\alpha d}, \quad 0 < \alpha \leq 1$$

In the **offline** stage -

- The cost of greedy approach by evaluating the train space
  Also for SCM, EIM

$$\mu_{i+1} = \arg \sup_{\mu \in \Pi_{train}} \varepsilon_N(\mu)$$

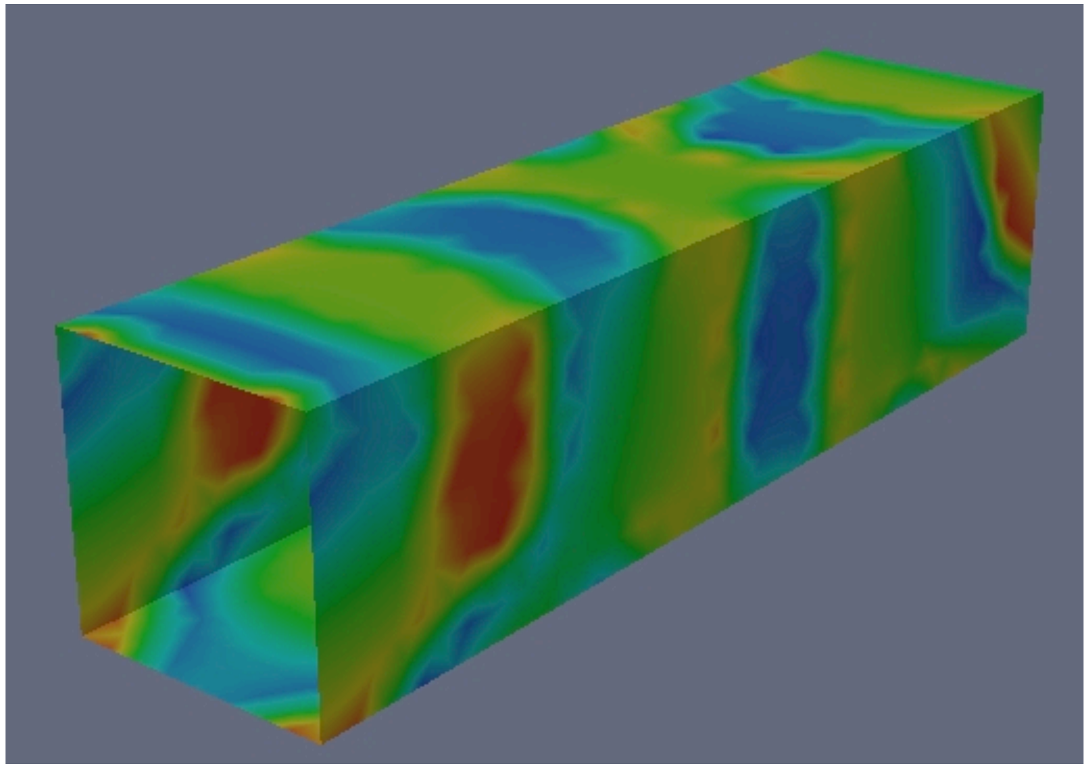
Non-affine problems

Let us consider the extension of these techniques to problems described by integral equations

Electric field integral equation (EFIE)

$$ik \int_{\Gamma\times\Gamma} G_k(\boldsymbol{x},\boldsymbol{y}) \left[\boldsymbol{j}(\boldsymbol{x}) \cdot \boldsymbol{j}^t(\boldsymbol{y}) - \tfrac{1}{k^2}\mathrm{div}_\Gamma \boldsymbol{j}(\boldsymbol{x}) \mathrm{div}_\Gamma \boldsymbol{j}^t(\boldsymbol{y})\right] d\boldsymbol{x} d\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{j}^t)$$

$$G_k(\boldsymbol{x},\boldsymbol{y}) := \frac{e^{ik|\boldsymbol{x}-\boldsymbol{y}|}}{|\boldsymbol{x}-\boldsymbol{y}|}.$$

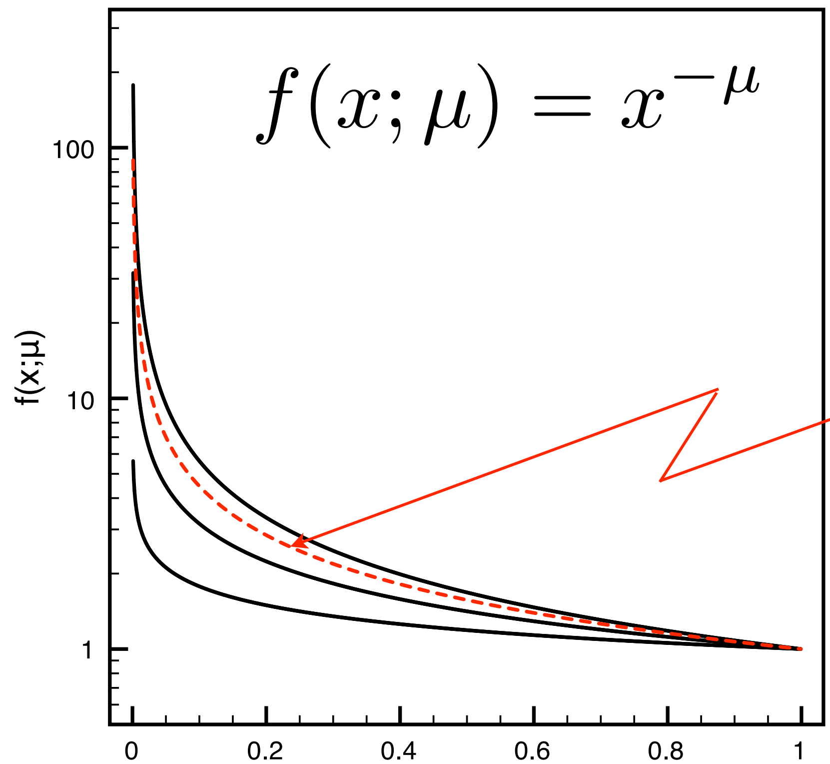


Truth approximation is a standard MoM solver.

CERFACS

Empirical Interpolation

Basic example



$$f(x;\mu) \approx \sum_{m=1}^{3} \alpha_m(\mu) x^{-\mu_m}$$

EFIE operators

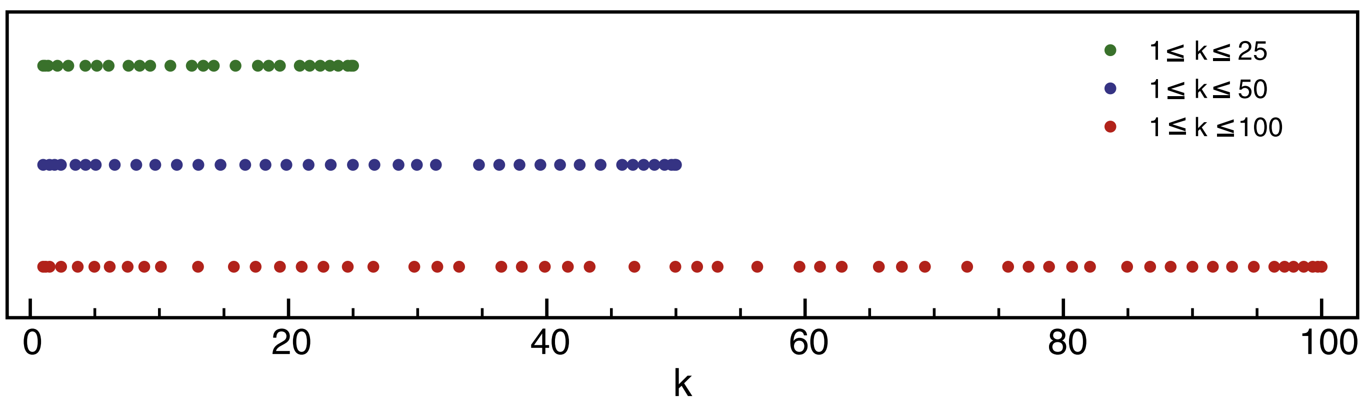
$$G_k^{ns}(r) = G^{ns}(r;k) = \frac{e^{ikr} - 1}{r}, \quad r \in \mathbb{R}^+, k \in \mathbb{R}^+$$

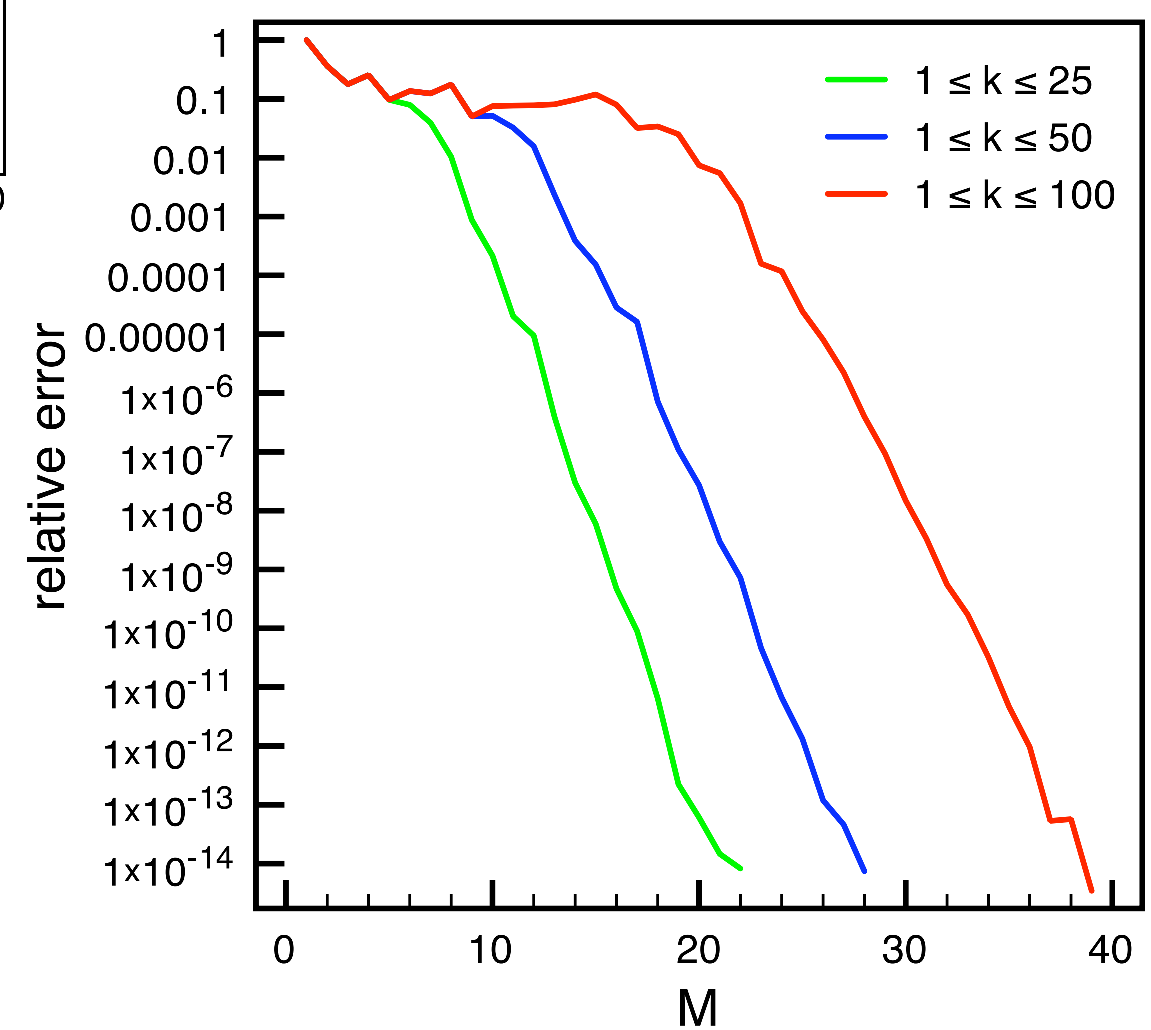
$$\mathbf{E}^i(\boldsymbol{x};\boldsymbol{\mu}) = -\boldsymbol{p}\, e^{ik\hat{\boldsymbol{s}}(\theta,\phi)\cdot\boldsymbol{x}}, \quad \boldsymbol{x} \in \Gamma, \boldsymbol{\mu} \in \mathcal{D}$$

EIM example

Basic test case

$$f(x;k) = \frac{e^{ikx} - 1}{x}, \quad x \in (0, R_{\max}], k \in [1, k_{\max}]$$





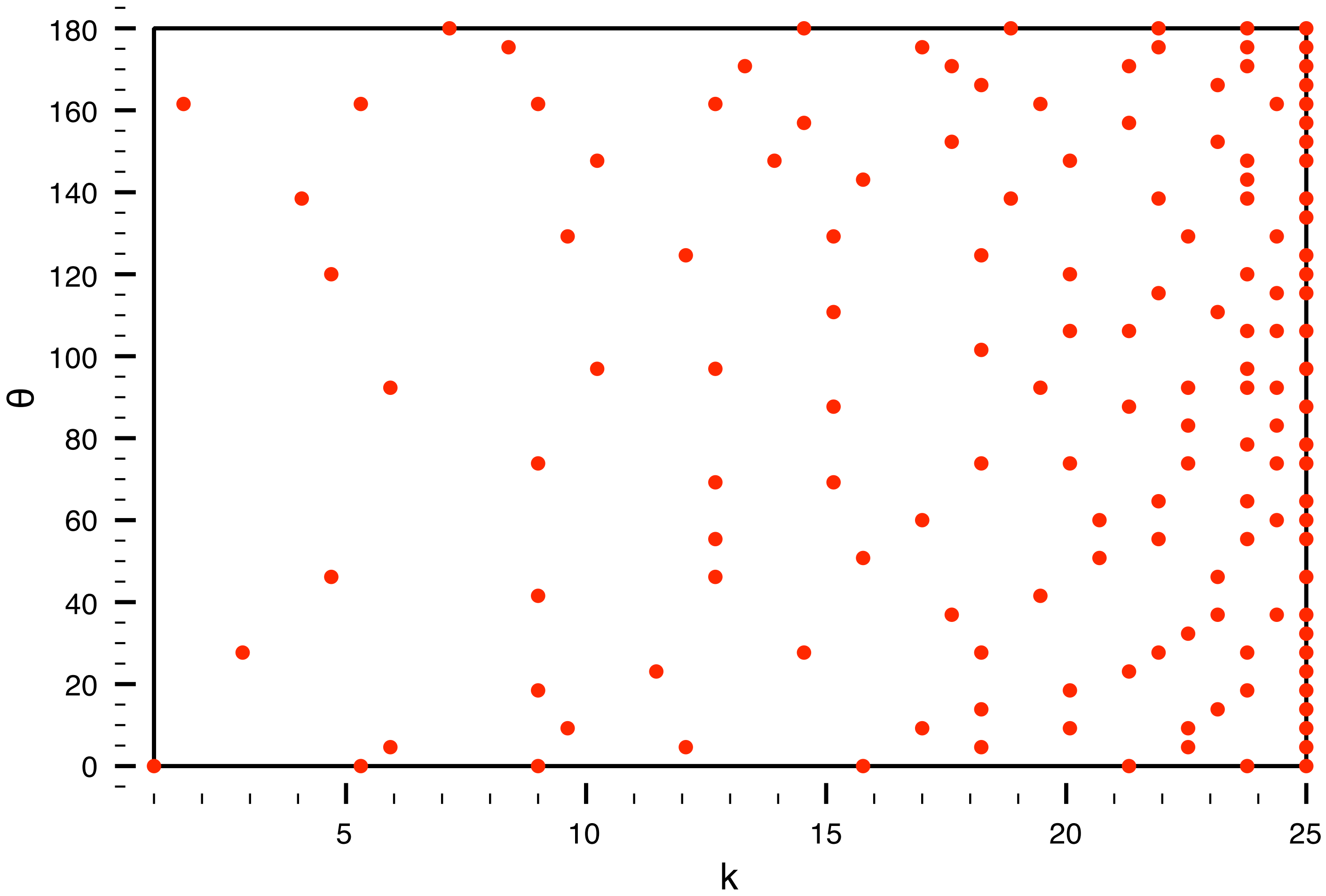
NOTE - Length of expansion directly impacts online cost

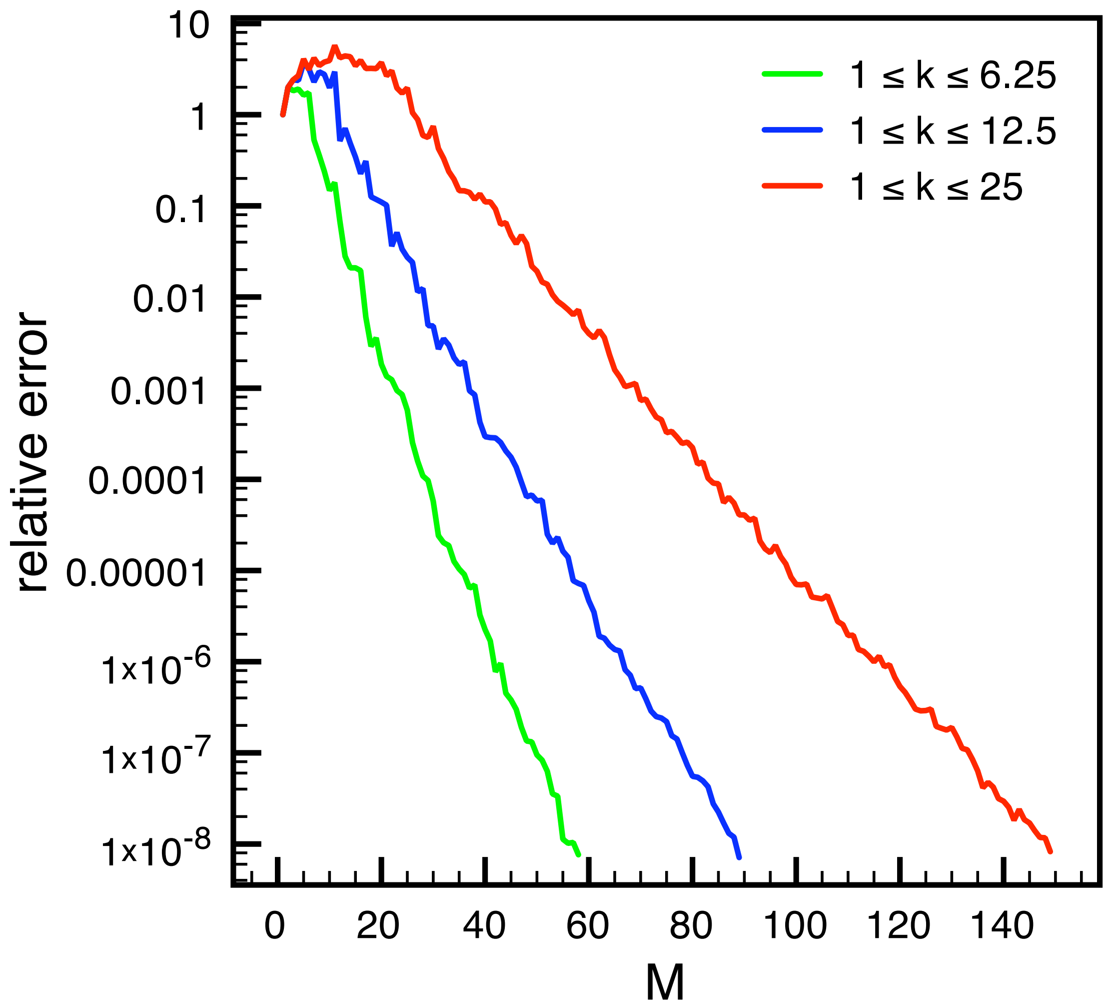
$$f(x,\mu) = \sum_{m=1}^{M} \alpha_m(\mu) f(x,\mu_m)$$

Integral equations

Results for EIM

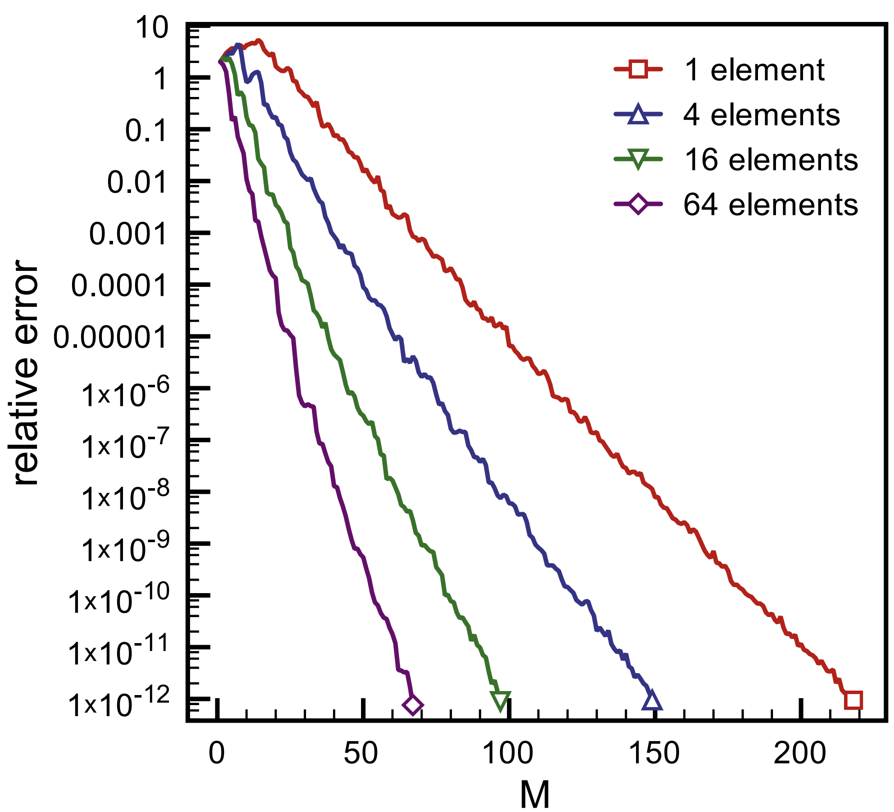
$$f(\boldsymbol{x};\boldsymbol{\mu}) = e^{ik\hat{\boldsymbol{s}}(\theta,\phi)\cdot\boldsymbol{x}}, \quad \boldsymbol{x}\in\Gamma, \boldsymbol{\mu}\in\mathcal{D},$$
$$\boldsymbol{\mu} = (k,\theta), \quad \phi \text{ fixed},$$
$$\mathcal{D} = [1, k_{\max}] \times [0,\pi]$$

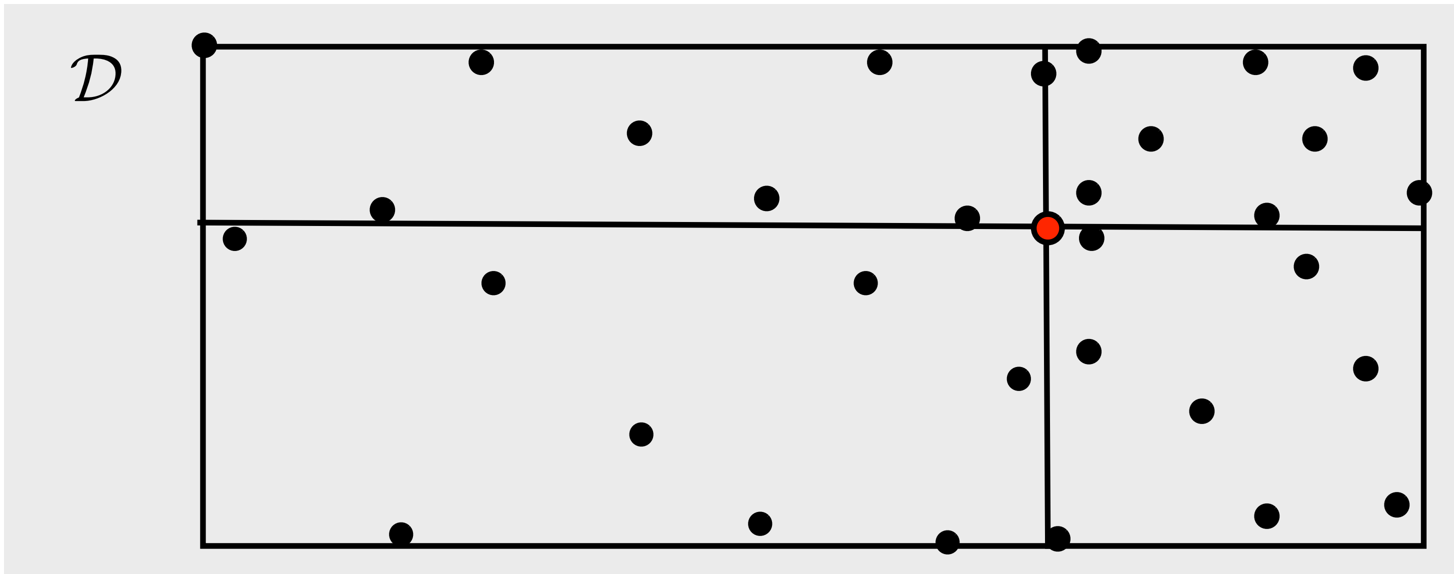


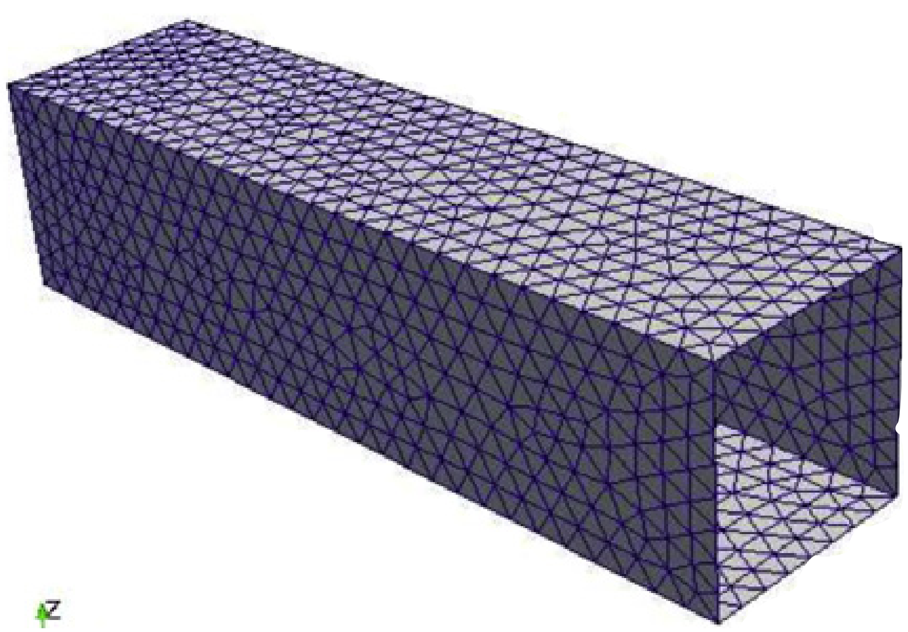


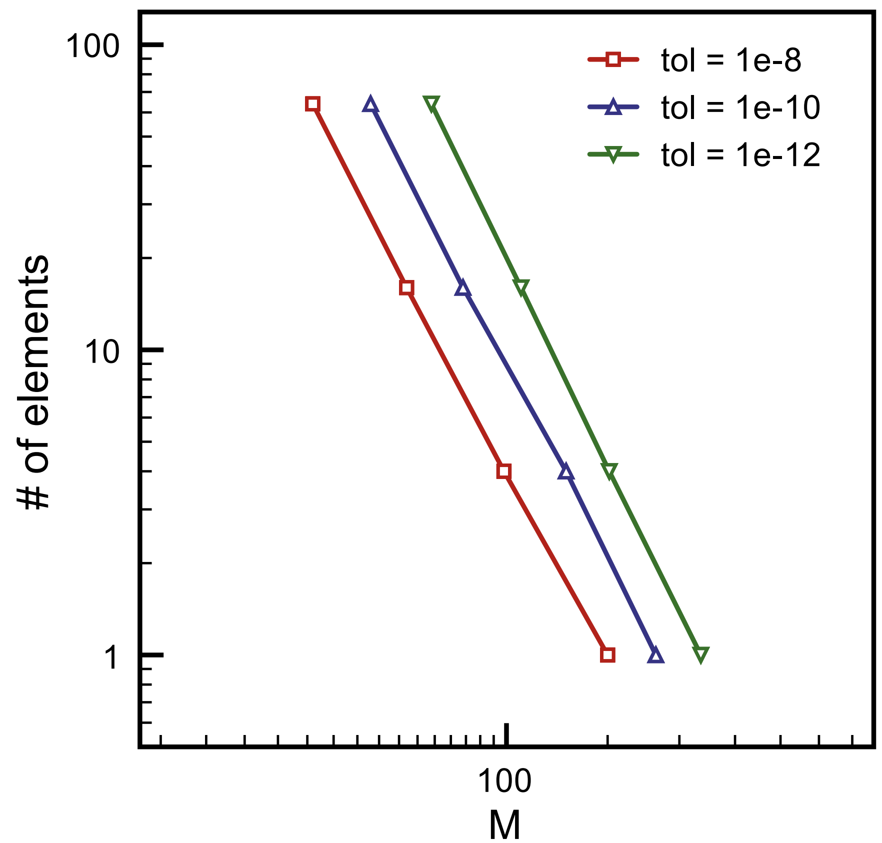
Element based EIM

Extension to an element based EIM





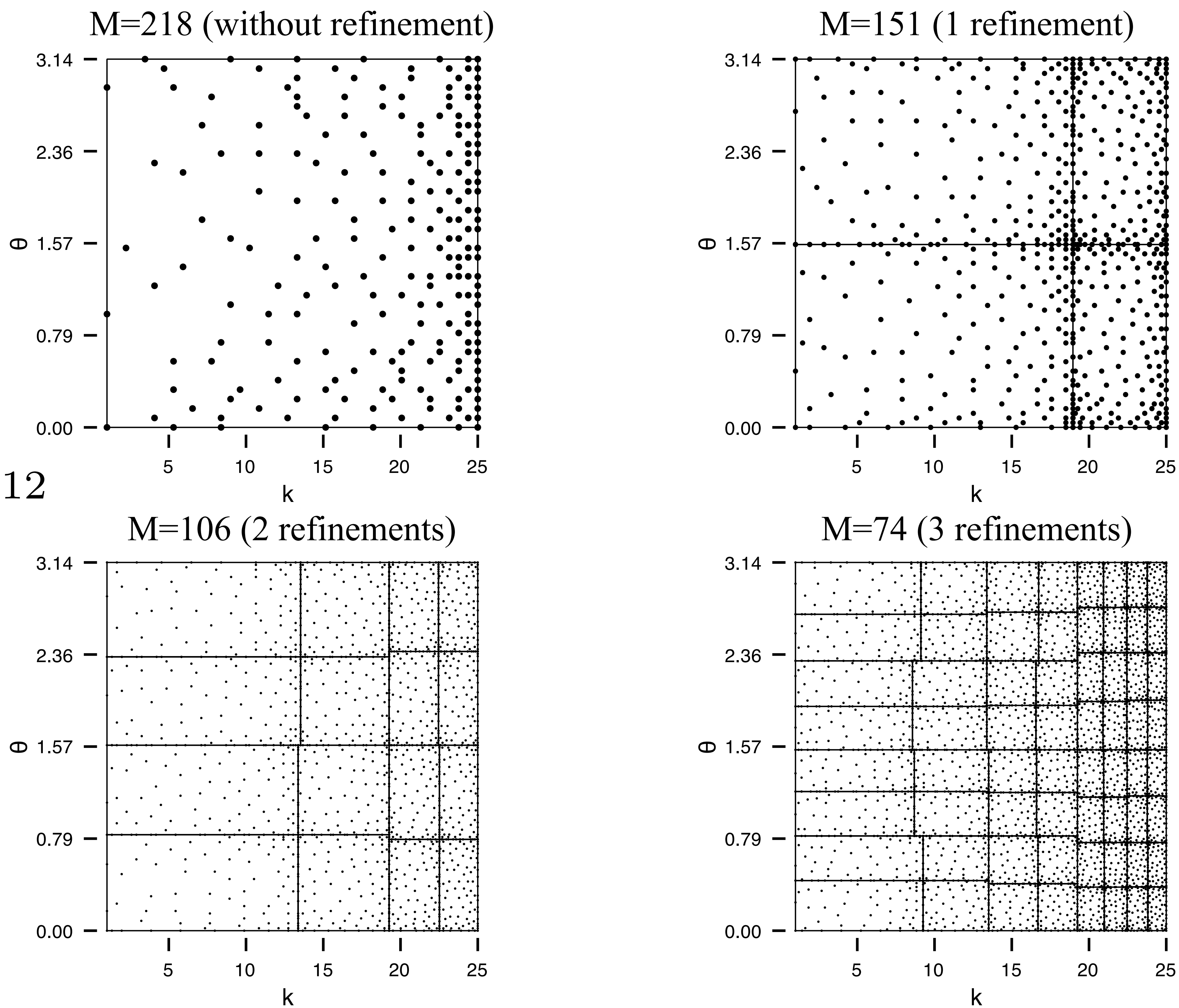




Objective is to reduce online cost

Adaptive EIM

Extension to multilevel EIM



$\varepsilon = 10^{-12}$

Bottlenecks

So where are the bottlenecks ?

In the **online** stage -

- Large number of terms in the affine expansion

$$a(u,\mu,v) = \sum_{k=1}^{Q_a} \Theta_k(\mu) a_k(u,v)$$

- Large number of terms in basis

$$N_d \propto N_1^{\alpha d}, \quad 0 < \alpha \leq 1$$

In the **offline** stage -

- The cost of greedy approach by evaluating the train space
  Also for SCM, EIM

$$\mu_{i+1} = \arg \sup_{\mu \in \Pi_{train}} \varepsilon_N(\mu)$$

Strategy for high-d sampling

A typical greedy approach

Input: A train set $\Xi_{train} \subset \mathcal{D}$, a tolerance $tol > 0$
Output: S_N and W_N

Initialization: Choose an initial parameter value $\boldsymbol{\mu}^1 \in \Xi_{train}$, set $S_1 = \{\boldsymbol{\mu}^1\}$, compute $v(\boldsymbol{\mu}^1)$, set $W_1 = \{v(\boldsymbol{\mu}^1)\}$, and $N = 1$;
while $\max_{\boldsymbol{\mu}\in\Xi} \eta(\mu; W_N) > tol$ **do**
  $\boxed{\text{For all } \boldsymbol{\mu} \in \Xi_{train}, \text{ compute } \eta(\boldsymbol{\mu}; W_N)\ ;}$
  Choose $\boldsymbol{\mu}^{N+1} = \operatorname{argmax}_{\boldsymbol{\mu}\in\Xi_{train}} \eta(\boldsymbol{\mu}; W_N)$;
  Set $S_{N+1} = S_N \cup \{\boldsymbol{\mu}^{N+1}\}$;
  Compute $v(\boldsymbol{\mu}^{N+1})$, and set $W_{N+1} = W_N \cup \{v(\boldsymbol{\mu}^{N+1})\}$;
  $N \leftarrow N + 1$;
end while

For a high-d problem with a large training set, this is expensive

Strategy for high-d sampling

We introduce <u>the saturation assumption</u>

$$\eta(\boldsymbol{\mu}; W_M) \leq C_{sa}\, \eta(\boldsymbol{\mu}; W_N) \quad \textit{for some} \quad C_{sa} > 0 \quad \textit{for all} \quad 0 < N < M$$

Different interpretations

- For $C_{sa} < 1$ - error is strictly decreasing
- For $C_{sa} \geq 1$ - error is allowed to increase (intermittently)

We shall use this approach to propose two different sampling

- An approach using just this saturation assumption
- An adaptive sampling with additional benefits

Strategy for high-d sampling

Strategy I -

When looking for the max error over the training set, recompute estimator only for those points where

$$C_{sa}\eta(\mu, W_N) \geq error_{tmpmax}$$

while $\max_{\boldsymbol{\mu}\in\Xi_{train}} \eta_{saved}(\boldsymbol{\mu}) \geq tol$ **do**

  $error_{tmpmax} = 0;$

  for all $\boldsymbol{\mu} \in \Xi_{train}$ **do**

    if $C_{sa}\eta_{saved}(\boldsymbol{\mu}) > error_{tmpmax}$ **then**

      Compute $\eta(\boldsymbol{\mu}; W_N)$, and let $\eta_{saved}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}, W_N);$

      if $\eta_{saved}(\boldsymbol{\mu}) > error_{tmpmax}$ **then**

        $error_{tmpmax} = \eta_{saved}(\boldsymbol{\mu})$, and let $\boldsymbol{\mu}_{max} = \boldsymbol{\mu};$

      end if

    end if

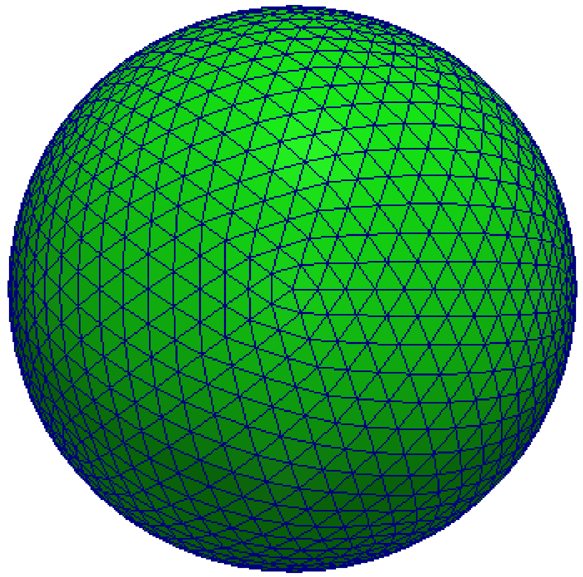
  end for

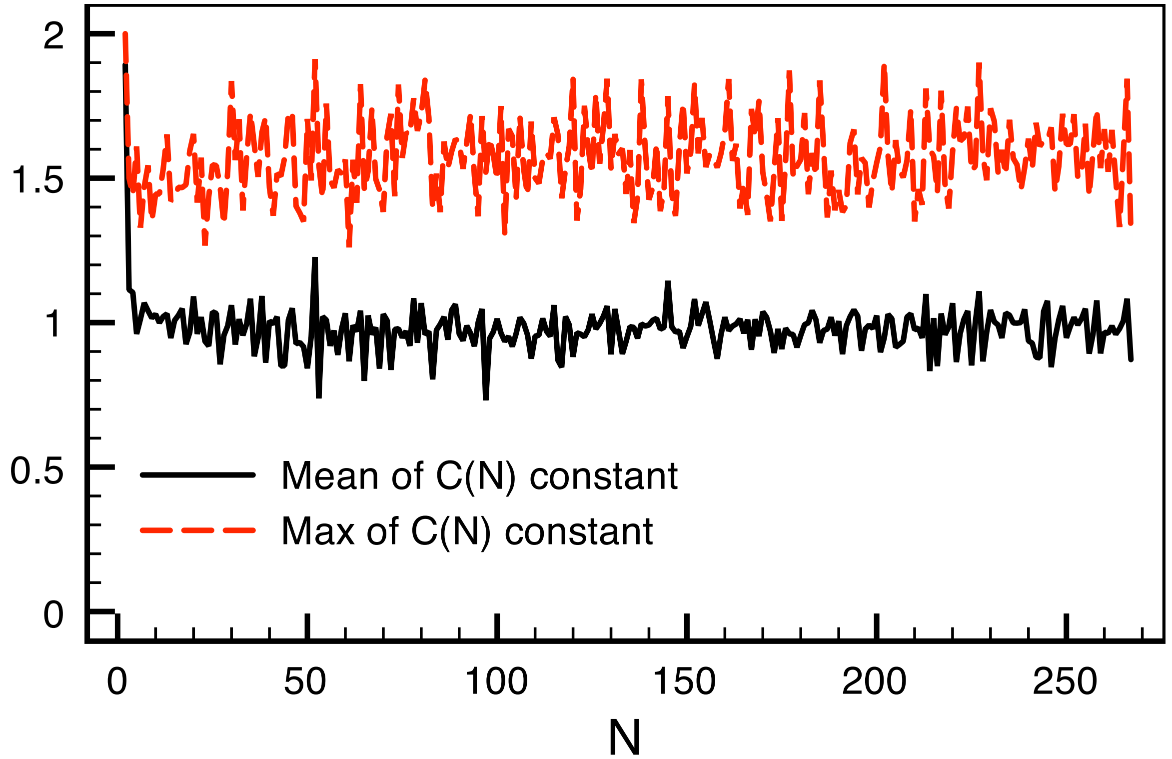
Strategy for high-d sampling

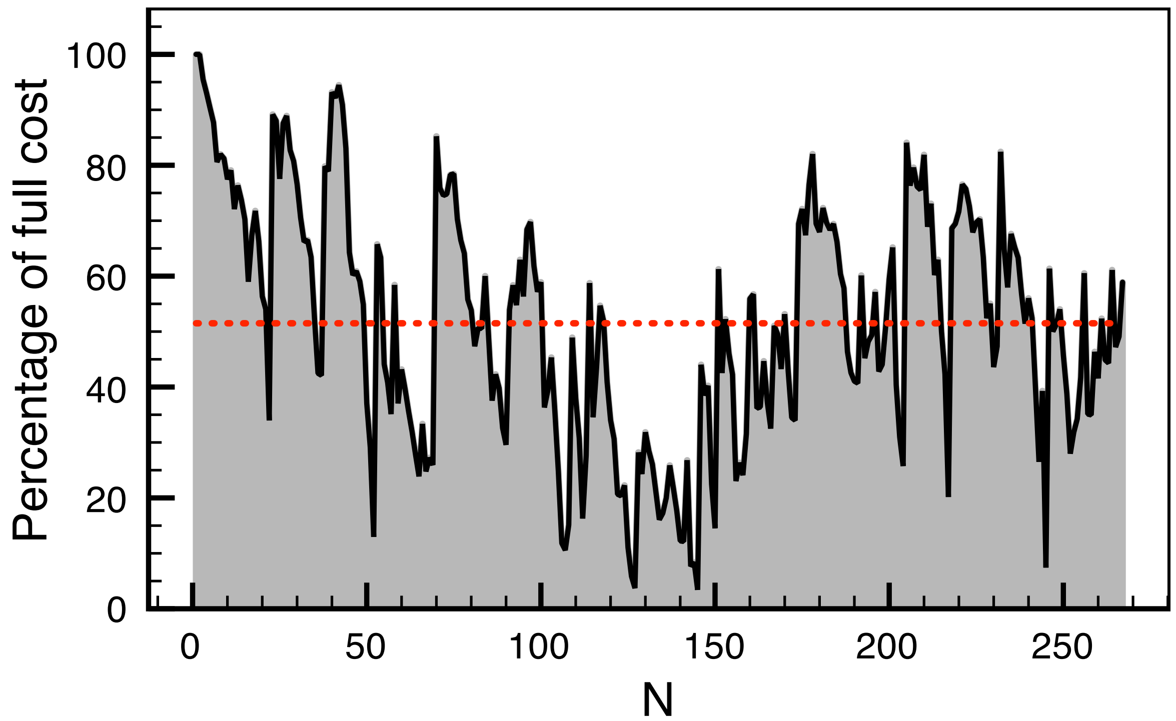
Consider EIM example

$$\mathcal{F}_2(\mathbf{x};\boldsymbol{\mu}) = e^{ik\hat{\boldsymbol{k}}\cdot\mathbf{x}}$$

$$\hat{\boldsymbol{k}} = -(\sin\theta\,\cos\varphi, \sin\theta\,\sin\varphi, \cos\theta)^T$$







$$C(N) = \frac{\eta(\boldsymbol{\mu}, W_N)}{\eta(\boldsymbol{\mu}, W_{N-1})}$$

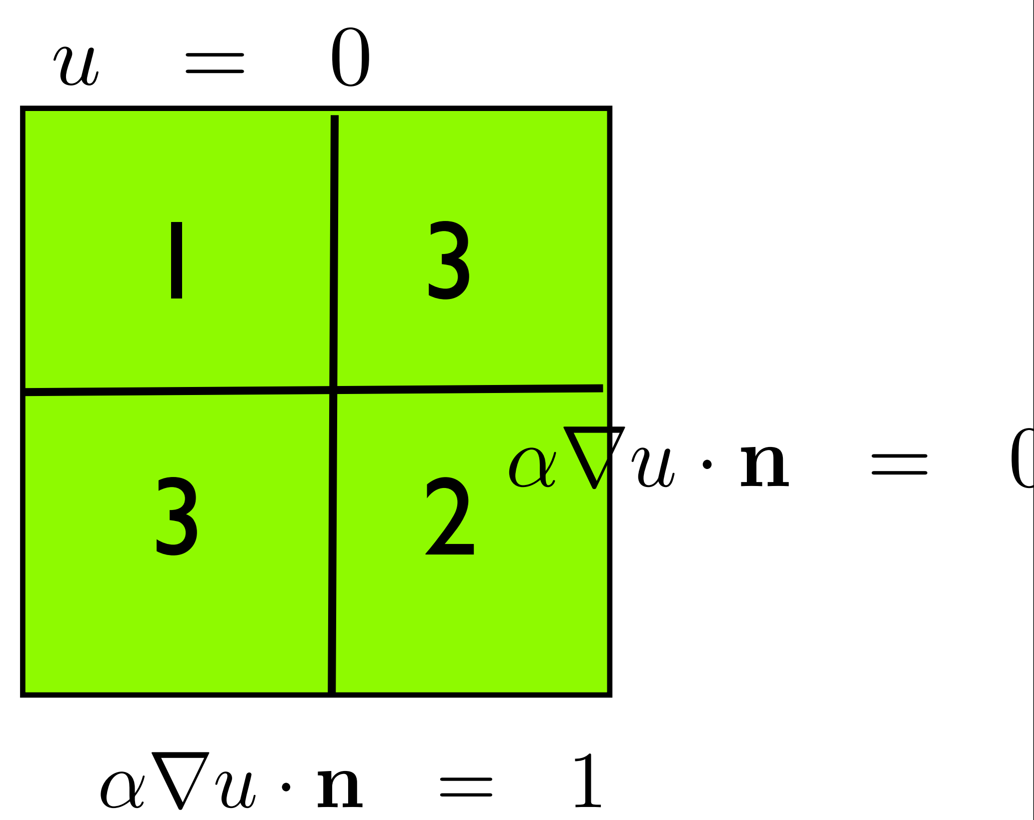
Strategy for high-d sampling

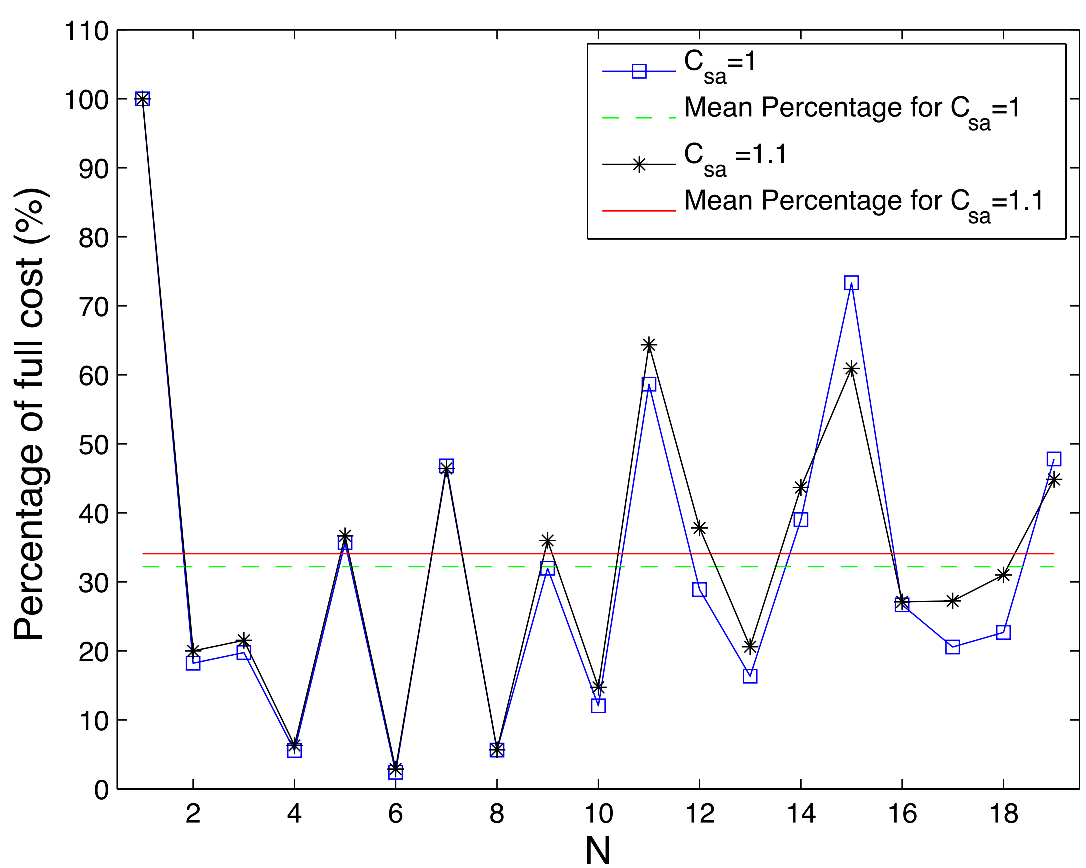
Consider a RBM example

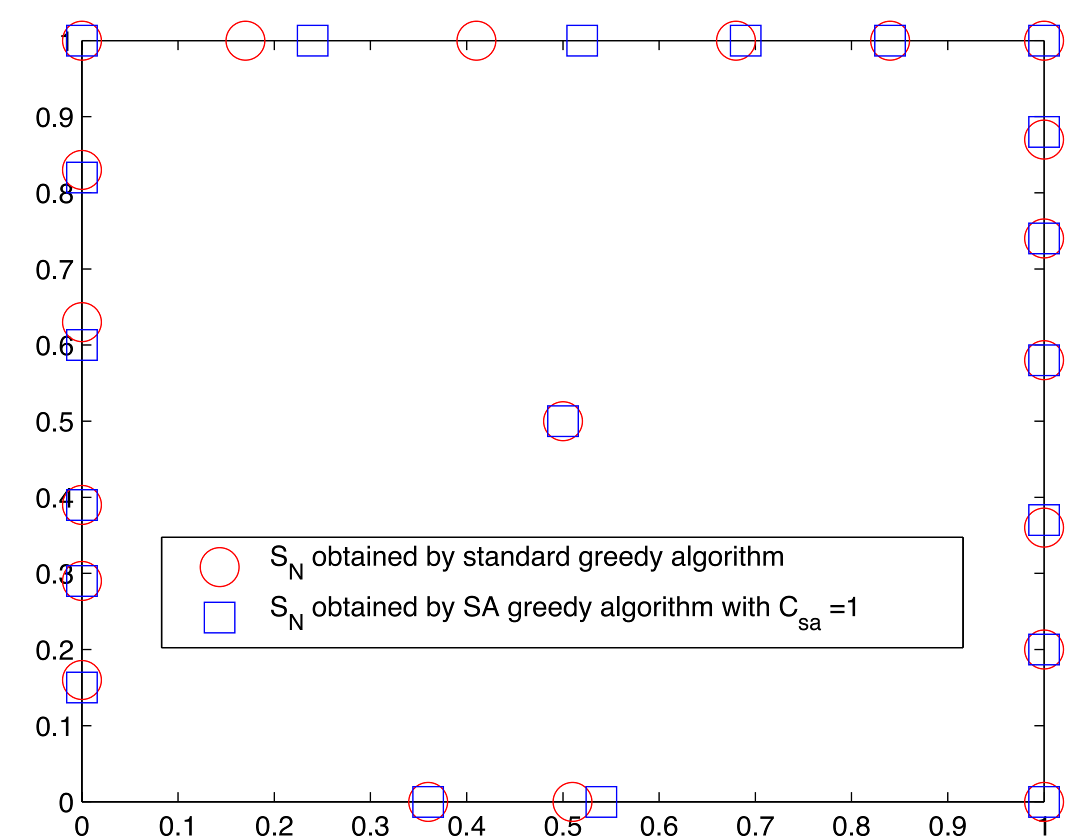
$$-\nabla \cdot (\alpha \nabla u) \quad = \quad 0 \quad \text{in} \quad \Omega,$$

$$\alpha = \begin{cases} \alpha_i & = & 100^{2\mu_i - 1}, & x \in R_i, i = 1, 2, \\ \alpha_3 & = & 1, & x \in R_3, \end{cases}$$

$$\boldsymbol{\mu} = (\mu_1, \mu_2) \in [0, 1]^2.$$

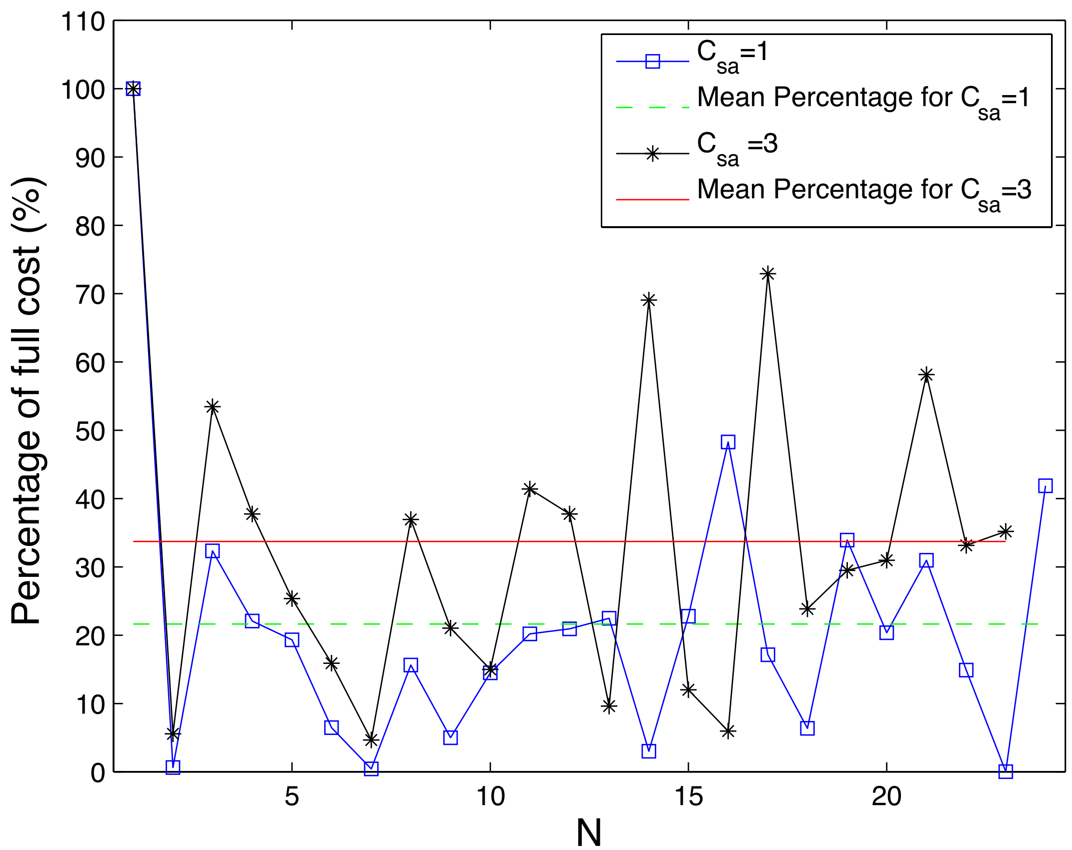






Strategy for high-d sampling

Dependence on safety factor



Observations -

- Easy modification of standard greedy
- Relies on a reasonable assumption of convergence
- Potential offline saving of close to an order of magnitude
- Still requires a fixed (large) training set

Strategy for high-d sampling

Strategy II -

- Choose training set size as can be afforded and sample randomly
- Resample #points for which $\eta(\mu, W_N) < tol$
- Perform a safety check

```
if η_saved(μ) < tol then
   flag μ; // all flagged parameters will be removed
end if
```

if $\eta_{saved}(\boldsymbol{\mu}) < tol$ **then**
flag $\boldsymbol{\mu}$; `// all flagged parameters will be removed`
end if

Discard all flagged parameters from Ξ_{train} and their corresponding saved error estimation in η_{saved};

Generate $M - \text{sizeof}(\Xi_{train})$ new samples, add them into Ξ_{train} such that $\text{sizeof}(\Xi_{train}) = M$; set η_{saved} of all new points to ∞;

Discard Ξ_{train}, generate M new parameters to form Ξ_{train}

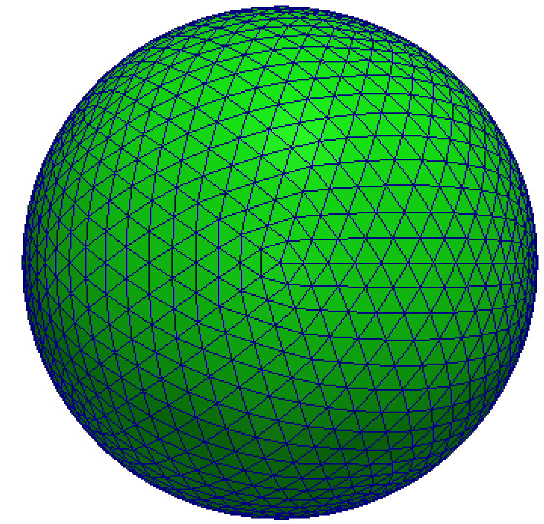
Strategy for high-d sampling

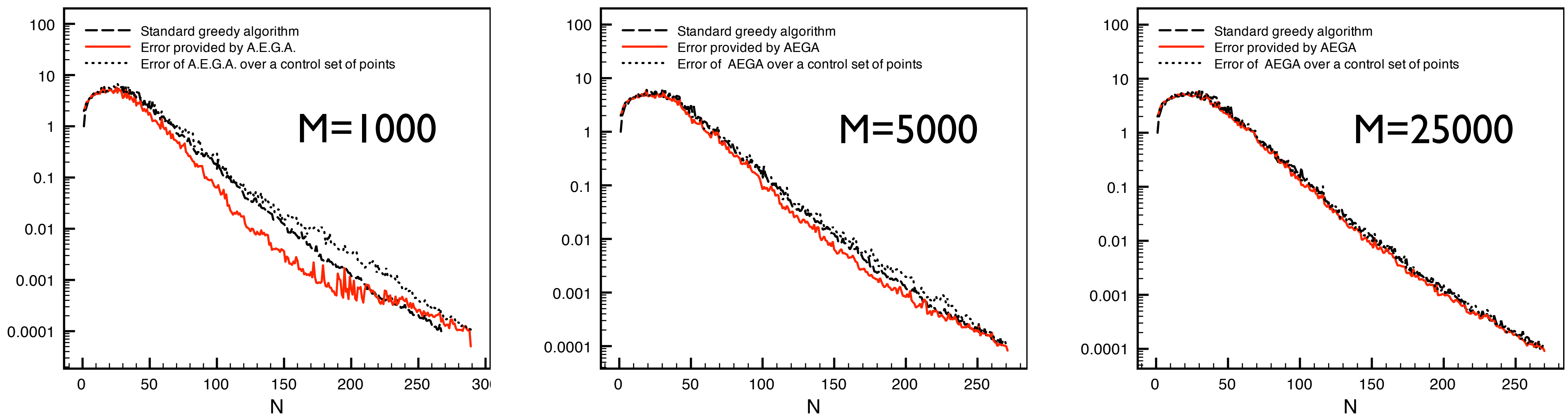
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

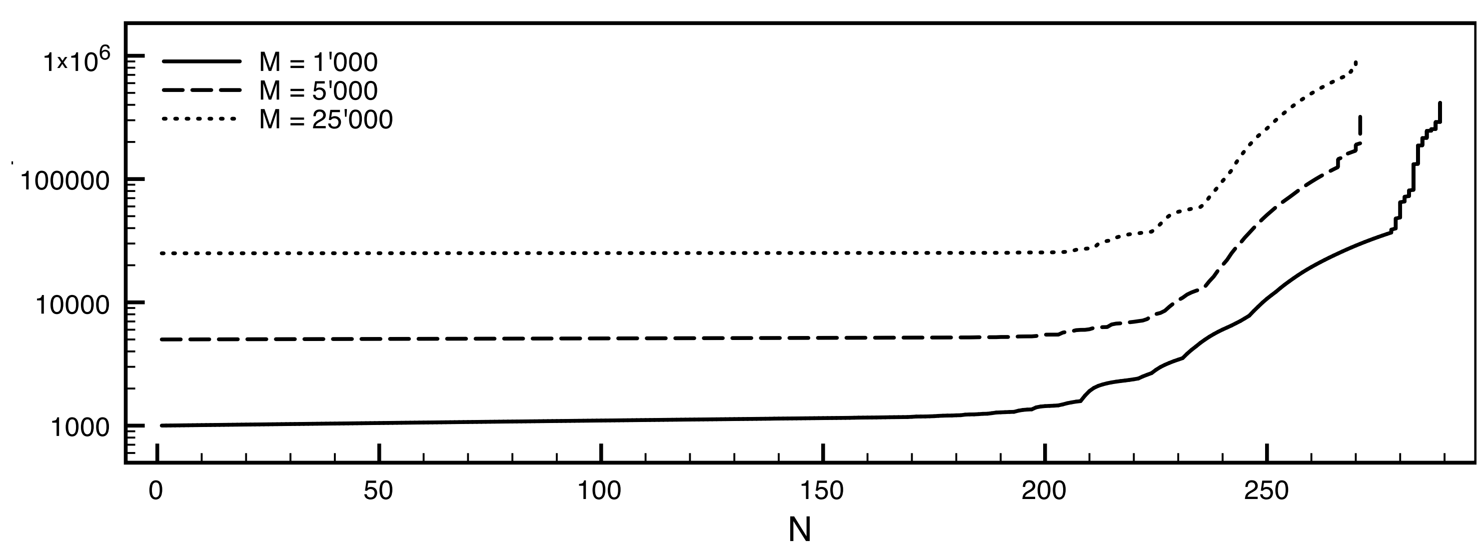
Consider EIM example

$$\mathcal{F}_2(\mathbf{x};\boldsymbol{\mu}) = e^{ik\hat{\boldsymbol{k}}\cdot\mathbf{x}}$$

$$\hat{\boldsymbol{k}} = -(\sin\theta\,\cos\varphi, \sin\theta\,\sin\varphi, \cos\theta)^T$$







Strategy for high-d sampling

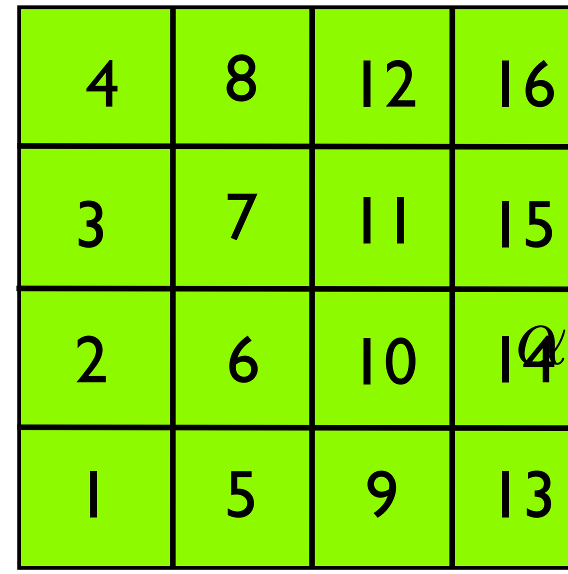
Consider a RBM example

$$-\nabla \cdot (\alpha \nabla u) = 0 \quad \text{in} \quad \Omega,$$

$$\alpha = \begin{cases} \alpha_k = 5^{2\mu_k - 1}, & x \in R_k, \quad k = 1, 2, \cdots, 15, \\ \alpha_{16} = 1, & x \in R_{16}. \end{cases}$$

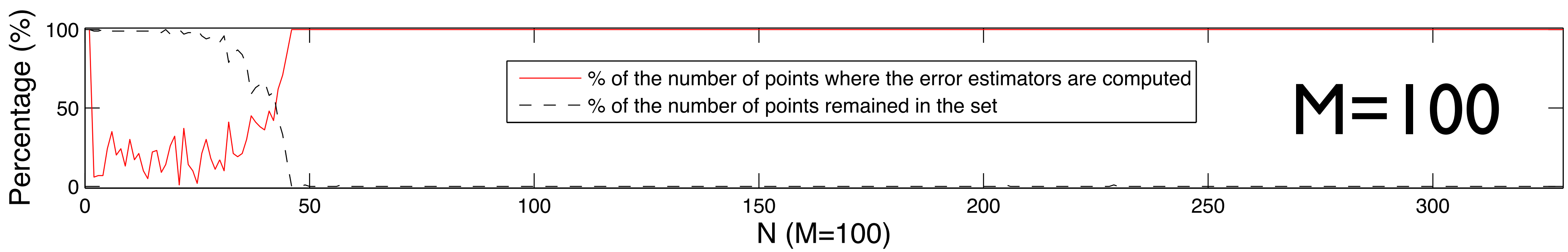
$$\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_{15}) \in [0, 1]^{15}$$

$$u = 0$$

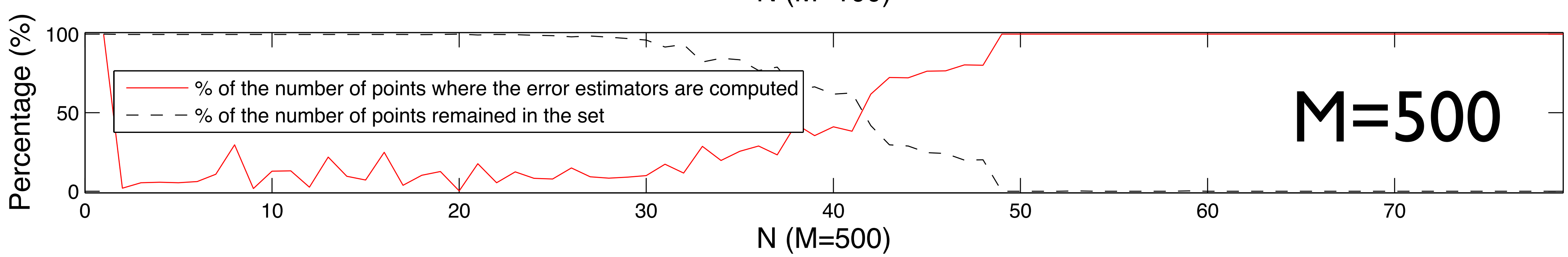


$$\alpha \nabla u \cdot \mathbf{n} = 0$$

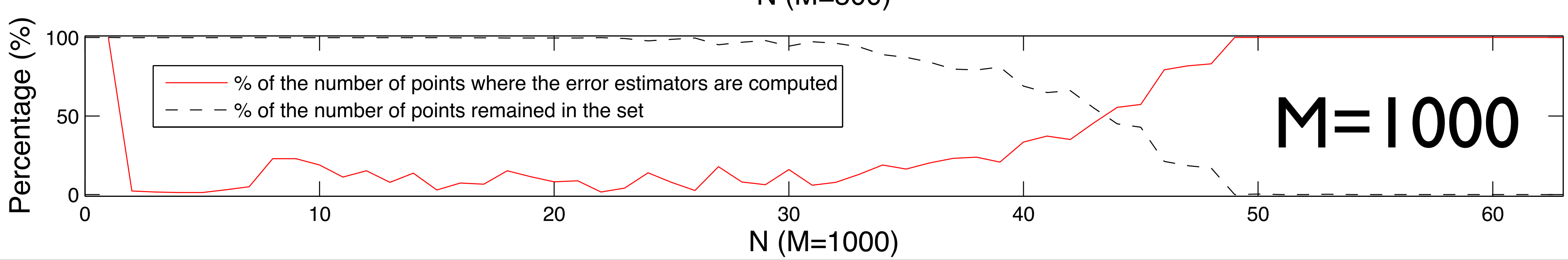
$$\alpha \nabla u \cdot \mathbf{n} = 1$$



$$N_{rb} = 52$$



$$N_{rb} = 50$$



$$N_{rb} = 50$$

Fundamental problem remains

While these tricks remain valuable, both offline and online cost typically scales with

$$M \propto (QN)^{\alpha d}, \quad 0 < \alpha \leq 1$$

For d>>1 this quickly becomes very expensive

Goal

Reduce the dimensionality of the problem without impacting the predictive accuracy

✓ Compression by ANOVA expansions
✓ Problem segmentation

ANOVA Expansions

In many cases we need to evaluate

$$f(\mathbf{X}(x)) \qquad \int f(\mathbf{X}(x))\,dx \qquad \mathbf{X} = (X_1, \ldots, X_d), \quad d \gg 1$$

which quickly becomes an expensive exercise.

<u>DEF</u>: The ANOVA expansion (exact)

$$f(\mathbf{X}) = f_0 + \sum_{t \subseteq \mathcal{D}} f_t(\mathbf{X}^t)$$

$$f_t(\mathbf{X}^t) = \int_{A^{d-|t|}} f(\mathbf{X}) d\mathbf{X}_{\mathcal{D}/t} - \sum_{w \subset t} f_w(\mathbf{X}^w) - f_0$$

$$f_0 = \int_{A^d} f(\mathbf{X})\,d\mathbf{X}, \quad \int_{A^0} f(\mathbf{X})\,d\mathbf{X}^0 = f(\mathbf{X})$$

$$\mathcal{D} = \{1, \ldots, d\}$$

$$\Omega = [0,1]^d$$

$$A^{|t|}$$

|t| dimensional hypercube

$$\mathbf{X}^t$$

t indexed sub-vector

ANOVA Expansions

A few characteristics -

- The ANOVA expansion is unique and exact
- It is a finite expansion with 2^d terms
- All terms are mutually orthogonal

Example:

$$f(\alpha_1, \alpha_2, \alpha_3) = f_0 + \sum_{i=1}^{3} \hat{f}_i(\alpha_i) + \sum_{1=i<j<d} \hat{f}_{ij}(\alpha_i, \alpha_j)$$

We have not achieved much yet.

Now consider the truncated expansion

$$f(\mathbf{X}, s) = f_0 + \sum_{t \subseteq \mathcal{D}; |t| \leq s} f_t(\mathbf{X}^t)$$

S = truncation dimension

ANOVA Expansions

Let us first introduce

$$V_t(f) = \int_{A^d} (f_t(\mathbf{X}^t))^2 \, d\mathbf{X}, \quad V(f) = \sum_{|t|>0} V_t(f)$$

... dimension-specific variances

Define the effective dimension through

$$\sum_{0<|t|\leq p_s} V_t(f) \geq qV(f) \qquad\qquad q \leq 1$$

Then one can prove $\quad \mathrm{Err}(\mathbf{X}, p_s) \leq 1 - q$

Sobol'90

$$\mathrm{Err}(\mathbf{X}, p_s) = \frac{1}{V(f)} \int_{A^d} [f\mathbf{X} - f(\mathbf{X}, p_s)]^2 \, d\mathbf{X}$$

NOTE: If p<<d there is hope!

Parametric compression

Subset specific variances

$$V_t(f) = \int_{A^d} (f_t(\mathbf{X}^t))^2 \, d\mathbf{X}, \;\; V(f) = \sum_{|t|>0} V_t(f)$$

and introduce sensitivities

$$S(t) = \frac{V_t}{V},$$

We can now estimate sensitivity of output on specific parameter through

$$\sum_{i \in t} S(i)$$

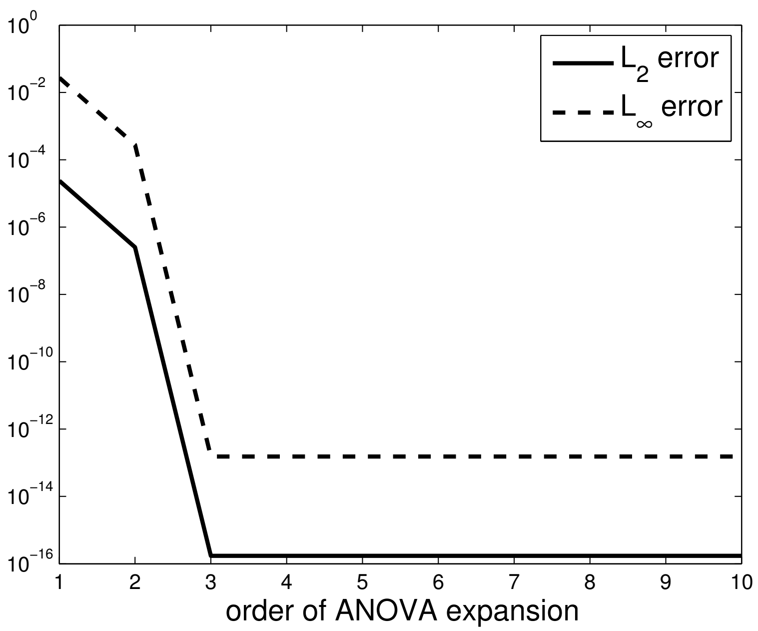
- Compute approximate ANOVA expansion - learn
- Identity important parameters and compress
- Compute lower dimensional model

ANOVA Expansions

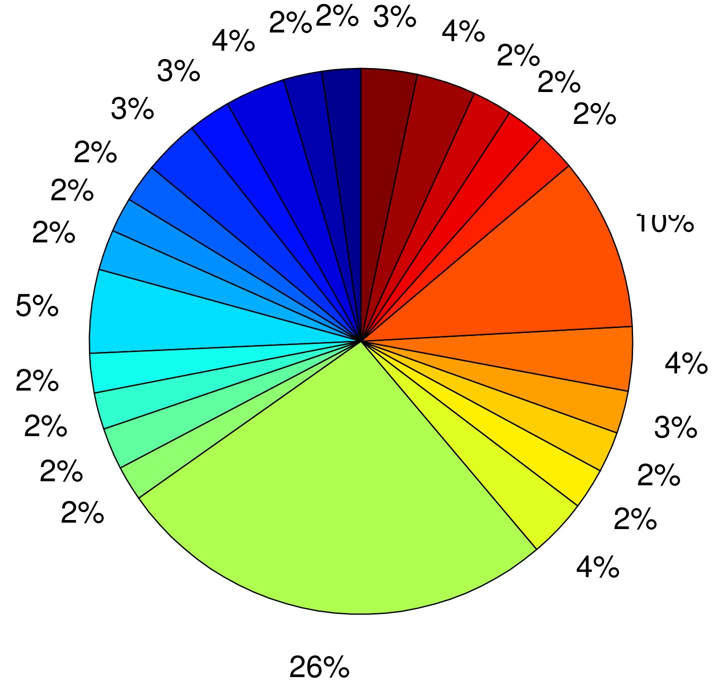
Example: 25 planets of uncertain mass pull in a unit mass space-ship

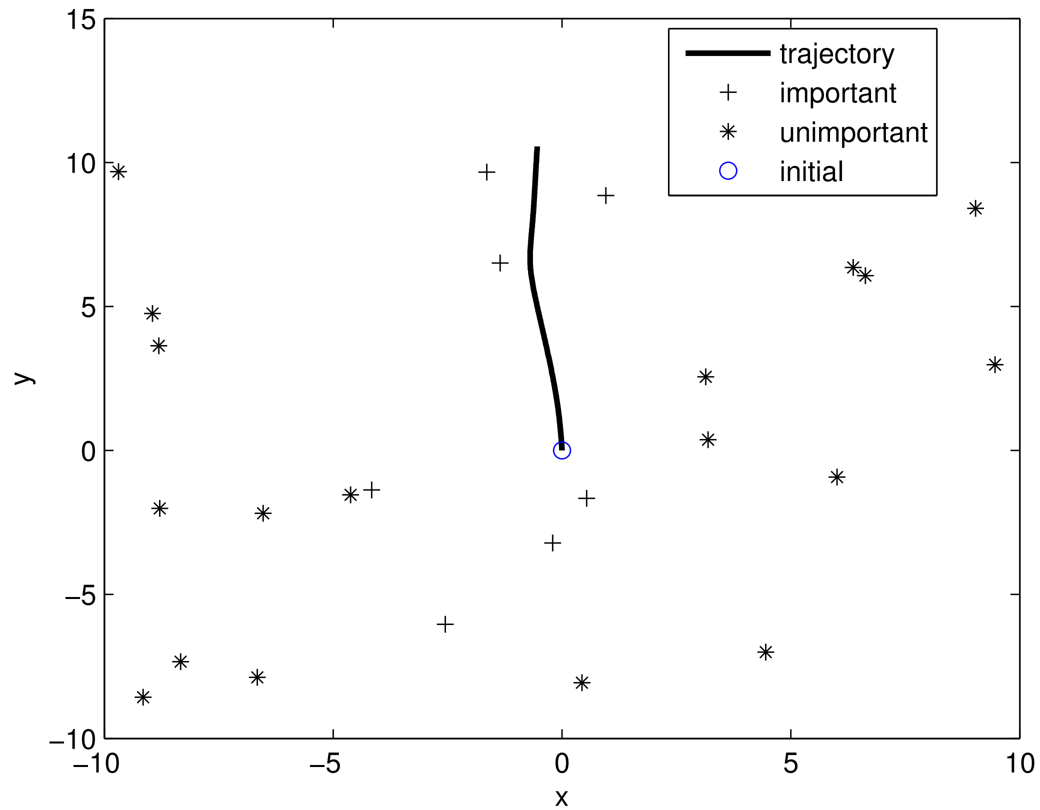
$$\ddot{\mathbf{x}}(t) = \sum_{i=1}^{p} m_i \hat{\mathbf{r}}_i / r_i^2, \qquad \mathbf{x}(\mathbf{t_0}) = \mathbf{x_0}. \qquad m_i = \frac{1}{p+1}[1 + 0.1 * U(-1,1)]$$

Full ANOVA based on Stroud-3



Sensitivity index





Active and passive "planets"

Active # of parameters is 7 >3%

Parameter compression for RBM

When extending this to PDEs and RBMs, key issue is

How to evaluate sensitivity at small cost ?

With the ability to build RB models, the following approach appears interesting

- Build a very coarse RBM over all parameters.
- Use RBM to build crude response surface
- Explore this very coarse model to estimate sensitivities
- Compress and develop RBM for important parameters

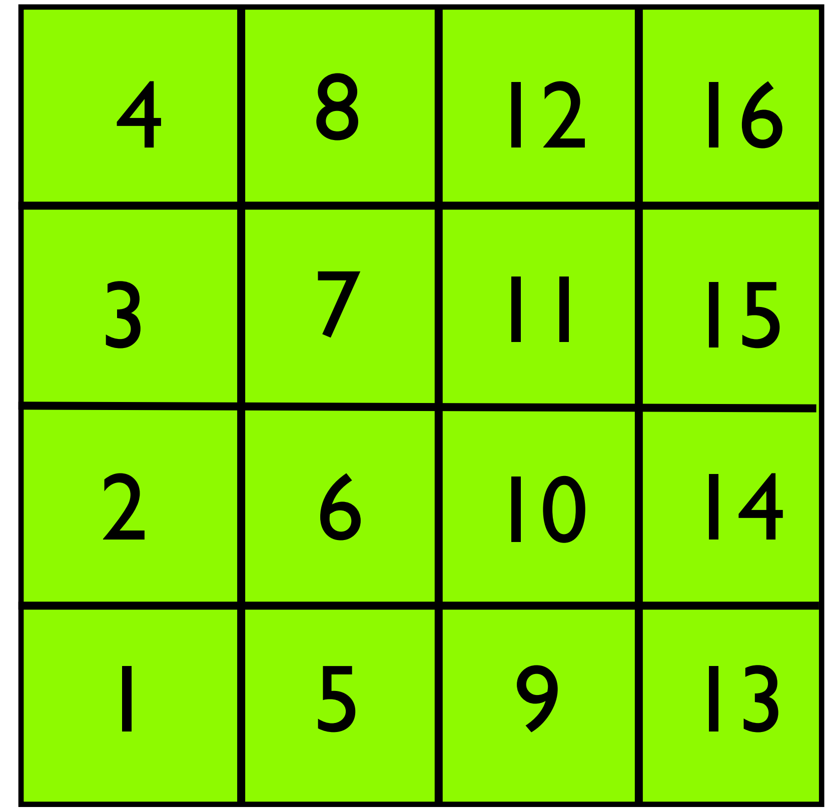
Heat equation test

Basic setup -

$$u = 0$$

$$-\nabla \cdot (\alpha \nabla u) = 0 \quad \text{in} \quad \Omega,$$

$$\alpha \nabla u \cdot \mathbf{n} = 0$$



$$\alpha \nabla u \cdot \mathbf{n} = 1$$

Piecewise constant material

$$\alpha = \begin{cases} \alpha_k = 100^{2\mu_k - 1}, \\ \alpha_k = 1.1^{2\mu_k - 1}, \end{cases}$$

$$\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_{16}) \in [0, 1]^{16}$$

Output of interest - $$s(\boldsymbol{\mu}) = \int_{\Gamma_{base}} u(\boldsymbol{\mu}) ds,$$

Heat equation test

We proceed with the following

- Build a coarse RBM over 16 parameters - tol = one
  Need 33 elements
- Perform sensitivity analysis on output using RBM
  Reveals the 1,5,9,13 controls 99% variation
- Build new RBM over 4 parameters
- Compress and develop RBM for important parameters

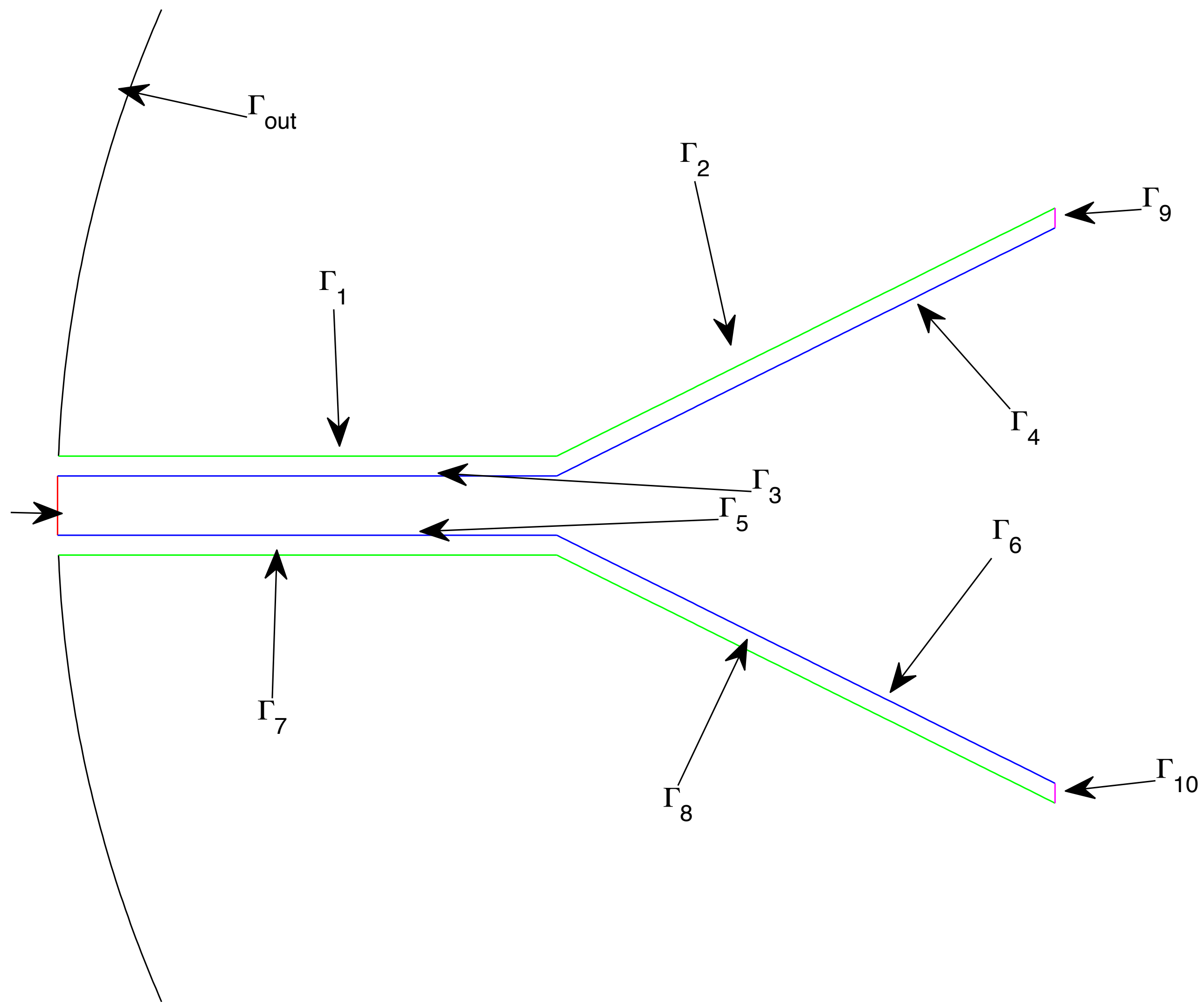
$$e_i^{rel} = \frac{|\bar{s}_i^{fe} - \bar{s}_i^{rb}|}{|\bar{s}_i^{fe}|}, \quad i = 1, \ldots, 100.$$

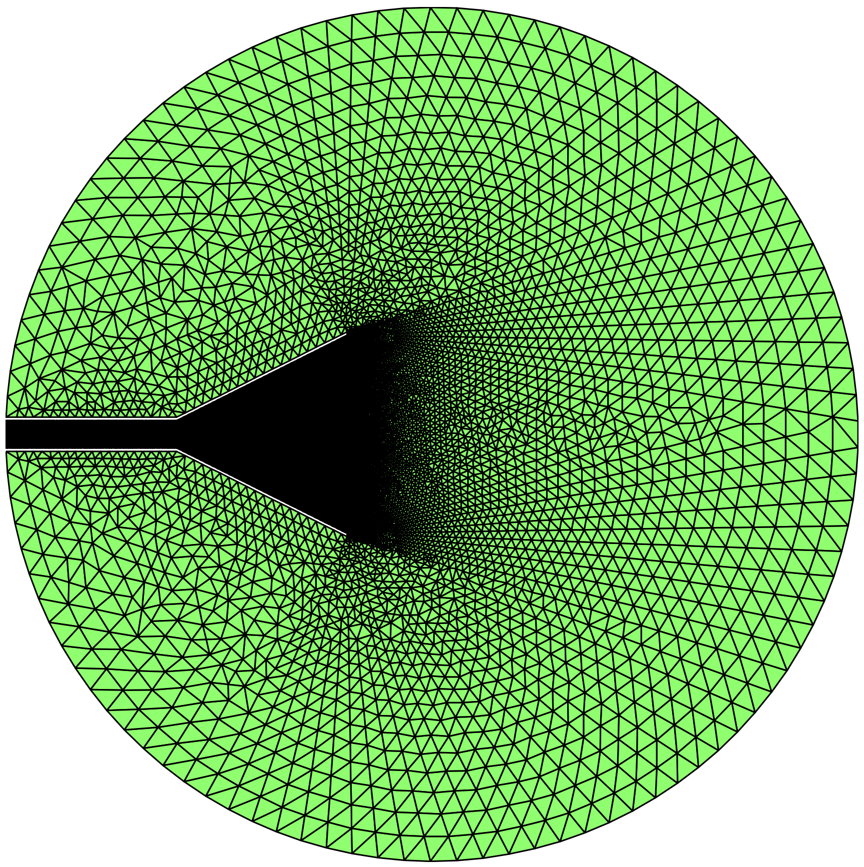
16-d test space

tol	Number of RB	e_{max}	e_{ave}
100	16	5.091×10^{-2}	7.732×10^{-3}
10	21	3.912×10^{-2}	7.177×10^{-3}
1	24	3.900×10^{-2}	7.192×10^{-3}
10^{-1}	30	3.893×10^{-2}	7.190×10^{-3}
10^{-2}	32	3.892×10^{-2}	7.190×10^{-3}

Acoustic horn test

We consider a similar approach for the acoustic horn





8 parameters, describing wall impedance in horn

Combining RB and ANOVA

The model is

$$\begin{cases} \Delta u + 4u & = & 0 & \text{in} & \Omega, \\ (2i + \frac{1}{25})u + \frac{\partial u}{\partial n} & = & 0 & \text{on} & \Gamma_{out}, \\ 2iu + \frac{\partial u}{\partial n} & = & 4i & \text{on} & \Gamma_{in}, \\ i\mu_j u + \frac{\partial u}{\partial n} & = & 0 & \text{on} & \Gamma_j, j = 1, \cdots, 8, \\ \frac{\partial u}{\partial n} & = & 0 & \text{on} & \text{other boundaries.} \end{cases}$$

Total of 8 parameters - boundary impedance

$$\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_8) \in [0, 1]^8.$$

Functional output -

$$s(\boldsymbol{\mu}) = \ell(u) = \text{real}(\int_{\Gamma_{in}} uds).$$

Combining RB and ANOVA

The approach is as follows

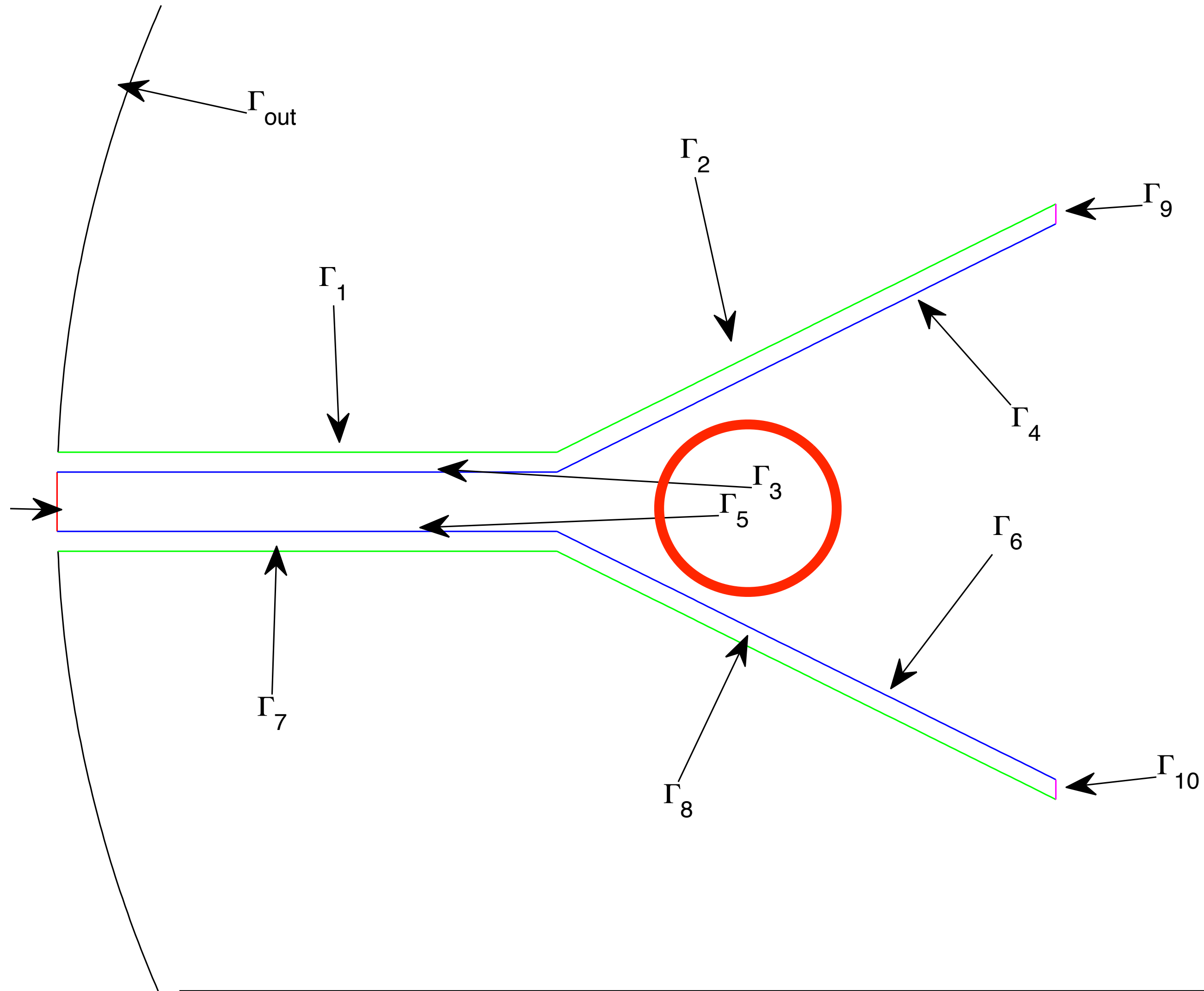
- Build coarse RB with high tolerance
- Tolerance of 10E-3 leads to 31 RB for 8 parameter problem.
- Use this coarse RB to compute ANOVA expansion of output and compute sensitivity.
- Results are

$$S_3 = 0.4321,\ S_5 = 0.4314, \quad S_{35} = 0.1256$$

$$S_3 + S_5 + S_{35} = 0.9891$$

- Similar results with tolerance of 1E-2 - 22 RB

Combining RB and ANOVA

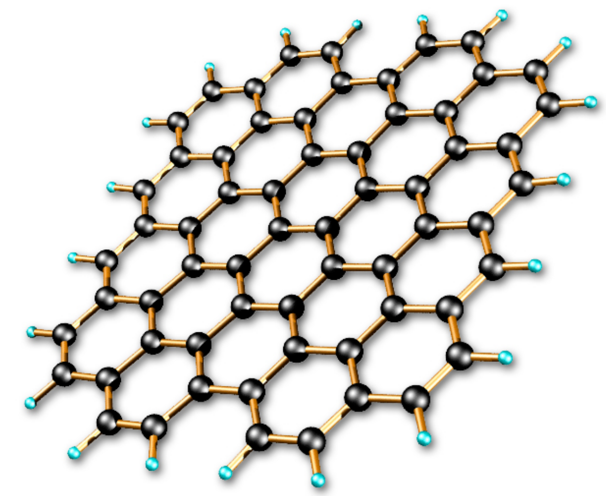


Two boundaries are responsible for >99% of all variation parameters.

tol	Number of RB	e_{max}	e_{ave}
10^{-2}	6	1.172×10^{-2}	2.404×10^{-3}
10^{-3}	11	1.214×10^{-2}	1.516×10^{-3}
10^{-4}	15	1.1213×10^{-2}	1.516×10^{-3}
10^{-5}	17	1.1213×10^{-2}	1.516×10^{-3}

Many object problems

Exploring related ideas for many body scattering



- Build an RB for each element
- Build an RB for the interaction operation
- Combine through iteration to enable rapid modeling of complex scatterer configurations

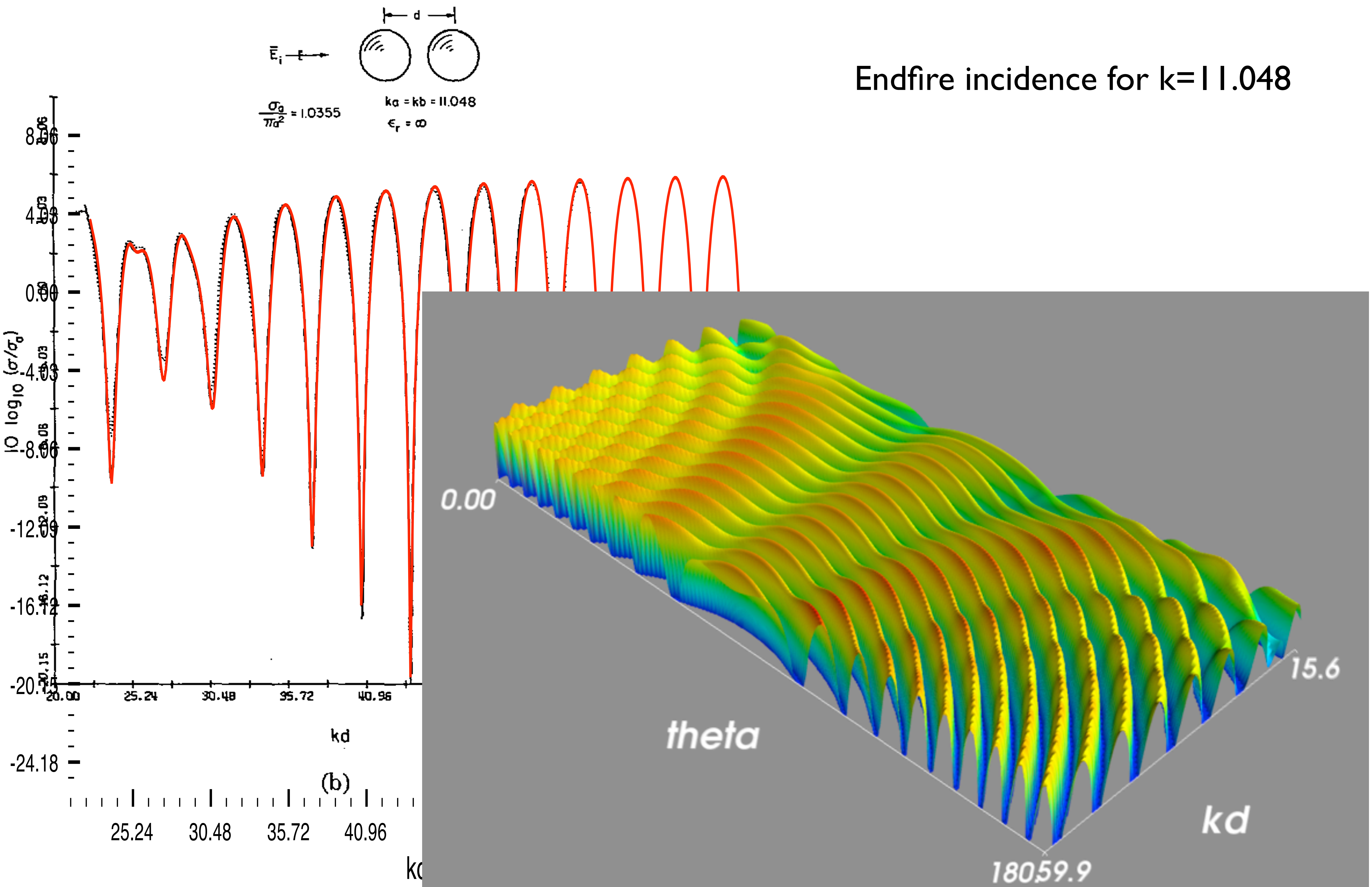
This is not a reduced basis method is the classic sense

.. but using RB ideas allows us to solve problems that are otherwise very hard to approach

.. and dramatically reduces number of parameters

Towards multiple scattering

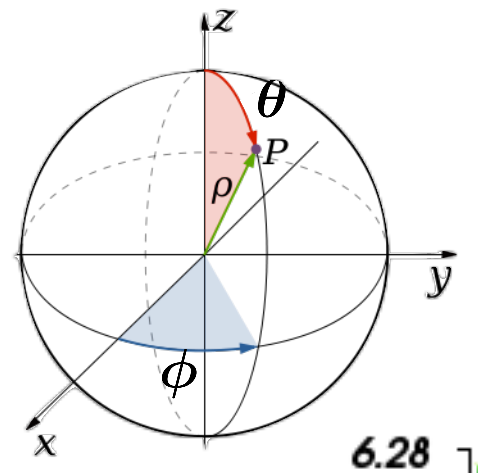
Endfire incidence for k=11.048

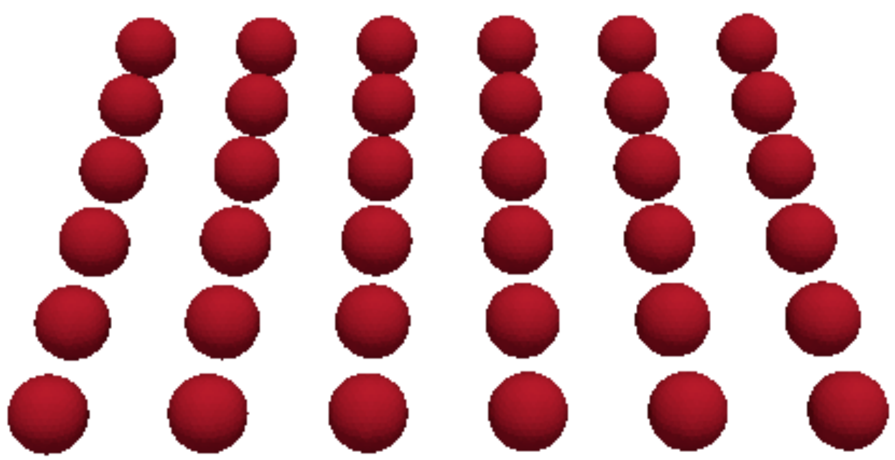


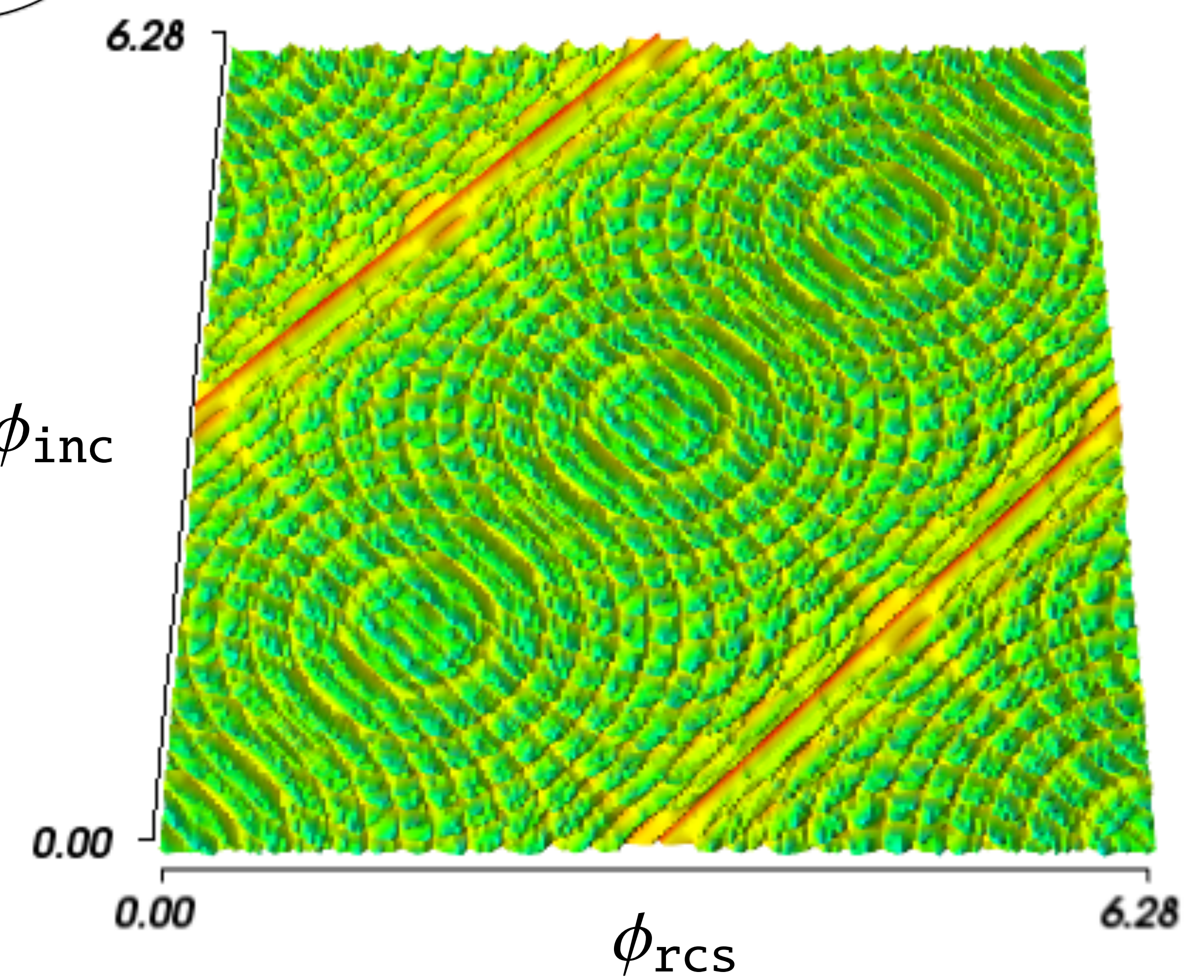
Multiple scattering problems

$$\phi \in [0, 2\pi]; k = 3, \theta = \pi/2$$

$$ka = 1; kd = 4$$







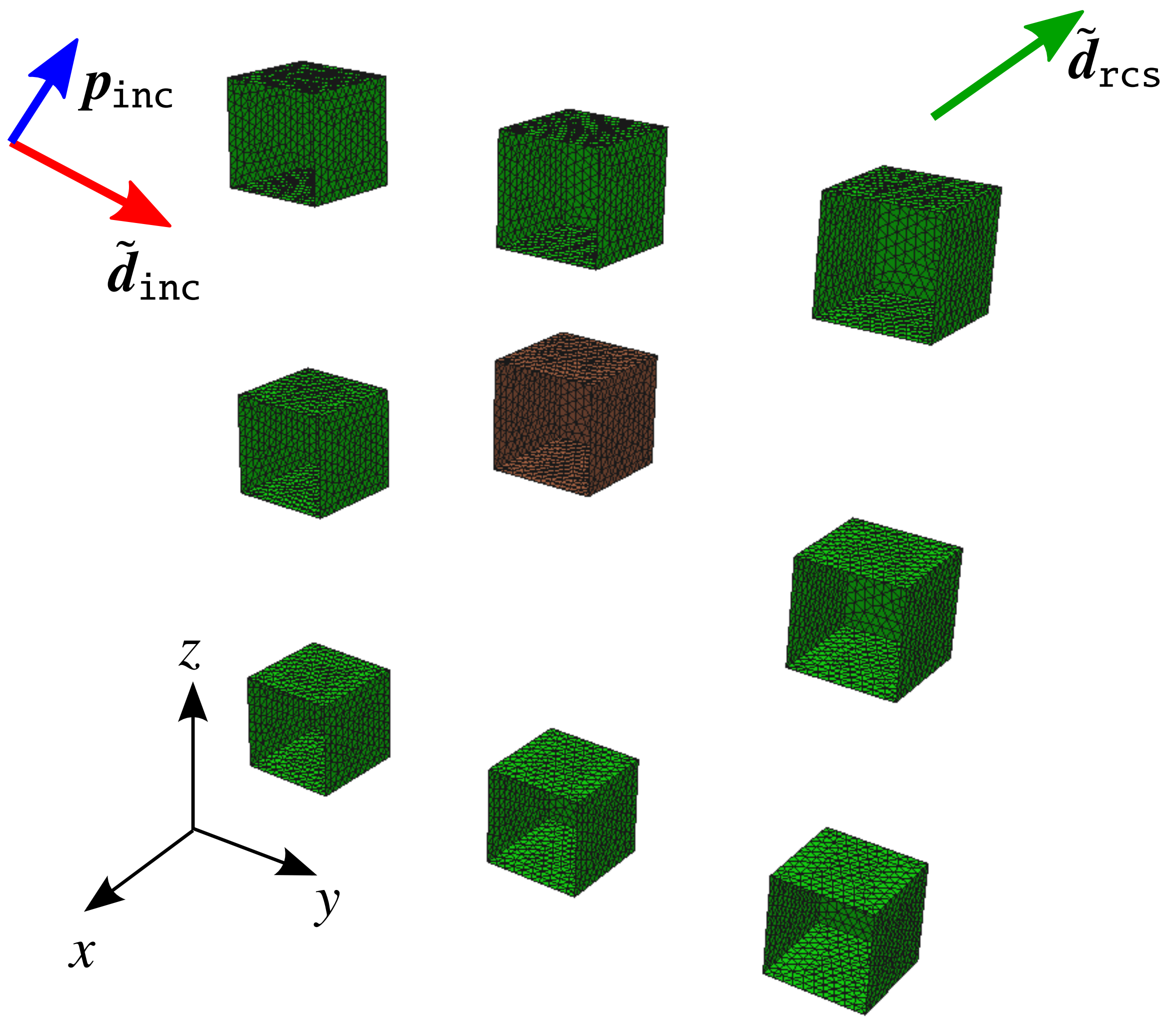
RB for single scatterer has 5 parameters (frequency(1), angle (2), polarization (2))

RB for interaction operator has 8 parameters (frequency(1), relative size(1), distance (2), rotation (2), polarization (2))

Full scattering result computed with iteration

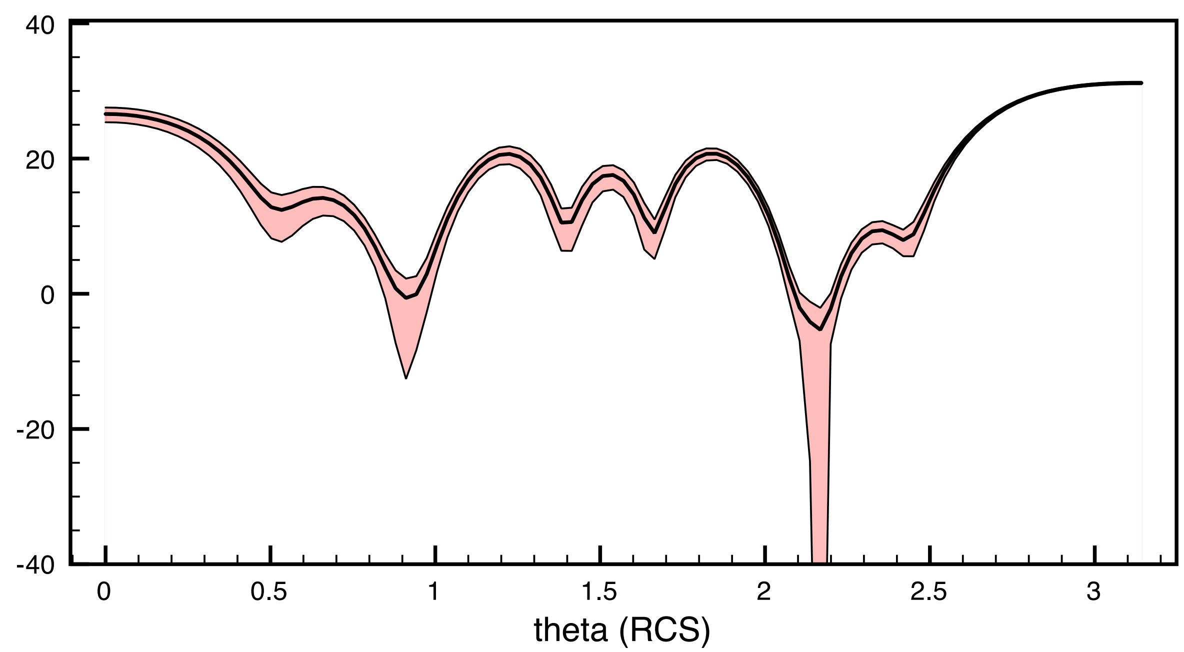
Full RCS computed in less than 3 minutes for 36 spheres

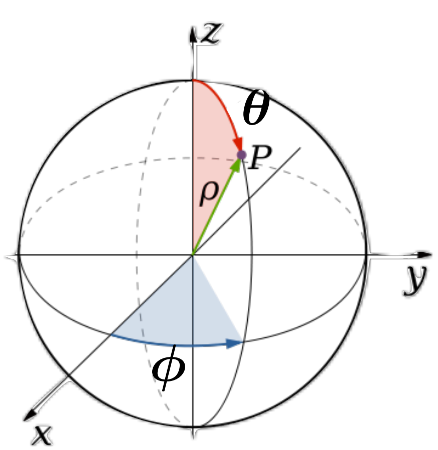
Multiple scattering problem



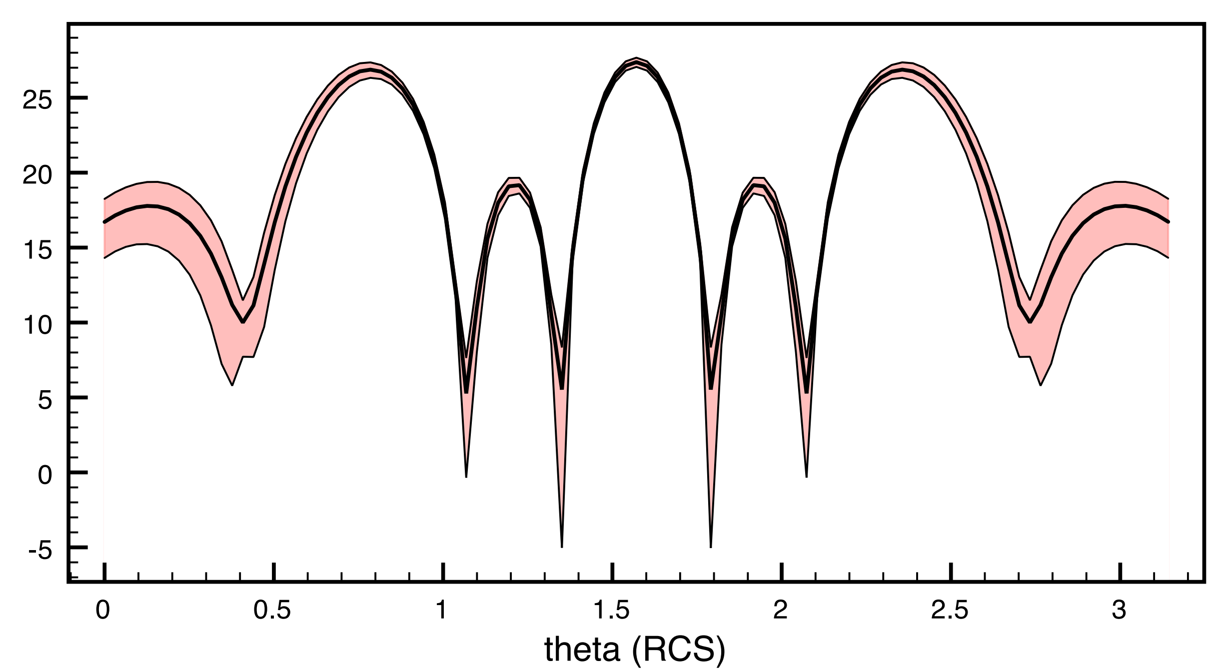
$$k = 3, \phi^i = 0, \theta^i = 0, 90$$

$$\phi^o = 0, \theta^o = 0 - 180$$





Vertical position of middle cavity uniformly distributed within [-1,1]



Remarks

A few ideas on how to deal with the high-d problem

- Multi element EIM for improved online performance
- Sampling techniques to reduce offline cost
- Parametric compression through ANOVA expansions
- Problem splitting and iteration

Combining these techniques allows for the practical use of RBM for high-dimensional problems


AS YOU CAN CLEARLY SEE IN SLIDE 397...
GAAAAH!
"POWERPOINT" POISONING.
www.dilbert.com
scottadams@aol.com
© 2000 United Feature Syndicate, Inc.
8/16/00

Thank you