Physically-based Simulation
Exercise 1

Alexander Lelidis (14-907-562),
Andreas Emch (08-631-384),
Uroš Tešić (17-950-346)

October 11, 2017

1 Analytic solution and results analysis

1.1 Analytic solution

To find the parameters c_1 and c_2 we use the inital values from the given codebase. The starting position $(t = 0)$ is at $y = -1$.

$$y(t) = c_1 e^{\alpha t} \cos(\beta t) + c_2 e^{\alpha t} \sin(\beta t) - L - \frac{mg}{k} \tag{1}$$

By differentiating this equation with respect to t we get.

$$y'(t) = c_1 e^{\alpha t}(\alpha \cos(\beta t) - \beta \sin(\beta t)) + c_2 e^{\alpha t}(\alpha \sin(\beta t) + \beta \cos(\beta t)) \tag{2}$$

By inserting $t = 0$ and $y_0 = -1$ into (1) we get:

$$-1 = c_1 e^{\alpha 0} \cos(\beta 0) + c_2 e^{\beta 0} \sin(\beta 0) - L - \frac{mg}{k} \tag{3}$$

$$-1 = c_1 e^{\alpha 0} - L - \frac{mg}{k} \tag{4}$$

$$c_1 = -1 + L + \frac{mg}{k} \tag{5}$$

To calculate c_2 we replace the starting speed with $y'(0) = 0$.

$$0 = c_1 e^{\alpha 0}(\alpha \cos(\beta 0) - \beta \sin(\beta 0)) + c_2 e^{\alpha 0}(\alpha \sin(\beta 0) + \beta \cos(\beta 0)) \tag{6}$$

$$0 = c_1 \alpha \cos(\beta 0) + c_2 \beta \cos(\beta 0) \tag{7}$$

$$0 = c_1 \alpha + c_2 \beta \tag{8}$$

$$c_2 = \frac{-c_1 \alpha}{\beta} \tag{9}$$

Subsituting c_1 to ge the value for c_2.

$$c_2 = -\frac{\alpha}{\beta}(-1 + L + \frac{mg}{k}) \tag{10}$$

1.2 Error convergence analysis

Please note that in the following two tables only the error convergence ($\frac{e_i}{e_{i+1}}$) with no damping for each method is shown.

1.2.1 Velocity change table:

step	euler	symplectic-euler	midpoint	backwards-euler
0.500000	1.503101	1.503101	77.334118	0.636422
0.250000	3.442190	3.442190	1.064007	3.509106
0.125000	4.090020	4.090020	4.998766	1.235972
0.062500	4.153509	4.153509	6.778438	3.245808
0.031250	4.104957	4.104957	7.453338	4.044384
0.015630	4.059688	4.059688	7.747549	4.169692
0.007810	4.031496	4.031496	7.846154	4.125019
0.003910	4.016445	4.016445	8.666667	4.074097
0.001950	4.007605	4.007605	6.666667	4.039235
0.000980				

1.2.2 Displacement table:

step	euler	symplectic-euler	midpoint	backwards-euler
0.500000	7.380872	4.529371	4.915443	1.527721
0.250000	5.375022	5.502809	7.579571	1.682016
0.125000	1.701236	5.106089	8.263111	6.642051
0.062500	3.129251	4.618190	8.264813	23.329893
0.031250	3.622619	4.319823	8.171322	0.748414
0.015630	3.825027	4.161173	8.087349	2.847380
0.007810	3.914942	4.081192	8.097561	3.514909
0.003910	3.958042	4.041136	8.200000	3.776851
0.001950	3.981702	4.018103	7.692308	3.893626
0.000980				

1.2.3 Average error table:

The following tables shows the average error of the velocity change and displacement. (The data for the damping factor 0.5 can be found in the attached excel).

Velocity	damp = 0	damp = 0.5
euler	3.712112	2.117509
symplectic$_e$uler	3.712112	2.117509
midpoint	14.283967	12.611934
backwards$_e$uler	3.231082	1.890506

Displacement	damp = 0	damp = 0.5
euler	4.098746	2.400098
symplectic$_e$uler	4.486432	2.589986
midpoint	7.696831	2.510173
backwards$_e$uler	5.329207	2.111750

1.2.4 Analysis:

Based on the top two tables, we can see that the explicit, sympletic- and backward-Euler converge with $O(h^2)$ and the midpoint-method with $O(h^3)$. Therefore, the error converges faster only for the midpoint-method. This holds for both, the velocity change as well the displacement. If the damping is set to 0.5, the errors for all methods does not converge as fast anymore as without damping. It seems like there is an additional error by adding the damping-factor. Some of the causes of the slower convergence and larger error could be rounding errors due to more calculations involved, and different methods underestimating or overestimating the damping.

1.3 Stability analysis

1.3.1 Damping: 0

step	euler	symplectic-euler	midpoint	backwards-euler	analytic
0.000100	2.2	2.2	2.2	2.2	2
0.000200	2.22	2.2	2.2	2.2	2
0.000400	2.32	2.2	2.2	2.2	2
0.000800	$8.47 \cdot 10^{34}$	2.2	2.2	2.19	2
0.001600	∞	2.2	2.2	2.19	2
0.003200	∞	2.2	2.2	2.19	2
0.006400	∞	2.2	2.22	2.19	2
0.012800	∞	2.2	$1.89 \cdot 10^{307}$	2.18	2
0.025600	∞	2.2	$1.84 \cdot 10^{307}$	2.16	2
0.051200	∞	2.2	∞	2.14	2.01

1.3.2 Damping: 0.5

step	euler	symplectic-euler	midpoint	backwards-euler	analytic
0.000100	2.14	2.14	2.14	2.14	2
0.000200	2.14	2.14	2.14	2.14	2
0.000400	2.14	2.14	2.14	2.14	2
0.000800	2.14	2.14	2.14	2.14	2
0.001600	2.14	2.14	2.14	2.14	2
0.003200	2.14	2.14	2.14	2.14	2
0.006400	2.15	2.14	2.14	2.14	2
0.012800	2.15	2.14	2.14	2.13	2
0.025600	2.16	2.14	2.14	2.13	2
0.051200	3.26	2.14	2.14	2.12	2.01

1.3.3 Analysis:

Two methods tend to gain energy over the time (explicit Euler and midpoint Euler), the others seem relatively stable and do not gain energy by doubling the step-sizes. The more accurate a method is, the more stable it seems to be.
After adding a 0.5 damping, the results changed significantly. Most of the methods are stable now, but still the explicit Euler seems to gain energy after doubling the steps 10 times. The damping factor works as a regularizer and keeps the energy mostly within the correct range. However, when the step-sizes are too large for the Euler, it will still fail.
Furthermore, backwards Euler tends to lose energy, which can be just barely seen from the maximum amplitude shrinking, but is evident during a transient simulation.