

## Übungsaufgabe Numerische Lösung DGL

1. Bestimmen Sie die Lösung der DGL

$$\dot{x} = (t-2)^2 + \frac{t}{x} \quad \text{mit} \quad x(0) = 1$$

im Bereich von  $t = 0$  bis  $t = 5$  mit Hilfe des Euler'schen Streckenzugverfahrens, des Leapfrog-Verfahrens und des Runge-Kutta-Verfahrens 4. Ordnung für Schrittweiten entsprechend 2, 5, 10, 25 und 50 Schritten. Vergleichen Sie die Lösungen untereinander und mit der analytischen Lösung. (Hinweis: den Algorithmus für die einzelnen Lösungen sowie die Anwendung auf 2 bzw. 5 Schritte in den regulären Übungen, da sich das mit Verstand und Taschenrechner lösen lässt. Die Numerik und die Lösung für große Zahlen von Schritten in den MatLab-Tutorien.)

*Lösung:* die Frage nach der analytischen Lösung ist eine Fangfrage. Multiplikation der DGL mit  $x$  zeigt, dass es sich um eine nicht-lineare DGL handelt:

$$x \dot{x} = x(t-2)^2 + t.$$

Da wir vor dem ersten Term auf der rechten Seite noch ein  $x$  stehen haben, ist diese DGL auch nicht separabel. Daher steht uns kein Standardverfahren zur Bestimmung der analytischen Lösung zur Verfügung. Das bedeutet auch, dass wir bei der numerischen Lösung sehr sorgfältig vorgehen müssen, da wir etwaige Ungenauigkeiten oder gar Fehler nicht durch Vergleich mit der analytischen Lösung aufspüren können.

(a) Euler'sches Streckenzug-Verfahren: Ziel ist die numerische Lösung des Anfangswertproblem

$$\dot{x} = (t-2)^2 + \frac{t}{x}$$

mit dem Anfangswert  $x(0) = x_0$  in einem Intervall  $a \leq t \leq b$ . Als erstes wird die Differentialgleichung in eine Differenzgleichung umgewandelt

$$\frac{\Delta x}{\Delta t} = \frac{x_n - x_{n-1}}{\Delta t} = f(x, t)$$

und das zu betrachtende Intervall  $[a, b]$  in  $M$  gleich große Teile der Länge

$$h = \Delta x = \frac{b-a}{M}$$

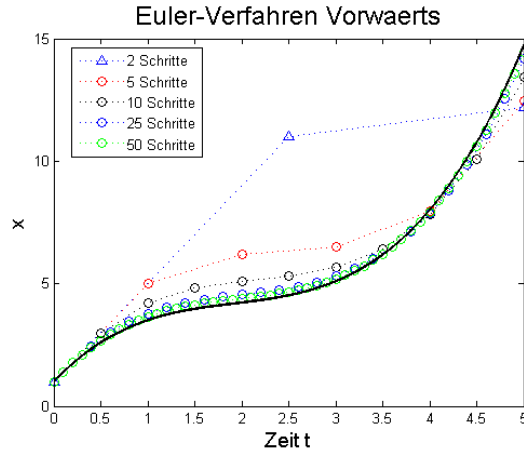
zerlegt.  $h = \Delta x$  ist die *Schrittweite* des numerischen Schemas. Die Lösungsfunktion wird punktweise an den durch die Diskretisierung ausgewählten Gitterpunkten  $t_n = t_0 + n \Delta t$ ,  $n = 0, 1, 2, 3, \dots$  bestimmt gemäß

$$x(t_{i+1}) = x(t_i) + \Delta t f(x_i, t_i).$$

Übertragen auf die Funktion der Aufgabenstellung ergibt sich

$$x(t_{i+1}) = x(t_i) + \Delta t (t_i - 2)^2 + \frac{t_i}{x_i}.$$

Die Lösungen für die verschiedenen Schrittweiten sind in der Tabelle weiter unten gegeben, die dazu gehörigen Funktionsgraphen der folgenden Abbildung.



Da keine analytische Lösung vorliegt, können wir nur die numerischen Lösungen für verschiedene Schrittweiten untereinander vergleichen. Mit zunehmender Schrittweite liegen die Lösungskurven dichter zusammen, allerdings zeigen sich selbst zwischen den Lösungen mit Schrittweiten von 0.1 und 0.2 (entsprechend 50 bzw. 25 Schritten) noch Unterschiede. Mit weiter zunehmender Schrittzahl wird der Fehler des numerischen Verfahrens kleiner, d.h. Kurven mit größerer Schrittzahl sollten weniger von der 50-Schritte Kurve abweichen als es die 25-Schritte-Kurve tut. Die Differenz zwischen der Lösung für 50 und 25 Schritte können wir daher als grobe Annäherung an den Fehler unseres numerischen Verfahrens betrachten. Das wird auch durch die durchgezogene Kurve bestätigt, die für eine Schrittweite von 0.001 bestimmt ist, entsprechend 5000 Gitterpunkten.

(b) Beim Leapfrog-Verfahren werden zwei ineinander geschachtelte Gitter iterativ verwendet. Ein Gitter ist wie beim Euler-Verfahren durch die Randpunkte des zu betrachtenden Intervalls fest gelegt, allerdings wird die Steigung nicht im Anfangspunkt des Intervalls sondern in der Intervallmitte betrachtet:

$$x_{i+1} = x_i + f(t_{i+\frac{1}{2}}, x_{i+\frac{1}{2}})\Delta t \quad \text{mit} \quad x_0 = x_0 .$$

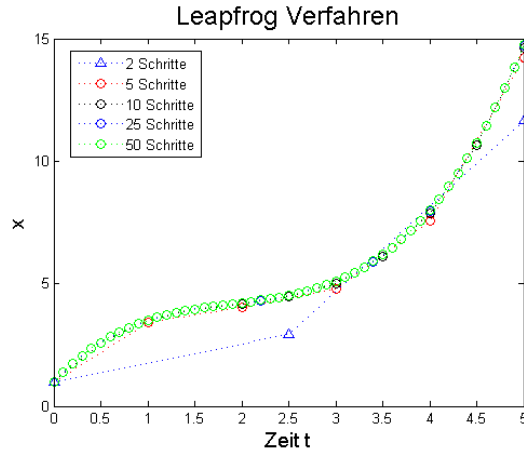
Um letztere zu bestimmen, wird ein zweites Gitter benötigt, dass an den Intervallmitten fest gelegt ist und als Steigung jeweils die Steigungen in seinen eigenen Intervallmitten (und damit an den Anfangspunkten der Intervalle des ersten Gitters) betrachtet:

$$x_{i+\frac{1}{2}} = x_{i-\frac{1}{2}} + f(t_i, x_i)\Delta t .$$

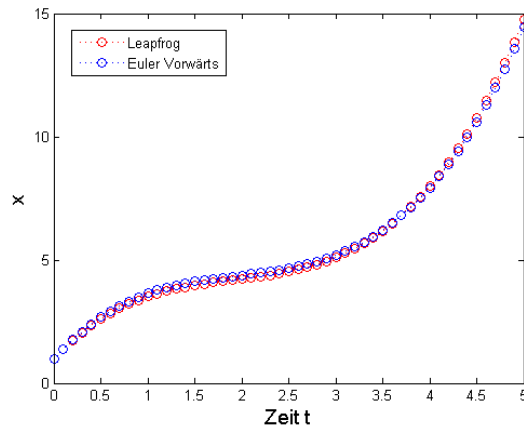
Da nur ein Anfangswert gegeben ist, muss zuerst der Anfangswert für das zweite Gitter aus dem ersten Anfangswert bestimmt werden:

$$x_{-\frac{1}{2}} = x_0 - f(t_0, x_0)\frac{1}{2}\Delta t .$$

Die Lösungen an den Punkten des ursprünglichen Gitters für verschiedene Schrittweiten sind in der Tabelle am Ende gegeben, die dazu gehörigen Funktionsgraphen in der folgenden Abbildung:



Das Leapfrog-Verfahren liefert wesentlich schneller eine genaue Lösung: die 50-Schritt Kurve liegt in der Abbildung im Rahmen der Zeichengenauigkeit bereits auf der 25-Schritt Kurve (deshalb sind die blauen Kreise der letzteren von den grünen Kreisen der ersteren überlagert und nicht mehr als getrennte Symbole zu erkennen). Der höhere Rechenaufwand durch die Berücksichtigung des Zwischengitters führt bereits früh zu einer genaueren Lösung als beim Euler-Verfahren.



Ein Vergleich der beiden 50-Schritt Lösungen für Euler Vorwärts und Leapfrog zeigt Abweichungen, die denen zwischen der 50-Schritt Lösung von Euler-Vorwärts und der 5000-Schritt Lösung in Euler Vorwärts entsprechen, d.h. das Leapfrog Verfahren hat mit 50 Schritten die exakte Lösung bereits sehr gut angenähert (die fast verschwindende Abweichung zwischen der 25- und der 50-Schritt Lösung gibt ja den Fehler des Verfahrens).

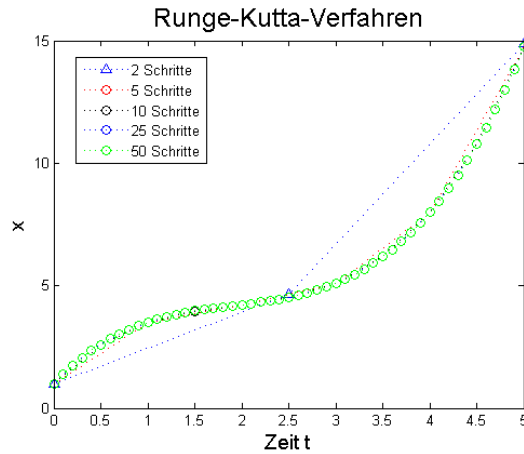
(c) Beim Runge-Kutta Verfahren 4. Ordnung wird können wir uns, wie beim Euler-Verfahren, wieder auf ein Gitter beschränken. Die Steigung wird hierbei aus den Werten an den Intervallrändern und in der Intervallmitte zusammen gesetzt:

$$x(t_i) = x_i = x_{i-1} + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

mit

$$\begin{aligned} k_1 &= \Delta t f(t_{i-1}, x_{i-1}), & k_2 &= \Delta t f\left(t_{i-\frac{1}{2}}, x_{i-1} + \frac{k_1}{2}\right), \\ k_3 &= \Delta t f\left(t_{i-\frac{1}{2}}, x_{i-1} + \frac{k_2}{2}\right), & k_4 &= \Delta t f(t_{i-1} + \Delta t, x_{i-1} + k_3). \end{aligned}$$

Dabei müssen die  $k_i$  für jeden Rechenschritt  $\Delta x$  neu berechnet werden. Die Ergebnisse für die einzelnen Schrittweiten sind in der Tabelle am Ende gegeben, die Plots in der folgenden Abbildung:



Bei Runge-Kutta sind bereits die Lösungen für 10, 25 und 50 Schritte im Rahmen der Zeichengenauigkeit nicht mehr zu unterscheiden, d.h. das Verfahren liefert bereits bei relativ großen Schrittweiten sehr genaue Lösungen – größere Schrittweiten machen hier keinen im Gegensatz zum Euler-Verfahren keinen Sinn mehr.

MATLAB Funktionen für die verschiedenen Verfahren finden Sie unter stud.ip. Zur Anwendung müssen Sie eine Funktion über ein Handle definieren, z.B.

```
f=@(t,T) -0.33*(T-(3-5*cos(pi/12*(t-2))))
```

für Aufgabe 5. Die Funktionen für die numerischen Verfahren lauten `eulervorwaerts`, `eulerrueckwaerts`, `leapfrog` und `rungekutta`, der Aufruf folgt jeweils der Syntax

```
[t,x]=rungekutta(f,a,b,dt,x0)
```

mit den Argumenten `f` für die über das Handle definierte Funktion, `a` und `b` als untere und obere Grenze des Integrationsintervalls, `dt` als der Schrittweite und `x0` als dem Anfangswert.

t	Euler				
	2	5	10	25	50
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.1	-	-	-	-	1.4000
0.2	-	-	-	1.8000	1.7681
0.3	-	-	-	-	2.1035
0.4	-	-	-	2.4702	2.4067
0.5	-	-	3.0000	-	2.6793
0.6	-	-	-	3.0146	2.9320
0.7	-	-	-	-	3.1395
0.8	-	-	-	3.4464	3.3308
0.9	-	-	-	-	3.4988
1.0	-	5.0000	4.2083	3.7808	3.6456
1.1	-	-	-	-	3.7730
1.2	-	-	-	4.0337	3.8831
1.3	-	-	-	-	3.9791
1.4	-	-	-	4.2212	4.0597
1.5	-	-	4.8271	-	4.1302
1.6	-	-	-	4.3596	4.1915
1.7	-	-	-	-	4.2457
1.8	-	-	-	4.4650	4.2947
1.9	-	-	-	-	4.3407
2.0	-	6.2000	5.1075	4.5536	4.3854
2.1	-	-	-	-	4.4310
2.2	-	-	-	4.6414	4.4794
2.3	-	-	-	-	4.5325
2.4	-	-	-	4.7442	4.5923
2.5	11.0000	-	5.3033	-	4.6605
2.6	-	-	-	4.8774	4.7392
2.7	-	-	-	-	4.8300
2.8	-	-	-	5.0560	4.9349
2.9	-	-	-	-	5.0557
3.0	-	6.5226	5.6640	5.3948	5.1940
3.1	-	-	-	-	5.3518
3.2	-	-	-	5.6081	5.5307
3.3	-	-	-	-	5.7326
3.4	-	-	-	6.0102	5.9592
3.5	-	-	6.4288	-	6.2122
3.6	-	-	-	6.5154	6.4936
3.7	-	-	-	-	6.8050
3.8	-	-	-	7.1379	7.1484
3.9	-	-	-	-	7.5255
4.0	-	7.9825	7.8260	7.8923	7.9383
4.1	-	-	-	-	8.3887
4.2	-	-	-	8.7937	8.8786
4.3	-	-	-	-	9.4099
4.4	-	-	-	9.8572	9.9846
4.5	-	-	10.0816	-	10.6047
4.6	-	-	-	11.0985	11.2721
4.7	-	-	-	-	11.9889
4.8	-	-	-	12.5334	12.7571
4.9	-	-	-	-	13.5788
5.0	12.1932	12.4836	13.4298	14.1780	14.4558

t	Leapfrog				
	2	5	10	25	50
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.1	-	-	-	-	1.3844
0.2	-	-	-	1.7363	1.7362
0.3	-	-	-	-	2.0556
0.4	-	-	-	2.3433	2.3437
0.5	-	-	2.5938	-	2.6021
0.6	-	-	-	2.8315	2.8325
0.7	-	-	-	-	3.0369
0.8	-	-	-	3.2153	3.2171
0.9	-	-	-	-	3.3751
1.0	-	3.4167	3.4914	3.5105	3.5128
1.1	-	-	-	-	3.6324
1.2	-	-	-	3.7328	3.7358
1.3	-	-	-	-	3.8251
1.4	-	-	-	3.8985	3.9023
1.5	-	-	3.9344	-	3.9693
1.6	-	-	-	4.0239	4.0283
1.7	-	-	-	-	4.0812
1.8	-	-	-	4.1249	4.1300
1.9	-	-	-	-	4.1767
2.0	-	4.0161	4.1750	4.2176	4.2234
2.1	-	-	-	-	4.2718
2.2	-	-	-	4.3176	4.3240
2.3	-	-	-	-	4.3819
2.4	-	-	-	4.4403	4.4474
2.5	2.9271	-	4.4607	-	4.5222
2.6	-	-	-	4.6006	4.6083
2.7	-	-	-	-	4.7074
2.8	-	-	-	4.8130	4.8213
2.9	-	-	-	-	4.9519
3.0	-	4.7879	5.0270	5.0919	5.1008
3.1	-	-	-	-	5.2698
3.2	-	-	-	5.4513	5.4608
3.3	-	-	-	-	5.6754
3.4	-	-	-	5.9054	5.9154
3.5	-	-	6.0972	-	6.1827
3.6	-	-	-	6.4684	6.4790
3.7	-	-	-	-	6.8062
3.8	-	-	-	7.1549	7.1661
3.9	-	-	-	-	7.5606
4.0	-	7.5833	7.8950	7.9799	7.9916
4.1	-	-	-	-	8.4611
4.2	-	-	-	8.9587	8.9710
4.3	-	-	-	-	9.5232
4.4	-	-	-	10.1069	10.1197
4.5	-	-	10.6550	-	10.7626
4.6	-	-	-	11.4405	11.4538
4.7	-	-	-	-	12.1954
4.8	-	-	-	12.9755	12.9894
4.9	-	-	-	-	13.8378
5.0	11.6535	14.2445	14.6243	14.7283	14.7427

t	Runge-Kutta 4ter-Ordnung				
	2	5	10	25	50
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.1	-	-	-	-	1.3843
0.2	-	-	-	1.7361	1.7362
0.3	-	-	-	-	2.0556
0.4	-	-	-	2.3438	2.3438
0.5	-	-	2.6017	-	2.6023
0.6	-	-	-	2.8328	2.8329
0.7	-	-	-	-	3.0373
0.8	-	-	-	3.2176	3.2176
0.9	-	-	-	-	3.3757
1.0	-	3.5123	3.5130	3.5136	3.5136
1.1	-	-	-	-	3.6333
1.2	-	-	-	3.7368	3.7368
1.3	-	-	-	-	3.8262
1.4	-	-	-	3.9034	3.9035
1.5	-	-	3.9701	-	3.9706
1.6	-	-	-	4.0296	4.0297
1.7	-	-	-	-	4.0827
1.8	-	-	-	4.1316	4.1316
1.9	-	-	-	-	4.1785
2.0	-	4.2246	4.2247	4.2252	4.2252
2.1	-	-	-	-	4.2738
2.2	-	-	-	4.3261	4.3261
2.3	-	-	-	-	4.3841
2.4	-	-	-	4.4496	4.4497
2.5	4.6730	-	4.5242	-	4.5246
2.6	-	-	-	4.6108	4.6108
2.7	-	-	-	-	4.7100
2.8	-	-	-	4.8240	4.8240
2.9	-	-	-	-	4.9547
3.0	-	5.1033	5.1033	5.1037	5.1037
3.1	-	-	-	-	5.2728
3.2	-	-	-	5.4638	5.4639
3.3	-	-	-	-	5.6785
3.4	-	-	-	5.9187	5.9187
3.5	-	-	6.1857	-	6.1860
3.6	-	-	-	6.4824	6.4824
3.7	-	-	-	-	6.8097
3.8	-	-	-	7.1697	7.1697
3.9	-	-	-	-	7.5644
4.0	-	7.9958	7.9952	7.9955	7.9955
4.1	-	-	-	-	8.4651
4.2	-	-	-	8.9750	8.9750
4.3	-	-	-	-	9.5273
4.4	-	-	-	10.1239	10.1239
4.5	-	-	10.7666	-	10.7669
4.6	-	-	-	11.4582	11.4582
4.7	-	-	-	-	12.1999
4.8	-	-	-	12.9939	12.9939
4.9	-	-	-	-	13.8424
5.0	14.8981	14.7481	14.7471	14.7474	14.7474