Supplement for the Manuscript "Bayesian Solution Uncertainty Quantification for Differential Equations"

Oksana A. Chkrebtii*, David A. Campbell† Ben Calderhead‡ and Mark A. Girolami§

Appendix A: Brief overview of differential equation models

Ordinary Differential Equation (ODE) models represent the evolution of system states $u \in \mathcal{U}$ over time $t \in [0, L]$, $L > 0$ implicitly through the equation $u_t(t, \theta) = f\{t, u(t, \theta), \theta\}$. The standard assumptions about the known function f are that it is continuous in the first argument and Lipschitz continuous in the second argument, ensuring the existence and uniqueness of a solution corresponding to a known initial state $u^*(0)$. Inputs and boundary constraints define several model variants.

The *Initial Value Problem* (IVP) of order one models time derivatives of system states as known functions of the states constrained to satisfy initial conditions, $u^*(0)$,

$$\begin{cases} u_t(t) = f\{t, u(t), \theta\}, & t \in [0, L],\ L > 0, \\ u(0) = u^*(0). \end{cases} \tag{A.1}$$

The existence of a solution is guaranteed under mild conditions (see for example, Butcher, 2008; Coddington and Levinson, 1955). Such models may be high dimensional and contain other complexities such as algebraic components, functional inputs, or higher order terms.

While IVP models specify a fixed initial condition on the system states, the *Mixed Boundary Value Problem* (MBVP) may constrain different states at different time points. Typically these constraints are imposed at the boundary of the time domain. For example, the two state mixed boundary value problem is,

$$\begin{cases} [u_t(t), v_t(t)] = f\{t, [u(t), v(t)], \theta\}, & t \in [0, L],\ L > 0, \\ \phi\{v(0), u(L)\} = 0, \end{cases} \tag{A.2}$$

and can be straightforwardly generalized to higher dimensions and extrapolated beyond the final time point L. Whereas a unique IVP solution exists under relatively mild conditions, imposing mixed boundary constraints can result in multiple solutions (e.g. Keller, 1968) introducing possible problems for parameter estimation.

*Department of Statistics, The Ohio State University, Columbus, OH, USA oksana@stat.osu.edu
†Department of Statistics & Actuarial Science, Simon Fraser University, Canada dac5@stat.sfu.ca
‡Department of Mathematics, Imperial College London, London, UK b.calderhead@ucl.ac.uk
§Department of Statistics, The University of Warwick, Coventry, UK M.Girolami@warwick.ac.uk

The *Delay Initial Function Problem* (DIFP) generalizes the initial constraint of IVPs to an initial function, $\phi(t)$, thereby relating the derivative of a process to both present and past states at lags $\tau_j \in [0, \infty)$,

$$\begin{cases} u_t(t) &= f\left\{t, u(t-\tau_1), \dots, u(t-\tau_d), \theta\right\}, \quad t \in [0, L],\, L > 0, \\ u(t) &= \phi(t), \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\;\; t \in [0 - \max_j(\tau_j), 0]\,. \end{cases} \tag{A.3}$$

DIFPs are well suited to describing biological and physical dynamics that take time to propagate through systems. However, they pose challenges to numerical techniques due to potentially large and structured truncation error (Bellen and Zennaro, 2003).

Partial Differential Equation (PDE) models represent the derivative of states with respect to multiple arguments, for example time and spatial variables. PDE models are ubiquitous in the applied sciences, where they are used to study a variety of phenomena, from animal movement, to the propagation pattern of a pollutant in the atmosphere. The main classes of PDE models are based on elliptic, parabolic and hyperbolic equations. Further adding to their complexity, functional boundary constraints and initial conditions make PDE models more challenging, while their underlying theory is less developed compared to ODE models (Polyanin and Zaitsev, 2004).

Appendix B: Problem specific algorithms

This section provides algorithms for the extension of the probabilistic modelling of solution uncertainty to deal with the MBVP and DIFP. Algorithm 3 provides a Metropolis-Hastings implementation of the Markov chain Monte Carlo algorithm from the two state mixed boundary value problem (A.2) with boundary constraint $[v(t=0), u(t=L)] - [v^*(t=0), u^*(t=L)] = [0, 0]$. Since these models may be subject to solution multiplicity, Algorithm 4 provides the corresponding parallel tempering (Geyer, 1991) implementation to permit efficient exploration of the space of $u(t=0)$. Algorithm 5 describes forward simulation for the delay initial function problem (A.3). In the context of the inverse problem, Algorithm 6 describes a Metropolis-Hastings implementation of the Markov chain Monte Carlo sampler for drawing realizations from the posterior distribution of the unknowns in the JAK-STAT system problem described in Section 4.1. Algorithm 7 describes its corresponding parallel tempering implementation. Algorithm 8 constructs a probabilistic solution for the heat equation PDE boundary value problem (5.6).

d

Algorithm 3 For the ODE mixed boundary value problem (A.2), draw K samples from the forward model conditional on boundary values $u^*(t=L), v^*(t=0), \theta, \Psi$, and a discretization grid $\mathbf{s} = (s_1, \cdots, s_N)$

At time $s_1 = 0$ initialize the unknown boundary value $u(t=0) \sim \pi(\cdot)$ by sampling from the prior π;
Use Algorithm 1 to conditionally simulate $(\mathbf{u}; \mathbf{v}) = (u(t_1), \ldots, u(t_T); v(t_1), \ldots, v(t_T))$ from the probabilistic solution of model (A.1) given $u(t=0), v^*(t=0), \theta, \Psi$;
for $k = 1 : K$ **do**
  Propose unknown boundary value $u'(t=0) \sim q\{\cdot \mid u(t=0)\}$ from a proposal density q;
  Use Algorithm 1 to conditionally simulate the probabilistic solution of model (A.1), $(\mathbf{u}'; \mathbf{v}')$, given $u'(t=0), v^*(t=0), \theta, \Psi$;
  Compute the rejection ratio,

$$\rho = \frac{q\{u(t=0) \mid u'(t=0)\}}{q\{u'(t=0) \mid u(t=0)\}} \frac{\pi\{u(t=0)\}}{\pi\{u'(t=0)\}} \frac{p\{u(t=L) \mid u^*(t=L)\}}{p\{u'(t=L) \mid u^*(t=L)\}};$$

  if $\min\{1, \rho\} > \mathrm{U}[0,1]$ **then**
    Update $u(t=0) \leftarrow u'(t=0)$;
    Update $(\mathbf{u}; \mathbf{v}) \leftarrow (\mathbf{u}'; \mathbf{v}')$;
  end if
  Return $(\mathbf{u}; \mathbf{v})$.
end for

Algorithm 4 Parallel tempering implementation of Algorithm 3 with C chains.

Choose the probability ξ of performing a swap move between two randomly chosen chains at each iteration, and define a temperature vector $\gamma \in (0,1]^C$, where $\gamma_C = 1$;
At time $s_1 = 0$ initialize the unknown boundary value $u(t=0)_{(1)} \sim \pi(\cdot)$ by sampling from the prior π and set $u(t=0)_{(c)} = u(t=0)_{(1)}$ for $c = 2, \ldots, C$;
Use Algorithm 1 to conditionally simulate $(\mathbf{u};\mathbf{v})_{(1)}$ from the probabilistic solution of model (A.1) given $u(t=0)_{(1)}, v^*(t=0), \theta, \Psi$ and set $(\mathbf{u};\mathbf{v})_{(c)} = (\mathbf{u};\mathbf{v})_{(1)}$ for $c = 2, \ldots, C$;
for $k = 1 : K$ **do**
  if $\xi > \mathrm{U}[0,1]$ **then**
    Propose a swap between $i, j \sim q(i,j)$, $1 \le i, j \le C, i \ne j$, where q is a proposal.
    Compute the rejection ratio,

$$\rho = \frac{p\left\{u(t=L)_{(i)} \mid u^*(t=L)\right\}^{\gamma_i}}{p\left\{u(t=L)_{(i)} \mid u^*(t=L)\right\}^{\gamma_j}} \frac{p\left\{u(t=L)_{(j)} \mid u^*(t=L)\right\}^{\gamma_j}}{p\left\{u(t=L)_{(j)} \mid u^*(t=L)\right\}^{\gamma_i}};$$

    if $\min(1,\rho) > \mathrm{U}[0,1]$ **then**
      Swap initial conditions $u(t=0)_{(i)}$ and $u(t=0)_{(j)}$;
    end if
  end if
  for $c = 1 : C$ **do**
    Perform one iteration of Metropolis-Hastings Algorithm 3, using instead a tempered likelihood to compute the rejection ratio:

$$\rho = \frac{q\left\{u(t=0)_{(c)} \mid u'(t=0)_{(c)}\right\}}{q\left\{u'(t=0)_{(c)} \mid u(t=0)_{(c)}\right\}} \frac{\pi\left\{u(t=0)_{(c)}\right\}}{\pi\left\{u'(t=0)_{(c)}\right\}} \frac{p\left\{u(t=L)_{(c)} \mid u^*(t=L)\right\}^{\gamma_c}}{p\left\{u'(t=L)_{(c)} \mid u^*(t=L)\right\}^{\gamma_c}},$$

    with temperature γ_c;
  end for
  Return $(\mathbf{u};\mathbf{v})_{(C)}$.
end for

Algorithm 5 For the ODE delay initial function problem (A.3), draw one sample from the forward model over grid $\mathbf{t} = (t_1, \cdots, t_T)$ given ϕ, τ, θ, Ψ, and a discretization grid $\mathbf{s} = (s_1, \cdots, s_N)$

At time $s_1 = 0$, sample $u^0(0-\tau) \sim \phi(0-\tau)$ and initialize the derivative $\mathrm{f}_1 = f\left\{s_1, u^*(0), u^0(0-\tau), \theta\right\}$ and define m^0, m_t^0, C^0, C_t^0 as in Section 2;

for $n = 1 : N$ **do**

If $n = 1$, set $g_1 = C_t^0(s_1, s_1) + C_t^0(s_1 - \tau, s_1 - \tau) + 2C_t^0(s_1, s_1 - \tau)$, otherwise set $g_n = C_t^{n-1}(s_n, s_n) + C_t^{n-1}(s_n - \tau, s_n - \tau) + 2C_t^{n-1}(s_n, s_n - \tau) + r_{n-1}(s_n)$;

Compute for each system component,

$$
\begin{aligned}
m^n(\mathbf{s}) &= m^{n-1}(\mathbf{s}) + g_n^{-1} \int_0^{\mathbf{s}} C_t^{n-1}(z, s_n)\mathrm{d}z \left\{\mathrm{f}_n - m_t^{n-1}(s_n)\right\}, \\
m^n(\mathbf{s}-\tau) &= m^{n-1}(\mathbf{s}-\tau) + g_n^{-1} \int_0^{\mathbf{s}-\tau} C_t^{n-1}(z, s_n)\mathrm{d}z \left\{\mathrm{f}_n - m_t^{n-1}(s_n)\right\}, \\
m_t^n(\mathbf{s}) &= m_t^{n-1}(\mathbf{s}) + g_n^{-1}\, C_t^{n-1}(\mathbf{s}, s_n) \left\{\mathrm{f}_n - m_t^{n-1}(s_n)\right\}, \\
C^n(\mathbf{s}, \mathbf{s}) &= C^{n-1}(\mathbf{s}, \mathbf{s}) - g_n^{-1} \int_0^{\mathbf{s}} C_t^{n-1}(z, s_n)\mathrm{d}z \left\{\int_0^{\mathbf{s}} C_t^{n-1}(z, s_n)\mathrm{d}z\right\}^\top, \\
C_t^n(\mathbf{s}, \mathbf{s}) &= C_t^{n-1}(\mathbf{s}, \mathbf{s}) - g_n^{-1} C_t^{n-1}(\mathbf{s}, s_n)\, C_t^{n-1}(s_n, \mathbf{s}), \\
\int_0^{\mathbf{s}} C_t^n(z, \mathbf{s})\mathrm{d}z &= \int_0^{\mathbf{s}} C_t^{n-1}(z, \mathbf{s})\mathrm{d}z - g_n^{-1} \int_0^{\mathbf{s}} C_t^{n-1}(z, s_n)\mathrm{d}z\, C_t^{n-1}(s_n, \mathbf{s}), \\
\int_0^{\mathbf{s}-\tau} C_t^n(z, \mathbf{s})\mathrm{d}z &= \int_0^{\mathbf{s}-\tau} C_t^{n-1}(z, \mathbf{s})\mathrm{d}z - g_n^{-1} \int_0^{\mathbf{s}-\tau} C_t^{n-1}(z, s_n)\mathrm{d}z\, C_t^{n-1}(s_n, \mathbf{s});
\end{aligned}
$$

if $n < N$ **then**

Sample step ahead realization $u^n(s_{n+1})$ from the predictive distribution of the state,

$$
p\left\{u(s_{n+1}) \mid \mathrm{f}_n, \Psi\right\} = \mathcal{N}\left\{u(s_{n+1}) \mid m^n(s_{n+1}), C^n(s_{n+1}, s_{n+1})\right\};
$$

if $s_{n+1} - \tau < 0$ **then**

Sample $u^n(s_{n+1} - \tau) \sim \phi(s_{n+1} - \tau)$,

else

Sample $u^n(s_{n+1} - \tau) \sim \mathcal{N}\left\{m^n(s_{n+1} - \tau), C^n(s_{n+1} - \tau, s_{n+1} - \tau)\right\}$;

end if

Interrogate the model by computing $\mathrm{f}_{n+1} = f\left\{s_{n+1}, u^n(s_{n+1}), u^n(s_{n+1} - \tau), \theta\right\}$;

end if

end for

Return $\mathbf{u} = (u(t_1), \cdots, u(t_T)) \sim \mathcal{GP}\left\{m^N(\mathbf{t}), C^N(\mathbf{t}, \mathbf{t})\right\}$, where $\mathbf{t} \subset \mathbf{s}$.

Algorithm 6 Draw K samples from (1.2) with density $p\{\theta, \tau, \phi, \alpha, \lambda, \mathbf{u} \mid \mathbf{y}, \Sigma\}$ given observations of the transformed solution of delay initial function problem (A.3).

Initialize $\theta, \tau, \phi, \alpha, \lambda \sim \pi(\cdot)$ where π is the prior density;
Use Algorithm 5 to conditionally simulate a realization, $\mathbf{u} = (u(t_1), \ldots, u(t_T))$, from the probabilistic solution of ODE delay initial function problem (A.3) given θ, τ, ϕ, and discretization grid $\mathbf{s} = (s_1, \cdots, s_N)$;
for $k = 1 : K$ **do**
  Propose $\theta', \tau', \phi' \sim q(\cdot \mid \theta, \tau, \phi)$ where q is a proposal density;
  Use Algorithm 5 to conditionally simulate a vector of realizations, $\mathbf{u}'$, from the forward model given $\theta', \tau', \phi', \alpha, \lambda$;
  Compute the rejection ratio,

$$\rho = \frac{q(\,\theta\,,\tau,\phi \mid \theta',\tau',\phi'\,)}{q(\,\theta'\,,\tau',\phi' \mid \theta,\tau,\phi\,)}\,\frac{\pi(\,\theta,\tau,\phi\,)}{\pi(\,\theta'\,,\tau',\phi'\,)}\,\frac{p\,(\,\mathbf{y} \mid \mathbf{u},\theta,\Sigma\,)}{p\,(\,\mathbf{y} \mid \mathbf{u}',\theta',\Sigma\,)};$$

  if $\min(1, \rho) > \mathrm{U}[0, 1]$ **then**
    Update $(\theta, \tau, \phi) \leftarrow (\theta'\,, \tau', \phi')$;
    Update $\mathbf{u} \leftarrow \mathbf{u}'$;
  end if
  Propose $\alpha', \lambda' \sim q(\cdot \mid \alpha, \lambda)$ where q is a proposal density;
  Use Algorithm 5 to conditionally simulate a vector of realizations, $\mathbf{u}'$, of the associated probabilistic solution for the ODE delay initial function problem given $\theta, \tau, \phi, \alpha', \lambda'$, and discretization grid $\mathbf{s} = (s_1, \cdots, s_N)$;
  Compute the rejection ratio,

$$\rho = \frac{q(\alpha, \lambda \mid \alpha', \lambda')}{q(\alpha', \lambda' \mid \alpha, \lambda)}\,\frac{\pi(\alpha, \lambda)}{\pi(\alpha', \lambda')}\,\frac{p\,(\,\mathbf{y} \mid \mathbf{u}, \theta, \Sigma\,)}{p\,(\,\mathbf{y} \mid \mathbf{u}', \theta, \Sigma\,)};$$

  if $\min(1, \rho) > \mathrm{U}[0, 1]$ **then**
    Update $(\alpha, \lambda) \leftarrow (\alpha', \lambda')$;
    Update $\mathbf{u} \leftarrow \mathbf{u}'$;
  end if
  Return $(\theta, \tau, \phi, \alpha, \lambda, \mathbf{u})$.
end for

Algorithm 7 Parallel tempering implementation of Algorithm 6

Define probability ξ of performing a swap move between two randomly chosen chains at each iteration, and define a temperature vector $\gamma \in (0,1]^C$, where $\gamma_C = 1$;
Initialize $(\theta,\tau,\phi,\alpha,\lambda)_{(1)} \sim \pi(\cdot)$ where π is the prior density and set $(\theta,\tau,\phi,\alpha,\lambda)_{(c)} = (\theta,\tau,\phi,\alpha,\lambda)_{(1)}$ for $c = 2,\dots,C$;
Use Algorithm 5 to conditionally simulate a realization, $\mathbf{u}_{(1)} = (u(t_1),\dots,u(t_T))_{(1)}$, from the forward model and set $\mathbf{u}_{(c)} = \mathbf{u}_{(1)}$ for $c = 2,\dots,C$;
for $k = 1 : K$ **do**
  if $\xi > \mathrm{U}[0,1]$ **then**
    Propose a swap between $i,j \sim q(i,j)$, $1 \le i,j \le C, i \ne j$, where q is a proposal.
    Compute the rejection ratio,

$$\rho = \frac{p\left(\mathbf{y} \mid \mathbf{u}_{(i)},\theta_{(i)},\Sigma\right)^{\gamma_i}}{p\left(\mathbf{y} \mid \mathbf{u}_{(i)},\theta_{(i)},\Sigma\right)^{\gamma_j}} \cdot \frac{p\left(\mathbf{y} \mid \mathbf{u}_{(j)},\theta_{(j)},\Sigma\right)^{\gamma_j}}{p\left(\mathbf{y} \mid \mathbf{u}_{(j)},\theta_{(j)},\Sigma\right)^{\gamma_i}};$$

    if $\min(1,\rho) > \mathrm{U}[0,1]$ **then**
      Swap parameter vectors $(\theta,\tau,\phi,\alpha,\lambda)_{(i)} \leftrightarrow (\theta,\tau,\phi,\alpha,\lambda)_{(j)}$;
    end if
  end if

  for $c = 1 : C$ **do**
    Perform one iteration of Metropolis-Hastings Algorithm 6, using instead a tempered likelihood to compute the rejection ratios:

$$\rho = \frac{q(\theta_{(c)},\tau_{(c)},\phi_{(c)} \mid \theta'_{(c)},\tau'_{(c)},\phi'_{(c)})}{q(\theta'_{(c)},\tau'_{(c)},\phi'_{(c)} \mid \theta_{(c)},\tau_{(c)},\phi_{(c)})}\,\frac{\pi(\theta_{(c)},\tau_{(c)},\phi_{(c)})}{\pi(\theta'_{(c)},\tau'_{(c)},\phi'_{(c)})} \cdot \frac{p\left(\mathbf{y} \mid \mathbf{u}_{(c)},\theta_{(c)},\Sigma\right)^{\gamma_c}}{p\left(\mathbf{y} \mid \mathbf{u}'_{(c)},\theta'_{(c)},\Sigma\right)^{\gamma_c}},$$

    and,

$$\rho = \frac{q(\alpha_{(c)},\lambda_{(c)} \mid \alpha'_{(c)},\lambda'_{(c)})}{q(\alpha'_{(c)},\lambda'_{(c)} \mid \alpha_{(c)},\lambda_{(c)})}\,\frac{\pi(\alpha_{(c)},\lambda_{(c)})}{\pi(\alpha'_{(c)},\lambda'_{(c)})}\,\frac{p\left(\mathbf{y} \mid \mathbf{u}_{(c)},\theta_{(c)},\Sigma\right)^{\gamma_c}}{p\left(\mathbf{y} \mid \mathbf{u}'_{(c)},\theta_{(c)},\Sigma\right)^{\gamma_c}},$$

    with temperature γ_c;
  end for
  Return $(\theta,\tau,\phi,\alpha,\lambda,\mathbf{u})_{(C)}$.
end for

Algorithm 8 Sampling from the formward model evaluated over the grid $\mathbf{X} \otimes \mathbf{T}$ for the heat equation (5.6), given κ, Ψ, N, M.

Define temporal discretization grid, $\mathbf{s} = (s_1, \cdots, s_N)$, and for each s_n define spatial discretization grid $\mathbf{z}_n = (z_{1,n}, \cdots, z_{M,n}), 1 \leq n \leq N$, and construct the $N \times M$ design matrix $\mathbf{Z} = (\mathbf{z}_1; \cdots; \mathbf{z}_N)$;

At time $s_1 = 0$ initialize the second spatial derivative using the boundary function and compute the temporal derivative $\mathbf{f}_1 = \kappa \left[u_{xx}(z_{1,1}, s_1), \cdots, u_{xx}(z_{M,1}, s_1)\right]$;

Define the prior covariance as in Section 5.4;

for $n = 1 : N$ **do**

If $n = 1$, set $G_1 = C_t^0([\mathbf{z_1}, s_1], [\mathbf{z_1}, s_1])$, otherwise set $G_n = C_t^{n-1}([\mathbf{z}_n, s_n], [\mathbf{z}_n, s_n]) + r_{n-1}\left([\mathbf{z}_n, s_n]\right)$;

Compute,

$$
\begin{aligned}
m_{xx}^n(\mathbf{Z}, \mathbf{s}) &= m_{xx}^{n-1}(\mathbf{Z}, \mathbf{s}) + \frac{\partial^2}{\partial \mathbf{Z}^2} \int_0^{\mathbf{s}} C_t^{n-1}([\mathbf{Z}, t], [\mathbf{z}_n, s_n]) \mathrm{d}t \, G_n^{-1} \left\{ \mathbf{f}_n - m_t^{n-1}(\mathbf{z}_n, s_n) \right\}, \\
m_t^n(\mathbf{Z}, \mathbf{s}) &= m_t^{n-1}(\mathbf{Z}, \mathbf{s}) + C_t^{n-1}([\mathbf{Z}, \mathbf{s}], [\mathbf{z}_n, s_n]) \, G_n^{-1} \left\{ \mathbf{f}_n - m_t^{n-1}(\mathbf{z}_n, s_n) \right\}, \\
C_{xx}^n([\mathbf{Z}, \mathbf{s}], [\mathbf{Z}, \mathbf{s}]) &= C_{xx}^{n-1}([\mathbf{Z}, \mathbf{s}], [\mathbf{Z}, \mathbf{s}]) \\
&\quad - \left\{ \frac{\partial^2}{\partial \mathbf{Z}^2} \int_0^{\mathbf{s}} C_t^{n-1}([\mathbf{Z}, t], [\mathbf{z}_n, s_n]) \mathrm{d}t \right\} G_n^{-1} \left\{ \frac{\partial^2}{\partial \mathbf{Z}^2} \int_0^{\mathbf{s}} C_t^{n-1}([\mathbf{Z}, t], [\mathbf{z}_n, s_n]) \mathrm{d}t \right\}^\top, \\
\frac{\partial^2}{\partial \mathbf{Z}^2} \int_0^{\mathbf{s}} C_t^n([\mathbf{Z}, t], [\mathbf{Z}, \mathbf{s}]) \mathrm{d}t &= \frac{\partial^2}{\partial \mathbf{Z}^2} \int_0^{\mathbf{s}} C_t^{n-1}([\mathbf{Z}, t], [\mathbf{Z}, \mathbf{s}]) \mathrm{d}t \\
&\quad - \left\{ \frac{\partial^2}{\partial \mathbf{Z}^2} \int_0^{\mathbf{s}} C_t^{n-1}([\mathbf{Z}, t], [\mathbf{z}_n, s_n]) \right\} \mathrm{d}t \, G_n^{-1} C_t^{n-1}([\mathbf{z}_n, s_n], [\mathbf{Z}, \mathbf{s}]), \\
C_t^n([\mathbf{Z}, \mathbf{s}], [\mathbf{Z}, \mathbf{s}]) &= C_t^{n-1}([\mathbf{Z}, \mathbf{s}], [\mathbf{Z}, \mathbf{s}]) - C_t^{n-1}([\mathbf{Z}, \mathbf{s}], [\mathbf{z}_n, s_n]) \, G_n^{-1} C_t^{n-1}([\mathbf{z}_n, s_n], [\mathbf{Z}, \mathbf{s}]);
\end{aligned}
$$

if $n < N$ **then**

Sample one-time-step-ahead realization of the second spatial derivative of the state, $u_{xx}(\mathbf{z}_n, s_{n+1})$ from the predictive distribution,

$$
\begin{aligned}
&p\left([u_{xx}(z_{1,n}, s_{n+1}), \cdots, u_{xx}(z_{M,n}, s_{n+1})] \mid \mathbf{f}_n, \kappa, \Psi\right) \\
&= \mathcal{N}\left([u_{xx}(z_{1,n}, s_{n+1}), \cdots, u_{xx}(z_{M,n}, s_{n+1})] \mid m_{xx}^n(\mathbf{z}_n, s_{n+1}), C_{xx}^n([\mathbf{z}_n, s_{n+1}], [\mathbf{z}_n, s_{n+1}])\right);
\end{aligned}
$$

and interrogate the PDE model by computing $\mathbf{f}_{n+1} = \kappa \left[u_{xx}(z_{1,n+1}, s_{n+1}), \cdots, u_{xx}(z_{M,n+1}, s_{n+1})\right]$

end if

end for

Return $\mathbf{U} = \begin{bmatrix} u(x_{1,1}, t_1) & \cdots & u(x_{X,1}, t_1) \\ \vdots & \vdots & \vdots \\ u(x_{1,1}, t_T) & \cdots & u(x_{X,T}, t_T) \end{bmatrix} \sim \mathcal{GP}\left(m^N(\mathbf{X}, \mathbf{T}), C^N([\mathbf{X}, \mathbf{T}], [\mathbf{X}, \mathbf{T}])\right)$,

where $\mathbf{X} \subset \mathbf{Z}$ and $\mathbf{T}_{\cdot,1} \subset \mathbf{s}$.

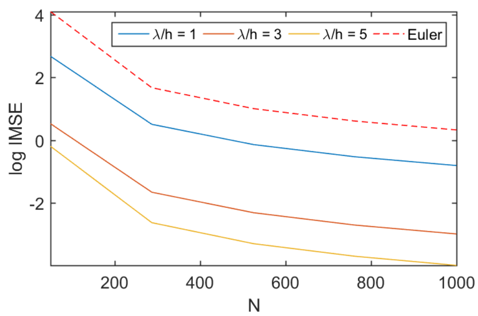

Figure 10: Left: log integrated mean squared error is shown for the one-step Euler method (dashed line) and for the mean one-step-ahead probabilistic solution (solid lines) against the discretization grid size (N) for different length-scales expressed as proportions of the time step. Right: log computational time against grid size is shown for one iteration of the one-step Euler method (red) and 100 iterations of the probabilistic one-step-ahead solver with squared exponential covariance (blue).

Appendix C: Computational Comparison

Computational comparison between a one-step-ahead probabilistic algorithm (UQDES-1) and the one-step explicit Euler method (Euler) are provided under difference discretization grids and regimes for the initial value ODE problem (2.3). As discussed in the paper, a probabilistic formalism can yield higher order algorithms by varying the sampling scheme. Therefore, comparison to higher-order numerical methods, such as Runge-Kutta, were outside the scope of our paper. Figure 10 shows the computation times as well as integrated log mean square error (IMSE) for each choice of solver. The improvement in log IMSE under appropriate length-scale specification illustrates the advantage of the probabilistic method within the inverse problem, where this hyperparameter can be estimated.

Appendix D: Derivations and Proofs

D.1 Probabilistic solution as latent function estimation

We take a process convolution view of the model presented in Section 3. Let $\mathcal{F}$ be the space of square-integrable random functions and $\mathcal{F}^*$ be its dual space of linear functionals. The solution and its derivative will be modelled by an integral transform using the linear continuous operators R and Q defining a mapping from $\mathcal{F}$ to $\mathcal{F}^*$. The associated kernels are the deterministic, square-integrable function $R_\lambda : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and its integrated version $Q_\lambda(t_j, t_k) = \int_0^{t_k} R_\lambda(z, t_k)\mathrm{d}z$. The operators $R, Q, R^\dagger, Q^\dagger$ are defined, for $u \in \mathcal{F}$ and $v \in \mathcal{F}^*$, as $Ru(t) = \int R_\lambda(t, z)u(z)\mathrm{d}z$ and $Qu(t) = \int Q_\lambda(t, z)u(z)\mathrm{d}z$, with adjoints $R^\dagger v(t) = \int R_\lambda(z, t)v(z)\mathrm{d}z$ and $Q^\dagger v(t) = \int Q_\lambda(z, t)v(z)\mathrm{d}z$ respectively.

Estimating the solution of a differential equation model may be restated as a problem of inferring an underlying latent process $\zeta \in \mathcal{F}$ from noisy realizations. We consider the white noise process, $\zeta \sim \mathcal{N}(0, K)$, with covariance $K(t_j, t_k) = \alpha^{-1}\delta_{t_j}(t_k)$. We model the derivative of the solution as the integral transform,

$$u_t(t_j) = m_t^n(t_j) + R\zeta(t_j), \quad t_j \in [0, L],\, L > 0,\, 0 \leq n \leq N. \tag{D.1}$$

The differential equation solution model at time $t_j \leq b$ is then obtained by integrating the derivative u_t with respect to time over the interval $[0, t_j]$ as follows,

$$u(t_j) = \int_0^{t_j} u_t(z)\mathrm{d}z = m^n(t_j) + Q\zeta(t_j), \quad t_j \in [0, L],\, L > 0,\, 0 \leq n \leq N. \tag{D.2}$$

Let $\mathbf{f}_{1:n} = (\mathrm{f}_1, \ldots, \mathrm{f}_n)$ and omit dependence on θ and Ψ, which are fixed. We wish to update the prior $[u \mid \mathbf{f}_{1:n-1}]$ given a new model interrogation, f_n, to obtain the updated process $[u \mid \mathbf{f}_{1:n}]$, where $1 \leq n \leq N$.

Lemma D.1 (Updating)**.** *The stochastic process* $\{u(t) \mid \mathbf{f}_{1:n-1}, t \in [0, L]\}$ *and its time derivative* $\{u_t(t) \mid \mathbf{f}_{1:n-1}, t \in [0, L]\}$ *are well-defined and distributed according to a Gaussian probability measure with marginal mean functions, covariance and cross-covariance operators given by,*

$$\begin{aligned}
m^n(t_j) &= m^{n-1}(t_j) + g_n^{-1}\left\{f_n - m_t^{n-1}(s_n)\right\} \int_0^{t_j} C_t^{n-1}(z, s_n)\,dz,\\
m_t^n(t_j) &= m_t^{n-1}(t_j) + g_n^{-1}\left\{f_n - m_t^{n-1}(s_n)\right\}\, C_t^{n-1}(t_j, s_n),\\
C^n(t_j, t_k) &= C^{n-1}(t_j, t_k) - g_n^{-1} \int_0^{t_j} C_t^{n-1}(z, s_n)\,dz \int_0^{t_k} C_t^{n-1}(s_n, z)\,dz,\\
C_t^n(t_j, t_k) &= C_t^{n-1}(t_j, t_k) - g_n^{-1} C_t^{n-1}(t_j, s_n)\, C_t^{n-1}(s_n, t_k),\\
\int_0^{t_j} C_t^n(z, t_k)\,dz &= \int_0^{t_j} C_t^{n-1}(z, t_k)\,dz - g_n^{-1} \int_0^{t_j} C_t^{n-1}(z, s_n)\,dz\, C_t^{n-1}(s_n, t_k),\\
\int_0^{t_k} C_t^n(t_j, z)\,dz &= \left\{\int_0^{t_k} C_t^n(z, t_j)\,dz\right\}^\top
\end{aligned}$$

where,

$$g_n := \begin{cases} C_t^0(s_1, s_1) & \text{if } n = 1,\\ C_t^{n-1}(s_n, s_n) + r_{n-1}(s_n) & \text{if } n > 1, \end{cases}$$

and where m^0 *and* m_t^0 *are the prior means and* C^0 *and* C_t^0 *the prior covariances of the state and derivatives defined in Section 2.*

Proof. We are interested in the conditional distribution of the state $u(t) - m^{n-1}(t) \in \mathcal{F}^*$ and time derivative $u_t(t) - m_t^{n-1}(t) \in \mathcal{F}^*$ given a new model interrogation on a mesh vertex $s_n \in [0, L], L > 0$ under the Gaussian error model,

$$\mathrm{f}_n - m_t^{n-1}(s_n) = R\zeta(s_n) + \eta(s_n),$$

where $\eta(s_n) \sim \mathcal{N}(0, \gamma_n)$.

Construct the vector

$$\left(u_t - m_t^{n-1}, \mathrm{f}_n - m_t^{n-1}(s_n)\right) = (R\zeta, R\zeta(s_n) + \eta(s_n)) \in \mathcal{F}^* \oplus \mathbb{R},$$

where the first element is function-valued and the second element is a scalar. This vector is jointly Gaussian with mean $M = (0, 0)$ and covariance operator C with positive definite cross-covariance operators,

$$\begin{array}{ll} C_{11} = RKR^\dagger & C_{12} = RKR^\dagger \\ C_{21} = RKR^\dagger & C_{22} = RKR^\dagger + \gamma_n. \end{array} \tag{D.3}$$

Since both $\mathcal{F}^*$ and $\mathbb{R}$ are separable Hilbert spaces, it follows from Theorem 6.20 in Stuart (2010) that the random variable $[u_t - m_t^{n-1} \mid \mathrm{f}_n - m_t^{n-1}(s_n)]$ is well-defined and distributed according to a Gaussian probability measure with mean and covariance,

$$\begin{aligned} \mathrm{E}\left(u_t - m_t^{n-1} \mid \mathrm{f}_n - m_t^{n-1}(s_n)\right) &= C_{12}C_{22}^{-1}(\mathrm{f}_n - m_t^{n-1}(s_n)), \\ \mathrm{Cov}\left(u_t - m_t^{n-1} \mid \mathrm{f}_n - m_t^{n-1}(s_n)\right) &= C_{11} - C_{12}C_{22}^{-1}C_{21}. \end{aligned}$$

Similarly, consider the vector $\left(u - m^{n-1}, \mathrm{f}_n - m_t^{n-1}(s_n)\right) = (Q\zeta, R\zeta(s_n) + \eta(s_n)) \in \mathcal{F}^* \oplus \mathbb{R}$, with mean $M = (0, 0)$ and cross-covariances,

$$\begin{array}{ll} C_{11} = QKQ^\dagger & C_{12} = QKR^\dagger \\ C_{21} = RKQ^\dagger & C_{22} = RKR^\dagger + \gamma_n. \end{array}$$

By Theorem 6.20 (Stuart, 2010), the random variable $[u - m^{n-1} \mid \mathrm{f}_n - m_t^{n-1}(s_n)]$ is well-defined and distributed according to a Gaussian probability measure with mean and covariance,

$$\begin{aligned} \mathrm{E}\left(u - m^{n-1} \mid \mathrm{f}_n - m_t^{n-1}(s_n)\right) &= C_{12}C_{22}^{-1}(\mathrm{f}_n - m_t^{n-1}(s_n)), \\ \mathrm{Cov}\left(u - m^{n-1} \mid \mathrm{f}_n - m_t^{n-1}(s_n)\right) &= C_{11} - C_{12}C_{22}^{-1}C_{21}. \end{aligned}$$

Cross-covariances are found analogously.

□

The kernel convolution approach to defining the covariances guarantees that the cross-covariance operators, (D.3), between the derivative and n derivative model realizations, are positive definite.

D.2 Proof of Theorem 3.1

Denote the exact solution satisfying (A.1) by $u^*(t)$. For clarity of exposition, we assume that $u(0) = 0$ and let $m^0(t) = 0$ for $t \in [0, L], L > 0$. We define $h = \max_{n=2,\ldots,N}(s_n - s_{n-1})$ to be the maximum step length between subsequent discretization grid points. We would like to show that the forward model $\{u(t) \mid N, t \in [0, L]\}$ (sampled via Algorithm 1) with $r_n(s) = 0, s \in [0, L]$ converges in L^1 to $u^*(t)$ as $h \to 0$ and $\lambda, \alpha^{-1} = O(h)$, concentrating at a rate proportional to h.

Let $\mathbf{f}_{1:n} = (\mathrm{f}_1, \dots, \mathrm{f}_n)$ and omit dependence on θ and Ψ, which are fixed. For $t \in [0, L]$ we find n such that $t \in [s_n, s_{n+1}]$, and wish to bound the expected absolute difference $\beta_n(t)$ between the exact solution and a single mixture component of (1.2). Let Φ be the standard normal cdf, and $A = \mathrm{E}\left(u(t) - u^*(t) \mid \mathbf{f}_{1:n}\right) / \sqrt{C^n(t,t)}$.

$$
\begin{aligned}
\beta_n(t) &= \mathrm{E}\left(|u(t) - u^*(t)| \mid \mathbf{f}_{1:n}\right) \\
&= \mathrm{E}\left(u(t) - u^*(t) \mid \mathbf{f}_{1:n}\right) \{1 - 2\Phi\left(-A\right)\} + \sqrt{\frac{2}{\pi}\, C^n(t,t)}\, \exp\left\{-A^2/2\right\}, \\
&\leq \left|\mathrm{E}\left(u(t) - u^*(t) \mid \mathbf{f}_{1:n}\right)\right| + O(h^2)
\end{aligned} \tag{D.4}
$$

The second equality uses the expectation of a folded normal distribution (Leone et al., 1961) and Lemma D.3. Next bound the first term of (D.4) as follows,

$$
\begin{aligned}
&\mathrm{E}\left(u(t) - u^*(t) \mid \mathbf{f}_{1:n}\right) \\
&= \mathrm{E}\left(u(s_n) - u^*(s_n) \mid \mathbf{f}_{1:n}\right) + (t - s_n)\mathrm{E}\left(u_t(s_n) - u_t^*(s_n) \mid \mathbf{f}_{1:n}\right) + O(h^2) \\
&= \mathrm{E}\left(u(s_n) - u^*(s_n) \mid \mathbf{f}_{1:n-1}\right) + \frac{C^{n-1}(s_n, s_n)}{C_t^{n-1}(s_n, s_n)}\mathrm{E}\left(\mathrm{f}_n - u_t(s_n) \mid \mathbf{f}_{1:n-1}\right) \\
&\quad + (t - s_n)\mathrm{E}\left(\mathrm{f}_n - u_t^*(s_n) \mid \mathbf{f}_{1:n-1}\right) + O(h^2).
\end{aligned} \tag{D.5}
$$

The nth updated process is mean-square differentiable on [0,L], so a Taylor expansion around s_n gives us the first equality, and the recursion from Lemma D.1 gives us the second equality. Next, we use Jensen's inequality to obtain,

$$
\beta_n(t) \leq \beta_{n-1}(s_n) + |t - s_n|\, \mathrm{E}\left(|\mathrm{f}_n - u_t^*(s_n)| \mid \mathbf{f}_{1:n-1}\right) + \tfrac{C^{n-1}(s_n,s_n)}{C_t^{n-1}(s_n,s_n)}\, \mathrm{E}\left(|\mathrm{f}_n - u_t(s_n)| \mid \mathbf{f}_{1:n-1}\right) + O(h^2).
$$

By the Lipschitz continuity of f, boundedness of $\mathrm{E}\left(|\mathrm{f}_n - u_t(s_n)| \mid \mathbf{f}_{1:n-1}\right)$, and recursive construction of the covariance, we obtain,

$$
\begin{aligned}
\beta_n(t) \quad &\leq \beta_{n-1}(s_n) + L|t - s_n|\beta_{n-1}(s_n) + O(h^2), \\
&= \beta_{n-1}(s_n)\left(1 + L|t - s_n|\right) + O(h^2).
\end{aligned}
$$

It can be shown (e.g. Butcher, 2008, p.67-68) that the following inequality holds:

$$
\beta_n(t) \leq \begin{cases} \beta_0(s_1) + hB(t-a), & L = 0, \\ \exp\{(t-a)L\}\beta_0(s_1) + \exp\{(t-a)L - 1\}hB/L, & L > 0, \end{cases}
$$

where B is the constant upper bound on all the remainders. This expression tends to 0 as $\alpha^{-1}, \lambda, h \to 0$, since $\beta_0(s_1) = 0$. Then, taking the expectation of $\beta_n(t)$ with respect to $\mathbf{f} = (\mathrm{f}_1, \dots, \mathrm{f}_N)$, we obtain,

$$
\mathrm{E}\left(|u(t) - u^*(t)| \mid N\right) = O(h), \quad \text{as } \alpha^{-1}, \lambda \to 0.
$$

Thus, the probabilistic solution (1.2) converges in L^1 to $u^*(t)$ at the rate $O(h)$. Note that the assumption that auxiliary parameters, λ and α^{-1}, associated with the solver tend to zero with the step size is analogous to maintaining a constant number of steps in a k-step numerical method regardless of the step size.

D.3 Properties of the covariance

In this section we present some results regarding covariance structures that are used in the proof of Theorem 3.1.

Lemma D.2. *For $1 < n \leq N$ and $t \in [0, L], L > 0$, the variances for the state and derivative obtained sequentially via Algorithm 1 satisfy:*

$$
\begin{aligned}
C^n(t,t) &\leq C^1(t,t),\\
C^n_t(t,t) &\leq C^1_t(t,t).
\end{aligned}
$$

Proof. We use the fact that $C^n_t(t,t) \geq 0$ for all n and the recurrence from Lemma D.1, we obtain:

$$
\begin{aligned}
C^n_t(t,t) &= C^{n-1}_t(t,t) - g_n^{-1} C^{n-1}_t(t,s_n)\, C^{n-1}_t(s_n,t),\\
&\leq C^{n-1}_t(t,t) \leq \cdots \leq C^1_t(t,t).
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
C^n(t,t) &= C^{n-1}(t,t) - g_n^{-1} \int_0^t C^{n-1}_t(z,s_n)\mathrm{d}z \int_0^t C^{n-1}_t(s_n,z)\mathrm{d}z,\\
&\leq C^{n-1}(t,t) \leq \cdots \leq C^1(t,t).
\end{aligned}
$$

□

Lemma D.3. *The covariances, $C^n(t_j,t_k)$ and $C^n_t(t_j,t_k)$, obtained at each step of Algorithm 1 tend to zero at the rate $O(h^4)$, as $h \to 0$ and $\lambda, \alpha^{-1} = O(h)$ if the covariance function R_λ is stationary and satisfies:*

$$
RR(t,t) - \frac{RR(t+d,t)RR(t+d,t)}{RR(t,t)} = O(h^4), \quad \lambda,\ \alpha^{-1} = O(h),\ h \to 0, \tag{D.6}
$$

$$
QQ(t,t) - \frac{QR(t+d,t)QR^\dagger(t+d,t)}{RR(t,t)} = O(h^4), \quad \lambda,\ \alpha^{-1} = O(h),\ h \to 0, \tag{D.7}
$$

where $d > 0$, $t, t+d \in [0, L]$, and $\lambda \geq h$.

Proof. From Lemma D.2 and assumption (D.6) we obtain,

$$
C^n_t(t,t) \leq C^1_t(t,t) = RR(t,t) - \frac{RR(t,s_1)RR(s_1,t)}{RR(s_1,s_1)} = O(h^4), \quad \lambda, \alpha^{-1} = O(h),\ h \to 0,\ t \in [0,L],\ 1 \leq n \leq N.
$$

Similarly, using Lemma D.2 and assumption (D.7) yields,

$$
C^n(t,t) \leq C^1(t,t) = QQ(t,t) - \frac{QR(t,s_1)QR^\dagger(t,s_1)}{RR(s_1,s_1)} = O(h^4), \quad \lambda, \alpha^{-1} = O(h),\ h \to 0,\ t \in [0,L],\ 1 \leq n \leq N.
$$

Then, by the Cauchy-Schwarz inequality,

$$
\left|C^n_t(t_j,t_k)\right|, \left|C^n(t_j,t_k)\right| = O(h^4),\ \lambda, \alpha^{-1} = O(h),\ h \to 0,\ t_j, t_k \in [0,L],\ 1 \leq n \leq N.
$$

□

The square exponential and uniform covariance functions considered in this paper are stationary and symmetric and satisfy conditions (D.6) and (D.7).

Lemma D.4. *The covariance $C_t^{n-1}(s_n, s_n)$, $1 \le n \le N$ in Algorithm 1, tends to zero at the rate $O(h^4)$ as $N \to \infty$ and $h \to 0$ when conditions (D.6) and (D.7) are satisfied.*

Proof. The proof follows immediately from Lemma D.3. □

D.4 Some covariance kernels and their convolutions

The examples in this paper utilize two types of covariance functions, although the results presented are more generally applicable. Imposing unrealistically strict smoothness assumptions on the state space by choice of covariance structure may introduce estimation bias if the exact solution is less smooth than expected. In this paper we work with stationary derivative covariance structures for simplicity, however there will be classes of problems where non-stationary kernels may be more appropriate.

The infinitely differentiable squared exponential covariance is based on the kernel $R_\lambda(t_j, t_k) = \exp\left\{-(t_j - t_k)^2/2\lambda^2\right\}$. We utilize this covariance structure in the forward models of the Lorenz63 system, the Navier-Stokes equations, the Lane-Emden mixed boundary value problem, and the heat equation. In contrast, solutions for systems of delay initial function problems are often characterized by second derivative discontinuities at the lag locations. For these forward models we utilize the uniform derivative covariance structure, obtained by convolving the kernel $R_\lambda(t_j, t_k) = 1_{(t_j-\lambda, t_j+\lambda)}(t_k)$ with itself, is non-differentiable. Given the shape of the uniform kernel, choosing the length-scale λ, to be greater than one half of the maximum step length ensures that the resulting step ahead prediction captures information from at least one previous model interrogation.

Closed form expressions for the pairwise convolutions for the two covariance functions are provided below and implemented in the accompanying software. Let R_λ be the squared exponential kernel and let Q_λ be its integrated version. Then,

$$
\begin{aligned}
\alpha RR(t_j, t_k) &= \sqrt{\pi}\lambda \exp\left\{-\tfrac{(t_j-t_k)^2}{4\lambda^2}\right\}, \\
\alpha QR(t_j, t_k) &= \pi\lambda^2 \left(\mathrm{erf}\left\{\tfrac{t_j-t_k}{2\lambda}\right\} + \mathrm{erf}\left\{\tfrac{t_k-a}{2\lambda}\right\}\right), \\
\alpha QQ(t_j, t_k) &= \pi\lambda^2 \left((t_j - a)\mathrm{erf}\left\{\tfrac{t_j-a}{2\lambda}\right\} - (t_k - t_j)\mathrm{erf}\left\{\tfrac{t_k-t_j}{2\lambda}\right\} + (t_k - a)\mathrm{erf}\left\{\tfrac{t_k-a}{2\lambda}\right\}\right), \\
&\quad + 2\sqrt{\pi}\lambda^3 \left(\exp\left\{-\tfrac{(t_j-a)^2}{4\lambda^2}\right\} - \exp\left\{-\tfrac{(t_k-t_j)^2}{4\lambda^2}\right\} + \exp\left\{-\tfrac{(t_k-a)^2}{4\lambda^2}\right\} - 1\right).
\end{aligned}
$$

Next, let R_λ be the uniform kernel and let Q_λ be its integrated version. Then,

$$
\begin{aligned}
\alpha RR(t_j,t_k) \quad &= \{\min(t_j,t_k) - \max(t_j,t_k) + 2\lambda\}\, 1_{(0,\infty)}\{\min(t_j,t_k) - \max(t_j,t_k) + 2\lambda\},\\
\alpha QR(t_j,t_k) \quad &= 2\lambda\{\min(t_j-\lambda, t_k+\lambda) - \max(a+\lambda, t_k-\lambda)\}\\
&\quad\; 1_{(0,\infty)}\{\min(t_j-\lambda, t_k+\lambda) - \max(a+\lambda, t_k-\lambda)\}\\
&+ (t_j - a)\{\min(t_j,t_k) - a - 2\lambda\}\, 1_{(0,\infty)}\{\min(t_j,t_k) - a - 2\lambda\}\\
&+ \left[(t_j+\lambda)\{\min(t_j,t_k) - \max(a+\lambda, t_j-\lambda, t_k-\lambda) + \lambda\}\right.\\
&\quad \left. - \tfrac{1}{2}\min(t_j+\lambda, t_k+\lambda)^2 + \tfrac{1}{2}\max(a+\lambda, t_j-\lambda, t_k-\lambda)^2\right]\\
&\quad\; 1_{(0,\infty)}\{\min(t_j,t_k) + \lambda - \max(a+\lambda, t_j-\lambda, t_k-\lambda)\}\\
&+ \left[(\lambda - a)\{\min(a+\lambda, t_j-\lambda, t_k+\lambda) - (t_k-\lambda)\}\right.\\
&\quad \left. + \tfrac{1}{2}\min(a+\lambda, t_j-\lambda, t_k+\lambda)^2 - \tfrac{1}{2}(t_k-\lambda)^2\right]\\
&\quad\; 1_{(0,\infty)}\{\min(a+\lambda, t_j-\lambda, t_k+\lambda) - (t_k-\lambda)\},\\
\alpha QQ(t_j,t_k) \quad &= 4\lambda^2\{\min(t_j,t_k) - a - 2\lambda\}\, 1_{(0,\infty)}\{\min(t_j,t_k) - a - 2\lambda\}\\
&+ 2\lambda\left[(t_k+\lambda)\{\min(t_j-\lambda, t_k+\lambda) - \max(a+\lambda, t_k-\lambda)\}\right.\\
&\quad \left. - \tfrac{1}{2}\min(t_j-\lambda, t_k+\lambda)^2 + \tfrac{1}{2}\max(a+\lambda, t_k-\lambda)^2\right]\\
&\quad\; 1_{(0,\infty)}\{\min(t_j-\lambda, t_k+\lambda) - \max(a+\lambda, t_k-\lambda)\}\\
&+ \left[\tfrac{1}{3}\min(a+\lambda, t_j-\lambda, t_k-\lambda)^3 - \tfrac{1}{3}(a-\lambda)^3\right.\\
&\quad + (\lambda-a)\{\min(a+\lambda, t_j-\lambda, t_k-\lambda)^2 - (a-\lambda)^2\}\\
&\quad \left. + (\lambda-a)^2\{\min(a+\lambda, t_j-\lambda, t_k-\lambda) - (a-\lambda)\}\right]\\
&\quad\; 1_{(0,\infty)}\{\min(a+\lambda, t_j-\lambda, t_k-\lambda) - (a-\lambda)\}\\
&+ (t_j - a)\left[\tfrac{1}{2}\min(a+\lambda, t_k-\lambda)^2 - \tfrac{1}{2}(t_j-\lambda)^2\right.\\
&\quad \left. + (\lambda-a)\{\min(a+\lambda, t_k-\lambda) - (t_j-\lambda)\}\right]\\
&\quad\; 1_{(0,\infty)}\{\min(a+\lambda, t_k-\lambda) - (t_j-\lambda)\}\\
&+ (t_k - a)\left[\tfrac{1}{2}\min(a+\lambda, t_j-\lambda)^2 - \tfrac{1}{2}(t_k-\lambda)^2\right.\\
&\quad \left. + (\lambda-a)\{\min(a+\lambda, t_j-\lambda) - (t_k-\lambda)\}\right]\, 1_{(0,\infty)}\{\min(a+\lambda, t_j-\lambda) - (t_k-\lambda)\}\\
&+ 2\lambda\left[(t_j+\lambda)\{\min(t_j+\lambda, t_k-\lambda) - \max(a+\lambda, t_j-\lambda)\}\right.\\
&\quad \left. - \tfrac{1}{2}\min(t_j+\lambda, t_k-\lambda)^2 + \tfrac{1}{2}\max(a+\lambda, t_j-\lambda)^2\right]\\
&\quad\; 1_{(0,\infty)}\{\min(t_j+\lambda, t_k-\lambda) - \max(a+\lambda, t_j-\lambda)\}\\
&+ \left[(t_j+\lambda)(t_k+\lambda)\{\min(t_j,t_k) + \lambda - \max(a+\lambda, t_j-\lambda, t_k-\lambda)\}\right.\\
&\quad - \tfrac{1}{2}(t_j+t_k+2\lambda)\{\min(t_j+\lambda, t_k+\lambda)^2 - \max(a+\lambda, t_j-\lambda, t_k-\lambda)^2\}\\
&\quad \left. + \tfrac{1}{3}\min(t_j+\lambda, t_k+\lambda)^3 - \tfrac{1}{3}\max(a+\lambda, t_j-\lambda, t_k-\lambda)^3\right]\\
&\quad\; 1_{(0,\infty)}\{\min(t_j,t_k) + \lambda - \max(a+\lambda, t_j-\lambda, t_k-\lambda)\}\\
&+ (t_j - a)(t_k - a)\{a + 2\lambda - \max(t_j,t_k)\}\, 1_{(0,\infty)}\{a + 2\lambda - \max(t_j,t_k)\}.
\end{aligned}
$$

References

Bellen, A. and Zennaro, M. (2003). *Numerical Methods for Delay Differential Equations*. Clarendon Press. 2

Butcher, J. (2008). *Numerical Methods for Ordinary Differential Equations*. John Wiley and Sons Ltd. 1, 12

Coddington, A. and Levinson, N. (1955). *Theory of ordinary differential equations*. International series in pure and applied mathematics. McGraw-Hill. 1

Geyer, C. (1991). "Markov chain Monte Carlo maximum likelihood." In *Computing Science and Statistics, Proceedings of the 23rd Symposium on the Interface, 156*. American Statistical Association. 2

Keller, H. B. (1968). *Numerical Methods for Two-point Boundary-value Problems*. Blaisdell Publishing Company. 1

Leone, F. C., Nelson, L. S., and Nottingham, R. B. (1961). "The folded Normal distribution." *Technometrics*, 3: 543–550. 12

Polyanin, A. D. and Zaitsev, V. F. (2004). *Handbook of Nonlinear Partial Differential Equations*. Chapman and Hall, CRC Press. 2

Stuart, A. M. (2010). "Inverse problems: A Bayesian perspective." *Acta Numerica*, 19: 451–559. 11