Reduced Basis Method for the parametrized Electric Field Integral Equation (EFIE)

Matrix Computations and Scientific Computing Seminar

University of California, Berkeley, 15.09.2010

Benjamin Stamm

Department of Mathematics and LBL

University of California, Berkeley

Project description

- In collaboration with
  - Prof. J. Hesthaven and Prof. Y. Maday (Brown U. and Paris VI)
  - Prof. R. Nochetto (U. of Maryland)
  - Dr. M'Barek Fares (CERFACS, Toulouse)
  - J. Eftang (MIT), Prof. A. Patera (MIT), Prof. M. Grepl (RWTH Aachen)
  - Prof. M. Ganesh (Colorado School of Mines)

- From mathematics to applied computations with many practical applications

- Sponsored by SNF grant PBELP2 - 123078, Brown University, UC Berkeley

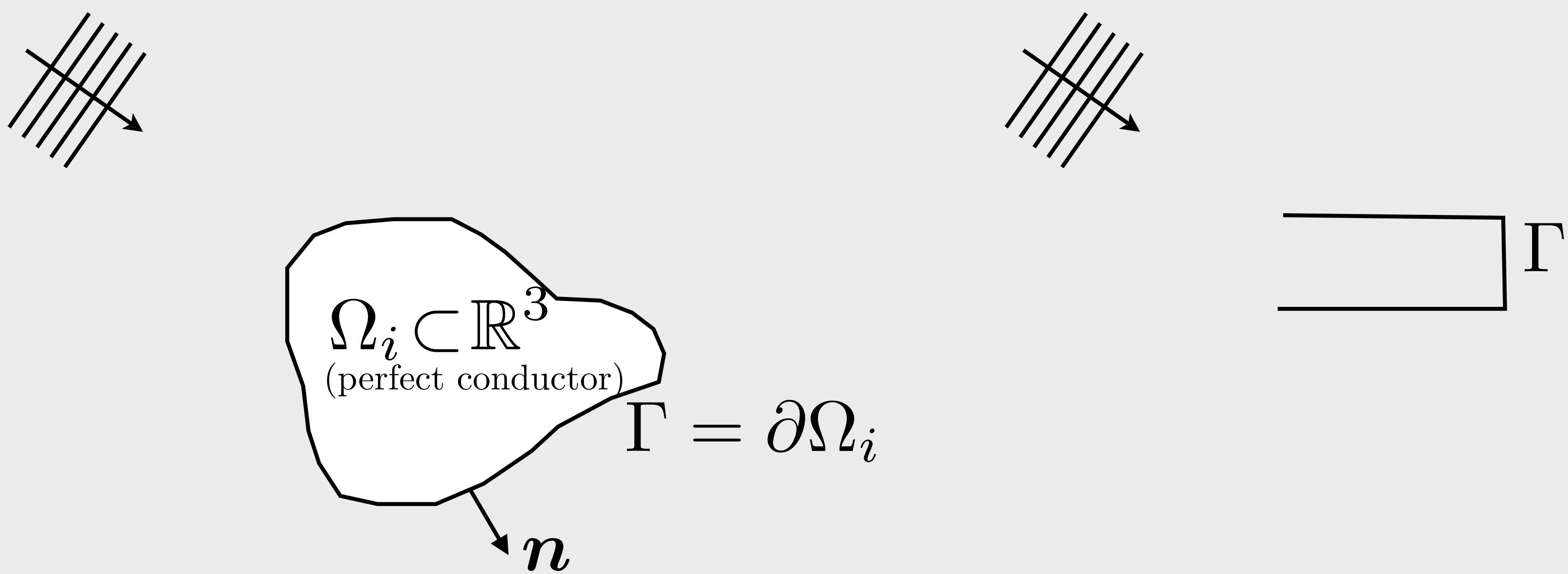
Outline

- Introduction to parametrized scattering problems
- The Reduced Basis Method
- The Empirical Interpolation Method
- Numerical results
- First results on multi-object scattering
- Conclusions

Introduction to parametrized Electromagnetic scattering

Parametrized Electromagnetic Scattering
(time-harmonic ansatz)

$$\boldsymbol{E}^i(\boldsymbol{x};\boldsymbol{\mu}) = -\boldsymbol{p}\, e^{ik\boldsymbol{x}\cdot\hat{\boldsymbol{k}}(\theta,\phi)} \qquad \Omega = \mathbb{R}^3\backslash\overline{\Omega_i}$$

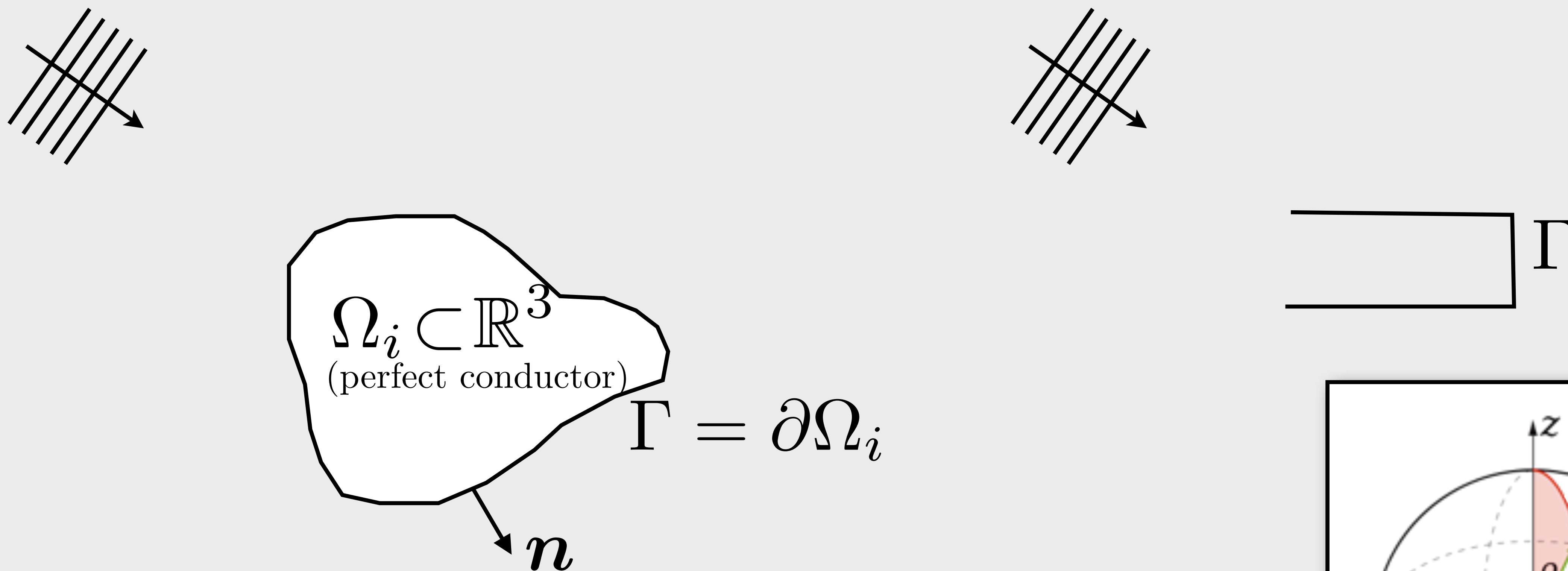


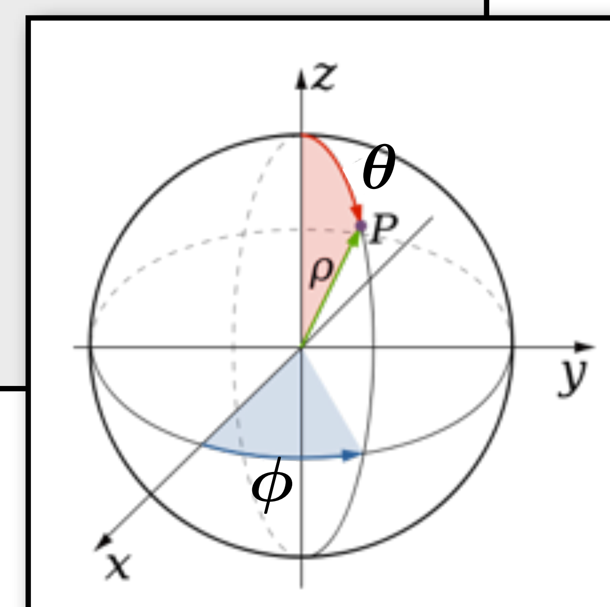
where $\boldsymbol{\mu} = (k, \theta, \phi, \boldsymbol{p}) \in \mathcal{D} \subset \mathbb{R}^7$ is a vector of parameters:
1) k: wave number
2) $\hat{\boldsymbol{k}}(\theta, \phi)$: wave direction in spherical coordinates
3) $\boldsymbol{p}$: polarization (is complex and lies in the plane perpendicular to $\hat{\boldsymbol{k}}(\theta, \phi)$)

Parametrized Electromagnetic Scattering

(time-harmonic ansatz)

$$\boldsymbol{E}^i(\boldsymbol{x};\boldsymbol{\mu}) = -\boldsymbol{p}\, e^{ik\boldsymbol{x}\cdot\hat{\boldsymbol{k}}_{(\theta,\phi)}} \qquad \Omega = \mathbb{R}^3 \backslash \overline{\Omega_i}$$





where $\boldsymbol{\mu} = (k, \theta, \phi, \boldsymbol{p}) \in \mathcal{D} \subset \mathbb{R}^7$ is a vector of parameters:

1) k: wave number

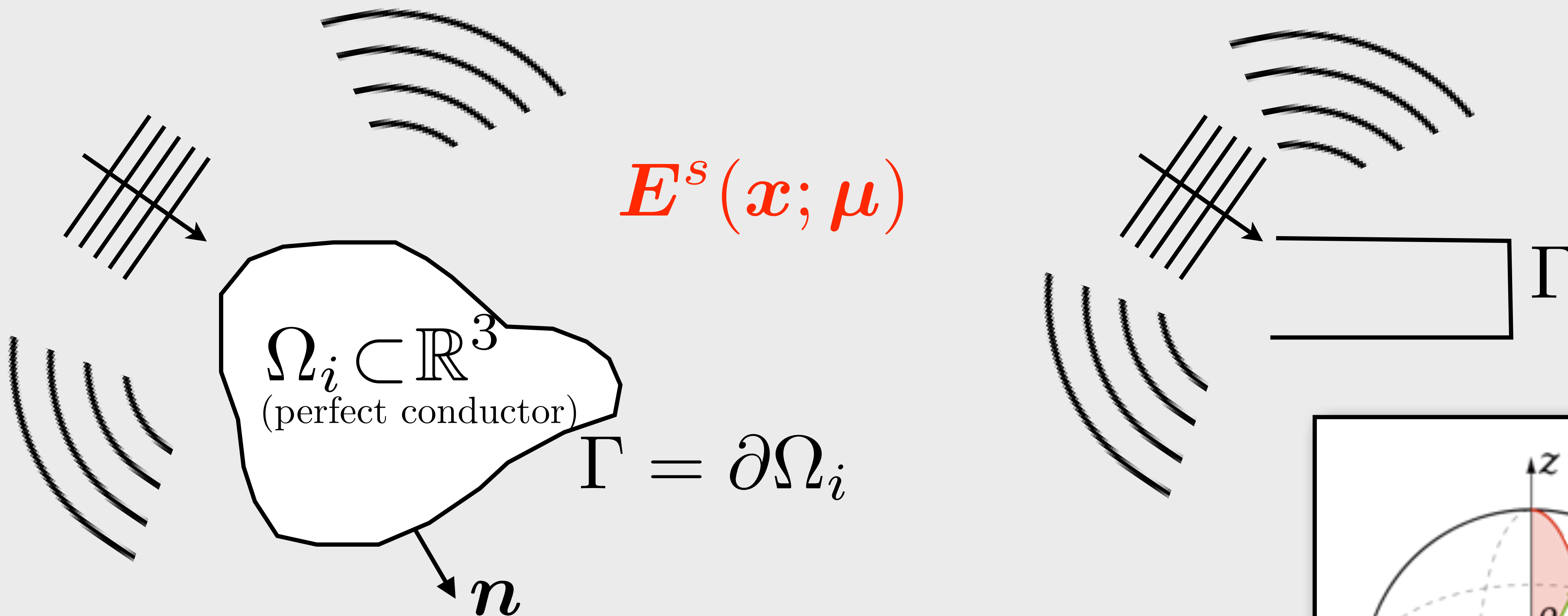
2) $\hat{\boldsymbol{k}}(\theta, \phi)$: wave direction in spherical coordinates

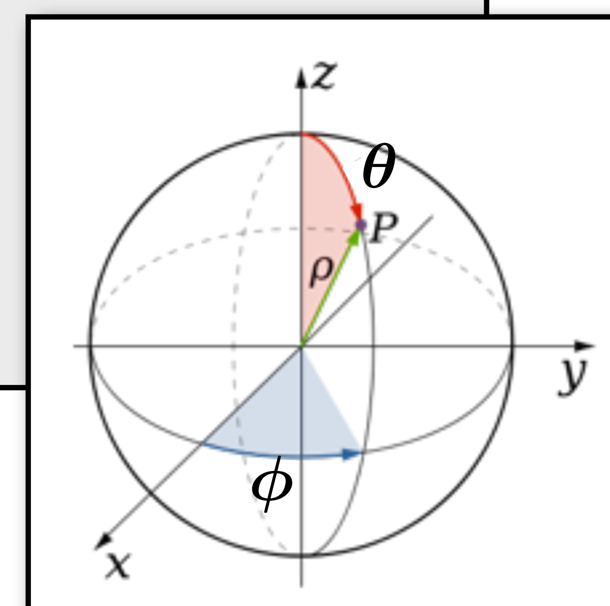
3) $\boldsymbol{p}$: polarization (is complex and lies in the plane perpendicular to $\hat{\boldsymbol{k}}(\theta, \phi)$)

Parametrized Electromagnetic Scattering

(time-harmonic ansatz)

$$\boldsymbol{E}^i(\boldsymbol{x};\boldsymbol{\mu}) = -\boldsymbol{p}\, e^{ik\boldsymbol{x}\cdot\hat{\boldsymbol{k}}(\theta,\phi)} \qquad \Omega = \mathbb{R}^3\backslash\overline{\Omega_i}$$





where $\boldsymbol{\mu} = (k, \theta, \phi, \boldsymbol{p}) \in \mathcal{D} \subset \mathbb{R}^7$ is a vector of parameters:

1) k: wave number

2) $\hat{\boldsymbol{k}}(\theta, \phi)$: wave direction in spherical coordinates

3) $\boldsymbol{p}$: polarization (is complex and lies in the plane perpendicular to $\hat{\boldsymbol{k}}(\theta, \phi)$)

Governing equations
(not parametrized for sake of simplicity)

Assume that Ω is a homogenous media with magnetic permeability μ and electrical permittivity ε.

Then, the electric field $\mathbf{E} = \mathbf{E}^i + \mathbf{E}^s \in \boldsymbol{H}(\mathrm{curl}, \Omega)$ satisfies

$$
\begin{aligned}
\mathrm{curl}\ \mathrm{curl}\ \mathbf{E} - k^2\mathbf{E} &= 0 && \text{in } \Omega, && \boxed{\text{Maxwell}} \\
\mathbf{E} \times \boldsymbol{n} &= 0 && \text{on } \Gamma, && \boxed{\text{boundary condition}} \\
\left| \mathrm{curl}\mathbf{E}^s(\boldsymbol{x}) \times \frac{\boldsymbol{x}}{|\boldsymbol{x}|} - ik\mathbf{E}^s(\boldsymbol{x}) \right| &= \mathcal{O}\left(\frac{1}{|\boldsymbol{x}|}\right) && \text{as } |\boldsymbol{x}| \to \infty. && \boxed{\text{Silver-Müller radiation condition}}
\end{aligned}
$$

Boundary condition is equivalent to $\gamma_t \mathbf{E} = 0$ where γ_t denotes the tangential trace operator on surface Γ, $\gamma_t \mathbf{E} = \boldsymbol{n} \times (\mathbf{E} \times \boldsymbol{n})$.

Integral representation

Stratton-Chu representation formula:

$$\mathbf{E}(\boldsymbol{x}) = \mathbf{E}^i(\boldsymbol{x}) + ikZ\mathbf{T}_k\boldsymbol{u}(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \Omega$$

where

$\mathbf{T}_k$: single layer potential

$\boldsymbol{u}$: electrical current on surface

$Z = \sqrt{\mu/\varepsilon}$: impedance $\quad$ ($k = \omega\sqrt{\mu\varepsilon}$: wave number)

Applying the tangential trace operator and invoking the boundary conditions yield the **strong form** of the **Electric Field Integral Equation (EFIE)**: Find $\boldsymbol{u} \in \mathbb{V}$ s.t.

$$ikZ\gamma_t(\mathbf{T}_k\boldsymbol{u})(\boldsymbol{x}) = -\gamma_t\mathbf{E}^i(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \Gamma$$

for some appropriate *complex* functional space $\mathbb{V}$ on Γ.

Integral representation

Stratton-Chu representation formula:

$$\mathbf{E}(\boldsymbol{x}) = \mathbf{E}^i(\boldsymbol{x}) + ikZ\mathbf{T}_k\boldsymbol{u}(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \Omega$$

where

$\mathbf{T}_k$: single layer potential

$\boldsymbol{u}$: electrical current on surface

$Z = \sqrt{\mu/\varepsilon}$: impedance $\quad$ ($k = \omega\sqrt{\mu\varepsilon}$: wave number)

Applying the tangential trace operator and invoking the boundary conditions yield the **strong form** of the **Electric Field Integral Equation (EFIE)**: Find $\boldsymbol{u} \in \mathbb{V}$ s.t.

$$ikZ\gamma_t(\mathbf{T}_k\boldsymbol{u})(\boldsymbol{x}) = -\gamma_t\mathbf{E}^i(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \Gamma$$

for some appropriate *complex* functional space $\mathbb{V}$ on Γ.

Model reduction: 3d ⇨ 2d problem

Variational formulation of the EFIE
(also called the Rumsey principle)

Multiplying by a test function $\boldsymbol{v} \in \mathbb{V}$ and taking the scalar product yields: Find $\boldsymbol{u} \in \mathbb{V}$ s.t.

$$\boxed{ikZ\langle \gamma_t(\mathbf{T}_k \boldsymbol{u}), \boldsymbol{v}\rangle_\Gamma = -\langle \gamma_t \mathbf{E}^i(\boldsymbol{x}), \boldsymbol{v}\rangle_\Gamma, \qquad \forall \boldsymbol{v} \in \mathbb{V}.}$$

Variational formulation of the EFIE

(also called the Rumsey principle)

Multiplying by a test function $\boldsymbol{v} \in \mathbb{V}$ and taking the scalar product yields: Find $\boldsymbol{u} \in \mathbb{V}$ s.t.

$$\boxed{ikZ\langle\gamma_t(\mathbf{T}_k\boldsymbol{u}),\boldsymbol{v}\rangle_\Gamma = -\langle\gamma_t\mathbf{E}^i(\boldsymbol{x}),\boldsymbol{v}\rangle_\Gamma, \qquad \forall\boldsymbol{v}\in\mathbb{V}.}$$

After integration by parts and introducing the parameter dependence we get: for any fixed $\boldsymbol{\mu} \in \mathcal{D}$, find $\boldsymbol{u}(\boldsymbol{\mu}) \in \mathbb{V}$ s.t.

$$\boxed{a(\boldsymbol{u}(\boldsymbol{\mu}),\boldsymbol{v};\boldsymbol{\mu}) = f(\boldsymbol{v};\boldsymbol{\mu}), \qquad \forall\boldsymbol{v}\in\mathbb{V}}$$

with

$$a(\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) = ikZ\int_\Gamma\int_\Gamma G_k(\boldsymbol{x},\boldsymbol{y})\left\{\boldsymbol{u}(\boldsymbol{x})\cdot\overline{\boldsymbol{v}(\boldsymbol{y})} - \tfrac{1}{k^2}\mathrm{div}_{\Gamma,\boldsymbol{x}}\boldsymbol{u}(\boldsymbol{x})\overline{\mathrm{div}_{\Gamma,\boldsymbol{y}}\boldsymbol{v}(\boldsymbol{y})}\right\}d\boldsymbol{x}\,d\boldsymbol{y}$$

$$f(\boldsymbol{v};\boldsymbol{\mu}) = -\int_\Gamma\gamma_t\mathbf{E}^i(\boldsymbol{x};\boldsymbol{\mu})\cdot\overline{\boldsymbol{v}(\boldsymbol{x})}\,d\boldsymbol{x}$$

Variational formulation of the EFIE

(also called the Rumsey principle)

Multiplying by a test function $\boldsymbol{v} \in \mathbb{V}$ and taking the scalar product yields: Find $\boldsymbol{u} \in \mathbb{V}$ s.t.

$$ikZ\langle \gamma_t(\mathbf{T}_k\boldsymbol{u}), \boldsymbol{v}\rangle_\Gamma = -\langle \gamma_t \mathbf{E}^i(\boldsymbol{x}), \boldsymbol{v}\rangle_\Gamma, \qquad \forall \boldsymbol{v} \in \mathbb{V}.$$

After integration by parts and introducing the parameter dependence we get: for any fixed $\boldsymbol{\mu} \in \mathcal{D}$, find $\boldsymbol{u}(\boldsymbol{\mu}) \in \mathbb{V}$ s.t.

$$a(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{v}; \boldsymbol{\mu}) = f(\boldsymbol{v}; \boldsymbol{\mu}), \qquad \forall \boldsymbol{v} \in \mathbb{V}$$

with

$$a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu}) = ikZ \int_\Gamma \int_\Gamma G_k(\boldsymbol{x}, \boldsymbol{y}) \left\{ \boldsymbol{u}(\boldsymbol{x}) \cdot \overline{\boldsymbol{v}(\boldsymbol{y})} - \tfrac{1}{k^2} \mathrm{div}_{\Gamma,\boldsymbol{x}} \boldsymbol{u}(\boldsymbol{x}) \overline{\mathrm{div}_{\Gamma,\boldsymbol{y}} \boldsymbol{v}(\boldsymbol{y})} \right\} d\boldsymbol{x}\, d\boldsymbol{y}$$

$$f(\boldsymbol{v}; \boldsymbol{\mu}) = -\int_\Gamma \gamma_t \mathbf{E}^i(\boldsymbol{x}; \boldsymbol{\mu}) \cdot \overline{\boldsymbol{v}(\boldsymbol{x})}\, d\boldsymbol{x}$$

1) Sesquilinear form $a(\cdot, \cdot; \boldsymbol{\mu})$ is symmetric but not coercive
2) $G_k(\boldsymbol{x}, \boldsymbol{y}) = \frac{e^{ik|\boldsymbol{x}-\boldsymbol{y}|}}{|\boldsymbol{x}-\boldsymbol{y}|}$ is the fundamental solution of the Helmholtz operator $\Delta + k^2$ and depends on the paramter k.

Variational formulation of the EFIE

(also called the Rumsey principle)

Multiplying by a test function $\boldsymbol{v} \in \mathbb{V}$ and taking the scalar product yields: Find $\boldsymbol{u} \in \mathbb{V}$ s.t.

$$\boxed{ikZ\langle \gamma_t(\mathbf{T}_k \boldsymbol{u}), \boldsymbol{v}\rangle_\Gamma = -\langle \gamma_t \mathbf{E}^i(\boldsymbol{x}), \boldsymbol{v}\rangle_\Gamma, \qquad \forall \boldsymbol{v} \in \mathbb{V}.}$$

After integration by parts and introducing the parameter dependence we get: for any fixed $\boldsymbol{\mu} \in \mathcal{D}$, find $\boldsymbol{u}(\boldsymbol{\mu}) \in \mathbb{V}$ s.t.

$$\boxed{a(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{v}; \boldsymbol{\mu}) = f(\boldsymbol{v}; \boldsymbol{\mu}), \qquad \forall \boldsymbol{v} \in \mathbb{V}}$$

with

1) Sesquilinear form $a(\cdot, \cdot; \boldsymbol{\mu})$ is symmetric but not coercive
2) $G_k(\boldsymbol{x}, \boldsymbol{y}) = \frac{e^{ik|\boldsymbol{x}-\boldsymbol{y}|}}{|\boldsymbol{x}-\boldsymbol{y}|}$ is the fundamental solution of the Helmholtz operator $\Delta + k^2$ and depends on the paramter k.

[Colton, Kress 1992],[Nédélec 2001]

Parametrized EFIE and its discretization

Galerkin approach: replace continuous space $\mathbb{V}$ by the finite dimensional subspace $\mathbb{V}_h$: For any fixed parameter $\boldsymbol{\mu} \in \mathcal{D}$, find $\boldsymbol{u}_h(\boldsymbol{\mu}) \in \mathbb{V}_h$ such that

$$a(\boldsymbol{u}_h(\boldsymbol{\mu}), \boldsymbol{v}_h; \boldsymbol{\mu}) = f(\boldsymbol{v}_h; \boldsymbol{\mu}) \qquad \forall \boldsymbol{v}_h \in \mathbb{V}_h. \tag{1}$$

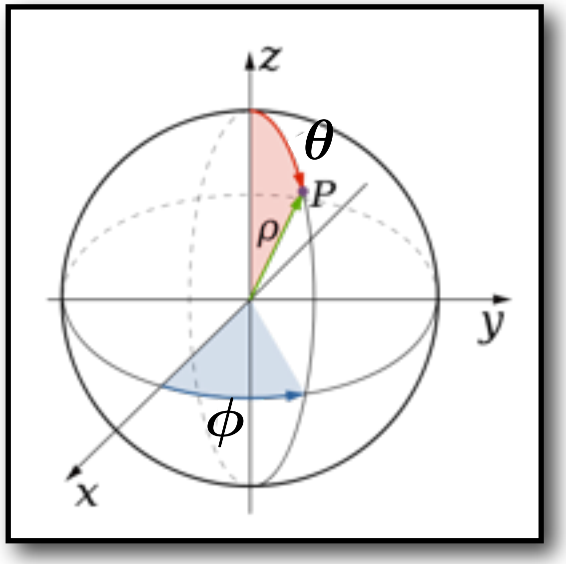
For $\mathbb{V}_h$ we use the lowest order (complex) Raviart-Thomas space $\boldsymbol{RT}_0$, also called Rao-Wilton-Glisson (RWG) basis in the electromagnetic community.

Boundary Element Method (BEM). In practice the code CESC is used, CESC: **C**ERFACS **E**lectromagnetic **S**olver **C**ode.

[Bendali 1984],[Schwab, Hiptmair 2002],[Buffa et al. 2002,2003], [Christiansen 2004]

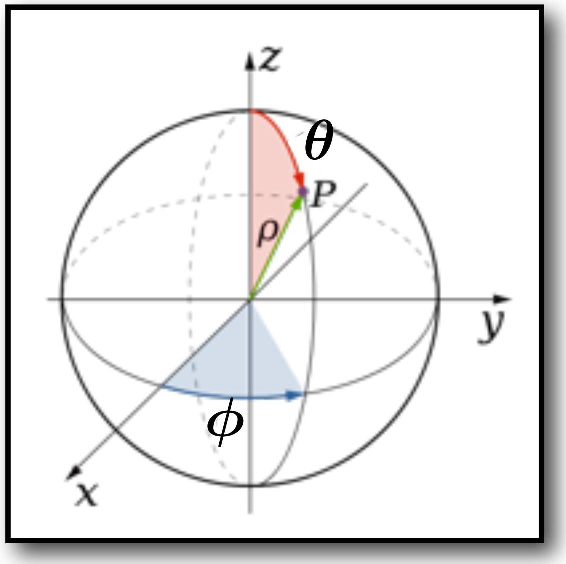
Example of parametrized solution

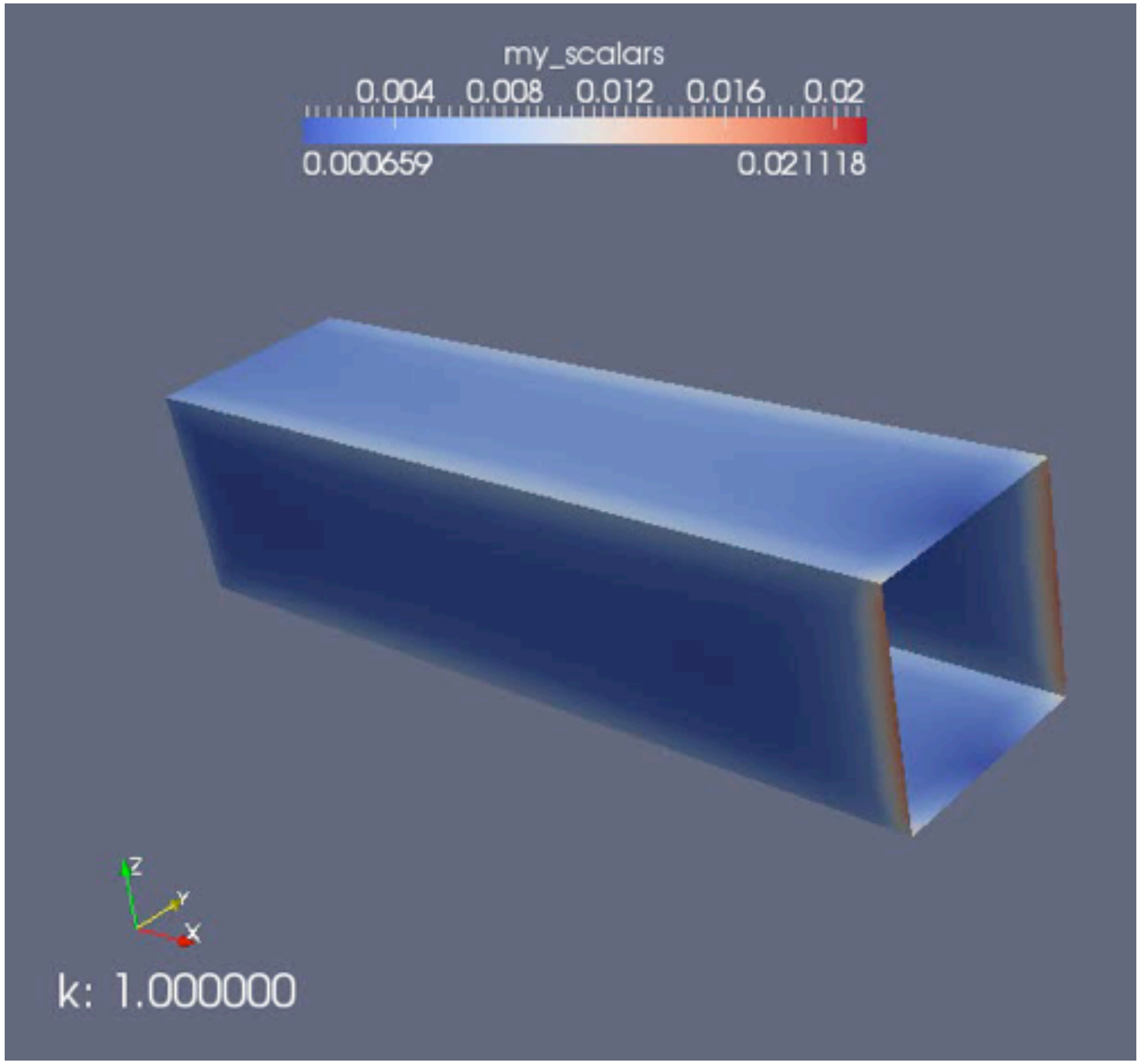
Incident plane wave parametrized by $\boldsymbol{E}^i(\boldsymbol{x};k) = -\boldsymbol{p}\, e^{ik\boldsymbol{x}\cdot\hat{\boldsymbol{k}}_{(\frac{\pi}{4},0)}}$.



Example of parametrized solution

Incident plane wave parametrized by $\boldsymbol{E}^i(\boldsymbol{x};k) = -\boldsymbol{p}\, e^{ik\boldsymbol{x}\cdot\hat{\boldsymbol{k}}_{(\frac{\pi}{4},0)}}$.





Output functional: Radar Cross Section (RCS)

- Describes pattern/energy of electrical field at infinity
- Functional of the current on body

$$A_\infty(\boldsymbol{u}, \hat{\boldsymbol{d}}) = \frac{ikZ}{4\pi} \int_\Gamma \hat{\boldsymbol{d}} \times (\boldsymbol{u}(\boldsymbol{x}) \times \hat{\boldsymbol{d}}) e^{-ik\boldsymbol{x}\cdot\hat{\boldsymbol{d}}} d\boldsymbol{x}$$

$$\mathrm{RCS}(\boldsymbol{u}, \hat{\boldsymbol{d}}) = 10 \log_{10} \left(\frac{|A_\infty(\boldsymbol{u}, \hat{\boldsymbol{d}})|^2}{|A_\infty(\boldsymbol{u}, \hat{\boldsymbol{d}}_0)|^2} \right)$$

where

$\boldsymbol{u}$: current on surface

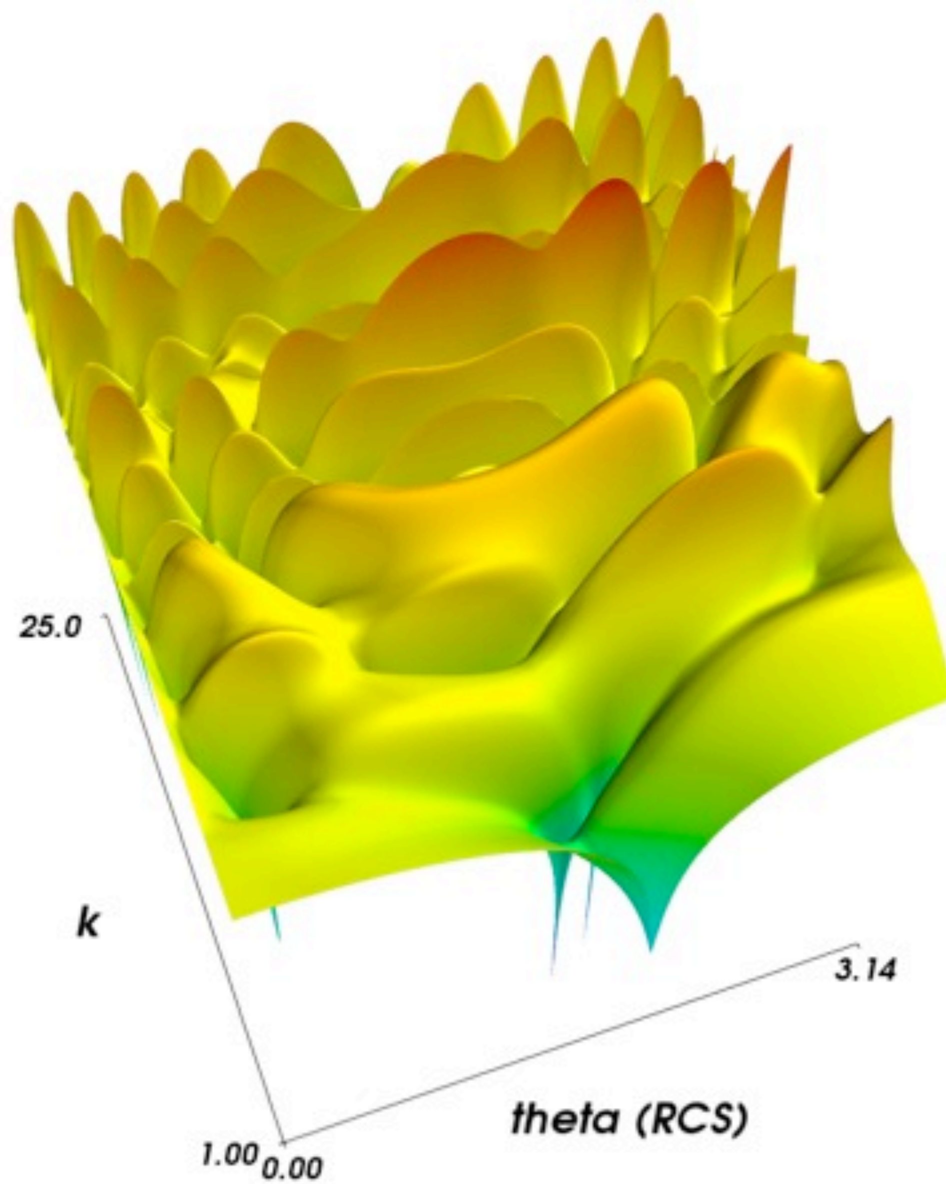
$\hat{\boldsymbol{d}}$: given directional unit vector

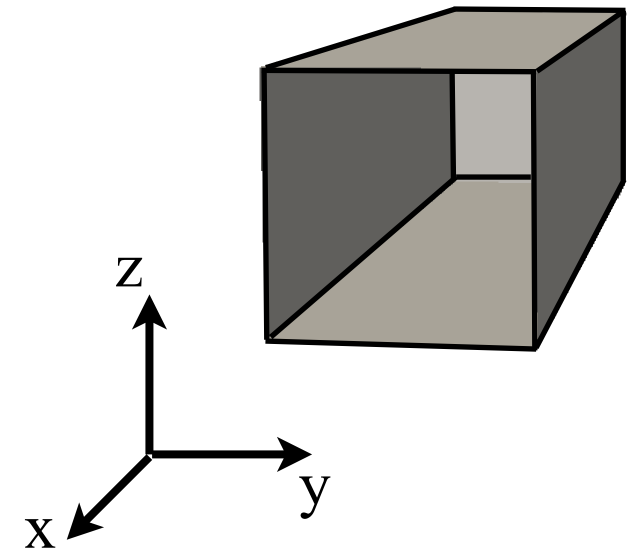
$\hat{\boldsymbol{d}}_0$: reference unit direction

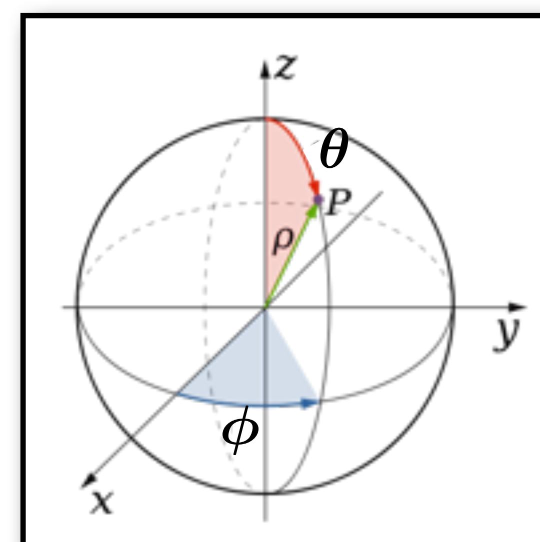
Output functional: Radar Cross Section (RCS)

Incident plane wave parametrized by $\boldsymbol{E}^i(\boldsymbol{x};k) = -\boldsymbol{p}\, e^{ik\boldsymbol{x}\cdot\hat{\boldsymbol{k}}_{(\frac{\pi}{4},0)}}$.

Directional unit vector given by $\hat{\boldsymbol{d}} = \hat{\boldsymbol{d}}_{(\theta,\phi)}$ with $\theta \in [0,\pi]$, $\phi = 0$.







Reduced basis method

Reduced basis method: Overview

Assume that we want to compute the scattered field for many different values of the parameters:

- Applying the BEM many times is too expensive and unnecessary since the parametrized solutions lie often on a low order manifold.

On a discrete level, assume that:

Assumption (Existence of "ideal" reduced basis):
The subspace $\mathcal{M}_h := \{u_h(\boldsymbol{\mu}) \,|\, \forall \boldsymbol{\mu} \in \mathbb{D}\}$, is of low dimensionality, i.e.

$$\mathcal{M}_h \overset{Tol}{=} \mathrm{span}\{\boldsymbol{\zeta}_i \,|\, i = 1, \ldots, N\}$$

up to a certain given tolerance Tol for some properly chosen $\{\boldsymbol{\zeta}_i\}_{i=1}^N$ and moderate $N \ll \mathcal{N} = \dim(\mathbb{V}_h))$. More precisely, we assume an exponentially decreasing Tol in function of N.

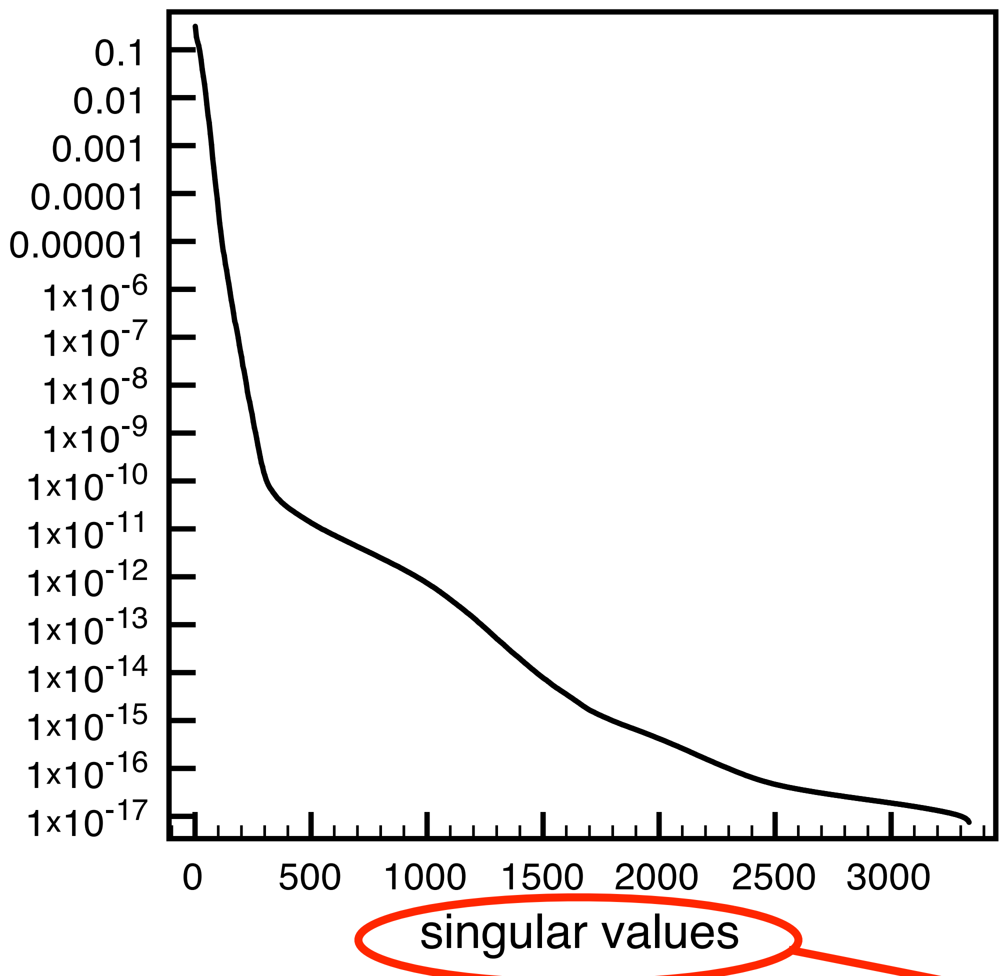
- The reduced basis method is a tool to construct an approximation $\{\boldsymbol{\xi}_i\}_{i=1}^N$ of the "ideal" reduced basis.

Example: Existence of an "ideal" reduced basis

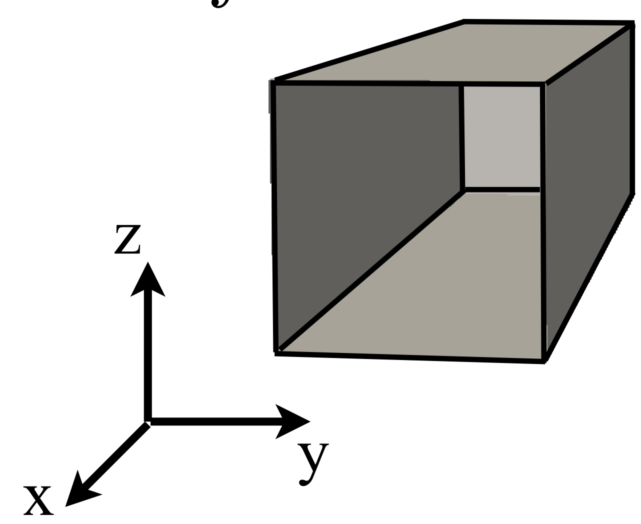
POD: Proper Orthogonal Decomposition

Parameters: $(k, \theta) \in [1, 25] \times [0, \pi]$, ϕ is fixed.

For a fine discretization of $[1, 25] \times [0, \pi]$, compute the BEM-solution for each parameter value. Save all solutions in a matrix and compute the singular values.



Geometry:



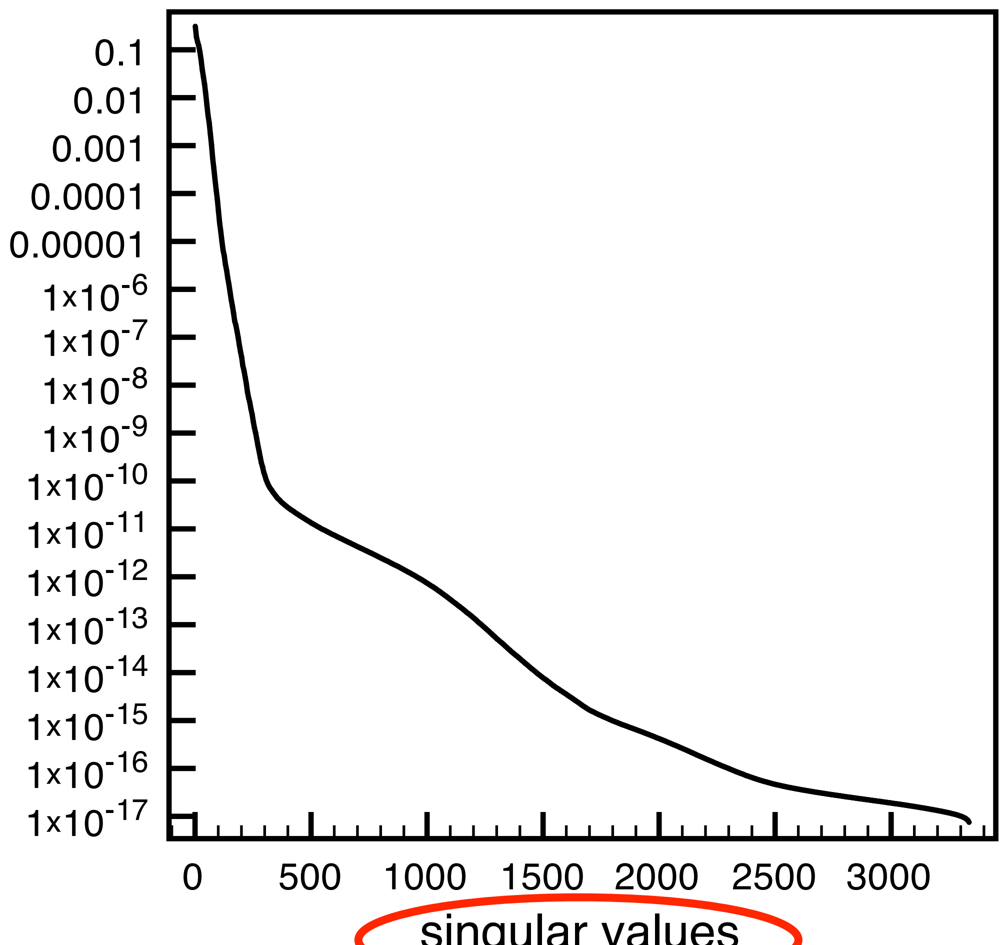
Indication of linear dependence of solutions

Example: Existence of an "ideal" reduced basis

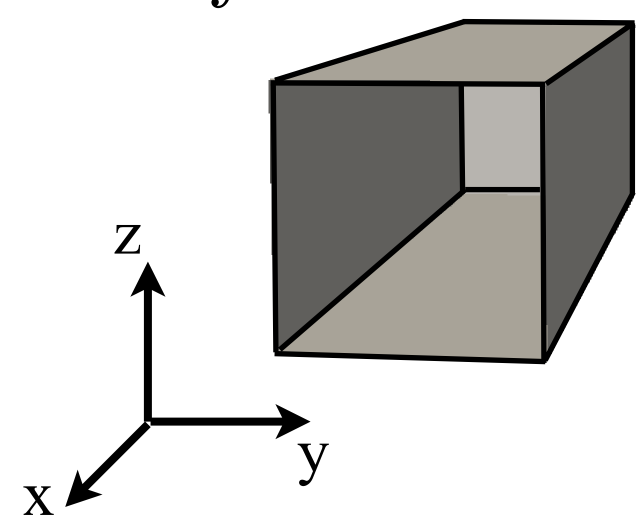
POD: Proper Orthogonal Decomposition

Parameters: $(k, \theta) \in [1, 25] \times [0, \pi]$, ϕ is fixed.

For a fine discretization of $[1, 25] \times [0, \pi]$, compute the BEM-solution for each parameter value. Save all solutions in a matrix and compute the singular values.



Geometry:



With 200 basis functions you can reach a precision of 1e-7!

Indication of linear dependence of solutions

Reduced basis method - Interpolation between snapshots

In practice we use $\{\boldsymbol{\xi}_i\}_{i=1}^N$ as reduced basis where
1) $\boldsymbol{\xi}_i = \boldsymbol{u}_h(\boldsymbol{\mu}_i)$ are solutions of (1) with $\boldsymbol{\mu} = \boldsymbol{\mu}_i \in \mathcal{D}$ (snapshots),
2) $\mathbb{S}_N = \{\boldsymbol{\mu}_i\}_{i=1}^N$ carefully chosen.

$\Rightarrow$ **Requires only N BEM computations.**

The reduced basis approximation is the solution of: For $\boldsymbol{\mu} \in \mathcal{D}$, find $\boldsymbol{u}_N(\boldsymbol{\mu}) \in \mathbb{W}_N$ such that:

$$a(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N; \boldsymbol{\mu}) = f(\boldsymbol{v}_N; \boldsymbol{\mu}) \qquad \forall \boldsymbol{v}_N \in \mathbb{W}_N \tag{2}$$

with $\mathbb{W}_N = \mathrm{span}\{\boldsymbol{\xi}_i \,|\, i = 1, \ldots, N\}$.

$\Rightarrow$ **Parameter dependent Ritz-type projection onto reduced basis.**

Questions:
1) **Accuracy**: How to choose $\mathbb{S}_N$?
2) **Efficiency**: How to solve (2) in a fast way?

Reduced basis method - Interpolation between snapshots

In practice we use $\{\boldsymbol{\xi}_i\}_{i=1}^N$ as reduced basis where
1) $\boldsymbol{\xi}_i = \boldsymbol{u}_h(\boldsymbol{\mu}_i)$ are solutions of (1) with $\boldsymbol{\mu} = \boldsymbol{\mu}_i \in \mathcal{D}$ (snapshots),
2) $\mathbb{S}_N = \{\boldsymbol{\mu}_i\}_{i=1}^N$ carefully chosen.
$\Rightarrow$ **Requires only N BEM computations.**

The reduced basis approximation is the solution of: For $\boldsymbol{\mu} \in \mathcal{D}$, find $\boldsymbol{u}_N(\boldsymbol{\mu}) \in \mathbb{W}_N$ such that:

$$a(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N; \boldsymbol{\mu}) = f(\boldsymbol{v}_N; \boldsymbol{\mu}) \qquad \forall \boldsymbol{v}_N \in \mathbb{W}_N \tag{2}$$

with $\mathbb{W}_N = \operatorname{span}\{\boldsymbol{\xi}_i \,|\, i = 1, \ldots, N\}$.

$\Rightarrow$ **Parameter dependent Ritz-type projection onto reduced basis.**

Questions:
1) **Accuracy**: How to choose $\mathbb{S}_N$?
2) **Efficiency**: How to solve (2) in a fast way?

See [Rozza et al. 2008] for a review.

Reduced basis method - Overall strategy

Step 1: Construct an approximation $\mathbb{W}_N \subset \mathbb{V}_h$ (reduced basis) to the solution space

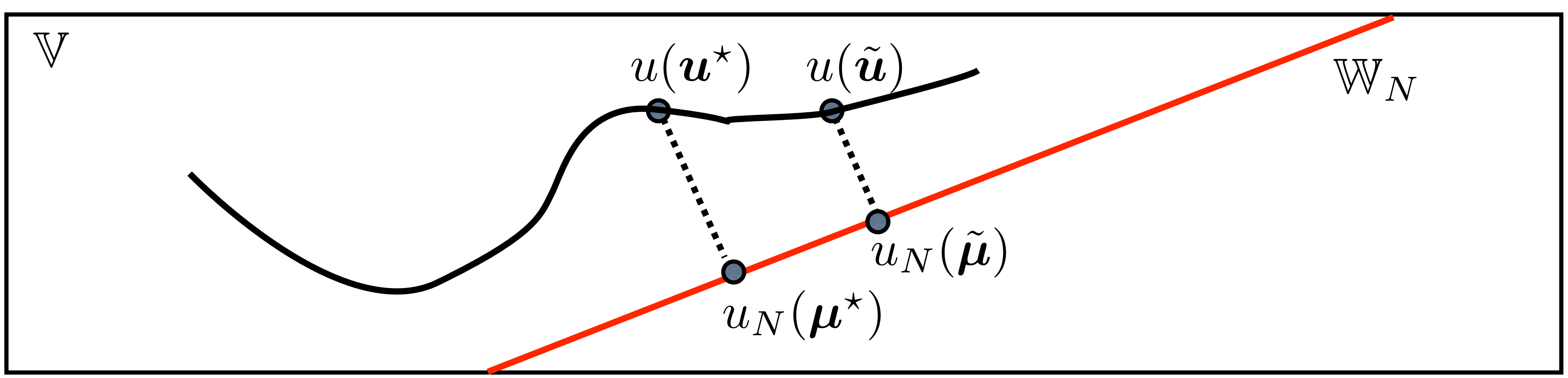
$$\mathbb{W}_N \approx \text{span}\{\mathcal{M}_h\} \quad \text{with} \quad \mathcal{M}_h = \{u_h(\boldsymbol{\mu}) \,|\, \boldsymbol{\mu} \in \mathcal{D}\}.$$

Step 2: Project the exact solution $u(\boldsymbol{\mu})$ onto the reduced basis using a parameter dependent Ritz-projection:

$$P_N(\boldsymbol{\mu}) : \mathbb{V} \to \mathbb{W}_N.$$

In other words: find $u_N(\boldsymbol{\mu}) \in \mathbb{W}_N$ such that

$$\boxed{a(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = f(v_N; \boldsymbol{\mu}), \qquad \forall v_N \in \mathbb{W}_N.}$$



Accuracy: Choice of reduced basis (Greedy algorithm)

Offline Initialization:

Choose initial parameter value $\boldsymbol{\mu}_1 \in \Xi$ and set $\mathbb{S}_1 = \{\boldsymbol{\mu}_1\}$, put $\mathbb{W}_0 = \emptyset$

For $N = 1, \ldots, N_{max}$

$\Xi \subset \mathcal{D}$ is a finite dimensional pointset of $\mathcal{D}$.

Offline Loop:

1) Compute the truth solution $\boldsymbol{u}_h(\boldsymbol{\mu}_N)$, solution of (1), with $\boldsymbol{\mu} = \boldsymbol{\mu}_N$

2) $\mathbb{W}_N = \mathbb{W}_{N-1} \cup \{\boldsymbol{u}_h(\boldsymbol{\mu}_N)\}$

3) For all $\boldsymbol{\mu} \in \Xi$, do:

i) Compute $\boldsymbol{u}_N(\boldsymbol{\mu}) \in \mathbb{W}_N$ solution of (2)

ii) Compute a posteriori error estimation $\eta(\boldsymbol{\mu}) \approx \|\boldsymbol{u}_h(\boldsymbol{\mu}) - \boldsymbol{u}_N(\boldsymbol{\mu})\|$

4) Choose $\boldsymbol{\mu}_{N+1} = \arg\max_{\boldsymbol{\mu} \in \Xi} \eta(\boldsymbol{\mu})$

5) $\mathbb{S}_{N+1} = \mathbb{S}_N \cup \{\boldsymbol{\mu}_{N+1}\}$

Online Loop:

Accuracy: Choice of reduced basis (Greedy algorithm)

Offline Initialization:

Choose initial parameter value $\boldsymbol{\mu}_1 \in \Xi$ and set $\mathbb{S}_1 = \{\boldsymbol{\mu}_1\}$, put $\mathbb{W}_0 = \emptyset$

For $N = 1, \ldots, N_{max}$

$\Xi \subset \mathcal{D}$ is a finite dimensional pointset of $\mathcal{D}$.

Offline Loop:

1) Compute the truth solution $\boldsymbol{u}_h(\boldsymbol{\mu}_N)$, solution of (1), with $\boldsymbol{\mu} = \boldsymbol{\mu}_N$
2) $\mathbb{W}_N = \mathbb{W}_{N-1} \cup \{\boldsymbol{u}_h(\boldsymbol{\mu}_N)\}$
3) For all $\boldsymbol{\mu} \in \Xi$, do:
   - i) Compute $\boldsymbol{u}_N(\boldsymbol{\mu}) \in \mathbb{W}_N$ solution of (2)
   - ii) Compute a posteriori error estimation $\eta(\boldsymbol{\mu}) \approx \|\boldsymbol{u}_h(\boldsymbol{\mu}) - \boldsymbol{u}_N(\boldsymbol{\mu})\|$
4) Choose $\boldsymbol{\mu}_{N+1} = \arg\max_{\boldsymbol{\mu} \in \Xi} \eta(\boldsymbol{\mu})$
5) $\mathbb{S}_{N+1} = \mathbb{S}_N \cup \{\boldsymbol{\mu}_{N+1}\}$

Online Loop:

Accuracy: Choice of reduced basis (Greedy algorithm)

Offline Initialization:
Choose initial parameter value $\boldsymbol{\mu}_1 \in \Xi$ and set $\mathbb{S}_1 = \{\boldsymbol{\mu}_1\}$, put $\mathbb{W}_0 = \emptyset$

For $N = 1, \ldots, N_{max}$

$\Xi \subset \mathcal{D}$ is a finite dimensional pointset of $\mathcal{D}$.

Offline Loop:

1) Compute the truth solution $\boldsymbol{u}_h(\boldsymbol{\mu}_N)$, solution of (1), with $\boldsymbol{\mu} = \boldsymbol{\mu}_N$
2) $\mathbb{W}_N = \mathbb{W}_{N-1} \cup \{\boldsymbol{u}_h(\boldsymbol{\mu}_N)\}$
3) For all $\boldsymbol{\mu} \in \Xi$, do:
   i) Compute $\boldsymbol{u}_N(\boldsymbol{\mu}) \in \mathbb{W}_N$ solution of (2)
   ii) Compute a posteriori error estimation $\eta(\boldsymbol{\mu}) \approx \|\boldsymbol{u}_h(\boldsymbol{\mu}) - \boldsymbol{u}_N(\boldsymbol{\mu})\|$
4) Choose $\boldsymbol{\mu}_{N+1} = \arg\max_{\boldsymbol{\mu} \in \Xi} \eta(\boldsymbol{\mu})$
5) $\mathbb{S}_{N+1} = \mathbb{S}_N \cup \{\boldsymbol{\mu}_{N+1}\}$

Online Loop:

1) For any new $\boldsymbol{\mu} \in \mathcal{D}$, compute $\boldsymbol{u}_N(\boldsymbol{\mu}) \in \mathbb{W}_{N_{max}}$ solution of (2)
2) Compute the output functional of $\mathrm{RCS}(\boldsymbol{u}_N(\boldsymbol{\mu}), \hat{\boldsymbol{d}})$

Efficiency: Affine assumption

Assumption:

$$a(\boldsymbol{w},\boldsymbol{v};\boldsymbol{\mu}) = \sum_{m=1}^{M} \Theta^m(\boldsymbol{\mu})\, a^m(\boldsymbol{w},\boldsymbol{v}),$$

$$f(\boldsymbol{v};\boldsymbol{\mu}) = \sum_{m=1}^{M} \Theta_f^m(\boldsymbol{\mu})\, f^m(\boldsymbol{v}),$$

where for $m = 1,\ldots,M$

$$\begin{aligned} \Theta^m, \Theta_f^m &: \mathcal{D} \to \mathbb{C} & \quad \boldsymbol{\mu} &- \text{dependent functions},\\ a^m &: \mathbb{V}_h \times \mathbb{V}_h \to \mathbb{C} & \quad \boldsymbol{\mu} &- \text{independent forms},\\ f^m &: \mathbb{V}_h \to \mathbb{C} & \quad \boldsymbol{\mu} &- \text{independent forms}, \end{aligned}$$

Efficiency: Affine assumption

Assumption:

$$a(\boldsymbol{w}, \boldsymbol{v}; \boldsymbol{\mu}) = \sum_{m=1}^{M} \Theta^m(\boldsymbol{\mu})\, a^m(\boldsymbol{w}, \boldsymbol{v}),$$

$$f(\boldsymbol{v}; \boldsymbol{\mu}) = \sum_{m=1}^{M} \Theta_f^m(\boldsymbol{\mu})\, f^m(\boldsymbol{v}),$$

where for $m = 1, \ldots, M$

$$\begin{aligned} \Theta^m, \Theta_f^m &: \mathcal{D} \to \mathbb{C} && \boldsymbol{\mu} - \text{dependent functions}, \\ a^m &: \mathbb{V}_h \times \mathbb{V}_h \to \mathbb{C} && \boldsymbol{\mu} - \text{independent forms}, \\ f^m &: \mathbb{V}_h \to \mathbb{C} && \boldsymbol{\mu} - \text{independent forms}, \end{aligned}$$

Caution: This is not feasible in the framework of the EFIE!

$$a(\boldsymbol{u}_h, \boldsymbol{v}_h; \boldsymbol{\mu}) = ikZ \int_\Gamma \int_\Gamma \frac{e^{ik|\boldsymbol{x}-\boldsymbol{y}|}}{|\boldsymbol{x}-\boldsymbol{y}|} \left\{ \boldsymbol{u}_h(\boldsymbol{x}) \cdot \overline{\boldsymbol{v}_h(\boldsymbol{y})} - \tfrac{1}{k^2} \mathrm{div}_{\Gamma,\boldsymbol{x}} \boldsymbol{u}_h(\boldsymbol{x}) \cdot \overline{\mathrm{div}_{\Gamma,\boldsymbol{y}} \boldsymbol{v}_h(\boldsymbol{y})} \right\} d\boldsymbol{x}\, d\boldsymbol{y}$$

$$f(\boldsymbol{v}_h; \boldsymbol{\mu}) = \boldsymbol{n} \times (\boldsymbol{p} \times \boldsymbol{n}) \int_\Gamma e^{ik\boldsymbol{x} \cdot \hat{\boldsymbol{s}}_{(\theta,\phi)}} \cdot \overline{\boldsymbol{v}_h(\boldsymbol{x})}\, d\boldsymbol{x}$$

Efficiency: Affine assumption

Assumption:

$$a(\boldsymbol{w}, \boldsymbol{v}; \boldsymbol{\mu}) = \sum_{m=1}^{M} \Theta^m(\boldsymbol{\mu})\, a^m(\boldsymbol{w}, \boldsymbol{v}),$$

$$f(\boldsymbol{v}; \boldsymbol{\mu}) = \sum_{m=1}^{M} \Theta_f^m(\boldsymbol{\mu})\, f^m(\boldsymbol{v}),$$

where for $m = 1, \ldots, M$

$$\begin{aligned} \Theta^m, \Theta_f^m &: \mathcal{D} \to \mathbb{C} && \boldsymbol{\mu} - \text{dependent functions}, \\ a^m &: \mathbb{V}_h \times \mathbb{V}_h \to \mathbb{C} && \boldsymbol{\mu} - \text{independent forms}, \\ f^m &: \mathbb{V}_h \to \mathbb{C} && \boldsymbol{\mu} - \text{independent forms}, \end{aligned}$$

Caution: This is not feasible in the framework of the EFIE!

Luckily this problem can be fixed (later in this talk), assume for now that the assumption holds approximatively

Efficiency: How to solve (2) in a fast way?

Offline:

Given $\mathbb{W}_N = \mathrm{span}\{\boldsymbol{\xi}_i \,|\, i = 1, \dots, N\}$ precompute

$$(A^m)_{i,j} = a^m(\boldsymbol{\xi}_j, \boldsymbol{\xi}_i), \qquad \forall 1 \le i, j \le N,$$
$$(F^m)_i = f^m(\boldsymbol{\xi}_i), \qquad \forall 1 \le i \le N.$$

Rem. Depends on $\mathcal{N} = \dim(\mathbb{V}_h)$.
Rem. Size of A^m and F^m is N^2 resp. N.

Online:

For a given parameter value $\boldsymbol{\mu} \in \mathcal{D}$
1) Assemble (depending on M and N, i.e. $\sim MN^2$ resp. $\sim MN$)

$$A = \sum_{m=1}^{M} \Theta^m(\boldsymbol{\mu}) A^m \qquad F = \sum_{m=1}^{M} \Theta_f^m(\boldsymbol{\mu}) F^m$$

2) Solve $A\boldsymbol{u}_N(\boldsymbol{\mu}) = F$. (depending on N, i.e $\sim N^3$ for LU factorization)

Efficiency: How to solve (2) in a fast way?

Offline:

Given $\mathbb{W}_N = \mathrm{span}\{\boldsymbol{\xi}_i \,|\, i = 1, \dots, N\}$ precompute

$$
\begin{aligned}
(A^m)_{i,j} &= a^m(\boldsymbol{\xi}_j, \boldsymbol{\xi}_i), && \forall 1 \le i, j \le N, \\
(F^m)_i &= f^m(\boldsymbol{\xi}_i), && \forall 1 \le i \le N.
\end{aligned}
$$

Rem. Depends on $\mathcal{N} = \dim(\mathbb{V}_h)$.
Rem. Size of A^m and F^m is N^2 resp. N.

Online:

For a given parameter value $\boldsymbol{\mu} \in \mathcal{D}$
1) Assemble (depending on M and N, i.e. $\sim MN^2$ resp. $\sim MN$)

$$
A = \sum_{m=1}^{M} \Theta^m(\boldsymbol{\mu}) A^m \qquad F = \sum_{m=1}^{M} \Theta_f^m(\boldsymbol{\mu}) F^m
$$

2) Solve $A\boldsymbol{u}_N(\boldsymbol{\mu}) = F$. (depending on N, i.e $\sim N^3$ for LU factorization)

- In the same vein we can compute the a posteriori estimate and the RCS/output functional
- Computation time also depends on M!

Efficiency: Empirical Interpolation Method (EIM)

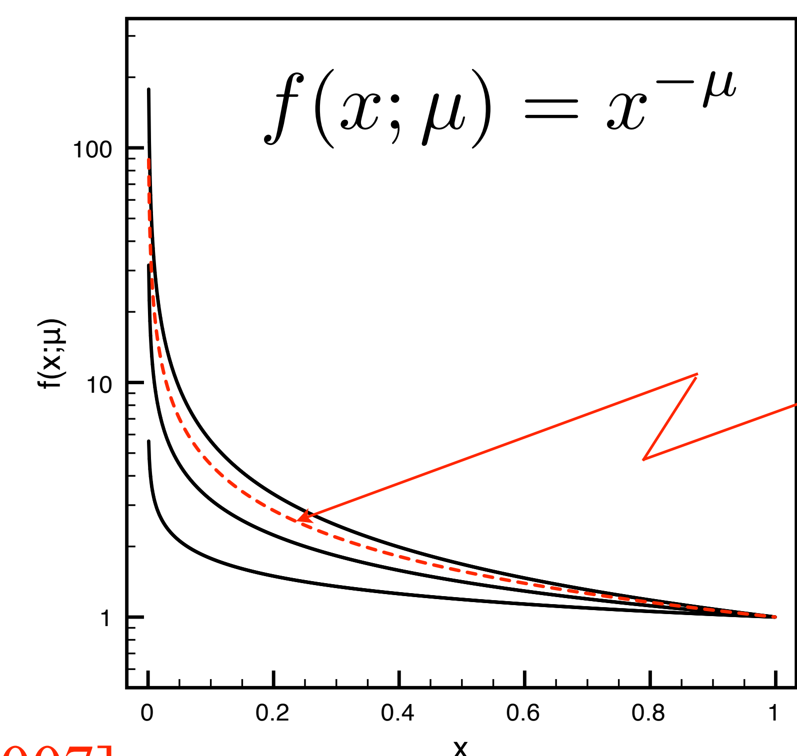
(allows to realize the affine assumption approximatively)

Efficiency: EIM

Let $f : \Omega \times \mathcal{D} \rightarrow \mathbb{C}$ such that $f(\,\cdot\,;\boldsymbol{\mu}) \in C^0(\Omega)$ for all $\boldsymbol{\mu} \in \mathcal{D}$. The **EIM** is a procedure that provides $\{\boldsymbol{\mu}_m\}_{m=1}^{M}$ such that

$$\boxed{\mathcal{I}_M(f)(\boldsymbol{x};\boldsymbol{\mu}) = \sum_{m=1}^{M} \alpha_m(\boldsymbol{\mu}) f(\boldsymbol{x};\boldsymbol{\mu}_m)}$$

is a good approximation of $f(\boldsymbol{x};\boldsymbol{\mu})$ for all $(\boldsymbol{x},\boldsymbol{\mu}) \in \Omega \times \mathcal{D}$. Uses also a greedy algorithm to pick the parameters $\{\boldsymbol{\mu}_m\}_{m=1}^{M}$.



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

[Grepl et al. 2007], [Maday et al. 2007]

Efficiency: EIM

Let $f : \Omega \times \mathcal{D} \to \mathbb{C}$ such that $f(\,\cdot\,;\boldsymbol{\mu}) \in C^0(\Omega)$ for all $\boldsymbol{\mu} \in \mathcal{D}$. The **EIM** is a procedure that provides $\{\boldsymbol{\mu}_m\}_{m=1}^M$ such that

$$\boxed{\mathcal{I}_M(f)(\boldsymbol{x};\boldsymbol{\mu}) = \sum_{m=1}^{M} \alpha_m(\boldsymbol{\mu}) f(\boldsymbol{x};\boldsymbol{\mu}_m)}$$

is a good approximation of $f(\boldsymbol{x};\boldsymbol{\mu})$ for all $(\boldsymbol{x},\boldsymbol{\mu}) \in \Omega \times \mathcal{D}$. Uses also a greedy algorithm to pick the parameters $\{\boldsymbol{\mu}_m\}_{m=1}^M$.

Examples:
1) Non-singular part of kernel function:

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

2) Incident plane wave:

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

with $\boldsymbol{\mu} = (k, \theta, \phi)$.

Efficiency: EIM implementation for EFIE

1) Split the kernel function into the singular part and non-singular part

$$G_k(r) = r^{-1} + G_k^{ns}(r)$$

Efficiency: EIM implementation for EFIE

1) Split the kernel function into the singular part and non-singular part

$$G_k(r) = r^{-1} + G_k^{ns}(r)$$

2) Insert it into the sequilinear form

$$a(\boldsymbol{w}, \boldsymbol{v}; k) = \int_{\Gamma\times\Gamma} \frac{1}{4\pi|\boldsymbol{x}-\boldsymbol{y}|} \left\{\boldsymbol{w}(\boldsymbol{x}) \cdot \overline{\boldsymbol{v}(\boldsymbol{y})} - \frac{1}{k^2}\mathrm{div}_\Gamma \boldsymbol{w}(\boldsymbol{x})\, \overline{\mathrm{div}_\Gamma \boldsymbol{v}(\boldsymbol{y})}\right\} d\boldsymbol{x}\, d\boldsymbol{y}$$
$$+ \int_{\Gamma\times\Gamma} G_k^{ns}(|\boldsymbol{x}-\boldsymbol{y}|) \left\{\boldsymbol{w}(\boldsymbol{x}) \cdot \overline{\boldsymbol{v}(\boldsymbol{y})} - \frac{1}{k^2}\mathrm{div}_\Gamma \boldsymbol{w}(\boldsymbol{x})\, \overline{\mathrm{div}_\Gamma \boldsymbol{v}(\boldsymbol{y})}\right\} d\boldsymbol{x}\, d\boldsymbol{y}.$$

Efficiency: EIM implementation for EFIE

1) Split the kernel function into the singular part and non-singular part

$$G_k(r) = r^{-1} + G_k^{ns}(r)$$

2) Insert it into the sequilinear form

3) Replace non-singular kernel function by its EIM interpolant
$G_k^{ns}(r) \approx \sum_{m=1}^{M} \alpha_m(k) G_{k_m}^{ns}(r)$

$$\begin{aligned} a(\boldsymbol{w}, \boldsymbol{v}; k) = & \int_{\Gamma\times\Gamma} \frac{1}{4\pi|\boldsymbol{x}-\boldsymbol{y}|} \left\{ \boldsymbol{w}(\boldsymbol{x}) \cdot \overline{\boldsymbol{v}(\boldsymbol{y})} - \frac{1}{k^2} \mathrm{div}_\Gamma \boldsymbol{w}(\boldsymbol{x}) \, \overline{\mathrm{div}_\Gamma \boldsymbol{v}(\boldsymbol{y})} \right\} d\boldsymbol{x}\, d\boldsymbol{y} \\ & + \int_{\Gamma\times\Gamma} G_k^{ns}(|\boldsymbol{x}-\boldsymbol{y}|) \left\{ \boldsymbol{w}(\boldsymbol{x}) \cdot \overline{\boldsymbol{v}(\boldsymbol{y})} - \frac{1}{k^2} \mathrm{div}_\Gamma \boldsymbol{w}(\boldsymbol{x}) \, \overline{\mathrm{div}_\Gamma \boldsymbol{v}(\boldsymbol{y})} \right\} d\boldsymbol{x}\, d\boldsymbol{y}. \end{aligned}$$

Efficiency: EIM implementation for EFIE

1) Split the kernel function into the singular part and non-singular part

$$G_k(r) = r^{-1} + G_k^{ns}(r)$$

2) Insert it into the sequilinear form

3) Replace non-singular kernel function by its EIM interpolant
$G_k^{ns}(r) \approx \sum_{m=1}^{M} \alpha_m(k) G_{k_m}^{ns}(r)$

$$\begin{aligned}
a(\boldsymbol{w}, \boldsymbol{v}; k) \approx\ & 1 \int_{\Gamma\times\Gamma} \frac{\boldsymbol{w}(\boldsymbol{x}) \cdot \overline{\boldsymbol{v}(\boldsymbol{y})}}{4\pi|\boldsymbol{x}-\boldsymbol{y}|} d\boldsymbol{x}\, d\boldsymbol{y} \\
& - \frac{1}{k^2} \int_{\Gamma\times\Gamma} \frac{\operatorname{div}_\Gamma \boldsymbol{w}(\boldsymbol{x})\, \overline{\operatorname{div}_\Gamma \boldsymbol{v}(\boldsymbol{y})}}{4\pi|\boldsymbol{x}-\boldsymbol{y}|} d\boldsymbol{x}\, d\boldsymbol{y} \\
& + \sum_{m=1}^{M} \alpha_m(k) \int_{\Gamma\times\Gamma} G_{k_m}^{ns}(|\boldsymbol{x}-\boldsymbol{y}|) \boldsymbol{w}(\boldsymbol{x}) \cdot \overline{\boldsymbol{v}(\boldsymbol{y})} d\boldsymbol{x}\, d\boldsymbol{y} \\
& - \sum_{m=1}^{M} \frac{\alpha_m(k)}{k^2} \int_{\Gamma\times\Gamma} G_{k_m}^{ns}(|\boldsymbol{x}-\boldsymbol{y}|) \operatorname{div}_\Gamma \boldsymbol{w}(\boldsymbol{x})\, \overline{\operatorname{div}_\Gamma \boldsymbol{v}(\boldsymbol{y})} d\boldsymbol{x}\, d\boldsymbol{y}
\end{aligned}$$

blue: parameter independent
red: parameter dependent

Efficiency: EIM implementation for EFIE

$$
\begin{aligned}
a(\boldsymbol{w},\boldsymbol{v};k) \approx\ & 1 \int_{\Gamma\times\Gamma} \frac{\boldsymbol{w}(\boldsymbol{x})\cdot\overline{\boldsymbol{v}(\boldsymbol{y})}}{4\pi|\boldsymbol{x}-\boldsymbol{y}|} d\boldsymbol{x}\, d\boldsymbol{y} \\
& - \frac{1}{k^2} \int_{\Gamma\times\Gamma} \frac{\mathrm{div}_\Gamma \boldsymbol{w}(\boldsymbol{x})\, \overline{\mathrm{div}_\Gamma \boldsymbol{v}(\boldsymbol{y})}}{4\pi|\boldsymbol{x}-\boldsymbol{y}|} d\boldsymbol{x}\, d\boldsymbol{y} \\
& + \sum_{m=1}^{M} \alpha_m(k) \int_{\Gamma\times\Gamma} G^{ns}_{k_m}(|\boldsymbol{x}-\boldsymbol{y}|)\boldsymbol{w}(\boldsymbol{x})\cdot\overline{\boldsymbol{v}(\boldsymbol{y})} d\boldsymbol{x}\, d\boldsymbol{y} \\
& - \sum_{m=1}^{M} \frac{\alpha_m(k)}{k^2} \int_{\Gamma\times\Gamma} G^{ns}_{k_m}(|\boldsymbol{x}-\boldsymbol{y}|)\mathrm{div}_\Gamma \boldsymbol{w}(\boldsymbol{x})\, \overline{\mathrm{div}_\Gamma \boldsymbol{v}(\boldsymbol{y})} d\boldsymbol{x}\, d\boldsymbol{y}
\end{aligned}
$$

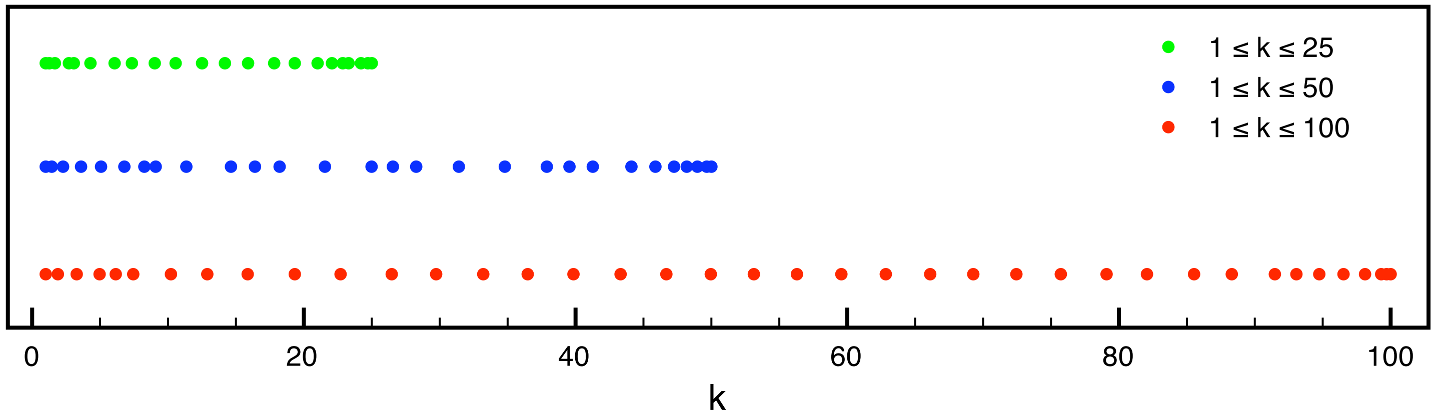
blue: parameter independent
red: parameter dependent

In the same manner for

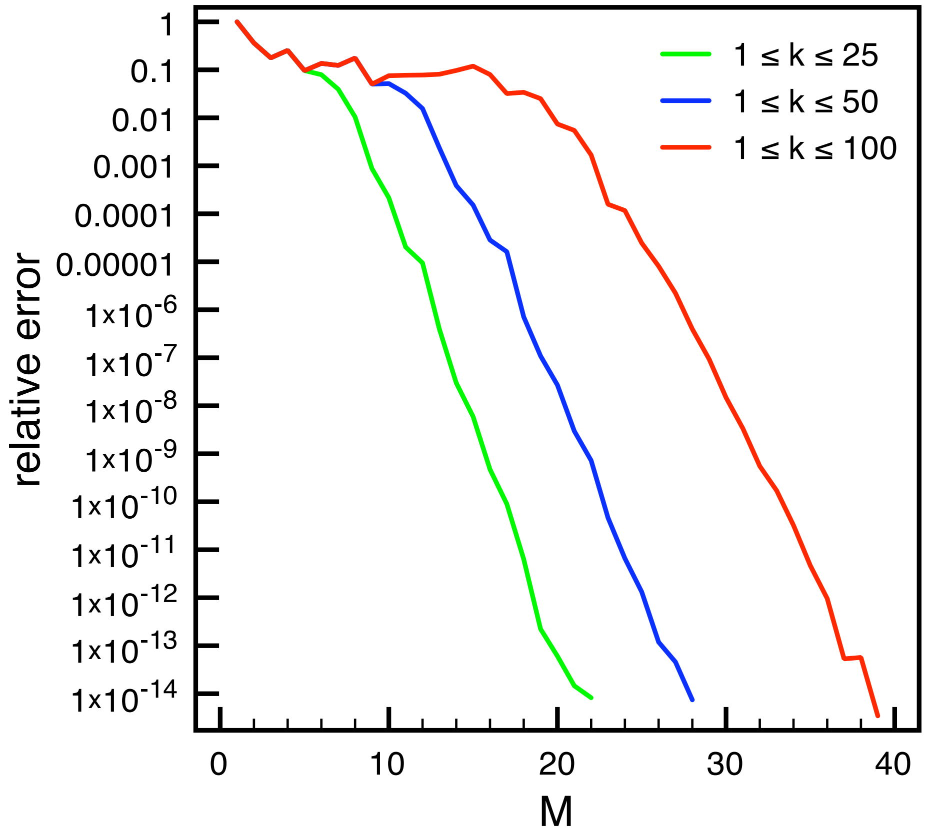
$$
F(\boldsymbol{v};\boldsymbol{\mu}) \approx \sum_{m=1}^{M_f} \alpha_f(\boldsymbol{\mu}) \int_\Gamma \gamma_t \mathbf{E}^i(\boldsymbol{y};\boldsymbol{\mu}_m)\cdot\overline{\boldsymbol{v}(\boldsymbol{y})} d\boldsymbol{y}
$$

Numerical results for EIM

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



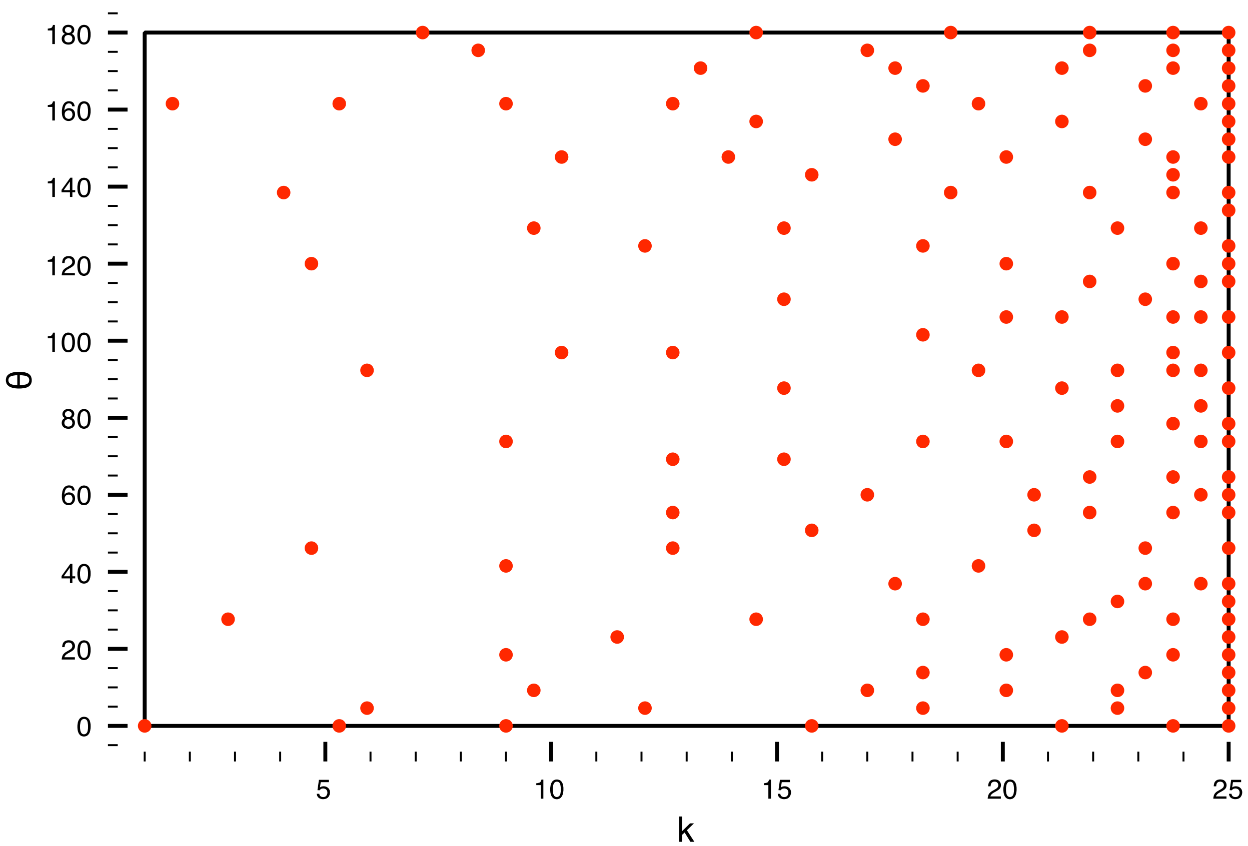
Picked parameters k_m in the parameter domain



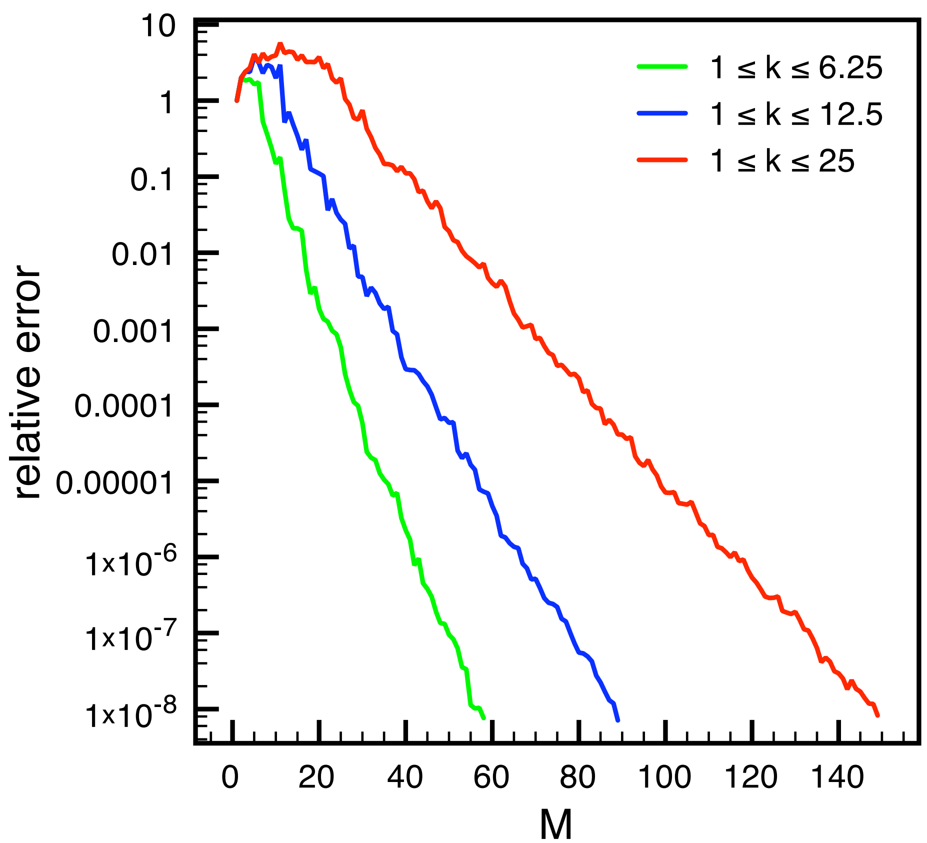
Interpolation error depending on the length of the expansion

Numerical results for EIM

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

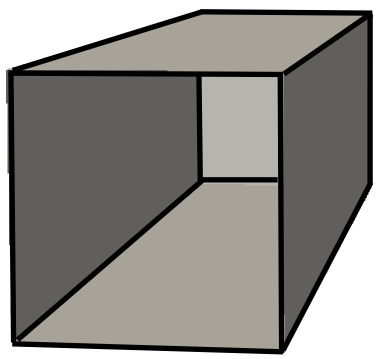


Picked parameters in the parameter domain for k_{max}=25



Interpolation error depending on the length of the expansion

Surface Γ given by:

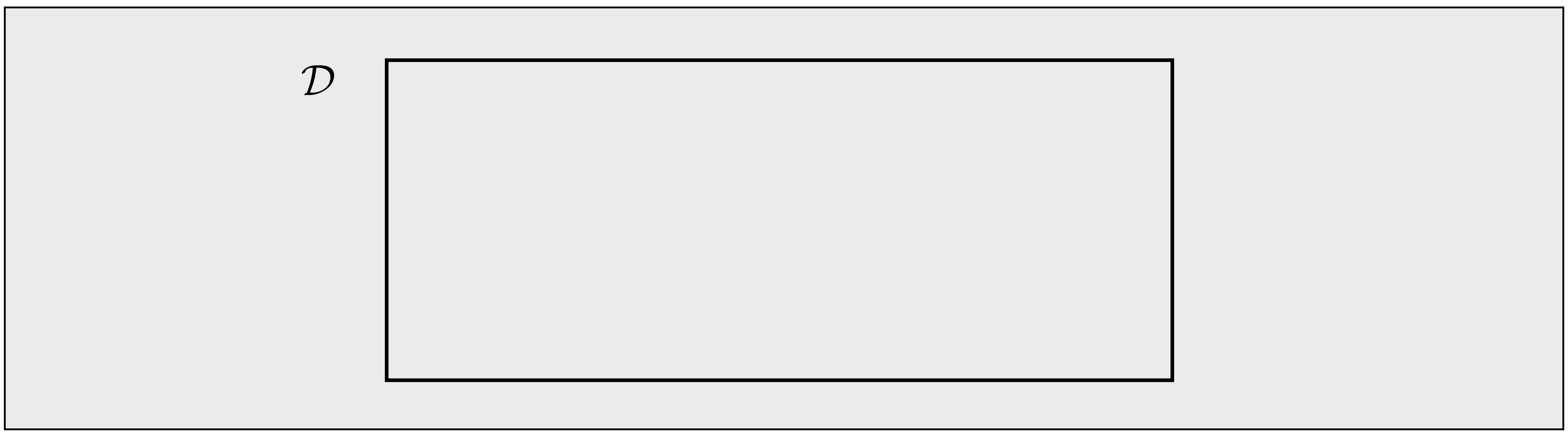


Efficiency: Elementwise EIM/*hp*-Interpolant

- Problems with large parameter domains, the expansion becomes too large and this can become that severe that the computing time of the RB solution (only online time) is in the order of a direct computation.
- As solution, the *parameter* domain can adaptively be split into subelements on which the function is approximated by a different Magic point expansion.
- Schematic illustration (2d parameter domain):

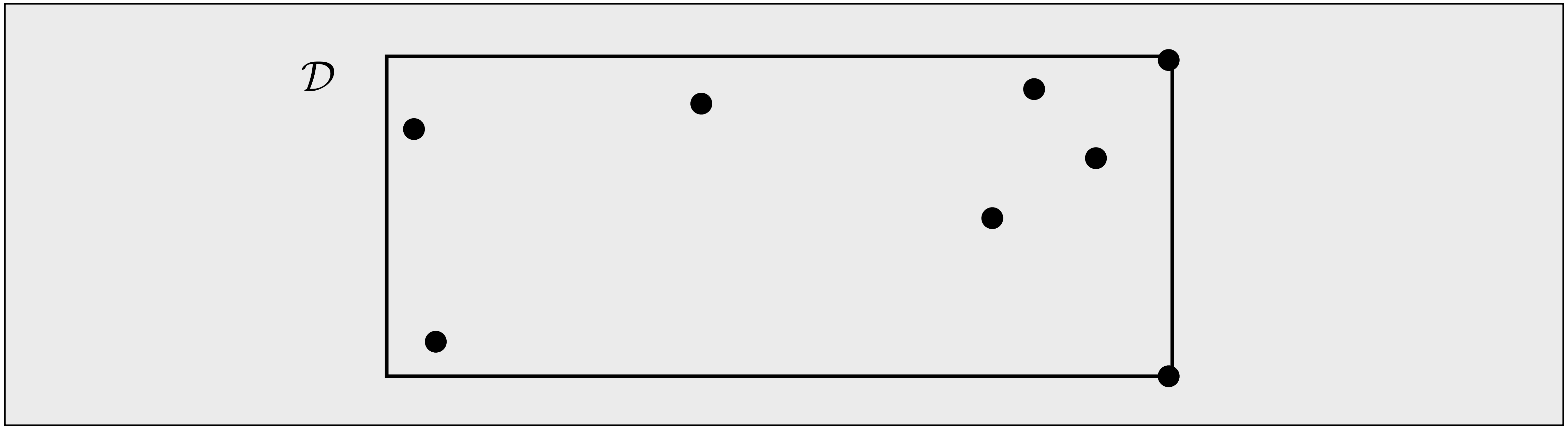
Efficiency: Elementwise EIM/*hp*-Interpolant

- Problems with large parameter domains, the expansion becomes too large and this can become that severe that the computing time of the RB solution (only online time) is in the order of a direct computation.
- As solution, the *parameter* domain can adaptively be split into subelements on which the function is approximated by a different Magic point expansion.
- Schematic illustration (2d parameter domain):



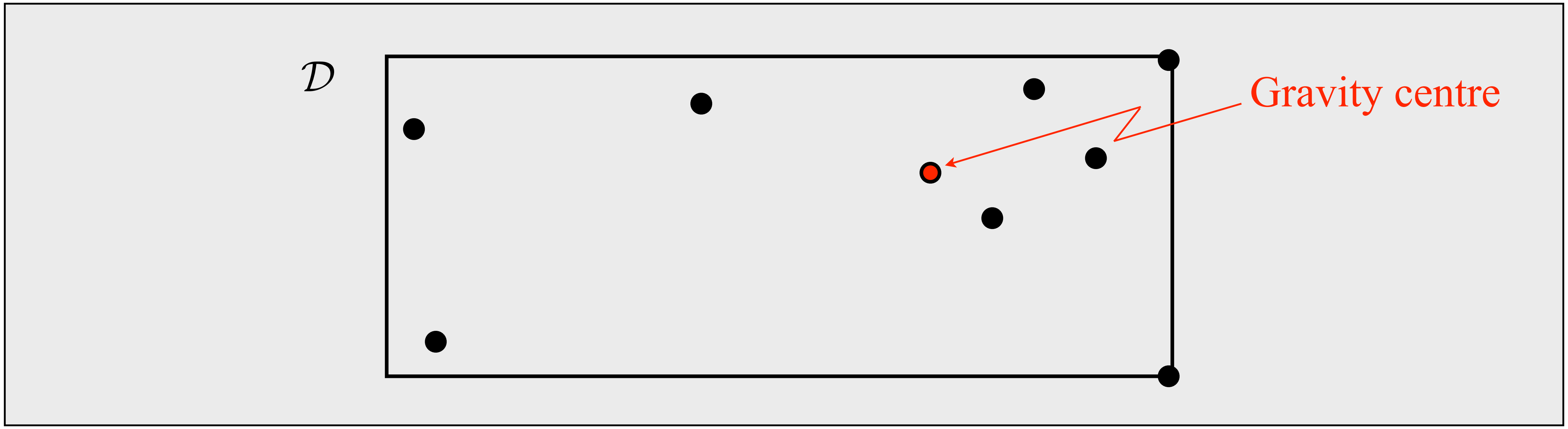
Efficiency: Elementwise EIM/*hp*-Interpolant

- Problems with large parameter domains, the expansion becomes too large and this can become that severe that the computing time of the RB solution (only online time) is in the order of a direct computation.
- As solution, the *parameter* domain can adaptively be split into subelements on which the function is approximated by a different Magic point expansion.
- Schematic illustration (2d parameter domain):



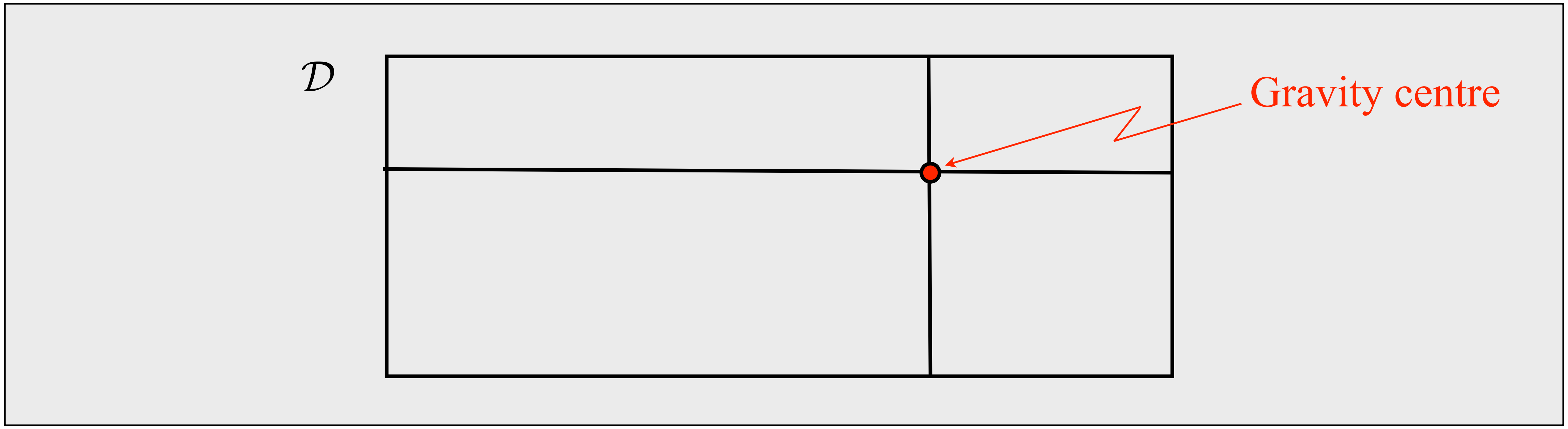
Efficiency: Elementwise EIM/*hp*-Interpolant

- Problems with large parameter domains, the expansion becomes too large and this can become that severe that the computing time of the RB solution (only online time) is in the order of a direct computation.
- As solution, the *parameter* domain can adaptively be split into subelements on which the function is approximated by a different Magic point expansion.
- Schematic illustration (2d parameter domain):



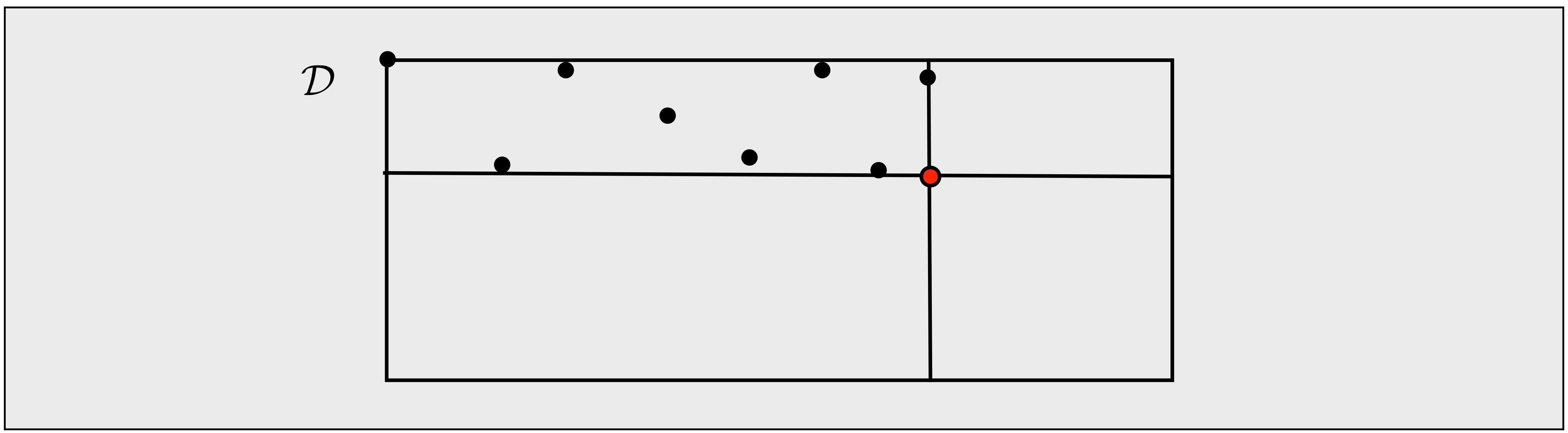
Efficiency: Elementwise EIM/*hp*-Interpolant

- Problems with large parameter domains, the expansion becomes too large and this can become that severe that the computing time of the RB solution (only online time) is in the order of a direct computation.
- As solution, the *parameter* domain can adaptively be split into subelements on which the function is approximated by a different Magic point expansion.
- Schematic illustration (2d parameter domain):



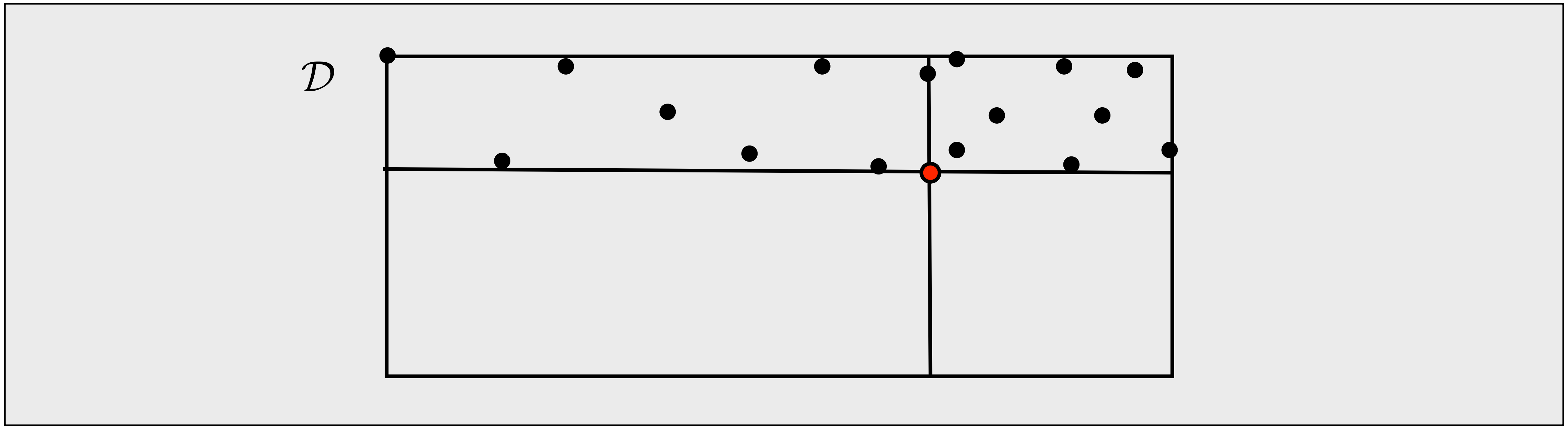
Efficiency: Elementwise EIM/*hp*-Interpolant

- Problems with large parameter domains, the expansion becomes too large and this can become that severe that the computing time of the RB solution (only online time) is in the order of a direct computation.
- As solution, the *parameter* domain can adaptively be split into subelements on which the function is approximated by a different Magic point expansion.
- Schematic illustration (2d parameter domain):



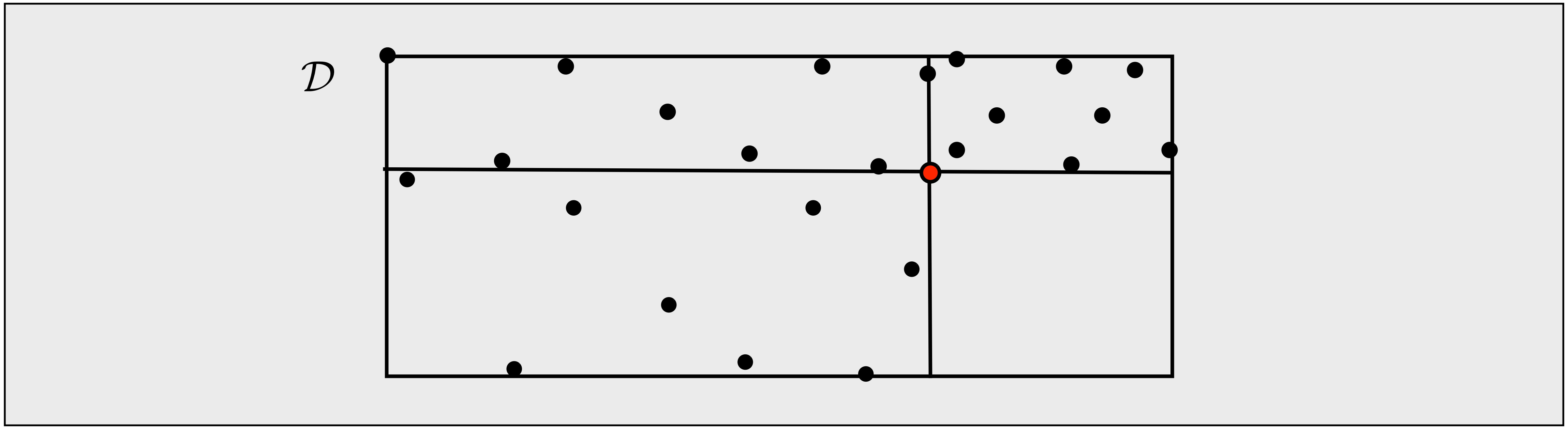
Efficiency: Elementwise EIM/*hp*-Interpolant

- Problems with large parameter domains, the expansion becomes too large and this can become that severe that the computing time of the RB solution (only online time) is in the order of a direct computation.
- As solution, the *parameter* domain can adaptively be split into subelements on which the function is approximated by a different Magic point expansion.
- Schematic illustration (2d parameter domain):



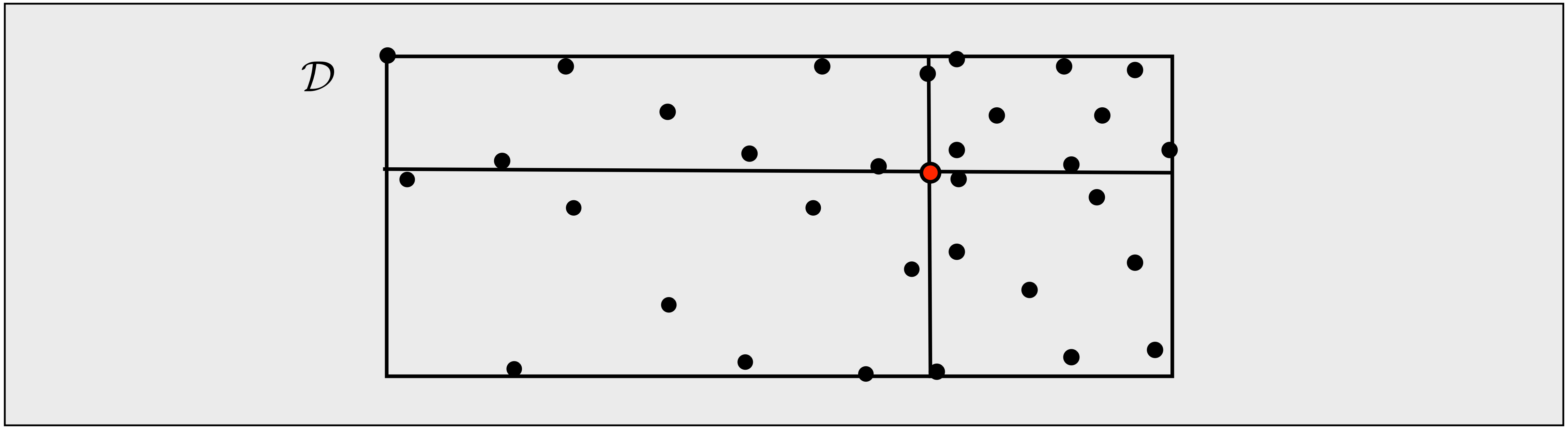
Efficiency: Elementwise EIM/*hp*-Interpolant

- Problems with large parameter domains, the expansion becomes too large and this can become that severe that the computing time of the RB solution (only online time) is in the order of a direct computation.
- As solution, the *parameter* domain can adaptively be split into subelements on which the function is approximated by a different Magic point expansion.
- Schematic illustration (2d parameter domain):



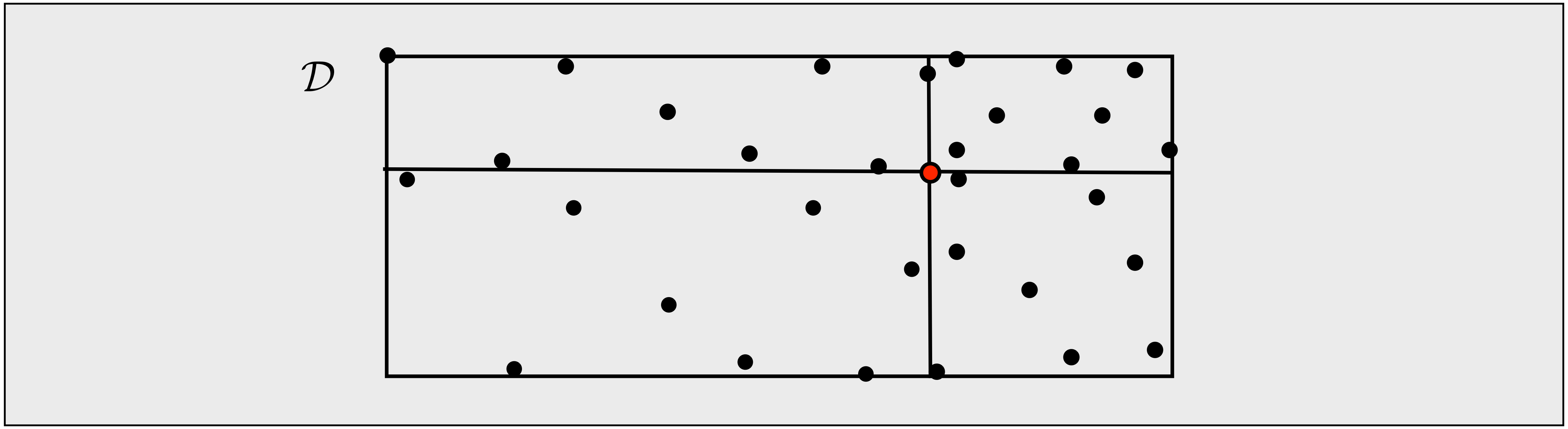
Efficiency: Elementwise EIM/*hp*-Interpolant

- Problems with large parameter domains, the expansion becomes too large and this can become that severe that the computing time of the RB solution (only online time) is in the order of a direct computation.
- As solution, the *parameter* domain can adaptively be split into subelements on which the function is approximated by a different Magic point expansion.
- Schematic illustration (2d parameter domain):



Efficiency: Elementwise EIM/*hp*-Interpolant

- Problems with large parameter domains, the expansion becomes too large and this can become that severe that the computing time of the RB solution (only online time) is in the order of a direct computation.
- As solution, the *parameter* domain can adaptively be split into subelements on which the function is approximated by a different Magic point expansion.
- Schematic illustration (2d parameter domain):

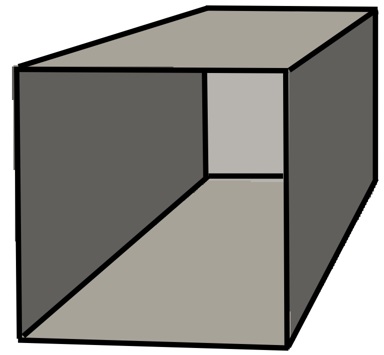


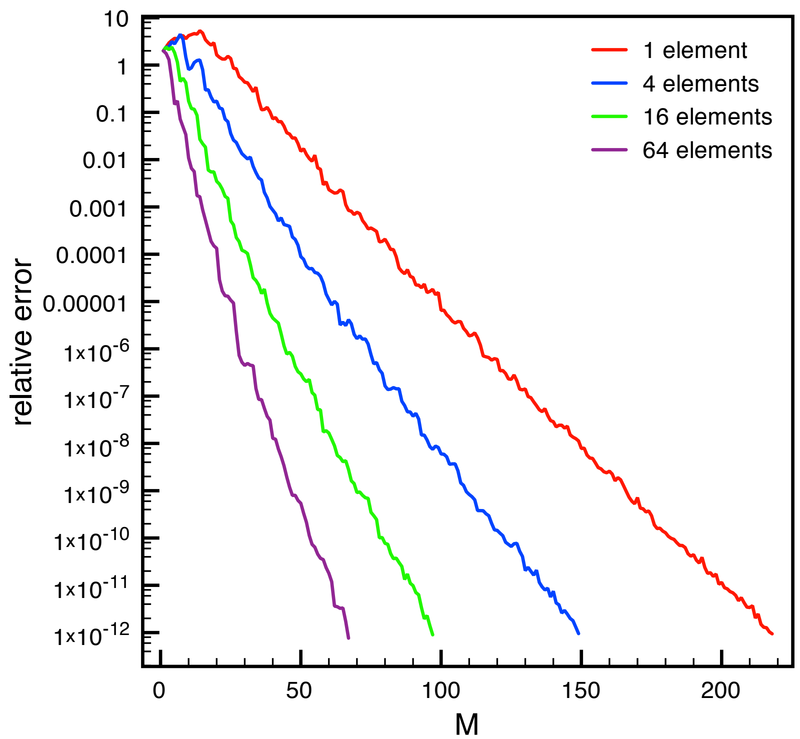
- Refinement until on each subdomain a certain tolerance is reached
- Parameter domain only is refined
- Generalization to any dimension of the parameter space possible

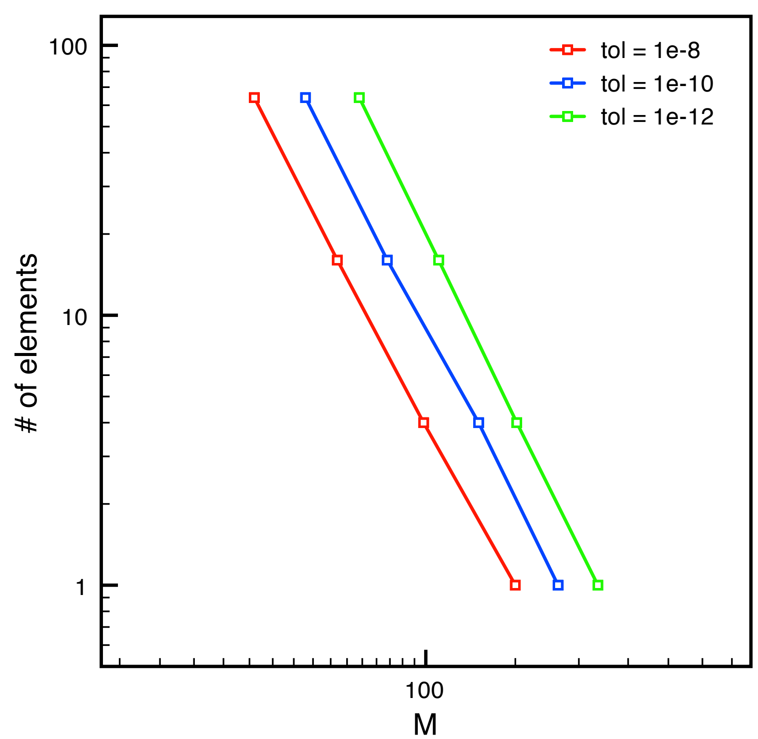
Numerical results elementwise EIM

$$
\begin{aligned}
f(\boldsymbol{x};\boldsymbol{\mu}) &= e^{ik\hat{\boldsymbol{k}}(\theta,\phi)\cdot\boldsymbol{x}}, \quad \boldsymbol{x}\in\Gamma, \boldsymbol{\mu}\in\mathcal{D},\\
\boldsymbol{\mu} &= (k,\theta), \quad \phi \text{ fixed},\\
\mathcal{D} &= [1,25]\times[0,\pi]
\end{aligned}
$$

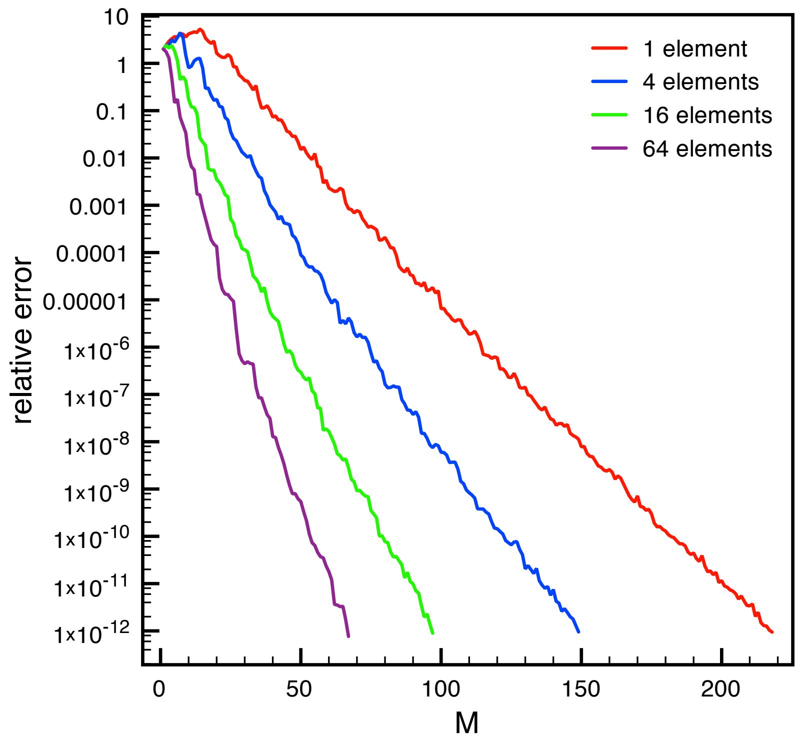
Surface Γ given by:

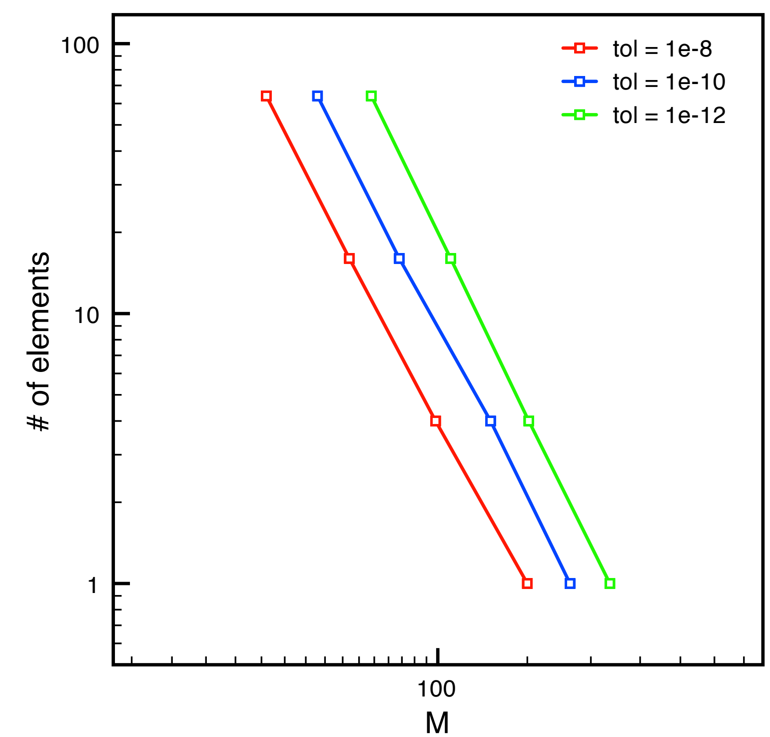






Numerical results elementwise EIM





Conclusion: A reduction of M implies a algebraic increase of number of elements (and dofs) needed. In this case:

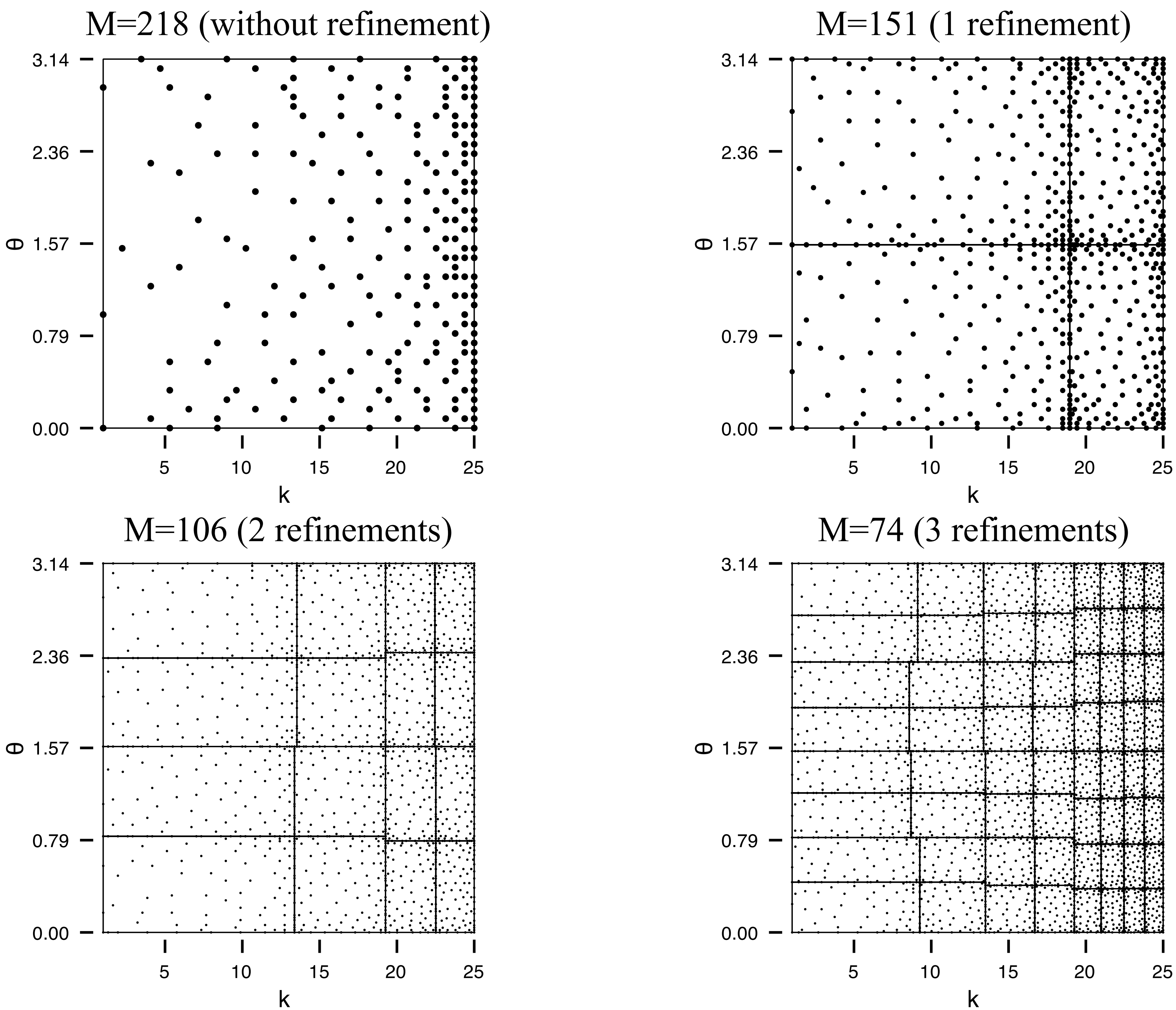
$$\#\text{elements} \approx CM^{-3.7}$$

But it reduces the online computing time (at the cost of a longer Offline procedure and more memory)

Shift of workload from Online part to Offline part

Numerical results elementwise EIM

Picked parameter values and EIM elements
(tol=1e-12):

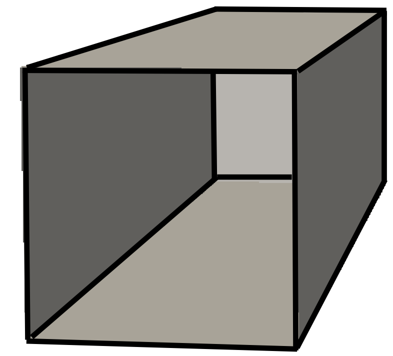


Numerical result for reduced basis method

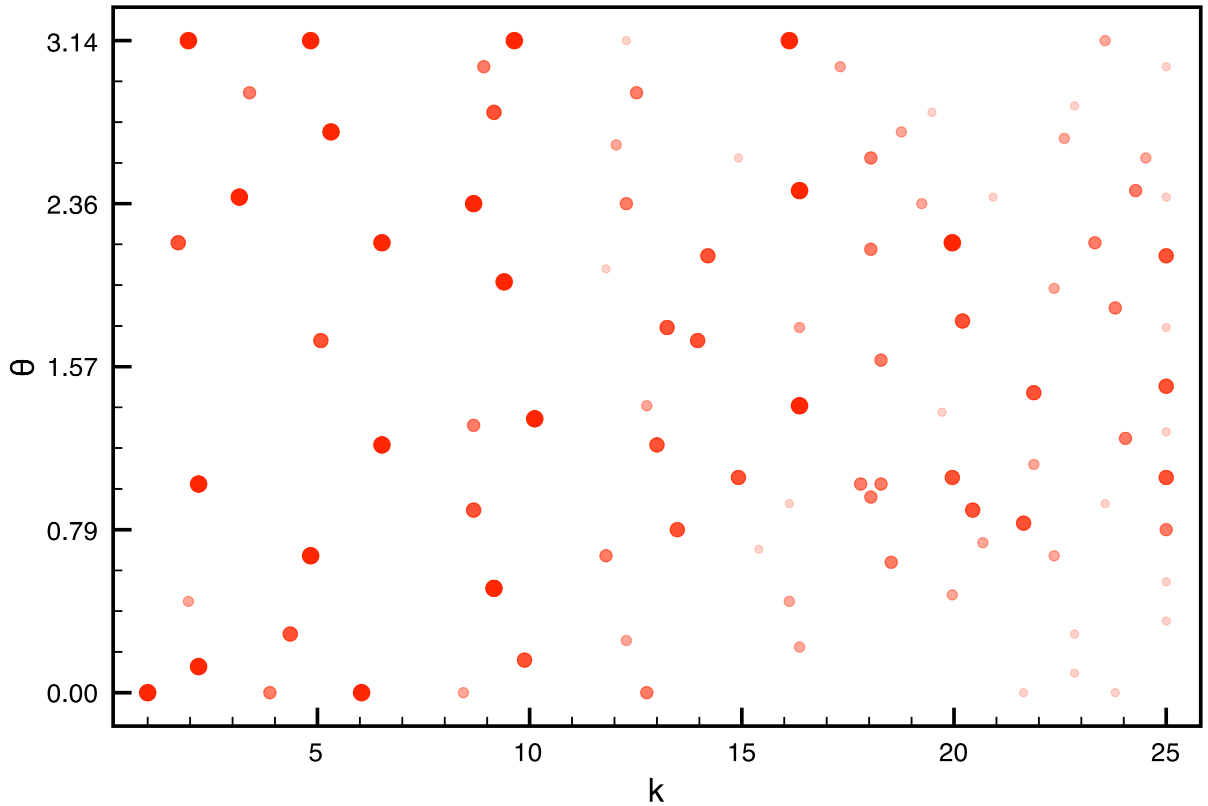
Numerical results: test 1

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

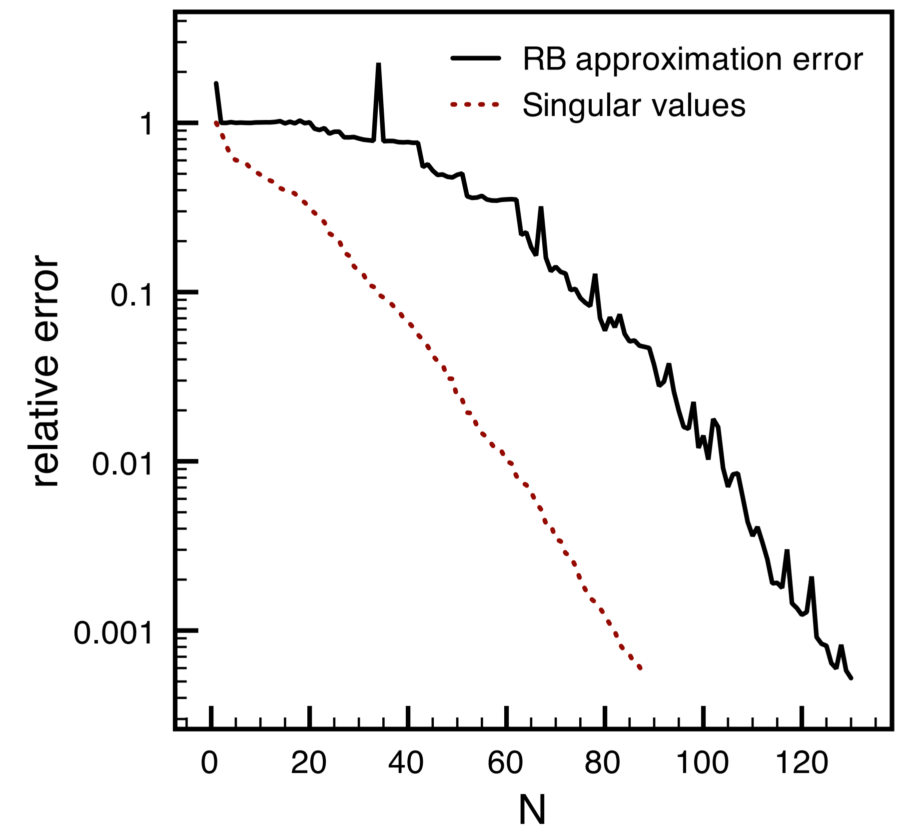
Surface Γ given by:



Picked parameters:



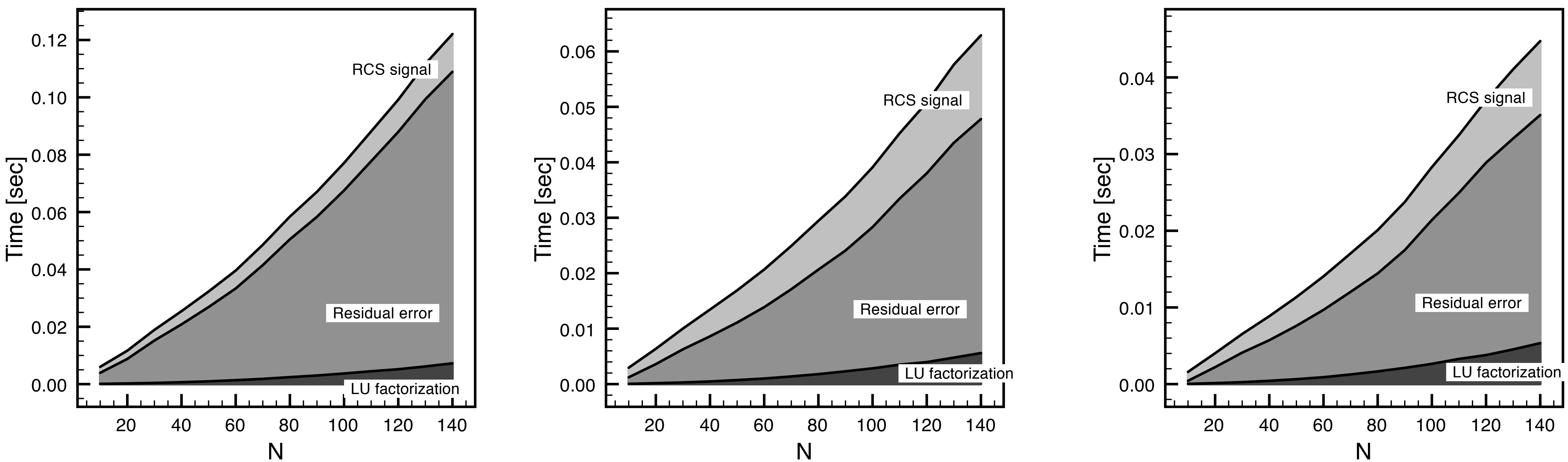
Convergence:



$$\text{relative error} = \frac{\max_{\boldsymbol{\mu}} \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_{L^2(\Omega)}}{\max_{\boldsymbol{\mu}} \|u_h(\boldsymbol{\mu})\|_{L^2(\Omega)}}$$

Numerical results: test 1

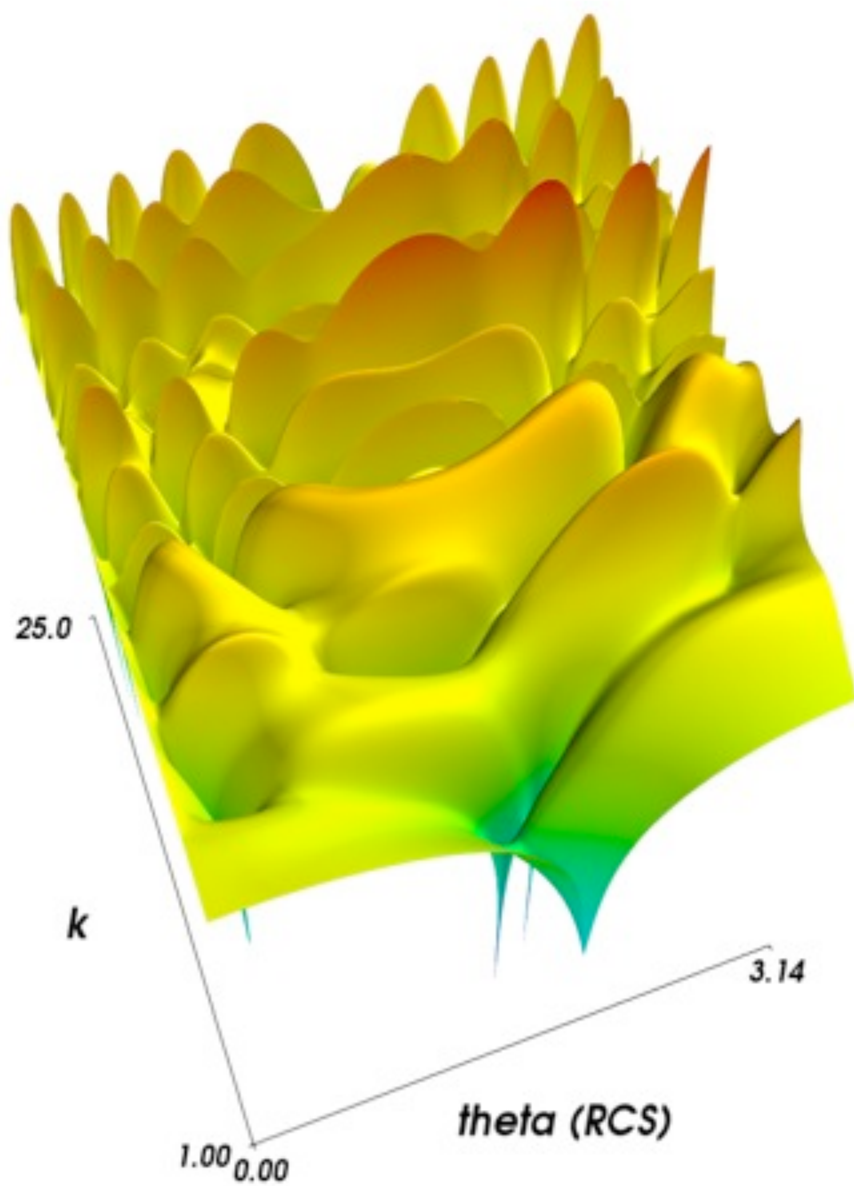
2 parameters, $\boldsymbol{\mu} = (k, \theta)$ with $\mathcal{D} = [1, 25] \times [0, \pi]$
$\phi = 0$ fixed



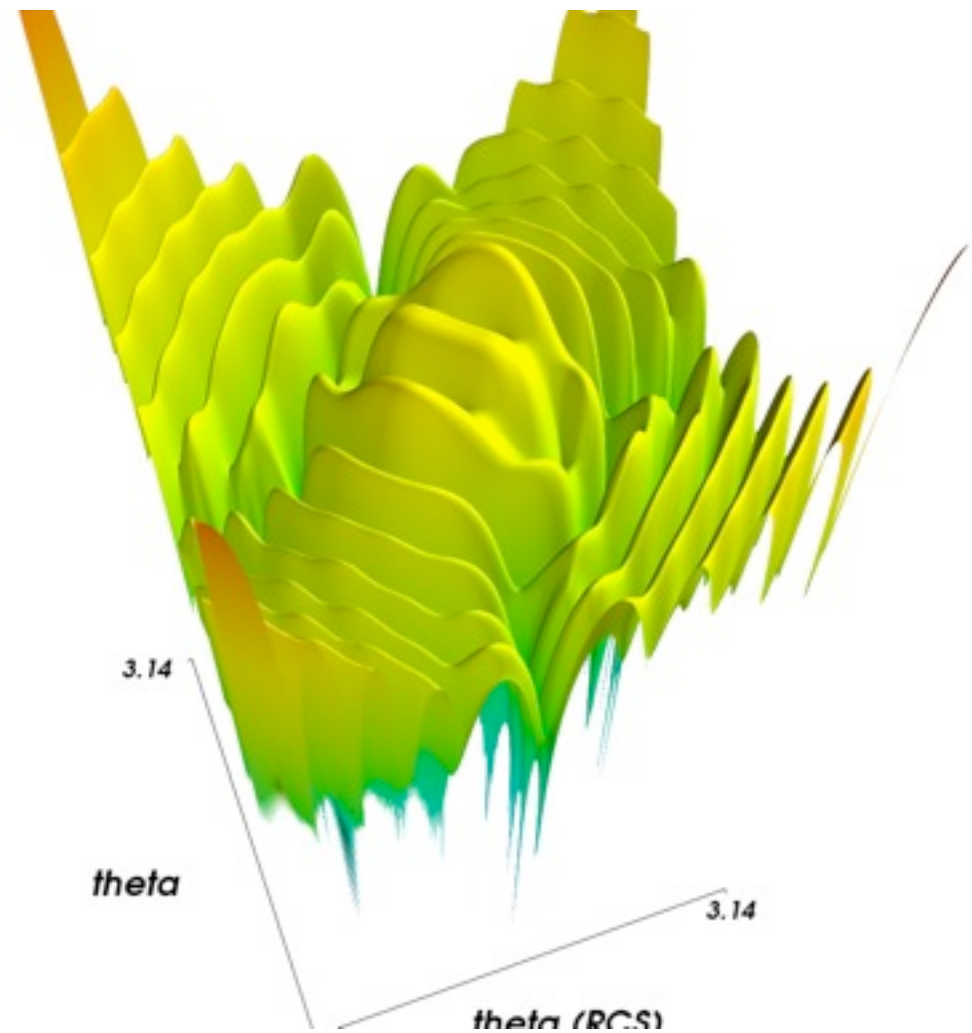
Number of elements used for EEIM			
Kernel function	1	4	16
Right hand side	1	4	16
RCS	4	16	64

Numerical results: test 1

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



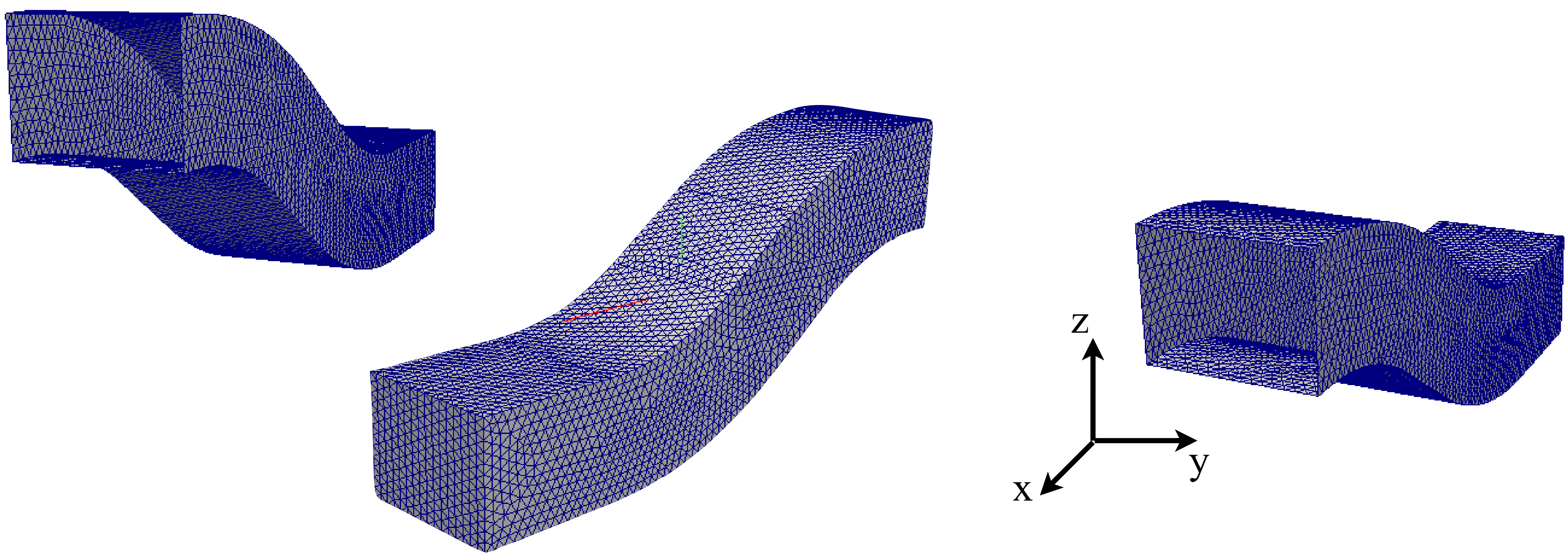
$\boldsymbol{\theta = \pi/4}$



$\boldsymbol{k = 25}$

More complex scatterer and parallelization

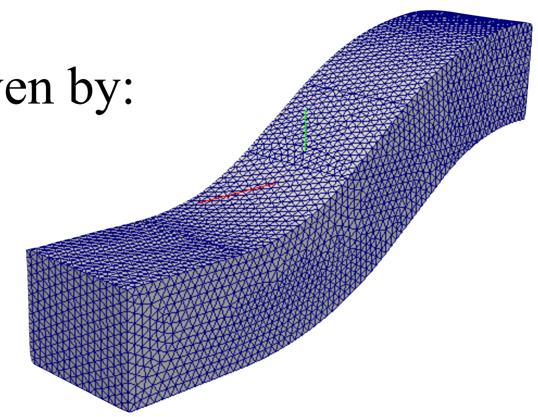
- 12620 complex double unknowns
- BEM matrix has 160 Mio complex double entries
- Used 160 processors with distributed memory for computations
- Solving linear system: Cyclic distribution by Scalapack: parallel LU-factorization
- Matrix-matrix, matrix-vector multiplication: Blockwise computations using blacs/blas



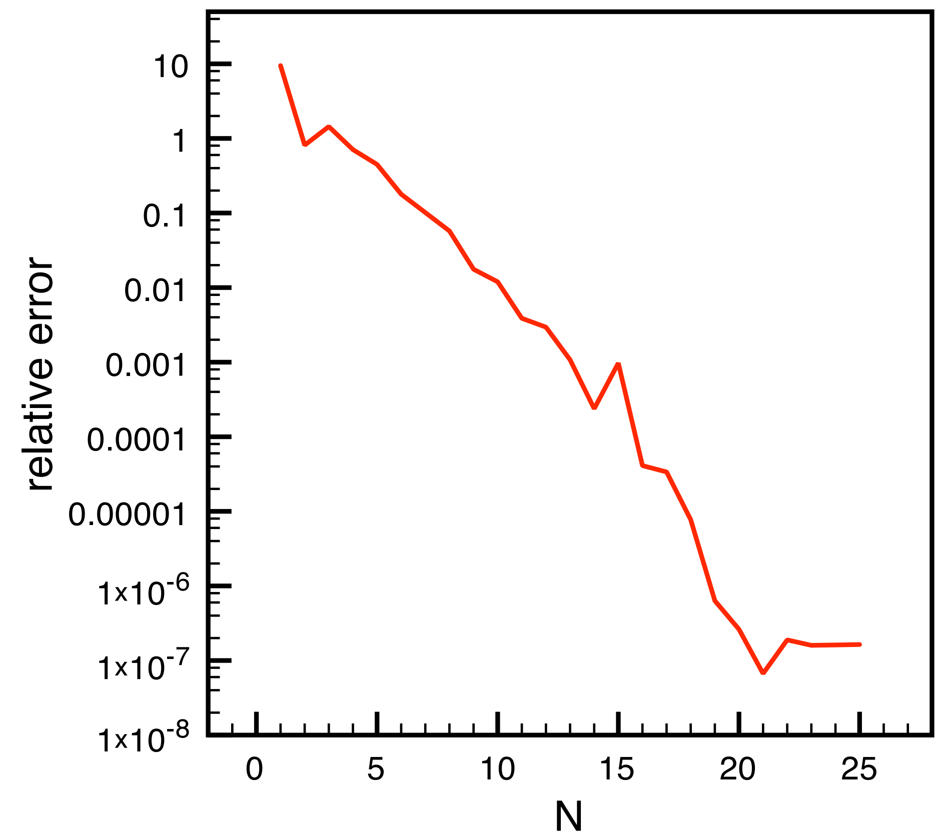
Numerical results: test 2

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

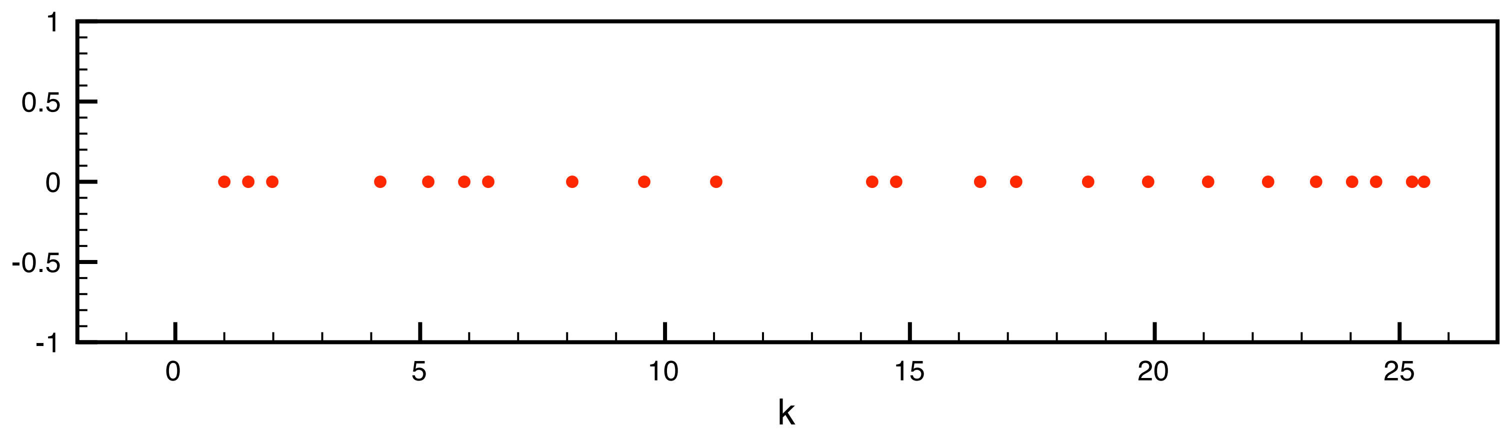
Surface Γ given by:



Convergence:



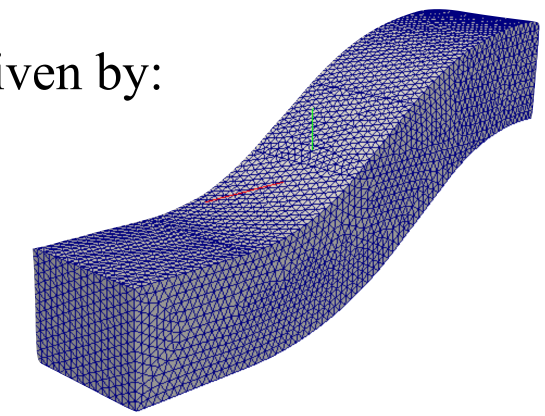
Repartition of 23 first picked parameters:



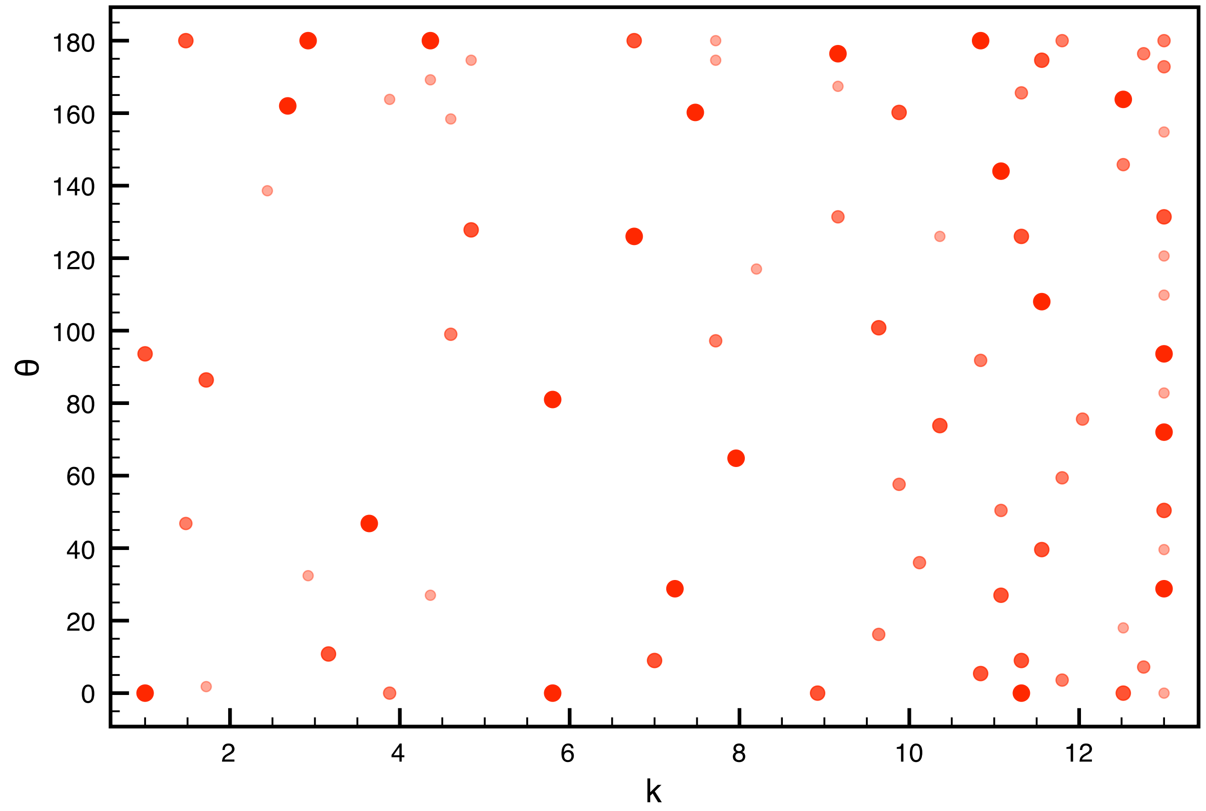
Numerical results: test 3

2 parameters, $\boldsymbol{\mu} = (k, \theta)$ with $\mathcal{D} = [1, 13] \times [0, \pi]$
$\phi = 0$ fixed

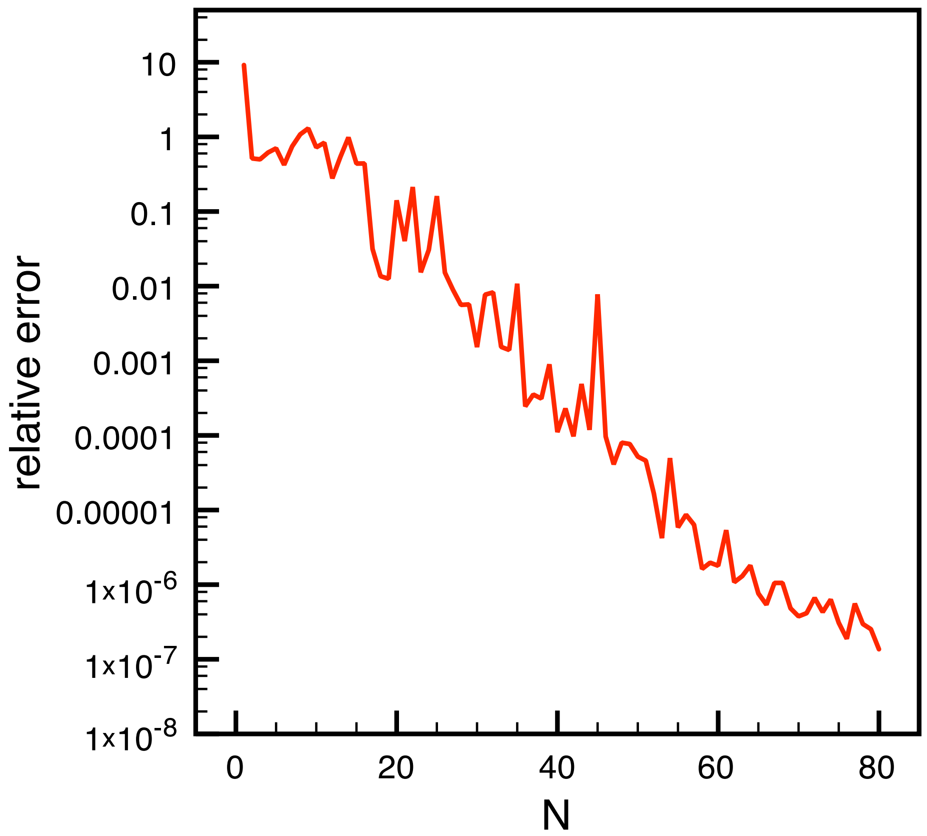
Surface Γ given by:



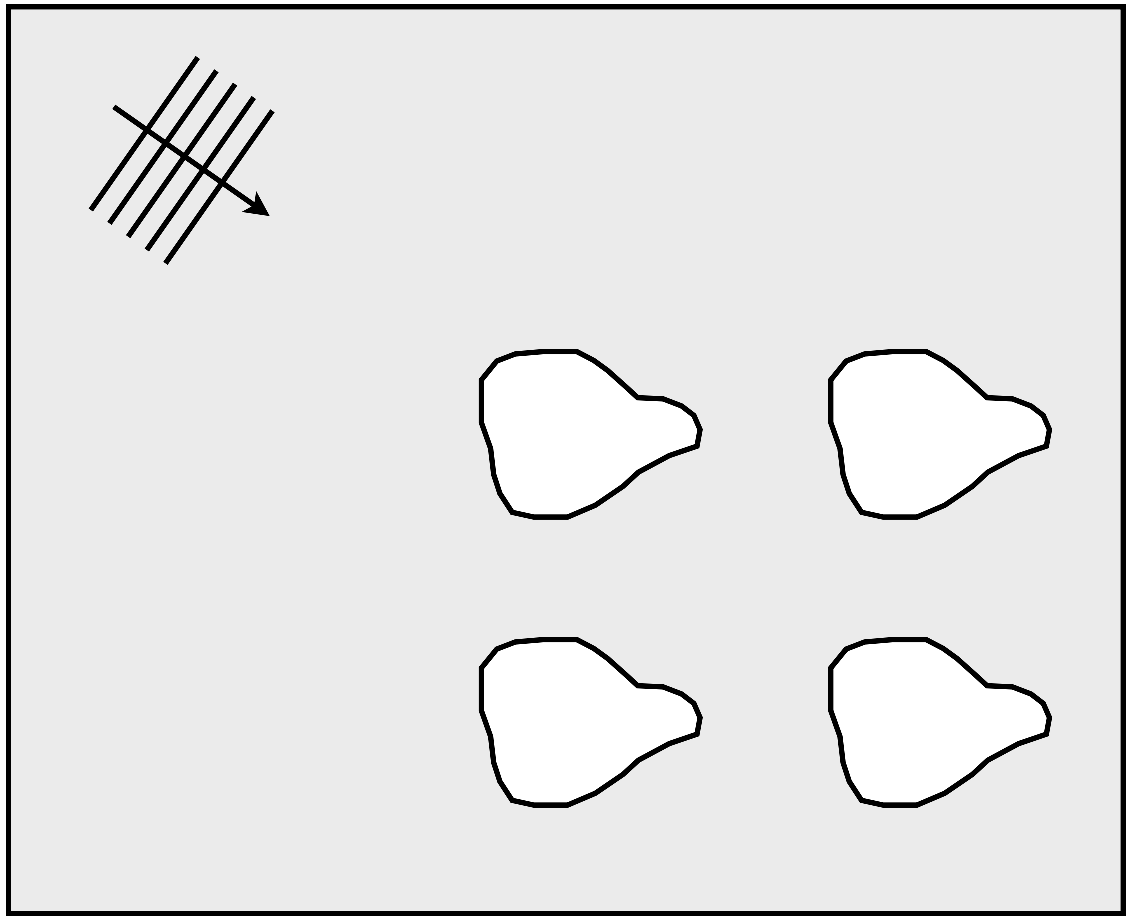
Picked parameters:



Convergence:



Current research - Multi object scattering



Strategy:
1) Train a reduced basis for each type of geometry to be accurate for all incident angles, polarizations and wavenumber.

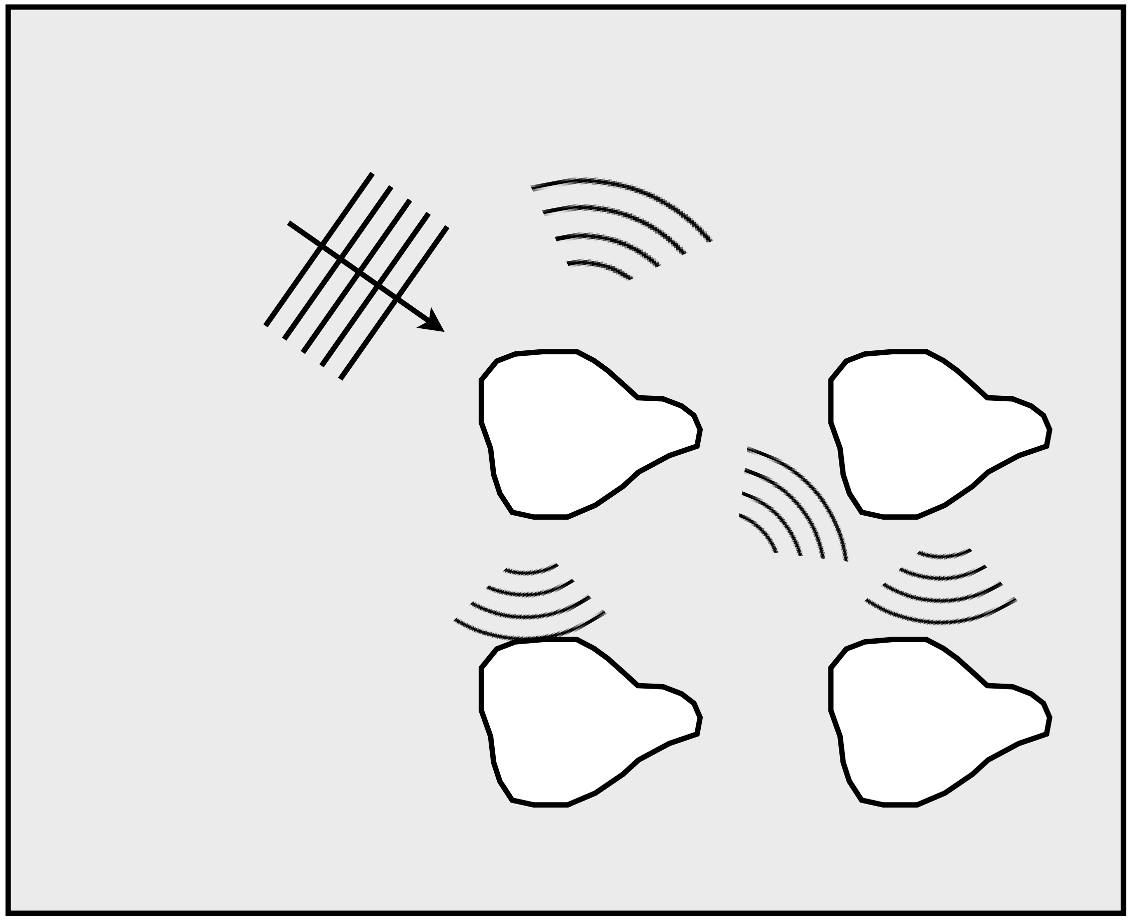
2) Approximate the interaction matrices between each pair of bodies using the EIM.
-> parametrization of the location of each object.

3) For a fixed wavenumber, incident angle and polarization we solve the problem in the reduced basis space using a Jacobi-type iteration scheme.

Remark: Observe that only for each new geometry a reduced basis needs to be assembled. The reduced basis is invariant under translation.

Example: for a lattice of 100x100 identical objects we need to assemble one reduced basis!

Current research - Multi object scattering



Strategy:
1) Train a reduced basis for each type of geometry to be accurate for all incident angles, polarizations and wavenumber.

2) Approximate the interaction matrices between each pair of bodies using the EIM.
-> parametrization of the location of each object.

3) For a fixed wavenumber, incident angle and polarization we solve the problem in the reduced basis space using a Jacobi-type iteration scheme.

Remark: Observe that only for each new geometry a reduced basis needs to be assembled. The reduced basis is invariant under translation.

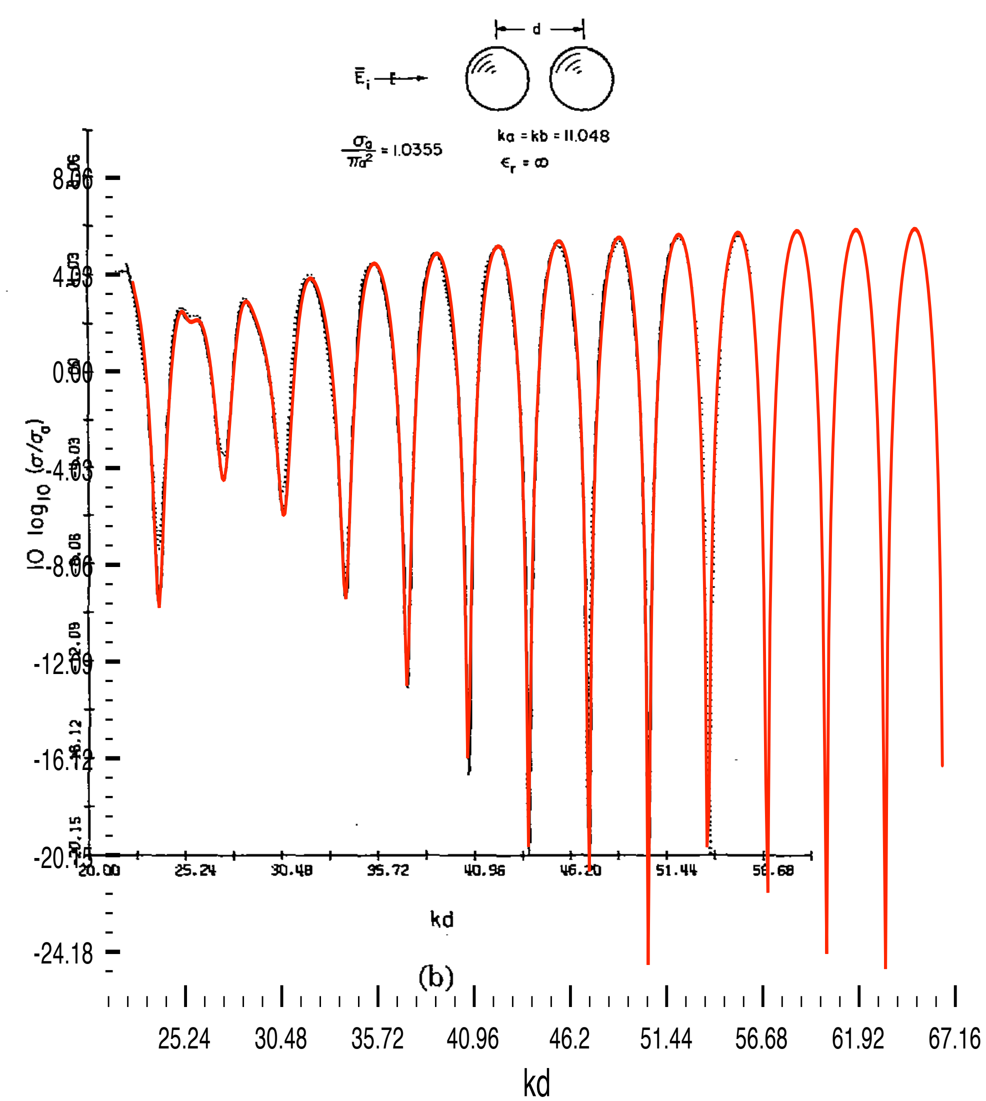
Example: for a lattice of 100x100 identical objects we need to assemble one reduced basis!

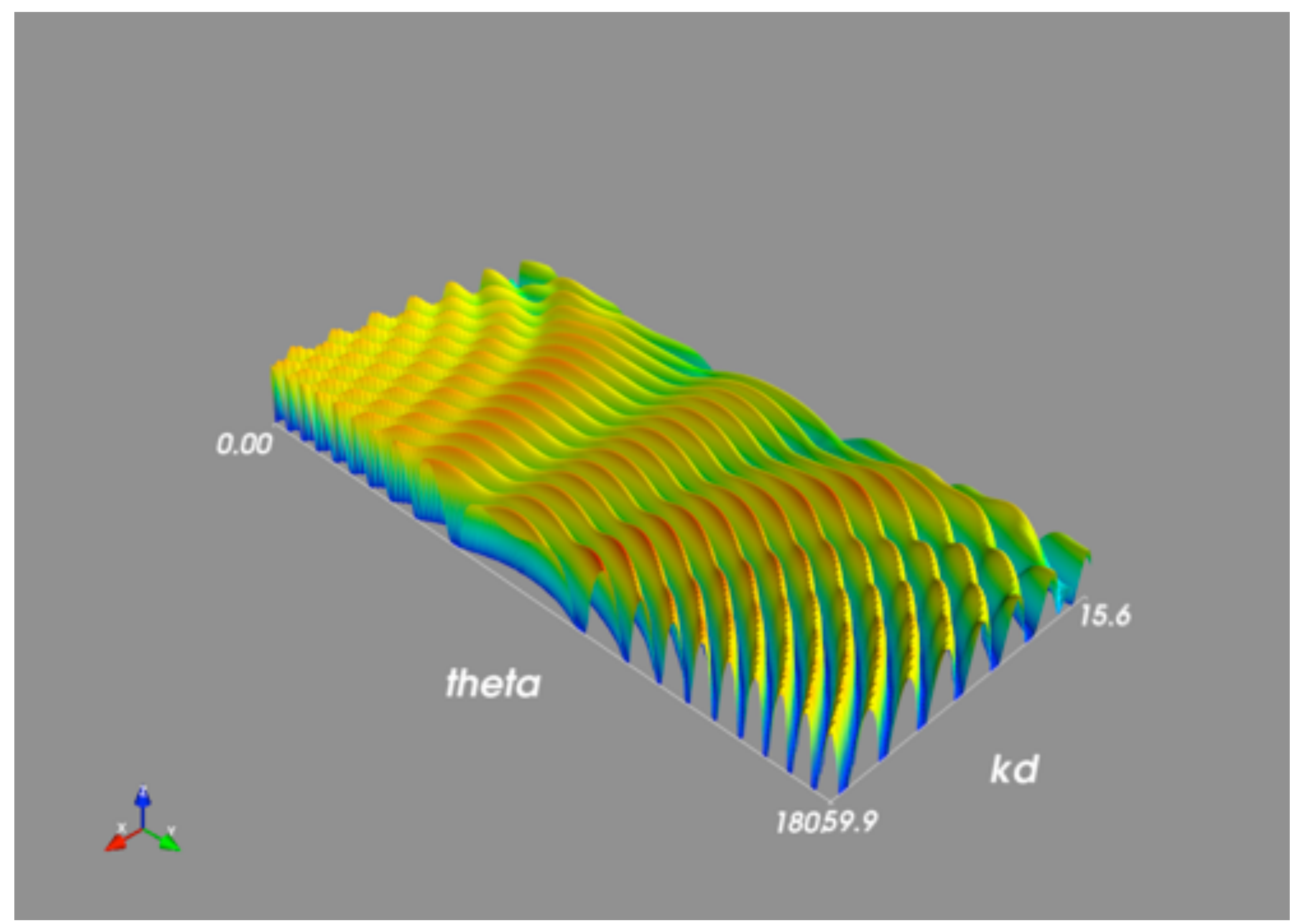
Current research - Multi object scattering

Endfire incidence for k=11.048

Benchmark available in literature

From broadside to endfire to broadsice for k=7.41

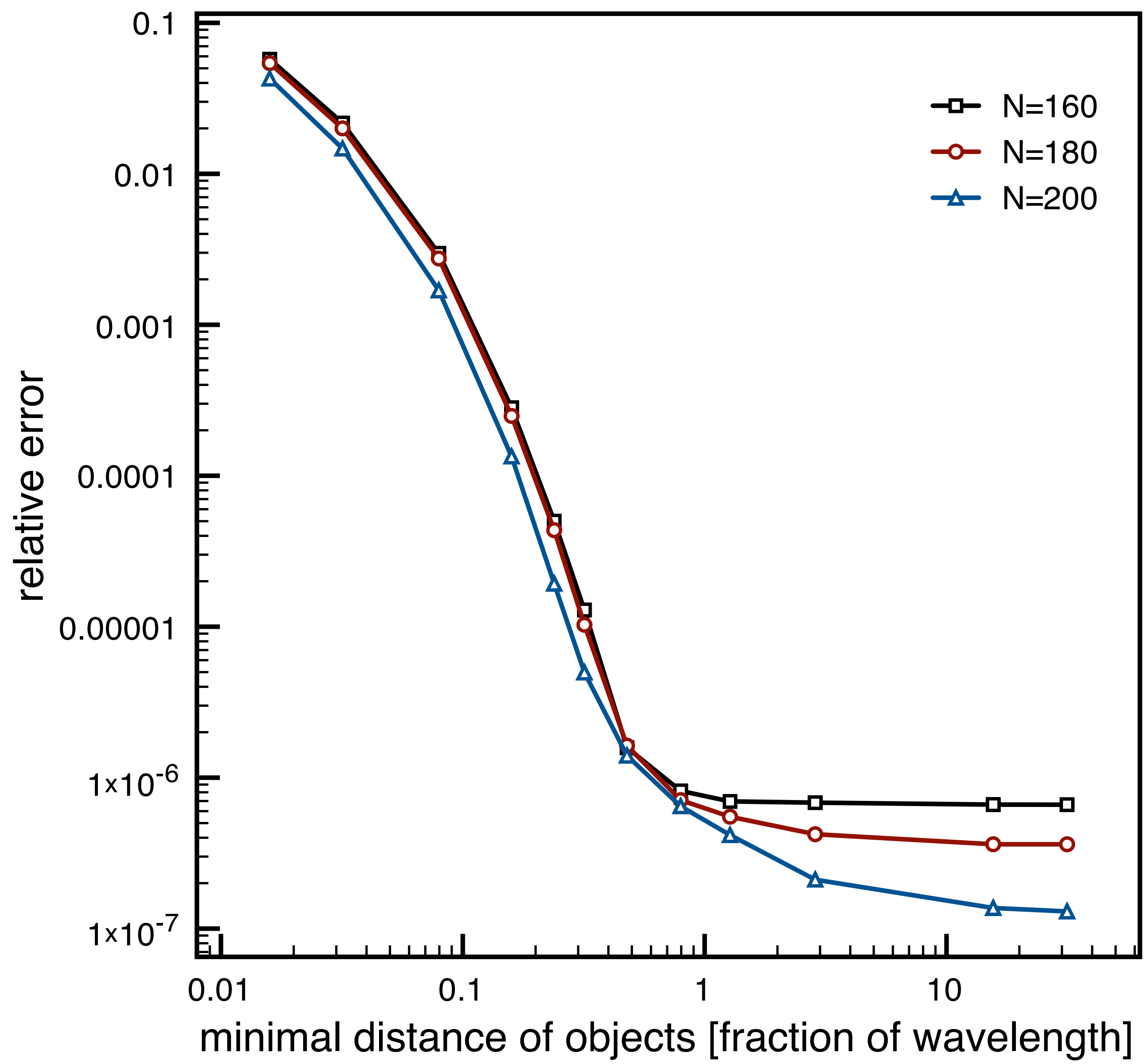




Accuracy - 2 spheres

2 spheres of variable distance:

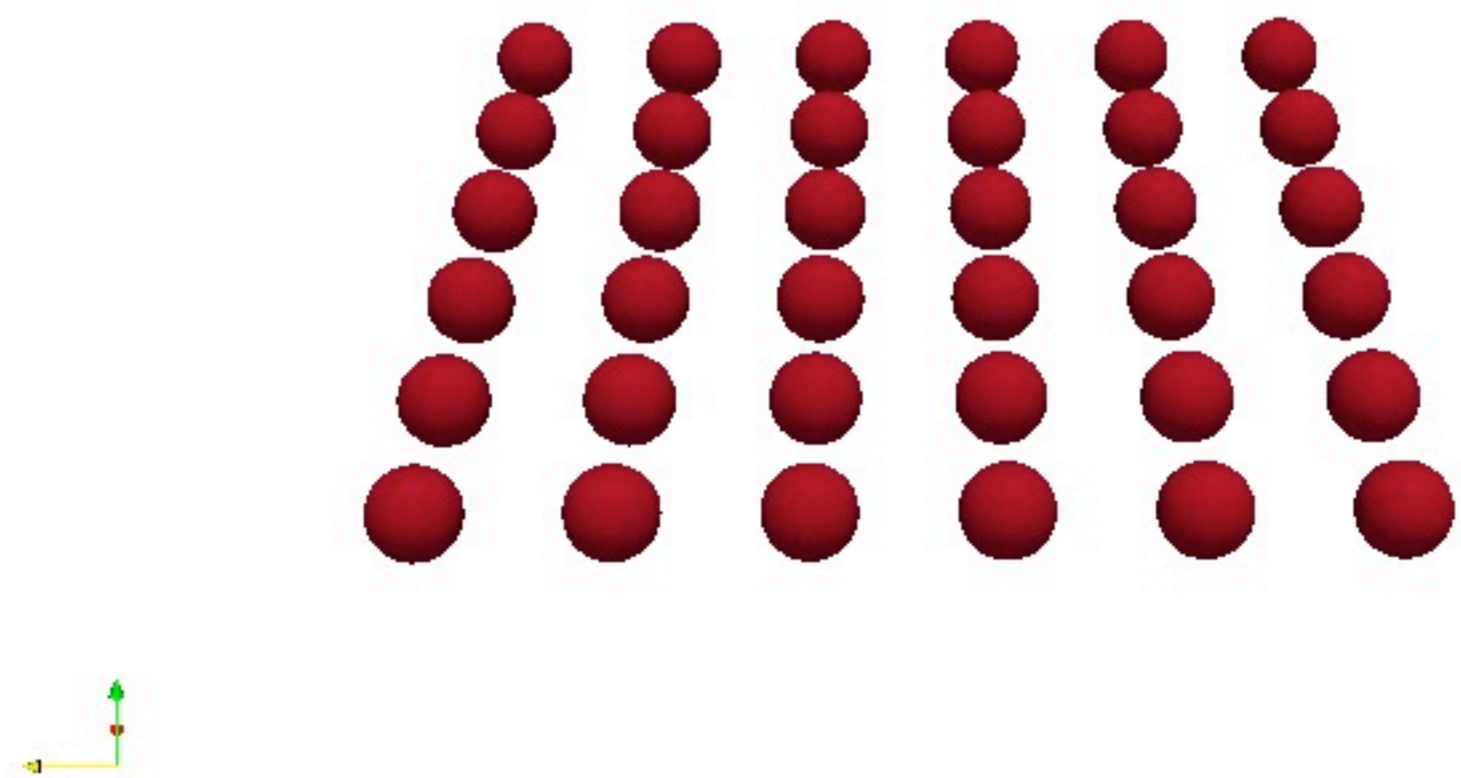
Relative error of the currents.



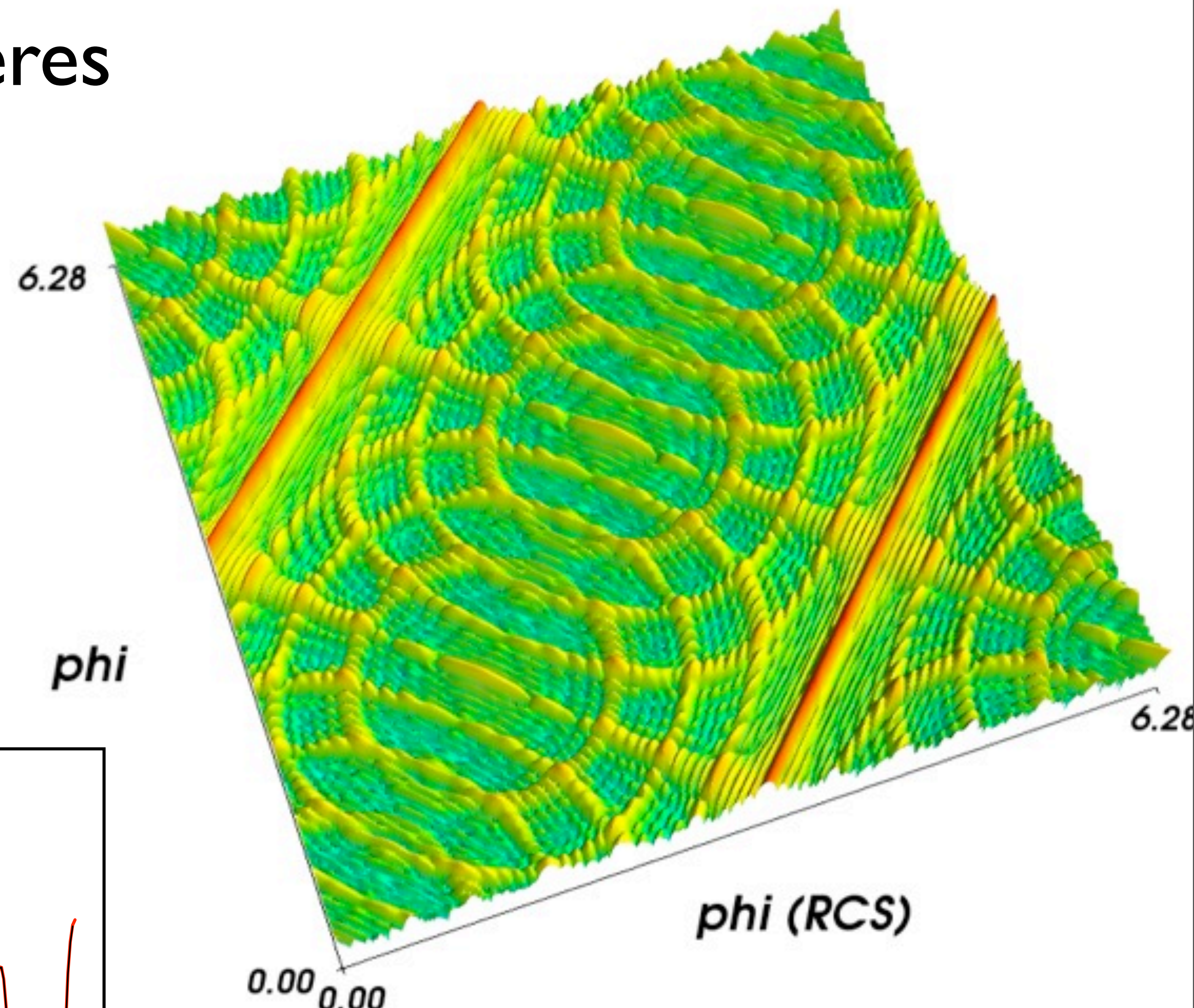
Lattice 6 x 6 spheres

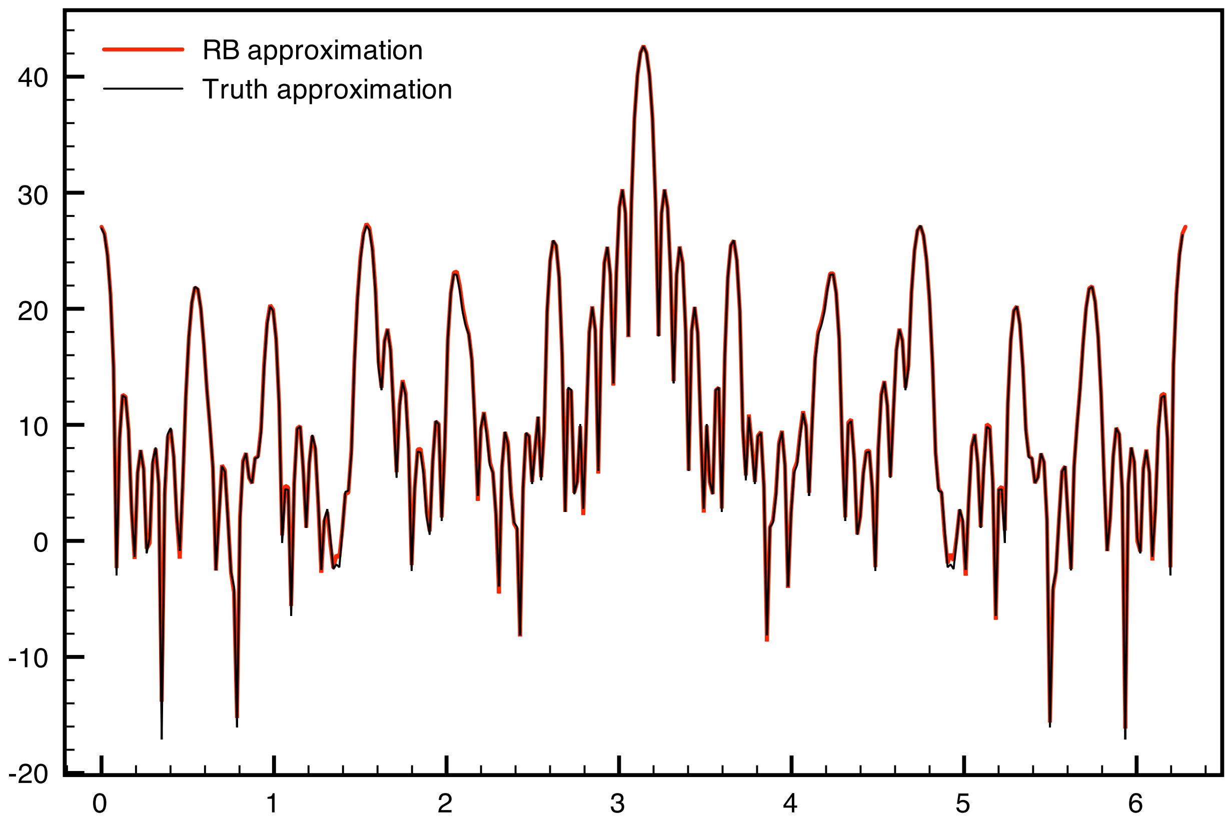
The wavenumber is fixed to $k = 3$ for the simulation. The parameter is the angle $\phi = 1, \ldots, 2\pi$ and θ is fixed at 90 degrees.

The RCS is also measured for $\phi = 0, \ldots, 2\pi$ and θ is fixed at 90 degrees.



Lattice 6 x 6 spheres

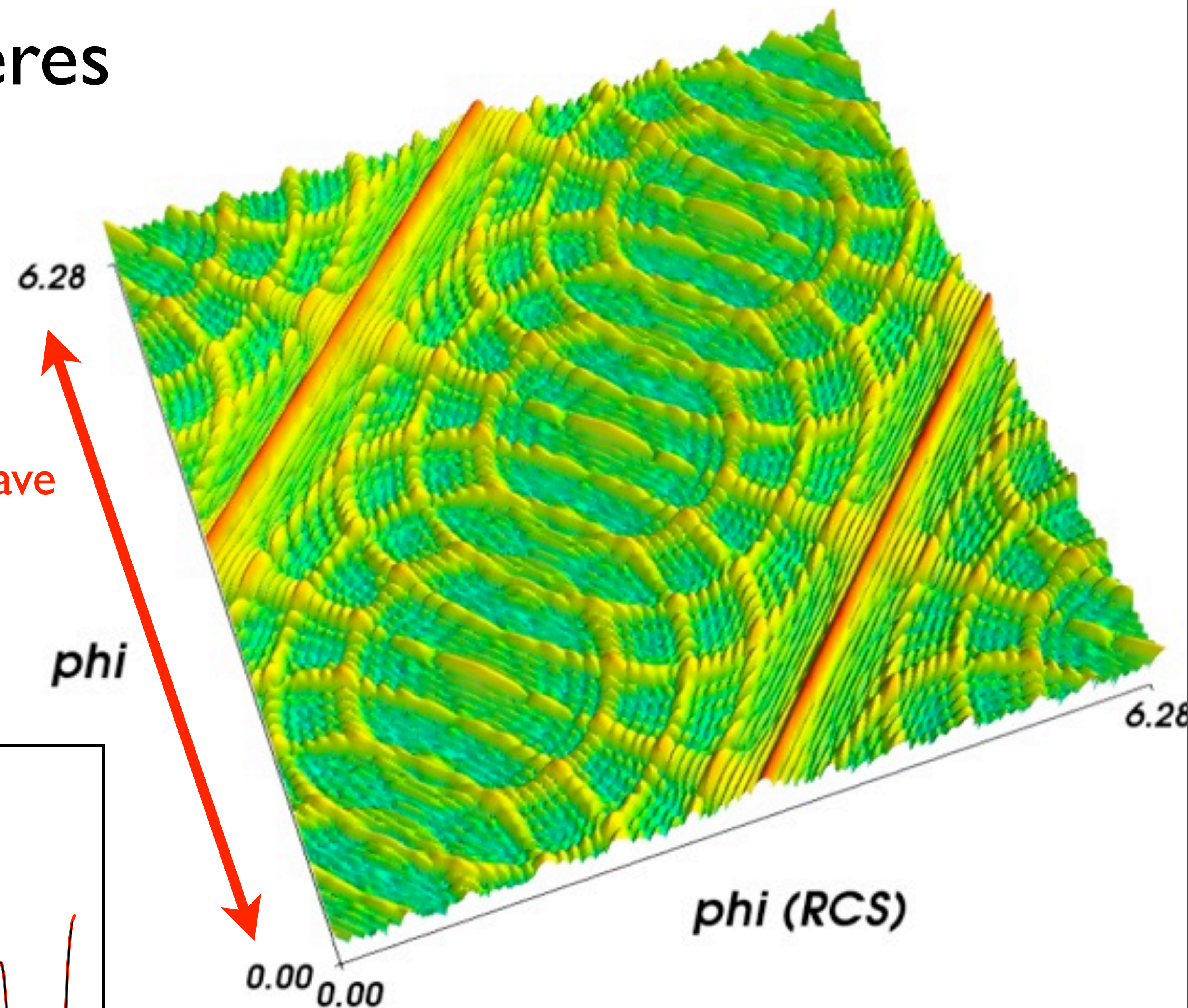


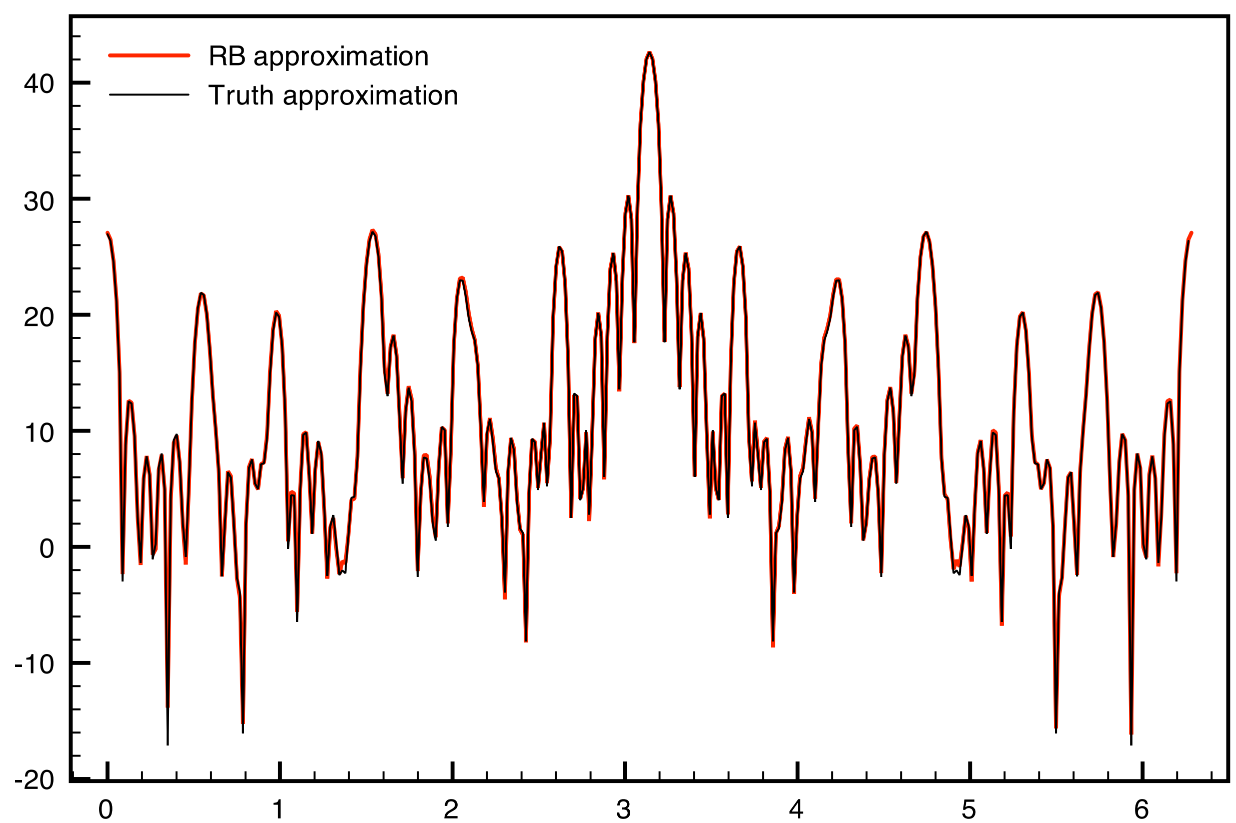


The Jacobi iterations need between 60 and 90 iterations, depending on the angle.
The total "online" simulation time for that plot was around 3-5 minutes, which involved 360 computations/different angles.

Lattice 6 x 6 spheres

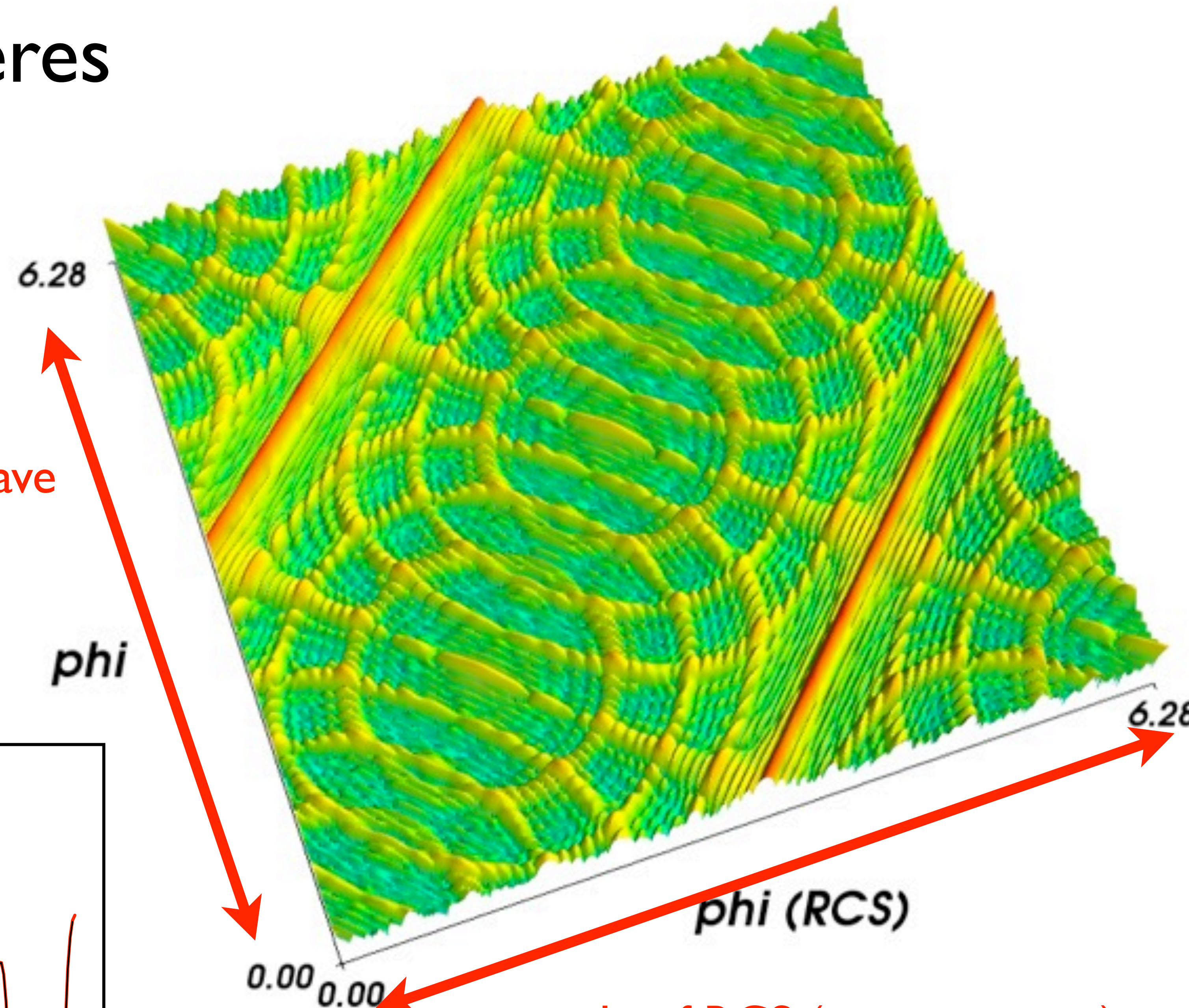
angle of incident plane wave
(input)





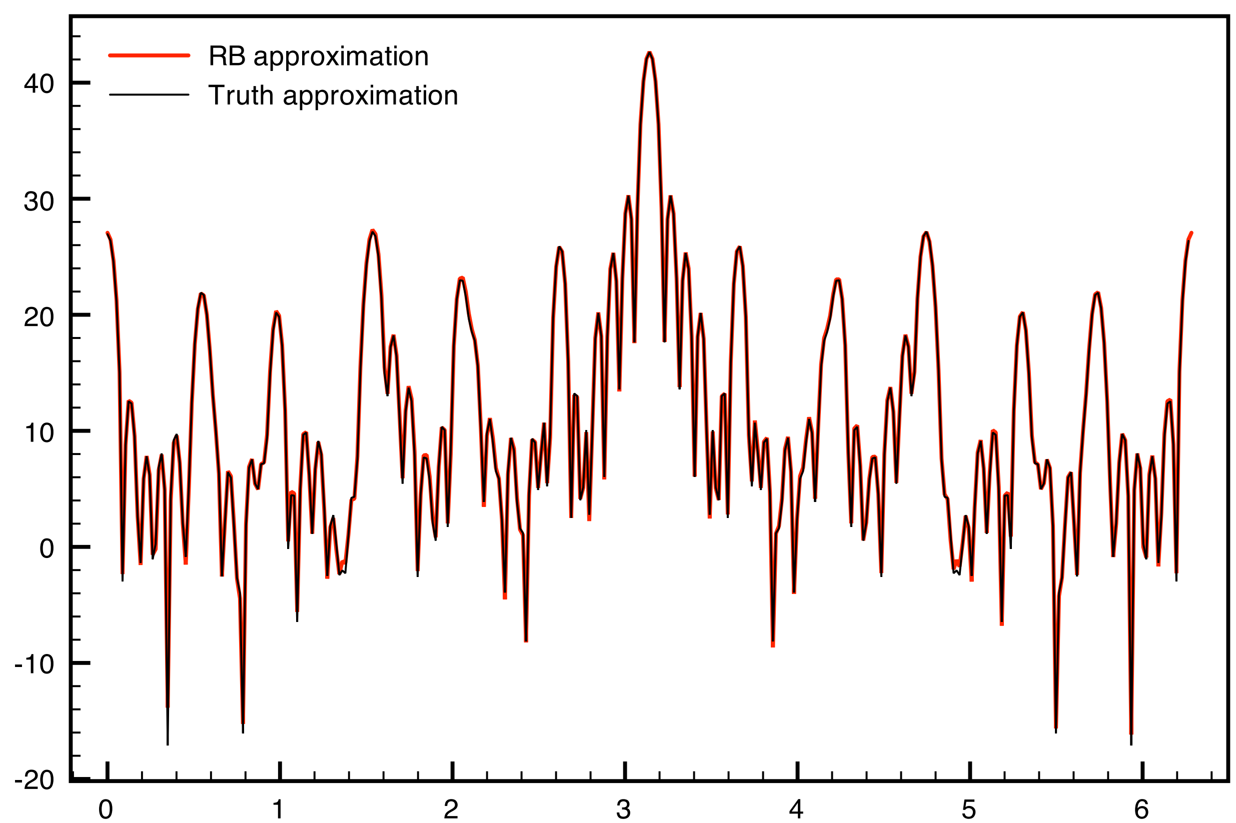
The Jacobi iterations need between 60 and 90 iterations, depending on the angle.
The total "online" simulation time for that plot was around 3-5 minutes, which involved 360 computations/different angles.

Lattice 6 x 6 spheres



angle of incident plane wave (input)

angle of RCS (measurement)



The Jacobi iterations need between 60 and 90 iterations, depending on the angle.
The total "online" simulation time for that plot was around 3-5 minutes, which involved 360 computations/different angles.

Conclusions:

- For the first time, the reduced basis method is applied to integral equations.
- EIM interpolation is an essential tool for parametrized integral equations due to the kernel function ⇨ Efficiency
- For large parameter domains EIM elements are used to speed up the computation of the “online” routine

Current

- Promising initial results for multi-object scattering using RB

Future

- *hp*-RBM for large parameter domains (and dimensions)
- CFIE for wavenumber parametrization for scatterers with volume

Thank you for your attention