

Über Kurven, Anfangswertaufgaben und Numerik

Manfred Ries*

08. Juli 2013

Inhaltsverzeichnis

Abbildungsverzeichnis	2
1 Kurve in \mathbb{R}^2	2
1.1 Kreis	2
1.1.1 Bogenlänge	2
1.1.2 Fläche	3
1.2 Lissajous-Figur	4
1.2.1 Bogenlänge	4
1.2.2 Fläche	4
1.2.3 Fläche mit Schulmathematik	5
1.3 Weitere Lissajous-Figuren	5
1.3.1 1:3	5
1.3.2 3:4	5
2 Differentialgleichung	6
3 Euler	7
4 Euler rückwärts	9
5 Trapezregel	9
6 Gauss-Seidel	11
7 Anhang	11
7.1 scilab-xcos	11
7.2 matlab-simulink	11
7.3 matlab	13
7.4 python	13
7.4.1 euler	13
7.4.2 odeint	13

*ries@uni-trier.de

Programmverzeichnis

1	Mathematica	6
2	Eulermethode	8
3	Eulergaussmethode	11
4	matlab	14
5	Eulermethode in matlab	14
6	Eulermethode in python	15
7	Anfangswertaufgabe in python	16

Abbildungsverzeichnis

1	Kreis als Kurve	3
2	Lissajous mit Parametern 1:2	4
3	Lissajous mit Parametern 1:3	5
4	Lissajous mit Parametern 3:4	6
5	Euler	8
6	Euler rückwärts	9
7	Trapez	10
8	Phasenbild Gauss	12
9	Lissajous mit $x\cos$	12
10	Lissajous mit <i>simulink</i>	13

1 Kurve in \mathbb{R}^2

1.1 Kreis

Die Abbildung

$$u : \mathbb{R} \supset T \rightarrow \mathbb{R}^2, t \mapsto \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

beschreibt eine Kurve in \mathbb{R}^2 .

Für das Beispiel $x(t) = \cos t$, $y(t) = \sin t$ und $T = [0, 2\pi]$ ergibt dies den (geschlossenen) Rand des Einheitskreises; die Kurve startet bei $u(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ und läuft entgegen dem Uhrzeigersinn, mathematisch positiv, auf dem Einheitskreis bis für den Parameter $t = 2\pi$ wieder $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ erreicht wird.

1.1.1 Bogenlänge

Der Umfang der Kurve bestimmt sich aus der Bogenlänge

$$\int_0^{2\pi} \|u'(t)\| dt = \int_0^{2\pi} \left\| \begin{pmatrix} \cos' t \\ \sin' t \end{pmatrix} \right\| dt =$$

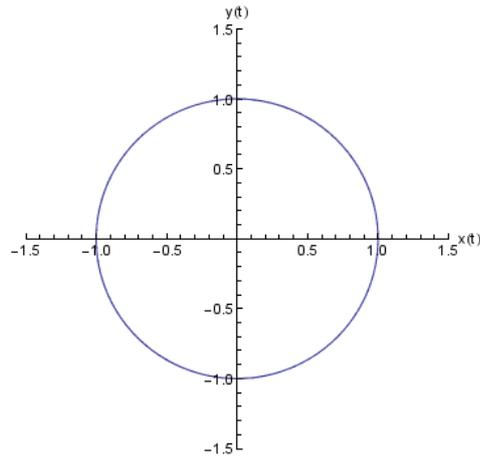


Abbildung 1: Kreis als Kurve

$$\int_0^{2\pi} \left\| \begin{pmatrix} -\sin t \\ \cos t \end{pmatrix} \right\| dt = \int_0^{2\pi} \sqrt{(-\sin t)^2 + (\cos t)^2} dt = \int_0^{2\pi} 1 dt = [t]_0^{2\pi} = 2\pi$$

1.1.2 Fläche

Um die Fläche des durch die Kurve $\gamma(t)$ berandeten⁽ⁱ⁾ Gebietes Ω zu bestimmen, soll der Green'sche Satz [2] verwendet werden:

$$\int_{\gamma} Ldx + Mdy = \iint_{\Omega} \left(\frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} \right) dxdy$$

Damit $\frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} = 1$ gilt, also auf der rechten Seite die Fläche des Gebietes steht, kann man das Differential z.B. als $F = \frac{1}{2}(-ydx + xdy)$ wählen⁽ⁱⁱ⁾

Für das Kurvenintegral (2. Art) gilt:

$$\int_{\gamma} Fdt = \int_{\gamma} F_1dx + F_2dy = \int_{\gamma} F(\gamma(t)) \gamma'(t) dt.$$

Im Fall des Kreises gilt mit $F = \frac{1}{2}(-ydx + xdy)$, also

$$F = \frac{1}{2} \begin{pmatrix} -y \\ x \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -\sin t \\ \cos t \end{pmatrix} \quad \text{und} \quad \gamma'(t) = \begin{pmatrix} \cos' t \\ \sin' t \end{pmatrix} = \begin{pmatrix} -\sin t \\ \cos t \end{pmatrix}.$$

Ausgerechnet

$$\int_{\gamma} F(\gamma(t)) \gamma'(t) dt = \frac{1}{2} \int_0^{2\pi} (-\sin t) \cdot (-\sin t) + \cos t \cdot \cos t dt =$$

⁽ⁱ⁾Orientierung mathematisch positiv

⁽ⁱⁱ⁾Alternativ ist auch xdy oder $-ydx$ möglich.

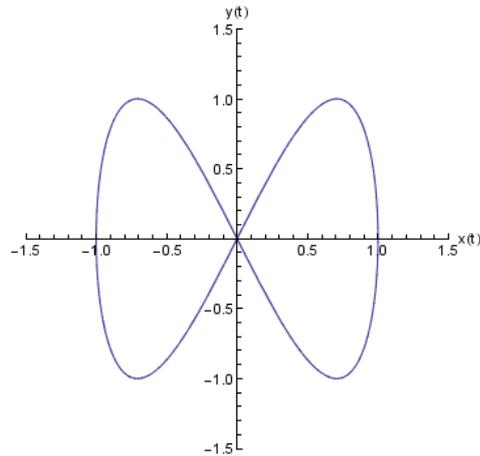


Abbildung 2: Lissajous mit Parametern 1:2

$$\frac{1}{2} \int_0^{2\pi} \sin^2 t + \cos^2 t \, dt = \frac{1}{2} \int_0^{2\pi} 1 \, dt = \frac{1}{2} [t]_0^{2\pi} = \pi.$$

1.2 Lissajous-Figur

Für $x(t) = \cos t$, $y(t) = \sin 2t$ und $T = [0, 2\pi]$ ergibt sich die Lissajousfigur [1], aus Abbildung 2. Diese entsteht auf dem Oszilloskop, wenn die X- und Y-Spannung wie hier eingegeben wird. Vgl. dazu 7.1 auf Seite 11.

1.2.1 Bogenlänge

$$\int_0^{2\pi} \left\| \begin{pmatrix} \cos' t \\ \sin' 2t \end{pmatrix} \right\| dt = \int_0^{2\pi} \left\| \begin{pmatrix} -\sin t \\ 2 \cos 2t \end{pmatrix} \right\| dt = \int_0^{2\pi} \sqrt{-\sin^2 t + 4 \cos^2 2t} \, dt \doteq 9.42942^{(iii)}$$

1.2.2 Fläche

Mit der Parametrisierung $\gamma : t \mapsto \begin{pmatrix} \cos t \\ \sin 2t \end{pmatrix}$ ergibt sich für die (mathematisch negativ umlaufene) linke Halb-Fläche mit $t \in [\frac{\pi}{2}, \frac{3\pi}{2}]$

$$F = \frac{1}{2} \begin{pmatrix} -y \\ x \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -\sin 2t \\ \cos t \end{pmatrix} \quad \text{und} \quad \gamma'(t) = \begin{pmatrix} \cos' t \\ \sin' 2t \end{pmatrix} = \begin{pmatrix} -\sin t \\ 2 \cos 2t \end{pmatrix}.$$

Zusammen

$$\int_{\gamma} F(\gamma(t)) \gamma'(t) \, dt = \frac{1}{2} \int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} (-\sin t) \cdot (-\sin 2t) + \cos t \cdot 2 \cos 2t \, dt = \frac{1}{2} \left(-\frac{8}{3}\right) = -\frac{4}{3}$$

⁽ⁱⁱⁱ⁾ Mathematica kann das Integral nicht exakt ausrechnen.

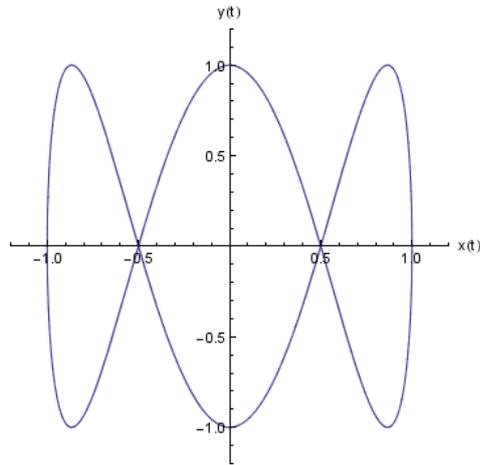


Abbildung 3: Lissajous mit Parametern 1:3

(iv). Der (vorzeichenlose) Flächeninhalt der gesamten *Fliege* beträgt damit $\frac{8}{3}$.

1.2.3 Fläche mit Schulmathematik

Der rechte obere Bogen der Kurve, also $t \in [0, \frac{\pi}{2}]$ und damit $x \in [0, 1]$, kann durch Umformulierung als Integration einer Funktion $x \mapsto f(x), x \in [0, 1]$ geschrieben werden:

Mit $x(t) = \cos t, t = \arccos x$ (v) ergibt sich $y(t) = \sin 2t = \sin(2 \arccos x)$. Das Integral beträgt $\int_0^1 \sin(2 \arccos x) dx = \frac{2}{3}$ (vi)

1.3 Weitere Lissajous-Figuren

1.3.1 1:3

$$u : [0, 2\pi] \rightarrow \mathbb{R}^2, t \mapsto \begin{pmatrix} \cos(1 \cdot t) \\ \sin(3 \cdot t) \end{pmatrix}, \text{Abbildung 3}$$

1.3.2 3:4

$$u : [0, 2\pi] \rightarrow \mathbb{R}^2, t \mapsto \begin{pmatrix} \cos(3 \cdot t) \\ \sin(4 \cdot t) \end{pmatrix}, \text{Abbildung 4}$$

(iv) $\int \sin t \cdot \sin 2t dt = \int \sin t \cdot 2 \sin t \cos t dt = \frac{2}{3} \sin^3 t + C; \int \cos t \cdot \cos 2t dt = \cos^3 t - \sin^2 t \cos t + C$

(v) Mathematica: $\arccos = \text{ArcCos}$

(vi) Mit der *Bauerregel*, $\int_0^1 \sin(2 \arccos(x)) dx, x = \cos(t), \frac{dx}{dt} = -\sin(t), dx = -\sin(t) dt$, ergibt sich $\int_{\arccos(0)}^{\arccos(1)} \sin(2 \arccos(\cos(t))) (-\sin(t)) dt = -\int_{\arccos(0)}^{\arccos(1)} \sin(2t) (\sin(t)) dt = -\int_{\frac{\pi}{2}}^0 \sin(2t) (\sin(t)) dt = \int_0^{\frac{\pi}{2}} \sin(2t) (\sin(t)) dt$. Mit $\sin 2t = 2 \sin t \cos t$, ergibt sich $\int_0^{\frac{\pi}{2}} 2 \sin^2(t) (\cos(t)) dt$. Die *Kettenregel* (invers) ergibt: $2 \int_0^{\frac{\pi}{2}} \frac{1}{3} \sin(3t) dt = 2 \frac{1}{3} \sin(3t) \Big|_0^{\frac{\pi}{2}} = 2 \frac{1}{3} = \frac{2}{3}$

Die *Substitutionsformel* lautet allgemein $\int_a^b f(x) dx = \int_{\phi^{-1}(a)}^{\phi^{-1}(b)} f(\phi(t)) \phi'(t) dt$. Mit $\phi : t \mapsto x, x = \phi(t)$, hier $x = \phi(t) = \cos(t)$ ergibt sich das gleiche Ergebnis wie mit der *Bauernregel*.

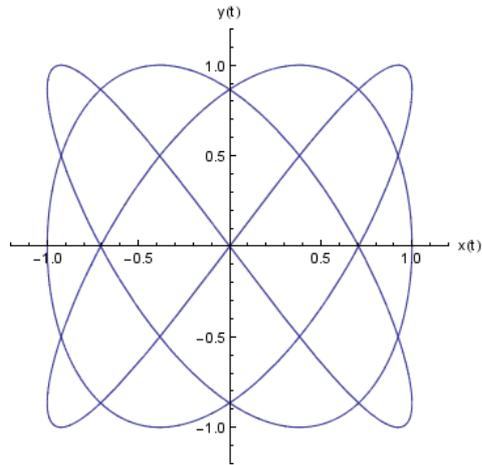


Abbildung 4: Lissajous mit Parametern 3:4

```

1 system = {x'[t] == -y[t], y'[t] == x[t], x[0] == 1, y[0] == 0};
2 lsg = NDSolve[system, {x, y}, {t, 0, 2 Pi}];
3 Plot[{Evaluate[x[t] /. lsg], Evaluate[y[t] /. lsg]},
4   {t, 0, 2 Pi},
5   PlotStyle -> {Red, Green}];
6 ParametricPlot[Evaluate[{x[t], y[t]} /. lsg],
7   {t, 0, 2 Pi},
8   AxesLabel -> {"x(t)", "y(t)"}, PlotRange -> 1.2]

```

Listing 1: Mathematica

2 Differentialgleichung

Die Funktionen $x(t) = \cos t$ und $y(t) = \sin t$ erfüllen die Differentialgleichungen

$$x'(t) = \cos' t = -\sin t = -y(t)$$

$$y'(t) = \sin' t = \cos t = x(t)$$

Zusammen mit den Anfangsdaten ergibt sich die Anfangswertaufgabe

$$x'(t) = -y(t)$$

$$y'(t) = x(t)$$

$$x(0) = 1$$

$$y(0) = 0$$

Als lineare Aufgabe kann das System als

$$u'(t) = Au(t), \quad A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

geschrieben werden. Die Eigenwerte sind beide komplex, $\lambda_1 = i$ und $\lambda_2 = -i$. Die dazu gehörenden Eigenvektoren sind $v_1 = \begin{pmatrix} i \\ 1 \end{pmatrix}$ und $v_2 = \begin{pmatrix} -i \\ 1 \end{pmatrix}$. Damit ist $\left\{ \begin{pmatrix} i \\ 1 \end{pmatrix} e^{it}, \begin{pmatrix} -i \\ 1 \end{pmatrix} e^{-it} \right\}$ ein Fundamentalsystem. Mit $e^{it} = \cos t + i \sin t$ ergibt sich

$$\alpha \begin{pmatrix} i \\ 1 \end{pmatrix} (\cos t + i \sin t) + \beta \begin{pmatrix} -i \\ 1 \end{pmatrix} (\cos(-t) + i \sin(-t)) = \begin{pmatrix} \alpha i \cos t - \alpha \sin t - \beta i \cos t - \beta \sin t \\ \alpha \cos t + \alpha i \sin t + \beta \cos t - \beta i \sin t \end{pmatrix}.$$

Aus den Anfangsbedingungen $x(0) = 1$ und $y(0) = 0$ folgt für die Koeffizienten α und β aus dem Gleichungssystem

$$\begin{bmatrix} i & -i \\ 1 & 1 \end{bmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

die Lösung mit $\alpha = \frac{-i}{2}$ und $\beta = \frac{i}{2}$. Damit ist

$$u(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \cos t + \frac{1}{2} \cos t \\ \frac{1}{2} \sin t + \frac{1}{2} \sin t \end{pmatrix} = \begin{pmatrix} \cos t \\ \sin t \end{pmatrix}$$

Anmerkung : `NDSolve`: Da Mathematica zur Lösung der Anfangswertaufgabe ein Numerisches Verfahren benutzt, zeigen die Ergebnisse nur für entsprechend kleine Schrittweiten h und Zeiten kleine Rundungsfehler. Für z.B. $t_{ende} = 100$ oder 1000 wird das Phasenbild immer *fetter*, bis schließlich die numerische Lösung ganz falsch wird (Laut Dokumentation nutzt Mathematica als Standard-Methode eine *predictor-corrector Adams method with orders 1 through 12*)

Vgl. dazu auch Abschnitt 5 auf Seite 10.

3 Euler

Die Diskretisierung der Ordnung 1 $u'(t) \doteq \frac{u(t+h)-u(t)}{h}$ (mit Schrittweite h) führt auf die Eulermethode zur Lösung der Anfangswertaufgabe:

$$\begin{aligned} x(t+h) &= x(t) - hy(t) \\ y(t+h) &= y(t) + hx(t) \end{aligned}$$

D.h. in jedem Zeitschritt $t \rightarrow t+h$ werden gleichzeitig die neuen Werte aus den alten berechnet. (Siehe auch Abschnitt 6 auf Seite 11.)

Mit einer Schrittweite von $h = \frac{2\pi}{n}, n = 100$ sammelt Mathematica die berechneten x - und y -Werte in den Listen X und Y . Dabei werden aus Zeit- und Speicherplatzgründen die neuen Werte nicht *exakt*, sondern *numerisch* (`N[]`) bestimmt.

Der erste Plot-Befehl zeigt beide Funktionen einzeln, der zweite das Phasendiagramm: Mit wachsendem Parameter t verlassen die berechneten Wertepaare den Rand des Einheitskreises nach außen. Aus der Matrixdarstellung

$$\begin{pmatrix} x(t+h) \\ y(t+h) \end{pmatrix} \begin{bmatrix} 1 & -h \\ h & 1 \end{bmatrix} = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

können die beiden Eigenwerte zu $\lambda_1 = 1 - ih$ und $\lambda_2 = 1 + ih$ berechnet werden. Das Produkt liefert die Determinante $1 + h^2$, was erklärt, warum die berechneten Punkte nach außen wandern.

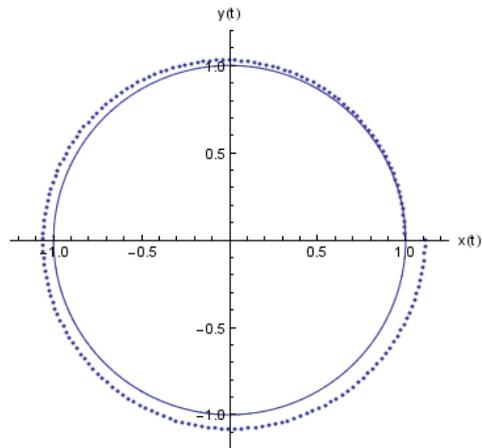


Abbildung 5: Euler

```

1 n = 100; h = 2 Pi/n;
2 X = {}; Y = {};
3 x = 1 ; y = 0;
4 Do[
5   xneu = N [ x - h * y ];
6   yneu = N [ y + h * x ];
7   AppendTo[X, xneu]; AppendTo[Y, yneu];
8   x = xneu; y = yneu
9   , {i, 1, n}
10  ]
11
12 euler = ListPlot[{X,Y}]
13
14 eulerphase = ListPlot[Transpose[{X,Y}], AspectRatio->Automatic];

```

Listing 2: Eulermethode

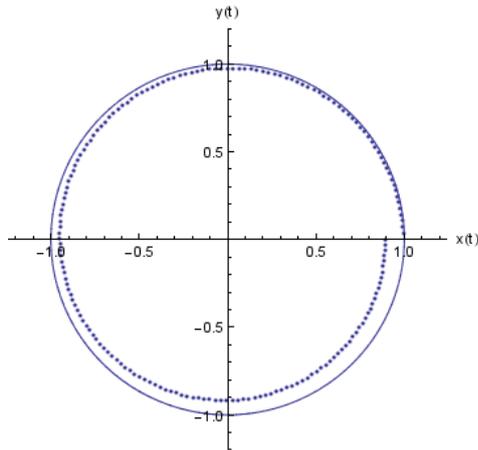


Abbildung 6: Euler rückwärts

4 Euler rückwärts

$$\begin{aligned}x(t+h) &= x(t) - hy(t+h) \\ y(t+h) &= y(t) + hx(t+h)\end{aligned}$$

Umgeformt ergibt dies

$$\begin{aligned}x(t+h) + hy(t+h) &= x(t) \\ -hx(t+h) + y(t+h) &= y(t)\end{aligned}$$

also ein Lineares Gleichungssystem für $\begin{pmatrix} x(t+h) \\ y(t+h) \end{pmatrix}$ mit Matrix $\begin{bmatrix} 1 & h \\ -h & 1 \end{bmatrix}$ und rechter Seite $\begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$.

Die Inverse Matrix lautet $\frac{1}{1+h^2} \begin{bmatrix} 1 & -h \\ h & 1 \end{bmatrix}$, so dass die Rechenvorschrift nun explizit

$$\begin{aligned}x(t+h) &= \frac{1}{1+h^2} (x(t) - hy(t)) \\ y(t+h) &= \frac{1}{1+h^2} (hx(t) + y(t))\end{aligned}$$

lautet. Wie in Abschnitt 3, Seite 7, erklärt auch hier die Determinante der Matrix, dass die berechneten Punkte bei dieser Methode nach innen driften: die Determinante beträgt $(1+h^2)^{-1} = \frac{1}{1+h^2}$.

5 Trapezregel

$$\begin{aligned}x(t+h) &= x(t) - \frac{1}{2}h(y(t) + y(t+h)) \\ y(t+h) &= y(t) + \frac{1}{2}h(x(t) + x(t+h))\end{aligned}$$

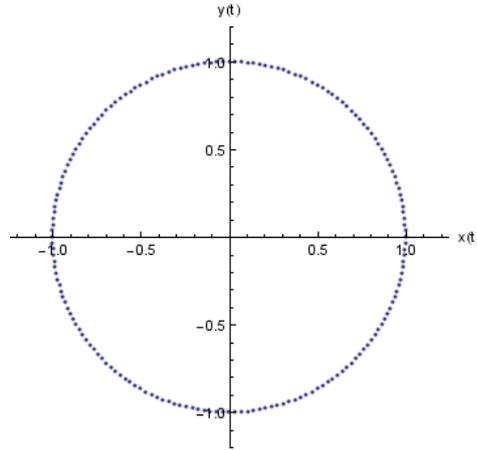


Abbildung 7: Trapez

Auch diese implizite Formel kann als Lineares Gleichungssystem umgeschrieben werden

$$\begin{aligned} x(t+h) + \frac{1}{2}hy(t+h) &= x(t) - \frac{1}{2}hy(t) \\ -\frac{1}{2}hx(t+h) + y(t+h) &= y(t) + \frac{1}{2}hx(t) \end{aligned}$$

mit der Matrix $\begin{bmatrix} 1 & \frac{1}{2}h \\ -\frac{1}{2}h & 1 \end{bmatrix}$ und der rechten Seite $\begin{pmatrix} x(t) - \frac{1}{2}hy(t) \\ y(t) + \frac{1}{2}hx(t) \end{pmatrix}$.

Mit der Inversen Matrix $\frac{1}{1+\frac{1}{4}h^2} \begin{bmatrix} 1 & -\frac{1}{2}h \\ \frac{1}{2}h & 1 \end{bmatrix}$ kann die Rechnung nun explizit

$$\begin{aligned} x(t+h) &= \frac{1}{1+\frac{1}{4}h^2} \left(x(t) - \frac{1}{2}hy(t) - \frac{1}{2}hy(t) - \frac{1}{4}h^2x(t) \right) \\ y(t+h) &= \frac{1}{1+\frac{1}{4}h^2} \left(\frac{1}{2}hx(t) - \frac{1}{4}h^2y(t) + y(t) + \frac{1}{2}hx(t) \right) \end{aligned}$$

oder

$$\begin{aligned} x(t+h) &= \frac{1}{1+\frac{1}{4}h^2} \left(x(t) - hy(t) - \frac{1}{4}h^2x(t) \right) \\ y(t+h) &= \frac{1}{1+\frac{1}{4}h^2} \left(y(t) + hx(t) - \frac{1}{4}h^2y(t) \right) \end{aligned}$$

formuliert werden.

Der Term $\frac{1}{4}h^2$ kann natürlich auch als $(\frac{h}{2})^2$ - passend zur Konvergenzordnung 2 der Trapezmethode - gelesen werden.

Wie das Phasendiagramm 7 zeigt, sind die berechneten Werte für $x(t)$ und $y(t)$ (innerhalb der Rechengenauigkeit) exakt. Der Grund dafür ist, dass die Trapezregel, die zur Integration der rechten Seite $f(z, u(t))$ (der Differentialgleichung) genutzt wird, Polynome bis zum Grad

```

1 Do[
2   xneu = N [ x - h * y ];
3   yneu = N [ y + h * xneu ]; (* !!! *)
4   AppendTo[X, xneu]; AppendTo[Y, yneu];
5   x = xneu; y = yneu
6   , {i, 1, n}
7 ]

```

Listing 3: Eulergaussmethode

1 exakt integriert:

$$\begin{aligned}
 u'(t) &= f(t, u(t)) \\
 \int_t^{t+h} u'(t) dt &= \int_t^{t+h} f(t, u(t)) dt + C \\
 [u(t)]_t^{t+h} &= \frac{1}{2}h(f(t, u(t)) + f(t+h, u(t+h))) + C \\
 u(t+h) &= u(t) + \frac{1}{2}h(f(t, u(t)) + f(t+h, u(t+h))) + 0
 \end{aligned}$$

In dem Beispiel sind beide Komponenten von $u(t)$ lineare Funktionen!
(Und die Determinante der Matrix 5 berechnet sich als 1!).

6 Gauss-Seidel

Im Gegensatz zur Eulermethode, siehe 3, wird hier der neue y -Wert mit Hilfe des inzwischen schon bekannten neuen x -Wertes berechnet! Dies ergibt deutlich besserer Resultate (siehe Plot). (vii)

Dieser sofortige *Update* erinnert an die Updatevarianten beim Jacobi- und und Gauß-Seidel Verfahren zur interativen Lösung von Linearen Gleichungssystemen.

7 Anhang

7.1 scilab-xcos

Gibt man im Programm `scilab` den Befehl `xcos` ein, so kann man interaktiv die Schaltung wie in 9 erstellen: Die beiden `Sinusioid`-Generatoren liefern den Input, den das `CScopXY` zeigen soll; als Zeitgeber dient `Clock_c`. Mit `strg-B` (oder rechte Maustaste) werden die Parameter der Generatoren definiert.

7.2 matlab-simulink

Gibt man im Programm `matlab` den Befehl `simulink` ein, so kann man ebensso wie in `xcos` die Schaltung 10 erstellen.

(vii) Die Begründung dafür: Hamiltonian Mechanik [3]

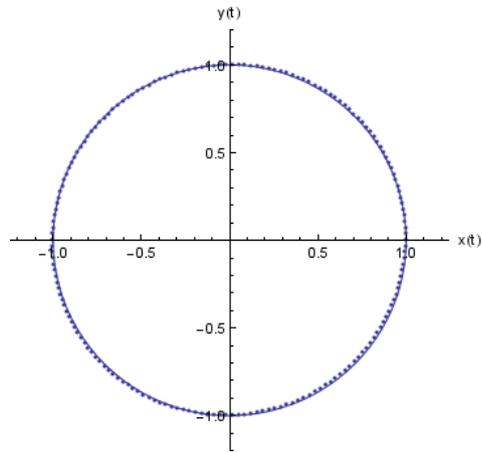


Abbildung 8: Phasenbild Gauss

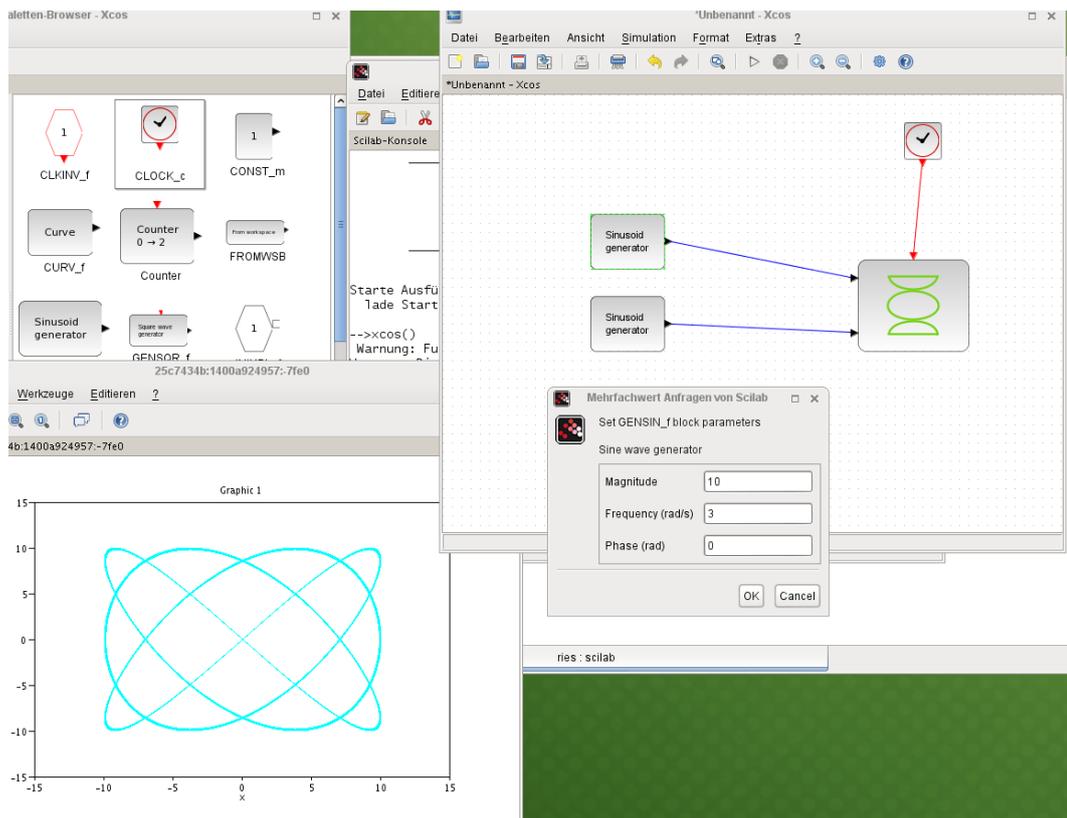


Abbildung 9: Lissajous mit $x\cos$

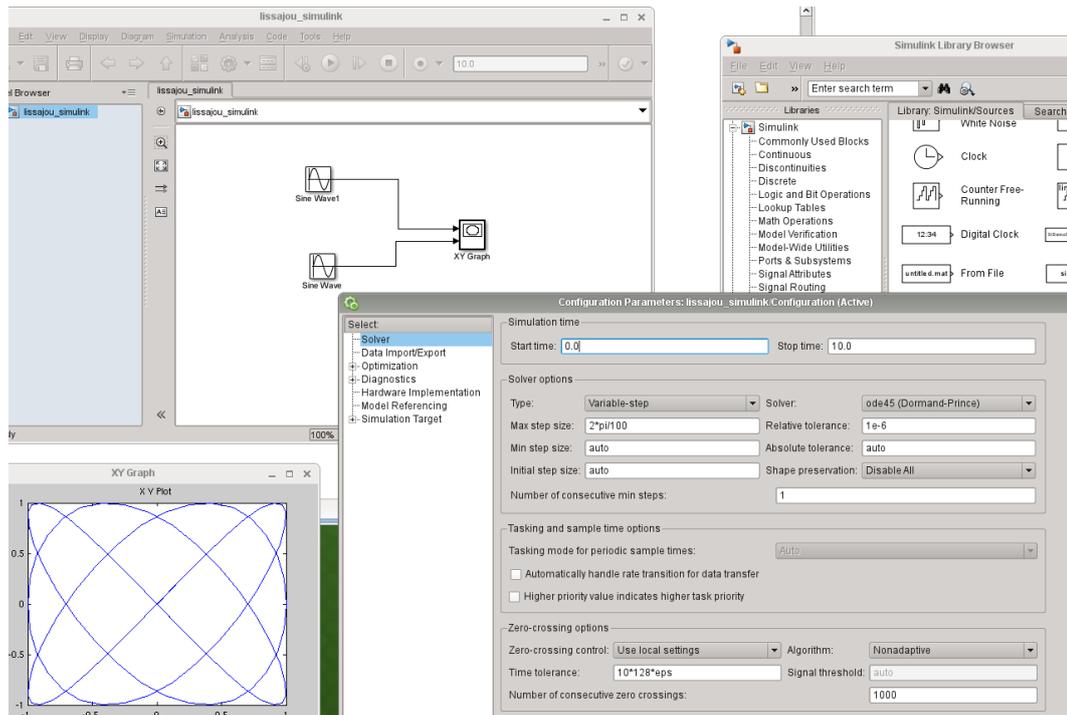


Abbildung 10: Lissajous mit *simulink*

Achtung: Die Schrittweite sollte angepasst werden, hier auf $0.06 \approx 2\pi/100$.

7.3 matlab

Hier die entsprechenden `matlab`-Programme. Die Ergebnisse sind (natürlich) äquivalent. Auch die in `matlab` verwendeten Löser wie `ode23` und `ode45` zeigen für kleine Schrittweiten h und längere Zeit Fehler in den Phasenbildern.

7.4 python

7.4.1 euler

Als Beispiel findet sich hier die Euler-Methode als `python`-Programm. Die Grafik wird mit dem `matplotlib`-Paket^(viii) ähnlich wie in `matlab` erstellt.

7.4.2 odeint

Die Python-Erweiterung `scipy` stellt die Routine `odeint`^(ix) zur Verfügung. Der prinzipielle Aufruf ist ähnlich dem von `matlab`.

^(viii) http://matplotlib.org/users/pyplot_tutorial.html

^(ix) Ein etwas längeres Beispiel zu den Möglichkeiten von `scipy` zeigt die *Räuber-Beute*-Modellierung (Lotka-Volterra) bei <http://wiki.scipy.org/Cookbook/LotkaVolterraTutorial>

```

1 function kreis_ode()
2 pi = 4*atan(1)
3 [t,y] = ode23( @mydy,[0,2*pi],[1 0] ); %I ca. [0,20]
4                                     % nach innen
5 %[t,y] = ode45( @mydy,[0,2*pi],[1 0] ); % I ca. [0,200]
6                                     % nach innen
7 plot( y(:,1),y(:,2), '*' )
8 end
9
10 function dy = mydy(t,y)
11 dy = zeros(2,1); % system mit 2
12 dy(1) = -y(2) ;
13 dy(2) = y(1) ;
14 end

```

Listing 4: matlab

```

1 function euler()
2 clf
3 %axis auto
4 axis( 1.2*[-1 1 -1 1] )
5 axis square
6 hold on
7 n = 100;
8 h = 2 * 4*atan(1) /n % 2 pi/n
9 x = 1; y = 0;
10 for i = 1 : n
11     xneu = x - h*y;
12     yneu = y + h*x;
13     x = xneu; y = yneu;
14     X(i) = x; Y(i) = y;
15 end % for
16 plot(X,Y, '*')
17 end

```

Listing 5: Eulermethode in matlab

```

1 #!/usr/bin/python
2 # -*- coding: utf-8 -*- # utf-8 Zeichen; als erster Kommentar
3
4 import matplotlib.pyplot as plt
5 import numpy as np
6
7 plt.axis( [-1.2, 1.2, -1.2, 1.2] )
8 plt.axes().set_aspect('equal') # x-y gleich skaliert
9 n = 100
10 h = 2* 4*np.arctan(1) /n # 2 pi/n
11 x = 1
12 y = 0
13 XX = []; YY = []
14 for i in range(0, n): # geht von 0 bis n-1, 0..99
15     xneu = x - h*y; # Tab !!
16     yneu = y + h*x;
17     x = xneu; y = yneu;
18     XX.append(x); YY.append(y)
19 # end for
20 plt.plot(XX,YY, '*')
21
22 plt.show() # erst dann wird plot gezeigt

```

Listing 6: Eulermethode in python

```

1 #!/usr/bin/python
2 # -*- coding: utf-8 -*- # utf-8 Zeichen; als erster Kommentar
3
4 import matplotlib.pyplot as plt
5 import numpy as np
6 from scipy import integrate
7
8 def dU_dt(U, t):
9     return np.array( [-U[1] , U[0] ] )
10
11 plt.axis( [-1.2, 1.2, -1.2, 1.2] )
12 plt.axes().set_aspect('equal') # x-y gleich skaliert
13
14 t = np.linspace(0, 6.283185, 100)
15 U0 = np.array([1,0])
16
17 U = integrate.odeint( dU_dt, U0, t)
18
19 # end for
20 plt.plot(U[:,0], U[:,1])
21
22 plt.show() # erst dann wird plot gezeigt

```

Listing 7: Anfangswertaufgabe in python

8 Literatur

- [1] Lissajous Figuren, <http://de.wikipedia.org/wiki/Lissajous-Figur>, online gelesen 08.07.2013
- [2] Murray R. Spiegel, Advanced Calculus, Seite 204, Probleme 8 und 9, Schaum's Outline Series, McGraw-Hill, 1963, ISBN 07-060229-S
- [3] Hamiltonian Mechanics, Wikipedia, https://en.wikipedia.org/wiki/Hamiltonian_mechanics, online gelesen 08.07.2013