On the Computation of Continuous, Error-Controlled Numerical Solutions of Nonlocal Parabolic Partial Differential Equations via Reformulation*

Graeme Fairweather[†] Victoria Guenter[‡]
Paul Muir[§] Raymond J. Spiteri[¶]

Abstract

Many problems arising in science and engineering are modeled by *nonlocal* initial–boundary value problems (IBVPs) in one space variable. In such a problem, the governing partial differential equation (PDE) and/or boundary conditions involve spatial integrals of the solution. Although high-quality software exists for solving standard IBVPs, it cannot handle such nonlocal problems directly. In this report, a reformulation technique, previously applied to nonlocal two-point boundary value problems, is extended to nonlocal IBVPs to enable their solution to a user-prescribed accuracy using a state-of-the-art error control software package. The efficacy of this approach is demonstrated using several nonlocal problems from the literature.

Subject Classification: 35K20, 45K05, 65M20, 65M50, 65M70
Keywords: Nonlocal parabolic initial–boundary value problems, Reformulation techniques, Adaptive B-spline collocation, Error-controlled numerical solutions.

---

*This work was supported by the Natural Sciences and Engineering Research Council of Canada and the Cooperative Institute for Research to Operations in Hydrology (CIROH) with funding under award NA22NWS4320003 from the NOAA Cooperative Institute Program. The statements, findings, conclusions, and recommendations are those of the author(s) and do not necessarily reflect the opinions of NOAA.

[†]Mathematical Reviews, American Mathematical Society, Ann Arbor, Michigan 48103, U.S.A. gxf@umich.edu

[‡]Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada, S7N 5C9. victoria.g@usask.ca

[§]Department of Mathematics and Computing Science, Saint Mary's University, Halifax, NS, Canada, B3H 3C3. muir@smu.ca

[¶]Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada, S7N 5C9. spiteri@cs.usask.ca

1 Introduction

Many problems arising in science and engineering are modeled by two-point boundary value problems for ordinary differential equations (TPBVPs) or initial–boundary value problems (IBVPs) for parabolic partial differential equations which are *nonlocal*. In such a problem, the governing differential equation or boundary conditions involve a spatial integral that depends on the solution of the problem. Applications involving nonlocal IBVPs arise in various areas such as thermo-elasticity, thermodynamics, heat conductivity, plasma physics, chemical diffusion, population dynamics, and nuclear reactor dynamics.

There exist several high-quality software packages for the numerical solution of TPBVPs, but none can solve *nonlocal* TPBVPs directly. In [27], several types of nonlocal TPBVPs from the literature are reformulated to transform them into a system of TPBVPs with no nonlocal elements. *The major advantage of this approach is that this reformulated system can then be solved to a user-prescribed accuracy using a state-of-the-art error control software package*, such as `COLNEWSC` [2]. This package is based on `COLNEW` [10] (which in turn is based on `COLSYS` [6, 7]) employs an adaptive B-spline collocation method. *It returns a continuous, error-controlled numerical solution. Error control means that the computation is adapted until a high-quality estimate of the error in the returned continuous numerical solution satisfies a given user-specified tolerance.*

The goal of this work is to extend the general approach of [27] to derive reformulations of nonlocal parabolic IBVPs in one space variable, such that the reformulated problem does not possess nonlocal elements. The reformulation leads to a coupled system of partial differential equations (PDEs), which can be solved in an error-controlled framework using the software package `eBACOLI-PE` (extended `BACOLI` to parabolic–elliptic equations), an updated version of `eBACOLI` [30] which is based on the earlier packages, `BACOL` [55] and `BACOLI` [43]. These software packages employ an adaptive B-spline collocation method for the spatial discretization of the PDE system coupled with the `DASSL` differential-algebraic equation (DAE) solver [42, 16, 17] which implements an adaptive error control algorithm based on a family of Backward Differentiation Formulas (BDFs) for the time integration. *An essential feature of `eBACOLI-PE` is that it returns a continuous, error-controlled numerical solution. The computation is adapted so that the returned continuous numerical solution has an error estimate that satisfies a given user-provided tolerance.* Additional details are provided in Section 2.

Numerous nonlocal parabolic IBVPs in one space variable appear in the literature. Many of these can be classified into two categories, the first of which comprises problems of the form

$$u_t(x,t) = u_{xx}(x,t) + f(x,t), \quad (x,t) \in [0,1] \times [0,T],$$

subject to the initial condition

$$u(x,0) = \phi(x), \quad x \in [0,1],$$

and the boundary conditions

$$u(0,t) = \int_0^1 k_0(x)\ u(s,t)ds + g_0(t), \quad u(1,t) = \int_0^1 k_1(x)\ u(s,t)ds + g_1(t),$$

where $t \in [0,T]$, f, ϕ, k_0, g_0, k_1, and g_1 are given functions, and T is the termination time. For the solution of such problems, various methods have been proposed including high-order finite difference methods [4, 5, 20, 28, 34, 49], a spectral method [54], a Sinc method [59], a homotopy perturbation method [29], Chebyshev collocation [53], a Ritz–Galerkin method [57], cubic spline collocation [36], and orthogonal spline collocation [13].

The second category, often referred to as nonlocal problems with specification of mass [19], comprises problems of the form

$$u_t(x,t) = u_{xx}(x,t) + f(x,t), \quad (x,t) \in [0,1] \times [0,T],$$

with

$$u(x,0) = \phi(x), \quad x \in [0,1],$$

and the conditions

$$\int_0^1 k_0(x)\ u(s,t)ds = g_0(t), \quad \int_0^1 k_1(x)\ u(s,t)ds = g_1(t), \quad t \in [0,T],$$

where f, ϕ, k_0, g_0, k_1, and g_1 are given functions, and T is the termination time. Numerous approaches have also been proposed to solve this problem class, including a Galerkin method [15], radial basis functions [52, 22], the homotopy perturbation method [3], finite differences [18, 45, 46, 47], and spectral collocation [11]. Additional examples of numerical methods for each problem class are provided in [12, 50].

The approaches employed in the literature for the two problem classes often involve the use of a standard spatial discretization method, e.g., finite differences or collocation, coupled with a standard numerical quadrature rule, e.g., the compound trapezoidal rule or Simpson's rule to approximate the integrals, coupled with a standard time-stepping method, such as the forward Euler method or the classical four-stage fourth-order explicit Runge–Kutta method, with fixed, uniform spatial meshes and uniform time-steps. Usually, numerical results comprise a discrete numerical solution at a set of uniformly distributed points across the spatial domain at the final time. All problems considered have exact solutions that can be used to compute errors of the numerical solutions. Typically, numerical results are provided over a range of spatial meshes and time-steps, with reported errors that are 10^{-4} or smaller. Implementation details are rarely provided, but, in some cases, the use of software platforms such as Maple, Mathematica, or Matlab to compute numerical results is reported. Occasionally, CPU times are provided. However, it is difficult to compare timing results across papers because of the variety of software platforms and hardware environments employed. In none of the papers is the computation of error estimates or the use of adaptive error-control algorithms considered. Moreover, none provides software that users could access in order to use the proposed methods.

In this report, we demonstrate how nonlocal IBVPs can be reformulated to enable their solution, in a straightforward fashion, using widely available state-of-the-art software to obtain continuous, error-controlled numerical solutions. For example, for a problem with a nonlocal condition of the form

$$u(1,t) = \int_0^1 u(s,t)\,ds,$$

the key technique in the reformulation process is the introduction of an auxiliary function

$$w(x,t) = \int_0^x u(s,t)\,ds.$$

The nonlocal condition can then be rewritten in a "local" form as

$$u(1,t) = w(1,t),$$

and, from the definition of the auxiliary function, we have the differential equation and associated boundary condition

$$w_x(x,t) = u(x,t), \quad w(0,t) = 0.$$

The problem now comprises the original PDE with local boundary conditions, coupled with a differential equation and its boundary condition. The introduction of this type of auxiliary function is also considered in [3].

An outline of the remainder of this report is as follows. In Section 2, we provide an overview of the `eBACOLI-PE` package used to solve reformulated nonlocal problems chosen from the literature. Each of these problems is presented in Section 3, with a description of the reformulation process applied to the problem to enable its solution using `eBACOLI-PE`. In Section 4, numerical results comparing the computed and exact solutions for each problem at its termination time are presented. Finally, in Section 5, conclusions and future projects are presented.

2 Description of `eBACOLI-PE`

The `eBACOLI-PE` software package is an update of the `eBACOLI` software package for solving one-dimensional PDE systems and is a member of the `BACOL` family of adaptive B-spline collocation codes. In this work, we use this package to solve nonlocal problems reformulated to yield a coupled system comprising (i) a parabolic PDE sub-system and (ii) a PDE sub-system representing differential constraints. The package, `eBACOLI-PE`, is designed to solve problems in the form

$$\begin{aligned}
\mathbf{u}_t &= \mathbf{f}(t, x, \mathbf{u}, \mathbf{u}_x, \mathbf{u}_{xx}, \mathbf{v}, \mathbf{w}, \mathbf{w}_x, \mathbf{w}_{xx}),\\
\mathbf{v}_t &= \mathbf{g}(t, x, \mathbf{u}, \mathbf{v}, \mathbf{w}),\\
\mathbf{0} &= \mathbf{h}(t, x, \mathbf{u}, \mathbf{u}_x, \mathbf{v}, \mathbf{w}, \mathbf{w}_x, \mathbf{w}_{xx}), \qquad (x,t) \in [x_L, \le x_R] \times [0, T],
\end{aligned}$$

subject to the initial conditions

$$\mathbf{u}(x,0) = \mathbf{u}_0(x), \quad \mathbf{v}(x,0) = \mathbf{v}_0(x), \quad \mathbf{w}(x,0) = \mathbf{w}_0(x), \quad x \in [x_L, x_R],$$

and the boundary conditions

$$\mathbf{B}_L(t, \mathbf{u}(x_L,t), \mathbf{u}_x(x_L,t), \mathbf{v}(x_L,t), \mathbf{v}_x(x_L,t), \mathbf{w}(x_L,t), \mathbf{w}_x(x_L,t)) = \mathbf{0},$$
$$\mathbf{B}_R(t, \mathbf{u}(x_R,t), \mathbf{u}_x(x_R,t), \mathbf{v}(x_R,t), \mathbf{v}_x(x_R,t), \mathbf{w}(x_R,t), \mathbf{w}_x(x_R,t)) = \mathbf{0},$$

where $t \in [0,T]$. It is important to note that `eBACOLI-PE` requires the boundary conditions to be separated; i.e., each boundary condition must depend on solution information at only one boundary point.

If n_u, n_v, and n_w denote the number of components in $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$, respectively, the total number of PDEs is denoted by

$$n = n_u + n_v + n_w.$$

A requirement of `eBACOLI-PE` is that $n_u \geq 1$ and $n_v, n_w \geq 0$. Furthermore, *the number of boundary conditions that must be imposed at each boundary point is n.* In addition, the set of n boundary conditions at each endpoint must *collectively* depend on all solution components (or their spatial derivatives) in order that the overall system of equations is well-posed within the solution framework employed by `eBACOLI-PE`.

In the use of `eBACOLI-PE` for reformulated nonlocal problems considered in this report, there are no equations of the form $\mathbf{v}_t = \mathbf{g}(\cdot)$, i.e., $\mathbf{v}$ is absent from the problem specification,[1] and $\mathbf{w}$ appears only through PDEs involving first-order spatial derivatives.[2] Hence, $\mathbf{w}$ is specified by PDEs of the form

$$\mathbf{w}_x = \mathbf{h}(t, x, \mathbf{u}, \mathbf{w}). \tag{1}$$

Only $2n_u$ boundary conditions are naturally prescribed (the boundary conditions of the original PDE problem after reformulation). Another n_w follow from the definitions of auxiliary functions, $\mathbf{w}$. Mathematically, this information is sufficient to determine $\mathbf{w}(\cdot, t)$. However, `eBACOLI-PE` is formulated using a collocation framework in which each dependent variable requires two boundary conditions. To satisfy this software requirement, the remaining n_w boundary conditions are obtained by *directly collocating components of* (1) *at $x = x_L$ or $x = x_R$.* How such conditions are obtained is described in detail for each example in Section 3.

Underlying numerical method

The underlying numerical method implemented in `eBACOLI-PE` employs a method-of-lines strategy in which the spatial discretization is performed by adaptive B-spline collocation, and the resulting semi-discrete system is advanced in time using a modified version of the implicit DAE time integrator `DASSL`.

[1] Auxiliary functions $v(x,t)$ that appear in Section 3 are not to be confused with $\mathbf{v}$.
[2] Auxiliary functions $w(x,t)$ that appear in Section 3 may not be directly related to $\mathbf{w}$.

(i) Spatial discretization by C^1 B-spline collocation. At each time step, each component of the solution is represented on a spatial mesh

$$\pi = \{x_i\}_{i=0}^{N}, \quad x_L = x_0 < \cdots < x_N = x_R,$$

by a C^1 piecewise-polynomial B-spline expansion of degree p. On each mesh subinterval, the PDE residual is enforced at $p-1$ Gauss collocation points. Together with the boundary constraints, this yields a semi-discrete system whose unknowns are the B-spline coefficients.

(ii) Semi-discrete system and ABD structure. The collocation equations and the boundary conditions define a system of DAEs of index 1 in time. Since each collocation equation involves only a small number of neighboring B-spline coefficients, the Jacobian matrices arising in Newton iterations have an *almost block diagonal* (ABD) structure. The `eBACOLI-PE` solver exploits this structure to solve efficiently linear systems arising in the implicit time integration using the `COLROW` package [23].

(iii) Time integration and temporal error control. The index-1 DAE system obtained from the method-of-lines spatial discretization is integrated in time using a variable-step, variable-order BDF method (as implemented in `DASSL`). At each step, a Newton iteration is applied to the implicit BDF equations. The time-step size and the BDF order are adapted to satisfy user-specified absolute and relative tolerances efficiently, yielding temporal error control.

(iv) Spatial error estimation and adaptive remeshing. After a successful time-step, `eBACOLI-PE` computes a spatial error estimate by comparing the computed B-spline solution with an inexpensive auxiliary interpolant. Two options for the interpolant are: (i) a superconvergent interpolation (SCI), which constructs a higher-accuracy interpolant using superconvergent solution values available from the collocation solution, (ii) a lower-order interpolation (LOI) which constructs a lower-accuracy interpolant based on certain collocation solution values. If the spatial error estimate does not satisfy the requested tolerance, the step is rejected, and a new mesh is constructed to equidistribute the estimated error approximately. The computation is then restarted from the last accepted time step by interpolating the solution and the time integrator history onto the new mesh. See [43] for further details.

(v) Continuous numerical solution. Since the spatial approximation is based on a C^1 B-spline basis, the numerical solution is available as a C^1 continuous function of x. Combined with the dense-output/interpolation facilities of the time integrator, this yields a continuous numerical solution that can be evaluated at arbitrary, user-selected points in space and time within the domain, with error control that adapts the computation to satisfy user-prescribed tolerances.

In summary, `eBACOLI-PE` computes continuous, error-controlled numerical solutions using (i) adaptive, Gaussian B-spline collocation in space, with mesh adaptation driven by interpolation-based (SCI/LOI) spatial error estimation, and (ii) an adaptive index-1 DAE integrator in time, with a time-step and BDF order adaptation driven by temporal error estimation.

3 Test Problems and Their Reformulations

3.1 [ZoSh2013-Ex1]

This is Example 1 from [59].

$$u_t(x,t) = u_{xx}(x,t), \quad (x,t) \in [0,1] \times [0,1],$$

with the initial condition

$$u(x,0) = 1 + \cos x, \quad x \in [0,1], \tag{2}$$

and the nonlocal boundary conditions

$$u(0,t) = \int_0^1 (s+t)\ u(s,t)ds + \frac{1}{2} + 2e^{-t} - t - e^{-t}[\cos 1 + (1+t)\sin 1],$$

and

$$u(1,t) = \int_0^1 te^{-s}\ u(s,t)\,ds + 1 + e^{-t}\cos 1 - \frac{t}{2e}[2(-1+e) + e^{-t}(e - \cos 1 + \sin 1)],$$

where $t \in [0,1]$. The exact solution is

$$u(x,t) = 1 + e^{-t}\cos x. \tag{3}$$

Reformulation: To reformulate the nonlocal boundary conditions, we introduce the auxiliary functions

$$w(x,t) = \int_x^1 (s+t)\ u(s,t)ds \quad \text{and} \quad v(x,t) = \int_0^x te^{-s}\ u(s,t)ds. \tag{4}$$

Then

$$w_x(x,t) = -(x+t)u(x,t), \quad w(1,t) = 0, \tag{5}$$

and

$$v_x(x,t) = te^{-x}u(x,t), \quad v(0,t) = 0. \tag{6}$$

The nonlocal boundary conditions then become

$$u(0,t) = w(0,t) + \frac{1}{2} + 2e^{-t} - t - e^{-t}\left[\cos 1 + (1+t)\sin 1\right],$$

and

$$u(1,t) = v(1,t) + 1 + e^{-t}\cos 1 - \frac{t}{2e}\left[2(-1+e) + e^{-t}(e - \cos 1 + \sin 1)\right],$$

respectively.

To obtain the remaining boundary conditions, we collocate (5) at $x = 0$ giving

$$w_x(0,t) + tu(0,t) = 0,$$

and collocate (6) at $x = 1$, giving

$$v_x(1,t) - te^{-1}\, u(1,t) = 0.$$

The `eBACOLI-PE` solver requires initial conditions for $w(x,t)$ and $v(x,t)$. Substituting (2) in (4), we obtain

$$w(x,0) = \frac{1-x^2}{2} + (\sin 1 - x\sin x) + (\cos 1 - \cos x),$$

and

$$v(x,0) = 0.$$

To check the accuracy of the computed approximations to $w(x,t)$ and $v(x,t)$, we derive their exact solutions by substituting (3) in (4). Thus

$$\begin{aligned} w(x,t) \quad &= \quad \frac{1-x^2}{2} + t(1-x) + e^{-t}(\sin 1 + \cos 1 - x\sin x - \cos x) \\ &\quad + te^{-t}(\sin 1 - \sin x), \end{aligned}$$

and

$$v(x,t) \quad = \quad t(1-e^{-x}) + \frac{te^{-t}}{2}\left(e^{-x}(\sin x - \cos x) + 1\right).$$

3.2 [KhSa2012-Ex1]

This is Example 1 from [36].

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) - \left(x(x-1) + \frac{\delta}{6(1+\delta)} + 2\right)e^{-t}, \quad (x,t) \in [0,1]\times[0,3], \\ \\ u(x,0) = x(x-1) + \frac{\delta}{6(1+\delta)}, \quad t \in [0,3], \\ \\ u(0,t) = -\delta\int_0^1 u(s,t), \quad x \in [0,1], \\ \\ u(1,t) = -\delta\int_0^1 u(s,t)ds, \quad x \in [0,1], \end{cases}$$

with exact solution

$$u(x,t) = \left(x(x-1) + \frac{\delta}{6(1+\delta)} \right) e^{-t}. \tag{7}$$

Reformulation: If

$$w(x,t) = \int_0^x u(s,t)ds, \tag{8}$$

then

$$w_x(x,t) = u(x,t), \quad w(0,t) = 0, \tag{9}$$

and the boundary conditions can be rewritten as

$$u(0,t) = -\delta w(1,t), \quad u(1,t) = -\delta w(1,t),$$

respectively.

Recall from Section 2 that `eBACOLI-PE` requires that the boundary conditions be separated. As a consequence, since the first boundary condition is non-separated, further reformulation is required. Thus, if

$$y(x,t) = w(1,t), \tag{10}$$

then

$$y_x(x,t) = 0, \tag{11}$$

and

$$y(0,t) = w(1,t).$$

The non-separated boundary condition can then be rewritten as

$$u(0,t) = -\delta y(0,t).$$

The remaining boundary conditions are then obtained by collocating (11) at $x = 0$, giving

$$y_x(0,t) = 0,$$

and by collocating (9) at $x = 1$, giving

$$w_x(1,t) = u(1,t).$$

Then, using (7) in (8),

$$w(x,t) = \left(\frac{x^3}{3} - \frac{x^2}{2} + \frac{\delta\, x}{6 + 6\,\delta} \right) e^{-t}, \quad w(x,0) = \frac{x^3}{3} - \frac{x^2}{2} + \frac{\delta\, x}{6 + 6\,\delta},$$

and using (10),

$$y(x,t) = \left(-\frac{1}{6} + \frac{\delta}{6 + 6\,\delta} \right) e^{-t}, \quad y(x,0) = -\frac{1}{6} + \frac{\delta}{6 + 6\,\delta}.$$

3.3 [KhSa2012-Ex4]

This is Example 4 from [36].

$$\begin{cases} u_t(x,t) = \frac{x}{2}u_{xx}(x,t) - u^2(x,t) + \left(\frac{\pi^4}{16}x^2 - \frac{\pi^3}{4}x + \frac{\pi^2}{4} + \sin^2(\pi x)e^{-2\pi t}\right), \\ \text{where} \quad (x,t) \in [0,1] \times [0,2], \\ u(x,0) = \frac{-\pi^2}{4}x + \frac{\pi}{2} + \sin(\pi x), \quad x \in [0,1], \\ u_x(0,t) = \frac{-\pi^2}{4} + \pi e^{-\pi t}, \quad t \in [0,2], \\ u(1,t) = \int_0^1 u(s,t)ds - \left(\frac{\pi^2}{8} + \frac{2}{\pi}e^{-\pi t}\right), \quad t \in [0,2]. \end{cases}$$

The exact solution is

$$u(x,t) = \frac{-\pi^2}{4}x + \frac{\pi}{2} + \sin(\pi x)e^{-\pi t}. \tag{12}$$

Reformulation: Let

$$w(x,t) = \int_0^x u(s,t)ds.$$

Then

$$w_x(x,t) = u(x,t), \quad w(0,t) = 0, \tag{13}$$

and the integral boundary condition can be rewritten as

$$u(1,t) = w(1,t) - \left[\frac{\pi^2}{8} + \frac{2}{\pi}e^{-\pi t}\right].$$

The remaining boundary condition is obtained by collocating (13) at $x = 1$, giving

$$w_x(1,t) - u(1,t) = 0.$$

Also

$$w(x,t) = \frac{-\pi^2}{8}x^2 + \frac{\pi}{2}x - \frac{e^{-\pi t}\cos(\pi x)}{\pi} + \frac{e^{-\pi t}}{\pi},$$

$$w(x,0) = \frac{-\pi^2}{8}x^2 + \frac{\pi}{2}x - \frac{\cos(\pi x)}{\pi} + \frac{1}{\pi}.$$

3.4 [ElEl2021-Ex4]

This is Example 4 from [24].

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + (2e^{-t} - te^{-t} - 1)(1+x), \quad (x,t) \in [0,1] \times [0,0.1], \\ \\ u(x,0) = 0, \quad x \in [0,1], \\ \\ u(0,t) = \frac{6}{5} \int_0^1 s\ u(s,t)\,ds, \quad u(1,t) = \frac{4}{3} \int_0^1 u(s,t)\,ds, \quad t \in [0,0.1], \end{cases}$$

with exact solution

$$u(x,t) = (1-t)(1-e^{-t})(1+x).$$

Reformulation:

Let

$$w(x,t) = \int_x^1 su(s,t)\,ds \quad \text{and} \quad v(x,t) = \int_0^x u(s,t)\,ds.$$

Then

$$w_x(x,t) = -xu(x,t), \qquad w(1,t) = 0, \tag{14}$$

and

$$v_x(x,t) = u(x,t), \quad v(0,t) = 0. \tag{15}$$

The integral boundary conditions can then be rewritten as

$$u(0,t) = \frac{6}{5} w(0,t) \quad \text{and} \quad u(1,t) = \frac{4}{3} v(1,t),$$

respectively.

The remaining boundary conditions are obtained by collocating (14) at $x = 0$, giving

$$w_x(0,t) = 0.$$

and collocating (15) at $x = 1$, giving

$$v_x(1,t) - u(1,t) = 0.$$

Also

$$w(x,t) = \frac{1}{6}(1-t)(1-e^{-t})\left(5 - 3x^2 - 2x^3\right), \quad w(x,0) = 0,$$

$$v(x,t) = \frac{1}{2}(1-t)(1-e^{-t})\left(2x + x^2\right), \quad v(x,0) = 0.$$

3.5 [EEE2022-Ex5]

This is the corrected Example 5 in [25] where $u(0,t)$ should be $u(1,t)$.

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + (\pi^2-1)e^{-t}\left(\sin(\pi x)+\cos(\pi x)\right), \quad (x,t)\in[0,1],\times[0,1], \\ u(x,0) = \sin(\pi x)+\cos(\pi x), \quad x\in[0,1], \\ u(0,t) = 2\int_0^1 \sin(\pi s)\ u(s,t)ds, \quad t\in[0,1], \\ u(1,t) = -2\int_0^1 \cos(\pi s)\ u(s,t)ds, \quad t\in[0,1], \end{cases}$$

with exact solution

$$u(x,t) = e^{-t}\left(\sin(\pi x)+\cos(\pi x)\right). \tag{16}$$

Reformulation:

If

$$w(x,t) = \int_x^1 \sin(\pi s)\ u(s,t)ds \quad \text{and} \quad v(x,t) = \int_0^x \cos(\pi s)\ u(s,t)ds,$$

then

$$w_x(x,t) = -\sin(\pi x)\ u(x,t), \quad w(1,t)=0, \tag{17}$$

and

$$v_x(x,t) = \cos(\pi x)\ u(x,t), \quad v(0,t)=0. \tag{18}$$

The boundary conditions then become

$$u(0,t) = 2w(0,t), \quad u(1,t) = -2v(1,t),$$

respectively.

The remaining boundary conditions are obtained by first collocating (17) at $x=0$, yielding

$$w_x(0,t) = 0,$$

and then (18) at $x=1$, yielding

$$v_x(1,t) - \cos(\pi)u(1,t) = 0 \implies v_x(1,t)+u(1,t) = 0.$$

Then, from (16), we obtain

$$w(x,t) = e^{-t}\left[\frac{1}{2} - \left(\frac{x}{2} - \frac{\sin(2\pi x)}{4\pi} + \frac{\sin^2(\pi x)}{2\pi}\right)\right],$$

$$w(x,0) = \frac{1-x}{2} + \frac{\sin(2\pi x)+\cos(2\pi x)-1}{4\pi},$$

and

$$v(x,t) = e^{-t}\left(\frac{\sin^2(\pi x)}{2\pi} + \frac{x}{2} + \frac{\sin(2\pi x)}{4\pi}\right),$$

$$v(x,0) = \frac{x}{2} + \frac{\sin(2\pi x) - \cos(2\pi x) + 1}{4\pi}.$$

3.6 [KhSa2012-Ex3]

This is Example 3 from [36].

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + \left[1 + \left(\frac{\pi^2}{4} - 1\right)t\right] e^{-t}\cos\left(\frac{\pi x}{2}\right), \quad (x,t) \in [0,1]\times[0,3], \\ u(x,0) = 0, \quad x \in [0,1], \\ \int_0^1 \sin\left(\frac{\pi s}{2}\right) u(s,t)\, ds = \frac{te^{-t}}{\pi}, \quad u_x(1,t) = -\frac{\pi}{2}te^{-t}, \quad t \in [0,3]. \end{cases}$$

The exact solution is

$$u(x,t) = te^{t}\cos\left(\frac{\pi x}{2}\right).$$

Reformulation: If

$$w(x,t) = \int_x^1 \sin\left(\frac{\pi s}{2}\right) u(s,t) ds,$$

the integral boundary condition can be rewritten as

$$w(0,t) = \frac{te^{-t}}{\pi}.$$

Also

$$w_x(x,t) = -\sin\left(\frac{\pi x}{2}\right) u(x,t), \quad w(1,t) = 0. \tag{19}$$

The remaining boundary condition is obtained by collocating (19) at $x = 0$, giving

$$w_x(1,t) = 0.$$

Then, at $x = 1$,

$$u_x(1,t) + \frac{\pi}{2}te^{-t} = 0 \quad \text{and} \quad w(1,t) = 0.$$

While the set of right boundary conditions includes dependence on both variables or their derivatives, this is not the case for the set of left boundary conditions. Because the kernel of the integral that appears in the original boundary *condition is zero at $x = 0$, this removes the dependence on $u(x,t)$ from the second of the left boundary conditions. Thus, there is no dependence on $u(x,t)$ or $u_x(x,t)$ within the left set of boundary conditions.* As mentioned earlier, this is an issue because `eBACOLI-PE` requires dependence on all the solution variables or their derivatives in both sets of boundary conditions.

In order to overcome this issue, we apply a phase shift of $+\frac{\pi}{2}$ to the original problem. This implies that $\sin\left(\frac{\pi x}{2}\right)$ and $\cos\left(\frac{\pi x}{2}\right)$ will be swapped where they appear in the problem definition. The exact solution to this phase-shifted problem is

$$u(x,t) = te^{-t}\sin\left(\frac{\pi x}{2}\right),$$

and the rest of this phase-shifted problem is as follows. We have

$$u_t(x,t) = u_{xx}(x,t) + \left[1 + \left(\frac{\pi^2}{4} - 1\right)t\right]e^{-t}\sin\left(\frac{\pi x}{2}\right), \quad (x,t) \in [0,1]\times[0,3],$$

The initial condition is

$$u(x,0) = 0, \quad x \in [0,1],$$

and the boundary conditions are

$$\int_0^1 \cos\left(\frac{\pi s}{2}\right)u(s,t)\,ds = \frac{te^{-t}}{\pi} \quad \text{and} \quad u_x(1,t) = 0, \quad t \in [0,3].$$

Here, the $u_x(1,t) = 0$ condition is obtained by making use of the exact solution to this phase-shifted problem.

We now apply a standard reformulation to this phase-shifted problem. If

$$w(x,t) = \int_x^1 \cos\left(\frac{\pi s}{2}\right)u(s,t)ds,$$

then

$$w_x(x,t) = -\cos\left(\frac{\pi x}{2}\right)u(x,t), \quad w(1,t) = 0. \tag{20}$$

The integral boundary condition can then be rewritten as

$$w(0,t) = \frac{te^{-t}}{\pi}.$$

The remaining boundary condition is obtained by collocating (20) at $x = 0$, giving

$$w_x(0,t) + \cos(0)\,u(0,t) = 0, \implies w_x(0,t) + u(0,t) = 0.$$

In this case, we observe that both the left and right sets of boundary conditions have dependence on both variables or their derivatives.

The exact solution for $w(x,t)$ is

$$w(x,t) = \frac{te^{-t}}{2\pi}(1 + \cos(\pi x)),$$

and the initial condition for $w(x,t)$ is $w(x,0) = 0$.

We note that once the solution to this phase-shifted problem is computed, we can recover the solution to the original problem by phase shifting the computed solution $u(x,t)$ to the phase-shifted problem by $-\frac{\pi}{2}$.

3.7 [EEE2022-Ex3]

This is Example 3 from [25].

$$\begin{cases} u_t(x,t) = u_{xx}(x,t), \quad (x,t) \in [0,1] \times [0,1], \\ u(x,0) = \cos\left(\frac{\pi x}{2}\right), \quad x \in [0,1], \\ \begin{cases} u(0,t) = e^{\frac{-\pi^2 t}{4}}, \\ \int_0^1 u(s,t)ds = \frac{2}{\pi} e^{-\frac{\pi^2 t}{4}}. \end{cases} \quad t \in [0,1], \end{cases}$$

The exact solution is

$$u(x,t) = e^{\frac{-\pi^2 t}{4}} \cos\left(\frac{\pi x}{2}\right). \tag{21}$$

Reformulation: Let

$$w(x,t) = \int_0^x u(s,t)ds.$$

Then

$$w_x(x,t) = u(x,t), \quad w(0,t) = 0, \tag{22}$$

and the integral boundary condition can be rewritten as

$$w(1,t) = \frac{2}{\pi} e^{-\frac{\pi^2 t}{4}}.$$

The remaining boundary condition is obtained by collocating (22) at $x = 1$ giving

$$w_x(1,t) - u(1,t) = 0.$$

Finally, from (21),

$$w(x,t) = \frac{2}{\pi} e^{-\frac{\pi^2 t}{4}} \sin\left(\frac{\pi x}{2}\right), \quad w(x,0) = \frac{2}{\pi} \sin\left(\frac{\pi x}{2}\right).$$

3.8 [EEE2022-Ex4]

This is Example 4 from [25].

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + e^t(x^2-2), \quad (x,t) \in [0,1]\times[0,1], \\ u(x,0) = x^2, \quad x \in [0,1], \\ \begin{cases} u(0,t) = 0, \quad t \in (0,1], \\ \int_0^1 u(s,t)ds = \frac{e^t}{3}, \quad t \in (0,1], \end{cases} \end{cases}$$

with exact solution

$$u(x,t) = e^t x^2. \tag{23}$$

Reformulation: Let

$$w(x,t) = \int_0^x u(s,t)ds.$$

Then

$$w_x(x,t) = u(x,t), \quad w(0,t) = 0, \tag{24}$$

and the integral condition becomes

$$w(1,t) = \frac{e^t}{3}.$$

The remaining boundary condition is obtained by collocating (24) at $x = 1$ giving

$$w_x(1,t) - u(1,t) = 0.$$

Also, using (23),

$$w(x,t) = \frac{e^t x^3}{3} \quad \text{and} \quad w(x,0) = \frac{x^3}{3}.$$

3.9 [KhSa2012-Ex2]

This is Example 2 from [36].

$$\begin{cases} u_t(x,t) = u_{xx}(x,t), & (x,t) \in [0,1]\times[0,3], \\ u(x,0) = \cos\Big(\dfrac{\pi x}{2}\Big) + \cos\Big(\dfrac{\pi x}{3}\Big), & x \in [0,1], \\ \displaystyle\int_0^{1/2} u(s,t)\,ds = \frac{1}{2\pi}\Big(2\sqrt{2}\,e^{-\pi^2 t/4} + 3e^{-\pi^2 t/9}\Big), & t \in [0,3], \\ u(1,t) = \frac{1}{2}\,e^{-\pi^2 t/4} + \frac{1}{2}\,e^{-\pi^2 t/9}, & t \in [0,3], \end{cases}$$

with exact solution

$$u(x,t) = e^{-\pi^2 t/4}\cos\left(\frac{\pi x}{2}\right) + e^{-\pi^2 t/9}\cos\left(\frac{\pi x}{3}\right).$$

Reformulation: The upper limit of $\frac{1}{2}$ in the integral boundary condition introduces an additional complication. To address this issue, we divide $[0,1]$ in two and map each half to the computational domain $z \in [0,1]$; see [8], Section 11.1.2, for a description of this type of mapping process. These mappings, each of which maps $x = \frac{1}{2}$ to $z = 0$, are defined as follows:

- For $[0, \frac{1}{2}]$, let
$$x(z) = \tfrac{1}{2} - \tfrac{1}{2}z,$$
so that $z = 0 \Leftrightarrow x = \frac{1}{2}$ and $z = 1 \Leftrightarrow x = 0$.
- For $[\frac{1}{2}, 1]$, let
$$x(z) = \tfrac{1}{2} + \tfrac{1}{2}z,$$
so that $z = 0 \Leftrightarrow x = \frac{1}{2}$ and $z = 1 \Leftrightarrow x = 1$.

We then denote $u(x,t)$ on $[0, \frac{1}{2}]$ by $u_L(x,t)$ and, on $[\frac{1}{2}, 1]$ by $u_R(x,t)$, so that the original PDE is replaced by

$$(u_L)_t\,(z,t) = 4\ \ (u_L)_{zz}\,(z,t), \quad z \in [0, \tfrac{1}{2}],$$

and,

$$(u_R)_t\,(z,t) = 4\ \ (u_R)_{zz}\,(z,t), \quad z \in [\tfrac{1}{2}, 1].$$

The corresponding initial conditions for $u_L(z,t)$ and $u_R(z,t)$ are

$$u_L(z,0) = \cos\left(\frac{\pi\ x(z)}{2}\right) + \cos\left(\frac{\pi\ x(z)}{3}\right), \quad \text{where } x(z) = 0.5(1-z),$$

and

$$u_R(z,0) = \cos\left(\frac{\pi\ x(z)}{2}\right) + \cos\left(\frac{\pi\ x(z)}{3}\right), \quad \text{where } x(z) = 0.5(1+z),$$

obtained from the original initial condition for $u(x,0)$. The exact solutions are

$$u_L(z,t) = \frac{e^{-\pi^2 t)}}{4}\cos\left(\frac{\pi\ x(z)}{2}\right) + \frac{e^{-\pi^2 t)}}{9}\cos\left(\frac{\pi\ x(z)}{3}\right), \quad x(z) = 0.5(1-z),$$

and

$$u_R(z,t) = \frac{e^{-\pi^2 t)}}{4}\cos\left(\frac{\pi\ x(z)}{2}\right) + \frac{e^{-\pi^2 t)}}{9}\cos\left(\frac{\pi\ x(z)}{3}\right), \quad x(z) = 0.5(1+z),$$

respectively, on using the original exact solution, $u(x,t)$.

Now define

$$w(z,t) = \int_0^{x(z)} u_L(s,t)\, ds, \quad x(z) = 0.5(1-z),$$

on $[0, \frac{1}{2}]$. Then

$$w_z(z,t) = -\frac{1}{2} u_L(z,t), \tag{25}$$

with corresponding boundary condition, $w(1,t) = 0$ (because $x(1) = 0$ on $[0, \frac{1}{2}]$).

The integral boundary condition can now be written as

$$w(0,t) = \frac{1}{2\pi}\left(2\sqrt{2}\ e^{-\frac{\pi^2 t}{4}} + 3\ e^{-\frac{\pi^2 t}{9}}\right),$$

since $x = \frac{1}{2}$ maps into $z = 0$ on $[0, \frac{1}{2}]$. The original right boundary condition becomes

$$u_R(1,t) = \frac{1}{2}\left(e^{-\frac{\pi^2 t}{9}}\right),$$

since $x(1) = 1$ on $[\frac{1}{2}, 1]$.

Since the `eBACOLI-PE` solver computes an approximate solution that has C^1-continuity, we enforce C^1-continuity on $u_L(x,t)$ and $u_R(x,t)$ at the interface between the two halves of the physical domain (where $x = \frac{1}{2}$ corresponds to $z = 0$ in both mappings). In the computational domain, these continuity conditions are

$$u_L(0,t) = u_R(0,t) \implies u_L(0,t) - u_R(0,t) = 0,$$

and

$$\left(u_L\right)_z(0,t) = -\left(u_R\right)_z(0,t) \implies \left(u_L\right)_z(0,t) + \left(u_R\right)_z(0,t) = 0.$$

Regarding the boundary conditions, at $z = 0$, we have

$$w(0,t) - \frac{1}{2\pi}\left(2\sqrt{2}\ e^{-\frac{\pi^2 t}{4}} + 3\ e^{-\frac{\pi^2 t}{9}}\right) = 0,$$

and the continuity conditions

$$u_L(0,t) - u_R(0,t) = 0, \quad \left(u_L\right)_z(0,t) + \left(u_R\right)_z(0,t) = 0.$$

Then, at $z = 1$

$$w(1,t) = 0, \quad u_R(1,t) - \frac{1}{2}\left(e^{-\frac{\pi^2 t}{9}}\right) = 0,$$

and the remaining boundary condition is obtained by collocating (25) at $z = 1$, giving

$$w_z(1,t) + \frac{1}{2} u_L(1,t) = 0.$$

The exact solution and the initial condition for $w(z,t)$ are obtained from the exact solution for $u_L(z,t)$, so that

$$w(z,t) = \frac{2}{\pi}\frac{e^{-\pi^2 t}}{4} \sin\left(\frac{\pi\ x(z)}{2}\right) + \frac{3}{\pi}\frac{e^{-\pi^2 t}}{9} \sin\left(\frac{\pi\ x(z)}{3}\right), \quad x(z) = 0.5(1-z),$$

and

$$w(z,0) = \frac{2}{\pi}\sin\left(\frac{\pi\ x(z)}{2}\right) + \frac{3}{\pi}\sin\left(\frac{\pi\ x(z)}{3}\right), \quad x(z) = 0.5(1-z).$$

Note that, in order to obtain the computed solution to the original problem on $[0, \frac{1}{2}]$, one must plot $1-x$ vs. $u_L(x,t)$. On the other hand, one can simply plot x vs. $u_R(x,t)$ to obtain the solution to the original problem on $[\frac{1}{2}, 1]$.

3.10 [AlYo2023-Ex1]

This is Example 1 from [3].

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) - u(x,t), & (x,t) \in [0,\pi] \times [0,1], \\ u(x,0) = \sin x, & x \in [0,\pi], \\ \displaystyle\int_0^\pi s\, u(s,t)\, ds = \pi e^{-2t}, & t \in [0,1], \\ \displaystyle\int_0^\pi (1-s)\, u(s,t)\, ds = (2-\pi)e^{-2t}, & t \in [0,1]. \end{cases}$$

The exact solution is

$$u(x,t) = e^{-2t}\sin x. \tag{26}$$

Reformulation:

If

$$w(x,t) = \int_0^x s\ u(s,t)\, ds,$$

and

$$v(x,t) = \int_0^x (1-s)\, u(s,t) ds,$$

then

$$w_x(x,t) = x\, u(x,t), \quad w(0,t) = 0, \tag{27}$$

and

$$v_x(x,t) = (1-x)\, u(x,t), \quad v(0,t) = 0. \tag{28}$$

The integral boundary conditions then become

$$w(\pi,t) = \pi e^{-2t} \quad \text{and} \quad v(\pi,t) = (2-\pi)e^{-2t}.$$

To obtain the remaining boundary conditions, we first collocate (28) at $x = 0$ to obtain

$$v_x(0,t) - u(0,t) = 0,$$

and then collocate (27) at $x = \pi$ to obtain

$$w_x(\pi,t) - \pi u(\pi,t) = 0.$$

Finally, from (26),

$$w(x,t) = e^{-2t}(\sin x - x\cos x),$$

$$w(x,0) = \sin x - x\cos x,$$

and

$$v(x,t) = e^{-2t}\left(1 + (x-1)\cos x - \sin x\right),$$
$$v(x,0) = 1 + (x-1)\cos x - \sin x.$$

3.11 [Sun1996-Ex1-Modified]

This example is a modified version of Example 1 from [51], namely,

$$u_t(x,t) = u_{xx}(x,t) + \frac{1}{x}u(x,t) + \frac{1}{2}\left(\int_1^x s\ u(s,t)\,ds\right) + f(x,t),\ (x,t) \in [1,2]\times[0,1],$$

with

$$f(x,t) = e^x\cos t - e^x\sin t\left(\frac{x^2+x+2}{2x}\right),$$

$$u(x,0) = 0, \quad x \in [0,1],$$

and

$$u(1,t) = e\ \sin t, \quad u_x(2,t) + u(2,t) = 2e^2\ \sin t, \qquad t \in [0,1].$$

The exact solution is

$$u(x,t) = e^x\sin t. \tag{29}$$

Reformulation: If

$$w(x,t) = \int_1^x s\ u(s,t)ds,$$

then

$$w_x(x,t) = x\ u(x,t), \quad w(1,t) = 0, \tag{30}$$

and the PDE may be written as

$$u_t(x,t) = u_{xx}(x,t) + \frac{1}{x}u(x,t) + \frac{1}{2}w(x,t) + f(x,t),$$

and

$$u(1,t) = e\sin t, \quad w(1,t) = 0,$$

$$u_x(2,t) + u(2,t) = 2e^2\ \sin t, \quad w_x(2,t) = 2u(2,t),$$

on collocating (30) at $x = 2$.

Also, from (29),

$$w(x,t) = (x-1)e^x\sin t, \quad w(x,0) = 0.$$

4 Numerical Results

In this section, we provide numerical results for the problems considered in Section 3. The `eBACOLI-PE` solver was provided with absolute and relative tolerances of 10^{-6} and an initial mesh of 11 points uniformly distributed across the spatial domain. We emphasize that the software, as the time integration progresses, can change the number of mesh points and the locations of the points according to the solution behavior.

The software developed for each problem and a link to the `eBACOLI-PE` solver is available at
`https://cs.smu.ca/~muir/SoftwareForNonLocalPDEs.shtml`.

For each problem in Section 3, we provide a plot of the computed and exact solutions for $u(x,t)$ at T, and report the ℓ_2 error for $u(x,t)$ at the output points. These points are a set of points uniformly distributed across the spatial domain, but we emphasize that, since the software provides a continuous numerical solution, solution values at any points in the spatial domain at any point in time can be obtained in a straightforward manner.

The computations were performed on a Linux system running Ubuntu 22.04.4 LTS on an Intel(R) Xeon(R) Platinum 8160 CPU @ 2.10GHz with 8 cores. The Fortran compiler was GNU Fortran (Ubuntu 11.4.0-1ubuntu1 22.04.2) 11.4.0.

For all problems, the results show excellent agreement between the computed solutions and the exact solutions for $u(x,t)$, ℓ_2 norm of the error for each of the computed solutions being approximately 10^{-6} or less. For each problem, the CPU time taken by `eBACOLI-PE` was less than 0.5 seconds.

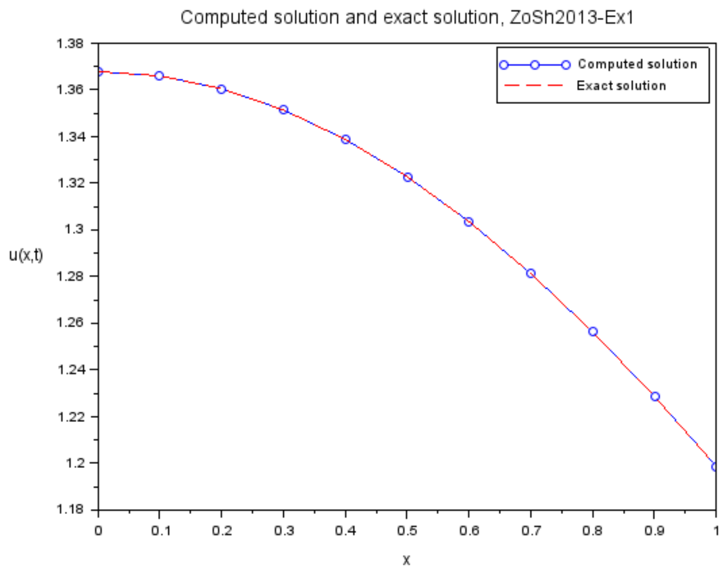

Figure 1: Computed solution and exact solution for ZoSh2013-Ex1 at $T = 1.0$. Discrete ℓ_2 error $= 2.3 \times 10^{-7}$.

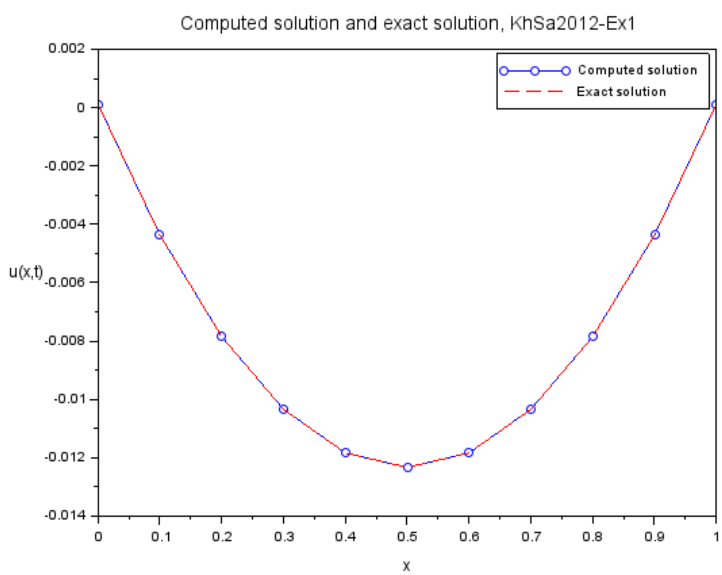

Figure 2: Computed solution and exact solution for KhSa2012-Ex1 at $T = 3.0$. Discrete ℓ_2 error $= 2.5 \times 10^{-8}$.

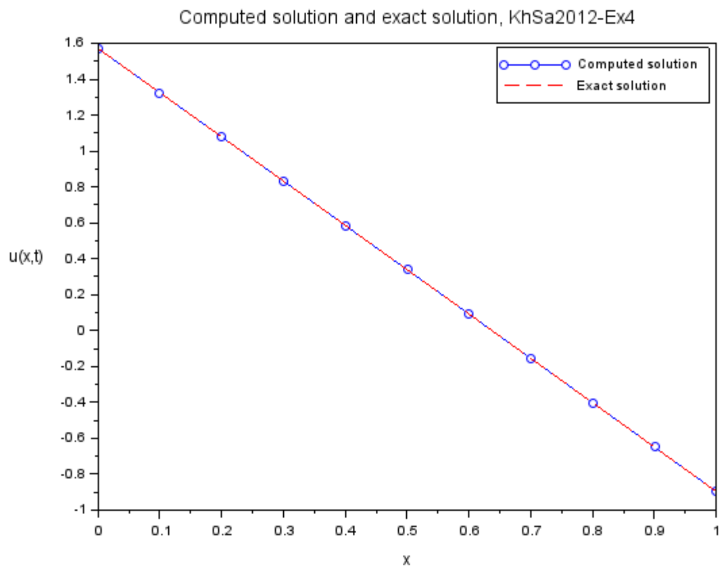

Figure 3: Computed solution and exact solution for KhSa2012-Ex4 at $T = 2.0$. Discrete ℓ_2 error $= 3.5 \times 10^{-7}$.

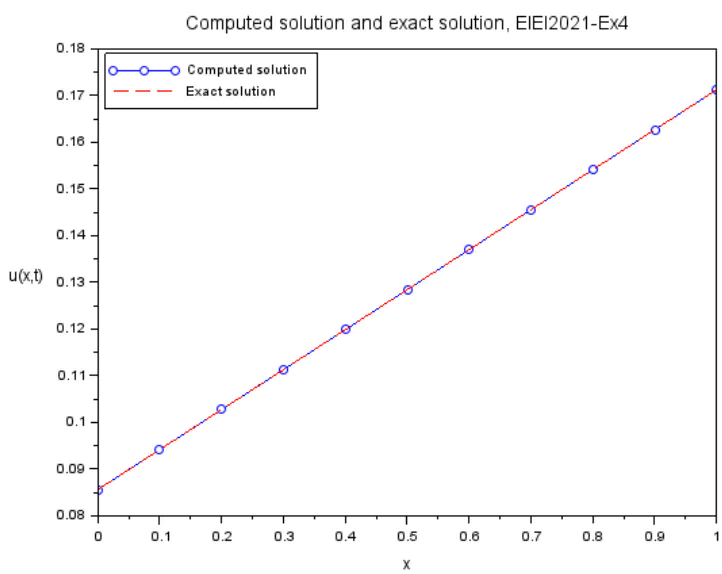

Figure 4: Computed solution and exact solution for ElEl2021-Ex4 at $T = 0.1$. Discrete ℓ_2 error $= 1.8 \times 10^{-6}$.

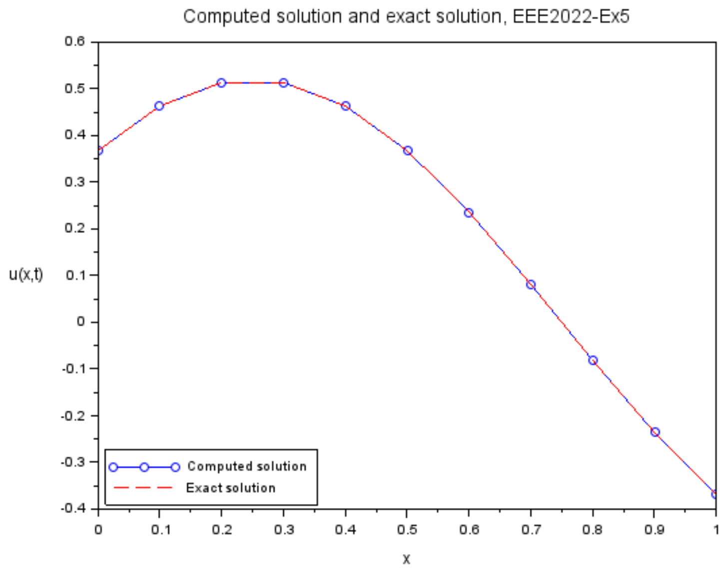

Figure 5: Computed solution and exact solution for EEE2022-Ex5 at $T = 1.0$. Discrete ℓ_2 error $= 6.0 \times 10^{-8}$.

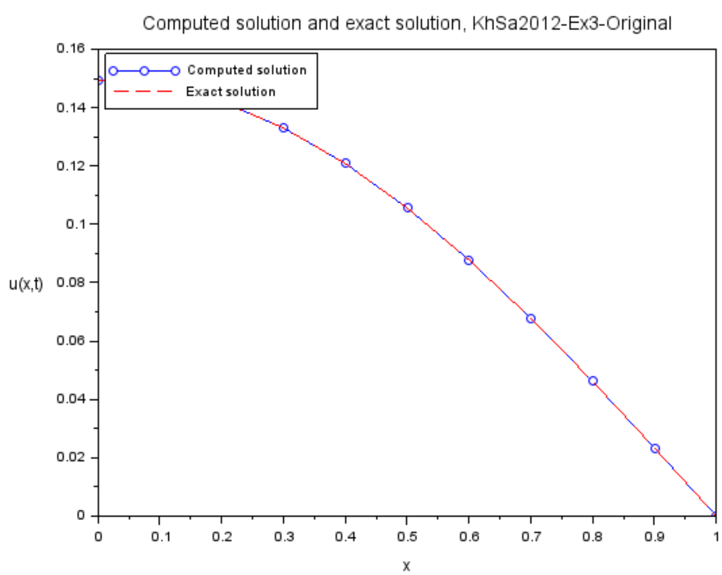

Figure 6: Computed solution and exact solution for KhSa2012-Ex3 (Original Problem) at $T = 3.0$. Discrete ℓ_2 error $= 1.1 \times 10^{-7}$.

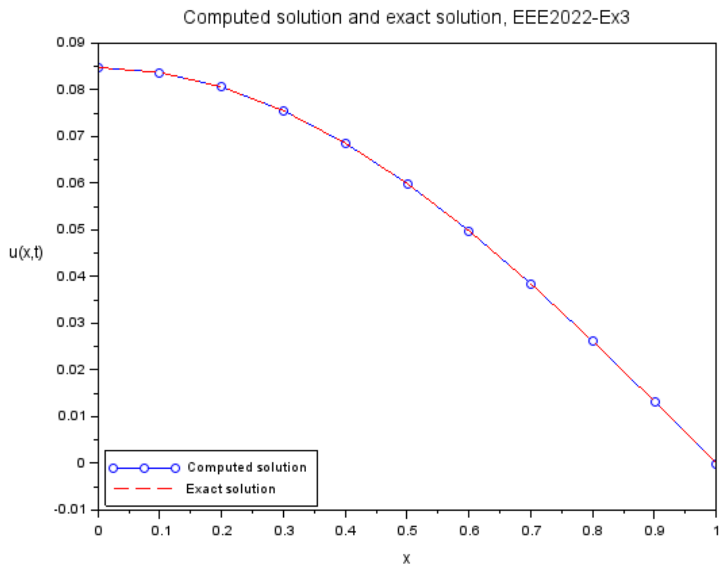

Figure 7: Computed solution and exact solution for EEE2022-Ex3 at $T = 1.0$. Discrete ℓ_2 error $= 1.1 \times 10^{-7}$.

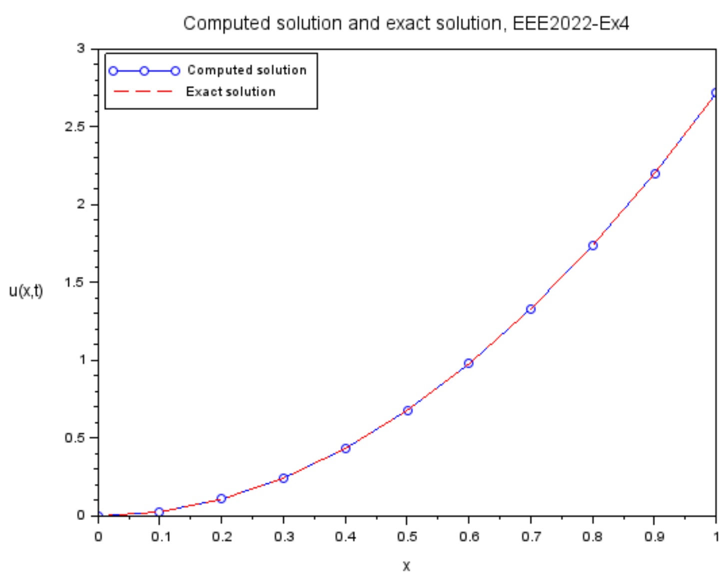

Figure 8: Computed solution and exact solution for EEE2022-Ex4 at $T = 1.0$. Discrete ℓ_2 error $= 5.1 \times 10^{-8}$.

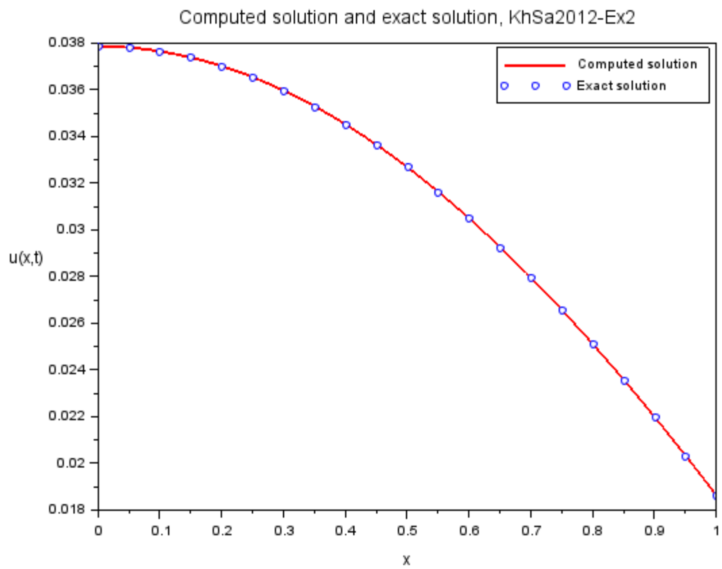

Figure 9: Computed solution and exact solution for KhSa2012-Ex2 at $T = 3.0$. Discrete ℓ_2 error $= 9.0 \times 10^{-8}$.

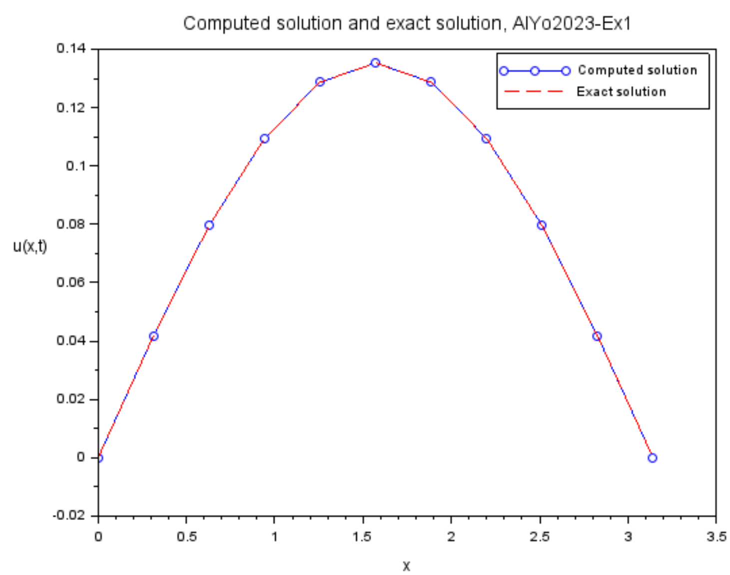

Figure 10: Computed solution and exact solution for AlYo2023-Ex1 at $T = 1.0$. Discrete ℓ_2 error $= 4.6 \times 10^{-6}$.

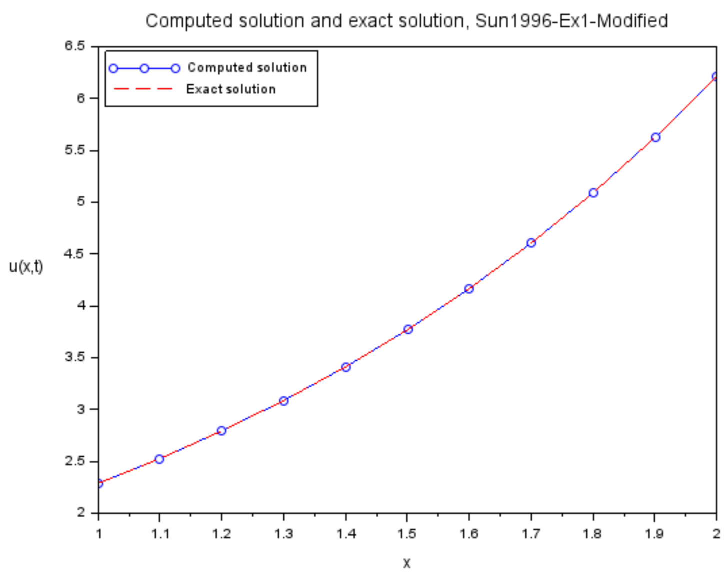

Figure 11: Computed solution and exact solution for Sun1996-Ex1-Modified at $T = 1.0$. Discrete ℓ_2 error $= 1.3 \times 10^{-6}$.

5 Conclusions and Future Work

This report describes an approach for reformulating nonlocal IBVPs to enable their solution using the state-of-the-art software package, `eBACOLI-PE`. This package returns a continuous, error-controlled numerical solution that can be evaluated at any point in the spatial domain and at any time in the temporal domain. The continuous numerical solution has an associated high-quality error estimate that satisfies a given user tolerance.

As the numerical results presented herein demonstrate, `eBACOLI-PE` computes, efficiently and accurately, numerical solutions to a selection of nonlocal IBVPs from the literature. The variety of reformulations can serve as templates for the reformulation of other nonlocal IBVPs.

A topic for future research is to investigate the extension of the reformulation processes to other types of IBVPs. One such class of problems comprises nonlocal first-order hyperbolic PDEs that occur in studies of population dynamics; see [38], [26], [1], and [9]. For example, consider

$$u_t(x,t) + u_x(x,t) = F(x, I(t))u(x,t), \quad 0 \le x \le A, \quad 0 \le t \le T,$$

with initial condition

$$u(x,0) = u_0(x), \quad 0 \le x \le A,$$

and the nonlocal boundary condition

$$u(0,t) = g\left(\int_0^A b(s, I(t))u(s,t)\, ds, t\right), \quad 0 \le t \le T,$$

where

$$I(t) = \int_0^A u(x,t)dx, \quad 0 \le t \le T,$$

and F, u_0, b, and g are given functions, A is a given constant, and T is the termination time. This problem describes the evolution in time of the age structure of a population. In [9] and [14], the authors consider a general first-order hyperbolic problem with a nonlocal boundary condition of the form

$$g\left(\int_0^A b(s, I(t))u(s,t)\, ds, t\right) = G(t), \quad 0 \le t \le T,$$

where G is a given function.

Several authors have discussed nonlocal problems involving the wave equation of the form

$$u_{tt}(x,t) - u_{xx}(x,t) = f(x,t), \quad 0 \le x \le 1, \quad 0 \le t \le T,$$

with

$$u(x,0) = f_1(x), \quad u_t(x,0) = f_2(x), \quad 0 \le x \le 1,$$

and

$$u(0,t) = 0, \quad \int_0^1 u(x,t)dx = 0, \quad 0 \le t \le T,$$

where f, f_1, and f_2 are given functions; see, for example, [31, 35, 39, 21, 37, 56, 58].

In [48, 56], a problem involving the telegraph equation is considered, viz.,

$$u_{tt}(x,t)+\alpha u_t(x,t) = \beta^2 u_{xx}(x,t)+\gamma u(x,t)+F(x,t)u(x,t), \; 0 \le x \le 1, \; 0 \le t \le T,$$

with

$$u(x,0) = f_1(x), \quad u_t(x,0) = f_2(x), \quad 0 \le x \le 1,$$

and

$$u(0,t) = g(t), \quad \int_0^1 u(x,t)dx = q(t), \quad 0 \le t \le T,$$

where F, f_1, f_2, g, and q are given functions, and α, β, and γ are given constants.

A nonlocal problem involving a pseudoparabolic equation of the form

$$u_t(x,t) = u_{xx}(x,t) + \tau u_{txx}(x,t) + f(x,t), \quad 0 \le x \le 1, \quad 0 \le t \le T,$$

with

$$u(x,0) = \phi(x), \quad 0 \le x \le 1,$$

$$\int_0^1 u(x,t)dx = \mu_1(t), \quad \int_0^1 x\ u(x,t)dx = \mu_2(t), \quad 0 \le t \le T,$$

where f, ϕ, μ_1, and μ_2, are given functions, τ is a given constant, is considered in [33].

Problems in which a nonlocal term appears in the governing equation are also appear in the literature; for example

$$u_t(x,t) = u_{xx}(x,t) + \frac{1}{2}u_x(x,t) + \nu \int_a^b \rho u_t(\rho,t)d\rho, \quad 0 \le x \le 1, \quad 0 \le t \le T,$$

where ν, a, and b are given constants, with standard initial and boundary conditions [51]. Another example of such a problem is the dynamic beam system [41, 40, 32], comprising

$$\begin{aligned} w_{tt}(x,t) &= \left(cd - a + b\int_0^1 w_x^2(x,t)dx\right) w_{xx}(x,t) - cd\psi_x(x,t), \\ \psi_{tt}(x,t) &= c\psi_{xx}(x,t) - c^2 d(\psi(x,t) - w_x(x,t)), \quad 0 \le x \le 1, \quad 0 \le t \le T, \end{aligned}$$

where a, b, c, and d are given constants, subject to standard initial and boundary conditions.

Another topic of future research is the merging of `eBACOLI-PE` with the latest version of the `BACOLI` family, `BACOLIKR` [44]. This would extend `eBACOLI-PE` to provide it with a capability for time and space dependent event detection, such as determining the time at which the spatial integral of the solution equals a specified value.

6 Acknowledgments

The authors acknowledge the technical assistance provided by Gemini (Google). This AI was utilized to assist with the reformulation steps that were applied to the nonlocal test problems in order to transform them into local problems. It was also used to help with the translation of some of the equations generated during the reformulation process into Fortran expressions that were then used in the code we developed to use `eBACOLI-PE` to solve the test problems. It was also used to assist with the translation of some of the equations generated during the reformulation process into LaTeX expressions that we then employed in the descriptions of the reformulation processes. Moreover, it was used to help with the generation of the Scilab scripts used to obtain the figures that appear in this report.

References

[1] L. M. Abia, Ó. Angulo, J. C. López-Marcos, and M. Á. López-Marcos. Numerical approximation of a nonlinear age-structured population model with finite life-span. *Journal of Scientific Computing*, 105:Article 2, 2025.

[2] M. Adams and P. Muir. Differential equation software for the computation of error-controlled continuous approximate solutions. *Numerical Algorithms*, 96:1021–1044, 2024.

[3] W. Al-Hayani and M. T. Younis. The homotopy perturbation method for solving nonlocal initial-boundary value problems for parabolic and hyperbolic partial differential equations. *European Journal of Pure and Applied Mathematics*, 16:1550–1567, 2023.

[4] N. Arar, L. Ait Kaki, and A. Ben Makhlouf. Highly efficacious sixth-order compact approach with nonclassical boundary specifications for the heat equation. *Mathematical Problems in Engineering*, 2022:8224959, 2022.

[5] A. Araújo and F. A. Oliveira. Semi-discretization method for the heat equation with non-local boundary conditions. *Communications in Numerical Methods in Engineering*, 10:751–758, 1994.

[6] U. Ascher, J. Christiansen, and R. D. Russell. Algorithm 569: COLSYS: Collocation software for boundary-value ODEs. *ACM Transactions on Mathematical Software*, 7:223–229, 1981.

[7] U. Ascher, J. Christiansen, and R. D. Russell. Collocation software for boundary-value ODEs. *ACM Transactions on Mathematical Software*, 7:209–222, 1981.

[8] U. M. Ascher, R. M. Mattheij, and R. D. Russell. *Numerical solution of boundary value problems for ordinary differential equations*. SIAM, 1995.

[9] B. Aylaj and A. Noussair. Convergence of numerical schemes to a nonlinear kinetic model of population dynamics with nonlocal boundary conditions. *SIAM Journal on Numerical Analysis*, 48:1707–1732, 2010.

[10] G. Bader and U. Ascher. A new basis implementation for a mixed order boundary value ODE solver. *SIAM Journal on Scientific and Statistical Computing*, 8:483–500, 1987.

[11] M. Bastani and D. K. Salkuyeh. Numerical studies of a nonlocal parabolic partial differential equations by spectral collocation method with preconditioning. *Computational Mathematics and Modeling*, 24:81–89, 2013.

[12] B. Bialecki, G. Fairweather, and J. López-Marcos. The Crank-Nicolson Hermite cubic orthogonal spline collocation method for the heat equation with nonlocal boundary conditions. *Advances in Applied Mathematics and Mechanics*, 5:442–460, 2013.

[13] B. Bialecki, G. Fairweather, and J. C. López-Marcos. The extrapolated Crank–Nicolson orthogonal spline collocation method for a quasilinear parabolic problem with nonlocal boundary conditions. *Journal of Scientific Computing*, 62:265–283, 2015.

[14] L. Bougoffa. An efficient method for solving nonlocal initial-boundary value problems for linear and nonlinear first-order hyperbolic partial differential equations. *Journal of Applied Mathematics and Computing*, 43:31–54, 2013.

[15] A. Bouziani, N. Merazga, and S. Benamira. Galerkin method applied to a parabolic evolution problem with nonlocal boundary conditions. *Nonlinear Analysis: Theory, Methods & Applications*, 69:1515–1524, 2008.

[16] P. Brown, A. Hindmarsh, and L. R. Petzold. Using Krylov methods in the solution of large-scale differential-algebraic systems. *SIAM J. Sci. Comp.*, 15:1467–1488, 1994.

[17] P. Brown, A. Hindmarsh, and L. R. Petzold. Consistent initial condition calculation for differential-algebraic systems. *SIAM J. Sci. Comp.*, 19:1495–1512, 1998.

[18] B. Cahlon, D. M. Kulkarni, and P. Shi. Stepwise stability for the heat equation with a nonlocal constraint. *SIAM Journal on Numerical Analysis*, 32:571–593, 1995.

[19] J. R. Cannon. The solution of the heat equation subject to the specification of energy. *Quarterly of Applied Mathematics*, 21:155–160, 1963.

[20] M. Dehghan. Numerical solution of a parabolic equation with non-local boundary specifications. *Applied Mathematics and Computation*, 145:185–194, 2003.

[21] M. Dehghan and A. Shokri. A meshless method for numerical solution of the one-dimensional wave equation with an integral condition using radial basis functions. *Numerical Algorithms*, 52:461–477, 2009.

[22] M. Dehghan and M. Tatari. Use of radial basis functions for solving the second-order parabolic equation with nonlocal boundary conditions. *Numerical Methods for Partial Differential Equations: An International Journal*, 24:924–938, 2008.

[23] J. Diaz, G. Fairweather, and P. Keast. Fortran packages for solving certain almost block diagonal linear systems by modified alternate row and column elimination. *ACM Transactions on Mathematical Software (TOMS)*, 9(3):358–375, 1983.

[24] M. El-Gamel and M. Abd El-Hady. On using Sinc collocation approach for solving a parabolic PDE with nonlocal boundary conditions. *Journal of Nonlinear Sciences and Applications*, 14:29–38, 2021.

[25] M. El-Gamel, G. I. El-Baghdady, and M. A. El-Hady. Highly efficient method for solving parabolic PDE with nonlocal boundary conditions. *Applied Mathematics*, 13:101–119, 2022.

[26] G. Fairweather and J. López-Marcos. A box method for a nonlinear equation of population dynamics. *IMA Journal of Numerical Analysis*, 11:525–538, 1991.

[27] G. Fairweather and P. Muir. Efficient computation of accurate continuous numerical solutions of nonlocal two-point boundary value problems. In *Go20 Conference on Scientific Computing and Software*, pages 101–119. Springer, 2025.

[28] M. Ghoreishi, A. M. Ismail, and A. Rashid. A new fourth-order explicit finite difference method for the solution of parabolic partial differential equation with nonlocal boundary conditions. *Journal of Computational Analysis and Applications*, 16:528–539, 2014.

[29] M. I. Ghoreishi and B. AI. The homotopy perturbation method (HPM) for nonlinear parabolic equation with nonlocal boundary conditions. *Applied Mathematical Sciences*, 5:113–123, 2011.

[30] K. R. Green and R. J. Spiteri. Extended BACOLI: Solving one-dimensional multiscale parabolic PDE systems with error control. *ACM Transactions on Mathematical Software*, 45:1–19, 2019.

[31] A. Guezane-Lakoud and N. Boumaza. Galerkin method applied for a nonlocal problem. *International Journal of Applied Mathematics & Statistics*, 19:63–71, 2010.

[32] T. Guo, Y. Li, and W. Qiu. Compact difference method for Euler-Bernoulli beams and plates with nonlinear nonlocal damping. *Applied Numerical Mathematics*, 2025.

[33] J. Jachimavičienė, Ž. Jesevičiūtė, and M. Sapagovas. The stability of finite-difference schemes for a pseudoparabolic equation with nonlocal conditions. *Numerical Functional Analysis and Optimization*, 30:988–1001, 2009.

[34] M. Javidi. The MOL solution for the one-dimensional heat equation subject to nonlocal conditions. In *International Mathematical Forum*, volume 12, pages 597–602, 2006.

[35] S. A. Khuri and A. Sayfy. A spline collocation approach for a generalized wave equation subject to non-local conservation condition. *Applied Mathematics and Computation*, 217:3993–4001, 2010.

[36] S. A. Khuri and A. Sayfy. A spline collocation approach for a generalized parabolic problem subject to non-classical conditions. *Applied Mathematics and Computation*, 218:9187–9196, 2012.

[37] V. Korzyuk, V. Erofeenko, and J. Sheika. Classical solution for initial–boundary value problem for wave equation with integral boundary condition. *Mathematical Modelling and Analysis*, 17:309–329, 2012.

[38] J. López-Marcos. An upwind scheme for a nonlinear hyperbolic integro-differential equation with integral boundary condition. *Computers & Mathematics with Applications*, 22:15–28, 1991.

[39] H. Marasi, R. Faghiri, and M. Hashimi. A series solution of non local hyperbolic equations. *Communications in Numerical Analysis*, 2012:cna–00113, 2012.

[40] J. Peradze and Z. Kalichava. A numerical algorithm for the nonlinear Timoshenko beam system. *Numerical Methods for Partial Differential Equations*, 36:1318–1347, 2020.

[41] J. Peradze, N. Papukashvili, and V. Odisharia. An approximate method for a nonlinear beam equation. *Reports of Enlarged Session of the Seminar of I.Vekua Institute of Applied Mathematics*, 19:83–87, 2004.

[42] L. R. Petzold. A description of DASSL: a differential-algebraic system solver. In *Scientific computing (Montreal, Que., 1982)*, IMACS Trans Sci Comput., I, pages 65–68. IMACS, New Brunswick, NJ, 1983.

[43] J. Pew, Z. Li, and P. Muir. Algorithm 962: BACOLI: B-spline adaptive collocation software for PDEs with interpolation-based spatial error control. *ACM Transactions on Mathematical Software*, 42:1–17, 2016.

[44] J. Pew, C. Tannahill, and P. Muir. Error-control B-spline Gaussian collocation PDE software with event detection. *AIMS Mathematics*, 11:3243–3268, 2026.

[45] M. Rehman and M. Taj. Efficient numerical technique for homogeneous heat equation subject to a boundary integral specification. *International Journal of Computational and Applied Mathematics*, 2:181–190, 2007.

[46] M. Rehman and M. Taj. Efficient techniques for the diffusion equation subject to specification of energy. *Applied Mathematical Sciences*, 3:2681–2694, 2009.

[47] M. Rehman, M. Taj, and M. Butt. Fifth-order numerical methods for heat equation subject to a boundary integral specification. *Acta Mathematica Universitatis Comenianae. New Series*, 79:89–104, 2010.

[48] D. K. Salkuyeh and H. R. Ghehsareh. Convergence of the variational iteration method for the Telegraph equation with integral conditions. *Numerical Methods for Partial Differential Equations*, 27:1442–1455, 2011.

[49] M. Sapagovas, T. Meškauskas, and F. Ivanauskas. Influence of complex coefficients on the stability of difference scheme for parabolic equations with non-local conditions. *Applied Mathematics and Computation*, 332:228–240, 2018.

[50] A. Štikonas. A survey on stationary problems, Green's functions and spectrum of Sturm–Liouville problem with nonlocal boundary conditions. *Nonlinear Analysis: Modelling and Control*, 19:301–334, 2014.

[51] Z.-Z. Sun. A second-order accurate finite difference scheme for a class of nonlocal parabolic equations with natural boundary conditions. *Journal of Computational and Applied Mathematics*, 76:137–146, 1996.

[52] M. Tatari and M. Dehghan. On the solution of the non-local parabolic partial differential equations via radial basis functions. *Applied Mathematical Modelling*, 33:1729–1738, 2009.

[53] E. Tohidi. Application of Chebyshev collocation method for solving two classes of non-classical parabolic PDEs. *Ain Shams Engineering Journal*, 6:373–379, 2015.

[54] E. Tohidi and A. Kılıçman. An efficient spectral approximation for solving several types of parabolic PDEs with nonlocal boundary conditions. *Mathematical Problems in Engineering*, 2014:369029, 2014.

[55] R. Wang, P. Keast, and P. Muir. BACOL: B-spline adaptive collocation software for 1-D parabolic PDEs. *ACM Transactions on Mathematical Software*, 30:454–470, 2004.

[56] H. Yao. Reproducing kernel method for the solution of nonlinear hyperbolic Telegraph equation with an integral condition. *Numerical Methods for Partial Differential Equations*, 27:867–886, 2011.

[57] S. Yousefi, Z. Barikbin, and M. Dehghan. Ritz-Galerkin method with Bernstein polynomial basis for finding the product solution form of heat equation with non-classic boundary conditions. *International Journal of Numerical Methods for Heat & Fluid Flow*, 22:39–48, 2012.

[58] Z.-f. Zhang, J.-r. Niu, and J. Luo. A weighted implicit difference scheme for the wave equation with nonlocal boundary condition. *Journal of Sichuan University (Natural Science Edition)*, 49, 2012.

[59] R. Zolfaghari and A. Shidfar. Solving a parabolic PDE with nonlocal boundary conditions using the Sinc method. *Numerical Algorithms*, 62:411–427, 2013.