Stabilität — Konsistenz — Konvergenz

Bei der Definition der Konvergenz eines MSV stellt sich das Problem, daß die durch das Verfahren gelieferten numerischen Lösungen ja *Folgen* $\{y_0, y_1, y_2, \ldots,\}$ sind, die exakten Lösungen aber richtige *Funktionen* $y(x)$. Es handelt sich also um mathematisch verschiedene Objekte, die nicht ohne weiteres miteinander zu vergleichen sind. Wir wollen hier nicht auf den allgemeinen Konvergenzbegriff für MSV eingehen, sondern nur eine Bedingung aufstellen, die ein „vernünftiges" (= für große Klassen von AWP konvergentes) Verfahren offensichtlich erfüllen muß.

Wir betrachten dazu ein AWP auf dem Intervall $[0,1]$. Die Stützstellen seien gegeben durch Unterteilung von $[0,1]$ in m Intervalle gleicher Länge: $x_k = kh$ mit $h = 1/m$, $m \in \mathbb{N}$. Dann ist $y_k^{(h)}$ eine Approximation zu $y(kh)$, speziell also $y_m^{(h)} = y_m^{(1/m)}$ eine Approximation zu $y(mh) = y(1)$. Für die vom MSV benötigten r Anfangsdaten gilt entsprechend: $y_i^{(1/m)} \simeq y(ih) = y(i/m)$ für $i = 0, \ldots, r-1$. Die Anfangsdaten des Verfahrens sollten also so gewählt sein, daß

$$\lim_{m\to\infty} y_i^{(1/m)} = \lim_{m\to\infty} y(i/m) = y(0) \tag{7.17}$$

für $i = 0, \ldots, r-1$.

Damit ergibt sich folgende Bedingung, die ein „vernünftiges" MSV (7.14) auf jeden Fall erfüllen sollte:

> Wendet man (7.14) auf ein „einfaches" Testproblem $y' = f(x,y)$, $y(0) = y_0$ an, welches eine auf $[0,1]$ differenzierbare eindeutige Lösung $y(x)$ besitzt, und wählt dabei die r Anfangswerte so, daß (7.17) gilt, dann konvergiert das Verfahren punktweise in $x = 1$:
>
> $$\lim_{m\to\infty} y_m^{(1/m)} = y(1). \tag{7.18}$$

Beispiel: Das Eulerverfahren (7.9), angewendet auf das Testproblem $y' = y$, $y(0) = 1$, erfüllt diese Bedingung: Da es sich um ein Einschrittverfahren handelt, ist nur $y_0^{(1/m)} = y(0)$ vorzugeben, (7.17) ist also erfüllt. Das Verfahren liefert die numerischen Lösungen $y_k^{(h)} = (1+h)^k$, wie wir gezeigt haben, also speziell $y_m^{(1/m)} = (1+1/m)^m \to e = y(1)$ für $m \to \infty$.

Wir definieren zu einem gegebenen MSV (7.14) das Polynom

$$\rho(z) := \sum_{i=0}^{r} a_i z^i. \tag{7.19}$$

mit dessen Hilfe sich nun die Stabilitätsbedingung formulieren läßt, die ein MSV erfüllen muß, um zumindest (punktweise) konvergent im obigen Sinne zu sein.

Satz 7.2 (Stabilität von MSV) *Ein MSV der Form* (7.14) *sei konvergent im Sinne von* (7.17), (7.18). *Dann gilt für das Polynom $\rho(z)$:*

1. *Jede Nullstelle von ρ ist vom Betrag ≤ 1, d.h. $\rho(z_0) = 0$ impliziert $|z_0| \leq 1$.*
2. *Jede mehrfache Nullstelle von ρ ist vom Betrag < 1, d.h. $\rho(z_0) = \rho'(z_0) = 0$ impliziert $|z_0| < 1$.*

Sind diese beiden Bedingungen erfüllt, so heißt das MSV stabil, *andernfalls* instabil.

Beweis: Es sei $z_0 \in \mathbb{C}$ eine Nullstelle von ρ. Wir zeigen die beiden Bedingungen, indem wir das spezielle Test–AWP $y' = 0$, $y(0) = 0$ zu verschiedenen Anfangswerten betrachten und jeweils beweisen, daß das Verfahren nicht konvergieren kann, falls eine der Bedingungen verletzt ist. Das MSV für dieses AWP ist einfach

$$\sum_{i=0}^{r} a_i y_{k+i}^{(h)} = 0.$$

1. Als Anfangswerte wählen wir zunächst

$$y_i^{(1/m)} = h z_0^i = z_0^i / m, \quad i = 0, \ldots, r-1, \tag{7.20}$$

welche sicherlich die Bedingung (7.17) erfüllen (Dazu muß man das MSV derart erweitern, daß komplexe Anfangswerte zugelassen werden, was aber offensichtlich möglich ist). Wegen

$$\sum_{i=0}^{r} a_i (h z_0^{k+i}) = z_0^k h \sum_{i=0}^{r} a_i z_0^i = z_0^k h \rho(z_0) = 0$$

ist demnach die Folge $y_k^{(h)} = h z_0^k$, $k = r, \ldots, m$ genau die Folge, welche man mit dem MSV zu den r Anfangswerten (7.20) erhält. Nach der Vorausetzung gilt (7.18), also

$$\lim_{m\to\infty} y_m^{(1/m)} = \lim_{m\to\infty} \frac{z_0^m}{m} = y(1) = 0,$$

woraus $|z_0| \leq 1$ folgt.

2. Als Anfangswerte wählen wir diesmal ($z_0 = 0$ kann ausgeschlossen werden, in dem Fall ist nichts mehr zu zeigen)

$$y_i^{(1/m)} = h i z_0^{i-1} = i z_0^{i-1} / m, \quad i = 0, \ldots, r-1, \tag{7.21}$$

welche ebenfalls (7.17) erfüllen. Wegen

$$\begin{aligned}\sum_{i=0}^{r} a_i (h(k+i) z_0^{k+i-1}) &= \sum_{i=0}^{r} a_i h \frac{dz^{k+i}}{dz}\bigg|_{z=z_0} = \frac{d}{dz} h \sum_{i=0}^{r} a_i z^{k+i}\bigg|_{z=z_0} = \frac{d}{dz} h z^k \sum_{i=0}^{r} a_i z^i\bigg|_{z=z_0} \\ &= \frac{d}{dz} h z^k \rho(z)\bigg|_{z=z_0} = h k z_0^{k-1} \rho(z_0) + h z_0^k \rho'(z_0) = 0\end{aligned}$$

ist hier die Folge $y_k^{(h)} = hkz_0^{k-1} = kz_0^{k-1}/m$, $k = r, \ldots, m$ genau die Folge, welche man mit dem MSV zu den r Anfangswerten (7.21) erhält. Auch hier gilt (7.18), also

$$\lim_{m\to\infty} y_m^{(1/m)} = \lim_{m\to\infty} z_0^{m-1} = y(1) = 0,$$

woraus $|z_0| < 1$ folgt.

$\square$

Beispiel:

1. Wir betrachten noch einmal das Verfahren (7.16). Das zugehörige Polynom ist hier $\rho(z) = z^2 + 4z - 5 = (z-1)(z+5)$, welches offensichtlich die Stabilitätsbedingungen verletzt. Das Verfahren ist instabil und damit unbrauchbar, wie unsere numerischen Ergebnisse schon vermuten ließen.

2. Die Verfahren von Adams-Bashforth sind stabil: Für die Koeffizienten gilt $a_r = 1$, $a_{r-1} = -1$, $a_i = 0$ für $i = 0, \ldots, r-2$. Das zugehörige Polynom ist demnach $\rho(z) = z^r - z^{r-1} = z^{r-1}(z-1)$. ρ hat eine einfache Nullstelle $z_0 = 1$ und eine mehrfache Nullstelle $z_0 = 0$.

Bemerkung: Der Begriff der Stabilität eines MSV ist verschieden vom allgemeinen Begriff der numerischen Stabilität eines Algorithmus, was die Fortpflanzung von Rundungsfehlern betrifft (vgl. Kap. 1). Es ist eine rein algebraische Bedingung, und die schlechten Ergebnisse des Verfahrens (7.16) wären auch bei exakter Rechnung, d.h. ohne Rundungsfehler, aufgetreten.
Umgekehrt kann ein Algorithmus für ein im obigen Sinne stabiles MSV durchaus numerisch instabil sein.

Man wird vermuten, daß eine weitere notwendige Bedingung für die Konvergenz eines MSV ist, daß das Verfahren eine Konsistenzordnung von (mindestens) $O(h)$ hat. Diese Vermutung erweist sich in der Tat als richtig. Der Beweis hierzu verläuft ähnlich zum Beweis des vorigen Satzes und soll hier nicht gebracht werden.
Umgekehrt kann man zeigen, daß Stabilität und Konsistenz auch hinreichend dafür ist, daß ein MSV überall punktweise konvergiert. Es gilt also insgesamt:

Ein lineares MSV ist genau dann konvergent, wenn es stabil und konsistent ist.

7.5 Konstruktion von Einschrittverfahren höherer Ordnung

Lineare Mehrschrittverfahren lassen sich zwar bis zu beliebiger Ordnung in h entwickeln. Ihr Nachteil ist aber, wie schon erwähnt, daß man zu Beginn des Verfahrens r Startwerte benötigt, wo nur ein Startwert durch das AWP vorgegeben ist. Das einzig uns bisher bekannte Verfahren, welches zur Berechnung eines neuen Wertes nur den zuletzt berechneten Wert benötigt, das Eulerverfahren, ist nur von der Ordnung $O(h)$. In diesem Abschnitt zeigen wir, wie sich die Vorteile beider Methoden miteinander kombinieren lassen. Man erhält so Einschrittverfahren höherer Ordnung. Dieser Vorteil wird aber dadurch bezahlt, daß man pro Schritt mehrere Auswertungen der rechten Seite f machen muß und die Fehleranalyse aufwendiger ist.

Die Konstruktion des Eulerverfahrens beruhte im wesentlichen auf der Approximation

$$y(x_{k+1}) \simeq y(x_k) + hf(x_k, y(x_k)). \tag{7.22}$$

Wir wollen versuchen, den Term auf der rechten Seite zu erweitern, ohne daß „große" Änderungen entstehen. Dabei nutzen wir im Prinzip nur aus, daß für eine stetige Funktion $g(x) \simeq g(y)$, falls $x \simeq y$.

Für alle $c, a \in [0,1]$ gilt

$$\begin{aligned} y(x_{k+1}) &\simeq y(x_k) + h[cf(x_k, y(x_k)) + (1-c)f(x_k, y(x_k))] \\ &\simeq y(x_k) + h[cf(x_k, y(x_k)) + (1-c)f(x_k + ah, y(x_k + ah))] \\ &\simeq y(x_k) + h[cf(x_k, y(x_k)) + (1-c)f(x_k + ah, y(x_k) + ahf(x_k, y(x_k)))]. \end{aligned}$$

Dabei haben wir im letzten Schritt

$$y(x_k + ah) \simeq y(x_k) + ahf(x_k, y(x_k)),$$

also im Prinzip wieder das Eulerverfahren, benutzt. Das gesuchte Verfahren ergibt sich jetzt daraus, daß „$\simeq$" durch „$=$" und $y(x_k)$ durch y_k ersetzt wird. Wir erhalten

$$y_{k+1} = y_k + h[cf(x_k, y_k) + (1-c)f(x_k + ah, y_k + ahf(x_k, y_k))]. \tag{7.23}$$

Die bisher freien Parameter c, a müssen nun so bestimmt werden, daß man ein Verfahren möglichst hoher Ordnung erhält, d.h. der Fehler

$$e(h) = y(x+h) - y(x) - h[cf(x, y(x)) + (1-c)f(x + ah, y(x) + ahf(x, y(x)))]$$

den man erhält, wenn man die exakte Lösung $y(x)$ in (7.23) einsetzt (vgl. die Herleitung der Definition der Konsistenzordnung für lineare MSV im vorigen Abschnitt), muß möglichst klein sein. Für $c = 1$, den Fall des Eulerverfahrens, gilt schon $e(h) = O(h^2)$. Wir sollten also c, a so wählen können, daß mindestens $e(h) = O(h^3)$ dabei herausspringt, wir also ein Einschrittverfahren 2. Ordnung in h erhalten. Wir entwickeln dazu $e(h)$ in einer Taylorreihe um $h = 0$.

Zur Abkürzung lassen wir dabei die Argumente x weg, d.h. wir schreiben

$$y = y(x), \quad y' = y'(x), \quad y'' = y''(x), \quad f = f(x, y(x)), \quad f_x = \frac{\partial f}{\partial x}(x, y(x)), \quad \text{usw.}$$

Zunächst erhalten wir durch Ableiten der DGL $y' = f$

$$y'' = f_x + f_y y'.$$

Weiter gilt

$$\begin{aligned} y(x+h) &= y + hy' + \frac{h^2}{2}y'' + O(h^3), \\ f(x+ah, y+ahf) &= f + f_x ah + f_y ahf + O(h^2) = y' + ah(f_x + f_y y') + O(h^2) \\ &= y' + ahy'' + O(h^2). \end{aligned}$$

Damit ist

$$\begin{aligned} e(h) &= y(x+h) - y - h[cf + (1-c)f(x+ah, y+ahf)] \\ &= y + hy' + \frac{h^2}{2}y'' + O(h^3) - y - h[cy' + (1-c)(y' + ahy'' + O(h^2))] \\ &= \frac{h^2}{2}y'' - h^2(1-c)ay'' + O(h^3) \\ &= (\tfrac{1}{2} - (1-c)a)h^2 y'' + O(h^3). \end{aligned}$$

Das Verfahren ist also von 2. Ordnung in h, falls

$$\frac{1}{2} - (1-c)a = 0$$

gilt. Eine Lösung hiervon ist z.B. $c = \frac{1}{2}$, $a = 1$. Das Verfahren lautet hier

$$y_{k+1} = y_k + \frac{h}{2}[f(x_k, y_k) + f(x_k + h, y_k + hf(x_k, y_k))],$$

oder übersichtlicher in algorithmischer Schreibweise

$$k_1 = f(x_k, y_k), \quad k_2 = f(x_k + h, y_k + hk_1), \quad y_{k+1} = y_k + \frac{h}{2}[k_1 + k_2]. \tag{7.24}$$

Dies ist das sog. *Verfahren von Heun.* Ein anderes Verfahren erhält man für $c = 0$, $a = \frac{1}{2}$:

$$y_{k+1} = y_k + h[f(x_k + 0.5h, y_k + 0.5hf(x_k, y_k))]$$

oder

$$k_1 = f(x_k, y_k), \quad y_{k+1} = y_k + hf(x_k + 0.5h, y_k + 0.5hk_1). \tag{7.25}$$

Dies ist das sog. *modifizierte Euler-Verfahren.*

Auf ähnliche Art und Weise können im Prinzip Einschrittverfahren beliebig hoher Ordnung hergeleitet werden. Wie man sich denken kann, werden die entsprechenden Terme bei der Herleitung entsprechend komplizierter. Hier kann man aber z.B. auf geeignete Softwarepakete zur Symbolmanipulation (Mathematica, Maple,...) zurückgreifen, die in der Lage sind, obige Taylorexpansionen automatisch durchzuführen. Wir wollen dies zum Abschluß am Beispiel des klassischen Runge-Kutta-Verfahrens 4. Ordnung

$$k_1 = f(x_k, y_k), \quad k_2 = f\left(x_k + \frac{1}{2}h, y_k + \frac{1}{2}hk_1\right),$$
$$k_3 = \left(x_k + \frac{1}{2}h, y_k + \frac{1}{2}hk_2\right), \quad k_4 = f(x_k + h, y_k + hk_3),$$
$$y_{k+1} = \frac{h}{6}[k_1 + 2k_2 + 2k_3 + k_4]$$

demonstrieren. Dieses Verfahren ist eines der am meisten verwendeten Methoden zu Lösung von Einschrittproblemen. Wir wollen Mathematica dazu benutzen, um zu zeigen, daß das Verfahren tatsächlich von 4. Ordnung ist, daß also für den entsprechenden Fehlerterm $e(h) = O(h^5)$ gilt. Dazu geben wir die Koeffizienten $k_1, \ldots, k_4$ einfach in das Programm ein ein, bilden daraus $e(h)$ und lassen diesen Term in eine Taylorreihe um $h = 0$ bis zur einschließlich 4. Potenz von h ausgeben. Vorher müssen wir noch die DGL $y' = f$ sowie ihre Ableitungen als Regeln eingeben, die bei der Taylorexpansion zur Vereinfachung von Termen benutzt werden sollen.

Die folgende Beispielsession zeigt, daß $e(h)$ tatächlich von 5. Ordnung in h ist. Wir haben also mit Hilfe des Computers die $O(h^4)$-Fehlerordnung des Verfahrens bewiesen (vorausgesetzt, man kann davon ausgehen, daß die Berechnungen in Mathematica fehlerfrei erfolgen).

```
galerkin% math
Mathematica 3.0 for Solaris
Copyright 1988-97 Wolfram Research, Inc.
 -- Motif graphics initialized --

In[1]:= k1:=f[x,y[x]] (* Eingabe der Koeffizienten des Runge-Kutta-Verf. *)

In[2]:= k2:=f[x+h/2,y[x]+h*k1/2]

In[3]:= k3:=f[x+h/2,y[x]+h*k2/2]

In[4]:= k4:=f[x+h,y[x]+h*k3]

In[5]:= err:=y[x+h]-y[x]-h*(k1+2*k2+2*k3+k4)/6 (* Lokaler Diskr.-Fehler *)

In[6]:= g[x_]:=f[x,y[x]] (* Bringe Mathematica die DGL bei *)
```

```
In[7]:= y'[x]=g[x]; y''[x]=g'[x]; y'''[x]=g''[x]; y''''[x]=g'''[x];

In[8]:= Series[err,{h,0,4}] (* Taylorreihe von err um h=0 bis 4. Potenz *)

              5
Out[8]= O[h]

In[9]:= Quit
```