Physically-based Simulation
Exercise 1

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

October 9, 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

1.2.1 Velocity change table:

step	euler	symplectic-euler	midpoint	backwards-euler
0.500000	1.5	1.5	77.33	0.64
0.250000	3.44	3.44	−1.06	−3.51
0.125000	4.09	4.09	5	1.24
0.062500	4.15	4.15	6.78	3.25
0.031250	4.1	4.1	7.45	4.04
0.015630	4.06	4.06	7.75	4.17
0.007810	4.03	4.03	7.85	4.13
0.003910	4.02	4.02	8.67	4.07
0.001950	4.01	4.01	6.67	4.04
0.000980				

1.2.2 Displacement table:

step	euler	symplectic-euler	midpoint	backwards-euler
0.500000	7.38	4.53	4.92	−1.53
0.250000	−5.38	5.5	7.58	1.68
0.125000	1.7	5.11	8.26	6.64
0.062500	3.13	4.62	8.26	−23.33
0.031250	3.62	4.32	8.17	0.75
0.015630	3.83	4.16	8.09	2.85
0.007810	3.91	4.08	8.1	3.51
0.003910	3.96	4.04	8.2	3.78
0.001950	3.98	4.02	7.69	3.89
0.000980				

1.2.3 Average error table:

Velocity	damp = 0	damp = 0.5
euler	3.71	2.9
symplectic$_e$uler	3.71	4.49
midpoint	14.05	7.7
backwards$_e$uler	2.45	−0.19
Displacement	**damp = 0**	**damp = 0.5**
euler	2.87	1.98
symplectic$_e$uler	2.94	2
midpoint	2.89	1.99
backwards$_e$uler	1.27	2

1.2.4 Analysis:

Based on the top two tables, we can see that the explicit, sympletic- and backward-euler converges 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 (due to additional terms and roundings probably).

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.