A Numerical Study of the Computation of Accurate Continuous Numerical Solutions of Nonlocal Two–Point Boundary Value Problems

Graeme Fairweather ‡ Paul Muir§

April 5, 2024

Abstract

A non-local two-point boundary value problem is a system of ordinary differential equations in which the equations or the boundary conditions contain terms or coefficients that depend on an integral of a function that depends on one of the solution components. We discuss the reformulation of nonlocal two-point boundary problems to enable their approximate solution using widely available software packages that provide a continuous numerical approximation to a user-prescribed accuracy. Such a package is used to solve a collection of problems from the literature and comparisons are made with results that have been obtained using published methods. We show that, using the reformulation approach, these problems can be solved simply and efficiently to almost machine accuracy.

Key words: Nonlocal two-point boundary value problems, reformulation, continuous numerical solutions, error control algorithms
AMS subject classifications:

1 Introduction

The numerical analysis of two-point boundary value problems (TPBVPs) has been widely studied and, for their solution, several excellent software packages have been developed and widely disseminated. Such a package typically provides a continuous approximate solution to a user-specified accuracy. Many problems in science and engineering are modeled by TPBVPs. However, a given TPBVP may not be in the form required by a selected package. For example, the package may require the problem to be formulated as a first-order system of ordinary differential equations (ODEs) while other packages may handle mixed-order systems directly. Of course, rewriting a higher-order equation as a first-order system is straightforward. All the packages can handle not only Dirichlet boundary conditions (BCs), but also Neumann and Robin BCs. Most packages require separated BCs and therefore a TPBVP with periodic BCs would require reformulation, resulting in an increase in the size of the problem. Also, a problem posed on a semi-infinite interval such as that in [27], would also require special treatment as would a TPBVP involving an interface; see, for example, [4]. There are other situations where reformulation is essential, and many of these are described in [7, 6]; see [22] for similar techniques for time-dependent problems.

This approach to solving TPBVPs is certainly not new. Reliable software packages for solving TPBVPs, such as COLSYS [5], have been available for more than forty years. They have been used with much success to solve challenging engineering problems; see, for example, [10, 20], [18], and [23, 29] for their application in chemical engineering, mechanical engineering, and heat transfer studies, respectively. However, the availability of these high quality TPBVP solvers appears to have gone unnoticed by many in the numerical analysis community, resulting in the continued development of methods for specific BVPs which, in general,

†Corresponding Author
‡Mathematical Reviews, American Mathematical Society, Ann Arbor, MI 48103, U.S.A. (graeme.fairweather@gmail.com)
§Mathematics and Computing Science, Saint Mary's University, Halifax, NS, B3H 3C3, Canada, (muir.smu@gmail.com)

could be solved efficiently, effectively and in a straightforward manner using a well-developed and widely available software package.

The focus of this report is on a several types of BVPs which are *nonlocal* (NBVPs), i.e., the governing differential equations or boundary conditions contain terms or coefficients that depend on an integral of a function of the solution. The most commonly considered NBVPs, which were introduced in [12, 13], are of the form,

$$-\alpha(q)u''(x) = f(x) \quad x \in (0,1), \quad u(0) = a, \;\; u(1) = b, \tag{1.1}$$

and

$$-\alpha(q)u''(x) + \gamma[u(x)]^{2n+1} = 0, \quad x \in (0,1), \quad u(0) = a, \;\; u(1) = b, \tag{1.2}$$

where

$$q = \int_0^1 u(t)dt,$$

$f(x)$ is a continuous function, $\alpha(q)$ is a continuous, positive function, n is a positive integer, and γ, a and b are constants. It should be noted that the general form (1.2) is slightly different from what is given in [13]. The examples considered in [13] do not agree with the general form given in that paper. In this report, since we consider examples from [13], we modify the general form so that it is consistent with the examples.

Theoretical properties of (1.1) and (1.2) are derived in [12], and, in both [12] and [13], a finite difference method is introduced and shown to be second-order accurate. In [26], the authors extend the work of [12] and provide the results of numerical experiments supporting their results. Moreover, this paper includes a comprehensive collection of problems in which equations of this form arise in the modeling of a broad range of physical phenomena.

In [39], a third type of NBVP is considered, namely,

$$-\phi(q)u''(x) = f(x), \quad x \in (0,1), \quad u(0) = u(1) = 0, \tag{1.3}$$

where

$$q = a\int_0^1 u(t)dt + b\int_0^1 [u'(t)]^2 dt,$$

$f(x)$ is a continuous function, $\phi(q)$ is a positive continuous function, and a and b are constants. The authors investigate theoretical properties of this equation and consider a standard finite difference method for its approximate solution. Numerical results are provided to support the theory.

In [37], the authors consider a fourth type of NBVPs which have the general form,

$$\frac{1}{x^k}\left(x^k u'(x)\right)' = \frac{ku'(x)}{x} + u''(x) = f(k, x, u(x)), \tag{1.4}$$

with boundary conditions,

$$u(0) = \gamma, \quad u(1) = \left(\int_0^1 g(t)u(t)dt\right) + \beta,$$

where $f(k, x, u(x))$ is a continuous function, $g(t)$ is a continuous function, and k, γ and β are constants. The authors prove the uniqueness of the solution, $u(x)$, and present a method for its numerical solution. Numerical results are provided to demonstrate the efficacy (in memory of Pat Keast) of the numerical method.

In [36], the authors consider a fifth type of NBVPs comprising integro-differential TPBVPs of the form,

$$u''(x) = g(x) + \int_a^x K_1(x,t)f_1(t,u(t))dt + \int_a^b K_2(x,t)f_2(t,u(t))dt, \tag{1.5}$$

with boundary conditions,

$$u(a) = \alpha, \quad u(b) = \beta,$$

where a, b, α and β are constants, and $g(x), K_1(x,t), K_2(x,t), f_1(t,u(t))$ and $f_2(t,u(t))$ are functions. The significance of this problem type is that the upper limit of the first integral is x rather than a constant as is the case in the previous problem classes. The authors develop a sufficient condition that guarantees a unique solution of (1.5) and describe a numerical method based on the Adomian decomposition method and the use of Green's function for its solution. An analysis of the proposed numerical method is also provided as are the results of several numerical experiments.

These problem classes as well as several other types of NBVPS have attracted the attention of many researchers in recent years, resulting in a substantial number of papers on the development of various numerical methods for their solution. Examples include, [24], a spectral collocation algorithm using clique functions, a method based on a combination of the Adomian decomposition method and Green's function [36], a technique based on the homotopy analysis method with Green's function [35], the variational method [26], and a semianalytical iterative technique [38]. While there is some value in the theoretical results involving the existence and uniqueness of approximate solutions, it is typically the case that the numerical method developed provides only a discrete approximate solution at a set of points uniformly distributed across the problem domain. In particular, no error estimate is computed, no adaptive error control algorithm is considered, and no continuous approximate solution is provided. Furthermore, the numerical methods are not implemented in a software package that could be used by an applications expert.

A brief outline of this report is as follows. In Section 2, we describe the process by which NBVPs can be converted to a standard form that can be easily solved using any of a number of well-known widely available TPBVP solvers. An important advantage of this approach is that these solvers are able to return an *error-controlled, continuous solution approximation.* This means that *the approximate solution is available to the user at any point in the domain and that a high-quality estimate of the error in the continuous approximate solution satisfies a user-prescribed tolerance.* In Section 3, a selection of test problems from the literature which involve various NBVPs is presented. In Section 4, a widely available software package for solving TPBVPs is described and a recent update of this package is discussed. In Section 5, an extensive collection of numerical examples demonstrate the efficacy of the reformulation approach. Concluding remarks are given in Section 6.

2 Examples of the Reformulation of NBVPs

All of the well-known, widely available TPBVP solvers treat BVPs written as a first-order system of n ODEs of the form

$$\mathbf{y}'(x) = \mathbf{f}\left(x, \mathbf{y}(x)\right), \tag{2.1}$$

with separated boundary conditions,

$$\mathbf{b}_L\left(\mathbf{y}(a)\right) = \mathbf{0}_L, \quad \mathbf{b}_R\left(\mathbf{y}(b)\right) = \mathbf{0}_R, \tag{2.2}$$

where

$$x \in (a,b), \quad \mathbf{y} : \Re \to \Re^n, \quad \mathbf{f} : \Re \times \Re^n \to \Re^n,$$

and

$$\mathbf{b}_L : \Re^n \to \Re^{n_L}, \quad \mathbf{b}_R : \Re^n \to \Re^{n_R}, \mathbf{0}_L \in \Re^{n_L}, \quad \mathbf{0}_R \in \Re^{n_R},$$

and $n_L + n_R = n$.

We first demonstrate the reformulation process for one of the standard forms of NBVPs, namely (1.1). First, we define

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

which implies that $w'(x) = u(x)$ and $w(0) = 0$. Then (1.1) can be rewritten as the system,

$$-\alpha\left(w(1)\right)u''(x) = f(x), \quad w'(x) = u(x), \tag{2.3}$$

with boundary conditions,

$$u(0) = a, \quad w(0) = 0, \quad u(1) = b.$$

Next, we introduce $v(x)$ such that $v'(x) = 0$ and $v(1) = w(1)$. Then (2.3) can be written in the form,

$$u''(x) = -\frac{f(x)}{\alpha\left(v(x)\right)}, \quad w'(x) = u(x), \quad v'(x) = 0,$$

with boundary conditions,

$$u(0) = a, \quad w(0) = 0, \quad u(1) = b, \quad v(1) = w(1).$$

The final step is the conversion to a first-order system of the form, (2.1)-(2.2). To accomplish this, let

$$y_1(x) = u(x), \quad y_2(x) = u'(x), \quad y_3(x) = w(x), \quad y_4(x) = v(x),$$

and note that $y_1'(x) = y_2(x)$. Then the resulting reformulation of (1.1) is

$$y_1'(x) = y_2(x), \quad y_2'(x) = -\frac{f(x)}{\alpha\left(y_4(x)\right)}, \quad y_3'(x) = y_1(x), \quad y_4'(x) = 0,$$

with boundary conditions,

$$y_1(0) = a, \quad y_3(0) = 0, \quad y_1(1) = b, \quad y_4(1) = y_3(1).$$

A similar approach can be used to reformulate problems of the form (1.2).

We next consider the reformulation of (1.3). The approach is similar to that just described except that, to deal with the second integral term, we introduce two additional solution components, namely, $y_5(x)$ and $y_6(x)$, such that

$$y_5(x) = \int_0^x [u'(t)]^2 dt = \int_0^x [y_2(t)]^2 dt,$$

which implies that

$$y_5'(x) = [y_2(x)]^2, \quad y_5(0) = 0,$$

and

$$y_6'(x) = 0, \quad y_6(1) = y_5(1).$$

The resulting reformulation of (1.3) is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \frac{-f(x)}{\phi\left(ay_4(x) + by_6(x)\right)}, \quad y_3'(x) = y_1(x), \quad y_4'(x) = 0, \quad y_5'(x) = [y_2(x)]^2, \quad y_6'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_5(0) = 0, \quad y_1(1) = 0, \quad y_4(1) = y_3(1), \quad y_6(1) = y_5(1).$$

We next consider the reformulation of problems of the form (1.4). As before, we introduce $y_1(x) = u(x)$ and $y_2(x) = u'(x)$. To deal with the integral in the right boundary condition, we introduce

$$y_3(x) = \int_0^x g(t)u(t)dt = \int_0^x g(t)y_1(t)dt,$$

which implies that

$$y_3'(x) = g(x)y_1(x), \quad y_3(0) = 0.$$

The resulting reformulation of (1.4) is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \frac{-ky_2(x)}{x} + f(k, x, y_1(x)), \quad y_3'(x) = g(x)y_1(x),$$

with boundary conditions,

$$y_1(0) = \gamma, \quad y_3(0) = 0, \quad y_1(1) = y_3(1) + \beta.$$

We conclude this section with a description of the reformulation of the problem class (1.5). We begin by introducing $y_1(x) = u(x)$ and $y_2(x) = u'(x)$. We next define

$$y_3(x) = \int_a^x K_1(x,t)f_1(t, u(t))dt = \int_a^x K_1(x,t)f_1(t, y_1(t))dt,$$

which implies that

$$y_3'(x) = K_1(x,x)f_1(x, y_1(x)), \quad y_3(a) = 0.$$

We also define

$$y_4(x) = \int_a^x K_2(x,t)f_2(t,u(t))dt = \int_a^x K_2(x,t)f_2(t,y_1(t))dt,$$

which implies that

$$y_4'(x) = K_2(x,x)f_2(x, y_1(x)), \quad y_4(a) = 0.$$

Finally, we define

$$y_5'(x) = 0, \quad y_5(b) = y_4(b).$$

The resulting reformulation of (1.5) is

$$y_1'(x) = y_2(x), \quad y_2'(x) = g(x) + y_3(x) + y_5(x), \quad y_3'(x) = K_1(x,x)f_1(x, y_1(x)),$$
$$y_4'(x) = K_2(x,x)f_2(x, y_1(x)), \quad y_5'(x) = 0,$$

with boundary conditions,

$$y_1(a) = \alpha, \quad y_3(a) = a, \quad y_4(a) = 0, \quad y_1(b) = \beta, \quad y_5(b) = y_4(b).$$

3 Test Problems

In this section, we consider a collection of NBVPs from the literature and, for each, we provide the reformulated version of the problem in the form (2.1)–(2.2). The first five problems are Problems 1–5 in the battery of problems in [26] and have the general form of (1.1) (problems i)-iii) or (1.2) (problems iv)-v)).

i) [26, Problem 1] With

$$\alpha(q) = q^{\frac{1}{3}}, \quad f(x) = -\left(\frac{6}{\sqrt[3]{4}}\right)x, \quad a = 0, \quad b = 1,$$

the exact solution is

$$u(x) = x^3.$$

The reformulated problem is then

$$y_1'(x) = y_2(x), \quad y_2'(x) = \frac{\left(\frac{6}{\sqrt[3]{4}}\right)x}{[y_4(x)]^{\frac{1}{3}}}, \quad y_3'(x) = y_1(x), \quad y_4'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_1(1) = 1, \quad y_4(1) = y_3(1).$$

ii) [26, Problem 2] Let

$$\alpha(q) = q^2, \quad f(x) = \frac{3}{4}\cos\left(\frac{2\pi}{3}x\right), \quad a = 1, \quad b = -\frac{1}{2}.$$

The exact solution is

$$u(x) = \cos\left(\frac{2\pi}{3}x\right),$$

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \frac{-\frac{3}{4}\cos\left(\frac{2\pi}{3}x\right)}{[y_4(x)]^2}, \quad y_3'(x) = y_1(x), \quad y_4'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_1(1) = -\frac{1}{2}, \quad y_4(1) = y_3(1).$$

iii) [26, Problem 3] With

$$\alpha(q) = (1+q)^2, \quad f(x) = \frac{49}{18}, \quad a = 0, \quad b = 0,$$

the exact solution is

$$u(x) = x(1-x),$$

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = -\frac{\frac{49}{18}}{(1+y_4(x))^2}, \quad y_3'(x) = y_1(x), \quad y_4'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_1(1) = 0, \quad y_4(1) = y_3(1).$$

Problems 4 and 5 are of the general form (1.2)

iv) [26, Problem 4] With

$$\alpha(q) = \frac{1}{q}, \quad \gamma = \left(\frac{3}{4(2\sqrt{2}-2)}\right), \quad n = 2, \quad a = 1, \quad b = \frac{\sqrt{2}}{2},$$

the exact solution is

$$u(x) = \frac{1}{\sqrt{1+x}},$$

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \left(\frac{3}{4(2\sqrt{2}-2)}\right) y_4(x)[y_1(x)]^5, \quad y_3'(x) = y_1(x), \quad y_4'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 1, \quad y_3(0) = 0, \quad y_1(1) = \frac{\sqrt{2}}{2}, \quad y_4(1) = y_3(1).$$

v) [26, Problem 5] With

$$\alpha(q) = q, \quad \gamma = \left(\frac{3(2\sqrt{2}-2)}{4}\right), \quad n = 2, \quad a = 1, \quad b = \frac{\sqrt{2}}{2},$$

the exact solution is

$$u(x) = \frac{1}{\sqrt{1+x}},$$

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \left(\frac{3(2\sqrt{2}-2)}{4}\right) \frac{[y_1(x)]^5}{y_4(x)}, \quad y_3'(x) = y_1(x), \quad y_4'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 1, \quad y_3(0) = 0, \quad y_1(1) = \frac{\sqrt{2}}{2}, \quad y_4(1) = y_3(1).$$

The next six problems are from [39] and each has the general form (1.3).

vi) [39, Problem 1] With

$$\phi(q) = q^2 + 1, \quad a = \frac{1}{2}, \quad b = \frac{1}{10}, \quad f(x) = K(2e^x - 1 + 4x - x^2)e^{-x},$$

where

$$K = \frac{46441}{14400} - \frac{537}{50}e^{-1} + \frac{29851}{2400}e^{-2} + \frac{63}{50}e^{-3} + \frac{49}{1600}e^{-4},$$

the exact solution is

$$u(x) = (1 - x^2)(1 - e^{-x}),$$

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = -\frac{K(2e^x - 1 + 4x - x^2)e^{-x}}{\left[\frac{1}{2}y_4(x) + \frac{1}{10}y_6(x)\right]^2 + 1},$$

$$y_3'(x) = y_1(x), \quad y_4'(x) = 0, \quad y_5'(x) = [y_2(x)]^2, \quad y_6'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_5(0) = 0, \quad y_1(1) = 0, \quad y_4(1) = y_3(1), \quad y_6(1) = y_5(1).$$

vii) [39, Problem 2] With

$$\phi(q) = q^2 + 1, \quad a = 1, \quad b = 1, \quad f(x) = K(2e^x - 1 + 4x - x^2)e^{-x},$$

where

$$K = \frac{1105}{16} - 330e^{-1} + \frac{2969}{8}e^{-2} + 70e^{-3} + \frac{49}{16}e^{-4},$$

the exact solution is

$$u(x) = (1 - x^2)(1 - e^{-x}),$$

and the reformulated problem is

$$\begin{aligned} &y_1'(x) = y_2(x), \\ &y_2'(x) = -\frac{K(2e^x - 1 + 4x - x^2)e^{-x}}{[y_4(x) + y_6(x)]^2 + 1}, \\ &y_3'(x) = y_1(x), \quad y_4'(x) = 0, \quad y_5'(x) = [y_2(x)]^2, \quad y_6'(x) = 0, \end{aligned}$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_5(0) = 0, \quad y_1(1) = 0, \quad y_4(1) = y_3(1), \quad y_6(1) = y_5(1).$$

viii) [39, Problem 3] With

$$\phi(q) = q^{\frac{1}{3}} + 1, \quad a = 1, \quad b = 1, \quad f(x) = K(4 - 5x + x^2)e^{-x},$$

where

$$K = \left(-\frac{3}{4} + 3e^{-1} - \frac{3}{4}e^{-2}\right)^{\frac{1}{3}} + 1,$$

the exact solution is

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

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = -\frac{K(4-5x+x^2)e^{-x}}{[y_4(x)+y_6(x)]^{\frac{1}{3}}+1},$$

$$y_3'(x) = y_1(x), \quad y_4'(x) = 0, \quad y_5'(x) = [y_2(x)]^2, \quad y_6'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_5(0) = 0, \quad y_1(1) = 0, \quad y_4(1) = y_3(1), \quad y_6(1) = y_5(1).$$

ix)[39, Problem 4] With

$$\phi(q) = \frac{1}{1+q} + 3, \quad a = 1, \quad b = 1, \quad f(x) = K(4-5x+x^2)e^{-x},$$

where

$$K = \frac{1}{\frac{1}{4} + 3e^{-1} - \frac{3}{4}e^{-2}} + 3,$$

the exact solution is

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

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = -\frac{K(4-5x+x^2)e^{-x}}{\frac{1}{1+y_4(x)+y_6(x)}+3},$$

$$y_3'(x) = y_1(x), \quad y_4'(x) = 0, \quad y_5'(x) = [y_2(x)]^2, \quad y_6'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_5(0) = 0, \quad y_1(1) = 0, \quad y_4(1) = y_3(1), \quad y_6(1) = y_5(1).$$

The following problems are from [39] and have the general form (1.3).

x) [39, Problem 5] With

$$\phi(q) = qe^{-q} + 2, \qquad a = 1, \quad b = 1, \quad f(x) = K(4-5x+x^2)e^{-x},$$

where

$$K = \left(-\frac{3}{4} + 3e^{-1} - \frac{3}{4}e^{-2}\right) e^{-\left(-\frac{3}{4}+3e^{-1}-\frac{3}{4}e^{-2}\right)} + 2,$$

the exact solution is

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

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = -\frac{K(4-5x+x^2)e^{-x}}{(y_4(x)+y_6(x))\, e^{-(y_4(x)+y_6(x))}+2},$$

$$y_3'(x) = y_1(x), \quad y_4'(x) = 0, \quad y_5'(x) = [y_2(x)]^2, \quad y_6'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_5(0) = 0, \quad y_1(1) = 0, \quad y_4(1) = y_3(1), \quad y_6(1) = y_5(1).$$

xi) [39, Problem 6] With

$$\phi(q) = qe^{-q} + 1, \quad a = 10, \quad b = 10, \quad f(x) = e^{[\sin(x)]^2},$$

the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = -\frac{e^{(\sin(x))^2}}{[10y_4(x) + 10y_6(x)]\, e^{-(10y_4(x)+10y_6(x))} + 1},$$

$$y_3'(x) = y_1(x), \quad y_4'(x) = 0, \quad y_5'(x) = [y_2(x)]^2, \quad y_6'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_5(0) = 0, \quad y_1(1) = 0, \quad y_4(1) = y_3(1), \quad y_6(1) = y_5(1).$$

This example has no known exact solution. The next five problems are from [37], and have the general form (1.4).

xii) [37, Problem 1] With

$$f(k, x, u(x)) = 12x^6 u(x)^5 - 2(3+k)x^2 u(x)^3, \quad g(t) = \frac{t}{2}, \quad \gamma = \frac{1}{2}, \quad \beta = \frac{1}{\sqrt{5}} - \frac{1}{4}\operatorname{arcsinh}\left(\frac{1}{2}\right).$$

The exact solution is

$$u(x) = \frac{1}{\sqrt{4+x^2}}.$$

We choose the parameter $k = \frac{1}{2}$ for our testing. The reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \frac{-ky_2(x)}{x} + 12x^6 y_1(x)^5 - 2(3+k)x^2 y_1(x)^3, \quad y_3'(x) = \frac{x}{2} y_1(x),$$

with boundary conditions,

$$y_1(0) = \frac{1}{2}, \quad y_3(0) = 0, \quad y_1(1) = y_3(1) + \frac{1}{\sqrt{5}} - \frac{1}{4}\operatorname{arcsinh}\left(\frac{1}{2}\right).$$

xiii) [37, Problem 2] With

$$f(k, x, u(x)) = 16x^6 e^{2u(x)} - 4(3+k)x^2 e^{u(x)}, \quad g(t) = \frac{1}{2}, \quad \gamma = \ln\left(\frac{1}{4}\right), \quad \beta = -2 + \arctan(2).$$

The exact solution is

$$u(x) = \ln\left(\frac{1}{4+x^4}\right).$$

We choose the parameter $k = \frac{1}{2}$ for our testing. The reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \frac{-ky_2(x)}{x} + 16x^6 e^{2y_1(x)} - 4(3+k)x^2 e^{y_1(x)}, \quad y_3'(x) = \frac{1}{2} y_1(x),$$

with boundary conditions,

$$y_1(0) = \ln\left(\frac{1}{4}\right), \quad y_3(0) = 0, \quad y_1(1) = y_3(1) - 2 + \arctan(2).$$

xiv) [37, Problem 4] With

$$f(k, x, u(x)) = -[u(x)]^5, \quad g(t) = \frac{1}{100}, \quad k = 2, \quad \beta = -\frac{\sqrt{3}}{100}\left(-50 + \operatorname{arcsinh}\left(\frac{1}{\sqrt{3}}\right)\right),$$

with the boundary condition, $u(0) = \gamma$, replaced by $u'(0) = 0$. The exact solution is

$$u(x) = \sqrt{\frac{3}{3+x^2}},$$

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \frac{-2y_2(x)}{x} - [y_1(x)]^5, \quad y_3'(x) = \frac{1}{100} y_1(x),$$

with boundary conditions,

$$y_1'(0) = 0, \quad y_3(0) = 0, \quad y_1(1) = y_3(1) - \frac{\sqrt{3}}{100}\left(-50 + \operatorname{arcsinh}\left(\frac{1}{\sqrt{3}}\right)\right).$$

xv) [37, Problem 2] With

$$f(k, x, u(x)) = -e^{u(x)}, \quad g(t) = \frac{1}{10}, \quad k = 1, \quad \beta = \frac{1}{20}\left(-8 + (1+\sqrt{2})\pi\right).$$

and again we take $u'(0) = 0$. The exact solution is

$$u(x) = 2\ln\left(\frac{4 - 2\sqrt{2}}{(3 - 2\sqrt{2})x^2 + 1}\right),$$

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \frac{-y_2(x)}{x} - [y_1(x)]^5, \quad y_3'(x) = \frac{1}{10} y_1(x),$$

with boundary conditions,

$$y_1'(0) = 0, \quad y_3(0) = 0, \quad y_1(1) = y_3(1) + \frac{1}{20}\left(-8 + (1+\sqrt{2})\pi\right).$$

xvi) [37, Problem 7] Here,

$$f(k, x, u(x)) = xe^{2u(x)} - ke^{u(x)}, \quad g(t) = \frac{1}{4}, \quad \gamma = \ln\left(\frac{1}{2}\right) \quad \beta = \ln\left(\frac{1}{3}\right) + \frac{1}{4}\left(-1 + \ln\left(\frac{27}{4}\right)\right),$$

For this problem, the left-hand side of the ODE differs slightly from the general form (1.4); it is

$$\frac{1}{x^{k-1}}\left(x^k u'(x)\right)'.$$

We choose the parameter $k = \frac{1}{2}$ in our numerical experiments. The exact solution is then

$$u(x) = \ln\left(\frac{1}{2+x}\right)$$

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \frac{-ky_2(x) + xe^{2y_1(x)} - ke^{y_1(x)}}{x}, \quad y_3'(x) = \frac{1}{4} y_1(x),$$

with boundary conditions,

$$y_1(0) = \ln\left(\frac{1}{2}\right), \quad y_3(0) = 0, \quad y_1(1) = y_3(1) + \ln\left(\frac{1}{3}\right) + \frac{1}{4}\left(-1 + \ln\left(\frac{27}{4}\right)\right).$$

The next three problems are from [36] and have the general form (1.5).

xvii)[36, Problem 1] With

$$g(x) = 1, \quad K_1(x,t) = e^{-t}, \quad f_1(t,u(t)) = u^2(t), \quad K_2(x,t) = 0, \quad f_2(t,u(t)) = 0,$$

and $a = 0, b = 1, \alpha = 1, \beta = e$, the exact solution is

$$u(x) = e^x.$$

and the reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = 1 + y_3(x), \quad y_3'(x) = e^{-x} y_1^2(x),$$

with boundary conditions,

$$y_1(0) = 1, \quad y_3(0) = 0, \quad y_1(1) = e.$$

xviii) [36, Problem 2] With

$$g(x) = -2x^2 - \frac{x^3}{6} - \frac{1}{(4+x)^2}, \quad K_1(x,t) = x - t, \quad f_1(t,u(t)) = e^{u(t)}, \quad K_2(x,t) = 0, \quad f_2(t,u(t)) = 0,$$

and $a = 0, b = 1, \alpha = \ln(4), \beta = \ln(5)$, the exact solution is

$$u(x) = \ln(4 + x).$$

Since $K_1(x,x) = 0$, we depart slightly from the standard formulation and define

$$y_3(x) = \int_0^x e^{u(t)} dt, \quad y_4(x) = \int_0^x t e^{u(t)} dt,$$

with $y_3(0) = 0$ and $y_4(0) = 0$. Then

$$y_3'(x) = e^{u(x)}, \quad y_4'(x) = x e^{u(x)}.$$

The reformulated problem is

$$y_1'(x) = y_2(x), \quad y_2'(x) = \left(-2x^2 - \frac{x^3}{6} - \frac{1}{(4+x)^2}\right) + x y_3(x) - y_4(x), \quad y_3'(x) = e^{y_1(x)}, \quad y_4'(x) = x e^{y_1(x)},$$

with boundary conditions,

$$y_1(0) = \ln(4), \quad y_3(0) = 0, \quad y_4(0) = 0, \quad y_1(1) = \ln(5).$$

xix)[36, Problem 3] With

$$g(x) = \frac{-3x}{2} - \frac{1}{(1+x)^2} + \frac{5}{6}, \quad K_1(x,t) = 0, \quad f_1(t,u(t)) = 0, \quad K_2(x,t) = x - t, \quad f_2(t,u(t)) = e^{u(t)},$$

and $a = 0, b = 1, \alpha = 0, \beta = \ln(2)$, the exact solution is

$$u(x) = \ln(1 + x).$$

Since $K_2(x,x) = 0$, we again depart slightly from the standard formulation and define

$$y_3(x) = \int_0^x e^{u(t)} dt, \quad y_4(x) = \int_0^x t e^{u(t)} dt,$$

with $y_3(0) = 0$ and $y_4(0) = 0$. Then

$$y_3'(x) = e^{u(x)}, \quad y_4'(x) = xe^{u(t)}.$$

We also define $y_5'(x) = 0$ and $y_6'(x) = 0$, with $y_5(1) = y_3(1)$ and $y_6(1) = y_4(1)$. The reformulated problem is then

$$y_1'(x) = y_2(x), \quad y_2'(x) = \left(\frac{-3x}{2} - \frac{1}{(1+x)^2} + \frac{5}{6}\right) + xy_5(x) - y_6(x), \quad y_3'(x) = e^{y_1(x)},$$
$$y_4'(x) = xe^{y_1(x)}, \quad y_5'(x) = 0, \quad y_6'(x) = 0,$$

with boundary conditions,

$$y_1(0) = 0, \quad y_3(0) = 0, \quad y_4(0) = 0, \quad y_1(1) = \ln(2), \quad y_5(1) = y_3(1), \quad y_6(1) = y_4(1).$$

4 Description of the Software

Once a problem has been reformulated, it has a form that allows it to be solved by any one of a number of well-known BVP solvers. Examples of such solvers include those provided at the Test Set for BVP Solvers website [1]: TPBVPC [16, 14], TPBVPLC [15, 16], ACDCC [17], COLYSYS [5], COLNEW [9], COLMOD [17, 40], and BVP_SOLVER-2 [11]. In this study, we use a new version of the COLNEW package, known as COLNEWSC [2, 3]. Here, we begin with a description of the COLNEW package followed by a description of the COLNEWSC package.

4.1 COLNEW

COLNEW was developed from the earlier package, COLSYS [5]. Both packages assume the existence of a mesh of points that partition the problem domain and then represent the approximate solution in terms of a *known* C^0-continuous piecewise polynomial basis of a specified degree p defined on the mesh. The unknown coefficients are determined from the solution of a nonlinear system consisting of boundary conditions and *collocation* conditions. The boundary conditions are obtained by requiring that the approximate solution satisfy the given BVODE boundary conditions (2.2). The collocation conditions are obtained by requiring that the approximate solution satisfy the ODEs (2.1) at $p-1$ points on each subinterval of the mesh. These collocation points are chosen to be the images of the set of $p-1$ Gauss points mapped onto each subinterval. The resulting nonlinear system is solved using a Newton-type algorithm; see [7] for details. This gives a C^0-continuous approximate solution across the problem domain.

Based on this approximate solution, an error estimation algorithm is employed to obtain a rough estimate of the error of the continuous collocation solution on each mesh subinterval. If the subinterval error estimates are sufficiently different in magnitude, a new mesh is constructed that approximately equidistributes these error estimates over the subintervals of the new mesh. Otherwise, a new mesh is obtained by "doubling" the current mesh; this means that each subinterval of the current mesh is split in half. In most cases, COLSYS/COLNEW obtain the new mesh by doubling the current mesh. If the current mesh is a doubling of the previous mesh, then the codes compute a more reliable error estimate, based on the approximate solution from the previous mesh and the approximate solution from the current mesh, using a Richardson extrapolation algorithm. If this error estimate satisfies the user tolerance, then the computation terminates. Otherwise, the computation continues, using the new mesh. See [7] for details.

For a first-order BVODE system (2.1), it can be shown that the error of the collocation solution at an arbitrary point in the problem domain has an error that is $O(h^{p+1})$, where h is the maximum subinterval size. Since the solvers employ collocation at Gauss points, it can also be shown that, at the mesh points, the error of the collocation solution is $O(h^{2(p-1)})$. Thus, for larger p values, the accuracy of the solution at the mesh points is substantially higher than at an arbitrary point; the mesh point solution values are said to be *superconvergent*. It can also be shown that the first derivative of the collocation solution is superconvergent at the mesh points.

COLSYS represents the piecewise polynomial basis using a B-spline basis [19], whereas COLNEW employs a monomial basis. Due to the differences in the basis representation, each solver employs different linear algebra software to solve the Newton systems that arise. See [7] for details.

4.2 COLNEWSC

COLNEWSC is a modification of COLNEW that exploits the existence of the superconvergent mesh-point collocation solution and derivative values in order to obtain a C^1-continuous superconvergent approximation throughout the problem domain. Initially, the COLNEWSC algorithm performs the same steps as COLNEW: a collocation solution is computed on the current mesh as described previously. However, once this approximate solution is obtained, the error estimates and the associated mesh refinement algorithm are based on estimates of the error of the collocation solution *only at the mesh points*. Because the error of the collocation solution at the mesh points is substantially smaller than the error of the continuous collocation solution across the problem domain, COLNEWSC is able to terminate on a much coarser mesh than is COLNEW. Moreover, since the cost computational cost is proportional to the number of mesh subintervals, COLNEWSC can be substantially more efficient than COLNEW.

Once the computation that depends on the mesh-point error estimates has terminated, COLNEWSC constructs a superconvergent solution approximation based on the collocation solution and the use of continuous mono-implicit Rung-Kutta methods [30]. The error of this superconvergent interpolant is $O(h^{2(p-1)})$, the same as the order of accuracy of the collocation solution at the mesh points. The algorithm that COLNEWSC uses to construct the superconvergent interpolant is described in [21].

Once the superconvergent interpolant has been constructed, a Richardson extrapolation-type algorithm is employed to obtain an estimate of the error of the superconvergent interpolant across the problem domain. If this error estimate satisfies the tolerance, then the computation terminates. In most cases, this error estimate does in fact satisfy the tolerance. However, occasionally, it does not, and, in such cases, the computation continues, using a new mesh obtained by doubling the current mesh. The computation terminates when the error estimate for the superconvergent interpolant satisfies the user tolerance.

In [2], COLNEWSC is shown to generally be substantially more efficient than either COLSYS, COLNEW, or BVP_SOLVER-2.

5 Numerical Experiments

In this section, we provide numerical results based on the application of COLNEWSC to the reformulated test problems introduced in Section 3. In the papers where these problems appear, numerical results are generally provided in the form of a table giving, for a uniform set of 10 points distributed across the problem domain, the exact solution value, the approximate solution value, and the absolute error. Consequently, we present our numerical results in a similar manner.

We use COLNEWSC with the number of collocation points per subinterval equal to 4, a uniform initial mesh of 5 subintervals, and a tolerance of 10^{-6}. For each test problem, we provide the initial approximation to the solution.

For each problem, COLNEWSC easily obtains a continuous numerical solution with an error estimate satisfying the desired tolerance. The solver first computes an approximate solution on the initial mesh of 5 subintervals. In order to obtain an estimate of the error, the solver then computes a second approximate solution on a uniform mesh of 10 subintervals. For every test problem, this error estimate satisfies the tolerance.

A typical execution time is approximately 0.01 seconds on an Intel(R) Xeon(R) CPU E7-8860 v3, 2.20GHz, 8 core, x86_64 processor with 48G of RAM running Ubuntu 22.04 LTS. The Fortran compiler is version 11.4.0 of gfortran.

In Table 1, we provide the initial solution approximations for each test problem from Section 3.

5.1 Numerical Results

Numerical results for each test problem from Section 2 are provided in Tables 3-20.

	$y_1(x)$	$y_2(x)$	$y_3(x)$	$y_4(x)$	$y_5(x)$	$y_6(x)$
i) [26, Problem 1]	x	1	$\frac{x^2}{2}$	$\frac{1}{2}$	-	-
ii) [26, Problem 2]	$1-x$	-1	$x-\frac{x^2}{2}$	$\frac{1}{2}$	-	-
iii) [26, Problem 3]	1	0	x	1	-	-
iv) [26, Problem 4]	$1+x$	1	$x+\frac{x^2}{2}$	$\frac{3}{2}$	-	-
v) [26, Problem 5]	$1+x$	1	$x+\frac{x^2}{2}$	$\frac{3}{2}$	-	-
vi) [39, Problem 1]	$1+x$	1	$x+\frac{x^2}{2}$	$\frac{3}{2}$	$x+\frac{x^2}{2}$	$\frac{3}{2}$
vii)[39, Problem 2]	$1+x$	1	$x+\frac{x^2}{2}$	$\frac{3}{2}$	$x+\frac{x^2}{2}$	$\frac{3}{2}$
viii) [39, Problem 3]	$x(1-x)$	1-2x	$-\frac{x^3}{3}+\frac{x^2}{2}$	$\frac{1}{6}$	$(2x-1)^2$	1
ix) [39, Problem 4]	$x(1-x)$	1-2x	$-\frac{x^3}{3}+\frac{x^2}{2}$	$\frac{1}{6}$	$(2x-1)^2$	1
x)[39, Problem 5]	$x(1-x)$	1-2x	$-\frac{x^3}{3}+\frac{x^2}{2}$	$\frac{1}{6}$	$(2x-1)^2$	1
xi) [39, Problem 6]	$x(1-x)$	1-2x	$-\frac{x^3}{3}+\frac{x^2}{2}$	$\frac{1}{6}$	$(2x-1)^2$	1
xii) [37, Problem 1]	$\frac{1}{2}-\frac{x}{20}$	$-\frac{1}{20}$	$\frac{x}{2}-\frac{x^2}{40}$	-	-	-
xiii) [37, Problem 2]	$-1.4-0.2x$	-0.2	$-1.4-0.2x$	-	-	-
xiv) [37, Problem 4]	$1-x$	-1	$x-\frac{x^2}{2}$	-	-	-
xv) [37, Problem 5]	$0.32-0.3x^2$	-0.6	$0.32x-0.1x^3$	-	-	-
xvi) [37, Problem 7]	$-0.7-0.4x$	-0.4	$-0.7x-0.2x^2$	-	-	-
xvii) [36, Problem 1]	$1+x$	1	$\frac{x^2}{2}+x$	-	-	-
xviii) [36, Problem 2]	$1+x$	1	$\frac{x^2}{2}+x$	$\frac{x^3}{2}+x^2$	-	-
xix) [36, Problem 3]	x	1	$\frac{x^2}{2}$	$\frac{x^3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$

Table 1: Initial solution approximations for test problems from Section 2.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	0.000E+00	0.000E+00	0.000E+00
0.1	1.000E-03	1.000E-03	2.162E-16
0.2	8.000E-03	8.000E-03	4.198E-16
0.3	2.700E-02	2.700E-02	3.319E-14
0.4	6.400E-02	6.400E-02	7.633E-16
0.5	1.250E-01	1.250E-01	7.910E-16
0.6	2.160E-01	2.160E-01	7.652E-14
0.7	3.430E-01	3.430E-01	1.062E-13
0.8	5.120E-01	5.120E-01	6.661E-16
0.9	7.290E-01	7.290E-01	4.441E-16
1.0	1.000E+00	1.000E+00	1.941E-13

Table 2: **[26, Problem 1]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error .

x	Exact Solution	COLNEWSC	Absolute Error
0.0	1.0000000000E+00	1.0000000000E+00	0.000E+00
0.1	9.7814760073E-01	9.7814760073E-01	9.215E-15
0.2	9.1354545764E-01	9.1354545764E-01	1.554E-14
0.3	8.0901699437E-01	8.0901699438E-01	1.991E-13
0.4	6.6913060636E-01	6.6913060636E-01	1.932E-14
0.5	5.0000000000E-01	5.0000000000E-01	1.776E-14
0.6	3.0901699437E-01	3.0901699438E-01	1.715E-13
0.7	1.0452846327E-01	1.0452846327E-01	1.784E-13
0.8	-1.0452846327E-01	-1.0452846327E-01	6.523E-15
0.9	-3.0901699437E-01	-3.0901699437E-01	2.720E-15
1.0	-5.0000000000E-01	-5.0000000000E-01	1.292E-13

Table 3: **[26, Problem 2]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error .

x	Exact Solution	COLNEWSC	Absolute Error
0.0	0.000E+00	0.000E+00	0.000E+00
0.1	9.000E-02	9.000E-02	1.388E-16
0.2	1.600E-01	1.600E-01	1.943E-16
0.3	2.100E-01	2.100E-01	8.765E-14
0.4	2.400E-01	2.400E-01	3.331E-16
0.5	2.500E-01	2.500E-01	3.886E-16
0.6	2.400E-01	2.400E-01	9.298E-15
0.7	2.100E-01	2.100E-01	2.559E-14
0.8	1.600E-01	1.600E-01	2.220E-16
0.9	9.000E-02	9.000E-02	1.388E-16
1.0	0.000E+00	6.455E-14	6.455E-14

Table 4: **[26, Problem 3]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error .

x	Exact Solution	COLNEWSC	Absolute Error
0.0	1.0000000000E+00	1.0000000000E+00	0.000E+00
0.1	9.5346258925E-01	9.5346258925E-01	9.770E-14
0.2	9.1287092918E-01	9.1287092918E-01	1.215E-13
0.3	8.7705801931E-01	8.7705801931E-01	5.529E-14
0.4	8.4515425473E-01	8.4515425473E-01	1.042E-13
0.5	8.1649658093E-01	8.1649658093E-01	8.726E-14
0.6	7.9056941504E-01	7.9056941504E-01	4.807E-14
0.7	7.6696498885E-01	7.6696498885E-01	3.220E-14
0.8	7.4535599250E-01	7.4535599250E-01	3.408E-14
0.9	7.2547625011E-01	7.2547625011E-01	1.688E-14
1.0	7.0710678119E-01	7.0710678119E-01	1.266E-14

Table 5: **[26, Problem 4]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	1.0000000000E+00	1.0000000000E+00	0.000E+00
0.1	9.5346258925E-01	9.5346258925E-01	1.045E-13
0.2	9.1287092918E-01	9.1287092918E-01	1.327E-13
0.3	8.7705801931E-01	8.7705801931E-01	6.894E-14
0.4	8.4515425473E-01	8.4515425473E-01	1.191E-13
0.5	8.1649658093E-01	8.1649658093E-01	1.018E-13
0.6	7.9056941504E-01	7.9056941504E-01	6.140E-14
0.7	7.6696498885E-01	7.6696498885E-01	4.341E-14
0.8	7.4535599250E-01	7.4535599250E-01	4.208E-14
0.9	7.2547625011E-01	7.2547625011E-01	2.132E-14
1.0	7.0710678119E-01	7.0710678119E-01	1.255E-14

Table 6: **[26, Problem 5]**.For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	0.0000000000E+00	0.0000000000E+00	0.000E+00
0.1	9.4210956144E-02	9.4210956144E-02	4.163E-17
0.2	1.7401847705E-01	1.7401847705E-01	1.110E-16
0.3	2.3585541918E-01	2.3585541918E-01	1.082E-13
0.4	2.7693116133E-01	2.7693116133E-01	2.776E-16
0.5	2.9510200522E-01	2.9510200522E-01	2.776E-16
0.6	2.8876055290E-01	2.8876055290E-01	6.439E-15
0.7	2.5674149507E-01	2.5674149507E-01	2.753E-14
0.8	1.9824157292E-01	1.9824157292E-01	2.498E-16
0.9	1.1275176465E-01	1.1275176465E-01	1.388E-16
1.0	0.0000000000E+00	8.1171180888E-14	8.117E-14

Table 7: **[39, Problem 1]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	0.0000000000E+00	0.0000000000E+00	0.000E+00
0.1	9.4210956144E-02	9.4210956144E-02	1.166E-15
0.2	1.7401847705E-01	1.7401847705E-01	2.054E-15
0.3	2.3585541918E-01	2.3585541918E-01	1.112E-13
0.4	2.7693116133E-01	2.7693116133E-01	3.164E-15
0.5	2.9510200522E-01	2.9510200522E-01	3.386E-15
0.6	2.8876055290E-01	2.8876055290E-01	2.887E-15
0.7	2.5674149507E-01	2.5674149507E-01	2.420E-14
0.8	1.9824157292E-01	1.9824157292E-01	2.248E-15
0.9	1.1275176465E-01	1.1275176465E-01	1.249E-15
1.0	0.0000000000E+00	8.1226692039E-14	8.123E-14

Table 8: **[39, Problem 2]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	0.0000000000E+00	0.0000000000E+00	0.000E+00
0.1	8.1435367623E-02	8.1435367623E-02	5.329E-15
0.2	1.3099692049E-01	1.3099692049E-01	8.604E-15
0.3	1.5557182634E-01	1.5557182634E-01	5.412E-14
0.4	1.6087681105E-01	1.6087681105E-01	1.057E-14
0.5	1.5163266493E-01	1.5163266493E-01	9.909E-15
0.6	1.3171479266E-01	1.3171479266E-01	7.966E-15
0.7	1.0428291380E-01	1.0428291380E-01	1.582E-14
0.8	7.1892634259E-02	7.1892634259E-02	4.580E-15
0.9	3.6591269377E-02	3.6591269377E-02	2.227E-15
1.0	0.0000000000E+00	2.5444923946E-14	2.544E-14

Table 9: [**39, Problem 3**]. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	0.0000000000E+00	0.0000000000E+00	0.000E+00
0.1	8.1435367623E-02	8.1435367623E-02	2.720E-15
0.2	1.3099692049E-01	1.3099692049E-01	4.385E-15
0.3	1.5557182634E-01	1.5557182634E-01	3.869E-14
0.4	1.6087681105E-01	1.6087681105E-01	5.412E-15
0.5	1.5163266493E-01	1.5163266493E-01	5.135E-15
0.6	1.3171479266E-01	1.3171479266E-01	2.107E-14
0.7	1.0428291380E-01	1.0428291380E-01	2.615E-14
0.8	7.1892634259E-02	7.1892634259E-02	2.484E-15
0.9	3.6591269377E-02	3.6591269377E-02	1.291E-15
1.0	0.0000000000E+00	2.5361657219E-14	2.536E-14

Table 10: [**39, Problem 4**]. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	0.0000000000E+00	0.0000000000E+00	0.000E+00
0.1	8.1435367623E-02	8.1435367623E-02	2.859E-15
0.2	1.3099692049E-01	1.3099692049E-01	4.635E-15
0.3	1.5557182634E-01	1.5557182634E-01	4.943E-14
0.4	1.6087681105E-01	1.6087681105E-01	5.634E-15
0.5	1.5163266493E-01	1.5163266493E-01	5.301E-15
0.6	1.3171479266E-01	1.3171479266E-01	1.199E-14
0.7	1.0428291380E-01	1.0428291380E-01	1.898E-14
0.8	7.1892634259E-02	7.1892634259E-02	2.429E-15
0.9	3.6591269377E-02	3.6591269377E-02	1.256E-15
1.0	0.0000000000E+00	2.5299207174E-14	2.530E-14

Table 11: [**39, Problem 5**]. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	COLNEWSC
0.0	0.0000000000E+00
0.1	4.6067710747E-02
0.2	8.3446843402E-02
0.3	1.1187728886E-01
0.4	1.3092051911E-01
0.5	1.3995348393E-01
0.6	1.3816302111E-01
0.7	1.2454394102E-01
0.8	9.7904930908E-02
0.9	5.6887106418E-02
1.0	4.1300296516E-14

Table 12: **[39, Problem 6]**. For $x = 0.0, 0.1, \ldots, 1.0$, COLNEWSC Solution. These results agree with the plot of the approximate solution shown in Fig. 2 from [39].

x	Exact Solution	COLNEWSC	Absolute Error
0.0	5.0000000000E-01	5.0000000000E-01	0.000E+00
0.1	4.9999375012E-01	4.9999375012E-01	3.723E-13
0.2	4.9990002999E-01	4.9990002999E-01	3.678E-13
0.3	4.9949451757E-01	4.9949451757E-01	3.751E-13
0.4	4.9840763927E-01	4.9840763927E-01	4.047E-13
0.5	4.9613893836E-01	4.9613893836E-01	4.541E-13
0.6	4.9209166199E-01	4.9209166199E-01	4.937E-13
0.7	4.8563720436E-01	4.8563720436E-01	4.746E-13
0.8	4.7621207360E-01	4.7621207360E-01	3.673E-13
0.9	4.6343502188E-01	4.6343502188E-01	2.072E-13
1.0	4.4721359550E-01	4.4721359550E-01	9.537E-14

Table 13: **[37, Problem 1]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error

x	Exact Solution	COLNEWSC	Absolute Error
0.0	-1.3862943611E+00	-1.3862943611E+00	0.000E+00
0.1	-1.3863193608E+00	-1.3863193608E+00	9.122E-13
0.2	-1.3866942811E+00	-1.3866942811E+00	8.655E-13
0.3	-1.3883173136E+00	-1.3883173136E+00	8.527E-13
0.4	-1.3926739681E+00	-1.3926739681E+00	8.973E-13
0.5	-1.4017985477E+00	-1.4017985477E+00	9.948E-13
0.6	-1.4181805500E+00	-1.4181805500E+00	1.088E-12
0.7	-1.4445868539E+00	-1.4445868539E+00	1.090E-12
0.8	-1.4837839824E+00	-1.4837839824E+00	9.304E-13
0.9	-1.5381781879E+00	-1.5381781879E+00	6.128E-13
1.0	-1.6094379124E+00	-1.6094379124E+00	3.191E-13

Table 14: **[37, Problem 2]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	1.0000000000E+00	1.0000000000E+00	1.277E-14
0.1	9.9833748846E-01	9.9833748846E-01	1.186E-13
0.2	9.9339926780E-01	9.9339926780E-01	1.071E-13
0.3	9.8532927816E-01	9.8532927816E-01	1.045E-13
0.4	9.7435470369E-01	9.7435470369E-01	7.161E-14
0.5	9.6076892283E-01	9.6076892283E-01	5.285E-14
0.6	9.4491118252E-01	9.4491118252E-01	4.907E-14
0.7	9.2714554082E-01	9.2714554082E-01	3.642E-14
0.8	9.0784129900E-01	9.0784129900E-01	1.044E-14
0.9	8.8735650942E-01	8.8735650942E-01	1.332E-15
1.0	8.6602540378E-01	8.6602540378E-01	9.104E-15

Table 15: **[37, Problem 4]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	3.1669436764E-01	3.1669436764E-01	4.419E-14
0.1	3.1326585050E-01	3.1326585050E-01	4.746E-14
0.2	3.0301542283E-01	3.0301542283E-01	4.680E-14
0.3	2.8604726530E-01	2.8604726530E-01	4.652E-14
0.4	2.6253112746E-01	2.6253112746E-01	4.563E-14
0.5	2.3269678387E-01	2.3269678387E-01	4.458E-14
0.6	1.9682680569E-01	1.9682680569E-01	4.355E-14
0.7	1.5524810668E-01	1.5524810668E-01	4.199E-14
0.8	1.0832276344E-01	1.0832276344E-01	4.083E-14
0.9	5.6438602469E-02	5.6438602469E-02	3.932E-14
1.0	0.0000000000E+00	-3.7747582837E-14	3.775E-14

Table 16: **[37, Problem 5]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	-6.9314718056E-01	-6.9314718056E-01	0.000E+00
0.1	-7.4193734473E-01	-7.4193734473E-01	4.630E-14
0.2	-7.8845736036E-01	-7.8845736036E-01	1.121E-14
0.3	-8.3290912294E-01	-8.3290912294E-01	7.261E-14
0.4	-8.7546873735E-01	-8.7546873735E-01	2.665E-15
0.5	-9.1629073187E-01	-9.1629073187E-01	1.887E-15
0.6	-9.5551144503E-01	-9.5551144503E-01	3.075E-14
0.7	-9.9325177301E-01	-9.9325177301E-01	2.964E-14
0.8	-1.0296194172E+00	-1.0296194172E+00	1.110E-15
0.9	-1.0647107370E+00	-1.0647107370E+00	8.882E-16
1.0	-1.0986122887E+00	-1.0986122887E+00	2.265E-14

Table 17: **[37, Problem 7]**. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	1.000E+00	1.000E+00	0.000E+00
0.1	1.105E+00	1.105E+00	0.000E+00
0.2	1.221E+00	1.221E+00	2.220E-16
0.3	1.350E+00	1.350E+00	2.236E-13
0.4	1.492E+00	1.492E+00	4.441E-16
0.5	1.649E+00	1.649E+00	6.661E-16
0.6	1.822E+00	1.822E+00	1.430E-13
0.7	2.014E+00	2.014E+00	1.581E-13
0.8	2.226E+00	2.226E+00	4.441E-16
0.9	2.460E+00	2.460E+00	4.441E-16
1.0	2.718E+00	2.718E+00	1.821E-13

Table 18: [**36, Problem 1**]. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	1.386E+00	1.386E+00	0.000E+00
0.1	1.411E+00	1.411E+00	0.000E+00
0.2	1.435E+00	1.435E+00	0.000E+00
0.3	1.459E+00	1.459E+00	4.086E-14
0.4	1.482E+00	1.482E+00	4.441E-16
0.5	1.504E+00	1.504E+00	4.441E-16
0.6	1.526E+00	1.526E+00	1.821E-14
0.7	1.548E+00	1.548E+00	1.799E-14
0.8	1.569E+00	1.569E+00	4.441E-16
0.9	1.589E+00	1.589E+00	2.220E-16
1.0	1.609E+00	1.609E+00	1.377E-14

Table 19: [**36, Problem 2**]. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

x	Exact Solution	COLNEWSC	Absolute Error
0.0	0.000E+00	0.000E+00	0.000E+00
0.1	9.531E-02	9.531E-02	8.127E-14
0.2	1.823E-01	1.823E-01	1.055E-13
0.3	2.624E-01	2.624E-01	3.386E-14
0.4	3.365E-01	3.365E-01	9.587E-14
0.5	4.055E-01	4.055E-01	8.116E-14
0.6	4.700E-01	4.700E-01	1.199E-14
0.7	5.306E-01	5.306E-01	1.776E-15
0.8	5.878E-01	5.878E-01	3.075E-14
0.9	6.419E-01	6.419E-01	1.499E-14
1.0	6.931E-01	6.931E-01	3.508E-14

Table 20: [**36, Problem 3**]. For $x = 0.0, 0.1, \ldots, 1.0$, Exact Solution, COLNEWSC Solution and Absolute Error.

5.2 Discussion of Numerical Results

From the results presented in the previous subsection, it can be seen that all test problems, once they have been reformulated, can be solved using a state-of-the-art TPBVP solver, with little effort required on the part of the solver. None of the computations require more than about 0.01 of a second. The resultant continuous solutions are shown to be very accurate; the errors are almost at the level of machine precision.

On the other hand, the corresponding results from [26, 39, 37, 36] involve complicated numerical methods that deliver only discrete approximate solutions for which the corresponding errors are typically shown to be in the range from 10^{-4} to 10^{-6}.

6 Concluding Remarks

In this report, we have reformulated numerous NBVPs from the literature so that they are amenable to solution using high-quality software packages. The results of numerical experiments using the software package, COLNEWSC, involving problems from the literature demonstrate the efficacy of our approach. For all the test problems considered in this report, we are able to obtain, in a straightforward manner, much more accurate solutions than those reported in the literature. Furthermore, the solutions we obtain are continuous across the problem domain and have been computed in an error-controlled framework, which means that an estimate of the error of the approximate solution satisfies a tolerance requirement. On the other hand, the algorithms considered in the literature in this area provide only a discrete solution approximate with no error estimate or algorithm for adapting the computation to meet a requested user tolerance.

In general, it would be exceedingly difficult to reproduce results presented in many of the papers cited herein because Matlab, Mathematica or Maple is used in the analysis and/or numerical experiments without any details being provided. Invariably authors claim that, based on the results of numerical experiments, their method is more accurate than another without considering computational cost. Moreover, in many papers, it is claimed that the method under consideration is efficient without any indication of how this property is measured.

There are many other classes of problems, in addition to NBVPs, that can be solved using the software considered in this report; for example,

- Bratu-type problems [8, 25, 33, 34, 35]:

$$y'' + \delta e^y = 0, \quad x \in (0,1), \qquad y(0) = y(1) = 0.$$

  An example of the solution of a Bratu problem using a software package is given in [32].

- Troesch's problem [27, 35]:

$$y'' = \lambda \sinh(\lambda y), \quad x \in (0,1), \qquad y(0) = 0, \;\; y(1) = 1.$$

- Thomas-Fermi equations [31, 37]:

$$\frac{1}{q(x)}(p(x)y'(x))' = f(x,y), \quad x \in (0,1), \qquad y(0) = \gamma, \;\; y(1) = \int_0^1 g(s)y(s)ds + \beta.$$

- The Lane-Emden-Fowler equation [28, 37]:

$$y'' + \frac{2}{x}y' + f(x,y) = g(x), \quad x \in (0,1), \qquad y(0) = 0, \;\; y'(0) = 0.$$

  It would be straightforward to solve these problems using widely available state-of-the-art TPBVP software.

References

[1] Test Set for BVP Solvers. *https://archimede.uniba.it/∼bvpsolvers/testsetbvpsolvers/*.

[2] M. Adams and P. Muir. Differential equation software for the computation of error-controlled continuous approximate solutions. *Numer. Alg.*, 2024.

[3] M. Adams, C. Tannahill, and P. Muir. Error control Gaussian collocation software for boundary value ODEs and 1D time-dependent PDEs. *Numer. Alg.*, 81:1505–1519, 2019.

[4] A. Akyurtlu, J. F. Arkyurtlu, C. E. Hamrin, and G. Fairweather. Reformulation and numerical solution of the equations for a catalytic, porous wall, gas-liquid reactor. *Computers Chem. Eng.*, 10:361–365, 1986.

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

[6] U. Ascher and R. D. Russell. Reformulation of boundary value problems into "standard" form. *SIAM Rev.*, 23:238–254, 1981.

[7] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Dfferential Euations*, volume 13 of *Classics in Applied Mathematics.* (SIAM), Philadelphia, PA, 1995.

[8] R. Assadi, S. Khuri, and I. Louhichi. Numerical solution of fractional Bratu-type BVPs: a generalized variational iterative approach. *Int. J. Heat. Fluid Flow*, 31:293–307, 2021.

[9] G. Bader and U. Ascher. A new basis implementation for a mixed order boundary value ODE solver. *SIAM J. Sci. Stat. Comput.*, 8:483–500, 1987.

[10] D. Bhattacharyya, M. Jevtitch, J. T. Schrodt, and G. Fairweather. Prediction of membrane separation characteristics by pore distribution measurements and surface force-pore flow model. *Chem. Eng. Commun.*, 42(1-3):111–128, 1986.

[11] J. J. Boisvert, P. H. Muir, and R. J. Spiteri. A Runge-Kutta BVODE solver with global error and defect control. *ACM Trans. Math. Software*, 39:Art. 11, 22, 2013.

[12] J. R. Cannon and D. J. Galiffa. A numerical method for a nonlocal elliptic boundary value problem. *J. Integral Equ. Appl.*, 20:243–261, 2008.

[13] J. R. Cannon and D. J. Galiffa. On a numerical method for a homogeneous, nonlinear, nonlocal, elliptic boundary value problem. *Nonlinear Anal.*, 74:1702–1713, 2011.

[14] J. R. Cash and F. Mazzia. A new mesh selection algorithm, based on conditioning, for two-point boundary value codes. *J. Comput. Appl. Math.*, 184:362–381, 2005.

[15] J. R. Cash and F. Mazzia. Hybrid mesh selection algorithms based on conditioning for two-point boundary value problems. *J. Numer. Anal. Indust. Appl. Math*, 1:81–90, 2006.

[16] J. R. Cash and F. Mazzia. Conditioning and hybrid mesh selection algorithms for two-point boundary value problems. *Scalable Comput. Prac.*, 10, 2009.

[17] J. R. Cash, G. Moore, and R.W. Wright. An automatic continuation strategy for the solution of singularly perturbed linear two-point boundary value problems. *J. Comput. Phys.*, 122:266–279, 1995.

[18] M. Davis and G. Fairweather. On the use of spline collocation for boundary value problems arising in chemical engineering. *Comput. Methods Appl. Mech. Eng.*, 28:179–189, 1981.

[19] C. de Boor. *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 1978.

[20] K. S. Denison, C. E. Hamrin Jr., and G. Fairweather. Solution of boundary value problems using software packages: DD04AD and COLSYS. *Chem. Eng. Commun.*, 22:1–9, 1983.

[21] W. H. Enright and P. H. Muir. Superconvergent interpolants for the collocation solution of boundary value ordinary differential equations. *SIAM J. Sci. Comput.*, 21:227–254, 1999.

[22] G. Fairweather and R. D. Saylor. The reformulation and numerical solution of certain nonclassical initial-boundary value problems. *SIAM J. Sci. Stat. Comput.*, 12:127–144, 1991.

[23] G. Fairweather and M. Vedha-Nayagam. An assessment of numerical software for solving two-point boundary-value problems arising in heat transfer. *Numer. Heat Transf. A*, 11:281–293, 1987.

[24] M. Izadi, J. Singh, and S. Noeiaghdam. Simulating accurate and effective solutions of some nonlinear nonlocal two-point BVPs: Clique and QLM-clique matrix methods. *Heliyon*, 9, 2023.

[25] R. Jalilian. Non-polynomial spline method for solving Bratu's problem. *Comput. Phys. Commun.*, 181:1868–1872, 2010.

[26] S. Khuri and A-M Wazwaz. A variational approach for a class of nonlocal elliptic boundary value problems. *J. Math. Chem.*, 52, 2014.

[27] S. A. Khuri and A. Sayfy. A fixed point iteration method using green's functions for the solution of nonlinear boundary value problems over semi-infinite intervals. *Int. J. Comp. Math.*, 97:1303–1319, 2020.

[28] M. Lakestani and M. Dehghan. Four techniques based on the B-spline expansion and the collocation approach for the numerical solution of the Lane–Emden equation. *Math. Methods Appl. Sci.*, 36:2243–2253, 2013.

[29] P. H. Muir, G. Fairweather, and M. Vedha-Nayagam. The effect of Prandtl number on heat transfer from an isothermal rotating disk with blowing at the wall. *Int. Commun. Heat Mass Transf.*, 10:287–297, 1983.

[30] P. H. Muir and B. Owren. Order barriers and characterizations for continuous mono-implicit Runge-Kutta schemes. *Math. Comp.*, 61:675–699, 1993.

[31] J. Shahni and R. Singh. Numerical solution and error analysis of the Thomas-Fermi type equations with integral boundary conditions by the modified collocation techniques. *J. Comput. Appl. Math.*, 441, 2024.

[32] L. F. Shampine and P. H. Muir. Estimating conditioning of BVPs for ODEs. *Math. Comput. Model.*, 40:1309–1321, 2004.

[33] H. Singh. Solving a class of local and nonlocal elliptic boundary value problems arising in heat transfer. *Heat Transf.*, 51:1524–1542, 2022.

[34] H. Singh. Chebyshev spectral method for solving a class of local and nonlocal elliptic boundary value problems. *Int. J.Numer. Simul.*, 24:899–915, 2023.

[35] R. Singh. An iterative technique for solving a class of local and nonlocal elliptic boundary value problems. *J. Math. Chem.*, 24:1874–1894, 2020.

[36] R. Singh, G. Nelakanti, and J. Kumar. A new efficient technique for solving two-point boundary value problems for integro-differential equations. *J. Math. Chem.*, 52:2030–2051, 2014.

[37] R. Singh and A-M. Wazwaz. An efficient method for solving the generalized Thomas-Fermiand and Lane-Emden-Fowler type equations with nonlocal integral type boundary conditions. *Int. J. Appl. Comput. Math.*, 8:68, 2022.

[38] H. Temimi and A. R. Ansari. A semi-analytical iterative technique for solving nonlinear problems. *Comput. Math. Appl.*, 61:203–210, 2011.

[39] W. Themistoclakis and A. Vecchio. On the numerical solution of a nonlocal boundary value problem. *J. Comput. Appl. Math.*, 292:720–731, 2016.

[40] R. Wright, J. Cash, and G. Moore. Mesh selection for stiff two-point boundary value problems. *Numer. Alg.*, 7:205–224, 1994.