Numerik gewöhnlicher Differentialgleichungen

1 Grundidee

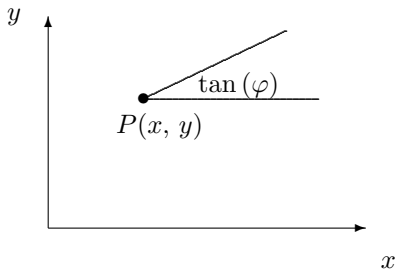
Richtungsfeld und Vektorfeld

Gegeben ist eine Differentialgleichung oder ein System von Differentialgleichungen (Dgl):

$$y'(x) = f(x, y(x)) \tag{1}$$

mit der Anfangsbedingung, bzw. den Anfangsbedingungen (AB):

$$y(x_0) = y_0 \tag{2}$$



Jedem Punkt $P(x, y)$ des Definitionsbereichs von $f(x, y)$ wird durch die Differentialgleichung (1) eindeutig eine Steigung $y' = tan(\varphi)$ und damit eine Richtung zugeordnet. ($\rightarrow$ *Richtungsfeld*).

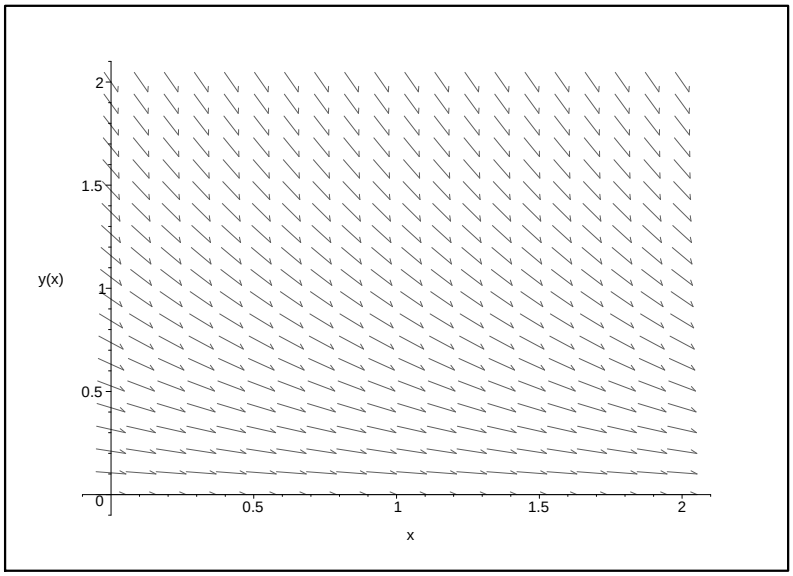


Abbildung 1: Richtungsfeld der Differentialgleichung $y' = -y$ mit dem Maple - Befehl `dfieldplot(DGL,[y(x)],x=0..2,y=0..2)` im File `rf_1.mws`

Geometrische Interpretation:

Die Differentialgleichung (1) lösen heisst jetzt, eine Kurve $y = y(x)$ so zeichnen, dass sie in jedem Punkt die durch das Richtungsfeld vorgeschriebene Richtung hat. So bekommen wir eine sogenannte *Trajektorie*.

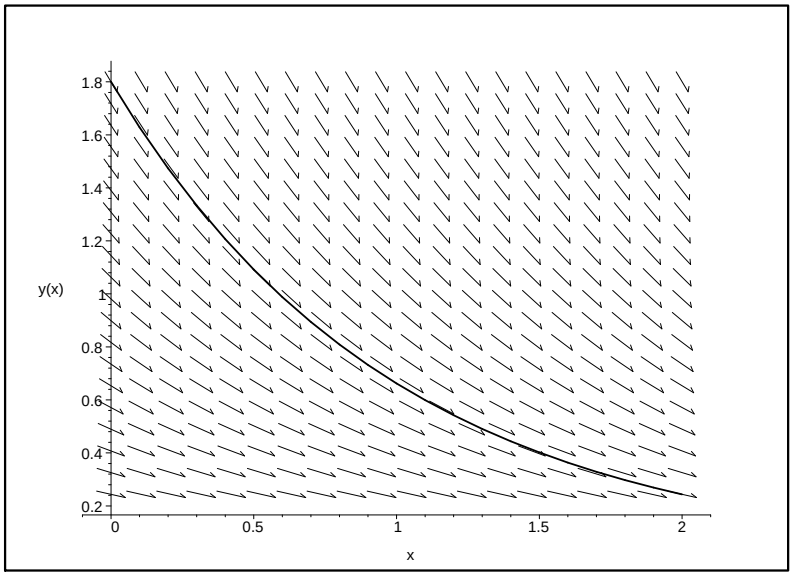


Abbildung 2: Richtungsfeld der Differentialgleichung mit einer Trajektorie
`phaseportrait(D(y)(x)=-y(x), y(x), x = 0..2, [[y(0) = 1.8]], colour=black, linecolour=black);`

Mit dem Matlab-Befehl `quiver` erhalten wir eine graphische Darstellung durch Pfeile, dabei gilt:

- Länge der Pfeile ist proportional zur Geschwindigkeit.
- Richtung der Pfeile ist tangential längs einer Trajektorie.

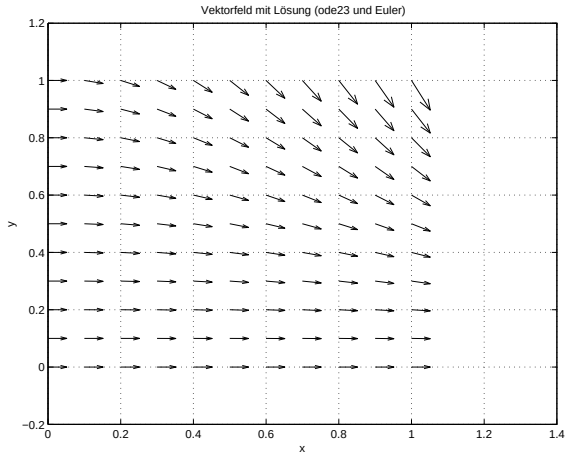


Abbildung 3: Vektorfeld der Differentialgleichung (3)
mit dem Matlab - Befehl `quiver` im File `vektorfeld1.m`

Mit einer solchen Darstellung bekommen wir das sogenannte *Vektorfeld* der Differentialgleichung.

Ziel: $y(x)$ soll für weitere Werte des Arguments x berechnet werden und zwar zunächst an einer Stelle x_1 in der Nachbarschaft von x_0

$$h = x_1 - x_0$$

Ein erster Schritt mit der Schrittweite h wird durchgeführt:

Idee:

Die Differentialgleichung (1) wird gelöst, indem beide Seiten über das Intervall $[x_0, x_1]$ integriert werden:

$$\int_{x_0}^{x_1} y'(x)\,dx = \int_{x_0}^{x_1} f(x,y)\,dx$$

$$y(x_1) - y(x_0) = \int_{x_0}^{x_1} f(x,y)\,dx$$

und damit

$$y(x_1) = y(x_0) + \int_{x_0}^{x_1} f(x,y)\,dx$$

Da $y = y(x)$ unbekannt ist, sind wir auf eine *approximative*, d.h. numerische Integration angewiesen.

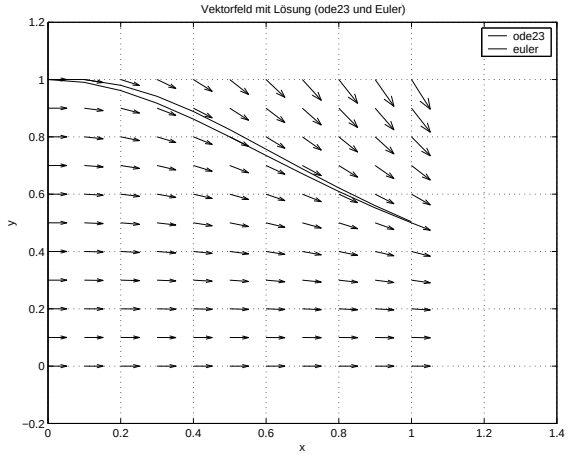


Abbildung 4: Vektorfeld von (3) mit gerechneter Lösung
im File `vektorfeld1.m`

2 Einschrittmethoden

2.1 Die Methode von Euler

Das gegebene Problem ist als Anfangswertproblem (AWP) definiert:

$$\begin{aligned} y'(x) &= f(x,y) \\ y(x_0) &= y_0 \end{aligned}$$

Gesucht ist dabei $y = y(x)$.

Einfachste numerische Methode: approximiere die Lösungskurve $y = y(x)$ durch die Tangente im Punkt $P(x_0, y_0)$.
(Linearisierung der Funktion, vgl. Differential von f).

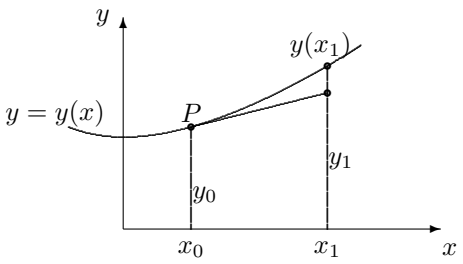


Abbildung 5: Methode von Euler, erster Schritt von x_0 nach $x_1 = x_0 + h$

Mit der Schrittweite h und den zugehörigen äquidistanten Stützstellen

$$x_k = x_0 + k \cdot h \qquad k = 1, 2, \ldots$$

bekommen wir die Näherungen y_k für die exakten Werte $y(x_k)$ mit der Rekursionsformel:

$$y_{k+1} = y_k + h \cdot f(x_k, y_k) \qquad k = 0, 1, 2, \ldots$$

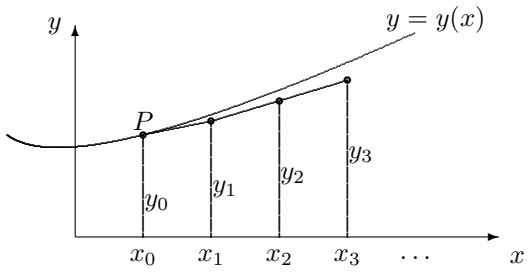


Abbildung 6: Methode von Euler, Schrittweite h, $x_k = x_0 + k \cdot h$, $k = 0, 1, 2, \ldots$

Die Integrationsmethode von Euler benutzt in den einzelnen Näherungspunkten (x_k, y_k) die Steigung des Richtungsfelds (durch die Differentialgleichung definiert) dazu, den nächsten Näherungswert y_{k+1} zu bestimmen.

Wegen der anschaulich geometrischen Konstruktion der Näherung wird das Verfahren auch als *Polygonzugmethode* bezeichnet.

Diese Methode stellt den einfachsten Repräsentanten einer *Einschrittmethode* dar, die zur Berechnung der Näherung y_{k+1} an der Stelle $x_{k+1} = x_k + h$ einzig den bekannten Näherungswert y_k an der Stützstelle x_k verwendet.

Beispiel mit Lösung zu Testzwecken

$$y'(x) = -2xy^2 \tag{3}$$
$$y(0) = 1 \tag{4}$$

mit der exakten Lösung:

$$y(x) = \frac{1}{x^2+1}$$

Mit der Methode von Euler erhalten wir die in folgender Tabelle zusammengestellten Näherungswerte y_k für verschiedene Schrittweiten h an den selben diskreten Stützstellen x_k, sowie die zugehörigen Fehler

$$e_k := y(x_k) - y_k$$

		$h = 0.1$		$h = 0.01$		$h = 0.001$	
x_k	$y(x_k)$	y_k	e_k	y_k	e_k	y_k	e_k
0	1.00000	1.00000	-	1.00000	-	1.00000	-
0.1	0.99010	1.00000	-0.00990	0.99107	-0.00097	0.99020	-0.00010
0.2	0.96154	0.98000	-0.01846	0.96330	-0.00176	0.96171	-0.00018
0.3	0.91743	0.94158	-0.02415	0.91969	-0.00226	0.91766	-0.00022
0.4	0.86207	0.88839	-0.02632	0.86448	-0.00242	0.86231	-0.00024
0.5	0.80000	0.82525	-0.02525	0.80229	-0.00229	0.80023	-0.00023
0.6	0.73529	0.75715	-0.02185	0.73727	-0.00198	0.73549	-0.00020

Tabelle 2.1 *Methode von Euler, verschiedene Schrittweiten. Der Fehler nimmt etwa proportional zur Schrittweite h ab.*

Bemerkung 1 *Die Methode von Euler kann meistens* **nur** *für kleine Schrittweiten h gute Näherungswerte liefern.*

2.2 Die Methode der Taylorreihe

Mit der Taylorreihe mit Restglied kann in der Umgebung des Startpunktes (x_0, y_0) eine bedeutend bessere Approximation der gesuchten Lösung $y(x)$ bestimmt werden:

$$y(x) = y(x_0) + \frac{y'(x_0)}{1!}(x-x_0) + \frac{y''(x_0)}{2!}(x-x_0)^2 + \cdots + \frac{y^{(p)}(x_0)}{p!}(x-x_0)^p + R_{p+1}$$

Wird in dieser Formel das Restglied R_{p+1} vernachlässigt, so erhalten wir mit der Schrittweite $h := x_1 - x_0$ den Näherungswert

$$y_1 = y_0 + \frac{y_0'}{1!}h + \frac{y_0''}{2!}h^2 + \cdots + \frac{y_0^{(p)}}{p!}h^p \ ,$$

wobei $y_0^{(m)}$ = Wert der $m-$ten Ableitung im Punkt (x_0, y_0), $m = 0, 1, \ldots, p$.

Und analog in einem allgemeinen Punkt (x_k, y_k) für die Schrittweite $h := x_{k+1} - x_k$ den Näherungswert

$$y_{k+1} = y_k + \frac{y_k'}{1!}h + \frac{y_k''}{2!}h^2 + \cdots + \frac{y_k^{(p)}}{p!}h^p \ ,$$

wobei $y_k^{(m)}$ = Wert der $m-$ten Ableitung im Punkt (x_k, y_k), $m = 0, 1, \ldots, p$.

Diese Methode erfordert die Kenntnis der Ableitungen der Funktion $y(x)$ bis zu einer vorgegebenen Ordnung p an der Stelle x_k. Die zweiten und höheren Ableitungen lassen sich durch sukzessive Differentiation der gegebenen Dgl nach x und wiederholte Substitution von y' bestimmen.

Die entstehenden Ausdrücke in den partiellen Ableitungen der gegebenen Funktion $f(x, y)$ werden rasch kompliziert. Auf diese Weise ist dieses Verfahren nur in einfachen Fällen praktikabel.

Die Methode der Taylorreihe ist aber bei anderer Durchführung oft sehr erfolgreich. Dazu wird die gesuchte Taylorreihe mit unbekannten Koeffizienten c_i in der Form

$$y(x) = y(x_0) + c_1(x - x_0) + c_2(x - x_0)^2 + c_3(x - x_0)^3 + \dots$$

angesetzt. Da dieser Ansatz eine Lösung der gegebenen Differentialgleichung sein soll, wird er in die gegebene Differentialgleichung eingesetzt. Durch Koeffizientenvergleich lassen sich anschliessend Bedingungsgleichungen für die Koeffizienten c_i formulieren, aus denen sich die c_i rekursiv berechnen lassen.

Beispiel:

$$\begin{aligned} y'(x) &= -2xy^2 \\ y(0) &= 1 \end{aligned}$$

Im allgemeinen Näherungspunkt $(x_k,\, y_k)$ lautet der *Ansatz*

$$y(x) = y_k + c_1(x - x_k) + c_2(x - x_k)^2 + c_3(x - x_k)^3 + \dots$$

- Diese Entwicklung wird zusammen mit ihrer ersten Ableitung in die Dgl eingesetzt.
- Anschliessend ist ein Koeffizientenvergleich bzgl. der Potenzen von $h := (x - x_k)$ durchzuführen; deshalb wird in der Dgl auch die Variable $x = x_k + h$ verwendet.
- Wir machen hier eine "Entwicklung um x_k", d.h. eine Entwicklung in einer Umgebung von x_k.

Somit:

$$\begin{aligned} c_1 + 2c_2h + 3c_3h^2 + \dots &= -2(x_k + h) \cdot [y_k + c_1h + c_2h^2 + c_3h^3 + \dots]^2 \\ &= -2x_ky_k^2 + (-2y_k^2 - 4c_1x_ky_k)h + \left\{-4c_1y_k - 2x_k(c_1^2 + 2c_2y_k)\right\} h^2 \\ &\quad + \left\{-2(c_1^2 + 2c_2y_k) - 4x_k(c_1c_2 + c_3y_k)\right\} h^3 + \dots \end{aligned}$$

Koeffizientenvergleich bis und mit zur dritten Potenz in h:

$$\begin{aligned} h^0: \quad c_1 &= -2x_ky_k^2 \\ h^1: \quad c_2 &= -(y_k + 2c_1x_k)y_k \\ h^2: \quad c_3 &= -\tfrac{1}{3}\left\{4c_1y_k + 2x_k(c_1^2 + 2c_2y_k)\right\} \\ h^3: \quad c_4 &= -\left\{\tfrac{1}{2}c_1^2 + c_2y_k + x_k(c_1c_2 + c_3y_k)\right\} \end{aligned}$$

Durch Vorwärtseinsetzen lassen sich die c_i rekursiv berechnen. Der Näherungswert y_{k+1} an der Stelle $x_{k+1} = x_k + h$ ist bei gegebener Schrittweite h somit

$$y_{k+1} = y_k + c_1h + c_2h^2 + c_3h^3 + c_4h^4 \; .$$

Die folgende Tabelle enthält die mit der Schrittweite $h = 0.1$ berechneten Werte y_k, die zugehörigen Koeffizienten c_i, sowie die Fehler $e_k := y(x_k) - y_k$.

Da in der Taylorreihe die Ableitungen bis und mit zur 4-ten Ordnung berücksichtigt sind, stellen die resultierenden y_k im Vergleich zur Methode von Euler bedeutend bessere Näherungen dar.

x_k	y_k	c_1	c_2	c_3	c_4	e_k
0	1.0000000	0.0000000	-1.0000000	0.0000000	1.0000000	-
0.1	0.9901000	-0.1960596	-0.9414743	0.3805494	0.8567973	-0.0000010
0.2	0.9615455	-0.3698279	-0.7823272	0.6565037	0.4997400	-0.0000071
0.3	0.9174459	-0.5050242	-0.5637076	0.7736352	0.0913102	-0.0000147
0.4	0.8620892	-0.5945582	-0.3331480	0.7423249	-0.2247569	-0.0000202
0.5	0.8000218	-0.6400348	-0.1279930	0.6144390	-0.3891673	-0.0000218
0.6	0.7353139	-0.6488238	0.0318205	0.4490115	-0.4195953	-0.0000197
0.7	0.6711567	-0.6306319	0.1421026	0.2897305	-0.3676095	-0.0000158
0.8	0.6097675	-0.5949063	0.2085909	0.1592475	-0.2825582	-0.0000114
0.9	0.5524938	-0.5494489	0.2411714	0.0637248	-0.1966194	-0.0000076
1.0	0.5000047					-0.0000047

Tabelle 2.2 *Tayloralgorithmus mit* $h = 0.1$

- Die Anzahl der Glieder der Taylorreihe lässt sich im Prinzip beliebig erhöhen und damit die Qualität der Approximation steigern.
- *Nachteil*: Für jede Differentialgleichung muss der zugehörige Satz von Rekursionsformeln zuerst bestimmt werden. (hier kann mit Vorteil die Computeralgebra mit MATHEMATICA oder MAPLE zum Einsatz kommen).

2.3 Diskretisationsfehler und Fehlerordnung

Zusammenstellung einiger Aussagen über Fehlerabschätzungen:

Grundkonzept: wir betrachten eine Rechenvorschrift in der Form

$$y_{k+1} = y_k + h \cdot \Phi(x_k, y_k, h) \qquad k = 0, 1, \ldots$$

Die Funktion $\Phi(x_k, y_k, h)$ beschreibt die zugrundeliegende Methode, wie aus den Grössen (x_k, y_k) und h der neue Näherungswert y_{k+1} für x_{k+1} zu bestimmen ist.

$$\begin{aligned} \text{Euler :} \quad & \Phi(x_k, y_k, h) &&= f(x_k, y_k) \\ \text{Taylor :} \quad & \Phi(x_k, y_k, h) &&= c_1 + c_2 h + c_3 h^2 + \cdots + c_p h^{p-1} \end{aligned}$$

Für die Methode von Euler ist $\Phi(x_k, y_k, h)$ *unabhängig* von h!

Bei der Methode von Taylor sind die Koeffizienten c_i sowohl von der zu lösenden Differentialgleichung $y' = f(x, y)$, als auch vom Punkt (x_k, y_k) abhängig.

Konsistenzbedingung für ein numerisches Verfahren

$$\lim_{h \to 0} \left\{ \frac{y_{k+1} - y_k}{h} \right\} = y'(x_k) = \Phi(x_k, y_k, 0)$$

Definition 2.1 *Ein Einschrittverfahren* $y_{k+1} = y_k + h \cdot \Phi(x_k, y_k, h)$ *heisst mit der Differentialgleichung* $y' = f(x, y)$ *konsistent, falls* $\Phi(x, y, 0) = f(x, y)$ *gültig ist.*

Diese Bedingung ist für die Methode von Euler offensichtlich erfüllt.

Für die Methode der Taylorreihe haben wir: $c_1 = y'(x_k) = f(x_k, y_k)$.

- Für die folgenden Betrachtungen wird vereinfachend eine konstante Schrittweite h vorausgesetzt, sodass die Stützstellen $x_k = x_0 + k \cdot h$ äquidistant sind.

- Zu den Bezeichnungen:
  $y(x)$ = exakte Lösung der Dgl $y'(x) = f(x, y(x))$ mit der AB $y(x_0) = y_0$.
  $y(x_k)$ = Wert der exakten Lösung an der Stelle x_k.
  y_k = Wert der Approximation an der Stelle x_k.
- Es wird exakte Arithmetik vorausgesetzt, Rundungsfehler bleiben unberücksichtigt.

Definition 2.2 *Lokaler Diskretisationsfehler d_{k+1} an der Stelle x_{k+1}:*

$$d_{k+1} := y(x_{k+1}) - y(x_k) - h \cdot \Phi(x_k, y(x_k), h) \tag{5}$$

d_{k+1} beschreibt die Abweichung, mit welcher die exakte Lösung $y(x)$ die verwendete Integrationsvorschrift *in einem einzelnen Schritt nicht* erfüllt, falls an der Stelle x_k vom exakten Wert $y(x_k)$ ausgegangen wird. d_{k+1} gibt an, wie das numerische Verfahren bei der Ausführung eines Schrittes, also *lokal*, die Lösung verfälscht.

Geometrische Interpretation von d_{k+1}, falls nicht bei $y(x_k)$ gestartet wird:
(Methode von *Euler* oder Methode der *Taylorreihe*)

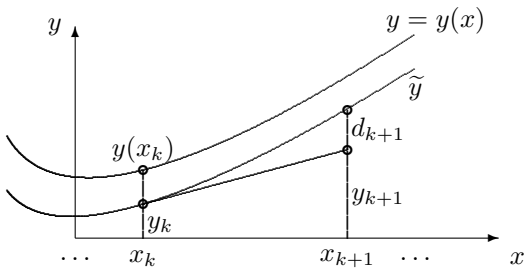


Abbildung 7: Lokaler Diskretisationsfehler, Methode von Euler, Methode der Taylorreihe,
$y = y(x)$ gesuchte Lösung, $\widetilde{y}$ = "exakte" Lösung mit $\widetilde{y}(x_k) = y_k$, d.h. "Lösung" mit neuer AB,
y_{k+1} = Wert mit dem numerischen Verfahren, $d_{k+1} := \widetilde{y}(x_{k+1}) - y_{k+1}$.

d_{k+1} := Differenz zwischen dem "exakten" Wert $\widetilde{y}(x_{k+1})$ und dem erhaltenen Näherungswert y_{k+1}, falls an der Stelle x_k vom "exakten" Wert $\widetilde{y}(x_k)$ ausgegangen wird.

Bemerkung 2 *Für allgemeinere Methoden trifft diese Interpretation meistens nicht zu.*

Für die Rechenpraxis aber von Bedeutung ist der *totale Fehler*, den die Näherung nach mehreren Integrationsschritten gegenüber der exakten Lösung aufweist.

Definition 2.3 *Globaler Diskretisationsfehler g_k an der Stelle x_k:*

$$g_k := y(x_k) - y_k.$$

Bemerkung 3 *Für den ersten Schritt von x_0 nach $x_1 = x_0 + h$ sind der lokale und der globale Fehler identisch.*

2.3.1 Abschätzung des globalen Diskretisationsfehlers:

Um den globalen Fehler abschätzen zu können, muss $\Phi(x, y, h)$ in einem geeignet gewählten Bereich $\mathbb{B}$ bzgl. der Variablen y eine *Lipschitz-Bedingung*

$$|\Phi(x, y_1, h) - \Phi(x, y_2, h)| \leq L \cdot |y_1 - y_2|$$

für $x, y_1, y_2, h \in \mathbb{B}$ und $0 < L < \infty$ erfüllen.

Bemerkung 4 *Für die Methode von Euler ist dies gerade die übliche Lipschitz-Bedingung an die Funktion $f(x,y)$, die für die Existenz und Eindeutigkeit der Lösung von $y' = f(x,y)$ ohnehin verlangt werden muss.*

Aus der Definition des lokalen Diskretisationsfehlers bekommen wir

$$y(x_{k+1}) = y(x_k) + h \cdot \Phi(x_k, y(x_k), h) + d_{k+1}$$

und durch Subtraktion von

$$y_{k+1} = y_k + h \cdot \Phi(x_k, y_k, h)$$

erhalten wir

$$g_{k+1} = g_k + h \cdot \{\Phi(x_k, y(x_k), h) - \Phi(x_k, y_k, h)\} + d_{k+1} \ .$$

Mit der Lipschitz-Bedingung für $\Phi(x,y,h)$:

$$\begin{aligned} |g_{k+1}| &\leq |g_k| + h\,|\Phi(x_k, y(x_k), h) - \Phi(x_k, y_k, h)| + |d_{k+1}| \\ &\leq |g_k| + hL\,|y(x_k) - y_k| + |d_{k+1}| \\ &= (1+hL)\,|g_k| + |d_{k+1}| \end{aligned}$$

Wir machen an dieser Stelle folgende vereinfachende *Annahme*. Für den Betrag des lokalen Diskretisationsfehlers kann eine obere Schranke D angeben werden, d.h. es soll gelten:

$$\max_k |d_k| \leq D$$

Damit erhalten wir für den globalen Fehler schliesslich die Rekursion:

$$|g_{k+1}| \leq (1+hL)\,|g_k| + D \qquad k = 0, 1, 2, \ldots \tag{6}$$

Behauptung

Es gilt

$$\begin{aligned} |g_n| &\leq \frac{(1+hL)^n - 1}{hL} \cdot D + (1+hL)^n \cdot |g_0| \\ &\leq \frac{D}{hL}\left\{e^{nhL} - 1\right\} + e^{nhL} \cdot |g_0| \end{aligned}$$

Beweis:

- Für die erste Ungleichung wiederholte Anwendung der Ungleichung (6).
- Für die zweite Ungleichung brauchen wir $1 + x \leq e^x$, d.h. dass die Funktion $y = e^x$ konvex ist.

- $g_0 = y(x_0) - y_0 = 0$

Satz

Für den globalen Diskretisationsfehler g_n an der Stützstelle $x_n = x_0 + nh$ gilt die Abschätzung

$$|g_n| \leq \frac{D}{hL}\left\{e^{nhL} - 1\right\} \leq \frac{D}{hL} \cdot e^{nhL}$$

Bemerkung 5

1) Für die Schranken von g_n ist neben der Lipschitzkonstanten L der Funktion $\Phi(x,y,h)$ der maximale Betrag D des lokalen Diskretisationsfehlers entscheidend.

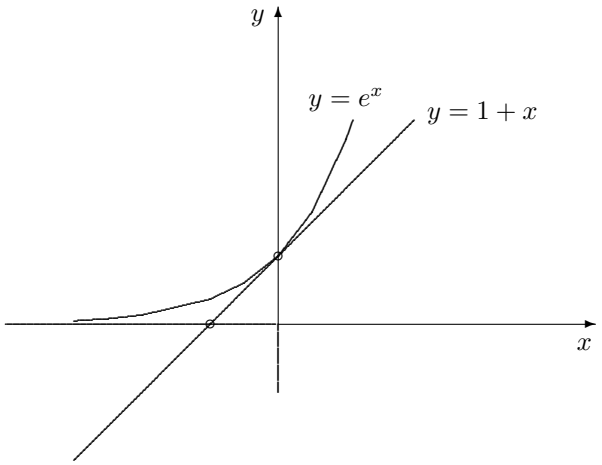


Abbildung 8: Konvexität von $y = e^x$

2) *Zur qualitativen Beurteilung eines Einschrittverfahrens spielt also der lokale Diskretisationsfehler d_{k+1} die zentrale Rolle.*

3) Zur Bestimmung von d_{k+1} werden Taylorentwicklungen zu Hilfe genommen. Dabei wird vorrausgesetzt, dass $f(x, y)$ und $y(x)$ hinreichend oft stetig differenzierbar sind.

Die Methode von Euler

Wir betrachten zunächst die einfachst mögliche Methode. Mit (5) erhalten wir:

$$d_{k+1} = y(x_{k+1}) - y(x_k) - h \cdot f(x_k, y(x_k))$$

$y(x_{k+1})$ wird durch die Taylorentwicklung von $y(x)$ im Punkt x_k mit Restglied ersetzt.

Dabei wird verwendet, dass $y'(x_k) = f(x_k, y(x_k))$ erfüllt ist:

$$\begin{aligned} d_{k+1} &= y(x_k) + h \cdot y'(x_k) + \frac{1}{2} \cdot y''(x_k + \Theta h) \cdot h^2 - y(x_k) - h \cdot y'(x_k) \\ &= \frac{1}{2} \cdot h^2 \cdot y''(x_k + \Theta h)\,, \qquad 0 < \Theta < 1 \end{aligned}$$

Sei

$$D := \max_{0 \le k \le n-1} |d_{k+1}| \le \frac{1}{2} \cdot h^2 \max_{x_0 \le \xi \le x_n} |y''(\xi)| =: \frac{1}{2} \cdot h^2 \cdot M_2$$

Damit bekommen wir

$$|g_n| \le \frac{hM_2}{2L} \cdot e^{L(x_n - x_0)}$$

für den globalen Diskretisationsfehler.

- Wird die Stützstelle x_n festgehalten und wird die Schrittweite $h = (x_n - x_0)/n$ für $n \longrightarrow \infty$ kleiner gemacht, so zeigt die Abschätzung, dass die Fehlerschranke proportional zu h abnimmt.
- D.h. der Wert $y_n \cong y(x_n)$ konvergiert mit $h \to 0$ gegen den exakten Wert $y(x_n)$. (Bei Abwesenheit von Rundungsfehlern)
- Die Konvergenz ist linear bzgl. der Schrittweite h.
  Die Methode von Euler hat die Fehlerordnung Eins.

Definition 2.4 *Ein Einschrittverfahren $y_{k+1} = y_k + h \cdot \Phi(x_k, y_k, h)$ hat die Fehlerordnung p, falls für den lokalen Diskretisationsfehler d_k die Abschätzung*

$$\max_{1 \le k \le n} |d_k| \le D = konst \cdot h^{p+1} = O(h^{p+1})$$

gilt, sodass der globale Diskretisationsfehler $|g_n|$ beschränkt ist durch

$$|g_n| \le e^{nhL} \cdot h^p = O(h^p)$$

Landau'sche Ordnungssysmbole

Zur Beschreibung der Grössenordnung einer mathematischen Grösse dienen die beiden folgenden Symbole „gross - O " bzw. „klein - o"

Es werden Grenzwerte für $x \to a$ betrachtet:
„gross - O" :

$$f(x) = O(g(x)) \quad \Longleftrightarrow \quad \lim_{x \to a} \frac{|f(x)|}{|g(x)|} = C = \quad \text{konstant}$$

„klein - o" :

$$f(x) = o(g(x)) \quad \Longleftrightarrow \quad \lim_{x \to a} \frac{|f(x)|}{|g(x)|} = 0$$

Die Methode der Taylorreihe

Mit obigem Ansatz

$$y(x) = y_k + c_1 \cdot (x - x_k) + c_2 \cdot (x - x_k)^2 + \cdots + c_p \cdot (x - x_k)^p$$

für die gesuchte Lösung bekommen wir für die Approximation

$$y_{k+1} = y_k + \frac{y'_k}{1!} \cdot h + \frac{y''_k}{2!} \cdot h^2 + \cdots + \frac{y_k^{(p)}}{p!} \cdot h^p \, .$$

Diese Methode hat die Fehlerordnung p, da

$$d_{k+1} = \frac{h^{p+1}}{(p+1)!} \cdot y^{(p+1)}(x_k + \Theta h) \qquad 0 < \Theta < 1$$

vgl. Tabelle 2.2. Die im Beispiel angewandte Integrationsmethode hat die Fehlerordnung 4.

D.h. z.B: wird h um einen Faktor 10 verkleinert, also $h \longrightarrow \frac{h}{10}$ so hat das eine Reduktion um den Faktor 10^{-4} des Fehlers zur Folge.

Also je grösser die Fehlerordnung, desto genauer das entsprechende Verfahren.

2.4 Verbesserte Polygonzugmethode Trapezmethode, Verfahren von Heun

Falls von einer Methode die Fehlerordnung bekannt ist, kann man eine *Extrapolation* durchführen. Eine Extrapolation ist eine raffinierte Linearkombination verschieden gerechneter Approximationen derselben Grösse zur Erhöhung der Fehlerordnung dieser Grösse.

Die Methode von Euler hat die Fehlerordnung eins, d.h. $O(h)$. Eine Extrapolation wird nun wie folgt durchgeführt:

Voraussetzung:

Mit der Polygonzugmethode $y_{k+1} = y_k + h \cdot f(x_k, y_k)$ seien bis zu einer gegebenen Stelle x zwei verschiedene Integrationen durchgeführt worden; zuerst mit der Schrittweite $h_1 = h$ und anschliessend mit der Schrittweite $h_2 = \frac{h}{2}$.

Für die erhaltenen Werte y_n und y_{2n} nach n bzw. $2n$ Integrationsschritten gilt näherungsweise:

$$\begin{aligned} y_n &\approx y(x) + c_1 \cdot h + O(h^2) \\ y_{2n} &\approx y(x) + c_1 \cdot \frac{h}{2} + O(h^2) \end{aligned}$$

Richardson Extrapolation

Durch Linearkombination von y_n und y_{2n} wird ein neuer Wert (=*extrapolierter* Wert) gebildet:

$$\widetilde{y} := 2\, y_{2n} - y_n \approx y(x) + O(h^2). \tag{7}$$

Der Fehler von $\widetilde{y}$ in (7) ist gegenüber $y(x)$ von *zweiter* Ordnung in h, d.h. er ist qualitativ *besser*!

- Anstatt eine Dgl. nach der Methode von Euler mit zwei verschiedenen Schrittweiten parallel zu integrieren, ist es zweckmässiger, die Extrapolation direkt auf die Werte anzuwenden, die einerseits von einem Intergationsschritt mit der Schrittweite h und andererseits von einem Doppelschritt mit der halben Schrittweite geliefert werden.
- In beiden Fällen geht man vom Näherungspunkt $(x_k,\, y_k)$ aus.

Die Methode von Euler, Schrittweite h:

$$y^{(1)}_{k+1} = y_k + h \cdot f(x_k, y_k)$$

Die Methode von Euler, Schrittweite $\frac{h}{2}$:

$$\begin{aligned} y^{(2)}_{k+\frac{1}{2}} &= y_k + \frac{h}{2} \cdot f(x_k, y_k) \\ y^{(2)}_{k+1} &= y^{(2)}_{k+\frac{1}{2}} + \frac{h}{2} \cdot f(x_k + \frac{h}{2},\, y^{(2)}_{k+\frac{1}{2}}) \end{aligned}$$

Richardson-Extrapolation:

$$\widetilde{y} = y_{k+1} = 2 \cdot y^{(2)}_{k+1} - y^{(1)}_{k+1} = y_k + h \cdot f\left(x_k + \frac{h}{2},\, y_k + \frac{h}{2} \cdot f(x_k, y_k)\right)$$

Algorithmische Formulierung

$$\begin{aligned} k_1 &= f(x_k, y_k) \\ k_2 &= f(x_k + \frac{h}{2}, y_k + \frac{1}{2}\, h\, k_1) \\ y_{k+1} &= y_k + h\, k_2 \end{aligned}$$

Diese Methode ist die sogenannte *verbesserte Polygonzug-Methode von Euler*.
Ein einzelner Schritt erfordert die Auswertung der Funktion $f(x, y)$ an zwei Stellen:

- k_1 = Steigung des Vektorfelds im Punkt $P(x_k,\, y_k)$.
- Damit wird der Hilfspunkt $P_{\text{hilf}}(x_k + \frac{h}{2},\, y^{(2)}_{k+\frac{1}{2}})$ bestimmt.
- k_2 = Steigung des Vektorfelds im Hilfspunkt.
- y_{k+1} wird anschliessend mit Hilfe von k_2 berechnet, womit die Änderung des Vektorfelds berücksichtigt wird.

Geometrische Interpretation des Verfahrens

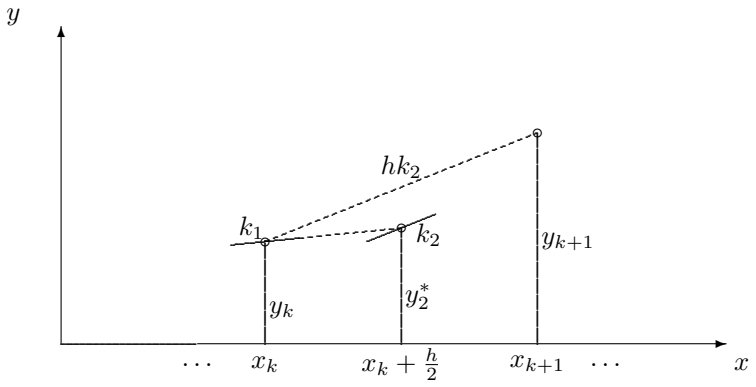


Abbildung 9: Verbesserter Polygonzug, $y_2^* = y_k + \frac{h}{2} \cdot k_1$ = Prädiktor und y_{k+1} = Korrektor

Bestimmung der Fehlerordnung der verbesserten Polygonzugmethode

Bestimmung der höheren Ableitungen von y:

$$\begin{aligned} y'(x) &= f(x,y) \\ y''(x) &= f_x + f \cdot f_y =: F(x, y(x)) \\ y'''(x) &= (f_{xx} + 2\, f \cdot f_{xy} + f^2 \cdot f_{yy}) + (f_x + f \cdot f_y)\, f_y =: G + F \cdot f_y \end{aligned}$$

Dabei wurden $G(x, y(x)) := (f_{xx} + 2\, f \cdot f_{xy} + f^2 \cdot f_{yy})$ und $F(x, y(x)) := (f_x + f \cdot f_y)\, f_y$ gesetzt.

Somit erhalten wir mit (5) für den lokalen Diskretisationsfehler:

$$d_{k+1} = y(x_{k+1}) - y(x_k) - h \left\{ f\left(x_k + \frac{h}{2},\, y(x_k) + \frac{h}{2}\, f\left(x_k, y(x_k)\right)\right)\right\}$$

Entwicklung in Taylorreihen, Entwicklungszentrum x_k und mit $x_{k+1} = x_k + h$:

$$\begin{aligned} d_{k+1} &= \frac{1}{6}\, h^3\, y'''(x_k) - \frac{1}{8}\, h^3\, G(x_k, y(x_k)) + O(h^4) \\ &= \frac{1}{6} \left\{ \frac{1}{4}\, G + F\, f_y \right\} \cdot h^3 + O(h^4) \end{aligned}$$

Der dominante Term in d_{k+1} ist i. allg. proportional zu h^3 und somit ist die Fehlerordnung der verbesserten Polygonzugmethode zwei, d.h. $O(h^2)$.

Trapezmethode

$$y'(x) = f(x, y(x))$$

Integration über das Integral $[x_k,\, x_{k+1}]$:

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x, y(x))\, dx$$

- Hier haben wir eine zur gegebenen Differentialgleichung äquivalente Integralgleichung.
- Für das Integral auf der rechten Seite kann i. allg. keine Stammfunktion angegeben werden, da $y(x)$ die gesuchte Funktion ist.
- Numerische Approximation des Integrals, z.B. durch die einfache Trapezregel.
- Die obige Gleichung wird daher nur noch genähert gültig sein, deshalb wird $y(x_k)$ durch den Näherungswert y_k ersetzt.

Damit erhalten wir

$$y_{k+1} = y_k + \frac{h}{2}\,\{f(x_k, y_k) + f(x_{k+1}, y_{k+1})\} \qquad k = 0,\ 1,\ 2, \ldots \tag{8}$$

(8) ist eine *implizite* Gleichung für die Unbekannte y_{k+1}.

D.h. mit der Trapezmethode bekommen wir eine *implizite Integrationsmethode*: jeder Integrationsschritt verlangt die Lösung einer *nicht* - linearen Gleichung (8).

Für den *lokalen* Diskretisationsfehler erhalten wir mit (5) und mit Hilfe von Taylor-Entwicklungen:

$$\begin{aligned} d_{k+1} &= y(x_{k+1}) - y(x_k) - \frac{h}{2}\,\{f(x_k, y(x_k)) + f(x_{k+1}, y(x_{k+1}))\} \\ &= -\frac{1}{12}\,h^3\,y'''(x_k) + O(h^4) \end{aligned}$$

Auf Grund des dominanten Terms in d_{k+1} ist die Fehlerordnung der Trapezmethode zwei, d.h. $O(h^2)$.

Sie ist also gleich derjenigen der verbesserten Polygonzugmethode. Die Trapezmethode hat jedoch spezielle *Stabilitätseigenschaften*, vgl. *steife* Differentialgleichungen.

Lösung von (8):

Für nicht-lineare Differentialgleichungen hat die zu lösende implizite Gleichung für y_{k+1} *Fixpunktform*. Es geht nun um die Lösung dieser Gleichung.

Der Wert $f(x_k, y_k)$ muss ohnehin berechnet werden. Damit bekommen wir einen geeigneten Startwert $y_{k+1}^{(0)}$ für die Fixpunktiteration:

$$y_{k+1}^{(0)} = y_k + h\,f(x_k, y_k)$$

Die Wertefolge

$$y_{k+1}^{(n+1)} = y_k + \frac{h}{2}\,\left\{f(x_k, y_k) + f(x_{k+1}, y_{k+1}^{(n)})\right\} \qquad n = 0,\ 1,\ 2,\ \ldots$$

konvergiert gegen den Fixpunkt y_{k+1}, falls

a) die Funktion $f(x, y)$ die übliche Lipschitz-Bedingung mit der Lipschitz-Konstanten $L < 1$ erfüllt und falls

b) $\frac{1}{2}\,hL < 1$ ist, weil dann die Voraussetzungen des lokalen Fixpunktsatzes von Banach erfüllt sind.

Da y_{k+1} ohnehin eine Näherung für $y(x_{k+1})$ darstellt, wird in obiger Fixpunktiteration *nur ein Schritt* ausgeführt, die Iteration wird somit vorzeitig abgebrochen.

Die Methode von Heun

$$\begin{aligned} y_{k+1}^{(P)} &= y_k + h\, f(x_k, y_k) \\ y_{k+1} &= y_k + \frac{h}{2}\left\{ f(x_k, y_k) + f(x_{k+1}, y_{k+1}^{(P)}) \right\} \end{aligned}$$

- *Prädiktorwert* $y_{k+1}^{(P)}$:
  mit der expliziten Methode von Euler der Ordnung $O(h)$.
- *Korrektor* y_{k+1}:
  $y_{k+1}^{(P)}$ wird anschliessend mit der impliziten Trapezmethode der Ordnung $O(h^2)$ zum Wert y_{k+1} korrigiert
- Für die Fehlerordnung dieser neuen Methode geht man analog wie bei der verbesserten Polygonzug-Methode vor.

Algorithmische Formulierung

$$\begin{aligned} k_1 &= f(x_k, y_k) \\ k_2 &= f(x_k + h, y_k + hk_1) \\ y_{k+1} &= y_k + \tfrac{h}{2}\,\{k_1 + k_2\} \end{aligned}$$

- Zur Bestimmung von y_{k+1} werden die beiden Steigungen k_1 in $(x_k,\, y_k)$ und k_2 in $(x_{k+1},\, y_{k+1}^{(P)})$ gemittelt.
- Sowohl die verbesserte Polygonzugmethode als auch die Methode von Heun sind Repräsentanten von *expliziten 2-stufigen Runge-Kutta-Verfahren* mit der Fehlerordnung 2, also $O(h^2)$

Geometrische Interpretation des Verfahrens

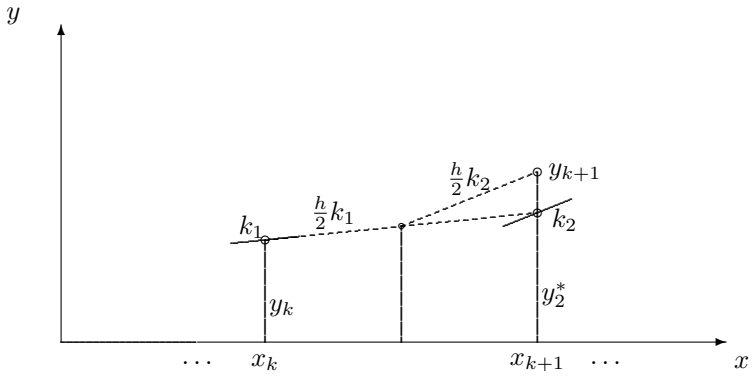


Abbildung 10: Methode von Heun, $y_2^* = y_k + h \cdot k_1 =$ Prädiktor und $y_{k+1} =$ Korrektor

Wir betrachten erneut unser Testbeispiel:

$$\left\{ \begin{array}{rcl} y'(x) &=& -2xy^2 \\ y(0) &=& 1 \end{array} \right.$$

	Verbesserte Polygonzugmethode				Methode von Heun			
	$h = 0.1$		$h = 0.05$		$h = 0.1$		$h = 0.05$	
x_k	y_k	g_k	y_k	g_k	y_k	g_k	y_k	g_k
0	1.00000		1.00000		1.00000		1.00000	
0.1	0.99000	0.00010	0.99007	0.00002	0.99000	0.00010	0.99009	0.00001
0.2	0.96118	0.00036	0.96145	0.00009	0.96137	0.00017	0.96152	0.00002
0.3	0.91674	0.00069	0.91727	0.00016	0.91725	0.00019	0.91742	0.00001
0.4	0.86110	0.00096	0.86184	0.00023	0.86195	0.00011	0.86208	−0.00001
0.5	0.79889	0.00111	0.79974	0.00026	0.80003	−0.00003	0.80004	−0.00004
0.6	0.73418	0.00111	0.73503	0.00026	0.73553	−0.00023	0.73538	−0.00009
0.7	0.67014	0.00100	0.67091	0.00023	0.67159	−0.00045	0.67128	−0.00014
0.8	0.60895	0.00080	0.60957	0.00018	0.61040	−0.00064	0.60993	−0.00018
0.9	0.55191	0.00058	0.55236	0.00013	0.55329	−0.00080	0.55270	−0.00021
1.0	0.49964	0.00036	0.49992	0.00008	0.50092	−0.00092	0.50024	−0.00024

Tabelle 2.3 *Verbesserte Polygonzugmethode und die Methode von Heun.*

2.5 Explizite Runge-Kutta Verfahren, ERK

Diese Methoden werden allgemein als sogenannte $RK-$Methoden bezeichnet. Unter dieser Abkürzung figurieren sie auch in den verschiedensten Simulations-Werkzeugen, wie z.B. SIMULINK oder in MATLAB.

- Das Prinzip der Herleitung von Einschrittmethoden höherer Fehlerordnung wir dargelegt.
- Ein dreistufiges Runge-Kutta Verfahren wird ausführlich behandelt.
- Die entsprechenden Entwicklungen werden rasch kompliziert und umfangreich.

Als Ausgangspunkt dient uns die Gleichung, (s.o.)

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x, y(x))\, dx$$

Allgemein geht es nun darum, das Integral auf der rechten Seite dieser Gleichung durch eine Quadraturformel zu approximieren.

Vorgehen für ein 3-stufiges Verfahren

- Der Wert des Integrals wird durch eine Quadraturformel approximiert, die auf drei Stützstellen ξ_1, ξ_2 und ξ_3 im Intervall $[x_x,\ x_{k+1}]$ mit den zugehörigen Integrationsgewichte c_1, c_2 sowie c_3 beruht.
- Die Lage der ξ_i sowie die Werte der c_i sind zunächst beliebig.
- Sie werden anschliessend so bestimmt, dass das resultierende Verfahren eine *möglichst hohe Fehlerordnung* hat.
- Mit dieser Forderung gelangen wir zu folgendem *Ansatz* für die Approximation y_{k+1}

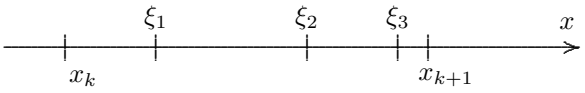
$$y_{k+1} = y_k + h\ \{c_1\ f(\xi_1, y(\xi_1)) + c_2\ f(\xi_2, y(\xi_2)) + c_3\ f(\xi_3, y(\xi_3))\} \tag{9}$$

Dieser Ansatz ist völlig analog zur Trapezmethode zu interpretieren. Festzulegen sind nun die ξ_i und die unbekannten Werte $y(\xi_i)$, sowie die Gewichte c_i, $i = 1, 2, 3$.

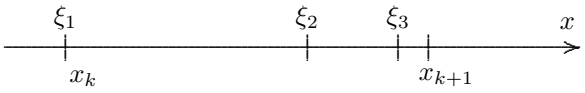
Die $y(\xi_i)$ werden wie folgt festgelegt:

- Das Prinzip der *Prädiktor - Korrektor - Methode* soll angewendet werden.
- Die Methode soll *explizit* werden.

allgemein:



speziell:



Diese Zielsetzungen bedingen für die ξ_i:

$$\xi_1 = x_k \qquad \xi_2 = x_k + a_2 \cdot h \qquad \xi_3 = x_k + a_3 \cdot h \qquad 0 \le a_2,\, a_3 \le 1$$

und somit

$$\xi_1 = x_k \quad \Longrightarrow \quad y(\xi_1) = y_k$$

Der erste Prädiktorwert $y_1^* := y_k$ ist bekannt.

Wir brauchen Prädiktorwerte für die beiden restlichen Werte. Dazu verwenden wir folgende Ansätze:

$$\begin{aligned} y(\xi_2): \quad y_2^* &:= y_k + b_{21} \cdot hf(x_k, y_k) \\ y(\xi_3): \quad y_3^* &:= y_k + b_{31} \cdot hf(x_k, y_k) + b_{32} \cdot hf(x_k + a_2 \cdot h, y_2^*) \end{aligned}$$

Geometrische Interpretation

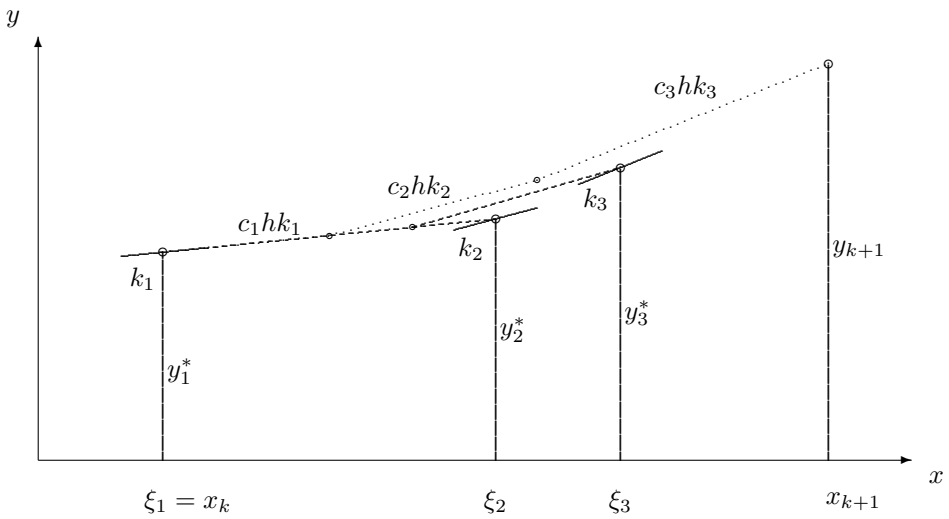


Abbildung 11: 3–stufiges Runge–Kutta Verfahren,
Prädiktorwerte $y_1^* = y_k$, $y_2^* = y_k + b_{21}hk_1$ und $y_3^* = y_k + b_{31}hk_1 + b_{32}hk_2$.

- y_2^* hängt von der Steigung im Punkt $P(x_k, y_k)$ ab.

- y_3^* hängt von den Steigungen in den Punkten $P(x_k,\, y_k)$ und $Q(\xi_2,\, y_2^*)$ ab.
- Die Parameter b_{21}, b_{31} und b_{32} können gewählt werden.

Werden alle diese Ansätze oben in Gleichung (9) eingesetzt, bekommen wir eine explizite Einschrittmethode

$$y_{k+1} = y_k + h\,\Phi(x_k, y_k, h) \qquad k = 0,\ ,1\,,2,\ \ldots$$

Algorithmische Formulierung:

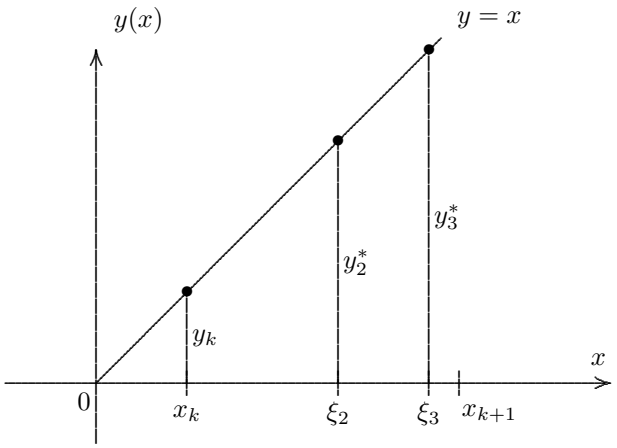
$$\begin{aligned} k_1 &= f(x_k, y_k) \\ k_2 &= f(x_k + a_2 \cdot h, y_k + b_{21} \cdot hk_1) \\ k_3 &= f(x_k + a_3 \cdot h, y_k + b_{31} \cdot hk_1 + b_{32} \cdot hk_2) \\ y_{k+1} &= y_k + h \cdot \{c_1 k_1 + c_2 k_2 + c_3 k_3\} \end{aligned}$$

Folgende Dinge sind zu beachten:

- Da die Funktion $f(x, y)$ pro Iterationsschritt dreimal ausgewertet wird, spricht man von einem *drei-stufigen RK-Verfahren.*
- y_k, $y_k + b_{21} \cdot hk_1$ und $y_k + b_{31} \cdot hk_1 + b_{32} \cdot hk_2$ heissen *Prädiktorwerte.*
- y_{k+1} heisst der *Korrektor.*
- Die acht Parameter a_2, a_3, b_{21}, b_{31}, b_{32}, c_1, c_2 und c_3 sind so zu bestimmen, dass die Fehlerordnung möglichst gross wird, d.h. Fehlerordnung $= O(h^p)$, p maximal.
- Weitere *Einschränkung* der Parameter:
  dazu betrachten wir das folgende Testbeispiel

  (10) $\qquad y'(x) = 1 \qquad$ d.h. $f(x, y) = 1 \qquad$ mit der AB: $y(0) = 0.$

  y_2^* und y_3^* sollen so sein, dass (10) mit unserem numerischen Verfahren *exakt* gelöst wird.



$$\begin{aligned} y_2^* &= y_k + b_{21} \cdot hf(x_k, y_k) \\ &= y_k + b_{21} \cdot h \\ &= x_k + a_2 \cdot h \qquad \Longrightarrow \qquad a_2 = b_{21} \end{aligned}$$

$$\begin{aligned} y_3^* &= y_k + b_{31} \cdot hf(x_k, y_k) + b_{32} \cdot hf(x_k + a_2 \cdot h, y_2^*) \\ &= y_k + b_{31} \cdot h + b_{32} \cdot h \\ &= x_k + a_3 \cdot h \qquad \Longrightarrow \quad a_3 = b_{31} + b_{32} \end{aligned}$$

und somit haben wir zusätzlich zwei Bedingungen:

$$a_2 = b_{21} \tag{11}$$

$$a_3 = b_{31} + b_{32} \tag{12}$$

(11) und (12) wurden in der Abbildung 11 bereits verwendet. (11) damit man bis ξ_2 und (12) damit man bis ξ_3 kommt.

Der lokale Diskretisationsfehler (5) ist hier wie folgt:

$$d_{k+1} = y(x_{k+1}) - y(x_k) - h\ \left\{c_1\overline{k}_1 + c_2\overline{k}_2 + c_3\overline{k}_3\right\} \tag{13}$$

- Wird in k_i die Näherung y_k durch den exakten Wert $y(x_k)$ ersetzt, so bekommen wir eine neue Grösse $\overline{k}_i$, für $i = 1,\ 2,\ 3$.
- $\overline{k}_1$, $\overline{k}_2$, und $\overline{k}_3$ werden nun sukzessive in Taylorreihen an der Stelle x_k entwickelt und in (44) eingesetzt.
- Dabei verwenden wir

$$\begin{aligned} y''(x) &= f_x + f \cdot f_y \\ &=: F(x, y(x)) \\ y'''(x) &= (f_{xx} + 2\, f \cdot f_{xy} + f^2 \cdot f_{yy}) + (f_x + f \cdot f_y)\, f_y \\ &=: G + F \cdot f_y \end{aligned}$$

Resultat

$$\begin{aligned} d_{k+1} &= h \cdot f\,\{1 - c_1 - c_2 - c_3\} + h^2 \cdot F\left\{\frac{1}{2} - a_2c_2 - a_3c_3\right\} \\ &\quad + h^3 \cdot \left\{F\, f_y \left[\frac{1}{6} - a_2c_3b_{32}\right] + G\left[\frac{1}{6} - \frac{1}{2}\, a_2^2c_2 - \frac{1}{2}\, a_3^2c_3\right]\right\} + O(h^4) \end{aligned} \tag{14}$$

Damit das Verfahren mindestens die Fehlerordnung 3 hat, müssen folgende Gleichungen gelten:

$$\begin{aligned} c_1 + c_2 + c_3 &= 1 \\ a_2c_2 + a_3c_3 &= \frac{1}{2} \\ a_2c_3b_{32} &= \frac{1}{6} \\ a_2^2c_2 + a_3^2c_3 &= \frac{1}{3} \end{aligned} \tag{15}$$

Auch (15) wurde in Abbildung 11 bereits verwendet, denn sonst käme man gar nicht bis an das Intervallende x_{k+1}.

Wir haben also 4 Gleichungen für 6 Unbekannte, d.h. wir haben zwei freie Parameter.

Es stellt sich aus diesem Grund folgende *Frage*:

Ist ein Verfahren der Fehlerordnung 4 möglich? Ist zufällig mehr herauszuholen, da das Gleichungssystem unterbestimmt ist (mehr Unbekannte als Gleichungen).

In der Taylorentwicklung (14) für d_{k+1} kommt im Koeffizienten von h^4 ein Term vor, der von den obigen 6 Parametern $unabhängig$ ist!

Die $Antwort$ lautet deshalb leider nein.

D.h. allgemein gilt der

Satz 2.1 *Die maximal erreichbare Fehlerordnung p für explizite 3-stufige RK-Verfahren ist drei.*

Das obige Gleichungssystem hat eine zweiparametrige Lösungsmenge. Z.B. können a_2 und a_3 als Parameter frei gewählt werden. Dies ist übrigens die übliche Wahl.

Damit erhalten wir aus der 2-ten und 4-ten Gleichung:

$$c_2 = \frac{3a_3 - 2}{6a_2(a_3 - a_2)} \qquad c_3 = \frac{2 - 3a_2}{6a_3(a_3 - a_2)}$$

und aus der 1-ten und 3-ten Gleichung:

$$c_1 = \frac{6a_2a_3 + 2 - 3(a_2 + a_3)}{6\,a_2a_3} \qquad b_{32} = \frac{a_3(a_3 - a_2)}{a_2(2 - 3a_2)}$$

Die Festlegung der Parameter in dieser $2-$ parametrigen Schar von $3-$ stufigen RK-Verfahren mit der Fehlerordnung 3 erfolgt nach verschiedenen Kriterien, wie z.B.

- Ob es sich um eine spezielle Klasse von Differentialgleichungen handelt oder
- ob eine Schrittweitensteuerung im Vordergrund steht.

Es ist üblich, die verschiedenen Methoden in einem $Tableau$ zu symbolisieren:

$$\begin{array}{r|lll} (\xi_1 = x_k)\ a_1 = 0 & & & \\ a_2 & b_{21} & & \\ a_3 & b_{31} & b_{32} & \\ \hline & c_1 & c_2 & c_3 \end{array}$$

Ein erstes RK-Verfahren: wähle z.B. $a_2 = \frac{1}{3}$ und $a_3 = \frac{2}{3}$

$$\begin{array}{r|lll} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array}$$

$$\begin{aligned} k_1 &= f(x_k, y_k) \\ k_2 &= f\left(x_k + \frac{1}{3}\,h, y_k + \frac{1}{3}\,h\,k_1\right) \\ k_3 &= f\left(x_k + \frac{2}{3}\,h, y_k + \frac{2}{3}\,h\,k_2\right) \\ y_{k+1} &= y_k + \frac{1}{4}\,h \cdot \{k_1 + 3k_3\} \end{aligned}$$

Dies ist die Methode von Heun, $p = 3$. Es handelt sich um eine Methode dritter Ordnung.

Geometrische Interpretation

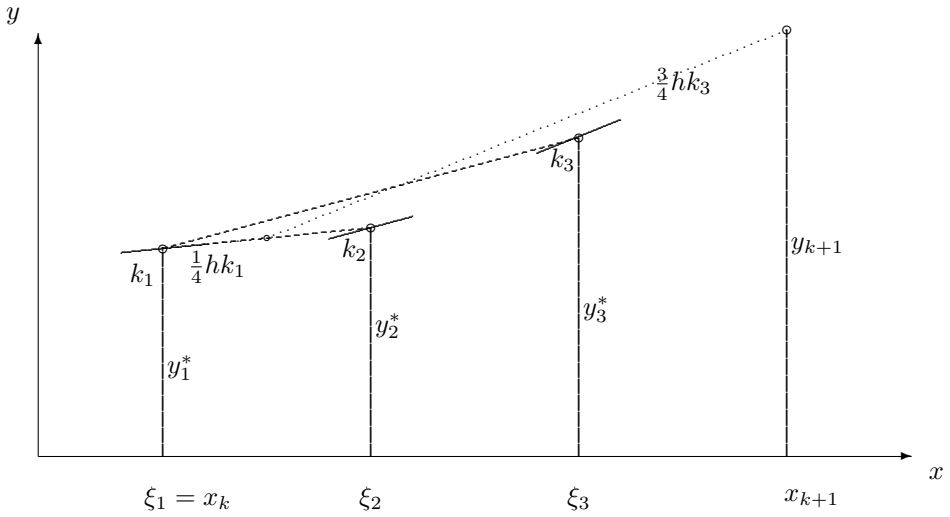


Abbildung 12: Methode von Heun $p = 3$,
Prädiktorwerte $y_1^* = y_k$, $y_2^* = y_k + \frac{1}{3} \cdot hk_1$ und $y_3^* = y_k + \frac{2}{3} \cdot hk_2$.

Ein anderes RK-Verfahren: wähle z.B. $a_2 = \frac{1}{2}$ und $a_3 = 1$

(16)

0			
$\frac{1}{2}$	$\frac{1}{2}$		
1	-1	2	
	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$

$$
\begin{aligned}
k_1 &= f(x_k, y_k) \\
k_2 &= f\left(x_k + \frac{1}{2}\, h, y_k + \frac{1}{2}\, h\, k_1\right) \\
k_3 &= f\left(x_k + h, y_k - h\, k_1 + 2h\, k_2\right) \\
y_{k+1} &= y_k + \frac{1}{6}\, h\, \left\{k_1 + 4\, k_2 + k_3\right\}
\end{aligned}
$$

Bei diesem Verfahren erscheint die *Regel von Simpson* als Quadraturformel. Es handelt sich um die Methode von Kutta dritter Ordnung, die er bereits 1901 angegeben hat.

Bemerkung 6

Auch die oben kennengelernten Methoden zweiter Ordnung lassen sich mit diesem Tableau definieren:

Verbesserte Polygonzugmethode

als ERK-Methode mit $p = 2$

(17)

0		
$\frac{1}{2}$	$\frac{1}{2}$	
	0	1

Heun

interpretiert als ERK-Methode mit $p = 2$

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

Cleve Moler, cf. Matlab

a_2 muss nicht kleiner sein als a_3, z.B. darf $a_2 = 1$ und $a_3 = \frac{1}{2}$ gewählt werden:

$$\begin{array}{c|ccc} 0 & & & \\ 1 & 1 & & \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \\ \hline & \frac{1}{6} & \frac{1}{6} & \frac{4}{6} \end{array}$$

d.h.

$$\begin{aligned} k_1 &= f(x_k,\, y_k) \\ k_2 &= f(x_k + h,\, y_k + hk_1) \\ k_3 &= f\left(x_k + \frac{h}{2},\, y_k + \frac{h}{4}\, k_1 + \frac{h}{4}\, k_2\right) \\ y_{k+1} &= y_k + \frac{h}{6}\, \{k_1 + k_2 + 4k_3\} \end{aligned}$$

Aufgabe 2.1

Skizzieren Sie analog zur Abbildung 12 das Diagramm für diese Methode.

2.6 Schrittweitensteuerung

- Das Prinzip der Schrittweitensteuerung wird gezeigt. Dazu soll der *lokale* Diskretisationsfehler der verwendeten Methode mit Hilfe eines Verfahrens höherer Fehlerordnung geschätzt werden.
- Um den Aufwand zur Berechnung des Schätzwertes möglichst klein zu halten, sollte das RK-Verfahren höherer Fehlerordnung soweit wie möglich die gleichen k_i - Werte verwenden, wie die betrachtete Methode.

Als Beispiel dazu können wir die verbesserte Polygonzugmethode (17) und die Methode von Kutta (16) mit $p = 3$ betrachten. Diese beiden Methoden erfüllen unseren Wunsch.

Für (17) $p = 2$ haben wir also

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

d.h.

$$\begin{aligned} k_1 &= f(x_k,\, y_k) \\ k_2 &= f(x_k + \frac{h}{2},\, y_k + \frac{h}{2}k_1) \\ y_{k+1} &= y_k + hk_2 \end{aligned}$$

und für (16) $p = 3$

$$\begin{array}{c|ccc} 0 & & & \\ 1/2 & 1/2 & & \\ 1 & -1 & 2 & \\ \hline & 1/6 & 2/3 & 1/6 \end{array}$$

mit

$$\begin{aligned} k_1 &= f(x_k,\, y_k) \\ k_2 &= f(x_k + \frac{h}{2},\, y_k + \frac{h}{2}k_1) \\ k_3 &= f\left(x_k + h,\, y_k - hk_1 + 2hk_2\right) \\ y_{k+1} &= y_k + \frac{h}{6}\left\{k_1 + 4k_2 + k_3\right\} \end{aligned}$$

Für die Methode (16) muss einzig k_3 zusätzlich gerechnet werden!

Der lokale Diskretisationsfehler der Methode (17) ist

$$d_{k+1}^{(VP)} = y(x_{k+1}) - y(x_k) - h\,\overline{k}_2 = O(h^3) \tag{18}$$

und der lokale Diskretisationsfehler der Methode (16) ist

$$d_{k+1}^{(RK3)} = y(x_{k+1}) - y(x_k) - \frac{h}{6}\left\{\overline{k}_1 + 4\,\overline{k}_2 + \overline{k}_3\right\} = O(h^4) \tag{19}$$

Mit Hilfe von (19) wird nun ein besserer Schätzwert für $d_{k+1}^{(VP)}$ bestimmt, indem die Differenz $y(x_{k+1}) - y(x_k)$ mit (19) ersetzt und in (18) eingesetzt wird, d.h.

$$d_{k+1}^{(VP)} = \frac{h}{6}\left\{\overline{k}_1 + 4\,\overline{k}_2 + \overline{k}_3\right\} - h\,\overline{k}_2 + d_{k+1}^{(RK3)}$$

und damit

$$d_{k+1}^{(VP)} = \frac{h}{6}\left\{\overline{k}_1 - 2\,\overline{k}_2 + \overline{k}_3\right\} + O(h^4).$$

Dabei wurde verwendet, dass $d_{k+1}^{(RK3)} = O(h^4)$.

Nun werden die unbekannten $\overline{k}_i$ durch die berechneten k_i ersetzt, woraus wir schliesslich:

$$d_{k+1} \approx \frac{h}{6}\ \{k_1 - 2\,k_2 + k_3\} + O(h^4)$$

erhalten.

D.h. mit *einer* zusätzlichen Funktionsauswertung für k_3 ist

$$\frac{h}{6}\ \{k_1 - 2\,k_2 + k_3\} = O(h^3) \tag{20}$$

ein *verbesserter* Schätzwert für $d_{k+1}^{(VP)}$.

Aufgrund des Betrages von (20) kann nun entschieden werden, ob für die Methode (17)!!

- die Schrittweite verkleinert werden muss oder,
- ob die Schrittweite für den folgenden Iterationsschritt vergrössert werden darf,

um eine vorgegebene Genauigkeitsanforderung zu erfüllen. Falls also ein Schritt akzeptiert wird, muss der Korrektor der Methode (17) verwendet werden. Die Differenz der beiden Approximationen ist (zumindest asymptotisch) eine Abschätzung des lokalen Fehlers und obiger Algorithmus garantiert, dass diese Abschätzung kleiner als eine gegebene Toleranz bleibt, cf. E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Kapitel II.4. p 166ff.

Bemerkung 2.1

Falls der Korrektor der Methode (16) verwendet wird, wird das Konzept der 'Fehlerabschätzung' verlassen und die Differenz der beiden Approximationen wird nur noch für die Wahl der Schrittweite gebraucht. Dies mag gerechtfertigt sein, da die lokalen Fehler i. allg. wenig mit dem globalen Fehler gemeinsam haben. (Der numerische Algorithmus weiss nichts über die Instabilitätseigenschaften des Dgl-systems). Der Algorithmus, der mit der genaueren Methode weiterrechnet heisst 'lokale Extrapolation'.

Bemerkung 2.2

Vergleichen Sie dazu, wie dieses Problem in MATLAB gelöst wird, insbes. z.B. bei der Routine `ode23`.

x_k	y_k	$10^8\ g_k$	h	k_1	k_2	k_3	$10^8\ d_{k+1}^{(H)}$
0	1.00000000	0	0.050	0	−0.1000000	−0.0498751	416 (!)
			0.025	0	−0.0500000	−0.0249844	26
0.025	0.99937500	39	0.025	−0.0499375	−0.0996257	−0.0747662	26
0.050	0.99750546	77	0.025	−0.0950179	−0.1485091	−0.1239909	24
0.075	0.99440533	114	0.025	−0.1483263	−0.1962962	−0.1722985	21
0.100	0.99009754	147	0.025	−0.1960586	−0.2426528	−0.2193460	16
0.125	0.98461365	173	0.025	−0.2423660	−0.2872707	−0.2648130	9
0.150	0.97799319	192	0.050	−0.2869412	−0.3714456	−0.3291543	130 (!)
			0.025	−0.2869412	−0.3298718	−0.3084071	1
0.175	0.97028303	199	0.050	−0.3295072	−0.4093871	−0.3694444	9
0.225	0.95181067	372	0.050	−0.4076746	−0.4771559	−0.4425056	−301 (!)
			0.025	−0.4076746	−0.4433230	−0.4255273	−48
0.250	0.94117320	327	0.025	−0.4429035	−0.4757979	−0.4593917	−68
0.275	0.92968944	260	0.025	−0.4753774	−0.5054196	−0.4904532	−91
0.300	0.91742947	172	0.025	−0.5050061	−0.5321361	−0.5186407	−116 (!)

Tabelle 2.4 *Methode von Heun zweiter Ordnung mit Schrittweitensteuerung.* $tol = 10^{-6}$. *Falls der Absolutbetrag der Schätzung* $< 10^{-7}$ *wird die Schrittweite von* h *auf* $2h$ *gesetzt.*

Ansatz für ein 4-stufiges RK-Verfahren

Das Vorgehen ist analog zur 3-stufigen Methode:

$$y_{k+1} = y_k + h\ \{c_1 f(\xi_1, y(\xi_1)) + c_2 f(\xi_2, y(\xi_2)) + c_3 f(\xi_3, y(\xi_3)) + c_4 f(\xi_4, y(\xi_4))\}$$

und mit der Schreibweise eines Tableaus:

$$\begin{array}{c|cccc} 0 & & & & \\ a_2 & b_{21} & & & \\ a_3 & b_{31} & b_{32} & & \\ a_4 & b_{41} & b_{42} & b_{43} & \\ \hline & c_1 & c_2 & c_3 & c_4 \end{array}$$

Es sind 13 Parameter zu bestimmen unter der Nebenbedingung:

$$\begin{aligned} a_2 &= b_{21} \\ a_3 &= b_{31} + b_{32} \\ a_4 &= b_{41} + b_{42} + b_{43} \end{aligned}$$

$$a_k = \sum_{j=1}^{k-1} b_{kj} \qquad k = 2,\ 3,\ 4$$

D.h. die Prädiktorwerte y_2^*, y_3^* und y_4^* müssen für die Dgl. $y'(x) = 1$ *exakt* sein (s.o.).

Auch hier wird der lokale Diskretisationsfehler in eine Taylorreihe nach Potenzen der Schrittweite h entwickelt. Die Forderung nach einer Methode der Fehlerordnung 4 führt nach Elimination von 3 Parametern gemäss obiger Nebenbedingungen auf ein nicht-lineares Gleichungssystem mit 8 Gleichungen für 10 Unbekannte. Dieses System hat eine 2-parametrige Lösung, die in Abhängigkeit von a_2 und a_3, analog zu 3-stufigen Verfahren, angegeben werden kann.

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

RK, $p = 4$, *die* Runge-Kutta Methode, sehr populäre Methode, vgl. Hairer, Nørsett und Wanner, Band I, p133 ff

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{3} & \frac{1}{3} & & & \\ \frac{2}{3} & \frac{-1}{3} & 1 & & \\ 1 & 1 & -1 & 1 & \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array}$$

RK, $p = 4$, weniger populär, dafür genauer, cf. MATHEMATICA

2.7 Stabilität

Gegeben ist ein Differentialgleichungssystem 1-ter Ordnung.

Gesucht ist eine numerische Approximation der Lösung.

Bei der Wahl eines bestimmten Verfahrens sind die Eigenschaften der gegebenen Differentialgleichung und der resultierenden Lösungsfunktion zu berücksichtigen. Wird dies unterlassen, kann es geschehen, dass die berechneten Lösungen mit den exakten Lösungen sehr wenig zu tun haben oder schlicht sinnlos sind (man spricht in diesem Zusammenhang auch von Geisterlösungen).

Testproblem

Wir betrachten das Problem

$$\left\{ \begin{array}{rcl} y'(x) & = & \lambda\, y(x) \qquad \lambda \in \mathbb{R} \quad \text{oder} \quad \lambda \in \mathbb{C} \\ y(0) & = & 1 \end{array} \right. \tag{21}$$

mit der exakten Lösung $y(x) = e^{\lambda x}$, $x \geq 0$.

Die folgenden Betrachtungen werden an diesem Testproblem vorgenommen. Die Aussagen bleiben auch für *nicht-lineare* Differentialgleichungen gültig, da diese lokal nach einer Linearisierung (mit Hilfe der Jacobi-Matrix, cf. Abschnitt 2.8.1) in Bezug auf y durch eine lineare Differentialgleichung approximiert werden können. Für einen Integrationsschritt der Schrittweite h ($h > 0$ klein) verhalten sich die Näherungswerte qualitativ gleich.

RK-Verfahren 4-ter Ordnung

Wie wirkt dieses Verfahren auf das Testproblem?

$$\begin{array}{rcll} k_1 & = & \lambda\, y_k & \\ k_2 & = & \lambda\,(y_k + \frac{1}{2} h\, k_1) & = \quad (\lambda + \frac{1}{2} h\, \lambda^2)\, y_k \\ k_3 & = & \lambda\,(y_k + \frac{1}{2} h\, k_2) & = \quad (\lambda + \frac{1}{2} h\, \lambda^2 + \frac{1}{4} h^2 \lambda^3)\, y_k \\ k_4 & = & \lambda\,(y_k + h\, k_3) & = \quad (\lambda + h\, \lambda^2 + \frac{1}{2} h^2 \lambda^3 + \frac{1}{4} h^3 \lambda^4)\, y_k \end{array}$$

Mit diesen Werten k_1, k_2, k_3 und k_4 bekommen wir den Korrektor

$$y_{k+1} = y_k + \frac{h}{6} \cdot \{k_1 + 2\, k_2 + 2\, k_3 + k_4\} = \left(1 + h\, \lambda + \frac{h^2}{2!}\, \lambda^2 + \frac{h^3}{3!}\, \lambda^3 + \frac{h^4}{4!}\, \lambda^4\right)\, y_k$$

D.h. y_{k+1} entsteht aus y_k durch Multiplikation mit dem Faktor

$$R(h\lambda) = \left(1 + h\lambda + \frac{h^2}{2!}\, \lambda^2 + \frac{h^3}{3!}\, \lambda^3 + \frac{h^4}{4!}\, \lambda^4\right)$$

$R(h\lambda)$ hängt nur von $h\lambda$ ab. Offensichtlich stellt $R(h\lambda)$ gerade den Anfang der Taylorreihe von $e^{h\lambda}$ dar (mit dem Fehler $O(h^5)$).

Für die exakte Lösung muss gelten:

$$y(x_{k+1}) = e^{h\lambda} \cdot y(x_k)$$

- $R(h\lambda)$ stellt für betragskleine Werte $h\lambda$ eine gute Approximation für $e^{h\lambda}$ dar.
- $\lambda \in \mathbb{R}$ und $\lambda > 0 \Longrightarrow R(h\lambda) > 1$, d.h. die Näherungswerte y_k werden qualitativ richtig berechnet.
- $\lambda \in \mathbb{R}$ und $\lambda < 0$: die Näherungslösung y_k klingt ab wie $y(x_k) \Longleftrightarrow |R(h\lambda)| < 1$.
  Als Polynom 4-ten Grades ist wegen $\lim\limits_{h\lambda \to -\infty} R(h\lambda) = \infty$ diese Bedingung nicht für alle negativen Werte von $h\lambda$ erfüllt.
- $\lambda \in \mathbb{C}$: oszillierende, exponentiell abklingende Komponenten.
  $|R(h\lambda)| < 1 \Longleftrightarrow \mathrm{Re}(\lambda) < 0$
  D.h. falls y_k wie $y(x_k)$ abklingt, muss die notwendige und hinreichende Bedingung

  $$|R(h\lambda)| < 1$$

  gelten.

Setze $z := h\lambda \in \mathbb{C}$:

$$R(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{z^4}{4!} + O(h^5)$$

$R(z)$ ist die sogenannte *Stabilitätsfunktion*.

Allgemein gilt:

Falls das RK-Verfahren von der Ordnung p ist, gilt:

$$R(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \cdots + \frac{z^p}{p!} + O(h^{p+1}) \qquad \text{wobei} \quad z = h\lambda$$

Definition 2.5

Für ein Einschrittverfahren, für das das Testproblem auf $y_{k+1} = R(z)\, y_k$ führt, heisst die Menge $\mathbb{S} := \{\mu \in \mathbb{C} / |R(\mu)| < 1\}$ *Gebiet der absoluten Stabilität* oder *Stabilitätsgebiet*.

Beispiel 2.1 *Stabilitätsgebiet für die Methode von Euler*

Diese Methode ist wie folgt gegeben:

$$\begin{aligned} k_1 &= f(x_k,\, y_k) \\ y_{k+1} &= y_k + h\, k_1 \end{aligned}$$

und für das Testproblem: $y' = \lambda\, y$

$$\begin{aligned} k_1 &= \lambda\, y_k \\ y_{k+1} &= y_k + h\, \lambda\, y_k = (1 + h\lambda)\, y_k \end{aligned}$$

und somit die *Stabilitätsfunktion* $R(z) := 1 + z$, wobei $z = h\lambda$.

Das Stabilitätsgebiet wird somit:

$$\mathbb{S} = \left\{ \mu \in \mathbb{C} \,\middle|\, |R(\mu)| < 1 \right\} = \left\{ z \in \mathbb{C} \,\middle|\, |R(z)| < 1 \right\}$$

also

$$(1 + x)^2 + y^2 < 1 \qquad \text{das Innere des Kreises mit} \quad M(-1,\, 0) \quad \text{und} \quad r = 1.$$

Falls $\lambda \in \mathbb{R}$ ist die Methode von Euler *stabil*, falls $h\lambda \in (-2,\, 0)$

Beispiel 2.2 $y' = -50\ (y - \cos(x))$

vgl. Abschnitt 3.2
$\lambda = -50$, d.h. $-2 < h\lambda < 0$, also sind $h_1 = \frac{1.974}{50}$ und $h_2 = \frac{1.875}{50}$ innerhalb von S, d.h. für diese beiden Schrittweiten sind die Verfahren stabil! Die Fragen nach der Genauigkeit und ob die Resultate überhaupt brauchbar sind, sind dabei ausser acht gelassen worden.

Beispiel 2.3 *Stabilitätsgebiet für die Methode vom impliziten Euler*

Diese Methode ist wie folgt gegeben:

$$\begin{aligned} k_1 &= f(x_{k+1},\, y_{k+1}) \\ y_{k+1} &= y_k + h\, k_1 \end{aligned}$$

und für das Testproblem: $y' = \lambda\, y$

$$\begin{aligned} k_1 &= \lambda\, y_{k+1} \\ y_{k+1} &= y_k + h\, \lambda\, y_{k+1} \end{aligned}$$

und aufgelöst nach y_{k+1} (da das Testproblem linear ist, kann die implizite Gleichung aufgelöst werden):

$$y_{k+1} = \frac{1}{1-h\lambda}\, y_k$$

und somit die *Stabilitätsfunktion* $R(z) := \frac{1}{1-z}$, wobei $z = h\lambda$.

Das Stabilitätsgebiet wird somit:

$$\mathbb{S} = \left\{ \mu \in \mathbb{C} \,\middle|\, |R(\mu)| < 1 \right\} = \left\{ z \in \mathbb{C} \,\middle|\, |R(z)| < 1 \right\}$$

also

$$(1-x)^2 + y^2 > 1 \qquad \text{das Äussere des Kreises mit} \quad M(1,\,0) \quad \text{und} \quad r = 1.$$

Beispiel 2.4 $y' = -50\ (y - \cos(x))$

vgl. Abschnitt 3.2
$\lambda = -50$, d.h. $h\lambda < 0$, für jedes h, somit ist der implizite Euler ist für *jedes* h stabil.

Beispiel 2.5 *Stabilitätsgebiet für die Trapezmethode*

Diese Methode ist wie folgt gegeben:

$$\int_{x_k}^{x_{k+1}} f(x,\, y)\, dx \simeq \frac{h}{2}\, \{f(x_k,\, y_k) + f(x_{k+1},\, y_{k+1})\}$$

und für das Testproblem: $y' = \lambda\, y$

$$y_{k+1} = y_k + \frac{h}{2}\, \{\lambda\, y_k + \lambda\, y_{k+1}\}$$

und aufgelöst nach y_{k+1} (da das Testproblem linear ist, kann die implizite Gleichung aufgelöst werden):

$$y_{k+1} = \frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}}\, y_k$$

und somit die *Stabilitätsfunktion* $R(z) := \frac{2+z}{2-z}$, wobei $z = h\lambda$ und $z \neq 2$.

Das Stabilitätsgebiet wird somit:

$$\mathbb{S} = \left\{ \mu \in \mathbb{C} \,\middle|\, |R(\mu)| < 1 \right\} = \left\{ z \in \mathbb{C} \,\middle|\, |R(z)| < 1 \right\}$$

also

$$x = \Re(z) < 0 \qquad \text{die ganze linke Halbebene der Gauss'schen Ebene}$$

Definition 2.6 *absolute Stabilität*

Eine Methode für die gilt: $\mathbb{S} = \left\{ z \in \mathbb{C} \,\middle|\, \Re(z) < 0 \right\}$ heisst *absolut* stabil.

D.h. die Trapezmethode ist absolut stabil, für die Schrittweiten sind *keine* Grenzen gesetzt.

Beispiel 2.6 *Stabilitätsgebiet für das Verfahren von Heun*

Diese Methode ist wie folgt gegeben:

$$\begin{aligned} k_1 &= f(x_k,\, y_k) \\ k_2 &= f(x_k + h,\, y_k + h k_1) \\ y_{k+1} &= y_k + \frac{h}{2}\, (k_1 + k_2) \end{aligned}$$

und für das Testproblem: $y' = \lambda\, y$

$$\begin{aligned} k_1 &= \lambda\, y_k \\ k_2 &= \lambda\,(y_k + hk_1) = \lambda\,(1+h\lambda) y_k \\ y_{k+1} &= y_k + \frac{h}{2}\,\{\lambda\, y_k + \lambda(1+h\lambda)\, y_k\} = \left(1 + h\lambda + \frac{h^2\lambda^2}{2}\right)\, y_k \end{aligned}$$

und somit die *Stabilitätsfunktion* $R(z) := 1 + z + \frac{z^2}{2}$, wobei $z = h\lambda$.

Das Stabilitätsgebiet wird somit:

$$\mathbb{S} = \left\{\mu \in \mathbb{C} \,\middle|\, |R(\mu)| < 1\right\} = \left\{z \in \mathbb{C} \,\middle|\, |R(z)| < 1\right\}.$$

Hier wird es hier etwas "hässlicher", bzw. aufwendiger:

$$\left|1 + z + \frac{z^2}{2}\right| < 1 \qquad \text{also} \qquad \left|1 + x + \frac{1}{2}(x^2 - y^2) + j(xy + y)\right| < 1$$

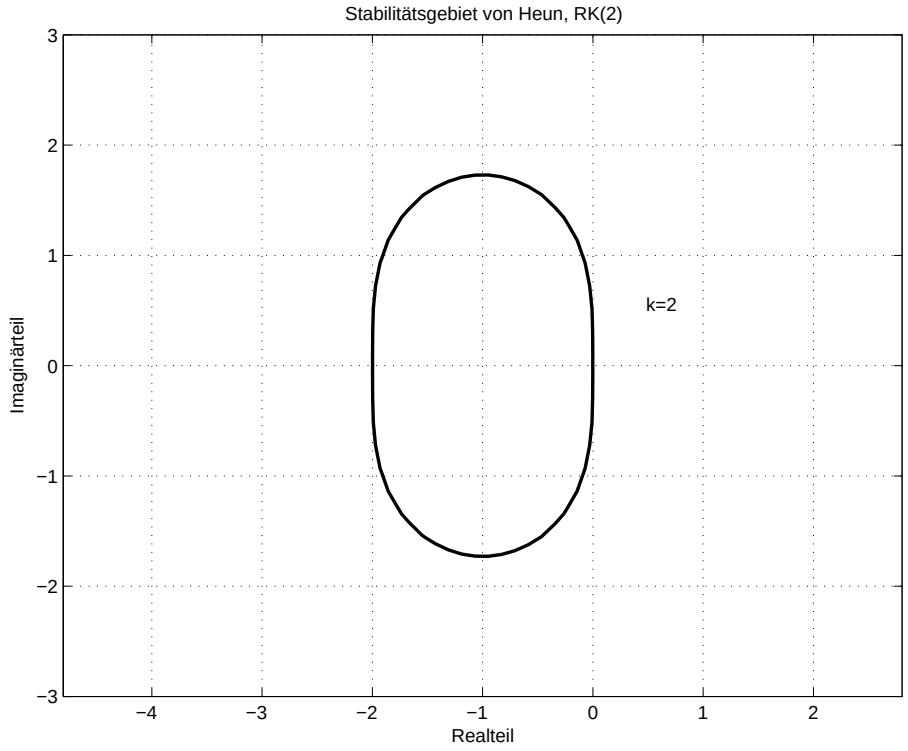


Abbildung 13: Stabilitätsgebiet der Methode von Heun

Das zugehörige MATLAB - File lautet:

```
% graphische Darstellung eines
% Stabilitätsgebietes (RK):
% Verfahren von Heun:

x = linspace(-3, 0.5, 30);
y = linspace(-3, 3, 30);

[xx, yy] = meshgrid(x,y);
```

```
f = abs(1+xx+0.5*xx.^2-0.5*yy.^2+i*(yy+xx.*yy));
[ku, ha] = contour(xx, yy, f, [1 1], 'r');
set(ha, 'LineWidth', 1.8);
grid
axis('equal')
gtext('k=2')
title('Stabilitätsgebiet von Heun, RK(2)')
xlabel('Realteil')
ylabel('Imaginärteil')
```

Ein Mass für die Grösse des Stabilitätsgebiets $\mathbb{S}$ für reelle negative Eignewerte λ ist das *Stabilitätsintervall* auf der reellen Achse der Gauss'schen Ebene:

explizite Runge–Kutta Methoden der Ordnung p					
p	1	2	3	4	5
Intervall	$(-2,\ 0)$	$(-2,\ 0)$	$(-2.51,\ 0)$	(-2.78, 0)	(-3.21, 0)

Bemerkung 2.3

Frage: Zusammenhang Stabilität mit Lipschitzbedingung?

Die Lipschitzbedingung an Φ ist eine schwache Bedingung. Damit nun ein numerisches Verfahren brauchbare Approximationen liefert, muss mehr gelten: es muss z.B. stabil sein ($|R(z)| < 1$ liefert uns eine konvergente Iteration, d.h. eine Fixpunktiteration mit $L < 1$), cf. dazu die Abschätzung des globalen Diskretisationsfehlers, Abschnitt 2.3.1 und obige Definition der Stabilität 2.3.

2.8 Steife Differentialgleichungen

Physikalische, chemische oder biologische Vorgänge die sich aus Anteilen zusammensetzen, die *stark verschieden rasch exponentiell abklingen*.

Vergeichen Sie dazu das Beispiel in Abschnitt 3.2. Für $c = 50$ haben wir *stark* unterschiedlich abklingende Lösungskomponenten. Dabei ist

$$\frac{c^2}{c^2+1}\cos(x)$$

der *dominante* Teil der Lösung (46).

- Offensichtlich existiert in der Nähe von $y \simeq \cos(x)$ eine „brave" Lösung.
- Alle anderen Lösungen erreichen diese „brave" Lösung nach einer sehr kurzen „Übergangsphase" (= *transiente* Phase). Mit andern Worten haben wir ein sogenanntes „Runterfallen" auf $y \simeq \cos(x)$.
- Solche transienten Phasen sind typisch für steife Differentialgleichungen (aber weder hinreichend noch notwendig!)
  Z.B. hat die Lösung mit der Anfangsbedingung $y(0) = 1$ oder genauer $y(0) = \frac{2500}{2501}$ *keine* transiente Phase. Für diese AB besteht die Lösung (46) nur aus dem mittleren Term.
- Falls die Schrittweite h etwas zu gross ist, hat die numerische Lösung mit der gesuchten Lösung *nichts* zu tun.

Für dieses Beispiel mit $\lambda = -50$ heisst das für den expliziten Euler: $-2 < h\lambda < 0$ bzw. $0 < h < \frac{2}{50}$ und für $h \geq \frac{2}{50}$ haben wir Instabilität des Verfahrens, cf. Folien.

Stabilität ist das eine, das andere ist die Genauigkeit. Selbst dann, wenn h so gewählt wird, dass die Stabilitätsbedingung erfüllt wird, hat unter Umständen die gerechnete Lösung nicht viel mit der

gesuchten Lösung zu tun, cf. Folien.

Das Beispiel aus dem Abschnitt 3.2 soll jetzt nicht nur stabil, sondern auch genau numerisch integriert werden. Dabei wird der klassische Runge-Kutta der Ordnung $p = 4$ verwendet.

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

mit

$$R(h\lambda) = 1 + (h\lambda) + \frac{(h\lambda)^2}{2!} + \frac{(h\lambda)^3}{3!} + \frac{(h\lambda)^4}{4!}$$

Global sollen z.B. 4 Stellen korrekt sein, d.h. lokal muss 5 stellig korrekt gerundet werden, was

$$\left|R(h\lambda) - e^{h\lambda}\right| \simeq \left|\frac{(h\lambda)^5}{5!}\right| \leq 10^{-4} \qquad \text{also} \qquad h \simeq 8.25 \cdot 10^{-3}$$

zur Folge hat.

Wann ist der Term e^{-50x} klein genug, d.h. $e^{-50x} \leq 10^{-4}$, sodass er bei der weiteren Integration keinen Einfluss mehr auf die ersten 4 Stellen hat? Für $x_1 \geq \frac{\ln(10^{-4})}{50} \simeq 0.1842\ldots$ ist dies der Fall.

Für $x > x_1$ kann dieser Term gegenüber den beiden anderen Termen vernachlässigt werden. Diese rasch abklingende und bereits kleine Komponente braucht ab x_1 nicht mehr so genau integriert zu werden. Die Schrittweite darf *vergrössert* werden, aber nur so, dass die Stabilitätsbedingung weiterhin gültig bleibt.

Falls weitere exponentiell abklingende Komponenten vorhanden sind, wird analog vorgegangen.

Beispiel 2.7

Betrachten wir das folgende Differentialgleichungssystem $y' = Ay$

$$(22) \qquad \left\{\begin{array}{rcrcrcr} y_1' &=& -0.5y_1 &+& 32.6y_2 &+& 35.7y_3 \\ y_2' &=& &-& 48y_2 &+& 9y_3 \\ y_3' &=& & & 9y_2 &-& 72y_3 \end{array}\right. \quad \text{mit den AB} \quad \left\{\begin{array}{rcr} y_1(0) &=& 4 \\ y_2(0) &=& 13 \\ y_3(0) &=& 1 \end{array}\right.$$

(22) ist linear und homogen, d.h. die exakte Lösung kann durch Entkopplung angegeben werden.

$$y_h(x) = c_1 e^{\lambda_1 x} \cdot v^{(1)} + c_2 e^{\lambda_2 x} \cdot v^{(2)} + c_3 e^{\lambda_3 x} \cdot v^{(3)},$$

wobei (EWP von A):

$$\lambda_1 = -0.5 \quad v^{(1)} = \mu_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \lambda_2 = -45 \quad v^{(1)} = \mu_1 \begin{pmatrix} -3 \\ 3 \\ 1 \end{pmatrix} \quad \lambda_3 = -75 \quad v^{(1)} = \mu_1 \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix}$$

Bestimmung der Konstanten c_k, $k = 1, 2, 3$ mit den AB ergibt die Lösung

$$(23) \qquad y(x) = 15 \cdot e^{-0.5x} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + 4 \cdot e^{-45x} \begin{pmatrix} -3 \\ 3 \\ 1 \end{pmatrix} + e^{-75x} \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix} \qquad x \geq 0.$$

Eine adaptive Schrittweitensteuerung muss folgendes können:

Da sich die Lösung $y(x)$ in (23) aus drei völlig unterschiedlichen Termen zusammengesetzt ist, muss h zuerst für den dritten Term, anschliessend für den zweiten und schliesslich für den ersten Term angepasst werden.

Soll (22) nun z.B. numerisch mit der Methode "verbesserter Polygonzug" $4-$stellig korrekt gelöst werden, so muss die Schrittweite wie folgt gesteuert werden:

- die Stabilität verlangt, falls $\lambda < 0$:

$$-2 < h\lambda < 0 \tag{24}$$

- die Genauigkeit verlangt: $\varepsilon = 5 \cdot 10^{-5}$
- "verbesserter Polygonzug": $R(h\lambda) = 1 + (h\lambda) + \frac{(h\lambda)^2}{2!}$

Also für $\lambda_3 = -75$

$$\frac{|h\lambda|^3}{3!} < \varepsilon \Longrightarrow h_1 = 8.9258e - 004$$

Die einzelnen Komponenten des dritten Terms von (23) sind genau genug, falls

$$e^{-75x} < \varepsilon \Longrightarrow x_1 > -\frac{1}{75} \cdot \ln(5 \cdot 10^{-5}) = 0.1320$$

Bis zu diesem x_1 werden mit h_1 bereits $n_1 = \frac{x_1}{h_1} = 147.9385$, also $n_1 = 148$ Schritte gemacht.

Eine analoge Überlegung für $\lambda_2 = -45$ liefert:

$$h_2 = 0.0015 \qquad \text{bis zu } x_2 = 0.2201 \qquad \text{mit } n_2 = \frac{x_2 - x_1}{h_2} = 59.1754$$

d.h. mit der zweiten Schrittweite werden $n_2 = 60$ Schritte durchgeführt.

Für $\lambda_1 = -0.5$ schliesslich liefert obige Umfromung: $h_3 < \sqrt[3]{2.4} \cdot 10^{-1}$.
Dieses h_3 verletzt die Stabilitätsbedingung (24)!
Für h_3 muss auch die folgende Ungleichung gelten $0 < h_3 < \frac{2}{75}$.

Allgemeines Vorgehen

Immer zuerst die am schnellsten abklingende Komponente betrachten, dann die nächst schnellste Komponente, usw. ... unter Einhaltung von (24).

2.8.1 Lokale Linearisierung - Jacobi Matrix

cf. Abschnitt 2.7

Um den Zusammenhang zwischen unserem Testproblem (21) in Abschnitt 2.7 und der allgemeinen Situation herzustellen, brauchen wir die *Jacobi-Matrix* eines Differentialgleichungssystems. Dazu diene uns das

Beispiel 2.8

$$\left\{\begin{array}{rcrcrcl} \dot{y}_1(t) &=& -0.001\,y_1(t) &+& 0.001\,y_2(t) & & \\ \dot{y}_2(t) &=& y_1(t) &-& y_2(t) &-& y_1(t)y_3(t) \\ \dot{y}_3(t) &=& y_1(x)y_2(t) &-& 100\,y_3(t) & & \end{array}\right. \quad \text{mit} \quad \left\{\begin{array}{rcl} y_1(0) &=& 0 \\ y_2(0) &=& 1 \\ y_3(0) &=& 1 \end{array}\right. \tag{25}$$

Das Beispiel beschreibt eine kinetische Reaktion dreier chemischer Substanzen Y_1, Y_2 und Y_3 nach dem Massenwirkungsgesetz, wobei $y_1(t)$, $y_2(t)$ und $y_3(t)$ die entsprechenden Konzentrationen der Substanzen zum Zeitpunkt t sind. Die verschiedenen Reaktionen laufen mit *sehr* unterschiedlichen Zeitkonstanten ab.

Definition 2.7 *Steifigkeit*

Die Steifigkeit eines gegebenen Differentialgleichungssystems ist mit

$$S(t) = \frac{\max |\mathrm{Re}(\lambda(t))|}{\min |\mathrm{Re}(\lambda(t))|}$$

gegeben. Dabei sind $\lambda(t)$ die Eigenwerte der zugehörigen *Jacobi-Matrix*

$$J(t,y) = \left(\frac{\partial f}{\partial y}\right) = \left(\frac{\partial f_i}{\partial y_j}\right) \qquad 1 \le i,\, j \le n$$

In obigem Beispiel haben wir

$$\left\{ \begin{array}{lcll} \dot{y}_1(t) & = & f_1(y_1,\, y_2,\, y_3) & \\ \dot{y}_2(t) & = & f_2(y_1,\, y_2,\, y_3) & \dot{y} = f(y) \\ \dot{y}_3(t) & = & f_3(y_1,\, y_2,\, y_3) & \end{array} \right.$$

Hier kommt t in f nicht explizit vor, es handelt sich um ein *autonomes* Differentialgleichungssystem.
In unserem Beispiel haben wir also

$$\begin{array}{llll} \frac{\partial f_1}{\partial y_1} = -0.01 & \frac{\partial f_1}{\partial y_2} = 0.01 & \frac{\partial f_1}{\partial y_3} = 0 & i = 1,\, j = 1,2,3 \\ \frac{\partial f_2}{\partial y_1} = 1 - y_3 & \frac{\partial f_2}{\partial y_2} = -1 & \frac{\partial f_2}{\partial y_3} = -y_1 & i = 2,\, j = 1,2,3 \\ \frac{\partial f_3}{\partial y_1} = y_2 & \frac{\partial f_3}{\partial y_2} = y_1 & \frac{\partial f_3}{\partial y_3} = -100 & i = 3,\, j = 1,2,3 \end{array}$$

und somit

$$J(t,y) = \begin{pmatrix} -0.01 & 0.01 & 0 \\ 1 - y_3 & -1 & -y_1 \\ y_2 & y_1 & -100 \end{pmatrix} = J(y) \qquad \text{die Zeit } t \text{ kommt nicht explizit vor}$$

Für $t = 0, y(0) = y_0$

$$J(y_0) = \begin{pmatrix} -0.01 & 0.01 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & -100 \end{pmatrix}$$

Mit Hilfe des Eigenwertproblems von $J(0)$ kann das Differentialgleichungssystem entkoppelt werden, d.h.

$$\dot{z} = Dz \qquad D = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

und wir haben formal das Testproblem (21).

Die Eigenwerte von $J(y_0)$ sind $\lambda_1 = -100$, $\lambda_2 = -1$ und $\lambda_3 = -0.01$ und damit bekommen wir $S(0) = 10^4$ für die Steifigkeit zur Zeit $t = 0$.

Nach dem ersten Zeitschritt $t_1 = t_0 + \Delta t$ haben wir dasselbe Prozedere durchzufühhren, usw. Auf diese Weise ist es möglich, zu jedem Zeitpunkt die Steifigkeit $S(t)$ zu bestimmen. Daneben haben wir mit den Eigenwerten von $J(t)$ komponentenweise auch die Stabiltätsbedingungen im nicht-linearen Fall.

Beispiel 2.9

Als weiteres Beispiel betrachten wir das Räuber-Beute Modell in der folgenden Formulierung

$$\begin{cases} \dot{y}_1 &= ay_1(1-y_2) &= f_1(y_1,y_2) \\ \dot{y}_2 &= y_2(y_1-1) &= f_2(y_1,y_2) \end{cases} \quad \text{mit den AB} \quad \begin{cases} y_1(0) &= 3 \\ y_2(0) &= 1 \end{cases} \quad \text{z.B. für } a=10$$

Hier haben wir

$$J(y) = \begin{pmatrix} a(1-y_2) & -ay_1 \\ y_2 & y_1-1 \end{pmatrix}$$

und

$$J(y_0) = \begin{pmatrix} 0 & -30 \\ 1 & 2 \end{pmatrix}$$

mit den beiden Eigenwerten $\lambda_{1.2} = 1 \pm j\sqrt{29}$. Die Steifigkeit in diesem Fall ist also $S(0) = 1$. Das wird für alle Zeiten so sein, da das System mit den gegebenen Anfangsbedingungen für den Wert $a = 10$ schwingt. Die Eigenwerte treten konjugiert komplex auf und wir haben nur zwei Eigenwerte, somit ist $\max \mathrm{Re}|\lambda(t)| = \min \mathrm{Re}|\lambda(t)|$. Hier ist die Steifigkeit also kein Problem.

Bemerkung 2.4

Bei Simulink Modellen ist es oft gut, die System–Matrix der lokalen Linearisierung zur Verfügung zu haben.

Im Submenu „Linearization and Trimming Commands – Alphabetical List“ von Simulink findet man den Befehl `linmod`.

Mit `[A,B,C,D] = linmod('name')` erhalten wir die System–Matrix A womit wir die Kondition als auch die Steifigkeit von A bestimmen können.

Dabei wird das System in den Zustandsvariablen wie folgt beschrieben:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

wobei x und u Input- und y eine Outputgrösse ist.

Bemerkung 2.5

Ist das gegebene Differentialgleichungssystem linear (cf. Beispiel (22)), so ist die zughörige Jacobi–Matrix identisch mit der gegebenen Systemmatrix und *konstant*.

2.8.2 Implizite Verfahren, Euler implizit, Trapezmethode

Als Beispiel zur Illustration soll uns

$$\ddot{x} + x = \varepsilon x^3 \qquad \varepsilon > 0, \text{ klein, z.B. } \varepsilon = 10^{-1}, 10^{-2} \tag{26}$$

dienen. (26) beschreibt einen nicht-linear gestörten Harmonischen Oszillator.

Mit der Substitution $y_1 = x$ und $y_2 = \dot{x}$ erhalten wir

$$\dot{y} = f(y) = \begin{pmatrix} y_2 \\ -y_1 + \varepsilon y_1^3 \end{pmatrix} \qquad y_0 = \begin{pmatrix} -2\pi \\ v_0 \end{pmatrix} \quad v_0 = 2.9, 3.8, 4.5, 5.5 \tag{27}$$

(27) ist ein sog. *autonomes* Differentialgleichungssystem, da in f die Zeit t nicht explizit vorkommt.

Jede explizite Methode funktioniert auch hier "problemlos" (abgesehen von Genauigkeit und Stabilität), also auch die Methode **Euler explizit**

$$y_{k+1} = y_k + h \cdot f(y_k) = y_k + h \cdot \begin{pmatrix} y_{k_2} \\ -y_{k_1} + \varepsilon y_{k_1}^3 \end{pmatrix}$$

wobei $y_k = \begin{pmatrix} y_{k_1} \\ y_{k_2} \end{pmatrix}$.

Euler implizit

$$\begin{aligned} y_{k+1} &= y_k + h \cdot f(y_{k+1}) \\ h \cdot f(y_{k+1}) - y_{k+1} + y_k &=: F(y_{k+1}) = 0 \end{aligned} \tag{28}$$

(28) ist ein nicht–lineares Gleichungssystem, das mit Newton–Raphson gelöst werden kann, cf. **Abschnitt 3.6.2**.

Für jeden Schritt der Schrittweite h ist also ein "Newton–Raphson" mit k Iterationen durchzuführen. dabei ist die Jacobi–Matrix

$$J_{\text{Euler implizit}}(y_{k+1}) = \left(\frac{\partial F_i}{\partial y_{k+1_j}}\right) = h \cdot \left(\frac{\partial f_i}{\partial y_{k+1_j}}\right) - I_2 \qquad 1 \le i,j \le 2 \tag{29}$$

Für (27) ist (29) wie folgt

$$J_{\text{Euler implizit}}(y) = h \cdot \begin{pmatrix} 0 & 1 \\ -1 + 3\varepsilon y_1^2 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad y := y_{k+1}$$

Um die Notation zu vereinfachen, wurde in dieser letzten Formel y_{k+1} durch y ersetzt.
Für jeden Zeitschritt ist ein nicht–lineares Gleichungssystem $F(y) = h \cdot f(y) - y + y_k = 0$ zu lösen.

erster Zeitschritt $y_0 =: y_1^{(0)} \longrightarrow y_1^{(k+1)}$

- Newton–Raphson: erste Wiederholung
  - i) $y_1^{(0)} := y_0$ Starwert. $F(y_1^{(0)}) = z^{(0)}$
  - ii) "Tangente" E_0 durch den Punkt $P_0(y_1^{(0)}, z^{(0)})$ an die Fläche $z = F(y)$

    $$E_0 \colon J_{\text{Euler implizit}}(y_1^{(0)})(y - y_1^{(0)}) = z - z^{(0)}$$

  - iii) Schnittpunkt $y_1^{(1)}$ von E_0 mit der $z_1 z_2-$ Ebene $(z = 0)$

    $$J_{\text{Euler implizit}}(y_1^{(0)})(y - y_1^{(0)}) = -z^{(0)} \quad \Longrightarrow (y - y_1^{(0)}) =: \Delta y = -\left(J_{\text{Euler implizit}}(y_1^{(0)})\right)^{-1} z^{(0)}$$

    und damit

    $$y_1^{(1)} = y_1^{(0)} + \Delta y = y_1^{(0)} - \left(J_{\text{Euler implizit}}(y_1^{(0)})\right)^{-1} z^{(0)}$$

    Zur Bestimmung von Δy muss ein lineares Gleichungssystem mit der Systemmatrix $J_{\text{Euler implizit}}(y_1^{(0)})$ gelöst werden.

- Newton–Raphson: zweite Wiederholung
  - i) $F(y_1^{(1)}) = z^{(1)}$
  - ii) "Tangente" E_1 durch den Punkt $P_1(y_1^{(1)}, z^{(1)})$ an die Fläche $z = F(y)$

    $$E_1 \colon J_{\text{Euler implizit}}(y_1^{(1)})(y - y_1^{(1)}) = z - z^{(1)}$$

  - iii) Schnittpunkt $y_1^{(2)}$ von E_1 mit der $z_1 z_2-$ Ebene $(z = 0)$

    $$J_{\text{Euler implizit}}(y_1^{(1)})(y - y_1^{(1)}) = -z^{(1)} \quad \Longrightarrow (y - y_1^{(1)}) =: \Delta y = -\left(J_{\text{Euler implizit}}(y_1^{(1)})\right)^{-1} z^{(1)}$$

    und damit

    $$y_1^{(2)} = y_1^{(1)} + \Delta y = y_1^{(1)} - \left(J_{\text{Euler implizit}}(y_1^{(1)})\right)^{-1} z^{(1)}$$

Zur Bestimmung von Δy muss ein lineares Gleichungssystem mit der Systemmatrix $J_{\text{Euler implizit}}(y_1^{(1)})$ gelöst werden.

...

- Newton–Raphson: $(k+1)-$ Wiederholung
  i) $F(y_1^{(k)}) = z^{(k)}$
  ii) "Tangente" E_k durch den Punkt $P_k(y_1^{(k)}, z^{(k)})$ an die Fläche $z = F(x)$

$$E_k\colon J_{\text{Euler implizit}}(y_1^{(k)})(y - y_1^{(k)}) = z - z^{(k)}$$

  iii) Schnittpunkt $y_1^{(k+1)}$ von E_k mit der z_1z_2- Ebene $(z = 0)$

$$J_{\text{Euler implizit}}(y_1^{(k)})(y-y_1^{(k)}) = -z^{(k)} \quad \Longrightarrow (y-y_1^{(k)}) =: \Delta y = -\left(J_{\text{Euler implizit}}(y_1^{(k)})\right)^{-1} z^{(k)}$$

  und damit

$$y_1^{(k+1)} = y_1^{(k)} + \Delta y = y_1^{(k)} - \left(J_{\text{Euler implizit}}(y_1^{(k)})\right)^{-1} z^{(k)}$$

zweiter Zeitschritt $y_1^{(k+1)} =: y_2^{(0)} \longrightarrow y_2^{(k+1)}$
Das Resultat des ersten Zeitschritts wird als Statwert für den zweiten Zeitschritt verwendet.

Allgemein gilt:

- Zur Bestimmung von Δy muss jeweils ein lineares Gleichungssystem mit der Systemmatrix aus (29) gelöst werden.
- Damit $(k+1)-$ te Wiederholung beim Newton–Raphson durchführbar ist, muss die jeweilige Jacobi–Matrix regulär sein.
- Im $j-$ ten Zeitschritt ist die Folge der $y_j^{(k)}$, $k = 0, 1, 2, \ldots$ konvergent, falls $\|\Delta y\| \longrightarrow 0$ für $k \longrightarrow \infty$.
- Da bei jedem Zeitschritt das Resultat des vorangegangenen Zeitschritts als Startwert verwendet wird, ist die Konvergenz i. allg. gut.

Trapezmethode Die Trapezmethode ist ebenso eine implizite Methode.

$$\begin{aligned} y_{k+1} &= y_k + \frac{h}{2} \cdot \{f(y_k) + f(y_{k+1})\} \\ \frac{h}{2} \cdot f(y_{k+1}) - y_{k+1} + \frac{h}{2} f(y_k) + y_k &=: F(y_{k+1}) = 0 \end{aligned} \tag{30}$$

Auch hier ist (30) ist ein nicht–lineares Gleichungssystem $F(y) = \frac{h}{2} \cdot f(y) - y + \frac{h}{2} f(y_k) + y_k = 0$, $y := y_{k+1}$, das mit Newton–Raphson gelöst werden kann, cf. Abschnitt 3.6.2.

$$J_{\text{Trapez}}(y_{k+1}) = \left(\frac{\partial F_i}{\partial y_{k+1_j}}\right) = \frac{h}{2} \cdot \left(\frac{\partial f_i}{\partial y_{k+1_j}}\right) - I_2 \qquad 1 \le i, j \le 2 \tag{31}$$

Für (27) ist (31) wie folgt

$$J_{\text{Trapez}}(y) = \frac{h}{2} \cdot \begin{pmatrix} 0 & 1 \\ -1 + 3\varepsilon y_1^2 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad y := y_{k+1}$$

Um die Notation zu vereinfachen, wurde in dieser letzten Formel y_{k+1} durch y ersetzt.
Ansonsten ist das Prozedere analog zur Methode "Euler imlizit".

2.8.3 Implizite Verfahren, implizite Runge–Kutta Verfahren (IRK)

Zur Lösung von steifen Differentialgleichungen sind implizite Verfahren bestens geeignet.
Bereits kennengelernt haben wir den impliziten Euler und die Trapezmethode.

Wir betrachten als nächstes implizite Runge-Kutta Verfahren.
Neu an diesen Verfahren ist, dass die Prädiktoren durch *implizite* Gleichungen definiert sind.

Prinzip für ein $1-$stufiges Verfahren:

$$\begin{aligned} k_1 &= f(x_k + a_1 h,\, y_k + b_{11} h k_1) \\ y_{k+1} &= y_k + h c_1 k_1 \end{aligned}$$

- Wie bei den expliziten Verfahren, muss auch hier die Gleichung $y'(x) = 1$ mit $y(0) = 0$ mit der Lösung $y(x) = x$ exakt integriert werden können.
- Diese Forderung führt auf die Nebenbedingung $a_1 = b_{11}$.
- Die Parameter sollen so bestimmt werden, dass das Verfahren eine möglichst hohe Fehlerordnung hat.

Statt mit den Näherungen k_1 und y_k wird nun mit den exakten Werten $\overline{k}_1$ und $y(x_k)$ gerechnet:

$$\begin{aligned} \overline{k}_1 &= \overline{k}_1(x_k,\, y(x_k)) \\ &= f(x_k + a_1 h,\, y(x_k) + b_{11} h \overline{k}_1) \quad \hat{=} \quad f(x_0 + \Delta x,\, y_0 + \Delta y) \\ &= f(x_k,\, y(x_k)) + a_1 h \left\{ f_x + \overline{k}_1 f_y \right\} + \tfrac{1}{2} a_1^2 h^2 \left\{ f_{xx} + 2\overline{k}_1 f_{xy} + \overline{k}_1^2 f_{yy} \right\} + O(h^3) \end{aligned} \tag{32}$$

Für die dritte Zeile wurde $(x_0,\, y_0) = (x_k,\, y(x_k))$ als Entwicklungszentrum für die Taylorreihe von $f(x_0 + \Delta x,\, y_0 + \Delta y)$ verwendet.

Diese implizite Gleichung (32) für $\overline{k}_1$ soll nach $\overline{k}_1$ „aufgelöst“ werden. Dazu wird $\overline{k}_1$ als Potenzreihe in h angesetzt.

$$\overline{k}_1 = \alpha_0 + \alpha_1 h + \alpha_2 h^2 + \ldots \tag{33}$$

(33) in (32) einsetzen

$$\begin{aligned} \alpha_0 + \alpha_1 h + \alpha_2 h^2 + \ldots &= f(x_k,\, y(x_k)) + a_1 h \left\{ f_x + (\alpha_0 + \alpha_1 h) f_y \right\} \\ &\quad + \tfrac{1}{2} a_1^2 h^2 \left\{ f_{xx} + (2\alpha_0) f_{xy} + \alpha_0^2 f_{yy} \right\} + O(h^3) \end{aligned}$$

und anschliessender Koeffizientenvergleich liefert:

$$\begin{aligned} h^0: &\quad \alpha_0 = f(x_k,\, y(x_k)) \\ h^1: &\quad \alpha_1 = a_1 f_x + a_1 \alpha_0 f_y = a_1 \left\{ f_x + f f_y \right\} &&= a_1 F \\ h^2: &\quad \alpha_2 = a_1 \alpha_1 f_y + \frac{1}{2} a_1^2 (f_{xx} + 2 f f_{xy} + f^2 f_{yy}) &&= a_1^2 \left\{ F f_y + \frac{1}{2} G \right\} \end{aligned}$$

Mit diesen Werten lässt sich der lokale Diskretisationsfehler abschätzen:

$$\begin{aligned} d_{k+1} &= y(x_{k+1}) - y(x_k) - h c_1 \overline{k}_1 \\ &= y(x_{k+1}) - y(x_k) - h c_1 \left\{ f + a_1 F h + a_1^2 \left\{ F f_y + \tfrac{1}{2} G \right\} h^2 \ldots \right\} \\ &= \left\{ 1 - c_1 \right\} h f + \left\{ \tfrac{1}{2} - a_1 c_1 \right\} h^2 F + \left\{ \left(\tfrac{1}{6} - a_1^2 c_1 \right) F f_y + \left(\tfrac{1}{6} - \tfrac{1}{2} a_1^2 c_1 \right) G \right\} h^3 + O(h^4) \end{aligned}$$

Mit der Taylorentwicklung von $y(x_{k+1})$ um x_k:

$$y(x_{k+1}) = y(x_k) + f h + F \frac{h^2}{2!} + (F f_y + G) \frac{h^3}{3!} + \ldots$$

erhalten wir schliesslich:

$$d_{k+1} = \{1 - c_1\}\ hf + \left\{\frac{1}{2} - a_1 c_1\right\} h^2 F + \left\{\left(\frac{1}{6} - a_1^2 c_1\right) F f_y + \left(\frac{1}{6} - \frac{1}{2} a_1^2 c_1\right) G\right\} h^3 + O(h^4)$$

Möglichst hohe Fehlerordnung heisst, möglichst viele Koeffizienten der Entwicklung Null setzen. Mit $c_1 = 1$ und $a_1 = \frac{1}{2}$ werden die beiden ersten geschweiften Klammern Null, der Koeffizient von h^3 kann nicht gleichzeitig Null gesetzt werden.

algorithmische Formulierung

$$k_1 = f\left(x_k + \frac{1}{2}h,\, y_k + \frac{1}{2} hk_1\right) \tag{34}$$

$$y_{k+1} = y_k + hk_1 \tag{35}$$

Dieses IRK Verfahren hat die Fehlerordnung $p = 2$.

- Jeder Iterationsschritt verlangt die Lösung einer impliziten Gleichung für k_1. Diese Gleichung kann z.B. mit Fixpunktiteration numerisch gelöst werden.
- Im Vergleich zur Trapezmethode, die ebenfalls implizit ist, bietet diese Methode keine besonderen Vorteile.

2.8.4 Stabilitätsgebiete für IRK

Dazu werden lediglich zwei Beispiele betrachtet. Zur Bestimmung dieser Gebiete wird das Testbeispiel $y' = \lambda y$ verwendet.

Beispiel 2.10 $1-$ *stufiges Verfahren*

$$\begin{aligned} k_1 &= \lambda\left(y_k + \ hk_1\right) \\ k_1\left(1 - \frac{1}{2} h\lambda\right) &= \lambda y_k \\ k_1 &= \frac{\lambda}{\left(1 - \frac{1}{2} h\lambda\right)} y_k \end{aligned}$$

und somit

$$y_{k+1} = y_k + hk_1 = \left(\frac{1 + \frac{1}{2} h\lambda}{1 - \frac{1}{2} h\lambda}\right) y_k$$

Mit $z := h\lambda$ wird die Satbilitätsfunktion

$$R(z) = \frac{2 + z}{2 - z}$$

und das Stabilitätsgebiet

$$\mathbb{S} = \left\{\mu \in \mathbb{C} \middle| \, |R(\mu)| < 1\right\} = \left\{\mu \in \mathbb{C} \middle| \, \mathrm{Re}(\mu) < 0\right\} = \text{linke Halbebene der Gauss'schen Zahlenebene}$$

d.h. auch diese Methode ist *absolut stabil*.

Ansatz für ein $2-$ stufiges Verfahren

$$\begin{aligned} k_1 &= f\left(x_k + a_1 h,\, y_k + h b_{11} k_1 + h b_{12} k_2\right) \\ k_2 &= f\left(x_k + a_2 h,\, y_k + h b_{21} k_1 + h b_{22} k_2\right) \\ y_{k+1} &= y_k + h\left\{c_1 k_1 + c_2 k_2\right\} \end{aligned}$$

Hier ist $m = 2$. Wir haben 8 Parameter eingeführt. Wie für die expilziten Verfahren muss das Problem $y' = 1$ mit $y(0) = 0$ *exakt* nachgebildet werden können, d.h.

$$\begin{aligned} a_1 &= b_{11} + b_{12} \\ a_2 &= b_{21} + b_{22} \end{aligned}$$

Mit diesen beiden Nebenbedingungen haben wir $2 \cdot 3$ freie Parameter. Die maximale Fehlerordnung ist $p = 4$.

allgemein gilt:

Ein $m-$stufiges implizites Runge-Kutta Verfahren hat mit geeigneter Wahl der $m \cdot (m+1)$ Parameter eine maximal erreichbare Fehlerordnung von $p = 2m$.

Beispiel 2.11 $2-$ *stufiges Verfahren*

Hier ein spezielles Beispiel zu obigem allgemeinen Ansatz:

$a_1 = \frac{3-\sqrt{3}}{6}$, $b_{11} = \frac{1}{4}$, $b_{12} = \frac{3-2\sqrt{3}}{12}$ mit $a_1 = b_{11} + b_{12}$
$a_2 = \frac{3+\sqrt{3}}{6}$, $b_{21} = \frac{3+2\sqrt{3}}{12}$, $b_{22} = \frac{1}{4}$ mit $a_2 = b_{21} + b_{22}$ und damit

$$\begin{aligned} k_1 &= f\left(x_k + \frac{3-\sqrt{3}}{6}\, h,\, y_k + \frac{1}{4}\, h k_1 + \frac{3-2\sqrt{3}}{12}\, h k_2\right) \\ k_2 &= f\left(x_k + \frac{3+\sqrt{3}}{6}\, h,\, y_k + \frac{3+2\sqrt{3}}{12}\, h k_1 + \frac{1}{4}\, h k_2\right) \\ y_{k+1} &= y_k + \frac{h}{2}\left(k_1 + k_2\right) \end{aligned}$$

- Alle impilziten Verfahren sind eindeutig bis auf eine triviale Vertauschung der $k-$ Werte.
- Die Analyse des lokalen Fehlers d_{k+1} ist aufwendig.

3 Anhang

3.1 Herleitung von d_{k+1} für RK, $p = 3$, cf. (14)

Da die betrachteten Verfahren der Ordnung $p = 3$ sind, muss bis und mit dem Term dritter Ordnung entwickelt werden.

Taylorentwicklung von $y(x_{k+1})$ mit Entwicklungszentrum x_k:

$$y(x_{k+1}) = y(x_k) + \frac{y'(x_k)}{1!} \cdot h + \frac{y''(x_k)}{2!} \cdot h^2 + \frac{y'''(x_k)}{3!} \cdot h^3 + O(h^4)$$

Mit

$$y''(x) = f_x + f \cdot f_y =: F \tag{36}$$

$$y'''(x) = (f_{xx} + 2\, f \cdot f_{xy} + f^2 \cdot f_{yy}) + (f_x + f \cdot f_y)\, f_y =: G + F \cdot f_y \tag{37}$$

erhalten wir

$$y(x_{k+1}) = y(x_k) + f \cdot h + \frac{1}{2!} F \cdot h^2 + \frac{1}{3!}(G + F \cdot f_y) \cdot h^3 + O(h^4) \tag{38}$$

$$\overline{k}_1 = f(x_k, y(x_k)) =: f \tag{39}$$

$$\overline{k}_2 = f\left(x_k + a_2 h, y(x_k) + a_2 h f\right) \tag{40}$$

$$\overline{k}_3 = f\left(x_k + a_3 h, y(x_k) + b_{31} h f + b_{32} h \overline{k}_2\right) \tag{41}$$

Taylorentwicklungen von $\overline{k}_2$ und $\overline{k}_3$ mit Entwicklungszentrum $(x_k, y(x_k)) =: (x, y)$:

$$\begin{aligned}
\overline{k}_2 = {} & f + f_x \cdot (a_2 h) + f_y \cdot (a_2 h f) && \text{erste partielle Ableitung je nach } x \text{ und } y \\
& + \frac{1}{2} f_{xx} \cdot (a_2 h)^2 && \text{zweite partielle Ableitung nach } x \\
& + f_{xy} \cdot (a_2 h)(a_2 h f) && \text{gemischte partielle Ableitungen nach } x \text{ und } y, \text{ wobei } f_{xy} = f_{yx} \\
& + \frac{1}{2} f_{yy} \cdot (a_2 h f)^2 && \text{zweite partielle Ableitung nach } y \\
& + O(h^3)
\end{aligned}$$

Mit (36) und (37) erhalten wir

$$\overline{k}_2 = f + a_2 F \cdot h + \frac{1}{2} a_2^2 G \cdot h^2 + O(h^3) \tag{42}$$

$$\begin{aligned}
\overline{k}_3 = {} & f + f_x \cdot (a_3 h) + f_y \cdot (b_{31} h f + b_{32} h \overline{k}_2) && \text{erste partielle Ableitung je nach } x \text{ und } y \\
& + \frac{1}{2} f_{xx} \cdot (a_3 h)^2 && \text{zweite partielle Ableitung nach } x \\
& + f_{xy} \cdot (a_3 h)(b_{31} h f + b_{32} h \overline{k}_2) && \text{gemischte partielle Ableitungen nach } x \text{ und } y, \text{ wobei } f_{xy} = f_{yx} \\
& + \frac{1}{2} f_{yy} \cdot (b_{31} h f + b_{32} h \overline{k}_2)^2 && \text{zweite partielle Ableitung nach } y \\
& + O(h^3)
\end{aligned}$$

Mit (42) erhalten wir (bis und mit dem Term in h^2)

$$\begin{aligned}
\overline{k}_3 = {} & f \\
& + \{a_3 f_x + (b_{31} + b_{32}) f f_y\} \cdot h \\
& + \left\{ a_2 b_{32} F f_y + \frac{1}{2} a_3^2 f_{xx} + a_3 (b_{31} + b_{32}) f f_{xy} + \frac{1}{2} (b_{31} + b_{32})^2 f^2 f_{yy} \right\} \cdot h^2 \\
& + O(h^3)
\end{aligned}$$

und mit (36), (37) schliesslich

$$\overline{k}_3 = f + a_3 F \cdot h + \left\{ a_2 b_{32} F f_y + \frac{1}{2} a_3^2 G \right\} \cdot h^2 + O(h^3) \tag{43}$$

In (43) wurde verwendet, dass $a_3 = b_{31} + b_{32}$ erfüllt ist.
Damit erhalten wir für

$$\begin{aligned} d_{k+1} &= y(x_{k+1}) - y(x_k) \\ &\quad -h \cdot \left\{ c_1 \overline{k}_1 + c_2 \overline{k}_2 + c_3 \overline{k}_3 \right\} \\ &= f \cdot h + \frac{1}{2!} F \cdot h^2 + \frac{1}{3!}(G + F \cdot f_y) \cdot h^3 \\ &\quad -h c_1 f \\ &\quad -h c_2 \left(f + a_2 F \cdot h + \frac{1}{2} a_2^2 G \cdot h^2 + O(h^3) \right) \\ &\quad -h c_3 \left(f + a_3 F \cdot h + \left\{ a_2 b_{32} F f_y + \frac{1}{2} a_3^2 G \right\} \cdot h^2 + O(h^3) \right) \end{aligned}$$

und geordnet nach Potenzen in h:

$$\begin{aligned} d_{k+1} &= (1 - c_1 - c_2 - c_3) f \cdot h \\ &\quad + \left(\frac{1}{2} - a_2 c_2 - a_3 c_3 \right) F \cdot h^2 \\ &\quad + \left(\frac{1}{6}(G + F \cdot f_y) - \frac{1}{2} a_2^2 c_2 G - a_2 b_{32} c_3 F f_y - \frac{1}{2} a_3^2 c_3 G \right) \cdot h^3 \\ &\quad + O(h^4) \end{aligned}$$

und schliesslich

$$\begin{aligned} d_{k+1} &= (1 - c_1 - c_2 - c_3) f \cdot h \\ &\quad + \left(\frac{1}{2} - a_2 c_2 - a_3 c_3 \right) F \cdot h^2 \\ &\quad + \left\{ \left[\frac{1}{6} - a_2 b_{32} c_3 \right] F f_y + \left[\frac{1}{6} - \frac{1}{2} a_2^2 c_2 - \frac{1}{2} a_3^2 c_3 \right] G \right\} \cdot h^3 \\ &\quad + O(h^4) \end{aligned}$$

□

3.2 Lineare Differentialgleichung mit Anregung

Als interessantes Beispiel im Zusammenhang mit der Stabilität und Genauigkeit dient uns das Beispiel

$$y' = -c\,(y - \cos(x)) \qquad c > 0 \qquad y(0) = 0 \tag{44}$$

mit der allgemeinen Lösung $y_a(x) = y_h(x) + y_p(x)$.

Dabei ist $y_h(x) = c_1\, e^{-cx}$ die allgemeine Lösung der homogenen Differentialgleichung

$$y' = -c\, y$$

und für $y_p(x)$ verwenden wir die Methode der *Variation der Konstanten*, also

Ansatz für

$$y_p(x) = c_1(x)\, e^{-cx} \tag{45}$$

(45) wird nun in (44) eingesetzt, was uns für

$$y_p(x) = \frac{c^2}{c^2+1}\cos(x) + \frac{c}{c^2+1}\sin(x)$$

liefert und schliesslich

$$y_a(x) = c_1\, e^{-cx} + \frac{c^2}{c^2+1}\cos(x) + \frac{c}{c^2+1}\sin(x)$$

Die Anfangsbedingung in (44) erlaubt c_1 zu bestimmen, was uns die spezielle Lösung von (44) liefert

$$y_(x) = -\frac{c^2}{c^2+1}\, e^{-cx} + \frac{c^2}{c^2+1}\cos(x) + \frac{c}{c^2+1}\sin(x) \tag{46}$$

In (46) ist der mittlere Term

$$\frac{c^2}{c^2+1}\cos(x)$$

für grosse Werte von $c > 0$ dominant, insbesondere für $c = 50$.

Der erste Term geht mit $x > 0$ sehr schnell gegen Null und der Koeffizient des dritten Terms ist c mal kleiner als derjenige des zweiten Tems.

3.3 Lösung einer separierbaren Differentialgleichung

Als Beispiel für diese Methode dient uns

$$y' = \frac{dy}{dx} = -2xy^2 \qquad \text{mit} \qquad y(0) = 1 \tag{47}$$

Wie der Name im Titel sagt, werden die Variablen getrennt. D.h. (47) wird algebraisch so umgeformt, dass alles was „y“ bzw. alles was „x“ heisst, je auf einer Seite der Gleichung steht, nämlich:

$$\frac{1}{y^2}\, dy = -2x\, dx$$

Nun folgt eine unbestimmte Integration, links über y bzw. rechts über x. Bei jeder unbestimmten Integration gibt es zusätzlich eine Integrationskonstante, die z.B. auf der rechten Seite berücksichtigt werden kann:

$$-\frac{1}{y} = -x^2 + C$$

Auflösen nach y liefert dei allgemeine Lösung von (47)

$$y(x) = \frac{1}{x^2 + C}$$

(dabei wurde $-C$ wieder durch C ersetzt) und mit der Anfangsbedingung erhalten wir schliesslich die spezielle Lösung

$$y(x) = \frac{1}{x^2+1} \tag{48}$$

von (47).

(47) mit (48) wird als Testbeispiel für unsere numerischen Verfahren verwendet.

Aufgabe 3.1

Versuchen Sie mit derselben Methode das Problem

$$y' = y \qquad \text{mit} \quad y(0) = c_0$$

zu lösen.
Lösung: $y(x) = c_0\, e^x$

3.4 Lösung einer Differentialgleichung mit Hilfe einer Potenzreihe

Als Beispiel für diese Methode dient uns

$$y'' + 2y' = 0 \qquad \text{mit} \qquad y(0) = 2 \quad y'(0) = 1 \tag{49}$$

Wie der Name im Titel besagt, wird als *Ansatz* für die Lösung eine Potenzreihe

$$y(x) = \sum_{k=0}^{\infty} a_k x^k \tag{50}$$

verwendet und in (49) eingesetzt. Die Koeffizienten von (50) werden mit Koeffizientenvergleich bestimmt. Mit

$$y'(x) = \sum_{k=1}^{\infty} k a_k x^{k-1} \qquad \text{und} \qquad y''(x) = \sum_{k=2}^{\infty} k(k-1) a_k x^{k-2}$$

erhalten wir

$$\begin{array}{lll}
x^0: & 2 \cdot 1 \cdot a_2 + 2 \cdot 1 \cdot a_1 & = 0 \\
x^1: & 3 \cdot 2 \cdot a_3 + 2 \cdot 2 \cdot a_2 & = 0 \\
x^2: & 4 \cdot 3 \cdot a_4 + 2 \cdot 3 \cdot a_3 & = 0 \\
x^3: & 5 \cdot 4 \cdot a_5 + 2 \cdot 4 \cdot a_4 & = 0 \\
\ldots: & \ldots \quad \ldots & \ldots \\
x^k: & (k+2) \cdot (k+1) \cdot a_{k+2} + 2 \cdot (k+1) \cdot a_{k+1} & = 0 \\
\ldots: & \ldots \quad \ldots & \ldots
\end{array}$$

ein *rekusives* Gleichungssystem für die Unbekannten a_k (∞ viele Gleichungen für ∞ viele Unbekannte).

Mit den Anfangsbedingungen erhalten wir $a_0 = 2$ und $a_1 = 1$ was erlaubt, die Lösung rekursiv zu bestimmen: $a_2 = -1$, $a_3 = \frac{2}{3}$, $a_4 = -\frac{1}{3}$ ...

Die die Rekursion lautet

$$a_{k+1} = -\frac{2a_k}{k+1} \qquad k = 1, 2 \ldots$$

Mit volständiger Induktion kann bewiesen werden, dass

$$a_k = -\frac{1}{2} \frac{(-2)^k}{k!} \qquad k = 1, 2 \ldots$$

gültig ist. Damit bekommen wir

$$y(x) = \frac{5}{2} - \frac{1}{2} e^{-2x}$$

als Lösung von (49).

Aufgabe 3.2

Versuchen Sie mit derselben Methode das Problem

$$y'' + y = 0 \qquad \text{mit} \quad y(0) = y'(0) = 1$$

zu lösen.
Lösung: $a_{k+2} = -\frac{a_k}{(k+2)(k+1)}$ und mit vollständiger Induktion: $a_{2k} = (-1)^k \frac{1}{(2k)!}$ und $a_{2k-1} = (-1)^{k+1} \frac{1}{(2k-1)!}$ und somit $y(x) = \cos(x) + \sin(x)$

3.5 Allgemeine Bedingungen für ein ERK mit $p = 4$

Literatur: Hairer, Nørsett und Wanner, Band I, p 137/138.

Taylorentwicklungen für den lokalen Fehler, analog wie für eine Methode der Stufe drei, cf. Abschnitt 2.5.
zu lösende Gleichungen:

$$c_1 + c_2 + c_3 + c_4 = 1 \tag{51}$$
$$c_2 a_2 + c_3 a_3 + c_4 a_4 = \frac{1}{2} \tag{52}$$
$$c_2 a_2^2 + c_3 a_3^2 + c_4 a_2 = \frac{1}{3} \tag{53}$$
$$c_2 a_2^3 + c_3 a_3^3 + c_4 a_3 = \frac{1}{4} \tag{54}$$

$$c_3 a_3 b_{32} a_2 + c_4 a_4 \left(b_{42} a_2 + b_{43} a_3\right) = \frac{1}{8} \tag{55}$$
$$c_3 b_{32} + c_4 b_{42} = c_2 \left(1 - a_2\right) \tag{56}$$
$$c_4 b_{43} = c_3 \left(1 - a_3\right) \tag{57}$$
$$0 = c_4 \left(1 - a_4\right) \tag{58}$$

$$c_3 c_4 a_2 \left(1 - a_3\right) \neq 0 \tag{59}$$

Aus (59) folgt mit (58) sofort, dass

$$a_4 = 1.$$

Die Bedingungen (51) – (54) inkl. (58) beinhalten nichts anderes, als dass die c_k die Gewichte und die a_k die Stützstellen einer Quadraturformel $4-$ ter Ordnung auf dem Standard–Intervall $[0, 1]$ sind, wobei speziell $a_1 = 0$ und $a_4 = 1$.

Zudem *muss* das Testproblem

$$y'(x) = 1 \tag{60}$$
$$y(0) = 0 \tag{61}$$

exakt integriert werden können. Mit (60) und (61) erhalten wir die folgenden Nebenbedingungen:

$$\begin{cases} a_2 = b_{21} \\ a_3 = b_{31} + b_{32} \\ a_4 = b_{41} + b_{42} + b_{43} \end{cases} \tag{62}$$

3.5.1 Zur Lösung dieser Gleichungen gibt es vier typische Fälle:

Fall 1)

$a_2 = u$ und $a_3 = v$, wobei $0, u, v, 1$ alle voneinander verschieden.

In diesem Fall bilden die Gleichungen (51) – (54) ein reguläres lineares Gleichungssystem für die Gewichte c_k:

$$c_1 = \frac{1 - 2(u+v) + 6uv}{12uv} \qquad c_2 = \frac{2v - 1}{12u(1-u)(v-u)} \tag{63}$$
$$c_3 = \frac{1 - 2u}{12v(1-v)(v-u)} \qquad c_4 = \frac{3 - 4(u+v) + 6uv}{12u(1-u)(v-u)} \tag{64}$$

Wegen (59) müssen u und v so gewählt werden, dass $c_3 \neq 0$ und $c_4 \neq 0$.

Die drei anderen Fälle mit einem Doppelknoten bauen auf der Regel von *Simpson* auf.

Fall 2)

$a_3 = 0$ und $a_2 = \frac{1}{2}$: $c_3 = w \neq 0$, $c_1 = \frac{1}{6} - w$, $c_2 = \frac{4}{6}$ und $c_4 = \frac{1}{6}$.

Fall 3)

$a_2 = a_3 = \frac{1}{2}$: $c_1 = \frac{1}{6}$, $c_3 = w \neq 0$, $c_2 = \frac{4}{6} - w$ und $c_4 = \frac{1}{6}$.

Fall 4)

$a_2 = 1$, $a_3 = \frac{1}{2}$: $c_4 = w \neq 0$, $c_2 = \frac{1}{6} - w$, $c_1 = \frac{1}{6}$ und $c_3 = \frac{4}{6}$.

Sind die a_k und c_k einmal, bestimmt, so folgt aus (57) $b_{43} = \ldots$

Anschliessend erhalten wir aus (55) und (56) die Grössen b_{32} und b_{42}. (55) und (56) bilden ein reguläres lineares Gleichungssystem für b_{32} und b_{42}, denn

$$\begin{vmatrix} c_3 & c_4 \\ c_3a_3a_2 & c_4a_4a_2 \end{vmatrix} = c_3c_4a_2(a_4 - a_3) \neq 0 \qquad \text{mit (59)}$$

Schliesslich erhalten wir b_{21}, sowie b_{31} und b_{41} mit (62).

3.5.2 Zwei spezielle Verfahren

Aus dem Fall 3) mit $w = \frac{2}{6}$:

(65)

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$

Aus dem Fall 1) mit $u = \frac{1}{3}$ und $v = \frac{2}{3}$:

(66)

0				
$\frac{1}{3}$	$\frac{1}{3}$			
$\frac{2}{3}$	$-\frac{1}{3}$	1		
1	1	-1	1	
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

(65) ist *das* Runge–Kutta Verfahren $4-$ ter Ordnung und (66) ist etwas genauer, aber weniger populär.

3.6 Anwendung der Jacobi–Matrix

3.6.1 Methode von Newton für eine skalare Gleichung

Zu lösen ist die nicht–lineare Gleichung

$$f(x) = 0.$$

Bei dieser Methode wird das gegebene Problem in jedem Schritt lokal linearisiert. Die Tangente (= lineare Funktion) an die Kurve wird mit der $x-$ Achse geschnitten.

erster Schritt

- $x_0 =$ Starwert. $f(x_0) = y_0$
- Tangente t_0 durch den Punkt $P_0(x_0, y_0)$ an die Kurve $y = f(x)$

$$t_0 \colon f'(x_0)(x - x_0) = y - y_0 \qquad \text{Punkt–Richtungsform der Geradengleichung}$$

- Schnittpunkt x_1 von t_0 mit der $x-$ Achse ($y = 0$)

$$f'(x_0)(x - x_0) = -y_0 \quad \Longrightarrow (x - x_0) =: \Delta x = -(f'(x_0))^{-1}y_0$$

und damit

$$x_1 = x_0 + \Delta x = x_0 - (f'(x_0))^{-1}y_0$$

zweiter Schritt

- $f(x_1) = y_1$

- Tangente t_1 durch den Punkt $P_1(x_1, y_1)$ an die Kurve $y = f(x)$

$$t_1 \colon f'(x_1)(x - x_1) = y - y_1 \qquad \text{Punkt–Richtungsform der Geradengleichung}$$

- Schnittpunkt x_2 von t_1 mit der $x-$ Achse ($y = 0$)

$$f'(x_1)(x - x_1) = -y_1 \quad \Longrightarrow (x - x_1) =: \Delta x = -(f'(x_1))^{-1} y_1$$

und damit

$$x_2 = x_1 + \Delta x = x_1 - (f'(x_1))^{-1} y_1$$

...

$(k+1)-$ **ter Schritt**

- $f(x_k) = y_k$
- Tangente t_k durch den Punkt $P_k(x_k, y_k)$ an die Kurve $y = f(x)$

$$t_k \colon f'(x_k)(x - x_k) = y - y_k \qquad \text{Punkt–Richtungsform der Geradengleichung}$$

- Schnittpunkt x_{k+1} von t_k mit der $x-$ Achse ($y = 0$)

$$f'(x_k)(x - x_k) = -y_k \quad \Longrightarrow (x - x_k) =: \Delta x = -(f'(x_k))^{-1} y_k$$

und damit

$$x_{k+1} = x_k + \Delta x = x_k - (f'(x_k))^{-1} y_k$$

Allgemein gilt:

- Der $(k+1)-$ te Schritt ist durchführbar, falls $f'(x_k) \neq 0$.
- Die Folge der x_k, $k = 0, 1, 2, \ldots$ ist konvergent, falls $|\Delta x| \longrightarrow 0$ für $k \longrightarrow \infty$.
- Falls wir einen guten Startwert wählen, konvergiert das Verfahren sehr gut. (quadratisch).

3.6.2 Methode von Newton für nicht–lineare Gleichungssysteme

Zu lösen ist das nicht–lineare Gleichungssystem

$$f(x) = 0 \qquad f \in \mathbb{R}^n \qquad x \in \mathbb{R}^n$$

d.h. n nicht–lineare Gleichungen in n Unbekannten.

Beispiel 3.1 *Sei speziell* $n = 2$

$$\left\{ \begin{array}{lcl} f_1(x_1, x_2) & = & x_1^2 + x_2^2 + 0.6x_2 - 0.16 \\ f_2(x_1, x_2) & = & x_1^2 - x_2^2 + x_1 - 1.6x_2 - 0.14 \end{array} \right. \qquad f(x) = 0 = \left(\begin{array}{c} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{array} \right) = \left(\begin{array}{c} 0 \\ 0 \end{array} \right)$$

Das zu lösende Problem soll eindeutig lösbar sein. Hier werden die Starwerte $x_1^{(0)} = 0.6$ und $x_2^{(0)} = 0.25$ gegeben, also Startwertvektor

$$x^{(0)} = \left(\begin{array}{c} x_1^{(0)} \\ x_2^{(0)} \end{array} \right) = \left(\begin{array}{c} 0.6 \\ 0.25 \end{array} \right)$$

Jede Iterierte ist jetzt im Gegensatz zu obigem Abschnitt ein Vektor $x \in \mathbb{R}^2$.

Lokale Linearisierung: der Ableitung im obigen Abschnitt entspricht hier die Jacobi–Matrix von f

$$J(x) = \frac{\partial f}{\partial x} = \left(\frac{\partial f_i}{\partial x_j} \right) \quad 1 \leq i, j \leq n$$

$i =$ Zeilen- und $j =$ Spaltenindex. Ansonsten ist das Prozedre völlig analog.
Für das Beispiel

$$J(x) = \left(\begin{array}{cc} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{array} \right) = \left(\begin{array}{cc} 2x_1 & 2x_2 + 0.6 \\ 2x_1 + 1 & -2x_2 - 1.6 \end{array} \right)$$

erster Schritt

- $x^{(0)} =$ Starwert. $f(x^{(0)}) = y^{(0)}$

- "Tangente" E_0 durch den Punkt $P_0(x^{(0)}, y^{(0)})$ an die Fläche $y = f(x)$
$$E_0 \colon J(x^{(0)})(x - x^{(0)}) = y - y^{(0)}$$
- Schnittpunkt $x^{(1)}$ von E_0 mit der x_1x_2- Ebene $(y = 0)$
$$J(x^{(0)})(x - x^{(0)}) = -y^{(0)} \quad \Longrightarrow (x - x^{(0)}) =: \Delta x = -\left(J(x^{(0)})\right)^{-1} y^{(0)}$$
und damit
$$x^{(1)} = x^{(0)} + \Delta x = x^{(0)} - \left(J(x^{(0)})\right)^{-1} y^{(0)}$$

Zur Bestimmung von Δx muss ein lineares Gleichungssystem mit der Systemmatrix $J(x^{(0)})$ gelöst werden.

zweiter Schritt
- $f(x^{(1)}) = y^{(1)}$
- "Tangente" E_1 durch den Punkt $P_1(x^{(1)}, y^{(1)})$ an die Fläche $y = f(x)$
$$E_1 \colon J(x^{(1)})(x - x^{(1)}) = y - y^{(1)}$$
- Schnittpunkt $x^{(2)}$ von E_1 mit der x_1x_2- Ebene $(y = 0)$
$$J(x^{(1)})(x - x^{(1)}) = -y^{(1)} \quad \Longrightarrow (x - x^{(1)}) =: \Delta x = -\left(J(x^{(1)})\right)^{-1} y^{(1)}$$
und damit
$$x^{(2)} = x^{(1)} + \Delta x = x^{(1)} - \left(J(x^{(1)})\right)^{-1} y^{(1)}$$

Zur Bestimmung von Δx muss ein lineares Gleichungssystem mit der Systemmatrix $J(x^{(1)})$ gelöst werden.

...

$(k+1)-$ **ter Schritt**
- $f(x^{(k)}) = y^{(k)}$
- "Tangente" E_k durch den Punkt $P_k(x^{(k)}, y^{(k)})$ an die Fläche $y = f(x)$
$$E_k \colon J(x^{(k)})(x - x^{(k)}) = y - y^{(k)}$$
- Schnittpunkt $x^{(k+1)}$ von E_k mit der x_1x_2- Ebene $(y = 0)$
$$J(x^{(k)})(x - x^{(k)}) = -y^{(k)} \quad \Longrightarrow (x - x^{(k)}) =: \Delta x = -\left(J(x^{(k)})\right)^{-1} y^{(k)}$$
und damit
$$x^{(k+1)} = x^{(k)} + \Delta x = x^{(k)} - \left(J(x^{(k)})\right)^{-1} y^{(k)}$$

Allgemein gilt:
- Zur Bestimmung von Δx muss ein lineares Gleichungssystem mit der Systemmatrix $J(x^{(k)})$ gelöst werden.
- Der $(k+1)-$ te Schritt ist durchführbar, falls $J(x^{(k)})$ regulär.
- Die Folge der $x^{(k)}$, $k = 0, 1, 2, \ldots$ ist konvergent, falls $\|\Delta x\| \longrightarrow 0$ für $k \longrightarrow \infty$.
- Falls wir einen guten Startvektor wählen, konvergiert das Verfahren sehr gut. (quadratisch).
- Das vorgestellte Verfahren ist die Methode von *Newton–Raphson*.

Geometrische Interpretation dieses Beispiels:
Die erste Kurve $f_1(x_1, x_2) = 0$ beschreibt eine Kreislinie und die zweite Kurve $f_2(x_1, x_2) = 0$ beschreibt eine Hyperbel, d.h. hier wird ein Schnittpunkt dieser beiden Kurven bestimmt.

Aufgabe 3.3

Führen Sie mit Hilfe von Matlab obiges Beispiel numerisch durch.
Zur Kontrolle: nach 5 Iterationen werden Sie
$$x^{(5)} = \begin{pmatrix} 0.2718845063 \\ 0.1196433776 \end{pmatrix}$$
erhalten.

Aufgabe 3.4 *Sei wieder* $n = 2$

Zu lösen ist das folgende nicht–lineare Gleichungssystem:
$$\begin{cases} f_1(x_1, x_2) &= e^{x_1x_2} + x_1^2 + x_2 - 1.2 \\ f_2(x_1, x_2) &= x_1^2 + x_2^2 + x_1 - 0.55 \end{cases} \qquad f(x) = 0 \qquad \text{mit dem Startvektor } x^{(0)} = \begin{pmatrix} 0.4 \\ 0.25 \end{pmatrix}$$

Literatur

[1] H. R. Schwarz, N. Köckler, *Numerische Mathematik*, Teubner Stuttgart, 2004.

[2] W. Gander, *Computer Mathematik*, Birkhäuser Basel, 1985.

[3] P. Henrici, Vorlesungsnotizen zu *„Numerische Methoden“*, Seminar für Angewandte Mathematik, ETH Zürich, Studienjahr 1977-1978.

[4] P. Henrici, *Elemente der numerischen Analysis*, Band 1, Hochschultaschenbücher Band 551, B.I. Mannheim 1972.

[5] R. Jeltsch, *Numerische Mathematik*, Skript zur Vorlesung, ???