Computational and Applied Mathematics (2023) 42:36
https://doi.org/10.1007/s40314-022-02180-y

Data-driven learning of differential equations: combining data and model uncertainty

Karl Glasner[1]

Received: 9 June 2022 / Revised: 25 October 2022 / Accepted: 21 December 2022

Abstract
Data-driven discovery of differential equations relies on estimating model parameters using information about a solution that is often incomplete and corrupted by noise. Moreover, the sizes of the uncertainties in the model and data are usually unknown as well. This paper develops a likelihood-type cost function which incorporates both sources of uncertainty and provides a theoretically justified way of optimizing the balance between them. This approach accommodates missing information about model solutions, allows for considerable noise in the data, and is demonstrated to provide estimates which are often superior to regression methods currently used for model discovery and calibration. Practical implementation and optimization strategies are discussed both for systems of ordinary differential and partial differential equations. Numerical experiments using synthetic data are performed for a variety of test problems, including those exhibiting chaotic or complex spatiotemporal behavior.

Keywords Parameter estimation · Model error · Learning differential equations

Mathematics Subject Classification 62J02 · 65M32 · 65L09

Introduction

Increases in computing power and the potential of leveraging machine learning concepts in scientific modeling have led to growing interest in data-driven model discovery and calibration. This is especially necessary in fields such as materials science, geophysics and biology, where first principle model derivations are elusive. Even when this can be done successfully, inference of constitutive details using observations can prove more satisfactory than theoretical derivations because of inherent uncertainties about the model itself.

There have been a variety of methods proposed recently for data-driven discovery of differential equations (Brunton et al. 2016; Rudy et al. 2017; Schaeffer 2017; Wang et al.

Communicated by Vinicius Albani.

✉ Karl Glasner
kglasner@math.arizona.edu

[1] Department of Mathematics, University of Arizona, Tucson, AZ 85721, USA

Published online: 07 January 2023

2019; Maddu et al. 2019; Long et al. 2019; Raissi et al. 2019). Discovery of model equations has two aspects: determining the model's structure based on criteria that are either physically inspired or rely on notions of sparsity; and secondly, estimation of model parameters within this framework. The first aspect relies to a great degree on the second, and this is the one which we will focus on. Estimation is often accomplished by optimizing a cost function that accounts for uncertainties in the data, model equation, or both. In the latter case, the relative trade-off between terms in the cost function has not been extensively investigated, and theoretical justifications for such a cost functions are generally not provided. This paper's aim is to provide a theoretically motivated regression method that is able to automatically assess the relative sizes of data and model uncertainty.

The importance of considering uncertainties in both data and underlying models has been understood for some time, in particular in the context of Bayesian inference (Kennedy and O'Hagan 2001; Brynjarsdottir and O'Hagan 2014; Plumlee 2017; Sargsyan et al. 2019). The high dimensionality in the problems considered here creates significant challenges for Bayesian estimation, which likely explains its absence in many popular techniques for data-driven discovery of differential equations. In the spirit of these works, we consider the simpler task of parameter estimation via regression and leave generalizations for future work.

Regression techniques have long enjoyed success in estimating parameters in differential equations. A common approach is to minimize the difference (typically least squares) between the model output and the provided data over a set of model parameters (Fullana et al. 1997; Ackleh et al. 1998; Ashyraliyev et al. 2008; Garvie et al. 2010; Jin and Maass 2012; Croft et al. 2015; Ramsay and Hooker 2017; Sgura et al. 2019; Zhao et al. 2020). The major drawbacks of this method are inherent to all non-convex optimization problems: algorithms for their solution may be inefficient, and proliferation of local minima can make the problem intractable. Another less obvious issue is that most models are themselves uncertain, and the statistical underpinning of this type of regression is predicated on uncertainty in the data alone.

An alternative approach to determining parameters involves minimizing the residual error, which is the discrepancy in the model equations evaluated using data (Ramsay and Hooker 2017; Brunel 2008; Gugushvili and Klaassen 2012; Liang and Hulin 2008; Brunton et al. 2016; Schaeffer 2017). For models that are linear in their parameters, this amounts to a standard least squares problem, whose solution is both unique and easily obtained. On the other hand, it is well accepted that this method is highly susceptible to noisy data (Schaeffer and McCalla 2017; Reinbold et al. 2020). This is not surprising, as in this case the cost function may be interpreted as a measure of model uncertainty alone.

It also possible to combine the two regression approaches (Ramsay et al. 2007; Voss et al. 2004; Raissi and Karniadakis 2018). The balance between the two loss function terms is provided by a interpolating hyperparameter which is often selected according to the user's intuition or taste. The approach developed in this paper falls into the category of hybrid methods but in our case the interpolation parameter is selected automatically by a rationally-based criteria.

Outside of these regression techniques, there are many other proposals for estimation and model discovery in differential equations. These include long-standing approaches like Kalman filter techniques (Evensen 2009), Gaussian process regression (Raissi et al. 2017; Raissi and Karniadakis 2018), as well as implementations of Bayesian inference in specific applications (Dewar et al. 2010; Barajas-Solano and Tartakovsky 2019; Yoshinaga and Tokuda 2020; Zhao et al. 2021). In addition, neural networks and deep learning technologies have been explored for learning partial differential equations from data (Long et al. 2019; Raissi and Karniadakis 2018; Raissi et al. 2019; Xu et al. 2019).

This paper relies on a classical approach to parameter identification by deriving a likelihood function incorporating uncertainty in both the supplied data and the model itself. This results in a two-stage optimization procedure, which is solved using techniques drawn from both optimization and numerical partial differential equations. We find several virtues to this formulation:

- Estimates are generally superior to other regression techniques, especially when significant noise is added to both the model and data.
- Incomplete data is utilized in a natural way, and missing data is reconstructed as a by-product of the method.
- Difficulties inherent in non-convex optimization are mitigated by a convex term that effectively regularizes the problem.

This paper is organized as follows. Section 1 describes the problem setup and reviews existing regression-type methods. Our new formulation is derived in Sect. 2, where practical implementation issues are discussed. Applications to ordinary differential equations are given in Sect. 3. Finally, some examples of pattern formation and spatio-temporal behavior in PDEs is given in 4.

1 Optimization formulations of parameter estimation

This paper considers a model of the form

$$N(u;\theta) = 0, \tag{1}$$

where u is called the state variable and θ is a vector comprising the unknown parameters. $N()$ is assumed to be a smooth, generic nonlinear operator, which may arise from either algebraic or differential equations. Although the state variable may formally be in an infinite dimensional space (as for differential equations), in practice, discretization renders it finite. It is therefore sufficient to suppose that $u \in \mathbb{R}^n$ and $N : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$, where n is the number of degrees of freedom in u and p the number of parameters. The fundamental challenge is to infer values of θ given information about the state u, denoted $\hat{u}$. Here the domain D of $\hat{u}$ may be smaller than u itself, representing incomplete information about the state u.

One widely adopted approach to this problem is to find the best fit between solutions of (1) and the supplied data (Fullana et al. 1997; Ackleh et al. 1998; Ashyraliyev et al. 2008; Garvie et al. 2010; Jin and Maass 2012; Croft et al. 2015; Ramsay and Hooker 2017; Sgura et al. 2019; Zhao et al. 2020). Supposing that for each θ this equation has a unique solution $\overline{u}(\theta)$, the parameter estimate θ^* can be obtained as a nonlinear least squares (NLS) problem

$$\theta^* = \text{argmin}_\theta \left\| \overline{u}(\theta)|_D - \hat{u} \right\|^2, \tag{2}$$

where $|_D$ is restriction to the domain of $\hat{u}$. In principle, various choices for norm $||\cdot||$ can be made, but for simplicity this will always denote the usual Euclidean norm here. The formulation (2) is generally robust to noise in the data, but suffers from the usual problems with non-convex optimization; in particular, there are often a great number of irrelevant local minima (see, for example, Sect. 3.2).

A complimentary method is to employ the estimate

$$\theta^* = \text{argmin}_\theta ||N(\hat{u};\theta)||^2. \tag{3}$$

instead. This formulation has been alternately called residual minimization (RM) or "gradient matching" in the differential equations context (Ramsay and Hooker 2017). Provided $N()$ is linear in its parameters, this is simply a linear least squares problem. This type of formulation is the basis for recent sparse regression techniques (Brunton et al. 2016; Schaeffer 2017; Rudy et al. 2017), and can be solved very efficiently. The main drawback, especially in differential equations, is that noise in the data easily corrupts the estimate. In addition, incomplete data is not automatically accommodated.

Remedies for using residual-minimization type estimates to handle noisy and incomplete data have been proposed. The simplest way of dealing with noise in the data is by low-pass filtering (Schaeffer 2017). This process can improve the quality of estimates but requires a tuning of the filter which is not a-priori known (Glasner 2021). An attractive alternative is to replace the residual with integral terms, akin to weak formulations of differential equations (Schaeffer and McCalla 2017; Reinbold et al. 2020; Messenger and Bortz 2021).

It is tempting to combine (2) and (3) in order to mitigate the drawbacks of each, for example by minimization over a linear combination (Voss et al. 2004; Raissi and Karniadakis 2018). Another hybrid method is to minimize both over the state and parameter values, for example

$$\min_{u,\theta} \left\| N(u;\theta) \right\|^2 + \lambda \left\| u|_D - \hat{u} \right\|^2, \tag{4}$$

where λ is an adjustable hyperparameter. This is a form of "profiled estimation" (Ramsay et al. 2007; Ramsay and Hooker 2017), which has been observed to mitigate some of the drawbacks of both nonlinear least squares (2) and residual minimization (3). Notice that as $\lambda \to \infty$, the minimizer in (4) is approximately $u = \hat{u}$, and therefore θ approximately minimizes (3). Conversely, if $\lambda \to 0$, then $u \approx \overline{u}(\theta)$, which means that θ is selected as in (2). Therefore (4) can be thought of as an interpolation of both regression approaches, and limits to each one for large or small λ; we exploit this fact when comparing our approach to others.

The optimization problem derived below is in some ways an extension of (4), and provides a statistical interpretation that justifies its importance. In addition, a selection mechanism is provided for the interpolation parameter λ.

2 Approximate model inference

To capture uncertainty in the observed state $\hat{u}$ and the model, it is supposed that

$$\hat{u} = u|_D + \tilde{u}, \quad N(u;\theta) + \tilde{N} = 0, \tag{5}$$

where the unknown discrepancies $\tilde{u}, \tilde{N}$ are regarded as component-wise random variables with distribution functions f_u, f_N, respectively, which have single isolated maxima. It is sufficient to assume that $\tilde{u}, \tilde{N}$ have zero mean, since the parameter set can be amended to include the means m_u, m_N, and $\hat{u}$ and $\tilde{N}$ can be replaced with $\hat{u} - u_m$ and $\tilde{N} - m_N$, respectively.

The approximations which follow are based on an assumption of small variances. In this limit, it is reasonable to suppose that f_u, f_N are approximated by normal distributions with variances σ_1^2, σ_2^2, respectively, as tails of the distribution provide only a subdominant contribution. The conditional probability distribution of observing data $\hat{u}$ given the actual

state u is therefore

$$P(\hat{u}|u) \approx (\sqrt{2\pi}\sigma_1)^{-\hat{n}} \exp\left(-\frac{||\hat{u} - u|_D||^2}{2\sigma_1^2}\right), \tag{6}$$

where $\hat{n} \leq n$ is the number of components in the provided data. The probability distribution of u given parameter θ is likewise

$$P(u|\theta) \approx (\sqrt{2\pi}\sigma_2)^{-n} \exp\left(-\frac{||N(u;\theta)||^2}{2\sigma_2^2}\right) |\det \nabla N(u)|. \tag{7}$$

Assuming $\tilde{u}$, $\tilde{N}$ are independent, it follows that the probability of the observation for a given parameter value can be approximated

$$P(\hat{u}|\theta) = (\sqrt{2\pi}\sigma_1)^{-\hat{n}} (\sqrt{2\pi}\sigma_2)^{-n} \int_{\mathbb{R}^n} \exp\left(-\frac{||\hat{u} - u|_D||^2}{2\sigma_1^2} - \frac{||N(u;\theta)||^2}{2\sigma_2^2}\right) |\det \nabla N(u;\theta)| du. \tag{8}$$

For small variances, the integral may be approximated by Laplace's method. Setting $\lambda = \sigma_2^2/\sigma_1^2$, the resulting approximation is

$$P(\hat{u}|\theta) \approx \frac{\lambda^{\hat{n}/2}}{(\sqrt{2\pi}\sigma_2)^{\hat{n}} \sqrt{|\det \nabla^2 F(u^*, \theta, \lambda)|}} \exp\left(-\frac{F(u^*, \theta, \lambda)}{2\sigma_2^2}\right) |\det \nabla N(u^*;\theta)|, \tag{9}$$

where

$$u^*(\theta, \lambda) \equiv \text{argmin}_u F(u, \theta, \lambda), \quad F \equiv \lambda \left\| \hat{u} - u|_D \right\|^2 + \left\| N(u;\theta) \right\|^2. \tag{10}$$

Here it is assumed that the minimizer in (10) is unique, which does not seem problematic in practice.

The corresponding log-likelihood function is

$$\boldsymbol{l}(\theta, \sigma_2, \lambda) = \frac{\hat{n}}{2} \ln(\lambda/\sigma_2^2) - \frac{1}{2} \ln |\det \nabla^2 F(u^*, \theta, \lambda)| - \frac{1}{2\sigma_2^2} F(u^*, \theta, \lambda) + \ln |\det \nabla N(u^*;\theta)|. \tag{11}$$

Maximizing this over σ_2 leads to

$$\sigma_2^2 = \frac{F(u^*, \theta, \lambda)}{\hat{n}}. \tag{12}$$

which provides an estimate of the variance. Inserting into (11) leads to the reduced likelihood function

$$\boldsymbol{l}_2(\theta, \lambda) = \frac{\hat{n}}{2} \ln \lambda - \frac{1}{2} \ln |\det \nabla^2 F(u^*, \theta, \lambda)| - \frac{\hat{n}}{2} \ln F(u^*, \theta, \lambda) + \ln |\det \nabla N(u^*;\theta)|. \tag{13}$$

The estimates for both θ and λ can be obtained by setting

$$(\theta, \lambda) = \text{argmax} \boldsymbol{l}_2(\theta, \lambda) \tag{14}$$

The estimation procedure outlined will be referred hereafter as approximate model inference (AMI).

2.1 Optimization methods

The problem (14) involves two levels of optimization, an "inner" problem for the minimizing state u^* and an "outer" maximization over parameters. This can be accomplished either by determining $u^* = u^*(\theta, \lambda)$ exactly to evaluate (13), or by simultaneous minimization of $F(u, \theta, \lambda)$ and $\boldsymbol{l}_2(\theta, \lambda; u)$, where u is regarded as a parameter in the latter objective function. The best choice appears to be problem specific, and is described for the examples below.

There are a variety of algorithms that can be used for unconstrained optimization in (10). It should be noted for differential equations, gradient descent of (10) is numerically equivalent to a parabolic partial differential equation, often of high order. Therefore, explicit gradient descent methods popular in machine learning contexts (e.g. Kingma and Ba 2014) are not appropriate. Alternatively, Newton and quasi-Newton methods may have difficulty converging for highly non-convex problems. A reasonable compromise between gradient descent and Newton approaches is the Levenberg-Marquardt (LM) method. The LM method is practical when the associated Hessian is not too difficult to invert. This is not typically the case for discretizations of PDE, however. Two possible alternatives are semi-implicit gradient descent methods, and nonlinear conjugate gradient (NCG) methods (Hager and Zhang 2006). For the latter, efficiency requires a good choice of preconditioner; we explain how this is done in the context of our examples.

Optimization over parameters (14) is generally a low dimensional problem, however, the implicit dependence on u^* makes analytic evaluations of the gradient and Hessian complicated. This can be avoided by quasi-Newton methods which rely solely on evaluations of the objective function.

The function (10) is generally non-convex, and might appear to create intractable difficulties associated with a multitude of local minima. In practice, this does not seem to be the case, which might be due to the fact that the term involving λ partially convexifies the problem. In general, the choice of a moderate (typically between 1 and 100) initial guess for λ seems to be beneficial for both efficiency of the optimization algorithm and reasonableness of the resulting estimates (Glasner 2021).

In the following experiments, we will make comparisons to approximate NLS (2) and RM (3) estimates. We do this by fixing $\lambda = 10^{-3}$ and $\lambda = 10^3$, respectively in (10) and carry out maximization only over the second term in (13) using the same algorithm as for the AMI estimate (14).

3 Ordinary differential equations

Here we consider problems of the form

$$u_t = f(u; \theta), \quad u(0) = U, \tag{15}$$

where $u : \mathbb{R} \to \mathbb{R}^k$ and $0 < t < T$. To discretize this system, the values of u at times $t_j = jh$ are given by u_j, where h is the step size, and $j = 0, 1, 2, ..., m$ with $m = T/h$. A simple approximation of (15) can be obtained by the trapezoid rule, so that the nonlinear operator is defined by

$$N_j(u; \theta) \equiv \frac{u_{j+1} - u_j}{h} - hf([u_j + u_{j+1}]/2; \theta), \quad j = 1, 2, \ldots, m. \tag{16}$$

To account for the initial condition, we define $N_0 \equiv u_0 - U$, so that all together N has $k(m + 1)$ equations and unknowns. Although the initial condition U can be regarded as

another parameter to be determined, U can be analytically eliminated since maximization in (13) over U leads to $U = u_0$. We can therefore reduce (10) to

$$F(u, \theta, \lambda) = \sum_{j=1}^{m} |N_j(u, \theta)|^2 + \lambda \sum_{j=0}^{m} |\hat{u}_j - u_j|^2. \tag{17}$$

Here it is supposed to begin with that the data $\hat{u}$ is supplied as pointwise values $(t_j, \hat{u}_j)$ (of course in practical applications, this may require interpolation of non-aligned data). The situation of incomplete data is addressed in the section 3.3.

The optimization procedure involves minimization of (17) for given values of θ, λ. This is accomplished efficiently by an adaptive version of the Levenberg-Marquardt algorithm, which interpolates between Gauss-Newton and gradient descent methods. This requires an initial guess for u, which is provided by $\hat{u}$ in the first evaluation of the minimum of (17), and subsequently by the results of the previous minimization.

Once the minimizer is determined, the likelihood function (13) may be evaluated. The log-determinant terms are straightforward to compute: $\nabla^2 F$ is provided by the Hessian used in the LM minimization, and the determinant is easily found from diagonal entries in an LU-decomposition. When the vector components u_j^k are ordered in a natural way, $\nabla N(u)$ represents a block lower triangular matrix. The determinant of this term results from the product of determinants of the diagonal blocks.

Finally, optimization for $\boldsymbol{l}_2(\theta, \lambda)$ was accomplished by a general purpose quasi-Newton method that only requires evaluation of the objective function (rather than numerical gradients). In general, this produced good convergence with relatively few function evaluations, typically $100 - 200$.

MATLAB code for parameter estimates (14) suitable for arbitrary systems of ODEs is publicly available at https://github.com/karlglasner/AMI. This repository also includes codes used for specific subsequent examples.

3.1 Example: FitzHugh–Nagumo equations

To illustrate the proposed method and compare it to other estimation methods, we consider the well-known FitzHugh–Nagumo system

$$\frac{dV}{dt} = c^2(V - V^3/3 + R), \tag{18}$$

$$\frac{dR}{dt} = -(V - a - bR), \tag{19}$$

where $\theta = (a, b, c)$ are the parameters to be estimated. Synthetic data were obtained by numerical integration (fourth order Runge-Kutta), using the initial condition $V(0) = .1$, $R(0) = .2$ and exact parameters $\theta^* = (3.179, 0.258, 3)$. The solution was sampled at 500 evenly spaced values on the interval $[0, 10]$ (see Fig. 1), corresponding to $h = 0.02$ in (15). Normally distributed noise with variance n_d^2 was added pointwise to the sampled values. To incorporate model uncertainty, Wiener process (Brownian) noise with variance hn_m^2 was added to (18–19).

Estimation was carried out with three methods: the approximate NLS and RM methods discussed in Sect. 2.1, and AMI optimization. In all cases, initial guesses $(a, b, c) = (1, 1, 1)$ and $u = \hat{u}$ were used. The estimated parameter values were compared to the exact values by computing the relative error, defined as the Euclidean norm of $[(a-a^*)/a^*, (b-b^*)/b^*, (c-$

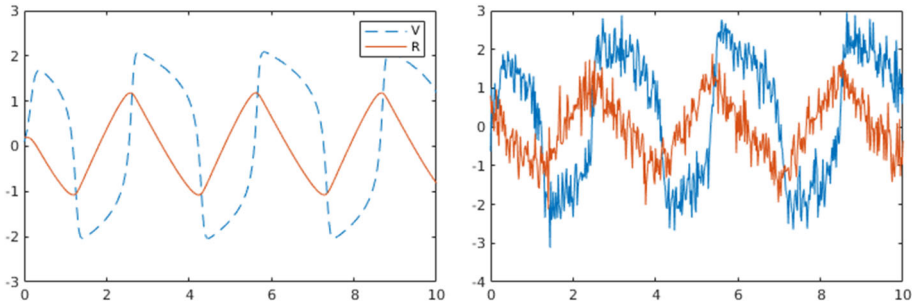

Fig. 1 Left: numerical solution to (18) without noise. Right: sampled data with noise levels $n_d = 0.4$ and $n_m = 0.1$

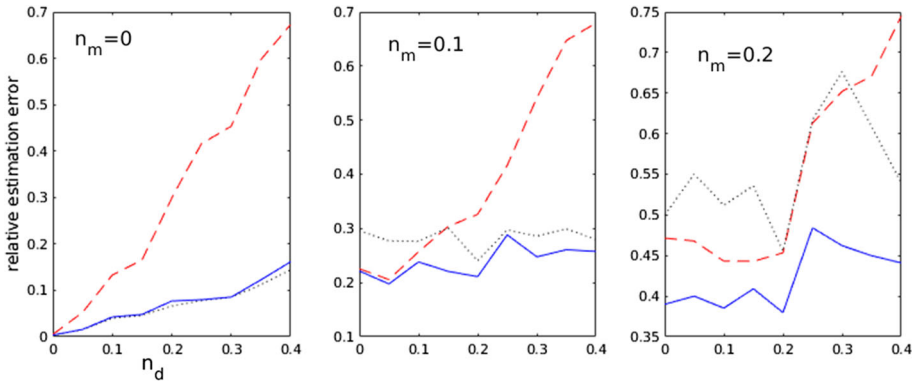

Fig. 2 Relative errors of parameter estimates for the FitzHugh-Nagumo system, using nonlinear least squares (dotted, black), residual minimization (dashed, red), and the AMI estimate (solid, blue). The errors were averaged over 20 realizations of synthetic data for each combination of n_d and n_m

$c^*)/c^*]$. For each value of n_d, n_m, 20 realizations of randomized noise were used, and the reported relative errors were averaged over these.

The results of the estimation procedure with differing levels of noise is shown in Fig. 2. In the absence of any noise, all estimation procedures yielded a learned system which reproduced the input solution almost exactly. The small discrepancy, in this case, can be ascribed to discretization error alone. With only noise added to the data (Fig. 2 left), the AMI and NLS estimates were nearly identical. Residual minimization, on the other hand, produced generally poor results, which might be expected since it essentially relies on differentiating noisy data. When only model noise was added, again the AMI and NLS results were similar, and better than the RM estimate. In this case, the parameter estimation error can be ascribed to the uncertainty in the model rather than the data. When both model and data noise are included (Fig. 2 middle and right), the AMI formulation produced superior estimates.

Notice that there is a crossover between the quality of the NLS and RM estimates, depending on whether data or model noise is dominant, consistent with the theory in Sect. 2. The virtue of the proposed method is to form a compromise between the two, determining the proper balance (via the parameter λ) from the data alone.

3.2 Example: Lorenz equations

Here we consider the well-studied Lorenz system

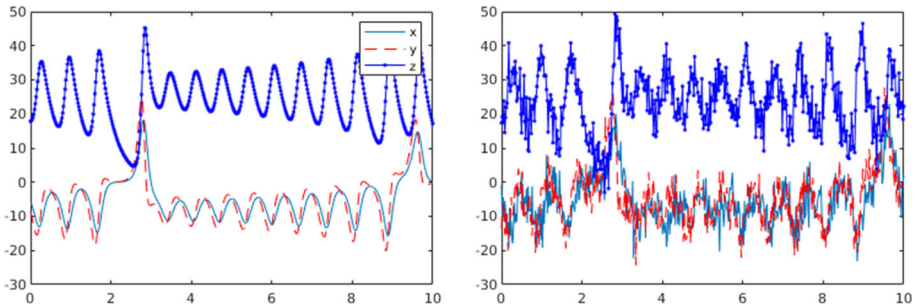

Fig. 3 Left: numerical solution to (20) without noise. Right: sampled data with noise levels $n_d = 4$ and $n_m = 2$

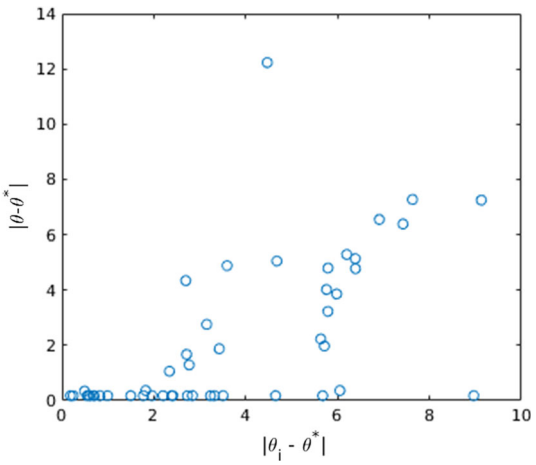

Fig. 4 Difference between NLS estimates and the exact parameter value θ^* (y-axis) versus the difference between initial guesses of the exact parameter value. This illustrates the proliferation of local minimizers

$$\frac{dx}{dt} = \sigma(y - x), \tag{20}$$

$$\frac{dy}{dt} = x(\rho - z) - y, \tag{21}$$

$$\frac{dz}{dt} = xy - \beta z, \tag{22}$$

where $\theta = (\sigma, \rho, \beta)$ is the parameter set. In the experiments below, synthetic data is created using $\theta^* = (10, 28, 8/3)$ and initial data $(x, y, z) = (-5.6866, -8.4929, 17.8452)$. These produce a trajectory near the classic chaotic attractor (see Fig. 3).

Various levels of model and data noise were tested, using the same procedure as in the previous example. Initial parameter guesses were also set to one as before. Significantly, optimization for the NLS method was typically unable to find reasonable estimates, even without noise, unless initial parameter guesses were artificially adjusted to be very close to the exact values. To illustrate this, 100 random initial guesses for θ were tried. NLS estimates (Fig. 4) were computed for each, showing the difficulty posed by local minimizers of (2). While methods to deal with this issue in non-convex optimization exist, it is not clear how practical they might be.

The results of the numerical experiments are shown in Fig. 5. The relative parameter errors were computed as before and averaged over 10 realizations of random noise. The estimates

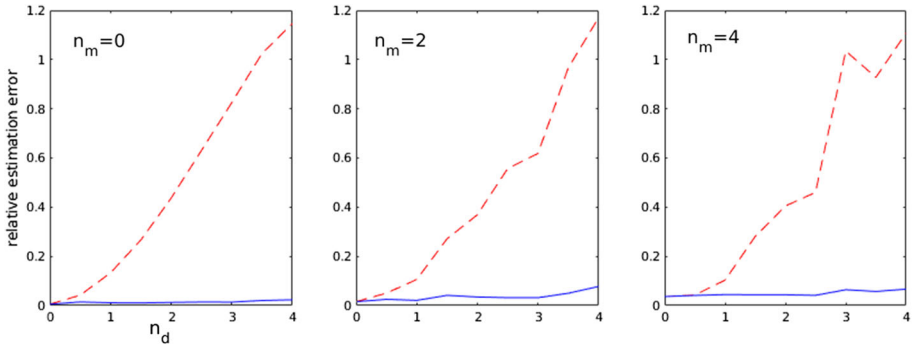

Fig. 5 Relative errors of parameter estimates for the Lorenz system, using residual minimization (dashed, red), and the AMI estimate (solid, blue). The errors were averaged over 10 realizations of synthetic data for each combination of n_d and n_m. Estimates using nonlinear least squares were extremely poor and are not shown

for NLS are not shown, since they were universally poor. As expected, the RM estimate is good at handling noise in the model but not the data. In contrast, the AMI method obtained good estimates for even the most extreme cases.

3.3 Incomplete data and data reconstruction

One virtue of the proposed methodology is that data may be supplied only over part of the domain. The estimation process not only provides parameter estimates but yields a state u^* which reconstructs the missing data.

To implement the optimization strategy detailed above, it is necessary to provide a reasonable initial guess for u. We find this is best done by first simply setting missing data points to zero, and optimizing with an intermediate fixed value of lambda ($\lambda = 100$ here) to obtain a crude estimate for parameters. The initial guess for u was then reset by filling in the missing data regions by integration of (20), using parameters obtained by the crude estimates. Finally, optimization with variable λ was performed with the new initial values for both u and θ.

To illustrate the performance when a limited amount of data is provided, noisy ($n_d = 0.4$) synthetic data for the system (18) was provided, and a certain percentage removed at random. Ten instances of random data removal were averaged for each percentage level. Figure 6 shows the quality of the estimates as a function of the data fraction. Reasonable estimates were observed when only 5% of the original data was retained.

To illustrate the ability to reconstruct data missing from an entire subdomain, synthetic data for the Lorenz system was generated as before, and data noise ($n_d = 4$) was added before the domain of $\hat{u}$ was restricted to $(0, 3) \cup (7, 10)$ (Fig. 7). The reconstruction is shown in the bottom panel of Fig. 7. The corresponding estimated parameters were $\theta = (10.1242, 28.0102, 2.6718)$, whose relative error was 1.26%. The L^∞ norm between the unadulterated data and the reconstructed state (top and bottom plots in Fig. 7) was 2.58.

The chaotic behavior of the Lorenz system likely limits the size of the interval that is removed from the data. Indeed, if the interval was much larger than the Lyapunov time, it would be impossible to accurately estimate the missing trajectories within this interval, even if the exact system was known. On the other hand, if the interest was in estimating only the parameters and the complete state, the remaining data might still be informative.

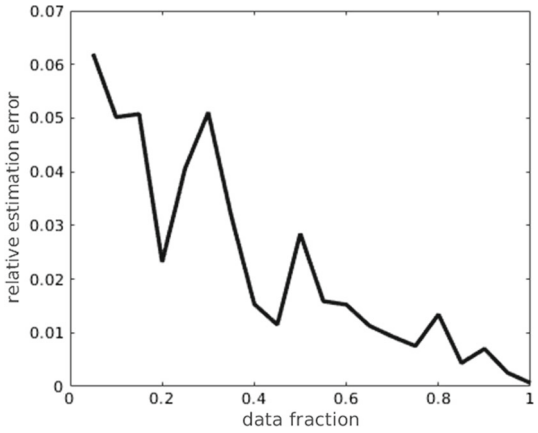

Fig. 6 Estimation error for the FitzHugh–Nagumo system as a function of the fraction of data values used in the AMI estimate

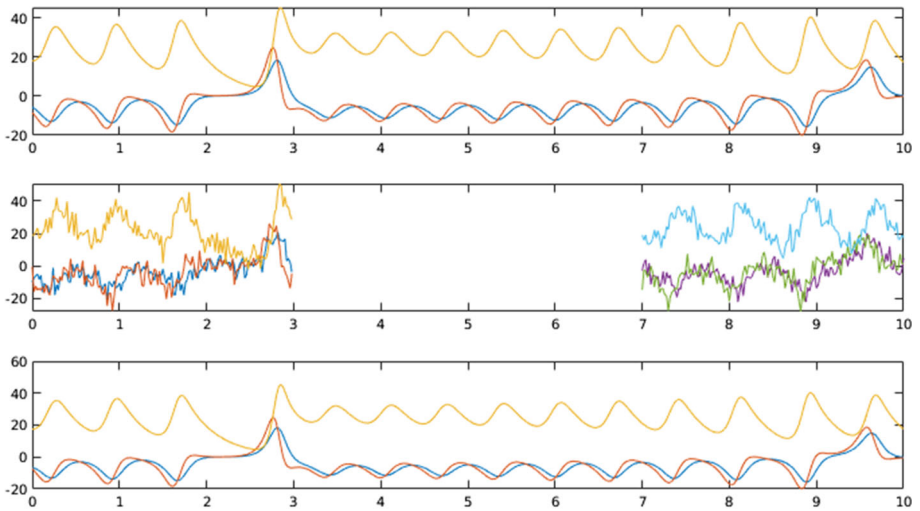

Fig. 7 Top: numerical simulation of the Lorenz system; middle: incomplete data with noise added; bottom: resulting state u^* after parameter optimization was complete

4 Partial differential equations

The formulation in Sect. 2 can be adapted in a straightforward manner for partial differential equations. There are, however, some unique challenges these problems pose, largely stemming from the high dimensionality of the discretized systems. Optimization algorithms that rely on the inversion of the exact Hessian, such as the LM method used above, becomes inefficient. We instead employ either a nonlinear conjugate gradient approach, or semi-implicit gradient descent, as explained below.

Another challenge involves the computation of the log-determinant terms in (13). Direct factorization is impractical, since the Hessian and gradient matrices are not narrow-banded along the diagonal. Various strategies to approximate the log-determinant of large matrices have been proposed (Barry and Pace 1999; Boutsidis et al. 2017), generally involving Monte-Carlo estimates the trace of a series approximation of the matrix logarithm. This was tried

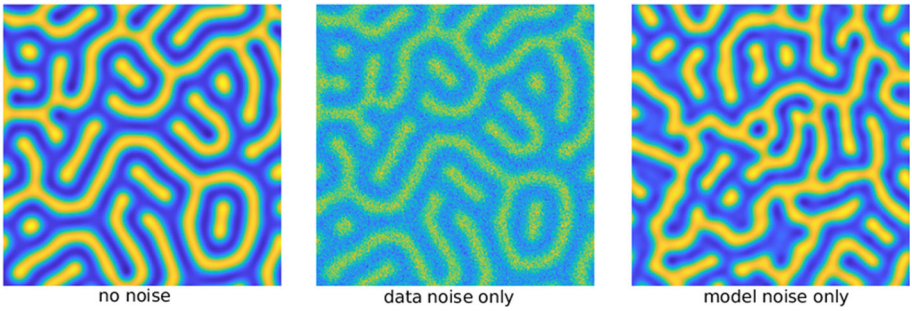

Fig. 8 Left: numerical solution to (24) without noise. Middle: the same state with data noise $n_d = 0.4$ added. Right: uses the same initial data as other plots, but noise was added to the model with $n_m = 0.2$. All simulations use the same initial condition

here, but the randomness introduced led to undesirable artifacts and poor performance. A more satisfactory approach was obtained by noting that most of the large eigenvalues of both $\nabla^2 F$ and ∇N arise from the differential operators. Approximating by excluding the non-differential terms results (in our examples) in constant coefficient operators. Spectral discretization of the linear operators provides analytic expressions for eigenvalues from which log determinants can be evaluated.

With this approximation, optimization in (13) and (10) can be done simultaneously, since the log determinant terms no longer depend explicitly on u^*. An equivalent problem is therefore to minimize

$$-\frac{\hat{n}}{2}\ln\lambda + \frac{1}{2}\ln|\det\nabla^2 F| + \frac{\hat{n}}{2}\ln F(u,\theta,\lambda) - \ln|\det\nabla N|. \tag{23}$$

over u,θ and λ at the same time.

4.1 Example: patterns in the Swift–Hohenberg equation

To illustrate the methodology in the context of partial differential equations, we consider the Swift–Hohenberg equation (Cross and Hohenberg Jul 1993)

$$u_t = -(\Delta + K^2)^2 u + \alpha u + \beta u^2 - \gamma u^3 \equiv N(u,\theta), \tag{24}$$

where the parameter set considered here is $\theta = (K, \alpha, \beta, \gamma)$. The spatial domain is a square $[0, 100]^2$, equipped with periodic boundary conditions. Our interest is in data representing spatial patterns, regarded as equilibria or near-equilibria of (24).

The spatial discretization of (24) uses a regular grid of size 256^2, and derivatives are evaluated pseudo-spectrally (Trefethen 2000). The function

$$F(u;\theta,\lambda) = \int_D ||N(u,\theta)||^2 + \lambda||u-\hat{u}||^2 dx.$$

is correspondingly evaluated using the trapezoid rule.

Synthetic data was obtained by simulating (24) using random initial conditions up to a specified large time $T = 10^4$. Normally distributed noise in the model and data was added pointwise to $N(u,\theta)$ and the final state $u(x,T)$, with variances n_m^2, n_d^2, respectively. Typical synthetic data is shown in Fig. 8.

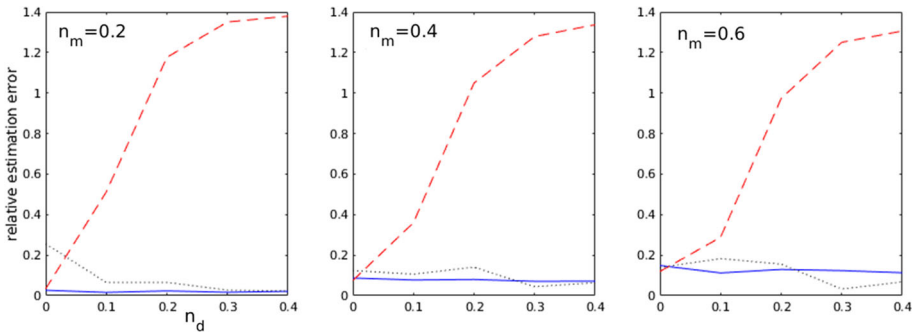

Fig. 9 Relative errors of parameter estimates for the Swift-Hohenberg equation, using nonlinear least squares (dotted, black), residual minimization (dashed, red), and the AMI estimate (solid, blue). The errors were averaged over 10 realizations of synthetic data for each combination of n_d and n_m

For log determinant computations, the Hessian of the system was approximated by the operator $\nabla^2 F \approx 2[-(\Delta + K^2)^4 + \lambda]$. Similarly, ∇N was approximated by $-(\Delta + K^2)^2$. Two different optimization methods were used: a semi-implicit operator splitting method (Glasner and Orizaga 2016) for the gradient decent, and the NCG method using the Polak–Ribière update formula. The preconditioner for the latter was given by the approximate Hessian operator, which is easily inverted spectrally. Gradient descent was used until convergence (measured as a reduction of the gradient norm) became poor; at this point the NCG method was used.

Numerical experiments were conducted for various combinations of data and model noise. As in the previous section, parameter estimates were obtained using approximate NLS and RM for comparison. The relative parameter errors were computed as before, and averaged over ten different randomly generated states.

The results of the experiments are shown in Fig. 9. Consistent with prior experiments, noise in the data leads to poor performance of RM estimation. In general, AMI estimates were best, although in a few circumstances NLS performed somewhat better. It is not clear if this can be ascribed to the approximations of the determinant terms.

4.2 Example: the Kuramoto–Sivashinsky equation and spatiotemporal data

Space and time dependence is now considered using the Kuramoto–Sivashinsky equation

$$u_t = auu_x + bu_{xx} + cu_{xxxx}. \tag{25}$$

Here, $u(x,t) : [0, L]\times[0, T] \to \mathbb{R}$ with $L = 40, T = 64$, and spatial boundary conditions are periodic. An initial condition is applied to generate synthetic data, but is otherwise determined automatically in the parameter estimation process, analogous to the method for ODEs.

In this model, a pseudo-spectral discretization is used for the spatial dependence on a regular grid with $N = 256$ points. The temporal discretization was the same as in Sect. 3, which is essentially the semi-implicit trapezoid rule with $M = 257$ points and timestep $\Delta t = T/(M-1)$. Writing $u_{i,j}$ for approximation at the ith spatial grid-point and the jth timestep, the corresponding optimization function F is given by

$$F(u; a, b, c, \lambda) = \sum_{j=0}^{M-1} \sum_{i=1}^{N} \left(\frac{u_{i,j+1} - u_{i,j}}{\Delta t} - K_{i,j}(\overline{u}) \right)^2$$

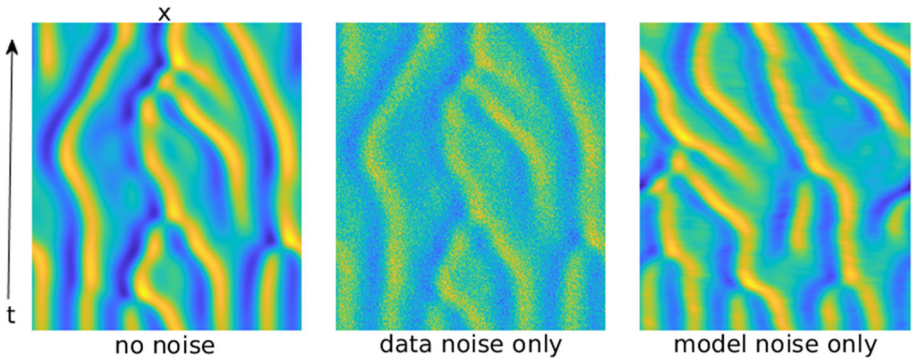

Fig. 10 Numerical solution to (25) without noise (left), with noise $n_d = 1$ added to the solution (middle) and noise $n_m = 1$ added to model (right)

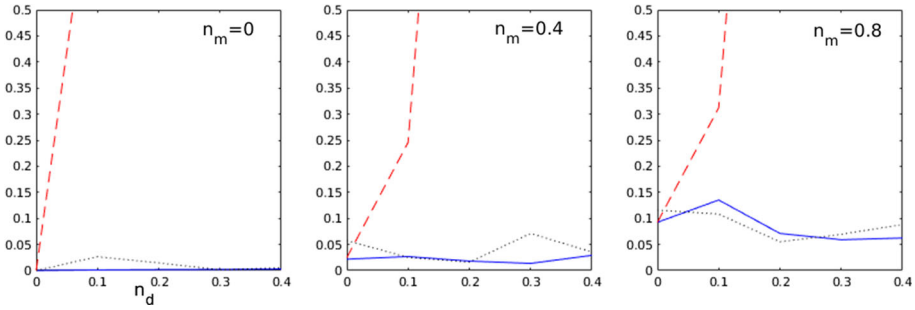

Fig. 11 Relative errors of parameter estimates for the Kuramoto-Sivashinsky equation, using nonlinear least squares (dotted, black), residual minimization (dashed, red), and the AMI estimate (solid, blue). The errors were averaged over 10 realizations of synthetic data for each combination of n_d and n_m

$$+\lambda \sum_{j=0}^{M} \sum_{i=1}^{N} (u_{i,j} - \hat{u}_{i,j})^2, \quad \overline{u}_{i,j} = \frac{u_{i,j} + u_{i,j+1}}{2},$$

where K is the spectrally discretized version of the right hand side in (25).

To general synthetic data, parameters $a = 1, b = -1, c = -1$ were used together with the initial condition

$$u(x, 0) = r_1 \sin(12\pi x/L) + r_2 \exp(-(x-2)^2/5);$$

where r_1 and r_2 are random and normally distributed. Noise in the model and data was introduced as in prior experiments, applying normally distributed fluctuations pointwise to K and $\hat{u}$, respectively. The resulting spatiotemporal patterns are shown in Fig. 10.

The Hessian was approximated by

$$\nabla^2 F \approx 2[\partial_{tt} + (b\partial_{xx} + c\partial_{xxxx})^2 + \lambda], \tag{26}$$

where derivatives correspond to their discrete counterparts, with periodic boundary conditions in x and Neumann boundary conditions in t. This operator is diagonalized by a Fourier transform in space and discrete cosine transform in time. The eigenvalues can therefore be

easily obtained as

$$\lambda_{i,j} = 2\Big(\omega_j + (-bk_i^2 + ck_i^4)\Big)^2 + \lambda$$

where k_i are the Fourier wavenumbers and ω_j are eigenvalues of the spectral discretization of ∂_{tt} (Trefethen 2000). An analogous computation was done for the ∇N term.

The hybrid gradient descent/ NCG optimization was used here as in the previous problem. In this case, the NCG method used the Fletcher-Reeves update formula, and the Hessian approximation (26) as a preconditioner.

Numerical experiments were conducted as before, making a comparison to NLS and RM estimates by averaging the estimation error over ten realizations of synthetic data for each combination of data and model noise. Figure 11 shows that the AMI method generally obtains better estimates, although these might be somewhat compromised by approximations made to the likelihood function. The RM estimates were generally very poor (and in many cases too high to appear in the figures).

5 Conclusion

We have demonstrated that uncertainties on both the data and underlying model must be taken into account for the accurate estimation of parameters. Our method could be viewed as an interpolation between NLS and RM regression, which individually may give poor estimates. Our procedure provides a principled approach to determining the interpolation parameter λ from the data alone.

The methodology can be readily adapted to data that is not evenly distributed in time or space. This can be done using several approaches. If the data is not sparse, it could be interpolated onto a regular grid, for example. Alternatively, the underlying numerical discretization can be designed to fit the structure of the data. Finally, the data can be treated as incomplete, which was shown to nonetheless produce reasonable estimates.

We have not attempted to quantify uncertainty in the estimates. This is often done in the framework of Bayesian inference (Dewar et al. 2010; Barajas-Solano and Tartakovsky 2019; Yoshinaga and Tokuda 2020; Zhao et al. 2021). This would require sampling techniques which are a challenging undertaking for systems with a large number of degrees of freedom. Our method might be a point or departure for this type of estimation, however.

Another natural extension of this approach is model discovery. This could be done in the context of sparse regression methods (Brunton et al. 2016; Schaeffer 2017), for example, by including terms in the objective function which penalize the number of model terms. Other extensions include accommodating multiple training data sets, which can be done in a straightforward manner in the probabilistic formulation. Beyond model calibration and system identification, there are numerous potential uses for the method developed here, including data restoration and analysis.

Acknowledgements The author was supported through NSF award DMS-1908968.

References

Ackleh Azmy S, Ferdinand Robert R, Simeon R (1998) Numerical studies of parameter estimation techniques for nonlinear evolution equations. Kybernetika 34(6):693–712

Ashyraliyev M, Jaeger J, Blom JG (2008) Parameter estimation and determinability analysis applied to drosophila gap gene circuits. BMC Syst Biol 2(1):83
Barajas-Solano David A, Tartakovsky Alexandre M (2019) Approximate Bayesian model inversion for pdes with heterogeneous and state-dependent coefficients. J Comput Phys 395:247–262
Barry RP, Pace RK (1999) Monte Carlo estimates of the log determinant of large sparse matrices. Linear Algebra Appl 289(1–3):41–54
Boutsidis C, Drineas P, Kambadur P, Kontopoulou E-M, Zouzias A (2017) A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. Linear Algebra Appl 533:95–117
Brunel NJB (2008) Parameter estimation of ode's via nonparametric estimators. Electron J Stat 2:1242–1267
Brunton Steven L, Proctor Joshua L, Kutz J Nathan (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci 113(15):3932–3937
Brynjarsdottir J, O'Hagan A (2014) Learning about physical parameters: the importance of model discrepancy. Inverse Probl 30(11):114007
Croft W, Elliott CM, Ladds Graham SB, Chandrasekhar V, Cathryn W (2015) Parameter identification problems in the modelling of cell motility. J Math Biol 71(2):399–436
Cross MC, Hohenberg PC (1993) Pattern formation outside of equilibrium. Rev Mod Phys 65(3):851
Dewar MA, Kadirkamanathan V, Opper M, Sanguinetti G (2010) Parameter estimation and inference for stochastic reaction-diffusion systems: application to morphogenesis in D melanogaster. BMC Syst Biol 4(1):21
Evensen G (2009) Data assimilation: the ensemble Kalman filter. Springer, New York
Fullana JM, Le Gal P, Rossi M, Zaleski S (1997) Identification of parameters in amplitude equations describing coupled wakes. Phys D Nonlinear Phenom 102(1–2):37–56
Garvie MR, Maini PK, Trenchea C (2010) An efficient and robust numerical algorithm for estimating parameters in turing systems. J Comput Phys 229(19):7058–7071
Glasner K (2021) Optimization algorithms for parameter identification in parabolic partial differential equations. Comput Appl Math 40(4):1–22
Glasner K, Orizaga S (2016) Improving the accuracy of convexity splitting methods for gradient flow equations. J Comput Phys 315:52–64
Gugushvili S, Klaassen CAJ (2012) $\sqrt{n}$-consistent parameter estimation for systems of ordinary differential equations: bypassing numerical integration via smoothing. Bernoulli 18(3):1061–1098
Hager William W, Hongchao Z (2006) A survey of nonlinear conjugate gradient methods. Pac J Optim 2(1):35–58
Jin B, Maass P (2012) Sparsity regularization for parameter identification problems. Inverse Probl 28(12):123001
Kennedy MC, O'Hagan A (2001) Bayesian calibration of computer models. J R Stat Soc Ser B Stat Methodol 63(3):425–464
Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
Liang H, Hulin W (2008) Parameter estimation for differential equation models using a framework of measurement error in regression models. J Am Stat Assoc 103(484):1570–1583
Long Z, Lu Y, Dong B (2019) Pde-net 2.0: learning pdes from data with a numeric-symbolic hybrid deep network. J Comput Phys 399:108925
Maddu S, Cheeseman BL, Sbalzarini IF, Müller CL (2019) Stability selection enables robust learning of partial differential equations from limited noisy data. arXiv preprint arXiv:1907.07810
Messenger Daniel A, Bortz David M (2021) Weak sindy for partial differential equations. J Comput Phys 443:110525
Plumlee M (2017) Bayesian calibration of inexact computer models. J Am Stat Assoc 112(519):1274–1285
Raissi M, Karniadakis GE (2018) Hidden physics models: machine learning of nonlinear partial differential equations. J Comput Phys 357:125–141
Raissi M, Perdikaris P, Karniadakis GE (2017) Machine learning of linear differential equations using gaussian processes. J Comput Phys 348:683–693
Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686-707
Ramsay J, Hooker G (2017) Dynamic data analysis. Springer, New York
Ramsay JO, Hooker G, Campbell D, Cao J (2007) Parameter estimation for differential equations: a generalized smoothing approach. J R Stat Soc Ser B Stat Methodol 69(5):741–796
Reinbold Patrick AK, Gurevich Daniel R, Grigoriev Roman O (2020) Using noisy or incomplete data to discover models of spatiotemporal dynamics. Phys Rev E 101(1):010203

Rudy Samuel H, Brunton Steven L, Proctor Joshua L, Nathan KJ (2017) Data-driven discovery of partial differential equations. Sci Adv 3(4):e1602614
Sargsyan K, Huan X, Najm HN (2019) Embedded model error representation for Bayesian model calibration. Int J Uncertain Quantif 9(4)
Schaeffer H (2017) Learning partial differential equations via data discovery and sparse optimization. Proc R Soc A Math Phys Eng Sci 473(2197):20160446
Schaeffer H, McCalla SG (2017) Sparse model selection via integral terms. Phys Rev E 96(2):023302
Sgura I, Lawless AS, Bozzini B (2019) Parameter estimation for a morphochemical reaction-diffusion model of electrochemical pattern formation. Inverse Probl Sci Eng 27(5):618–647
Trefethen LN (2000) Spectral methods in MATLAB, vol 10. SIAM, Philadelphia
Voss Henning U, Jens T, Jürgen K (2004) Nonlinear dynamical system identification from uncertain and indirect measurements. Int J Bifurcat Chaos 14(06):1905–1933
Wang Z, Huan X, Garikipati K (2019) Variational system identification of the partial differential equations governing the physics of pattern-formation: Inference under varying fidelity and noise. Comput Methods Appl Mech Eng 356:44–74
Xu H, Chang H, Zhang D (2019) Dl-pde: deep-learning based data-driven discovery of partial differential equations from discrete and noisy data. arXiv preprint arXiv:1908.04463
Yoshinaga N, Tokuda S (2020) Bayesian modelling of pattern formation from one snapshot of pattern. arXiv preprint arXiv:2006.06125
Zhao H, Storey BD, Braatz RD, Bazant MZ (2020) Learning the physics of pattern formation from images. Phys Rev Lett 124(6):060201
Zhao H, Braatz RD, Bazant MZ (2021) Image inversion and uncertainty quantification for constitutive laws of pattern formation. J Comput Phys 436:110279