Aufgaben zur Vorlesung
Numerik II

Wintersemester 2012/13
Übungsblatt 8

W.–J. Beyn
D. Otten

Abgabe: Mittwoch, 05.12.2012, vor Beginn der Übung

Übung: Mi. 12:15–13:45, V5-148

Aufgabe 22: [Probleme auf unbeschränkten Intervallen]

Man betrachte für $a \in \mathbb{R}$ das explizite und implizite Eulerverfahren für das 2-dimensionale AWP

$$\begin{pmatrix} u_1'(t) \\ u_2'(t) \end{pmatrix} = \begin{pmatrix} -u_2(t) \\ u_1(t) \end{pmatrix}, \; t > 0, \quad \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} a \\ 0 \end{pmatrix}. \tag{1}$$

Das Gitter Ω_h sei in beiden Fällen für $h > 0$ durch

$$\Omega_h = \{t_j = jh | j \in \mathbb{N}_0\}$$

gegeben.

a) Bestimmen Sie die exakte Lösung von (1) und ihr Verhalten für $t \to \infty$.

b) Zeigen Sie, dass die Lösung u_h des expliziten Eulerverfahrens $\lim_{j\to\infty} \|u_h(t_j)\|_2 = \infty$ erfüllt, die des impliziten aber $\lim_{j\to\infty} \|u_h(t_j)\|_2 = 0$.

(6 Punkte)

Aufgabe 23: [Asymptotische Entwicklung des Konvergenzfehlers]

Beweisen Sie den Satz über die Entwicklung des Konvergenzfehlers (Kap. II, Satz 2.17) für den Spezialfall $q = p + 1$ und $n = 1$. Leiten Sie dazu zuerst eine explizite Darstellung der Anfangswertaufgabe für die Koeffizientenfunktion $e_1(t)$ her und gehen Sie dann wie im Fall $q = p$ vor.

(6 Punkte)

Aufgabe 24: [Implementation der adaptiven Schrittweitensteuerung]

a) Lösen Sie mit dem (expliziten) klassischen Runge–Kutta–Verfahren und den Schrittweiten $h = 10^{-i}$, $i = 1, 2, 3$ die Anfangswertaufgabe (das sogenannte Lorenz-System),

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}' = \begin{pmatrix} \sigma(y - x) \\ \lambda x - y - xz \\ -\mu z + xy \end{pmatrix}, \quad \begin{pmatrix} x(0) \\ y(0) \\ z(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \\ 9 \end{pmatrix}$$

für die Parameterwerte $\sigma = 10$, $\lambda = 28$, $\mu = \frac{8}{3}$ und $0 \leqslant t \leqslant 50$.
Geben Sie die numerischen Näherungen nur zu den Zeiten $t_k = 5k$, $k = 0, \ldots, 10$ in einer Tabelle aus und vergleichen Sie die Ergebnisse für die verschiedenen Schrittweiten.

b) Steuern Sie in einer zweiten Rechnung das klassische Runge–Kutta–Verfahren mit den Daten

$$\kappa = 10, \; h_0 = 0.2, \; h_{\max} = 0.5, \; \mathrm{tol} = 10^{-4}, \, 10^{-6}, \, 10^{-8}$$

und geben Sie ebenfalls die Näherungswerte für die Zeitpunkte $t_k = 5k$ aus. Man verwende dazu jeweils beim Überschreiten eines solchen Zeitpunktes einen gesonderten Runge–Kutta–Schritt mit geeigneter Schrittweite. Dieser Zwischenschritt dient nur der Berechnung eines Näherungswertes und soll die Steuerung ansonsten unbeeinflusst lassen.
Zeichnen Sie die gesteuerte Lösung zu $\mathrm{tol} = 10^{-6}$ und die beste äquidistante Lösung zu $h = 10^{-3}$ in ein (t, x) bzw. ein (t, z) Diagramm.

(6 Punkte)

Numerik II (WS 12/13)
Übungsblatt 08
Lösungen

Aufgabe 22:

$$\begin{pmatrix} u_1'(t) \\ u_2'(t) \end{pmatrix} = \begin{pmatrix} -u_2(t) \\ u_1(t) \end{pmatrix}, \quad t>0, \qquad \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} a \\ 0 \end{pmatrix}, \quad a \in \mathbb{R}$$

Betrachte dazu das explizite & implizite Eulerverfahren mit (äquidistantem) Gitter

$$\Omega_h = \{ t_j = j \cdot h \mid j \in \mathbb{N}_0 \}, \quad h>0$$

a) Bestimmen Sie die exakte Lösung und ihr Verhalten für $t \to \infty$

b) Zeigen Sie, dass die Lösung u_h
- des expliziten Eulerverfahrens $\lim\limits_{j\to\infty} \|u_h(t_j)\|_2 = \infty$

und
- des impliziten Eulerverfahrens $\lim\limits_{j\to\infty} \|u_h(t_j)\|_2 = 0$

erfüllt.

Lösung:

zu (a): • Analytische Lösung:

- reelle Version: $A := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

$$\begin{pmatrix} u_1' \\ u_2' \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} a \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = e^{At} \cdot \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix} \begin{pmatrix} a \\ 0 \end{pmatrix} = \begin{pmatrix} a\cos t \\ a \sin t \end{pmatrix}$$

$$\exp\begin{pmatrix} x & -y \\ y & x \end{pmatrix} = e^x \cdot \begin{pmatrix} \cos y & -\sin y \\ \sin y & \cos y \end{pmatrix}, \quad x,y \in \mathbb{R}$$

- komplexe Version: $u(t) := u_1(t) + i u_2(t)$

$$u'(t) = u_1'(t) + i u_2'(t) = -u_2(t) + i u_1(t) = i\,(u_1(t) + i u_2(t)) = i u(t)$$
$$u(0) = u_1(0) + i u_2(0) = a$$
$$\Rightarrow u(t) = a \cdot e^{it} = a(\cos t + i \sin t)$$

• Verhalten für $t \to \infty$: Die Funktionen u_1, u_2 sind in t periodisch und sind für alle t durch $\pm a$ beschränkt. Weiter:

$$\lim_{t\to\infty} \left\| \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} \right\|_2^2 = \lim_{t\to\infty} |a|^2 \underbrace{(\cos^2(t) + \sin^2(t))}_{=1} = |a|^2 = a^2$$

zu (b): $u_h := \begin{pmatrix} u_{h1} \\ u_{h2} \end{pmatrix}$, $A := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$, $f(u) := Au$

explizites Eulerverfahren:

$$u^{j+1} = u^j + h\,\varphi(t_j, u^j, h) = u^j + h\,f(u^j) = (I_2 + hA)u^j = (I_2 + hA)^{j+1} u^0$$

implizites Eulerverfahren:

$$u^{j+1} = u^j + h\,\varphi(t_j, u^j, h) = u^j + h\,f(u^{j+1}) = u^j + hAu^{j+1}$$
$$\Rightarrow u^{j+1} = (I_2 - hA)^{-1} u^j = (I_2 - hA)^{-(j+1)} u^0$$

Nun gilt

$$(I_2+hA)=\begin{pmatrix}1 & -h\\ h & 1\end{pmatrix}$$

$$(I_2-hA)^{-1}=\begin{pmatrix}1 & h\\ -h & 1\end{pmatrix}^{-1}=\frac{1}{1+h^2}\begin{pmatrix}1 & -h\\ h & 1\end{pmatrix}=\frac{1}{1+h^2}(I_2+hA)$$

explizites Eulerverfahren:

$$\|U_h(t_{j+1})\|_2^2=\|U^{j+1}\|_2^2=\|(I_2+hA)^{j+1}U^0\|_2^2$$

$$\underset{(22.1)}{=}(1+h^2)^{j+1}\|U_0\|_2^2=|a|^2(1+h^2)^{j+1}$$

(22.1) $$\|(I_2+hA)V\|_2^2=\left\|\begin{pmatrix}1 & -h\\ h & 1\end{pmatrix}\begin{pmatrix}v_1\\ v_2\end{pmatrix}\right\|_2^2=\left\|\begin{pmatrix}v_1-hv_2\\ hv_1+v_2\end{pmatrix}\right\|_2^2$$

$$=v_1^2-2hv_1v_2+h^2v_2^2+h^2v_1^2+v_2^2+2hv_1v_2$$

$$=(1+h^2)(v_1^2+v_2^2)=(1+h^2)\|V\|_2^2$$

$$\Rightarrow\lim_{j\to\infty}\|U_h(t_{j+1})\|_2^2=\lim_{j\to\infty}|a|^2\cdot\underbrace{(1+h^2)^{j+1}}_{>1,\ da\ h>0}=\infty.$$

$\uparrow$ $1+h^2>1$

implizites Eulerverfahren:

$$\|U_h(t_{j+1})\|_2^2=\|U^{j+1}\|_2^2=\|(I_2-hA)^{-(j+1)}U^0\|_2^2$$

$$\underset{(22.2)}{=}(1+h^2)^{-(j+1)}\|U^0\|_2^2=|a|^2(1+h^2)^{-(j+1)}$$

(22.2) $$\|(I_2-hA)^{-1}V\|_2^2=\left\|\frac{1}{1+h^2}(I_2+hA)V\right\|_2^2\underset{(22.1)}{=}\left(\frac{1}{1+h^2}\right)^2\cdot(1+h^2)\|V\|_2^2$$

$$=(1+h^2)^{-1}\|V\|_2^2$$

$$\Rightarrow\lim_{j\to\infty}\|U_h(t_{j+1})\|_2^2=\lim_{j\to\infty}|a|^2\cdot\underbrace{(1+h^2)^{-(j+1)}}_{>0}=0.$$

$\uparrow$ $1+h^2>0$

FAZIT: Beide Verfahren geben das Verhalten der exakten Lösung im Unendlichen nicht wieder!

Aufgabe 23:

Beweisen Sie Satz 2.17 für $q=p+1$:

Satz: Sei $\bar{u}\in C^{q+2}([t_0,t_E],\mathbb{R}^n)$ Lösung der AWA

$$u' = f(t,u),\ t\in[t_0,t_E],\quad u(t_0)=u^0.$$

Sei $U\subset\mathbb{R}^{n+1}$ Umgebung des Graphen von $\bar{u}$ und $\varphi\in C^{q+1}(U\times[0,\bar{h}],\mathbb{R}^n)$ die Verfahrensfunktion eines Einschrittverfahrens der Ordnung p, $p\le q$. Dann gibt es Funktionen

$$e_j\in C^{q-p+2-j}([t_0,t_E],\mathbb{R}^n),\ j=0,\dots,q-p \quad \text{mit } e_j(t_0)=0,\ j=0,\dots,q-p$$

und

$$\|u_h-\left(\bar{u}+\sum_{j=0}^{q-p}h^{p+j}e_j\right)|_h\|_\infty = \mathcal{O}(h^{p+1}).$$

Lösung:

Vorbemerkung:

Taylor-Formel, Satz von Taylor: $I\subset\mathbb{R}$ Intervall, $f\in C^{n+1}(I,\mathbb{R}^n)$, $n\in\mathbb{N}_0$, $a\in I$.

Dann gilt:

$$f(x) = T_n(x)+R_n(x)\qquad \forall x\in I$$

mit

$$T_n(x)=\sum_{k=0}^{n}\frac{f^{(k)}(a)}{k!}\cdot(x-a)^k \qquad \text{(n-tes Taylorpolynom an der Entwicklungsstelle a)}$$

und

$$R_n(x)=\int_a^{\varphi(1)=x}\frac{(x-t)^n}{n!}\cdot f^{(n+1)}(t)\,dt \qquad \text{(n-tes Restglied, Integralform)}$$

$$\overset{\varphi(s)=x}{=}\int_0^1\frac{(x-a-s(x-a))^n}{n!}\cdot f^{(n+1)}(a+s(x-a))\cdot ds\cdot\underbrace{(x-a)}_{\varphi'(s)},\quad R_n(x)=\mathcal{O}(|x-a|^{n+1})$$

↑ Int. durch Sub. $\varphi(s)=a+s(x-a)$

Beweis: ($q=p+1$)

$\bar{u}\in C^{p+3}([t_0,t_E],\mathbb{R}^n)$, $\varphi\in C^{p+2}(U\times[0,\bar{h}],\mathbb{R}^n)$, $e_j\in C^{3-j}([t_0,t_E],\mathbb{R}^n)$, $j=0,1$ (d.h. $e_0\in C^3$ und $e_1\in C^2$). Wir zeigen:

$$\|T_h\left((\bar{u}+h^p e_0+h^{p+1}e_1)|_h\right)\|_\infty=\mathcal{O}(h^{p+2})$$

- Veränderungen gegenüber dem Fall $q=p$

Es gilt für $t\in\Omega_h\setminus\{t_E\}$:

$$T_h\left((\bar{u}+h^p e_0+h^{p+1}e_1)|_h\right)(t+h)$$

$$\overset{\text{nach Def. von } T_h}{=}\frac{1}{h}\left((\bar{u}+h^p e_0+h^{p+1}e_1)(t+h)-(\bar{u}+h^p e_0+h^{p+1}e_1)(t)\right)-\varphi\left(t,(\bar{u}+h^p e_0+h^{p+1}e_1)(t),h\right)$$

$$\overset{\text{umsortieren des Summanden}}{=}\frac{1}{h}\Big(\underbrace{\bar{u}(t+h)-\bar{u}(t)}_{①}+h^{p+1}\Big(\underbrace{e_0(t+h)}_{②}-e_0(t)\Big)+h^p\Big(\underbrace{e_1(t+h)}_{③}-e_1(t)\Big)-\underbrace{\varphi\left(t,(\bar{u}+h^p e_0+h^{p+1}e_1)(t),h\right)}_{④}\Big)$$

① $\bar{u}(t+h)=\sum_{k=0}^{p+2}\frac{1}{k!}\left(\frac{d^k}{dt^k}\bar{u}\right)(t)\cdot h^k+\underbrace{\int_0^1\frac{(1-s)^{p+2}}{(p+2)!}\cdot\bar{u}^{(p+3)}(t+sh)\,ds\cdot h^{p+3}}_{\mathcal{O}(h^{p+3})},\quad \bar{u}\in C^{p+3}$

$x=t+h,\ a=t,\ n=p+2,\ f=\bar{u}$

② $e_0(t+h)=\sum_{k=0}^{2}\frac{1}{k!}\left(\frac{d^k}{dt^k}e_0\right)(t)\cdot h^k+\underbrace{\int_0^1\frac{(1-s)^2}{2!}e_0^{(3)}(t+sh)\,ds\cdot h^3}_{\mathcal{O}(h^3)},\quad e_0\in C^3$

$x=t+h,\ a=t,\ n=2,\ f=e_0$

③ $e_1(t+h)=\sum_{k=0}^{1}\frac{1}{k!}\left(\frac{d^k}{dt^k}e_1\right)(t)\,h^k+\underbrace{\int_0^1\frac{(1-s)}{1!}\cdot e_1^{(2)}(t+sh)\,ds\cdot h^2}_{\mathcal{O}(h^2)},\quad e_0\in C^2$

$x=t+h,\ a=t,\ n=1,\ f=e_1$

④: $\varphi(t,(\bar u+h^l e_0+h^{l+1}e_1)(t/,h) = \sum_{k=0}^{p+1} \frac{1}{k!}\cdot\left(\frac{\partial^k}{\partial s^k}\varphi\right)(t,(\bar u+h^l e_0+h^{l+1}e_1)(t/,0)\,h^k$

$+ \underbrace{\int_0^1 \frac{(1-s)^{p+1}}{(p+1)!}\cdot\left(\frac{\partial^{p+2}\varphi}{\partial s^{p+2}}\right)(t,(\bar u+h^l e_0+h^{l+1}e_1)(t/,sh)\,ds\cdot h^{p+2}}_{\Theta(h^{p+2})}$, $\varphi\in C^{p+2}$

$x=h,\ a=0,\ n=p+1,\ f=\varphi$

$= \underbrace{\frac{1}{h}\sum_{k=1}^{p+2}\frac{1}{k!}\left(\frac{d^k}{dt^k}\bar u\right)(t)h^k}_{=!} + \Theta(h^{p+2}) + h^{p-1}\underbrace{\sum_{k=1}^{2}\frac{1}{k!}\cdot\left(\frac{d^k}{dt^k}e_0\right)(t)\cdot h^k}_{=!} + \Theta(h^{p+2})$

p+1. & p+2. Summanden abspalten & Indexverschiebung

$+ h^l\underbrace{\sum_{k=1}^{1}\frac{1}{k!}\left(\frac{d^k}{dt^k}e_1\right)(t)h^k}_{=!} + \Theta(h^{p+2}) - \underbrace{\sum_{k=0}^{p+1}\frac{1}{k!}\left(\frac{\partial^k}{\partial s^k}\varphi\right)(t,(\bar u+h^l e_0+h^{l+1}e_1)(t/,0)\,h^k}$

0. & p+1. Summanden abspalten

$-\Theta(h^{p+2})$

$= \bar u'(t) + \sum_{k=1}^{p-1}\frac{1}{(k+1)!}\left(\frac{d^{k+1}}{dt^{k+1}}\bar u\right)(t)h^k + \frac{1}{(p+1)!}\cdot\left(\frac{d^{p+1}}{dt^{p+1}}\bar u\right)(t)\cdot h^l + \frac{1}{(p+2)!}\cdot\left(\frac{d^{p+2}}{dt^{p+2}}\bar u\right)(t)\,h^{l+1}$

$+ h^l e_0'(t) + \frac{1}{2}h^{l+1}e_0''(t) + h^{l+1}e_1'(t) - \underbrace{\varphi(t,\bar u(t)+h^l e_0(t)+h^{l+1}e_1(t/,0)}_{⑤}$

$- \sum_{k=1}^{p}\frac{1}{k!}\underbrace{\left(\frac{\partial^k}{\partial s^k}\varphi\right)(t,\bar u(t)+h^l e_0(t)+h^{l+1}e_1(t/,0)}_{⑥}h^k - \frac{1}{(p+1)!}\underbrace{\left(\frac{\partial^{p+1}}{\partial s^{p+1}}\varphi\right)(t,\bar u(t)+h^l e_0(t)+h^{l+1}e_1(t/,0)}_{⑦}h^{p+1}$

$+\Theta(h^{p+2})$

⑤: $\varphi(t,\bar u(t)+h^l e_0(t)+h^{l+1}e_1(t/,0) = \sum_{k=0}^{1}\frac{1}{k!}\cdot\left(\frac{\partial^k}{\partial v^k}\varphi\right)(t,\bar u(t),0)\cdot(h^l e_0(t)+h^{l+1}e_1(t))^k$

$+ \underbrace{\int_0^1 \frac{(1-s)}{1!}\left(\frac{\partial^2}{\partial v^2}\varphi\right)(t,\bar u(t)+s(h^l e_0(t)+h^{l+1}e_1(t)),0)\,ds\,(h^l e_0(t)+h^{l+1}e_1(t))^2}_{\Theta(h^{2l})}$

$x=\bar u(t)+h^l e_0(t)+h^{l+1}e_1(t),\ a=\bar u(t),\ n=1,\ f=\varphi$

⑥: $\left(\frac{\partial^k}{\partial s^k}\varphi\right)(t,\bar u(t)+h^l e_0(t)+h^{l+1}e_1(t),0) = \sum_{j=0}^{1}\frac{1}{j!}\cdot\left(\frac{\partial^j}{\partial v^j}\frac{\partial^k}{\partial s^k}\varphi\right)(t,\bar u(t),0)\cdot(h^l e_0(t)+h^{l+1}e_1(t))^j$

$+ \underbrace{\int_0^1 \frac{(1-s)}{1!}\left(\frac{\partial^2}{\partial v^2}\frac{\partial^k}{\partial s^k}\varphi\right)(t,\bar u(t)+s(h^l e_0(t)+h^{l+1}e_1(t)),0)\,ds\,(h^l e_0(t)+h^{l+1}e_1(t))^2}_{\Theta(h^{2l})}$, $k=1,\dots,p$

$x=\bar u(t)+h^l e_0(t)+h^{l+1}e_1(t),\ a=\bar u(t),\ n=1,\ f=\frac{\partial^k}{\partial s^k}\varphi$

⑦: $\left(\frac{\partial^{p+1}}{\partial s^{p+1}}\varphi\right)(t,\bar u(t)+h^l e_0(t)+h^{l+1}e_1(t),0) = \left(\frac{\partial^{p+1}}{\partial s^{p+1}}\varphi\right)(t,\bar u(t),0)$

$+ \underbrace{\int_0^1 \frac{1}{0!}\cdot\left(\frac{\partial}{\partial v}\frac{\partial^{p+1}}{\partial s^{p+1}}\varphi\right)(t,\bar u(t)+s(h^l e_0(t)+h^{l+1}e_1(t)),0)\,ds\,(h^l e_0(t)+h^{l+1}e_1(t))}_{\Theta(h)}$

$x=\bar u(t)+h^l e_0(t)+h^{l+1}e_1(t),\ a=\bar u(t),\ n=0,\ f=\frac{\partial^{p+1}}{\partial s^{p+1}}\varphi$

$$= \bar{u}'(t) + \sum_{k=1}^{p-1} \frac{1}{(k+1)!} \underbrace{\bar{u}^{(k+1)}(t)}_{=\frac{\partial^k f}{\partial t^k}(t,\bar{u}(t))} h^k + \frac{1}{(p+1)!} \bar{u}^{(p+1)}(t) h^p + \frac{1}{(p+2)!} \bar{u}^{(p+2)}(t) h^{p+1}$$

$$+ h^p e_0'(t) + \frac{1}{2} h^{p+1} e_0''(t) + h^{p+1} e_1'(t) - \sum_{k=0}^{1} \frac{1}{k!} \left(\frac{\partial^k}{\partial v^k}\varphi\right)(t,\bar{u}(t),0)\left(h^p e_0(t) + h^{p+1} e_1(t)\right)^k$$

$$- \mathcal{O}(h^{2p}) - \sum_{k=1}^{p} \frac{1}{k!} \left[\sum_{j=0}^{1} \frac{1}{j!} \left(\frac{\partial^j}{\partial v^j} \frac{\partial^k}{\partial s^k}\varphi\right)(t,\bar{u}(t),0) \cdot \left(h^p e_0(t) + h^{p+1} e_1(t)\right)^j + \mathcal{O}(h^{2p})\right] h^k \quad = \mathcal{O}(h^{p+k})$$

$$- \frac{1}{(p+1)!} \cdot \left[\left(\frac{\partial^{p+1}}{\partial s^{p+1}}\varphi\right)(t,\bar{u}(t),0) + \underbrace{\mathcal{O}(h)}_{\mathcal{O}(h^{p+2})}\right] \cdot h^{p+1} + \mathcal{O}(h^{p+2})$$

$\Rightarrow j=1: k=1$
$\rightsquigarrow e_0$ in h^{p+1} Gleichung
e_1 in $\mathcal{O}(h^{p+2})$
$k=2,\dots,p$
$\rightsquigarrow$ in $\mathcal{O}(h^{p+2})$
$j=0: k=1,\dots,p-1$
$\rightsquigarrow$ Konsistenz
$k=p$
$\rightsquigarrow$ in h^p Gleichung

$$= \underbrace{u'(t) - \underbrace{\varphi(t,\bar{u}(t),0)}_{=f(t,\bar{u}(t))}}_{=0 \text{ (da } \bar{u} \text{ Lösung von } u'=f(t,u))} + \sum_{k=1}^{p-1} h^k \underbrace{\left[\frac{1}{(k+1)!}\left(\frac{\partial^k}{\partial t^k} f\right)(t,\bar{u}(t)) - \frac{1}{k!}\left(\frac{\partial^k}{\partial s^k}\varphi\right)(t,\bar{u}(t),0)\right]}_{=0 \text{ (da Verfahren konsistent der Ordnung } p)}$$

$$+ h^p \cdot \underbrace{\left[\frac{1}{(p+1)!}\bar{u}^{(p+1)}(t) + e_0'(t) - \left(\frac{\partial}{\partial v}\varphi\right)(t,\bar{u}(t),0)\cdot e_0(t) - \frac{1}{p!}\left(\frac{\partial^p}{\partial s^p}\varphi\right)(t,\bar{u}(t),0)\right]}_{\stackrel{!}{=}0 \text{ (Gleichung für } e_0)}$$

$$+ h^{p+1} \cdot \underbrace{\left[\frac{1}{(p+2)!}\bar{u}^{(p+2)}(t) + \frac{1}{2}e_0''(t) + e_1'(t) - \left(\frac{\partial}{\partial v}\varphi\right)(t,\bar{u}(t),0)\cdot e_1(t) - \left(\frac{\partial}{\partial v}\frac{\partial}{\partial s}\varphi\right)(t,\bar{u}(t),0)\cdot e_0(t) - \frac{1}{(p+1)!}\left(\frac{\partial^{p+1}}{\partial s^{p+1}}\varphi\right)(t,\bar{u}(t),0)\right]}_{\stackrel{!}{=}0 \text{ (Gleichung für } e_1)}$$

$$+ \mathcal{O}(h^{p+2})$$

$$e_0'(t) = -\frac{1}{(p+1)!}\bar{u}^{(p+1)}(t) + \left(\frac{\partial}{\partial v}\varphi\right)(t,\bar{u}(t),0)\cdot e_0(t) + \frac{1}{p!}\cdot\left(\frac{\partial^p}{\partial s^p}\varphi\right)(t,\bar{u}(t),0)$$

$$e_1'(t) = -\frac{1}{2}e_0''(t) - \frac{1}{(p+2)!}\bar{u}^{(p+2)}(t) + \left(\frac{\partial}{\partial v}\varphi\right)(t,\bar{u}(t),0)\cdot e_1(t) + \left(\frac{\partial}{\partial v}\frac{\partial}{\partial s}\varphi\right)(t,\bar{u}(t),0)\,e_0(t)$$
$$+ \frac{1}{(p+1)!}\left(\frac{\partial^{p+1}}{\partial s^{p+1}}\varphi\right)(t,\bar{u}(t),0)$$

$$e_0(t_0) = 0\,,\quad e_1(t_0) = 0$$

Man erkennt, dass die Lösungen der Bedingung $e_0 \in C^3$, $e_1 \in C^2$ genügen, aufgrund der Glattheit von $\bar{u}$ & φ.

```
% Aufgabenteil (a)
%% Initialisierung
sigma = 10;
lambda = 28;
mu = 8/3;
f = @(t,v) [sigma*(v(2)-v(1));lambda*v(1)-v(2)-v(1)*v(3);-mu*v(3)+v(1)*v(2)];
u_init=[1;4;9];
t_init=0;
t_end=50;
h=10.^(-(1:3));

%% Runge-Kutta-Tableau (m=4) Klassisches Runge-Kutta-Verfahren (aequidistant)
alpha=[0 1/2 1/2 1];
beta=[0 0 0 0;1/2 0 0 0;0 1/2 0 0;0 0 1 0];
gamma=[1/6 1/3 1/3 1/6];

for i=1:3
  [tn_rkclassic{i},un_rkclassic{i}]=rk_explicit(f,u_init,t_init,t_end,h(i),
alpha,beta,gamma);
end

%% Graphische Ausgabe (aequidistant)
set(0,'DefaultAxesFontSize',15.0);
set(0,'DefaultTextFontSize',15.0);
figure
for i=1:3
    subplot(2,2,i)
    hold on
    plot(tn_rkclassic{i},un_rkclassic{i}(1,:),'b');
    plot(tn_rkclassic{i},un_rkclassic{i}(2,:),'r');
    plot(tn_rkclassic{i},un_rkclassic{i}(3,:),'g');
    hold off
    title(['a) Loesungen x, y und z fuer h=',num2str(h(i))]);
    xlabel('t'); ylabel('x(t),y(t),z(t)');
    legend('(t,x)','(t,y)','(t,z)','Location','Best');
end

% Aufgabenteil (b)
%% Initialisierung
sigma = 10;
lambda = 28;
mu = 8/3;
f = @(t,v) [sigma*(v(2)-v(1));lambda*v(1)-v(2)-v(1)*v(3);-mu*v(3)+v(1)*v(2)];
u_init=[1;4;9];
t_init=0;
t_end=50;

%% Initialisierung fuer die Schrittweitensteuerung
kappa = 10;
h0 = 0.2;
hmax = 0.5;
tol = 10.^(-2*(2:4));

%% Runge-Kutta-Tableau (m=4) Klassisches Runge-Kutta-Verfahren (adaptiv)
alpha=[0 1/2 1/2 1];
beta=[0 0 0 0;1/2 0 0 0;0 1/2 0 0;0 0 1 0];
```

```
gamma=[1/6 1/3 1/3 1/6];
p=4;
tk = 5*(0:10);
phi=@(t,u,h) phi_RK(t,u,h,f,alpha,beta,gamma);

for i=1:3
  [tn_rkclassic_adaptive{i},un_rkclassic_adaptive{i},un_tk_rkclassic_adaptive
{i}]=adapstepcont(phi,u_init,t_init,t_end,h0,hmax,tol(i),kappa,p,tk);
end

%% Graphische Ausgabe (adaptiv)
set(0,'DefaultAxesFontSize',15.0);
set(0,'DefaultTextFontSize',15.0);
figure
for i=1:3
    subplot(2,2,i)
    hold on
    plot(tn_rkclassic_adaptive{i},un_rkclassic_adaptive{i}(1,:),'b');
    plot(tn_rkclassic_adaptive{i},un_rkclassic_adaptive{i}(2,:),'r');
    plot(tn_rkclassic_adaptive{i},un_rkclassic_adaptive{i}(3,:),'g');
    hold off
    title(['b) Loesungen x, y und z fuer tol=',num2str(tol(i))]);
    xlabel('t'); ylabel('x(t),y(t),z(t)');
    legend('(t,x)','(t,y)','(t,z)','Location','Best');
end

%% Graphische Ausgabe (aequidistant vs. adaptiv)
figure
for i=1:3
    subplot(2,2,i)
    hold on
    plot(tn_rkclassic{3},un_rkclassic{3}(i,:),'b')
    plot(tn_rkclassic_adaptive{3},un_rkclassic_adaptive{3}(i,:),'r')
    title([num2str(i),'.-Komponente des Lorenz-Systems']);
    xlabel('t'); ylabel([num2str(i),'.-Komponente']);
    legend('aequidist. RK','Schrittweitensteuerung','Location','Best');
end

%% Graphische Ausgabe (3D)
figure;
plot3(un_rkclassic{end}(1,:),un_rkclassic{end}(2,:),un_rkclassic{end}(3,:));
title('Loesung fuer (x(t),y(t),z(t)), vgl. Lorenzattraktor');
xlabel('x(t)'); ylabel('y(t)'); zlabel('z(t)');

%% Textausgabe (Vergleich an den Stuetzstellen tk)
fprintf(['\n\nErgebnisse des klassischen Runge-Kutta-Verfahrens'...
    ' ohne Schrittweitensteuerung:'])
for i = 1:length(h)
  ind{i}=1+tk/h(i);
  fprintf(['\n\nSchrittweite h = ' num2str(h(i))])
  fprintf('\n  t_k        x(t)          y(t)          z(t)')
  fprintf('\n-------------------------------------------------')
  for j = 1:length(tk)
    fprintf('\n  %2.0f   %11.6f   %11.6f   %11.6f',tk(j),un_rkclassic{i}(:,ind{i}
(j)))
  end
```

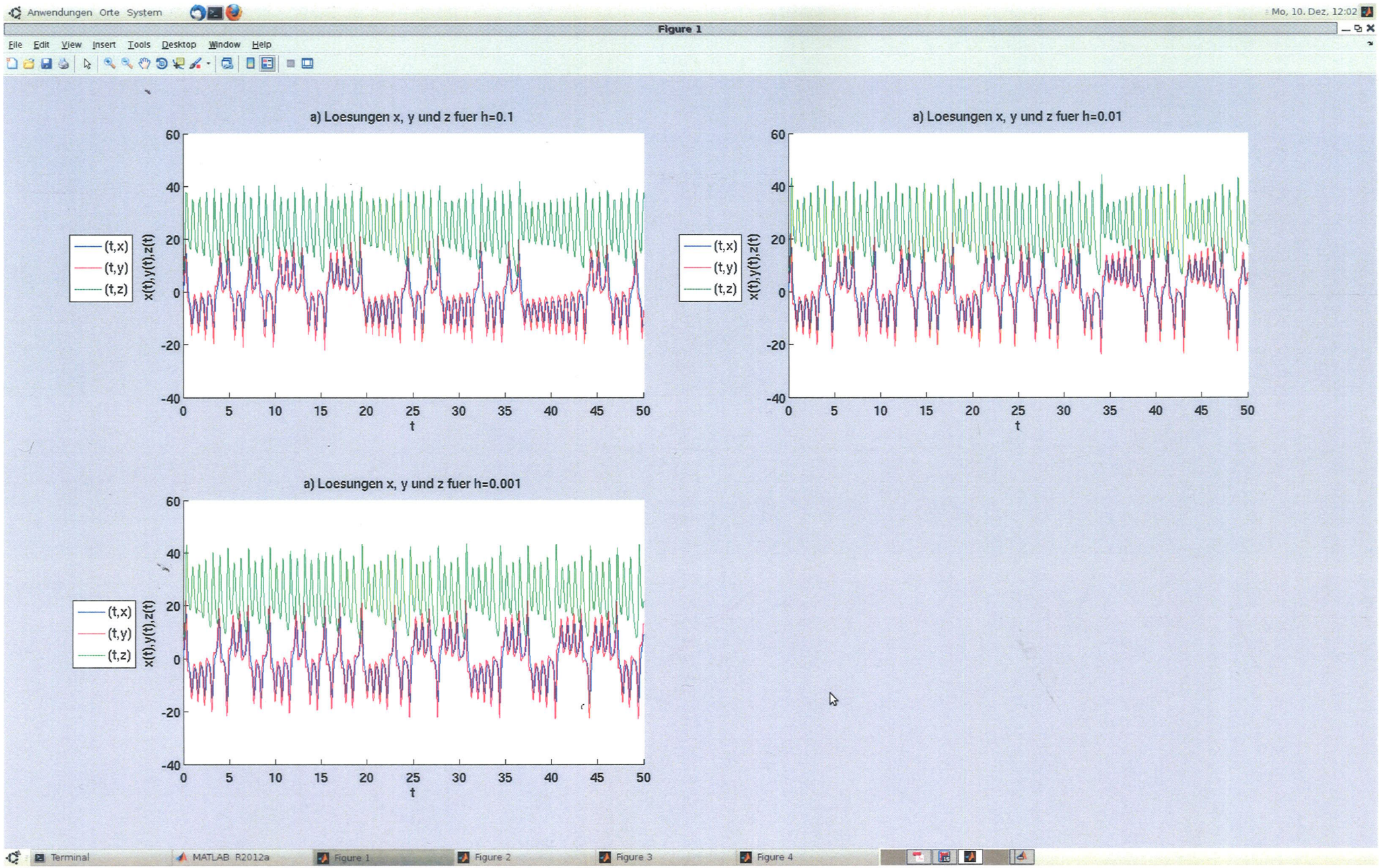
```
end
fprintf('\n')

fprintf(['\n\nErgebnisse des klassischen Runge-Kutta-Verfahrens'...
    ' mit Schrittweitensteuerung:'])
for i = 1:length(tol)
    fprintf(['\n\nToleranz tol = ' num2str(tol(i))])
    fprintf('\n  t_k       x(t)          y(t)          z(t)')
    fprintf('\n-------------------------------------------------')
    for j = 1:length(tk)
        fprintf('\n  %2.0f   %11.6f   %11.6f   %11.6f',tk(j),
un_tk_rkclassic_adaptive{i}(:,j))
    end
end
fprintf('\n')
```

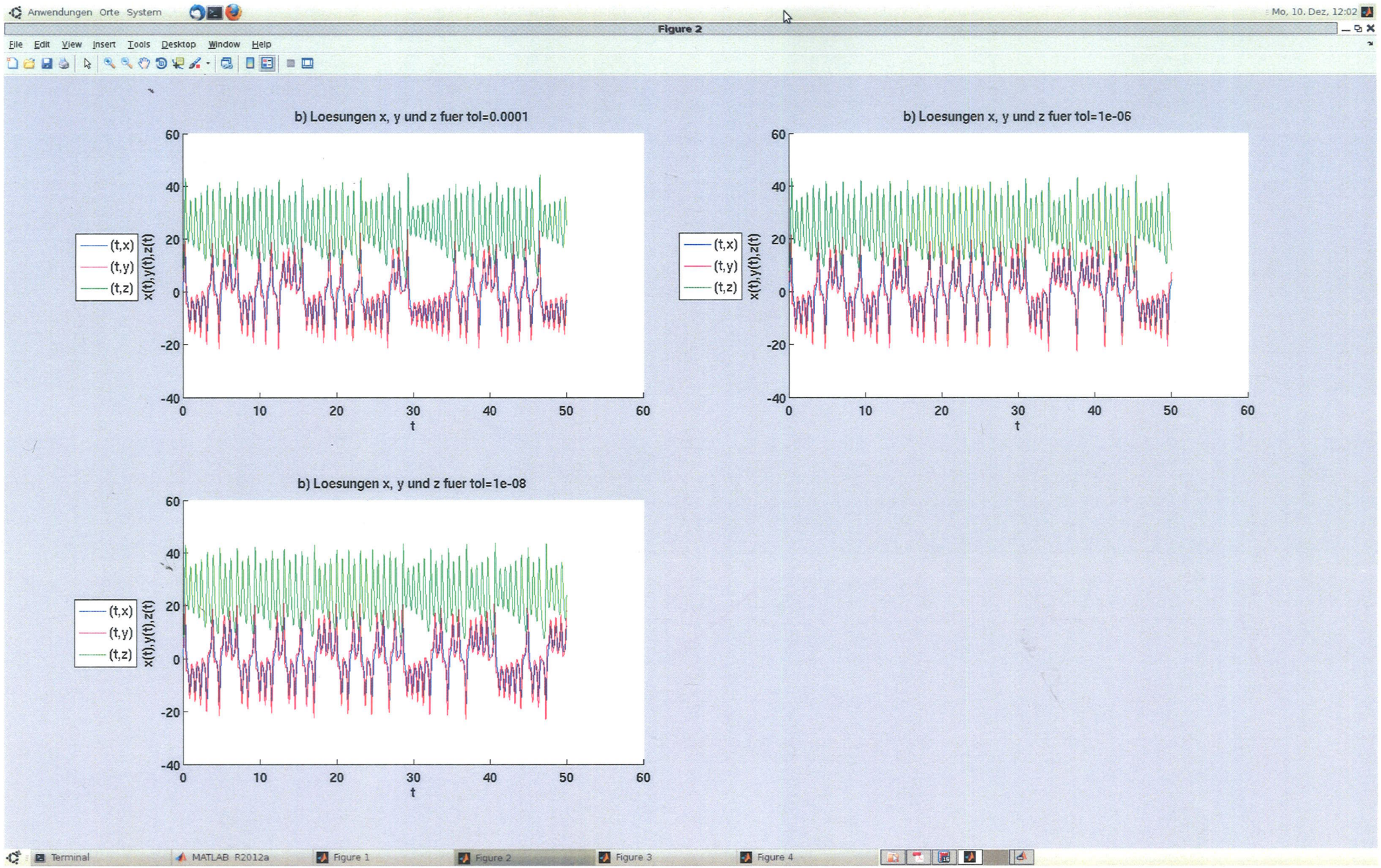
```
function [tn,un] = rk_explicit(f,u_init,t_init,t_end,h,alpha,beta,gamma)
%RK_EXPLICIT loest Anfangswertprobleme der Form
%              u'(t) = f(t,u(t)), u(t_init) = u_init
%            mit expliziten Runge-Kutta Verfahren (alpha,beta,gamma)
%            zur konstanten Zeitschrittweite h auf dem endlichen
%            Zeitintervall [t_init,t_end].

%% Initialisierung
  m=length(gamma);                  % Stufe des RK-Verfahrens
  tn=t_init:h:t_end;                % diskretes Zeitintervall
  time_steps=length(tn);            % Anzahl der Zeitschritte
  space_dim=length(u_init);         % Dimension der DGL, Raumdimension
  un=zeros(space_dim,time_steps); % Speicherreservierung fuer RK-Iteration
  un(:,1)=u_init;                   % Initialisierung der Anfangsdaten
  k=zeros(space_dim,m);             % Speicherreservierung der k_i's

%% Explizites Runge-Kutta-Verfahren
  for j=2:time_steps
    % Berechnung der k_i's
    for i=1:m
      temp_k=zeros(space_dim,1);
      for l=1:i-1
        temp_k=temp_k+beta(i,l)*k(:,l);
      end
      k(:,i)=f(tn(j-1)+alpha(i)*h,un(:,j-1)+h*temp_k);
    end

    % Berechnung der u^j's
    temp_phi=zeros(space_dim,1);
    for i=1:m
      temp_phi=temp_phi+gamma(i)*k(:,i);
    end
    un(:,j)=un(:,j-1)+h*temp_phi;
  end
end
```

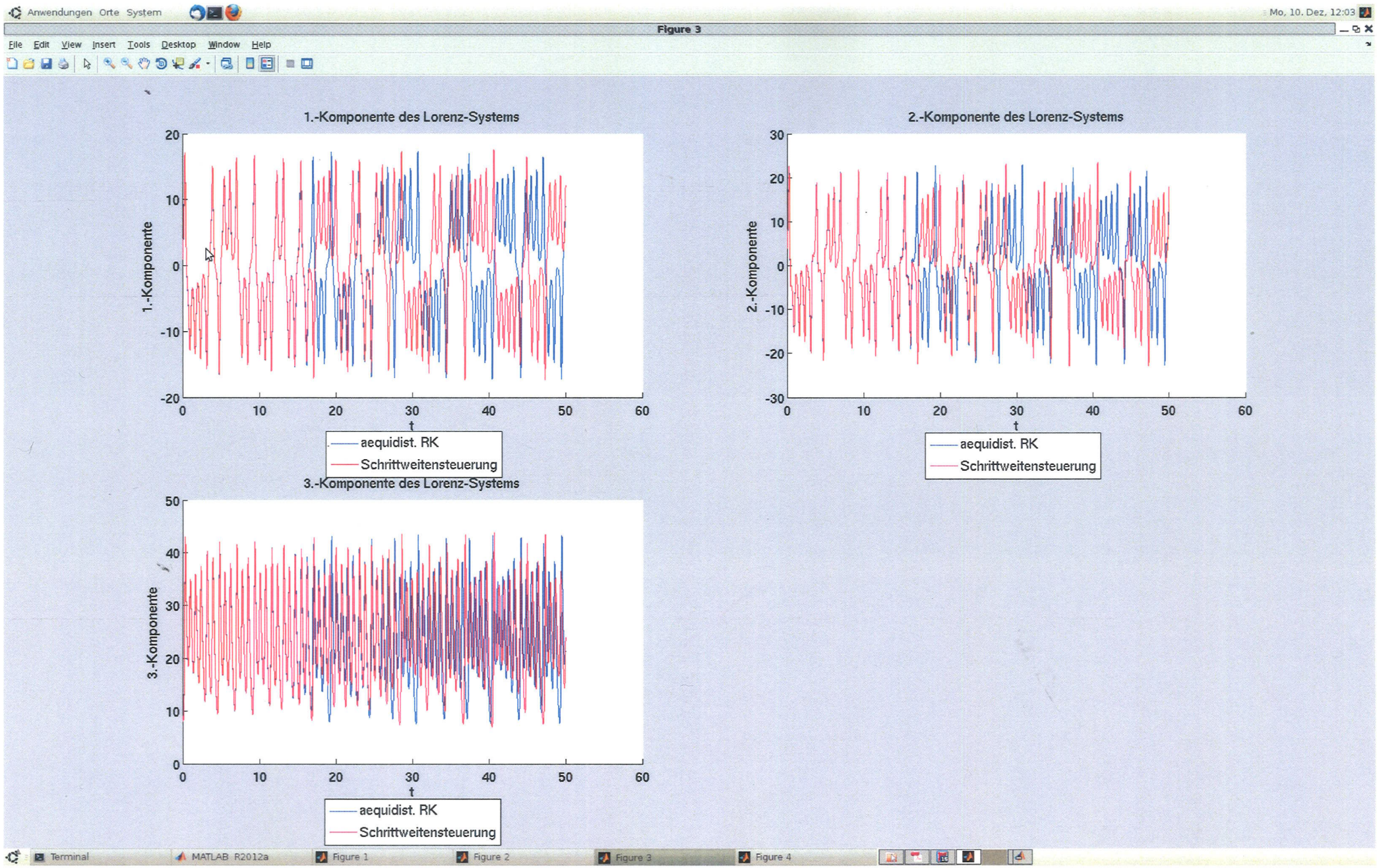
```
function [tn,un,un_tk] = adapstepcont(phi,u_init,t_init,t_end,h0,h_max,tol,
kappa,p,tk)
%ADAPSTEPCONT ist der (h,h/2)-Schrittweitensteuerungsalgorithmus
%   phi     : function handle fuer die Verfahrensfunktion des verwendeten
%             numerischen Verfahrens der Form phi(t,v,h)
%   u_init  : Anfangswert
%   t_init  : Anfangszeitpunkt
%   t_end   : Endzeitpunkt
%   h0      : Anfangsschrittweite
%   h_max   : obere Schranke fuer die Schrittweite
%   tol     : Toleranz
%   kappa   : Skalierungsfaktor, kontrolliert die Schrittweite
%   p       : Konvergenzordnung des numerischen Verfahrens
%   tk      : Vektor mit speziellen Zeitpunkten im Intervall [t_init,t_end].
%             Die Funktion adapstepcont liefert Approximationswerte der DGL
%             zu den Zeitpunkten tk. Die Wahl der tk's hat keinen Einfluss
%             auf den Algorithmus.
%   tn      : adaptives Zeitgitter
%   un      : Approximation der DGL zu den Zeitpunkten tn
%   un_tk   : Approximation der DGL zu den Zeitpunkten tk

maxsteps = 10000; % maximale Anzahl an Iterationsschritten

tau  = t_init;
u    = u_init;
h    = h0;

space_dim = length(u_init); % Dimension der DGL, Raumdimension

% initialising memory
tn    = zeros(1,maxsteps);
un    = zeros(space_dim,maxsteps);
un_tk = zeros(space_dim,length(tk));

un(:,1) = u;

% counters
z1 = 1;
z2 = 1; % counter for vector tk

% stepsize control algorithm
while (tau < t_end) && (z1 < maxsteps)
    v = u + h     * phi(tau,u,h);
    z = u + (h/2) * phi(tau,u,h/2);
    w = z + (h/2) * phi(tau+h/2,z,h/2);

    gamma  = (1/(2^p - 1)) * max(abs(v-w)./max(1,abs(u)));
    hprime = h*(tol/(2*gamma))^(1/(p+1));

    if gamma <= tol
        u = w;
        tau = tau+h;
        h = min([hprime, kappa*h, h_max]);

        % checking if tau is larger than the next time point in tk
        if (z2 <= length(tk))&&(tau > tk(z2))
```

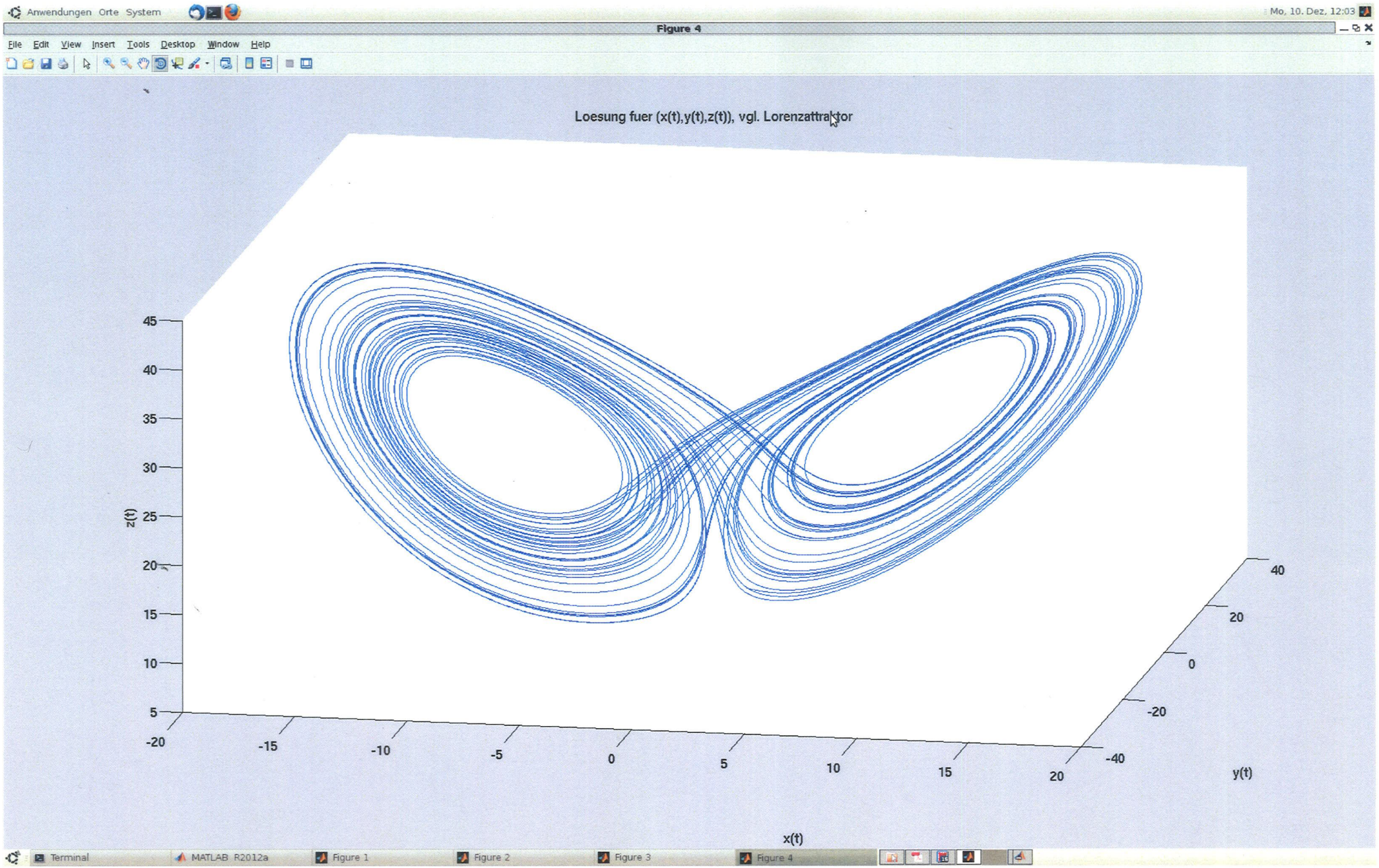
```
            % if true, we make an extra step with the numerical scheme from
            % the previous time point to tk(z2)
            htilde      = tk(z2) - tn(z1);
            un_tk(:,z2) = un(:,z1) + htilde*phi(tn(z1),un(:,z1),htilde);
            z2 = z2 + 1;
        end

        z1 = z1 + 1;
        % saving the new approximations
        tn(z1)    = tau;
        un(:,z1) = u;
    else
        h = max(hprime,h/kappa);
    end


end

if z1 == maxsteps
    fprintf(['\nWarning in adapstepcont: Maximal number of steps exceeded, '...
        'numerical scheme did not reach the end of the interval'])
else
    un = un(:,1:z1);
    tn = tn(1:z1);
end

end
```

```
function out = phi_RK(t,u,h,f,alpha,beta,gamma)

% Initialisierung
m=length(gamma);          % Stufe des RK-Verfahrens
space_dim = length(u);    % Dimension der DGL, Raumdimension
k = zeros(space_dim,m);   % Speicherreservierung der k_i's

% Berechnung der k_i's
for i = 1:m
  k(:,i) = f(t + alpha(i)*h,u+h*k*beta(i,:)');
end
out = k*gamma';

end
```


Anwendungen Orte System
Mo, 10. Dez, 12:02
Figure 1
File Edit View Insert Tools Desktop Window Help
a) Loesungen x, y und z fuer h=0.1
a) Loesungen x, y und z fuer h=0.01
a) Loesungen x, y und z fuer h=0.001
(t,x)
(t,y)
(t,z)
x(t),y(t),z(t)
t
Terminal
MATLAB R2012a
Figure 1
Figure 2
Figure 3
Figure 4


Anwendungen Orte System
Mo, 10. Dez, 12:02
Figure 2
File Edit View Insert Tools Desktop Window Help
b) Loesungen x, y und z fuer tol=0.0001
b) Loesungen x, y und z fuer tol=1e-06
b) Loesungen x, y und z fuer tol=1e-08
(t,x)
(t,y)
(t,z)
x(t),y(t),z(t)
t
Terminal
MATLAB R2012a
Figure 1
Figure 2
Figure 3
Figure 4


Anwendungen Orte System
Mo, 10. Dez, 12:03
Figure 3
File Edit View Insert Tools Desktop Window Help
1.-Komponente des Lorenz-Systems
1.-Komponente
t
aequidist. RK
Schrittweitensteuerung
2.-Komponente des Lorenz-Systems
2.-Komponente
t
aequidist. RK
Schrittweitensteuerung
3.-Komponente des Lorenz-Systems
3.-Komponente
t
aequidist. RK
Schrittweitensteuerung
Terminal
MATLAB R2012a
Figure 1
Figure 2
Figure 3
Figure 4


Anwendungen Orte System
Mo, 10. Dez, 12:03
Figure 4
File Edit View Insert Tools Desktop Window Help
Loesung fuer (x(t),y(t),z(t)), vgl. Lorenzattraktor
45
40
35
30
25
20
15
10
5
z(t)
-20
-15
-10
-5
0
5
10
15
20
x(t)
40
20
0
-20
-40
y(t)
Terminal
MATLAB R2012a
Figure 1
Figure 2
Figure 3
Figure 4

Ergebnisse des klassischen Runge-Kutta-Verfahrens ohne Schrittweitensteuerung:

Schrittweite h = 0.1

t_k	x(t)	y(t)	z(t)
0	1.000000	4.000000	9.000000
5	11.782650	2.831553	39.075782
10	-7.936518	2.250581	35.249941
15	-0.578804	-0.911490	11.643803
20	-12.722358	-10.930672	34.549504
25	-3.830672	-6.662106	13.569722
30	-10.236146	-3.770811	35.311700
35	0.359695	1.835639	20.258237
40	-7.831848	-2.671624	31.848692
45	-5.144418	-9.384353	11.506879
50	-5.787571	-1.909826	28.882546

Schrittweite h = 0.01

t_k	x(t)	y(t)	z(t)
0	1.000000	4.000000	9.000000
5	-0.205894	3.766911	26.050564
10	-12.912296	-15.941356	29.319727
15	-1.363929	-3.013372	20.676049
20	-12.418212	-6.293827	37.737237
25	2.143587	3.671245	18.826581
30	13.587500	13.612837	33.747569
35	4.031635	5.354932	18.718316
40	-7.081940	1.362344	34.128058
45	1.824365	3.207368	12.178317
50	7.754503	2.983653	31.550039

Schrittweite h = 0.001

t_k	x(t)	y(t)	z(t)
0	1.000000	4.000000	9.000000
5	-0.204574	3.766235	26.047472
10	-12.856925	-15.909129	29.203778
15	0.939397	1.850450	17.325806
20	-11.598548	-14.983985	26.794765
25	2.938885	4.827153	20.335237
30	3.005529	-0.864806	26.983876
35	10.775667	14.926279	24.313105
40	2.772693	3.944942	16.645130
45	-9.617627	-4.482781	33.739733
50	9.104705	14.494411	18.770210

Ergebnisse des klassischen Runge-Kutta-Verfahrens mit Schrittweitensteuerung:

Toleranz tol = 0.0001

t_k	x(t)	y(t)	z(t)
0	1.000000	4.000000	9.000000
5	-0.108853	3.679517	25.760905
10	15.072398	17.803292	33.105590
15	0.419734	0.485825	14.835970
20	-7.905851	0.204024	34.557179
25	-8.072918	-12.623682	18.471347
30	-9.379721	-6.116954	31.741029
35	0.282110	1.174791	18.844524
40	13.962720	16.989980	30.974382
45	-0.523299	-2.330316	21.745626
50	-4.444881	-0.870601	27.543159

Toleranz tol = 1e-06

t_k	x(t)	y(t)	z(t)
0	1.000000	4.000000	9.000000
5	-0.202704	3.764604	26.042102
10	-12.700997	-15.827098	28.865253
15	0.620543	0.695714	15.752053

```
  20    14.758311    11.414916    38.524166
  25     1.197942     2.336901    18.306733
  30    14.561389    13.366877    36.335114
  35     4.384266     6.043092    18.431543
  40     1.941381     1.067134    21.399111
  45     1.085856     2.078532     7.790274
  50     3.971763     6.420826    15.883243

Toleranz tol = 1e-08
  t_k      x(t)         y(t)         z(t)
-----------------------------------------
   0     1.000000     4.000000     9.000000
   5    -0.204543     3.766207    26.047383
  10   -12.854396   -15.907751    29.198360
  15     0.785155     1.591980    17.013402
  20    15.847120    13.127772    39.557574
  25     2.643878     4.298547    18.036594
  30   -10.627302    -5.671017    34.684707
  35     6.154111     9.927900    16.483012
  40     0.458389    -0.047691    18.468376
  45    14.606596     6.779821    41.766211
  50    11.614965    17.523931    22.835602
```