Parametric Problems, Uncertainty Quantification, and Model Reduction

Hermann G. Matthies

J.-B. Colliat, M. Hautefeuille, A. Keese, D. Liu, T. Srisupattarawanit, E. Zander

Institute of Scientific Computing, TU Braunschweig
Brunswick, Germany

`wire@tu-bs.de`

`http://www.wire.tu-bs.de`

Overview

1. Parameter dependent problems
2. Associated linear process
3. Factorisations/ Reparametrisations
4. Model reduction and sparse representation
5. Examples and conclusion

Parameter dependent entities

Assume $r(p)$ to be the parameter dependent
input, the properties, or the output
of some system dependent on a parameter $p \in \mathcal{P}$.

Special case: $\mathcal{P}$ is a probability space.

Here p can be a number, function, field, or similar,
e.g. a random variable (RV), a stochastic process, a random field.

The parameter space $\mathcal{P}$ may be high-dimensional.
How to re-parametrise problem?
How to approximate $p \to r(p)$ to keep computation and memory low?

Computation of $r(p)$ involves approximation.
Allow comparable error to have sparse representation.

Deterministic model, discretisation, solution

Consider operator equation, physical system modelled by A:

$$A(u) = f \qquad u \in \mathcal{U},\ f \in \mathcal{F},$$

$$\Leftrightarrow \forall v \in \mathcal{U}: \qquad \langle A(u), v \rangle = \langle f, v \rangle,$$

$\mathcal{U}$ — space of states, $\mathcal{F}$ — dual space of actions / forcings.

Solution is usually by first discretisation

$$\boldsymbol{A}(\boldsymbol{u}) = \boldsymbol{f} \qquad \boldsymbol{u} \in \mathcal{U}_N \subset \mathcal{U},\ \boldsymbol{f} \in \mathcal{F}_N \subset \mathcal{F},$$

and then (iterative) numerical solution process

$$\boldsymbol{u}_{k+1} = \boldsymbol{\Phi}(\boldsymbol{u}_k), \qquad \lim_{k \to \infty} \boldsymbol{u}_k = \boldsymbol{u}.$$

$\boldsymbol{\Phi}$ evaluates (pre-conditioned) residua $\boldsymbol{f} - \boldsymbol{A}(\boldsymbol{u}_k)$.

Model with uncertainties

With uncertainties modelled by appropriate probab. space $(\Omega, \mathbb{P}, \mathfrak{A})$:

$$A[\omega](u(\omega)) = f(\omega) \qquad \text{a.s. in } \omega \in \Omega,$$

State $u(\omega)$ is $\mathcal{U}$-valued random variable (RV),
solution is in a tensor space $\mathcal{W} = \mathcal{U} \otimes \mathcal{S}$.

Variational statement: $\forall v \in \mathcal{W}: \quad \mathbb{E}\left(\langle A[\cdot](u(\cdot)), v\rangle\right) = \mathbb{E}\left(\langle f(\cdot), v\rangle\right).$

Similarly after semi-discretisation in $\mathcal{U}$:

$$\boldsymbol{A}[\omega](\boldsymbol{u}(\omega)) = \boldsymbol{f}(\omega) \qquad \text{a.s. in } \omega \in \Omega,$$

assume $\{\boldsymbol{v}_j\}_{j=1}^N$ a basis in $\mathcal{U}_N$, then the approx. solution in $\mathcal{U}_N \otimes \mathcal{S}$

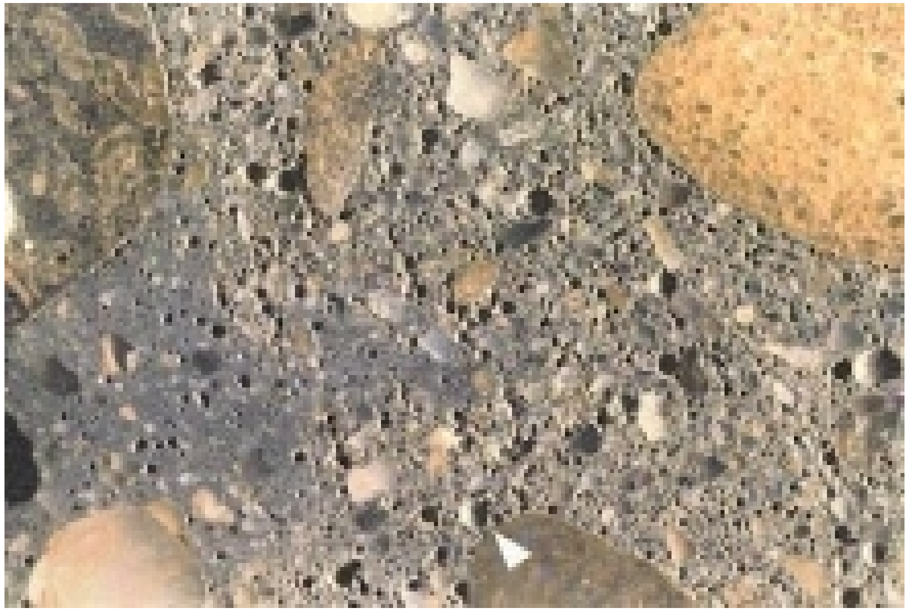
$$\boldsymbol{u}(\omega) = \sum_{j=1}^{N} u^j(\omega) \boldsymbol{v}_j.$$

Uncertainty in modelling of concrete

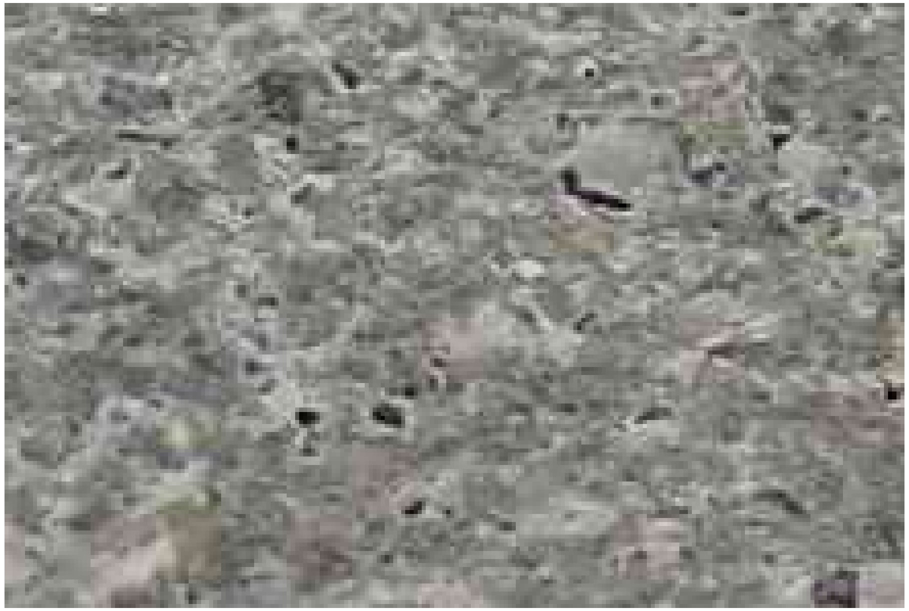
Viaduct de Millau

16×32 specimen

Scales in concrete

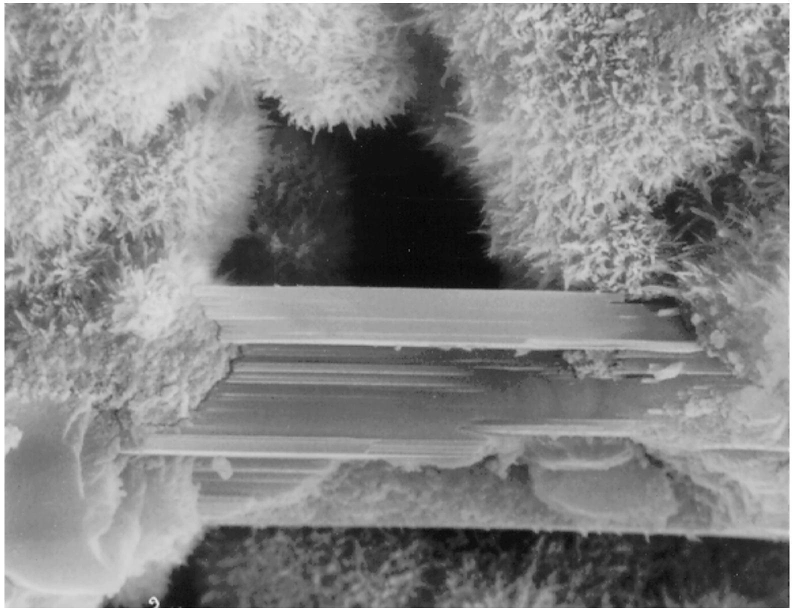


Concrete

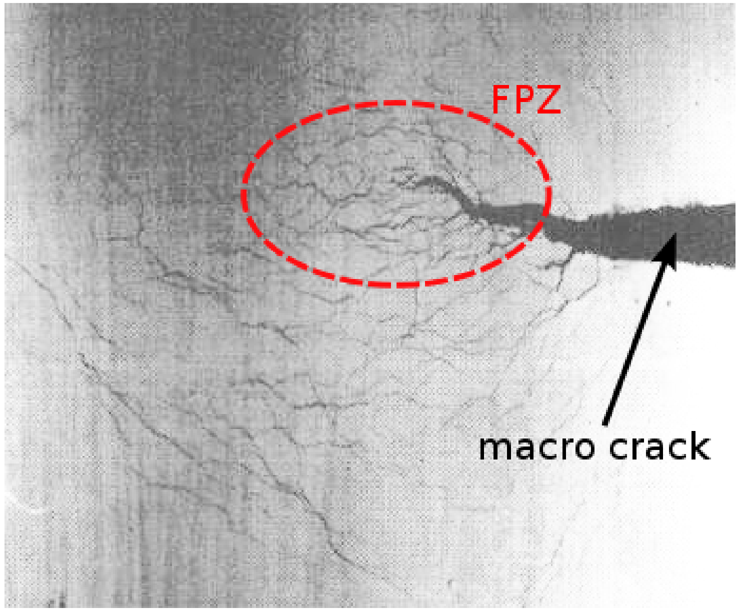


mortar

Heterogeneous on many scales



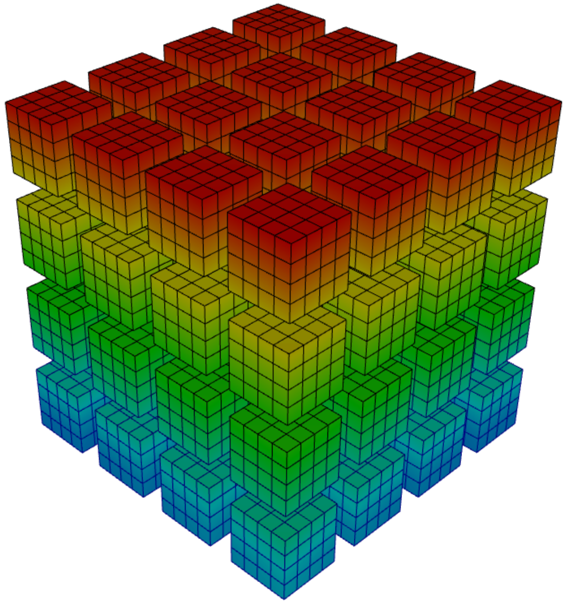
close-up



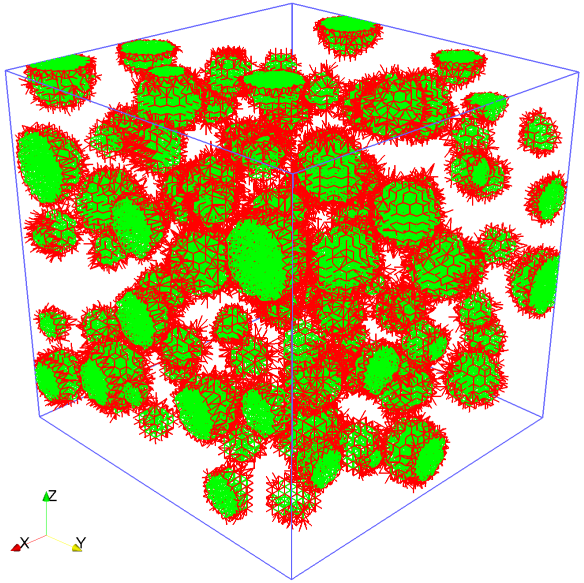
fracture process zone

Multi-scale coupling strategy

Strategy key ideas: Sub-domain wrapped in interface layer, interface layer coarse interpolation, localised Lagrange multipliers: independent dual compatibility.

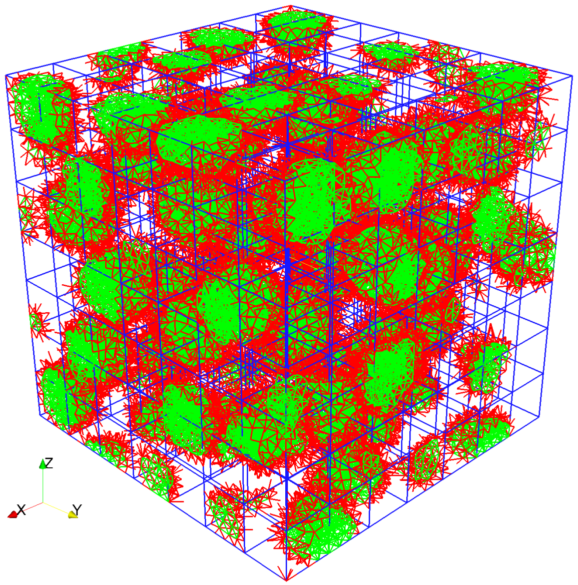


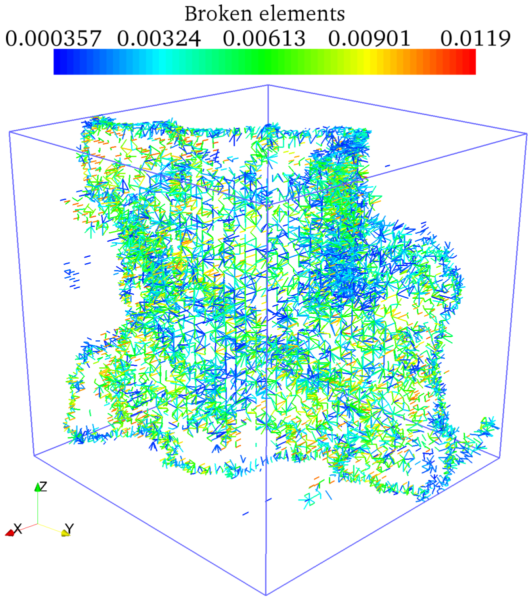
Macro-Micro Cube



One Micro-Cube with Random Inclusions

Computation





micro inclusions strain

fractured bonds

Direct integration / sampling solution

Builds on fact that ultimately a quantity of interest $\mathbb{E}\left(\Psi(u)\right)$ is wanted.

$$\mathbb{E}\left(\Psi(u)\right) = \int_{\Omega} \Psi(\omega, u(\omega))\,\mathbb{P}(\mathrm{d}\omega) \approx \sum_{z=1}^{Z} \mathrm{w}_z \Psi(\omega_z, u(\omega_z))$$

Pick (e.g. Monte Carlo) $\{\omega_z\}_{z=1}^{Z}$ points, $\forall \omega_z$ do solution process

$$\boldsymbol{u}_{k+1}(\omega_z) = \boldsymbol{\Phi}[\omega_z](\boldsymbol{u}_k(\omega_z)),$$

giving $\boldsymbol{u}(\omega_z) = \sum_j u^j(\omega_z)\boldsymbol{v}_j = \sum_j u_z^j \boldsymbol{v}_j$.

Effectively choosing $\mathcal{S}_Z \subset \mathcal{S}$, solution is in $\mathcal{W}_{N,Z} := \mathcal{U}_N \otimes \mathcal{S}_Z$.

(Usually $\boldsymbol{u}(\omega_z)$ discarded after use in integration.)

Random state represented by solution samples $[\boldsymbol{u}(\omega_0), \dots, \boldsymbol{u}(\omega_Z)]$,

or the tensor $\mathbf{u}_Z^N := \{u_z^j\}_{z=1,\dots,Z}^{j=1,\dots,N}$.

Solution by emulation / functional approximation

Emulation — replace expensive simulation by inexpensive approximation
(alias response surfaces, proxy / surrogate models, etc.)

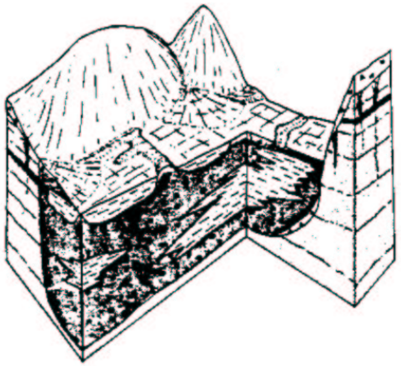
Choose subspace $\mathcal{S}_B \subset \mathcal{S}$ with basis $\{X_\beta\}_{\beta=1}^B$,
make ansatz for each $u_j(\omega) \approx \sum_\beta u_j^\beta X_\beta(\omega)$, giving

$$\boldsymbol{u}(\omega) = \sum_{j,\beta} u_j^\beta X_\beta(\omega) \boldsymbol{v}_j = \sum_{j,\beta} u_j^\beta X_\beta(\omega) \otimes \boldsymbol{v}_j .$$

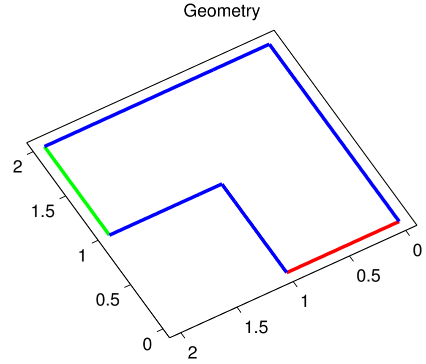
Solution is in tensor product $\mathcal{W}_{N,B} := \mathcal{U}_N \otimes \mathcal{S}_B \subset \mathcal{U} \otimes \mathcal{S} = \mathcal{W}$.

Random state $\boldsymbol{u}(\omega)$ represented by tensor $\mathbf{u}_N^B := \{u_j^\beta\}_{j=1,\dots,N}^{\beta=1,\dots,B}$,
computed by sampling (pre-conditioned) residua $\boldsymbol{f}(\omega) - \boldsymbol{A}[\omega](\boldsymbol{u}_k(\omega))$.

Model problem with uncertainties



Aquifer



2D model domain $\mathcal{G}$

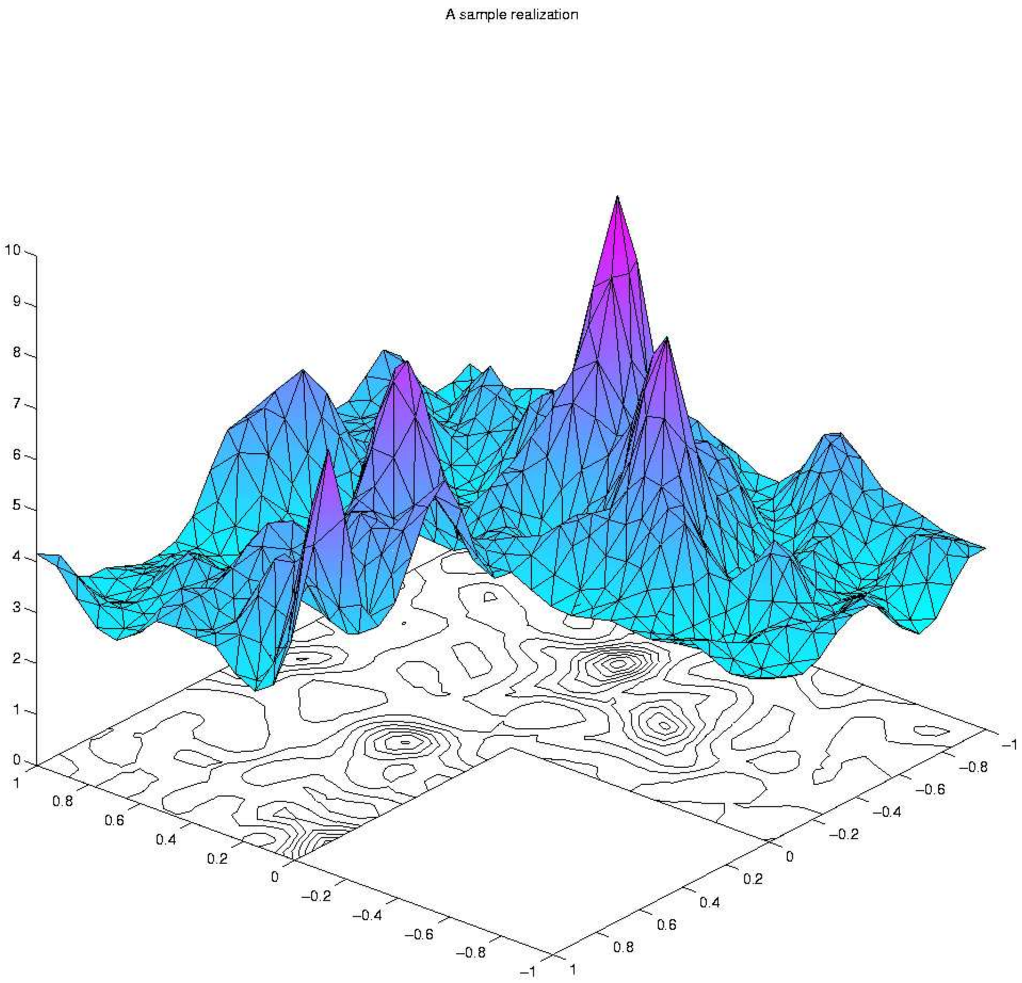
Simple stationary model of groundwater flow with uncertain data

$$-\nabla \cdot (\kappa(x,\omega)\nabla u(x,\omega)) = f(x,\omega) \qquad x \in \mathcal{G} \subset \mathbb{R}^d \qquad \text{\& b.c.}$$

Probabilistic modelling of uncertainty.

Parameter $\omega \in \Omega$ (realisations of κ and f) is in a probability space.

Realisation of $\kappa(x,\omega)$ — β-distributed



Stochastic PDE and variational form

Solution $u(x,\omega)$ is sought in tensor product space

$$\mathcal{W} := \mathcal{U} \otimes \mathcal{S} = \mathring{H}^1(\mathcal{G}) \otimes L_2(\Omega).$$

Variational formulation: find $u \in \mathcal{W}$ such that $\forall\ v \in \mathcal{W}$:

$$\mathsf{a}(v,u) := \mathbb{E}\left(\int_{\mathcal{G}} \nabla v(x,\omega) \cdot \kappa(x,\omega) \cdot u(x,\omega)\,\mathrm{d}x \right)$$

$$= \mathbb{E}\left(\int_{\mathcal{G}} v(x,\omega) f(x,\omega)\,\mathrm{d}x \right) =: \langle\!\langle v, f \rangle\!\rangle.$$

Lax-Milgram lemma → well-posedness.

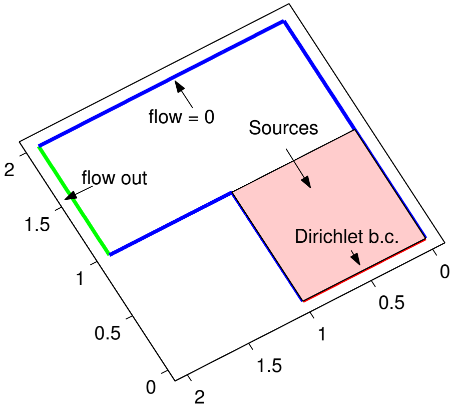
Galerkin discretisation on $\mathcal{W}_{B,N} = \mathcal{U}_N \otimes \mathcal{S}_B \subset \mathcal{U} \otimes \mathcal{S} = \mathcal{W}$ leads to

$$\mathbf{A}\,\mathbf{u} = \left(\sum_{m=1}^{M} \xi_m \boldsymbol{A}_m \otimes \boldsymbol{\Delta}^{(m)} \right) \mathbf{u} = \mathbf{f}.$$

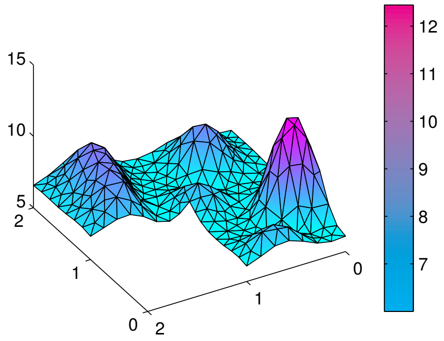
Céa's lemma → Galerkin converges.

Example solution

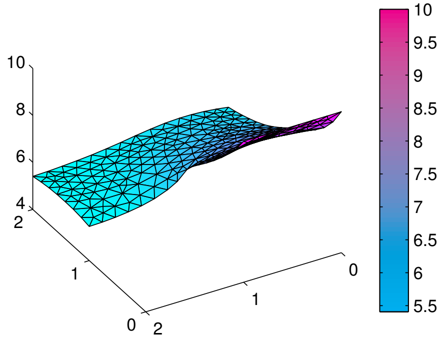
Geometry



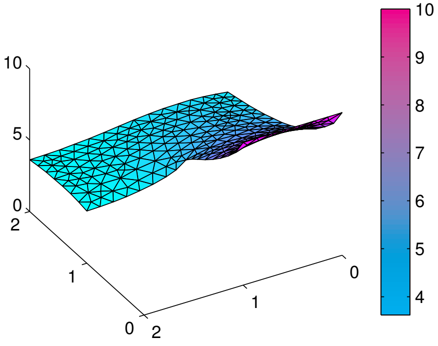
Realization of κ



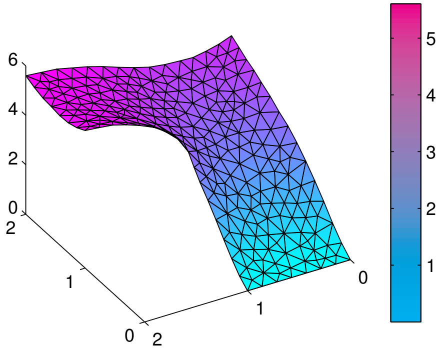
Realization of solution

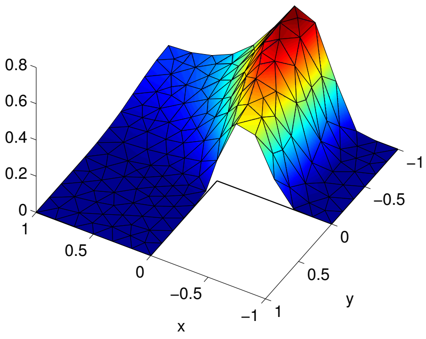


Mean of solution



Variance of solution





$\Pr\{u(x) > 8\}$

Parametric problems

For each p in a parameter set $\mathcal{P}$, let $r(p)$ be a 'solution' to some problem in a Hilbert space $\mathcal{V}$ (for simplicity).

With $r : \mathcal{P} \to \mathcal{V}$, denote $\mathcal{U} = \overline{\operatorname{span}}\, r(\mathcal{P}) = \overline{\operatorname{span}}\, \operatorname{im} r$.

What we are after: other representations of r or $\mathcal{U} = \overline{\operatorname{span}}\, \operatorname{im} r$.

To each function $r : \mathcal{P} \to \mathcal{U}$ corresponds a linear map $R : \mathcal{U} \to \tilde{\mathcal{R}}$:

$$R : \mathcal{U} \ni u \;\mapsto\; \langle r(\cdot) | u \rangle_{\mathcal{V}} \in \tilde{\mathcal{R}} = \operatorname{im} R \subset \mathbb{R}^{\mathcal{P}}.$$

By construction R is injective. Use this to make $\tilde{\mathcal{R}}$ a pre-Hilbert space:

$$\forall \phi, \psi \in \tilde{\mathcal{R}} : \; \langle \phi | \psi \rangle_{\mathcal{R}} := \langle R^{-1}\phi | R^{-1}\psi \rangle_{\mathcal{U}}.$$

R^{-1} is unitary on completion $\mathcal{R}$.

RKHS and classification

$\mathcal{R}$ is a reproducing kernel Hilbert space —RKHS— with kernel

$$\varkappa(p_1, p_2) = \langle r(p_1) | r(p_2) \rangle_{\mathcal{U}} \in \mathbb{R}^{\mathcal{P} \times \mathcal{P}}$$

Reproducing property:

$$\forall \phi \in \mathcal{R} : \; \langle \varkappa(p, \cdot) | \phi(\cdot) \rangle_{\mathcal{R}} = \phi(p).$$

In other settings (classification, machine learning, SVM), when different subsets of $\mathcal{P}$ have to be classified, the space $\mathcal{U}$ and the map $r : \mathcal{P} \to \mathcal{U}$ is not given, but can be freely chosen.

It is then called the feature map.
The whole procedure is called the kernel trick.

‘Correlation’

If there is another inner product $\langle\cdot|\cdot\rangle_{\mathcal{Q}}$ on a subspace $\mathcal{Q} \subset \mathbb{R}^{\mathcal{P}}$, (e.g. if $(\mathcal{P}, \mu)$ is a measure space define $\mathcal{Q} := L_2(\mathcal{P}, \mu)$), a linear map C may be defined in $\mathcal{U}$ by

$$\forall u, v \in \mathcal{U};\; \langle Cu, v\rangle_{\mathcal{U}'\times\mathcal{U}} = \langle Ru|Rv\rangle_{\mathcal{Q}}.$$

C is the ‘correlation’ operator (adjoint in $\mathcal{Q} = L_2(\mathcal{P})$):

$$C := R^* R = \int_{\mathcal{P}} r(p) \otimes r(p)\, \mu(\mathrm{d}p)$$

is self-adjoint and positive semi-definite → has spectrum $\sigma(C) \subseteq \mathbb{R}_+$.

Spectral decomposition with projectors E_λ

$$Cu = \int_0^\infty \lambda\, \mathrm{d}E_\lambda u = \sum_{\lambda_j \in \sigma_p(C) \subset \mathbb{R}_+} \lambda_j \langle v_j|u\rangle_{\mathcal{U}}\, v_j + \int_{\mathbb{R}_+ \setminus \sigma_p(C)} \lambda\, \mathrm{d}E_\lambda u.$$

Spectral decomposition

Often C has a pure point spectrum (e.g. C compact)
$\Rightarrow$ last integral vanishes, i.e. $\sigma(C) = \sigma_p(C)$:

$$Cu = \sum_j \lambda_j \sum_k^{\text{mult.}\lambda_j} \langle v_j^k | u \rangle_{\mathcal{U}} \, v_j^k = \sum_{\lambda_j \in \sigma_p(C)} \lambda_j \sum_k^{\text{mult.}\lambda_j} \left(v_j^k \otimes v_j^k \right) u.$$

If $\sigma(C) \neq \sigma_p(C)$: generalised eigenvectors v_λ and Gelfand triplets (rigged Hilbert spaces) for the continuous spectrum:

$$\int_{\mathbb{R}_+ \setminus \sigma_p(C)} \lambda \, \mathrm{d}E_\lambda u = \sum_k^{\text{mult.}} \int_{\mathbb{R}_+} \lambda \, \left(v_\lambda^k \otimes v_\lambda^k \right) u \, \varrho_k(\mathrm{d}\lambda).$$

Representation as sum / integral of rank-1 operators.
Numerical approximation will give a sum. Assumed from now on.

Singular value decomposition

C unitarily equivalent to multiplication operator M_k, with $k \geq 0$:

$$C = V M_k V^* = (V M_k^{1/2})(V M_k^{1/2})^*, \text{ with } M_k^{1/2} = M_{\sqrt{k}}.$$

$$(M_k f(\zeta) := k(\zeta) f(\zeta))$$

This connects to the singular value decomposition (SVD) of $R = S M_k^{1/2} V^*$, with a (here) unitary S.

With $\sqrt{\lambda_m}\, s_m := R v_m \in \mathcal{R}$:

$$(Ru)(p) = \langle r(p)|u\rangle_{\mathcal{U}} = \sum_m \sqrt{\lambda_m}\, \langle v_m|u\rangle_{\mathcal{U}}\, s_m(p),$$

$$R = \sum_m \sqrt{\lambda_m}\, (v_m \otimes s_m).$$

Model reduction possible by truncating the sum.

Model reduction

For partly continuous spectrum we get

$$r(p) = \sum_{k}^{\text{mult.}} \int_{\mathbb{R}_+} \sqrt{\lambda}\,\langle v_\lambda^k, u\rangle\, s_\lambda^k(p)\, \varrho_k(\mathrm{d}\lambda)$$

With approximation or only point spectrum

$$r(p) = \sum_m \sqrt{\lambda_m}\, s_m(p) v_m, \quad r \in \mathcal{U} \otimes \mathcal{Q}.$$

This is the Karhunen-Loève-expansion, due to the SVD.
A sum of rank-1 operators / tensors.

Observe that r is linear in the “coordinates” s_m, and also $\varsigma := \sqrt{\lambda}$.

A representation of r, model reduction possible by truncation of sum.

Kernel spectral decomposition

For $\phi, \psi \in L_2(\mathcal{P})$ we have also

$$\langle R^*\phi | R^*\psi\rangle_{\mathcal{U}} = \iint_{\mathcal{P}\times\mathcal{P}} \phi(p_1)\,\varkappa(p_1,p_2)\,\psi(p_2)\;\mu(\mathrm{d}p_1)\,\mu(\mathrm{d}p_2).$$

To compute R^*, define an operator $\hat{C} = RR^*$ on $L_2(\mathcal{P})$ by

$$(\hat{C}\phi)(p_1) := \int_{\mathcal{P}} \varkappa(p_1,p_2)\,\phi(p_2)\,\mu(\mathrm{d}p_2) = \langle \varkappa(p_1,\cdot)|\phi\rangle_{L_2(\mathcal{P})}.$$

Eigenvalue problem for $\hat{C}$ gives (Mercer's theorem)

$$\varkappa(p_1,p_2) = \sum_m \lambda_m\, s_m(p_1) s_m(p_2),$$

$\{s_m\}$ is CONS in $L_2(\mathcal{P})$, $\{\sqrt{\lambda_m}\, s_m\}$ is CONS in $\mathcal{R}$.

$$R^*\phi = \sum_m \sqrt{\lambda_m}\; v_m \langle s_m|\phi\rangle_{L_2(\mathcal{P})}, \quad R^{-1}\phi = \sum_m \lambda_m^{-1/2}\; v_m \langle s_m|\phi\rangle_{L_2(\mathcal{P})}.$$

Factorisations

R^* may serve as a representation. This is a factorisation of C.
Let $C = B^*B$ be an arbitrary one. Some possible ones:

$$C = R^*R = (VM_k^{1/2})(VM_k^{1/2})^* = C^{1/2}C^{1/2} = B^*B.$$

Each factorisation leads to a representation—all unitarily equivalent.
When C is a matrix, a favourite is Cholesky: $C = LL^*$).

Assume that $C = B^*B$ and $B : \mathcal{U} \to \mathcal{H}$, let $\{e_k\}$ be CONS in $\mathcal{H}$.

Unitary $Q : \ell_2 \ni \boldsymbol{a} = (a_1, \dots, a_n, \dots) \mapsto \sum_k a_k e_k \in \mathcal{H}$.

Let $\tilde{r}(\boldsymbol{a}) := B^*Q\boldsymbol{a} := \tilde{R}^*\boldsymbol{a}$, i.e. $\tilde{R}^* : \ell_2 \to \mathcal{U}$. Then

$$\tilde{R}^*\tilde{R} = (B^*Q)(Q^*B) = B^*B = C.$$

Integral decompositions

More decompositions and representations possible via $\hat{C}$.

Let $\varkappa(\omega_1, \omega_2) = \int_Y g(\omega_1, y) g(\omega_2, y)\, \nu(\mathrm{d}y)$.

Set $p(y) = R^{-1}(g(\cdot, y))$ to give

$$R^*\phi = \int_Y p(y) \langle g(\cdot, y) | \phi \rangle_{L_2(\Omega)}\, \nu(\mathrm{d}y) = \int_Y p(y) (G\phi)(y)\, \nu(\mathrm{d}y),$$

where $(G\phi)(y) = \int_\Omega g(\omega, y) \phi(\omega)\, \mu(\mathrm{d}\omega)$ is an integral transform.

We can arrange $\mathcal{U} = \overline{\mathrm{span}}\, \mathrm{im}\, r = \overline{\mathrm{span}}\, \mathrm{im}\, p$.

Then $p(y)$ gives a representation over Y: for $f \in L_2(Y, \nu)$

$$\hat{R}^* f = \int_Y p(y) f(y)\, \nu(\mathrm{d}y).$$

Representations

We have seen several ways to represent the solution space by a—hopefully—simpler space.
These can all be used for model reduction, choosing a smaller subspace.

- The RKHS together with R^{-1}.
- The spectral decomposition over $\sigma(C)$ or via $V M_k^{1/2}$.
- The Karhunen-Loève expansion based on SVD via R^*.
- Other multiplicative decompositions, such as $C = B^* B$.
- The kernel decompositions and representation on $L_2(Y, \nu)$.

Choice depends on what is wanted / needed.

Examples and interpretations

- If $\mathcal{V}$ is a space of centred RVs, r is a random field / stochastic process indexed by $\mathcal{P}$, kernel $\varkappa(p_1, p_2)$ is covariance function.

- If in this case $\mathcal{P} = \mathbb{R}^d$ and moreover $\varkappa(p_1, p_2) = c(p_1 - p_2)$ (stationary process / homogeneous field), then diagonalisation V is real Fourier transform, typically $\sigma_p(C) = \emptyset$ $\Rightarrow$ need Gelfand triplets.

- If μ is a probability measure on $\mathcal{P} = \Omega$ $(\mu(\Omega) = 1)$, and r is a centred $\mathcal{V}$-valued RV, then C is the covariance.

- If $\mathcal{P} = \{1, 2, \dots, n\}$ and $\mathcal{R} = \mathbb{R}^n$, then $\varkappa$ is the Gram matrix of the vectors $r_1, \dots, r_n$.

- If $\mathcal{P} = [0, T]$ and $r(t)$ is the response of a dynamical system, then R^* leads to proper orthogonal decomposition (POD).

Further factorisation

We have found representations in $\mathcal{U} \otimes \mathcal{S}$, where

$$\mathcal{S} = \mathcal{R}, L_2(\mathcal{P}), L_2(\sigma(C)), \bigoplus_k L_2(\mathbb{R}, \mu_k), \ell_2, L_2(Y), \dots$$

Combinations may occur, so that $\mathcal{S} = \mathcal{S}_I \otimes \mathcal{S}_{II} \otimes \mathcal{S}_{III} \otimes \dots$

This was only a basic decomposition.

Often the problem allows $\mathcal{U} = \bigotimes_k \mathcal{U}_k$.

Or the parameters allow $\mathcal{S} = \bigotimes_j \mathcal{S}_j$.

In case of random fields / stochastic processes

$$\mathcal{S} = L_2(\Omega) \cong \bigotimes_j L_2(\Omega_j) \cong L_2(\mathbb{R}^{\mathbb{N}}, \Gamma) \cong \bigotimes_{k=1}^{\infty} L_2(\mathbb{R}, \Gamma_1) \dots$$

So $\mathcal{U} \otimes \mathcal{S} \cong \left(\bigotimes_j \mathcal{U}_j\right) \otimes (\bigotimes_k \mathcal{S}_{I,k}) \otimes (\bigotimes_m \mathcal{S}_{II,m}) \otimes \dots$

Discretisation — model reduction

On continuous level discretisation is choice of subspace

$$\mathcal{W}_{N,B} := \mathcal{U}_N \otimes \mathcal{S}_B \subset \mathcal{U} \otimes \mathcal{S} =: \mathcal{W}$$

and—important for computation—good basis in it.

On discrete level reduced models find sub-manifold $\mathcal{W}_R \subset \mathcal{W}_{N,B}$ with smaller dimensionality $\dim \mathcal{W}_R = R \ll N \times B = \dim \mathcal{W}_{N,B}$.

They can work on $\mathcal{S}_B$ or $\mathcal{U}_N$, or both.

Different approaches to choose reduced model:

- Before the solution process (e.g. modal projection).
- After the solution process (essentially data compression).
- During solution, computing solution and reduction simultaneously.

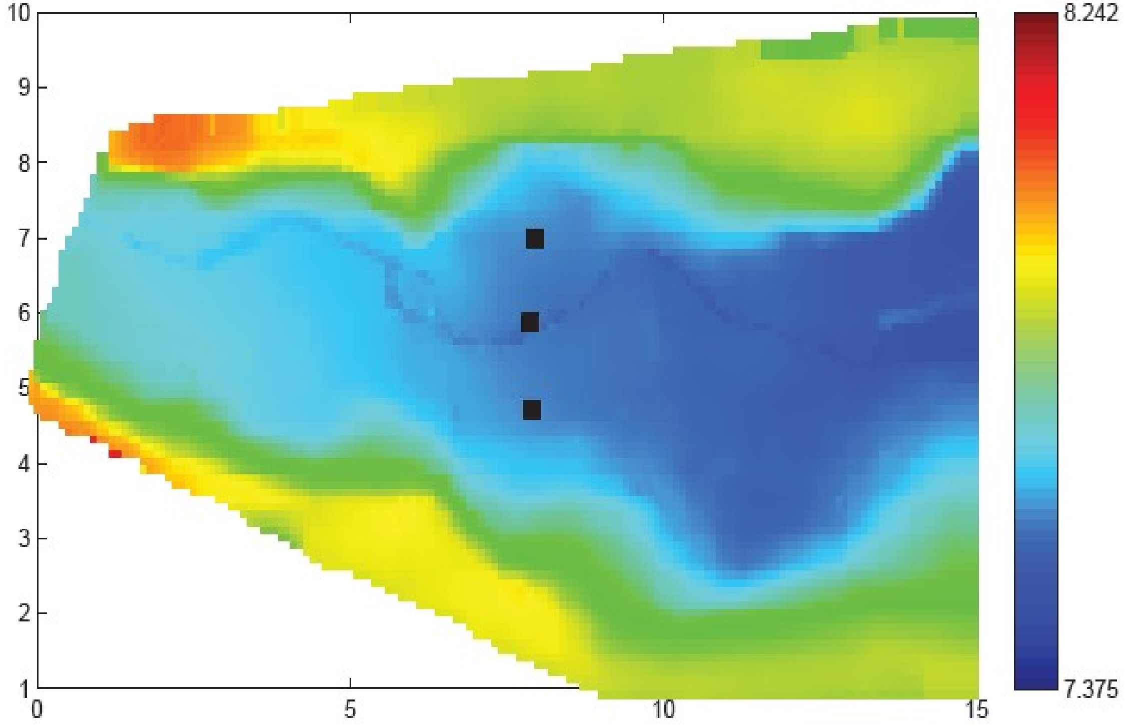
Use in time-space sampling I

Example: UQ-computations of time-dependent shallow-water flow.

1:50 Scale model in Milano of Toce river (Italy) (D. Liu)

Use in time-space sampling II

Topography in model — uncertain elevation.

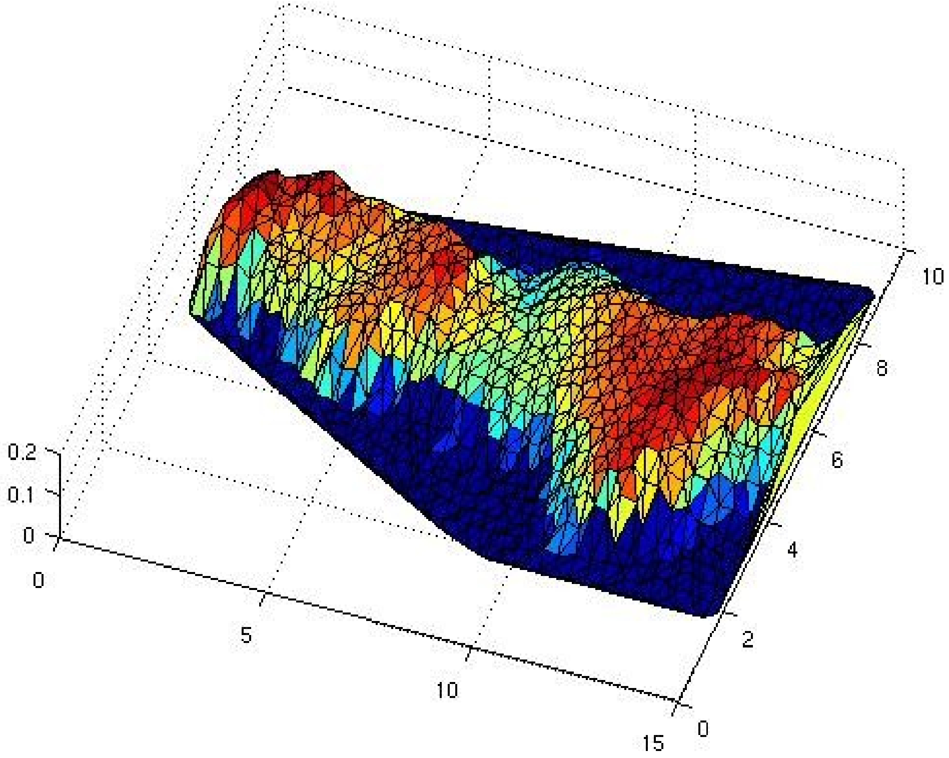


Also uncertain inflow and bed friction—Manning's coefficient.

Use in time-space sampling III

Computation with QMC-sampling

Water level with 5 % exceedance probability



Use in emulation

Solution process to obtain co-efficients for stochastic problem

$$\mathbf{u}_{k+1} = \boldsymbol{\Phi}(\mathbf{u}_k)$$

may be written as tensorised mapping

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \boldsymbol{\Xi}(\mathbf{u}_k) = \mathbf{u}_k - \left(\sum_{m=1}^{M} \boldsymbol{Y}_m \otimes \boldsymbol{G}^m \right) (\mathbf{u}_k).$$

With $\mathbf{u}_0 = \sum_{j=1}^{R_0} \boldsymbol{y}_{0,j} \otimes \boldsymbol{g}^{0,j}$, this gives

$$\mathbf{u}_1 = \sum_{j=1}^{R_0} \boldsymbol{y}_{0,j} \otimes \boldsymbol{g}^{0,j} - \sum_{m=1}^{M} \boldsymbol{Y}_m(\mathbf{u}_0) \otimes \boldsymbol{G}^m(\mathbf{u}_0).$$

Rank of $\mathbf{u}_{k+1}$ grows by M.

Possible for pre-conditioned linear iteration,

and modified-, full-, inexact- and quasi-Newton iteration.

Truncated low-rank iteration

Sparse representation entails

- reduce $\mathbf{u}_N^B := [u_j^\beta]$ to important information $\mathbf{u} \approx \mathbf{u}_N^B$,
- never store all of $\mathbf{u}_N^B$, but only $\mathbf{u}$,
- operate efficiently on sparse representation $\mathbf{u}$.

If iteration and rank-truncation $\mathbf{T}_\epsilon$ are alternated, rank stays low.

Here rank-truncation by updated SVD.

$$\hat{\mathbf{u}}_{k+1} = \sum_{j=1}^{R_k} \boldsymbol{y}_{k,j} \otimes \boldsymbol{g}^{k,j} - \sum_{m=1}^{M} \boldsymbol{Y}_m(\mathbf{u}_k) \otimes \boldsymbol{G}^m(\mathbf{u}_k),$$

$$\mathbf{u}_{k+1} = \mathbf{T}_\epsilon(\hat{\mathbf{u}}_{k+1}) \qquad \text{with} \quad \|\mathbf{T}_\epsilon(\mathbf{v}) - \mathbf{v}\| \leq \epsilon.$$

Truncated iteration converges until stagnation for linearly convergent process with contraction factor ϱ to stagnation range $\epsilon/(1-\varrho)$.

Computational complexity for linear case

Residuum is $\mathbf{f} - \mathbf{A}\,\mathbf{u}_k = \mathbf{f} - \left(\sum_{m=1}^{M} \xi_m\, \boldsymbol{A}_m \otimes \boldsymbol{\Delta}^{(m)}\right)\, \mathbf{u}_k$.

Computation on full $\mathbf{u}_k$ needs
$M \times B$ $\boldsymbol{A}$-multiplications $+$ $M \times N$ $\boldsymbol{\Delta}$-multiplications.

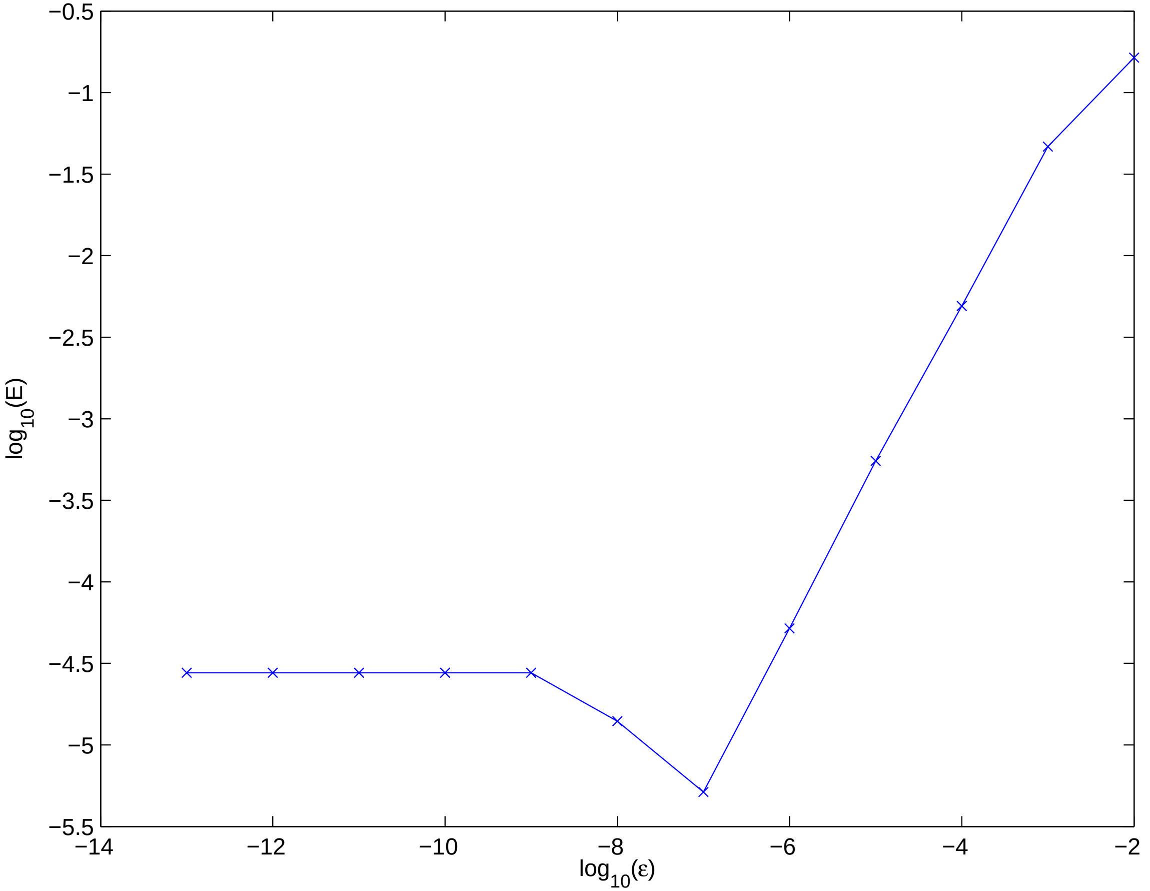
Computation on low rank-R tensor product $\mathbf{u}_k$ needs
$M \times R$ $\boldsymbol{A}$-multiplications $+$ $M \times R$ $\boldsymbol{\Delta}$-multiplications,
which is much less.

Pre-conditioner $\mathbf{P}$ should be used as $\mathbf{P} = \sum_{p=1}^{P} \boldsymbol{\Lambda}^{(p)} \otimes \boldsymbol{P}_p$.
Simplest example: mean value pre-conditioner $\mathbf{P} = \boldsymbol{I} \otimes \boldsymbol{P}_0$
with $\boldsymbol{P}_0$ pre-conditioner for $\mathbb{E}\,(\boldsymbol{A})$.

Similar savings as before, with M replaced by P.

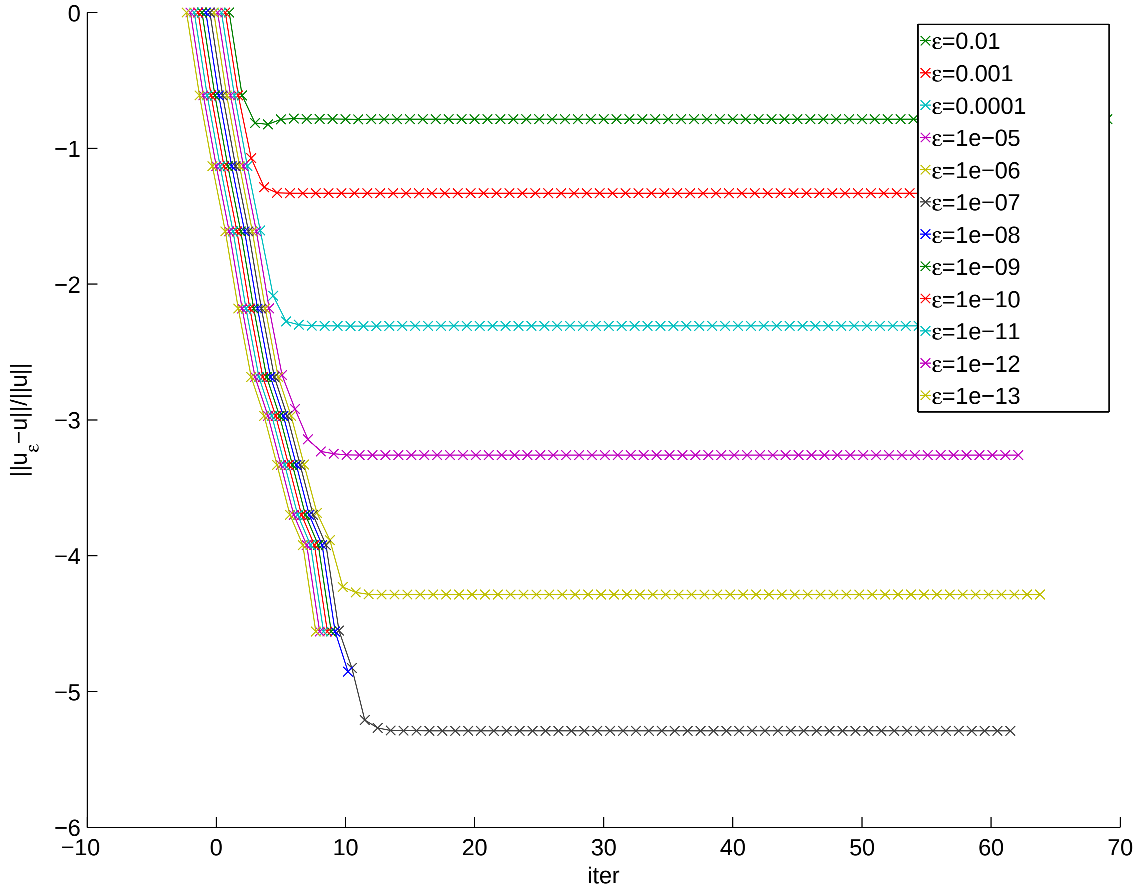
Truncation accuracy

Accuracy of k-term tensor approximation.



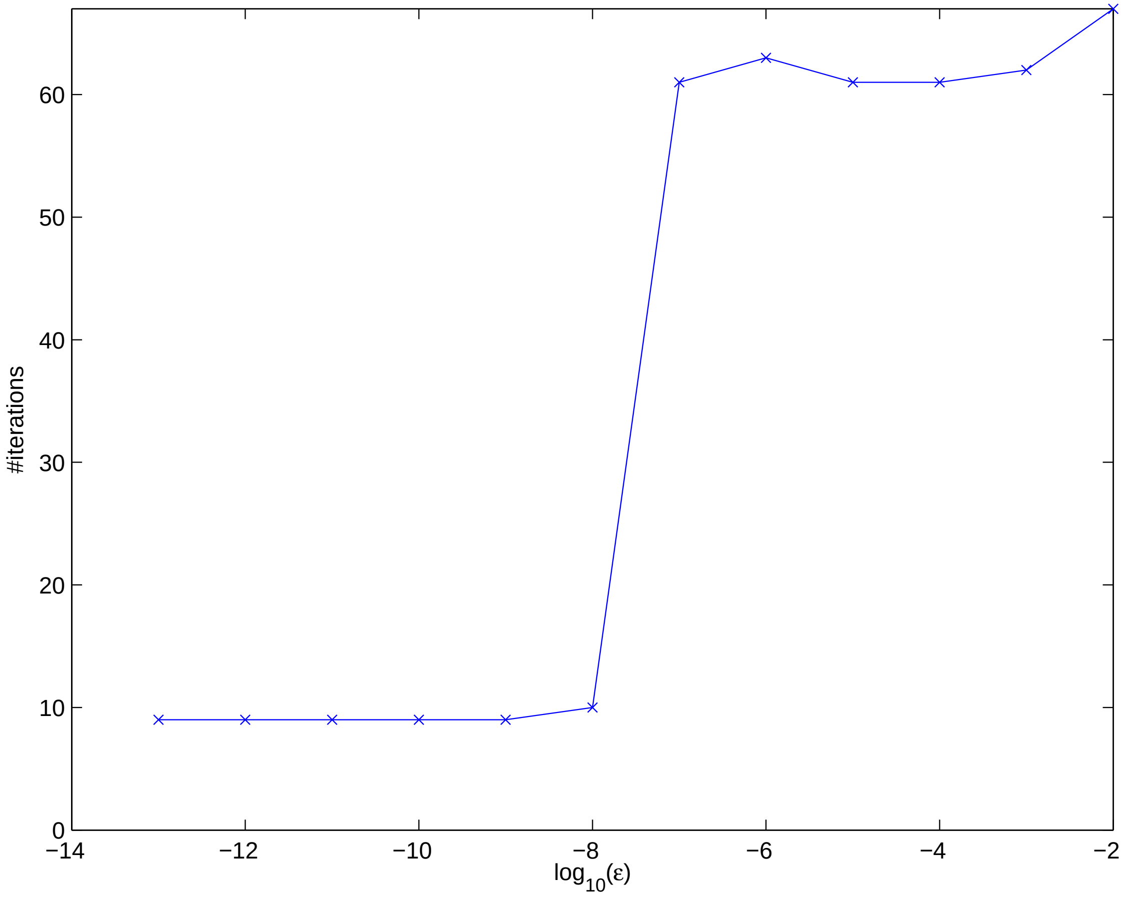
Iteration accuracy

Convergence of truncated iteration.



Number of iterations

Iteration count of truncated iteration.



Dual weighted residuals

Given some functional of interest: $\boldsymbol{\psi}(\mathbf{u}) \approx \mathbb{E}\left(\Psi(\hat{u})\right)$
Error in functional $\epsilon = \boldsymbol{\psi}(\mathbf{u}) - \boldsymbol{\psi}(\mathbf{u}_R) \approx \langle \delta\boldsymbol{\psi}, \mathbf{u} - \mathbf{u}_R \rangle$

For simplicity assume that $\mathbf{u}$ solves a linear system: $\mathbf{A}\mathbf{u} = \mathbf{f}$,
reduced solution has residual $\mathbf{r} = \mathbf{f} - \mathbf{A}\mathbf{u}_R$.
Solve adjoint system $\mathbf{A}^*\mathbf{y} = \delta\boldsymbol{\psi}$ for sensitivity $\mathbf{y}$:

$$\langle \delta\boldsymbol{\psi}, \mathbf{u} \rangle = \langle \mathbf{A}^*\mathbf{y}, \mathbf{u} \rangle = \langle \mathbf{y}, \mathbf{A}\mathbf{u} \rangle = \langle \mathbf{y}, \mathbf{f} \rangle,$$

giving

$$\epsilon \approx \langle \delta\boldsymbol{\psi}, \mathbf{u} - \mathbf{u}_R \rangle = \langle \mathbf{y}, \mathbf{A}(\mathbf{u} - \mathbf{u}_R) \rangle = \langle \mathbf{y}, \mathbf{f} - \mathbf{A}\mathbf{u}_R \rangle = \langle \mathbf{y}, \mathbf{r} \rangle.$$

ϵ may be used for adaptive steering of model reduction process.

Conclusion

- parametric models lead naturally to a number of factorisations
- For efficiency try and use sparse representation throughout: ansatz in low-rank tensor products, as well as storage of solution and residuum—and iterator in tensor products.
- Works in sampling and emulation /functional approximation.
- Works also for non-linear problems and solvers, time-dependent problems.