7 Supplementary Material

7.1 Derivatives of a Gaussian process

7.1.1 Definitions

Following Papoulis and Pillai (2002), we define stochastic convergence and stochastic differentiability.

Definition 2 *The RV x_n* ***converges*** *to x in the MS sense (limit in mean) if for some x*

$$\lim_{n\to\infty} \mathbb{E}(|x_n - x|) = 0 \tag{20}$$

Definition 3 *The stochastic process x(t) is MS* ***differentiable*** *if for some $x'(t)$*

$$\lim_{\delta t\to 0} \mathbb{E}\left|\frac{x(t+\delta t)-x(t)}{\delta t} - x'(t)\right| = 0 \tag{21}$$

Definition 4 *A stochastic process x(t) is called a* ***Gaussian Process****, if any finite number of samples of its trajectory are jointly Gaussian distributed according to a previously defined mean function $\mu(t)$ and a covariance matrix, that can be constructed using a predefined kernel function $k_\phi(t_i, t_j)$*

7.1.2 GP and its derivative are jointly Gaussian

Let $t_0, \delta t \in \mathbb{R}$.
Let $x(t)$ be a Gaussian Process with constant mean μ and kernel function $k_\phi(t_1, t_2)$, assumed to be MS differentiable.

From the definition of GP, we know that $x(t_0 + \delta t)$ and $x(t_0)$ are jointly Gaussian distributed.

$$\begin{bmatrix} x(t_0) \\ x(t_0+\delta t) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mu\\ \mu\end{bmatrix}, \boldsymbol{\Sigma}\right) \tag{22}$$

where $\boldsymbol{\Sigma}_{i,j} = k_\phi(\mathbf{t}_i, \mathbf{t}_j)$ using $\mathbf{t} = [t_0, t_0 + \delta t]$.

Using the linear transformation

$$\mathbf{T} = \frac{1}{\delta t}\begin{bmatrix}\delta t & 0\\ -1 & 1\end{bmatrix} \tag{23}$$

one can show that

$$\begin{bmatrix} x(t_0) \\ \frac{x(t_0+\delta t)-x(t_0)}{\delta t} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mu\\ 0\end{bmatrix}, \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}^T\right) \tag{24}$$

So it is clear that for all δt, $x(t_0)$ and $\frac{x(t_0+\delta t)-x(t_0)}{\delta t}$ are jointly Gaussian distributed. Using the assumption that x is differentiable according to the definition in eq. 21 and the fact that convergence in expectation implies convergence in distribution, it is clear that $x(t_0)$ and $\dot{x}(t_0)$ are jointly Gaussian.

This fact can be easily extended to any finite set of sample times $\mathbf{t} = [t_0, t_1, ..., t_N]$. One can use the exact same procedure to show that the resulting vectors $\mathbf{x}(\mathbf{t})$ and $\dot{\mathbf{x}}(\mathbf{t})$ are jointly Gaussian as well.

7.1.3 Moments of the joint distribution

As shown in the previous section, any finite set of samples $\mathbf{x}$ is jointly Gaussian together with its derivatives $\dot{\mathbf{x}}$. To calculate the full distribution, it thus suffices to calculate the mean and the covariance between the elements of the full vector

$$\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{C}_\phi & \mathbf{C}'_\phi\\ {}'\mathbf{C}_\phi & \mathbf{C}''_\phi\end{bmatrix}\right) \tag{25}$$

$\mathbf{C}_\phi$ is the predefined kernel matrix of the Gaussian Process.

${}'\mathbf{C}_\phi$ can be calculated by directly using the linearity of the covariance operator.

$$\begin{aligned} {}'\mathbf{C}_{\phi_{i,j}} &= \operatorname{cov}(\dot{x}(t_i), x(t_j)) \\ &= \operatorname{cov}\left(\frac{d}{da}x(a)\Big|_{a=t_i}, x(t_j)\right) \\ &= \frac{d}{da}\operatorname{cov}\left(x(a), x(t_j)\right)_{a=t_i} \\ &= \frac{d}{da}k_\phi(a, t_j)_{a=t_i} \end{aligned} \tag{26}$$

Obviously, $\mathbf{C}'_\phi$ is just the transposed of ${}'\mathbf{C}_\phi$, while $\mathbf{C}''_\phi$ can be calculated in exactly the same manner to obtain

$$\mathbf{C}''_{\phi_{i,j}} = \frac{d}{da}\frac{d}{db}k_\phi(a,b)_{a=t_i, b=t_j} \tag{27}$$

7.1.4 Conditional GP for derivatives

To obtain the GP over the derivatives given the states, the joint distribution

$$\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{C}_\phi & \mathbf{C}'_\phi\\ {}'\mathbf{C}_\phi & \mathbf{C}''_\phi\end{bmatrix}\right) \tag{28}$$

has to be transformed. This can be done using standard techniques as described e.g. in section 8.1.3 of Petersen et al. (2008). There, it is written:

Define

$$\mathbf{x} = \begin{bmatrix}\mathbf{x_a}\\ \mathbf{x_b}\end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix}\boldsymbol{\mu_a}\\ \boldsymbol{\mu_b}\end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma_a} & \boldsymbol{\Sigma_c}\\ \boldsymbol{\Sigma_c}^T & \boldsymbol{\Sigma_b}\end{bmatrix} \tag{29}$$

Then

$$p(\mathbf{x_b}|\mathbf{x_a}) \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_b, \hat{\boldsymbol{\Sigma}}_b) \tag{30}$$

where

$$\hat{\boldsymbol{\mu}}_b = \boldsymbol{\mu_b} + {\boldsymbol{\Sigma_c}}^T {\boldsymbol{\Sigma_a}}^{-1}(\mathbf{x_a} - \boldsymbol{\mu_a}) \tag{31}$$

$$\hat{\boldsymbol{\Sigma}}_b = \boldsymbol{\Sigma_b} - {\boldsymbol{\Sigma_c}}^T {\boldsymbol{\Sigma_a}}^{-1}\boldsymbol{\Sigma_c} \tag{32}$$

Applied to the above probability distribution, this leads to

$$p(\dot{\mathbf{x}}|\mathbf{x}) \sim \mathcal{N}(\mathbf{Dx}, \mathbf{A}) \tag{33}$$

using

$$\mathbf{D} = {}'\mathbf{C}_\phi {\mathbf{C}_\phi}^{-1} \tag{34}$$

$$\mathbf{A} = \mathbf{C}''_\phi - {}'\mathbf{C}_\phi {\mathbf{C}_\phi}^{-1}\mathbf{C}'_\phi \tag{35}$$

7.2 Proof of theorem 1

Proof The proof of this statement follows directly by combining all the previous definitions and marginalizing out all the random variables that are not part of the end result.

First, one starts with the joint density over all variables as stated in equation (15)

$$p(\mathbf{x},\dot{\mathbf{x}},\mathbf{y},\mathbf{F_1},\mathbf{F_2},\boldsymbol{\theta}|\boldsymbol{\phi},\sigma,\gamma) = p_{\mathrm{GP}}(\mathbf{x},\dot{\mathbf{x}},\mathbf{y}|\boldsymbol{\phi},\sigma)p_{\mathrm{ODE}}(\mathbf{F_1},\mathbf{F_2},\boldsymbol{\theta}|\mathbf{x},\dot{\mathbf{x}},\gamma),$$

where

$$p_{\mathrm{GP}}(\mathbf{x},\dot{\mathbf{x}},\mathbf{y}|\boldsymbol{\phi},\sigma) = p(\mathbf{x}|\boldsymbol{\phi})p(\dot{\mathbf{x}}|\mathbf{x},\boldsymbol{\phi})p(\mathbf{y}|\mathbf{x},\sigma)$$

and

$$p_{\mathrm{ODE}}(\mathbf{F_1},\mathbf{F_2},\boldsymbol{\theta}|\mathbf{x},\dot{\mathbf{x}},\gamma) = p(\boldsymbol{\theta})p(\mathbf{F_1}|\boldsymbol{\theta},\mathbf{x})p(\mathbf{F_2}|\dot{\mathbf{x}},\gamma\mathbf{I})\delta(\mathbf{F_1}-\mathbf{F_2}).$$

To simplify this formula, p_{ODE} can be reduced by marginalizing out $\mathbf{F_2}$ using the properties of the Dirac delta function and the probability density defined in equation (14). The new p_{ODE} is then independent of $\mathbf{F_2}$.

$$p_{\mathrm{ODE}}(\mathbf{F_1},\boldsymbol{\theta}|\mathbf{x},\dot{\mathbf{x}},\gamma) = p(\boldsymbol{\theta})p(\mathbf{F_1}|\boldsymbol{\theta},\mathbf{x})\mathcal{N}(\mathbf{F_1}|\dot{\mathbf{x}},\gamma\mathbf{I}).$$

Inserting equation (13) yields

$$p_{\mathrm{ODE}}(\mathbf{F_1},\boldsymbol{\theta}|\mathbf{x},\dot{\mathbf{x}},\gamma) = p(\boldsymbol{\theta})\delta(\mathbf{F_1}-\mathbf{f}(\mathbf{x},\boldsymbol{\theta}))\mathcal{N}(\mathbf{F_1}|\dot{\mathbf{x}},\gamma\mathbf{I}).$$

Again, the properties of the Dirac delta function are used to marginalize out $\mathbf{F_1}$. The new p_{ODE} is now independent of $\mathbf{F_1}$,

$$p_{\mathrm{ODE}}(\boldsymbol{\theta}|\mathbf{x},\dot{\mathbf{x}},\gamma) = p(\boldsymbol{\theta})\mathcal{N}(\mathbf{f}(\mathbf{x},\boldsymbol{\theta})|\dot{\mathbf{x}},\gamma\mathbf{I}).$$

This reduced p_{ODE} is now combined with p_{GP}. Observing that the mean and the argument of a normal density are interchangeable and inserting the definition of the GP prior on the derivatives given by equation (5) leads to

$$p(\mathbf{x},\dot{\mathbf{x}},\mathbf{y},\boldsymbol{\theta}|\boldsymbol{\phi},\sigma,\gamma) = p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\phi})\mathcal{N}(\dot{\mathbf{x}}|\mathbf{Dx},\mathbf{A})p(\mathbf{y}|\mathbf{x},\sigma)\mathcal{N}(\dot{\mathbf{x}}|\mathbf{f}(\mathbf{x},\boldsymbol{\theta}),\gamma\mathbf{I}).$$

$\dot{\mathbf{x}}$ can now be marginalized by observing that the product of two normal densities of the same variable is again a normal density. The formula can be found, e.g., in Petersen et al. (2008). As a result, one obtains

$$p(\mathbf{x},\mathbf{y},\boldsymbol{\theta}|\boldsymbol{\phi},\sigma,\gamma) = p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\phi})p(\mathbf{y}|\mathbf{x},\sigma)\mathcal{N}(\mathbf{f}(\mathbf{x},\boldsymbol{\theta})|\mathbf{Dx},\mathbf{A}+\gamma\mathbf{I}).$$

It should now be clear that after inserting equations (3) and (4) and renormalizing, we get the final result

$$p(\mathbf{x},\boldsymbol{\theta}|\mathbf{y},\boldsymbol{\phi},\gamma,\sigma) \propto p(\boldsymbol{\theta})\mathcal{N}(\mathbf{x}|\mathbf{0},\boldsymbol{C_\phi})\mathcal{N}(\mathbf{y}|\mathbf{x},\sigma^2\mathbf{I})\mathcal{N}(\mathbf{f}(\mathbf{x},\boldsymbol{\theta})|\mathbf{Dx},\mathbf{A}+\gamma\mathbf{I}),$$

concluding the proof of this theorem. □

7.3 Hyperparameter and kernel selection

As discussed before, the Gaussian process model is defined by a kernel function $k_\phi(t_i,t_j)$. For both the hyperparameters $\boldsymbol{\phi}$ and the functional form of k there exist many possible choices. Even though the exact choice might not be too important for consistency guarantees in GP regression (Choi and Schervish, 2007), this choice directly influences the amount of observations that are needed for reasonable performance. While there exist some interesting approaches to learn the kernel directly from the data, e.g., Duvenaud et al. (2013) and Gorbach et al. (2017b), these methods can not be applied due to the very low amount of observations of the systems considered in this paper. As in previous approaches, the kernel functional form is thus restricted to simple kernels with few hyperparameters, whose behaviors have already been investigated by the community, e.g., in the kernel cookbook by Duvenaud (2014). Once a reasonable kernel is chosen, it is necessary to fit the hyperparameter and depending on the amount of expert knowledge available, there are different methodologies.

7.3.1 Maximizing the data likelihood

As mentioned e.g. in Rasmussen (2004), it is possible for a Gaussian process model to analytically calculate the marginal likelihood of the observations $\mathbf{y}$ given the

evaluation times $\mathbf{t}$ and hyperparameters $\boldsymbol{\phi}$ and σ.

$$\begin{aligned}\log(p(\mathbf{y}|\mathbf{t},\boldsymbol{\phi},\sigma)) = \\ -\frac{1}{2}\mathbf{y}^T(\mathbf{C}_\phi+\sigma\mathbf{I})^{-1} \\ -\frac{1}{2}\log|\mathbf{C}_\phi+\sigma\mathbf{I}| \\ -\frac{n}{2}\log 2\pi \end{aligned} \tag{36}$$

where σ is the GPs estimate for the standard deviation of the observation noise and n is the amount of observations.

This equation is completely independent of the ODE system one would like to analyze and depends only on the observations of the states. To fit the GP model to the data, equation (36) can be maximized w.r.t. $\boldsymbol{\phi}$ and σ, without incorporating any prior knowledge.

7.3.2 Concurrent optimization

In AGM of Dondelinger et al. (2013), the hyperparameters are not calculated independent of the ODEs. Instead, a prior is defined and their posterior distribution is determined simultaneously with the posterior distribution over states and parameters by sampling from equation (8).

This approach has several drawbacks. As we shall see in section 5, its empirical performance is significantly depending on the hyperpriors. Furthermore, optimizing the joint distribution given equation (8) requires calculating the inverse of the covariance matrices $\mathbf{C}_\phi$ and $\mathbf{A}$, which has to be done again and again for each new set of hyperparameters. Due to the computational complexity of matrix inversion, this is significantly slowing down the optimization.

For these reasons, if strong prior knowledge about the hyperparameters is available, it might be better to incorporate it into the likelihood optimization presented in the previous section. There, it could be used as a hyperprior to regularize the likelihood.

7.3.3 Manual tuning

In the variational inference approach Gorbach et al. (2017a), the hyperparameters were assumed to be provided by an expert. If such expert knowledge is available, it should definitely be used since it can improve the accuracy drastically.

7.4 Adjusting the GP model

To make VGM more comparable to AGM, the hyperparameters of the kernel must be learned from the data. However, maximizing the data likelihood described in equation (36) directly using the prior defined in equation (4) will lead to very bad results.

7.4.1 Zero mean observations

The main reason for the poor performance without any pretreatment of $\mathbf{y}$ is the fact that the zero mean assumption in equation (36) is a very strong regularization for the amount of data available. As its effect directly depends on the distance of the true values to zero, it will be different for different states in multidimensional systems, further complicating the problem. Thus, it is common in GP regression to manipulate the observations such that they have zero mean.

This procedure can be directly incorporated into the joint density given by equation (8). It should be noted that for multidimensional systems this joint density will factorize over each state k, whose contribution will be given by

$$\begin{aligned} p(\mathbf{x_k}|\mathbf{y_k},\boldsymbol{\theta},\boldsymbol{\phi},\gamma,\sigma) \propto \mathcal{N}(\tilde{\mathbf{x}}_\mathbf{k}|\mathbf{0},\boldsymbol{C}_{\boldsymbol{\phi}_k}) \\ \times \mathcal{N}(\mathbf{y_k}|\mathbf{x_k},\sigma^2\mathbf{I}) \\ \times \mathcal{N}(\mathbf{f}_k(\mathbf{x_k},\boldsymbol{\theta})|\mathbf{D_k}\tilde{\mathbf{x}}_\mathbf{k},\mathbf{A_k}+\gamma\mathbf{I}) \end{aligned} \tag{37}$$

where

$$\tilde{\mathbf{x}}_\mathbf{k} = \mathbf{x_k} - \mu_{y,k}\mathbf{1}$$

using $\mu_{y,k}$ to denote the mean of the observations of the k-th state and $\mathbf{1}$ to denote a vector of ones with appropriate length.

It is important to note that this transformation is not just equivalent to exchanging $\mathbf{x_k}$ and $\tilde{\mathbf{x}}_\mathbf{k}$. While the transformation is not necessary for the observation term, as $\mathbf{x_k}$ and $\mathbf{y_k}$ would be shifted equally, the original $\mathbf{x_k}$ is needed as input to the ODEs. This allows for this transformation without the need to manually account for this in the structural form of the differential equations.

This trick will get rid of some of the bias introduced by the GP prior. In the simulations, this made a difference for all systems, including the most simple one presented in section 5.1.

7.4.2 Standardized states

If the systems get more complex, the states might be magnitudes apart from each other. If one were to use the same hyperparameters $\boldsymbol{\phi}$ for all states, then a deviation $(\mathbf{F_k}-\mathbf{D_k}\tilde{\mathbf{x}}_\mathbf{k}) = 10^{-4}$ would contribute equally to the change in probability, independent of whether the states $\tilde{\mathbf{x}}_\mathbf{k}$ are of magnitude 10^{-8} or 10^3. Thus, small relative deviations from the mean of states with large values will lead to stronger changes in the joint probability than large relative deviations of states with small

values. This is not a desirable property, which can be partially alleviated by calculating a new set of hyperparameters for each state. However, this problem can be completely nullified by standardizing the data $\mathbf{y}$. For the k-th state, this would change its contribution to the joint density to

$$\begin{aligned} p(\mathbf{x_k}|\mathbf{y_k}, \boldsymbol{\theta}, \boldsymbol{\phi}, \gamma, \sigma) \propto & \mathcal{N}\left(\frac{1}{\sigma_{y,k}}\tilde{\mathbf{x}}_\mathbf{k} \middle| \mathbf{0}, \boldsymbol{C}_{\boldsymbol{\phi_k}}\right) \\ & \times \mathcal{N}\left(\frac{1}{\sigma_{y,k}}\mathbf{y_k} \middle| \frac{1}{\sigma_{y,k}}\mathbf{x_k}, \sigma^2\mathbf{I}\right) \\ & \times \mathcal{N}\left(\frac{1}{\sigma_{y,k}}\mathbf{f}_k(\mathbf{x_k}, \boldsymbol{\theta}) \middle| \frac{1}{\sigma_{y,k}}\mathbf{D_k}\tilde{\mathbf{x}}_\mathbf{k}, \mathbf{A_k} + \gamma\mathbf{I}\right) \end{aligned} \tag{38}$$

where

$$\tilde{\mathbf{x}}_\mathbf{k} = \mathbf{x_k} - \mu_{y,k}\mathbb{1}$$

and $\sigma_{y,k}$ is the standard deviation of the observations of the k-th state.

For complex systems with states on different orders of magnitude, standardization is a must to obtain reasonable performance. Even for the system presented in section 5.2, standardization has a significantly beneficial effect, although the states do not differ greatly in magnitude.

7.5 Algorithmic details

For all experiments involving AGM, the R toolbox deGradInfer (Macdonald and Dondelinger, 2017) published alongside Macdonald (2017) was used. For comparability, no priors were used, but the toolbox needed to be supplied with a value for the standard deviation of the observational noise and the true standard deviation of the noise was used. For all other parameters, e.g., amount of chains or samples, the values reported in Dondelinger et al. (2013) were used.

For MVGM and FGPGM, the hyperparameters of the GP model were determined in a preprocessing step identical for both algorithms. After calculating the hyperparameters, the MVGM parameters were inferred using the implementation used by Gorbach et al. (2017a).

Both MVGM and FGPGM need to be supplied with γ, which was treated as a tuning parameter. In principle, this parameter could be found by evaluating multiple candidates in parallel and choosing based on data fit.

For both experiments, the amount of iterations recommended by the AGM toolbox (Macdonald and Dondelinger, 2017) have been used, namely 100'000 iterations for Lotka Volterra and 300'000 iterations for Protein Transduction. This is the same setup that has been used to obtain the parameter estimates shown throughout the paper. The simpler sampling setup of FGPGM clearly leads to running time savings of about a third compared to AGM. Thus, FGPGM is not only significantly more accurate, but also significantly faster than AGM. Dondelinger et al. (2013) have shown that this also implies order of magnitude improvements if compared to the running time of approaches based on numerical integration.

All experiments were performed using the ETH cluster.[2]. It should be noted that the algorithms were implemented in different programming languages and the cluster consists of cores with varying computation power, adding some variability to the running time estimates.

7.6 Lotka Volterra

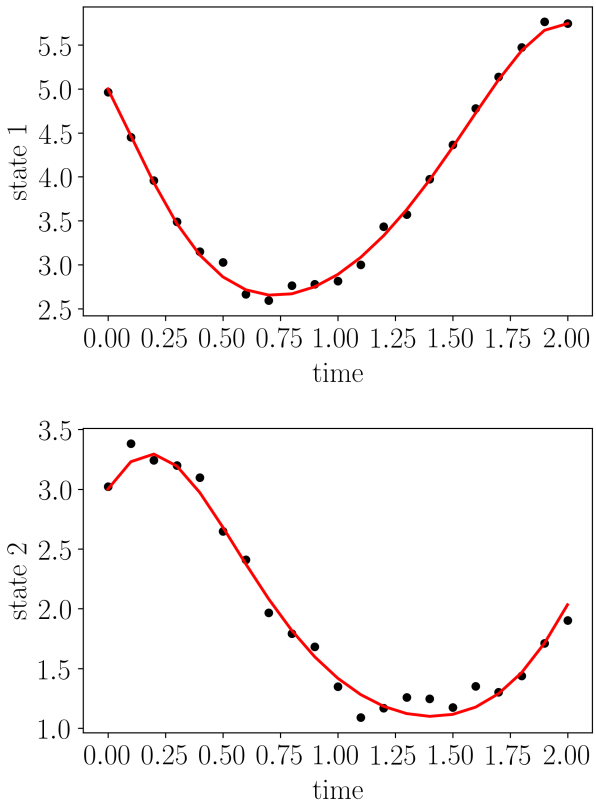


Figure 9: Example rollout of the Lotka Volterra system showing the state evolution over time. The dots denote the observations while the line represents the ground truth obtained by numerical integration.

As in the previous publications, a squared exponential kernel was used. For FGPGM and MVGM, γ was set to 0.3, while AGM was provided with the true observation noise standard deviations σ. The standard

[2] https://scicomp.ethz.ch/wiki/Euler#Euler_I

deviation of the proposal distribution of FGPGM was chosen as 0.075 for state proposals and as 0.09 for parameter proposals to roughly achieve an acceptance rate of 0.234. For all algorithms, it was decided to use only one GP to fit both states. This effectively doubles the amount of observations and leads to more stable hyperparameter estimates. As the state dynamics are very similar, this approximation is feasible.

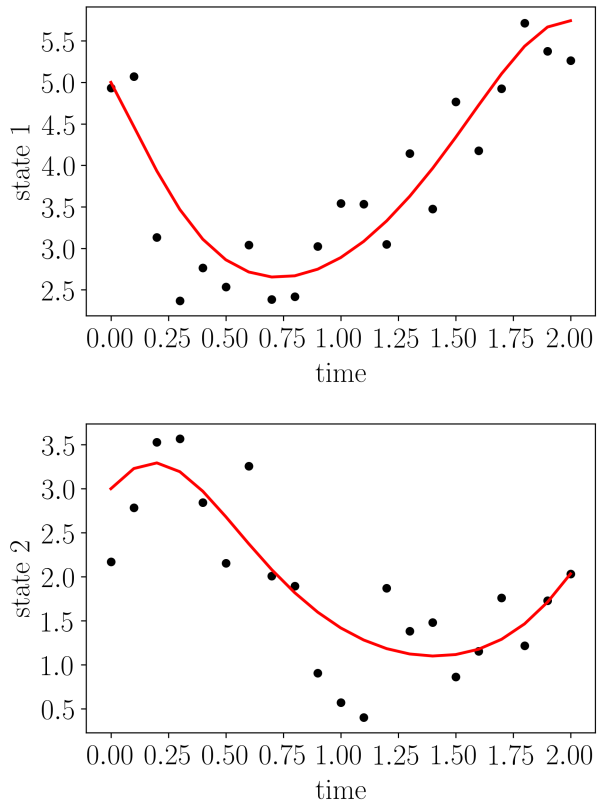


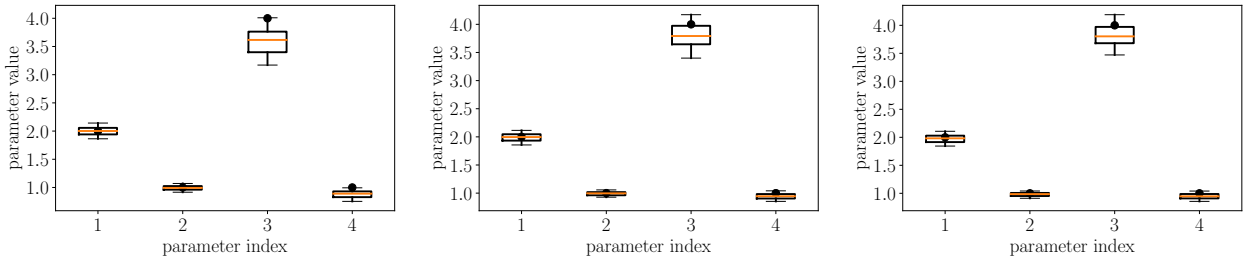
Figure 10: Example rollout of the Lotka Volterra system showing the state evolution over time for the high noise case. The dots denote the observations while the line represents the ground truth obtained by numerical integration.

7.7 Parameter Distribution

The MCMC approach of FGPGM allows to infer the probability distribution over parameters. This is shown for one example rollout in Figure 14. The inferred distributions are close to Gaussian in shape. This likely explains the sampling-like performance of the variational approach MVGM, as their assumptions of using a factorized Gaussian proxy distribution over the parameters seems to be a good fit for the true distribution.

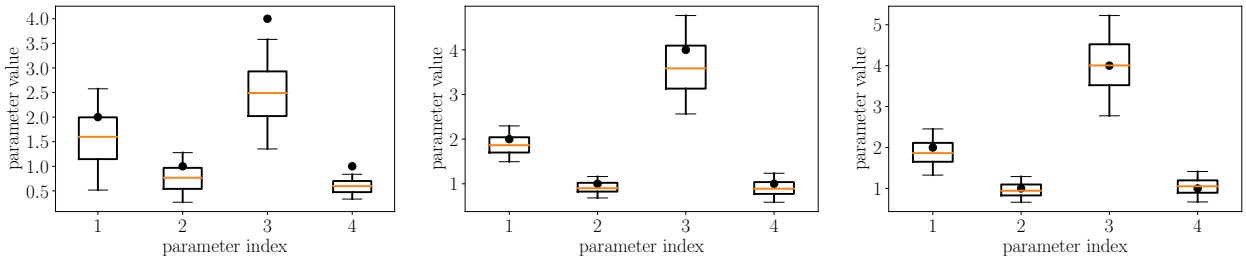
7.8 Protein Transduction

Figure 15 and Figure 16 show median plots for the states obtained by numerical integration of the inferred parameters of the Protein Transduction system.



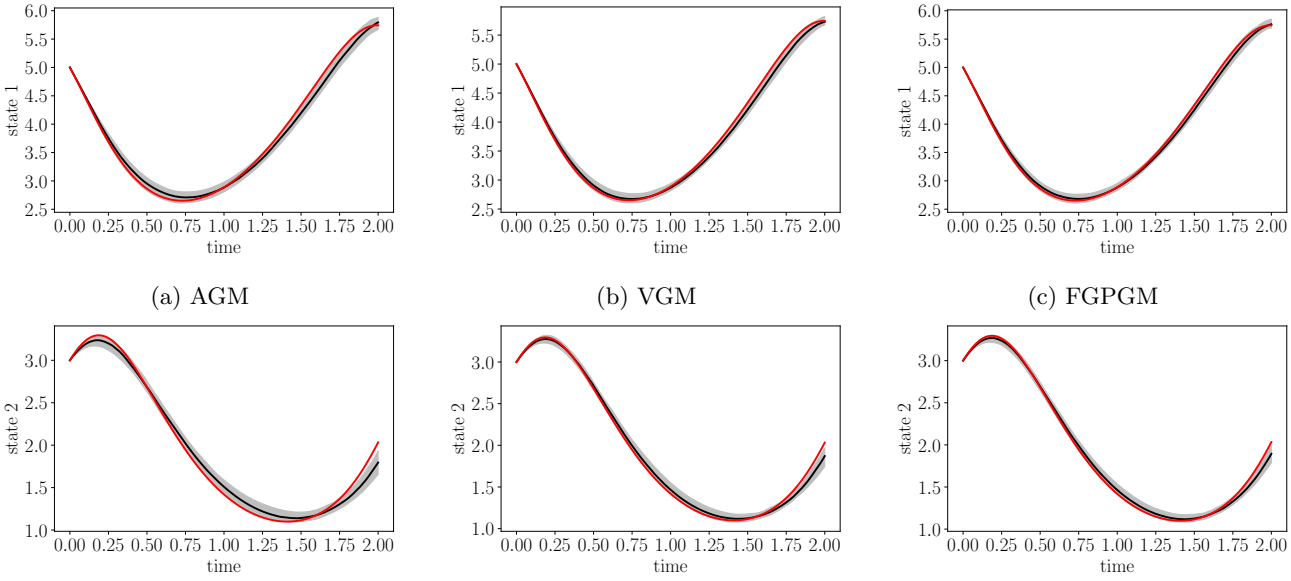
(a) AGM (b) FGPGM (c) MVGM

Figure 11: Inferred parameters for the low noise case. Ground truth (black dots), median (orange line), 50% (boxes) and 75% (whiskers) quantiles evaluated over 100 independent noise realizations are shown.



(a) AGM (b) FGPGM (c) MVGM

Figure 12: Boxplots showing the inferred parameters over 100 runs for the Lotka Volterra dynamics for the high noise case. The black dots represent the ground truth, the orange line denotes the median of the 100 parameter estimates while the boxes and whiskers denote 50% and 75% quantiles.



(a) AGM (b) VGM (c) FGPGM

Figure 13: Median Plots for all states of Lotka Volterra with low noise. State 1 is in the top row, state 2 is in the bottom row. The red line is the ground truth, while the black line and the shaded area denote the median and the 75% quantiles of the results of 100 independent noise realizations. As was already to be expected by the parameter estimates, FGPGM and VGM are almost indistinguishable while AGM falls off a little bit.

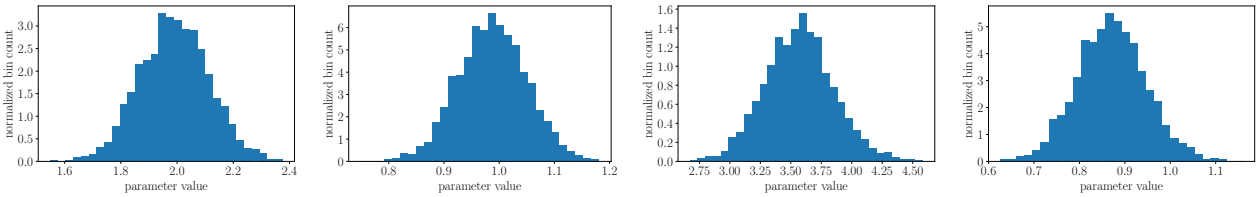


Figure 14: Histograms representing the MCMC samples obtained for one example run of the Lotka Volterra system. Each histogram represents the marginal distribution of one ODE parameter.

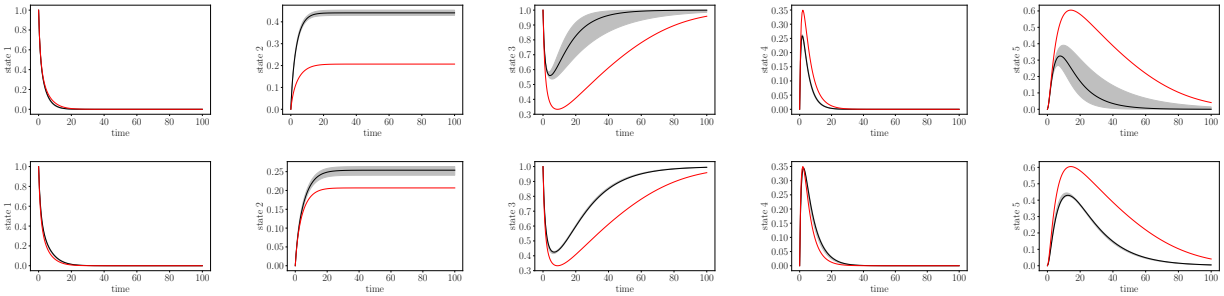


Figure 15: Median plots for all states of the most difficult benchmark system in the literature, Protein Transduction. The red line is the ground truth, while the black line and the shaded area denote the median and the 75% quantiles of the results of 100 independent noise realizations. FGPGM (middle) is clearly able to find more accurate parameter estimates than AGM (top).

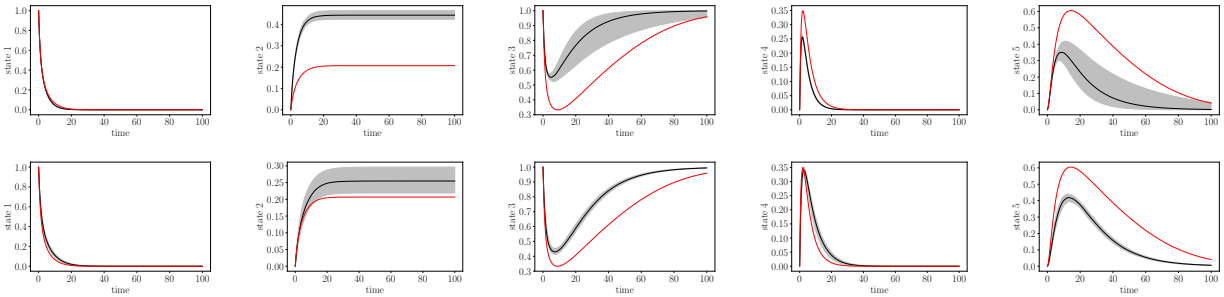
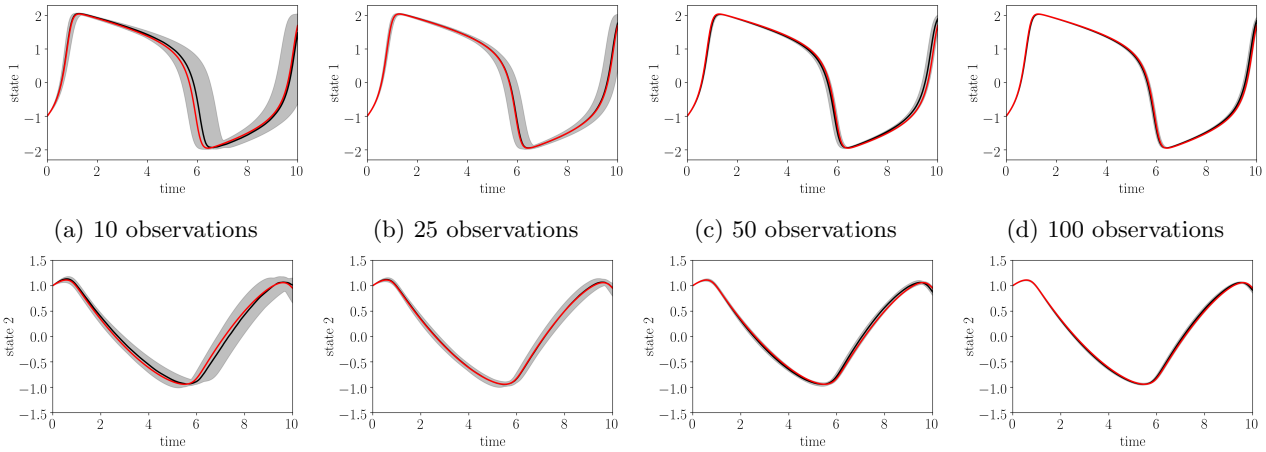


Figure 16: Median Plots for all states of the most difficult benchmark system in the literature, Protein Transduction, for the high noise case.The red line is the ground truth, while the black line and the shaded area denote the median and the 75% quantiles of the results of 100 independent noise realizations. FGPGM (middle) is clearly able to find more accurate parameter estimates than AGM (top).



(a) 10 observations (b) 25 observations (c) 50 observations (d) 100 observations

Figure 17: Median plots of the numerically integrated states after parameter inference for the FHN system with SNR 10. Ground truth (red), median (black) and 75% quantiles (gray) over 100 independent noise realizations.