4. Zeitschrittverfahren

4.1 Klassifikation

Durch das Ersetzen von räumlichen und zeitlichen Ableitungen mit finiten Differenzen kann man für jede Gleichung zahlreiche numerische Schemen generieren. Die grosse Zahl dieser Schemen legt es nahe, eine systematische Klassifikation durchzuführen. Die wichtigste Klassifikation dieser Art basiert auf der zeitlichen Diskretisierung. Dabei darf allerdings nicht vergessen werden, dass eine abschliessende Beurteilung der Eigenschaften von Schemen nur anhand der gleichzeitigen Berücksichtigung von zeitlicher und räumlicher Diskretisierung möglich ist. Dies wird durch den Forward-Schritt illustriert, der absolut instabil (in Kombination mit zentrierten räumlichen Differenzen), aber auch stabil (in Kombination mit räumlichen Upstream Differenzen) sein kann.

Im Folgenden betrachten wir die Gleichung

$$\frac{\partial \phi}{\partial t} = F(\phi) \tag{4.1}$$

in welcher F einen beliebigen Operator darstellt. Für die Klassifikation der Zeitschrittverfahren wird allein die Ableitung $\partial\phi / \partial t$ in (4.1) diskretisiert. Dabei werden drei Merkmale betrachtet:

(I) EXPLIZIT / IMPLIZIT

Bei einem *expliziten Schema* kommt in der diskretisierten Form von (4.1) das neue Zeitniveau auf der rechten Seite nicht vor. Zu den Verfahren dieser Kategorie gehört der Forward-Schritt (oder Euler-Schritt)

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F(\phi^n) \tag{4.2}$$

Im Gegensatz dazu ist das Backward-Verfahren

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F\left(\phi^{n+1}\right) \tag{4.3}$$

sowie die Trapez-Regel

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \frac{1}{2}\left[F\left(\phi^{n+1}\right) + F\left(\phi^n\right)\right] \tag{4.4}$$

implizite Verfahren. In der hier verwendeten Form setzen diese quasi das Resultat voraus, um die zeitliche Tendenz zu berechnen.

In einfachen Fällen können implizite Verfahren in eine explizite Form gebracht werden. Im Falle der Reibungsgleichung

$$\frac{\partial u}{\partial t} = -\kappa \cdot u \tag{4.4}$$

ergibt sich zum Beispiel nach Anwendung der Trapez-Regel

$$\frac{u^{n+1}-u^n}{\Delta t} = -\kappa \frac{1}{2}\left(u^{n+1}+u^n\right).$$

Dieser implizite Zeitschritt kann in die explizite Form

$$u^{n+1} = \frac{1-\Delta t\,\kappa/2}{1+\Delta t\,\kappa/2} u^n \tag{4.6}$$

gebracht werden. Existiert eine explizite Form eines impliziten Zeitschrittverfahrens, so bezeichnet man das Verfahren auch als *pseudo-implizit*.

(II) ANZAHL INVOLVIERTER ZEITNIVEAUS

Forward-, Backward- und Trapez-Verfahren sind *Zwei-Niveau-Verfahren*, denn diese Zeitschritte basieren allein auf dem neuen Zeitniveau ϕ^{n+1} sowie dem nächstälteren Niveau ϕ^n. Der Leapfrog-Zeitschritt ist andererseits ein *Drei-Niveau-Verfahren*, zusätzlich wird noch das Niveau ϕ^{n-1} benötigt.

Zwei-Niveau-Verfahren haben den Vorteil, dass kein rechnerischer Mode auftritt, wie wir ihn für das Leapfrog-Schema in Kapitel 3.4.3 kennengelernt haben. Zudem ist bei Drei-Niveau-Verfahren die Abspeicherung der Variablen zu einem zusätzlichen Zeitniveau notwendig, was bei komplexen Modellen einen bedeutenden Mehrbedarf and Speicherplatz zur Folge haben kann. Andererseits sind alle expliziten Zwei-Niveau-Verfahren diffusiv (d.h. $|\lambda|<1$) während bereits das einfache Leapfrog-Schema neutral ist (d.h. $|\lambda|\equiv 1$).

(III) EINSTUFIGE UND MEHRSTUFIGE VERFAHREN

Alle bis anhin betrachteten Verfahren sind einstufige Verfahren. Bei mehrstufigen Verfahren wird zuerst ein provisorischer Wert, z.B. $\tilde{\phi}^{n+1}$ (predictor-Schritt), berechnet, aus dem in einer zweiten Stufe die endgültige Prognose von ϕ^{n+1} (corrector-Schritt) folgt.

4.2 Implizite Advektion

Die Anwendung der Trapez-Regel (4.4) auf die lineare Advektionsgleichung (3.2) ergibt

$$\frac{1}{\Delta t}\left(\phi_j^{n+1}-\phi_j^n\right) = -u\frac{1}{2}\left[\frac{\phi_{j+1}^n-\phi_{j-1}^n}{2\Delta x}+\frac{\phi_{j+1}^{n+1}-\phi_{j-1}^{n+1}}{2\Delta x}\right], \tag{4.11}$$

wobei zentrierte Differenzen im Raum verwendet wurden. Dieses Gleichungssystem lässt sich schreiben als

$$\phi_j^n - \frac{\alpha}{4}\left(\phi_{j+1}^n - \phi_{j-1}^n\right) = \phi_j^{n+1} + \frac{\alpha}{4}\left(\phi_{j+1}^{n+1} - \phi_{j-1}^{n+1}\right). \tag{4.12}$$

Dies stellt ein gekoppeltes algebraisches Gleichungssystem dar, wobei jeder Gitterpunkt eine Gleichung beisteuert. In unserem speziellen Beispiel lässt sich eine geschlossene Lösung finden. Die linke Seite von (4.12) lässt sich zur Zeit n auswerten, und das Resultat wird mit A_j^n bezeichnet. Damit ergibt sich die Matrixform

$$\begin{bmatrix} A_1^n \\ A_2^n \\ A_3^n \\ . \\ . \\ . \\ A_J^n \end{bmatrix} = \begin{bmatrix} 1 & \frac{\alpha}{4} & 0 & . & . & . & 0 & -\frac{\alpha}{4} \\ -\frac{\alpha}{4} & 1 & \frac{\alpha}{4} & 0 & . & . & . & 0 \\ & & . & & & & & \\ 0 & 0 & -\frac{\alpha}{4} & 1 & \frac{\alpha}{4} & 0 & . & 0 \\ 0 & . & 0 & -\frac{\alpha}{4} & 1 & \frac{\alpha}{4} & 0 & 0 \\ & & & & & . & & \\ \frac{\alpha}{4} & 0 & . & . & . & 0 & -\frac{\alpha}{4} & 1 \end{bmatrix} \cdot \begin{bmatrix} \phi_1^{n+1} \\ \phi_2^{n+1} \\ \phi_3^{n+1} \\ . \\ . \\ . \\ \phi_J^{n+1} \end{bmatrix} \tag{4.13a}$$

beziehungsweise

$$\mathbf{A}^n = \mathbf{M} \cdot \Phi^{n+1}, \tag{4.13b}$$

wobei $\mathbf{A}^n$ und Φ^{n+1} Spaltenvektoren bezeichnen. Es gilt also ein algebraisches System von gekoppelten linearen Gleichungen zu lösen, wobei die Grösse der Matrix **M** durch die Anzahl Gitterpunkte J bestimmt wird. **M** hat eine tridiagonale Form (abgesehen von geringfügigen Modifikationen in der linken unteren und rechten oberen Ecke, welche durch die periodischen Randbedingungen bedingt sind). Matrizen dieser Art sind invertierbar, und mit Hilfe der Inversen $\mathbf{M^{-1}}$ lässt sich die Lösung von (4.13) schreiben als

$$\Phi^{n+1} = \mathbf{M}^{-1} \cdot \mathbf{A}^n \quad \text{oder} \quad \phi_j^{n+1} = \sum_l M_{jl}^{-1} A_l^n. \tag{4.14}$$

In der Praxis wird die fehleranfällige Inversion der Matrix **M** vermieden. Stattdessen wird ein iteratives numerisches Verfahren zur Lösung des Systems (4.13) verwendet (z.B. Gauss'sche Elimination).

Das Auftreten eines gekoppelten Systems von vielen Gleichungen ist typisch für die implizite Methode. Die Lösung des resultierenden gekoppelten Systems ist relativ einfach, solange dieses linear ist. Im Allgemeinen versucht man ein nichtlineares Gleichungssystem zu vermeiden, denn die Lösung solcher Systeme ist unter Umständen ausserordentlich schwierig, und es existieren möglicherweise mehrere Lösungen.

Implizite Zeitschritt-Verfahren sind trotz der erhöhten Komplexität sehr effizient, da sie ausserordentlich attraktive Stabilitätseigenschaften aufweisen. Letztere werden kurz anhand der

Von-Neumann-Methode analysiert. Diese hat die wunderbare Eigenschaft, dass sie auch ohne eine explizite Schemenvorschrift anwenbar ist, d.h. die Stabilität kann untersucht werden, ohne dass die Matrix **M** wirklich invertiert wird. Der lineare Ansatz (3.20) kann direkt in Gleichung (4.11) eingesetzt werden, womit sich

$$\lambda - 1 = -\frac{\alpha}{4}\left[\left(e^{ik\Delta x} - e^{-ik\Delta x}\right) + \lambda\left(e^{ik\Delta x} - e^{-ik\Delta x}\right)\right]$$

ergibt. Nach dem Ersetzen der Exponentialfaktoren durch die Sinus-Funktion und dem Auflösen nach λ ergibt sich

$$\lambda = \frac{1 - i\,p/2}{1 + i\,p/2} \qquad \text{mit} \qquad p = \alpha \sin\left(k\,\Delta x\right). \tag{4.15}$$

Es kann leicht nachgeprüft werden, dass

$$|\lambda| \equiv 1 \quad \text{für} \quad \forall \alpha\,, \tag{4.16}$$

d.h. die implizite Advektion ist absolut stabil und neutral. Das CFL-Kriterium muss nicht erfüllt werden! Dies ist konsistent mit der Betrachtung des numerischen Einfluss-Gebietes in Fig.3.6, denn die Kopplung der Gleichungen (4.13) von Gitterpunkt zu Gitterpunkt lässt das numerische Einflussgebiet beliebig gross werden.

Fig.4.1 zeigt den Phasenfehler der impliziten Advektion für α = 0.5, 1, 2, 5, 10 als Funktion der inversen Wellenlängen $1/(l\,\Delta x)$. Es ist sofort ersichtlich, dass der Phasenfehler massiv ist (immer schlechter als beim Upstream-Schema), und mit zunehmender Courant-Zahl stark zunimmt.

Das implizite Schema verdankt seine Popularität primär seiner Stabilität, und keineswegs seiner Genauigkeit. Die Stabilität ist häufig ein ausschlaggebendes Kriterium. In vielen

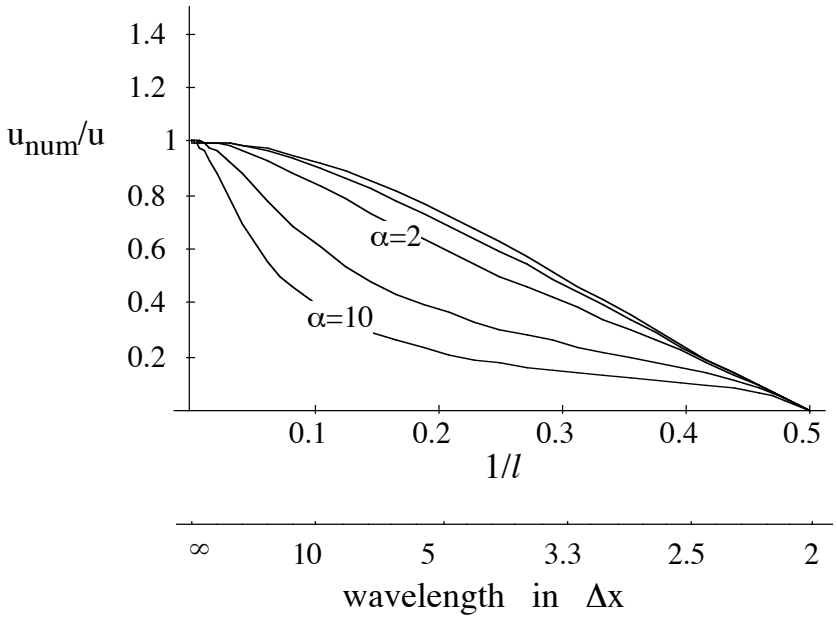


Fig.4.1: Phasenfehler für implizite lineare Advektion als eine Funktion der horizontalen Wellenlänge und für die Courant-Zahlen α = 0.5, 1, 2, 5, 10.

atmosphärischen Anwendungen treten schnelle Wellen auf, an deren physikalischen Effekten man nicht interessiert ist. Trotzdem bestimmen die schnellsten Wellen über das CFL-Kriterium den Zeitschritt von expliziten Formulierungen. Ein Beispiel dazu: In nicht-hydrostatischen, kompressiblen Formulierungen der atmosphärischen Gleichungen sind die Schallwellen enthalten. Ihr Einfluss auf die Dynamik ist vernachlässigbar. In expliziten Formulierungen zwingt ihre vertikale Ausbreitung jedoch zu rechenintensiven kleinen Zeitschritten ($\Delta t \approx 0.2$ sec). Wird die vertikale Ausbreitung der Schallwellen implizit behandelt, sind etwa 10-mal grössere Zeitschritte möglich. Ähnliches trifft für Schwerewellen in grossskaligen hydrostatischen Modellen (z.B. Klimamodellen) zu.

Da nur bestimmte Terme für die schnelle Wellenpropagation verantwortlich sind, ist es oft ausreichend, nur einige wenige Terme implizit zu behandeln, und die anderen in einer expliziten Formulierung zu belassen. In diesem Zusammenhang spricht man von *semi-impliziten Verfahren.*

4.3 Schemen mit mehreren Zeitniveaus

Bei der Diskussion in Kapitel 3.4.3 wurde bereits festgestellt, dass das Leapfrog-Schema nebst dem "physikalischen Mode" mit $\lambda_1 \approx 1$ einen "numerischen Mode" mit $\lambda_2 \approx -1$ besitzt. Für die lineare Advektion kann das leicht gesehen werden, indem man $\alpha=0$ setzt. In diesem Fall ergibt sich aus (3.11)

$$\phi_j^{n+1} = \phi_j^{n-1}$$

was mit

$$\phi_j^{n+1} = \phi_j^n \qquad \text{oder} \qquad \phi_j^{n+1} = -\phi_j^n$$

äquivalent ist. Die Ursache des rechnerischen Modes steckt direkt in der Verwendung von mehr als zwei Zeitniveaus. Alle linearen Schemen mit k Zeitniveaus haben $k-2$ rechnerische Moden, welche unerwünscht sind und eliminiert werden müssen. Hat der λ-Faktor dieser Moden einen kleinen Betrag, so geschieht dies automatisch. Ist andererseits $|\lambda|=1$ wie beim rechnerischen Mode des Leapfrog-Schemas, so sind spezielle Massnahmen oft unerlässlich.

(A) DAS ASSELIN-FILTER

Eine beim Leapfrog-Schema sehr beliebte Methode zur Unterdrückung des rechnerischen Modes ist das Asselin-Filter (Asselin 1972). Dabei wird nach jedem Leapfrog-Zeitschritt

$$\phi^{n+1} = \overline{\phi}^{n-1} + 2\,\Delta t\, F\left(\phi^n\right) \tag{4.17}$$

die Filter-Operation

$$\bar{\phi}^n = \phi^n + \gamma\left(\bar{\phi}^{n-1} - 2\phi^n + \phi^{n+1}\right) \tag{4.18}$$

durchgeführt. Dieses Filter eliminiert die hochfrequenten Anteile des Spektrums, wozu auch der oszillierende rechnerische Mode gehört. Typische Werte für den Filter-Parameter γ sind dabei $0.05 \le \gamma \le 0.25$.

Das Asselin-Filter hat jedoch auch Auswirkungen auf den physikalischen Mode und die Stabilität eines Schemas. Dies kann mit der *Matrix-Methode*, welche eine Verallgemeinerung der Von-Neumann-Methode darstellt, nachgeprüft werden. Für die Stabilitätsanalyse betrachten wir dabei das lineare Advektionsproblem mit zentrierten räumlichen Differenzen, also

$$\phi_j^{n+1} = \bar{\phi}_j^{n-1} - \alpha(\phi_{j+1}^n - \phi_{j-1}^n), \tag{4.19}$$

kombiniert mit dem Asselin-Filter

$$\bar{\phi}_j^n = \phi_j^n + \gamma(\bar{\phi}_j^{n-1} - 2\phi_j^n + \phi_j^{n+1}). \tag{4.20}$$

Im Rahmen der Matrix-Methode wird hier die räumliche Variabilität einer Welle durch Exponentialfaktoren dargestellt, d.h.

$$\phi_{j\pm1}^n = e^{\pm ik\Delta x}\phi_j^n.$$

Einsetzen in (4.19) ergibt

$$\phi_j^{n+1} = \bar{\phi}_j^{n-1} - 2ip\phi_j^n \qquad \text{mit} \qquad p = \alpha\sin(k\Delta x). \tag{4.21}$$

Analog verfährt man mit dem Asselin-Schritt (4.20), wobei zuerst ϕ_j^{n+1} aus (4.21) eingesetzt wird. Dies ergibt

$$\begin{aligned}\bar{\phi}_j^n &= \phi_j^n + \gamma(\bar{\phi}_j^{n-1} - 2\phi_j^n + \bar{\phi}_j^{n-1} - 2ip\phi_j^n) \\ &= \phi_j^n(1 - 2\gamma - 2\gamma ip) + 2\gamma\bar{\phi}_j^{n-1}\end{aligned}. \tag{4.22}$$

Mit (4.21) und (4.22) kann der Leapfrog/Asselin-Schritt (4.19, 4.20) in Matrixform geschrieben werden als

$$\begin{pmatrix}\phi_j^{n+1} \\ \bar{\phi}_j^n\end{pmatrix} = \begin{pmatrix}-2ip & 1 \\ 1-2\gamma(1+ip) & 2\gamma\end{pmatrix}\begin{pmatrix}\phi_j^n \\ \bar{\phi}_j^{n-1}\end{pmatrix}. \tag{4.23}$$

Die durch die Matrix (im folgenden mit Λ bezeichnet) beschriebene Abbildung erhöht dabei das Zeitniveau von $\left(\phi_j^n, \bar{\phi}_j^{n-1}\right)$ um eine Stufe, ganz analog wie das früher bei der Von-Neumann-Methode mit $\phi_j^{n+1} = \lambda\phi_j^n$ ausgedrückt wurde. Dementsprechend wird die Stabilität bestimmt durch die Eigenwerte der Matrix Λ. Berechnet man diese gemäss

$$\|\Lambda - \lambda I\| = 0,$$

so ergibt sich die quadratische Gleichung

$$\lambda^2 + \lambda 2(ip - \gamma) - 2ip\gamma - 1 + 2\gamma = 0.$$

Die dazugehörigen Lösungen λ_1 und λ_2 werden in Fig.4.2 gezeigt und sollten verglichen werden mit dem Resultat ohne Asselin-Filter, d.h.

$$|\lambda_{1,2}| \equiv 1 \quad \text{für} \quad |\alpha| \le 1.$$

Für die Courant-Zahl α = 0.5 und den Filter-Wert γ = 0.25 (siehe Fig.4.2a) wird der rechnerische Mode λ_2 über den ganzen Wellenlängenbereich gedämpft, während der physikalische Mode λ_1 nur geringfügig beeinflusst wird. Dies demonstriert den vom Asselin-Filter erhofften Effekt. Für α=0.8 (siehe Fig.4.2b) wird andererseits eine Instabilität sichtbar, welche sich durch $|\lambda_2|>1$ in der Umgebung der 4-Gitterpunkt-Wellen manifestiert. Dies ist eine unerwünschte Nebenwirkung. Eine Übersicht über diese Asselin-Instabilität ist in Fig.4.2c ersichtlich, wo $|\lambda_{1,2}|$ als Funktion von α für die kritische Wellenlänge l=4 aufgetragen ist. Obwohl weit verbreitet, ist deshalb bei der Verwendung des Asselin-Filters Vorsicht geboten. Insbesondere sollten Werte von $\gamma > 0.1$ vermieden werden.

(B) DIE LEAPFROG-TRAPEZREGEL ("KURIHARA"-SCHRITT)

Eine Alternative zum Asselin-Filter stellt die Leapfrog-Trapezregel dar. Diese basiert auf einem normalen Leapfrog-Schritt

$$\tilde{\phi}^{n+1} = \phi^{n-1} + 2\Delta t\, F(\phi^n), \tag{4.31}$$

gefolgt von einem modifizierten Trapez-Schritt

$$\phi^{n+1} = \phi^n + \frac{\Delta t}{2}\left(F(\phi^n) + F(\tilde{\phi}^{n+1})\right). \tag{4.32}$$

Dieses Schema ist stabil für

$$\alpha \leq \sqrt{2}$$

und eine starke Dämpfung des rechnerischen Modes wird erreicht für $\alpha \leq \sim 1.2$. Das Schema ist etwa 50% teurer als das Leapfrog-Schema, ohne dass der Phasenfehler reduziert wird.

(C) SCHEMEN MIT HÖHERER ORDNUNG IN DER ZEIT

Es gibt Zeitschritt-Verfahren mit einem Diskretisierungsfehler höherer Ordnung, welche wohl rechnerische Moden haben, diese jedoch dämpfen. Ein Beispiel ist das Adams-Bashford-Verfahren 3. Ordnung:

$$\phi^{n+1} = \phi^n + \frac{\Delta t}{12}\left[23F(\phi^n) - 16F(\phi^{n-1}) + 5F(\phi^{n-2})\right]. \tag{4.33}$$

Dieses Schema hat nebst dem physikalischen Mode zwei rechnerische Moden, welche jedoch beide automatisch gedämpft werden.

Schemen dieser Art sind vergleichsweise rechenintensiv. Dazu kommt ein bedeutender Speicherplatzbedarf. Im Falle des Adams-Bashford-Schemas 3. Ordnung müssen pro Gitterpunktvariable vier Zeitniveaus abgespeichert werden.

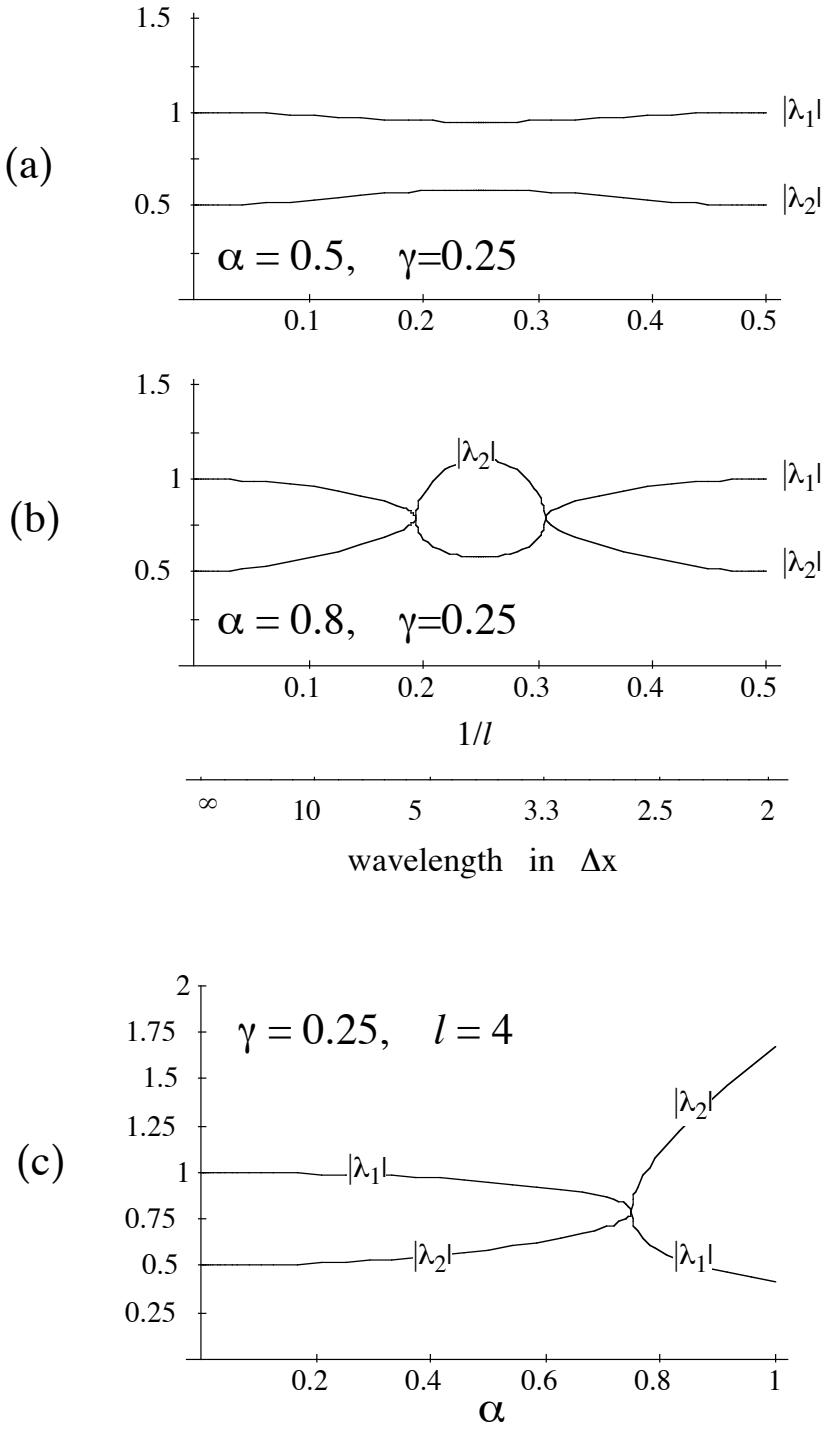


Fig.4.2: Amplitudenfaktor für die lineare Advektion mit dem Leapfrog-Schema kombiniert mit einem Asselin-Filter. Diagramm (a) und (b) zeigen das Verhalten für fixe Werte der Asselin-Konstanten γ und der Courant-Zahl α, und als Funktion der horizontalen Wellenlänge $l\,\Delta x$. Diagramm (c) zeigt die Situation als Funktion der Courant-Zahl α.

4.4 Mehrstufige Methoden

(I) ÜBERSICHT

Die im vorherigen Kapitel 4.3 besprochenen Verfahren mit mehreren Zeitniveaus können höhere Zeit-Ordnung bieten, weisen allerdings Probleme wie nicht-physikalische Moden auf. Eine Alternative sind explizit mehrstufige Verfahren. Diese verwenden grundsätzlich nur zwei Zeitniveaus, werten aber die Funktion F an temporären Stellen innerhalb eines Zeitintervals aus. Zum Beispiel, wenn eine zusätzliche Stelle $n+a$ eingeführt wird, erhält man eine Methode der Form:

$$\tilde{\phi}^{n+a} = \phi^n + a\Delta t\, F(\phi^n) \tag{4.34a}$$

$$\phi^{n+1} = \phi^n + b\Delta t\, F(\tilde{\phi}^{n+a}) + (1-b)\Delta t\, F(\phi^n) \tag{4.34b}$$

wobei $0 \le a, b \le 1$. Es ist ersichtlich, dass F an zwei Stellen n und $n+a$ ausgewertet wird, und man spricht daher von einer *zweistufigen* Methode.

Eine wichtige Klasse der mehrstufigen Methoden bilden die *Runge-Kutta Methoden*. Ein explizites Runge-Kutta Schema mit s Stufen wertet F an s Stützstellen folgendermassen aus:

$$\begin{aligned} k_1 &= F(\phi^n) \\ k_2 &= F(\phi^n + \Delta t a_{21} k_1) \\ k_3 &= F(\phi^n + \Delta t[a_{31}k_1 + a_{32}k_2]) \\ &\vdots \\ k_s &= F(\phi^n + \Delta t[a_{s1}k_1 + a_{s2}k_2 + \ldots + a_{s,s-1}k_s]) \end{aligned} \quad . \tag{4.35}$$

Ansätze für k_1 bis k_3 ergeben sich durch „Euler-Schritte“. Es können Stützwerte an diesen Stellen definiert werden (vergleiche die folgenden Formeln mit Gleichungen 4.2 und 4.37):

$$\begin{aligned} \phi_1^* &= \phi^n \\ \phi_2^* &= \phi^n + \Delta t\, a_{21} F\left(\phi_1^*\right) \\ \phi_3^* &= \phi^n + \Delta t\, a_{31}\left[F\left(\phi_1^*\right) + \Delta t\, a_{32} F\left(\phi_2^*\right)\right] \\ &\vdots \end{aligned} \tag{4.36}$$

In ϕ_2^* wird die Steigung an der Stelle ϕ_1^* berücksichtigt, in ϕ_3^* zusätzlich noch diejenige an der Stelle ϕ_2^*, usw. Hier lässt sich bereits der Unterschied zur Euler-Methode diskutieren. Im Euler-Verfahren

$$\phi^{n+1} = \phi^n + \Delta t \cdot F\left(\phi^n\right) \tag{4.37}$$

geht nämlich nur die Ableitung $F(\phi^n)$ am Anfang des Intervalls $[t_n, t_{n+1}]$ ein. Die Formel ist in dieser Hinsicht asymmetrisch. Beim Runge-Kutta-Verfahren wird versucht, diese Asymmetrie

zu beheben, indem auch die Ableitungen an Zwischenstellen benutzt werden. Der Funktionswert bei Zeitschritt $n+1$ wird approximiert durch

$$\phi^{n+1} = \phi^n + \Delta t[b_1 k_1 + b_2 k_2 + \cdots + b_s k_s] \tag{4.38}$$

Ein RK (Runge-Kutta) Schema kann schematisch durch seine *Butcher-Tabelle* beschrieben werden:

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} \\ \hline & b_1 & b_2 & \ldots & b_{s-1} \quad b_s \end{array} \tag{4.39}$$

Die Koeffizienten c_i definieren die Stützstellen, an denen die Funktion F ausgewertet werden:

$$\{t + 0\Delta t, t + c_2\Delta t, t + c_3\Delta t, \cdots, t + c_s\Delta t\}$$

Weil F in diesem Fall nicht direkt von t abhängt, sind diese hier nicht weiter von Interesse. Die Koeffizienten a_{ij} und b_j werden so gewählt, dass die Fehlerordnung maximal (und der Fehler minimal) wird. Dieses Kriterium lässt aber gewisse Freiheitsgrade zu, sodass es immer mehrere RK Methoden mit s Stufen gibt. Bei den zwei-stufigen Methoden gibt es zum Beispiel zwei bekannten Schemen:

- verbesserter Euler:

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array} \tag{4.40}$$

- Heun-Methode:

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{4.41}$$

Diese Methoden haben gewisse Nachteile (leichte Instabilität, starke Dämpfung) und werden für partielle Differentialgleichungen nur in speziellen Situationen empfohlen. Einige Nachteile können vermieden werden, wenn auf die maximale Ordnung verzichten wird, z.B. mit dem Matsuno Schema, das im Sinne von (4.34) definiert wird durch:

- Matsuno (Forward-Backward): $a = b = 1$

Das Matsuno Schema vermeidet die Verstärkung von hochfrequenten Wellen, ist aber nur erster Ordnung, und somit keine RK Methode. Der Matsuno-Schritt wirkt stark diffusiv, und wird

deshalb gelegentlich für den Beginn der Integration von numerischen Modellen mit leicht inkonsistenten Anfangsbedingungen (anstelle einer Initialisierung) eingesetzt. Für weitere Integration werden meistens höherstufige RK Schemen verwendet.

(II) HERLEITUNG VON RUNGE-KUTTA-VERFAHREN

Die Herleitung einer RK Methode geht von der Differentialgleichung

$$\frac{d\phi}{dt} = F(\phi) \tag{4.42}$$

aus. Diese wird in eine äquivalente Integralgleichung umgeformt:

$$\phi^{n+1} = \phi^n + \int_{t_n}^{t_{n+1}} dt' \, F\left[\phi(t')\right] \tag{4.43}$$

Beim Runge-Kutta-Verfahren wird das Integral in (4.43) durch eine Quadraturformel mit Stützstellen im Intervall $[t_n,\ t_{n+1}.]$ approximiert. Im Folgenden wird dies am Beispiel der RK Methode dritter Ordnung durchgeführt. Die k_i werden nach Taylor in ϕ entwickelt (mit den Abkürzungen $F \equiv F\left(\phi^n\right), F' = F'\left(\phi^n\right), F'' = F''\left(\phi^n\right)$):

$$\begin{aligned}
k_1 &= F \\
k_2 &= F + \Delta t a_{21} k_1 F' + \frac{(\Delta t a_{21} k_1)^2}{2} F'' + \mathrm{O}(\Delta t^3) \\
&= F + \Delta t a_{21} F F' + \frac{(\Delta t a_{21})^2}{2} F^2 F' + \mathrm{O}(\Delta t^3) \\
k_3 &= F + \Delta t (a_{31} k_1 + a_{32} k_2) F' + \frac{\Delta t^2 (a_{31} k_1 + a_{32} k_2)^2}{2} F'' + \mathrm{O}(\Delta t^3) \\
&= F + \Delta t (a_{31} F + a_{32} (F F' + \Delta t a_{21} (F + \mathrm{O}(\Delta t))) F' \\
&\quad + \frac{\Delta t^2 (a_{31} F + a_{32} (F + \mathrm{O}(\Delta t)))^2}{2} F'' + \mathrm{O}(\Delta t^3)
\end{aligned} \tag{4.44}$$

Einsetzten von (4.44) in (4.38) liefert den folgenden Ausdruck:

$$\begin{aligned}
\phi^{n+1} = \phi^n &+ \Delta t \left[b_1 F + b_2 F + b_3 F \right] \\
&+ \Delta t^2 \left[a_{21} F F' + b_3 \left(a_{31} F + a_{32} F \right) F' \right] \\
&+ \Delta t^3 \left[\tfrac{1}{2} a_{21}^2 b_2 F^2 F'' + b_3 a_{32} a_{21} F'^2 F + \tfrac{1}{2} b_3 \left(a_{31} + a_{32} \right)^2 F^2 F'' \right] \\
&+ O\left(\Delta t^4 \right)
\end{aligned} \tag{4.45}$$

Anderseits kann man ϕ^{n+1} auch mit einer Taylor Entwicklung in der Zeit approximieren. Unter Verwendung von (4.42) und der Kettenregel können anschliessend Zeitableitungen von ϕ durch Ableitungen von F ersetzt werden:

$$\begin{aligned}\phi^{n+1} &= \phi^n + \Delta t\frac{d\phi}{dt} + \tfrac{1}{2}\Delta t^2\frac{d^2\phi}{dt^2} + \tfrac{1}{6}\Delta t^3\frac{d^3\phi}{dt^3} + O\left((\Delta t)^4\right) \\ &= \phi^n + \Delta t\,F + \tfrac{1}{2}\Delta t^2\,F'F + \tfrac{1}{6}\Delta t^3\left(F''F^2 + F'^2F\right) + O\left((\Delta t)^4\right)\end{aligned} \tag{4.46}$$

Ein Koeffizientenvergleich zwischen (4.45) und (4.46) liefert:

$$\begin{aligned} &b_1 + b_2 + b_3 = 1 \\ &a_{21}b_2 + a_{31}b_3 + a_{32}b_3 = \tfrac{1}{2} \\ &\tfrac{1}{2}a_{21}^2 b_2 + \tfrac{1}{2}b_3\left(a_{31} + a_{32}\right)^2 = \tfrac{1}{6} \\ &b_3\,a_{32}\,a_{21} = \tfrac{1}{6}\end{aligned} \tag{4.47}$$

Sind diese Bedingungen erfüllt, so ergibt sich ein RK Schema der Ordnung 3. Man stellt leicht fest, dass die Koeffizienten a_{ij} und b_j durch (4.47) nicht vollständig bestimmt sind. Es verbleiben zwei Freiheitsgrade, welche eine grosse Flexibilität in der Wahl des Schemas erlauben. Ein wichtiges Kriterium ist die rechnerische Effizienz der Methode, insbesondere der Speicherbedarf. Williamson (1980) untersuchte alle Methoden der Ordnung 3 auf ihren Speicherplatzbedarf und schlug folgendes Schema vor:

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{3}{4} & -\frac{3}{16} & \frac{15}{16} & \\ \hline & \frac{1}{6} & \frac{3}{10} & \frac{8}{15} \end{array} \tag{4.48}$$

Dieses Schema kann wie folgt umgeschreiben werden:

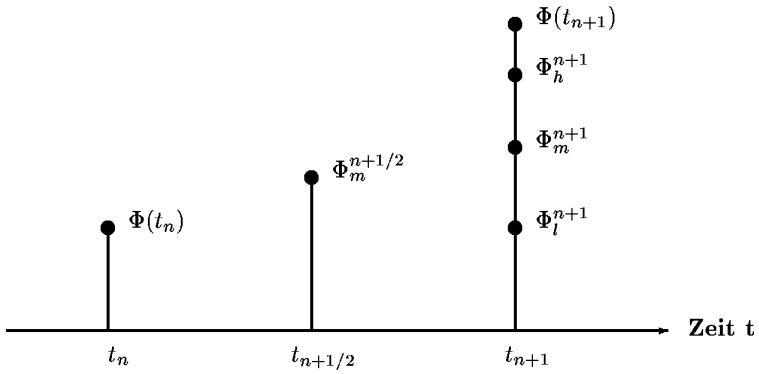
$$\begin{aligned} q_1 &= \Delta t\,F(\phi^n) & \phi^{(1)} &= \phi^n + \frac{q_1}{3} \\ q_2 &= \Delta t\,F(\phi^{(1)}) - \frac{5}{9}q_1 & \phi^{(2)} &= \phi^{(1)} + \frac{15}{16}q_2 \\ q_3 &= \Delta t\,F(\phi^{(2)}) - \frac{153}{128}q_2 & \phi^{n+1} &= \phi^{(2)} + \frac{8}{15}q_3 \end{aligned} \tag{4.49}$$

Mit dem Williamson RK Schema kann sehr viel Speicherplatz eingespart werden. In praktischen Anwendungen mit zeitabhängigen partiellen Differentialgleichungen kann ϕ^n ein sehr grosser Vektor von unbekannten Grössen sein (Temperatur, Druck, Geschwindigkeiten ... auf jedem Gitterpunkt eines 3D-Gitters). Ist m die Anzahl der Unbekannten in ϕ, so kommt das obige Schema mit $2m$ Speicherplätzen aus (je m für ϕ und q), indem die Felder ϕ und q sukzessive überschrieben werden. Das Schema braucht damit gleichviel Speicherplatz wie das Leapfrog-Schema, aber weniger als das Leapfrog-Schema mit Asselin-Filter ($3m$).

Das oben ausgeführte Vorgehen lässt sich leicht auf Verfahren höherer Ordnung verallgemeinern. Oft werden RK Methoden der Ordnung 4 eingesetzt. Ebenfalls gibt es implizite RK Methoden, in denen die Koeffizienten a_{ij} eine Rechtecksmatrix statt einer Dreiecksmatrix bilden. Diese Schemata sind rechnerisch teuer, weil in jedem Zeitschritt die

Koeffizienten durch einen iterativen Prozess bestimmt werden müssen. Sie sind für steife gewöhnliche Differentialgleichung geeignet, finden aber in der Umweltphysik selten Anwendung.

Bei gewöhnlichen Differentialgleichungen lohnt es sich häufig, die Schrittweite Δt nicht konstant zu setzen, sondern der „Glattheit" der Lösungsfunktion anzupassen. An Stellen, wo die Lösung stark variiert, sollte Δt klein sein, an Stellen wo die Lösung hingegen sehr glatt ist, reicht auch ein grösserer Zeitschritt aus. Eine solche Schrittweitensteuerung garantiert, dass die Rechenzeit für die interessanten (variablen) Bereiche eingesetzt wird, und die uninteressanten möglichst wenig Rechenzeit beanspruchen. Das Vorgehen bei der Schrittweitensteuerung ist in der untenstehenden Skizze dargestellt:



Zur Zeit t_n hat die exakte Lösung den Wert $\phi(t_n)$, zur Zeit t_{n+1} den Wert $\phi(t_{n+1})$. Ein Zeitschritt Δt mit einem numerischen Schema führt – ausgehend vom exakten Wert $\phi(t_n)$ – auf ϕ_l^{n+1}. Die Differenz $d_{n+1}=\phi(t_{n+1})-\phi_l^{n+1}$ ist der lokale Diskretisierungsfehler des Schemas. Der Zeitschritt Δt soll nun so gewählt werden, dass der lokale Diskretisierungsfehler d_{n+1} kleiner als eine vorgegebene Schranke ε_1 bleibt. Das Problem beim obigen Vorgehen ist natürlich, dass der exakte Wert $\phi(t_{n+1})$ unbekannt ist, und damit auch d_{n+1}. Um den Wert d_{n+1} zu *schätzen*, bieten sich zwei Möglichkeiten an: Entweder ein Schema höherer Ordnung mit der Schrittweite Δt (resultierend in ϕ_h^{n+1}), oder die Zerlegung des Intervalls $[t_n, t_{n+1}]$ in zwei Abschnitte der Länge $\Delta t/2$ (resultierend in $\phi_m^{n+1/2}$ für den ersten Teilschritt und ϕ_m^{n+1} für den zweiten Teilschritt). Die Schätzung für d_{n+1} ist dann die Differenz zwischen ϕ_h^{n+1} und ϕ_l^{n+1} bzw. ϕ_m^{n+1} und ϕ_l^{n+1}. Sobald nun der geschätzte lokale Diskretisierungsfehler die Schranke ε_1 unterschreitet, soll die Schrittweite halbiert werden. Fällt sie unter eine Schranke $\varepsilon_2<\varepsilon_1$, so soll die Schrittweite wieder verdoppelt werden.

Wichtig beim obigen Verfahren, ist, dass der Diskretisierungsfehler mit möglichst wenig zusätzlichem Rechenaufwand (Funktionsauswertungen) auskommt. Wir wollen das hier nicht weiter verfolgen, auch deshalb, weil die Schrittweitensteuerung bei partiellen

Differentialgleichungen in der Regel nicht zur Anwendung kommt. Dort ist der Gewinn zu gering. Die limitierenden Faktoren sind eher die räumlichen Diskretisierungen.

Schlussendlich soll noch die Stabilität und der Phasenfehler des obigen Runge-Kutta-Verfahrens diskutiert werden. Das Resultat ist in Fig.4.3 dargestellt. Hierbei bezeichnet α die Courant-Zahl Das Schema bleibt ungefähr bis α=1.75 stabil. Der Phasenfehler ist im Vergleich zum Leapfrog-Schema deutlich vermindert.

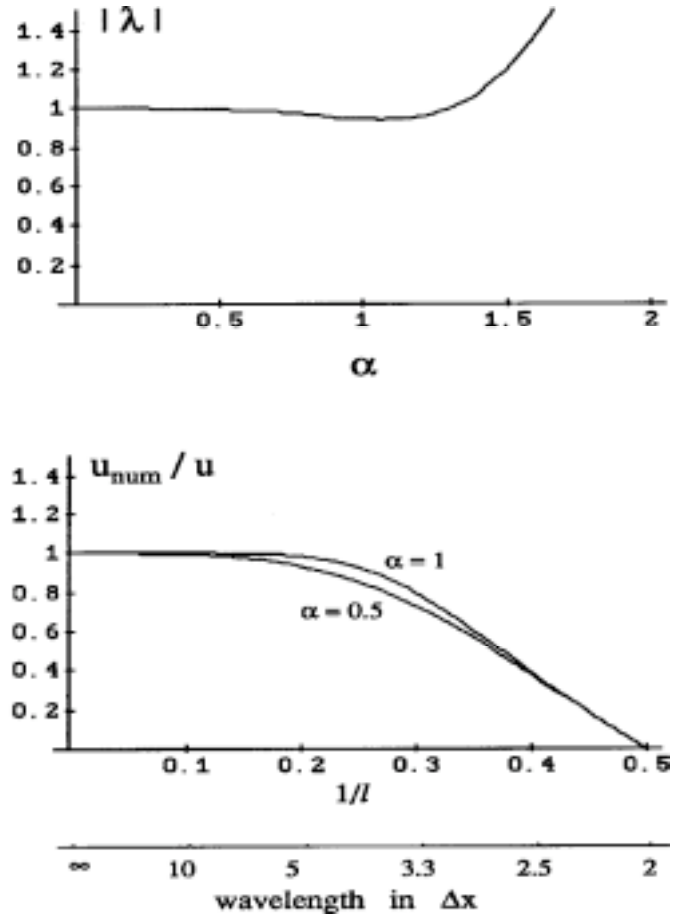


Fig.4.3: Amplituden- und Phasenfehler des Runge-Kutta-Schemas dritter Ordnung. Verstärkungsfaktors $|\lambda|$ als Funktion der Courant-Zahl α (oben) und Phasenfehler als Funktion der Wellenlänge und für zwei Courant-Zahlen (unten).