Uncertainty Quantification in Incompressible Flow using Sparse Grids

Friedrich Menhorn (menhorn@in.tum.de), Tobias Neckel (neckel@in.tum.de)

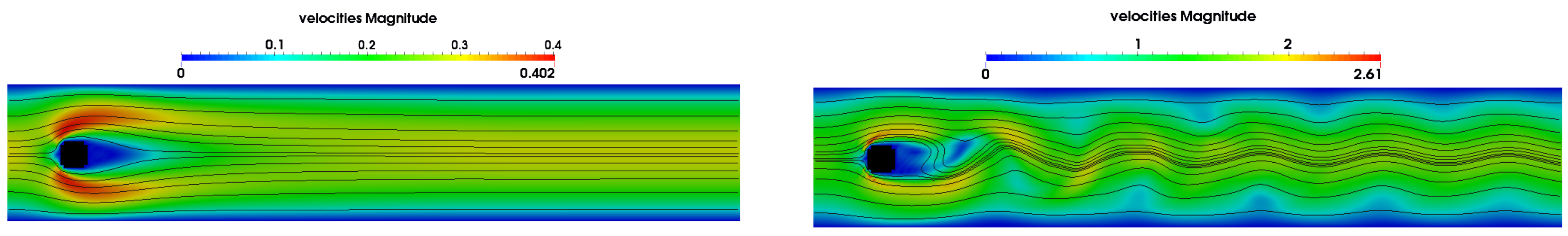
03/15/2015, SIAM CSE 2015, Technische Universität München

Goal:

$$\mathbb{E}[y] = \mathbb{E}[g(x)] = \int_{-\infty}^{\infty} g(x) p_X(x) dx$$

CFD Scenario

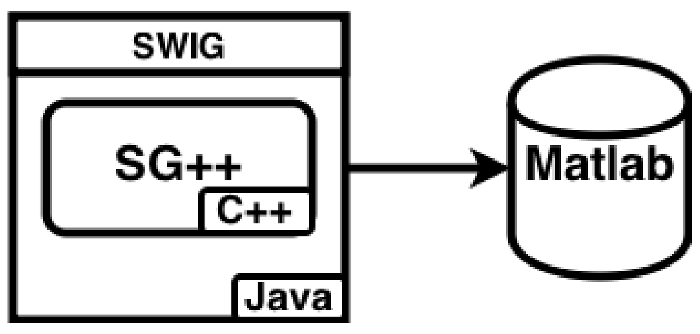
- ***Scenario and solver:***
- Navier Stokes equation for incompressible flow
- Finite elements solver "quickfluid" [1] in Matlab
- Following the benchmark scenario of Turek and Schäfer (1996) [2]:



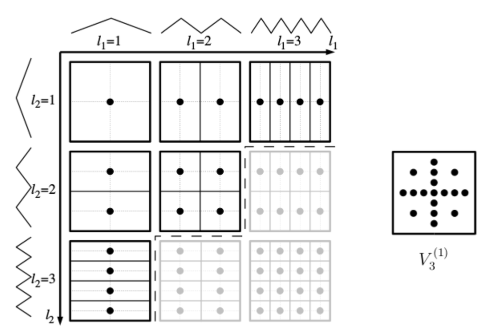
Benchmark cylinder flow scenario as described in [2] for Reynold's number RE=20 (left) and RE=100 (right).

Sparse Grids

- Powerful and memory efficient method to discretize a domain
- Interesting for UQ since it is able to cope with the curse of dimensionality
- Implementation using the C++-library SG++[3] via a Java interface in Matlab
- Here: Grid used as collocation points, for interpolation and for quadrature



Implementation of SG++ in Matlab



Compostion of a 2-d sparse grid with level 3 [3]

Uncertainty Quantification

- Quantify the influence of input parameters with an underlying randomness
- Different approaches:
  - Monte Carlo
  - Polynomial Chaos Expansion
  - Used here: Stochastic collocation
- Usual quantities of interest:
  - Mean
  - Variance/ standard deviation

Idea:
1. Use sparse grid points as stochastic collocation points
2. Evaluate function at collocation points
3. Apply quadrature rule to derive QoI

Create sparse grid based on number of random input parameters

↓

Calculate transformation borders

↓

Transform sparse grid points to stochastic collocation points

↓

Run an independent simulation run for each stochastic collocation point

↓

Use sparse grid interpolation to approximate the numerical solution

↓

Use quadrature on interpolation function to approximate mean

↓

Monte-Carlo quadrature:

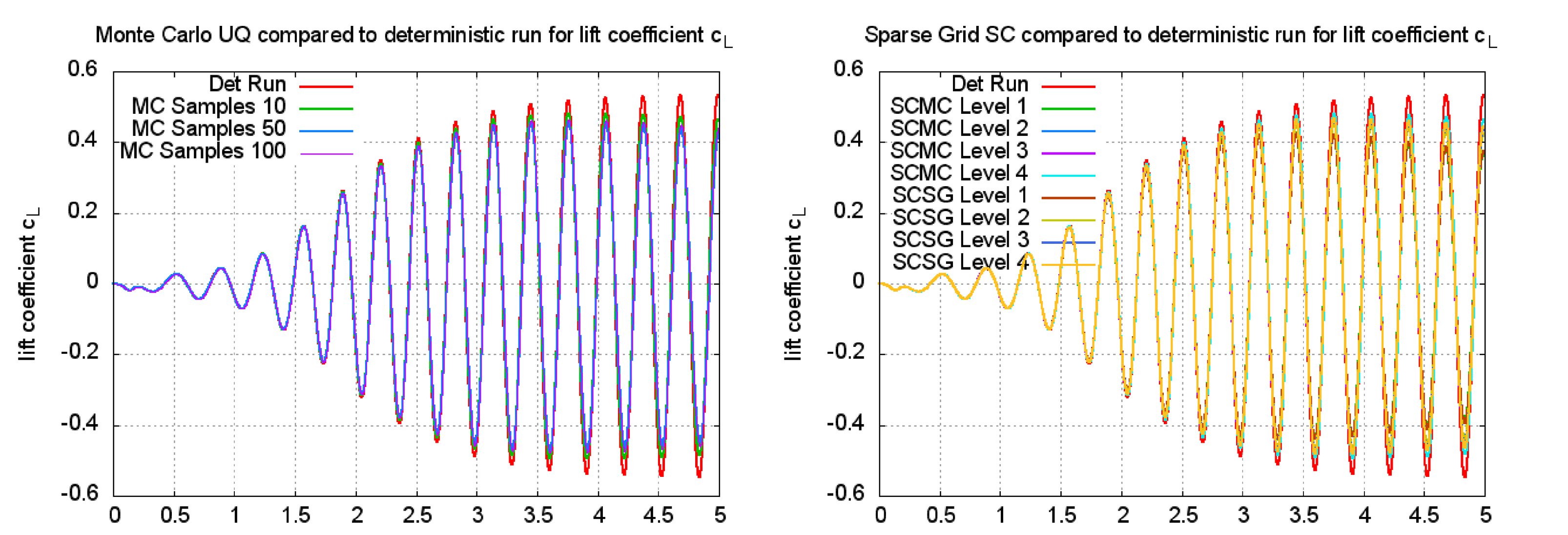
$$\mathbb{E}[y] \approx \frac{1}{n} \sum_{i=1}^{n} f_{SG}(x_i)$$

Sparse grid quadrature:

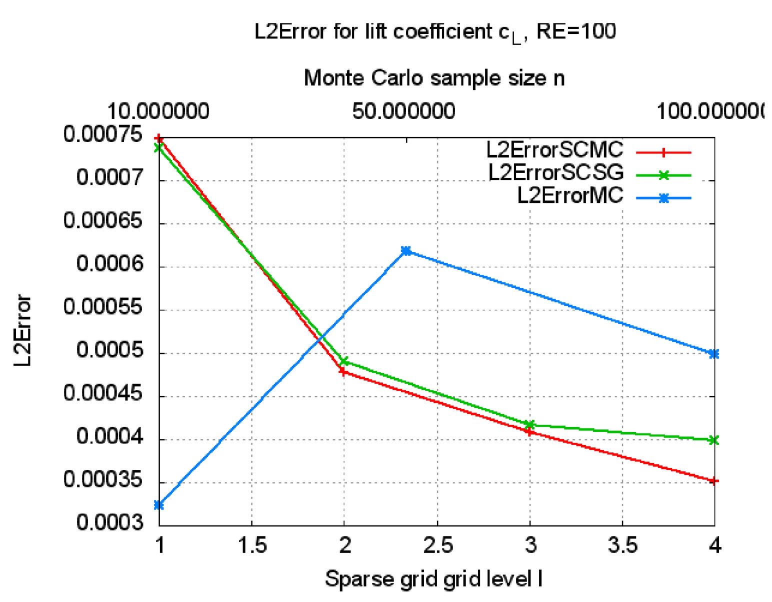
$$\mathbb{E}[y] \approx vol * Q_l^{(d)}(f_{SG} * p_X(x))$$

Results

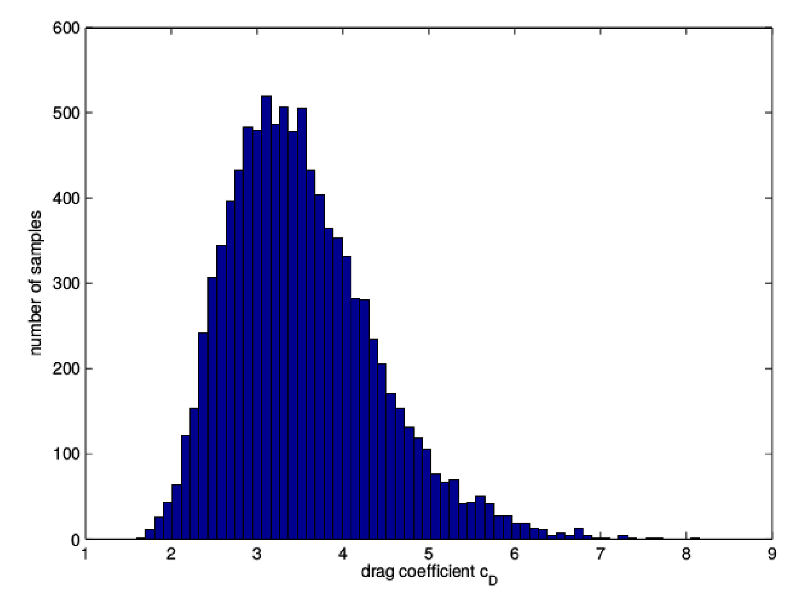
- Random input parameters: Inlet velocity u, density ρ and viscosity η following an underlying normal distribution
- Output parameters: Drag coefficient c_D and lift coefficient c_L resulting from the force on the cylinder



Results for the lift coefficient with RE=100 with the different methods in comparison.



L2 Error for the lift coefficient, RE=100



Histogramm showing the distribution of the drag coefficient using three random input parameters

[1] Chair of Scientific Computing. *Quickfluid*. http://www5.in.tum.de/wiki/index.php/Software_Developments#quickfluid
[2] S. Turek and M. Schäfer. *Benchmark computations of laminar flow around a cylinder*. In E. H. Hirschel, editor, Flow Simulation with High-Performance Computers II, number 52 in NNFM. Vieweg, 1996.
[3] Chair of Scientific Computing. *SG++ toolbox*. http://www5.in.tum.de/SGpp/releases/main.html