

Hamilton Jacobi Equations

Thomas Schmelz

16.11.2010

Einführung

In diesem Kapitel werden die numerischen Methoden zur Lösung von allgemeinen Hamilton-Jacobi Gleichungen (HJG) behandelt. Hamilton-Jacobi Gleichungen sind von der Form:

$$\phi_t + H(\nabla\phi) = 0$$

Befinden wir uns in einem dreidimensionalen Raum, lässt sich diese ebenfalls wie folgt schreiben:

$$\phi_t + H(\phi_x, \phi_y, \phi_z) = 0$$

In den vorherigen Kapiteln hatten wir es bereits mit einigen HJE zu tun. So ist das von außen erzeugte Geschwindigkeitsfeld (siehe Gleichung (3.2)) ein Beispiel mit $H(\nabla\phi) = \vec{V} \cdot \nabla\phi$. Auch die Gleichung (4.4) mit $H(\nabla\phi) = V_n \cdot \nabla\phi$ ist eine HJE, wobei V_n von \vec{x} , t , oder auch $\nabla\phi/|\nabla\phi|$ abhängen kann.

Verbindung mit Erhaltungsgleichungen

Eindimensionale skalare Erhaltungsgleichungen haben folgende Form:

$$u_t + F(u)_x = 0$$

Ein Beispiel wäre die Kontinuitätsgleichung $\rho_t + (\rho u)_x = 0$, die die Erhaltung der Masse in einem System beschreibt. Dabei entspricht ρ der Dichte des zu betrachtenden Materials.

Betrachtet man die allgemeine Form der HJE (in diesem Fall eindimensional) und der Erhaltungsgleichungen etwas genauer, lässt sich folgender Zusammenhang erkennen. Leitet man die HJE nach x ab, erhält man

$$(\phi_t)_x + H(\phi_x)_x = 0$$

da davon ausgegangen wird, dass ϕ stetig differenzierbar ist, lässt sich die Reihenfolge der Ableitungen vertauschen

$$(\phi_x)_t + H(\phi_x)_x = 0$$

Substituiert man nun $u = \phi_x$ erhält man

$$(u)_t + H(u)_x = 0$$

was in der Form einer Erhaltungsgleichung entspricht. Man kann also die Schlussfolgerung ziehen, leitet man die Lösung ϕ einer HJE ab, erhält man die Lösung u einer Erhaltungsgleichung.

Numerische Diskretisierung

Eine HJE lässt sich wie folgt diskretisieren (expliziter Euler):

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + \hat{H}^n(\phi_x^-, \phi_x^+; \phi_y^-, \phi_y^+; \phi_z^-, \phi_z^+) = 0$$

\hat{H} bezeichnet man als numerischen Hamiltonian. Damit Konsistenz gewährleistet ist muss für \hat{H} gelten, $\hat{H}(\phi_x, \phi_x; \phi_y, \phi_y; \phi_z, \phi_z) = H(\phi_x, \phi_y, \phi_z)$.

Die approximierten Ableitungen werden entweder über one-side differencing oder über genauere HJ ENO bzw. HJ WENO Verfahren (siehe Kapitel 3) bestimmt. Im Folgenden betrachten wir monotone Verfahren, d.h. ϕ^{n+1} fällt nicht für alle ϕ^n . Der explizite Euler kann auch mit TVD Runge Kutta Verfahren höherer Ordnung ersetzt. Dabei muss allerdings um Stabilität und Konvergenz zu gewährleisten die CFL Bedingung erfüllt sein:

$$\Delta t \max \left\{ \frac{|H_1|}{\Delta x} + \frac{|H_2|}{\Delta y} + \frac{|H_3|}{\Delta z} \right\} < 1$$

Dabei entsprechen H_1, H_2, H_3 den partiellen Ableitungen von H nach ϕ_x, ϕ_y, ϕ_z .

Lax-Friedrichs Methode (LF)

Als erstes Beispiel betrachten wir die Approximation der Lax-Friedrich-Methode.

$$\hat{H} = H \left(\frac{\phi_x^- + \phi_x^+}{2}, \frac{\phi_y^- + \phi_y^+}{2} \right) - \alpha^x \left(\frac{\phi_x^+ - \phi_x^-}{2} \right) - \alpha^y \left(\frac{\phi_y^+ - \phi_y^-}{2} \right)$$

α^x und α^y beschreiben hier die Größe der numerischen Viskosität und sind definiert als

$$\alpha^x = \max |H_1(\phi_x, \phi_y)|, \quad \alpha^y = \max |H_2(\phi_x, \phi_y)|$$

Die Wahl der Koeffizienten kann sich als schwierig erweisen. In der ursprünglichen Implementierungsform der LF Methode werden die ϕ_x^- bzw. ϕ_x^+ über das gesamte karthesische Gitter ausgerechnet und somit das Intervall $I^x = [\phi_x^{min}, \phi_x^{max}]$ ermittelt. Identisch ermittelt man das Intervall $I^x = [\phi_x^{min}, \phi_x^{max}]$. Jetzt berechnet man nur noch die maximalen Werte von $|H_1(\phi_x, \phi_y)|$ und $|H_2(\phi_x, \phi_y)|$ mit der Nebenbedingung $\phi_x \in I^x$ und $\phi_y \in I^y$.

In Beispiel (3.3) gilt $H_1 = u$ und $H_2 = v$. Daraus folgt $\alpha^x = |u|$ und $\alpha^y = |v|$. In Beispiel (4.4) gilt $H_1 = V_N \phi_x / |\nabla \phi|$ und $H_2 = V_N \phi_y / |\nabla \phi|$ falls V_N von ϕ_x und ϕ_y unabhängig ist. Falls ϕ eine signed distance funktion ist, gilt zusätzlich $|\nabla \phi| = 1$, wodurch sich $H_1 = V_N \phi_x$ und $H_2 = V_N \phi_y$ vereinfachen.

α allgemein über das ganze Gebiet zu bestimmen, kann zu großen Approximationsfehlern führen. Deswegen haben sich neben der LF Methode einigen anderen Varianten entwickelt:

Die SLF Methode (stencil Lax-Friedrich) betrachtet für einen Punkt $\vec{x}_{i,j}$ nur die in der Nähe liegenden Punkte $\vec{x}_{i-3,j}$ bis $\vec{x}_{i+3,j}$, analog für die y-Richtung, wobei HJ WENO zur Bestimmung von ϕ_y^\pm benutzt wird.

Die LLF Methode (local Lax-Friedrich) geht noch einen Schritt weiter. Um α^x zu berechnen, betrachtet man für die Bestimmung von I^x nur die beiden Werte ϕ_x^+ und ϕ_x^- . Um I^y zu ermitteln benutzt man weiterhin die SLF oder LF Methode. Für α^y geht man analog vor.

Die LLLF Methode (local local Lax-Friedrich) arbeitet wie die LLF Methode, mit dem Unterschied, dass für alle Richtungen nur die direkten Nachbarn betrachtet werden.

In der Praxis hat sich gezeigt, dass LF und SLF Verfahren in der Regel zu ungenaue Ergebnisse liefern.

Roe-Fix Methode

Das Roe-Fix Schema ähnelt dem von Lax-Friedrichs, mit dem Unterschied, dass das Prinzip des *upwindings* genutzt wird. \hat{H} wird wie folgt definiert.

$$\hat{H} = H(\phi_x^*, \phi_y^*) - \alpha^x \left(\frac{\phi_x^+ - \phi_x^-}{2} \right) - \alpha^y \left(\frac{\phi_y^+ - \phi_y^-}{2} \right)$$

I^x und I^y werden ermittelt, indem wir ϕ_x^\pm und ϕ_y^\pm über die bereits erläuterte LLLF Methode ausrechnen. Ändert sich das Vorzeichen von $H_1(\phi_x, \phi_y)$ und $H_2(\phi_x, \phi_y)$ für alle $\phi_x \in I^x$ und $\phi_y \in I^y$ nicht, benutzen wir *upwindings*. Bei $H_1 > 0$ setzen wir $\phi_x^* = \phi_x^-$, bei $H_1 < 0$ $\phi_x^* = \phi_x^+$. Analoges gilt für H_2 . α^x und α^y werden gleich 0 gesetzt.

Ändert sich das Vorzeichen von H_1 bzw. H_2 liegt ein sogenannter *sonic point* vor. Dies lässt auf nichteindeutige Lösungen zurückführen. In diesem Fall wechseln wir von der RF Methode zu der LLF Methode um die Viskosität zu ermitteln. Wiederum betrachten wir das Vorzeichen von H_1 und H_2 für $\phi_x \in I_{LLF}^x$ und $\phi_y \in I_{LLF}^y$. Ändert sich das Vorzeichen von H_1 nicht, gilt wieder $\phi_x^* = \phi_x^\pm$ je nach Vorzeichen. Das hat zur Folge, dass H_2 ein *sonic point* hat. Hier wird $\phi_x^* = \frac{\phi_x^- + \phi_x^+}{2}$ gesetzt und α^y wie in LLF bestimmt.

Bei der Durchführung von RF fällt folgendes Problem auf. Obwohl ϕ_x^\pm ausgerechnet werden muss, wird (abgesehen im Fall eines *sonic point*) nur einer dieser beiden Werte benutzt. Um Rechenleistung zu senken, bietet es sich deswegen an, ϕ_x^\pm erst mit Hilfe der backward bzw. forward Methode zu approximieren, mit diesen Werten zu entscheiden, welche dieser beiden Werte gleich ϕ_x^* gesetzt wird und diesen dann wesentlich genauer mit der HJ ENO oder HJ WENO Methode auszurechnen, die allerdings etwas kostspieliger ist.

Godunov Methode

Diese Methode liefert eine exakte Lösung eines Riemannproblems für eindimensionale skalare Erhaltungsgleichungen. Riemannprobleme sind Anfangswertprobleme, bei dem zum Zeitpunkt $t = 0$ alle Werte konstant sind, mit Ausnahme eines Punktes, der eine Unstetigkeitsstelle beschreibt.

Die numerische Approximation des Verfahrens lautet:

$$\hat{H} = ext_x ext_y H(\phi_x, \phi_y)$$

I^x und I^y werden über die LLLF Methode ermittelt. Falls $\phi_x^- > \phi_x^+$ nimmt $ext_x H$ den minimalen Wert für alle $\phi_x \in I^x$ an. Falls $\phi_x^- < \phi_x^+$ nimmt $ext_x H$ den maximalen Wert für alle $\phi_x \in I^x$ an. Bei Gleichheit, wird einfach dieser Wert in H eingesetzt (I^x besteht ja dann auch nur aus einen Wert). Analog wird bei y vorgegangen. Man beachte, dass $ext_x ext_y H \neq ext_y ext_x H$ solange H nicht trennbar ist.

Die Implementierung von Godunov kann sich als schwierig erweisen, ist aber in manchen speziellen Fällen dennoch leicht umzusetzen. Betrachten wir nochmal das Beispiel (3.2) des von außen erzeugten Geschwindigkeitsfeldes. In diesem Fall ist H trennbar. Es gilt also $ext_x ext_y H = ext_x(u\phi_x) + ext_y(v\phi_y)$. In diesem Beispiel gilt:

Falls $\phi_x^- < \phi_x^+$ wollen wir den Wert $u\phi_x$ möglichst minimal. Also wählen wir für $u > 0$ ϕ_x^- und für $u < 0$ ϕ_x^+ .

Falls $\phi_x^- > \phi_x^+$ wollen wir den Wert $u\phi_x$ möglichst maximal. Also wählen wir für $u > 0$

ϕ_x^- und für $u < 0$ ϕ_x^+ .

Zusammengefasst gilt also:

Falls $u > 0$, benutze ϕ_x^-

Falls $u < 0$, benutze ϕ_x^+

Falls $u = 0$, setze $u\phi_x = 0$

Dies entspricht genau dem *upwind differencing*, welches bereits in Kapitel 3 erklärt wurde.