

AUFGABE 5 GLEICHUNGSDATEN UND KONVERGENZGESCHWINDIGKEIT DES LÖSERS

In dieser Aufgabe untersuchen wir experimentell die Abhängigkeit der Löserkonvergenz von den Daten der partiellen Differentialgleichung, die wir diskretisieren. Dazu verwenden wir das Programm `sequentialmodelproblems.cc` aus dem Modul `dune-parsolve`, das auch verwendet wurde, um die numerischen Beispiele aus dem Skript und der Vorlesung zu erstellen. Der Aufbau des Programms wurde in der Übung bereits erläutert.

In den Header-Dateien `problemA.hh` bis `problemE.hh` sind Gleichungsdaten für verschiedene Modellprobleme gekapselt. Die zugehörigen Differentialgleichungen werden vom Hauptprogramm in zwei oder drei Raumdimensionen mit verschiedenen linearen Lösern näherungsweise gelöst. Wir interessieren uns in dieser Aufgabe nur für den Gradienten- und den CG-Löser mit SSOR-Vorkonditionierung in drei Raumdimensionen. (Alle anderen Kombinationen können auskommentiert werden.) Außerdem beschränken wir uns auf das Gitterlevel 5, indem wir die Variablen `j3d` (minimales Gitterlevel) und `J3d` (maximales Gitterlevel) auf 5 setzen. Die gewünschte Defektreduktion `red` belassen wir bei 10^{-8} . Ihre Aufgabe ist es, die Gleichungsdaten in einem beliebigen der Problemdata-Header so zu modifizieren, dass die beiden Verfahren eine möglichst große Zahl an Iterationen benötigen, um zu konvergieren, dass also die entstehenden Steifigkeitsmatrizen möglichst schlecht konditioniert sind. Dabei sollen natürlich die Voraussetzungen an die Gleichungsdaten, die garantieren, dass wir unsere Verfahren überhaupt anwenden können, nicht verletzt werden. Insbesondere soll der Permeabilitätstensor stets symmetrisch und positiv definit sein. Für das Problem D können Sie auch in Zeile 373 vom Hauptprogramm die Korrelationslänge modifizieren. (Falls Sie *wirklich* schlecht konditionierte Probleme finden, müssen Sie noch im Hauptprogramm die maximale Iterationszahl `maxit` erhöhen.) Ihre „besten“ Ergebnisse können Sie wieder per Mail an `sven.marnach@iwr.uni-heidelberg.de` einsenden (einfach die entsprechenden Header-Dateien schicken).

Um `sequentialmodelproblems.cc` zu kompilieren, müssen Sie `dune-pdelab` und `dune-parsolve` auf den aktuellsten Stand aus dem Repository bringen. Wenn Sie sich die Module auf ihrem eigenen Rechner aus dem SVN-Repository heruntergeladen haben, können Sie einfach

```
svn update
```

verwenden, um die aktuellste Version zu erhalten. Im CIP-Pool funktioniert das nur für `dune-parsolve`, da `dune-pdelab` nicht über `svn` installiert wurde. Daher ist hier der einfachste Weg, zunächst im Verzeichnis `dune-course` das Unterverzeichnis `dune-pdelab` komplett zu löschen (`rm -rf dune-pdelab`) und dann die aktuelle Version mittels

```
svn checkout https://svn.dune-project.org/svn/dune-pdelab/branches/1.2snapshot dune-pdelab
```

aus dem Repository zu holen. Dann können Sie alles neu kompilieren mit dem Befehl

```
dune-common/bin/dunecontrol --opts=parallel-debug.opts all
```

ausgeführt in dem Verzeichnis, in das Sie die Module kopiert haben. (Bei Ihnen zuhause haben Sie evtl. keine Optionen-Datei, so dass Sie den `--opts`-Parameter einfach weglassen können.)

12 Punkte

AUFGABE 6 PARALLELE BERECHNUNG DER L^2 -NORM MIT DUNE

In dieser Aufgabe soll die Berechnung der L^2 -Norm aus Aufgabe 4 parallelisiert werden. Dazu sollen mehrere Prozesse über einen Teil des Gebietes integrieren und die Ergebnisse nachher aufsummieren. Die nötigen Teile des Interfaces wurden in der Übung vorgestellt. Zusätzlich ist die Klasse `CollectiveCommunication` nützlich – schauen Sie sich am besten die Online-Dokumentation dieser Klasse an. Den im Konstruktoraufbau benötigten MPI-Communicator erhalten Sie mit der `getCommunicator()`-Methode der `MPIHelper`-Instanz.

Um DUNE auf dem eigenen Rechner mit Unterstützung paralleler Programmierung zu kompilieren, müssen Sie eine MPI-Bibliothek installiert haben, beispielsweise MPICH. In allen Linux-Distributionen gibt es fertige Pakete dafür. Außerdem müssen alle Pakete mittels

```
dune-common/bin/dunecontrol --configure-opts="--enable-parallel" all
```

neu kompiliert werden. Auf den Rechnern im CIP-Pool ist DUNE bereits „parallel“ kompiliert.

8 Punkte