Uncertainty Quantification and High-Performance Computing with Application to Model Reduction

Vincent Heuveline
Joint work with Michael Schick

---

Reduced-Order Models in Computational Science and Engineering, RWTH Aachen, 30-31 January 2014
Computational Science and Engineering (CompSE)
Engineering Mathematics and Computing Lab (EMCL)

• Research group at IWR in Scientific Computing and High Performance Computing

• Link mathematical modeling, numerical simulation, optimization and hardware aware computing

• Interdisciplinary research activities with application in
  • Meteorology, climate and environment
  • Medical Engineering
  • Energy research

• Interface with the industry, technology transfer
Heidelberg Institute for Theoretical Studies (HITS gGmbH)

- Private, non-profit research institute since 2010
- Funding from Klaus-Tschira-Stiftung

- 10 research groups:
  - Mathematics, Computer Science, Biology, Astronomy, Chemistry…

- Since May 2013: **Data Mining and Uncertainty Quantification**

- Close collaboration with Heidelberg University
Adaptivity for tropical cyclones

- Multiple scales in space and time are relevant for the dynamics of the atmosphere
- In large domains, smallest scales cannot be resolved → adaptive methods
- Goal-oriented adaptivity: User-defined features of interest are in focus
Adaptivity for tropical cyclones: storm-storm interaction

Goal functional

\[ J(v) := \int_{B(x_0, r)} (\nabla \times v)(T, x) \, dx, \quad T = 96h \]
Climate Change and Air Quality

Goal: Efficient integration of models of gas phase chemistry into numerical weather-, climate- and air-quality-models
Climate Change and Air Quality: Model reduction

→ Where do we use which model?
→ How do different models couple?
→ Which influence has a reduced model on the quality of the solution?
Outline

1. Motivation
2. Modeling and numerical methods
3. Numerical examples
4. Outlook
Uncertainty Quantification

- Uncertainty present in data, parameters, models, …
  - Measurement errors
  - Model errors

- How can we quantify uncertainty?
  - Mathematical models
  - Numerical methods
  - Visualization
Types of uncertainty

- **Aleatory uncertainty**
  - Intrinsic variability
  - Non-reducible

- **Examples**
  - Material properties
  - Operating conditions

- **Epistemic uncertainty**
  - Lack of knowledge
  - Reducible

- **Examples**
  - Turbulence models
  - Chemical reactions
Error contributions

• Model error
  – How good is your mathematical model?
  – Representation of physical phenomena

• Method error
  – What is the error of your numerical computation?
  – Round-off errors, convergence errors,…

• Data error
  – How does uncertain data influence your solution?
Schematic view on Uncertainty Quantification

Backward Propagation:
Bayes‘ update,
Parameter estimation, …

Forward Propagation:
Ensembles, Monte Carlo,
Galerkin projection, …

Uncertainty in data / parameter → Phys. Model ← Model output / Observations
Computational intensity

Computational intensity =

# floating point operations per transferred byte

Consequences:

- Bandwidth limitations on many devices
- Only a small fraction of peak performance can be achieved
Computational intensity

Numerical simulation of PDEs and CFD

- Stencil
- Blas 1+2
- LBM
- SpMV

- FFT
- BLAS 3
- Particle methods

- $O(1)$
- $O(\log(N))$
- $O(N)$
The Challenge

- Make your algorithms ready for
  - Fine-grained parallelism
  - Scalability with respect to thousands of threads
  - Data locality

Single-core

Multi-core

Many-core

But how?
The Big Picture

**Hardware evolution**

- Memory wall: Data movement cost prohibitively expensive
- Power wall: Nuclear power plant for each machine (in the cloud)?
- ILP wall: ‘Automagic’ maximum resource utilisation?
- Memory wall + power wall + ILP wall = brick wall

**Inevitable paradigm shift: Parallelism and heterogeneity**

- In a single chip: singlecore → multicore, manycore, ...
- In a workstation (cluster node): NUMA, CPUs and GPUs, ...
- In a big cluster: different nodes, communication characteristics, ...

**This is our problem as applied mathematicians**

- Affects all machines we use, including workstations and laptops
1. Motivation

2. Modeling and numerical methods

3. Numerical examples

4. Outlook
Modeling uncertainty

- Use parameterization with independent random variables for uncertain data \( \xi = (\xi_1, \xi_2, \ldots) \)

- **Polynomial Chaos**
  - Expand stochastic quantities by a finite orthogonal series
  - Use special polynomial interpolation
    \[
    X(\xi) = \sum_{i=0}^{P} x_i \psi_i(\xi)
    \]

- **Karhunen-Loève Expansion**
  - Series expansion based on singular value decomposition
  - Covariance must be known
Non-intrusive vs. intrusive methods (example)

Standard Monte Carlo Method (decoupled/non-intrusive)

Random Number → Det. Input → D. P. → Pathsolution

\[\vdots\]

Random Number → Det. Input → D. P. → Pathsolution

Stoch. solution
Moments, ...

Standard Galerkin projection (coupled/intrusive)

Stoch. Input

![Diagram]

Stoch. solution
Moments, Pathsolutions, ...

D. P.

D. P.

\[\vdots\]

D. P.

D. P.

\[\vdots\]

D. P.

D. P.

\[\vdots\]

D. P.

D. P.
Non-intrusive approaches

• Monte Carlo
  – Standard
  – Quasi Monte Carlo
  – Latin-Hypercube sampling
  – Multilevel Monte Carlo
  – ....

• Collocation methods
  – Sparse grid collocation
  – Numerical quadrature rules

• Variants of Galerkin projection
  – High accuracy
  – More implementation effort
Multilevel/Multigrid methods

• Applicable if hierarchy in model is available
  – Polynomial order, multiscale properties,…

• Compute solution on different levels

• Smooth fine level errors by some (simple) smoothing iterative method
Polynomial Chaos (PC)

Problem: Given a random vector $\xi \in L^2(\Theta, \mathcal{F}, \mathbb{P})$, find $u \in L^2(\Theta, \mathcal{F}, \mathbb{P})$ such that

$$\mathcal{M}(u, \xi) = 0,$$

w.r.t. some probability space $(\Theta, \mathcal{F}, \mathbb{P})$

Decompose $u$ in terms of basis functionals $\psi_i \in L^2(\Theta, \mathcal{F}, \mathbb{P})$, such that

$$u(\xi) = \sum_{i=0}^{\infty} u_i \psi_i(\xi), \quad \text{a.s. in } \Theta$$

**Generalized Polynomial Chaos (gPC):**

$\{\psi_i\}_{i=0}^{\infty}$ defined as a set of orthogonal polynomials

Incompressible Navier-Stokes equations

Incompressible Navier-Stokes equations subject to random boundary and initial conditions (SNSE)

\[
\frac{\partial}{\partial t} u + (u \cdot \nabla) u - \nu \Delta u + \nabla p = 0 \quad \text{in } \Sigma, \text{ a.s. in } \Theta
\]

\[
\nabla \cdot u = 0 \quad \text{in } \Sigma, \text{ a.s. in } \Theta
\]

\[
BCs = BCs(\xi)
\]

\[
ICs = ICs(\xi)
\]

- \( t \in [0, T] \subset \mathbb{R} \)
- \( x \in \Omega \subset \mathbb{R}^d \)
- \( \Sigma := \Omega \times [0, T] \)
- \( \xi \in L^2(\Theta, \mathcal{F}, \mathbb{P}) \) a square-integrable random variable

\( \Rightarrow \) It holds: \( u = u(x, t, \xi) \)
Galerkin projection of a semi-discrete system

- $A(\xi)x(\xi) = b(\xi)$, with $A \in L^2(\Gamma; V_{det,h})$, $x, b \in L^2(\Gamma; V_{det,h})$.
- $V_{det,h}$: discrete deterministic (spatial/temporal) vector space with $\dim(V_{det,h}) < \infty$
- PC expansions $A = \sum_{i=0}^{Q} A_i \psi_i$, $x = \sum_{i=0}^{P} x_i \psi_i$ and $b = \sum_{i=0}^{P} b_i \psi_i$:
  \[
  \sum_{i=0}^{Q} \sum_{j=0}^{P} A_i x_j c_{ijk} = b_k, \quad \forall k = 0, \ldots, P \quad \text{(Galerkin projection).}
  \]
- Third order (sparse!) tensor $\mathbb{T}_3 = (c_{ijk})$:
  \[
  c_{ijk} = \frac{\langle \psi_i \psi_j, \psi_k \rangle}{\langle \psi_k, \psi_k \rangle}.
  \]
- $\langle \cdot, \cdot \rangle$ inner-product on $L^2(\Gamma)$. 
Hierarchy

- $S_p := \{ x \in L^2(\Gamma; V_{\text{det},h}) : x = \sum_{|\alpha| \leq p} x_\alpha \psi_\alpha \}, \quad p \in \mathbb{N}$.
- It holds: $S_0 \subseteq S_1 \subseteq \ldots \subseteq S_p$ (hierarchical PC basis).
- Solve $A^{(l)} x^{(l)} = b^{(l)}$, $x^{(l)} \in S_l$, $l = 0, \ldots, p$, i.e.,
  \[
  \sum_{i=0}^{Q} \sum_{j=0}^{\dim S_l} A_i x_j^{(l)} c_{ijk} = b_k^{(l)}, \quad k = 0, \ldots, \dim S_l.
  \]
- Increasing ordering of coefficients $x_j^{(l)}$ is assumed w.r.t. polynomial degree of PC basis.
- For $l = 0$ we obtain the mean problem
  \[
  A_0 x_0^{(0)} = b_0^{(0)}.
  \]
Prolongation and restriction operators

- Prolongation $P^{l-1}_l : S_{l-1} \rightarrow S_l$ as inclusion:
  $$P^{l-1}_l(x^{(l-1)}) := x^{(l-1)} \in S_{l-1} \subseteq S_l$$

- Restriction $R^{l}_{l-1} : S_l \rightarrow S_{l-1}$ as $L^2$-projection:
  $$R^{l}_{l-1}(x^{(l)}) := \sum_{|\alpha| \leq l-1} x^{(l)}_\alpha \psi_\alpha \in S_{l-1}, \quad x^{(l)} \in S_l$$
Smointer

- Use mean based preconditioner as a smoother, due to its decoupled block structure.
- Given $\tilde{x}^{(l)} \in S_l$ compute $x^{(l)} \in S_l$ by
  \[
  x^{(l)}_k = \left[ S(x^{(l)}, b^{(l)}) \right]_k := \tilde{x}_k^{(l)} + A_0^{-1} \left[ (b^{(l)} - A^{(l)}\tilde{x}^{(l)}) \right]_k,
  \]
  for all $k = 0, \ldots, \dim S_l$.
- Note, that employing full cycles to level $l = 0$, the multilevel approach is non-intrusive.
- Action of $A_0^{-1}$ can be applied independently on each mode.
- If $\dim V_{det,h}$ is not too large, then a direct $LU$ factorization can be applied:
  - Only one factorization needs to be computed a priori.
  - Action of $A_0^{-1}$ only requires forward and backward substitution.
  - This is only one example for solving systems with $A_0$.
  - The choice of the deterministic solver is certainly application dependent.
Algorithm

1: if \((l = 0)\) then
2:     solve \(A_0x^{(0)} = b^{(0)}\)
3: else
4:     \(x^{(l)} = S^{\nu_1}(x^{(l)}, b^{(l)})\) (pre-smoothing, \(\nu_1\) times)
5:     \(r^{(l)} := b^{(l)} - A^{(l)}x^{(l)}\) (compute residual on fine level)
6:     \(r^{(l-1)} = \mathcal{R}_{l-1}r^{(l)}\) (restrict residual)
7:     for \(m = 0\) to \(m < \mu\) do
8:         \(ML(c^{(l-1)}, r^{(l-1)}, l - 1)\) (compute coarse correction, \(\mu\) times)
9:     end for
10:    \(c^{(l)} = \mathcal{P}_{l}^{-1}(c^{(l-1)})\) (prolongate coarse correction)
11:    \(x^{(l)} = x^{(l)} + c^{(l)}\) (update fine level solution with correction)
12:    \(x^{(l)} = S^{\nu_2}(x^{(l)}, b^{(l)})\) (post-smoothing fine level solution, \(\nu_2\) times)
13: end if
Parallelization

• Domain decomposition for the spatial part (mesh)

\[ D = \bigcup_{i=1}^{N_s} D_i, \quad D_i \cap D_j = \emptyset \]

• OpenMP shared memory parallelization of Galerkin matrix vector product on each mesh subdomain (decoupled in k)

\[ \sum_{i=0}^{Q} \sum_{j=0}^{P} A_i x_j c_{ijk} = b_k, \quad \forall k = 0, \ldots, P \]

• OpenMP parallelization of decoupled smoother (block-Jacobi type)
Outline

1. Motivation
2. Modeling and numerical methods
3. Numerical examples
4. Outlook
Example: Flow over backward-facing step (FB2D)

- Parabolic inflow on $\Gamma_i$
- Random viscosity (Uniform distribution):
  \[ \nu(\vec{\xi}) = \nu_0 + \sigma\nu_0 \sum_{i=1}^{N} q^{i-1} \xi_i \]
- Natural outflow boundary conditions on $\Gamma_o$
- No-slip walls on $\Gamma_w$
- Taylor-Hood Finite-Elements
FB2D: Velocity magnitude

Mean w/ streamlines

StdDev w/ contourlines

Fixed N=4 random variables and p=2.
FB2D: Convergence properties

Convergence of inexact Newton method with respect to stochastic variation $\sigma$ and $q=0.5$. 
FB2D: Computational cost

Growth of computational time (in seconds) with respect to the stochastic dimension P+1 with fixed $\sigma = 0.1$ and $q=0.7$.

- **Red**: growing number of random variables with fixed polynomial degree
- **Blue**: growing polynomial degree with fixed number of random variables

4 MPI tasks each with 8 OpenMP threads @ Intel Xeon CPUs with 2.4 GHz
Example: Lid-Driven-Cavity 3D (LDC3D)

- Log-Normal Distribution of lid boundary condition
- Median Reynolds number Re ~ 100
- 5th order Hermite PC expansion
- 3,384,420 degrees of freedom / PC modes
LDC3D: PC Modes

Mode 0

Mode 1

Mode 2

Mode 3

Mode 4

Mode 5
LDC3D: Center line velocity trajectories

- Evaluation of velocity trajectories at center line \((x,0.5,0.5)\) for varying \(x\) in \([0,1]\)
- Plotted mean (black) ±3*standard_deviation (red)
SFB / TRR 125: „Cognition-guided Surgery“
– Knowledge- and model-based Surgery
Mathematical Modeling and Simulation of the elastic behaviour of Soft Tissue in the human body using the Software Toolkit HiFlow³.
HiFlow3:
Finite Element Toolbox / UQ / HPC

- Large scale problems modelled by PDEs
- Discretization with Finite Elements
- Efficient and accurate solution methods

Open source (LGPL): www.hiflow3.org
Conclusions

• Benchmark of a parallel multilevel solver for the incompressible Navier-Stokes equations
• Uncertainty in boundary conditions and kinematic viscosity
• Effective and robust

Outlook

• Further development for the unsteady case
• Coupling with advection-diffusion equations
• Adaptive choice of mesh refinements on each level
Thank you for your attention!