GOODE: A Gaussian Off-The-Shelf Ordinary Differential Equation Solver

David N. John [1 2] **Vincent Heuveline** [2] **Michael Schober** [3]

Abstract

There are two types of ordinary differential equations (ODEs): initial value problems (IVPs) and boundary value problems (BVPs). While many probabilistic numerical methods for the solution of IVPs have been presented to-date, there exists no efficient probabilistic general-purpose solver for nonlinear BVPs. Our method based on iterated Gaussian process (GP) regression returns a GP posterior over the solution of nonlinear ODEs, which provides a meaningful error estimation via its predictive posterior standard deviation. Our solver is fast (typically of quadratic convergence rate) and the theory of convergence can be transferred from prior non-probabilistic work. Our method performs on par with standard codes for an established benchmark of test problems.

1. Introduction

The field of probabilistic numerics (Hennig et al., 2015) seeks numerical methods that can be interpreted as probabilistic inference. Numerical algorithms should return *probability distributions* as a measure of uncertainty associated with the inherit numerical approximations (Cockayne et al., 2016) which may improve existing algorithms (e.g., Balles et al. (2017); Mahsereci & Hennig (2015)), add novel functionality (e.g., Xi et al. (2018); Hauberg et al. (2015)), or improve operation safety (Oates et al., 2017). Good probabilistic analogues exist for quadrature (Briol et al., 2019), linear solvers (Bartels et al., 2018), and optimization (Balles & Hennig, 2018). For an extensive list see Oates & Sullivan (2019).

Ordinary differential equations (ODEs) appear as mathematical models for many problems arising in a broad range of application areas, e.g., in dynamical systems, and optimal control. Recently, ODEs have also been introduced as building blocks in machine learning algorithms (Chen et al., 2018; E et al., 2018; Grathwohl et al., 2019; Salman et al., 2018; Zhang et al., 2018). As most problems lack analytic solutions, no numerical toolbox is complete without a general-purpose numerical ODE solver.

While many probabilistic algorithms for the solution of initial value problems (IVPs) have been proposed (Tronarp et al., 2018; Teymur et al., 2018; Schober et al., 2019), only one method has been presented for the special case of linear boundary value problems (BVPs) (Cockayne et al., 2016). However, many interesting applications require the solution of nonlinear BVPs (Ascher et al., 1994; Sontag, 1998).

We present a probabilistic numerical algorithm for the solution of nonlinear two-point BVPs. As many problems can be transformed into this standard form (Ascher & Russell, 1981), this closes a significant gap in the probabilistic numerical toolbox. The method extends earlier work of (Cockayne et al., 2016) for linear problems by reformulating older work of Bellman & Kalaba (1965) in the context of Gaussian process (GP) regression (Rasmussen & Williams, 2006). The algorithm is based on the Newton-Raphson method (Deuflhard, 2011) and is of quadratic convergence given a good enough initial guess.

Our method treats the problem as a black box and only requires a standard interface identical to other state-of-the-art methods (Kierzenka & Shampine, 2001). Thus, the proposed algorithm can be considered an *off-the-shelf* numerical method for general BVPs. The standardized API enables a fair comparison on an established benchmark of test problems (Mazzia, 2014).

The structure of this work is the following: Section 2 introduces the considered BVPs and recaps briefly on multi-output GP regression. Section 3 explains a GP solver for linear BVPs. Section 4 builds on this and introduces the concept of quasilinearization in order to solve nonlinear BVPs, resulting in the proposed solver GOODE. Numerical experiments and comparison to other solvers are presented in Section 5. In Section 6, we put our work into perspective within the probabilistic numerics field. We conclude in Section 7.

[1]Corporate Research, Robert Bosch GmbH, Renningen, Germany [2]Engineering Mathematics and Computing Lab, Interdisciplinary Center for Scientific Computing, Heidelberg University, Germany [3]Bosch Center for Artificial Intelligence, Renningen, Germany. Correspondence to: David N. John <david.john@de.bosch.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

2. Problem Formulation and Background

We consider the *two-point boundary value problem (BVP)*

$$\begin{aligned} y' &= f(t, y(t)), \quad a \leq t \leq b, \\ 0 &= g(y(a), y(b)), \end{aligned} \tag{1}$$

with respect to the unknown function $y : [a, b] \to \mathbb{R}^d$. Differentiable $f : [a, b] \times \mathbb{R}^d \to \mathbb{R}^d$ and $g : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ are given and might be nonlinear. This is a first-order BVP. Without loss of generality, we can assume first-order systems (Ascher & Russell, 1981).

One way to solve (1) is by introducing unknown variables θ for the missing initial values $y(a) = \theta$. Standard software for the solution of IVPs can be used in an inner loop to find a θ^* that satisfies (1) in an outer loop. Methods of this type are called (multiple) shooting or marching methods. They are not commonly used nowadays as they require substantial algorithmic overhead (Ascher et al., 1994, §4) to compensate for numerical stability issues which are often encountered in interesting problems (Cash, 2004).

Global methods (Ascher et al., 1994, §5) consider a finite mesh of knots $\Delta := \{t_1 = a, t_2, \ldots, t_{N-1}, t_N = b\}$ and a suitable approximation class of functions $\mathcal{Y}$, e.g. the set of piecewise polynomials $\mathcal{P}_K$ of degree K. The goal is to find a $\hat{y} \in \mathcal{Y}$ such that the *discretized problem*

$$\begin{aligned} \hat{y}'(t_i) &= f(t_i, \hat{y}(t_i)), \quad \forall t_i \in \Delta, \\ 0 &= g(\hat{y}(a), \hat{y}(b)), \end{aligned} \tag{2}$$

is solved exactly by the approximation $\hat{y}$. Define the functional $F : \mathcal{Y} \to \mathbb{R}^{d(N+1)}$ as

$$F(\hat{y}) = \begin{pmatrix} \hat{y}'(t_1) - f(t_1, \hat{y}(t_1)) \\ \vdots \\ \hat{y}'(t_N) - f(t_N, \hat{y}(t_N)) \\ g(\hat{y}(t_1), \hat{y}(t_N)) \end{pmatrix} \tag{3}$$

It is clear that a root $F(\hat{y}^*) = 0$ of (3) is a solution to the discretized problem (2). Thus, the BVP is reduced to a root-finding problem and may be solved with any algorithm from the Newton-Raphson family (Deuflhard, 2011).

For a more thorough introduction, we refer the reader to Ascher et al. (1994) and Deuflhard & Bornemann (2002).

2.1. Gaussian Processes Review and Notation

There is extensive literature on GPs and GP regression, e.g. (Rasmussen & Williams, 2006), hence we briefly introduce our notation for multi-output GP regression here.

Let $y = f(t)$, for $t \in \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}^d$. Consider the multidimensional regression problem, where f is unknown and only a data matrix $D = [y_i]_{i=1,\ldots,N} \in \mathbb{R}^{d \times N}$, with $y_i = f(t_i)$, at the corresponding discrete mesh $\Delta = \{t_1, \ldots, t_N \,|\, t_n \in \mathbb{R}\}$, $t_m < t_n$ for $m < n$, is given. Denote by $vec(D) \in \mathbb{R}^{dN}$ the vectorization of matrix D. Further, let $A \otimes B = C$ denote the Kronecker product of $A \in \mathbb{R}^{n \times m}, B \in \mathbb{R}^{p \times q}$, then $C = [A_{ij}B]_{i=1,\ldots,n,j=1,\ldots,m} \in \mathbb{R}^{np \times mq}$. For a covariance kernel $k(t, t')$, define for two sets Δ, Δ' containing m and n elements, respectively, the $m \times n$ matrix $K_{\Delta\Delta'}$ with $(K_{\Delta\Delta'})_{i,j} = k(t_i, t'_j)$.

GP regression assumes a prior $P(f(t)) = \mathcal{GP}(f(t); m(t), k(t, t') \otimes V)$, with prior mean $m(t) \in \mathbb{R}^d$, a covariance kernel $k(t, t')$ and $V \in \mathbb{R}^{d \times d}$ positive semi-definite. For example, the trivial choice $V := I_d$, where $I_d \in \mathbb{R}^{d \times d}$ denotes the *identity matrix*. Given the data, the predictive posterior GP is $P(f(t) \,|\Delta, D) = \mathcal{GP}(f(t); \underline{\mu}_{\mathcal{D}}(t), \underline{k}_{\mathcal{D}}(t, t'))$, with

$$\begin{aligned} \underline{\mu}_{\mathcal{D}}(t) &= m(t) - (K_{t\Delta} \otimes V)G^{-1}vec(D), \\ \underline{k}_{\mathcal{D}}(t, t') &= K_{tt'} - (K_{t\Delta} \otimes V)G^{-1}(K_{\Delta t'} \otimes V), \end{aligned} \tag{4}$$

where $G := K_{\Delta\Delta} \otimes V$. Later, we present an example of GP regression in Figure 1. For details see Bonilla et al. (2007).

Note that GPs are closed under linear transformations, e.g. derivatives of GPs are again GPs, since differentiation is a linear operator (Bogachev, 1998). For a covariance kernel $k(t, t')$, define $k^{\partial}(t, t') = \frac{\partial}{\partial t'}k(t, t')$, similarly ${}^{\partial}k(t, t') = \frac{\partial}{\partial t}k(t, t')$ and ${}^{\partial}k^{\partial}(t, t') = \frac{\partial^2}{\partial t \partial t'}k(t, t')$. Then, for $d = 1$, provided the derivatives exist, $P(\frac{d}{dt}f(t)) = \mathcal{GP}(f(t); \frac{d}{dt}m(t), {}^{\partial}k^{\partial}(t, t'))$. Furthermore, $cov(f(t), \frac{d}{dt}f(t')) = k^{\partial}(t, t')$ and vice versa $cov(\frac{d}{dt}f(t), f(t')) = {}^{\partial}k(t, t')$. For a detailed introduction see Solak et al. (2003).

Later, we will mainly use the squared exponential kernel $k(t, t') = exp((2\lambda^2)^{-1}(t - t')^2)$ with characteristic length scale $\lambda > 0$. Other covariance functions are described in Rasmussen & Williams (2006).

3. Solving Linear BVPs with GPs

It has been shown how GPs can be used to solve linear BVPs (Cockayne et al., 2016). In the following, we will present the theory in our notation. Consider the linear BVP

$$\begin{aligned} y'(t) &= A(t)y(t) + q(t), \quad a \leq t \leq b, \\ \eta &= B_a y(a) + B_b y(b), \end{aligned} \tag{5}$$

where $A : [a, b] \to \mathbb{R}^{d \times d}$, $q : [a, b] \to \mathbb{R}^d$, $B_a, B_b \in \mathbb{R}^{d \times d}$ and $\eta \in \mathbb{R}^d$. Rewrite the previous equation and separate into linear operator, function and inhomogeneous part:

$$\begin{aligned} \left[\frac{\mathrm{d}}{\mathrm{d}t} - A(t)\right] y(t) &= q(t), \\ \begin{bmatrix} B_a & B_b \end{bmatrix} \begin{bmatrix} y(a) \\ y(b) \end{bmatrix} &= \eta. \end{aligned} \tag{6}$$

Since GPs are closed under linear transformations, (6) can be used to directly form a predictive posterior belief. Define a discretization mesh $\Delta = \{t_1, \dots, t_N \,|\, t_n \in [a,b]\}$, $t_m < t_n$ for $m < n$ and also $\partial I = \{a, b\}$. Further, $y(\Delta) \in \mathbb{R}^{d\times N}$ and $vec(y(\Delta)) := [y(t_1), \dots, y(t_N)] \in \mathbb{R}^{dN}$. Denote by $\mathbb{A}(\Delta) \in \mathbb{R}^{Nd\times Nd}$ the block diagonal matrix composed of $A(t_i)$ as the i-th block matrix, for all $t_i \in \Delta$. Define

$$
\begin{aligned}
Q &= \begin{bmatrix} \eta \\ vec(q(\Delta)) \end{bmatrix} \in \mathbb{R}^{d(1+N)}, \\
Y &= \begin{bmatrix} y(\partial I) & y(\Delta) & y'(\Delta) \end{bmatrix} \in \mathbb{R}^{d\times(2+2N)}, \\
H &= \begin{bmatrix} B_a & B_b & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbb{A}(\Delta) & I_{dN} \end{bmatrix} \in \mathbb{R}^{d(1+N)\times d(2+2N)}.
\end{aligned} \tag{7}
$$

Then $Hvec(Y) = Q$ represents the discretized version of the linear BVP in (6) as one large system of equations.

Using the previous definitions and assuming a multidimensional prior GP $P(y(t)) = \mathcal{GP}(y(t); 0, k(t,t') \otimes V)$ for $y(t) \in \mathbb{R}^d$, with positive semi-definite $V \in \mathbb{R}^{d\times d}$ (later $V := I_d$, the identity matrix), the *predictive posterior GP*, conditional on the data $\mathcal{D} = \{\mathbb{A}(\Delta), q(\Delta), B_a, B_b, \eta\}$ can be written compactly as

$$
P(y(t) \,|\, \mathcal{D}) = \mathcal{GP}(y(t); \underline{\mu}_{\mathcal{D}}(t), \underline{k}_{\mathcal{D}}(t,t')). \tag{8}
$$

With the definitions

$$
\begin{aligned}
G &= \begin{bmatrix} K_{\partial I\partial I} & K_{\partial I\Delta} & K^{\partial}_{\partial I\Delta} \\ K_{\Delta\partial I} & K_{\Delta\Delta} & K^{\partial}_{\Delta\Delta} \\ {}^{\partial}K_{\Delta\partial I} & {}^{\partial}K_{\Delta\Delta} & {}^{\partial}K^{\partial}_{\Delta\Delta} \end{bmatrix} \in \mathbb{R}^{(2+2N)\times(2+2N)}, \\
F &= \begin{bmatrix} K_{t\partial I} & K_{t\Delta} & K^{\partial}_{t\,\Delta} \end{bmatrix} \in \mathbb{R}^{1\times(2+2N)}, \\
F^{\dagger} &= \begin{bmatrix} K_{\partial It'} & K_{\Delta t'} & {}^{\partial}K_{\Delta t'} \end{bmatrix}^{\intercal} \in \mathbb{R}^{1\times(2+2N)}, \\
G_H &= H(G \otimes V)H^{\intercal} \in \mathbb{R}^{d(1+N)\times d(1+N)},
\end{aligned} \tag{9}
$$

predictive posterior mean and *covariance* are given as

$$
\begin{aligned}
\underline{\mu}_{\mathcal{D}}(t) &= (F \otimes V)H^{\intercal}G_H^{-1}Q \in \mathbb{R}^d, \\
\underline{k}_{\mathcal{D}}(t,t') &= (K_{tt'} \otimes V) - (F \otimes V)H^{\intercal}G_H^{-1}H(F^{\dagger} \otimes V).
\end{aligned} \tag{10}
$$

Define by $\sigma(t) := diag(\underline{k}_{\mathcal{D}}(t,t))^{1/2} \in \mathbb{R}^d$ the *posterior standard deviation*.

Note that for each $i \in \{1, \dots, d\}$, G is the covariance matrix of the dimension-wise rows, i.e. the i-th row, of Y. And $G \otimes V$ is the covariance matrix for $vec(Y) \in \mathbb{R}^{d(2+2N)}$.

Example 1: For a $\varepsilon > 0$, the ODE $\varepsilon z''(t) - z(t) = 0$, with $z(0) = 1, z(1) = 0$ is a linear BVP. The exact solution is $z_{ex}(t) = \frac{\exp(-t/\sqrt{\varepsilon}) - \exp((t-2)/\sqrt{\varepsilon})}{1-\exp(-2/\sqrt{\varepsilon})}$ (Mazzia, 2014). In standard form, for $y_1 = z, y_2 = z'$ and $y = [y_1, y_2]^{\intercal}$ it reads

$$
y' = \begin{bmatrix} 0 & 1 \\ \frac{1}{\varepsilon} & 0 \end{bmatrix} y, \quad \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} y(0) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} y(1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.
$$

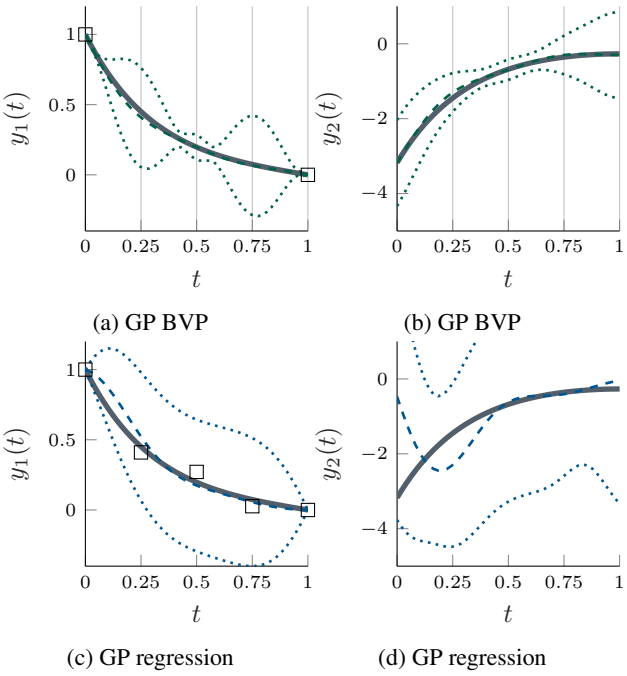


Figure 1. The first row shows the solution of Example 1 with a GP BVP solver. The second row shows standard GP regression results, if data points from the exact solution are given. Legend: $\underline{\mu}_{\mathcal{D}}(t)$ (dashed), $\underline{\mu}_{\mathcal{D}}(t) \pm 10\sigma(t)$ (dotted), exact $y(t)$ solid line, boundary conditions (square) and data points (square).

Figure 1 visualizes that a GP BVP solver is not just standard GP regression. For $\varepsilon = 0.1$, $\Delta = \{0, 0.25, 0.5, 0.75, 1\}$, an approximation of Example 1 obtained with the linear GP BVP solver in comparison to standard GP regression are displayed. Both results are obtained with a squared exponential kernel and a non-optimal length-scale for visualization purposes. For the BVP only $y(\partial I)$ (square marker in (a)) are given as direct observations of y. The remaining information about y and y' are indirectly given via *noise-free linear combinations* $y'(t_n) + Ay(t_n) = q(t_n)$ at $t_n \in \Delta$ (vertical grey lines in (a), (b)) with a Dirac likelihood, where only $q(t)$ and A are known. In standard GP regression, direct information about $y(t)$ is required. Here the data $y_1(\Delta) = z_{ex}(\Delta)$ (square marker in (c)) is used. Note that in the black-box setting, direct information is usually *not available* and needs to be replaced by noisy numerical approximations (here simulated via additive Gaussian noise). Based on this data, (c) displays the standard GP regression result for $y_1(t)$ and (d) shows the derivative of this GP for $y_2(t)$. Also note how H introduces a non-stationary dependence of the regression weights $(F \otimes V)H^{\intercal}G_H^{-1}$ and also on the posterior standard deviation $\sigma(t)$ in contrast to standard GP regression.

Remark: As BVPs *by definition* are multi-dimensional (or higher-order), a multi-output GP regression implementation is an absolute requirement for this application. Here, computational costs are largely dominated by inversion of

G_H which is $O((d+dN)^3)$. Unfortunately $A(t) \neq const$ in most cases, thus H cannot be reformulated as a Kronecker product and consequently G_H^{-1} can not be simplified to reduce the cost of the inversion, in contrast to Hennig & Hauberg (2014). In the few cases where $A(t) = const$, H can be written as some Khatri-Rao product, which is equivalent to the Kronecker product up to (unknown) permutation and selection matrices (Liu & Trenkler, 2008).

Remark: Our formulation is for first-order BVPs in standard form. In the few cases where this form is not possible, an analogous formulation for higher-order BVPs can be derived, if higher-order derivatives of $k(t,t')$ exist. Then set up $Y, H, G, F, F^\dagger$ not just for y, y', but also for all required higher derivatives of y.

3.1. Error Analysis for Linear BVPs

The error analysis for linear problems in Cockayne et al. (2016) is restated here in our notation. The following Proposition is similar to Prop. 4.1 in Cockayne et al. (2016).

Proposition 1 (local accuracy) Let $y(t)$ be the exact solution of a linear BVP. Then,

$$\forall t \in [a,b]: \quad \|\underline{\mu}_{\mathcal{D}}(t) - y(t)\| \leq \sigma(t)\|y\|. \tag{11}$$

Consequently, reducing $\sigma(t)$ improves estimation $\underline{\mu}_{\mathcal{D}}(t)$. An illustration of Proposition 1 follows later with Figure 4.

Let $h := \sup_{t\in[a,b]} \min_{t'\in\Delta}|t-t'|$ denote the fill distance of Δ. In case of an equidistant mesh $h = (b-a)/2N$. Proposition 4.2 in Cockayne et al. (2016) states that $\sigma(t) \leq Ch^p$, where $p = \beta - \rho - 1/2$ depends on the Sobolev space $\mathbb{H}^\beta$ that is norm-equivalent to the Reproducing Kernel Hilbert Space (RKHS) of the kernel $k(t,t')$. The parameter ρ corresponds to the order of the differential operator, in our case $\rho = 1$.

Theorem 4.4 in Cockayne et al. (2016) is a direct result of both stated propositions. We reformulate it for our purpose.

Theorem 2 Let $y(t)$ be the exact solution of a linear BVP. Then, $\|\underline{\mu}_{\mathcal{D}}(t) - y(t)\|_{L^2}^2 \leq C(b-a)h^{2p}$ with p as above and $C > 0$.

4. Solving Nonlinear BVPs by Quasilinearization

Quasilinearization is a direct application of Newton's method to nonlinear BVPs (1) in order to obtain a series of linear continuous BVPs (6). The original idea is due to Bellman & Kalaba (1965) and a conceptual introduction can be found in Ascher et al. (1994). The following derivation is similar to Mazzia & Sgura (2002).

Define $\mathcal{F}(y) = y' - f(t,y)$ and $\mathcal{G}(y) = g(y(a), y(b)) - \eta$. In the following, we will write $J_f(t,y) = \frac{\partial}{\partial y} f(t,y)$. The Fréchet derivative $\mathcal{F}'$ of $\mathcal{F}$, for a $s \in C^1([a,b], \mathbb{R}^d)$, is given by (Ascher et al., 1994, §2.3.4)

$$\mathcal{F}'(y)s = \left(\frac{\mathrm{d}}{\mathrm{d}\,t} - J_f(t,y)\right) s. \tag{12}$$

Now, linearization of the BVP $\mathcal{F}(y) = 0, \mathcal{G}(y) = 0$ around a given $y^{(k)} \in L^2([a,b])$ with respect to the Fréchet derivative yields Newton's method in function space

$$\begin{aligned} \mathcal{F}'(y^{(k)})s^{(k)} &= -\mathcal{F}(y^{(k)}), \\ y^{(k+1)} &= y^{(k)} + \nu\, s^{(k)}, \end{aligned} \tag{13}$$

with $\nu > 0$. For $\nu = 1$, it can be shown that (13) is equivalent to

$$\left[\frac{\mathrm{d}}{\mathrm{d}t} - J_f(t, y^{(k)})\right] y^{(k+1)} = f(t, y^{(k)}) - J_f(t, y^{(k)}) y^{(k)} \tag{14}$$

which is, for a given initial guess $y^{(0)} \in L^2([a,b])$, a series of linear BVPs with the boundary conditions

$$\begin{aligned} &B_a^{(k)} y_a^{(k+1)} + B_b^{(k)} y_b^{(k+1)} = \eta^{(k)}, \\ &\eta^{(k)} = B_a^{(k)} y_a^{(k)} + B_b^{(k)} y_b^{(k)} - g(y_a^{(k)}, y_b^{(k)}) + \eta, \\ &B_a^{(k)} = \frac{\partial}{\partial y_a} g(y_a^{(k)}, y_b^{(k)}),\ B_b^{(k)} = \frac{\partial}{\partial y_b} g(y_a^{(k)}, y_b^{(k)}). \end{aligned} \tag{15}$$

Here, for brevity $y_a^{(\cdot)} := y^{(\cdot)}(a)$ and $y_b^{(\cdot)} := y^{(\cdot)}(b)$.

4.1. GOODE

Our contribution is the combination of quasilinearization and the presented GP solver for linear BVPs, (8) and (10), to iteratively approximate nonlinear BVPs in a probabilistic fashion. In each iteration the GP solver is used to approximate a linear BVP out of the series (14) and (15), until a stopping criterion is reached. This yields a probabilistic solver providing a probability distribution as solution, naturally delivering uncertainty estimates, which is a novel and unique functionality in comparison to classical nonlinear BVP solvers. We have implemented our method in Matlab with an almost identical interface as the Matlab BVP solvers (Kierzenka & Shampine, 2001; 2008). Our goal is that users should be able to apply our method without needing to understand the exact inner working, treating the algorithm as a black-box off-the-shelf numerical method. We denote this method by GOODE: a *Gaussian Off-the-shelf Ordinary Differential Equation solver*[1].

Computational costs are largely dominated by inversion of G_H in each iteration which is $O((d+dN)^3)$. This is akin to classical (global) methods (Ascher et al., 1994, §7). Using efficient solvers, it can be hoped to achieve a speed-up via the application of Krylov subspace methods (de Roos & Hennig, 2017).

[1]Matlab code is available at https://github.com/boschresearch/GOODE

4.2. Error Analysis for Nonlinear BVPs

In each iteration the linear BVPs (14) and (15) are only approximated. Due to those *uncertain Newton steps* the method is an inexact Newton method in function space and standard convergence results apply.

Define the residual $r^{(k)} := \mathcal{F}'(y^{(k)})s^{(k)} + \mathcal{F}(y^{(k)})$ and the relative residual $\rho^{(k)} := \|r^{(k)}\|_{L^2}/\|\mathcal{F}(y^{(k)})\|_{L^2}$. If the condition $\rho^{(k)} \leq \nu^{(k)}$ holds, where the forcing factors $\nu^{(k)} = \min(0.5, C\|\mathcal{F}(y^{(k)})\|_{L^2}) < 1$, for some constant $C > 0$, then the inexact Newton method converges locally quadratic (Dembo et al., 1982; Dean, 1992; Mazzia & Sgura, 2002). Consequently, it is not necessary to be exact in each iteration. The accuracy and the associated computational costs of the approximations in each iteration could be controlled to satisfy only the upper bound of the condition. For our solver this could be used in future work.

For a given tolerance $tol > 0$, $\rho^{(k)} < tol$ or $\|s^{(k)}\|_{L^2} < tol$ can be used as stopping criterion. This is further discussed in Mazzia & Sgura (2002).

By combining this with Section 3.1 one obtains for nonlinear BVPs following: In each iteration the linear BVPs (14) need to be solved such that the local convergence condition on $\rho^{(k)}$ is satisfied. Then GOODE, i.e. the respective inexact Newton method, converges locally quadratic. One strategy could be to iteratively approximate the linear problems and, if the condition on the relative residual is not satisfied, then refine the mesh Δ and repeat. Currently, we only use an equidistant mesh Δ, thus refinement is achieved by increasing N. However, in future work one could benefit largely from Proposition 1; i.e. the posterior standard deviation $\sigma(t)$, as a local accuracy estimate, could be used to refine the grid locally.

In the next section Figure 3 illustrates the convergence of GOODE with respect to N and the Newton iterations.

4.3. Illustration

Example 2: For a $\varepsilon > 0$, the ODE $\varepsilon z''(t) - z'(t)^2 = 1$, with $z(0) = z_{ex}(0)$, $z(1) = z_{ex}(1)$ is a nonlinear BVP. The exact solution is $z_{ex}(t) = 1 + \varepsilon \ln \cosh((t - 0.745)/\varepsilon)$. In standard form, for $y_1 = z, y_2 = z'$, $y = [y_1, y_2]^\intercal$ and $f(y_2) = \frac{1}{\varepsilon}(1 - (y_2)^2)$, it reads

$$y' = \begin{bmatrix} y_2 \\ f(y_2) \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} y(0) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} y(1) = \begin{bmatrix} z_{ex}(0) \\ z_{ex}(1) \end{bmatrix}.$$

Figure 2 displays the iterative approximation of Example 2 ($\varepsilon = 0.1$) with GOODE in comparison to the exact solution, for an equidistant mesh Δ with $N = 31$ points, initialized with $y^{(0)} \equiv 0$. A squared exponential kernel, with length scale optimization on a fine grid with respect to the error relative to the exact solution, was used. Due to the non-informative initialization $y^{(0)} \equiv 0$ iteration $k = 1$ is far away from the exact solution, however $y_1^{(1)}$ already fulfills the boundary conditions, as all next iterations do. Following iterations $k+1$ are initialized by $y^{(k)}$ and converge fast to the exact solution until the stopping criterion, $\|s^{(k)}\|_{L^2} < tol$ for $tol = 10^{-6}$, is reached. A relative error of 10^{-4} with respect to the L2 norm is reached for iteration $k = 7$ as is displayed in Figure 3.

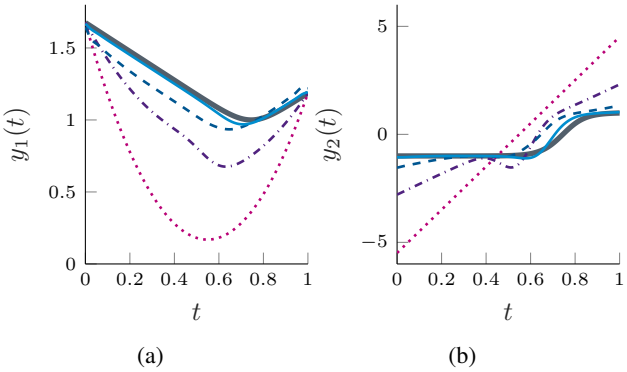


Figure 2. Iterative approximation of nonlinear Example 2 with GOODE. (a) displays $y_1(t)$ and (b) $y_2(t)$, with their respective approximating Newton iterations. Legend: Iteration $k = 1$ dotted, $k = 2$ dash-dotted, $k = 3$ dashed, $k = 4$ solid and exact solution solid thick (gray) line.

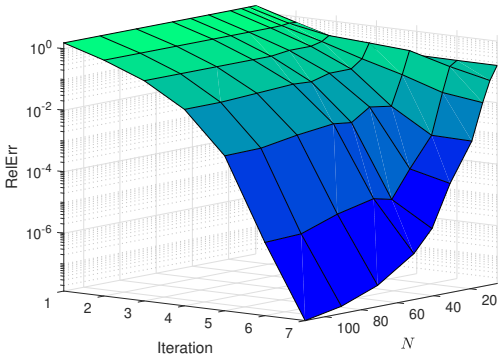
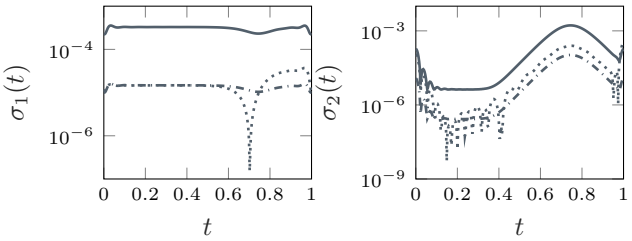


Figure 3. Relative error of the Newton iterations $k = 1, \ldots, 7$ for varying number of points N for Example 2.

Note that the results can be improved for a larger N. This can be done with the strategies presented earlier. In Figure 3 the evolution of the relative error (for Example 2 and with respect to the reference solution) of the Newton iterations $k = 1, \ldots, 7$ for increasing number N of equidistant points in Δ is displayed. Convergence of the Newton method improves largely for increasing N, as the series of linear BVPs approximating the nonlinear BVP are solved with higher accuracy.

Figure 4 illustrates that the result of Proposition 1 also

applies to the converged Newton iteration $k = 7$ approximating the nonlinear Example 2. The local accuracy $\|\underline{\mu}_{\mathcal{D}}(t) - y(t)\|$ and its upper bound $\sigma(t)\|y\|_{L^2}$ are plotted. At this point we emphasize that $\underline{\mu}_{\mathcal{D}}(t)$ and $\sigma(t)$ are the approximation of the linear BVP in iteration $k = 7$, but $y(t)$ is the analytical solution of the nonlinear BVP. Also $\sigma(t)$ is plotted to show that it has the same magnitude as $\|\underline{\mu}_{\mathcal{D}}(t) - y(t)\|$. We note that the overall qualitative trend has high similarity, thus confirming (at least for Example 2) that $\sigma(t)$ is valuable additional information delivered as add-on with GP solvers. It also confirms that $\sigma(t)$ can be a useful tool for local mesh refinement. Sticking to this example, mesh refinement around $t_r \approx 0.7$, due to the peak at $\sigma_2(t_r)$ (b), might be advisable.



(a) (b)

Figure 4. Illustration of Proposition 1, with nonlinear Example 2. For iteration $k = 7$, the local accuracy $\|\underline{\mu}_{\mathcal{D}}(t) - y(t)\|$ (dotted), $\sigma(t)\|y\|_{L^2}$ (solid) and $\sigma(t)$ (dash-dotted) are presented for dimensions 1 in (a) and 2 in (b).

4.4. A Bayesian Probabilistic Numerical Method?

Recently, Cockayne et al. (2017) have given a rigorous definition of what it means for a probabilistic numerical method to also be a *Bayesian* probabilistic numerical method. Wang et al. (2018) present a sufficient condition on the existence of a certain Lie group of transformations for this to be the case. While numerical algorithms have been developed to detect linearity (Birkisson & Driscoll, 2013), it is not clear how the more general condition can be exploited algorithmically.

For the general nonlinear case, the existence of a suitable Lie group cannot be expected. According to the definition in Cockayne et al. (2017), our method is *not* a Bayesian PNM. However, we suggest that GOODE is thought of as a Laplace approximation to the true posterior density. Using different initial guesses, this can be used to construct a GP mixture model for ODEs with multiple solutions. This view is explored empirically in Sect. 5.1.

4.5. Model Selection

First of all, a covariance kernel $k(t, t')$, with existing partial derivatives as required needs to be selected. Examples of possible kernels are squared exponential, Matérn $5/2$, Matérn $3/2$, rational quadratic (Rasmussen & Williams, 2006) and cubic spline kernels (Minka, 2000).

Based on the choice of the kernel $k(t, t')$ the kernel hyperparameters $\theta \in \mathbb{R}^K$ need to be set and optimized; e.g. for the squared exponential kernel $\theta = \lambda > 0$. Standard type-II log marginal likelihood (Rasmussen & Williams, 2006)

$$\begin{aligned} logP(y \,|\, \mathcal{D}, \theta) = &-\frac{1}{2} Q^\intercal G_H^{-1} Q - \frac{1}{2}\log(\det(G_H)) \\ &- \frac{1}{2}(d + dN)\log(2\pi), \end{aligned} \tag{16}$$

with the definitions above, is maximized via grid search on a fine grid $\Delta_{t'}\theta$. Gradient based optimization would either require differentiation of the kernel and its partial derivatives with respect to θ or corresponding approximations. However, as computation time is not a bottleneck for the problems considered in this work, grid search will be sufficient.

Remark: In the experiments later on we observe that, for the squared exponential kernel, in some problems the accuracy of the method for a given N strongly depends on the choice of λ. However, in some of those cases the maximum of the log marginal likelihood only corresponds to the best hyperparameter (obtained with respect to the reference), if the number of mesh points N is large enough.

It is known that using a universal kernel (Micchelli et al., 2006) and a fine enough mesh, any curve can be fitted. In this context, sub-optimal hyper-parameters might require an exponentially larger N (van der Vaart & van Zanten, 2011), nevertheless universality still holds.

5. Experiments

To benchmark our GP BVP solver GOODE, a comparison to established, non-probabilistic BVP solvers TOM, `bvptwp` and Matlab's `bvp4c` and `bvp5c` on an established set of BVP problems is carried out.

The code TOM is described in Mazzia & Sgura (2002) and `bvptwp` in Cash et al. (2013). `bvp4c` is described in Kierzenka & Shampine (2001) and `bvp5c` in Kierzenka & Shampine (2008). These methods are considered state-of-the-art (Cash & Mazzia, 2011). GOODE is implemented in Matlab with standard interface.

The testset can be obtained from Mazzia (2014). A short description is given in Mazzia & Cash (2014). It contains 33 two-point BVPs, which are in parts singularly perturbed problems, depending on an $\varepsilon > 0$; I.e. small ε provides a multi-scale character to the problem.

Problems 1-18 are linear and provide, except for no. 15, an exact solution. Problems 19-33 are nonlinear, with exact solutions only for 20 and 21. Most problems have dimension $d = 2$, problems 31 and 32 $d = 4$ and problem 33 $d = 6$.

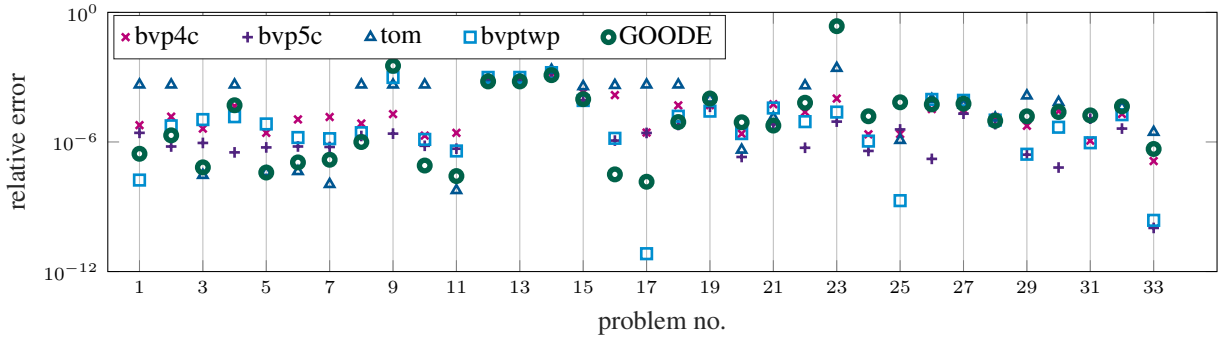


Figure 5. Relative errors of the approximations obtained by several BVP solver for the test set.

If not stated otherwise, we will use the following default setting to obtain the results: squared exponential kernel, equidistant mesh $\Delta \in \mathbb{R}^N$ including the boundary points, with $N = 31$, grid search for $\lambda \in [1.5h, 15h]$ with $M = 40$ logarithmic spaced grid points and $\varepsilon = 0.1$ for all the problems. For the problems without exact solution, we use an approximate reference solution obtained with `bvp5c` and relative tolerance of 1e-9 and absolute tolerance of 1e-12 (default values are 1e-3 and 1e-6, respectively.).

A comparison of GOODE to the other solvers, with respect to the relative error, is displayed in Figure 5. Overall, the performance of GOODE is comparable, with the advantage of a local error estimate $\sigma(t)$. Note that GOODE, currently with the disadvantage of the equidistant mesh, needs slightly more points to obtain those results. At this point we refer to Section 4.5 and mention again that for small, fixed N, hyperparameter optimization is essential, but fast. Runtime and mesh comparison can be found in the supplementary materials. For Problem 23 GOODE performs significantly worse. This is due to the equidistant mesh Δ and small N. A much larger N or an adaptive mesh would improve results, as all other solvers have adaptive mesh selection and locally refine at the right boundary of Problem 23, where y_1 changes rapidly. Tests with non-equidistant mesh and also with Matérn 5/2 kernel can be found in the supplements.

Figure 6 presents the difference of selecting the length scale as the global optimum based on the reference or based on maximizing the log likelihood (16). For some problems it is similar, but for others, e.g. problems 2, 4, 18, and 21, there is a significant difference. However, this can be improved by increasing the number of points N as presented in Figure 7. Here, for problems 1, 2, 20, and 21, the fast convergence of GOODE is displayed. It also shows that for N large enough the difference between best possible error and the error of log likelihood model selection vanishes. Note that both axis are in logarithmic scale.

5.1. Painlevé ODE

Nonlinear BVPs can have multiple solutions and the resulting probability distribution should thus be multi-modal. We illustrate this with the nonlinear Painlevé ODE $z(t)'' = z(t)^2 - t$ with boundary conditions $z(0) = 0$ and $z(10) = \sqrt{10}$, see Cockayne et al. (2017, §6.2). Let $y := [z, z']^\intercal$ to obtain the standard form. This problem has two solutions. To obtain both solutions the initialization $y^{(0)}$ needs to be adjusted. For one solution we use $y^{(0)} \equiv 0$ and for the other $y_1^{(0)}$ linear between $y_1^{(0)}(0) = -3$ and $y_1^{(0)}(10) = 3$, and $y_2^{(0)} \equiv 0$. For this setting GOODE finds both solutions, visualized in Figure 8 by the solid (light green lines). `bvp5c` only finds the first solution (with $N = 65$ and `bvp5c` error estimate $4.8e-5$) and fails to converge to the second solution (with $N = 1400$ and `bvp5c` error estimate 0.65).

6. Related Work in Probabilistic Numerics

Applications of kernel methods (Wendland, 2004) to the classical theory for BVPs (Ascher et al., 1994; Deuflhard & Bornemann, 2002) is established (Saitoh & Sawano, 2016, §5). Through the known equivalences of spline methods and GP regression (Kanagawa et al., 2018; Kimeldorf & Wahba, 1970), many theoretical results are readily obtainable.

In Brugnano & Trigiante (1998), methods for initial value problems (IVPs) are developed that focus on the application of BVP methods to IVPs. This treatment can circumvent in some cases unfavorable stability properties of shooting-based spline methods for IVPs such as Loscalzo (1969); Byrne & Chi (1972). It is an open question, how these results can be transfered to other probabilistic IVP solvers in the literature.

Probabilistic IVP solvers, e.g., Teymur et al. (2018); Tronarp et al. (2018); Schober et al. (2019) could be applied in (multiple) shooting schemes (Ascher et al., 1994, §4). However, it is not clear whether these methods are sufficiently stable to warrant efficient integration of BVPs.

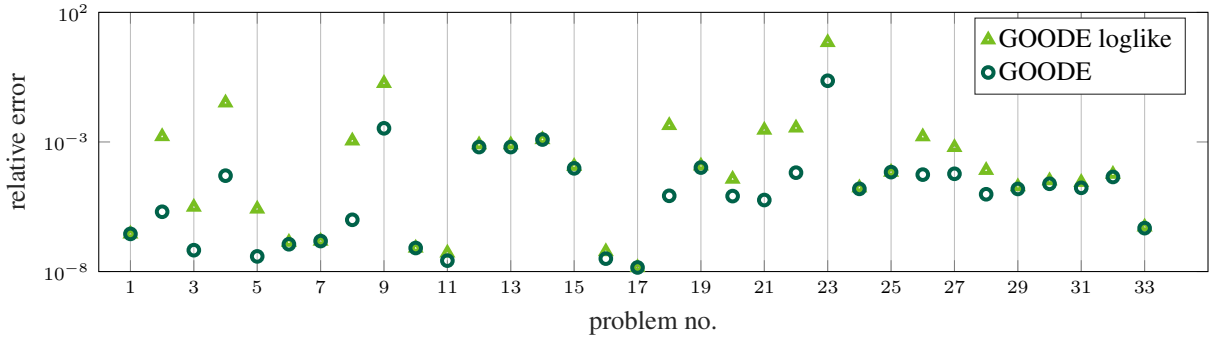
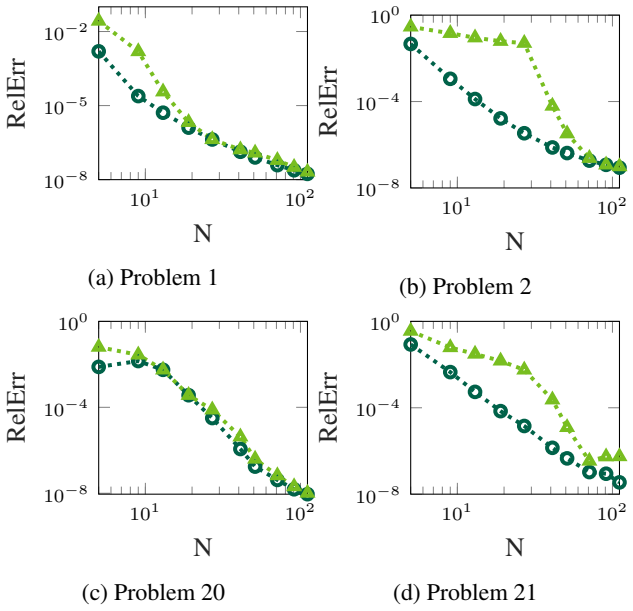


Figure 6. Comparison of global optimum (with respect to the reference) vs. log likelihood optimum, each computed on a fine grid for λ.



(a) Problem 1 (b) Problem 2

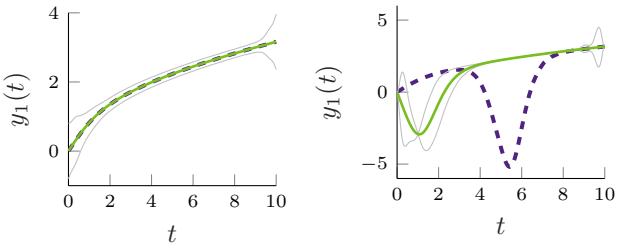
(c) Problem 20 (d) Problem 21

Figure 7. Convergence for Problems 1, 2, 20 and 21. Dark green circles display the best relative error, obtained on a fine grid, with respect to the reference. Light green triangles display the relative error obtained by using the length-scale from log likelihood maximization.

In the context of Riemannian manifolds, solvers for specialized BVPs have been presented in Arvanitidis et al. (2019); Hennig & Hauberg (2014).

7. Conclusion

We have presented GOODE, a probabilistic numerical algorithm for the solution of nonlinear two-point BVPs. The solver iteratively applies the Bayesian probabilistic numerical method of Cockayne et al. (2016) in the quasilinearization process of Bellman & Kalaba (1965), which results in a GP posterior distribution over the numerical approximation. In this work, we have favored a more expository presentation over classical convergence and stability results. Those are readily obtainable by transferring the vast literature on smoothing splines (Wendland, 2004) and other boundary value methods (Ascher et al., 1994) via the known equivalence with GPs (Kanagawa et al., 2018).



(a) Solution 1 (b) Solution 2

Figure 8. Two solutions of the Painlevé BVP. Solid (light green) lines show results obtained with GOODE (λ from log likelihood maximization) $\pm C\sigma_1(t)$ as dotted lines, for $C = 8e4$ in (a) and $C = 4e3$ in (b). Dashed (magenta) lines show `bvp5c` results.

As most ODE problems can be reformulated in this first-order form (Ascher & Russell, 1981), our method fills a gap in the ever-growing toolbox of probabilistic numerical methods. Results on a standard benchmark show that the algorithm performs on par with state-of-the-art codes. This performance is already achieved with the proof-of-concept version detailed above. In particular, further improvements may be expected through the addition of advanced functionality such as automatic mesh selection and step size adaptation.

Although the algorithm returns a probability distribution in its current form, it may be criticized that the iterative process is not phrased as probabilistic inference. Thus, future work might focus on a fully probabilistic method, e.g., by applying a general linear solver (Bartels et al., 2018) and by probabilistic hyperparameter selection, e.g., for the evaluation mesh (Oates et al., 2019; Chkrebtii & Campbell, In review; Chaloner & Verdinelli, 1995).

Acknowledgements

We would like to thank Simon Bartels, Peter Baumann and Hans Kersting for helpful discussions and proofreading.

References

Arvanitidis, G., Hauberg, S., Hennig, P., and Schober, M. Fast and robust shortest paths on manifolds learned from data. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proc. of the 22nd Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, pp. 1506–1515. PMLR, 2019.

Ascher, U. and Russell, R. D. Reformulation of boundary value problems into "standard" form. *SIAM Review*, 23 (2):238–254, 1981.

Ascher, U. M., Mattheij, R. M., and Russell, R. D. *Numerical solution of boundary value problems for ordinary differential equations*. Siam, 1994.

Balles, L. and Hennig, P. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, pp. 404–413. PMLR, 2018.

Balles, L., Romero, J., and Hennig, P. Coupling adaptive batch sizes with learning rates. In *Conference on Uncertainty in Artificial Intelligence (UAI) 2017*. AUAI Press, 2017.

Bartels, S., Cockayne, J., Ipsen, I. C. F., and Hennig, P. Probabilistic Linear Solvers: A Unifying View. *arXiv e-prints*, art. arXiv:1810.03398, Oct 2018.

Bellman, R. E. and Kalaba, R. E. *Quasilinearization and nonlinear boundary-value problems*. Rand Corporation, 1965.

Birkisson, A. and Driscoll, T. A. Automatic linearity detection. Technical report, SICS, 2013.

Bogachev, V. I. *Gaussian measures*. American Mathematical Soc., 1998.

Bonilla, E. V., Chai, K. M., and Williams, C. Multi-task gaussian process prediction. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T. (eds.), *Advances in Neural Information Processing Systems 20*, pp. 153–160. Curran Associates, Inc., 2007.

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., and Sejdinovic, D. Probabilistic integration: A role in statistical computation? *Statist. Sci.*, 34(1):1–22, 02 2019.

Brugnano, L. and Trigiante, D. *Solving Differential Equations by Multistep Initial and Boundary Value Methods*. CRC Press, 1998.

Byrne, G. D. and Chi, D. N. H. Linear multistep formulas based on g-splines. *SIAM Journal on Numerical Analysis*, 9(2):316–324, 1972.

Cash, J. R. A survey of some global methods for solving two-point bvps. *Applied Numerical Analysis & Computational Mathematics*, 1(1):7–17, 2004.

Cash, J. R. and Mazzia, F. Efficient global methods for the numerical solution of nonlinear systems of two point boundary value problems. In Simos, T. E. (ed.), *Recent Advances in Computational and Applied Mathematics*, pp. 23–39. Springer, 2011.

Cash, J. R., Hollevoet, D., Mazzia, F., and Nagy, A. M. Algorithm 927: The matlab code bvptwp.m for the numerical solution of two point boundary value problems. *ACM Trans. Math. Softw.*, 39(2):15:1–15:12, February 2013.

Chaloner, K. and Verdinelli, I. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 6572–6583. Curran Associates, Inc., 2018.

Chkrebtii, O. A. and Campbell, D. A. Adaptive step-size selection for state-space based probabilistic differential equation solvers. In review.

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. Probabilistic Numerical Methods for Partial Differential Equations and Bayesian Inverse Problems. *arXiv e-prints*, art. arXiv:1605.07811, May 2016.

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. Bayesian Probabilistic Numerical Methods. *arXiv e-prints*, art. arXiv:1702.03673, Feb 2017.

de Roos, F. and Hennig, P. Krylov Subspace Recycling for Fast Iterative Least-Squares in Machine Learning. *arXiv e-prints*, art. arXiv:1706.00241, June 2017.

Dean, E. An inexact newton method for nonlinear two-point boundary-value problems. *Journal of optimization theory and applications*, 75(3):471–486, 1992.

Dembo, R. S., Eisenstat, S. C., and Steihaug, T. Inexact newton methods. *SIAM Journal on Numerical analysis*, 19(2):400–408, 1982.

Deuflhard, P. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, volume 35. Springer Science & Business Media, 2011.

Deuflhard, P. and Bornemann, F. *Scientific Computing with Ordinary Differential Equations*. Springer, 2002.

E, W., Han, J., and Li, Q. A mean-field optimal control formulation of deep learning. *Research in the Mathematical Sciences*, 6(1):10, Dec 2018.

Grathwohl, W., Chen, R. T. Q., Bettencourt, J., and Duvenaud, D. Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*, 2019.

Hauberg, S., Schober, M., Liptrot, M., Hennig, P., and Feragen, A. A random riemannian metric for probabilistic shortest-path tractography. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, volume 18. Springer, September 2015.

Hennig, P. and Hauberg, S. Probabilistic Solutions to Differential Equations and their Application to Riemannian Statistics. In *Proc. of the 17th int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, volume 33. JMLR, W&CP, 2014.

Hennig, P., Osborne, M. A., and Girolami, M. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2179), 2015.

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. K. Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences. *arXiv e-prints*, art. arXiv:1807.02582, July 2018.

Kierzenka, J. and Shampine, L. F. A bvp solver based on residual control and the maltab pse. *ACM Trans. Math. Softw.*, 27(3):299–316, September 2001.

Kierzenka, J. and Shampine, L. F. A bvp solver that controls residual and error. *JNAIAM J. Numer. Anal. Ind. Appl. Math*, 3(1-2):27–41, 2008.

Kimeldorf, G. S. and Wahba, G. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41 (2):495–502, 1970.

Liu, S. and Trenkler, G. Hadamard, khatri-rao, kronecker and other matrix products. *Int. J. Inf. Syst. Sci*, 4(1): 160–177, 2008.

Loscalzo, F. R. An introduction to the application of spline functions to initial value problems. In *Theory and Applications of spline functions*, pp. 37–64. Academic Press New York, 1969.

Mahsereci, M. and Hennig, P. Probabilistic line searches for stochastic optimization. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 181–189. Curran Associates, Inc., 2015.

Mazzia, F. Test set for boundary value problem solvers, release 0.5. https://archimede.dm.uniba.it/~bvpsolvers/, July 2014. Department of Mathematics, University of Bari and INdAM, Research Unit of Bari.

Mazzia, F. and Cash, J. R. A Fortran Test Set for Boundary Value Problem Solvers. In *Proceedings of the International Conference on Numerical Analysis and Applied Mathematics (ICNAAM)*, 2014.

Mazzia, F. and Sgura, I. Numerical approximation of nonlinear bvps by means of bvms. *Applied Numerical Mathematics*, 42(1-3):337–352, 2002.

Micchelli, C. A., Xu, Y., and Zhang, H. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.

Minka, T. Deriving quadrature rules from Gaussian processes. Technical report, Statistics Department, Carnegie Mellon University, 2000.

Oates, C. J. and Sullivan, T. J. A Modern Retrospective on Probabilistic Numerics. *arXiv e-prints*, art. arXiv:1901.04457, January 2019.

Oates, C. J., Cockayne, J., Aykroyd, R. G., and Girolami, M. Bayesian Probabilistic Numerical Methods in Time-Dependent State Estimation for Industrial Hydrocyclone Equipment. *arXiv e-prints*, art. arXiv:1707.06107, Jul 2017.

Oates, C. J., Cockayne, J., Prangle, D., Sullivan, T. J., and Girolami, M. Optimality Criteria for Probabilistic Numerical Methods. *arXiv e-prints*, art. arXiv:1901.04326, January 2019.

Rasmussen, C. and Williams, C. *Gaussian Processes for Machine Learning*. MIT, 2006.

Saitoh, S. and Sawano, Y. *Theory of reproducing kernels and applications*. Springer, 2016.

Salman, H., Yadollahpour, P., Fletcher, T., and Batmanghelich, K. Deep Diffeomorphic Normalizing Flows. *arXiv e-prints*, art. arXiv:1810.03256, October 2018.

Schober, M., Särkkä, S., and Hennig, P. A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*, 29(1):99–122, Jan 2019.

Solak, E., Murray-smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. Derivative observations in gaussian process models of dynamic systems. In Becker, S., Thrun, S., and Obermayer, K. (eds.), *Advances in Neural Information Processing Systems 15*, pp. 1057–1064. MIT Press, 2003.

Sontag, E. D. *Mathematical control theory*. Springer Science & Business Media, 1998.

Teymur, O., Lie, H. C., Sullivan, T., and Calderhead, B. Implicit probabilistic integrators for odes. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 7255–7264. Curran Associates, Inc., 2018.

Tronarp, F., Kersting, H., Särkkä, S., and Hennig, P. Probabilistic Solutions To Ordinary Differential Equations As Non-Linear Bayesian Filtering: A New Perspective. *arXiv e-prints*, art. arXiv:1810.03440, October 2018.

van der Vaart, A. and van Zanten, J. Information rates of nonparametric Gaussian process methods. *Journal of Machine Learning Research*, 12:2095–2119, 2011.

Wang, J., Cockayne, J., and Oates, C. On the Bayesian Solution of Differential Equations. *arXiv e-prints*, art. arXiv:1805.07109, May 2018.

Wendland, H. *Scattered data approximation*, volume 17. Cambridge university press, 2004.

Xi, X., Briol, F.-X., and Girolami, M. Bayesian quadrature for multiple related integrals. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, pp. 5373–5382. PMLR, 2018.

Zhang, J., Mokhtari, A., Sra, S., and Jadbabaie, A. Direct Runge-Kutta Discretization Achieves Acceleration. *arXiv e-prints*, art. arXiv:1805.00521, May 2018.