Sensitivity Approximation and Verification

Solution to Harmonic Oscillator

Andrei Kramer <andreikr@kth.se>

May 3, 2021

This part of the documentation justifies the expressions used in `verify.R` where we check the results of the sensitivity approximation method against the expected analytical result. The solution to the initial value problem of the Harmonic Oscillator is of course well known. The important part is its parameter-sensitivity (parameter derivative). We treat the damping parameter c as a system input and the coefficiant k of the main *restoring force* as the *interesting* parameter for sensitivity calculations. This choice is arbitrary.

1 Model Definition

We will use a dampened harmonic oscillator model with unit mass:

$$\ddot{y} = -ky - c\dot{y}\,, \tag{1}$$

to compare numerical k-sensitivity approximation methods to the analytical solution sensitivity. This model is well suited for the purpose of illustration as the sensitivity is a scalar function.

To use ordinary differential equation solvers, we reformulate the model as a system of equatuions, with order one:

$$\begin{aligned}
\dot{v} &= -ky - cv\,, & v(0) &= v_0 \qquad (2)\\
\dot{y} &= v\,, & y(0) &= y_0 \qquad (3)\\
& & & \qquad (4)
\end{aligned}$$

This system has a known analytical solution:

$$
\begin{aligned}
\omega &= \sqrt{k}\,, \\
r &= \frac{c}{2\omega}\,, \\
\Rightarrow r\omega &= \frac{c}{2}\,, \\
y(t;k) &= a\exp(-r\omega t)\cos(\sqrt{1-r^2}\omega t+\phi)\,, \\
\Rightarrow v(t;k) &= -a\exp(-r\omega t)(r\omega)\cos(\sqrt{1-r^2}\omega t+\phi) \\
&\quad - a\exp(-r\omega t)\sin(\sqrt{1-r^2}\omega t+\phi)\sqrt{1-r^2}\omega\,, \\
&= -y(t;k)\left(r\omega+\tan(\sqrt{1-r^2}\omega t+\phi)\sqrt{1-r^2}\omega\right)\,,
\end{aligned}
\tag{5}
$$

where a and ϕ are to be determined from the initial conditions v_0 and y_0:

$$
\begin{aligned}
y(0;k) &= a\cos(\phi) = y_0\,, \\
v(0;k) &= -y_0 r\omega - y_0\tan(\phi)\sqrt{1-r^2}\omega = v_0\,,
\end{aligned}
\tag{6}
$$

In Section 2 we show that (5) solves the ordinary differential equation (1).

All four constants a, ω, ϕ, and r are functions of (k,c), so we could instead write: $a(k,c)$, $r(k,c)$, etc.. But, to ease notation and avoid too many parentheses, we will instead note the dependence on k via subscripts. Since we are interested in the sensitivity of y only with repsect to k we will drop the dependence on c in notation.

1.1 Phase and Amplitude

We solve the initial value equations (6) for ϕ:

$$
\frac{v_0 + y_0 r_k\omega_k}{y_0\sqrt{1-r_k^2}\omega_k} = -\tan(\phi_k)\,, \tag{7}
$$

$$
\arctan\left(-\frac{2v_0 + y_0 c}{2y_0\sqrt{1-r_k^2}\omega_k}\right) = \phi_k\,. \tag{8}
$$

Here, we note that this solution for ϕ is also a function of k and append this note as a subscript. This result makes a immediately available as

$$
a_k = \frac{y_0}{\cos(\phi_k)}\,, \tag{9}
$$

where we make the dependence on k noted once again. We disregard the dependence on

1.2 Sensitivity

Because we are calculating the derivative with respect to k it is useful to rewrite the solution and make it explicit when a term does depend on k, and show where k cancels:

$$
\begin{aligned}
\Gamma_k &:= \sqrt{1 - r_k^2}\,, \\
\Gamma_k \omega_k &= \sqrt{1 - \left(\frac{c}{2\sqrt{k}}\right)^2}\sqrt{k} = \sqrt{k - \frac{c^2 k}{4k}} = \sqrt{k - \frac{c^2}{4}}\,, \\
\Rightarrow y(t;k) &= a \exp\left(-\frac{c}{2}t\right) \cos\left(t\sqrt{k - \frac{c^2}{4}} + \phi_k\right)\,,
\end{aligned} \tag{10}
$$

The sensitivity of the solution $y(t;k)$ with respect to the parameter k can be obtained by straight forward differentiation, albeit with many terms. We collect a list of derivatives:

$$
a_k = \frac{y_0}{\cos(\phi_k)} \qquad \frac{da_k}{dk} = \frac{y_0 \tan(\phi_k)}{\cos(\phi_k)} \frac{d\phi_k}{dk} = a_k \tan(\phi_k) \frac{d\phi_k}{dk}\,, \tag{11}
$$

$$
\omega_k = \sqrt{k} \qquad \frac{d\omega_k}{dk} = \frac{1}{2\sqrt{k}}\,, \tag{12}
$$

$$
r_k = \frac{c}{2\omega_k} \qquad \frac{dr_k}{dk} = -\frac{c}{2\omega_k^2} \frac{d\omega_k}{dk} = -\frac{c}{2k}\frac{1}{2\sqrt{k}} = -\frac{r_k}{2k}\,, \tag{13}
$$

$$
r_k \omega_k = \frac{c}{2} \qquad \frac{d(r_k \omega_k)}{dk} = 0\,, \tag{14}
$$

$$
\Gamma_k = \sqrt{1 - r_k^2} \qquad \frac{d\Gamma_k}{dk} = \frac{2r_k}{2\sqrt{1 - r_k^2}} \frac{dr_k}{dk} = -\frac{r_k}{k}\frac{r_k}{2\Gamma_k}\,, \tag{15}
$$

$$
\Gamma_k \omega_k = \sqrt{k - \frac{c^2}{4}} \qquad \frac{d(\Gamma_k \omega_k)}{dk} = \frac{1}{2\sqrt{k - \frac{c^2}{4}}} = \frac{1}{2\Gamma_k \omega_k}\,, \tag{16}
$$

First we take the derivative of ϕ from (7):

$$
-\frac{d}{dk}\left(\frac{2v_0 + y_0 c}{2y_0 \Gamma_k \omega_k}\right) = \frac{2}{cos(2\phi + 1)} \frac{d\phi}{dk}\,, \tag{17}
$$

$$
-\left(-\frac{2v_0 + y_0 c}{4y_0 (\Gamma_k \omega_k)^3}\right) = \frac{2}{cos(2\phi + 1)} \frac{d\phi_k}{dk}\,, \tag{18}
$$

We solve for the derivative of ϕ:

$$
\frac{d\phi_k}{dk} = \cos(2\phi_k + 1) \frac{2v_0 + y_0 c}{8y_0 (\Gamma_k \omega_k)^3}\,, \tag{19}
$$

```
w <- function(k) sqrt(k)
r <- function(k) c/(2*w(k))                                # damping ratio
srr1 <- function(k) sqrt(1-r(k)^2)                         # convenience
## srr1*w
srr1w <- function(k) sqrt(k-0.25*c^2)
## phase and amplitude
f <- function(k) atan(-(2*v0+y0*c)/(2*y0*srr1w(k)))
a <- function(k) y0/cos(f(k))
## derivatives
dwdk <- function(k) 1/(2*w(k))
dfdk <- function(k) cos(2*f(k) + 1)*(2*v0 + y0*c)/(8*y0*srr1w(k)^3)
dadk <- function(k) y0 * (tan(f(k))/cos(f(k))) * dfdk(k)
## y and S = dy/dk
y <- function(x,k) a(k)*exp(-0.5*c*x)*cos(srr1w(k)*x + f(k))
S <- function(x,k) {
  y(x,k)*(  tan(f(k))*dfdk(k)
          - tan(srr1w(k)*x+f(k))*(0.5*x/srr1w(k) + dfdk(k)))
}
```

Listing 1: Calculation of the sensitivity in R. The variables have slightly different names: `srr1(k)` is Γ_k and `srr1w(k)` is $\Gamma_k\omega_k$. The independent variable t is `x` in the code.

Using these results, we assemble the full sensitivity:

$$\begin{aligned}
\frac{dy(t;k)}{dk} &= \frac{d}{dk}\left(a_k \exp\left(-\frac{c}{2}t\right)\cos(\Gamma_k\omega_k t + \phi_k)\right) , \\
&= \exp\left(-\frac{c}{2}t\right)\frac{d}{dk}\left(a_k\cos(\Gamma_k\omega_k t + \phi_k)\right) , \\
&= \exp\left(-\frac{c}{2}t\right)\left(\frac{da_k}{dk}\cos(\Gamma_k\omega_k t + \phi_k) - a_k\sin(\Gamma_k\omega_k t + \phi_k)\left(\frac{t}{2\Gamma_k\omega_k} + \frac{d\phi_k}{dk}\right)\right) , \\
&= \exp\left(-\frac{c}{2}t\right)\left(a_k\tan(\phi_k)\frac{d\phi_k}{dk}\cos(\Gamma_k\omega_k t + \phi_k) - a_k\sin(\Gamma_k\omega_k t + \phi_k)\left(\frac{t}{2\Gamma_k\omega_k} + \frac{d\phi_k}{dk}\right)\right) , \\
&= y(t;k)\left(\tan(\phi_k)\frac{d\phi_k}{dk} - \tan(\Gamma_k\omega_k t + \phi_k)\left(\frac{t}{2\Gamma_k\omega_k} + \frac{d\phi_k}{dk}\right)\right) ,
\end{aligned} \tag{20}$$

The last line is suitable for direct evaluation, given the parameters (k, c). In R we calculate this using the function defined in Listing 1.

2 Proof for the Solution

We insert the proposed solution (5) into (1). First we perform al needed derivatives:

$$\begin{aligned}
y(t;k) &= a\exp(-r\omega t)\cos(\sqrt{1-r^2}\omega t + \phi) , \\
\dot{y}(t;k) &= -y(t;k)r\omega - a\exp(-r\omega t)\sin(\sqrt{1-r^2}\omega t + \phi)\sqrt{1-r^2}\omega ,
\end{aligned} \tag{21}$$

The second derivative of y:

$$\begin{aligned}\ddot{y}(t;k) = -\dot{y}(t;k)r\omega \\ - \Big(a\exp(-r\omega t)(-r\omega)\sin(\sqrt{1-r^2}\omega t+\phi)\sqrt{1-r^2}\omega \\ + a\exp(-r\omega t)\cos(\sqrt{1-r^2}\omega t+\phi)\left(\sqrt{1-r^2}\omega\right)^2\Big)\,, \quad (22)\end{aligned}$$

which simplifies to:

$$\begin{aligned}\ddot{y}(t;k) &= -\dot{y}(t;k)r\omega - \left(\underbrace{-a\exp(-r\omega t)\sin(\sqrt{1-r^2}\omega t+\phi)\sqrt{1-r^2}\omega}_{\dot{y}+yr\omega} r\omega + y(t;k)\left(\sqrt{1-r^2}\omega\right)^2\right)\,,\\ &= -\dot{y}(t;k)r\omega - \left(\dot{y}(t;k)r\omega + y(t;k)(r\omega)^2 + y(t;k)\left(\sqrt{1-r^2}\omega\right)^2\right)\,,\\ &= -2\dot{y}(t;k)r\omega - \left(y(t;k)(r\omega)^2 + y(t;k)(1-r^2)\omega^2\right)\,,\\ &= -2\dot{y}(t;k)r\omega - \left(y(t;k)\omega^2\right)\,,\\ &= -2\dot{y}(t;k)\frac{c}{2} - y(t;k)k = -ky(t;k) - c\dot{y}(t;k)\,,\end{aligned} \quad (23)$$

which reconstructs the original ODE in (1).

3 Numerical Simulation Outcomes

In the following sections, we show comparisons of numerical solutions to the initial value problem (1) under various conditions: with or without damping (friction), with and without a constant driving force. In all cases, we do the numerical integration without forward sensitivity analysis, but use the analytical solution in (20).

By choice, the sensitivity is a scalar function, so we show its values directly. We have also included sensitivity calculations using finite differences and using the Cauchy integral formula on the analytical solution (5). The analytical solution for the state variable y is well known and simple enough to be implemented correctly. So, we use these discrete methods to check the analytical solution for the sensitivity.

In general, analytical solutions are not guaranteed to be available. An alternative is to use the sensitivity to predict a trajectory at slightly changed parameters $k+\Delta_k$, using a linear predictor:

$$y(t;k+\Delta_k) = y(t;k) + S_y(t;k)\cdot\Delta_k + \mathcal{O}(\Delta_k^2)\,, \quad (24)$$

For small enough Δ_k, we neglect the second order effects. But, since we are using an approximation S of the sensitivity, we make an order one error for each time point j:

$$y(t_j;k+\Delta_k) \approx y(t_j;k) + (\mathrm{S}_y(t_j;k) + \Delta_S(t_j;k))\cdot\Delta_k\,, \quad (25)$$

where $\Delta_S(t_j;k)$ is the error we made when approximating the sensitivity:

$$\Rightarrow \|\Delta_S(t_j;k)\cdot\Delta_k\| \approx \left\|y(t_j;k+\Delta_k) - (y(t_j;k) + \mathrm{S}_y(t_j;k)\cdot\Delta_k)\right\|\,. \quad (26)$$

This trajectory can of course be simulated directly using the changed parameters whenever analytical solutions are unavailable. This makes the above error estimate generally applicable.

The average sensitivity error can be estimated using the discrepancy between the prediction and direct simulation/solution:

$$\delta_S := \frac{1}{T}\sum_{j=1}^{T} \frac{\left\| y(t_j;k+\Delta_k) - (y(t_j;k) + S_y(t_j;k)\cdot\Delta_k)\right\|}{\|\Delta_k\|}\,, \tag{27}$$

where T is the number of discrete time points returned by the numerical solver.

3.1 Oscillations

Under the conditions $c = 0$, the system sustains oscilations of constant height (given non-zero initial values). We integrate the model using the `BDF` solver in the GSL library `gsl_odeiv2`.

The first step is to verify that the numerical solution is sufficiently accurate. Both the analytical and numerical trajectories are shown in Figure 1. They agree within the requested accuracy.

Figure 5 shows the difference between a prediction using the estimated sensitivity, and analytical solution for a small shift in k, as motivated in the previous Section. But, rather than a second simulation we use the analytical solution for y.

But, because the analytical solution for the sensitivity *is* available, we can also show it directly, see Figure 6. This is usually not possible. Additionally, in many cases, obtaining a sensitivity check through finite differences requires quite a bit of work for larger systems, maybe more than conventional forward sensitivity analysis. That is why the linearization error is fairly useful.

3.2 Damping

In the case of a damped oscillation, we continue to use the analytical solution and calculate the same error measures as in the previous section. Figure 4 shows the solution accuracy (disregarding sensitivities). Figure 5 shows the linearization error, as described in Section 3. Since we have the analytical solution for the sensitivity, we can depict it directly, the graphs are shown in Figure 6.

3.3 Driving Force

We can also add a constant force to the model:

$$\ddot{y} = -ky - c\dot{y} + F \tag{28}$$

Without following through all analytical calculations, we can restrict ourselves to numerical methods only. This is what we would do for large models. Since we are not using the analytical sensitivity solution, we calculate and depict the linearization error (27) only. To obtain a direct solution for the shifted trajectory, we use the solver from the `deSolve` package (in R).

Figure 7 shows the effects of a small force, while Figure 8 depicts the effects of a larger force.

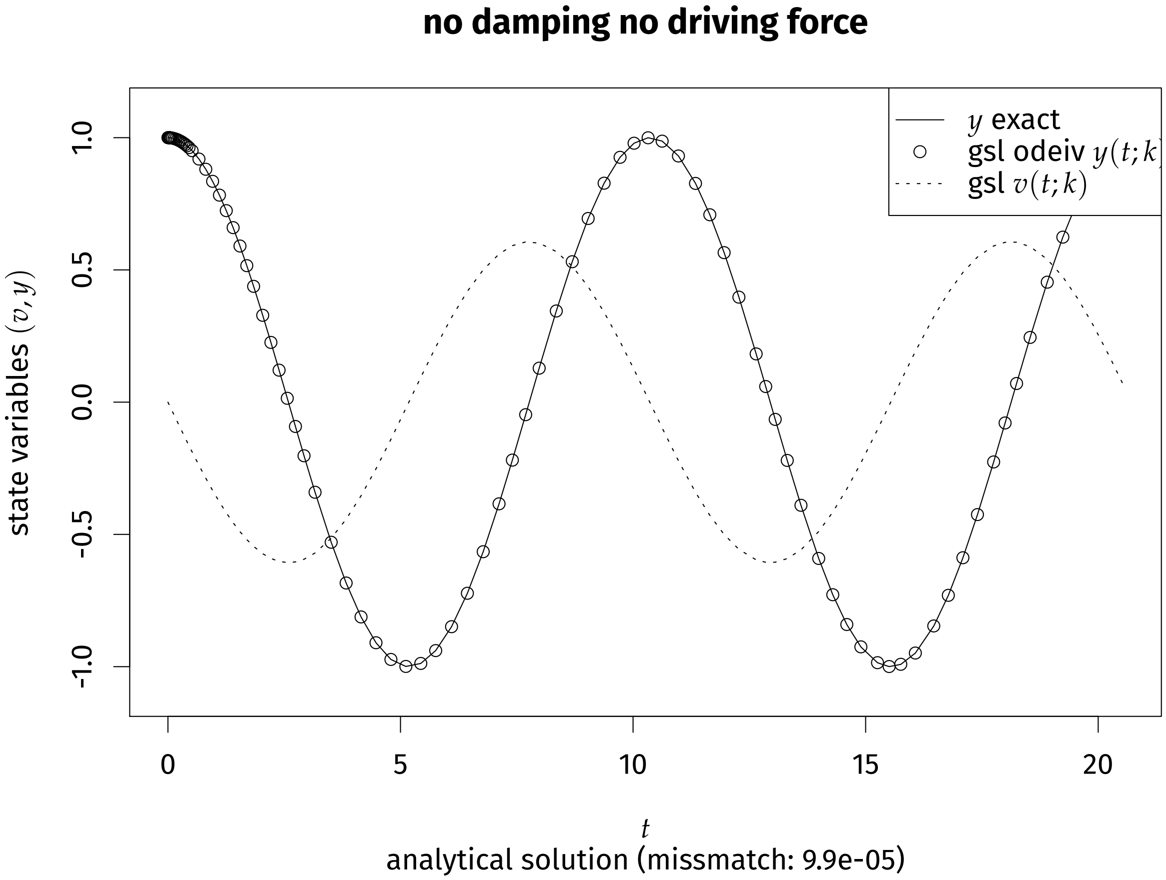


Figure 1: Comparison between *numerical integration* and the *analytical trajectory* solution. A perfect match within visual precision.

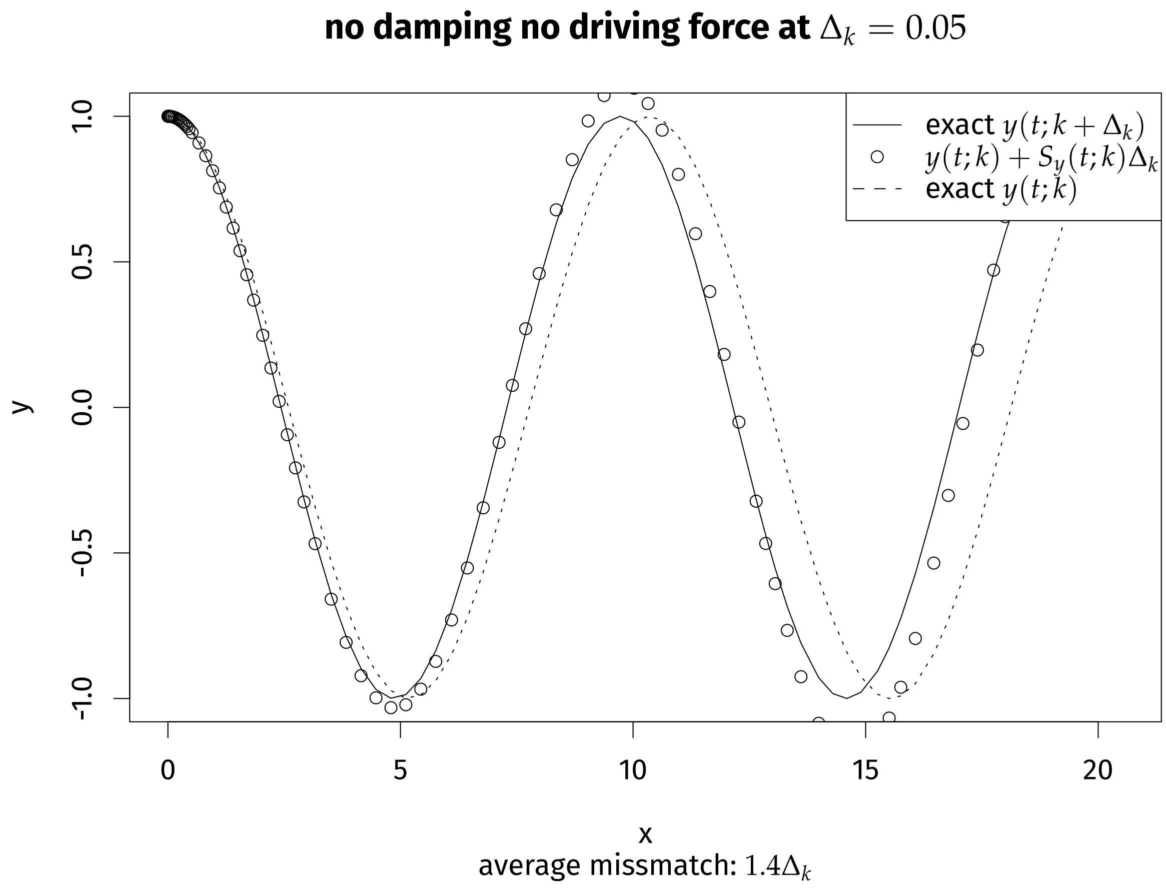


Figure 2: Linearization Error. The trajectory is shifted linearly to slightly changed parameters using the approximate sensitivity, plotted against a direct solution at $k + \Delta_k$.

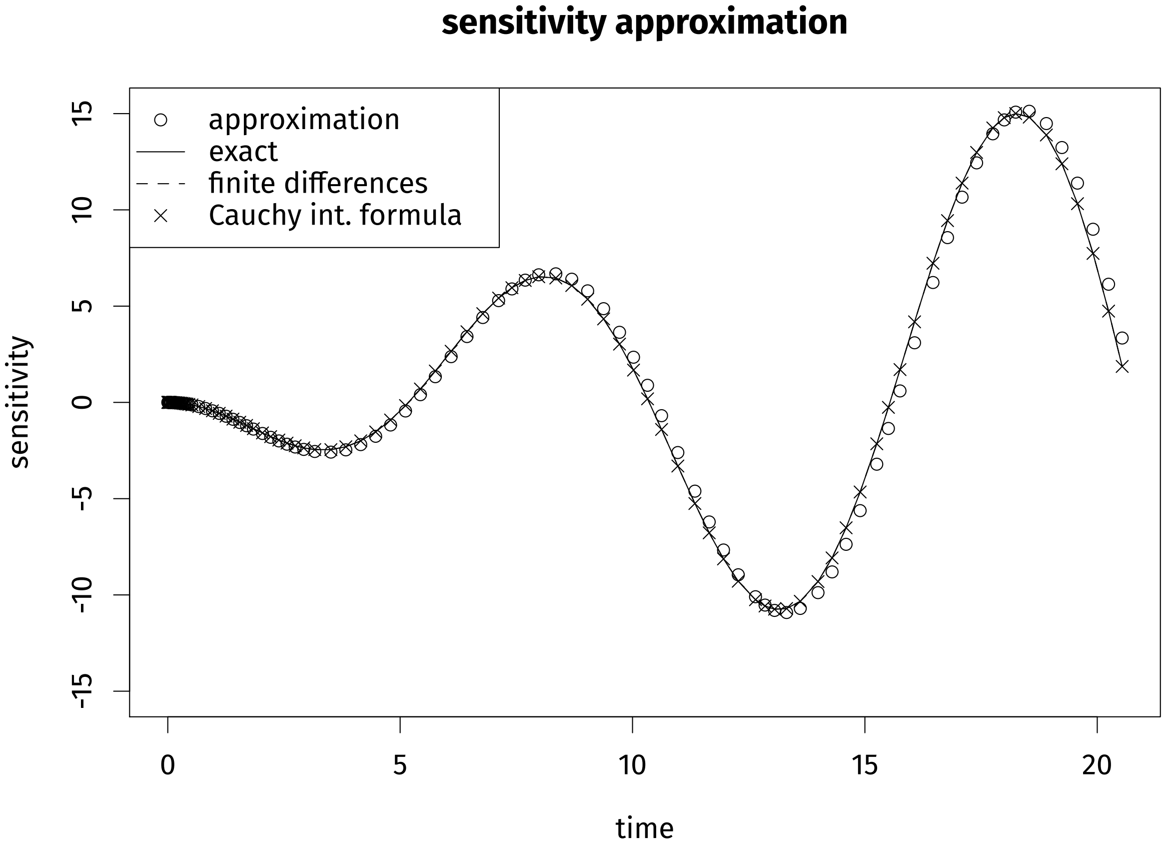


Figure 3: Comparison between the (scalar) approximate sensitivity and the analytical sensitivity solution $dy(t;k)/dk$

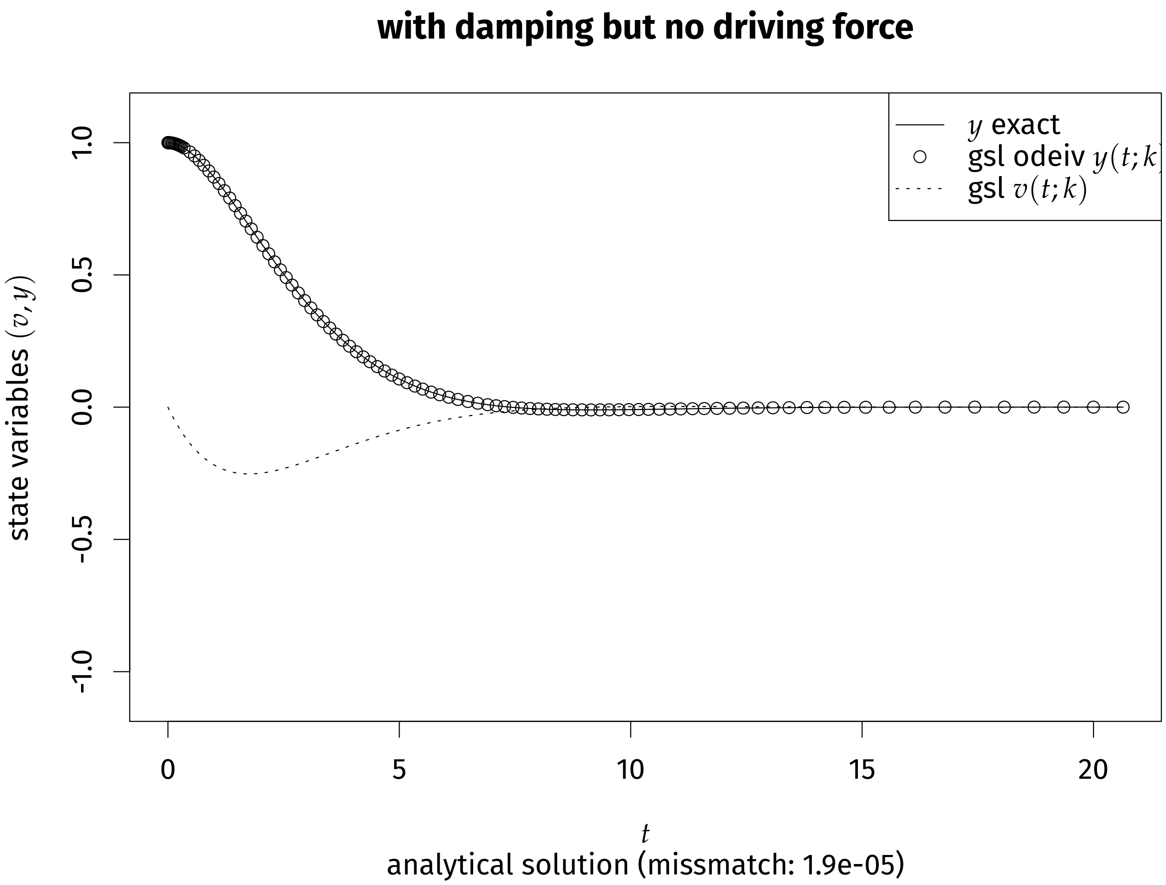


Figure 4: Comparison of the solution obatined using the solvers in the GSL to the analytical solution.

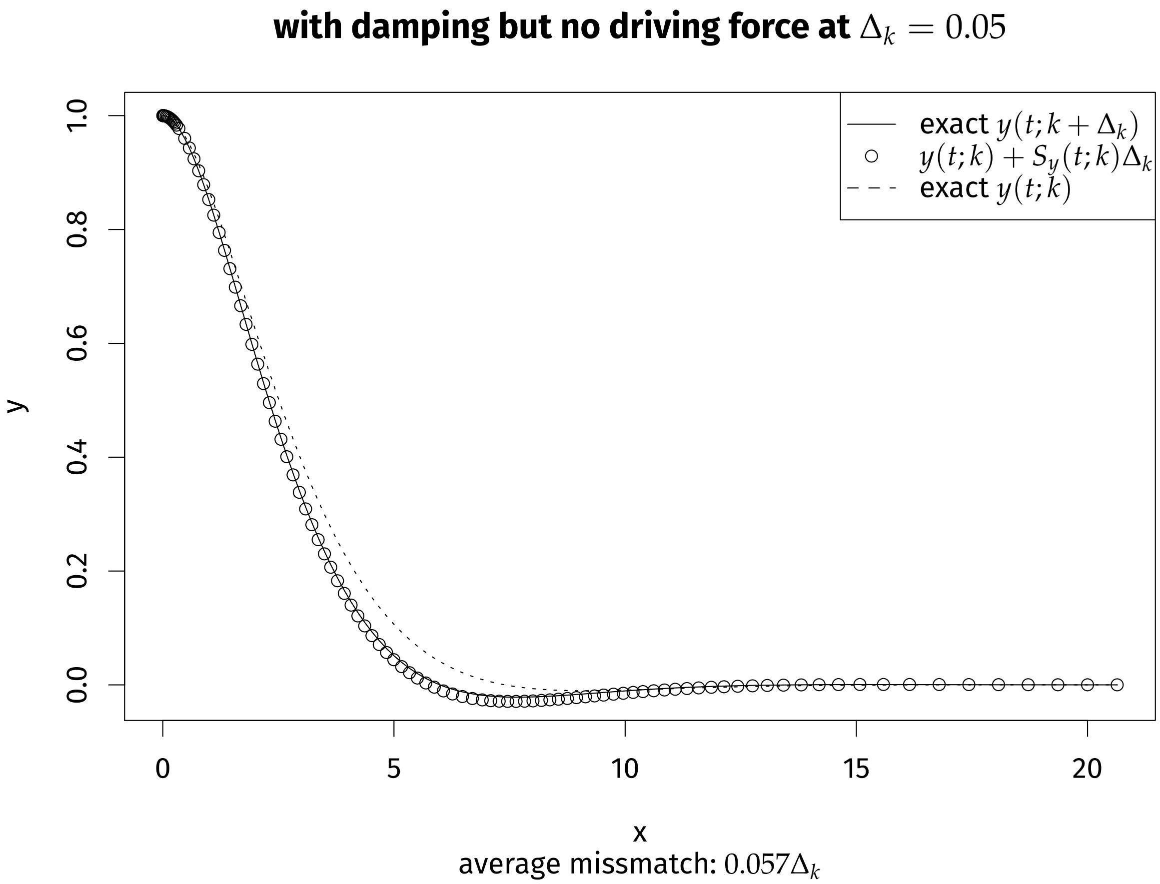


Figure 5: Linearization Error as described in Section 3. The error is very small.

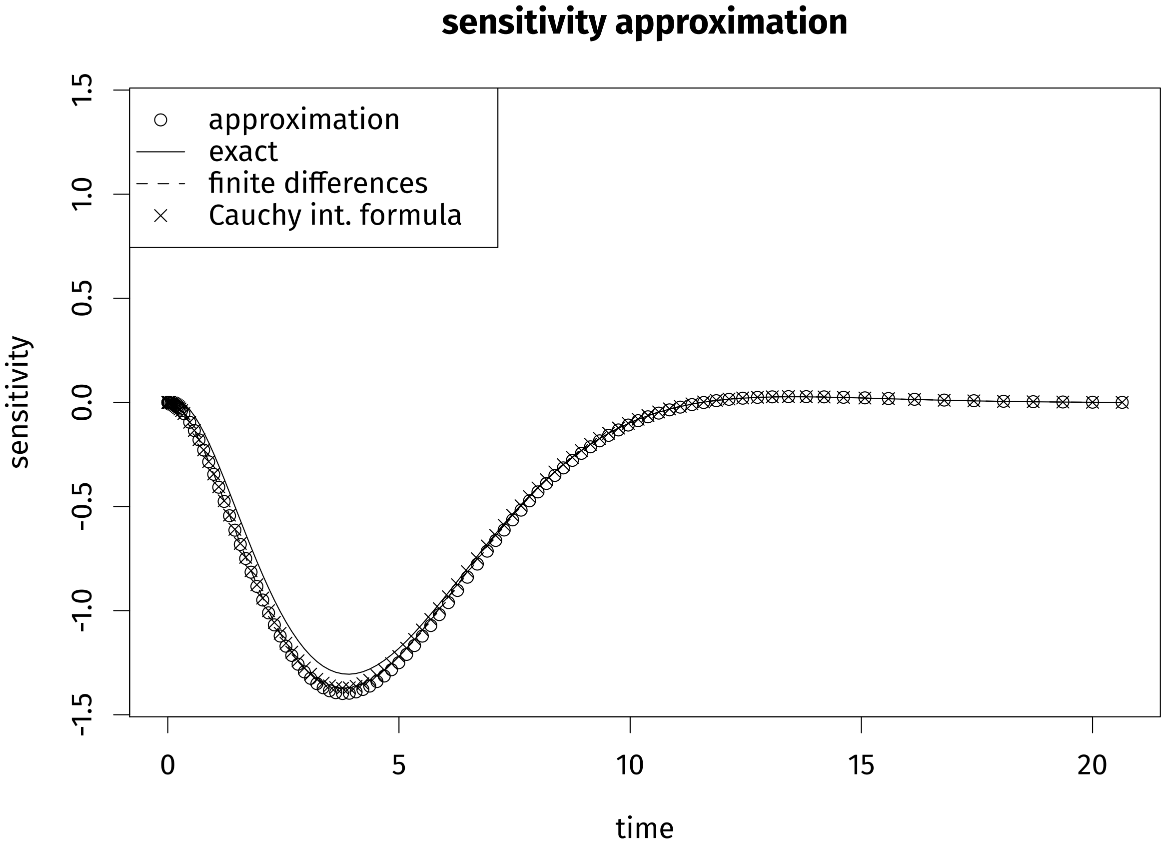
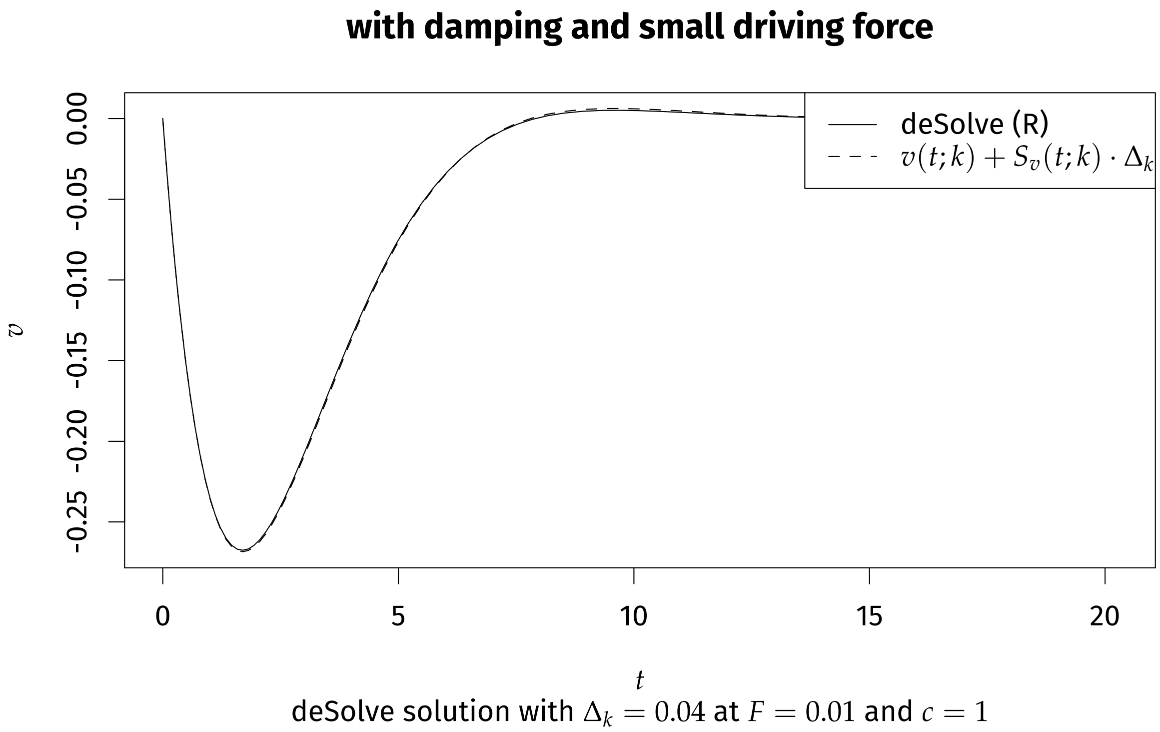


Figure 6: Comparison between the numerically obtained sensitivity approximation and the analytical sensitivity solution, for the case of a damped harmonic oscillator.



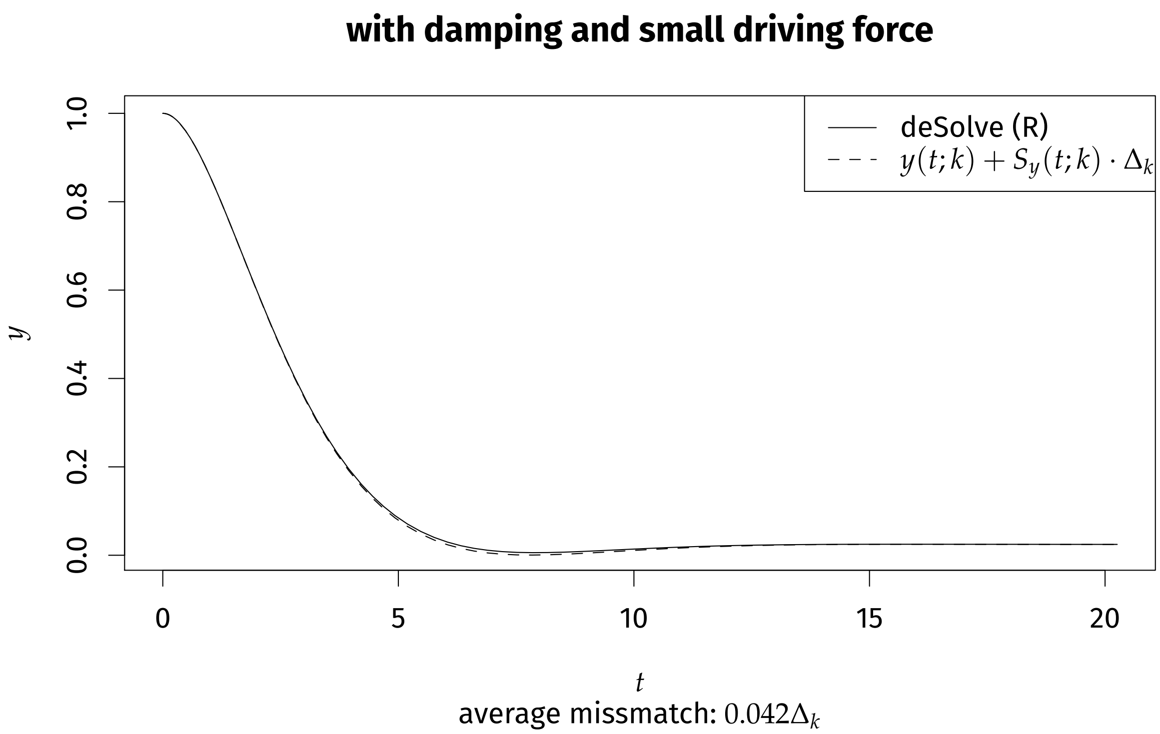


Figure 7: Linearization error compared to a direct solution with R's `deSolve`. The error is very small.

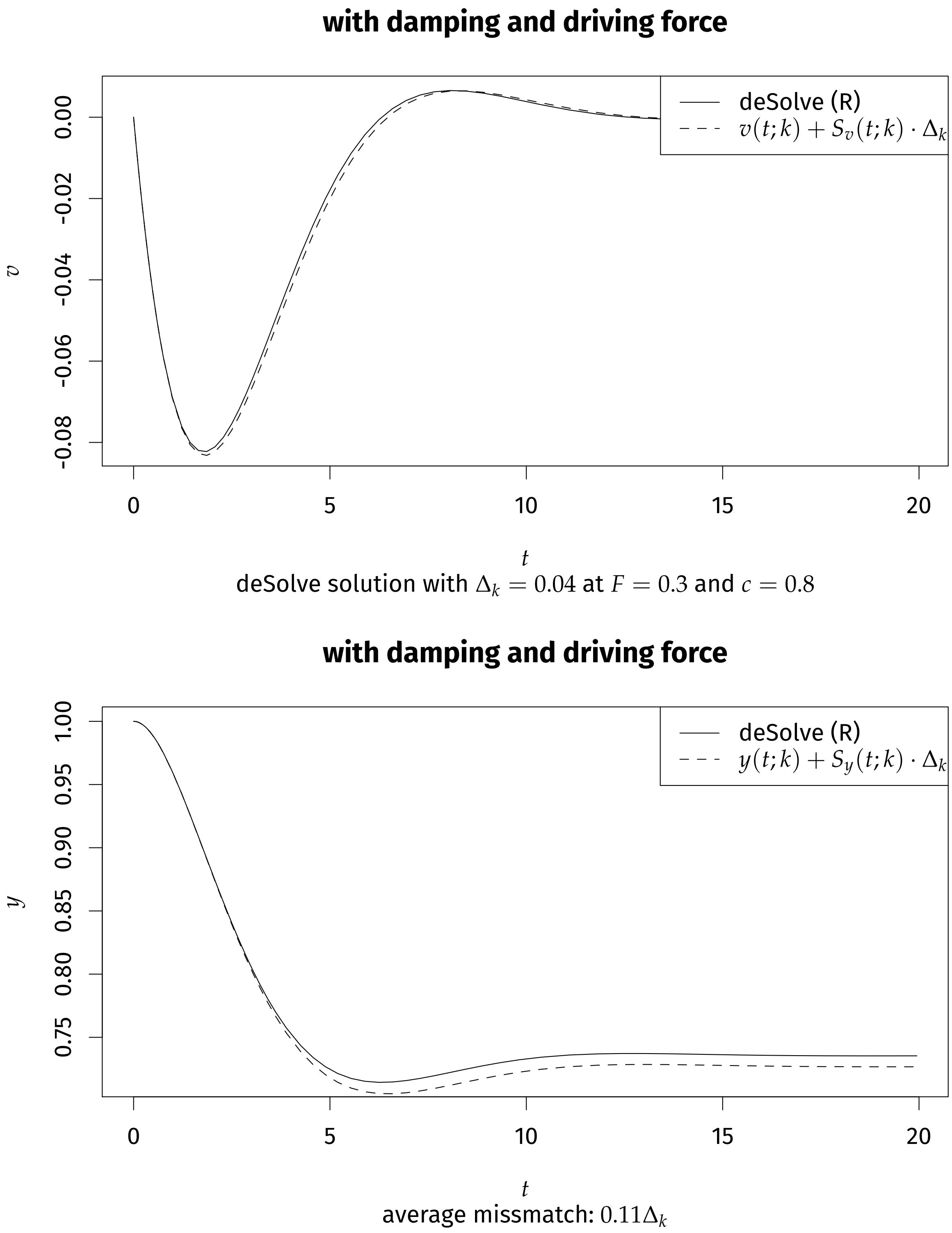


Figure 8: The linearization error for a bigger constant force, but otherwise calculated precisely as in Figure 7. The error is still small but clearly visible.