Pick-and-Mix Information Operators for Probabilistic ODE Solvers

Nathanael Bosch[1] **Filip Tronarp**[1] **Philipp Hennig**[1,2]
[1]University of Tübingen, [2]Max Planck Institute for Intelligent Systems, Tübingen, Germany
{nathanael.bosch, filip.tronarp, philipp.hennig}@uni-tuebingen.de

Abstract

Probabilistic numerical solvers for ordinary differential equations compute posterior distributions over the solution of an initial value problem via Bayesian inference. In this paper, we leverage their probabilistic formulation to seamlessly include additional information as general likelihood terms. We show that second-order differential equations should be directly provided to the solver, instead of transforming the problem to first order. Additionally, by including higher-order information or physical conservation laws in the model, solutions become more accurate and more physically meaningful. Lastly, we demonstrate the utility of flexible information operators by solving differential-algebraic equations. In conclusion, the probabilistic formulation of numerical solvers offers a flexible way to incorporate various types of information, thus improving the resulting solutions.

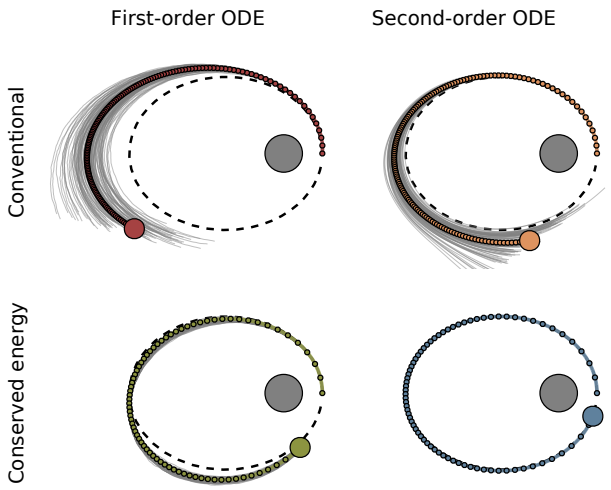

Figure 1: *Faithful modeling of ODE information improves probabilistic solutions.* Informing the solver about the second-order structure of the Kepler problem (●) increases the accuracy over its first-order counterpart (●). Adding physical information about the conservation of energy and angular momentum greatly improves the solution, *even for increased step sizes* (●). Full information leads to the best results (●). The dynamical system is described in Supplement A.6.

1 INTRODUCTION

Throughout science and engineering, dynamical systems are frequently described with ordinary differential equations (ODEs). But in many systems of interest, the differential equation harbors additional information not directly accessible to the numerical algorithm used to solve it. For example, physical systems often follow high-order dynamics and preserve quantities such as energy, mass, or angular momentum. To efficiently compute meaningful solutions, practitioners have to carefully choose from a wide range of numerical solvers, such as Runge–Kutta methods (Hairer et al., 1993), Nyström methods for second-order ODEs (Nyström, 1925), structure-preserving integrators (Hairer et al., 2006), and many more. Each of these methods can be seen as a laboriously custom-designed way to encode specific kinds of information. In this paper, we present a more flexible, unified approach to include additional knowledge into numerical ODE solutions, by leveraging the framework of *probabilistic numerics*.

In probabilistic numerics (Hennig et al., 2015; Oates and Sullivan, 2019), numerical problems are formulated as problems of probabilistic inference. Probabilistic numerical methods return *distributions over solutions*. Such methods can quantify their own approximation error through samples and other structured quantities – a functionality typically not provided by classic nu-

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

merical methods. This paper builds on probabilistic numerical ODE solvers based on Bayesian filtering and smoothing (Schober et al., 2019; Tronarp et al., 2019). These "ODE filters" have been shown to converge with polynomial rates (Kersting et al., 2020; Tronarp et al., 2021) and their efficiency has been demonstrated on a range of both non-stiff and stiff problems (Krämer and Hennig, 2020; Bosch et al., 2021).

Contributions Probabilistic ODE solvers are defined by two parts: the prior and the likelihood. In the basic version of such solvers, the likelihood is completely defined by the vector field. But, as we show in this work, their formulation is sufficiently flexible to allow for a much richer language. By formulating the likelihood in terms of flexible information operators, information about higher-order derivatives and conserved quantities can be represented with the same semantics as the ODE information itself. We demonstrate the utility of the proposed framework in four case studies:

1. *Second-order differential equations:* Solving second-order ODEs directly, instead of transforming them to first order, greatly improves the efficiency of probabilistic solvers.
2. *Additional second-derivative information:* Information about higher-order derivatives can be additionally included in the joint inference process to increase the solution accuracy.
3. *Systems with conserved quantities:* By including conservation laws into the model, probabilistic solutions become not only more accurate but also more physically meaningful.
4. *Differential-algebraic equations (DAEs):* With the corresponding information operator, probabilistic solvers can be extended to DAEs.

2 PROBABILISTIC ODE SOLVERS

This section introduces filtering-based probabilistic ODE solvers. Consider an initial value problem (IVP)

$$\dot{y} = f(y(t), t), \qquad \forall t \in [0, T], \tag{1}$$

with vector field $f : \mathbb{R}^{d+1} \to \mathbb{R}^d$ and initial value $y(0) = y_0 \in \mathbb{R}^d$. Instead of computing a single point estimate (as done by classic numerical algorithms), ODE filters compute *probabilistic* ODE solutions. That is, they approximate posterior distributions of the form

$$p\left(y(t) \mid y(t_0) = y_0, \{\dot{y}(t_n) = f(y(t_n), t_n)\}_{n=0}^{N}\right), \tag{2}$$

for a chosen time-discretization $\{t_n\}_{n=1}^{N}$. Thereby, they estimate not only the ODE solution, but also the unavoidable, global approximation error that arises due to discretization.

In the following, we pose the probabilistic numerical ODE solution as a problem of Bayesian state estimation, the solution of which can be efficiently approximated with extended Kalman filtering. For a more thorough introduction we refer to Tronarp et al. (2019).

2.1 Numerical ODE Solutions As Inference

Integrated Wiener Process Priors *A priori*, we model the unknown ODE solution $y(t)$ by a q-times integrated Wiener process (IWP). More precisely, define

$$Y(t) = \left[Y^{(0)}(t), Y^{(1)}(t), \ldots, Y^{(q)}(t)\right] \tag{3}$$

as the solution of a linear, time-invariant stochastic differential equation of the form

$$\mathrm{d}Y^{(i)}(t) = Y^{(i+1)}(t)\,\mathrm{d}t, \qquad i = 0, \ldots q-1, \tag{4a}$$

$$\mathrm{d}Y^{(q)}(t) = \Gamma^{1/2}\,\mathrm{d}W(t), \tag{4b}$$

$$Y(0) \sim \mathcal{N}\left(\mu_0, \Sigma_0\right), \tag{4c}$$

driven by a d-dimensional Wiener process W. The matrix $\Gamma^{1/2}$ is the symmetric square-root of some positive semi-definite matrix $\Gamma \in \mathbb{R}^{d\times d}$ and $\mu_0 \in \mathbb{R}^{d(q+1)}$, $\Sigma_0 \in \mathbb{R}^{d(q+1)\times d(q+1)}$ are the initial mean and covariance. Then, $Y^{(i)}$ models the i-th derivative of unknown ODE solution y and we write $y \sim \mathrm{IWP}(q)$.

Discrete-Time Transitions The process $Y(t)$ satisfies transition densities (Särkkä and Solin, 2019)

$$Y(t+h)\,|\,Y(t) \sim \mathcal{N}\left(A(h)Y(t), Q(h)\right). \tag{5}$$

The matrices $A(h), Q(h) \in \mathbb{R}^{d(q+1)\times d(q+1)}$ denote the transition matrix and the process noise covariance. For the chosen IWP(q) prior, it holds

$$A(h) = \breve{A}(h) \otimes I_d, \qquad Q(h) = \breve{Q}(h) \otimes \Gamma, \tag{6}$$

and the matrices $\breve{A}(h), \breve{Q}(h) \in \mathbb{R}^{(q+1)\times(q+1)}$ are known in closed form (Kersting et al., 2020):

$$\breve{A}_{ij}(h) = \mathbb{I}_{i\le j}\frac{h^{j-1}}{(j-i)!}, \tag{7a}$$

$$\breve{Q}_{ij}(h) = \frac{h^{2q+1-i-j}}{(2q+1-i-j)(q-i)!(q-j)!}. \tag{7b}$$

Measurement Process To relate the prior process to the ODE solution, we define a measurement model in terms of an *information operator* (Cockayne et al., 2019; Tronarp et al., 2019), similar to the likelihood models used in gradient matching (Calderhead et al., 2009; Wenk et al., 2020). Define

$$\mathcal{Z}[y](t) := \dot{y}(t) - f(y(t), t). \tag{8}$$

The operator $\mathcal{Z}$ maps the true ODE solution (see Eq. (1)) to a known quantity, namely the zero function; $\mathcal{Z}[y] \equiv 0$. On the other hand, the action of the information operator on the process Y can be expressed in terms of the following non-linear function

$$z(t, Y) := \mathcal{Z}[Y^{(0)}](t) = Y^{(1)}(t) - f\left(Y^{(0)}(t), t\right). \quad (9)$$

Once again, if $Y^{(0)}$ solves the ODE (Eq. (1)) exactly, we have $z(t, Y) \equiv 0$. Consequently, *inferring* the true ODE solution y reduces to conditioning the prior Y on $z(t, Y) = 0$. This inference problem is the subject of the next section.

2.2 Approximate Gaussian Inference

To enable tractable inference, we discretize time to a grid $\{t_n\}_{n=1}^N \subset [0, T]$ and condition the process $Y(t)$ only on discrete observations $z_n := z(t_n, Y(t_n)) = 0$. The resulting non-linear Gauss–Markov regression problem is well-known in the Bayesian filtering and smoothing literature (Särkkä, 2013). Its solution can be efficiently approximated with the extended Kalman filter (EKF), as Gaussian distributions

$$p\left(Y(t_n) \mid z_{1:n}\right) \approx \mathcal{N}\left(\mu_n, \Sigma_n\right). \quad (10)$$

In a nutshell, the EKF algorithm proceeds by iterating the following steps (Särkkä, 2013, Section 5.2):

- `PREDICT`: Given $Y(t_n) \mid z_{1:n} \sim \mathcal{N}(\mu_n, \Sigma_n)$ and the Gaussian transitions of Eq. (5), we can extrapolate to $Y(t_{n+1}) \mid z_{1:n} \sim \mathcal{N}(\mu_{n+1}, \Sigma_{n+1})$, with

$$\mu^-_{n+1} = A(h_n)\mu_n, \quad (11a)$$
$$\Sigma^-_{n+1} = A(h_n)\Sigma_n A(h_n)^\intercal + Q(h_n), \quad (11b)$$

  where $h_n := t_{n+1} - t_n$.

- `UPDATE`: To include information about the new measurement z_{n+1} into $Y(t_{n+1})$ approximate $Y(t_{n+1}) \mid z_{1:n+1} \sim \mathcal{N}(\mu_{n+1}, \Sigma_{n+1})$, with

$$\hat{z}_{n+1} = z\left(t_{n+1}, \mu^-_{n+1}\right) \quad (12a)$$
$$S_{n+1} = H_{n+1}\Sigma^-_{n+1}H^\intercal_{n+1}, \quad (12b)$$
$$K_{n+1} = \Sigma^-_{n+1}H^\intercal_{n+1}S^{-1}_{n+1}, \quad (12c)$$
$$\mu_{n+1} = \mu^-_{n+1} + K_{n+1}(z_{n+1} - \hat{z}_{n+1}), \quad (12d)$$
$$\Sigma_{n+1} = \Sigma^-_{n+1} - K_{n+1}S_{n+1}K^\intercal_{n+1}. \quad (12e)$$

  In a standard EKF, the matrix H_{n+1} denotes the Jacobian of the measurement model z, evaluated at μ^-_{n+1}. For z as defined in Eq. (9), we have $H_n := E_1 - J_f(E_0\mu_n, t_n)E_0$, where the matrices $E_i \in \mathbb{R}^{d \times d(q+1)}$ denote projection matrices to the i-th component of the state Y, that is $E_i Y = Y^{(i)}$.

  We call the resulting ODE solver `EK1` (Tronarp et al., 2019). Alternatively, Schober et al. (2019) use a zeroth-order approximation of the vector field, i.e. $H_n := E_1$. We refer to this solver as `EK0`.

Remark 1 (Smoothing)**.** *A Rauch–Tung–Striebel backward pass turns the filtering distribution into a smoothing posterior (Särkkä, 2013). At the final time point, the filtering and smoothing posteriors coincide.*

Remark 2 (Alternative inference schemes)**.** *The unscented Kalman filter (Julier and Uhlmann, 2004) can be used for Gaussian filtering, but requires multiple evaluations of the vector field at each time step. Particle filtering (Särkkä, 2013) can provide more descriptive, non-Gaussian ODE posterior estimates (Tronarp et al., 2019). But, similarly to sampling-based approaches to probabilistic ODE solutions (Chkrebtii et al., 2016; Conrad et al., 2017; Abdulle and Garegnani, 2020; Teymur et al., 2018), this expressivity comes at increased computational cost. In comparison, the EKF provides computationally efficient approximate inference.*

2.3 Practical Considerations

Calibration The posterior covariances returned by the solver depend on the choice of diffusion parameter Γ (recall Eq. (4)). Good uncertainty quantification therefore requires the estimation of Γ. In ODE filters, this is usually done by approximately maximizing the marginal likelihood of the observed data $p(z_{1:N})$ (Tronarp et al., 2019). This procedure also extends to more general, time-varying diffusion models Γ_n which have been proposed for greater flexibility (and for step-size adaptation; see below) (Schober et al., 2019). Refer to Bosch et al. (2021) for more detail.

Step-Size Adaptation In practice, computationally efficient ODE solvers typically rely on adaptive step-size selection (Hairer et al., 1993, Chapter II.4). We follow the presentation of Bosch et al. (2021) and control a local error estimate, derived from the measurement z, with a PI control algorithm (Gustafsson et al., 1988).

3 INFORMATION OPERATORS

The previous section established ODE filters as efficient algorithms for computing probabilistic numerical solutions of first-order ODEs. In the following, we extend their formulation to a broader class of problems and include additional types of information, by generalizing the underlying information operators.

The vector-field information enters the inference problem through the specified measurement model: f (recall Eq. (1)) only appears in the information operator $\mathcal{Z}$ (respectively z; see Eqs. (8) and (9)). However,

Table 1: Common problem settings and corresponding information operators.

Description	Equation	Information operator
First-order ODE	$\dot{y}(t) = f\left(y(t), t\right)$	$z(t, Y) := Y^{(1)} - f\left(Y^{(0)}, t\right)$
Second-order ODE	$\ddot{y}(t) = f\left(\dot{y}(t), y(t), t\right)$	$z(t, Y) := Y^{(2)} - f\left(Y^{(1)}, Y^{(0)}, t\right)$
Mass matrix DAE	$M\dot{y}(t) = f\left(y(t), t\right)$	$z(t, Y) := MY^{(1)} - f\left(Y^{(0)}, t\right)$
Invariances	$g\left(y(t), \dot{y}(t)\right) = 0$	$z(t, Y) := g\left(Y^{(0)}, Y^{(1)}\right)$
Chain rule	$\ddot{y}(t) = J_f\left(y(t)\right) \cdot \dot{y}(t)$	$z(t, Y) := Y^{(2)} - J_f\left(Y^{(0)}\right) \cdot Y^{(1)}$

the approximate inference algorithm itself (the EKF; see Section 2.2) does not rely on the specific form of the measurements; except for calibration and step-size adaptation, which we separately discuss below. To extend the ODE filter framework, we consider more general information operators, of the form

$$\mathcal{Z} \in \mathcal{I}_y := \{\mathcal{Z} : \mathcal{Z}[y] \equiv 0\}. \tag{13}$$

As before, they map some unknown function of interest y to the known zero function. But, this general form is not restricted to first-order ODEs. For example, given an energy-preserving system with second-order dynamics, we can formulate a corresponding operator to define its probabilistic solution (as will be shown in Section 4.3). Table 1 provides a summary of the problem settings and the corresponding operators considered in this paper, written in the functional form $z(t, Y) := \mathcal{Z}[Y^{(0)}](t)$. Before moving to our case studies, where each model will be explained in more detail, we discuss practical details and implementation.

Inference with Multiple Information Operators Some problems of interest provide multiple types of information about the true solution, for example as additional derivatives (Section 4.2) or physical conservation laws (Section 4.3). Formally, this amounts to an information operator $\mathcal{Z} \in \mathcal{I}_y$ that can be partitioned as $\mathcal{Z}[y] = [\mathcal{Z}_1[y]^\intercal, \mathcal{Z}_2[y]^\intercal]^\intercal$, with $\mathcal{Z}_1, \mathcal{Z}_2 \in \mathcal{I}_y$, and corresponding functional representation

$$z(t, Y) = [z_1(t, Y)^\intercal,\ z_2(t, Y)^\intercal]^\intercal. \tag{14}$$

It is still possible to update jointly on both measurement models in a single EKF update step on z; this strategy is chosen in Section 4.2. However, performing two separate update steps can sometimes be preferable (Raitoharju and Piché, 2019; Raitoharju et al., 2016, 2017). In this case, each measurement model is linearized separately in the partially updated state. This strategy is chosen in Section 4.3.

Calibration and Step-Size Adaptation The approaches for calibration and adaptive step-size selection discussed in Section 2.3 do not strictly depend on the specific information operator, but they were developed in the context of first-order ODEs (Bosch et al., 2021). There, the information operator is d-dimensional, i.e. $z(t, Y) \in \mathbb{R}^d$, and describes the *local defect*. We found that this formulation can be extended to settings with a different problem structure (in this work, second-order ODEs and DAEs), but for settings with multiple sources of information (here, additional derivatives or invariances) special care has to be taken. To conveniently consider user-specified relative tolerance levels, the local error should be of the same dimension as the ODE solution. Thus, in Sections 4.2 and 4.3, only the part of the measurement model that relates to the given differential equation is considered for calibration and step-size adaptation.

4 CASE STUDIES

We evaluate the presented framework in four case studies. First, we apply the probabilistic solver to second-order ODEs. We investigate the difference between solving such problems directly, by selecting the correct information operator, and solving the algebraically (but not numerically) equivalent first-order ODEs. Second, we augment the probabilistic numerical solver for first-order ODEs with second-derivative information, which can be computed from the ODE via the chain rule. Third, we consider Hamiltonian systems in which the total energy is conserved over time, and we evaluate the influence of this information on the probabilistic numerical solution. Fourth, we demonstrate how probabilistic solvers can be extended to solve semi-explicit differential-algebraic equations.

Implementation The implementation follows the practices suggested by Krämer and Hennig (2020) and includes exact initialization, preconditioned state transitions, and a square-root implementation. All experiments are implemented in the Julia programming language (Bezanson et al., 2017). Reference solutions are computed with DifferentialEquations.jl (Rackauckas and Nie, 2017). All experiments run on a single, consumer-level CPU. Code for the implementation and experiments is publicly available on GitHub.[1]

[1] `github.com/nathanaelbosch/pick-and-mix`

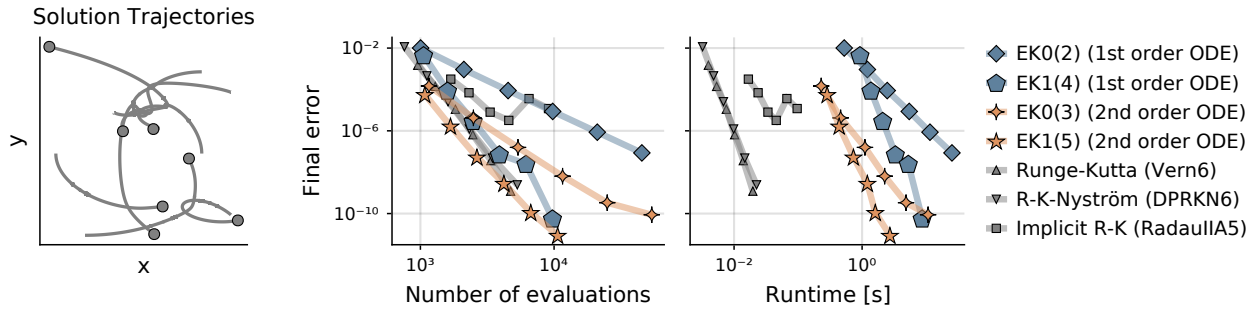

Figure 2: *Second-order ODEs should be solved directly.* The Pleiades system describes the motion of seven stars in a plane (left). Solving this problem directly in second order, compared to solving the equivalent first-order ODE, improves accuracy and efficiency, both in the number of function evaluations (center) and runtime (right).

4.1 Second-Order Differential Equations

This first case study demonstrates how information about the problem structure, such as the order of the ODE, can improve probabilistic solutions. To this end, consider an autonomous, *second*-order ODE

$$\ddot{y}(t) = f\left(\dot{y}(t), y(t)\right), \qquad \forall t \in [0, T], \tag{15}$$

with vector field $f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ and initial values $y(0) = y_0$, $\dot{y}(0) = \dot{y}_0$.

Second-order ODEs can be transformed to first order by defining a new variable $\widetilde{y} := (\dot{y}, y)$. They can therefore, in principle, be solved by any generic solver. However, doubling the dimension of the ODE can increase both the solver runtime and memory cost. Specialized non-probabilistic solvers such as Nyström methods have been specifically developed to circumvent this issue (Nyström, 1925; Hairer et al., 1993). In this section, we follow a similar (but much simpler) approach and present a direct application of probabilistic solvers to second-order ODEs.

The motivation is twofold. First, a duplication of the ODE dimension leads to a 4x increase in memory cost and 8x runtime, since the EKF algorithm relies on matrix-matrix operations on the state covariances. Second, the structure of the transformed problem is not a good fit for the integrated Wiener process prior. After transformation, the first derivative $\dot{y}$ appears both in $\widetilde{y}$ and $\frac{\mathrm{d}\widetilde{y}}{\mathrm{d}t}$. It is therefore modeled with both an IWP(q) and IWP$(q-1)$ prior *at the same time* (recall Section 2.1). Both of these shortcomings can be circumvented by solving the second-order problem directly.

Solver Setup The second-order ODE (Eq. (15)) induces an information operator of the form

$$z(t, Y) = Y^{(2)} - f\left(Y^{(1)}, Y^{(0)}\right). \tag{16}$$

We consider two linearizations:

$$H := \begin{cases} E_2, & (\texttt{EK0}) \\ E_2 - \frac{\partial f}{\partial y} \cdot E_0 - \frac{\partial f}{\partial \dot{y}} \cdot E_1, & (\texttt{EK1}) \end{cases} \tag{17}$$

named in correspondence to the existing probabilistic solvers for first-order problems presented in Section 2.2.

Experiment Setup We evaluate the solvers on the Pleiades problem (Hairer et al., 1993, Chapter II.10), a system of 14 second-order ODEs (full problem definition in Supplement A.1). All solvers use adaptive steps and a time-varying diffusion model (Bosch et al., 2021). We compare the resulting mean absolute errors at final time T, referred to as "final error". Thus, the solutions were not smoothed. For a fair comparison, the orders of the first-order solvers are lowered by one compared to their second-order counterparts, such that their highest modeled derivatives coincide.

Results The work-precision diagrams in Fig. 2 show that second-order ODEs are solved both more efficiently and more accurately than their first-order counterparts. We observe not only an improvement in absolute runtime, but also a reduced error even for comparable numbers of vector-field evaluations. Figure 2 also compares the solvers to well-established non-probabilistic methods, including an explicit Runge–Kutta solver (`Vern6`; Verner, 2010), a Nyström method (`DPRKN6`; Dormand and Prince, 1987), and an implicit solver (`RadauIIA5`; Hairer and Wanner, 1999). While these classic solvers require a comparable number of vector-field evaluations, they exhibit a reduced absolute runtime. Since probabilistic solvers have the same cubic complexity as the classic, implicit `RadauIIA5`, we suspect that this discrepancy is partly due to the well-optimized implementation of the DifferentialEquations.jl library (Rackauckas and Nie, 2017). On the other hand, probabilistic ODE solvers provide strictly more functionality than non-probabilistic methods, thus a certain increase

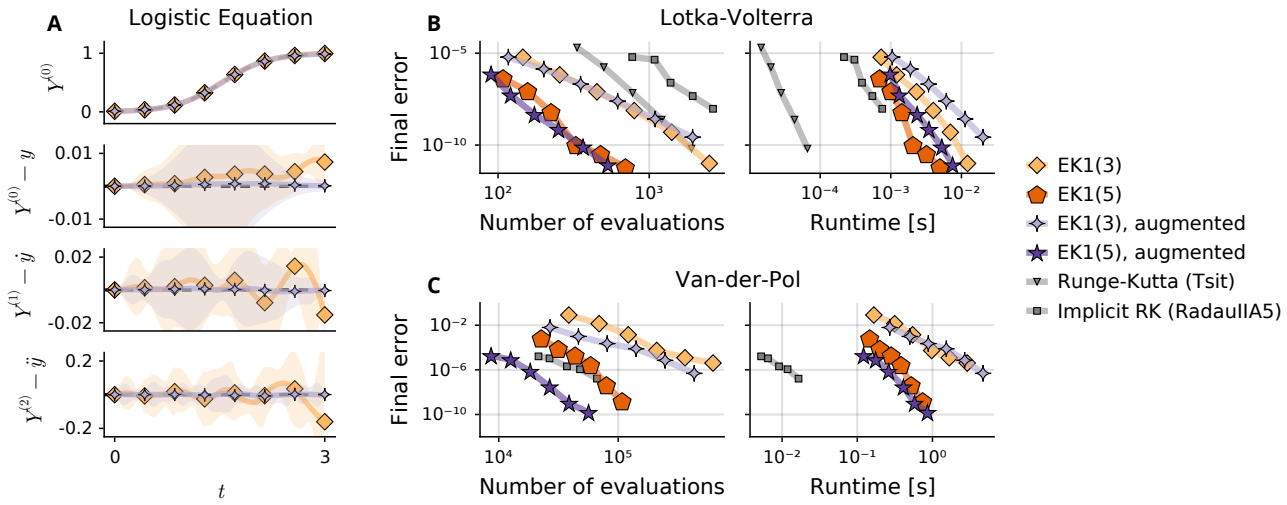

Figure 3: *Additional second-derivative information can improve probabilistic solutions.* On a fixed time discretization, additional information about second-derivatives reduces the approximation error (A). For adaptive-step solvers, it depends on the specific problem. On the non-stiff Lotka–Volterra problem, the utility of the additional information seems limited (B). However, the benefit of second-derivative information on the stiff Van–der–Pol problem outweighs the additional computational cost and leads to reduced runtimes (C).

in runtime is expected. As demonstrated by this case study, this paper further reduces the gap between probabilistic and non-probabilistic methods by providing ODE filters for second-order differential equations.

4.2 First-Order ODEs with Additional Second-Derivative Information

In this section, we augment the probabilistic solver with additional second-derivative information, that can be derived from a standard, first-order problem. For this, consider an autonomous, explicit, first-order ODE

$$\dot{y}(t) = f\left(y(t)\right), \qquad \forall t \in [0, T], \tag{18}$$

with vector field $f : \mathbb{R}^d \to \mathbb{R}^d$ and initial value $y(0) = y_0 \in \mathbb{R}^d$. Second derivatives of the true solution can be derived from Eq. (18) by differentiating both sides and applying the chain rule. We obtain

$$\ddot{y}(t) = J_f(y(t)) \cdot f\left(y(t)\right), \tag{19}$$

where J_f is the Jacobian of f.

Solver Setup Equations (18) and (19) motivate a measurement model $z(t, Y) := [z_1(t, Y)^\intercal, z_2(t, Y)^\intercal]^\intercal$,

$$z_1(t, Y) := Y^{(1)} - f\left(Y^{(0)}\right), \tag{20a}$$

$$z_2(t, Y) := Y^{(2)} - J_f(Y^{(0)}) f\left(Y^{(0)}\right). \tag{20b}$$

In this evaluation, we consider exact linearizations of both z_1 and z_2 (computed with automatic differentiation). Furthermore, the solvers update on both measurement models in a single, joint update step.

Fixed-Step Results We first evaluate the proposed method in a simplified setting to visualize the effect of additional second-derivative information. To this end, consider the logistic ODE (defined in Supplement A.2), and fixed-step solvers with $\Delta t = 3/7$. Figure 3 (A) shows the results. Both solvers approximate the true solution, but the more informed solver achieves lower approximation errors and has reduced uncertainties.

Adaptive-Step Results Next, we evaluate the proposed method on the non-stiff Lotka–Volterra problem (Supplement A.3) and the stiff Van–der–Pol model (Supplement A.4), in conjunction with adaptive step-size selection and a dynamic diffusion model (Bosch et al., 2021). Figure 3 shows the resulting work-precision diagrams. On the Lotka–Volterra problem (B), we observe that additional information does not strictly lead to improvements. Here, the original `EK1` solvers seem preferable. On the other hand, the additional second-derivative information leads to increased accuracy and even to a reduction in the number of vector-field evaluations on the stiff Van–der–Pol problem (C).

4.3 Systems with Conserved Quantities

In this case study, we demonstrate how additional knowledge about conserved quantities of the modeled dynamical system can be provided to the probabilistic solver. To this end, we consider *Hamiltonian problems*, a particular class of dynamical systems of the form

$$\dot{p} = -\frac{\partial H}{\partial q}(p, q), \qquad \dot{q} = \frac{\partial H}{\partial p}(p, q), \tag{21}$$

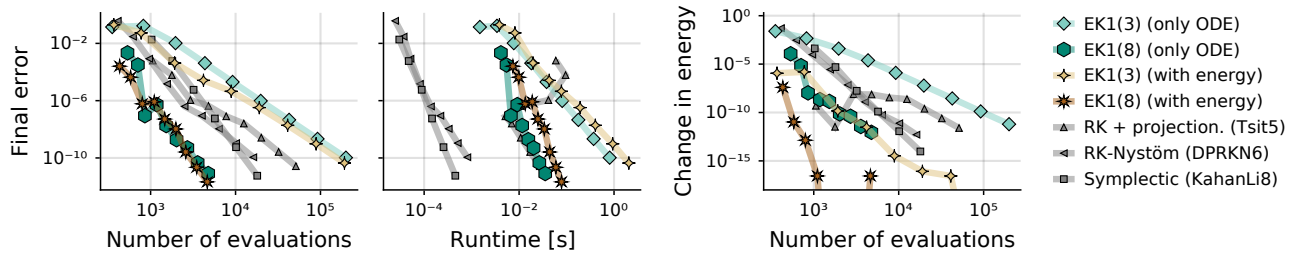

Figure 4: *Work-precision diagram of numerical solvers with and without energy conservation.* Additional information about the total energy in the dynamical Hénon–Heiles system can improve the accuracy of the solution (left). This comes with additional computational cost and increases the runtime (center). But as a result, the total energy is conserved more strictly and solutions become more physically meaningful (right).

where the Hamiltonian $H : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ describes the total energy in the dynamical system. Hamiltonian problems form an important class of ODEs in the context of geometric numerical integration (Hairer et al., 2006) since their trajectories *preserve the Hamiltonian*. That is, for a solution $(p(t), q(t))$ of such problems, the Hamiltonian $H(p(t), q(t))$ is constant, and it holds

$$g\left(p(t), q(t)\right) := H(p(t), q(t)) - H(p(0), q(0)) \equiv 0. \tag{22}$$

Geometric integrators aim to preserve this structure in their numerical approximation. In the following, we present a probabilistic solver for Hamiltonian problems that includes this additional information into its inference process to improve its solution estimates.

Solver Setup The problems considered in this section can all be written as second-order ODEs, with $(\dot{y}, y) := (p, q)$. Together with the conservation law of Eq. (22), this motivates a partitioned measurement model $z(t, Y) := [z_1(t, Y)^\intercal, z_2(t, Y)^\intercal]^\intercal$, with

$$z_1(t, Y) := Y^{(2)} - f\left(Y^{(0)}\right), \tag{23a}$$

$$z_2(t, Y) := g(Y^{(1)}, Y^{(0)}), \tag{23b}$$

where f denotes the vector field of the corresponding ODE. As in the previous section, all considered methods rely on exact linearizations of the measurement models. In addition, the solvers perform a *partitioned* EKF update. That is, they separately linearize and update first on the ODE information z_1 and then on the conserved quantity z_2 – a procedure that parallels established "projection methods" used with non-probabilistic ODE solvers (Hairer et al., 2006, Section IV.4).

Problem Setting We mainly consider the Hénon–Heiles model which describes a star moving around a galactic center (Henon and Heiles, 1964). The full problem definition is given in Supplement A.5. We compare probabilistic solvers with and without additional information about the conservation of energy, for various orders ($q \in \{3, 8\}$). All solvers use adaptive steps and dynamic diffusion models. Since we evaluate the error at the final time point, smoothing is not required.

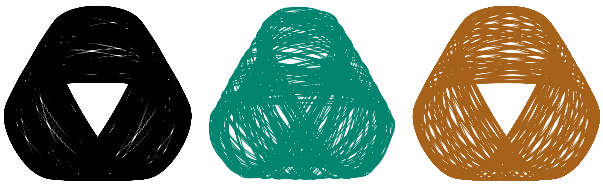

Figure 5: *Conservation stabilizes long simulations.* Probabilistic numerical simulations of the Hénon–Heiles problem over long time horizons, computed with adaptive steps and low precision, deteriorate over time (middle) and deviate strongly from the true trajectory (left). By including energy-preservation into the solver, long-term simulations become more accurate (right).

Results Figure 4 shows the results in multiple work-precision diagrams. We observe that the additional information leads, in some configurations, to improved accuracies, but comes with an increase in absolute runtime. However, the probabilistic solvers enforce the conservation of energy very strictly – even in comparison to non-probabilistic approaches that are particularly well suited for this problem setting, including a Runge–Kutta solver (`Tsit5`; Tsitouras, 2011) combined with a projection method (Hairer et al., 2006, Section IV.4), a Runge–Kutta–Nyström solver (`DPRKN6`; Dormand and Prince, 1987), and a symplectic integrator (`KahanLi8`; Kahan and Li, 1997). This structural preservation is of major concern to obtain physically meaningful solutions and stable long-term simulations of Hamiltonian systems (Hairer et al., 2006). The conservation of energy is therefore often of higher importance than a sole reduction in the (Euclidean) error. Following this motivation, Fig. 5 shows how energy preservation stabilizes long-term simulations with probabilistic solvers. Finally, Fig. 6 demonstrates on the Kepler problem

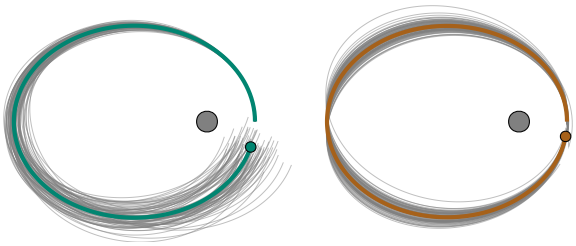

Figure 6: *Energy preservation affects the covariances.* (For a clearer visualization of the samples, covariances are inflated by a factor of 3 (left) and 300 (right).) The samples of a standard probabilistic solution of the Kepler problem appear physically implausible (left). By informing the solver about the conservation of energy and angular momentum, the posterior distribution becomes more physically meaningful (right).

(defined in Supplement A.6) how physical information influences not only the mean of the solution estimate, but also its covariances.

4.4 Differential-Algebraic Equations

In our final case study, we demonstrate how flexible information operators can be used to extend probabilistic solvers to completely new problem classes. To this end, we consider systems of the form

$$M\dot{y}(t) = f\left(y(t)\right), \qquad \forall t \in [0, T], \tag{24}$$

with vector field $f : \mathbb{R}^d \to \mathbb{R}^d$, initial values $y(0) = y_0$, and *mass matrix* $M \in \mathbb{R}^{d\times d}$. If M is singular, the system can not be rewritten as a regular ODE and we call Eq. (24) a differential-algebraic equation (DAE). For instance, in the Robertson DAE considered in this case study, we have $M = \mathrm{diag}\left([1, 1, 0]\right)$. The system thus describes two ODEs and one algebraic equation.

DAEs arise naturally in many dynamical systems, such as multi-body dynamics, chemical kinetics, or optimal control (Brenan et al., 1996). Their numerical simulation is notoriously challenging and often requires specialized methods; only a specific subset of classic ODE solvers is able to solve the problem given in Eq. (24) (Petzold, 1982). To the best of our knowledge, this work presents the first *probabilistic* DAE solver.

Solver Setup To encode the DAE information of Eq. (24), we define a measurement model

$$z(t, Y) := MY^{(1)} - f\left(Y^{(0)}\right). \tag{25}$$

In our experiments, we consider exact linearizations

$$H := M \cdot E_1 - J_f\left(Y^{(0)}\right) \cdot E_0, \tag{26}$$

together with adaptive step-size selection and dynamically calibrated diffusions. Smoothing is not required.

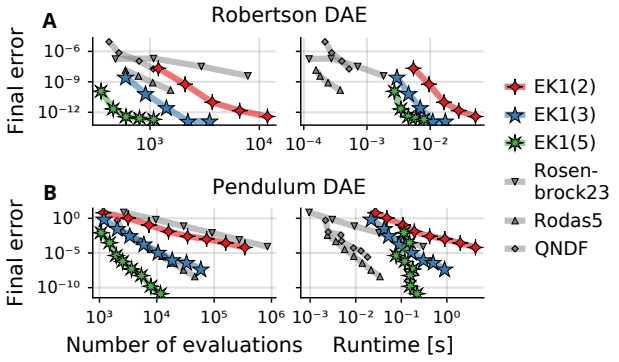

Figure 7: *With the correct information operator, probabilistic ODE solvers can solve semi-explicit DAEs.*

Experiment and Results To investigate the utility of the proposed methods, we compare probabilistic solvers with various orders ($q \in \{2, 3, 5\}$) to three non-probabilistic DAE solvers: Rosenbrock methods of order 2 and 5 (`Rosenbrock23` & `Rodas5`; Hairer and Wanner, 1996) and an adaptive-order multistep method (`QNDF`; Shampine and Reichelt, 1997). All methods are evaluated on the stiff Robertson DAE and on a non-stiff pendulum DAE (defined in Supplements A.7 and A.8). Figure 7 shows the resulting work-precision diagrams. As one would expect, increasing the number of steps leads to reduced error. In addition, we observe higher convergence rates for solvers of higher order. While the proposed solvers display higher runtimes than their classic counterparts, the differences are comparable to our results in the other case studies. In the number of vector-field evaluations, probabilistic and non-probabilistic solvers appear comparable. In summary, the proposed probabilistic solvers demonstrate good performance on the considered DAEs.

5 CONCLUSION

We have shown how to improve ODE solvers by drawing on various sources of information, within the framework of probabilistic numerics. The proposed algorithm performs efficient inference with extended Kalman filtering and can leverage existing methods for uncertainty calibration and step-size adaptation. In four case studies, we demonstrated how information about problem structure, additional derivatives, and conserved quantities can be used to improve the solver performance and the quality of the posterior distributions. By providing a flexible and efficient means to encode mechanistic knowledge beyond the ODE itself, our proposed framework further reduces the gap between probabilistic and non-probabilistic methods and thereby enriches the interface of mechanistic inference and simulation.

Acknowledgements

The authors gratefully acknowledge financial support by the German Federal Ministry of Education and Research (BMBF) through Project ADIMEM (FKZ 01IS18052B), and financial support by the European Research Council through ERC StG Action 757275 / PANAMA; the DFG Cluster of Excellence "Machine Learning - New Perspectives for Science", EXC 2064/1, project number 390727645; the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ: 01IS18039A); and funds from the Ministry of Science, Research and Arts of the State of Baden-Württemberg. The authors also thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting N. Bosch. The authors are grateful to Nicholas Krämer for many valuable discussions and thank Jonathan Wenger and Jonathan Schmidt for helpful feedback on the manuscript.

References

Abdulle, A. and Garegnani, G. (2020). Random time step probabilistic methods for uncertainty quantification in chaotic and geometric numerical integration. *Statistics and Computing*, 30.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59.

Bosch, N., Hennig, P., and Tronarp, F. (2021). Calibrated adaptive probabilistic ODE solvers. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*. PMLR.

Brenan, K., Campbell, S., Campbell, S., and Petzold, L. (1996). *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics.

Calderhead, B., Girolami, M., and Lawrence, N. (2009). Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc.

Chkrebtii, O. A., Campbell, D. A., Calderhead, B., and Girolami, M. A. (2016). Bayesian Solution Uncertainty Quantification for Differential Equations. *Bayesian Analysis*, 11(4).

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. (2019). Bayesian probabilistic numerical methods. *SIAM review*, 61.

Conrad, P. R., Girolami, M., Särkkä, S., Stuart, A., and Zygalakis, K. (2017). Statistical analysis of differential equations: introducing probability measures on numerical solutions. *Statistics and Computing*, 27.

Dormand, J. and Prince, P. (1987). Runge-Kutta-Nystrom triples. *Computers & Mathematics with Applications*, 13.

Gustafsson, K., Lundh, M., and Söderlind, G. (1988). A PI stepsize control for the numerical solution of ordinary differential equations. *BIT Numerical Mathematics*, 28.

Hairer, E., Lubich, C., and Wanner, G. (2006). *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg.

Hairer, E., Norsett, S., and Wanner, G. (1993). *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8. Springer-Verlag.

Hairer, E. and Wanner, G. (1996). *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, volume 14. Springer-Verlag.

Hairer, E. and Wanner, G. (1999). Stiff differential equations solved by Radau methods. *Journal of Computational and Applied Mathematics*, 111.

Hennig, P., Osborne, M. A., and Girolami, M. (2015). Probabilistic numerics and uncertainty in computations. *Proceedings. Mathematical, physical, and engineering sciences*, 471.

Henon, M. and Heiles, C. (1964). The applicability of the third integral of motion: Some numerical experiments. *The Astronomical Journal*, 69.

Julier, S. and Uhlmann, J. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92.

Kahan, W. and Li, R.-C. (1997). Composition constants for raising the orders of unconventional schemes for ordinary differential equations. *Mathematics of computation*, 66.

Kersting, H., Sullivan, T. J., and Hennig, P. (2020). Convergence rates of Gaussian ODE filters. *Statistics and Computing*, 30.

Krämer, N. and Hennig, P. (2020). Stable implementation of probabilistic ODE solvers. *arXiv:2012.10106 [stat.ML]*.

Nyström, E. (1925). *Über die numerische integration von differentialgleichungen*. Acta Societatis Scientiarum Fennicae. Finn. Literaturges.

Oates, C. J. and Sullivan, T. J. (2019). A modern retrospective on probabilistic numerics. *Statistics and Computing*, 29.

Petzold, L. (1982). Differential/algebraic equations are not ODE's. *Siam Journal on Scientific and Statistical Computing*, 3.

Rackauckas, C. and Nie, Q. (2017). DifferentialEquations.jl – a performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, 5.

Raitoharju, M. and Piché, R. (2019). On computational complexity reduction methods for Kalman filter extensions. *IEEE Aerospace and Electronic Systems Magazine*, 34.

Raitoharju, M., Piché, R., Ala-Luhtala, J., and Ali-Löytty, S. (2016). Partitioned update Kalman filter. *Journal of Advances in Information Fusion*, 11.

Raitoharju, M., Ángel F. García-Fernández, and Piché, R. (2017). Kullback–Leibler divergence approach to partitioned update Kalman filter. *Signal Processing*, 130.

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*, volume 3 of *Institute of Mathematical Statistics textbooks*. Cambridge University Press.

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

Shampine, L. F. and Reichelt, M. W. (1997). The Matlab ODE suite. *SIAM journal on scientific computing*, 18.

Särkkä, S. and Solin, A. (2019). *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press.

Teymur, O., Lie, H. C., Sullivan, T., and Calderhead, B. (2018). Implicit probabilistic integrators for ODEs. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Tronarp, F., Kersting, H., Särkkä, S., and Hennig, P. (2019). Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: a new perspective. *Statistics and Computing*, 29.

Tronarp, F., Särkkä, S., and Hennig, P. (2021). Bayesian ODE solvers: the maximum a posteriori estimate. *Statistics and Computing*, 31.

Tsitouras, C. (2011). Runge–Kutta pairs of order 5 (4) satisfying only the first column simplifying assumption. *Computers & Mathematics with Applications*, 62.

van der Pol, B. (1926). On "relaxation-oscillations". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2.

Verner, J. H. (2010). Numerically optimal Runge–Kutta pairs with interpolants. *Numerical Algorithms*, 53.

Wenk, P., Abbati, G., Osborne, M. A., Schölkopf, B., Krause, A., and Bauer, S. (2020). ODIN: ODE-informed regression for parameter and state inference in time-continuous dynamical systems. In *Proceedings of the 34th Conference on Artificial Intelligence (AAAI)*, volume 34. AAAI Press.

Supplementary Material: Pick-and-Mix Information Operators for Probabilistic ODE Solvers

A PROBLEM DEFINITIONS

A.1 Pleiades

The Pleiades system describes the motion of seven stars in a plane, with coordinates (x_i, y_i) and masses $m_i = i$, $i = 1, \ldots, 7$ (Hairer et al., 1993, II.10). It is given by a second-order ODE

$$\ddot{x}_i = \sum_{j \neq i} m_j (x_j - x_i)/r_{ij}, \qquad \ddot{y}_i = \sum_{j \neq i} m_j (y_j - y_i)/r_{ij}, \tag{27}$$

where $r_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2\right)^{3/2}$, for $i, j = 1, \ldots, 7$, on the time span $t \in [0, 3]$, with initial locations

$$x(0) = [3, 3, -1, -3, 2, -2, 2], \tag{28a}$$
$$y(0) = [3, -3, 2, 0, 0, -4, 4], \tag{28b}$$

and initial velocities

$$\dot{x}(0) = [0, 0, 0, 0, 0, 1.75, -1.5], \tag{28c}$$
$$\dot{y}(0) = [0, 0, 0, -1.25, 1, 0, 0]. \tag{28d}$$

A.2 Logistic Equation

The logistic equation is a simple IVP problem, given as

$$\dot{y}(t) = 3y(t)(1 - y(t)), \qquad t \in [0, 3], \qquad y(0) = 100, \tag{29}$$

for which the analytical solution is known to be

$$y(t) = \frac{\exp(3t)}{100 - 1 + \exp(3t)}. \tag{30}$$

A.3 Lotka–Volterra

The Lotka–Volterra model describes the dynamics of biological systems in which two species interact, one as a predator and the other as prey. The IVP is given by the ODE

$$\dot{x} = 1.5x - xy, \qquad \dot{y} = xy - 3y. \tag{31}$$

In our experiments, we consider initial values $x(0) = 1$, $y(0) = 1$ and a time span $t \in [0, 7]$.

A.4 Van–der–Pol

The Van der Pol model (van der Pol, 1926) describes a non-conservative oscillator with non-linear damping. In our experiment, we consider a notoriously stiff version of the model, given as

$$\dot{y_1}(t) = y_2(t), \qquad \dot{y_2}(t) = 10^6 \left(\left(1 - y_1^2(t)\right) y_2(t) - y_1(t)\right), \tag{32a}$$

on the time span $t \in [0, 10]$, with initial value $y(0) = [0, \sqrt{3}]^\intercal$.

A.5 Hénon–Heiles

The Hénon-Heiles model describes a star moving around a galactic center, with its motion restricted to a plane (Henon and Heiles, 1964). It is defined by a Hamiltonian

$$H(p,q) = \left[\frac{1}{2}\left(p_1^2 + p_2^2\right)\right] + \left[\frac{1}{2}\left(q_1^2 + q_2^2\right) + q_1^2 q_2 - \frac{1}{3}q_2^3\right], \tag{33}$$

which describes the kinetic and potential energy of the star with velocity p and location q. With $y(t) := q(t)$, we write the Hénon-Heiles problem as an IVP with second-order ODE, as

$$\ddot{y}_1(t) = -y_1(t) - 2y_1(t)y_2(t), \tag{34a}$$
$$\ddot{y}_2(t) = y_2^2(t) - y_2(t) - y_1^2(t), \tag{34b}$$

on the time span $t \in [0, 1000]$, with initial values $y(0) = (0, 0.1)$, $\dot{y}(0) = (0.5, 0)$. It further holds

$$g(\dot{y}(t), y(t)) := H(\dot{y}(t), y(t)) - H(\dot{y}_0(t), y_0(t)) = 0, \tag{35}$$

by conservation of the Hamiltonian (Hairer et al., 2006).

A.6 Kepler Problem

The Kepler problem is a special case of the two-body problem in celestial mechanics, and can be used to describe the movement of a planet around a star. It is given by a Hamiltonian $H : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$, with

$$H\left(p(t), q(t)\right) = \frac{\|p(t)\|^2}{2} - \frac{1}{\|q(t)\|}. \tag{36}$$

With $y(t) := q(t)$ and $\dot{y}(t) := p(t)$, it induces the second-order ODE

$$\ddot{y}(t) = -\frac{y(t)}{\|y(t)\|^3}. \tag{37}$$

In our experiments, the Kepler problem is solved on the time span $t \in \left[0, \frac{99}{100} \cdot 2\pi\right]$, with initial values $y(0) = [0.4, 0]$, $\dot{y}(0) = [0, 2]$. In addition to conserving the Hamiltonian, the Kepler system conserves angular momentum:

$$L\left(p(t), q(t)\right) = q_1(t)p_2(t) - q_2(t)p_1(t). \tag{38}$$

Thus, it holds

$$g(\dot{y}(t), y(t)) := \begin{bmatrix} H(\dot{y}(t), y(t)) - H(\dot{y}(0), y(0)) \\ L(\dot{y}(t), y(t)) - L(\dot{y}(0), y(0)) \end{bmatrix} = 0. \tag{39}$$

A.7 Robertson DAE

The Robertson DAE describes a system of chemical reactions and is a very popular problem to evaluate stiff ODE and DAE solvers (Hairer and Wanner, 1996). As a DAE, it is given by the equations

$$y_1(t) = -0.04y_1(t) + 10^4 y_2(t)y_3(t), \tag{40a}$$
$$y_2(t) = 0.04y_1(t) + 10^4 y_2(t)y_3(t) - (3 \cdot 10^7)y_2(t)^2, \tag{40b}$$
$$0 = y_1(t) + y_2(t) + y_3(t) - 1, \tag{40c}$$

and it therefore has a singular mass matrix of the form

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

We consider an initial value $y(0) = [1, 0, 0]$, and while the system is most often simulated on the time span $t \in [0, 10^5]$, we solve it on $t \in [0, 10^2]$ since we found the final error to be more informative in this setting since the values did then not saturate yet.

A.8 Pendulum DAE

A pendulum can be described in Cartesian coordinates with the following, index-reduced DAE (Hairer and Wanner, 1996)

$$\dot{x}(t) = v_x, \tag{41a}$$
$$\dot{v_x}(t) = xT, \tag{41b}$$
$$\dot{y}(t) = v_y, \tag{41c}$$
$$\dot{v_y}(t) = yT - g, \tag{41d}$$
$$0 = 2(v_x^2 + v_y^2 + y(yT - g) + Tx^2). \tag{41e}$$

In our experiments, we consider initial values $x(0) = 1$, $v_x(0) = 0$, $y(0) = 0$, $v_y(0) = 0$, $T(0) = 0$, and the gravitational acceleration $g = 9.81$. We simulate the system on the time span $t \in [0, 10]$.