Sensitivity Approximation and Verification

Solution to Affine System

Andrei Kramer <andreikr@kth.se>

April 29, 2021

Here we make use of a system with readily available analytical solutions: linear and affine systems. For this series of tests, the number of state variables can be arbitrarily large. This makes timing tests easy to do, without compromising the validity of the analytical solution. We generate a system with a symmetrical, negative definite Jacobian of arbitrary size using a *pseudo-random number generator*.

1 Model Definition

The general model is an ordinary differential equation:

$$\dot{x} = f(x,t;p) \qquad x \in \mathbb{R}^n, p \in \mathbb{R}^n\,, \tag{1}$$

$$x(0) = x_0\,, \qquad S(x,t;p)_i^{\,j} = \frac{dx_i(t;p)}{dp_j}\,, \tag{2}$$

where p is a parameter vector and S the sensitivity matrix.

1.1 Affine System

Here, we have a constant Jacobian Γ and further restrict:

$$\dot{x} = \Gamma x + b \qquad b_i = p_i^2 \tag{3}$$

$$\tag{4}$$

This can be transformed to a linear system:

$$\begin{aligned} L &= \Gamma^{-1}\,,\\ z &= x + Lb\,,\\ \dot{z} &= \Gamma z\,, \end{aligned} \tag{5}$$

where we define L to ease notation. The solution to (5) is:

$$z(t) = \exp\left(\Gamma t\right) z_0\,, \tag{6}$$

and consequently:

$$x(t;p) = \exp(\Gamma t)\,(x_0 + Lb) - Lb\,. \tag{7}$$

The analytical sensitivity of (7) is:

$$E_\Gamma(t) = \exp(\Gamma t)\ , \tag{8}$$

$$x(t;p) = E_\Gamma(t)(x_0 + Lb) - Lb + u \qquad b_i := p_i^2\,, \tag{9}$$

$$(Lb)_i = \sum_j L_{ij} b_j\,, \qquad \frac{d(Lb)_i}{dp_k} = 2\sum_j L_{ij} p_j [j = k]\,, \tag{10}$$

$$\Rightarrow \nabla_p (Lb) = 2L\,\mathrm{diag}(p)\,, \qquad \nabla_p x(t;p) = 2(E_\Gamma(t) - I_n)L\,\mathrm{diag}(b)\,, \tag{11}$$

where

$$[\text{condition}] = \begin{cases} 1 & \text{condition is true} \\ 0 & \text{otherwise} \end{cases}\,, \tag{12}$$

and $(I_n)_{ij} = [i = j]$.

Without changing the calculations for the sensitivity we can add an input u to the model for the possibility of different simulation runs.

2 Numerical Solution

We use the results from Section 1 to verify the accuracy of the numerical solution of the initial value problem (3). We use the *GNU Scientific Library* (`gsl`) to obtain a numerical solution x and approximate the sensitivity. The results of the returned trajectory at discrete time points t_j is displayed in Figure 1, the solution is within specified error tolerances. The accuracy of the estimated sensitivity matrix is displayed in Figure 2. Since we are interested in parameter estimation, we also investigate the Fisher information of a likelihood function that assumes a Gaussian error when measuring data (the data corresponds to the state variables in the model). Since this is a test for the sensitivity approximation, we don't actually need data, so we set the measurement noise to an additive unit Gaussian model: $\mathcal{N}(0, I_n)$ (iid).

Then the Fisher information $F(p)$ is related to the sensitivity as:

$$F(p) = \sum_j \underbrace{S(x(t_j), t_j; p)^{\mathrm{T}} S(x(t_j), t_j; p)}_{\mathtt{FI}(t_j;p)}\,, \tag{13}$$

where we also define the fisher information per time point: $\mathtt{FI}(t;p)$.

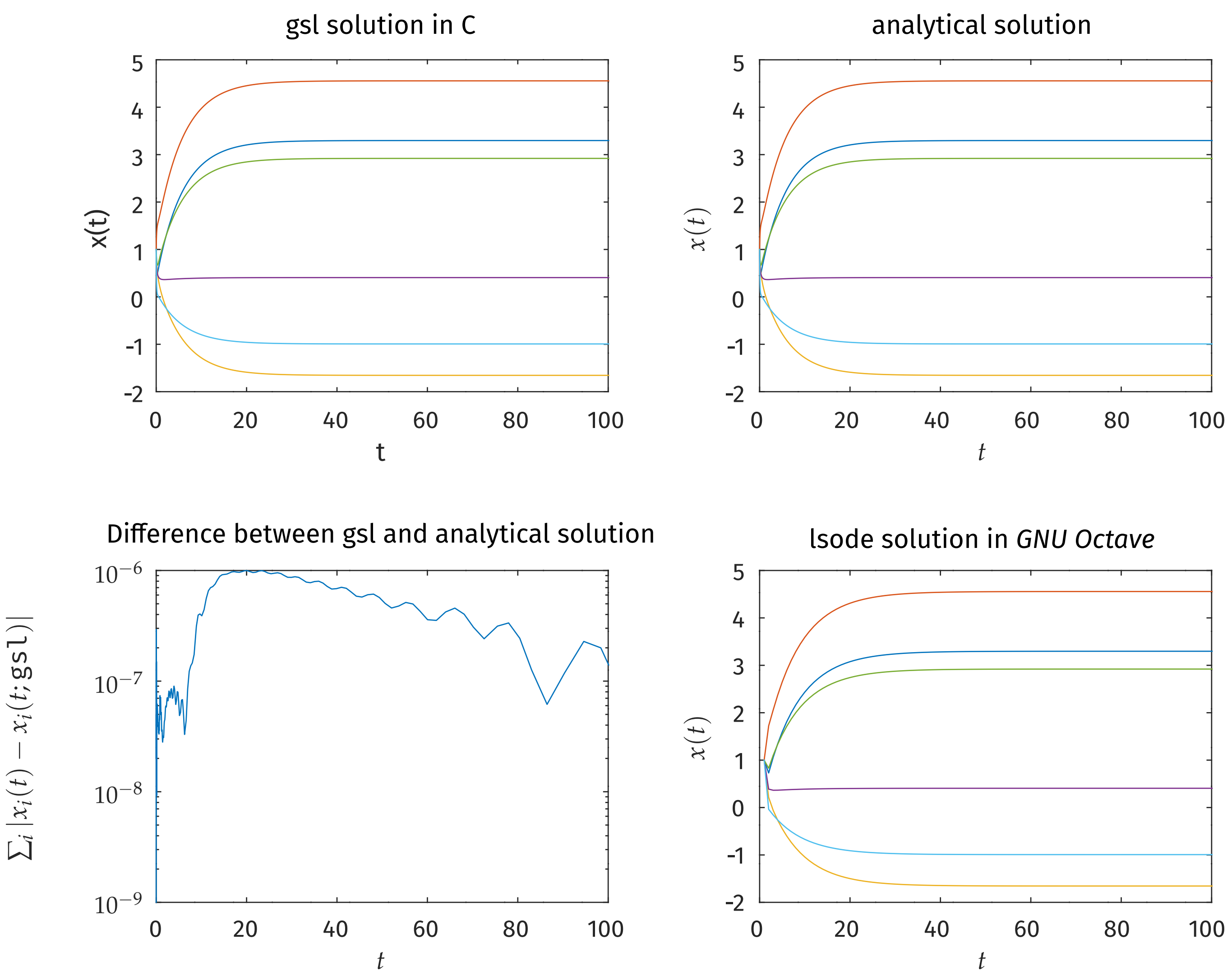

Figure 1: Accuracy of the trajectory $x(t_j)$ using the gsl solvers in `gsl_odeiv2.h` as compared to the analytical solution (7). For completeness, we also include the numerical solution from *GNU Octave*'s `lsode` solver for stiff problems. Bottom left: the absolute difference between the trajectories, summed over all state variables.

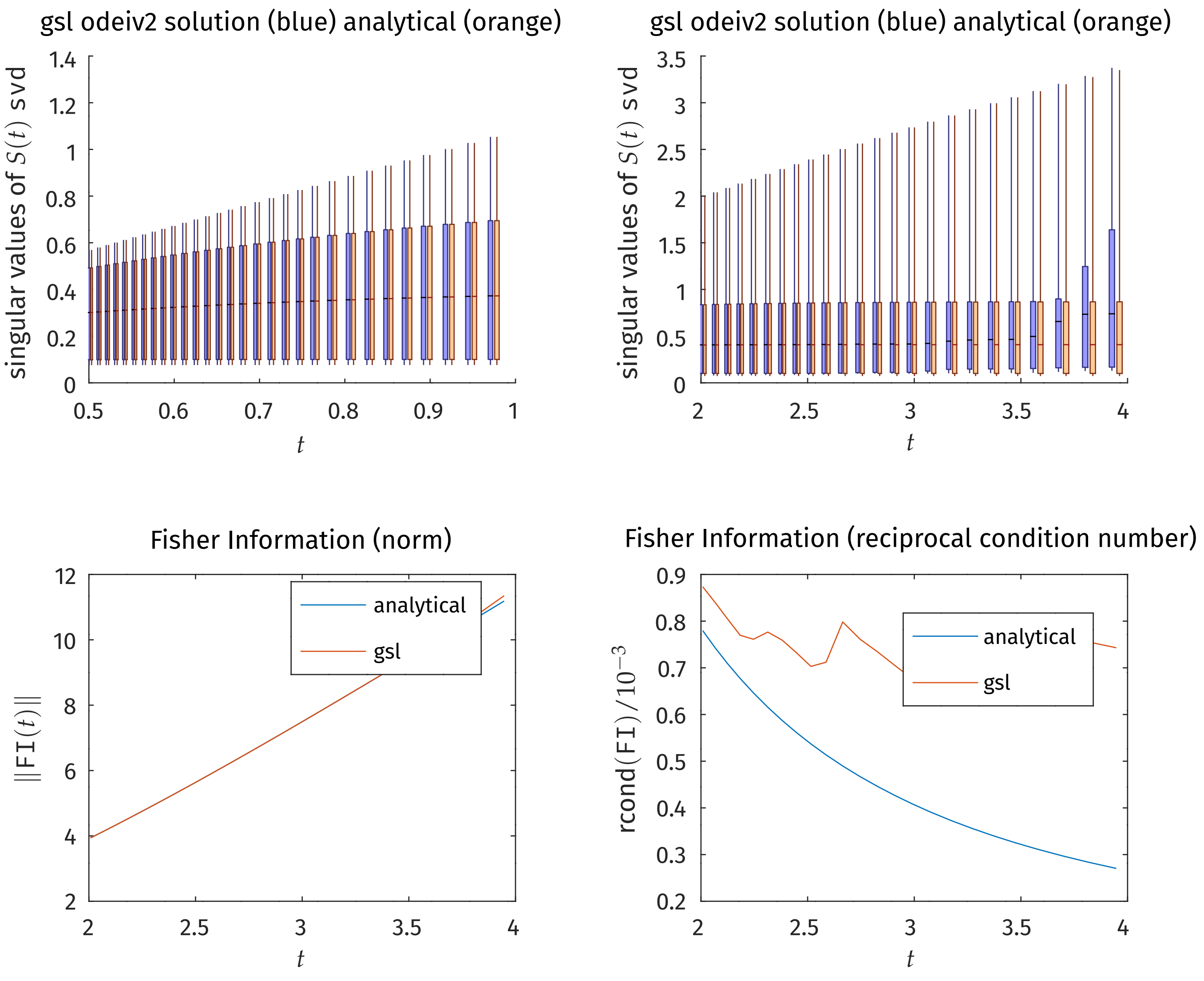

Figure 2: The accuracy of the sensitivity matrix is summarized as a box-and-whisker plot of the singular values of $S(x,t;p)$ (singular value decomposition, per time point), this is to reduce visual clutter, not to imply that the values are random. In (blue) the approximated sensitivity, in (orange) the analytical solution. The numerical approximation eventually diverges, when the system approaches steady state. Lower panels show the accuracy of the Fisher information: $\mathtt{FI} = S^{\mathsf{T}}S$ (this is assuming a Gaussian error model with unit noise parameter and corresponding Likelihood function).