

Wissenschaftliches Rechnen

Sommersemester 2019

Übungsblatt 11

Aufgabe 37 (Installieren und starten von MLMC-M++)

Loggen Sie sich mit Ihrem KIT-Account an einem der Poolraum-PCs ein und checken Sie die Praktikumsversion des Programms MLMC-M++ mit

```
git clone -b mlmc
--single-branch https://git.scc.kit.edu/yq8188/mpp.git
```

aus. Führen Sie im Projektordner

```
tar xjf sprng5.tar.bz2
```

und dann anschließend im Ordner sprng5

```
./configure --with-mpi=yes --with-fortran=no
```

aus. sprng5 ist ein Zufallszahlengenerator den Sie zum Schluss noch mit

```
make -j
```

bauen müssen. Wechseln Sie in den Projektordner und erzeugen Sie ein neues Verzeichnis mit `mkdir build`. Führen Sie nun im `build`-Verzeichnis

```
cmake ..
```

aus und anschließend

```
make -j
```

was zunächst ein Makefile passend zu Ihrem Betriebssystem erzeugt und anschließend mit diesem Makefile das Projekt compiliert.

Wenn das Projekt erfolgreich gebaut wurde finden Sie nun in Ihrem `build`-Verzeichnis die ausführbare Datei `MLMC-M++`. Sie können nun `MLMC-M++` mit Open MPI starten indem Sie `mpirun -np 4 MLMC-M++` oder `./MLMC-M++` aufrufen.

Aufgabe 38 (DG-Verfahren für KDR- Gleichung)

Verwenden Sie den Praktikums-Branch von M++ und die Konfigurationsdatei `dgreaction.conf` zur Bearbeitung der Aufgabe. Es soll nochmals die KDR-Gleichung, diesmal aber mit DG-Ansatzelementen, betrachtet werden. Verwenden Sie hierfür `HybridProblem=HybridDGReaction`, `T=1.6`, `dt=0.001`, `Reaction=0` und `Diffusion=0.000001`. Erstellen Sie `Mass`, `Outflow` und `Lösungsplots`.

Aufgabe 39 (Name)

Wir wollen nochmals das elliptische RW-Problem

$$-\operatorname{div}(\kappa(\omega, x)\nabla u(\omega, x)) = 0, \quad \text{auf } D = (0, 1)^2$$

betrachten, wobei die Randwerte wie bisher gegeben sind und $\kappa(\omega, x)$ ein log-normal verteiltes stochastisches Feld mit der Kovarianzfunktion

$$C(x, y) = \sigma^2 \exp(-\|x - y\|_2^\alpha / \lambda^\alpha)$$

darstellt. Zur Approximation des Erwartungswertes des L^2 -Funktional (in folge mit $Q(\omega)$ bezeichnet) der Lösung u verwenden wir die multilevel Monte Carlo (MLMC) Methode

$$\widehat{Q}_{h, \{M_l\}_{l=0}^L}^{MLMC} = \sum_{l=0}^L \widehat{Y}_{h, M_l}^{MC} = \sum_{l=0}^L \frac{1}{M_l} \sum_{i=1}^{M_l} Y_l(\omega_i),$$

wobei $Y_0(\omega) = Q_0(\omega)$ und $Y_l(\omega) = Q_l(\omega) - Q_{l-1}(\omega)$.

Damit die Methode konvergiert sind die folgenden Voraussetzungen zu überprüfen:

- Die Konvergenzrate α der FEM ist gegeben durch

$$|\mathbb{E}[Q_h - Q]| \lesssim h^\alpha, \quad |\mathbb{E}[Q_h - Q]| \lesssim N^{-\alpha}, \quad \text{wobei } N = \dim(V_h).$$

- Die Kosten der FEM können abgeschätzt werden durch

$$\mathcal{C}(Q_h) \lesssim h^{-\gamma}, \quad \mathcal{C}(Q_h) \lesssim N^\gamma.$$

- Die Varianz der Differenz $Q_{h_l} - Q_{h_{l-1}}$ fällt mit

$$|\mathbb{V}[Q_{h_l} - Q_{h_{l-1}}]| \lesssim h^\beta, \quad |\mathbb{V}[Q_{h_l} - Q_{h_{l-1}}]| \lesssim N^{-\beta}.$$

Bearbeiten Sie folgende Aufgaben:

1. Überprüfen Sie die Voraussetzungen der MLMC Methode, indem Sie in der Konfigurationsdatei `mlmc.conf` die Parameter zu `ConvergenceTest` laden. Das heißt verwenden Sie:

```
Experiment=ConvergenceTest, SubroutineEstimator=MCElliptic,
l_mc=3, L_mc=6, M=50, MCVerbose=0, MCPlotting=2
```

Sie können sich mit `plot_mc.py` im Ordner `tools` Konvergenzplots der Größen $\mathbb{E}[Q_{h_l}]$, $\mathbb{V}[Q_{h_l}]$, $\mathbb{E}[Q_{h_l} - Q_{h_{l-1}}]$ und $\mathbb{V}[Q_{h_l} - Q_{h_{l-1}}]$ erstellen. Untersuchen Sie die Konvergenzordnung in Abhängigkeit der Kovarianzfunktion, indem Sie in der Konfigurationsdatei `stochfield.conf` die Parameter

```
sigma=1.0, lambda1 = 0.15, lambda2 = 0.15, alpha = 1.8
```

verändern. Betrachten Sie außerdem die Sampleplots der Permeabilität `perm_level_sample.vtk` und überlegen Sie sich die Rolle der Parameter in der Kovarianzfunktion.

2. Verwenden Sie eine beliebige Konfiguration für das stochastische Feld für die Sie in der vorherigen Aufgabe Konvergenz nachgewiesen haben. Laden Sie weiter die Parameter zu `MLMCExperiment` in der `mlmc.conf`. Das heißt, verwenden Sie:

```
Experiment=MLMCExperiment, SubroutineEstimator=MCElliptic,  
Lmax=9, epsilon=0.5, l_init=3 4 5, M_init=30 10 10,  
MLMCVerbose=2, MCVerbose=1, MCPlotting=3
```

Versuchen Sie den MLMC-Algorithmus nachzuvollziehen. Experimentieren Sie mit verschiedenen stochastischen Feldern, Zielfunktionalen und vorgegebenen Fehler-schranken `epsilon`. Ein Blick in die `vtk`-Dateien kann ebenfalls helfen.

Sie können sich mit `MLMCExperimentOverEpsilon` auch drei verschiedene Fehler vorgeben und sich die Ergebnisse dann mit `plot_mlmc.py` plotten lassen. Interpretieren Sie die erreichten Ergebnisse. Stellen Sie vorher `MLMCVerbose=0`, `MCVerbose=0`, `MCPlotting=0` ein.

Aufgabe 40 (Praktikumsbericht # 5 - Freiwillige Abgabe)

Ergänzen Sie Ihren letzten Praktikumsbericht um eine kurze Zusammenfassung der MLMC-Methode und des DG-Verfahrens für die KDR-Gleichung. Fügen Sie Ihrem Bericht die Auswertungen Ihrer Konvergenzuntersuchungen bei und wählen Sie sich einige Sampleplots aus, anhand derer Sie die Methode nochmals erläutern können.

Homepage:

Unter dem Link <http://www.math.kit.edu/ianm3/lehre/einfwissrech2019s/de> erreichen Sie die Homepage zur Vorlesung. Dort finden Sie alle Informationen zur Vorlesung.