

CFD-Programmier-Seminar

Projekt 5: Limiter

Die Rekonstruktion der Lösung muss so gestaltet werden, dass die TVD-Eigenschaft nicht verletzt wird, d.h. es dürfen keine künstlichen Minima bzw. Maxima entstehen. Hierfür werden sog. Limiter eingesetzt, deren Aufgabe es ist, obige Bedingung zu erzwingen. Man kann das Wirken eines Limiters als Hinzufügen von Diffusion auffassen. Die Kunst bei der Konstruktion eines Limiters ist es, nur so viel Diffusion wie gerade nötig einzubringen, d.h. den Gradienten nur so wenig wie möglich zu limitieren.

Im Rahmen dieser Aufgabe sollen zwei Limiterfunktionen für unstrukturierte Gitter umgesetzt und verglichen werden, nämlich der älteste Limiter dieser Art von Barth und Jespersen sowie der heute aktuelle Limiter von Venkatakrishnan.

Der Limiter von Barth und Jespersen

Das Prinzip des Limiters ist es, an bestimmten Punkten einer Zelle den unlimitierten Wert einer jeden Variable zu rekonstruieren und mit den minimalen und maximalen Werten dieser Variable (z. B. der Dichte) in der Nachbarschaft der Zelle zu vergleichen. Ist der rekonstruierte Wert größer bzw. kleiner als einer der beiden Extremwerte, so wird ein Skalierungsfaktor ermittelt, der dafür sorgt, dass der Gradient so verkleinert wird, dass der Extremwert gerade erreicht wird.

In der ursprünglichen Variante haben Barth und Jespersen die Werte an den Eckpunkten des Elements rekonstruiert, es hat sich jedoch gezeigt, dass es ausreichend ist, dies an den Integrationspunkten an den Seiten (d. h. den Seitenmittelpunkten) zu tun. Auf diese Art und Weise erspart man sich das Abspeichern der Vektoren vom Baryzentrum zu den Eckpunkten.

Formelmäßig lässt sich der Limiter für eine Zelle i mit N Seiten folgendermaßen angeben:

$$\Psi_i = \min_{j=1..N} \begin{cases} \min\left(1, \frac{U_{max} - U_i}{\Delta_2}\right) & \text{falls } \Delta_2 > 0 \\ \min\left(1, \frac{U_{min} - U_i}{\Delta_2}\right) & \text{falls } \Delta_2 < 0 \\ 1 & \text{falls } \Delta_2 = 0 \end{cases}$$

Hierbei ist $\Delta_2 = \nabla U_i \cdot \vec{r}_j$, wobei \vec{r}_j der Vektor vom Baryzentrum der Zelle zum jeweiligen Seitenmittelpunkt ist. Um Divisionen durch null zu vermeiden, setzt man Δ_2 i.d.R. auf $\Delta_2 = \text{sign}(\Delta_2) (|\Delta_2| + \omega)$, wobei ω ein sehr kleiner Wert im Bereich der Maschinengenauigkeit ist (z.B. $\omega = 1E-16$ für REAL-Variablen doppelter Genauigkeit). U_{max} ist der maximale Wert der jeweiligen Variable der eigentlichen Zelle und der Nachbarn, also $U_{max} = \max(U_i, \max_{j=1..N}(U_j))$. Die Definition von U_{min} ist analog dazu, also $U_{min} = \min(U_i, \min_{j=1..N}(U_j))$.

Der Limiter von Venkatakrishnan

Das Prinzip dieses Limiters ist das Gleiche wie das des obigen Limiters von Barth und Jespersen. Der Unterschied liegt jedoch darin, dass der Gradient so limitiert wird, dass die minimal nötige Viskosität eingebracht wird, d. h. dass der Gradient weniger stark gedämpft wird. Im Gegensatz zum obigen Limiter ist der von Venkatakrishnan nicht parameterfrei, d.h. der Benutzer kann (und muss) das Verhalten durch einen frei wählbaren Parameter k (= "Fummelfaktor") beeinflussen. Leider lässt sich dieses Prinzip nicht mehr anschaulich erklären.

Formelmäßig lässt sich der Limiter für eine Zelle i mit N Seiten folgendermaßen angeben:

$$\Psi_i = \min_{j=1..N} \begin{cases} \frac{1}{\Delta_2} \left[\frac{(\Delta_{1,max}^2 + \epsilon^2) \Delta_2 + 2\Delta_2^2 \Delta_{1,max}}{\Delta_{1,max}^2 + 2\Delta_2^2 + \Delta_{1,max} \Delta_2 + \epsilon^2} \right] & \text{falls } \Delta_2 > 0 \\ \frac{1}{\Delta_2} \left[\frac{(\Delta_{1,min}^2 + \epsilon^2) \Delta_2 + 2\Delta_2^2 \Delta_{1,min}}{\Delta_{1,min}^2 + 2\Delta_2^2 + \Delta_{1,min} \Delta_2 + \epsilon^2} \right] & \text{falls } \Delta_2 < 0 \\ 1 & \text{falls } \Delta_2 = 0 \end{cases}$$

Die Definitionen von Δ_2 , U_{min} und U_{max} sind identisch zu denen beim Limiter von Barth und Jespersen, $\Delta_{1,max} = U_{max} - U_i$ und $\Delta_{1,min} = U_{min} - U_i$. Das Quadrat der Konstante ϵ ist definiert als

$$\epsilon^2 = \left(k \sqrt{\text{Zellflaeche}} \right)^3.$$

Dieser Wert ist in jeder Zelle abgelegt.

Programmierarbeiten

Der Aufruf des Limiters erfolgt über `CALL Limiter(CONST, MESH, aElem)` und übergibt damit nur den Pointer auf das Element. Der Limiter wird pro Element einmal aufgerufen. Ähnlich wie bei den Flussfunktionen heißt das, dass Sie ein lokales (elementenlokales) Problem behandeln. Alle folgenden Anweisungen beziehen sich somit auf Operationen innerhalb eines Elements.

Die folgenden Arbeitspunkte sind im Code CFDFV umzusetzen:

1. Aufruf des Limiters in der `SpatialReconstruction.f90`: Die Limitierung der Gradienten erfolgt noch vor der räumlichen Rekonstruktion. Das heißt, dass Sie in der gleichen Schleife wie der räumlichen Rekonstruktion den Limiter aufrufen müssen. Übergeben Sie der Limiter-Routine mit dem Aufruf `CALL Limiter(CONST, MESH, aElem)` den Pointer auf das Element. Der Limiter berechnet die maximalen Gradienten in x- und y-Richtung, `aElem%u_x` und `aElem%u_y`. Die räumliche Rekonstruktion verwendet dann die limitierten Gradienten zur Berechnung der Zustände auf den Seiten des Elements.
2. Limiter von Barth und Jespersen:
 - Setzen Sie den maximalen und minimalen Zustand innerhalb der Zelle auf den Wert im Baryzentrum, `uMax = aElem%pVar` und `uMin = aElem%pVar`.
 - Durchlaufen Sie alle Seiten des Elements und vergleichen den Zustand auf der Nachbarseite mit dem `uMax` und `uMin`:
`uMax(:) = MAX(uMax, aSide%connection%elem%pvar)`
`uMin(:) = MIN(uMin, aSide%connection%elem%pvar)`
Damit sollten sie nach dem Durchlaufen aller Seiten den maximalen und minimalen Zustand an den Kanten gefunden haben.
 - Jetzt bilden Sie die minimale und maximale Differenz der Zustände.
 - Definieren Sie einen Vektor Ψ der Länge Ihrer Zustandsvariable `pvar(:)`.
 - Durchlaufen Sie wieder alle Seiten. Auf jeder Seite programmieren Sie eine Schleife über alle Variablen und berechnen Δ_2 (pro Variable!) wie im Theorieabschnitt beschrieben. Führen Sie mittels eine IF-Abfrage die beschriebene Fallunterscheidung durch und berechnen das Ψ für jede Variable. Sparen Sie eine IF-Abfrage indem Sie Ψ für alle Variablen auf 1 setzen. Zum Schluss vergleichen Sie das Ψ , welches Sie für die Seite berechnet haben mit allen Ψ von den anderen Seiten. Das kleinere Ψ sollte hierbei als das Ψ für die Limitierung der Gradienten im Element gewählt werden.

Praktisch können Sie diese Abfrage umsetzen, indem Sie vor Beginn der Schleife über alle Seiten ein globales Ψ definieren, das für jede Variable 1 ist. Auf jeder Seite vergleichen Sie dieses Ψ mit dem der Seite mit einer MIN-Abfrage. Diese funktioniert auch für Vektoren in Fortran.

- Limitieren Sie die Gradienten $a_{\text{Elem}}u_x$ und $a_{\text{Elem}}u_x$ indem Sie sie mit Ψ multiplizieren.

3. Limiter von Venkatakrisnan:

Die Implementierung vom Venkatakrisnan Limiter läuft analog zu dem Limiter von Barth und Jespersen. Lediglich die IF-Abfrage muss, wie im Theoriekapitel diskutiert, entsprechend umgesetzt werden.

Beide Routinen befinden sich in der Datei `Limiter.f90`.

Hinweis: Die Limiter sind die mit Abstand rechenzeitaufwändigste Routine des Codes, so dass hier ganz besonders auf eine effiziente Umsetzung des Codes geachtet werden soll.

Validierung

Untersuchen Sie die beiden Limiter anhand der folgenden Testfälle:

- 1D-Riemannprobleme
- Richtmyer-Meshkov-Instabilität
- Profilmströmung

Variieren Sie dabei den Parameter k des Limiters von Venkatakrisnan und untersuchen Sie den Einfluss auf das Ergebnis. Bei den stationären Problemen soll vor allem der Einfluss des Limiters auf das Konvergenzverhalten untersucht werden.