



Published in final edited form as:

Int J Numer Methods Eng. 2009 March 19; 77(12): 1690–1730. doi:10.1002/nme.2473.

Matched interface and boundary (MIB) for the implementation of boundary conditions in high-order central finite differences

Shan Zhao^{1,*},† and G. W. Wei^{2,3}

¹ Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487, U.S.A.

² Department of Mathematics, Michigan State University, East Lansing, MI 48824, U.S.A.

³ Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, U.S.A.

SUMMARY

High-order central finite difference schemes encounter great difficulties in implementing complex boundary conditions. This paper introduces the matched interface and boundary (MIB) method as a novel boundary scheme to treat various general boundary conditions in arbitrarily high-order central finite difference schemes. To attain arbitrarily high order, the MIB method accurately extends the solution beyond the boundary by repeatedly enforcing only the original set of boundary conditions. The proposed approach is extensively validated via boundary value problems, initial-boundary value problems, eigenvalue problems, and high-order differential equations. Successful implementations are given to not only Dirichlet, Neumann, and Robin boundary conditions, but also more general ones, such as multiple boundary conditions in high-order differential equations and time-dependent boundary conditions in evolution equations. Detailed stability analysis of the MIB method is carried out. The MIB method is shown to be able to deliver high-order accuracy, while maintaining the same or similar stability conditions of the standard high-order central difference approximations. The application of the proposed MIB method to the boundary treatment of other non-standard high-order methods is also considered.

Keywords

high-order methods; central finite differences; complex boundary conditions; matched interface and boundary

1. INTRODUCTION

Finite difference (FD) method is the oldest while still a widely used approach for the numerical solution of partial differential equations [1–4]. To achieve high-order accuracy as well as high cost-efficiency for practical applications, numerous high-order FD methods have been developed in the literature [5–12], including, standard, Euler sum, non-standard, compact, spectrally weighted, and optimized FD schemes, to name only a few. Typically, these high-order FD methods use wide stencils. Thus, to maintain a designed high-order accuracy, special numerical treatments are required near boundaries where these FD kernels may refer to nodes outside the computational domain. However, it is numerically challenging to construct a boundary closure method that is not only highly accurate to maintain the designed level of

*Correspondence to: Shan Zhao, Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487, U.S.A.

†szhao@bama.ua.edu

accuracy, but also sufficiently robust to handle various boundary conditions arisen in practical problems, and free of non-physical spurious solutions. Indeed, the development of such boundary closure methods has attracted much of research attention in scientific and engineering computations.

The boundary closure of high-order FD schemes with wide stencils can be carried out in essentially two ways: one is to employ the information on a small fictitious domain outside the boundary, while the other relies only on the information inside the boundary. Many different types of boundary closure methods have been proposed in the literature in the framework of the latter one. For example, one type of method builds boundary conditions into differentiation kernels [13], so that both the differential equation and its boundary conditions can be satisfied simultaneously. However, this technique may not be robust enough to handle general boundary conditions. In another type, boundary conditions are imposed in the differential equation discretization by using penalty-like terms [14,15]. Apart from the construction of a delicate procedure to select a penalty factor, the main problem of the penalty method is the possible loss of high accuracy, which is at odds with the spirit of using high-order FD methods. If certain analytical features, such as boundary layers and singularities, are known *a priori* near the boundary, such local features could be included in numerical discretization to promote a more accurate simulation. To this end, the flexible local approximation method (FLAME) [16–20] can be employed, which provides a general framework for integrating analytical features into local FD approximations in a very simple manner. For time-dependent problems, summation-by-parts operators have been constructed for FD approximations of first and second derivatives [21,22]. Effective boundary closure schemes based on the simultaneous approximation term principle have been presented to maintain both high-order accuracy and stability [21–23]. The most commonly used boundary closure method for high-order FD approaches in this category is to employ progressively more asymmetric versions of differential kernels near the boundary [24,25]. In other words, one-sided FD (OFD) approximations are employed near boundaries, which do not involve nodes outside the computational domain. In practice, Chebyshev-type node clustering toward the ends of the domain is usually utilized to permit high accuracy. This kind of non-uniform grid is also widely used in the spectral collocation method. However, using the Chebyshev-type node clustering, the grid spacing h at the boundaries is much smaller than the interior ones. Consequently, such node clustering generally induces high conditional numbers in solving elliptic problems and severe stability constraints in solving time-dependent problems.

At present, it is of considerable interest to study the other type of boundary closure methods, i.e. the fictitious domain boundary method. Moreover, to avoid the difficulty associated with the node clustering, only uniform grid will be considered in this work. The basic assumption of the fictitious domain boundary closure methods is that for a given level of accuracy, fictitious values outside the computational domain is obtained by the smooth extension (or extrapolation) of the physical solution inside the computational domain. A treatment of boundary conditions using fictitious values was proposed in the discrete singular convolution algorithm [26]. In such an approach, the boundary conditions were discretized once to form a set of linear algebraic equations, from which the fictitious values could be determined [27]. As the number of required fictitious points is usually larger than that of the equations given by the boundary condition, it is assumed that there is a one-to-one correspondence between the inner nodes and the outer fictitious nodes on the boundary [27]. This fictitious domain boundary method handles well many boundary conditions [26–30]. However, it has difficulty to accommodate some complex boundary conditions, such as the Robin condition or the free edge support, because under such occasions, the one-to-one assumption might not be rigorously valid beyond the second-order accuracy.

Fornberg outlined a procedure for using fictitious grid points in FD schemes [31]. A detailed scheme, called local adaptive differential quadrature method, was proposed for treating multiple boundary conditions raised in high-order differential equations [32]. This boundary method admits the same number of fictitious points outside a boundary as the number of non-trivial boundary conditions, so that these fictitious points can be uniquely determined. It is capable of dealing with various boundary conditions, including free edges. However, the number of fictitious points determined from this boundary scheme is not sufficient for maintaining the translation invariance property of the high-order central FD kernel. Therefore, non-symmetric differential kernels have to be employed near boundaries [32]. In general, non-symmetric numerical differential kernels are subject to spurious solutions in boundary value and eigenvalue problems. Such unphysical solutions will further induce more constrained time stability conditions in evolution equations. In contrast, symmetric differential kernels produce far fewer spurious solutions or no spurious solution [33]. As a result, they have a better stability in dealing with evolution equations.

The objective of the present work is to construct arbitrarily high-order symmetric differential kernels for solving partial differential equations with general boundary conditions. This is accomplished by introducing the matched interface and boundary (MIB) method for boundary closure. Two criteria are used in the MIB scheme to determine fictitious values. First, the extrapolation of fictitious values should be numerically realized by enforcing given boundary conditions (i.e. a constrained extrapolation). Second, the number of fictitious values is determined by the order of high-order central FD scheme used in the computational domain. Owing to the fact that the number of fictitious values is usually larger than that of boundary conditions, we will repeatedly use the given set of boundary conditions. Technically, this may lead to linearly dependent rows and columns in the resulting matrix. We avoid this linear dependence by selecting a different set of grid partition when the same set of boundary condition is repeatedly used. The proposed MIB method maintains the collocation feature of central FD method over the entire computational domain without resorting to an optimization procedure as that of the FLAME [16–20]. The MIB method is originated from the hierarchical derivative matching method [34,35], originally proposed for simulating electromagnetic wave scattering and propagation in inhomogeneous media. For solving elliptic interface problems with curved interfaces, up to sixth-order MIB schemes have been constructed [36] as a generalization of the immersed boundary method [37], immerse interface method [38,39], and ghost fluid method [40]. In fact, the MIB can be cast in an interpolation formulation without referring to any fictitious value or node [41]. Therefore, the purpose of using fictitious values is to make the MIB presentation clear. In the present work, we reconstruct the MIB for implementing boundary conditions. We consider boundary value problems with arbitrary combinations of Dirichlet, Neumann, and Robin boundary conditions. We also tackle eigenvalue problems, initial-boundary value problems, and high-order differential equations. Extensive numerical experiments are carried out to validate the proposed MIB method and investigate its performance. The time stability of the MIB method is examined both theoretically and numerically.

The rest of this paper is organized as the follows. In Section 2, the numerical setting is laid out. Several relevant boundary closure methods are reviewed and the formulation of the MIB method for boundary treatments is introduced. Extensive numerical experiments are considered in the following sections to validate the proposed MIB method. Comparisons with the FD methods are given. Specifically, Section 3 is devoted to the solution of boundary value problems. Initial-boundary value problems are studied in Section 4. A detailed stability analysis is carried out in Section 4 as well. Solution of high-order differential equations is considered in Section 5. The implementation of boundary conditions in the discrete singular convolution algorithm is discussed in Section 6. Finally, a conclusion is given in Section 7.

2. THEORY AND ALGORITHM

It is well known that by employing a large stencil, the high-order central FD schemes encounter difficulty in dealing with complex boundary conditions, because a translation invariant central FD differentiation kernel will refer to grid points outside the domain, see Figure 1(a). This difficulty could be bypassed via using one-sided differentiations near boundaries, giving rise to OFD methods. Two typical OFD methods will be investigated in this paper, see Figure 1(b) and (c). There is no limit to consider other types of OFD matrix structures. However, such considerations will not affect the essential conclusion of the present study.

In this section, a general numerical setting considered in this paper is presented. Several existing boundary treatments for high-order FD are reviewed. Finally, the MIB method is developed to facilitate high-order central FD schemes for various differential equations.

2.1. General numerical setting

Let us consider a regular computational domain Ω , where Ω is chosen as the unit interval $[0,1]$, the unit square $[0,1] \times [0,1]$, and the unit cube $[0,1] \times [0,1] \times [0,1]$, respectively, in one (1D), two (2D), and three dimensions (3D). As shown in Figure 2, the boundaries of the domain Ω are denoted as the follows:

1D:	2D:	3D:
$\Gamma_1 := \{x x=0\}$	$\Gamma_1 := \{(x, y) x=0, 0 \leq y \leq 1\}$	$\Gamma_1 := \{(x, y, z) 0 \leq x \leq 1, 0 \leq y \leq 1, z=0\}$
$\Gamma_2 := \{x x=1\}$	$\Gamma_2 := \{(x, y) 0 \leq x \leq 1, y=0\}$	$\Gamma_2 := \{(x, y, z) x=0, 0 \leq y \leq 1, 0 \leq z \leq 1\}$
	$\Gamma_3 := \{(x, y) x=1, 0 \leq y \leq 1\}$	$\Gamma_3 := \{(x, y, z) 0 \leq x \leq 1, y=0, 0 \leq z \leq 1\}$
	$\Gamma_4 := \{(x, y) 0 \leq x \leq 1, y=1\}$	$\Gamma_4 := \{(x, y, z) x=1, 0 \leq y \leq 1, 0 \leq z \leq 1\}$
		$\Gamma_5 := \{(x, y, z) 0 \leq x \leq 1, y=1, 0 \leq z \leq 1\}$
		$\Gamma_6 := \{(x, y, z) 0 \leq x \leq 1, 0 \leq y \leq 1, z=1\}$

In this section, our primary concerns are three standard boundary conditions, i.e.

- Dirichlet boundary condition:

$$u = \varphi_j \quad \text{on } \Gamma_j \tag{1}$$

- Neumann boundary condition:

$$\frac{\partial u}{\partial n} = \varphi_j \quad \text{on } \Gamma_j \tag{2}$$

- Robin boundary condition:

$$iku - \frac{\partial u}{\partial n} = \varphi_j \quad \text{on } \Gamma_j \tag{3}$$

even though many non-conventional boundary conditions are also discussed in this paper. Here $\partial/\partial n$ stands for the outward normal derivative, $i = \sqrt{-1}$, and Γ_j are boundaries of the computational domain Ω . Various different combinations of these three boundary conditions are considered, including both homogeneous (i.e. $\varphi_j = 0$) and inhomogeneous ones (i.e. φ_j is a non-zero constant or a function).

A uniform grid is employed throughout this paper, with $N+1$ grid nodes along each dimension. The standard $(2M)$ th-order central FD approximation is considered, in which the derivative of a function is approximated by a weighted linear sum of the function values at $2M+1$ nodes,

$$u^{(n)}(x) = \sum_{j=-M}^M c_j^{(n)}(x) u(x_j) \tag{4}$$

where $u^{(n)}(x)$ is the n th-order derivative of $u(x)$, and the translation invariant FD kernel $c_j^{(n)}(x)$ is the n th-order derivative of the Lagrange interpolation kernel

$$c_j(x) = \prod_{k=-M, k \neq j}^M \frac{x - x_k}{x_j - x_k} \tag{5}$$

The differentiation

$$c_j^{(n)}(x) = \left(\frac{d}{dx} \right)^n c_j(x)$$

can be carried out analytically. For example, one has

$$c_j^{(1)}(x) = \sum_{k=-M, k \neq j}^M \frac{1}{x_j - x_k} \sum_{i=-M, i \neq k, j}^M \frac{x - x_i}{x_j - x_i} \tag{6}$$

$$c_j^{(2)}(x) = \sum_{k,m=-M, k \neq j, m \neq j, m \neq k}^M \frac{1}{(x - x_k)(x - x_m)} \sum_{i=-M, i \neq k, j, m}^M \frac{x - x_i}{x_j - x_i} \tag{7}$$

Recently, a recurrence relationship has been found for the n th-order FD kernel $c_j^{(n)}(x)$, so that the corresponding FD weighing coefficients can be determined conveniently [42]. More recently, a fast algorithm has been developed for determining weights in high-order FD formulas on arbitrarily spaced grids [43]. All central FD weights employed in this paper are generated via this fast algorithm.

2.2. Boundary closure methods for non-symmetric FD

In the high-order OFD method, in order to avoid the boundary closure difficulty of applying a central FD kernel in a translation invariant manner, progressively more asymmetric FD kernels are employed near boundaries. Thus, the OFD approximation is defined pointwisely

$$u^{(n)}(x_i) = \sum_{j=S_1}^{S_2} c_{i,j}^{(n)}(x_i) u(x_j) \tag{8}$$

where S_1 and S_2 are the summation limits. OFD kernels can be given as

$$c_{i,j}(x) = \prod_{k=S_1, k \neq j}^{S_2} \frac{x - x_k}{x_j - x_k} \tag{9}$$

$$c_{i,j}^{(n)}(x_i) = \left. \frac{d^n c_{i,j}(x)}{dx^n} \right|_{x=x_i} \tag{10}$$

In the present study, these OFD coefficients are generated by the fast algorithm [43]. We note that if the summation (8) is global, i.e. $S_1 = 0$ and $S_2 = N$, this actually gives a generalized differential quadrature approximation [42]. In fact, non-uniform grids are used in the generalized differential quadrature to stabilize the method [42].

Different choice of the summation limits S_1 and S_2 gives rise to different OFD matrix structure. Two typical OFD methods shown in Figure 1(b) and (c) will be considered in this paper. For both OFD methods, symmetric FD kernel with fixed bandwidth $2M + 1$ is used for interior nodes, i.e. $S_1 = i - M$ and $S_2 = i + M$, as long as it will not be beyond the domain. Here, M clearly characterizes the order of accuracy of the FD approximation. Near the boundaries, asymmetric FD kernels are employed. These two methods use different limits S_1 and S_2 for summation (8) at x_i

- OFD1:

$$S_1 = \max(i - M, 0), \quad S_2 = \min(i + M, N) \tag{11}$$

- OFD2:

$$S_1 = \max(\min(i - M, N - 2M), 0), \quad S_2 = \min(\max(2M, i + M), N) \tag{12}$$

where $0 \leq i \leq N$ and $2M \leq N$.

As shown in Figure 1, the matrix structure of OFD1 is the same as that of central FD. There seems no reason to consider an OFD method with even shorter stencil at the boundaries. The OFD2 method essentially aims to maintain the same order of accuracy throughout the domain by using OFD kernels with fixed bandwidth $2M + 1$ near the boundaries, see Figure 1(c). Even longer OFD kernels will not improve the order of convergence. We will thus only focus on these two OFD methods in the present study. In general, the OFD2 method is more accurate than the OFD1 method, while the former is more likely to produce spurious modes than the latter [33].

At boundary nodes, the boundary conditions are discretized according to these OFD approximations. For example at x_0 , boundary conditions (1)–(3) are approximated as

- Dirichlet boundary condition:

$$u(x_0)=\varphi \quad (13)$$

- Neumann boundary condition:

$$-\sum_{j=S_1}^{S_2} c_{0,j}^{(1)}(x_0)u(x_j)=\varphi \quad (14)$$

- Robin boundary condition:

$$iku(x_0)+\sum_{j=S_1}^{S_2} c_{0,j}^{(1)}(x_0)u(x_j)=\varphi \quad (15)$$

The boundary discretization at x_N can be similarly done. Consider the regular second-order finite difference method, which in fact can be regarded as a special case of the OFD1 method with $M = 1$. We have the following boundary discretizations

- Dirichlet boundary condition:

$$u(x_0)=\varphi \quad (16)$$

- Neumann boundary condition:

$$\frac{u(x_0) - u(x_1)}{h} = \varphi \quad (17)$$

- Robin boundary condition:

$$iku(x_0) + \frac{u(x_1) - u(x_0)}{h} = \varphi \quad (18)$$

where $h = 1/N$ is the grid spacing.

There are different boundary closure methods to incorporate boundary algebraic equations into the entire OFD discretization. We will consider the following two schemes:

- *Boundary closure scheme 1:* In scheme 1, algebraic equations attained from discretized boundary conditions at x_0 and x_N is simply coupled with the algebraic equations attained from the discretized differential equation at x_1, \dots, x_{N-1} . This straightforward boundary method is often assumed in text books of numerical analysis for the regular FD method. However, it may yield spurious solution in higher dimensions as shown in Reference [33].
- *Boundary closure scheme 2:* In scheme 2, one first solves two boundary algebraic equations to determine u_0 and u_N . In particular, u_0 and u_N will be represented as linear combinations of u_1, \dots, u_{N-1} . Then when u_0 and u_N are referred in discretizing the differential equation on inner nodes x_1, \dots, x_{N-1} , the representations of u_0 and u_N in terms of u_1, \dots, u_{N-1} will be supplied, so that the final FD matrix will not involve u_0 and u_N . To illustrate the idea, let us consider the regular FD method for the Robin boundary condition at the left boundary. Based on the discretized boundary condition, one can solve from (18) that

$$u(x_0) = \frac{h\varphi - u(x_1)}{ikh - 1} \tag{19}$$

Then derivative involved in the differential equation, say $u^{(2)}(x)$ at x_1 , is approximated as

$$u^{(2)}(x_1) = \frac{u(x_0)}{h^2} - \frac{2u(x_1)}{h^2} + \frac{u(x_2)}{h^2} = \frac{\varphi}{(ikh - 1)h} - \left(\frac{1}{(ikh - 1)h^2} + \frac{2}{h^2} \right) u(x_1) + \frac{u(x_2)}{h^2} \tag{20}$$

while the standard central difference is used to approximate $u^{(2)}(x)$ for x_2, \dots, x_{N-2} . The boundary treatment for the right end can be done similarly. Consequently, the dimension of discrete matrix reduces from $(N+1) \times (N+1)$ to $(N-1) \times (N-1)$. This type of boundary treatment is commonly used in the differential quadrature method [42].

We will focus only on the boundary closure schemes 1 and 2 in this paper, although we note that there are other boundary closure methods for the OFD formulation [32]. For low-order differential equations, the difference between numerical results of the boundary closure schemes 1 and 2 could be very small. Nevertheless, there are higher order differential equations, such as those considered in Section 5, that can be handled by scheme 2, but could not be directly resolved by scheme 1.

2.3. Boundary closure methods for central FD schemes

Consider a $(2M)$ th-order central FD approximation of an n th-order derivative, applied in a translation invariant manner

$$u^{(n)}(x_i) = \sum_{j=-M}^M C_j^{(n)} u(x_{i+1}), \quad i=0, 1, 2, \dots, N \tag{21}$$

on a 1D uniform grid $x_{-M} < \dots < x_0 = a < x_1 < \dots < x_N = b < \dots < x_{N+M}$. Clearly, in order to carry out these approximations, M fictitious points outside each boundary are required, see Figure 1(a). As a consequence, the central challenge in the fictitious domain boundary treatments for the central FD method is how to determine M unknown function values on fictitious points by using the given boundary conditions at each boundary with the number of boundary conditions being far fewer than M .

For some simple boundary conditions, this difficulty can be overcome by assuming that there is a one-to-one correspondence between the inner nodes and the outer fictitious nodes on the boundary [27]. For example, one assumes that for the left boundary

$$u(x_{-j}) - u(x_0) = a_j [u(x_j) - u(x_0)] \quad \text{for } j=1, 2, \dots, M \tag{22}$$

where a_j is the unknown representation coefficient to be determined from boundary conditions. With assumption (22), the central FD approximation (21) at x_0 can be modified as

$$u^{(n)}(x_0) = C_0^{(n)} u(x_0) + \sum_{j=1}^M C_j^{(n)} [(1+(-1)^n a_j)u(x_j) + (-1)^n (1 - a_j)u(x_0)] \tag{23}$$

For a class of boundary conditions,

$$u(x_0) = 0, \quad \sum_{n=1}^N K_n u^{(n)}(x_0) = 0 \tag{24}$$

where K_n are given coefficients, the corresponding discretized boundary equation in the central FD is

$$\sum_{j=1}^M \left(\sum_{n=1}^N K_n C_j^{(n)} (1+(-1)^n a_j) \right) u(x_j) = 0 \tag{25}$$

One way to satisfy Equation (25) is to set

$$\sum_{n=1}^N K_n C_j^{(n)} (1+(-1)^n a_j) = 0 \quad \text{for } j=1, 2, \dots, M$$

from which we have

$$a_j = \frac{\sum_{n=1}^N K_n C_j^{(n)}}{-\sum_{n=1}^N K_n (-1)^n C_j^{(n)}} \quad \text{for } j=1, 2, \dots, M \tag{26}$$

Such a fictitious domain boundary treatment has been frequently used in the discrete singular convolution algorithm [26,27,44] to successfully handle many boundary conditions, such as the periodic condition [26], the asymptotic Dirichlet condition [30], the perfect electric and magnetic wall conditions in electromagnetic [28,29], the simply supported, clamped and transversely supported edges in vibration analysis [27], etc. For example, at a clamped edge, the boundary conditions are given as

$$u(x_0) = 0, \quad u^{(1)}(x_0) = 0 \tag{27}$$

It can be derived from Equation (26) that one should take $a_j = 1$. This is the so called symmetric extension [26]. At a simply supported edge, boundary conditions are given as

$$u(x_0) = 0, \quad u^{(2)}(x_0) = 0 \tag{28}$$

These conditions can be imposed by choosing $a_j = -1$. This is the so called anti-symmetric extension [26].

However, for more complex boundary conditions, such as the Robin condition or the free edge support, the one-to-one assumption (22) might not be rigorously valid or can only be satisfied up to second-order accuracy. Under such an occasion, this method cannot maintain high-order accuracy at boundaries.

2.4. The MIB method

It is of great interest in this subsection to construct a systematic and robust boundary method, the MIB method, to accurately determine M fictitious values. We illustrate the idea by considering the Robin boundary condition (3) in 1D

$$iku + u_x = \varphi \quad \text{on } \Gamma_1 \tag{29}$$

With only one boundary condition available, it appears impossible to determine function values on M fictitious points, as $M \gg 1$. To overcome this difficulty, the MIB method will generate fictitious values iteratively by repeatedly matching the boundary condition across the boundary. Referring to Figure 3, we denote fictitious values on M fictitious points outside the domain as f_i for $i = 1, 2, \dots, M$, while function values of $L + 1$ grid points inside the domain as u_j for $j = 0, 1, 2, \dots, L$. We seek for a high-order approach to represent f_i in terms of u_j by means of discretizing the boundary condition (29).

At the first step, since only one boundary condition is available, one can only determine one fictitious point, i.e. f_1 . In order to achieve high-order accuracy for the boundary implementation, OFD approximations are considered, which involve $L + 1$ grid points on the inner side of the boundary; see Figure 4. Thus, the boundary condition (29) is approximated as

$$iku_0 + C_{2,1}^{(1)} f_1 + \sum_{i=2}^{L+2} C_{2,i}^{(1)} u_{i-2} = \varphi \tag{30}$$

where $C_{2,i}^{(1)}$ are OFD weights to approximate first derivative at u_0 by using $f_1, u_0, u_1, \dots, u_L$. Note that the first subscript of $C_{2,i}^{(1)}$ is 2, because u_0 is the second point in the present stencil. The only unknown f_1 in Equation (30) can be solved in terms of u_i for $i = 0, \dots, L$ and φ . Here, we note the flexibility of choosing the total number of terms used by varying L in the finite difference approximation. While the length of L determines the level of accuracy, it can be either larger or smaller than M . For time-independent problems, we usually choose $7 \leq L \leq 11$ to achieve high accuracy. Nevertheless, for unsteady problems, a very large L may render the MIB method unstable. This will be discussed in detail later.

To gain a sufficient number of function values at fictitious points, we use an iterative procedure as introduced in electromagnetic interface problems. By treating the previous calculated fictitious point as knowns, we seek for determining one more fictitious point as shown in Figure 4. Numerically, this is accomplished by discretizing the same boundary condition again, but with one new fictitious point

$$iku_0 + C_{3,1}^{(1)}f_2 + C_{3,2}^{(1)}f_1 + \sum_{i=3}^{L+3} C_{3,i}^{(1)}u_{i-3} = \varphi \tag{31}$$

where $C_{3,i}^{(1)}$ are OFD weights to approximate first derivative at u_0 by using $f_2, f_1, u_0, u_1, \dots, u_L$. Note that the first subscript of $C_{3,i}^{(1)}$ is 3, because u_0 is the third point in the present stencil. The grid partition considered in (31) still has $L + 1$ inner nodes, but two fictitious points outside the boundary. Thus, this partition is independent of the previous one. Since f_1 has already been determined from Equation (30), f_2 can be solved from (31). Through such an iterative procedure, the requested M fictitious points can be efficiently determined if $M \leq L$.

If $M > L$, more iterative steps are required. Through this procedure, at step L , one can determine f_i for $i = 1, 2, \dots, L$. Now, at step $L + 1$, central FD weights are employed so that boundary condition (29) is discretized as

$$iku_0 + \sum_{i=1}^{L+1} C_{L+2,i}^{(1)}f_{L+2-i} + \sum_{i=L+2}^{2L+3} C_{L+2,i}^{(1)}u_{i-L-2} = \varphi \tag{32}$$

where $C_{L+2,i}^{(1)}$ are central difference weights to approximate first derivative at u_0 by using $f_{L+1}, \dots, f_1, u_0, u_1, \dots, u_{L+1}$. Note that u_0 is the $(L + 2)$ th point in the present stencil. In other words, from step $L + 1$ onward, we add both one more fictitious point and one more grid point at each iterative step in the MIB iteration, as shown in Figure 4. This is because central finite difference approximations have higher accuracy than OFD approximations. In Equation (32), one still has only one unknown, i.e. f_{L+1} , which can be easily solved. One can repeat this procedure as many times as necessary, until the desired M fictitious points are all determined; see Figure 4.

In order to apply the MIB method to a boundary value or eigenvalue problem in which u_j is not readily available, a fundamental representation is essential for an implicit formulation

$$f_i = \mathbf{R}^i \cdot \mathbf{U} \quad \text{for } i = 1, 2, \dots, M \tag{33}$$

where vector $\mathbf{U} = (u_0, \dots, u_L, \varphi)$ and the elements of vector \mathbf{R}^i are the representation coefficients of f_i with respect to \mathbf{U} . With this representation, instead of solving f_i , one needs to determine \mathbf{R}^i . The representation coefficients \mathbf{R}^i are determined from essentially the same procedure presented above for f_i . The only difference is that now one boundary condition is discretized and coupled into $L + 2$ algebraic equations, since a fictitious value f_i is represented via $L + 2$ coefficients, which are the $L + 2$ elements of \mathbf{R}^i .

To better illustrate the MIB approach, we next present a detailed MIB formulation for a fourth-order central FD scheme with $M = 2$ and $L = 3$. Consequently, $\mathbf{U} = (u_0, \dots, u_3, \varphi)$. By denoting \mathbf{I}^i as a unit vector with its i th element being 1 and other $L + 1$ elements being 0, we have

$$u_i = \mathbf{I}^{i+1} \cdot \mathbf{U} \quad \text{for } i = 0, 1, \dots, L, \quad \varphi = \mathbf{I}^{L+2} \cdot \mathbf{U} \tag{34}$$

By using representation (33) and (34), Equation (30) is given as

$$ik\mathbf{I}^1 + C_{2,1}^{(1)}\mathbf{R}^1 + \sum_{i=2}^5 C_{2,i}^{(1)}\mathbf{I}^{i-1} = \mathbf{I}^5 \tag{35}$$

in which the common factor \mathbf{U} has been canceled. Thus, the fictitious value f_1 can be solved as

$$\mathbf{R}^1 = \frac{1}{C_{2,1}^{(1)}} \left(\mathbf{I}^5 - ik\mathbf{I}^1 - \sum_{i=2}^5 C_{2,i}^{(1)}\mathbf{I}^{i-1} \right) \tag{36}$$

Similarly, we have from the second step

$$ik\mathbf{I}^1 + C_{3,1}^{(1)}\mathbf{R}^2 + C_{3,2}^{(1)}\mathbf{R}^1 + \sum_{i=3}^6 C_{3,i}^{(1)}\mathbf{I}^{i-2} = \mathbf{I}^5 \tag{37}$$

$$\mathbf{R}^2 = \frac{1}{C_{3,1}^{(1)}} \left(\mathbf{I}^5 - ik\mathbf{I}^1 - C_{3,2}^{(1)}\mathbf{R}^1 - \sum_{i=3}^6 C_{3,i}^{(1)}\mathbf{I}^{i-2} \right) \tag{38}$$

Thus, representation coefficients read

$$\mathbf{R}^1 = \left(-\frac{10}{3} + 4hki, 6, -2, \frac{1}{3}, -4h\right), \quad \mathbf{R}^2 = \left(-\frac{80}{3} + 20hki, 40, -15, \frac{8}{3}, -20h\right)$$

where h is the grid spacing. With representations for f_1 and f_2 , the fourth-order central FD approximation at x_0 and x_1 should be correspondingly modified. For example, for the second derivative, we have

$$u^{(2)}(x_1) = \frac{1}{h^2} \left(-\frac{1}{12}f_1 + \frac{4}{3}u(x_0) - \frac{5}{2}u(x_1) + \frac{4}{3}u(x_2) - \frac{1}{12}u(x_3) \right) \tag{39}$$

$$u^{(2)}(x_0) = \frac{1}{h^2} \left(-\frac{1}{12}f_2 + \frac{4}{3}f_1 - \frac{5}{2}u(x_0) + \frac{4}{3}u(x_1) - \frac{1}{12}u(x_2) \right) \tag{40}$$

The MIB treatment of other boundary conditions can be similarly carried out. For the Dirichlet boundary condition (1), one way is to derive a new boundary condition based on the governing equation. This will be illustrated later in numerical studies. Another way is to directly impose the boundary condition by using an interpolation scheme that avoids the boundary point. An advantage of representation (33) is that fictitious point coefficients \mathbf{R}^i are independent of the boundary data φ , although f_i depends on φ . More precisely, it is sufficient in the MIB method to determine only one set of fictitious point coefficients \mathbf{R}^i for one boundary condition, even

when φ is a spatial function along the boundary or even time-dependent. Moreover, we note that in the MIB method, boundary conditions are enforced systematically so that it can achieve arbitrarily high orders in principle. Finally, we note boundary conditions are satisfied in fictitious point representations, which will be incorporated into the central FD approximation during the differential equation discretization. In this sense, the present MIB method is equivalent to the boundary closure scheme 2 of the OFD approaches considered in Section 2.2.

3. BOUNDARY VALUE PROBLEMS

In this section and the following ones, we examine the usefulness of the MIB method by testing its accuracy, convergence, and efficiency. For a comparison, regular FD and high-order OFD approaches (see Figure 1) are also considered. A uniform grid is employed in all cases, with $N + 1$ being the mesh size along each direction. The bandwidth of the central FD is $2M + 1$, which is the same as that of OFD for interior node. Standard algebraic iterative solvers are utilized in boundary value problems. Denoting u_h as the numerical solution, we use the following measures to estimate errors in numerical examples:

$$L_{\infty} = \frac{\max |u - u_h|}{\max |u|}, \quad L_2 = \sqrt{\frac{\sum_{i=0}^N |u - u_h|^2}{\sum_{i=0}^N |u|^2}}$$

Since accommodating boundary conditions is one of the major concerns for accurately solving elliptic boundary value problems [45–47], we consider first the application of the MIB to the Poisson equation and the Helmholtz equation. We will study the order of convergence and cost-efficiency of the MIB method. For this purpose, several boundary value problems with analytical solutions are considered in 1D, 2D, and 3D.

3.1. 1D boundary value problem

We first consider a 1D boundary value problems of the Helmholtz equation.

- Example 1 [48–51]:

$$\begin{aligned} u_{xx} + k^2 u &= 1 && \text{in } \Omega \\ u &= 0 && \text{on } \Gamma_1 \\ iku - u_x &= 0 && \text{on } \Gamma_2 \end{aligned} \tag{41}$$

The analytical solution is

$$u = \frac{1}{k^2} ((1 - \cos(kx) - \sin(k)\sin(kx)) + i(\cos(k) - 1)\sin(kx))$$

The interval is chosen as $\Omega = [0, 1]$. In order to demonstrate the high accuracy of the MIB approach, a highly oscillatory solution with $k = 20$ is studied, see Figure 5.

The MIB treatment of the Robin boundary condition is carried out as discussed in Section 2, while that of the Dirichlet boundary condition involves a little extra work. We derive a new boundary condition containing derivatives based on the Dirichlet boundary condition and the governing equation. In particular, at $x = 0$, we have both $u(0) = 0$ and $u_{xx}(0) + k^2 u(0) = 1$, so that obviously

$$u_{,xx} = 1 \quad \text{on } \Gamma_1$$

This is the boundary condition finally being used in the MIB modeling on Γ_1 . By taking $M = 6$ and $L = 12$, MIB results are also depicted in Figure 5. It is clear that our numerical results agree with the analytical solution very well.

We next quantitatively examine the numerical orders of the MIB, the regular FD, the OFD1, and the OFD2 methods in Table I. Based on successive mesh refinement, the numerically displayed order of convergence is calculated and reported. In the present study, the boundary closure scheme 2 of Section 2.2 is employed in the regular FD method and two OFD approaches. The boundary closure scheme 1 has been found to yield almost the same results for this 1D problem.

We note that the regular FD method can be regarded as the OFD1 method with $M = 1$. Essentially, the forward or backward difference is used to discretize boundary conditions. These approximations are of the first order of accuracy. It can be observed from Table I that the numerical order of the entire FD approximation is also about first order for problems involving Robin boundary condition. On the other hand, in the MIB method, by taking $M = L = 1$, the corresponding central FD method has exactly the same bandwidth as the regular FD method. Nevertheless, via the MIB boundary treatment, such a central FD method attains the second order of accuracy, i.e. the theoretical order.

We next examine high-order FD methods for several M values in Table I. For the OFD1 method, by using $M + 1$ nodes at the boundaries, the theoretical order is only M th order. Thus, it can be seen from the table that the numerically tested orders usually are slightly larger than M , while in general the convergence rate of the OFD1 is merely M th order. On the other hand, the OFD2 method makes use of $2M + 1$ nodes to approximate boundary conditions such that its theoretical order is maintained as $(2M)$ th order throughout the domain. This is numerically confirmed. By using the MIB boundary treatment, the central FD stencil is applied in a translation invariant manner, so that its theoretical order is guaranteed to be $(2M)$ th order. This is evident in Table I. Furthermore, it can be observed that the MIB method is about 100 times more accurate than the OFD2 method, although both methods attain $(2M)$ th order of convergence.

3.2. 2D boundary value problems

We then consider two 2D boundary value problems of the Helmholtz equation.

- Example 1 [48,49]:

$$\begin{aligned}
 \Delta u + k^2 u &= 0 \quad \text{in } \Omega \\
 iku + \frac{\partial u}{\partial n} &= i(k - k_1)e^{ik_2 y} \quad \text{on } \Gamma_1 \\
 iku + \frac{\partial u}{\partial n} &= i(k - k_2)e^{ik_1 x} \quad \text{on } \Gamma_2 \\
 iku + \frac{\partial u}{\partial n} &= i(k + k_1)e^{i(k_1 + k_2)y} \quad \text{on } \Gamma_3 \\
 iku + \frac{\partial u}{\partial n} &= i(k + k_2)e^{i(k_1 x + k_2 y)} \quad \text{on } \Gamma_4
 \end{aligned} \tag{42}$$

where $(k_1, k_2) = (k \cos \theta, k \sin \theta)$. The analytical solution is $u(x, y) = e^{i(k_1 x + k_2 y)}$.

- Example 2:

$$\begin{aligned}
 \Delta u + k^2 u &= (4+2ki)(x^2+y^2) + (2k^2+k^3i)x^2y^2 + (k^2+k^3i)xy + k^3i \quad \text{in } \Omega \\
 \frac{\partial u}{\partial x} &= ik_1 e^{ik_2 y} + (1+ki)y \quad \text{on } \Gamma_1 \\
 iku - \frac{\partial u}{\partial n} &= i(k+k_2)e^{ik_1 x} + (1+ki)x - k^2 \quad \text{on } \Gamma_2 \\
 iku - \frac{\partial u}{\partial n} &= i(k-k_1)e^{i(k_1+k_2y)} - (4+k^2)y^2 - (1+k^2)y - k^2 \quad \text{on } \Gamma_3 \\
 u &= e^{i(k_1 x + k_2 y)} + (2+ki)x^2 + (1+ki)x + ki \quad \text{on } \Gamma_4
 \end{aligned} \tag{43}$$

where $(k_1, k_2) = (k \cos \theta, k \sin \theta)$. The analytical solution is $u(x, y) = e^{i(k_1 x + k_2 y)} + (2+ki)x^2y^2 + (1+ki)xy + ki$.

In both examples, k_1 and k_2 is the wavenumber in the x - and y -directions, respectively, θ is the wave direction, and $\Omega = [0, 1] \times [0, 1]$.

Arbitrary combinations of three types of standard boundary conditions are considered in these two 2D examples. Again, new boundary conditions are derived for the Dirichlet boundary conditions in advance. For example, in Example 2, we have $u = e^{i(k_1 x + k_2 y)} + (2+ki)x^2 + (1+ki)x + ki$ on Γ_4 so that

$$\frac{\partial^2 u}{\partial x^2} \Big|_{y=1} = \frac{\partial^2}{\partial x^2} u \Big|_{y=1} = \frac{\partial^2}{\partial x^2} [e^{i(k_1 x + k_2 y)} + (2+ki)x^2 + (1+ki)x + ki] = -k_1^2 e^{i(k_1 x + k_2 y)} + (4+2ki)$$

Therefore, the new boundary condition used in the MIB is given as

$$\begin{aligned}
 \frac{\partial^2 u}{\partial y^2} \Big|_{y=1} &= \Delta u \Big|_{y=1} - u_{xx} \Big|_{y=1} \\
 &= [(4+2ki)(x^2+y^2) + (2k^2+k^3i)x^2y^2 + (k^2+k^3i)xy + k^3i]_{y=1} - k^2 u \Big|_{y=1} - [-k_1^2 e^{i(k_1 x + k_2 y)} + (4+2ki)] \\
 &= -k_2^2 e^{i(k_1 x + k_2 y)} + (4+2ki)x^2
 \end{aligned}$$

It is mentioned previously that one advantage of the MIB treatment is that fictitious coefficients in representation (33) are independent of boundary data φ_i . This advantage becomes more evident in 2D studies. For example, on Γ_1 of Example 1, φ_1 is a function of y , so that, precisely the boundary condition at a different y node is different. However, by using representation (33), one needs only conduct one MIB scheme, i.e. determines representation coefficients once, for entire boundary points on Γ_1 . Thus, the MIB treatment is carried out for a total of four times for a 2D computation. Moreover, usually, the computing time of the MIB treatment is very small compared with the CPU time required by the iterative solver. Therefore, the proposed MIB method is a very efficient approach to deal with arbitrary boundary conditions.

By setting the wave angle $\theta = \pi/8$ and the wave number $k = 20$, Figure 6 shows the mesh plots of the MIB solutions with $N^2 = 40^2$, $M = 6$, and $L = 12$. These results are in fact indistinguishable from the analytical solution. On the other hand, it is known that the approximation error of a numerical scheme usually depends on the wave direction θ [49]. Here, we study this dependence for the MIB method by considering Example 1. By using $k = 20$, $N^2 = 40^2$, $M = 6$, and $L = 12$, the numerical errors of the MIB approach for different θ are depicted in Figure 7. For both L_2 and L_∞ errors, a rotational symmetry with respect to $\theta = \pi/4$ is observed. This boundary value problem actually has the same symmetry property. As noticed in [49], this symmetry property in numerical errors is because the quality of the MIB approximation depends on the wavenumber $\max(k_1, k_2)$, instead of k or θ . Thus, the minimal numerical errors appear at $\theta = \pi/4$, where $\max(k_1, k_2)$ takes the minimum. In view of the same pattern in the present numerical

error and that in the literature [49], one may conclude that the MIB method is very robust to different wave direction θ .

We next examine the order of convergence, see Table II. The boundary closure scheme 2 of Section 2.2 is employed in the FD and two OFD approaches. Similar to 1D cases, the convergence rate of the regular FD is again about first order, while that of the OFD1 and the OFD2 is, respectively, M th and $(2M)$ th order. However, for both OFD approaches, when M is large, the standard iterative algebraic solver, i.e. the preconditioned biconjugate gradient method, fails to converge based on the dense mesh $N^2 = 80^2$. Error for such a case is marked with ∞ in Table II. For the OFD2 method, when M is even larger, the convergence stops at smaller N values 20 and 40. It is interesting to note that the convergence of both OFD approaches begins to fail at the same place, i.e. when there are 9 nodes involved in the complete one-sided approximation at the boundary ($M = 8$ in the OFD1 and $M = 4$ in the OFD2). The similar situation has been encountered in Reference [33], in which both OFD approaches begin to generate spurious modes by using one-sided approximations of the same length. Moreover, it is shown in Reference [33] that the production of the spurious modes in the OFD approaches is due to the use of severe one-sided approximations. The converging failure in the present study is believed to be due to the same cause, i.e. by using a severe one-sided approximation, the OFD discrete matrix becomes almost ill-conditioned so that the algebraic solver fails to converge. The convergence problem is not observed in the 1D, probably because the matrix dimension is small in 1D. Thus, the present results indicate that the problem of ill-condition becomes more serious and challenging for higher dimensional cases. Furthermore, a direct consequence of such a converging failure is that both OFD approaches can at most deliver about eighth order of accuracy in 2D cases.

In contrast, the MIB method still maintains its order of accuracy in the 2D. The numerically tested orders of the MIB method for Examples 1 and 2 are listed, respectively, in Tables II and III. It is clear from both tables that the MIB method attains the theoretical order of accuracy, i.e. $(2M)$ th order for $M = 1, 2, 4, \text{ and } 6$. When $M = 8$, certain numerical precision limit is reached so that it finally achieves about 14th order of accuracy. The MIB method is much more accurate than other high-order FD methods.

It is well known that the main merit of a high-order method in comparing with a low-order one is the cost-efficiency. The ultimate goal of developing high-order methods in the field of scientific computing is to save computational time when a high accuracy is required and the domain is quite regular. We next demonstrate the efficiency of our high-order method versus the regular FD method widely used in engineering and scientific computing. We consider Example 1 in Table IV to test the cost-efficiency. It is known that if the boundary conditions of the 2D Poisson equation are always of Dirichlet or Neumann type, a fast Poisson solver based on the fast sine or cosine transform can be utilized to solve the FD discretization matrix of the 2D boundary value problem in essentially $O(N \log N)$ operations. However, for the present test problems with complicated boundary conditions, such as the Robin boundary condition, such a fast solver is not trivially available. Thus, in the present study, the standard preconditioned biconjugate gradient solver is used in both the FD and MIB methods.

It can be seen from Table IV that by using an extremely coarse mesh $N^2 = 10^2$, the 16th order MIB method delivers an extremely high accuracy, $L_2 = 1.29 (-12)$ and $L_\infty = 1.54 (-12)$, while only 0.11 s CPU time is consumed. In Table IV, both the boundary closure scheme 1 and 2 of Section 2.2 are considered for the FD method. It can be seen that the numerical errors of the FD approaches with both boundary closure schemes are almost identical up to the successive mesh refinement of $N = 40$. However, the FD with the boundary closure scheme 1 of Section 2.2 breaks down when $N^2 = 80^2$, while the boundary closure scheme 2 is free of such issues.

The same problem has been observed in Reference [33]. In particular, the boundary closure scheme 1 yields spectral pollution spurious modes in the 2D, but scheme 2 does not.

We finally note the order of convergence of the FD method with both boundary closure schemes is just first order. Thus, by using an extremely dense mesh $N^2 = 1280^2$, the accuracy of the FD method is just about 10^{-4} . Further mesh refinement would be impractical. Based on the convergence pattern of the regular FD method, it can be easily estimated that to achieve the similar level of accuracy as the MIB method, one has to further refine the mesh 27 times. In other words, an intractable mesh size $N^2 = 171798691840^2$ is to be required for the FD method to give $L_2 = 1.36 (-12)$ and $L_\infty = 1.81 (-12)$, as listed in Table IV. On the other hand, the CPU increment ratio is also listed in Table IV. By roughly assuming that for each mesh refinement the CPU time would increase by 12 times, the corresponding FD computational time after 27 refinements is estimated to be 1.25 (+34) s. Therefore, the 16th-order MIB method could be 1.13 (+35) times faster than the widely used FD method in the present 2D problem.

3.3. 3D boundary value problems

We finally consider one 3D boundary value problem.

- Example 1:

$$\begin{aligned}
 \Delta u + k^2 u &= 2y^2 z^2 + 2x^2 y^2 + 2z^2 x^2 + k^2 x^2 y^2 z^2 + 2k^2 xyz + k^2 & \text{in } \Omega = [0, 1] \times [0, 1] \times [0, 1] \\
 \frac{\partial u}{\partial z} &= ik_3 e^{i(k_1 x + k_2 y)} + 2xy & \text{on } \Gamma_1 \\
 u &= e^{i(k_2 y + k_3 z)} + 1 & \text{on } \Gamma_2 \\
 iku - \frac{\partial u}{\partial n} &= i(k + k_2) e^{i(k_1 x + k_3 z)} + 2xz + ki & \text{on } \Gamma_3 \\
 \frac{\partial u}{\partial x} &= ik_1 e^{i(k_1 + k_2 y + k_3 z)} + 2y^2 z^2 + 2yz & \text{on } \Gamma_4 \\
 u &= e^{i(k_1 x + k_2 + k_3 z)} + x^2 z^2 + 2xz + 1 & \text{on } \Gamma_5 \\
 iku - \frac{\partial u}{\partial n} &= i(k - k_3) e^{i(k_1 x + k_2 + k_3)} + (ki - 2)x^2 y^2 + (2ki - 2)xy + ki & \text{on } \Gamma_6
 \end{aligned} \tag{44}$$

where

$$(k_1, k_2, k_3) = \left(\frac{k}{\sqrt{2}}, \frac{k}{\sqrt{3}}, \frac{k}{\sqrt{6}} \right)$$

The analytical solution is $u(x, y) = e^{i(k_1 x + k_2 y + k_3 z)} + (xyz + 1)^2$.

Similarly, new conditions need to be derived for the Dirichlet boundaries. For example, on Γ_2 , one attains

$$\frac{\partial^2 u}{\partial y^2} \Big|_{x=0} = \frac{\partial^2 u}{\partial y^2} \Big|_{x=1} = -k_2^2 e^{i(k_2 y + k_3 z)}, \quad \frac{\partial^2 u}{\partial z^2} \Big|_{x=0} = \frac{\partial^2 u}{\partial z^2} \Big|_{x=1} = -k_3^2 e^{i(k_2 y + k_3 z)}$$

Therefore, the new boundary condition used in the MIB is given as

$$\frac{\partial^2 u}{\partial x^2} \Big|_{x=0} = \Delta u|_{x=0} - u_{yy}|_{x=0} - u_{zz}|_{x=0} = -k_1^2 e^{i(k_2 y + k_3 z)} + 2y^2 z^2$$

Similar to 2D cases, the regular FD method is still of the first order of accuracy for the present 3D problem, while both high-order OFD approaches break down when M is large. These results

are omitted to save space. Nevertheless, the MIB method still attains the theoretical order of accuracy as can be observed in Table V. Slice plots of the MIB solution at $z = 0.5$ are given in Figure 8.

4. INITIAL-BOUNDARY VALUE PROBLEMS AND STABILITY ANALYSIS

We next consider the application of the MIB treatment to time-dependent boundary conditions involved in the unsteady problems. The time integration stability of the MIB discretization is investigated thoroughly by considering the following two 1D model problems.

- Example 1 [52]:

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} &= 0, \quad 0 \leq x \leq 1, \quad t \geq 0 \\ u(x, 0) &= \sin(2\pi x) \\ u(0, t) &= \sin(2\pi(-t)), \quad u(1, t) = \sin(2\pi(1-t)) \end{aligned} \tag{45}$$

The analytical solution is $u(x, t) = \sin(2\pi(x-t))$.

- Example 2:

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} &, \quad 0 \leq x \leq 1, \quad t \geq 0 \\ u(x, 0) &= C \sin x \\ u(0, t) &= 0, \quad \frac{\partial u}{\partial x} \Big|_{x=1} = C \cos(1) e^{-t} \end{aligned} \tag{46}$$

where $C = e^{10}$. The analytical solution is $u(x, t) = C \sin(x) e^{-t}$.

In both problems, we first derive new boundary conditions at the Dirichlet boundaries. For example, at the left end of Example 1, the MIB boundary procedure is carried out based on $\partial u / \partial x = -\cos(2\pi t)$. After MIB spatial discretization, both model problems can be rewritten into the following semi-discrete form:

$$\frac{d}{dt} U = AU + S(t) \tag{47}$$

where $U^T = [u_0, u_1, \dots, u_N]$ is the solution vector, A is the MIB spatial discretization matrix, and $S(t)$ is a source term. We note that although the present boundary conditions are time-dependent, the MIB representation coefficients of fictitious points are actually only needed to be calculated once to construct A . The constant matrix A can then be used at all time steps. The changing part in boundary conditions can be simply accounted in terms of the source term $S(t)$. Therefore, the proposed MIB method is very efficient in handling time-dependent boundary conditions.

Since the boundary data of two model problems are time-dependent, special boundary treatments are required in the time advancement schemes to maintain the overall formal accuracy [52]. In the present study, an advanced strong stability-preserving (SSP) Runge–Kutta method [53–56] is employed to solve Equation (47). The SSP methods are designed to maintain strong stability in certain norm, such as the total variation norm, as the first-order forward Euler scheme, while achieving higher order accuracy in time [53–55]. The extension of SSP methods to solve an autonomous system, such as Equation (47), has been introduced in [56]. By denoting $U_n = U(t_n)$, the general m th-order m stage SSP Runge–Kutta time discretization of Equation (47) can be given as [56]

$$\begin{aligned}
 U^{(0)} &= U_n \\
 U^{(i)} &= U^{(i-1)} + \Delta t A U^{(i-1)} + \Delta t S^{(i)}, \quad i=1, \dots, m \\
 U_{n+1} &= \sum_{k=0}^m \alpha_{m,k} U^{(k)}
 \end{aligned}
 \tag{48}$$

where Δt is the time increment and the coefficients $\alpha_{m,k}$ are given by [53,56]

$$\begin{aligned}
 \alpha_{1,0} &= 1, & \alpha_{m,k} &= \frac{1}{k} \alpha_{m-1,k-1}, & k &= 1, \dots, m-2 \\
 \alpha_{m,m} &= \frac{1}{m!}, & \alpha_{m,m-1} &= 0, & \alpha_{m,0} &= 1 - \sum_{k=1}^m \alpha_{m,k}
 \end{aligned}$$

To maintain high-order accuracy, the boundary source should be set according to [56]

$$S^{(i)} = \left(I + \Delta t \frac{\partial}{\partial t} \right)^{i-1} S(t_n)
 \tag{49}$$

where I is the identity operator. By choosing $m = 4$, a SSP fourth-order four-stage Runge–Kutta method (SSP-RK4) is used in this work.

The MIB results for these two initial-boundary value problems are shown in Table VI. We choose $M = 4$ and $L = 6$ in the MIB method. A uniform grid with $N = 100$ is employed in both examples. Sufficiently small Δt values are used so that MIB results shown in Table VI are all of extremely high accuracy. These results suggest that the MIB method works very well not only for boundary value problems, but also for initial-boundary value problems.

It is of great interest to explore the stability of the MIB spatial discretization together with the SSP-RK4 temporal discretization. We first examine the stability region of temporal discretization. It is known that although there are many different m th-order m -stage Runge–Kutta methods, their stability domains depend on m only if $m \leq 4$ [31]. Thus the present SSP-RK4 method has the same stability domain as the classical RK4 method. In particular, by denoting the eigenvalue of A being λ , the stability function of the SSP-RK4 can be given as

$$S(\Delta t, \lambda) = 1 + \Delta t \lambda + \frac{(\Delta t \lambda)^2}{2!} + \frac{(\Delta t \lambda)^3}{3!} + \frac{(\Delta t \lambda)^4}{4!}
 \tag{50}$$

The SSP-RK4 time integration will be stable provided that $|S(\Delta t, \lambda)| \leq 1$ for all eigenvalues of A . The stability region of the SSP-RK4 method is shown in Figure 9(a).

We next theoretically analyze the stability of the central FD approximation together with the SSP-RK4 scheme by conducting the Fourier analysis. For this type of analysis, a periodic boundary should be assumed. Consequently, the semi-discrete form is free of the source term $S(t)$

$$\frac{d}{dt} U = AU
 \tag{51}$$

where A is symmetric for second-order derivative, while anti-symmetric for first-order derivative.

Let us consider the hyperbolic equation

$$\frac{\partial u}{\partial t} = - \frac{\partial u}{\partial x}$$

in Example 1 first. We consider the Fourier modes e^{iwx} for wavenumber w in the range $-\pi/h \leq w \leq \pi/h$ with h being the spacing. For the central FD approximation to $-\partial/\partial x$ with $M = 1$ (second-order central FD), eigenvalues of A can be found to be

$$Ae^{iwx} = - \frac{e^{iw(x+h)} - e^{iw(x-h)}}{2h} = -i \frac{\sin wh}{h} e^{iwx} \tag{52}$$

Similarly, eigenvalues of A for $(2M)$ th-order central FD approximation can be found to be [31]

$$Ae^{iwx} = -i \frac{\sin wh}{h} \sum_{k=0}^{M-1} \frac{(k!)^2}{(2k+1)!} 2^{2k} \left(\sin \frac{wh}{2} \right)^{2k} e^{iwx} \tag{53}$$

It is clear from Equations (52) and (53) that eigenvalues of central FD approximation to first derivatives are all purely imaginary numbers. It is known that along the imaginary axis, the SSP-RK4 will be stable within the interval $i[-2\sqrt{2}, 2\sqrt{2}]$; see Figure 9(a). The critical number $2\sqrt{2}$ can also be determined from Equation (50) by taking λ being pure imaginary [34]. Denote ρ as the spectral radius of central FD matrix vA , i.e. $\rho = \max_{0 \leq i \leq M} |\lambda_i|$. We then have that the central FD scheme will be stable if $|\rho \Delta t| \leq 2\sqrt{2}$. For central FD with $M = 1$, one can derive from v Equation (52) that $\rho = 1/h$. Thus, the second-order central FD is stable if $\Delta t \leq 2\sqrt{2}h$. In other words, the corresponding Courant–Friedrichs–Levy (CFL) number is $2\sqrt{2}$. By using a computer algebra package, such as the Maple, one can calculate the maximum value of eigenvalues of high-order central FD matrix given in Equation (53). The corresponding analytical CFL numbers are listed in Table VII.

The stability analysis of the heat equation $\partial u/\partial t = \partial^2 u/\partial x^2$ in Example 2 can be similarly conducted. We first consider the second-order central FD approximation to $\partial^2/\partial x^2$. Eigenvalues of A are found to be

$$Ae^{iwx} = \frac{e^{iw(x+h)} - 2e^{iwx} + e^{iw(x-h)}}{h^2} = \frac{2\cos(wh) - 2}{h^2} e^{iwx} \tag{54}$$

We note that eigenvalues λ are all non-positive real numbers and the spectral radius can be simply calculated to be $\rho = 4/h^2$. In fact, this spectral radius can be calculated based on the stencil itself [34]

$$\rho = \left| \frac{1}{h^2} \right| + \left| -\frac{2}{h^2} \right| + \left| \frac{1}{h^2} \right| = \frac{4}{h^2} \tag{55}$$

Similarly, for the general $(2M)$ th-order central FD method, all eigenvalues are non-positive real numbers and the spectral radius can also be calculated as the absolute sum of corresponding stencil

$$\rho = \sum_{j=-M}^M |C_j^{(2)}| \tag{56}$$

Along the real axis, the SSP-RK4 will be stable within the interval $[-D, 0]$ (see Figure 9(a)) where $D = 2.7852935634052816$. For each M , the central FD method for the heat equation will be stable if $\rho \Delta t \leq D$. Consequently, the analytical CFL numbers can be computed as D/ρ . These results are given in Table VII.

We next numerically verify the analytical CFL numbers given in Table VII. To this end, we consider a central FD discretization with analytical boundary treatments, i.e. the fictitious values needed in the central FD approximation (see Figure 1) will be given directly based on analytical solutions. The semi-discrete form of such a central FD discretization takes the form of Equation (47), instead of Equation (51), but the corresponding source term $S(t)$ will not affect the time stability. Computationally, we note that in the present studies, the boundary data $S(t)$ should be processed as in Equation (49) for fractional time steps in the SSP-RK4 time integration. In both examples, we consider a time integration in the range $t \in [0, T]$ with a time increment Δt . Denote the total number of time steps to be N_t . We have $N_t = T/\Delta t$. We numerically search for the critical N_t values such that the computation is still stable. In particular, we choose $h = 0.001$ and $T = 100$ in Example 1, and $h = 0.01$ and $T = 10$ in Example 2. The critical N_t value is searched based on an increment of 10 time steps and 100 time steps in Examples 1 and 2, respectively. Due to the spatial resolution, a smaller increment of time steps will be insensitive. Based on the numerically detected critical N_t value, one can compute the CFL number to be T/hN_t and $T/h^2 N_t$, respectively, for Examples 1 and 2. It can be seen clearly from Table VII that the numerical CFL numbers of the central FD method are in excellent agreement with the analytical ones.

We finally analyze the stability of the MIB method. Consider again the semi-discrete form Equation (47). We first note that the analytical CFL numbers are very difficult to calculate for the MIB method, because of the complex structure of matrix A . Thus, we investigate the stability of the MIB method by considering two theoretical features of the discrete spectrum of A . First, in comparing with the central FD method with the same M value, the MIB matrix might have a different spectral radius ρ . Second, due to the loss of symmetry, the MIB method could yield spurious modes when L is large, even though the MIB method has been shown to produce fewer spurious modes than the OFD approaches do [33]. It is noted that the issue of spurious solutions of the MIB method has been investigated in details in [33]. However, the connection between the spurious modes and time instability has not been explored yet. In the present study, we show that both spectral radius and spurious modes could affect time stability dramatically. In fact, Examples 1 and 2 are designed to illustrate the influences of these two features.

We first study the stability of the MIB method for solving hyperbolic equation in Example 1. Following the same numerical setting of the central FD method, the critical CFL numbers of

the MIB methods are tested numerically; see Table VII. It is found that for each M value, there exists a critical L value L^* . When $L \leq L^*$, the MIB method is stable and the CFL number is the same as the corresponding central FD method. Nevertheless, the MIB method will be unstable if $L > L^*$. The stable ranges of L for tested M values are reported in Table VII. In particular, we note that when M is large, the critical L value takes a uniform upper bound 7 (or eight grid nodes since the grid index starts from 0). This is consistent with our previous finding on the stability of the hierarchical derivative matching method [34]. Two significant conclusions can be drawn based on the present studies. First, the MIB is a stable method for any M value. Second, when M is large and $L \leq 7$, the MIB method cannot only achieve higher order accuracy, but also maintain the same CFL stability condition as the standard higher order FD method.

The possible instability of the MIB method for the hyperbolic equation is due to the presence of spurious modes. As discussed above, the analytical eigenvalues of the central FD matrix for the first-order derivatives are pure imaginary numbers. We thus define the spurious modes (unphysical modes) in the present study as eigenvalues with non-vanishing real part [33]. To detect spurious modes numerically, we examine the largest real part, $\max\{\text{Re}(\lambda)\}$, of the discrete spectrum of the MIB method for different M and L values. By setting $N = 1000$, these results are presented in Table VIII. It can be observed from Table VIII that for each M , when L is small, a vanishing albeit non-zero real part is presented, due to perhaps numerical round-off. However, when L is too large, the real part becomes very large, indicating the presence of spurious modes. It is clear from Figure 9(a) that such spurious modes will be outside of the SSP-RK4 stability region, unless $\Delta t \rightarrow 0$. Therefore, the MIB method will be unstable for the hyperbolic equation when the spurious modes occur. We finally note that the critical L^* values in Table VIII are the same as those in Table VII, because in fact the latter ones are dictated by the former ones.

We next analyze the stability of the MIB method for solving heat equation in Example 2. Following the same numerical setting of the central FD method, the critical CFL numbers of the MIB method are tested numerically. It is found that unlike in Example 1, the MIB discretization of the heat equation is always stable for all tested M and L values. When $L = 1$, the CFL numbers of the MIB are found to be essentially the same as those of the central FD method; see Table VII and Figure 9(b). The numerical CFL numbers for a larger L are depicted in Figure 9(b). It can be observed that, except for $M = 2$, the CFL number will be declined when L is larger. Moreover, when $M \geq 3$, CFL curves for different M eventually merge into one. This suggests that when both L and M are large, the stability condition depends primarily on L . Again, two significant conclusions can be drawn. First, the MIB is always stable for second-order derivative approximation. Second, by using a large L value, the MIB method can achieve extremely high order of accuracy with a slightly smaller CFL number than that of the central FD method.

We next study the discrete spectrum of the MIB matrix. Since the analytical eigenvalues are negative real numbers, we define the spurious mode in the present context as eigenvalues with imaginary part. To detect spurious modes, we simply check the largest imaginary part of the discrete spectrum, $\max\{\text{Im}(\lambda)\}$. Besides this index, other two indices are also recorded for each discrete spectrum, i.e. the largest real part $\max\{\text{Re}(\lambda)\}$ and the smallest real part $\min\{\text{Re}(\lambda)\}$. The former is crucially related to the instability region of the SSP-RK4 scheme, while the latter primarily determines the spectral radius ρ . These indices for first four M values are given in Table IX. Results for other M values are found to be similar to that of $M = 4$, and are thus omitted to save the space. It is found that for each M , when L is very large, the MIB matrix still produce spurious modes, i.e. non-zero $\max\{\text{Im}(\lambda)\}$ in Table IX. However, the spurious modes will not affect the stability significantly. Physically, the spurious modes with both negative real part and non-vanishing imaginary part can always be scaled into the stability region of the SSP-RK4 (see Figure 9(a)) by using a proper Δt . Numerically, the largest real

part $\max\{\text{Re}(\lambda)\}$ remains to be vanishing when the spurious modes occur (see Table IX). Therefore, the MIB method is always stable, no matter the spurious modes are presented or not. Instead, the stability of the MIB method for the heat equation is affected by the spectral radius ρ only. Here ρ is essentially determined by the smallest real part $\min\{\text{Re}(\lambda)\}$. It can be seen from Table IX that except for $M = 2$, when L is larger, $\min\{\text{Re}(\lambda)\}$ is larger. This explains why the CFL number will decrease as L increases in Figure 9(b). For $M = 2$, $\min\{\text{Re}(\lambda)\}$ becomes smaller after L is large enough to provoke spurious modes. Thus, the corresponding CFL number increases eventually in Figure 9(b). We finally note that the current critical L values that are free of spurious modes are much larger than those in Example 1, especially when M is small. This is essentially consistent with our previous works [33,34].

5. HIGH-ORDER DIFFERENTIAL EQUATIONS

High-order differential equations are often associated with multiple boundary conditions, so that the problem is well posed. These multiple non-standard boundary conditions usually involve high-order derivatives, and have to be properly implemented in order to attain a correct numerical solution [24,32,57]. In this subsection, we validate the MIB method by considering a sixth-order and an eighth-order differential equations. The use of the MIB method for a fourth-order differential equation with a free-edged boundary was considered in [58].

5.1. A sixth-order eigenvalue problem

A circular ring structure that has rectangular cross-sections of constant width and parabolic variable thickness, can be formulated as an eigenvalue problem of a sixth-order differential equation. We denote w as the tangential displacement, Ω as the dimensionless frequency, and r as the variable related to the thickness of the cross-section of the ring. The eigenvalue problem for a half of the ring structure with constraints and a quarter of ring structure without constraints is given, respectively, in Examples 1 and 2.

- Example 1 [32]

$$\beta_1 w^{(6)} + \beta_2 w^{(5)} + \beta_3 w^{(4)} + \beta_4 w^{(3)} + \beta_5 w^{(2)} + \beta_6 w^{(1)} = \Omega^2 (f w^{(2)} + f^{(1)} w^{(1)} - \pi^2 f w) \tag{57}$$

$$w(0) = w^{(1)}(0) = w^{(3)}(0) = 0, \quad w(1) = w^{(1)}(1) = w^{(3)}(1) = 0 \tag{58}$$

for $x \in [0,1]$. Here $\beta_1 = \psi/\pi^4$, $\beta_2 = 3\psi^{(1)}/\pi^4$, $\beta_3 = (2\psi/\pi^2) + (3\psi^{(2)}/\pi^4)$, $\beta_4 = (4\psi^{(1)}/\pi^2) + (\psi^{(3)}/\pi^4)$, $\beta_5 = \psi + 3\psi^{(2)}/\pi^2$, and $\beta_6 = \psi^{(1)}/\psi^{(3)}/\pi^2$, in which $\psi = [f(x)]^3$ and $f = f(x) = -4(r-1)x^2 + 4(r-1)x + 1$.

- Example 2 [32]

$$\beta_1 w^{(6)} + \beta_2 w^{(5)} + \beta_3 w^{(4)} + \beta_4 w^{(3)} + \beta_5 w^{(2)} + \beta_6 w^{(1)} = \Omega^2 (f w^{(2)} + f^{(1)} w^{(1)} - \pi^2 f w/4) \tag{59}$$

$$\begin{aligned} w(0) = w^{(2)}(0) = 0, & \quad \psi^{(1)}(0) [w^{(1)}(0) + 4w^{(3)}(0)/\pi^2] + 4\psi(0)w^{(4)}(0)/\pi^2 = 0 \\ w(1) = w^{(2)}(1) = 0, & \quad \psi^{(1)}(1) [w^{(1)}(1) + 4w^{(3)}(1)/\pi^2] + 4\psi(1)w^{(4)}(1)/\pi^2 = 0 \end{aligned} \tag{60}$$

for $x \in [0,1]$. Here $\beta_1 = 16\psi/\pi^4$, $\beta_2 = 48\psi^{(1)}/\pi^4$, $\beta_3 = (8\psi/\pi^2) + (48\psi^{(2)}/\pi^4)$, $\beta_4 = (16\psi^{(1)}/\pi^2) + (16\psi^{(3)}/\pi^4)$, $\beta_5 = \psi + 12\psi^{(2)}/\pi^2$, $\beta_6 = \psi^{(1)} + 4\psi^{(3)}/\pi^2$, in which $\psi = [f(x)]^3$ and $f = f(x) = -(r-1)x^2 + 2(r-1)x + 1$.

In both examples, Dirichlet zero boundary conditions are directly enforced in the MIB boundary treatment. Unlike previous studies, additional boundary conditions are not derived, because governing equations are complicated. Only other two boundary conditions are used in the present MIB boundary treatment. For example, we consider the MIB treatment at the left end $x = 0$ for Example 1. Two fictitious points are determined based on two boundary conditions $w^{(1)}(0) = w^{(3)}(0) = 0$ in the first step. Then, only the lowest order boundary condition, i.e. $w^{(1)}(0) = 0$, is iteratively enforced to estimate one more fictitious point each step, until a sufficient number of fictitious points is attained. The MIB method for the right end and Example 2 can be similarly done. A standard eigenvalue solver is used to solve the eigenvalue problem resulting from the MIB discretization.

The frequencies of the ring structure calculated by the MIB method are listed in Tables X and XI, respectively for Examples 1 and 2. Since there is no exact solution for this problem, the literature results obtained by the differential quadrature method (DQM) [24], the Rayleigh–Ritz method [24], the generalized differential quadrature rule (GDQR) method [32,59], and the local adaptive differential quadrature method (LaDQM) [32] are adopted as references. We consider several mesh sizes N for each numerical scheme. We note that a fictitious domain boundary treatment is also used in the LaDQM. As a generalized differential quadrature method, the LaDQM makes use of Lagrange kernels, which can be regarded as OFD approximations near the boundary. The accuracy of the LaDQM is determined by the bandwidth M , similar to the MIB and OFD methods. In both tables, results are given for $M = N + 2$ in the LaDQM, while $M = L = N$ in the MIB. Thus, based on the same size N , the LaDQM supposes to be the more accurate than the MIB.

It can be observed from Tables X and XI that the MIB method converges to the same frequency parameter as that of the GDQR for all r values. For most cases, the GDQR slightly outperforms both LaDQM and MIB methods, in terms of convergence. However, in view of the complexity of the GDQR method, which involves the Hermite interpolating polynomials, both fictitious domain boundary methods are simpler. It is of great interest to further compare the accuracies of two fictitious domain methods. By considering the same mesh size $N = 10$ in Example 1, the accuracies of two approaches are very similar and the LaDQM is slightly better. Nevertheless, if the accuracy is compared in terms of the same bandwidth $M = 12$, i.e. $N = 10$ for the LaDQM and $N = 12$ for the MIB, the proposed method is actually more accurate. Moreover, the MIB method significantly outperforms the LaDQM in Example 2, in terms of both fixed N and M comparisons. In fact, for large r values, the LaDQM does not really converge to the reference value of the GDQR, while the MIB method does. The slight overshoot of the LaDQM might be due to the fact that the third boundary condition is very complex in Example 2. On the other hand, as a general framework to handle all type of boundary conditions, such a complex boundary condition imposes no difficulty at all to the MIB method. Thus, the MIB results for Example 2 are as well as for Example 1. This suggests that the MIB method is a very robust boundary closure approach.

5.2. An eighth-order boundary value problem

We next consider an eighth-order boundary value problem [32,57,60]. The problem is defined as

$$\begin{aligned}
 & y^{(8)} + \varphi(x)y = \psi(x), \quad a \leq x \leq b \\
 & y(a) = A_0, \quad y^{(2)}(a) = A_2, \quad y^{(4)}(a) = A_4, \quad y^{(6)}(a) = A_6 \\
 & y(b) = B_0, \quad y^{(2)}(b) = B_2, \quad y^{(4)}(b) = B_4, \quad y^{(6)}(b) = B_6
 \end{aligned}
 \tag{61}$$

where $y = y(x)$ and $\varphi(x)$ and $\psi(x)$ are continuous functions defined in the interval $x \in [a, b]$. Here A_i and B_i , ($i = 0, 2, 4, 6$), are finite real constants. Two examples with different coefficient setting and analytical solutions are studied.

- Example 1 [32]

$$\begin{aligned}\varphi(x) &= -x, & \psi(x) &= -(55+17x+x^2-x^3)e^3, & x &\in [-1, 1] \\ A_0 &= 0, & A_2 &= 2/e, & A_4 &= -4/e, & A_6 &= -18/e \\ B_0 &= 0, & B_2 &= -6e, & B_4 &= -20e, & B_6 &= -42e\end{aligned}$$

The analytical solution is $y(x) = (1-x^2)e^x$.

- Example 2 [32]

$$\begin{aligned}\varphi(x) &= -1, & \psi(x) &= 8(2x\sin(x) - 7\cos(x)), & x &\in [-1, 1] \\ A_0 &= 0, & A_2 &= -4\sin(1)+2\cos(1), & A_4 &= 8\sin(1) - 12\cos(1) \\ & & A_6 &= -12\sin(1)+30\cos(1) \\ B_0 &= 0, & B_2 &= -4\sin(1)+2\cos(1), & B_4 &= 8\sin(1) - 12\cos(1) \\ & & B_6 &= -12\sin(1)+30\cos(1)\end{aligned}$$

The analytical solution is $y(x) = (x^2-1)\cos(x)$.

The MIB method is implemented in the same manner as in the previous studies. At each boundary, three boundary conditions excluding the Dirichlet zero condition are used in the first MIB step to estimate three fictitious points. Then only the lowest order boundary condition is enforced repeatedly. The MIB results of both examples are listed in Tables XII and XIII. In both tables, only maximum absolute errors are reported in order to compare with the literature results of the spline method [57], the GDQR method [32, 59], and the LaDQM [32]. It can be seen from these two tables that the spline method does not converge near the boundaries, while other three methods work well there. In terms of accuracy, the GDQR method is obviously the best one, since it uses a Chebyshev grid. On the other hand, by using a simple uniform grid, the MIB method is almost as accurate as the LaDQM in all cases. In summary, the present studies on high-order differential equations indicate that the MIB scheme is a robust, accurate and reliable boundary approach for high-order FD methods.

6. IMPLEMENTING BOUNDARY CONDITIONS IN THE DISCRETE SINGULAR CONVOLUTION

Although the MIB boundary method is primarily applied to the central FD method in this paper, it in principle can be utilized in any non-standard high-order FD method [5–12] to impose boundary conditions. We illustrate this point in this subsection by considering the discrete singular convolution algorithm [26,44], which has been widely used for solving various challenging computational problems in fluid dynamics simulation [61], structural analysis [27,62], computational electromagnetics [28,29], and so on.

The mathematical foundation of the discrete singular convolution algorithm is the theory of distributions and the theory of wavelets. By appropriately selecting parameters of a kernel, the discrete singular convolution approach typically delivers the spectral accuracy of global spectral/pseudospectral methods for numerical integration [30,49]. In particular, it has been proved that the truncation error of the discrete singular convolution algorithm by using regularized Shannon's kernel (RSK) decays exponentially with respect to the increase in sampling points [63]. On the other hand, as the discrete singular convolution approach is still

a local method in the general sense, like the FD scheme, it thus could be more flexible than global method in terms of handling complex geometry and boundary conditions [30,61,62].

For simplicity, the discrete singular convolution algorithm is briefly described here. The reader is referred to original papers [26,27,44] for more details. In the discrete singular convolution algorithm, a function and its n th-order derivative are usually approximated via a discretized convolution

$$u^{(n)}(x) \approx \sum_{k=-M}^M \delta_{\alpha,\sigma}^{(n)}(x - x_k)u(x_k), \quad n=0, 1, 2, \dots \tag{62}$$

where $2M + 1$ is the computational bandwidth and $\delta_{\alpha,\sigma}(x - x_k)$ is a collective symbol for the (regularized) discrete singular convolution kernels. The high-order derivative terms $\delta_{\alpha,\sigma}^{(n)}(x - x_k)$ in (62) are given by

$$\delta_{\alpha,\sigma}^{(n)}(x - x_k) = \left(\frac{d}{dx}\right)^n \delta_{\alpha,\sigma}(x - x_k) \tag{63}$$

Here, the differentiation can be carried out analytically. Numerical solution of differential equations can be easily implemented by a FD (collocation) scheme using Equation (62).

Although many other discrete singular convolution kernels can be similarly employed, RSK [26] is employed in the present study,

$$\delta_{h,\sigma}(x - x_k) = \frac{\sin \frac{\pi}{h}(x - x_k)}{\frac{\pi}{h}(x - x_k)} e^{-(x-x_k)^2/2\sigma^2} \tag{64}$$

The parameter σ determines the width of the Gaussian envelop and often varies in association with the grid spacing, i.e. $\sigma=rh$, where r is a parameter chosen in computation. For a given problem, optimal parameters M and r can be chosen through a Fourier dispersion analysis [49].

In the following, we consider two numerical experiments to examine the MIB boundary treatment for the discrete singular convolution method. Boundary value problems of the Poisson equation in both 1D and 2D are employed for this purpose.

- Example 1

$$\begin{aligned} u_{xx} &= -k^2 \sin(kx) + 1 && \text{in } \Omega \\ u &= 1 && \text{on } \Gamma_1 \\ u_x &= k \cos(k) + 2 && \text{on } \Gamma_2 \end{aligned} \tag{65}$$

The analytical solution is $u = \sin(kx) + \frac{1}{2}x^2 + x + 1$.

- Example 2

$$\begin{aligned}
 \Delta u &= -2k^2 \sin(kx) \cos(ky) + (2xy + 1)(x^2 + y^2) && \text{in } \Omega \\
 u &= 0 && \text{on } \Gamma_1 \\
 \frac{\partial u}{\partial y} &= x && \text{on } \Gamma_2 \\
 \frac{\partial u}{\partial x} &= k \cos(k) \cos(ky) + y^3 + y^2 + y && \text{on } \Gamma_3 \\
 u &= \cos(k) \sin(kx) + \frac{1}{3}x^3 + \frac{1}{2}x^2 + x && \text{on } \Gamma_4
 \end{aligned} \tag{66}$$

The analytical solution is $u(x, y) = \sin(kx) \cos(ky) + \frac{1}{3}x^3 y^3 + \frac{1}{2}x^2 y^2 + xy$.

The MIB boundary treatment can be similarly conducted as in previous studies. By considering two mesh sizes and two wavenumbers $k = 10$ and 20 , numerical results are listed in Table XIV. Very good results are obtained in all cases. Example plots of numerical results are shown in Figure 10. The excellent agreement between the numerical and analytical solutions can be seen clearly in Example 1. Since a larger k corresponds to a more oscillatory solution, the numerical accuracy is lower for a fixed resolution in Table XIV. However, reasonably good results are still attained for the toughest case where only 21 grid points are utilized along each direction to resolve wavenumber $k = 20$. This is consistent with the literature study [28, 49] that the discrete singular convolution method is a dispersion vanishing method, thus it is suitable for solving short wave problems. Moreover, a comparison among different examples in Table XIV reveals that the present method performs equally well in all cases. This further demonstrates that robustness of the proposed MIB method and its potential for being used in non-standard FD methods [5–12].

7. CONCLUSION

This paper addresses a long-standing difficulty in the treatment of general boundary conditions in the high-order central finite difference (FD) solution of partial differential equations. A novel high-order boundary scheme, the matched interface and boundary (MIB) method is introduced to incorporate boundary conditions into high-order approximations. In the MIB method, boundary conditions are repeatedly utilized to systematically determine a large number of function values at fictitious points outside the domain. Consequently, translation invariant differential kernels can be applied near the boundary. Extensive numerical experiments have been carried out to validate the MIB method and to compare with other high-order boundary closure methods. Various problems in 1D, 2D, and 3D are considered to demonstrate the robustness of the MIB method. The implementation of boundary conditions on irregular domains is under our consideration.

In summary, the proposed MIB method has the following features:

- The MIB boundary treatment can achieve arbitrarily high-order accuracy in principle. Up to 16th-order MIB schemes have been numerically verified in higher dimensions. Moreover, compared with the one-sided finite difference (OFD) method that could have the same order of accuracy, the MIB is usually more accurate.
- In an application of the present scheme [33], the proposed MIB method has been shown to be free of spurious solutions by appropriately choosing L , while OFD approaches do suffer serious problems of spurious solutions when the half bandwidth M is large. Although one-sided FD approximations are employed in the MIB fictitious domain boundary treatment, the MIB method is less likely to produce spurious modes than the OFD approaches. This is primarily because unlike in the OFD approaches, the impact of non-symmetric approximations on the final discrete matrix is indirect in the MIB method and is further balanced by the central FD convolution [33].

However, it is previously unknown how the spurious modes will impact on the time instability of the MIB method, when it is applied to initial-boundary value problems.

- In this work, the MIB method has been shown to be a stable method for solving time-dependent problems. CFL time stability conditions of the central FD/MIB method for solving both hyperbolic and diffusion equations are established. The influence of both spectral radius and spurious mode to time stability of the MIB semi-discretization has been investigated in details. In particular, for first-order derivative approximation, the MIB method follows the same stability constraint as the central FD method when L is small. However, when L is larger than certain critical number, the MIB method is unstable due to the presence of spurious modes with non-vanishing positive real part. In general, the MIB approximation to odd order derivatives should all follow this stability pattern. For second-order derivative approximation, the MIB method is found to be always stable for all tested M and L values. The MIB method actually still permits spurious modes with non-vanishing imaginary parts. However, such modes are still within the stability region. It should be generally true that spurious modes will not affect time instability for even order derivatives. The spectral radius then plays an important role. It is found that for a given M , when L increases, the spectral radius usually increases, so that the CFL number declines. In short, in both studied cases, the MIB method can achieve very high order of accuracy, such as 8th or 12th order, while still maintaining strong stability.
- The MIB method is a very robust approach to implement boundary conditions for solving partial differential equations. This robustness has been verified by applying the MIB method to various boundary value problems, eigenvalue problems, initial-boundary value problems, and high-order differential equations. The MIB method is shown to be able to handle not only the standard Dirichlet, Neumann, and Robin boundary conditions, but also more general ones, such as multiple conditions and time-dependent conditions.
- The MIB method can be incorporated into any high-order approach, such as the compact FD, the Euler sum, optimized FD schemes, and the discrete singular convolution algorithm, to accommodate boundary conditions up to machine accuracy.
- The formulation of the MIB method is quite simple. Furthermore, the MIB coefficient generation can be carried out only once to deal with boundary conditions with spatial and temporal dependent inhomogeneous terms. Thus, the MIB boundary treatment is computationally cheap.
- Although only uniform grids are considered in this paper, the MIB method can be freely applied to non-uniform grids, as well as adaptive grids.
- The proposed MIB method works for non-smooth or discontinuous solutions as well. For problems with non-smooth solutions inside the domain boundaries, the MIB method can achieve up to sixth-order of accuracy for arbitrarily curved interface [36]. Using the MIB, it is also possible to extend non-smooth solutions to regions outside the boundary in a boundary value problem. This amounts to applying the MIB method to both the boundary and interface.

Acknowledgments

Contract/grant sponsor: NSF; contract/grant numbers: DMS-0731503, DMS-0616704

Contract/grant sponsor: NSF; contract/grant numbers: IIS-0430987, DMS-0616704

Contract/grant sponsor: NIH; contract/grant number: CA127189-01

The research of S. Zhao was supported in part by NSF grants DMS-0731503 and DMS-0616704. The research of G.W. Wei was supported in part by NSF grants IIS-0430987 and DMS-0616704, and NIH grant CA127189-01.

References

1. Ferziger, JH.; Perić, M. *Computational Methods for Fluid Dynamics*. Springer; Berlin, Heidelberg: 1996.
2. Forsythe, GE.; Wasow, WR. *Finite-Difference Methods for Partial Differential Equations*. Wiley; New York: 1960.
3. Waring E. Problems concerning interpolations. *Philosophical Transactions of the Royal Society* 1779;69:59–67.
4. Whittaker, ET.; Robinson, G. Lagrange's Formula of Interpolation 17 in *The Calculus of Observations: A Treatise on Numerical Mathematics*. 4. Dover: New York; 1967. p. 28-30.
5. Boyd JP. Sum-accelerated pseudospectral methods—finite-differences and sech-weighted differences. *Computer Methods in Applied Mechanics and Engineering* 1994;116:1–11.
6. Cole JB. High accuracy solution of Maxwell's equations using nonstandard finite differences. *Computational Physics* 1997;11:287–292.
7. Haras Z, Ta'asan S. Finite-difference schemes for long time integration. *Journal of Computational Physics* 1994;114:265–279.
8. Jurgens HM, Zingg DW. Numerical solution of the time-domain Maxwell equations using high accuracy finite-difference methods. *SIAM Journal on Scientific Computing* 2000;22:1675–1696.
9. Lele SK. Compact finite difference scheme with spectral-like resolution. *Journal of Computational Physics* 1992;103:16–42.
10. Lockard DP, Brentner KS, Atkins HL. High accuracy algorithms for computational aeroacoustics. *American Institute of Aeronautics and Astronautics Journal* 1995;33:246–251.
11. Tam CKW, Webb JC. Dispersion-relation-preserving finite difference scheme for computational acoustics. *Journal of Computational Physics* 1993;107:262–281.
12. Young JL, Gaitonde D, Shang JS. Towards the construction of a fourth-order difference scheme for transient EM wave simulation: staggered grid approach. *IEEE Transactions on Antennas and Propagation* 1997;45:1573–1580.
13. Huang WZ, Sloan DM. The pseudospectral method for solving differential eigenvalue problems. *Journal of Computational Physics* 1994;111:399–409.
14. Abarbanel S, Chertock AE. Strict stability of high order compact implicit finite-difference schemes: the role of boundary conditions for hyperbolic PDEs, I. *Journal of Computational Physics* 2000;160:42–66.
15. Carpenter MH, Gottlieb D, Abarbanel S. Time-stable boundary conditions for finite difference schemes solving hyperbolic systems: methodology and applications to high order compact schemes. *Journal of Computational Physics* 1994;111:222–236.
16. Tsukerman I. Flexible local approximation method for electro- and magnetostatic. *IEEE Transactions on Magnetics* 2004;40:941–944.
17. Tsukerman I. Electromagnetic applications of a new finite-difference calculus. *IEEE Transactions on Magnetics* 2005;41:2206–2225.
18. Tsukerman I. A class of difference schemes with flexible local approximation. *Journal of Computational Physics* 2006;211:659–699.
19. Dai J, Tsukerman I, Rubinstein A, Sherman S. New computational models for electrostatics of macromolecules in solvents. *IEEE Transactions on Magnetics* 2007;43:1217–1220.
20. Tsukerman, I. *Nanostructure Science and Technology Series*. Springer; Berlin: 2007. *Computational Methods for Nanoscale Applications: Particle, Plasmons, and Waves*.
21. Strand B. Summation by parts for finite difference approximations for d/dx . *Journal of Computational Physics* 1994;110:47–67.
22. Mattsson K, Nordstrom J. Summation by parts operators for finite difference approximations of second derivatives. *Journal of Computational Physics* 2004;199:503–540.
23. Gustafsson, B. *High Order Difference Methods for Time Dependent PDE*. Springer; Berlin: 2008.

24. Gutierrez RH, Laura PAA. Vibrations of non-uniform rings studied by means of the differential quadrature method. *Journal of Sound and Vibration* 1995;185:507–513.
25. Shu C, Wang CM. Treatment of mixed and nonuniform boundary conditions in GDQ vibration analysis of rectangular plates. *Engineering Structures* 1999;21:125–134.
26. Wei GW. Discrete singular convolution for solution of the Fokker–Planck equations. *Journal of Chemical Physics* 1999;110:8930–8942.
27. Wei GW, Zhao YB, Xiang Y. Discrete singular convolution and its application to the analysis of plates with internal supports. I. Theory and algorithm. *International Journal for Numerical Methods in Engineering* 2002;55:913–946.
28. Bao G, Wei GW, Zhao S. Local spectral time-domain method for electromagnetic wave propagation. *Optics Letters* 2003;28:513–515. [PubMed: 12696600]
29. Shao ZH, Wei GW, Zhao S. DSC time-domain solution of Maxwell's equations. *Journal of Computational Physics* 2003;189:427–453.
30. Zhao S, Wei GW. Comparison of the discrete singular convolution and three other numerical schemes for solving Fisher's equation. *SIAM Journal on Scientific Computing* 2003;25:127–147.
31. Fornberg, B. *A Practical Guide to Pseudospectral Methods*. Cambridge University Press; Cambridge: 1996.
32. Wang Y, Zhao YB, Wei GW. A note on the numerical solution of high order differential equations. *Journal of Computational and Applied Mathematics* 2003;159:387–398.
33. Zhao S. On the spurious solutions in the high order finite difference methods. *Computer Methods in Applied Mechanics and Engineering* 2007;196:5031–5046.
34. Zhao S, Wei GW. High order FDTD methods via derivative matching for Maxwell's equations with material interfaces. *Journal of Computational Physics* 2004;200:60–103.
35. Zhao S, Wei GW. Tensor product derivative matching for wave propagation in inhomogeneous media. *Microwave and Optical Technology Letters* 2004;43:69–77.
36. Zhou YC, Zhao S, Feig M, Wei GW. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *Journal of Computational Physics* 2006;213:1–30.
37. Peskin CS. Numerical analysis of blood flow in heart. *Journal of Computational Physics* 1977;25:220–252.
38. Fogelson AL, Keener JP. Immersed interface methods for Neumann and related problems in two and three dimensions. *SIAM Journal on Scientific Computing* 2000;22:1630–1654.
39. LeVeque RJ, Li ZL. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis* 1994;31:1019–1044.
40. Fedkiw RP, Aslam T, Merriman B, Osher S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics* 1999;152:457–492.
41. Zhou YC, Wei GW. On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method. *Journal of Computational Physics* 2006;219:228–246.
42. Shu, C. *Differential Quadrature and its Application in Engineering*. Springer; London: 2000.
43. Fornberg B. Calculation of weights in finite difference formulas. *SIAM Reviews* 1998;40:685–691.
44. Wei GW. A unified approach for solving the Fokker–Planck equation. *Journal of Physics A: Mathematical and General* 2000;33:4935–4953.
45. Badea L, Daripa P. A fast algorithm for two-dimensional elliptic problems. *Numerical Algorithms* 2002;30:199–239.
46. Braverman E, Israeli M, Averbuch A. A fast spectral solver for a 3D Helmholtz equation. *SIAM Journal on Scientific Computing* 1999;20:2237–2260.
47. Hyman JM, Shashkov M. Approximation of boundary conditions for mimetic finite-difference methods. *Computers and Mathematics with Applications* 1998;36:79–99.
48. Babuška F, Ihlenburg ET, Paik Sauter SA. A generalized finite element method for solving the Helmholtz equation in two dimensions with minimal pollution. *Computer Methods in Applied Mechanics and Engineering* 1995;128:325–359.
49. Bao G, Wei GW, Zhao S. Numerical solution of the Helmholtz equation with high wave numbers. *International Journal for Numerical Methods in Engineering* 2004;59:389–408.

50. Ihlenburg F, Babuška I. Finite element solution of the Helmholtz equation with high wavenumber Part I: the h -version of the FEM. *Computers and Mathematics with Applications* 1995;30:9–37.
51. Ihlenburg F, Babuška I. Finite element solution of the Helmholtz equation with high wavenumber Part II: the h - p -version of the FEM. *SIAM Journal on Numerical Analysis* 1997;34:315–358.
52. Carpenter MH, Gottlieb D, Abarbanel S, Don W-S. The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a study of the boundary error. *SIAM Journal on Scientific Computing* 1995;16:1241–1252.
53. Gottlieb S, Shu CW, Tadmor E. Strong stability-preserving high order time discretization methods. *SIAM Reviews* 2001;43:89–112.
54. Gottlieb S. On high order strong stability preserving Runge–Kutta and multi step time discretizations. *Journal of Scientific Computing* 2005;25:105–128.
55. Gottlieb S, Ruuth SJ. Optimal strong-stability-preserving time-stepping schemes with fast downwind spatial discretizations. *Journal of Scientific Computing* 2006;27:289–304.
56. Chen MH, Cockburn B, Reitich F. High order RKDG methods for computational electromagnetics. *Journal of Scientific Computing* 2005;22:205–226.
57. Siddiqi SS, Twizell EH. Spline solutions of linear eighth-order boundary value problem. *Computer Methods in Applied Mechanics and Engineering* 1996;131:309–325.
58. Zhao S, Wei GW, Xiang Y. DSC analysis of free-edged beams by an iteratively matched boundary method. *Journal of Sound and Vibration* 2005;284:487–493.
59. Wu TY, Liu GR. Application of generalized differential quadrature rule to sixth-order differential equations. *Communications in Numerical Methods in Engineering* 2000;16:777–784.
60. Liu GR, Wu TY. Differential quadrature solutions of eighth-order boundary-value differential equations. *Journal of Computational and Applied Mathematics* 2002;145:223–235.
61. Zhou YC, Patnaik BSV, Wan DC, Wei GW. DSC solution for flow in a staggered double lid driven cavity. *International Journal for Numerical Methods in Engineering* 2003;57:211–234.
62. Xiang Y, Zhao YB, Wei GW. Discrete singular convolution and its application to the analysis of plates with internal supports. II. Complex supports. *International Journal for Numerical Methods in Engineering* 2002;55:947–971.
63. Qian LW. On the regularized Whittaker–Kotel’nikov–Shannon sampling formula. *Proceedings of the American Mathematical Society* 2003;131:1169–1176.

APPENDIX: ACRONYM TABLE

The acronyms used in this paper are listed below, in which those marked with * are introduced in this work.

1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
CFL	Courant–Friedrichs–Levy
DQM	differential quadrature method [24]
FD*	finite difference
FLAME	flexible local approximation method [16–20]
GDQR	generalized differential quadrature rule [59]
LaDQM	local adaptive differential quadrature method [32]
MIB*	matched interface and boundary
OFD*	one-sided finite difference
OFD1*	one-sided finite difference type 1, see Figure 1(b)

OFD2*	one-sided finite difference type 2, see Figure 1(c)
RK4	fourth-order Runge–Kutta
SSP	strong stability-preserving [53–56]

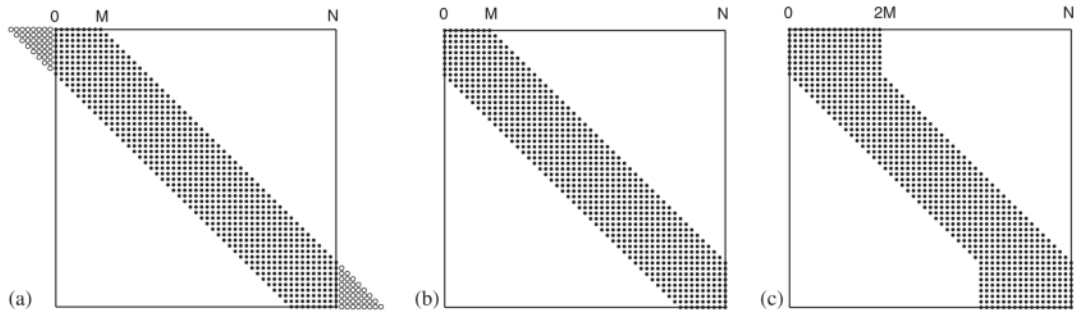


Figure 1. Illustration of the matrix structures of high-order finite difference methods: (a) central finite difference (FD); (b) one-sided finite difference type 1 (OFD1); and (c) one-sided finite difference type 2 (OFD2).

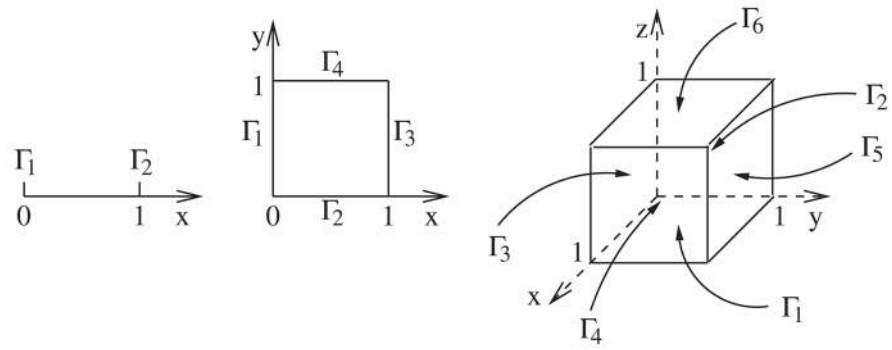


Figure 2.
Illustration of boundary notations in 1D, 2D, and 3D.

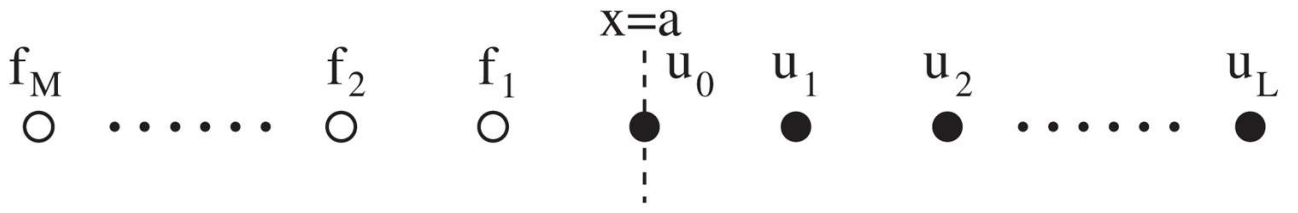


Figure 3.
Illustration of fictitious points near the left boundary.

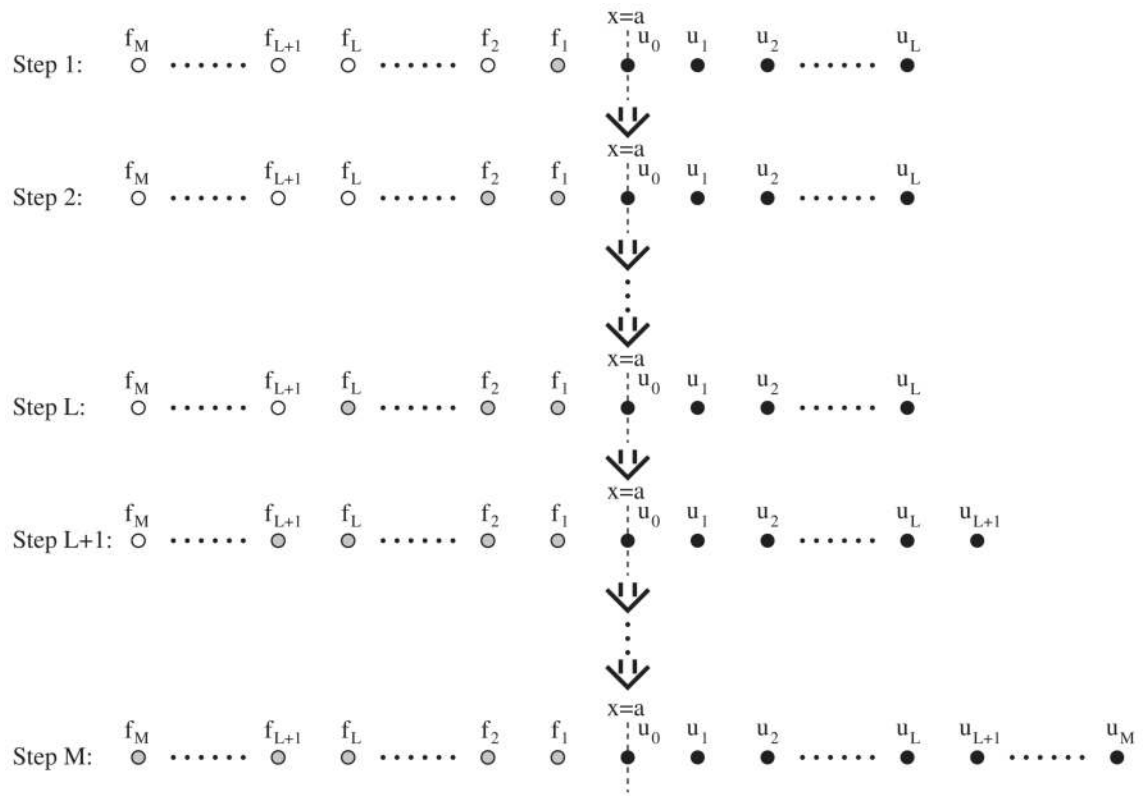


Figure 4.
Illustration of the iterative procedure.

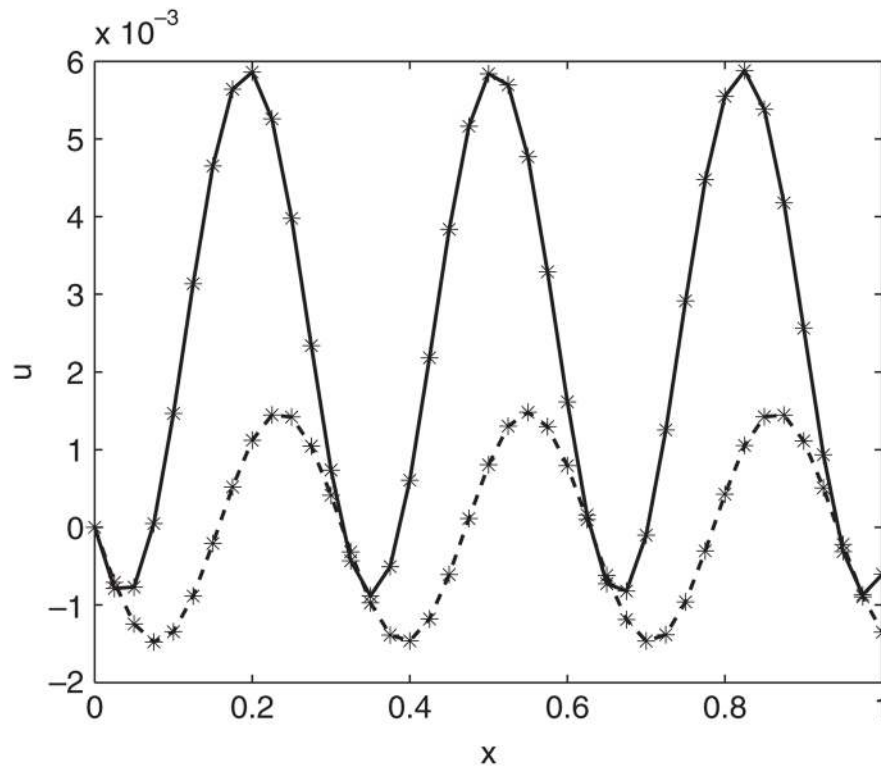


Figure 5. Analytical and numerical solutions of Example 1 in Section 3.1 for $k = 20$ and $N = 40$. Here solid and dashed lines denote, respectively, the real and imaginary parts of the analytical solution, while stars stand for the MIB result.

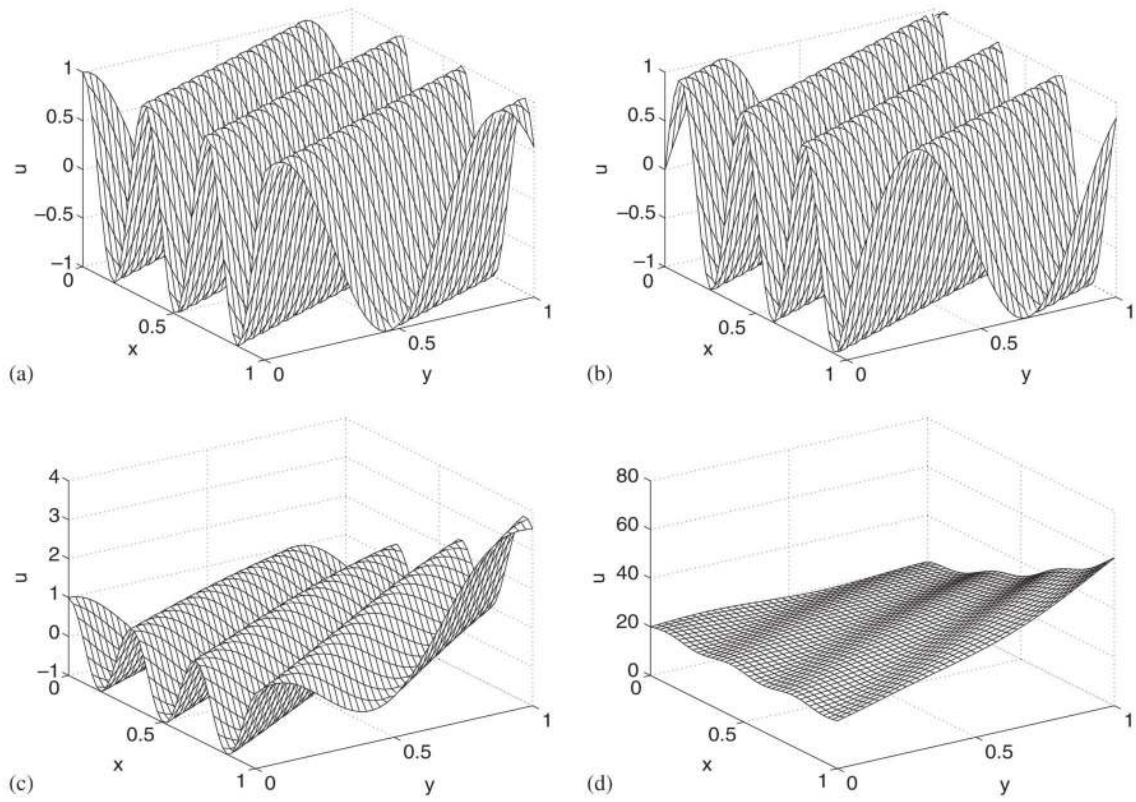


Figure 6. Numerical solutions of two 2D examples in Section 3.2. (a) Example 1, real part; (b) Example 1, imaginary part; (c) Example 2, real part; and (d) Example 2, imaginary part.

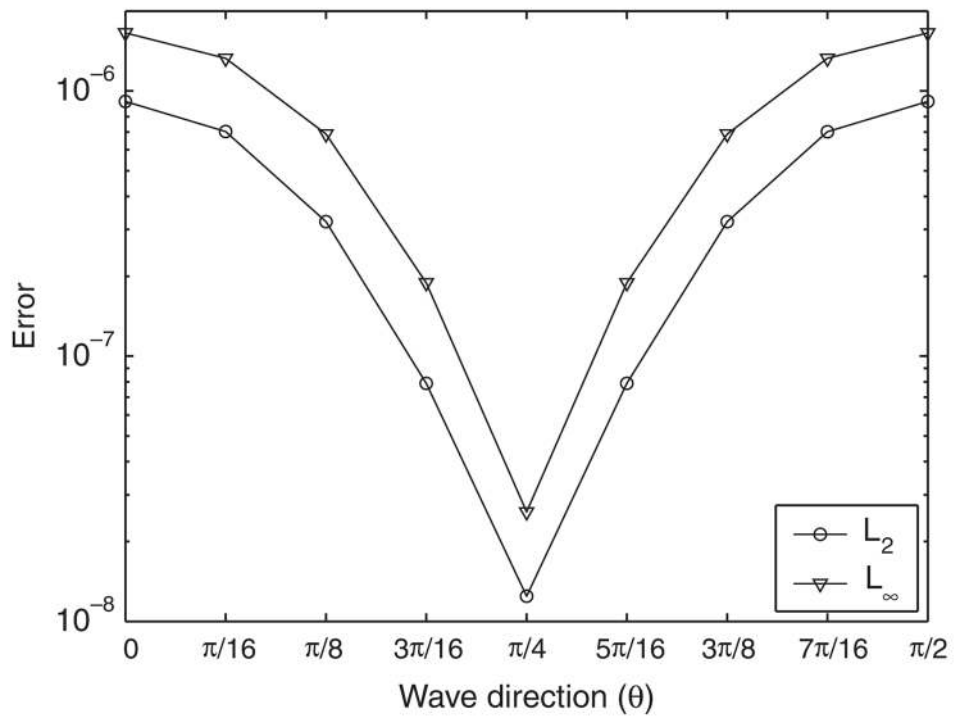


Figure 7. Dependence of MIB approximation errors on the wave angle θ .

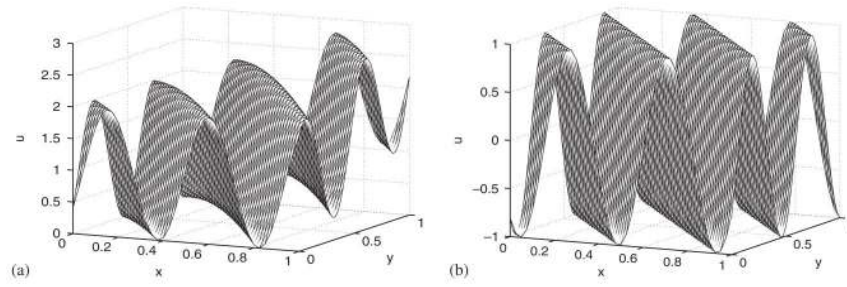


Figure 8. Slice plots at $z = 0.5$ of the numerical solution in Example 1 of Section 3.3 with $k = 20$ and $N^3 = 40^3$. In the MIB method, we set $M = 6$ and $L = 12$: (a) real part and (b) imaginary part.

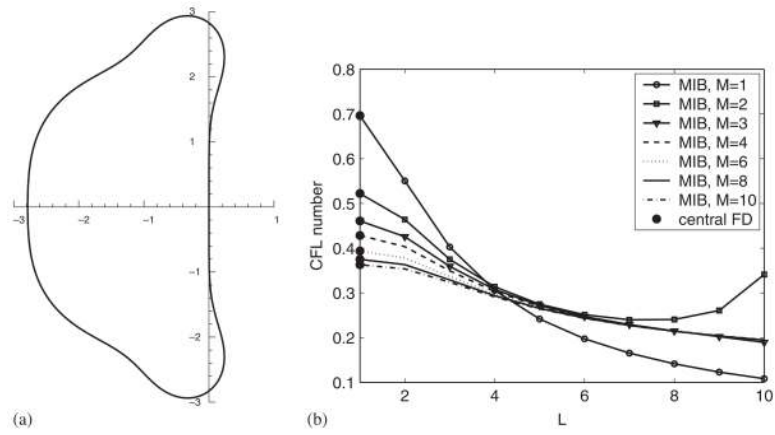


Figure 9. (a) Stability region of the SSP-RK4 method. (b) Numerical CFL numbers of the MIB and central FD methods for Example 2 in Section 4.

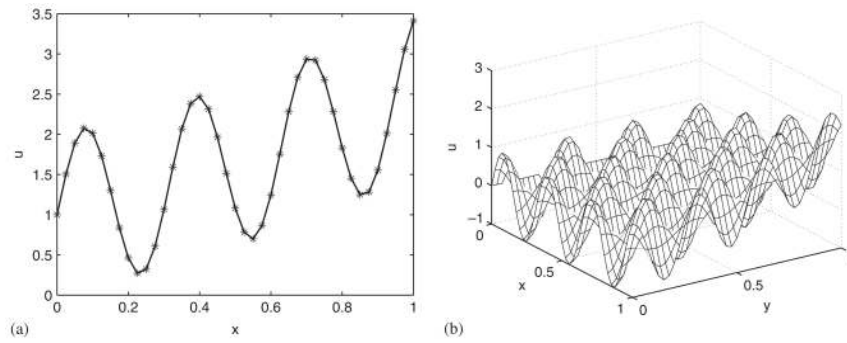


Figure 10. Solutions of Poisson equations of Section 6. (a) Analytical (solid line) and numerical solutions (stars) of Example 1 with $k = 20$, $M = 20$, $N = 40$, $r = 2.6$, and $L = 11$ and (b) numerical solutions of Example 2 with $k = 20$, $M = 20$, $N^2 = 40^2$, $r = 2.5$, and $L = 12$.

Table I

Numerical convergence tests of Example 1 in Section 3.1 with $k = 25$.

Scheme	M	Result	L_2						L_∞		
			$N = 20$	$N = 40$	$N = 80$	$N = 20$	$N = 40$	$N = 80$			
FD	1	Error	6.81 (-1)	9.30 (-2)	6.44 (-2)	8.24 (-1)	1.38 (-1)	7.65 (-2)			
		Order		2.87	0.53		2.58	0.85			
OFD1	2	Error	7.31 (-2)	1.38 (-2)	1.91 (-3)	1.06 (-1)	1.58 (-2)	1.92 (-3)			
		Order		2.40	2.85		2.75	3.04			
	4	Error	4.46 (-1)	1.23 (-2)	3.16 (-4)	4.09 (-1)	1.04 (-2)	2.73 (-4)			
		Order		5.18	5.28		5.29	5.26			
	6	Error	4.50 (-1)	5.25 (-3)	3.94 (-5)	3.98 (-1)	4.42 (-3)	3.40 (-5)			
		Order		6.42	7.06		6.49	7.02			
	8	Error	1.06 (-1)	1.80 (-3)	4.30 (-6)	9.39 (-2)	1.53 (-3)	3.72 (-6)			
		Order		5.88	8.71		5.94	8.69			
OFD2	2	Error	5.64 (-1)	3.31 (-2)	1.40 (-3)	5.83 (-1)	3.24 (-2)	1.48 (-3)			
		Order		4.09	4.57		4.17	4.46			
	4	Error	4.68 (-1)	3.04 (-3)	1.54 (-5)	4.38 (-1)	2.59 (-3)	1.34 (-5)			
		Order		7.27	7.63		7.40	7.60			
	6	Error	1.59 (-0)	1.93 (-4)	1.44 (-7)	1.39 (-0)	1.65 (-4)	1.24 (-7)			
		Order		13.01	10.39		13.04	10.38			
	8	Error	1.37 (-1)	7.07 (-5)	9.06 (-10)	1.26 (-1)	5.99 (-5)	7.83 (-10)			
		Order		10.92	16.25		11.03	16.22			
MIB	1	Error	7.81 (-1)	1.40 (-1)	3.43 (-2)	7.98 (-1)	1.90 (-1)	4.80 (-2)			
		Order		2.49	2.02		2.07	1.99			
	2	Error	7.64 (-2)	5.81 (-3)	4.07 (-4)	1.22 (-1)	8.15 (-3)	5.75 (-4)			
		Order		3.72	3.84		3.91	3.82			
	4	Error	1.05 (-2)	4.01 (-5)	1.20 (-7)	1.17 (-2)	4.80 (-5)	1.65 (-7)			
		Order		8.03	8.39		7.93	8.18			
	6	Error	2.08 (-2)	1.20 (-6)	2.47 (-10)	2.02 (-2)	1.12 (-6)	2.39 (-10)			
		Order		14.08	12.25		14.14	12.19			
	8	Error	3.94 (-3)	1.30 (-6)	1.60 (-11)	3.55 (-3)	1.11 (-6)	1.34 (-11)			

Scheme	M	Result	L_2			L_{∞}		
			$N = 20$	$N = 40$	$N = 80$	$N = 20$	$N = 40$	$N = 80$
	Order		11.57	16.30	16.30	11.65	16.33	

In the MITB method, L is set to be 1, 3, 10, 13, and 14, respectively, for $M = 1, 2, 4, 6,$ and 8 .

Table II

Numerical convergence tests of Example 1 in Section 3.2 with $k = 25$.

Scheme	M	Result	L_2						L_∞		
			$N^2 = 20^2$	$N^2 = 40^2$	$N^2 = 80^2$	$N^2 = 20^2$	$N^2 = 40^2$	$N^2 = 80^2$	$N^2 = 20^2$	$N^2 = 40^2$	$N^2 = 80^2$
FD	1	Error	6.61 (-1)	1.68 (-1)	7.37 (-2)	2.08 (-0)	3.96 (-1)	1.84 (-1)			
		Order		1.98	1.19		2.39	1.11			
OFD1	2	Error	2.49 (-1)	6.53 (-2)	1.67 (-2)	5.62 (-1)	1.44 (-1)	3.75 (-2)			
		Order		1.93	1.96		1.96	1.94			
	4	Error	3.56 (-1)	1.57 (-2)	8.78 (-4)	8.61 (-1)	3.61 (-2)	1.97 (-3)			
		Order		4.51	4.16		4.57	4.20			
	6	Error	3.22 (-1)	3.78 (-3)	5.29 (-5)	5.95 (-1)	8.49 (-3)	1.19 (-4)			
		Order		6.41	6.16		6.13	6.16			
	8	Error	2.37 (-1)	2.09 (-3)	∞	5.21 (-1)	3.80 (-3)	∞			
		Order		6.83			7.10				
OFD2	2	Error	3.66 (-1)	2.78 (-2)	1.71 (-3)	8.34 (-1)	6.40 (-2)	3.96 (-3)			
		Order		3.72	4.02		3.70	4.02			
	4	Error	3.35 (-1)	2.25 (-3)	∞	7.22 (-1)	4.66 (-3)	∞			
		Order		7.22			6.43				
MIB	1	Error	7.82 (-1)	1.77 (-1)	4.29 (-2)	2.05 (-0)	4.44 (-1)	1.05 (-1)			
		Order		2.14	2.04		2.21	2.07			
	2	Error	9.32 (-1)	6.67 (-3)	4.47 (-4)	2.55 (-1)	1.77 (-2)	1.17 (-3)			
		Order		3.81	3.90		3.85	3.92			
	4	Error	1.17 (-2)	2.54 (-5)	8.92 (-8)	2.72 (-2)	6.44 (-5)	2.22 (-7)			
		Order		8.85	8.15		8.72	8.18			
	6	Error	1.11 (-2)	2.15 (-6)	3.06 (-10)	2.47 (-2)	4.62 (-6)	6.83 (-10)			
		Order		12.33	12.78		12.39	12.72			
	8	Error	1.10 (-2)	5.82 (-7)	2.33 (-11)	2.23 (-2)	1.21 (-7)	1.67 (-10)			
		Order		14.20	14.61		14.17	12.83			

In the MIB method, L is set to be 1, 3, 10, 12, and 14, respectively, for $M = 1, 2, 4, 6$, and 8.

Table III

Numerical convergence tests of Example 2 in Section 3.2 with $k = 30$.

Scheme	M	Result	L_2					
			$N^2 = 20^2$	$N^2 = 40^2$	$N^2 = 80^2$	$N^2 = 20^2$	$N^2 = 40^2$	$N^2 = 80^2$
MIB	1	Error	3.94 (-2)	7.81 (-3)	1.92 (-3)	5.69 (-2)	1.15 (-2)	2.75 (-3)
		Order		2.33	2.02		2.30	2.07
	2	Error	5.13 (-3)	3.91 (-4)	2.71 (-5)	7.07 (-3)	5.34 (-4)	3.73 (-5)
		Order		3.71	3.85		3.73	3.84
	4	Error	1.95 (-3)	4.88 (-6)	1.41 (-8)	2.13 (-3)	5.76 (-6)	1.42 (-8)
		Order		8.65	8.43		8.53	8.67
	6	Error	2.81 (-3)	8.04 (-7)	1.58 (-10)	3.18 (-3)	8.34 (-7)	1.90 (-10)
		Order		11.77	12.32		11.90	12.10
	8	Error	4.39 (-3)	7.34 (-7)	3.94 (-11)	5.06 (-3)	8.85 (-7)	4.19 (-11)
		Order		12.55	14.18		12.48	14.37

In the MIB method, L is set to be 1, 3, 9, 12, and 13, respectively, for $M = 1, 2, 4, 6$, and 8.

Table IV

Numerical efficiency tests of Example 1 in Section 3.2 with $k = 1$.

Scheme	N^2	L_2		L_∞		CPU	
		Error	Order	Error	Order	Sec.	Ratio
MIB	10^2	1.29 (-12)		1.54 (-12)		0.11	
	10^2	2.35 (-2)		3.09 (-2)		0.03	
FD with boundary closure scheme 1	20^2	1.17 (-2)	1.00	1.55 (-2)	0.99	0.11	3.67
	40^2	5.84 (-3)	1.00	7.76 (-3)	1.00	1.03	9.36
	80^2	∞		∞		∞	
	10^2	2.36 (-2)		2.97 (-2)		0.03	
FD with boundary closure scheme 2	20^2	1.17 (-2)	1.01	1.52 (-2)	0.97	0.10	3.33
	40^2	5.85 (-3)	1.00	7.69 (-3)	0.98	0.54	5.40
	80^2	2.92 (-3)	1.00	3.87 (-3)	0.99	4.45	8.24
	160^2	1.46 (-3)	1.00	1.94 (-3)	1.00	37.92	8.52
	320^2	7.29 (-4)	1.00	9.71 (-4)	1.00	445.37	11.74
	640^2	3.65 (-4)	1.00	4.86 (-4)	1.00	7715.82	17.32
	1280^2	1.82 (-4)	1.00	2.43 (-4)	1.00	90869.82	11.78
	171798691840^2	1.36 (-12)	1.00	1.81 (-12)	1.00	1.25 (+34)	12.00

Both boundary closure scheme 1 and 2 of Section 2.2 are considered for the FD method. The 16th order MIB method with $M = 8$ and $L = 10$ is used. CPU time in second is reported.

Table V

Numerical convergence tests of Example 1 in Section 3.3 with $k = 12$.

Scheme	M	Result	L_2			L_∞		
			$N^3 = 12^3$	$N^3 = 24^3$	$N^3 = 48^3$	$N^3 = 12^3$	$N^3 = 24^3$	$N^3 = 48^3$
MIB	1	Error	1.28 (-1)	2.67 (-2)	2.12 (-1)	4.33 (-2)		
		Order		2.26		2.29		
	2	Error	5.19 (-3)	3.85 (-4)	1.11 (-2)	7.82 (-4)		
		Order		3.75		3.83		
	4	Error	1.08 (-4)	2.06 (-7)	1.76 (-4)	2.96 (-7)		
		Order		9.04		9.22		
	8	Error	2.38 (-5)	3.45 (-9)	2.71 (-5)	4.52 (-9)		
		Order		12.75		12.55		

In the MIB method, L is set to be 1, 3, 9, and 12, respectively, for $M = 1, 2, 4$, and 8.

Table VI

Numerical errors of the MIB method for the time-dependent equations of Section 4.

t	Example 1		Example 2	
	L_{∞}	L_2	L_{∞}	L_2
1	2.51 (-12)	2.64 (-12)	3.55 (-13)	2.95 (-13)
2	5.25 (-12)	6.02 (-12)	3.26 (-12)	2.86 (-12)
3	4.43 (-12)	6.00 (-12)	1.87 (-12)	1.69 (-12)
4	5.52 (-12)	6.65 (-12)	3.06 (-12)	2.73 (-12)
5	1.33 (-11)	1.55 (-11)	3.33 (-12)	2.97 (-12)
6	1.81 (-11)	2.23 (-11)	3.39 (-12)	3.02 (-12)
7	9.64 (-12)	1.61 (-11)	3.41 (-12)	3.04 (-12)
8	7.59 (-12)	8.56 (-12)	3.41 (-12)	3.04 (-12)
9	5.66 (-12)	1.25 (-11)	1.98 (-11)	1.75 (-11)
10	1.06 (-11)	1.59 (-11)	2.51 (-11)	2.22 (-11)

In Example 1, $\Delta t = 2.5 \times 10^{-4}$, while in Example 2, $\Delta t = 2.0 \times 10^{-5}$.

Table VII

CFL numbers in Section 4.

<i>M</i>	Example 1				Example 2			
	Central FD		MIB		Central FD		MIB	
	Analytical	Numerical	Numerical	<i>L</i>	Analytical	Numerical	Numerical	<i>L</i>
1	2.82843	2.82805	2.82805	1	0.696323	0.696379	0.695894	1
2	2.06120	2.06101	2.06101	1,2	0.522243	0.522193	0.522193	1
3	1.78340	1.78348	1.78348	1, ..., 5	0.460802	0.460829	0.460829	1
4	1.63436	1.63425	1.63425	1, ..., 6	0.428402	0.428449	0.428449	1
6	1.47249	1.47254	1.47254	1, ..., 7	0.393796	0.393701	0.393701	1
8	1.38318	1.38313	1.38313	1, ..., 7	0.375027	0.375094	0.375094	1
10	1.32512	1.32503	1.32503	1, ..., 7	0.362973	0.362976	0.363108	1

In Example 1, for each *M*, the MIB takes the same numerical CFL number for all reported *L* values. In Example 2, for each *M*, the MIB takes the reported CFL number only for *L* = 1. When *L* is larger, the MIB is stable under a smaller CFL number.

Table VIII

The largest real part of the discrete spectrum of the MIB method for Example 1 in Section 4.

M	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 6$	$L = 7$	$L = 8$
1	4.77 (-7)	1.16 (+3)						
2	2.91 (-10)	4.69 (-7)	2.16 (+1)					
3	7.97 (-7)	1.12 (-6)	1.26 (-6)	1.14 (-6)	1.13 (-10)	2.44 (+2)		
4	2.49 (-7)	9.14 (-7)	1.76 (-6)	2.90 (-6)	2.54 (-6)	2.67 (-6)	1.88 (+1)	
6	8.17 (-11)	1.27 (-10)	9.11 (-11)	8.28 (-11)	9.29 (-11)	2.52 (-10)	2.73 (-9)	7.94 (+1)
8	1.09 (-10)	1.14 (-10)	1.25 (-10)	1.85 (-6)	3.00 (-6)	7.22 (-6)	5.93 (-6)	5.89 (+1)
10	1.10 (-10)	1.04 (-10)	1.08 (-10)	1.85 (-10)	1.77 (-10)	1.98 (-10)	4.67 (-10)	4.06 (+1)

Table IX
 Analysis of the discrete spectrum of the MIB method for Example 2 in Section 4.

L	$M = 1$			$M = 2$		
	$\min\{\text{Re}(\lambda)\}$	$\max\{\text{Re}(\lambda)\}$	$\max\{\text{Im}(\lambda)\}$	$\min\{\text{Re}(\lambda)\}$	$\max\{\text{Re}(\lambda)\}$	$\max\{\text{Im}(\lambda)\}$
1	-4.00 (+4)	0.00 (+0)	0.00 (+0)	-5.33 (+4)	0.00 (+0)	0.00 (+0)
2	-5.06 (+4)	0.00 (+0)	0.00 (+0)	-6.00 (+4)	0.00 (+0)	0.00 (+0)
3	-6.92 (+4)	7.21 (-11)	0.00 (+0)	-7.42 (+4)	2.38 (-11)	0.00 (+0)
4	-9.11 (+4)	0.00 (+0)	0.00 (+0)	-8.87 (+4)	2.71 (-12)	0.00 (+0)
5	-1.15 (+5)	0.00 (+0)	0.00 (+0)	-1.01 (+5)	5.76 (-11)	0.00 (+0)
6	-1.41 (+5)	0.00 (+0)	0.00 (+0)	-1.10 (+5)	0.00 (+0)	0.00 (+0)
7	-1.68 (+5)	0.00 (+0)	0.00 (+0)	-1.16 (+5)	0.00 (+0)	0.00 (+0)
8	-1.97 (+5)	8.09 (-13)	0.00 (+0)	-1.15 (+5)	1.94 (-9)	2.84 (+3)
9	-2.27 (+5)	0.00 (+0)	0.00 (+0)	-1.06 (+5)	0.00 (+0)	1.09 (+4)
10	-2.57 (+5)	1.53 (-9)	1.29 (+3)	-8.05 (+4)	0.00 (+0)	1.93 (+4)
L	$M = 3$			$M = 4$		
	$\min\{\text{Re}(\lambda)\}$	$\max\{\text{Re}(\lambda)\}$	$\max\{\text{Im}(\lambda)\}$	$\min\{\text{Re}(\lambda)\}$	$\max\{\text{Re}(\lambda)\}$	$\max\{\text{Im}(\lambda)\}$
1	-6.04 (+4)	0.00 (+0)	0.00 (+0)	-6.50 (+4)	0.00 (+0)	0.00 (+0)
2	-6.54 (+4)	0.00 (+0)	0.00 (+0)	-6.90 (+4)	0.00 (+0)	0.00 (+0)
3	-7.74 (+4)	0.00 (+0)	0.00 (+0)	-7.97 (+4)	0.00 (+0)	0.00 (+0)
4	-9.05 (+4)	1.40 (-12)	0.00 (+0)	-9.18 (+4)	0.00 (+0)	0.00 (+0)
5	-1.02 (+5)	0.00 (+0)	0.00 (+0)	-1.03 (+5)	1.75 (-11)	0.00 (+0)
6	-1.12 (+5)	2.84 (-11)	0.00 (+0)	-1.13 (+5)	2.29 (-11)	0.00 (+0)
7	-1.21 (+5)	1.53 (-10)	0.00 (+0)	-1.21 (+5)	0.00 (+0)	0.00 (+0)
8	-1.29 (+5)	0.00 (+0)	0.00 (+0)	-1.29 (+5)	2.20 (-10)	0.00 (+0)
9	-1.37 (+5)	1.08 (-10)	4.62 (+3)	-1.36 (+5)	1.61 (-10)	4.45 (+3)
10	-1.46 (+5)	0.00 (+0)	8.68 (+3)	-1.42 (+5)	0.00 (+0)	9.10 (+3)

Table X

Comparison of fundamental frequencies for Example 1 of the sixth-order eigenvalue problem in Section 5.1.

Scheme	N	$r = 1.0$	$r = 1.1$	$r = 1.2$	$r = 1.3$	$r = 1.4$	$r = 1.5$
DQM [24]	—	2.268	2.417	2.561	2.701	2.839	2.976
Rayleigh-Ritz [24]	—	2.274	2.416	2.557	2.697	2.834	2.970
GDQR [32]	6	2.2631	2.4133	2.5597	2.7139	2.8946	3.1297
	7	2.2669	2.4137	2.5565	2.6944	2.8242	2.9407
	8	2.2667	2.4137	2.5567	2.6962	2.8318	2.9623
	9	2.2667	2.4137	2.5568	2.6966	2.8336	2.9681
	10	2.2667	2.4137	2.5568	2.6966	2.8335	2.9678
LaDQM [32]	6	2.2624	2.4135	2.5583	2.7019	2.8452	2.9878
	7	2.2647	2.4136	2.5576	2.6995	2.8400	2.9791
	8	2.2669	2.4137	2.5570	2.6976	2.8364	2.9738
	9	2.2668	2.4137	2.5569	2.6972	2.8353	2.9715
	10	2.2667	2.4137	2.5568	2.6968	2.8341	2.9694
MIB	8	2.2658	2.4136	2.5572	2.6982	2.8374	2.9747
	9	2.2665	2.4137	2.5570	2.6974	2.8357	2.9720
	10	2.2668	2.4137	2.5568	2.6970	2.8346	2.9702
	11	2.2668	2.4137	2.5568	2.6968	2.8341	2.9691
	12	2.2667	2.4137	2.5568	2.6967	2.8337	2.9685

In the LaDQM [32], $M = N + 2$, while in the MIB, $M = L = N$.

Table XI

Comparison of fundamental frequencies for Example 2 of the sixth-order eigenvalue problem in Section 5.1.

Scheme	N	$r = 1.0$	$r = 1.1$	$r = 1.2$	$r = 1.3$	$r = 1.4$	$r = 1.5$
DQM [24]	—	2.686	2.849	3.010	3.171	3.332	3.493
Rayleigh-Ritz [24]	—	2.687	2.846	3.006	3.167	3.326	3.486
GDQR [32]	5	2.6828	2.8452	3.0062	3.1666	3.3267	3.4861
	6	2.6833	2.8452	3.0062	3.1665	3.3263	3.4858
	7	2.6833	2.8452	3.0062	3.1665	3.3262	3.4857
	8	2.6833	2.8452	3.0062	3.1665	3.3263	3.4858
	9	2.6833	2.8452	3.0062	3.1665	3.3263	3.4858
LaDQM [32]	5	2.6956	2.8523	3.0199	3.1917	3.3595	3.5150
	6	2.6828	2.8488	3.0181	3.1884	3.3577	3.5251
	7	2.6830	2.8488	3.0182	3.1887	3.3579	3.5230
	8	2.6833	2.8489	3.0181	3.1884	3.3578	3.5252
	9	2.6833	2.8489	3.0181	3.1884	3.3578	3.5248
MIB	7	2.6854	2.8458	3.0067	3.1680	3.3289	3.4882
	8	2.6831	2.8452	3.0062	3.1667	3.3269	3.4867
	9	2.6832	2.8452	3.0062	3.1666	3.3267	3.4864
	10	2.6833	2.8452	3.0062	3.1665	3.3264	3.4859
	11	2.6833	2.8452	3.0062	3.1665	3.3263	3.4859

In the LaDQM [32], $M = N + 2$, while in the MIB, $M = L = N$.

Table XII

Maximum absolute errors of Example 1 of the eighth-order boundary value problem in Section 5.2.

$y^{(k)}$	Spline [57] ($N = 63$)		GDQR [32]		LaDQM [32]		MIB	
	$[x_3, x_{N-4}]$	Otherwise	$N = 6$	$N = 10$	$N = 6$	$N = 10$	$N = 6$	$N = 10$
$k = 0$	9.44 (-5)	4.94 (+3)	3.13 (-6)	1.54 (-11)	7.58 (-5)	2.71 (-9)	2.76 (-4)	3.90 (-8)
$k = 1$	1.45 (-4)	1.17 (+5)	5.09 (-6)	2.75 (-11)	1.30 (-4)	4.83 (-9)	4.44 (-4)	6.35 (-8)
$k = 2$	2.34 (-4)	4.98 (+7)	7.52 (-6)	4.07 (-11)	1.90 (-4)	7.31 (-9)	6.86 (-4)	9.57 (-8)
$k = 3$	3.58 (-4)	1.05 (+10)	1.50 (-5)	9.28 (-11)	4.10 (-4)	1.66 (-8)	1.17 (-3)	1.77 (-7)
$k = 4$	5.87 (-4)	2.45 (+12)	2.20 (-5)	1.94 (-10)	6.48 (-4)	2.82 (-8)	1.79 (-3)	2.49 (-7)
$k = 5$	8.61 (-4)	2.40 (+14)	1.83 (-4)	1.58 (-9)	2.34 (-3)	1.18 (-7)	3.01 (-3)	1.47 (-6)
$k = 6$	1.73 (-3)	1.20 (+16)	4.09 (-4)	3.21 (-9)	4.53 (-3)	2.69 (-7)	6.02 (-3)	2.73 (-6)
$k = 7$	1.21 (-2)	2.57 (+17)	8.81 (-3)	1.70 (-7)	7.77 (-2)	6.93 (-6)	1.28 (-1)	2.90 (-5)

In the LaDQM [32], $M = N + 3$, while in the MIB, $M = L = N$.

Table XIII

Maximum absolute errors of Example 2 of the eighth-order boundary value problem in Section 5.2.

$y^{(k)}$	Spline [57] ($N = 63$)		GDQR [32]		LaDQM [32]		MIB	
	$[x_3, x_{N-4}]$	Otherwise	$N = 6$	$N = 10$	$N = 6$	$N = 10$	$N = 6$	$N = 10$
$k = 0$	1.31 (-4)	2.24 (+3)	2.61 (-6)	1.49 (-11)	7.38 (-5)	4.05 (-9)	2.25 (-4)	1.24 (-8)
$k = 1$	2.02 (-4)	5.50 (+4)	4.16 (-6)	2.34 (-11)	1.16 (-4)	6.38 (-9)	3.52 (-4)	2.20 (-8)
$k = 2$	3.24 (-4)	4.69 (+6)	6.23 (-6)	3.64 (-11)	1.81 (-4)	9.97 (-9)	5.61 (-4)	3.29 (-8)
$k = 3$	5.01 (-4)	4.95 (+9)	1.13 (-5)	5.85 (-11)	2.90 (-4)	1.60 (-8)	8.48 (-4)	7.63 (-8)
$k = 4$	7.94 (-4)	1.15 (+12)	1.47 (-5)	8.49 (-11)	4.36 (-4)	2.41 (-8)	1.48 (-3)	1.31 (-7)
$k = 5$	1.26 (-3)	1.13 (+14)	4.75 (-5)	1.82 (-10)	8.29 (-4)	4.62 (-8)	2.11 (-3)	5.38 (-7)
$k = 6$	1.85 (-3)	5.67 (+15)	9.11 (-5)	4.66 (-10)	8.94 (-4)	5.21 (-8)	4.84 (-3)	2.25 (-6)
$k = 7$	2.15 (-3)	1.21 (+17)	1.30 (-3)	1.62 (-8)	1.19 (-2)	1.22 (-6)	4.06 (-2)	5.68 (-5)

In the LaDQM [32], $M = N + 3$, while in the MIB, $M = L = N$.

Table XIV

Numerical errors of the discrete singular convolution method with $M = 20$ for Example 1 and 2 of Section 6.

Example	k	$N = 20$		$N = 40$	
		L_2	L_{∞}	L_2	L_{∞}
1	10	9.37 (-7)	9.92 (-7)	1.05 (-10)	1.26 (-10)
	20	2.21 (-2)	2.08 (-2)	3.00 (-6)	2.81 (-6)
2	10	3.02 (-7)	3.44 (-7)	1.00 (-10)	1.94 (-10)
	20	7.25 (-4)	1.08 (-3)	2.03 (-7)	3.02 (-7)

In Example 1, $r = 2.6$ and $L = 11$. In Example 2, $r = 2.5$ and $L = 12$.