Numerische Verfahren

Numerische Methoden von gewöhnlichen Differentialgleichungen (AWP)

Prof. Dr.-Ing. K. Warendorf, Prof. Dr.-Ing. P. Wolfsteiner

Hochschule für Angewandte Wissenschaften München
Fakultät 03

WS 13/14

Inhalt

Einleitung

Gegeben sei das Anfangswertproblem:

$$\dot{y} = f(t, y), \quad y(t_0) = y_0$$

Aufgabe

Berechnung eines Näherungswertes an der Stelle $t_n = b$.

$$y_n \approx y(b)$$

mit der Diskretisierung:
Zerlegung des Integrationsintervalles $[a = t_0, b]$ in: $a = t_0 < t_1 < \cdots < t_n = b$
Feste Schrittweite: $h = \frac{b-a}{n} = t_i - t_{i-1}$

Achtung: Der Stabilitätsbereich eines expliziten Einschrittverfahrens und damit die Schrittweite h sind beschränkt. Für *zu große* h divergiert das Verfahren.
h ist auch nach unten beschränkt, da sonst die Rundungsfehler überwiegen!

Explizites Euler-Verfahren

Fortschreiterichtung: Steigung in t_{i-1}

Euler-Verfahren

Gegeben: Anfangswert: (t_0, y_0), Schrittweite: h

$$y_i = y_{i-1} + h\, f(t_{i-1}, y_{i-1}), \quad i = 1, ..., n$$

Globaler Diskretisierungsfehler: $O(h)$.
$\Rightarrow$ Verfahren 1. Ordnung (Konvergenz)

Runge-Kutta-Verfahren 4. Ordnung

Fortschreiterichtung: mittlere Steigung: Berechnet aus den 4 Steigungen in t_{i-1}, t_i und 2 Punkten in der Mitte

Runge-Kutta-Verfahren 4. Ordnung

Gegeben: Anfangswert: (t_0, y_0), Schrittweite: h

$$
\begin{aligned}
y_i &= y_{i-1} + \tfrac{1}{6}\,(k_1 + 2k_2 + 2k_3 + k_4), \quad i = 1, \ldots, n \\
k_1 &= h\,f(t_{i-1}, y_{i-1}) \\
k_2 &= h\,f(t_{i-1} + \tfrac{1}{2}h,\, y_{i-1} + \tfrac{1}{2}k_1) \\
k_3 &= h\,f(t_{i-1} + \tfrac{1}{2}h,\, y_{i-1} + \tfrac{1}{2}k_2) \\
k_4 &= h\,f(t_{i-1} + h,\, y_{i-1} + k_3)
\end{aligned}
$$

Globaler Diskretisierungsfehler: $O(h^4)$.
$\Rightarrow$ Verfahren 4. Ordnung (Konvergenz)

Matlab: `ode45(...)`. Dieses und entsprechende Verfahren sind mit automatischer Schrittweitensteuerung implementiert.

Implizites Euler-Verfahren

Implizites Verfahren: Die Funktion f wird nicht an dem schon berechneten Punkt (t_{i-1}, y_{i-1}) ausgewertet, sondern an dem *neuen* Punkt (t_i, y_i)

Implizites Euler-Verfahren

Gegeben: Anfangswert: (t_0, y_0), Schrittweite: h

$$y_i = y_{i-1} + h\,f(t_i, y_i), \quad i = 1, ..., n$$

Globaler Diskretisierungsfehler: $O(h)$.
$\Rightarrow$ Verfahren 1. Ordnung (Konvergenz) Das Verfahren ist unbeschränkt absolut stabil (A-stabil), d.h. h unterliegt aufgrund der Stabilität keiner Beschränkung nach oben.

Es entsteht eine (nicht)-lineare Gleichung (bzw. System). Diese muss mit geeigneten (numerischen) Verfahren (z.B.) Newton-Verfahren, Fixpunktiteration) gelöst werden. Implizite Verfahren werden insbesondere zur Lösung von steifen DGL-Systemen genutzt.

DGL-Systeme 1. Ordnung

Die numerischen Verfahren zur Lösung von DGLn 1. Ordnung lassen sich entsprechend auf Systeme 1. Ordnung übertragen. Die skalaren Operationen müssen als Vektoroperationen aufgefasst werden.

DGL-System 1. Ordnung

$$\left.\begin{array}{rcl} \dot{y}_1 & = & f_1(t, y_1, \ldots, y_n) \\ \dot{y}_2 & = & f_2(t, y_1, \ldots, y_n) \\ & \vdots & \\ \dot{y}_n & = & f_n(t, y_1, \ldots, y_n) \end{array}\right\} : \text{Vektorschreibweise: } \dot{\vec{y}} = F(t, \vec{y})$$

mit den Anfangsbedingungen
$y_1(t_0) = y_1^{(0)}, y_2(t_0) = y_2^{(0)}, \ldots, y_n(t_0) = y_n^{(0)}$ bzw. $\vec{y}(t_0) = \vec{y}_0$.

Explizites Euler-Verfahren für DGL-System 1. Ordnung

Euler-Verfahren

Gegeben: Anfangswerte: $y_1(t_0) = y_1^{(0)}, y_2(t_0) = y_2^{(0)}, \ldots, y_n(t_0) = y_n^{(0)}$
bzw. $\vec{y}(t_0) = \vec{y}_0$, Schrittweite: h

$$\vec{y}_i = \vec{y}_{i-1} + h \cdot F(t_{i-1}, \vec{y}_{i-1}), \quad i = 1, ..., n$$

Reduktion von DGLn höherer Ordnung auf ein System 1. Ordnung (Zustandsform)

Die numerische Lösung von DGLn höherer Ordnung oder von Systemen, erfordert ein Umschreiben in die sogenannte Zustandsform (System 1. Ordnung).
DGL n-ter Ordnung:

$$y^{(n)} = f(t, y, \dot{y}, \dots, y^{(n-1)}), \text{ Anfangsbedingungen: } y(t_0), \dot{y}(t_0), \dots, y^{(n-1)}(t_0)$$

Zustandsform

Zustandsgrößen: $z_1 = y, z_2 = \dot{y}, z_3 = \ddot{y}, \dots z_n = y^{(n-1)}$

Umformung in ein System 1. Ordnung:

$$\begin{array}{lcll}
\dot{z}_1 & = & z_2 & , \quad z_1(t_0) = y(t_0) \\
\dot{z}_2 & = & z_3 & , \quad z_2(t_0) = \dot{y}(t_0) \\
 & \vdots & & , \quad \vdots \\
\dot{z}_{n-1} & = & z_n & , \quad z_{n-1}(t_0) = y^{(n-2)}(t_0) \\
\dot{z}_n & = & f(t, z_1, \dots, z_n) & , \quad z_n(t_0) = y^{(n-1)}(t_0)
\end{array}$$

Zustandsform für DGLn 2. Ordnung I

DGL 2. Ordnung:

$$\ddot{y} = f(t, y, \dot{y}) \text{ mit AB: } y(t_0) = y_1^{(0)}, \dot{y}(t_0) = y_2^{(0)}$$

lässt sich durch $z_1 = y, z_2 = \dot{y}$ überführen in die Zustandsform:

$$\begin{aligned} \dot{z}_1 &= z_2 &, \quad z_1(t_0) = z_1^{(0)} = y_1^{(0)} \\ \dot{z}_2 &= f(t, z_1, z_2) &, \quad z_2(t_0) = z_2^{(0)} = y_2^{(0)} \end{aligned} \tag{1}$$

Vektorielle Schreibweise: $\dot{\vec{z}} = \begin{pmatrix} z_2 \\ f(t, \vec{z}) \end{pmatrix}$

Zustandsform für DGLn 2. Ordnung II

Euler-Verfahren für obiges System (1)

Gegeben: Anfangswerte: $(t_0, z_1^{(0)})$,$(t_0, z_2^{(0)})$ Schrittweite: h, Schrittanzahl: m

$$
\begin{aligned}
z_1^{(i)} &= z_1^{(i-1)} + h \cdot z_2^{(i-1)} \\
z_2^{(i)} &= z_2^{(i-1)} + h \cdot f(t_{i-1}, z_1^{(i-1)}, z_2^{(i-1)}) \quad i = 1, ..., m
\end{aligned}
$$

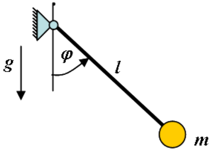
Vektorielle Schreibweise: $\vec{z}_i = \vec{z}_{i-1} + h \cdot \dot{\vec{z}}_{i-1}$

Zustandsform einer DGL 2. Ordnung: Punktpendel

Nichtlineare DGL 2. Ordnung in φ:

$$m \cdot l^2 \cdot \ddot{\varphi}(t) = -m \cdot g \cdot l \cdot \sin\left(\varphi(t)\right)$$

$$\Rightarrow \ddot{\varphi}(t) = -\frac{g}{l} \cdot \sin\left(\varphi(t)\right)$$



Anfangsbedingungen:
Anfangswinkel: $\varphi(0)$, Anfangswinkelgeschwindigkeit: $\dot{\varphi}(0)$

Zustandsform

Zustandsgrößen:
Lage: $z_1(t) = \varphi(t)$, Geschwindigkeit: $z_2(t) = \dot{\varphi}(t)$
Umformung in ein System 1. Ordnung:

$$\begin{aligned} \dot{z}_1 &= z_2 & , \quad & z_1(0) = \varphi(0) \\ \dot{z}_2 &= -\tfrac{g}{l} \cdot \sin\left(z_1(t)\right) & , \quad & z_2(0) = \dot{\varphi}(0) \end{aligned}$$

Zustandsform eines Systems 2. Ordnung: I

System von DGL 2. Ordnung in $y_1, \dots, y_n$:

$$\begin{aligned} \ddot{y}_1 &= f_1(t, y_1, \dots, y_n, \dot{y}_1 \dots, \dot{y}_n) \\ &\vdots \\ \ddot{y}_n &= f_n(t, y_1, \dots, y_n, \dot{y}_1 \dots, \dot{y}_n) \end{aligned}$$

Anfangsbedingungen:

$$y_1(t_0), \dots, y_n(t_0),$$

$$\dot{y}_1(t_0), \dots, \dot{y}_n(t_0)$$

Zustandsform eines Systems 2. Ordnung: II

Zustandsform

Zustandsgrößen:

$$z_1 = y_1, \dots, z_n = y_n$$

$$z_{n+1} = \dot{y}_1, \dots, z_{2n} = \dot{y}_n$$

Umformung in ein System 1. Ordnung:

$$\begin{array}{lcll}
\dot{z}_1 & = & z_{n+1} & , \quad z_1(t_0) = y(t_0) \\
 & \vdots & & , \quad \vdots \\
\dot{z}_n & = & z_{2n} & , \quad z_n(t_0) = y_n(t_0) \\
\hline
\dot{z}_{n+1} & = & f_1(t, z_1, \dots, z_n, z_{n+1} \dots, z_{2n}) & , \quad z_{n+1}(t_0) = \dot{y}_1(t_0) \\
 & \vdots & & , \quad \vdots \\
\dot{z}_{2n} & = & f_n(t, z_1, \dots, z_n, z_{n+1} \dots, z_{2n}) & , \quad z_{2n}(t_0) = \dot{y}_n(t_0)
\end{array}$$

Zustandsform eines Systems 2. Ordnung: Federpendel I

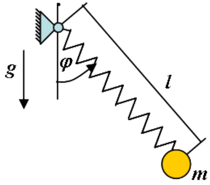
System von DGL 2. Ordnung in φ, l:

$$\begin{aligned} m\cdot\left(l(t)\ddot{\varphi}(t)+2\dot{l}(t)\dot{\varphi}(t)\right) &= -mg\sin(\varphi(t)) \\ m\cdot\left(\ddot{l}(t)-l(t)\dot{\varphi}^2(t)\right) &= -c(l(t)-l_F)+mg\cos(\varphi(t)) \end{aligned}$$

$\Rightarrow$

$$\ddot{\varphi}(t)=\frac{1}{l}\left(-g\sin(\varphi(t))-2\dot{l}(t)\dot{\varphi}(t)\right)$$

$$\ddot{l}(t)=-\frac{c}{m}(l(t)-l_F)+g\cos(\varphi(t))+l(t)\dot{\varphi}^2(t)$$



Anfangsbedingungen:
Anfangslage: $\varphi(0)$, $l(0)$
Anfangswinkelgeschwindigkeit: $\dot{\varphi}(0)$, $\dot{l}(0)$

Zustandsform eines Systems 2. Ordnung: Federpendel II

Zustandsform

Zustandsgrößen:
Lage: $z_1(t) = \varphi(t)$, $z_2(t) = l(t)$ Geschwindigkeit: $z_3(t) = \dot{\varphi}(t)$, $z_4(t) = \dot{l}(t)$
Umformung in ein System 1. Ordnung:

$$
\begin{aligned}
\dot{z}_1 &= z_3 &&, \quad z_1(0) = \varphi(0)\\
\dot{z}_2 &= z_4 &&, \quad z_2(0) = l(0)\\
\dot{z}_3 &= -\frac{1}{z_2}(g\sin(z_1) + 2z_4 z_3) &&, \quad z_3(0) = \dot{\varphi}(0)\\
\dot{z}_4 &= -\frac{c}{m}(z_2 - l_F) + g\cos(z_1) + z_2 z_3^2 &&, \quad z_4(0) = \dot{l}(0)
\end{aligned}
$$

Zustandsform eines Systems 2. Ordnung: Massenmatrix I

Falls das DGL-System nicht in expliziter Form gegeben ist und sich nur schwer nach den 2. Ableitungen auflösen lässt (starke Kopplung), ist eine andere Vorgehensweise zu empfehlen.

System von 2 DGL (linear in den 2. Ableitungen)

$$\begin{aligned} a_{11}\ddot{y}_1 + a_{12}\ddot{y}_2 &= f_1(t, y_1, y_2, \dot{y}_1, \dot{y}_2) \\ a_{21}\ddot{y}_1 + a_{22}\ddot{y}_2 &= f_2(t, y_1, y_2, \dot{y}_1, \dot{y}_2) \end{aligned}$$

a_{ij} können von $t, y_1, y_2, \dot{y}_1$ und $\dot{y}_2$ aber **nicht** von $\ddot{y}_1$ und $\ddot{y}_2$ abhängen.

Zustandsform eines Systems 2. Ordnung: Massenmatrix II

Zustandsform

Zustandsgrößen: $z_1 = y_1, z_2 = y_2, z_3 = \dot{y}_1, z_4 = \dot{y}_2$

$$\begin{aligned}
\dot{z}_1 &= z_3 &,\quad z_1(t_0) &= y_1(t_0)\\
\dot{z}_2 &= z_4 &,\quad z_2(t_0) &= y_2(t_0)\\
a_{11}\dot{z}_3 + a_{12}\dot{z}_4 &= f_1(t, z_1, z_2, z_3, z_4) &,\quad z_3(t_0) &= \dot{y}_1(t_0)\\
a_{21}\dot{z}_3 + a_{22}\dot{z}_4 &= f_2(t, z_1, z_2, z_3, z_4) &,\quad z_4(t_0) &= \dot{y}_2(t_0)
\end{aligned}$$

Matrix-Schreibweise:

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a_{11} & a_{12} \\ 0 & 0 & a_{21} & a_{22} \end{pmatrix}}_{\textbf{\textit{Massenmatrix:}}\ M=M(t,\vec{z})} \cdot \underbrace{\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{pmatrix}}_{\dot{\vec{z}}} = \underbrace{\begin{pmatrix} z_3 \\ z_4 \\ f_1(t, z_1, z_2, z_3, z_4) \\ f_2(t, z_1, z_2, z_3, z_4) \end{pmatrix}}_{F(t,\vec{z})}$$

Matlab kann diese gekoppelten Differentialgleichungen lösen. Dazu muss M mit dem Befehl ***odeset('Mass', @Matrix)*** bekannt gemacht werden.