Institut für Numerische und Angewandte Mathematik 21.01.2010
FB Mathematik und Informatik der Universität Münster
Prof. Dr. Mario Ohlberger, René Milk

Aufgaben zur Vorlesung

Wissenschaftliches Rechen

WS 2009/2010 — Blatt 12

Abgabe: 4.2.2010 in der Vorlesung / per Email

DG-Verfahren für Erhaltungsgleichungen in 1D

In diesem Aufgabenblatt wollen wir uns mit der Implementierung von DG-Verfahren zur Approximation von skalaren Erhaltungsgleichungen in einer Raumdimension beschäftigen. Das Ziel ist die Implementierung und Validierung eines limitierten Verfahrens, wie es in der Vorlesung in Abschnitt 4.3.2 vorgestellt wurde.
Da die Aufgabe genau analog zu Blatt 1 aus dem Praktikum im SS2009 für Finite Differenzenverfahren formuliert ist, kann eine Implementierung aus dem SS2009 als Vorlage verwendet werden.

Aufgabe 1 (Klassenkonzept) (8 Punkte)

Wir betrachten eine skalare Erhaltungsgleichung in einer Raum-Dimension

$$\begin{aligned} \partial_t u(x,t) + \partial_x f(u(x,t)) &= 0 \quad \text{in } \Omega \times [0,T] \\ u(x,0) &= u_0(x) \text{ in } \Omega \\ u(x,t) &= u_{dir}(x,t) \text{ in } \Gamma_{dir}. \end{aligned}$$

Hierbei bezeichnet $T > 0$ die Endzeit, $\Omega = [a,b] \subset \mathbb{R}$ ein beschränktes Intervall, $f \in C^1(\mathbb{R})$ die Flussfunktion, u_0 stückweise stetige, beschränkte Anfangsdaten und u_{dir} beschränkte Dirichlet-Randdaten, die auf dem Einfluss-Rand $\Gamma_{dir}(t) := \{x \in \partial\Omega | f'(u)n < 0\}$ mit äußerer Einheitsnormalen n vorgeschrieben werden. Der übrige Rand $\partial\Omega \backslash \Gamma_{dir}$ ist Ausfluss-Rand, auf dem keine Bedingungen vorgegeben werden.
Die Diskretisierung dieser Gleichung soll mittels einer expliziten DG-Approximation mit minmod-Limitierung realisiert werden. Hierzu sei K die Anzahl der Zeitintervalle, $\Delta t := T/K$ die Zeitschrittweite und es bezeichnen $t^k := k\Delta t, k = 0, \dots, K$ die diskreten Zeitpunkte. Es seien die N Gitterpunkte im Ort gegeben als $G := \{x_n\}_{n=1}^N$ mit $a = x_1 < \dots < x_N = b$. Über das Gitter G sei der DG-Raum $V_h = V_h^p$ definiert. Durch das DG-Verfahren wird eine Sequenz von diskreten Funktionen $u_h^k \in V_h$ generiert, indem eine Anfangsprojektion $u_h^0 := P_h(u_0)$ durchgeführt wird und anschließend sequenziell $u_h^{k+1} := u_h^k + \Delta t L_h(u_h^k)$ mit dem Ortsdiskretisierungsoperator $L_h : V_h \to V_h$ berechnet wird.

a) Entwerfen Sie ein objektorientiertes Klassenkonzept, d.h. geben Sie für die untenstehenden (und eventuell weiteren) notwendigen Klassen die Schnittstelle an, d.h. Methoden mit Argumenten und kurzer Funktionsbeschreibung. Wie hängen die Klassen zusammen und wie kann das durch Konstruktoren und Membervariablen berücksichtigt werden? Welche Methoden und Argumente können als const deklariert werden?

b) Schreiben Sie in wenigen Zeilen Pseudo-Code ein Hauptprogramm, das mit Hilfe der Klassen die Simulation durchführt.

Einige wichtigen Strukturen sind die folgenden:

Model: Eine Sammlung der Datenfunktionen der Differentialgleichung, d.h. Klasse, welche Punktauswertung der Flussfunktion, Anfangs- und Randwerte erlaubt.

DiscreteSpaceOperator: Eine Klasse, die die Anwendung des diskreten Ortsoperators $v_h = L_h[w_h]$ realisiert.

NumericalFlux: Eine Klasse, die einen numerischen Fluss repräsentiert.

Aufgabe 2 (Implementation) (12 Punkte)

Erstellen Sie basierend auf dem Klassenkonzept von Blatt 1 entsprechende Implementationen der Klassen in einer Datei `dg.hh` und ein entsprechendes Hauptprogramm in `dg.cc`. Die folgenden Spezifikationen konkretisieren weitere Anforderungen an die Programmteile:

- Das Hauptprogramm soll die Intervallgrenzen a und b, die Anzahl der Punkte N, die Endzeit T, Anzahl der Zeitintervalle K und einen Ausgabe-Dateinamen als Kommandozeilen-Argumente übergeben bekommen, die Simulation durchführen und die Lösung zur Endzeit in eine Datei ausgeben, die man dann z.B. mit gnuplot visualisieren kann.

- Als Modellgleichungen sollen die folgenden implementiert werden:

$$f(u) = \frac{1}{2}u^2 \qquad (1)$$

$$u_0(x) = \begin{cases} -\frac{1}{2}x + \frac{3}{2} & \text{für } x \in [1,3] \\ 0 & \text{sonst} \end{cases} \qquad (2)$$

$$u_{dir}(x,t) = 0. \qquad (3)$$

Zur Verifikation der Numerik lautet die exakte Lösung auf dem Gebiet $\Omega = [a,b] = [0,5]$ mit Endzeit $T = 4$

$$u(x,t) = \begin{cases} 0 & \text{für } x < 1 \\ \frac{1}{t}(x-1) & \text{für } 1 \le x < x_s(t) \\ \frac{1}{2-t}(x-3) & \text{für } x_s(t) \le x < 3, t \in [0,2) \\ 0 & \text{für } 3 \le x, t \in [0,2) \\ 0 & \text{für } x_s(t) \le x, 2 \le t \end{cases}$$

mit der Kurve

$$x_s(t) = \begin{cases} 1+t & \text{für } 0 \leq t < 2 \\ 1+\sqrt{2t} & \text{für } 2 \leq t. \end{cases}$$

Bei der Programmierung der Komponenten soll insbesondere Wert auf Dokumentation zu den Klassen und Methoden gelegt werden. Fügen Sie hierzu entsprechende C++-Kommentarzeilen mit Erläuterungen ein.

Aufgabe 3 (Experimente) (12 Punkte)

Mit Hilfe Ihrer Implementation des DG-Verfahrens sollen einige experimentelle Einsichten gewonnen werden. Gehen Sie dazu wie folgt für Polynomgrad $p = 0, 1, 2$ vor.

a) Erweitern Sie ihr Programm so, dass Sie den *experimental order of convergence (EOC)* des Verfahrens für die Endposition $x(T)$ bestimmen können. Was erhalten Sie asymptotisch für kleiner werdende Δt?

b) Bestimmen Sie für den Engquist-Osher-Fluss und gegebenem $N = 100$ empirisch ein K so dass das Verfahren eine plausible Lösung produziert. Erfüllen die zugehörigen Δx und Δt die CFL Bedingung? Wie äußert sich ein Verletzen der CFL-Bedingung, d.h. zu große Zeitschrittweite in der Lösung?

c) Visualisieren Sie eine numerische Lösung zu Zeiten $t = 0$, $t = 1$, $t = 2$ und $t = 4$ und beschreiben Sie qualitativ die Bewegung des Maximums.

d) Visualisieren Sie unter Verwendung des Lax-Friedrichs-Flusses die Lösung zum Endzeitpunkt für Gitter verschiedener Feinheit und beschreiben Sie die qualitative Änderung.

e) Vergleichen Sie die Lösungen zum Endzeitpunkt unter Verwendung des Lax-Friedrichs-Flusses und des Engquist-Osher-Flusses.

f) Führen Sie Laufzeitmessungen durch, um den Effekt der verschiedenen Optimierungslevel (-O1 bis -O3) bei Verwendung des CRTP gegenüber virtuellen Funktionen zu untersuchen.

g) Erstellen Sie eine Tabelle der Fehler von numerischer zu exakter Lösung $e_h := \left\| u_h^K - P_h(u(\cdot, T)) \right\|_{L^1(a,b)}$ für Gitter mit zunehmender Verfeinerung, d.h. Halbierung der Δx_n . Ermitteln Sie als weitere Spalte in der Tabelle den *experimental order of convergence (EOC)* zwischen jeweils zwei aufeinanderfolgenden Verfeinerungsstufen. Seien u_h^K und $u_{h'}^{K'}$ die Lösungen für gegebenes und verfeinertes Gitter. Damit ist der EOC definiert als

$$EOC := \frac{\log(e_h/e_{h'})}{\log 2}.$$