

## 3 Numerische Integration gewöhnlicher Differentialgleichungen

Entwicklungen in der Elektrotechnik sind heute ohne Computer-Unterstützung nicht mehr denkbar. Die Entwicklung und Simulation basiert dabei auf Modellen, die meist mit Hilfe von Differentialgleichungen beschrieben werden. Das aus diesen Differentialgleichungen abzuleitende (Zeit-)Verhalten des Systems kann durch - mit der Zeit - schrittweise fortschreitende Berechnung der Lösung unter den gegebenen Anfangsbedingungen und (Zeit-) Anregungsfunktionen (Störfunktionen) ermittelt werden.

Die Verfügbarkeit von leistungsfähigen Digitalrechnern insbesondere des PC's hat die Computerunterstützte Entwicklung (CAD) in der Schaltungstechnik zu einem unverzichtbaren Werkzeug gemacht, wobei dem Anwender für die unterschiedlichsten Probleme vielfältige Entwicklungswerkzeuge (Tools) zur Verfügung stehen.

Zur Einführung werden in Kap 3.1 und 3.2 kurz numerische Verfahren zur Differentiation und Integration vorgestellt. Danach werden Verfahren zur numerischen Integration gewöhnlicher Differentialgleichungen behandelt, die bei der Lösung von Anfangswertaufgaben eingesetzt werden. Einfache Beispiele zu Simulationsaufgaben aus der Schaltungstechnik und der Regeltechnik dienen der Veranschaulichung des Stoffes. Zur Vertiefung werde auf die verwendete Literatur [8] bis [14] verwiesen.

Im folgenden werde die unabhängige Variable mit  $x$  und die abhängige mit  $y$  oder  $y(x)$  oder  $f(x)$  bezeichnet. Handelt es sich um Zeitvorgänge, so verwenden wir  $t$  als unabhängige Variable und  $x(t)$ ,  $y(t)$ ,  $f(t)$  oder ähnlich als abhängige Variable. Um den Sachverhalt allgemein zu formulieren, werden wir oft zunächst als unabhängige Variable  $x$  verwenden, wobei dann darunter z.B. der Ort aber auch die Zeit oder eine andere Variable verstanden werden kann.

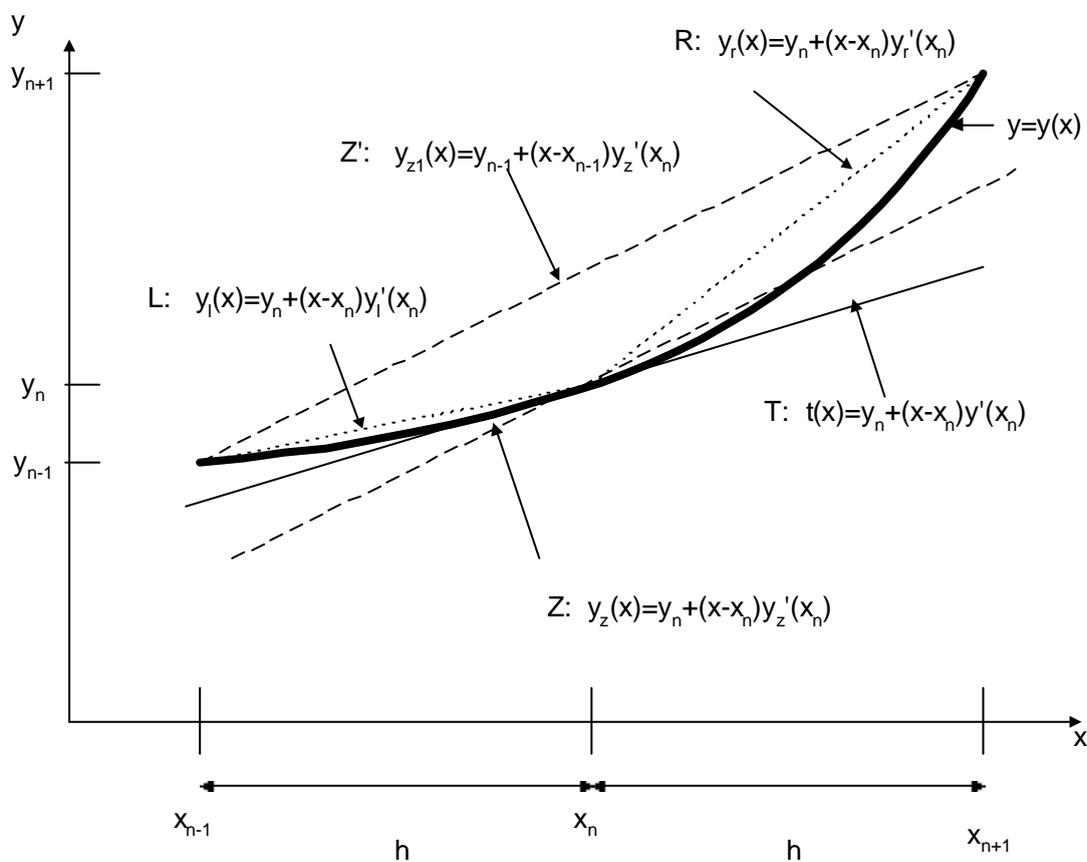
### 3.1 Numerische Differentiation

Es sei eine Funktion oder eine Abhängigkeit gegeben durch

- Wertetabelle  $(x_n, f(x_n)) ; (x_n, y_n); \quad n = 0, 1, \dots, N$
- Meßkurve oder online-Messwerte  $y = f(x), y = y(x)$

Günstig: äquidistante Schrittweite  $h$ , also  $x_{n+1} = x_n + h$  oder  $x_n = x_0 + nh$

In der Abbildung sind im Punkt  $(x_n, y_n)$  die Tangente  $T$  an die Kurve  $y = y(x)$  und drei Geraden  $R$ ,  $L$  und  $Z$  eingezeichnet, die mit den Steigungen des rechten, zentralen und linken Differenzenquotienten die Tangente approximieren. Man erkennt, daß die Fehler, die dabei auftreten von der Größe der Schrittweite ( der Verfeinerung bei der Diskretisierung) abhängen und daß der zentrale Differenzenquotient genauer als der linke oder rechte Differenzenquotient ist.



**rechtsseitiger (vorwärts) Differenzenquotient:**

$$y_r'(x_n) = \frac{y(x_n + h) - y(x_n)}{h} = \frac{y_{n+1} - y_n}{h}$$

**linksseitiger (rückwärts) Differenzenquotient:**

$$y_l'(x_n) = \frac{y(x_n) - y(x_n - h)}{h} = \frac{y_n - y_{n-1}}{h}$$

**zentraler (vorwärts und rückwärts) Differenzenquotient:**

$$y_z'(x_n) = \frac{y(x_n + h) - y(x_n - h)}{2h} = \frac{y_{n+1} - y_{n-1}}{2h}$$

**Fehlerabschätzung:**

Vorausgesetzt, die Funktion  $y = y(x)$  ist im Intervall  $I_n = [x_n, x_{n+1}]$  hinreichend oft stetig differenzierbar, dann können wir sie in eine Taylorreihe um den Punkt  $x_n$  mit einem Restglied entwickeln :

$$y(x_n + h) = y(x_n) + hy'(x_n) + \frac{1}{2!}h^2 y''(\mathbf{x})$$

$$R = \frac{1}{2!}h^2 y''(\mathbf{x})$$

$$x_n < \mathbf{x} < x_{n+1}$$

Daraus folgt :

$$y'(x_n) = \frac{y(x_n + h) - y(x_n)}{h} - \frac{1}{2}h \cdot y''(\mathbf{x}) = \frac{y_{n+1} - y_n}{h} + O(h)$$

Der Fehler bei Ersetzen der Ableitung  $y'(x_n)$  durch den rechtsseitigen Differenzenquotienten läßt sich aus der obigen Formel nicht berechnen, da nicht bekannt ist, an welcher Stelle die zweite Ableitung von  $y$  zu berechnen ist (Mittelwertsatz). Man kann jedoch eine **obere Grenze für den Fehler** angeben, indem man eine obere Schranke für die zweite Ableitung einsetzt. Außerdem sieht man, daß dann die Abhängigkeit des Korrekturterms linear von der Schrittweite  $h$  abhängt. Man sagt deshalb, der **Fehler ist von der Ordnung  $h$  und schreibt  $O(h)$ . Bei Halbierung der Schrittweite geht somit der Fehler auf die Hälfte zurück.**

Analog verhält es sich beim **linksseitigen Differenzenquotienten**

Zur Fehlerabschätzung beim **zentralen Differenzenquotienten** muß man  $y = y(x)$  im Punkt  $x_n$  bis zur dritten Ableitung entwickeln. Man erhält zwei Restglieder, die zu einem Fehlerterm zusammengefaßt werden. Man findet dann, daß der **Fehler von der Ordnung  $h^2$  ist, was bedeutet, daß der Fehler bei Halbierung der Schrittweite auf den vierten Teil zurück geht.**

$$y(x_n + h) = y(x_n) + h \cdot y'(x_n) + \frac{1}{2!}h^2 y''(x_n) + \frac{1}{3!}h^3 y'''(\mathbf{x}_r)$$

$$y(x_n - h) = y(x_n) - h \cdot y'(x_n) + \frac{1}{2!}h^2 y''(x_n) - \frac{1}{3!}h^3 y'''(\mathbf{x}_l)$$

$$y(x_n + h) - y(x_n - h) = 2h \cdot y'(x_n) + \frac{1}{3!}h^3 (y'''(\mathbf{x}_r) + y'''(\mathbf{x}_l))$$

$$x_n < \mathbf{x}_r < x_{n+1}$$

$$x_{n-1} < \mathbf{x}_l < x_n$$

$$y'(x_n) = \frac{y_{n+1} - y_{n-1}}{2h} - \frac{1}{12}h^2 (y'''(\mathbf{x}_r) + y'''(\mathbf{x}_l)) = \frac{y_{n+1} - y_{n-1}}{2h} - O(h^2)$$

Nach derselben Methode lassen sich Formeln für die numerische Berechnung der **2. Ableitung** einer Funktion aufstellen:

$$y(x_n + h) = y(x_n) + h \cdot y'(x_n) + \frac{1}{2!}h^2 y''(x_n) + \frac{1}{3!}h^3 y'''(x_n) + \frac{1}{4!}h^4 y^{(4)}(\mathbf{x}_r)$$

$$y(x_n - h) = y(x_n) - h \cdot y'(x_n) + \frac{1}{2!}h^2 y''(x_n) - \frac{1}{3!}h^3 y'''(x_n) + \frac{1}{4!}h^4 y^{(4)}(\mathbf{x}_l)$$

$$y(x_n + h) + y(x_n - h) = 2y(x_n) + h^2 y''(x_n) + \frac{1}{4!}h^4 (y^{(4)}(\mathbf{x}_r) + y^{(4)}(\mathbf{x}_l))$$

Daraus folgt:

$$y''(x_n) = \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} + O(h^2) \quad (\text{zentraler Differenzenquotient für 2. Ableitung})$$

$$y''(x_n) = \frac{y_{n+2} - 2y_{n+1} + y_n}{h^2} + O(h) \quad (\text{rechter Differenzenquotient für 2. Ableitung})$$

$$y''(x_n) = \frac{y_n - 2y_{n-1} + y_{n-2}}{h^2} + O(h) \quad (\text{linker Differenzenquotient für 2. Ableitung})$$

**Beispiel:**  $y(x) = x^3 + 1$

Da hier  $y'$  und  $y''$  analytisch berechnet werden können, können wir den numerisch errechneten Näherungswert für die Ableitung mit dem exakten Wert vergleichen. Wir berechnen die Werte für eine Schrittweite von  $h=1$  für  $x_n = -3$  bis  $+3$  und halbieren dann die Schrittweite.

$y'(x) = 3x^2, y''(x) = 6x$

h	$x_n$	$y(x_n)$	$y'(x_n)$	$y'_z(x_n)$	$y''(x_n)$	$y''_z(x_n)$
-----						
1	-3	-26	27		-18	
	-2	-7	12	13	-12	-12
	-1	0	3	4	-6	-6
	0	1	0	1	0	0
	1	2	3	4	6	6
	2	9	12	13	12	12
	3	28	27		18	
-----						
0.5	-3	-26	-27		-18	
	-2.5	-14.625	18.75	19	-15	-15
	-2	-7	12	12.25	-12	-12
	-1.5	-2.375	6.75	7	-9	-9
	-1	0	3	3.25	-6	-6
	-0.5	0.875	0.75	1	-3	-3
	0	1	0	.25	0	0
-----						
	2.5	16.625	18.75	19	15	15
	3	28	27		18	

**Randpunkte:**

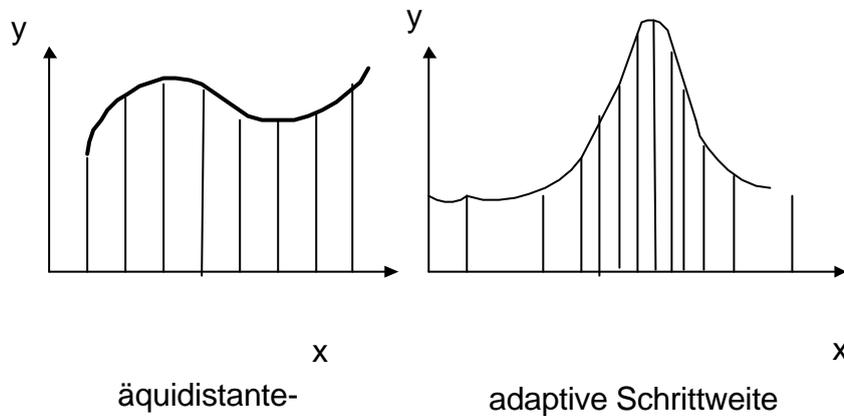
h = 1:	$y'(-3) = 27$ $y'(3) = 27$	$y'_r(-3) = 19$ $y'_{l1}(3) = 19$	$y''(-3) = -18$ $y''_r(-3) = -12$ $y''(+3) = +18$ $y''_{l1}(+3) = 12$
h = 0.5:	$y'(-3) = 27$ $y'(3) = 27$	$y'_r(-3) = 22.75$ $y'_{l1}(3) = 22.75$	$y''(-3) = -18$ $y''_r(-3) = -15$ $y''(-3) = +18$ $y''_{l1}(+3) = 15$

**Bem.:**

- Die numerischen Werte für die 2. Ableitung sind exakt, da  $y''(x) = 6$ .
- Bei Halbierung der Schrittweite geht der Fehler in der 1. Ableitung auf ein viertel zurück.

### 3.2 Numerische *Integration*

- bei einigen Verfahren wird für die Integration  $y(x)$  durch ein Interpolations-Polynom  $n$ -ter Ordnung  $P_n(x)$  ersetzt
- numerische Integration hat glättende Wirkung, Verfahren sind stabil
- wir erwähnen drei wichtige Verfahren :  
Sehnen-Trapez-Regel  
Simpson'sche Regel  
Romberg Verfahren,  
von denen wir die ersten beiden kurz besprechen.
- man unterscheidet Verfahren mit äquidistanter Schrittweite, solche mit adaptiver Schrittweite und solche für periodische Funktionen.



### 3.2.1 Sehnen Trapez- Verfahren:

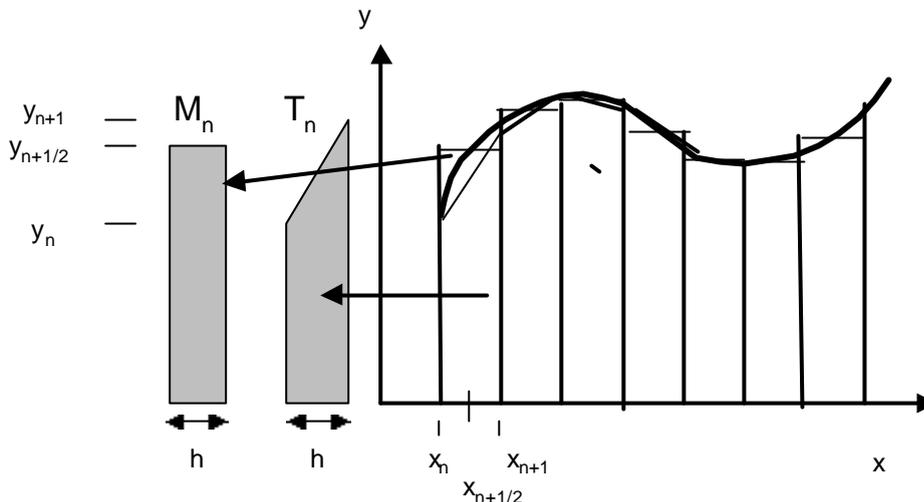
Das Integrationsintervall  $I = [a,b]$  wird in  $N$  äquidistante Abschnitte zerlegt.

$$a = x_0 < x_1 < x_2 < \dots < x_N = b$$

Schrittweite:  $h = (b-a)/N$

$$x_n = a + n \cdot h \quad n = 0, 1, \dots, N$$

$$y_n = y(x_n)$$



Trapezfläche:

$$T_n = \left[ y_n + \frac{1}{2}(y_{n+1} - y_n) \right] \cdot h = \frac{1}{2}(y_n + y_{n+1}) \cdot h$$

Trapeznäherung für das Integral:

$$T = \sum_{n=0}^{N-1} T_n = h \cdot \left( \frac{1}{2} y_0 + \frac{1}{2} y_1 + \frac{1}{2} y_1 + \frac{1}{2} y_2 + \dots + \frac{1}{2} y_{N-1} + \frac{1}{2} y_N \right)$$

$$T = h \cdot \left( \frac{1}{2} y_0 + y_1 + \dots + y_{N-1} + \frac{1}{2} y_N \right)$$

Den Fehler erhält man wieder durch Taylorreihenentwicklung:

$$I = \int_a^b y(x) dx$$

$$|T - I| < F_T, \quad F_T = \frac{b-a}{12} h^2 M, \quad \text{und } M \text{ so, daß } |y''(x)| < M \text{ für alle } x \in [a, b]$$

Mittelpunktssumme:

$$x_{n+1/2} = a + (n + 1/2)h$$

$$M = h \sum_{n=0}^{N-1} y(x_{n+1/2})$$

Nimmt man die Punkte  $x_n$  und die Mittelpunkte  $x_{n+1/2}$  zusammen, so ergibt sich eine Verfeinerung der Zerlegung des Intervalls  $[a,b]$  um das Doppelte oder eine Halbierung der Schrittweite. Eine

Betrachtung der Trapezsumme für die halbierte Schrittweite ergibt, daß sich diese aus der Trapezsumme und der Mittelpunktssumme für die ursprüngliche Schrittweite berechnen läßt.

$$T\left(\frac{h}{2}\right) = \frac{h}{2} \left[ \frac{1}{2} y(x_0) + y(x_{1/2}) + \sum_{n=1}^{N-1} [y(x_n) + y(x_{n+1/2})] + \frac{1}{2} y(x_N) \right]$$

$$\Rightarrow T\left(\frac{h}{2}\right) = [T(h) + M(h)] \quad (*)$$

Außerdem läßt sich zeigen , daß

$$|T - I| \leq |T - M|$$

ist.

Zur Berechnung des Näherungswertes für das Integral I wird zunächst  $h = b-a$  gesetzt und dann T und M berechnet. Dann wird die Schrittweite h halbiert und erneut T und M berechnet, wobei schon bekannte Funktionswerte aufgrund obiger Formel (\*) auf ökonomische Weise wieder verwertet werden. Dieses Verfahren wird solange fortgesetzt, bis der Fehler kleiner als eine vorgegebene Grenze  $\varepsilon$  ist. Falls die Abbruchbedingung nicht erreicht wird, wird die Rechnung nach 10 Durchläufen gestoppt , da sich sonst eine Dauerschleife ergäbe.

**Algorithmus:**

```

h ← b-a;      n ← 1;      T ← h*(y(a)+y(b))/2
für          k = 1,2,...,10;
M=0
für n=0,1,...,N-1:
    M ← M*y(a+(n+0.5)*h)
M ← h*M;  T ← (T+M)/2;  h ← h/2;  N ← 2*N;
falls |T-M| < ε : Stop
    
```

### 3.2.2 Integrationsformel nach Simpson:

#### Übung:

Die Punkte  $(x_n, f(x_n))$  werden durch eine Approximationsfunktion interpoliert. Dabei wird für jede zweite Stützstelle ein Polynom zweiten Grades ermittelt, das durch die Punkte  $(x_{n-1}, f(x_{n-1}))$ ,  $(x_n, f(x_n))$  und  $(x_{n+1}, f(x_{n+1}))$  verläuft. Das Integral wird durch die Fläche unter der so erzeugten Approximationsfunktion angenähert.

- a) Bestimmen Sie das Interpolationspolynom durch die ersten drei Stützwerte an den Stellen  $x_0, x_1 = x_0+h$  und  $x_2 = x_0+2h$ .

**Hilfe:** Machen Sie für das Interpolationspolynom den Ansatz

$$P_2(x) = a_2(x - x_1)^2 + a_1(x - x_1) + a_0$$

und bestimmen Sie aus den Bedingungsgleichungen

$$P_2(x_0) = f(x_0)$$

$$P_2(x_0 + h) = f(x_0 + h)$$

$$P_2(x_0 + 2h) = f(x_0 + 2h)$$

die Koeffizienten  $a_2, a_1, a_0$ .

**Ergebnis:**

$$a_2 = \frac{1}{2h^2} [f(x_0 + 2h) - 2f(x_0 + h) + f(x_0)]$$

$$a_1 = \frac{1}{2h} [f(x_0 + 2h) - f(x_0)]$$

$$a_0 = f(x_0 + h)$$

- b) Bestimmen Sie die Fläche unter der Approximationsfunktion  $P_2(x)$  für den Streifen zwischen  $x_0$  und  $x_2 = x_0+2h$

$$A_{\text{Streifen}} = \int_{x_0}^{x_2} P_2(x) dx$$

**Ergebnis:**

$$A_{\text{Streifen}} = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)]$$

Daraus folgt für  $N$  gerade die Simpson'sche Formel

$$S = \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \dots + 4f(x_{N-1}) + f(x_N)]$$

Der Verfahrensfehler, den man durch Taylorentwicklung berechnen kann, ergibt sich zu :

$$F_S = -\frac{b-a}{180} h^4 f^{(4)}(\mathbf{x}) \quad \text{mit } \mathbf{x} \in [a, b]$$

**Übung:**

Bei der Übertragung digitaler Daten als Pulssignal ( 1 ---> Spannung +U<sub>0</sub> , 0 ---> Spannung - U<sub>0</sub> ) hängt die Fehlerrate P<sub>F</sub> der detektierten Bits vom Verhältnis der Signalleistung zur Rauschleistung am Empfängereingang ( P<sub>S</sub>/P<sub>R</sub>) ab. Die Formel lautet:

$$P_F = \frac{1}{2} \operatorname{erfc} \left( \frac{1}{\sqrt{2}} \sqrt{\frac{P_S}{P_R}} \right)$$

Dabei ist erfc(x) das komplementäre Gauß'sche Fehlerintegral (siehe Kap. 1 )

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$$

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy$$

Man berechne die Fehlerrate mit Hilfe der Trapezmethode sowie mit der Simpson'schen Methode für N=10, Rechengenauigkeit mindestens 5 Stellen, für einen Signal-Rausch Abstand von 9,03 dB , entsprechend P<sub>S</sub>/P<sub>R</sub> = 7.9 und x = 2 .

**Lösung** (Numerische Werte mit Math-Cad):

$$\begin{array}{llll}
 a := 0 & b := 2 & N := 10 & h := \frac{b-a}{N} \quad h = 0.2 \\
 i := 0..10 & x(i) := i \cdot h & x1(i) := \left\{ i + \frac{1}{2} \right\} \cdot h & \\
 y(i) := \frac{2}{\sqrt{\pi}} \cdot \exp[-(i \cdot h)^2] & & y1(i) := \frac{2}{\sqrt{\pi}} \cdot \exp\left[-\left[\left\{ i + \frac{1}{2} \right\} \cdot h\right]^2\right] & \\
 \begin{array}{|c|} \hline y(i) \\ \hline 1.128379 \\ 1.084135 \\ 0.961541 \\ 0.787243 \\ 0.594986 \\ 0.415107 \\ 0.267344 \\ 0.158942 \\ 0.087229 \\ 0.044192 \\ 0.020667 \\ \hline \end{array} & & \begin{array}{|c|} \hline y1(i) \\ \hline 1.117152 \\ 1.031261 \\ 0.878783 \\ 0.691275 \\ 0.501969 \\ 0.33648 \\ 0.208208 \\ 0.11893 \\ 0.062711 \\ 0.030525 \\ 0.013716 \\ \hline \end{array} & 
 \end{array}$$

$$T := \frac{h}{2} \cdot (y(0) + 2 \cdot (y(1) + y(2) + y(3) + y(4) + y(5) + y(6) + y(7) + y(8) + y(9)) + y(10))$$

$$T = 0.995049$$

$$M := h \cdot (y1(0) + y1(1) + y1(2) + y1(3) + y1(4) + y1(5) + y1(6) + y1(7) + y1(8) + y1(9))$$

$$M = 0.995458$$

$$F := T - M$$

$$F = -4.098941 \cdot 10^{-4}$$

$$T1 := \frac{1}{2} \cdot (T + M)$$

$$T1 = 0.995253$$

$$S := \frac{h}{3} \cdot (y(0) + 4 \cdot y(1) + 2 \cdot y(2) + 4 \cdot y(3) + 2 \cdot y(4) + 4 \cdot y(5) + 2 \cdot y(6) + 4 \cdot y(7) + 2 \cdot y(8) + 4 \cdot y(9) + y(10))$$

$$S = 0.99531491$$

$$P := \frac{1}{2} \cdot (1 - S)$$

$$P = 2.342543 \cdot 10^{-3}$$

### 3.3 Numerische Lösung von gewöhnlichen Differentialgleichungen erster Ordnung

In der Schaltungsentwicklung und Regelungstechnik wird eine mathematische Beschreibung der Systemeigenschaften vorgenommen, die auf der Aufstellung von Differentialgleichungen basiert, welche die zeitliche Änderung einer oder mehrerer Zustandsgrößen beschreiben. Durch Lösung der Differentialgleichungen unter den gegebenen Anregungsfunktionen, Anfangs- und Randbedingungen ist es möglich, die Eigenschaften bzw. das zeitliche Verhalten des Systems voraus zu berechnen. Für die in der Praxis vorkommenden Schaltungen und Systeme ist meist eine analytische, geschlossene Lösung der Differentialgleichungen nicht mehr möglich und sinnvoll. Auch würde es großen Aufwand erfordern die Systemreaktion auf beliebige Anregungen zu ermitteln. Deswegen sind die analytischen Integrationsverfahren für Differentialgleichungen nur für einfache grundsätzliche Beispiele von Bedeutung, während in den Simulationsprogrammen numerische Verfahren eingesetzt werden, die hinreichend genaue Näherungswerte der gesuchten Lösungsfunktionen liefern. Im folgenden werden verschiedene Verfahren dargestellt und bestimmte Eigenschaften beschrieben, die bei der Anwendung der Rechenverfahren zu beachten sind.

Man kann die numerische Lösung einer Differentialgleichung auch so interpretieren: Physikalische Systeme sind durch die Naturgesetze in Form von Differentialgleichungen in der lokalen zeitlichen und räumlichen Umgebung in ihrem Verhalten festgelegt. Durch Näherung - insbesondere einer Linearisierung - der Gesetze in der lokalen Umgebung kann man auf einfache Weise näherungsweise den Zustand des Systems an einem um eine kleine Strecke oder einem kleinen Zeitintervall entfernten Ort bzw. Zeitpunkt bestimmen und von dort dasselbe Verfahren erneut solange anwenden, bis man das Verhalten des Systems über eine größere Strecke bzw. einen größeren Zeitraum angenähert berechnet hat.

Wir beginnen mit Lösungsverfahren für eine explizite Differentialgleichung erster Ordnung  $y' = f(x,y)$  und erweitern dann die Verfahren auf Systeme von Differentialgleichungen.

Unter einer Lösung der Dgl  $y' = f(x,y)$  versteht man eine Funktion  $y = y(x)$ , für die  $y'(x) = f(x,y(x))$  für alle  $x \in D$ , ( $D = \text{Definitionsintervall der Lösung}$ ) gilt. Zur Festlegung einer bestimmten Lösung aus einer einparametrischen Lösungsschar einer Differentialgleichung erster Ordnung ist eine Anfangsbedingung  $y(x_0) = y_0$  nötig, welche an der vorgegebenen Stelle  $x_0$  den Anfangswert  $y_0$  vorgibt.

Wir setzen für die Anwendung der Verfahren voraus, daß die Differentialgleichungen lösbar sind und eine eindeutige Lösung existiert, zu der die Näherungslösung konstruiert werden kann.

### 3.3.1 Die Methoden von Euler , Heun und Runge-Kutta

Durch die Differentialgleichung ist im Punkt  $(x_0, y_0)$  mit dem Wert  $y'(x_0) = f(x_0, y_0)$  die Steigung der Tangente der gesuchten Lösungsfunktion festgelegt.

Die Lösung wird schrittweise, ausgehend vom Punkt  $(x_0, y_0)$  durch Aneinandersetzen von Geradenstückchen konstruiert.

Diskretisierung des Integrationsintervalls (Stützstellen):

$$a = x_0, x_1, \dots, x_n, \dots$$

$$x_n = x_0 + n \cdot h, \quad n = 0, 1, 2, \dots$$

Die Werte für die Näherungslösung an den Stützstellen sollen der Einfachheit halber mit  $y_n$  bezeichnet werden. Man beachte, daß für Kapitel 3.3, 3.4 und 3.5 gilt:

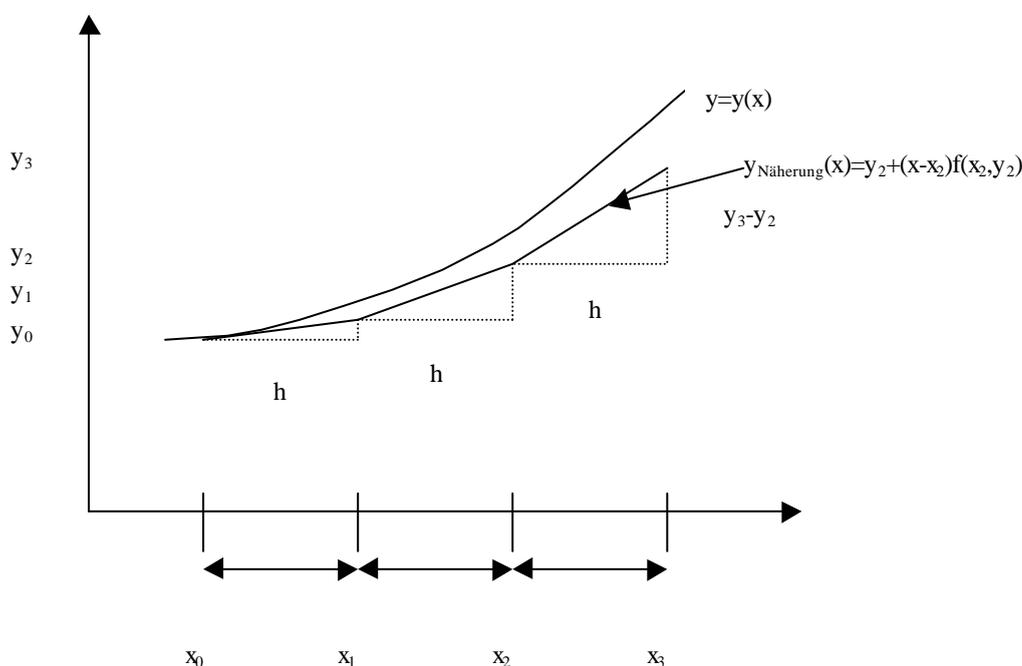
$$\boxed{\begin{array}{l} y_n = \text{Näherungswert für } y(x_n) \\ y_n \neq y(x_n) \end{array}}$$

#### Euler-Verfahren

Die Ableitung wird durch den rechten (vorwärts) Differenzenquotienten ersetzt.

Sukzessive Berechnung von Näherungswerten  $y_n$  für die exakten Lösungswerte  $y(x_n)$  durch Anwendung der Rechenvorschrift:

$$\boxed{\begin{array}{l} y_{n+1} = y_n + h \cdot f(x_n, y_n), \quad n = 0, 1, 2, 3 \\ y_0 = y(x_0) \end{array}}$$



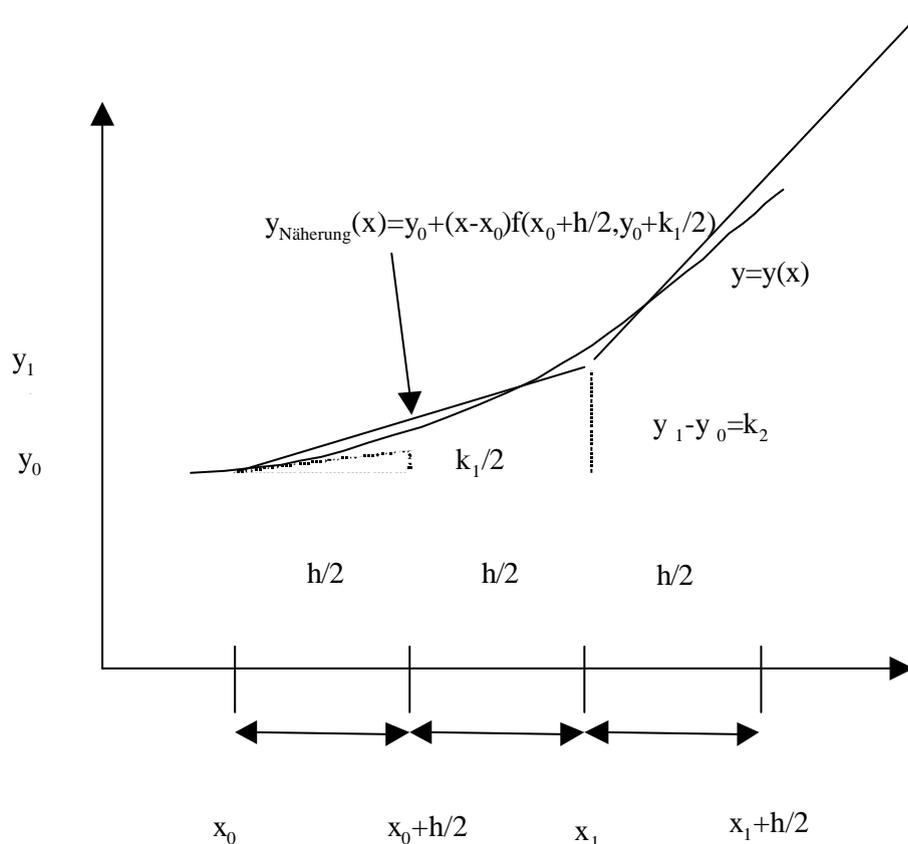
Die Methode ist sehr grob und liefert nur in den einfachsten Fällen brauchbare Ergebnisse.

### Verbessertes Euler-Verfahren (verbessertes Polygonzugverfahren oder zentrierte Differenzenmethode)

Eine verbessertes Verfahren ergibt sich, wenn man zur Berechnung der linearen Geradenabschnitte der Näherungslösung zwischen den zwei Wertepaaren  $(x_n, y_n)$  und  $(x_{n+1}, y_{n+1})$  den zentralen Differenzenquotienten statt dem rechten Differenzenquotienten heranzieht.

Als Algorithmus formuliert :

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\ y_{n+1} &= y_n + h \cdot k_2 \end{aligned}$$



### Klassisches Runge-Kutta-Verfahren 4. Ordnung

Noch genauer ist eine von Runge und Kutta angegebene Methode, bei der zur Berechnung der Steigung des linearen Stückes der Näherungslösung zwischen den Punkten  $(x_n, y_n)$  und  $(x_{n+1}, y_{n+1})$  an vier Stellen des Richtungsfeldes der Differentialgleichung Steigungen ausgerechnet werden und dann ein Mittelwert aus diesen verwendet wird (sog. klassisches Runge-Kutta-Verfahren 4. Ordnung).

Als Algorithmus geschrieben ergibt sich:

$$\begin{aligned}
 k_1 &= h \cdot f(x_n, y_n) \\
 k_2 &= h \cdot f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\
 k_3 &= h \cdot f\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) \\
 k_4 &= h \cdot f(x_n + h, y_n + k_3) \\
 k &= \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
 y_{n+1} &= y_n + k
 \end{aligned}$$

In der folgenden Zeichnung (auf der nächsten Seite) wird die Konstruktion der Lösung veranschaulicht. Man erkennt, daß von  $P_0$  ausgehend im Intervall  $[x_n, x_{n+1}]$  fünf Geraden, nämlich  $g_1(x)$ ,  $g_2(x)$ ,  $g_3(x)$ ,  $g_4(x)$  und  $y_{\text{Näherung}}(x)$ , eingezeichnet sind. Die erste Gerade  $g_1(x)$  hat die Steigung des Richtungsfeldes im Punkt  $P_0$ . Die zweite Gerade  $g_2(x)$  hat die Steigung des Richtungsfeldes in  $P_1$ , dem Mittelpunkt auf der ersten Geraden. Die dritte Gerade  $g_3(x)$  hat die Steigung des Richtungsfeldes in  $P_2$ , dem Mittelpunkt auf der zweiten Geraden und die vierte Gerade  $g_4(x)$  hat die Steigung des Richtungsfeldes in  $P_3$ , dem Endpunkt auf der dritten Geraden. Die Steigung der Geraden für die Näherungslösung wird aus dem Mittelwert dieser vier Steigungen bestimmt, wobei die beiden Steigungen, die aus zentralen Differenzquotienten gebildet werden, gegenüber den beiden an den Endpunkten durch rechten bzw. linken Differenzenquotienten gebildeten Steigungen doppelt gewertet werden.

$$\begin{aligned}
 g_1(x) &= y_n + (x - x_n)f(x_n, y_n) & g_1\left(x_n + \frac{h}{2}\right) &= y_n + \frac{k_1}{2} \\
 \Leftrightarrow \frac{k_1}{h} &= f(x_n, y_n) = \text{Steigung des Richtungsfeldes in } P_0 \\
 g_2(x) &= y_n + (x - x_n)f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) & g_2\left(x_n + \frac{h}{2}\right) &= y_n + \frac{k_2}{2} \\
 \Leftrightarrow \frac{k_2}{h} &= f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) = \text{Steigung des Richtungsfeldes in } P_1
 \end{aligned}$$

3.3 Numerische Lösung von gewöhnlichen Differentialgleichungen erster Ordnung

SS 99

$$g_3(x) = y_n + (x - x_n)f\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) \quad g_3(x_n + h) = y_n + k_3$$

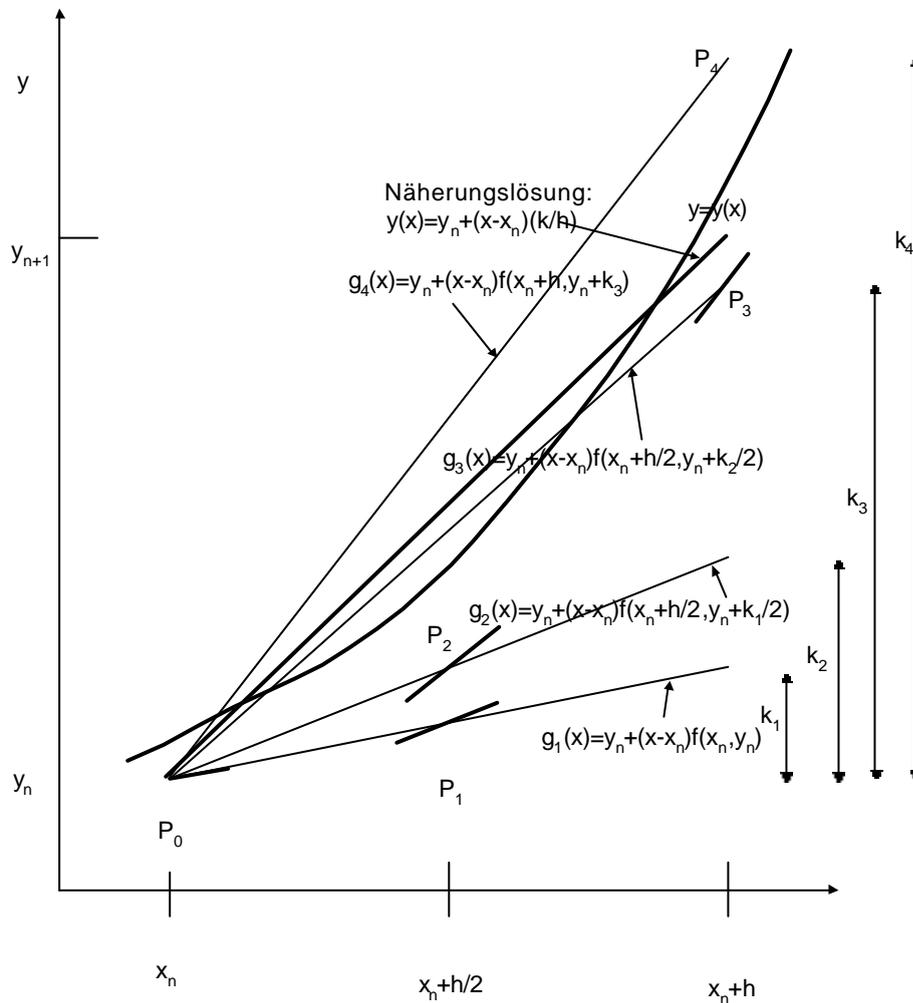
$$\Leftrightarrow \frac{k_3}{h} = f\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) = \text{Steigung des Richtungsfeldes in } P_2$$

$$g_4(x) = y_n + (x - x_n)f(x_n + h, y_n + k_3) \quad g_4(x_n + h) = y_n + k_4$$

$$\Leftrightarrow \frac{k_4}{h} = f(x_n + h, y_n + k_3) = \text{Steigung des Richtungsfeldes in } P_3$$

$$y_{\text{Näherung}}(x) = y_n + (x - x_n) \frac{k}{h} \quad \text{mit } k = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

für  $x_n \leq x \leq x_{n+1}$



Das klassische Runge- Kutta-Verfahren haben wir lediglich plausibel gemacht aber auf die Herleitung verzichtet. Da es eine ganze Reihe von Runge-Kutta-Verfahren gibt, die dem klassischen ähnlich sind, aber Unterschiede bzgl. Rechenaufwand und Genauigkeit aufweisen, wollen wir dennoch das bei diesen Methoden anwendbare, grundsätzliche Prinzip der Herleitung von ‚Einschrittverfahren höherer Fehlerordnung‘ am Beispiel der Methode von Heun vorstellen.

### Trapezverfahren von Heun

Dabei konstruiert man eine genäherte Lösung der Anfangswertaufgabe durch bestimmte Integration der Differentialgleichung  $y'(x)=f(x,y(x))$  bezüglich der unabhängigen Variablen  $x$  über das Intervall  $[x_n, x_{n+1}]$  der Länge  $h = x_{n+1} - x_n$

$$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f(x, y(x)) dx .$$

Auf der rechten Seite nähern wir das Integral mit der Trapezregel an. Die Gleichung wird demzufolge nur noch genähert gültig sein, weshalb wir  $y(x_n)$  durch  $y_n$  ersetzen:

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1})] \quad (**)$$

Für eine allgemeine nichtlineare Differentialgleichung haben wir damit eine implizite Gleichung für den unbekanntenen Näherungswert  $y_{n+1}$  gefunden. Jeder Integrationsschritt erfordert damit die Lösung einer nichtlinearen Gleichung.

Im Spezialfall einer linearen Differentialgleichung erster Ordnung

$$y'(x) = a(x)y(x) + b(x)$$

mit gegebenen Funktionen  $a(x)$  und  $b(x)$  führt die Trapezmethode auf die lineare Gleichung für  $y_{n+1}$

$$y_{n+1} = y_n + \frac{h}{2} [a(x_n)y_n + b(x_n) + a(x_{n+1})y_{n+1} + b(x_{n+1})]$$

Daraus ergibt sich die explizite Rekursionsformel

$y_{n+1} = \frac{[2 + h \cdot a(x_n)]y_n + h[b(x_n) + b(x_{n+1})]}{2 - h \cdot a(x_{n+1})} \quad (n = 0, 1, 2, 3, \dots)$
---

zur sukzessiven Berechnung der Werte  $y_1, y_2, y_3, \dots$

Für eine nichtlineare Differentialgleichung muß bei jedem Integrationsschritt eine Lösung der impliziten Gleichung

(\*\*)

$$y_{n+1} = \mathbf{j}(y_{n+1})$$

**3.3 Numerische Lösung von gewöhnlichen Differentialgleichungen erster Ordnung**

SS 99

bestimmt werden. Dies kann nach der Methode der sukzessiven Approximation erreicht werden. Man setzt einen Näherungswert für die Lösung in die linke Seite der Gleichung ein und berechnet damit einen verbesserten Näherungswert. Mit diesem verfährt man wieder in derselben Weise und so fort. Unter bestimmten Voraussetzungen

( es muß gelten

$$|f(x, y) - f(x, y^*)| \leq L|y - y^*| \quad \text{für alle } x, y, y^* \in B$$

und einer bestimmten Konstante  $0 < L < \infty$  sowie

$$\frac{1}{2} hL < 1 \quad )$$

konvergiert die Folge der so bestimmten Näherungswerte gegen den ‚Fixpunkt‘  $y_{n+1}$ .

Bei der **Methode nach Heun** setzt man als Startwert für die Iteration den nach Euler bestimmten Wert (Prädiktorwert):

$$y_{n+1}^{(P)} = y_n + hf(x_n, y_n)$$

und berechnet mit diesem Prädiktorwert mit nur einem Iterationsschritt einen verbesserten Wert, den sog. Korrektorwert:

$$y_{n+1}^{(K)} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(P)})]$$

Als Algorithmus geschrieben ergibt sich :

$k_1 = h \cdot f(x_n, y_n)$ $k_2 = h \cdot f(x_n + h, y_n + k_1)$ $y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$
--

### 3.3.2 Diskretisierungsfehler, Fehlerordnung, Schrittweitensteuerung

Der lokale Diskretisierungsfehler beschreibt die Abweichung, mit welcher die exakte Lösung  $y(x)$  die verwendete Integrationsvorschrift in einem einzelnen Schritt nicht erfüllt.

Der globaler Diskretisierungsfehler:

$e_n := y(x_n) - y_n$  beschreibt die Abweichung des exakten Lösungswertes  $y(x_n)$  vom Näherungswerte  $y_n$ .

Die Fehlerordnung eines Verfahrens beschreibt die Abhängigkeit des globalen Fehlers von der Schrittweite  $h$ :

Ein Einschrittverfahren besitzt die Fehlerordnung  $p$ , falls für seinen maximalen lokalen Diskretisierungsfehler  $d_{\max}$  gilt:

$$d_{\max} \leq \text{const} \cdot h^{p+1} = O(h^{p+1})$$

Insbesondere bedeutet das, daß bei einem Verfahren mit der Fehlerordnung  $p$  bei Halbierung der Schrittweite  $h$  der globale Fehler auf das  $\frac{1}{2^p}$  fache abnimmt.

Durch Taylorentwicklungen und Abschätzungen lassen sich für die genannten Verfahren folgende Ergebnisse bezüglich Fehlerordnung ableiten:

#### a) Euler-Verfahren :

lokale Fehlerordnung = 2. Globale Fehlerordnung = 1.

Neben dem Verfahrensfehler spielen auch noch Fehler aufgrund der endlichen Stellengenauigkeit beim Rechenvorgang (sog. **Rundungsfehler**) eine Rolle. **Während mit kleiner werdender**

**Schrittweite der lokale Fehler abnimmt, nimmt der Rundungsfehler zu** ( $\approx \frac{d_{\max}}{h}$ ). Es läßt sich

daher für das Verfahren eine **optimale Schrittweite** festlegen, gegeben durch

$$h_0 = \sqrt{\frac{2 \cdot d_{\max}}{M}}$$

wobei  $M$  eine obere Schranke für die zweite Ableitung der Lösungsfunktion  $y(x)$  im Definitionsintervall ist

$$|y''(x)| < M \quad \text{für alle } x \in [a, b]$$

$M$  muß geschätzt werden!

#### b) Verbessertes Euler-Verfahren und Heun-Verfahren:

Fehlerordnung = 2

**c) Klassisches Runge-Kutta-Verfahren:**

lokale Fehlerordnung = 4, globale Fehlerordnung = 4 .

Dieses Verfahren reagiert sehr empfindlich auf eine Veränderung der Schrittweite. Bei Halbierung der Schrittweite geht der lokale Fehler auf 1/32 zurück. Die optimale Schrittweite wird gemäß einer empirischen Beziehung festgelegt:

$$h \cdot K \approx 0.1$$

Dabei ist K die sog. Lipschitzkonstante, die eine obere Schranke der partiellen Ableitung der Funktion  $f(x,y)$  nach  $y$  darstellt. Sie muß geschätzt werden.

$$\left| \frac{\partial f(x, y)}{\partial y} \right| \leq K$$

Ein Maß für den lokalen Diskretisationsfehler ergibt sich aus dem Vergleich zwischen zwei Näherungswerten für  $y_{n+1}$ , die man aus  $y_n$  bei unterschiedlicher Schrittweite berechnet.

$$d_{\max} \approx \frac{y_{n+1}(h) - y_{n+1}\left(\frac{h}{2}\right)}{15}$$

**Schrittweitensteuerung bei Runge-Kutta:**

Wegen der empfindlichen Reaktion des Fehlers auf die Schrittweite sollte die Schrittweite nicht kleiner als nötig gewählt werden.

Die Schrittweite wird deswegen der jeweiligen Steigung der Lösungskurve angepaßt. Ein Maß für die Anpassung liefert der Parameters

$$q = \left| \frac{k_3 - k_2}{k_2 - k_1} \right|$$

Dieser Parameter wird nach jedem Schritt berechnet und abhängig davon wird die Schrittweite gemäß der folgenden Vorschrift angepaßt:

Für $q < 0,005$	Schrittweite verdoppeln
Für $q > 0,04$	Schrittweite halbieren und den vorangehenden Schritt mit dieser Schrittweite wiederholen
Für $0,005 < q < 0,04$	Schrittweite unverändert lassen

### 3.4 Differentialgleichungen höherer Ordnung und Systeme

Differentialgleichungen höherer Ordnung und Systeme von Differentialgleichungen lassen sich durch Einführung von Zwischenfunktionen auf Systeme von gewöhnlichen Differentialgleichungen erster Ordnung zurückführen .

#### Beispiel:

Differentialgleichung dritter Ordnung

$$y^{(3)}(x) = f(x, y(x), y'(x), y''(x))$$

Anfangsbedingungen:

$$y(x_0) = y_0, \quad y'(x_0) = y'_0, \quad y''(x_0) = y''_0$$

Umbenennung der unbekannteten Funktion  $y(x)$  in  $y_1(x)$  und

Einführung von zwei neuen Funktionen  $y_2(x)$  und  $y_3(x)$  durch die Definitionen:

$$y_2(x) = y_1'(x), \quad y_3(x) = y_2'(x) = y_1''(x)$$

Damit ergibt sich ein System von drei Dgl. erster Ordnung für die drei Funktionen  $y_1, y_2, y_3$

$$y_1'(x) = y_2(x)$$

$$y_2'(x) = y_3(x)$$

$$y_3'(x) = f(x, y_1(x), y_2(x), y_3(x))$$

mit den Anfangsbedingungen

$$y_1(x_0) = y_0, \quad y_2(x_0) = y'_0, \quad y_3(x_0) = y''_0$$

Einführen der Vektorschreibweise:

$$\mathbf{y}(x) = \begin{pmatrix} y_1(x) \\ y_2(x) \\ y_3(x) \end{pmatrix}, \quad \mathbf{f}(x, \mathbf{y}) = \begin{pmatrix} y_2(x) \\ y_3(x) \\ f(x, y_1(x), y_2(x), y_3(x)) \end{pmatrix}, \quad \mathbf{y}_0 = \begin{pmatrix} y_0 \\ y'_0 \\ y''_0 \end{pmatrix}$$

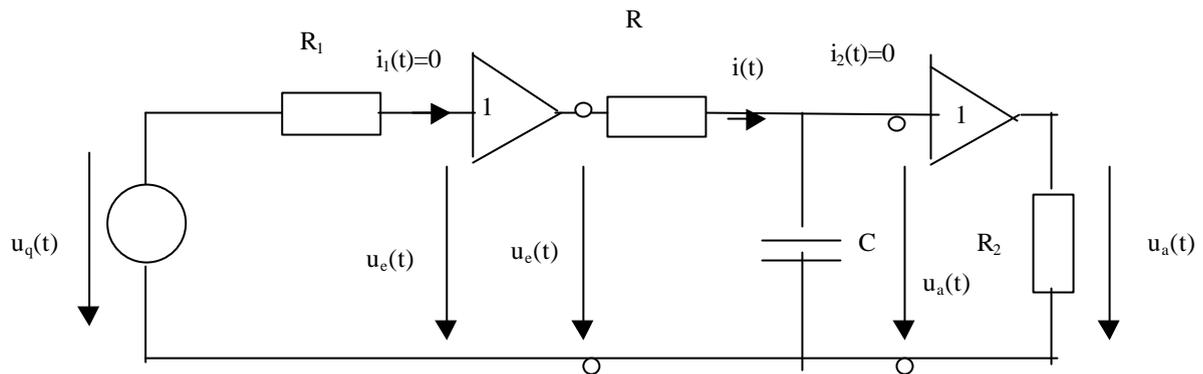
$$\mathbf{y}'(x) = \mathbf{f}(x, \mathbf{y}(x)), \quad \mathbf{y}(x_0) = \mathbf{y}_0$$

Die beschriebenen Einschrittmethoden können unmittelbar auf Vektorfunktionen übertragen werden. Formal muß man lediglich die skalaren Größen durch entsprechende Vektorgrößen ersetzen und bei der Anwendung die Vektorkomponenten ausrechnen.

### 3.5 Beispiele

Einige Beispiele werden mit dem Taschenrechner gelöst, für andere empfiehlt sich der Einsatz von Programmen in PASCAL oder C. Diese sind u.a. zu finden in [ 11 ] bis [ 14 ]. Bei einigen Beispielen wird das Programm MAYA aus dem Buch von Kose et al. mit dem Titel ‚Numerik sehen und verstehen‘ [ 9 ] eingesetzt. Das Buch enthält eine Diskette und ist in der Bibliothek verfügbar.

#### 3.5.1 RC-Glied



$$u_a(t) = \frac{1}{C} \int i(t) dt$$

$$R \cdot i(t) + \frac{1}{C} \int i(t) dt = u_e(t)$$

$$\Rightarrow \frac{du_a(t)}{dt} = -\frac{1}{RC} (u_a(t) - u_e(t))$$

ergibt Dgl. erster Ordnung für  $u_a(t) = y(t)$  vom Typ

$$\dot{y} = f(t, y) \quad \text{mit} \quad y = y(t)$$

oder

$$y' = f(x, y) \quad \text{mit} \quad y = y(x)$$

man kann die Anregungsfunktion auch noch explizit in die Funktion  $f(t,y)$  schreiben

$$\dot{y}(t) = f(t, u_e(t), y(t))$$

was aber für die Anwendung der besprochenen Verfahren keine Vorteile bringt. Wir wollen deswegen darauf verzichten.

Wird am Eingang eine Sprungfunktion angelegt, so ergibt sich bei ungeladenem Kondensator zur Zeit  $t = 0$  das Anfangswertproblem:

$$\frac{du_a(t)}{dt} = \frac{1}{RC} (1 - u_a(t)) \quad \text{für } t \geq 0$$

$$u_a(0) = 0$$

Die exakte Lösung erhält man durch Trennung der Variablen ( $T:=RC$ ):

$$\frac{du_a}{1-u_a} = \frac{dt}{T}, \quad -\ln(1-u_a) = \frac{t}{T} + C, \quad 1-u_a = C' e^{-\frac{t}{T}},$$

$$u_a(t; C') = (1 - C' e^{-\frac{t}{T}}) \quad u_a(0) = 0 \quad \Rightarrow \quad C' = 1$$

$$u_a(t) = (1 - e^{-\frac{t}{T}}) \quad t \geq 0$$

Lösung mit Euler-Verfahren ( $T = 1$ ):

$$\frac{u_{a,n+1} - u_{a,n}}{h} = f(t_n, u_{a,n})$$

$$u_{a,n+1} = u_{a,n} + k \quad \text{mit} \quad k = h \cdot f(t_n, u_{a,n}) = h \cdot (1 - u_{a,n})$$

$h = 0.2$

n	$t_n$	$u_{a,n}$	$f(t_n, u_{a,n}) = 1 - u_{a,n}$	k	$u_a(t_n)$ exakt
0	0	0	1	0.2	0
1	0.2	0.2	0.8	0.16	0.181
2	0.4	0.36	0.64	0.128	0.330
3	0.6	0.488	0.512	0.102	0.451
4	0.8	0.590			0.551

Lösung nach Runge-Kutta-Verfahren. Berechnung mit Hilfe einer Tabelle und dem Taschenrechner:

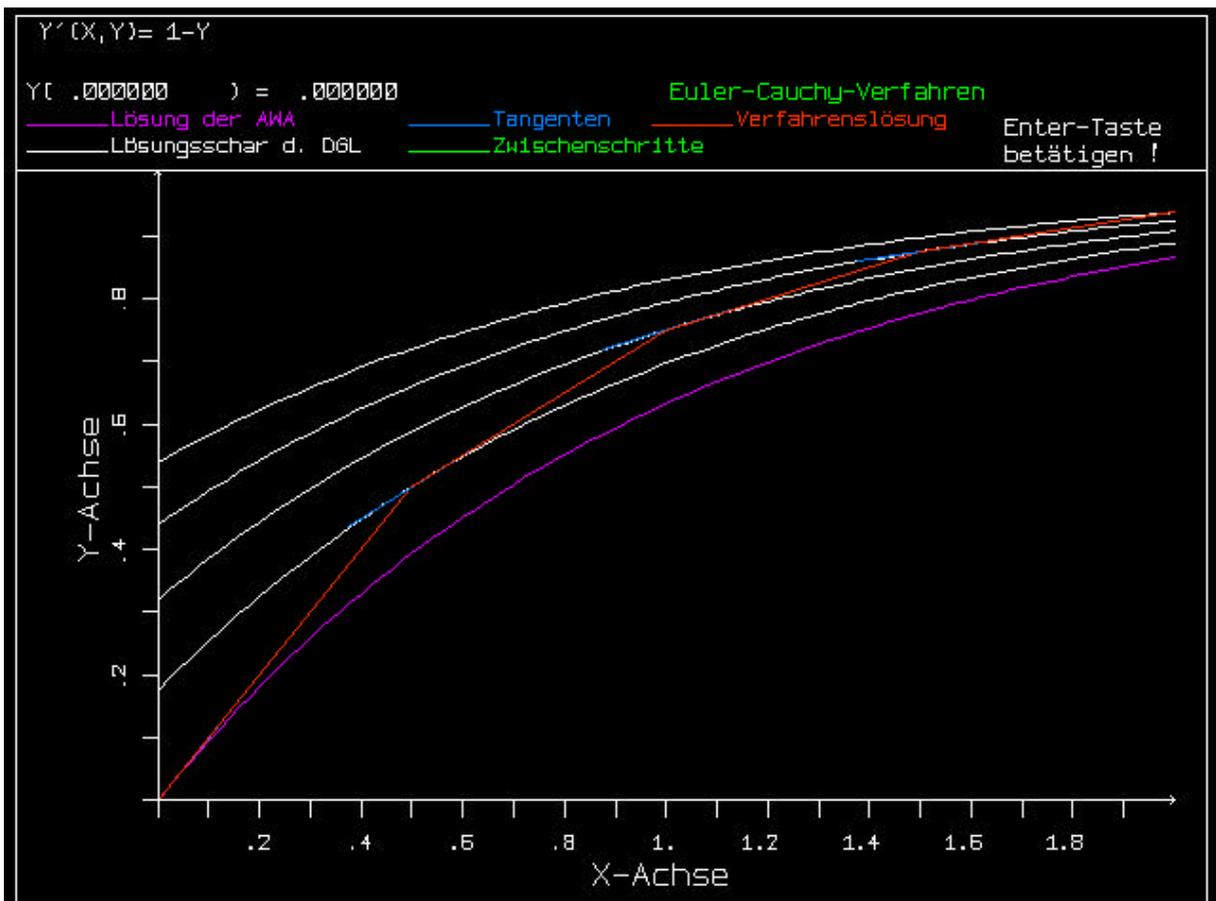
n	t	y	$f(t,y)$	$k_1^{(n)}$	
0	$t_0$	$y_0$	$f(x_0, y_0)$	$k_1^{(0)}$	$k_1^{(0)}$
	$t_0 + h/2$	$y_0 + k_1^{(0)}/2$	$f(t_0 + h/2, y_0 + k_1^{(0)}/2)$	$k_2^{(0)}$	$2k_2^{(0)}$
	$t_0 + h/2$	$y_0 + k_2^{(0)}/2$	$f(t_0 + h/2, y_0 + k_2^{(0)}/2)$	$k_3^{(0)}$	$2k_3^{(0)}$
	$t_0 + h$	$y_0 + k_3^{(0)}$	$f(t_0 + h, y_0 + k_3^{(0)})$	$k_4^{(0)}$	$k_4^{(0)}$
	$t_1 = t_0 + h$	$y_1 = y_0 + k^{(0)}$			$k^{(0)} = (1/6)\Sigma$

$h = 0.2$

n	t	y	$f(t,y)$	$k_1^{(n)}$	
0	0	0	1	0.2	0.2
	0.1	0.1	0.9	0.18	0.36
	0.1	0.09	0.91	0.182	0.364
	0.2	0.182	0.818	0.1636	0.1636
	0.2	0.181266			0.181266

	Exakt=	0.181269			
1	0.2	0.181266	0.818733	0.1637466	0.1637466
	0.3	0.263139	0.7368607	0.1473721	0.2947442
	0.3	0.254952	0.7450479	0.1490095	0.298013
	0.4	0.3302755	0.6697244	0.1339448	0.1339448
	0.4	0.3296741			0.1484081
	Exakt=	0.3296800			
n	t	y	f(t,y)	$k_i^{(n)}$	
2	0.4	0.3296741	0,6703259	0,1340651	0,1340651
	0.5	0.3967066	0,6032933	0,1206586	0,2413172
	0.5	0.3900034	0,6099965	0,1219993	0,2439986
	0.6	0.4516734	0,5483265	0,1096653	0,1096653
	0,6	<b>0,4511818</b>			0,1215077
	Exakt =	0.4511883			

Darstellung der Näherungslösung nach dem Euler-Verfahrens im Bereich von  $t = 0.. 2$  für  $h = 0,5$  mit dem Programm MAYA:



Berechnung eines Einschwingvorganges:

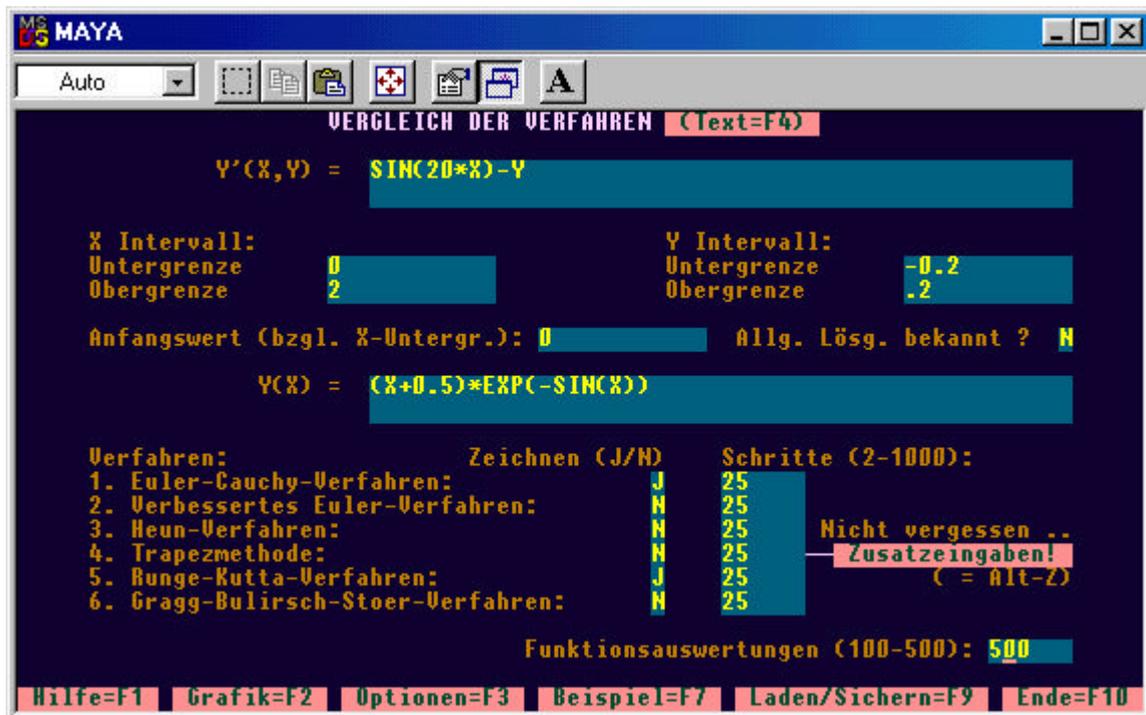
$$u_e(t) = \hat{u}_e \cos(\mathbf{w}t + \mathbf{j})$$

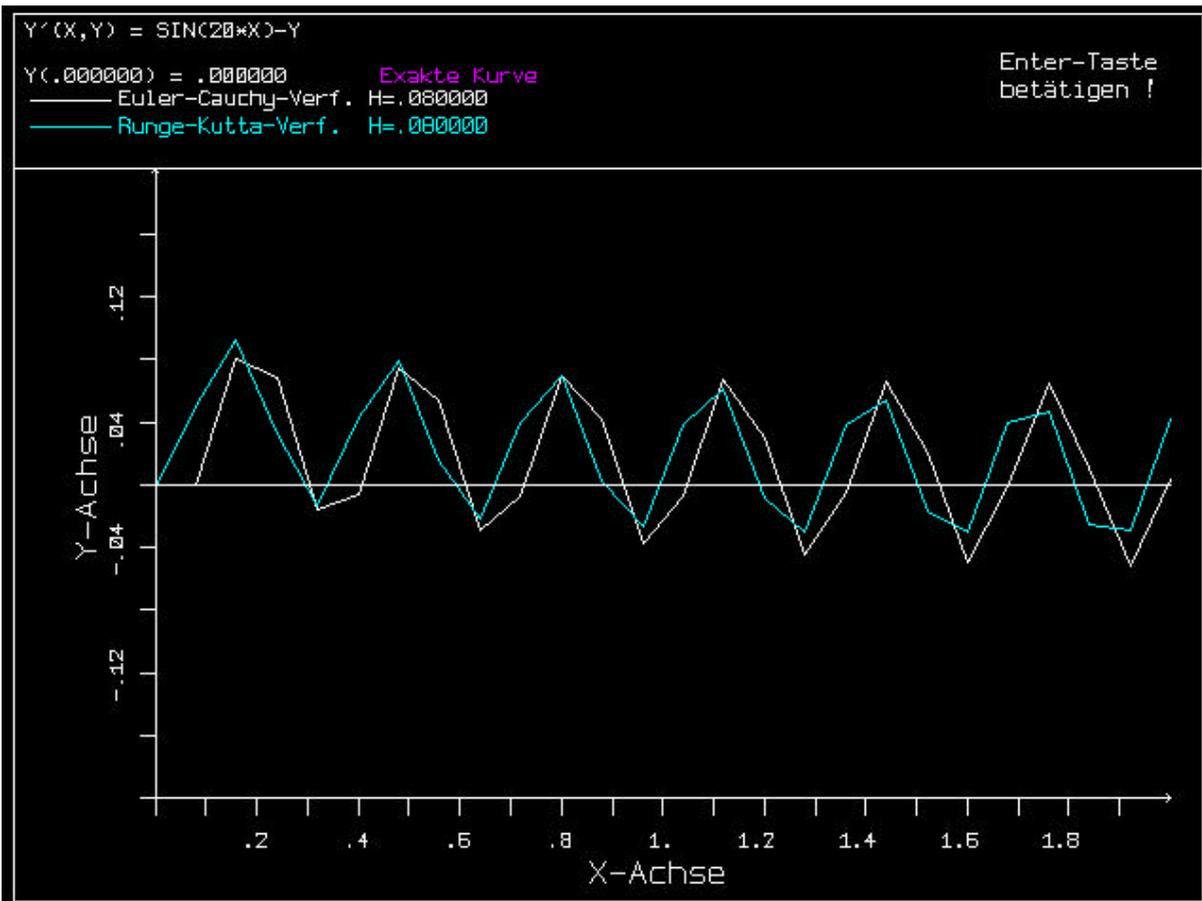
$$\hat{u}_e = 1, \quad \mathbf{w} = 20, \quad \mathbf{j} = -\frac{\mathbf{P}}{2} \quad \Rightarrow$$

$$u_e(t) = \sin(20t) \quad \text{für } t > 0$$

$$\text{und } u_e(t) = 0 \quad \text{für } t < 0$$

$$\frac{du_a}{dt} = -\frac{1}{RC}(u_a(t) - u_e(t)), \quad (RC = 1) \quad \Rightarrow \quad \frac{du_a(t)}{dt} = (\sin(20t) - u_e(t))$$





### 3.5.2 Nichtlineare Differentialgleichung erster Ordnung

$$y' = -2xy^2$$

$$y(0) = 1$$

exakte Lösung: durch Trennung der Variablen

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

$$-\frac{1}{y} + C_1 = -x^2 + C_2 \quad \frac{1}{y} = -x^2 + C' \quad y(x) = \frac{1}{x^2 + C'} \quad y(0) = 1 \quad \Rightarrow \quad C' = 1$$

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

Algorithmus für Euler:  $y_{n+1} = y_n - h \cdot 2x_n y_n^2 = y_n + h \cdot f(x_n, y_n)$

Globaler Diskretisationsfehler:  $e_n = y(x_n) - y_n$

$h = 0.2$

n	$x_n$	$y_{nEuler}$	$y(x_n)$	$f(x_n, y_{nE})$	$\epsilon_n$
0	0	1	1	0	0
1	0.2	1	0.9615385	-0.4	-0.0384615
2	0.4	0.92	0.862069	-0.67712	-0.057931
3	0.6	0.784576	0.7352941		-0.0492819

$h = 0.1$

n	$x_n$	$y_{nEuler}$	$y(x_n)$	$f(x_n, y_{nE})$	$\epsilon_n$
0	0	1	1	0	0
1	0.1	1	0.990099	-0.2	-0.009900
2	0.2	0.98	0.961538	-0.38416	-0.018462
3	0.3	0.94158	0.917431	-0.5319437	-0.024149
4	0.4	0.888389	0.862069	-0.631388	-0.026320
5	0.5	0.82525	0.800000	-0.6810376	-0.02525
6	0.6	0.757146	0.735294		-0.021852

Algorithmus für Heun:

$$k_1 = h \cdot f(x_n, y_n)$$

$$k_2 = h \cdot f(x_n + h, y_n + k_1)$$

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$$

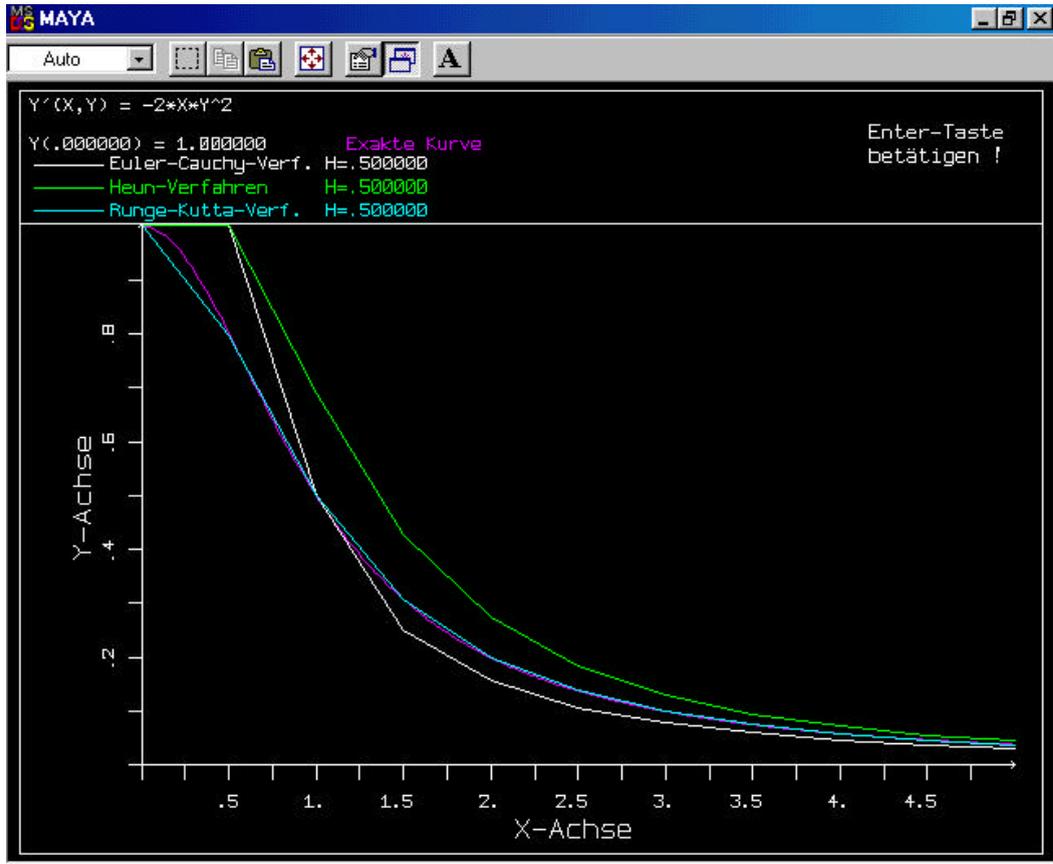
$h = 0.2$

n	$x_n$	$y_{nHeun}$	$y(x_n)$	$k_1 = hf(x_n, y_{nH})$	$k_2$	$\epsilon_n$
0	0	1	1	0	-0.08	
1	0.2	0.96	0.9615385	-0.073728	-0.1256765	-0.0015
2	0.4	0.8602978	0.862069	-0.118418	-0.1320926	-0.0017
3	0.6	0.7350425	0.7352941	-0.129669	-0.1172727	-0.00025
4	0.8	0.6115717	0.6097561			-0.0018

$h = 0.1$

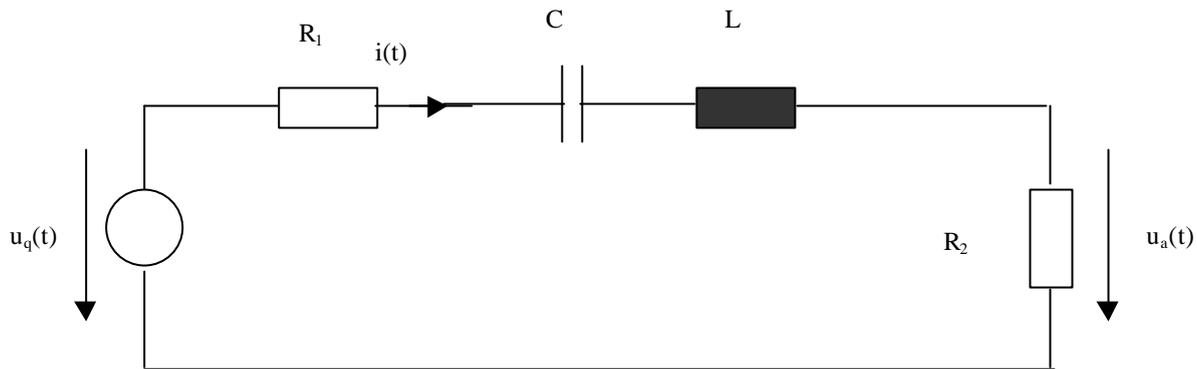
n	$x_n$	$y_{nHeun}$	$y(x_n)$	$k_1 = hf(x_n, y_{nH})$	$k_2$	$\epsilon_n$
0	0	1	1	0	-0.02	0
1	0.1	0.99	0.990099	-0.019602	-0.0376669	-0.000099
2	0.2	0.9613656	0.961538	-0.036969	-0.0512705	-0.00017
3	0.3	0.9172458	0.917431	-0.0504804	-0.0601026	-0.00018
4	0.4	0.8619543	0.862069	-0.0594372	-0.0644034	-0.00011
5	0.5	0.800034	0.800000	-0.0640054	-0.0650086	-0.000034
6	0.6	0.735527	0.735294	-0.06492	-0.0629599	-0.00023
7	0.7	0.671587	0.6711409			-0.00044

Graphische Darstellung der Lösung mit MAYA:



### 3.5.3 LC-Schwingkreis

Als Beispiel für Systeme 2. Ordnung mit 2 unabhängigen Energiespeichern



$$(R_1 + R_2)i(t) + \frac{1}{C} \int i(t) dt + L \frac{di}{dt} = u_q(t)$$

$$u_a(t) = R_2 i(t)$$

$$\frac{(R_1 + R_2)}{R_2} u_a(t) + \frac{1}{R_2 C} \int u_a(t) dt + \frac{L}{R_2} \frac{du_a}{dt} = u_q(t)$$

$$\frac{d^2 u_a}{dt^2} + \frac{(R_1 + R_2)}{R_2} \frac{du_a}{dt} + \frac{1}{LC} u_a = \frac{R_2}{L} \frac{du_q}{dt}$$

allgemein geschrieben:

$$\ddot{y} + 2\mathbf{d} \cdot \dot{y} + \mathbf{w}_0^2 y = s(t)$$

$s(t)$  = Störfunktion

$$y(0) = y_0, \quad \dot{y}(0) = v_0$$

exakte Lösung für homogenen Teil ( $s(t)=0$ , freie Schwingung):

über Exponentialansatz  $y(t) = y_0 e^{pt}$

$$\text{Eigenwertgleichung: } p^2 + 2\mathbf{d}p + \mathbf{w}_0^2 = 0 \quad \Rightarrow \quad p_{1/2} = -\mathbf{d} \pm \sqrt{\mathbf{d}^2 - \mathbf{w}_0^2}$$

Schwingfall :

$$\mathbf{d} < \mathbf{w}_0 : \quad p_{1/2} = -\mathbf{d} \pm j\mathbf{w}_r, \quad \mathbf{w}_r = \sqrt{\mathbf{w}_0^2 - \mathbf{d}^2}$$

$$\mathbf{d} < \mathbf{w}_0 \quad y(t) = e^{-\mathbf{d}t} \left[ y_0 \cos(\mathbf{w}_r t) + \frac{v_0 + \mathbf{d} \cdot y_0}{\mathbf{w}_r} \sin(\mathbf{w}_r t) \right] = \hat{y} e^{-\mathbf{d}t} \cos(\mathbf{w}_r t + \mathbf{f})$$

$$\mathbf{d} \ll \mathbf{w}_0 \quad \mathbf{w}_r \approx \mathbf{w}_0$$

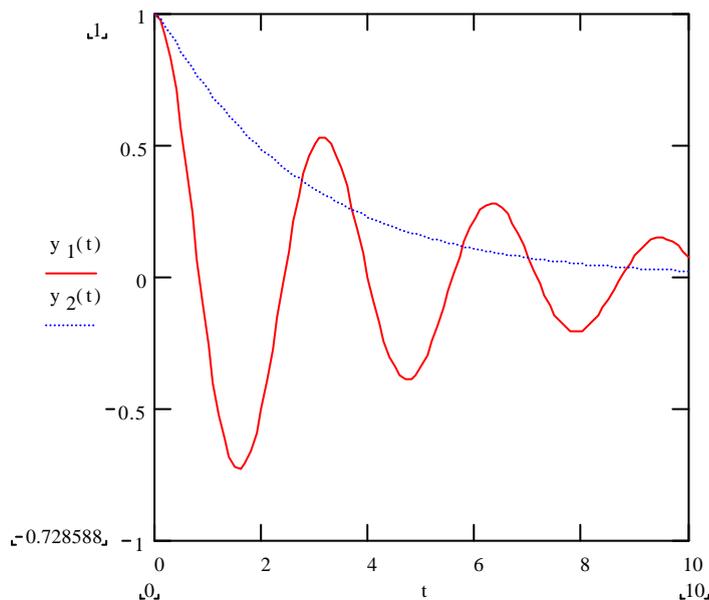
*aperiodischer Grenzfall :*

$$\mathbf{d} < \mathbf{w}_0 : p_1 = p_2 = -\mathbf{d} \quad y(t) = (1+t)e^{-\mathbf{d}t}$$

*Kriechfall :*

$$\mathbf{d} > \mathbf{w}_0 : p_{1/2} = -\mathbf{d} \pm \sqrt{\mathbf{d}^2 - \mathbf{w}_0^2} = -|p_{1/2}| < 0 \quad y(t) = \frac{y_0}{|p_2| - |p_1|} \left( |p_2| e^{-|p_1|t} - |p_1| e^{-|p_2|t} \right)$$

Zeichnung für  $y_0=1, v_0=0, \omega_0=2$  und den zwei Werte  $\delta_1=0.2$  (Kurve  $y_1(t)$ ) und  $\delta_2=5.5$  (Kurve  $y_2(t)$ ).



Numerische Berechnung der Lösung durch Überführen in System von Differentialgleichungen erster Ordnung:

$$v(t) := \frac{du_a}{dt}$$

$$\frac{du_a}{dt} = v(t)$$

$$\frac{dv}{dt} = -\frac{R_1 + R_2}{L}v(t) - \frac{1}{LC}u_a(t) + \frac{R_2}{L}\frac{du_q}{dt}$$

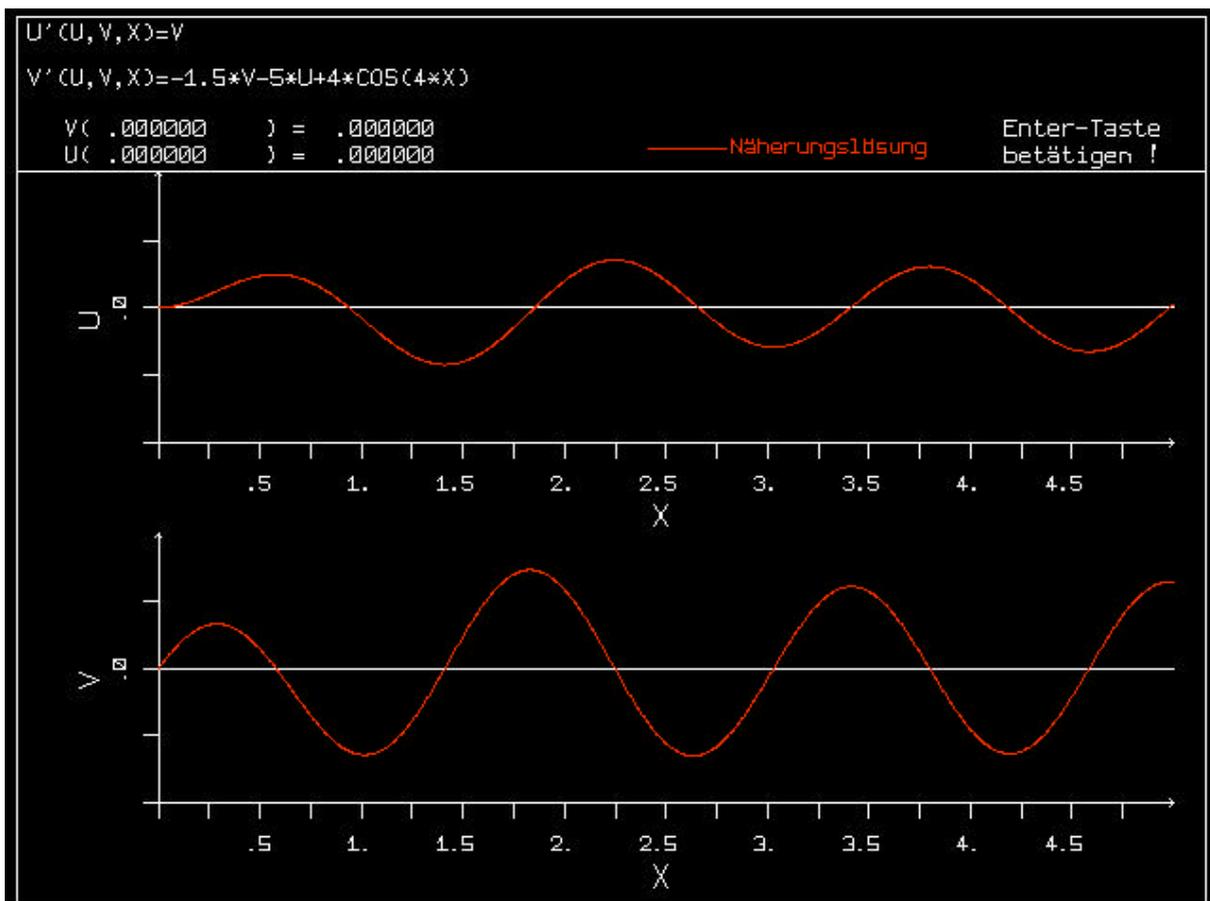
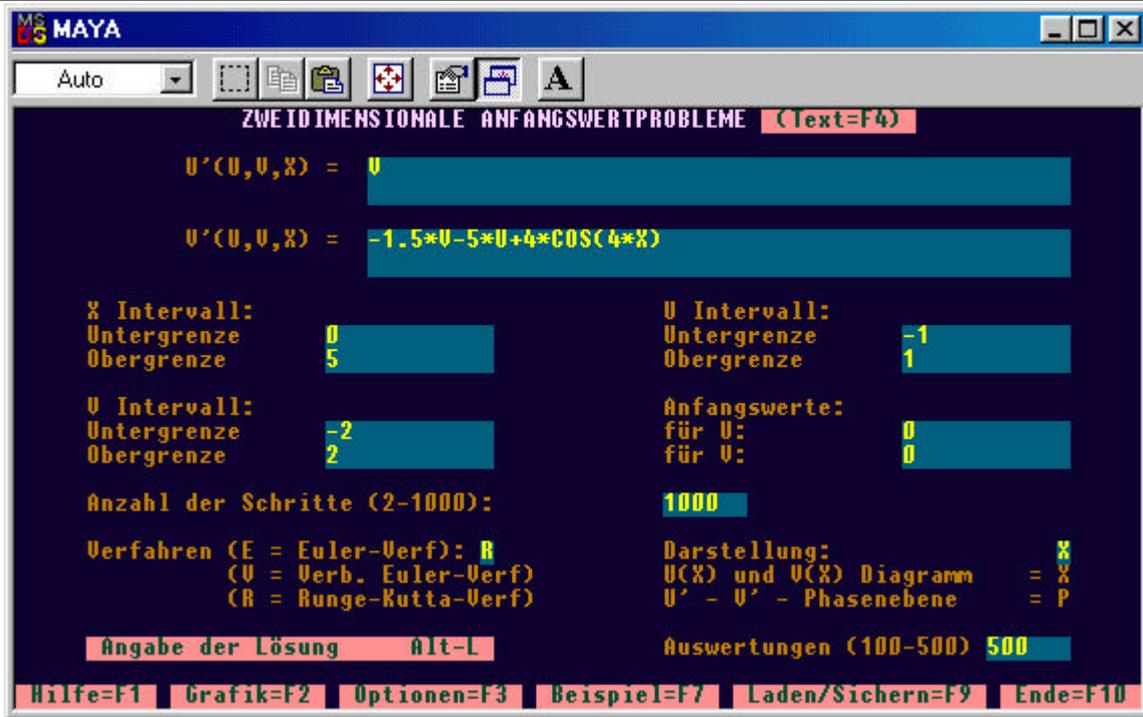
$$\mathbf{y} = \begin{pmatrix} u_a(t) \\ v(t) \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R_1 + R_2}{L} \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 0 \\ \frac{R_2}{L}\frac{du_q}{dt} \end{pmatrix} \quad \mathbf{y}_0 = \begin{pmatrix} u_a(0) \\ \frac{du_a}{dt}(0) \end{pmatrix}$$

$$\Rightarrow \dot{\mathbf{y}} = \mathbf{A}\mathbf{y} + \mathbf{b} \quad \text{mit} \quad \mathbf{y}(0) = \mathbf{y}_0$$

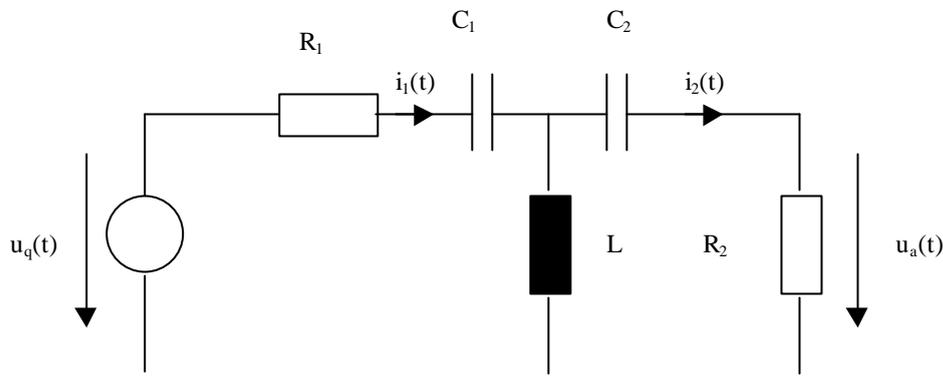
Einschwingvorgang bei Anregung des Schwingkreises mit Sinusspannung

Lösung mit MAYA :

$$(R_1+R_2)/L=1.5, \quad 1/(LC)=5, \quad R_2/L=1, \quad u_q(t)= 0 \quad \text{für } t < 0 \quad \text{und } u_q(t)= \sin(4t) \quad \text{für } t > 0$$



### 3.5.4 LC-Hochpaßfilter 3. Ordnung



Aufstellen der Gleichungen für die Ströme mit Hilfe der Kirchhoff'schen Maschenregeln

$$\begin{aligned}
 R_1 i_1(t) + \frac{1}{C_1} \int i_1(t) dt + L \frac{d}{dt} (i_1(t) - i_2(t)) &= u_q(t) \\
 R_2 i_2(t) + \frac{1}{C_2} \int i_2(t) dt - L \frac{d}{dt} (i_1(t) - i_2(t)) &= 0 \quad \Rightarrow \\
 R_1 \frac{d}{dt} i_1(t) + \frac{1}{C_1} i_1(t) + L \frac{d^2}{dt^2} i_1(t) - L \frac{d^2}{dt^2} i_2(t) &= \frac{d}{dt} u_q(t) \\
 R_2 \frac{d}{dt} i_2(t) + \frac{1}{C_2} i_2(t) - L \frac{d^2}{dt^2} i_1(t) + L \frac{d^2}{dt^2} i_2(t) &= 0
 \end{aligned}$$

Aus diesem linearen Differentialgleichungssystem für die beiden Ströme können Differentialgleichungen für den Strom  $i_1(t)$  und  $i_2(t)$  durch Elimination bestimmt werden. Die Koeffizienten  $-L_{ik}$  der Koeffizientenmatrix des linearen Gleichungssystems sind lineare Differentialoperatoren, die auf Zeitfunktionen wirken. Die Auflösung des Gleichungssystems nach einer der Unbekannten ergibt somit eine Differentialgleichung.

$$\begin{aligned}
 \left( \frac{1}{C_1} + R_1 \frac{d}{dt} + L \frac{d^2}{dt^2} \right) i_1(t) - L \frac{d^2}{dt^2} i_2(t) &= \frac{d}{dt} u_q(t) \\
 -L \frac{d^2}{dt^2} i_1(t) + \left( \frac{1}{C_2} + R_2 \frac{d}{dt} + L \frac{d^2}{dt^2} \right) i_2(t) &= 0 \quad \Leftrightarrow
 \end{aligned}$$

$$\begin{aligned}
 L_{11} i_1(t) + L_{12} i_2(t) &= b_1(t) \\
 L_{21} i_1(t) + L_{22} i_2(t) &= 0
 \end{aligned}$$

Auflösung nach  $i_2(t)$  mit der Cramer'schen Regel

$$i_2(t) = \frac{\begin{vmatrix} L_{11} & b_1(t) \\ L_{12} & b_2(t) \end{vmatrix}}{\begin{vmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{vmatrix}} \Leftrightarrow \begin{vmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{vmatrix} i_2(t) = \begin{vmatrix} L_{11} & b_1(t) \\ L_{12} & b_2(t) \end{vmatrix}$$

Also:

$$i_2(t) = \frac{\begin{vmatrix} \left( \frac{1}{C_1} + R_1 \frac{d}{dt} + L \frac{d^2}{dt^2} \right) \frac{d}{dt} u_q(t) \\ -L \frac{d^2}{dt^2} & 0 \end{vmatrix}}{\begin{vmatrix} \left( \frac{1}{C_1} + R_1 \frac{d}{dt} + L \frac{d^2}{dt^2} \right) & -L \frac{d^2}{dt^2} \\ -L \frac{d^2}{dt^2} & \left( \frac{1}{C_2} + R_2 \frac{d}{dt} + L \frac{d^2}{dt^2} \right) \end{vmatrix}}$$

Daraus ergibt sich :

$$\left[ \left( \frac{1}{C_1} + R_1 \frac{d}{dt} + L \frac{d^2}{dt^2} \right) \cdot \left( \frac{1}{C_2} + R_2 \frac{d}{dt} + L \frac{d^2}{dt^2} \right) - L^2 \frac{d^4}{dt^4} \right] i_2(t) = L \frac{d^3}{dt^3} u_q(t)$$

$$\left[ \frac{1}{C_1 C_2} + \left( \frac{R_2}{C_1} + \frac{R_1}{C_2} \right) \frac{d}{dt} + \left( \frac{L}{C_1} + \frac{L}{C_2} + R_1 R_2 \right) \frac{d^2}{dt^2} + (R_1 + R_2) L \frac{d^3}{dt^3} \right] i_2(t) = L \frac{d^3}{dt^3} u_q(t)$$

$$\frac{d^3}{dt^3} i_2(t) = -\frac{1}{C_1 C_2 (R_1 + R_2) L} i_2(t) - \frac{\left( \frac{R_2}{C_1} + \frac{R_1}{C_2} \right) d}{(R_1 + R_2) L dt} i_2(t)$$

$$- \frac{\left( \frac{L}{C_1} + \frac{L}{C_2} + R_1 R_2 \right) d^2}{(R_1 + R_2) L dt^2} i_2(t) + \frac{1}{(R_1 + R_2) L^2} \frac{d^3}{dt^3} u_q(t) \quad \Leftrightarrow$$

$$\frac{d^4}{dt^4} i_2(t) = a \cdot i_2(t) + b \frac{d}{dt} i_2(t) + c \frac{d^2}{dt^2} i_2(t) + s(t)$$

In der letzten Zeile wurden zur Vereinfachung Abkürzungen eingeführt.

Diese lineare Dgl. 3. Ordnung für  $i_2(t)$  mit der Anregungsfunktion  $s(t)$  kann in ein System von Differentialgleichungen erster Ordnung umgewandelt werden:

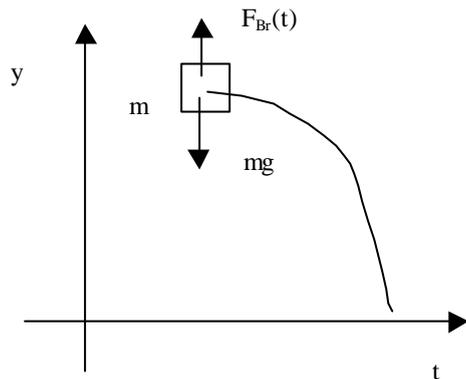
$$\begin{aligned}\frac{d}{dt}i_2(t) &= v_1(t) \\ \frac{d}{dt}v_1(t) &= v_2(t) \quad \left( = \frac{d^2}{dt^2}i_2(t) \right) \\ \frac{d}{dt}v_2(t) &= a \cdot i_2(t) + b \cdot v_1(t) + c \cdot v_2(t) + s(t) \quad \left( = \frac{d^3}{dt^3}i_2(t) \right)\end{aligned}$$

In Vektorschreibweise:

$$\mathbf{y}(t) = \begin{pmatrix} i_2(t) \\ v_1(t) \\ v_2(t) \end{pmatrix} \quad A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a & b & c \end{pmatrix} \quad \mathbf{b}(t) = \begin{pmatrix} 0 \\ 0 \\ s(t) \end{pmatrix}$$
$$\frac{d}{dt}\mathbf{y}(t) = A \cdot \mathbf{y}(t) + \mathbf{b}(t)$$

### 3.5.5 Mondlandung

Mit der Einführung der BASIC-programmierbaren Taschenrechner (1990) war es dem Anwender möglich selbst ein beliebtes Computerspiel auf seinem Taschenrechner zu programmieren: die Simulation einer Apollo Mondlandung [14 ] bis [16 ] . Man mußte den Abstieg der Raumfähre vom Mutterschiff zur Mondoberfläche und eine weiche Landung durch Online Eingabe (über die Tastatur) von Bremschub (Zahlen zwischen 0 und 9) steuern. Dabei stand nur ein begrenzter Treibstoffvorrat zur Verfügung.



Bewegungsgleichung (Abnahme des Gewichtes aufgrund Treibstoffabnahme wird vernachlässigt)

$$m\ddot{y}(t) = -mg + F_{Br}(t)$$

$$\ddot{y}(t) = -g + \frac{F_{Br}(t)}{m}$$

Treibstoffmenge T, Treibstoffabnahme  $dT/dt$

$$\frac{dT}{dt} = -kF_{Br}$$

Umwandlung in Differentialgleichungssystem erster Ordnung durch Einführen der Geschwindigkeit  $v(t) = dy/dt$

$$\frac{dy}{dt} = v(t)$$

$$\frac{dv}{dt} = -g + s(t)$$

$$\frac{dT}{dt} = -k_1 s(t)$$

$$s(t) = \frac{F_{Br}(t)}{m} = \text{Störfunktion}$$

$$k_1 = k \cdot m = \text{Konstante}$$

Zahlenwerte: z.B.  $y(0)=5000 \text{ m} \Leftrightarrow 100$  in geeigneten Einheiten,  $v(0)=0 \text{ m/s}$ ,  $g = 5 \text{ (m/s}^2\text{)}$

Schubbeschleunigung:  $0 < s(t) < 10$ ,

$s=0$  bedeutet freier Fall und  $s = 5$  bedeutet Schweben. ,

$k_1 = 0,1 \text{ kg/min}$

Algorithmus für Euler'sche Näherungsrechnung mit  $\Delta t = h$  (Zeitintervall):

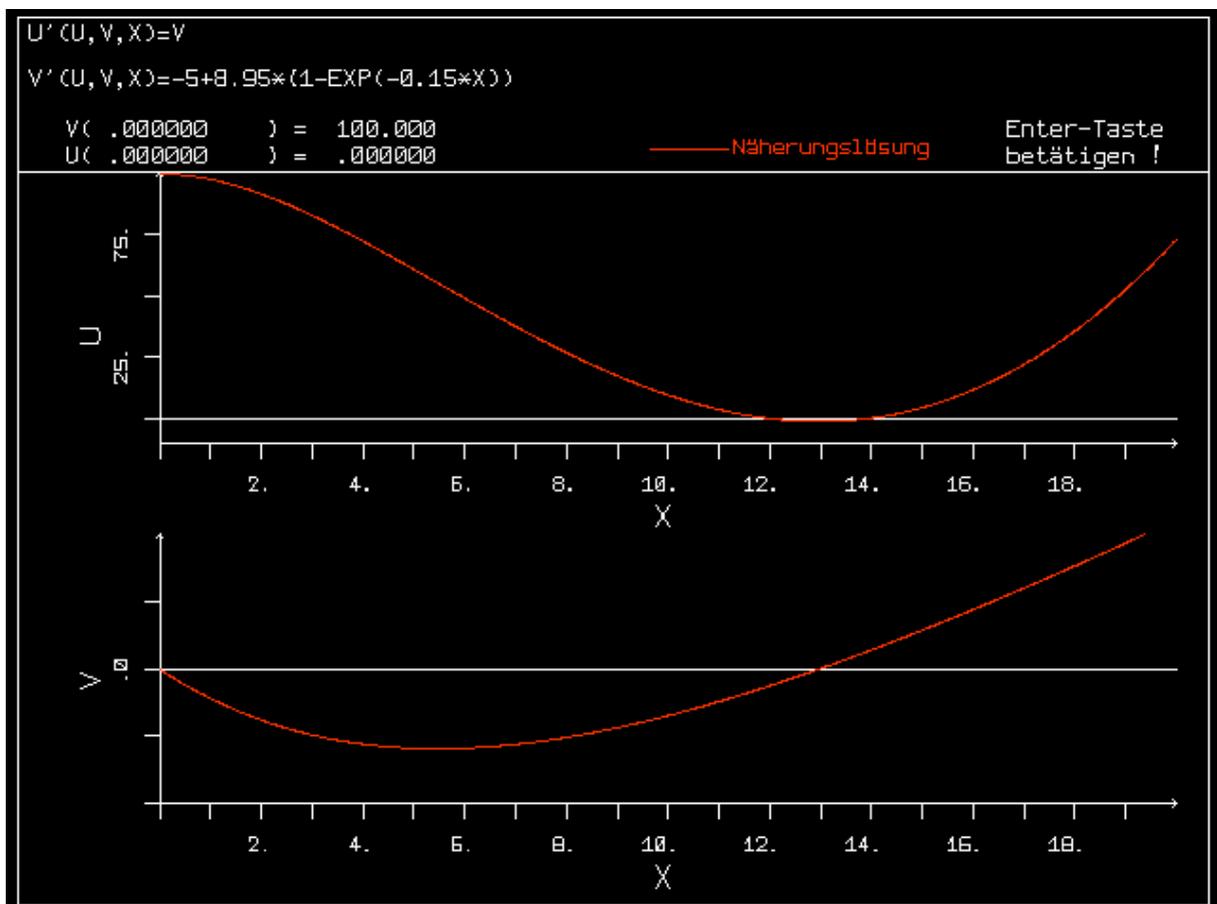
$$y_{n+1} = y_n + h \cdot v_n$$

$$v_{n+1} = v_n + h \cdot (-5 + s_n)$$

$$T_{n+1} = T_n - 0.1 \cdot h \cdot s_n$$

Eine sanfte Landung kann man mit einer geeigneten Bremsfunktion simulieren. Wir lösen das Problem mit MAYA und finden durch Probieren die Bremsfunktion

$s(t) = 8.95(1 - e^{-0.15t})$  mit dem folgenden Abstiegsverlauf :



### 3.5.6 Füchse und Hasen

Das folgende Räuber-Beute-Modell von Lothka-Volterra [ 10], [16] ist auf viele Situationen anzuwenden, wie eben Füchse und Hasen, Angebot und Nachfrage Modelle sowie Jobsituationsmodelle.

Die Füchse und Hasen befinden sich auf einem abgegrenzten Terrain. Ihre Zahlen als Funktionen der Zeit seien  $x(t)$  und  $y(t)$ , wobei wir für  $x(t)$  und  $y(t)$  stetige Funktionen annehmen (Interpolation oder Zahl der Füchse  $x(t)$  /Hasen  $y(t) \gg 1$  ).

Die in einem Zeitabschnitt  $\Delta t$  erfolgten Hasengeburten  $\Delta x$  werden als proportional zu dieser Zeit, der Zahl der Hasen  $x(t)$  und dem Platz, der jedem Hasen zur Verfügung steht, angesetzt.

$$\Delta x_{\text{Geburten}}(t) = k_1 x(t)(x_{\text{max}} - x(t))\Delta t$$

In demselben Zeitabschnitt sterben auch Hasen, weil sie von Füchsen gefressen werden.

Die Zahl der von Füchsen in diesem Zeitabschnitt getöteten Hasen wird als proportional der Häufigkeit, mit der Füchse auf Hasen treffen gesetzt (und ist auch proportional  $\Delta t$ ). Diese Häufigkeit ist proportional der Zahl der Hasen und der Zahl der Füchse, also proportional dem Produkt aus Hasen mal Füchsen.

$$\Delta x_{\text{Sterbefälle}}(t) = k_2 x(t) y(t)\Delta t$$

Die Geburtenrate der Füchse richtet sich nach dem Angebot von Hasen für jeden Fuchs und ist demnach auch proportional dem Produkt aus Hasen mal Füchsen.

$$\Delta y_{\text{Geburten}}(t) = k_3 x(t) y(t)\Delta t$$

Da die Füchse keine Feinde haben, ist für sie die Todesrate konstant. D.h. die Abnahme der Füchse dem Zeitabschnitt  $\Delta t$  ist proportional zur Zahl der Füchse ( und proportional zu  $\Delta t$ ).

$$\Delta y_{\text{Sterbefälle}}(t) = k_4 y(t)\Delta t$$

Dividiert man diese Gleichungen durch  $\Delta t$ , so gelangt man zu den Rategleichungen:

$$x = \text{Zahl der Hasen}$$

$$y = \text{Zahl der Füchse}$$

$$x_{\text{max}} = \text{maximal mögliche Zahl der Hasen}$$

$$\text{Rate} = \text{Geburtenrate} - \text{Sterberate}$$

$$\frac{dx(t)}{dt} = k_1 (x_{\text{max}} - x(t))x(t) - k_2 x(t)y(t)$$

$$\frac{dy(t)}{dt} = k_3 x(t)y(t) - k_4 y(t)$$

oder vereinfacht geschrieben:

$$\frac{dx}{dt} = k_1 (x_{\text{max}} - x)x - k_2 xy$$

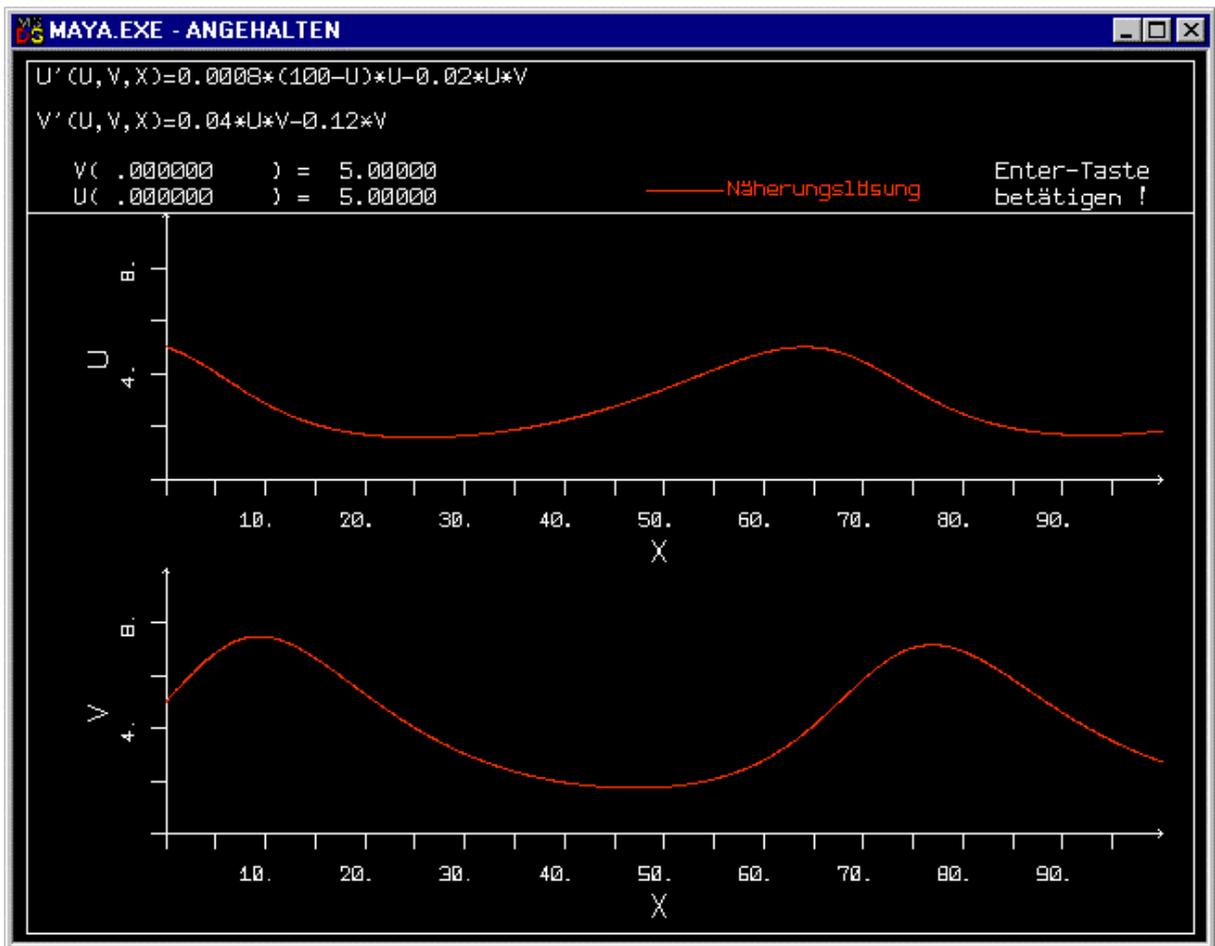
$$\frac{dy}{dt} = k_3 xy - k_4 y$$

Wir lösen das Problem mit MAYA . Durch Probieren findet man Lösungsfunktionen, die mit der gleichen Periode oszillieren. Die Maxima sind gegeneinander zeitversetzt. Bei großem Angebot an Hasen vermehren sich die Füchse stark. Dadurch werden mehr Hasen gefressen und die Zahl der

Hasen geht zurück. Bei verminderter Zahl der Hasen werden die Bedingungen für den einzelnen Hasen immer besser, so daß die Geburtenrate der Hasen steigt, während die Abnahme der Füchse konstant bleibt. Dadurch nehmen die Hasen wieder zu und das Spiel beginnt von neuem.

Für die Darstellung wurden kleine Zahlen verwendet. Dies erreicht man indem man x und y normiert (z.B. durch 100 teilt). Die folgenden Werte von x und y sind also mit 100 zu multiplizieren:

Werte:  $x_{\max} = 100$ ,  $k_1 = 0.0008$ ,  $k_2 = 0.02$ ,  $k_3 = 0.04$ ,  $k_4 = 0.12$  Anfangswerte  $x_0 = 5$ ,  $y_0 = 5$  :



## LITERATURVERZEICHNIS

- (1) Stingl, Peter: Statistik in 10 Stunden, Hanser-Verlag, 1981
- (2) Meyer, Martin: Signalverarbeitung, Analoge und digitale Signale und Filter ,Vieweg, Studienskript, 1998
- (3) Mildenberger Otto: Nachrichtenübertragungstechnik, Grundlagen analog und digital, Vieweg, 1997
- (4) Lathi B.P.: Modern Digital and Analog Communication Systems, Oxford University Press, 1995
  
- (5) Van den Enden A., Verhoeckx N. : Digitale Signalverarbeitung, Vieweg -Verlag, 1990
- (6) Bellanger Maurice: Digital Processing of Signals, Theory and Practice, Teubner, John Wiley, 1989
- (7) Gerdson P., Kröger P.: Digitale Signalverarbeitung in der Nachrichtenübertragung Springer-Verlag,
- (8) Weller Friedrich: Numerische Mathematik für Ingenieure und Naturwissenschaftler, Vieweg-Verlag, 1996
- (9) Kose K., Schröder R., Wieliczka ...: Numerik sehen und verstehen, Vieweg- Verlag, 1992
- (10) Schwarz H. R.: Numerische Mathematik, Teubner-Verlag
- (11) Engeln-Müllges G., Reutter F.: Numerische Mathematik für Ingenieure, BI –Verlag
- (12) Engeln-Müllges G., Reutter F.: Pascal-Programme zur Numerische Mathematik für Ingenieure, BI –Verlag
- (13) Engeln-Müllges G., Uhlig F.: Numerical Algorithms with C, mit CD –ROM , Springer – Verlag, 1996
- (14) Brand B.: Algorithmen zur praktischen Mathematik , 30 Verfahren mit BASIC- und PASCAL-Programmen für Kleincomputer , Oldenbourg
- (15) Menzel Klaus: BASIC in 100 Beispielen , Teubner-Verlag 1983
- (16) Erbs Heinz-Erich: 33 Spiele mit PASCAL und wie man sie (auch in BASIC) programmiert, Teubner 1984
- (17) Mittelbach Henning: Turbo-PASCAL in Beispielen, mit mehr als 100 Programmen, Teubner, 1997