Trigonometric Interpolation-Based Optimization for a 2nd Order Non-Linear ODE with Mixed Boundary Conditions

Xiaorong Zou (xiaorong.zou@bofa.com)
Global Market Risk Analytic, Bank of America

May 6, 2025

Abstract. In this paper, we propose a trigonometric interpolation-based algorithm for the numerical solution of a 2nd order non-linear ODE with mixed boundary conditions. It estimates the second derivative of the target solution by a trigonometric polynomial. The estimation is generated through an optimization process whose loss function is derived by the differential equation and the boundary conditions. The gradient vector of the associated loss function can be formulated by Fast Fourier Transform so that high-degree accuracy is feasible by using sufficient numbers of interpolation grid points. In case that the solution of the ODE is not unique, the algorithm has flexibility to converge to the desired solution that meets a certain constraint such as being positive/increasing/convex. Comprehensive numerical experiments have been conducted with satisfactory performance on convergence and constraint enforcement under various types of boundary conditions.

The algorithm can be extended for nonlinear ODE of a general order k although implementation complexity will increase as k gets larger.

Keywords: Ordinary Differential Equation, Runge-Kutta Scheme, Trigonometric Interpolation, Fast Fourier Transform, Optimization

MSC2000: Primary 65T40, Secondary 65T50

1. Introduction

A new trigonometric interpolation algorithm was recently introduced in [18]. It leverages Fast Fourier Transform (FFT) to achieve optimal computational efficiency and converges in a way aligned with smoothness of the target function. It can be used to approximate nonperiodic functions defined on bounded intervals. Considering the analytic attractiveness of a trigonometric polynomial, especially in handling differential and integral operations, the algorithm is expected to be applied in a wide spectrum. As an example, a trigonometric interpolation-based optimization method has been developed to solve a 1st order non-linear ODE and the test results show that it outperforms standard Runge-Kutta scheme significantly [18]. In this paper, we continue on the applications of the trigonometric interpolation algorithm to solve the following 2nd order non-linear ODE.

$$y''(x) = f(x, y, y'), \quad x \in [s, e] \tag{1}$$

$$d_{11}y(s) + d_{12}y'(s) + d_{13}y(e) + d_{14}y'(e) \; = \; \alpha, \tag{2}$$

$$d_{21}y(s) + d_{22}y'(s) + d_{23}y(e) + d_{24}y'(e) \; = \; \beta, \tag{3}$$

where $f(x, v, u)$ is continuously differentiable on the range $[s, e] \times R^2$, the matrix $D := (d_{ij})_{1 \le i \le 2, 1 \le j \le 4}$ meets certain technical assumption defined by Eq. (24), and α, β are two real numbers.

A wide variety of natural phenomena are modeled by 2nd order ODEs that have been applied to many problems in physics, engineering, mechanics and so on. In general, closed form solutions are not available, especially in non-linear cases. There are quite rich researches on the algorithms for numerical solutions. Relevant background on 2nd order ODE can be found in [2] and more details are referred to [3]-[14].

The proposed algorithm, labeled as TIBO [1] hereafter, approximates the second derivative y'' of the target solution y by a trigonometric interpolation. The two boundary conditions is naturally captured by two parameters in the derived close-form expressions of y and y'. Further more, FFT can be used to optimize relevant numerical computations such that sufficient numbers of grid points can be applied to improve the accuracy of the approximation. In addition, the algorithm can be extended for a k-th order non-linear ODE although the implementation complexity will increase as k gets larger [19].

TIBO is expected to be accurate when the target solution y is sufficient smooth as stated in Theorem 2.2, which is confirmed by the numerical experiments conducted in Section 4. Another advantage of the optimization-based algorithm is its flexibility to integrate certain constraint on the optimization process so that the algorithm converges to a desired solution in case that multiple solutions exists. In Section 4.4, four types of constraints are used to identify a desired solution, which can be a upper/lower bound of the initial $y'(s)$ [2], the target $y(x)$, the first derivative $y'(x)$ and the second derivative $y''(x)$ respectively. Each of the 4 constraints has been used to successfully identify a desired solution. Note that finding positive solutions of an ODE is an interesting research topic [15]-[16].

TIBO is comparable to the Adomian decomposition method in the sense that it uses analytical close form to globally approximate solutions of ODEs and avoid errors due to a discretization or Taylor-formula-based local approximation [6]. With the analytic form of a trigonometrical interpolation and the leverage on FFT, TIBO is expected to outperform the polynomial-based Adomian decomposition method.

The rest of paper is organized as follows. In Section 2, we summarize the relevant results of trigonometric interpolation algorithm developed

[1] As the abbreviation of trigonometric interpolation-based optimization
[2] It aims to solve a boundary problem where $y(s)$ is given.

in [18]. Section 3 is devoted to develop TIBO. The main idea is similar to what is used to address first order ODE in [18] although extra attention is required to cope with a more complicated objective function and initial conditions. Comprehensive numerical experiments are conducted in Section 4 for the performance assessment with four types of boundary conditions defined in Section 4.1. We compare TIBO with a benchmark method that combines the classic shooting method and Runge-Kutta scheme [17] and report the results in three subsections 4.2-4.4. The summary is made on Section 5.

2. Trigonometrical Interpolation on Non-Periodic Functions

In this section, we review the relevant results on the trigonometric interpolation algorithm developed in [18], starting with the following established theorem.

Theorem 2.1. *Let $f(x)$ be an odd periodic function [3] with period $2b$ and $N = 2M = 2^{q+1}$ for some integer $q \geq 1$, and x_j, y_j be defined by*

$$x_j = -b + j\lambda, \quad \lambda = \frac{2b}{N}, \quad 0 \leq j < N, \tag{4}$$

$$y_j = f(x_j). \tag{5}$$

Then there is a unique trigonometric polynomial of $M-1$ degree

$$\begin{aligned} f_M(x) &= \sum_{0\leq j<M} a_j \sin\frac{j\pi x}{b}, \\ a_j &= \frac{2}{N}\sum_{0\leq k<N} (-1)^j y_k \sin\frac{2\pi jk}{N}, \quad 0 \leq j < M \end{aligned}$$

such that it fits to all grid points, i.e.

$$f_M(x_k) = y_k, \quad 0 \leq k < N.$$

One can computer the coefficients by Inverse FFT

$$\{a_j(-1)^j\}_0^{N-1} = 2 \times Imag(ifft(\{y_k\}_{k=0}^{N-1})).$$

It is shown in [18] that $f_M(x)$ and the associated derivatives converge in an optional order.

[3] Similar results for even periodic function is available in [18].

Theorem 2.2. *Let $f(x)$ be an periodic function with period $2b$ and $|f^{(K+1)}(x)|$ exist with an upper bound D_{K+1}, then*

$$|f_M(x) - f(x)| \leq \frac{C_1(D_{K+1})}{N^K}, \tag{6}$$

$$|f_M^{(k)}(x) - f^{(k)}(x)| \leq \frac{C_2(D_{K+1})}{N^{K-k}}, \quad 1 \leq k < K. \tag{7}$$

where $C_1(D_{K+1})$ and $C_2(D_{K+1})$ are two constants depending on D_{K+1}.

Theorem 2.2 provides the theoretic foundation for trigonometric interpolation-based algorithms in solving a kth order non-linear ODE. For such applications, the trigonometric interpolation of a function and the derived approximations of its derivatives up to the order k are required to converge simultaneously in an effective way.

In [18], the algorithm 2.1 has been enhanced so it can be applied to a nonperiodic function f over a bounded interval $[s, e]$ whose $K+1$-th derivative $f^{(K+1)}(x)$ is bounded. To seek for a periodic extension with same smoothness as in f, we assume that f can be extended smoothly such that $f^{(K+1)}$ exists and is bounded over $[s-\delta, e+\delta]$ for certain $\delta > 0$. A smooth periodic extension of f can be achieved by a cut-off smooth function $h(x)$ with the following property.

$$h(x) = \begin{cases} 1 & x \in [s, e], \\ 0 & x < s - \delta \text{ or } x > e + \delta. \end{cases}$$

A cut-off function with closed-form analytic expression is proposed in [18]. Let

$$o = s - \delta, \quad b = e + \delta - o, \tag{8}$$

and define $F(x) := h(x+o)f(x+o)$ for $x \in [0, b]$. One can treat $F(x)$ as an odd periodic function with period $2b$. Apply Theorem 2.1 to generate the trigonometric interpolation of degree $M-1$ with $N = 2M$ evenly-spaced grid points over $[-b, b]$

$$F_M(x) = \sum_{0 \leq j < M} a_j \sin \frac{j\pi x}{b},$$

and let

$$\hat{f}_M(x) = F_M(x - o) = \sum_{0 \leq j < M} a_j \sin \frac{j\pi(x-o)}{b}.$$

$\hat{f}_M(x)|_{[s,e]}$ can be treated as an trigonometric interpolation of f since $\hat{f}_M(x_k) = f(x_k)$ for all grid points $x_k \in [s, e]$. Numerical tests on certain basic functions demonstrate that $\hat{f}$ approaches f with decent accuracy when f is sufficient smooth [18].

3. The development of TIBO

In this section, we aim to develop Algorithm 3.1 to solve the ODE (1-3). We shall follow the same idea that is used to solve first order ODE in [18], i.e. estimating the solution by an optimization problem with the loss function (Eq. (16)) based on the differential equation. The major part is to formulate the associated gradient vector by FFT. The derivation of the algorithm, on the other hand, becomes significantly more challenging due to the increased complexity of the differential equation and especially, the general boundary conditions.

We shall follow the notations used in Section 2. Assume that $f(x,v,u)$ in Eq (1) is continuous differential on $[s-\delta, e+\delta]\times R^2$ for certain $\delta > 0$. By a parallel shifting if needed, we further assume $s=\delta$ without loss of generality. Let h be the cut-off function specified in Section 2 and extend $f(x,v,u)$ to $F(x,v,u)$ as follows

$$F(x,v,u) = f(x,v,u)h(x) \qquad x\in[0,b]$$

Consider a solution $v(x)$ of the following ODE system

$$\begin{aligned} v''(x) &= F(x,v(x),v'(x)), \quad x\in[0,b], &(9)\\ \alpha &= d_{11}v(s)+d_{12}v'(s)+d_{13}v(e)+d_{14}v'(e), &(10)\\ \beta &= d_{21}v(s)+d_{22}v'(s)+d_{23}v(e)+d_{24}v'(e), &(11)\end{aligned}$$

where $\{d_{ij}\}_{1\le i\le 2, 1\le j\le 4}$ meets the condition (24).

It is clear that $v(x)|_{[s,e]}$ solves ODE (1-3). Let $u(x)=v'(x)$ and $z(x)=v''(x)$ to simplify notations. It is clear that $z(x)$ can be smoothly extended as an odd periodic function with period $2b$, which can be approximated by an odd trigonometric polynomial using Theorem 2.1. To that direction, let $\{(x_k,z_k)\}_{k=0}^{N-1}$ be a grid set of $z(x)$ such that

$$\begin{aligned} x_k &= -b+\frac{2b}{N}k, \qquad k=0,1,\cdots,N-1,\\ z_0 &= 0, \quad z_k=-z_{N-k}, \qquad 1\le k<M.\end{aligned}$$

Notice that

$$s = x_{M+m}, \qquad e = x_{M+m+n}.$$

Let

$$z_M(x) = \sum_{0\le j<M} b_j \sin\frac{j\pi x}{b} \tag{12}$$

be the interpolant of $z(x)$ by Theorem 2.1 such that for $0\le j<M$

$$b_j = \frac{2}{N}\sum_{k=0}^{N-1}(-1)^j z_k \sin\frac{2\pi jk}{N} = \frac{4}{N}\sum_{k=M}^{N-1}(-1)^j z_k\sin\frac{2\pi jk}{N}. \tag{13}$$

By Eq. (12), u and v can be derived accordingly

$$\tilde{u}_M(x) \;=\; a_0 - \frac{b}{\pi}\sum_{1\le j<M}\frac{b_j}{j}\cos\frac{j\pi x}{b}, \tag{14}$$

$$\tilde{v}_M(x) \;=\; a_1 + a_0 x - (\frac{b}{\pi})^2\sum_{1\le j<M}\frac{b_j}{j^2}\sin\frac{j\pi x}{b}. \tag{15}$$

We shall adopt the following notations in the rest of paper for convenience.

$$\begin{aligned}
u_k &= \tilde{u}_M(x_k), \quad v_k = \tilde{v}_M(x_k), \quad F_k = F(x_k, v_k, u_k),\\
X &= \{x_k\}_{0\le k<N}, \quad F = \{F_k\}_{0\le k<N},\\
Z &= \{z_k\}_{0\le k<N}, \quad U = \{u_k\}_{0\le k<N}, \quad V = \{v_k\}_{0\le k<N},\\
DF_{k;u} &= \frac{\partial F}{\partial u}(x_k, v_k, u_k), \quad DF_{k;v} = \frac{\partial F}{\partial v}(x_k, v_k, u_k),\\
DF_u &= \{DF_{k;u}\}_{0\le k<N}, \quad DF_v = \{DF_{k;v}\}_{0\le k<N},\\
X^R &= \{x_k\}_{M\le k<N}, \quad Z^R = \{z_k\}_{M\le k<N}, \quad F^R = \{F_k\}_{M\le k<N},\\
U^R &= \{u_k\}_{M\le k<N}, \quad V^R = \{v_k\}_{M\le k<N},\\
DF_u^R &= \{DF_{k;u}\}_{M\le k<N}, \quad DF_v^R = \{DF_{k;v}\}_{M\le k<N}.
\end{aligned}$$

ODE system (9-11) can be solved by minimizing the following objective function [4].

$$\phi(z_M, z_{M+1}, ..., z_{N-1}) = \frac{1}{2M}\sum_{M\le k<N}(z_k - F_k)^2. \tag{16}$$

We need an effective way to calculate the gradient $\frac{\partial\phi}{\partial Z}$ when M is not small. For $M \le t < N$,

$$\begin{aligned}
M\frac{\partial\phi}{\partial z_t} &= (G_k z_t - F_t)G_t - \sum_{M\le k<N}(G_k z_k - F_k)DF_{k,u}\frac{\partial u_k}{\partial z_t}\\
&\quad - \sum_{M\le k<N}(G_k z_k - F_k)DF_{k,v}\frac{\partial v_k}{\partial z_t}.
\end{aligned} \tag{17}$$

It is clear

$$\frac{\partial b_j}{\partial z_t} = \frac{4}{N}(-1)^j\sin\frac{2\pi j t}{N}, \quad 0\le j<M, \quad M\le t<N. \tag{18}$$

Define $\frac{1}{0} := 0$. Let 0_M be the M-dim zero vector and $sum(T)$ be the summation of all elements in a given vector T. $A_1 \circ A_2$ denotes

[4] z_M, F_M are always 0 since they are odd functions. We include them in Eq. (16) to simply certain FFT related expressions.

the Hadamard product that applies the element-wise product to two metrics of same dimension. Adopt the following notations to cope with boundary conditions.

$$
\begin{aligned}
J &= [0, 1, \frac{1}{2}, \dots, \frac{1}{M-1}, 0_M], \qquad A = [(-1)^j]_{0 \le j < N} \\
\Phi^s_m &= [\sin \frac{2\pi jm}{N}]_{j=0}^{M-1}, \quad \Phi^s_{m,N} = [\Phi^s, 0_M], \\
\Phi^s_{(m+n)} &= [\sin \frac{2\pi jm + n}{N}]_{j=0}^{M-1}, \quad \Phi^s_{m+n,N} = [\Phi^s_m, 0_M], \\
\Phi^c_m &= [\cos \frac{2\pi jm}{N}]_{j=0}^{M-1}, \quad \Phi^c_{m,N} = [\Phi^c_m, 0_M], \\
\Phi^c_{(m+n)} &= [\cos \frac{2\pi jm + n}{N}]_{j=0}^{M-1}, \quad \Phi^c_{m+n,N} = [\Phi^c_{m+n}, 0_M], \\
\Psi_u &= \{(z_j - F_j) DF_{j;u}\}_{j=M}^{N-1}, \quad \Psi^u_N = [0_M, \Psi_u], \\
\Psi_v &= \{(z_j - F_j) DF_{j;v}\}_{j=M}^{N-1}, \quad \Psi^v_N = [0_M, \Psi_v], \\
I_u &= sum(\Psi_u), \quad I_v = sum(\Psi_v), \quad II_v = sum(\Psi_v \circ X^R), \\
S_m &= \frac{b^2}{\pi^2} \sum_{0<j<M} \frac{b_j}{j^2} \sin \frac{2\pi jm}{N}, \quad C_m = \frac{b}{\pi} \sum_{0<j<M} \frac{b_j}{j} \cos \frac{2\pi jm}{N}, \\
S_{m+n} &= \frac{b^2}{\pi^2} \sum_{0<j<M} \frac{b_j}{j^2} \sin \frac{2\pi j(m+n)}{N}, \\
C_{m+n} &= \frac{b}{\pi} \sum_{0<j<M} \frac{b_j}{j} \cos \frac{2\pi j(m+n)}{N}.
\end{aligned}
$$

By Eq. (14) and (15), we have

$$
\begin{aligned}
v(s) &= v(x_{M+m}) = a_1 + a_0 s - S_m, \\
v(e) &= v(x_{M+m+n}) = a_1 + a_0 e - S_{m+n}, \\
u(s) &= u(x_{M+m}) = a_0 - C_m, \\
u(e) &= u(x_{M+m+n}) = a_0 - C_{m+n}.
\end{aligned}
$$

Let $\nabla w := (\frac{\partial w}{\partial z_i})_{M \leq i < N}$ be the gradient vector for a variable $w = w(z_M, \ldots z_{N-1})$. Applying Eq (18), we obtain

$$\begin{aligned}
\nabla S_m &= \frac{4b^2}{\pi^2} Imag(ifft(A \circ \Phi^s_{m,N} \circ J^2)), && (19)\\
\nabla S_{m+n} &= \frac{4b^2}{\pi^2} Imag(ifft(A \circ \Phi^s_{m+n,N} \circ J)), && (20)\\
\nabla C_m &= \frac{4b^2}{\pi^2} Imag(ifft(A \circ \Phi^c_{m,N} \circ J^2)), && (21)\\
\nabla C_{m+n} &= \frac{4b^2}{\pi^2} Imag(ifft(A \circ \Phi^c_{m+n,N} \circ J)). && (22)
\end{aligned}$$

One can solve a_0 and a_1 by Eq (10) and (11),

$$\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \frac{1}{m_{11}m_{22} - m_{12}m_{21}} \begin{pmatrix} m_{22} & -m_{12} \\ -m_{21} & m_{11} \end{pmatrix} \begin{pmatrix} \alpha + \mu \\ \beta + \nu \end{pmatrix}, \qquad (23)$$

where

$$\begin{aligned}
m_{11} &= d_{11}s + d_{12} + d_{13}e + d_{14}, \quad m_{12} = d_{11} + d_{13},\\
m_{21} &= d_{21}s + d_{22} + d_{23}e + d_{24}, \quad m_{22} = d_{21} + d_{23}.
\end{aligned}$$

and

$$\begin{aligned}
\mu &= d_{11}S_m + d_{12}C_m + d_{13}S_{m+n} + d_{14}C_{m+n}\\
\nu &= d_{21}S_m + d_{22}C_m + d_{23}S_{m+n} + d_{24}C_{m+n}.
\end{aligned}$$

In the derivation of Eq. (23), we assume

$$m_{11}m_{22} - m_{12}m_{21} \neq 0. \qquad (24)$$

We obtain by Eq (23), (19-22)

$$\begin{aligned}
\nabla a_0 &= \frac{1}{m_{11}m_{22} - m_{12}m_{21}} (m_{22}\nabla\mu - m_{12}\nabla\nu),\\
\nabla a_1 &= \frac{1}{m_{11}m_{22} - m_{12}m_{21}} (-m_{21}\nabla\mu + m_{11}\nabla\nu),
\end{aligned}$$

where $\nabla\mu$ and $\nabla\nu$ can be calculated by (19-22). It is worthwhile to point out that ∇a_0 and ∇a_1 are constant vectors, which makes the optimization process effective.

We need to solve U in term of Z to cope with $\frac{\partial U}{\partial Z}$ in Eq. (17). By Eq (13) and $z_0 = z_M = 0$, we obtain for $M \le k < N$

$$\begin{aligned} u_k &= a_0 - \sum_{0\le j<M} (-1)^j \frac{bb_j}{j\pi} \cos\frac{2\pi jk}{N} \\ &= a_0 - \frac{2b}{\pi N} \sum_{0\le j<M} \frac{1}{j} \cos\frac{2\pi jk}{N} \sum_{0\le l<N} z_l \sin\frac{2\pi jl}{N} \quad (25)\\ &= a_0 - \frac{4b}{\pi N} \sum_{M\le l<N} z_l \sum_{0\le j<M} \frac{1}{j} \cos\frac{2\pi jk}{N} \sin\frac{2\pi jl}{N}. \quad (26) \end{aligned}$$

The last step is due to $z_l \sin\frac{2\pi jl}{N} = z_{N-1} \sin\frac{2\pi j(N-l)}{N}$. One can rewrite Eq. (25) using FFT as follows:

$$U = a_0 - \frac{2bN}{\pi} Re\{ifft(J \circ Im\{ifft(Z)\})\}[M : N-1].$$

Similarly, we need express V in term of Z.

$$\begin{aligned} v_k &= a_1 + a_0 x_k - \frac{b^2}{\pi^2} \sum_{0\le j<M} (-1)^j \frac{b_j}{j^2} \sin\frac{2\pi jk}{N} \\ &= a_1 + a_0 x_k - \frac{2b^2}{\pi^2 N} \sum_{0\le j<M} \frac{1}{j^2} \sin\frac{2\pi jk}{N} \sum_{0\le l<N} z_l \sin\frac{2\pi jl}{N} \quad (27)\\ &= a_1 + a_0 x_k - \frac{4b^2}{\pi^2 N} \sum_{M\le l<N} z_l \sum_{0\le j<M} \frac{1}{j^2} \sin\frac{2\pi jk}{N} \sin\frac{2\pi jl}{N}. \quad (28) \end{aligned}$$

By (27),

$$V^R = a_1 + a_0 X^R - \frac{2b^2 N}{\pi^2} Im\{ifft(J^2 \circ Im\{ifft(Z)\})\}[M : N-1], \quad (29)$$

By (26) and (28), we obtain for $M \le t < N$

$$\begin{aligned} \frac{\partial u_k}{\partial z_t} &= \frac{\partial a_0}{\partial z_t} - \frac{4b}{\pi N} \sum_{0\le j<M} \frac{1}{j} \cos\frac{2\pi jk}{N} \sin\frac{2\pi jt}{N}, \quad (30)\\ \frac{\partial v_k}{\partial z_t} &= \frac{\partial a_1}{\partial z_t} + x_k \frac{\partial a_0}{\partial z_t} - \frac{4b^2}{\pi^2 N} \sum_{0\le j<M} \frac{1}{j^2} \sin\frac{2\pi jk}{N} \sin\frac{2\pi jt}{N}. \quad (31) \end{aligned}$$

We are in the position to attack the nontrivial term in Eq. (17). Define

$$\begin{aligned} \phi_t^u &:= \sum_{M\le k<N} (G_k z_k - F_{k,u}) DF_{k,u} \frac{\partial u_k}{\partial z_t}, \quad (32)\\ \phi_t^v &:= \sum_{M\le k<N} (G_k z_k - F_{k,v}) DF_{k,v} \frac{\partial v_k}{\partial z_t}. \quad (33) \end{aligned}$$

By (30) and (31),

$$\begin{aligned}
\phi_t^u &= \sum_{M \le k < N} (G_k z_k - F_{k;u}) DF_{k;u} \frac{\partial a_0}{\partial z_t} \\
&- \frac{4b}{\pi N} \sum_{0 \le j < M} J_j \sin \frac{2\pi j t}{N} \sum_{M \le k < N} (G_k z_k - F_{k;u}) DF_{k;u} \cos \frac{2\pi j k}{N}. \\
\phi_t^v &= \sum_{M \le k < N} (G_k z_k - F_{k;v}) DF_{k;v} (\frac{\partial a_1}{\partial z_t} + x_k \frac{\partial a_0}{\partial z_t}) \\
&- \frac{4b^2}{\pi^2 N} \sum_{0 \le j < M} J_j^2 \sin \frac{2\pi j t}{N} \sum_{M \le k < N} (G_k z_k - F_{k;v}) DF_{k;v} \sin \frac{2\pi j k}{N}.
\end{aligned}$$

The gradient vector (17) can be formulated by FFT as follows:

$$\phi^u = I_u \nabla a_0 - \frac{4bN}{\pi} Im\{ifft(J \circ Re[ifft(\Psi_N^u)])\}[M : N-1] \tag{34}$$

$$\begin{aligned}
\phi^v &= I_v \nabla a_1 + II_v \nabla a_0 \\
&- \frac{4b^2 N}{\pi^2} Im\{ifft(J^2 \circ Im[ifft(\Psi_N^v)])\}[M : N-1]
\end{aligned} \tag{35}$$

which implies

$$\nabla \phi = \frac{1}{M}((Z^R - F^R) - \phi^u - \phi^v). \tag{36}$$

In summary, we obtain the following algorithm for a numerical solution of ODE (1-3) [5].

Algorithm 3.1. *The TIBO for a 2nd order non-linear ODE .*

1. *Construct the cut-off function h with the parameter s, e, δ;*

2. *Solve the optimization problem (16) with the gradient vector defined by Eq. (36), where ϕ^u and ϕ^v are defined by Eq. (34) and (35) respectively;*

3. *Use the optimization output vector Z to find coefficients B by Eq (13);*

4. *Apply B to find a_0 and a_1 by Eq. (23) and thus obtain the trigonometric approximation $\tilde{v}_M$ (15).*

[5] We assume that $s = \delta$ in Algorithm 3.1. In general, one should first parallel shift x domain to left by $e - s$.

4. The numerical performance assessments

4.1. The description of performance tests

Let $\hat{y}(x;\theta) = x^n \cos\theta x$ with $x \in [1,3]$. For a fixed $D = (d_{ij})_{2\times 4}$ and $P = (p_{uu}, p_{uv}, p_{vv}, p_u, p_v)$, one can easily see that $\hat{y}$ solves the following

$$\begin{aligned} y'' &= \hat{y}'' - p_{uu}\hat{y}'^2 - p_{uv}\hat{y}\hat{y}' - p_{vv}\hat{y}^2 - p_u\hat{y}' - p_v\hat{y} \\ &+ p_{uu}y'^2 + p_{uv}yy' + p_{vv}y^2 + p_uy' + p_vy \qquad (37) \\ \alpha &= d_{11}y(s) + d_{12}y'(s) + d_{13}y(e) + d_{14}y'(e) \qquad (38) \\ \beta &= d_{21}y(s) + d_{22}y'(s) + d_{23}y(e) + d_{24}y'(e) \qquad (39) \end{aligned}$$

where α and β are defined by

$$\begin{aligned} \alpha &= d_{11}\hat{y}(s) + d_{12}\hat{y}'(s) + d_{13}\hat{y}(e) + d_{14}\hat{y}'(e) \\ \beta &= d_{21}\hat{y}(s) + d_{22}\hat{y}'(s) + d_{23}\hat{y}(e) + d_{24}\hat{y}'(e). \end{aligned}$$

$\hat{y}(x;\theta)$ will be refereed as the base solution of ODE (37)-(39), denoted by y_b in the rest of the section.

For all numerical experiments conducted in this section, we choose the associated parameters as follows

$$P = (0.1, 0.1, 1.0, 0.1, 1.0), \quad n = 1, \quad \theta \in \{\frac{\pi}{3}, \frac{3\pi}{2}\}, \qquad (40)$$

and consider the following types of boundary conditions:

Table I. The types of boundary condition used in this section

bdy type	d_{11}	d_{12}	d_{13}	d_{14}	d_{21}	d_{22}	d_{23}	d_{24}	condition on
Neumann	1	0	0	0	0	1	0	0	v_s, u_s
Dirichlet	1	0	0	0	0	0	1	0	v_s, v_e
Mix_1	1	0	0	0	0	0	0	1	v_s, u_e
Mix_2	1	1	0	0	0	0	1	1	$v_s + u_s, v_e + u_e$.

The first two types are associated to the classic Neumann and Dirichlet conditions; Mix_1 applies $y(s)$ and $y'(e)$ as the boundary condition. Mix_2 represents general case and apply a combination y, y' at s, e as the boundary condition.

For a benchmark, we use the classic Runge-Kutta scheme [17] (Section 9.4.1, page 284), labeled $rk4$, to assess the performance of Algorithm 3.1 with Neumann type. On Dirichlet type, we follow the standard shooting method that searches for $y'(s)$ such that $y(e) = \beta$ and then applies $rk4$. Similarly, on mix_1 and mix_2, we extend the shooting method and search $(y(s), y'(s))$ such that Equations (38-39) are satisfied, and then apply $rk4$. The benchmark method will be labeled as $rk4_e$.

In general, TIBO's performance is expected to be sensitive to the initial guesses of the optimization process in Algorithm 3.1, which is provided by the benchmark method in our testing. Especially, the selection of the initial values can determine how the optimization process converges when the solution is not unique. For each type in Table I except $Neumann$, we construct 25 test scenarios associated with the initial values created as follows:

1. Randomly simulate 10 numbers in the range $[-0.5, 0.5]$, place them in two groups as follows.
$$\begin{aligned} init_y &= [0.41, 0.41, -0.40, 0.05, 0.47] \\ init_{y'} &= [0.31, -0.37, 0.13, -0.22, 0.46] \end{aligned}$$

2. Create 25 initial $y(s), y'(s)$ by 5 groups for Mix_1 and Mix_2
$$[f(s) + i \cdot init_y, f'(s) + i \cdot init_{y'}], \quad i = 1, 2, -2, 3, -3.$$
$init_y$ is adjusted to $f(s)$ under $Dirichlet$ and Mix_1.

3. For a pair (i, j) $(1 \leq i, j \leq 5)$, the scenario in j position of group i is assigned with the scenario ID by
$$SID = (i-1) \cdot 5 + j. \tag{41}$$
For example, $(y(s) + 3 \cdot init_y(1), y'(s) + 3 \cdot init_{y'}(1))$ is assigned scenario id $(4-1) \cdot 5 + 1 = 16$.

For each of four boundary types, we conduct two sets of tests with $\theta \in (\frac{\pi}{3}, \frac{3\pi}{2})$ and each set contains 25 scenarios described above. $f(x; \theta)$ becomes more volatile as θ gets larger and, model performance with $\theta = \pi/3$ is expected to outperform that with $\theta = 3\pi/2$. For a given test type and θ, overall performance on 25 scenarios provide us how robust the optimization process is.

For a test scenario, part of the following information will be reported.

1. SID: The scenario ID defined by Eq. (41).

2. Bdy: The associated boundary type defined by Table I.

3. ST: The solution type which can be *base* and *second*. *base* refers to the case where it is identical to y_b over $[s, e]$; and *second* refers to a different solution.

4. $max|y'' - f|$: The max difference is calculated by applying identified approximation y_{tibo}, i.e. $y''_{tibo} - f(x, y_{tibo}, y'_{tibo})$. The max is taken over the set of 2^{10} equally-spaced points over $[0, b]$. Note that grid point set used in the optimization algorithm is determined by $q \leq 7$. As such, a negligible max indicates that y_{tibo} converges to a solution of the ODE over $[0, b]$ although it can be different from y_b.

5. $max(|y_{tibo} - y_b|)$: the max difference between the solution y_{tibo} by Algorithm 3.1 and y_b over the points in $[s, e]$ generated by the set of 2^{10} equally-spaced points over $[0, b]$ as explained above. A negligible value suggests that y_{tibo} converges to y_b.

6. $max(|y_{rk4_e} - y_s|)$: the similar metric as above but replacing $tibo$ by the benchmark method $rk4_e$.

4.2. The convergence Performance with q

Table II summarizes the impact of q on the convergence with $Neumann$ type. Algorithm 3.1 converges to the base solution y_b over $[s, e]$. $rk4_e$ outperforms $TIBO$ in a single case $\theta = \pi/3$ with $q = 5$. $TIBO$ performs significantly better, especially when solution becomes more oscillated with $\theta = 3\pi/2$ y_{opt}.

Table II. The impact of q on the convergence performance with $Neumann$ type

q	$max(\lvert y_{tibo} - y_b\rvert)$	$max(\lvert y_{rk4_e} - y_b\rvert)$	$max\lvert y'' - f\rvert$
		$\theta = \pi/3$	
5	1.4E-04	1.7E-05	4.4E-03
6	8.5E-07	1.0E-06	5.1E-05
7	9.1E-10	6.4E-08	1.1E-07
		$\theta = 3\pi/2$	
5	2.7E-04	2.7E-02	2.3E-01
6	1.4E-05	1.7E-03	1.2E-03
7	1.8E-08	1.1E-04	1.1E-06

4.3. The convergence Performance with boundary conditions

To assess the performance of Algorithm 3.1 under general boundary types in Table I, we conduct two sets of tests with $q = 7, \theta = \pi/3$ and $q = 7, \theta = 3\pi/2$ For each set, 25 scenarios defined in Subsection 4.1 are used for each of covered boundary types except Neumann where $rk4$ can be used directly to generate the initial values of the optimization process used in TIBO. We have following observations.

1. The optimization process of Algorithm 3.1 terminates normally under all scenarios. It converges to a solution of the ODE in all cases some except 6 scenarios under $(Mix_2, \theta = \pi/3)$ where the optimization process likely converges to a local min situation. Table III shows the max values of the metric $max|y'' - f|$. The rest of 19 scenarios under $(Mix_2, \theta = \pi/3)$ converges to a solution of ODE with $\max(max|y'' - f|) = 1.1E - 07$.

Table III. The distribution of $\max(max|y'' - f|)$

θ	$Neumann$	$Dirichlet$	mix_1	mix_2
$\pi/3$	1.1E-07	1.1E-07	1.1E-07	4.1E-01
$3\pi/2$	1.1E-06	1.1E-06	1.2E-06	1.1E-06

2. There are exact two solutions for each of 6 sets of 25-scenarios testing under

$$Bdy \in \{Dirichlet, Mix_1, Mix_2\}, \quad \theta \in \{\pi/3, 3\pi/2\}.$$

   Figure 1 shows the base solution y_b and the second solution v represented by a scenario with SID defined in Eq. (41). It also includes the two base solutions for $Neumann$ type.

3. Algorithm 3.1 converges quickly for $Neumann$ type, mainly due to its effective way to compute the gradient vector. For other boundary types, it uses the shooting method to generate initial feeds of the optimization process and the convergent speed largely depends on how fast the shooting method converges. For $Dirichlet$ and Mix_1 type, the shooting method indeed converges quickly in general. For Mix_2 type, the shooting method searches in two-dim space and hence take significant amount of time to converge, which slows down the convergence of Algorithm 3.1. One can consider to adopt a more effective way to generate the initial feed to cope with a general boundary type such as mix_2.

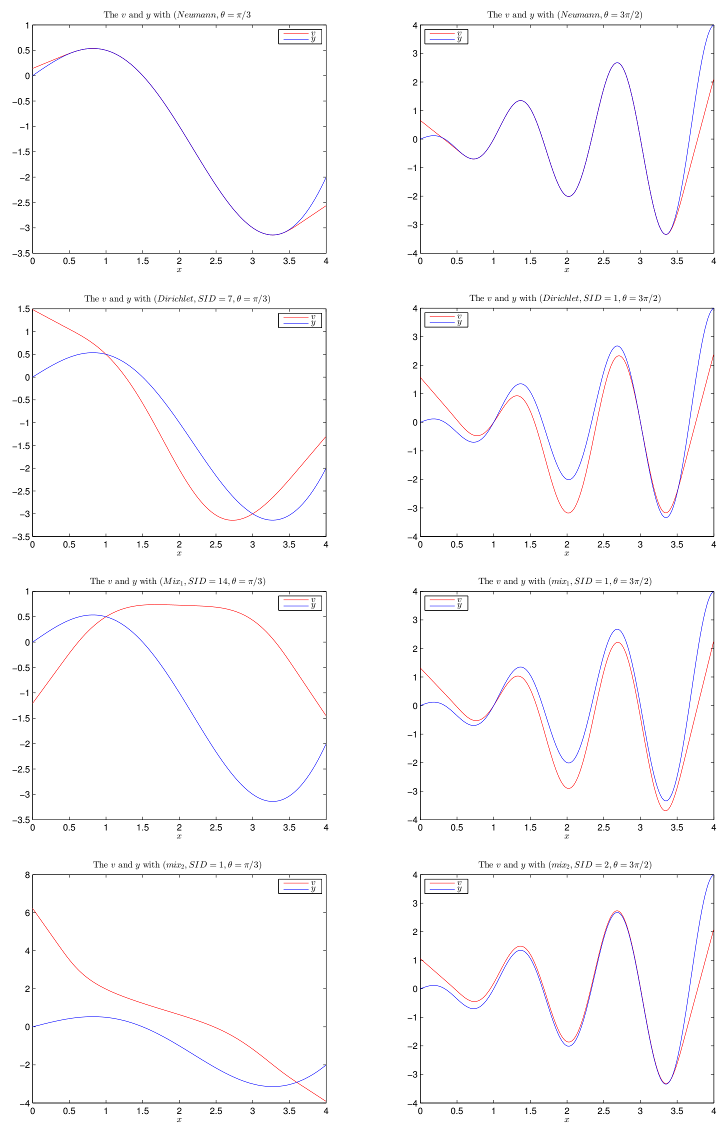

Figure 1. The plots of the base solution y_b (label as y in the plot) and the second solution v under $\theta = \pi/3$ (left) and $\theta = 3\pi/2$ (right). v is based on a representative scenario whose SID is defined in Eq. (41)

Table IV provides the performance metrics under the two representative scenarios that generate the base and second solutions except for *Neumann* type. Column "ST" shows the solution type of a scenario in the table. It also includes the base solution for *Neumann* type with $\theta \in \{\pi/3, 3\pi/2\}$. We have the following observations.

1. The metric $max|y'' - f|$ is negligible for all cases, indicating that Algorithm 3.1 indeed produces a solution for ODE Eqs. (37-39). [6]

2. Algorithm 3.1 converges to the base solution with better accuracy than the benchmark method, especially with $\theta = 3\pi/2$ where y_b becomes significantly more oscillated.

Table IV. The information of the two representative scenarios that generate the base/second solutions as well as the associated performance metrics. Column "ST" provides the solution type of each covered case.

Bdy	ST	SID	$max(\|y_{tibo} - y_b\|)$	$max(\|y_{rk4_e} - y_b\|)$	$max\|y'' - f\|$
			$\theta = \frac{\pi}{3}$		
$Neumann$	base	1	2.7E-09	6.4E-08	1.1E-07
$Dirichlet$	base	1	8.5E-10	1.9E-07	1.0E-07
$Dirichlet$	second	7	1.1E+00	1.1E+00	7.5E-08
mix_1	base	1	1.2E-09	1.8E-07	1.1E-07
Mix_1	second	14	3.4E+00	3.4E+00	7.7E-08
Mix_2	base	2	3.7E-09	3.2E-01	1.1E-07
Mix_2	second	1	2.1E+00	2.1E+00	9.1E-08
			$\theta = \frac{3\pi}{2}$		
$Neumann$	base	1	1.8E-08	1.1E-04	1.1E-06
$Dirichlet$	base	2	2.6E-10	4.4E-05	1.1E-06
$Dirichlet$	second	1	1.2E+00	1.2E+00	1.1E-06
Mix_1	base	4	1.1E-08	7.1E-05	1.1E-06
Mix_1	second	1	8.9E-01	8.9E-01	1.2E-06
Mix_2	base	1	6.5E-08	1.7E+00	1.1E-06
Mix_2	second	2	2.0E-01	9.3E-02	1.1E-06

4.4. The convergence Performance with extra constraints

As shown in Subsection 4.3, the solution of ODE Eqs. (9-11) is not unique in general. Algorithm 3.1 has the flexibility to integrate some extra constrains on the desired solution in addition to the required boundary conditions (2-3). In this subsection, we apply the extra constraints in Table V to address the uniqueness issue to force Algorithm 3.1 return the solution that meets one of those extra constant. Table VI provides the testing results for each of the four constraint types in Table V. The first four rows repeat the results reported in Table IV associated to $Bdy \in \{Dirichlet, Mix_1\}, \theta = \pi/3$ without any extra constrain. We observe the followings.

1. To test the constraint type D, the lower bound -0.5 is applied to the scenario 7 with $Bdy = Dirichlet, \theta = \pi/3$ as reported at the

[6] Note that TIBO enforces the boundary condition automatically.

Table V. The constraint types tested in this subsection

ID	Description	Bdy
D	imposing a upper/lower bound on $y'(s)$	*Dirichlet*
v	imposing a upper/lower bound on $y(x)$ over $[s, e]$	*Any*
u	imposing a upper/lower bound on $y'(x)$ over $[s, e]$	*Any*
z	imposing a upper/lower bound on $y''(x)$ over $[s, e]$	*Any*

fifth row in Table IV. The scenario generates the second solution under default setting without any extra conditions. Note that $y_b'(s)$ meets the condition while $v'(s)$ doesn't from the plot Figure 1 (the second panel on the left). With the new constraint, $max(|y_{tibo} - y_b|)$ and $max|y'' - f|$ becomes negligible, indicating that Algebra 3.1 converges to the base solution as desired. The new solution and base solution is plotted in Figure 2 on top left.

2. To test the constraint type v, the upper bound 0.5 is applied to the scenario 14 with $Bdy = mix_1, \theta = \pi/3$ as reported at the sixth row in Table IV. The scenario generates the second solution without any extra conditions. Note that y_b takes maximum value 0.5 at s while the second solution increases around $y = s$ as shown in Figure 1 [7]. With the new constraint, $max(|y_{tibo} - y_b|)$ and $max|y'' - f|$ becomes negligible, indicating that Algebra 3.1 converges to the base solution as desired. The new solution and base solution is plotted in Figure 2 on top right.

3. To test the constraint type u, the upper bound 0.0 is applied to the same scenario 14 in above Item 2 as reported at the seventh row in Table IV. Note that y_b decreases over $[s, e]$ while the second solution does not meet the constraint as explained in Item 2. With the new constraint, $max(|y_{tibo} - y_b|)$ and $max|y'' - f|$ becomes negligible, indicating that Algebra 3.1 converges to the base solution as desired. The new solution and base solution is plotted in Figure 2 on bottom left.

4. To test the constraint type z, the upper bound 0.0 is applied to the scenario 1 with $Bdy = mix_1, \theta = \pi/3$ as reported at the eighth row in Table IV. The scenario generates the base solution without any extra constraint. Note that y_b is not convexity-downward over $[s, e]$ and thus does not meet the condition while the second solution

[7] See the third panel on the left in Figure 1.

is indeed convexity-downward and meets the condition. With the new constraint, $max(|y_{tibo} - y_b|)$ is close to the value for the second solution shown in the fourth row under default setting without any constraints, shows that the scenario 1 converges to the second solution, which is further confirmed by the plot of the new solution in Figure 2 bottom right.

In summary, the testings shows that Algorithm 3.1 can cope with the uniqueness issue by imposing an constraint defined in Table V for a desired solution.

Table VI. The test results on four types of constraints in Table V. The detail refers to the discussion above this table.

Bdy	TT	SID	ST	$max(\|y_{tibo} - y_b\|)$	$max\|y'' - f\|$	EC	Thres.
$Dirichlet$	1	1	base	8.5E-10	1.0E-07	NA	NA
$Dirichlet$	1	7	second	1.1E+00	7.5E-08	NA	NA
MIX_1	1	1	base	1.2E-09	1.1E-07	NA	NA
MIX_1	1	14	second	3.430860	7.7E-08	NA	NA
$Dirichlet$	2	7	second	3.3E-12	1.1E-07	D_{low}	-0.5
Mix_1	3	14	second	6.7E-10	1.1E-07	v_{up}	0.5
Mix_1	4	14	second	6.7E-10	1.1E-07	u_{up}	0
Mix_1	5	1	base	3.430853	4.2E-03	z_{up}	0

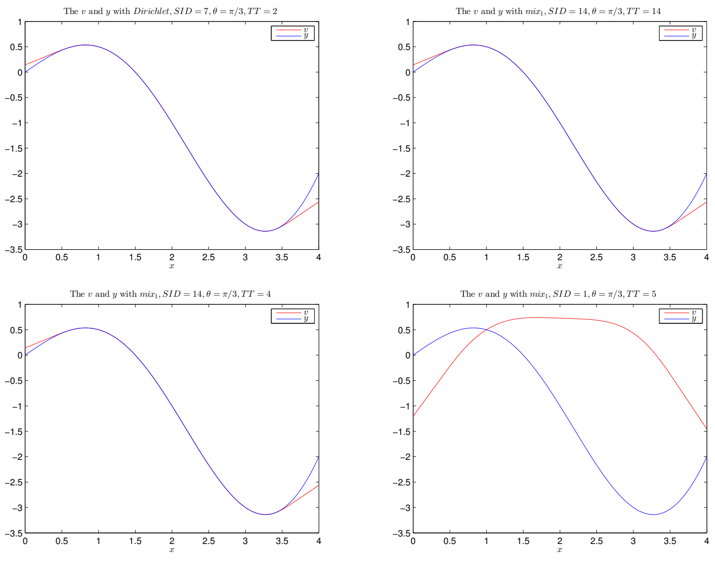

Figure 2. The plots of the base y and the second solution v by Algorithm 3.1 with extra constrains defined in xxx

5. Summary

In this paper, we propose a trigonometric interpolation-based optimization (TIBO) algorithm to approximate the solutions of a second order nonlinear ODE. The algorithm has a few advantages. It is able to cope with a general mixed boundary condition and the idea can be extended to high order nonlinear ODEs [19]. TIBO is expected to converge quickly with decent accuracy if the underlying solution is sufficient smooth. In case that there are multiple solutions, TIBO provides flexibility to identify a desired solution by applying various constraints in the optimization process as shown in Subsection 4.4.

Model performance has been tested under four types of boundary conditions, including classic Neumann, Dirichlet and other two mixed types as defined in Subsection 4.1. For Neumann type, TIBO converges with expected accuracy and outperforms rk4 significantly. Similar performance is observed with Dirichlet type, and TIBO outperforms a shooting-based benchmark. For the other two mixed types, TIBO converges in a robust way and outperforms significantly a benchmark based on a combination of the shooting method and rk4. With those mixed types, the convergence speed becomes more sensitive to the initial guess feeding to the optimization process and an enhancement is desired to improve TIBO's efficiency.

Multiple solutions are observed in the numerical experiments with the three boundary types except Neumann whose solution is supposed to be unique. Four types of constraints are used to identify a desired solution in Subsection 4.4. The constraint can be a upper/lower bound of the initial $y'(s)$ (Dirichlet only), the solution $y(x)$, $y'(x)$ and $y''(x)$ respectively. All four types of constraints have been used to successfully identify the desired solutions in Subsection 4.4.

References

1. N. A. Gasilov, *On the existence and uniqueness of a solution to the boundary value problem for linear ordinary differential equations*, Acta Math. Univ. Comenianae, 93 (2024), pp 205-224
2. Herbert B. Keller, *Numerical Methods for Two-Point Boundary-Value Problems*, Dover Publications, Inc. Mineola, New York, 2018,
3. S. Cuomo, A. Marasco, *A numerical approach to nonlinear two-point boundary value problems for ODEs*, Computers and Mathematics with Applications, 55 (2008), pages = 2476-2489,
4. José L. López, Ester Pérez Sinusía, Nico M. Temme, *Multi-point Taylor approximations in one-dimensional linear boundary value problems*, Applied Mathematics and Computation, https://doi.org/10.1016/j.amc.2008.11.015, 207 (2)(2009), pp = 519-527,
5. C. Chun, R. Sakthivel, *Homotopy perturbation technique for solving two-point boundary value problems comparison with other methods*, Comput. Phys. Commun, 181 (2010), pp 1021–1024,
6. G. Adomian, *A review of the decomposition method in applied mathematics*, J. Math. Analysis and Applications, 135 (1988), pp 501-544,
7. F. Geng, M. Cui, *A novel method for nonlinear two-point boundary value problems: combination of ADM and RKM*, Appl. Math. Comput, 217 (2011), pp 4676–4681,
8. A.S.V. Ravi Kanth, Y.N. Reddy, *Cubic spline for a class of singular two-point boundary value problems*, Appl. Math. Comput., 170 (2005), pp 733–740
9. Y. Zheng, Y. Lin, Y. Shen, *A new multiscale algorithm for solving second order boundary value problems*, Applied Numerical Mathematics, 156 (2020), pp 528–541,
10. *Topological methods for some boundary value problems*, H.B. Thompson, J. Comput. Math. Appl. , 42 (2001), pp 487–495,
11. M. Al-Smadi, O. Abu Arqub, N. Shawagfeh, et al., *Numerical investigations for systems of second-order periodic boundary value problems using reproducing kernel method*, Appl. Math. Comput.,291 (2016),pp 137–148
12. W. Jiang, M. Cui, *Solving nonlinear singular pseudoparabolic equations with nonlocal mixed conditions in the reproducing kernel space*, Int. J. Comput. Math., 87 (2010), pp 3430–3442,
13. Z.H. Zhao, Y.Z. Lin, J. Niu, *Convergence order of the reproducing kernel method for solving boundary value problems*, Math. Model. Anal. , 21 (2016), pp 466–477
14. S. Bushnaq, S. Momani, Y. Zhou, *A reproducing kernel Hilbert space method for solving integro-differential equations of fractional order*, J. Optim. Theory Appl., 2014 (2013), pp 96–105
15. R.P. Agarwal, F.H. Wong, W.C. Lian, *Positive Solutions for Nonlinear Sigular Boundary Value Problems*, Applied Math Letters, 12 (1999), pp 115-120,
16. F.H. Wong, *Existence of positive solutions of sigular boundary value Problems*, Nonlinear Analysis, 21 (1993), pp 397-406,
17. Victor S. Ryaben'kii, Semyon V. Tsynkov, *A Theoretical Introduction to Numerical Analysis*, Chapman and Hall/CRC, 2007
18. Xiarong Zou, *On Trigonometric Approximation and Its Applications*, https://arxiv.org/pdf/2505.02330.
19. Xiarong Zou, *Trigonometric Interpolation Based Optimization for High Order Non-Linear ODE*, To be appear.