Institut für Numerische und Angewandte Mathematik
FB Mathematik und Informatik der Universität Münster
Prof. Dr. M. Ohlberger, M. Drohmann

18.06.2009

Aufgaben zum Praktikum

Numerik Partieller Differentialgleichungen II

SS 2009 — Blatt 4

Abgabe: 24.05.2009 per Email

Auf dem vierten Aufgabenblatt geht es darum, die numerische C++-Toolbox DUNE kennen zu lernen. Mit Hilfe dieser Toolbox ist es möglich, die Finite Volumen Verfahren für skalare Erhaltungsgleichungen in mehreren Raumdimensionen, effizient und unter Verwendung verschiedener unstrukturierter Gitter in C++ zu implementieren.
Für diejenigen Teilnehmer des Kurses, die die Aufgaben 1-3 bereits im letzten Praktikum bearbeitet haben, ist die Aufgabe 4* gedacht, in der es darum geht, eine Beispielimplementation eines Finite Volumen Verfahrens mit Dune zu verstehen und zu ergänzen.

Aufgabe 1 (Dune Gitter-Typen)

Für die folgenden Aufgaben sei verwiesen auf das Dune-Grid-Howto der Kursseite und die Online-Dokumentation von Dune unter
`http://www.dune-project.org/doc-1.2/doxygen/html/classes.html`.

(a) Laden Sie das Archiv `dune-praktikum.tgz` von der Kursseite. Entpacken, kompilieren und starten sie das enthaltene Programm `gettingstarted`. Versuchen Sie, die Funktionsweise des Programms und die Ausgaben nachzuvollziehen.

(b) Ändern Sie das Programm, so dass es die folgenden Gittertypen verwendet. Gegebenenfalls sind bei Änderungen des Konstruktor-Aufrufs die Makrogitter-Dateien `cube.hexa`,`cube.tetra` und `2dgrid.al` und andere Parameter geeignet zu wählen: **SGrid** (dim=5), **ALUSimplexGrid** (dim=3), **ALUCubeGrid** (dim=3), **AlbertaGrid** (dim=2), **OneDGrid** (dim=1). Wieviele Knoten hat das Gitter jeweils?

Aufgabe 2 (Dune Gitter-Schnittstelle)

Gegeben ist eine Makrogitter-Datei `solid.dgf` mit weiteren Daten-Files zum Einlesen eines unregelmässigen 3D-Objektes im DGF (Dune-Grid-File) Format.

(a) Schreiben Sie ein Programm, welches dies in ein **ALUSimplexGrid** Gitter verwandelt. Dieses soll zunächst eine 2-fache globale Verfeinerung des Gitters durchführen. Wieviele Elemente existieren vor und nach der Verfeinerung?

(b) Schreiben Sie eine Funktion, die für ein Element mit Eckpunkten $x_i, i = 0, \ldots, n_p - 1$ das Baryzentrum $x_b := \frac{1}{n_p} \sum_{i=0}^{n_p-1} x_i$ berechnet. Realisieren Sie eine Iteration über die Level-0-Ebene des Gitters, und geben Sie die Baryzentren der Elemente aus.

(c) Anstelle von globaler Verfeinerung soll 4 mal eine lokale Verfeinerung geschehen in einem Gebiet ihrer Wahl.

(d) Visualisieren Sie das globale bzw. lokale adaptierte Gitter mit paraview oder grape.

(e) Berechnen Sie das Volumen des Objektes durch Aufsummieren der Element-Volumen.

Aufgabe 3 (Diskrete Funktionen)

Für diese Aufgaben sei verwiesen auf die Online-Dokumentation von `Dune-FEM` unter `http://dune.mathematik.uni-freiburg.de/doc/html-current/index.html` Als Gitter soll der Einheitswürfel verwendet werden, der mit einem `ALUSimplexGrid` diskretisiert ist, und durch das Makrogitter `cube.dgf` beschrieben wird. Es soll jeweils ein `LeafGridPart` verwendet werden
Unstetige Funktionen:

(i) Legen Sie eine diskrete Funktion mit Hilfe der Klasse `FiniteVolumeSpace` an, welche elementweise konstant ist. Dies erfordert vorheriges Anlegen eines Funktionenraumes, eines Gridparts und eines diskreten Funktionenraumes.

(ii) Führen Sie eine DOF-Iteration ueber die diskrete Funktion durch und weisen Sie jedem Freiheitsgrad seine Nummer in der Aufzählung zu.

(iii) Visualisieren Sie die resultierende diskrete Funktion in Grape oder Paraview mit Hilfe der `DataOutput` Klasse.

(iv) Werten Sie die diskrete Funktion lokal in einem Element aus (Hinweis: `localFunction(ElementType& el)` in der diskreten Funktion erlaubt Zugriff auf die lokalen Freiheitsgrade und lokale Auswertungen). Vergleichen Sie den erhaltenen Funktionswert mit dem zugehörigen DOF-Wert und erläutern Sie die Beobachtung.

Aufgabe 4* (Finite Volumen Verfahren Erhaltungsgleichungen in 2D)
Das Dune Modul dune-femhowto von der Kursseite enthält eine Beispielimplementierung eines Finite Volumen Verfahrens für ein Advektions-Diffusions Problem im $\mathbb{R}^2$. Dieses befindet im Verzeichnis `examples/finitevolume`.

(i) Entpacken und kompilieren Sie das Modul.

(ii) Verstehen Sie, wie das Programm funktioniert.

(iii) Erweitern Sie das Programm so, dass statt des bereits implementierten Engquist-Osher Flusses auch ein Lax-Friedrich Fluß verwendet werden kann.