

Modellierung dynamischer Systeme

Abgabe der Praktikumsaufgabe 1

Maria Lüdemann und Birger Kamp

April 1, 2016

1 Teilaufgabe 1 - Steife Differentialgleichungen

Es ist folgende DGL gegeben:

$$y' = 10 - 500y + 5000x, y(0) = 1$$

Dazu ist folgende analytische Lösung gegeben:

$$y = 10x + e^{-500x}$$

Für diese DGL ist ein Schaltbild in Simulink zu entwerfen. Des weiteren sollen die Verfahren(Runge-Kutta 2. Ordnung, expliziter Euler, impliziter Euler) erarbeitet und Iterationsgleichungen für die jeweiligen Verfahren angegeben werden. Im Anschluss soll ein Programm entwickelt(*stiff*) werden das das jedes Verfahren auf die DGL anwendet und in einem Plot zum Vergleich mit der gegebenen analytischen Lösung anzeigt. Zum weiteren Vergleich soll ein Plot berechnet werden der die Differenz zwischen den Verfahrenslösungen und der analytischen Lösung darlegt.

Das Programm (*stiff*) soll mit vier Parameter Varianten gestartet werden, die sich wie folgt beschreiben:

- $h = 0.001, x_{End} = 0.2$
- $h = 0.003, x_{End} = 0.2$
- $h = 0.004, x_{End} = 0.2$
- $h = 0.005, x_{End} = 0.2$

Schaltbild

Das Simulink-Schaltbild zu dieser Gleichung ist:

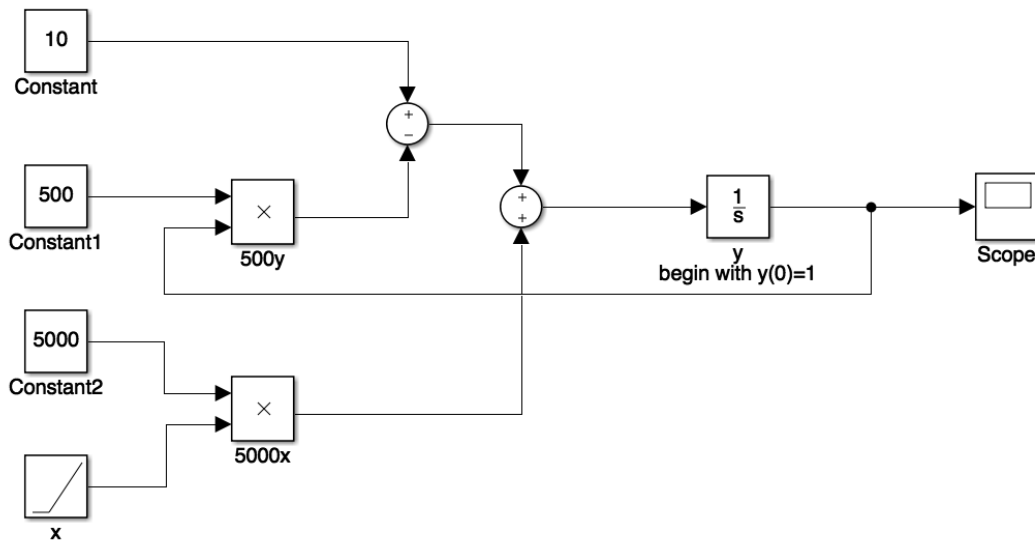


Figure 1: Simulink Schaltbild DGL1

1.1 Iterationsgleichungen

Im Folgenden die Iterationsgleichungen der jeweiligen Verfahren.

1.1.1 Explizites Euler-Verfahren

$$x_{n+1} = x_n + h \quad (1)$$

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

$$y_{n+1} = y_n + h * (10 - 500y_n + 5000x_n) \quad (3)$$

1.1.2 Implizites Euler-Verfahren

Das implizite Euler entspricht dem expliziten mit dem Unterschied das hier anstelle des y_n ein y_{n+1} genutzt wird, dass mithilfe geeigneter Verfahren zur Nullstellenberechnung approximiert werden muss.

$$x_{n+1} = x_n + h \quad (4)$$

$$y_{n+1} = y_n + h * f(x_{n+1}, y_{n+1}) \quad (5)$$

$$y_{n+1} = y_n + h * (10 - 500y_{n+1} + 5000x_{n+1}) \quad (6)$$

1.1.3 Runge-Kutta-Verfahren 2.Ordnung

$$x_{n+1} = x_n + h \quad (7)$$

$$k_1 = h * f(x_n, y_n) \quad (8)$$

$$k_1 = h * (10 - 500y_n + 5000x_n) \quad (9)$$

$$k_2 = h * f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \quad (10)$$

$$k_2 = h * (10 - 500 * (y_n + \frac{k_1}{2}) + 5000 * (x_n + \frac{h}{2})) \quad (11)$$

$$y_{n+1} = y_n + k_2 \quad (12)$$

1.2 Plot der Lösungen

Im Folgenden sind alle Plots der Ergebnisse der Verfahren dargestellt. Es wird außerdem jeweils die Ergebnis-Differenz eines Verfahrens zur analytischen Lösung gezeigt.

1.2.1 h=0.001

Das Ergebnis zeigt, dass die Verfahren bis $x=0.01$ sehr ähnliche Ergebnisse liefern und haben eine max. Differenz von ca. 0.12. Danach laufen alle Kurven kongruent.

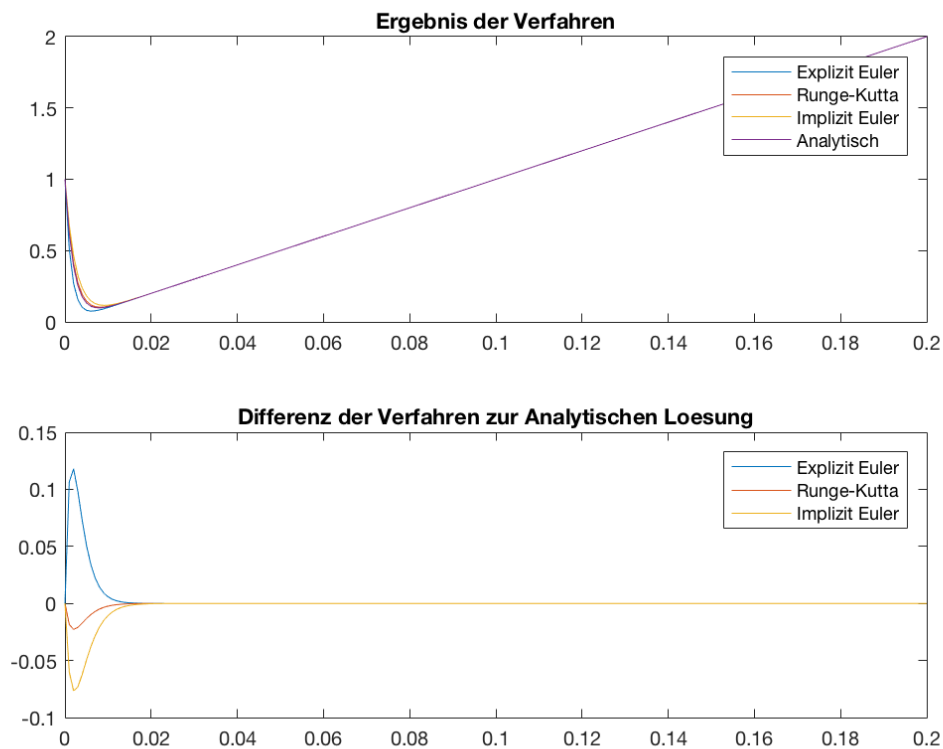


Figure 2: $h=0.001$

1.2.2 $h=0.003$

Das Ergebnis zeigt, dass die Verfahren bis ca. $x=0.03$ unterschiedliche Ergebnisse liefern. Die Ergebnisse des Runge-Kutta-Verfahrens verlaufen bis ca. $x=0.03$ nahezu parallel zur analytischen Lösung. In diesem Wertebereich liefert das explizite Euler-Verfahren sehr schwankende Werte. Ab ca. $x=0.03$ laufen alle Kurven kongruent. Das implizite Euler Verfahren gewinnt im Gegensatz zu den anderen Verfahren durch das Steigen der Schrittweite an Genauigkeit und nähert sich in diesem Ergebnis schneller der analytischen Lösung als bei der Schrittweite von 0.001.

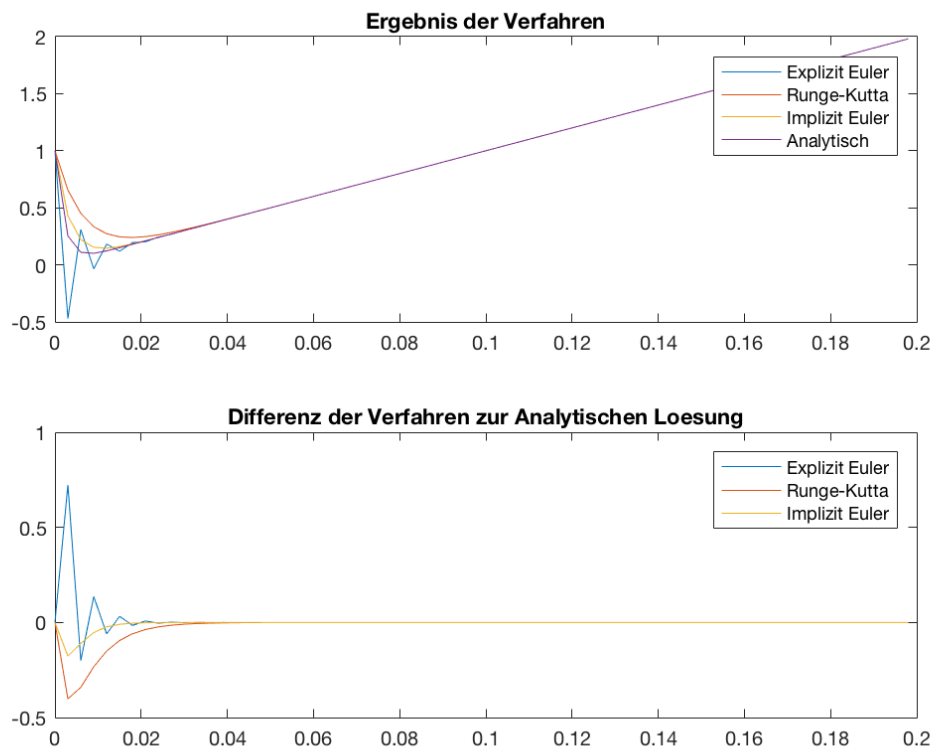


Figure 3: $h=0.003$

1.2.3 $h=0.004$

Bis ca. $x=0.01$ laufen alle Kurven bis auf das des impliziten Euler Verfahrens unterschiedlich zur analytischen Lösung. Ab dort laufen die Kurven des Runge-Kutta-Verfahrens und der analytischen Lösung parallel während das implizite Euler Verfahren eine nahezu kongruente Kurve liefert. Während der ganzen Laufzeit liefert das explizite Euler-Verfahren sehr schwankende Werte, die sich mit einer Differenz von ± 1 in der Nähe der analytischen Lösung befinden.

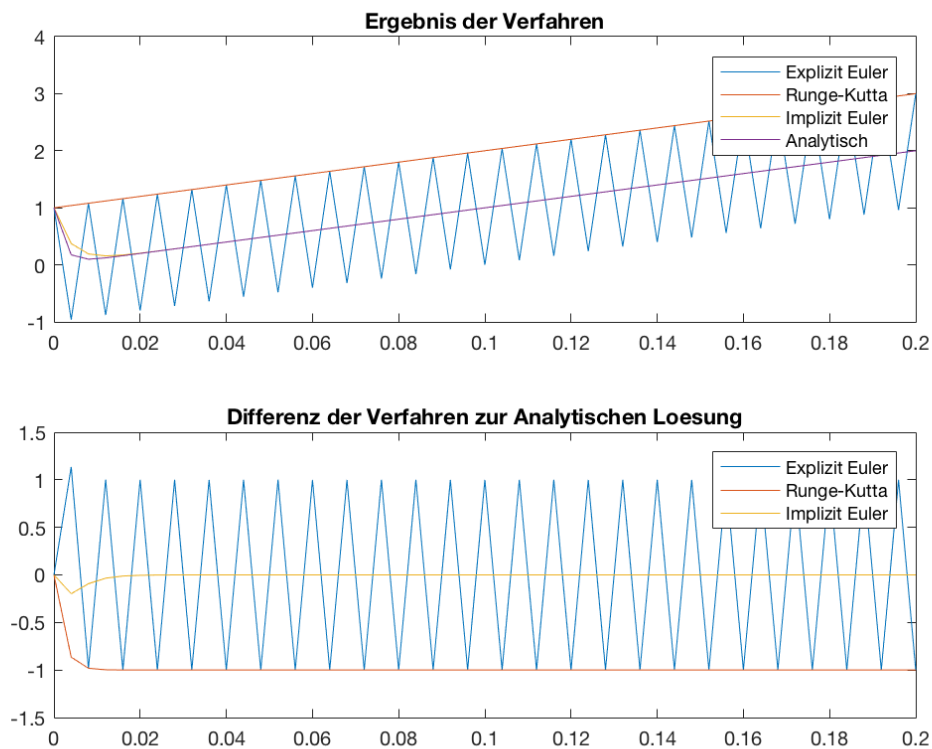


Figure 4: $h=0.004$

1.2.4 $h=0.005$

Bis ca. $x=0.15$ laufen alle Kurven kongruent. Ab dort schwanken die Werte des expliziten Euler-Verfahrens mit einer Differenz von ca. ± 0.1 um die Werte der analytischen Lösung. Die Werte des Runge-Kutta-Verfahrens hingegen werden ab ca. $x=0.15$ exponentiell größer und die Werte des impliziten Eulers bleiben kongruent mit der analytischen Lösung.

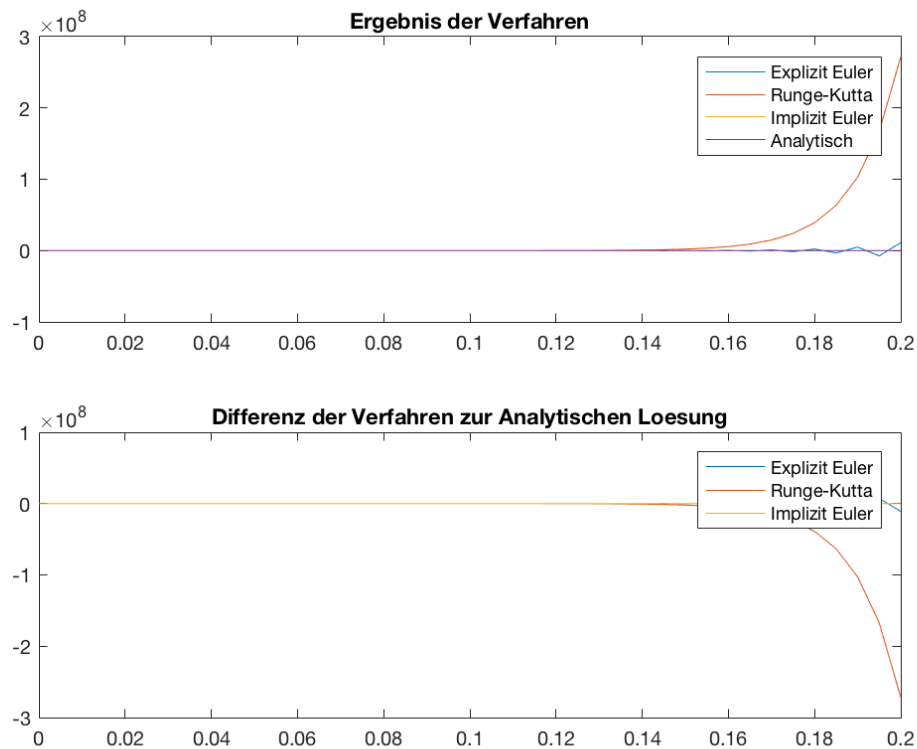


Figure 5: $h=0.005$

1.3 Interpretation der Ergebnisse

Die Ergebnisse zeigen, dass die expliziten Verfahren bei einer geringen Schrittweite genauere Ergebnisse liefern. Allerdings wird dadurch die benötigte Rechenzeit erhöht. Dahingegen liefert das implizite Euler Verfahren auch bei höherer Schrittweite recht genaue Lösungen.

2 Teilaufgabe 2 - Nicht Lineare DGL und Van der Pol

Die folgende DGL ist gegeben:

$$y'' = 6 * (1 - y^2) * y' - y,$$

$$y(0) = 0,$$

$$y'(0) = 1$$

Für diese DGL ist ein Schaltbild in Simulink zu entwerfen. Des weiteren sollen die Verfahren (Runge-Kutta 2. Ordnung und expliziter Euler) erarbeitet und Iterationsgleichungen für die jeweiligen Verfahren angegeben werden, dafür muss die DGL 2. Ordnung

in zwei DGLn 1. Ordnung umgewandelt werden. Im Anschluss soll ein Programm entwickelt (*vanderpol*) werden das jedes Verfahren auf die DGL anwendet und in einem Plot zum Vergleich mit der gegebenen analytischen Lösung anzeigt.

Das Programm (*stiff*) soll mit zwei Parameter Varianten gestartet werden, die sich wie folgt beschreiben:

- $h = 0.001, t_{End} = 31$
- $h = 0.02, t_{End} = 31$

2.1 Schaltbild

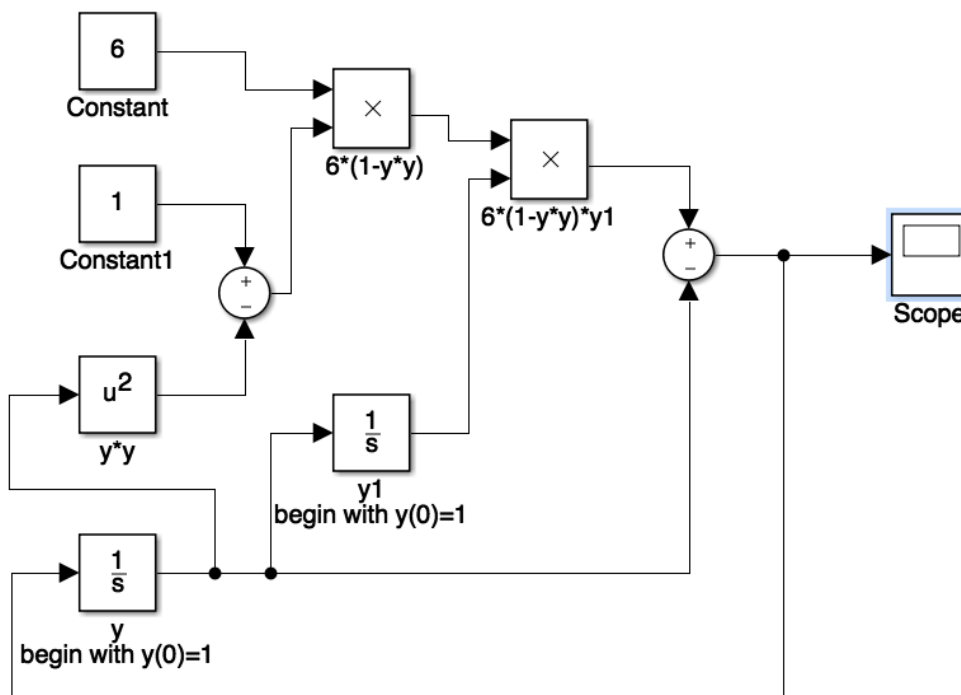


Figure 6: Simulink Schaltbild

2.2 DGLn der 1.Ordnung

$$\tilde{y} = y' \quad (13)$$

$$\tilde{y}' = 6 \cdot (1 - y^2) \cdot \tilde{y} - y \quad (14)$$

$$y(0) = 0 \quad (15)$$

$$\tilde{y}(0) = 1 \quad (16)$$

2.3 Iterationsgleichungen

2.3.1 Euler-Verfahren

$$x_{n+1} = x_n + h \quad (17)$$

$$\tilde{y}_{n+1} = \tilde{y}_n + h * (6 * (1 - y_n^2) * \tilde{y}_n - y_n) \quad (18)$$

$$y_{n+1} = y_n + h * \tilde{y}_n \quad (19)$$

2.3.2 Runge-Kutta-Verfahren 2.Ordnung

$$x_{n+1} = x_n + h \quad (20)$$

$$\tilde{k}_1 = h * (6 * (1 - y_n^2) * \tilde{y}_n - y_n) \quad (21)$$

$$k_1 = h * \tilde{y}_n \quad (22)$$

$$\tilde{k}_2 = h * (6 * (1 - (y_n + \frac{k_1}{2})^2) * (\tilde{y}_n + \frac{\tilde{k}_1}{2}) - (y_n + \frac{k_1}{2})) \quad (23)$$

$$k_2 = h * (\tilde{y}_n + \frac{k_1}{2}) \quad (24)$$

$$\tilde{y}_{n+1} = \tilde{y}_n + \tilde{k}_2 \quad (25)$$

$$y_{n+1} = y_n + k_2 \quad (26)$$

2.4 Plot der Lösungen

2.4.1 $h = 0.001$

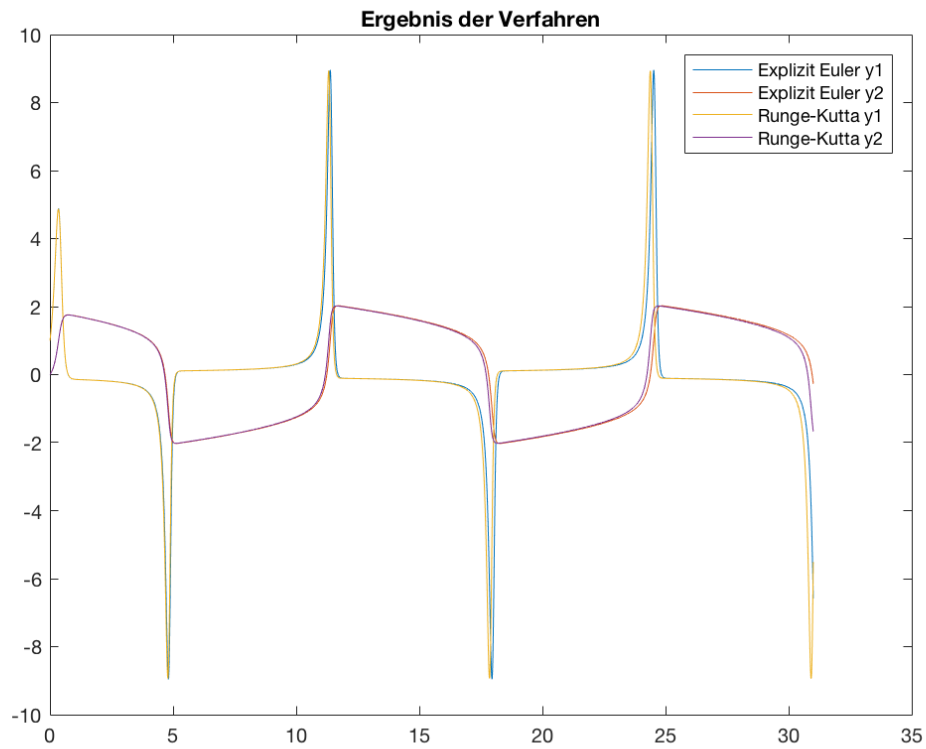


Figure 7: $h=0.001$

Anhand des Ergebnisses lässt sich erkennen, dass beide Verfahren (Runge-Kutta 2. Ordnung und Expliziter Euler) bei einer Schrittweite von 0.001 kongruente Ergebnisse liefern.

2.4.2 $h = 0.02$

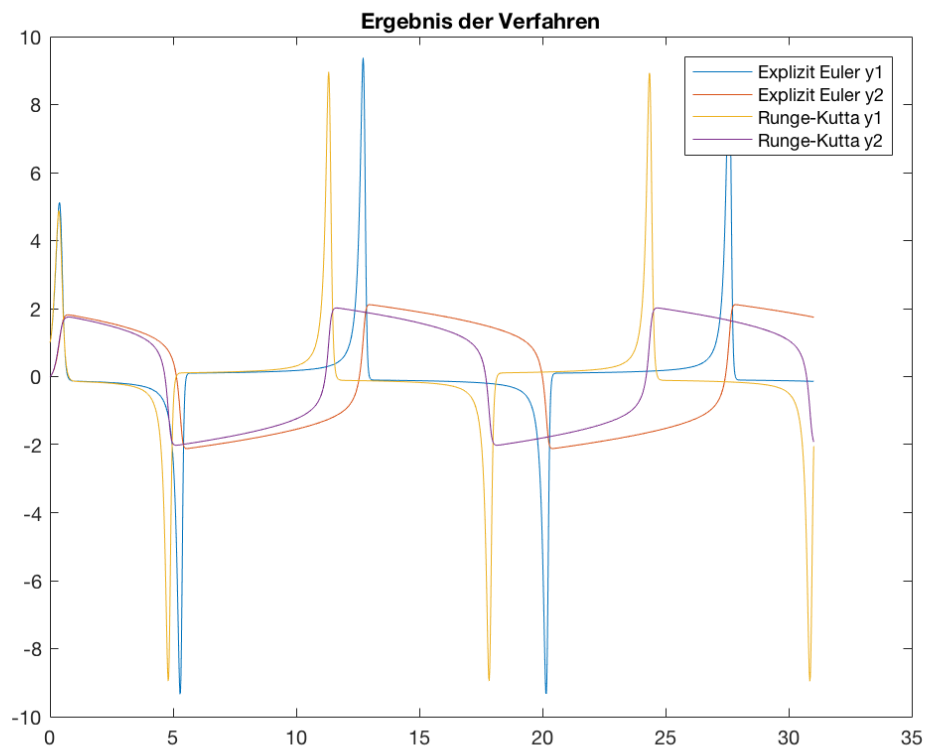


Figure 8: $h=0.02$

In diesem Experiment mit einer Schrittweite von 0.02 verändern sich die Ergebnisse des expliziten Euler verfahrens und verschieben sich merklich gegenüber der Lösung mit einer Schrittweite von 0.001. Dahingegen verändert sich das komplexere Runge-Kutta 2. Ordnung verfahren nicht und behält das Ergebnis bei.

3 Teilaufgabe 3 - Lorenz-Attraktor

Es ist folgendes DGL-System gegeben:

$$x' = -10 * (x * y), x(0) = 0,01$$

$$y' = (40 - z) * x - y, y(0) = 0,01$$

$$z' = x * y - 2,67 * z, z(0) = 0,00$$

Für dieses DGL System sollen die Iterationsgleichungen angegeben werden. Daraus wird ein Programm (*lorenz*) entwickelt, das das DGL System löst und in zwei verschiedenen Plots sollen die Ergebnisse von $x(t)$ und von $z(t)$ dargestellt werden. Das

DGL System soll des Weiteren mit Simulink gelöst werden.

Das Programm soll mit folgenden Parametern gestartet werden:

- $h = 0.002, t_{End} = 120$

Ein weiterer Vergleich soll stattfinden in dem die Funktion $y'(t)$ verändert wird. Sodass das System einmal mit $y' = (40 - z) * x - y$ und einmal mit $y' = (40.000000001 - z) * x - y$ gelöst wird. Die beiden Lösungen sollen gegenüber gestellt und verglichen werden.

3.1 Iterationsgleichungen

Im Folgenden die Iterationsgleichungen für das Runge-Kutta-Verfahren 2ter Ordnung bezogen auf das gegebene DGL-System.

$$k_1 = h * (-10 * (x_n * y_n)) \quad (27)$$

$$l_1 = h * ((40 - z) * x_n - y_n) \quad (28)$$

$$m_1 = h * (x_n * y_n - 2,67 * z_n) \quad (29)$$

$$k_2 = h * f(x_n + \frac{k_1}{2}, y_n + \frac{l_1}{2}, z_n + \frac{m_1}{2}) \quad (30)$$

$$k_2 = h * (-10 * ((x_n + \frac{k_1}{2}) * (y_n + \frac{l_1}{2}))) \quad (31)$$

$$l_2 = h * f(x_n + \frac{k_1}{2}, y_n + \frac{l_1}{2}, z_n + \frac{m_1}{2}) \quad (32)$$

$$l_2 = h * ((40 - (z_n + \frac{m_1}{2})) * (x_n + \frac{k_1}{2}) - (y_n + \frac{l_1}{2})) \quad (33)$$

$$m_2 = h * f(x_n + \frac{k_1}{2}, y_n + \frac{l_1}{2}, z_n + \frac{m_1}{2}) \quad (34)$$

$$m_2 = h * ((x_n + \frac{k_1}{2}) * (y_n + \frac{l_1}{2}) - 2,67 * (z_n + \frac{m_1}{2})) \quad (35)$$

$$x_{n+1} = x_n + k_2 \quad (36)$$

$$y_{n+1} = y_n + l_2 \quad (37)$$

$$z_{n+1} = z_n + m_2 \quad (38)$$

3.2 Plot der Lösung

Gelöst durch das Runge-Kutta-Verfahren 2ter Ordnung.

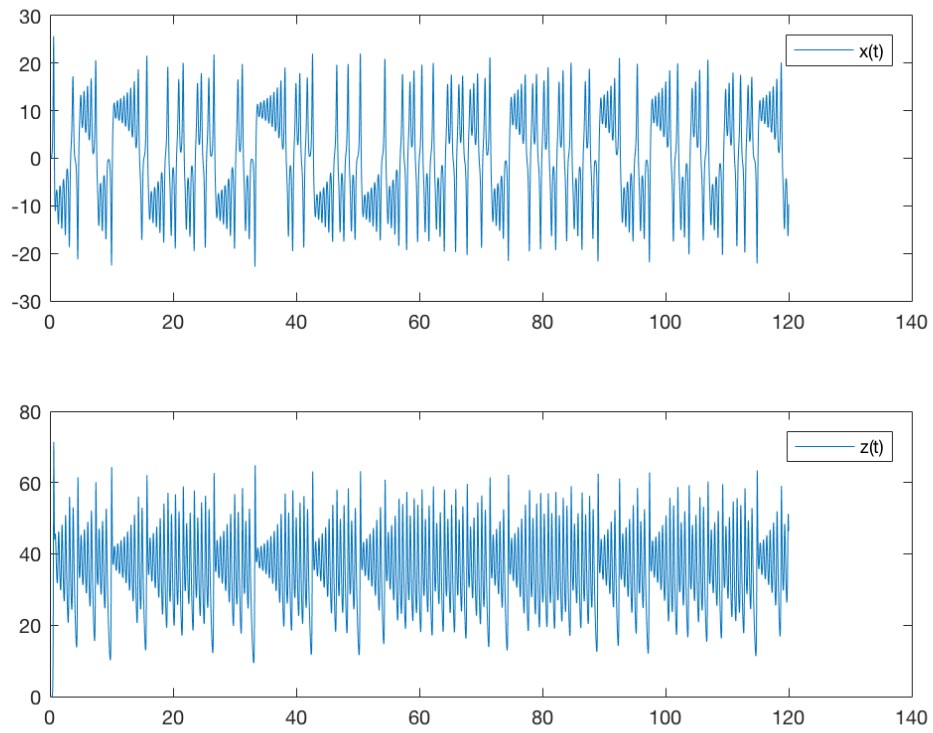


Figure 9: Vergleich von $x(t)$ und $z(t)$, $h=0.002$, $t_{end} = 120$

3.3 Schaltbild

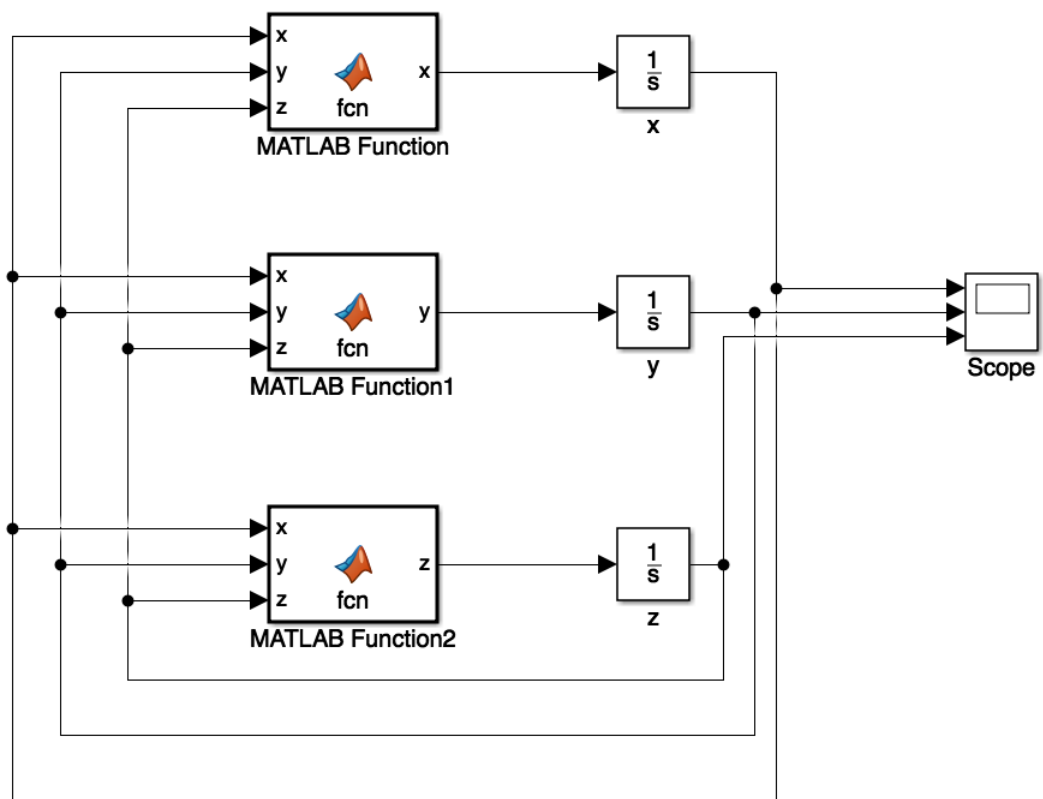


Figure 10: Simulink Schaltbild

3.4 Vergleich der Simulationen

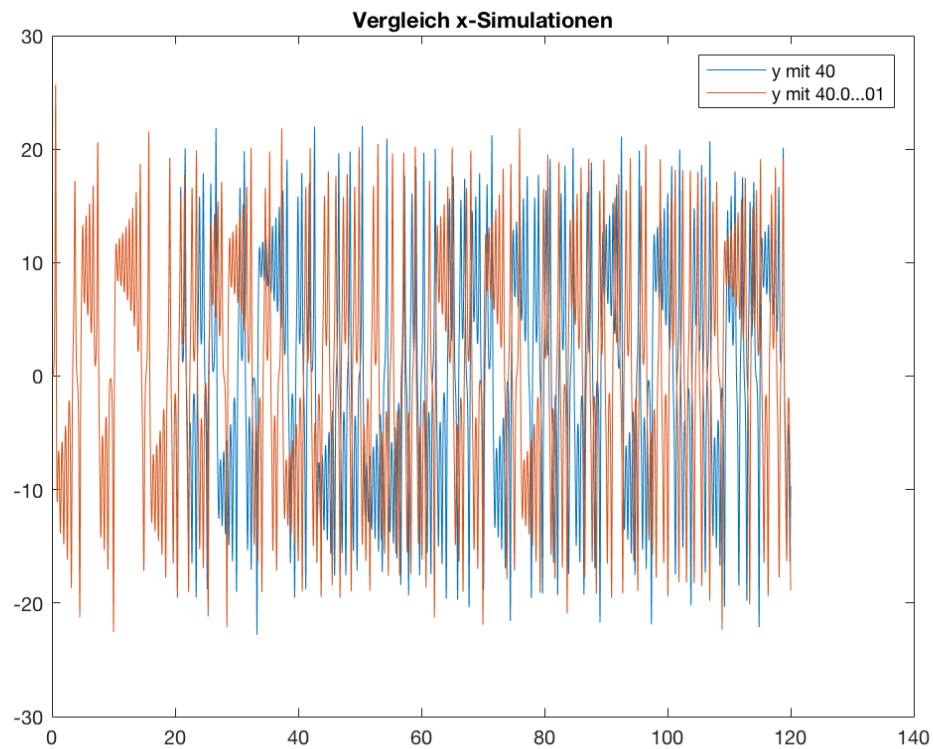


Figure 11:

Durch das ausführen der zweiten Funktion $y = \dots$ mit 40.000000001 anstatt von 40 ergibt sich obenstehende Verschiebung zwischen den beiden Funktionen. Die Grafik zeigt die Lösungen des Systems mit diesem marginalen Unterschied. Aufgrund von Rundungsfehlern scheint sich die Berechnung des Systems so stark zu verändern dass sich dies in der Grafik sichtbar niederschlägt.