Journal of Computational Physics 228 (2009) 55–74

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Fourier spectral embedded boundary solution of the Poisson's and Laplace equations with Dirichlet boundary conditions

Feriedoun Sabetghadam [a,*], Shervin Sharafatmandjoor [a], Farhang Norouzi [b]

[a] Mechanical and Aerospace Engineering Faculty, Science and Research Branch, Islamic Azad University (IAU), Pounak Square, Tehran, Iran
[b] Department of Mechanical Engineering, McGill University, Montreal, Quebec, Canada

ARTICLE INFO

Article history:
Received 19 November 2007
Received in revised form 13 August 2008
Accepted 14 August 2008
Available online 5 September 2008

Keywords:
Embedded boundary
FFT-based methods
Numerical boundaries
Relaxation methods
Saturation state

ABSTRACT

A Fourier spectral embedded boundary method, for solution of the Poisson's equation with Dirichlet boundary conditions and arbitrary forcing functions (including zero forcing function), is presented in this paper. This iterative method begins by transformation of the Dirichlet boundary conditions from the physical boundaries to some corresponding regular grid points (which are called the numerical boundaries), using a second order interpolation method. Then the transformed boundary conditions and the forcing function are extended to a square, smoothly and periodically, via multiplying them by some suitable error functions. Instead of direct solution of the resulting extended Poisson's problem, it is suggested to define and solve an equivalent transient diffusion problem on the regular domain, until achievement of the steady solution (which is considered as the solution of the original problem). Without need of any numerical time integration method, time advancement of the solution is obtained directly, from the exact solution of the transient problem in the Fourier space. Consequently, timestep sizes can be chosen without stability limitations, which it means higher rates of convergence in comparison with the classical relaxation methods. The method is presented in details for one- and two-dimensional problems, and a new emerged phenomenon (which is called the saturation state) is illustrated both in the physical and spectral spaces. The numerical experiments have been performed on the one- and two-dimensional irregular domains to show the accuracy of the method and its superiority (from the rate of convergence viewpoint) to the other classical relaxation methods. Capability of the method, in dealing with complex geometries, and in presence of discontinuity at the boundaries, has been shown via some numerical experiments on a four-leaf shape geometry.

1. Introduction

In the embedded boundary methods (EBM), a domain Ω with irregular boundary $\partial\Omega$ (where the solution is sought on it), is surrounded with a bigger domain D with regular boundaries (coinciding with the considered coordinate axes). Then the problem is extended from the irregular domain Ω to the whole of the regular domain D, and solution of the extended problem is obtained by the use of high efficiency numerical methods. A typical drawing for a two-dimensional Cartesian grid is shown in Fig. 1.

Although suggestion of such methods has a long history which returns to the early days of numerical methods [7], just recently (because of some deficiencies of the conventional structured and unstructured grids in dealing with the moving boundary and multi-body problems, multiphase flows and so forth), these methods have become in the center of attention of many academic, as well as applied researchers [2,10,16,12,18].

[*] Corresponding author. Tel.: +98 21 44869724; fax: +98 21 44869728.
E-mail address: fsabet@sr.iau.ac.ir (F. Sabetghadam).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.08.018

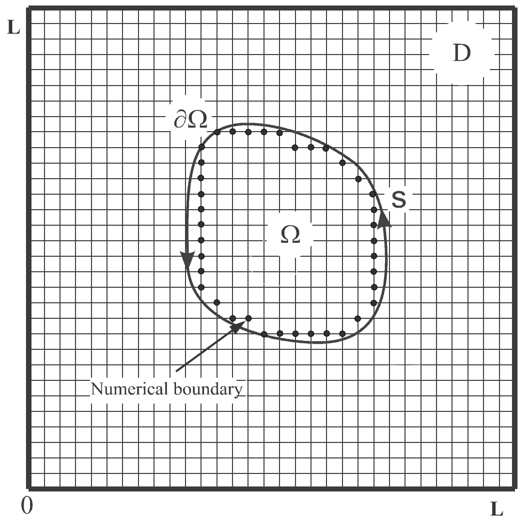


Fig. 1. Regular and irregular (physical) boundaries and the corresponding numerical boundary (generated for a typical uniform Cartesian grid), for a two-dimensional problem.

In this context, the fast Fourier transform-based (FFT-based) methods, which formally are $\mathcal{O}(N\log N)$, can be considered as the efficient methods for solution of the extended problems. Nevertheless, use of these methods in the EBM have been considered just recently, mainly because of their drawbacks in facing with discontinuities and their intrinsic restrictions in implementation of the general boundary conditions [2,4,6]. Prior to these works almost all of the spectral methods, suggested for the general irregular boundaries, employed a kind of the spectral element methods, or the boundary element methods in implementation of the Dirichlet boundary conditions [1,8].

The FFT-based embedded boundary solution of the Poisson's equation with homogeneous Neumann boundary condition and without any boundary condition (by appropriate extension of the forcing function), have been proposed in [2,4,6], respectively. The present article, as a generalization to these methods, is devoted to the Fourier embedded boundary solution of the Poisson's problem with Dirichlet boundary conditions.

In addition to their wide applications in the diffusion and filtering problems and the domain decomposition methods, these problems have a crucial role in the fluid flow simulations. In fact, the present method can be used directly in solution of the vorticity-stream function formulation of the two-dimensional incompressible Navier–Stokes equations [15,18]. Here, without loss of generality, only the one- and two-dimensional problems are discussed. But, as it will be seen, extension of the method to three-dimensional problems is straightforward. Moreover, the method is directly applicable to the Laplace equation (i.e. the Poisson's equation with zero forcing function), which is not the case for many other similar methods.

For a d-dimensional domain $\Omega \subset \mathbb{R}^d$ and its boundary $\partial\Omega$, and for the given functions $F : \Omega \to \mathbb{R}$ and $v_b : \partial\Omega \to \mathbb{R}$, the Poisson's problem with Dirichlet boundary condition is defined as

$$\sum_{i=1}^{d} \frac{\partial^2 v}{\partial x_i^2} = F \quad \text{in } \Omega, \tag{1}$$

$$v(\mathbf{S}) = v_s \quad \text{on } \partial\Omega. \tag{2}$$

In Fig. 1, the Ω has been shown as a two-dimensional domain with irregular closed boundary $\partial\Omega$, defined by a vector function $\mathbf{S}$. To stress their differences, the irregular boundary $\partial\Omega$ will be noted as the 'physical boundary', while the corresponding points on the uniform grid (marked by the bold circles in the figure), will be called the 'numerical boundary'. As a necessary condition for good representation of the physical boundaries, it is assumed that the vector function $\mathbf{S}$ is defined everywhere on the boundary, with a resolution sufficiently greater than the uniform computational grid.

Inspired by the methods of Bueno-Orovio et al. [6], Bueno-Orovio and Perez-Garcia [5], Boyd [2] and then Bueno-Orovio [4], the main idea is suitable extension of the forcing function F into $D \setminus (\Omega \cup \partial\Omega)$, such that the extended function can be transformed to the Fourier space. Then the solution of the extended problem will be found in the Fourier space, and finally the desired solution of the original problem (defined in Ω) can simply be found by ignoring the solution in the $D \setminus (\Omega \cup \partial\Omega)$.

Unfortunately, the Poisson's problem with Dirichlet boundary conditions in the above formulation, cannot be solved directly by the methods of Boyd [2], Bueno-Orovio [4] or Bueno-Orovio et al. [6], at least for the following two reasons:

(1) Absence of an explicit method for implementation of the Dirichlet boundary conditions in these works. However, addition of a homogeneous solution (which can be obtained from the boundary element method or capacity matrix

method) is suggested in [2] which is useful, when use of other than FFT-based methods is desired. In the present method, where it is aimed to solve the problem solely by applying the FFT directly, decomposition of the problem into homogeneous and non-homogeneous parts cannot help implementation of the Dirichlet boundary conditions.

(2) Absence of a method for transforming the Dirichlet boundary conditions from the physical boundaries to the numerical boundaries. In the solution of a discretized problem on a regular grid, implementation of the Dirichlet boundary conditions should be done via setting the dependent variable at some of the regular grid points, which we call them the numerical boundaries. On the other hand, these boundary conditions are originally defined at the physical boundaries. Therefore, accurate implementation of the Dirichlet boundary conditions requires a method for appropriate transformation of the boundary data [1,8,12,13].
Unfortunately, since the Fourier coefficients are global quantities and all the local information will be lost in transformation to the Fourier space, any local manipulation on the boundary data in the Fourier space is impossible. This is a challenge for any FFT-based embedded boundary method, applying on the problems with Dirichlet boundary conditions. For example, in [1] (where a method for boundary data transformation to the regular grid is proposed by the use of an optimization process), the necessity of localized data manipulation leads to use of the boundary element method.

Our suggestion in this paper is to replace the original Poisson's problem by an equivalent transient diffusion problem and following an iterative procedure, in which, all the local manipulations on the boundary conditions are done in the physical space, while the time advancement of the solution is obtained in the Fourier space. As it will be seen, this procedure results in a spectral relaxation method, which is basically different from all the aforementioned methods.

The idea of replacing an elliptic equation with its equivalent unsteady diffusion problem has been used many times in the literature (for example, see [9] for a finite difference formulation), but in the field of spectral methods, with the best knowledge of the authors, the present algorithm can be supposed as a new methodology.

Like the other embedded/immersed boundary methods, it is expected that the method would be more efficient than the conventional non-embedded boundary methods (like finite element methods or others); in dealing with complex and moving boundary problems with large mesh deformations or anywhere that the remeshing process is needed.

In the following sections, details of the mathematical formulation and our numerical experiments are presented. Section 2 describes mathematical formulation and our suggested iterative algorithm for one- and two-dimensional problems. Moreover, the saturation phenomenon (in implementation of the numerical boundary conditions) is explained and its effects on the accuracy of the final solution are discussed. Section 3 is devoted to our numerical experiments on the one- and two-dimensional problems, in which the flexibility of the method in dealing with fairly complex geometries and large amounts of discontinuities at the boundaries are assessed and discussed. Finally, the conclusions and our suggestions about the possible future works could be seen in the Section 4.

2. Mathematical formulation

For embedded boundary solution of the Poisson's equation (1) with Dirichlet boundary condition (2), after extension of the forcing function F (which will be called f), it is suggested to replace the problem with an equivalent relaxation problem as

$$\begin{aligned} &\frac{\partial u}{\partial t} = \sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2} - f \quad \text{on } D, \\ &u(\mathbf{x}, t=0) = u_0(\mathbf{x}), \end{aligned} \tag{3}$$

in which the transient solution $u(\mathbf{x}, t)$, is the equivalent (periodic extension) of $v(\mathbf{x})$, defined on the regular domain D. The initial condition $u_0(\mathbf{x})$ can be determined from the previous iteration or an initial guess. However, it must satisfy the Dirichlet boundary condition (2); that is,

$$u_0(\mathbf{S}) = v_s. \tag{4}$$

Now, since the functions $u(\mathbf{x}, t)$ and $f(\mathbf{x})$ are periodic: (i) the problem (3) (including the initial condition) can be transformed into the Fourier space, and (ii) the time integration can be performed exactly in the Fourier space without need of any numerical integration method (which means that fairly large timesteps can be chosen without any stability limitation). Particularly, this feature of the method makes it to be competitive (from the rate of convergence viewpoint) in the family of the relaxation methods.

In this section, the details of mathematical formulation of the method are discussed. At first, the method is fully described for the one-dimensional problems and then, extension to the two-dimensional problems is provided. Full coverage of the one-dimensional problems is due to the fact that the solution methodologies for one- and two-dimensional problems are essentially identical. As will be seen, the main difference between these problems is the way of boundary data transformation, which is just a subsidiary step in the hierarchy of the whole algorithm. Moreover, in the one-dimensional problems, the effects of discontinuities on the solution accuracy (which is one of the main difficulties for any spectral method), can be assessed more easily and accurately.

2.1. One-dimensional formulation

Consider the regular domain $D = \{x | x \in [0, 2\pi]\}$ and a subdomain $\Omega = \{x | x \in]a, b[,\ 0 < a < b < 2\pi\}$ on it as the irregular domain, and its boundaries $\partial\Omega = \{a, b\}$. The Poisson's problem with Dirichlet boundary conditions is defined on the irregular domain Ω as

$$\begin{aligned} &\frac{\partial^2 v}{\partial x^2} = F(x), \quad x \in \Omega, \\ &v(a) = v_a, \\ &v(b) = v_b. \end{aligned} \tag{5}$$

To discretize the problem, the domain D is covered by a uniform N-point grid, and to be a general embedded boundary problem, it is assumed that a and b are not coinciding with the regular grid points. The regular grid points are named x_i, where $i = 0, 1, 2, \ldots, N-1$. Furthermore, with a notation similar to Jomaa and Macaskill [13], it is assumed that $x_l \leqslant a \leqslant x_{l+1}$, and $x_r \leqslant b \leqslant x_{r+1}$ and also $x_{l+1} - a = \alpha_l \Delta x$ and $b - x_r = \alpha_r \Delta x$, where $\Delta x = \frac{2\pi}{N-1}$. Using these definitions, the numerical domain and numerical boundaries are defined as $\overline{\Omega} =]x_{l+1}, x_r[$ and $\partial\overline{\Omega} = \{x_{l+1}, x_r\}$, respectively. Fig. 2 shows a portion of the regular domain D, including the left physical and numerical boundaries.

Now the suggested iterative algorithm for solution of the problem (5) contains the following steps:

(1) Boundary data transformation; which means replacement of the Dirichlet boundary conditions (defined at the physical boundaries $\partial\Omega = \{a, b\}$) with some equivalent Dirichlet boundary conditions at the numerical boundaries $\partial\overline{\Omega} = \{x_{l+1}, x_r\}$. It will be done using a second order interpolation.
(2) Definition of an equivalent relaxation (diffusion) problem on $(\overline{\Omega} \cup \partial\overline{\Omega})$, which its final solution will be considered as an approximation for the desired solution.
(3) Extending the forcing function F and the initial condition, from the irregular domain $(\overline{\Omega} \cup \partial\overline{\Omega})$ to the regular domain D, smoothly and periodically. This step will be done in a way similar to Boyd [2] and Bueno-Orovio [4].
(4) Time advancement. This step itself contains: (i) transformation of the resulting (extended, periodic) problem into the Fourier space (via a FFT), (ii) time integration in the Fourier space for a timestep Δt (using the exact solution of the diffusion problem in the Fourier space), and (iii) transforming the results back into the physical space, using an Inverse FFT (IFFT).Note that at the end of this step, the numerical boundary values are slightly changed.
(5) Going back to the step (1).

The above loop should be repeated until achieving the desired accuracy. Finally, by ignoring the solution on $D \setminus (\overline{\Omega} \cup \partial\overline{\Omega})$ and addition of the physical boundary conditions, an approximation to the solution for the problem (5) on the irregular domain $(\Omega \cup \partial\Omega)$ is obtained. To give a better idea of the algorithm, a flow diagram for the above steps is provided in Fig. 3. Moreover, the steps (1)–(4) are discussed below in details.

2.1.1. Boundary data transformation

It is desired to find the equivalent numerical boundary conditions (that is, the values of $v(x_{l+1})$ and $v(x_r)$), which can replace the physical boundary conditions (that is, the values of v_a and v_b), in the problem (5). During the last years, a wide variety of boundary data treatments have been proposed and used in various EBM and IBM methods, including definition of some virtual forcing functions [15,16] and the boundary condition manipulations [12,19,13,18]. In this article, a second order interpolation, which is a combination of the methods of Sjogreen and Petersson [19] and Russell and Wang [18], has been used. The method is described for the left boundary and can be followed easily for the other one.

The solution domain is considered from x_{l+2} to x_{r-1} with its boundaries at x_{l+1} and x_r. Since it is desired to change the original problem with an unsteady problem, the values of $\{x_{l+2}, x_{l+3}, \ldots, x_{r-2}, x_{r-1}\}$ are known from the previous iteration or the initial guess. Now, at the left numerical boundary, we have

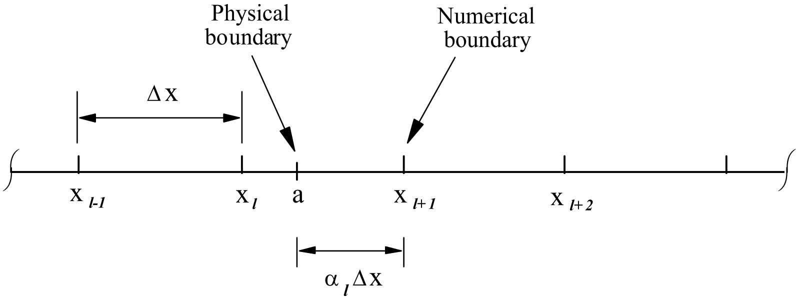


Fig. 2. Left physical and numerical boundaries for a one-dimensional irregular domain $(\Omega \cup \partial\Omega) = [a, b]$ overlaid with a uniform N-point grid.

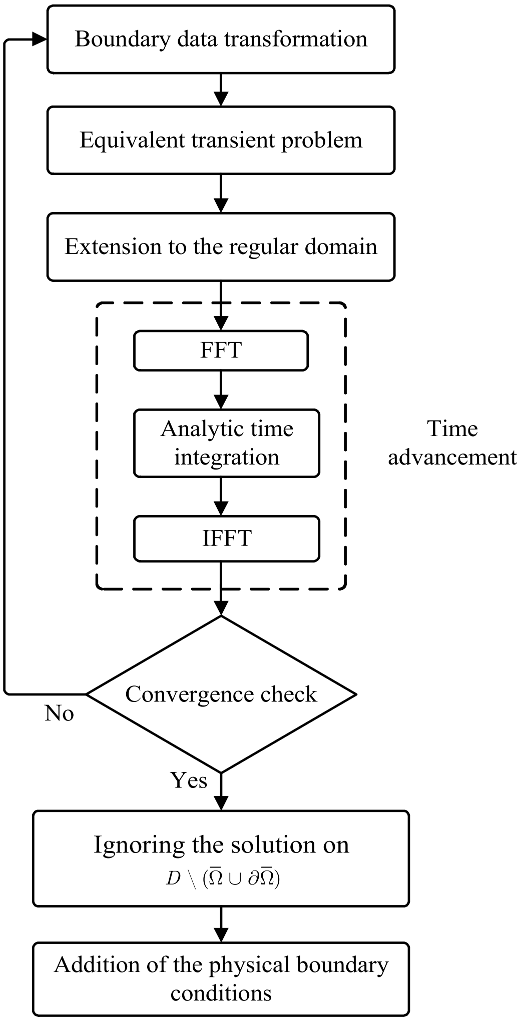

Boundary data transformation
Equivalent transient problem
Extension to the regular domain
FFT
Analytic time integration
IFFT
Time advancement
Convergence check
No
Yes
Ignoring the solution on $D \setminus (\bar{\Omega} \cup \partial\bar{\Omega})$
Addition of the physical boundary conditions

Fig. 3. The main steps of the solution algorithm.

$$\left.\frac{\partial^2 v}{\partial x^2}\right|_{l+1} = F(x_{l+1}),$$

therefore, by a second order finite differencing,

$$\left.\frac{\partial}{\partial x}\left(\frac{\partial v}{\partial x}\right)\right|_{l+1} = 2\frac{\alpha_l v_{l+2} - (1+\alpha_l)v_{l+1} + v_a}{\alpha_l(1+\alpha_l)\Delta x^2}.$$

After some algebra, for the left numerical boundary, we have:

$$v_{l+1} = \frac{-\frac{1}{2}\alpha_l(1+\alpha_l)\Delta x^2 F(x_{l+1}) + v_a + \alpha_l v_{l+2}}{1+\alpha_l}. \tag{6}$$

Similarly, an explicit relation for the right numerical boundary v_r is obtainable. In this way, the problem (5) with Dirichlet boundary conditions at the physical boundaries, is converted to a problem with Dirichlet boundary conditions at the numerical boundaries as

$$\frac{\partial^2 v}{\partial x^2} = F(x), \quad x \in \bar{\Omega}, \tag{7}$$

$$v(x_{l+1}) = v_{l+1},$$
$$v(x_r) = v_r. \tag{8}$$

The following remarks should be noted about this step:

(i) Obviously, it is a second order interpolation between the physical boundary, the numerical boundary and the first grid point inside the solution domain. As it will be emphasized in the numerical experiments section, presence of such an interpolation, plays a smoothing role which can prevent triggering the high frequency modes (in the Fourier space), especially for the problems with discontinuities at the boundaries.
(ii) Detail discussions about the effects of accuracy of the boundary condition implementation on the accuracy of the solution of the Poisson's equation can be found in many references, for example [13]. However, it should be noted that in the present method, as it is more explained in Sections 2.1.4 and 3.1, influences of the boundary data variation, during each timestep, are more dominant than the errors of the interpolation method.
(iii) However, this is just a subsidiary step and always can be modified without any essential changes in the whole of the solution algorithm.

2.1.2. Equivalent transient diffusion problem

After the boundary data transformation, the new Dirichlet boundary conditions are defined at some regular grid points (that is, $\partial\overline{\Omega} = \{x_{l+1}, x_r\}$). However, since the $\partial\overline{\Omega}$ is an internal region of the regular domain D, direct implementation of these boundary conditions to an elliptic equation (where its solution is dependent on all the boundary values), needs some special treatments. During the last years, to overcome this difficulty, the boundary element or spectral element methods were suggested and used [1,8,18,2].

As an alternative approach, replacement of the elliptic equation by an equivalent parabolic one is suggested in this article. Several previous experiences have shown that in the problems that an initial guess, close to the final solution, is at hand (e.g. solution of the unsteady Navier–Stokes equations, where usually a good initial guess is available from the last timestep), this strategy usually yields an efficient solution method.

Therefore, the equivalent parabolic problem is defined as

$$\begin{aligned} &\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2} - F(x), \quad x \in (\overline{\Omega} \cup \partial\overline{\Omega}), \\ &v(x,0) = \overline{v}_0(x), \quad x \in \overline{\Omega}, \end{aligned} \tag{9}$$

$$\begin{aligned} &v(x_{l+1,}, t) = v_l, \\ &v(x_r, t) = v_r. \end{aligned} \tag{10}$$

To unifying our notations, the initial condition $\overline{v}_0(x)$ (defined in the interior $\overline{\Omega}$) together with the boundary conditions $\{v_l, v_r\}$ will be referred to as $v_0(x)$. Any steady solution of the problem (9) (that is, $\frac{\partial v}{\partial t} = 0$), which also satisfies the boundary conditions (10), can be considered as an approximation to the solution of the problem (7).

It should be noted that decision on the solution method for the problem (9) has not been taken up to now, and the problem can be solved by any numerical method. On the other hand, since the unsteady diffusion problems (with periodic boundary conditions) have exact solutions in the Fourier space, attempts to find a spectral solution for this problem seems to be justified. However, transformation of the problem (9) and its boundary conditions (10) into the Fourier space need appropriate extension of them to the regular domain D.

2.1.3. Extension to the regular domain

In the following of the methods suggested in [2,4,6], the forcing function $F(x)$ and the initial condition $v_0(x)$ should be extended to the regular domain D periodically and smoothly, using an appropriate window function. Therefore, if we call the extended forcing function and initial condition as $f(x)$ and $u_0(x)$, respectively, we have:

$$f(x) = \begin{cases} F(x) & \text{on } (\overline{\Omega} \cup \partial\overline{\Omega}) \\ 0 & \text{otherwise} \end{cases}; \quad u_0(x) = \begin{cases} v_0(x) & \text{on } (\overline{\Omega} \cup \partial\overline{\Omega}) \\ 0 & \text{otherwise} \end{cases}.$$

To our knowledge, use of the window functions in spectral solution of the Helmholtz and Poisson's equations have been proposed in [11] and then [20] for the first times. In these works, windowing of the forcing function is used in one dimension (as a part of a domain decomposition algorithm). After about one decade, a more general discussion and classification was appeared in [3] and then some more practical formulations and implementations were suggested in [2,4,6], which the main difference between them is the method of definition of the window function.

In the language of Boyd [3], the problem (5) is a so-called "third kind" problem, which means that the forcing function $F(x)$ (and also the initial condition $v_0(x)$), are not defined on $D \setminus (\Omega \cup \partial\Omega)$. For the originally smooth problems (on $\Omega \cup \partial\Omega$), where the exponential rates of convergence are potentially achievable, smoothness of the extended part plays the key role in preserving the spectral rate of convergence of the final solution. According to the works of Boyd [3,2], Bueno-Orovio [4], Bueno-Orovio et al. [6] and Bueno-Orovio and Perez-Garcia [5], although a general non-singular extension could not be found for these "third kind" problems, the machine accuracies are possible; and for some particular problems, appropriate extension methods can be developed to preserve the exponential convergence rates.

In the present unsteady formulation, as it will be emphasized in the next section, accuracy of the boundary condition implementation and the rate of convergence, depend on the smoothness of the window function, in addition to minimizing the changes of the boundary values during timestepping. On the other hand, since the forcing function $f(x)$ is defined on Ω, rate of change of the boundary values (on $\partial\Omega$) and their extensions (on $D \setminus (\Omega \cup \partial\Omega)$) depend on the spatial curvature:

$$\frac{\partial u}{\partial t} \backsim \frac{\partial^2 u}{\partial x^2} \quad \text{on} \quad D \setminus \Omega. \tag{11}$$

So, we arrive at the idea of linear extension of the initial condition (with $\frac{\partial^2 u}{\partial x^2} \approx 0$) in vicinity of the boundaries, to achieving the minimum rates of change of the boundary values. In this way, smooth windowing of the forcing function and initial condition is done via multiplication of the linearly extended functions by an appropriate shifted error function.

The error function is shown in Fig. 4. In this figure, the outer smooth margin $(\Delta_0 + \Delta_1)$ is produced by a Δ_0 shifting in a standard error function with a suitable rising distance Δ_1.

In Fig. 5, the initial condition $v_0(x)$ is extended linearly to the regular domain D, with a slope which is obtained from its adjacent interior points in $\overline{\Omega}$, and then, it is multiplied by the error function of Fig. 4 to construction of the extended initial condition $u_0(x)$.

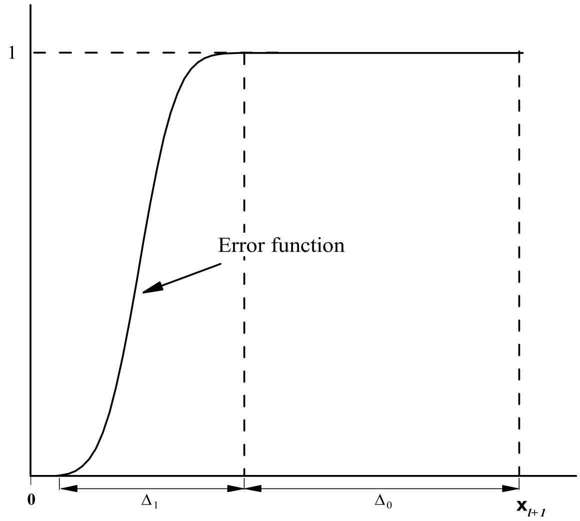


Fig. 4. The smooth outer margin for the left boundary. In addition to the decaying rate of the Fourier modes, it affects the acceptable timestep sizes and the final accuracy.

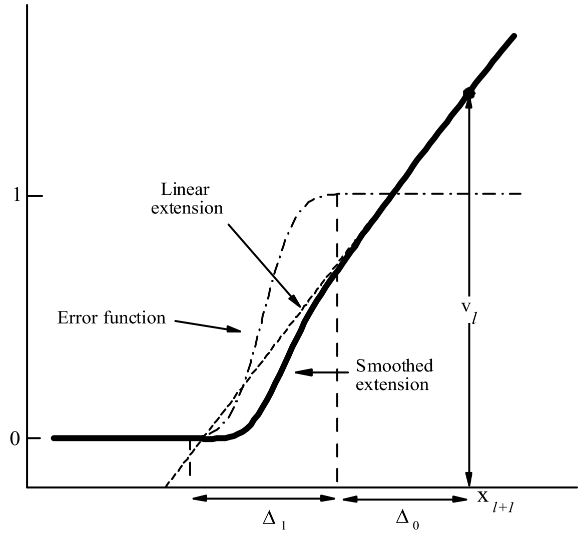


Fig. 5. Smooth extension of the forcing function and initial condition are obtained from multiplication of a linear extension (with a slope which is computed from the internal adjacent points) by a shifted error function.

Our numerical experiments showed that for non-coinciding boundaries (that is, a and b are not coincided with the regular grid), and for initial conditions with discontinuities at the boundaries (that is, when there are significant jumps between the Dirichlet boundary conditions and initial condition at the boundaries), some instabilities can be occurred as the solution proceeds. This is mainly because of the fast changes in the slope of the solution, near the boundaries, from one timestep to the next one. However, our numerical experiments showed that these instabilities are not serious (for originally smooth problems), and always will disappear with development of the solution. On the other hand, to have s safe solution procedure, one can use just a simple error function (without the linear extension) at first, and then switch on the linear extension process after sufficiently development of the solution (and disappearing of the discontinuities).

Moreover, while the slope of the linear extension is obtained from the internal points (adjacent to the boundaries), the exponential rates of convergence are obtainable for simple cases, and for more complex geometries, achieving the accuracies close to the machine precision are possible.

As a final remark, it is worth mentioning that the linear extension process is designed to achieving more accurate solutions and higher convergence rates. But a simple error function can be used satisfactorily with the advantage of simplicity in the solution procedure. Therefore, all of the following two-dimensional numerical experiments are performed using this simple windowing procedure.

2.1.4. Time advancement

Now, the problem is changed to

$$\begin{aligned}&\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2}-f(x),\quad x\in D,\\&u(x,t=0)=u_0(x),\quad x\in D,\\&u(0,t)=u(2\pi,t),\quad \frac{\partial}{\partial x}u(0,t)=\frac{\partial}{\partial x}u(2\pi,t),\ldots\end{aligned}\tag{12}$$

Due to periodicity of the forcing function $f(x)$ and the initial condition $u_0(x)$, this problem can be transformed to the Fourier space, where it has an exact solution

$$\hat{u}_k(t)=(\hat{u}_k)_0\exp(-k^2t)+[\exp(-k^2t)-1]\frac{\hat{f}_k}{k^2}.\tag{13}$$

In this relation $\hat{f}_k$ and $(\hat{u}_k)_0$ are the Fourier transforms of $f(x)$ and $u_0(x)$, respectively, and k is the wavenumber.

Since the exact solution (13) is available, time integration can be performed directly without need of any numerical integration method. Consequently, there is not any stability limitation on the time integration sizes. On the other hand, as it will be shown later, such time integration destroys the boundary conditions (8) and therefore, the time integration sizes are restricted by the desired accuracy in implementation of the boundary conditions. Hereafter, to reflect this restriction and also to emphasize the iterative nature of the solution algorithm, integration time in Eq. (13) will be referred to as Δt.

Timestepping. Time evolution of the linear diffusion problems with periodic boundary conditions in the Fourier space, expressed by Eq. (13), has some well-known features including:

(i) All the modes are uncoupled and the time evolution rate of each mode with $k\neq 0$ is solely proportional to its initial condition and the forcing function component in the same mode; and is exponentially proportional to the second power of its wavenumber.

(ii) Time evolution of the zeroth mode (that is, for $k=0$), can be determined uniquely from the zeroth mode of the forcing function

$$\frac{d\hat{u}_0}{dt}=-\hat{f}_0.$$

On the other words, distribution of $u(x,t)$ on D has an average which evolves (in time) with a rate, which is proportional to the average of the forcing function.

(iii) Since Eq. (13) is written in the Fourier space, the diffusion process governed by this equation is inherently periodic in the physical space (that is, $u(x,t)$ calculated from $\hat{u}_k(t)$ will remain periodic on the domain D during time evolution).

As a consequence of the above issues, the following circumstances will be observed in time integration of the problem (12) using Eq. (13):

(1) The Dirichlet boundary conditions v_l and v_r, which are implemented via the initial condition $u_0(x)$, will change during the time integration. It yields lack of accuracy in implementation of the boundary conditions. In fact, one can see that for say, $\Delta t\to\infty$, just the zeroth mode and the effect of the forcing function will remain and the boundary conditions

will not have any influence on the solution. Therefore, an accurate solution needs minimum changes of the boundary values for each timestep.It can easily be shown that the relative change in the boundary conditions during the timestep Δt, (for sufficiently small Δt), can be written as

$$\frac{(\hat{u}_{k_\Delta})_0 - \hat{u}_{k_\Delta}}{(\hat{u}_{k_\Delta})_0} = \mathcal{O}[k_\Delta^2 \Delta t] = \mathcal{O}\left[\left(\frac{2\pi}{\Delta}\right)^2 \Delta t\right], \tag{14}$$

where $k_\Delta = \frac{2\pi}{\Delta}$ is a typical wavenumber, which is dependent on the involved wavenumbers in the vicinity of the boundaries. It will be shown formally, in the numerical experiments section, that the value of Δ depends on the values of $\varDelta_0$ and $\varDelta_1$, as well as the sharpness of the solution field near the boundaries. According to Eq. (14), changes in the boundary conditions, during the time advancement Δt, is first order with respect to the timestep, and is second order with respect to the wavenumbers (which are involved in the boundaries neighborhood). Therefore, a strategy for minimizing variation of the boundary values, which is followed in this article, is to minimizing the involved wavenumbers near the boundaries. This strategy leads to the linear extension as described in Section 2.1.3.

(2) Despite its periodicity, the solution does not remain necessarily zero outside of the $(\overline{\Omega} \cup \partial\overline{\Omega})$ during the time advancement, especially near the boundaries of the regular domain D.

These issues are consequences of treatment of the numerical boundaries as the internal points and emphasizing on using the Fourier modes in construction of the solution space. In summary, in spite of using the exact solution in the time integration, the timestep sizes are limited by the desired values of errors. However, this limitation is not so strict in comparison with the stability limitations of the numerical time integration methods (as it will be shown in the Section 3).

On the other hand, our numerical experiments showed that after performing a number of the above iteration, solution shows a kind of saturation, in which, without achieving the steady state solution, major part of the solution is remained essentially unchanged during each timestep. A more comprehensive discussion about this phenomenon (which we called it the saturation state) and its effects on the solution methodology is offered in the Section 3.1.

2.2. Extension to the two-dimensional problems

In extension of the method to the two-dimensional problems, as it is shown in Fig. 1, we assume that the physical boundary $\partial\Omega$ is defined via an arbitrary curve $\mathbf{S}(x,y)$, where the boundary conditions are determined on it. In the method which is followed here, presence of an analytic relation for $\mathbf{S}(x,y)$ is not necessarily needed.

Solution of Eq. (1) with the boundary conditions (2) is sought on the irregular domain $(\Omega \cup \partial\Omega)$ by extension of the problem to the regular domain D, which is overlaid by a uniform grid (x_i, y_j). Obviously, all the five steps in the Section 2.1 could be followed, and among them, the most remarkable difference between the one- and two-dimensional problems is in the boundary data transformation process, which will be explained in details in the subsequent section.

2.2.1. Transformation of the boundary data

For boundary data transformation, first of all, it is needed to have a precise definition for the numerical boundary points. Among many possible definition methods (for example, see Johansen and Colella [12] for a fairly extensive discussion about a variety of related practical issues, or Jomaa and Macaskill [13] for a categorization of the numerical boundary points), a method is followed here in similarity to analytical definition of the boundaries and discontinuities (for example, see [14] or [17]).

Consider some d-dimensional balls C_{ij} (circles for two-dimensional problems), centered at (x_i, y_j) and with radius r_{ij}, where $r_{ij} = \min(\Delta x, \Delta y)$. Now a grid point (x_i, y_j) is a numerical boundary point, if and only if

(i) $(x_i, y_j) \in (\Omega \cup \partial\Omega)$.
(ii) C_{ij} contains at least one point of $D \setminus (\Omega \cup \partial\Omega)$.

Just like the one-dimensional problems, after determination of the numerical boundary points, the numerical boundary conditions should be set. Among the different possible approaches, a combination of the methods proposed in [18,19] is followed in this article (see Fig. 6).

By definition of a local curvilinear coordinate (ξ, η), as it is shown in Fig. 7, and transformation of Eq. (1) to this coordinate system (in the vicinity of the physical boundary $\mathbf{S}(x,y)$), we have:

$$\frac{\partial^2 v}{\partial \eta^2} + \kappa \frac{\partial v}{\partial \eta} + \frac{\partial^2 v}{\partial \xi^2} = F(\xi, \eta), \tag{15}$$

where $\kappa = \frac{1}{R}$ is the inverse of the local curvature radius of the physical boundary $\mathbf{S}(x,y)$.

Since a second order differencing is used at each timestep, a local grid with three nodes in depth is considered, similar to Russell and Wang [18]. On the other hand, the present method is different from Russell and Wang [18] in the way that, just the point ‘nb’ (that is, the numerical boundary), is on the regular grid. The other points should be obtained by interpolation on the physical boundary data and the initial guess $v(x,y)$.

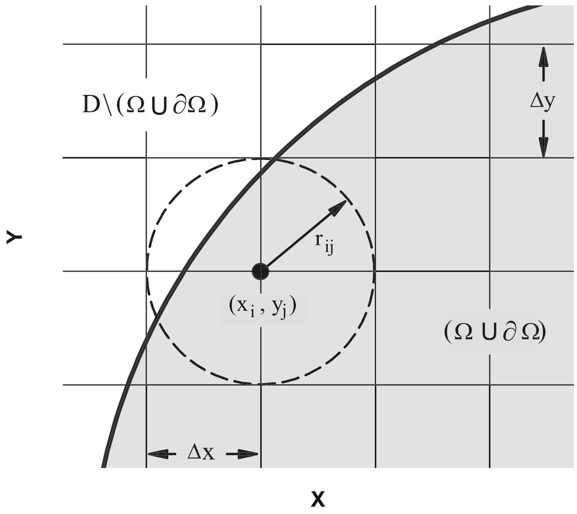


Fig. 6. Numerical boundary for a two-dimensional problem and uniform grid $\Delta x = \Delta y$. The point (x_i, y_j) is a numerical boundary point, since the circle C_{ij} contains some points in $D \setminus (\Omega \cup \partial\Omega)$.

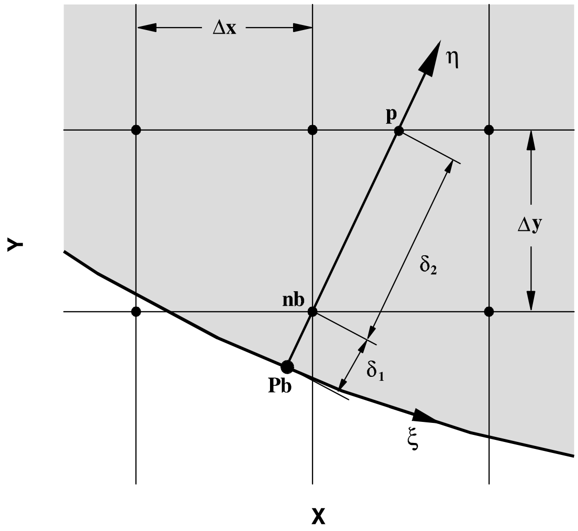


Fig. 7. The local curvilinear coordinate system (ξ, η), crossing the physical boundary and passing the numerical boundary point 'nb'.

Since $\left(\frac{\partial^2 v}{\partial \xi^2}\right)_{\text{pb}}$ and v_{p} are determined, from the boundary conditions and interpolation, discretization of Eq. (15) in the η direction (similar to the one-dimensional problems), yields a relation for the numerical boundary conditions as

$$v_{\text{nb}} = \left(\frac{\delta_1^2}{\kappa\delta_1 - 1}\right)\left[F_{\text{pb}} - \left(\frac{\partial^2 v}{\partial \xi^2}\right)_{\text{pb}} + \frac{v_{\text{pb}}[\kappa\delta_1(\delta_1 + \delta_2) - \delta_2] - \delta_1 v_{\text{p}}}{\delta_1^2(\delta_1 + \delta_2)}\right]. \tag{16}$$

Here, the subscripts 'pb' and 'nb' denote the physical boundary and the numerical boundary, respectively. It should be noted that since Eq. (15), which is a Poisson's equation in the orthogonal curvilinear coordinate, is used directly in derivation of Eq. (16), this transformation procedure is suitable just for the Poisson's and Laplace equations and for other kinds of elliptic equations should be adapted appropriately.

After boundary data transformation, the other steps of the algorithm (presented in Fig. 3) could be followed. In this way, by definition of an unsteady problem on the regular domain D and progressing the unsteady solution until achievement of the steady state solution, an approximation for the solution of the problem (1) will be obtained.

3. Numerical experiments

Our numerical experiments on the above algorithm are offered in this section. Firstly, the method is applied to a one-dimensional problem in order to assessing the errors and showing its superiority (from the rate of convergence viewpoint), in comparison with the usual relaxation methods. Moreover, the one-dimensional problems provide a good opportunity for explanation of the saturation phenomenon, both in the physical and spectral spaces, and its effects on the convergence rate and the final solution accuracy.

In the two-dimensional numerical experiments, it is shown that there are no essential differences in the error distributions of the one- and two-dimensional problems, and in either case the maximum errors occur inside the solution domain, not at the boundaries. In particular, this is similar to the quadratic boundary conditions implementation in [13]; which can be interpreted as suitability of the second order interpolation in boundary data transformation (at least for the converged final solution).

3.1. One-dimensional problems

As a one-dimensional test case, Eq. (5) is considered with $F(x) = -1$, and the boundary conditions $v(a) = v_a = 1.5$ and $v(b) = v_b = 1.5$, on the interval $(\Omega \cup \partial\Omega) = [a, b] = [0.613, 5.71]$. This problem has an exact solution

$$v(x) = -\frac{x^2}{2} + \frac{(a+b)}{2}x + \frac{(v_a + v_b)}{a-b}(x-a) - \frac{ab}{2} + v_a. \tag{17}$$

By extension of this problem to the regular domain $D = [0, 2\pi]$, and definition of a uniform 1024 point grid on it, we have $\alpha_l = \alpha_r = 0.2$. To observe the method capability in dealing with the discontinuities at the boundaries and the rate of convergence (in presence of discontinuities), the initial condition is considered as $v(x, 0) = \bar{v}(x) = \cos(x)$ on the interval $\Omega =]0.613, 5.71[$, which is too far from the final solution. Therefore, we are faced with discontinuities at both boundaries. Fig. 8 shows the result of boundary data transformation on the initial condition, for the left numerical boundary, which is smoothly extended to the regular domain D using a shifted error function. The following remarks should be noticed about this figure:

(i) Obviously, the boundary data transformation process has a smoothing effect in the vicinity of the numerical boundary. However, due to the remarkable difference between the physical boundary values and the initial guess, a jump is created. Although such jumps trigger the high frequency Fourier modes during the FFT process, our experiments showed that the solution can be remained alias-free, for rational timestep sizes. Moreover, these jumps enforced use of the error function, instead of the linear extension in the extension process, for the first timesteps (as described in Section 2.1.3). However, by approaching the steady state solution, these jumps will die gradually.

(ii) After determining the numerical boundary condition at the point x_{l+1}, smooth extension of the initial condition is done by the use of an error function, such that it approaches to the numerical boundary condition at the numerical boundary point.

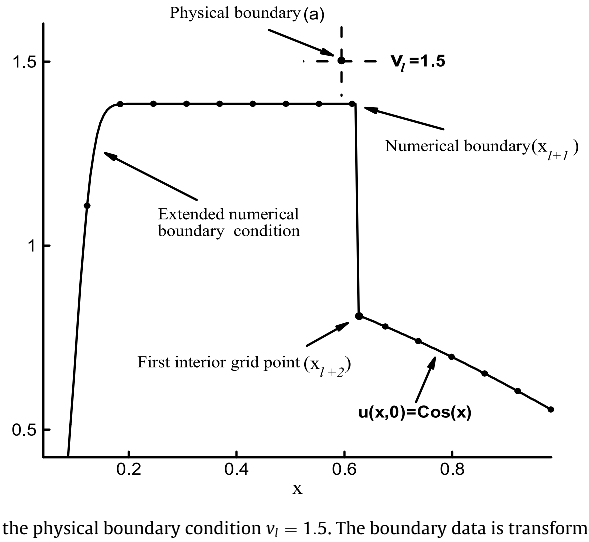


Fig. 8. Boundary data transformation for the physical boundary condition $v_l = 1.5$. The boundary data is transformed from the physical boundary point a to the numerical boundary point x_{l+1}.

With each selected timestep Δt, after some iterations, the solution leads to a situation which we call it the saturation state. For a saturated solution, without achieving the steady state solution, major part of the solution field remains unchanged during each timestep.

Although from viewpoint of many end users, only the final solution is in practical importance, in order to design an accurate and fast algorithm a clear understanding about the saturation phenomenon seems to be essential. Therefore, to explain the phenomenon, Fig. 9 illustrates changes in a saturated solution during a timestep $\Delta t = 0.5$. The curve symboled by '$\circ$' is the solution before time advancement, where the numerical boundary conditions are implemented and the resulting field is smoothly extended to the regular domain. To clarify the saturation phenomenon, the extension process is intentionally performed via a simple error function. Transforming this field to the Fourier space, time integration and then taking back to the physical space (as described in Section 2.1), resulted in the '∇' symboled curve. Apparently, most of the domain $(\Omega \cup \partial\Omega) = [0.613, 5.71]$, remained almost unchanged, while for the points around the boundaries, fairly large variations can be observed. These curves will be repeated identically in the next iterations, while the timestep Δt is unchanged.

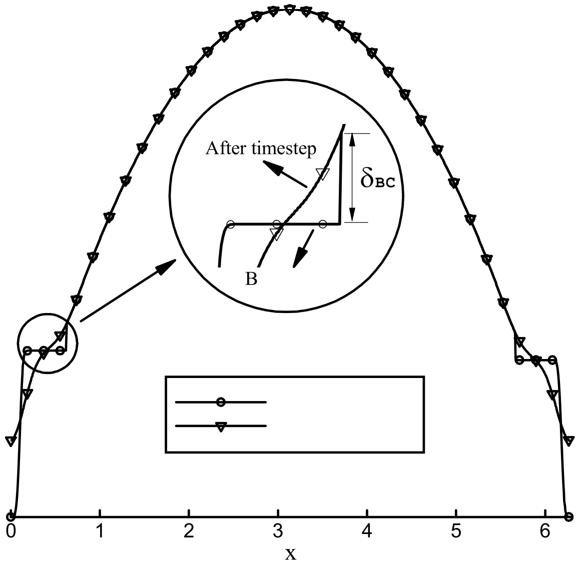
In fact, the saturation state is the final solution for a problem with variable boundary conditions with amount of $\delta_{\mathrm{BC}} \sim \left(\frac{2\pi}{\Delta}\right)^2 \Delta t$. Conversely, a solution can be called saturated, when the effect of these variable boundary conditions and the forcing function are completely diffused all over the solution field. With this respect, a converged solution can be defined as a saturated solution in which the boundary condition variation meets a pre-assumed criteria (that is, $\delta_{\mathrm{BC}} \leqslant \epsilon_{\mathrm{T}}$, where the ϵ_{T} is a desired threshold).

Some other aspects of the saturation phenomenon can be observed in the spectral space. Fig. 10 shows spectrum of the $\|\hat{v}_k\|$ and $\|\hat{f}_k\|$ (where $\|\cdot\|$ is the $\mathscr{L}_2$ norm), for a saturated solution before and after timestep $\Delta t = 0.001$ (which resulted in $\delta_{\mathrm{BC}} \sim (10^{-3})$). Since the solution is saturated, these spectra will be repeated identically in the next iterations. Note that the spectrum of the forcing function is scaled by $\frac{1}{k^2}$, in order to make easier comparison between the saturated solution and the forcing function spectra.

According to Eq. (13), it is expected to have $\hat{v}_k \to \frac{\hat{f}_k}{k^2}$ for a sufficiently long time integration (that is, $\sum(\Delta t) \approx t \to \infty$). In the other words, all the initial modes experience exponential decaying until settling down on the forcing function, where in this situation, all the Dirichlet boundary conditions are missed. In contrast, implementation of the boundary conditions causes deviation of the solution spectrum from $\frac{\hat{f}_k}{k^2}$.

In Fig. 10, since the overall time advancement has been sufficient to achieving a saturated solution, a good coincidence between the solution spectrum and the forcing function spectrum can be observed for high wavenumber modes (after the point R). On the other hand, the time integration has not had any influences on the very small wavenumber modes, as can be observed up to the point L. Meanwhile, the gap between L and R is a transient region which is under the influence of Dirichlet boundary conditions (and therefore its spectrum is different from $\frac{\hat{f}_k}{k^2}$), and also its wavenumbers are large enough to change due to the exponential decaying.

Having both a suitable convergence rate and an acceptable final accuracy, needs proper choosing of $(\varDelta_0 + \varDelta_1)$ and Δt simultaneously. However, in the practical applications the smooth margin $(\varDelta_0 + \varDelta_1)$ can be chosen first (more or less intuitively, and also based on the uniform grid resolution), and then during the solution progress, suitable convergence rates are affordable by appropriate choosing of the timestep sizes.



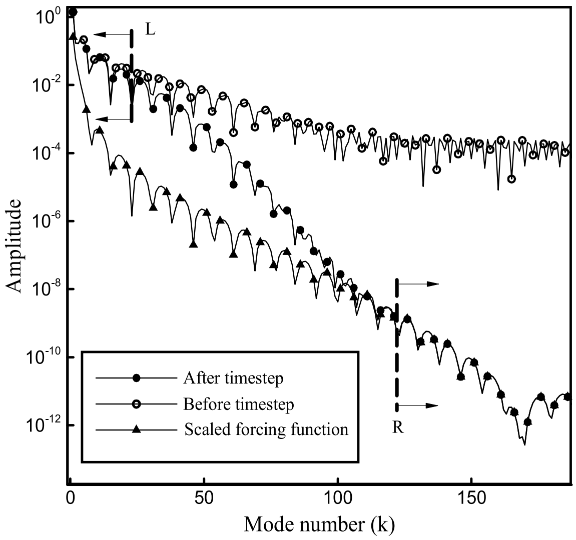


Fig. 10. Saturated solution in the Fourier space before and after the time integration. For the sake of clarity, the forcing function is scaled by $\frac{1}{k^2}$.

Various strategies can be followed for selection of the timestep sizes. One can simply choose a very small constant timestep (say, $\Delta t \backsim \frac{1}{h}$, where h is the mesh size), and continue until achieving the desired solution. It is not difficult to see that this strategy results in a very slow rate of convergence, which is somehow equivalent to the classical Gauss–Seidel (or even the Jacobi) method. Therefore, in order to increasing the rates of convergence, variable timesteps are essential.

A simple but still effective scheme, which is used in the present work, is to choosing an initial large timestep (which can be chosen based on the ratio of lengths of Ω and D), and then adapting the timesteps according to convergence of the Fourier modes from the smaller to the larger wavenumbers, respectively. In this way, saturation of the Fourier modes is checked one by one from the zeroth mode. At each step, the initial timestep will be divided by k^2 when the mode $(k-1)$ is saturated. The factor $\frac{1}{k^2}$ is mainly chosen because of presence of the $\exp(-k^2t)$ term in the time evolution equation (13). All the following numerical experiments have been performed using this strategy.

Fig. 11 shows convergence histories for the present method and two other classical relaxation methods, that is, the Jacobi and the Gauss–Seidel methods. For the spectral embedded boundary solution, the smooth outer margins are chosen as $\Delta_0 = 0.43$ and $\Delta_1 = 0.26$ and the initial timestep has been $\Delta t_0 = 0.5$. On the convergence curve of the present method, the spikes (such as the points A and B) are noticeable, which have been generated due to the timestep changes. Except for the point A, where the timestep changing has been done without achieving the saturation state (to show the effect of

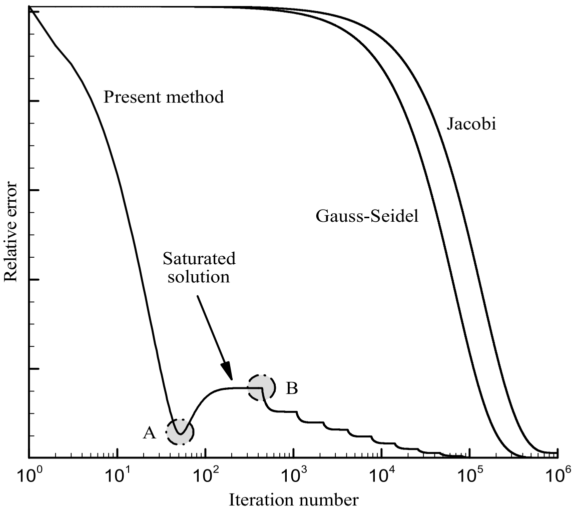


Fig. 11. Convergence history of the present method in comparison with two other classical relaxation methods. Except for the point A, the other timestep changes have been done in the saturation states.

timestep on the convergence rate); all the other timestep changes have been done after saturation (which are distinct by reductions in the curve slope).

In spite of using the above simple timestep adaptation algorithm, an obvious superiority of the method (from the convergence rate viewpoint), is observable. However, still higher convergence rates are achievable by using more complicated timestep adaptation algorithms. For the present numerical experiment, the solution procedure has been terminated at $\Delta t_{\text{final}} = 0.0013$ with the maximum error $\epsilon_{\max} = 8.76 \times 10^{-5}$.

For such simple one-dimensional problem, due to presence of the exact solution (that is, Eq. (17)), distribution of the relative error on the interval $(\overline{\Omega} \cup \partial\overline{\Omega})$ is attainable (see Fig. 12). Apparently, the error distribution has its maximum in the middle of the interval, while the minimum errors are generated at the boundaries. Particularly, this situation is the same as quadratic boundary condition implementation in [13], and therefore can be interpreted as suitability of the second order interpolation for the boundary data transformation.

As it is pointed out earlier, accuracy of the boundary condition implementation (in the final solution), depends on the final timestep sizes. This dependency is illustrated in Fig. 13, where the maximum error values, that is, the errors in the center of the solution interval, is plotted versus different values of Δt_{final}. As one can see, the maximum errors are $\mathcal{O}(\exp(\mathscr{A}\Delta t_{\text{final}}))$, for

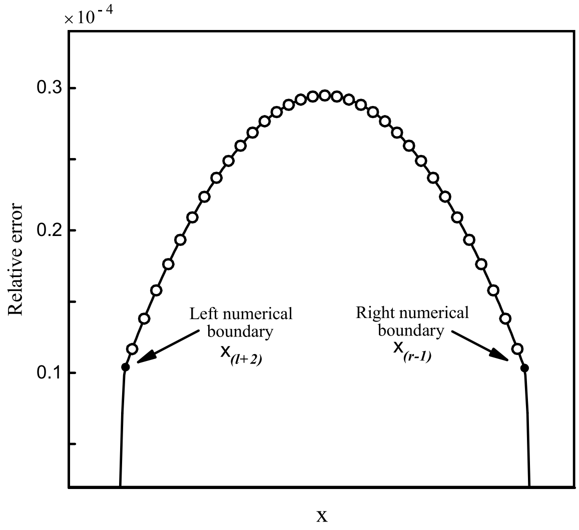


Fig. 12. Relative error on the regular domain D. Outside the embedded numerical domain $(\overline{\Omega} \cup \partial\overline{\Omega})$, the error is essentially zero.

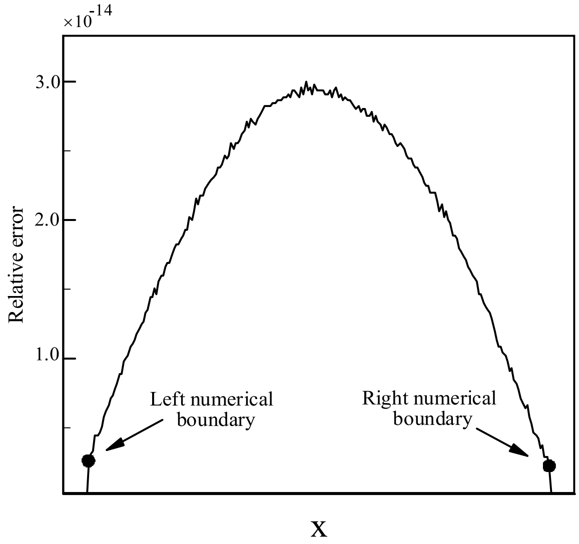


Fig. 13. Error distribution for a more accurate solution of the problem of Fig. 12, with $\Delta t_{\text{final}} = 0.0001$. More accurate solutions (close to the machine accuracy) are attainable with more iterations.

some enough small values of Δt_{final}. The coefficient $\mathscr{A}$ is a constant which is dependent on the $(\Delta_0 + \Delta_1)$, as well as sharpness of the solution in the vicinity of the boundaries. For the above numerical experiments, it has been obtained as $\mathscr{A} \sim \mathcal{O}(5)$ (see Fig. 14).

It is worth mentioning that, like the other relaxation methods, the present algorithm needs more iterations to achieving more accuracy. However, to show capability of the method in achieving arbitrary accuracies, error distribution for a more converged solution (with $\Delta t_{\text{final}} = 0.0001$), is presented in Fig. 13. As it can be seen, the errors are fairly acceptable for many applications.

3.2. Two-dimensional problems

As the first two-dimensional numerical experiment, the method is applied on a unit circle C, centered at $(x_c, y_c) = (\pi, \pi)$, and placed in a $(2\pi \times 2\pi)$ box as the regular domain D. The physical boundary points are defined on $N_{\text{PB}} = 4096$ points, and the cubic spline interpolation is used wherever the midpoint data are needed. The physical and the numerical boundaries for a (256×256) uniform grid are shown in Fig. 15.

At first, to see the effect of boundary data transformation on smoothness of the resulting numerical boundary conditions, a constant boundary condition $v(\mathbf{S}(x, y)) = 1$ is transformed to the numerical boundary by the use of Eq. (16). In addition, in

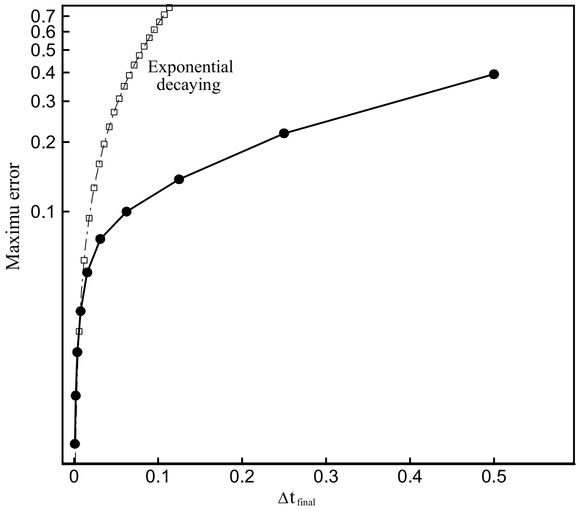


Fig. 14. The maximum error (that is, the error in the middle of the solution domain), versus final timestep sizes, compared with an exponential function.

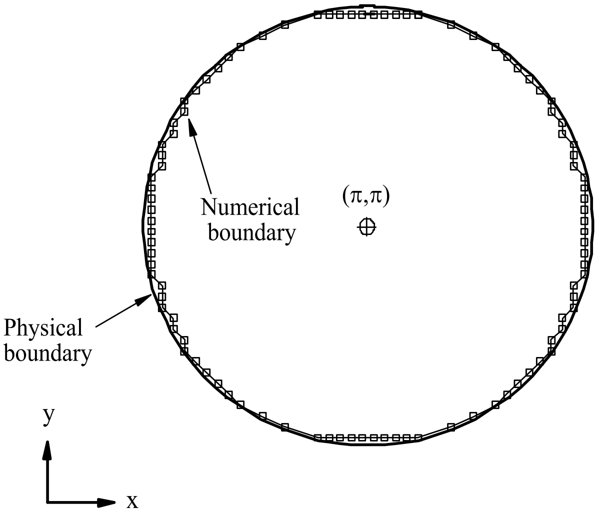


Fig. 15. Physical and numerical boundaries for a unit circle and a uniform (256 × 256) grid.

the sake of clarity, the initial condition is assumed to be zero (that is, $v_0(\mathbf{S}(x,y)) = 0$). In Fig. 16, the resulting numerical boundary conditions are shown for some different resolutions of the regular grids.

Apparently, contrary to the physical boundary conditions, the values of numerical boundary conditions are variable along the numerical boundary nodes. Mainly, this is a consequence of variation of δ_1 in Eq. (16). Therefore, increasing the resolution of the regular grid, which decreases both the values of δ_1 and its variations, reduces variation of the numerical Dirichlet boundary conditions and also approach them to the physical boundary conditions (in this case, a constant value $v(\mathbf{S}(x,y)) = 1$), as can be seen in Fig. 16.

It can be concluded that some discontinuities (at the numerical boundaries), can appear as a result of the boundary data transformation. However, as it will be pointed out in the following discussions, the solution accuracy and the rate of convergence are not influenced seriously by these discontinuities.

After the boundary data transformation and definition of an unsteady problem, the initial condition together with the forcing function should be extended to the domain D, smoothly and periodically. In this context, a crucial question is the effects of the above discontinuities (at the numerical boundary points), on the FFT process and then on the global rate of convergence. To probe this, similar to the method which was proposed in [2] and then followed in [4], the scatter diagrams of the absolute values of Fourier coefficients are shown in Fig. 17, for a saturated solution before and after timestep Δt.

In the left plate, which is the solution spectrum before time advancement, due to the presence of discontinuities at the numerical boundaries, the exponential convergence can be seen just for the low wavenumber modes, while for the higher wavenumbers, a fairly large deviation from the exponential decaying is observable. Right side of the figure, shows the same field after the timestep $\Delta t = 0.01$. Obviously, as a result of exponential decaying in time, except for a number of modes (which are under the influence of the extended forcing function spectrum), an exponential convergence is observable for almost whole of the wavenumbers. However, for the above observations it is worth mentioning that:

(1) In the above experiments (one- and two-dimensional), the discontinuous boundary conditions were selected intentionally to show the ability of the method in dealing with these kinds of problems. However, our experiments have shown that for continuous problems, perfect spectral rates of convergence are affordable.

(2) Fig. 17 is plotted for a fairly coarse regular grid (that is, (64×64)), which as mentioned earlier, causes more discontinuities. However, in the practical problems where the higher resolution grids are in common use, these discontinuities do not produce serious problems (as will be seen later in our following numerical experiments).

In summary, although the exponential rate of convergence can be missed (for discontinuous problems and low-resolution grids), however, the spectral method can work properly, with larger values of errors (in comparison with the spectral accuracy), as it is also notified in [2].

To check the method accuracy in the two-dimensional problems, the forcing function $F(x,y) = -2\cos(x+y)$ together with the physical boundary conditions $v(\mathbf{S}) = \cos(x_s + y_s)$ are considered on the circle C, which has an exact solution $v(x,y) = \cos(x+y)$.

Like as the one-dimensional problems, to observe the rate of convergence and capability of the method in dealing with discontinuities, the initial condition is considered too far from the final solution as $v_0(x,y) = 1$. The initial condition and

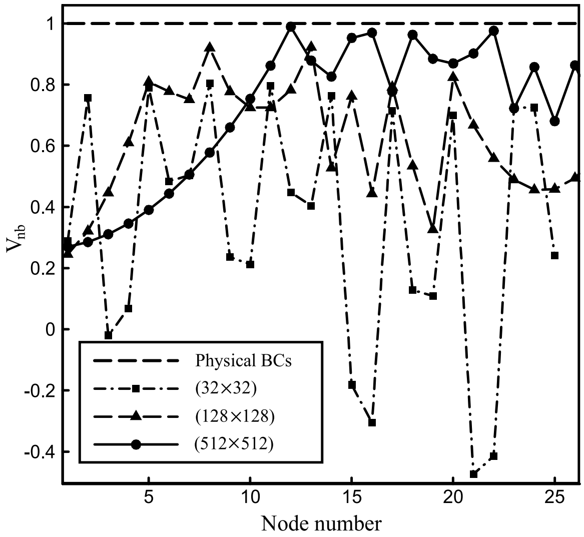


Fig. 16. Transformed Dirichlet numerical boundary conditions for a constant physical boundary condition $v(\mathbf{S}(x,y)) = 1$ and constant initial condition $v_0(x,y) = 0$ on various regular grid resolutions. Since the number of numerical boundary points are different for different grid resolutions, they are shown just for 25 boundary points, corresponding to the coarsest grid (32×32).

transformed numerical boundary conditions are shown in Fig. 18, where the discontinuities at the numerical boundaries are evident.

In the right side of Fig. 19, a three-dimensional visualization of the converged solution is presented on the whole regular domain, including the circle C and its smooth outer margins. Note that the crinkly appearance of the outer margin is due to the coarseness of the grid (as a (64×64) grid is used). To have an assessment of the convergence rate, some intermediate solutions for the $(y = \pi)$ plane is plotted on the left side of the figure.

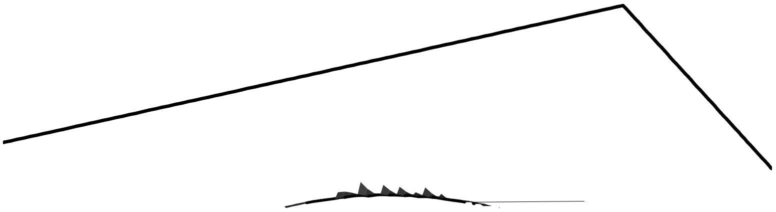


Fig. 19. Left: the boundary points errors. Right: errors for the interior points projected on $(y = x)$ plane, that is, a plane in 45° angle with the x-axis.

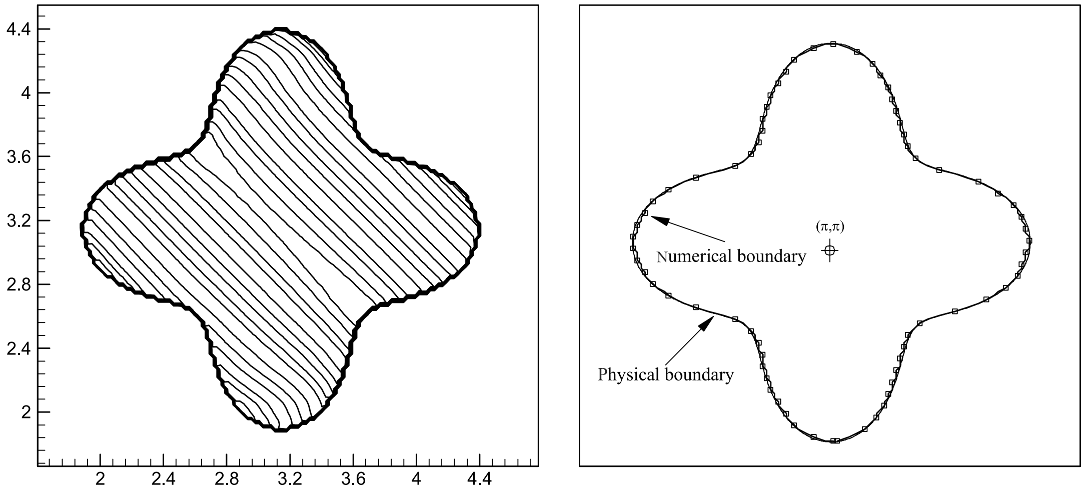


Fig. 21. Right: physical and numerical boundaries for a four-leaf shape body, placed in a regular (256 × 256) grid. To have a clear visualization, just a number of selected boundary points are shown. Left: contour plot for the final, converged solution with the same boundary conditions and the forcing function as in Fig. 19. The smooth outer margin is removed and the physical boundary conditions are added to the solution.

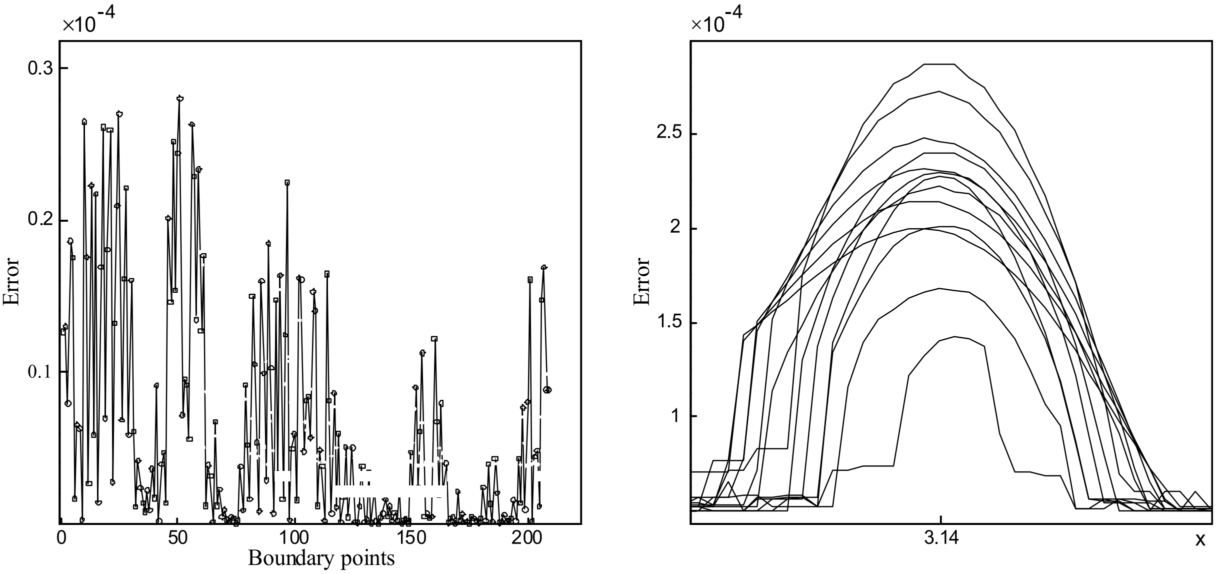


Fig. 22. The error distribution for the four-leaf body of Fig. 21. Left: absolute values of the relative errors for the boundary points. Right: errors in the interior points. In contrary to Fig. 20, the errors are projected on the $(x = \pi)$ plane.

corners on the physical boundaries (except for the 90° angle corners, for which definition of the problem (1) is paradoxical [14]), it is just needed to change the boundary data transformation algorithm to covering the problem.

4. Conclusions and future works

A spectral embedded boundary method for solution of the Poisson's equation with Dirichlet boundary conditions and arbitrary (non-zero, as well as zero) forcing functions has been proposed in this paper. The suggested algorithm, can be seen as a spectral relaxation method which is designed to benefiting from advantages of the FFT (its fastness, suitability for parallel programming, and so forth), in the solution of the linear elliptic equations.

The method has been formulated and implemented on the one- and two-dimensional problems with an emphasis on the problems with discontinuous initial conditions. Numerical boundary (points and conditions) are defined on the regular grid points, as a necessity of accurate implementation of the Dirichlet boundary conditions, and the boundary data transformation is done using a second order interpolation. Suitability of the interpolation method has been shown via some numerical experiments, with arguing that the minimum errors are occurred at the boundaries. Comparison of the method with some other classical relaxation methods has shown higher rates of convergence, which is a result of absence of numerical time integration methods (as the main source of numerical instabilities in the unsteady problems) in the algorithm. Flexibility of the method, in dealing with fairly complex geometries, and its ability in achieving desired accuracies, have been demonstrated via some numerical experiments on a four-leaf shape geometry.

Like some other relaxation methods, the present method can be used as the smoothing step of multigrid methods, where it has the obvious advantage of straightforward implementation of prolongation and restriction (i.e. the zero padding and filtering in the Fourier space). Therefore, development of such a multigrid method is considered as one of our first plans for future works in extension of the method.

As an alternative boundary data transformation method, some immersed boundary concepts (e.g. definition of local forcing functions and implementing them using the Dirac delta function), can be used. In this way, an iterative method (instead of the unsteady diffusion problem), should be followed for setting of these new forcing functions.

On the other hand, extension to the three-dimensional problems and use of the method in solution of the incompressible Navier–Stokes equations in two- and three-dimensions are some other possible future works.

Acknowledgments

The three reviewers and associate editor Prof. Bayliss are acknowledged for careful reading and their helpful comments.

References

[1] L. Badea, P. Daripa, On a Fourier method of embedding domains using an optimal distributed control, J. Numer. Algor. 32 (2–4) (2003) 261–273.

[2] J.P. Boyd, Fourier embedded domain methods: extending a function defined on an irregular region to a rectangle so that the extension is spatially periodic and C^∞, Appl. Math. Comput. 161 (2005).

[3] J.P. Boyd, A comparison of numerical algorithms for Fourier extension of the first, second, and third kinds, J. Comput. Phys. 178 (2002) 118–160.

[4] A. Bueno-Orovio, Fourier embedded domain methods: periodic and C^∞ extension of a function defined on an irregular region to a rectangle via convolution with Gaussian kernels, Appl. Math. Comput. 183 (2006).

[5] A. Bueno-Orovio, V.M. Perez-Garcia, Spectral smoothed boundary methods: the role of external boundary conditions, Numer. Method Part. Differ. Eq. 22 (2006) 435–448.

[6] A. Bueno-Orovio, V.M. Perez-Garcia, F.H. Fenton, Spectral methods for partial differential equations in irregular domains: the spectral smoothed boundary method, SIAM J. Sci. Comput. 28 (3) (2006) 886–900.

[7] L. Collatz, The Numerical Treatment of Differential Equations, Springer, Berlin, 1966.

[8] M. Elghaoui, R. Pasquetti, A spectral embedding method applied to the advection–diffusion equation, J. Comput. Phys. 125 (2) (1996) 464–476.

[9] F.H. Fenton, E.M. Cherry, A. Karma, W. Rappel, Modeling wave propagation in realistic heart geometries using the phase-field method, Chaos 15 (2005) 013502.

[10] B.E. Griffith, C.S. Peskin, On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems, J. Comput. Phys. 208 (1) (2005) 75–105.

[11] M. Israeli, L. Vozovoi, A. Averbuch, Spectral multidomain technique with local Fourier basis, J. Sci. Comput. (82) (1993) 135–193.

[12] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, J. Comput. Phys. 211 (1) (2006) 347–366.

[13] Z. Jomaa, C. Macaskill, The embedded finite difference method for the Poisson equation in a domain with an irregular boundary and Dirichlet boundary conditions, J. Comput. Phys. 202 (2) (2005) 488–506.

[14] P. Knabner, L. Angermann, Numerical Methods for Elliptic and Parabolic Partial Differential Equations, Springer-Verlag, 2003.

[15] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (1) (2005) 157–192.

[16] C.S. Peskin, The immersed boundary method, J. Acta Numer. (2002) 1–39.

[17] W.A. Poor, Differential Geometric Structures, McGraw-Hill Book Company, New York, 1981.

[18] D. Russell, Z.J. Wang, A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, J. Comput. Phys. 191 (1) (2003) 177–205.

[19] B. Sjögreen, N.A. Petersson, A Cartesian embedded boundary method for hyperbolic conservation laws, LLNL Report UCRL-JRNL-226709.

[20] L. Vozovoi, M. Israeli, A. Averbuch, Spectral multidomain technique with local Fourier basis. II: Decomposition into cells, J. Sci. Comput. 9 (3) (1994) 311–326.