An integral-collocation-based fictitious domain technique for solving elliptic problems

N. Mai-Duy*,†, H. See ‡ and T. Tran-Cong†
† Computational Engineering and Science Research Centre (CESRC)
The University of Southern Queensland, Toowoomba, QLD 4350, Australia
‡School of Chemical & Biomolecular Engineering
The University of Sydney, Sydney, NSW 2006, Australia

Submitted to *Commun. Numer. Meth. Engng*, 17-Apr-2007; revised, 5-Jun-2007

Short title: FICTITIOUS DOMAIN TECHNIQUE

*Corresponding author: Telephone +61 7 4631 1324, Fax +61 7 4631 2526, E-mail maiduy@usq.edu.au

SUMMARY

This paper presents a new fictitious-domain technique for numerically solving elliptic second-order partial-differential equations (PDEs) in complex geometries. The proposed technique is based on the use of integral collocation schemes and Chebyshev polynomials. The boundary conditions on the actual boundary are implemented by means of integration constants. The method works for both Dirichlet and Neumann boundary conditions. Several test problems are considered to verify the technique. Numerical results show that the present method yields spectral accuracy for smooth (analytic) problems.

KEY WORDS: fictitious domains; point collocation techniques; integrated Chebyshev polynomials; elliptic problems

1 INTRODUCTION

Solving PDEs in irregularly shaped domains presents a challenge in computational engineering. Well-known techniques used for handling complex geometries include coordinate transformations, domain decompositions, meshless discretizations and fictitious domains. Each technique has some advantages over the others for certain classes of problems.

Fictitious-domain techniques can be traced back to the early 1950s ([1] and references therein). These techniques have been very successful in solving complicated engineering problems (e.g. [2,3]). The basic idea behind fictitious-domain techniques is to extend domains of complicated shapes to those of simpler shapes for which the generation of meshes is simple and well-established efficient numerical solvers can be applied. Another advantage, when compared with coordinate transformation techniques, is that they are able to retain the PDE in a Cartesian form. It is noted that the transformation of the governing equation into generalized curvilinear coordinates that conform with complex boundaries usually introduces an additional error [4]. A main difficulty here lies in the method employed to take into account the boundary conditions. Glowinski et al. [5] have presented a family of fictitious-domain techniques which are based on the explicit use of Lagrange multipliers defined on the actual boundary and associated with the boundary conditions for Dirichlet elliptic problems. Since then, the Lagrange multiplier/fictitious-domain methods have become increasingly popular. Many further developments and applications have been reported: for instance, for the solution of the Navier-Stokes equations governing incompressible viscous flows (e.g. [6,7]), for the fluid/rigid-body interactions (e.g. [8,3,9]) and for the fluid/flexible-body interactions (e.g. [10]).

Spectral collocation methods/pseudo-spectral methods are global numerical solvers for PDEs and they are known to be very accurate (cf. [11-15]). The methods use a set of orthogonal polynomials such as Chebyshev polynomials (very smooth basis

functions) to represent the approximate solution of the PDE and take the cosine-type points (the zeros of $(1-x^2)T'_N$ in which N is the degree of the polynomial) as the grid points ($-1 \leq x \leq 1$). The conversion of the spectral space into the physical space can be carried out efficiently through a fast Fourier transform. Unlike Galerkin spectral methods, spectral collocation methods approximate the solution in terms of nodal variable values. The main advantages of pseudo-spectral methods lie in their accuracy and economy. For problems whose solutions are smooth (infinitely differentiable), they yield an exponential rate of convergence as the grid is refined or the order of the approximation N is increased (spectral accuracy). The interpolation error decreases more rapidly than any power of $1/N$ [11]. For two-dimensional problems, it requires that the problem domain be rectangular $[-1,-1] \times [1,1]$, which, at the beginning, limits the application of pseudo-spectral techniques to problems defined in simple geometries. There has been a considerable effort put into the development of these techniques in complex geometries. A general and popular way to deal with complex geometries is based on the use of domain decompositions and coordinate transformations. The problem domain is divided into several subdomains, and each subdomain is mapped onto the reference square. Orszag [16] has presented a technique for matching approximate solutions over contiguous regions, namely the patching technique. It requires the approximate solution and its first-order normal derivative to be continuous at the subdomain interfaces. Given a fixed number of subdomains, the approximation is still spectral when the grids on subdomains are refined. The patching technique normally provides an approximate solution that is C^1 function across the internal artificial boundaries. The reader is referred to the book of Karniadakis and Sherwin [17] for a detailed discussion of multi-domain spectral methods.

It is well known that integration is a smoothing operator and is more numerically stable than differentiation. The weak forms associated with finite-element techniques and the inverse statements associated with boundary-element techniques are derived from integrating a weighted residual statement by parts once and twice, respectively (cf. [18]). The integration process reduces the required order of continuity of the approximate solution. A weak solution satisfies the governing equation in an average sense. On the other hand, point collocation techniques such as pseudo-spectral and finite-difference methods are directly based on the strong form of the PDE. The main advantage of these techniques lies in their simplicity as there is no integration of the PDE involved. Grids/meshes are only required for the interpolation of the field variable. The governing equation is satisfied in a pointwise sense.

For the approximation of a function and its derivatives, it has been found that the use of integration, instead of conventional differentiation, to construct the radial-basis-function (RBF) approximations (integral collocation formulation) significantly improves the accuracy of the RBF scheme especially for evaluating derivative functions [19]. Since the introduction of the integral RBF collocation approach [19,20], Kansa et al. [21], based on the theoretical result of Madych and Nelson [22], have concluded that the decreasing rate of convergence for derivative functions caused by differentiation can be avoided in the integral RBF approach. When applying

the integral collocation formulation for the solution of differential equations, with RBFs or Chebyshev polynomials, the constants of integration have been found to be very useful. They provide an alternative way, which is very effective, for the implementation of multiple boundary conditions [23-25] and also allow a higher-order smoothness of the approximate solution across the subdomain interfaces [26]. It will be shown that, in the context of fictitious-domain techniques, the constants of integration can be utilized for the purpose of imposing the prescribed conditions on the actual boundary. This work seems to be the first report implementing the idea of fictitious domain in the context of pseudo-spectral methods; it provides a new way of handling irregularly shaped domains. Even in the case where domain decompositions are required, the use of fictitious domains can be seen to be more straightforward to implement than with the use of coordinate transformations. The presently proposed approach is underpinned by three main features, namely the high-order accuracy of the Chebyshev collocation technique, the effective implementation of boundary conditions of the integral collocation formulation, and the ability to deal with irregularly shaped domains of the fictitious-domain technique.

An outline of the paper is as follows. In section 2, a brief review of point collocation formulations is given. The proposed fictitious-domain technique, which is based on the integral collocation formulation, is described in section 3. The method is then verified through the solution of several test problems in section 4; these test examples involve simply-connected domains, multiply-connected domains, multi-domains, Dirichlet boundary conditions, Neumann boundary conditions and singular solutions. Section 5 gives some concluding remarks.

2 POINT COLLOCATION FORMULATIONS

The Chebyshev collocation technique consists in approximating the solution with Chebyshev polynomials, and forcing the differential equation and the boundary conditions to be satisfied exactly at the cosine-type points. The construction of the Chebyshev approximations representing the approximate solution of the PDE can be based on differentiation and integration.

2.1 Differential formulation

An approximate function f can be represented by the Chebyshev interpolant of degree N as follows

$$f(x) = \sum_{k=0}^{N} a_k T_k(x) = \sum_{k=0}^{N} a_k \cos(k \arccos(x)), \tag{1}$$

where $-1 \leq x \leq 1$, $\{a_k\}_{k=0}^N$ are unknown coefficients and $\{T_k\}_{k=0}^N$ are the Chebyshev polynomials. Expressions of derivatives of (1) will then be obtained through differentiation.

At the Gauss-Lobatto (G-L) points,

$$\{x_i\}_{i=0}^N = \left\{\cos\left(\frac{\pi i}{N}\right)\right\}_{i=0}^N, \tag{2}$$

the values of derivatives of f are simply computed by

$$\widehat{\frac{df}{dx}} = \mathbf{D}^{(1)}\widehat{f} = \mathbf{D}\widehat{f}, \tag{3}$$

$$\widehat{\frac{d^2f}{dx^2}} = \mathbf{D}^{(2)}\widehat{f} = \mathbf{D}^2\widehat{f}, \tag{4}$$

$$\cdots \quad \cdots \quad \cdots \quad \cdots$$

$$\widehat{\frac{d^pf}{dx^p}} = \mathbf{D}^{(p)}\widehat{f} = \mathbf{D}^p\widehat{f}, \tag{5}$$

where the symbol $\widehat{\cdot}$ is used to denote a vector, e.g. $\widehat{f} = (f_0, f_1, \cdots, f_N)^T$ and $\widehat{\frac{d^pf}{dx^p}} = \left(\frac{d^pf_0}{dx^p}, \frac{d^pf_1}{dx^p}, \cdots, \frac{d^pf_N}{dx^p}\right)^T$, and $\mathbf{D}^{(\cdot)}$ are the differentiation matrices. The entries of $\mathbf{D}$ ($\mathbf{D}^{(1)}$) are given by

$$D_{ij} = \frac{\bar{c}_i}{\bar{c}_j}\frac{(-1)^{i+j}}{x_i - x_j}, \qquad 0 \leq i, j \leq N, \quad i \neq j, \tag{6}$$

$$D_{ii} = -\frac{x_i}{2(1-x_i^2)}, \qquad 1 \leq i \leq N-1, \tag{7}$$

$$D_{00} = -D_{NN} = \frac{2N^2+1}{6}, \tag{8}$$

where $\bar{c}_0 = \bar{c}_N = 2$ and $\bar{c}_i = 1$ for $i = 1, 2, \cdots, N-1$. It is noted that the diagonal entries of $\mathbf{D}$ can also be obtained in the way that represents exactly the derivative of a constant

$$D_{ii} = -\sum_{j=0, j\neq i}^{N} D_{ij}. \tag{9}$$

For the case of smooth functions, the Chebyshev approximation scheme is known to be very accurate (exponential accuracy). The error is $O(N^{-\alpha})$, where α depends on the regularity of the function. It should be emphasized that there is a reduction in accuracy for the approximation of derivative functions; this reduction is an increasing function of derivative order (cf. [13]).

2.2 Integral formulation

This formulation uses a truncated Chebyshev series of degree N to represent a derivative of an unknown function f, e.g.

$$\frac{d^p f(x)}{dx^p} = \sum_{k=0}^{N} a_k T_k(x). \tag{10}$$

Expressions for lower-order derivatives and the function itself are then obtained through integration as

$$\frac{d^{p-1} f(x)}{dx^{p-1}} = \sum_{k=0}^{N} a_k I_k^{(p-1)}(x) + c_1, \tag{11}$$

$$\frac{d^{p-2} f(x)}{dx^{p-2}} = \sum_{k=0}^{N} a_k I_k^{(p-2)}(x) + c_1 x + c_2, \tag{12}$$

$$\cdots \quad \cdots \qquad \cdots \qquad \cdots \qquad \cdots$$

$$\frac{df(x)}{dx} = \sum_{k=0}^{N} a_k I_k^{(1)}(x) + c_1 \frac{x^{p-2}}{(p-2)!} + c_2 \frac{x^{p-3}}{(p-3)!} + \cdots + c_{p-2} x + c_{p-1}, \tag{13}$$

$$f(x) = \sum_{k=0}^{N} a_k I_k^{(0)}(x) + c_1 \frac{x^{p-1}}{(p-1)!} + c_2 \frac{x^{p-2}}{(p-2)!} + \cdots + c_{p-1} x + c_p, \tag{14}$$

where $I_k^{(p-1)}(x) = \int T_k(x)dx$, $I_k^{(p-2)}(x) = \int I_k^{(p-1)}(x)dx, \cdots, I_k^{(0)}(x) = \int I_k^{(1)}(x)dx$, and $c_1, c_2, \cdots, c_p$ are integration constants.

Unlike conventional differential schemes, the starting point of the integral collocation scheme can vary in use, depending on the particular application under consideration. In this regard, the concept of scheme order is introduced here. An integral collocation scheme (ICS) is said to be of pth order, denoted by ICSp, if the scheme starts with the Chebyshev approximation of the pth-order derivative of f. A differential collocation scheme can be considered as a special case of ICS by letting p be zero (ICS0).

The evaluation of (10)-(14) at the G-L points leads to

$$\widehat{\frac{d^p f}{dx^p}} = \mathcal{I}_{[p]}^{(p)} \widehat{s}, \tag{15}$$

$$\widehat{\frac{d^{p-1} f}{dx^{p-1}}} = \mathcal{I}_{[p]}^{(p-1)} \widehat{s}, \tag{16}$$

$$\cdots\cdots\cdots \tag{17}$$

$$\widehat{\frac{df}{dx}} = \mathcal{I}_{[p]}^{(1)} \widehat{s}, \tag{18}$$

$$f = \mathcal{I}_{[p]}^{(0)} \widehat{s}, \tag{19}$$

where subscript [.] and superscript (.) are used to indicate the orders of ICS and derivative function, respectively,

$$\widehat{s} = (a_0, a_1, \cdots, a_N, c_1, c_2, \cdots, c_p)^T,$$

$$\mathcal{I}^{(p)}_{[p]} = \begin{bmatrix} T_0(x_0), & T_1(x_0), & \cdots, & T_N(x_0), & 0, & 0, & \cdots, & 0, & 0 \\ T_0(x_1), & T_1(x_1), & \cdots, & T_N(x_1), & 0, & 0, & \cdots, & 0, & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ T_0(x_N), & T_1(x_N), & \cdots, & T_N(x_N), & 0, & 0, & \cdots, & 0, & 0 \end{bmatrix},$$

$$\mathcal{I}^{(p-1)}_{[p]} = \begin{bmatrix} I^{(p-1)}_0(x_0), & I^{(p-1)}_1(x_0), & \cdots, & I^{(p-1)}_N(x_0), & 1, & 0, & \cdots, & 0, & 0 \\ I^{(p-1)}_0(x_1), & I^{(p-1)}_1(x_1), & \cdots, & I^{(p-1)}_N(x_1), & 1, & 0, & \cdots, & 0, & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I^{(p-1)}_0(x_N), & I^{(p-1)}_1(x_N), & \cdots, & I^{(p-1)}_N(x_N), & 1, & 0, & \cdots, & 0, & 0 \end{bmatrix},$$

$\cdots\cdots$, and

$$\mathcal{I}^{(0)}_{[p]} = \begin{bmatrix} I^{(0)}_0(x_0), & I^{(0)}_1(x_0), & \cdots, & I^{(0)}_N(x_0), & \frac{x_0^{p-1}}{(p-1)!}, & \frac{x_0^{p-2}}{(p-2)!}, & \cdots, & x_0, & 1 \\ I^{(0)}_0(x_1), & I^{(0)}_1(x_1), & \cdots, & I^{(0)}_N(x_1), & \frac{x_1^{p-1}}{(p-1)!}, & \frac{x_1^{p-2}}{(p-2)!}, & \cdots, & x_1, & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I^{(0)}_0(x_N), & I^{(0)}_1(x_N), & \cdots, & I^{(0)}_N(x_N), & \frac{x_N^{p-1}}{(p-1)!}, & \frac{x_N^{p-2}}{(p-2)!}, & \cdots, & x_N, & 1 \end{bmatrix}.$$

Several advantages of pseudospectral techniques based on integrated basis functions over those based on differentiated basis functions for solving two-point boundary value problems have been reported in [27,24]. The present study employs the integral collocation formulation for the purpose of implementing the boundary conditions in the context of fictitious-domain pseudopectral techniques.

3 THE PROPOSED FICTITIOUS-DOMAIN TECHNIQUE

Consider the approximation of the solution of the differential problem consisting of the equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = b(x, y), \qquad (x, y) \in \Omega, \tag{20}$$

where u is the field/dependent variable, b is a driving/forcing function, and Ω is an irregular bounded domain, together with Dirichlet and Neumann boundary conditions on the boundary $\partial\Omega$.

For fictitious-domain techniques/domain embedding methods, a spatial domain of complicated shape is extended to a simple one, where structured grids/meshes can be used. It is worth mentioning that the grids are independent of the boundary definition.

Figure 1 shows an extension of Ω to a rectangular domain that is discretized using a tensor product grid formed by the G-L points. The Chebyshev discrete approximations representing the field variable u and its derivatives are constructed on these grids; their final forms are written in terms of the values of u at the grid points. Since the nodal points do not generally lie on the boundary of the actual domain, special treatments are required to implement the boundary conditions.

The present study attempts to include information on the boundary in the Chebyshev approximations. It can be done by using integral collocation schemes. Unlike conventional differential techniques, the integral collocation approach is capable of generating new coefficients (i.e. integration constants). This feature allows one to add some additional equations to the system that converts the spectral space into the physical space. These extra equations can be used to impose the prescribed conditions on the actual boundary. In what follows, the present method is presented in detail for two types of boundary conditions, namely Dirichlet and Neumann conditions.

3.1 Dirichlet boundary conditions

Lines aa', bb', cc', dd' and ee' in Figure 1 present typical cases for the approximation of $\partial u/\partial y$ and $\partial^2 u/\partial y^2$.

3.1.1 Case 1 - Line ee':

Along this line, there are no boundary points. The task thus becomes simple, i.e. simply expressing the values of $\partial u/\partial y$ and $\partial^2 u/\partial y^2$ at a grid point in terms of the nodal values of u along the line. This can be done by applying the ICS0 scheme. Its Chebyshev expressions are given by (3)-(5).

3.1.2 Case 2 - Line dd':

This line and the boundary $\partial\Omega$ intersect at two points, namely y_{b1} and y_{b2}. The first boundary point y_{b1} is also a grid node, and hence it is straightforward to implement u_{b1}. Assume that the second boundary point y_{b2} does not coincide with any grid nodes. To impose u_{b2}, one extra equation is needed and hence the ICS1 scheme can be applied here. The conversion system is formed as follows

$$\begin{pmatrix} \widehat{u} \\ u_{b2} \end{pmatrix} = \begin{bmatrix} \mathcal{I}^{(0)}_{[1]} \\ \mathcal{B} \end{bmatrix} \begin{pmatrix} \widehat{a} \\ c_1 \end{pmatrix} = \mathcal{C} \begin{pmatrix} \widehat{a} \\ c_1 \end{pmatrix}, \tag{21}$$

where $\mathcal{C}$ is the conversion matrix of dimension $(N_y{+}2){\times}(N_y{+}2)$, $\widehat{a} = \left(a_0, a_1, \cdots, a_{N_y}\right)^T$, $\widehat{u} = \left(u_0, u_1, \cdots, u_{N_y}\right)^T$, and

$$\mathcal{B} = \left[I_0^{(0)}(y_{b2}), I_1^{(0)}(y_{b2}), \cdots, I_{N_y}^{(0)}(y_{b2}), 1 \right]_{[1]}.$$

Solving (21) yields

$$\begin{pmatrix} \widehat{a} \\ c_1 \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ u_{b2} \end{pmatrix}. \tag{22}$$

The values of $\partial u/\partial y$ and $\partial^2 u/\partial y^2$ at the grid points are then computed by

$$\widehat{\frac{\partial u}{\partial y}} = \mathcal{I}_{[1]}^{(1)} \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ u_{b2} \end{pmatrix}, \tag{23}$$

$$\widehat{\frac{\partial^2 u}{\partial y^2}} = \mathcal{I}_{[1]}^{(2)} \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ u_{b2} \end{pmatrix}, \tag{24}$$

where

$$\mathcal{I}_{[1]}^{(2)} = \begin{bmatrix} \frac{dT_0}{dy}(y_0), & \frac{dT_1}{dy}(y_0), & \cdots, & \frac{dT_{N_y}}{dy}(y_0), & 0 \\ \frac{dT_0}{dy}(y_1), & \frac{dT_1}{dy}(y_1), & \cdots, & \frac{dT_{N_y}}{dy}(y_1), & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{dT_0}{dy}(y_{N_y}), & \frac{dT_1}{dy}(y_{N_y}), & \cdots, & \frac{dT_{N_y}}{dy}(y_{N_y}), & 0 \end{bmatrix}.$$

It is noted that ICS2 is also applicable here. The second integration constant c_2 can be used for the purpose of imposing the governing equation at $y = y_{b1}$ (also y_0). The conversion system thus becomes

$$\begin{pmatrix} \widehat{u} \\ \frac{\partial^2 u_{b1}}{\partial y^2} \\ u_{b2} \end{pmatrix} = \begin{bmatrix} \mathcal{I}_{[2]}^{(0)} \\ \mathcal{B} \end{bmatrix} \begin{pmatrix} \widehat{a} \\ c_1 \\ c_2 \end{pmatrix} = \mathcal{C} \begin{pmatrix} \widehat{a} \\ c_1 \\ c_2 \end{pmatrix}, \tag{25}$$

where $\mathcal{C}$ is the matrix of dimension $(N_y + 3) \times (N_y + 3)$ and

$$\mathcal{B} = \begin{bmatrix} T_0(y_{b1}), & T_1(y_{b1}), & \cdots, & T_{N_y}(y_{b1}), & 0, & 0 \\ I_0^{(0)}(y_{b2}), & I_1^{(0)}(y_{b2}), & \cdots, & I_{N_y}^{(0)}(y_{b2}), & y_{b2}, & 1 \end{bmatrix}_{[2]}.$$

In (25), the value of $\partial^2 u_{b1}/\partial y^2$ is known as it is obtained through (20).

It leads to

$$\widehat{\frac{\partial u}{\partial y}} = \mathcal{I}_{[2]}^{(1)} \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \frac{\partial^2 u_{b1}}{\partial y^2} \\ u_{b2} \end{pmatrix}, \tag{26}$$

$$\widehat{\frac{\partial^2 u}{\partial y^2}} = \mathcal{I}_{[2]}^{(2)} \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \frac{\partial^2 u_{b1}}{\partial y^2} \\ u_{b2} \end{pmatrix}. \tag{27}$$

3.1.3 Case 3 - Line cc':

There are two boundary points y_{b1} and y_{b2}, which are also grid nodes. Two schemes ICS0 and ICS2 can be applied here. For ICS0, the values of derivatives of u with respect to y are computed using (3)-(5). For ICS2, one can utilize two integration constants to force the governing equation to be satisfied exactly at the two boundary points

$$\begin{pmatrix} \widehat{u} \\ \frac{\partial^2 u_{b1}}{\partial y^2} \\ \frac{\partial^2 u_{b2}}{\partial y^2} \end{pmatrix} = \begin{bmatrix} \mathcal{I}^{(0)}_{[2]} \\ \mathcal{B} \end{bmatrix} \begin{pmatrix} \widehat{a} \\ c_1 \\ c_2 \end{pmatrix}, \tag{28}$$

where $\partial^2 u_{b1}/\partial y^2$ and $\partial^2 u_{b2}/\partial y^2$ are easily computed using the governing equation, and

$$\mathcal{B} = \begin{bmatrix} T_0(y_{b1}), & T_1(y_{b1}), & \cdots, & T_{N_y}(y_{b1}), & 0, & 0 \\ T_0(y_{b2}), & T_1(y_{b2}), & \cdots, & T_{N_y}(y_{b2}), & 0, & 0 \end{bmatrix}_{[2]}.$$

The remaining steps for obtaining the Chebyshev approximations of $\partial u/\partial y$ and $\partial^2 u/\partial y^2$ are similar to previous cases and therefore omitted here for brevity.

3.1.4 Case 4 - Line bb':

Along this line, there are two boundary points. Assume that they are not grid points. ICS2 can be employed to impose the two boundary conditions

$$\begin{pmatrix} \widehat{u} \\ u_{b1} \\ u_{b2} \end{pmatrix} = \begin{bmatrix} \mathcal{I}^{(0)}_{[2]} \\ \mathcal{B} \end{bmatrix} \begin{pmatrix} \widehat{a} \\ c_1 \\ c_2 \end{pmatrix}, \tag{29}$$

where

$$\mathcal{B} = \begin{bmatrix} I^{(0)}_0(y_{b1}), & I^{(0)}_1(y_{b1}), & \cdots, & I^{(0)}_{N_y}(y_{b1}), & y_{b1}, & 1 \\ I^{(0)}_0(y_{b2}), & I^{(0)}_1(y_{b2}), & \cdots, & I^{(0)}_{N_y}(y_{b2}), & y_{b2}, & 1 \end{bmatrix}_{[2]}.$$

3.1.5 Case 5 - Line aa':

A number of schemes can be applied here. In the following, two typical schemes are presented.

If the contact point y_b is not a grid node, one can use ICS1

$$\begin{pmatrix} \widehat{u} \\ u_b \end{pmatrix} = \begin{bmatrix} \mathcal{I}^{(0)}_{[1]} \\ \mathcal{B} \end{bmatrix} \begin{pmatrix} \widehat{a} \\ c_1 \end{pmatrix}, \tag{30}$$

where

$$\mathcal{B} = \begin{bmatrix} I^{(0)}_0(y_b), & I^{(0)}_1(y_b), & \cdots, & I^{(0)}_{N_y}(y_b), & 1 \end{bmatrix}_{[1]}.$$

If the contact point is also a grid node, one can employ ICS0 or ICS2. For the latter, the conversion system is given by

$$\begin{pmatrix} \widehat{u} \\ \frac{\partial u_b}{\partial y} \\ \frac{\partial^2 u_b}{\partial y^2} \end{pmatrix} = \begin{bmatrix} \mathcal{I}^{(0)}_{[2]} \\ \mathcal{B} \end{bmatrix} \begin{pmatrix} \widehat{a} \\ c_1 \\ c_2 \end{pmatrix}, \tag{31}$$

where

$$\mathcal{B} = \begin{bmatrix} I_0^{(1)}(y_b), & I_1^{(1)}(y_b), & \cdots, & I_{N_y}^{(1)}(y_b), & 1, & 0 \\ T_0(y_b), & T_1(y_b), & \cdots, & T_{N_y}(y_b), & 0, & 0 \end{bmatrix}_{[2]}.$$

In (31), $\partial u_b/\partial y$ and $\partial^2 u_b/\partial y^2$ are known values, which are derived from using boundary conditions.

The values of $\partial u/\partial x$ and $\partial^2 u/\partial x^2$ at the grid points along horizontal lines can be computed in a similar fashion.

The Chebyshev approximations of derivatives at a grid point are expressed in terms of the nodal values of u along the grid lines that goes through that point. It should be emphasized that they already contain information about the boundary of Ω (i.e. locations and boundary values). As with finite-difference and finite-element techniques, one will gather these approximations together to form the global matrices for the discretization of the PDE. This task is relatively simple since the grid used here is regular. By collocating the governing equation at the grid points and then deleting rows corresponding to points that lie on the boundary, a square system of algebraic equations is obtained, which is solved for the approximate solution.

3.2 Neumann boundary conditions

In the context of Cartesian-grid-based collocation methods, Neumann boundary conditions are known to be more difficult to implement than Dirichlet boundary conditions. It is particularly acute for the case of non-rectangular boundaries. Viswanathan [28] has proposed constructing a finite-difference approximation at a grid point that lies adjacent to the curved boundary by taking into account the rate of change of the normal gradient of the field variable along the boundary. In the work of Thuraisamy [29,30], the normal derivative at a boundary point was approximated using two lines that intersect at that point and make angles of $\pi/4$ on either side of the local normal direction. Recently, Sanmigue-Rojas et al. [31] have reported a technique for generating a non-uniform Cartesian grid in which all the boundary points are regular nodes of the grid.

In the present technique, like Dirichlet boundary conditions, Neumann boundary conditions are also imposed through the transformation of the spectral space into the physical space. The implementation also takes the advantage of fictitious domains and Chebyshev polynomials. A computational domain is now rectangular

(fictitious domains) and the spectral approximations are defined everywhere in the domain (Chebyshev polynomials). Hence, one can easily derive highly accurate approximations of $\partial u/\partial x$ and $\partial u/\partial y$ at any point in the fictitious domain, where in general a boundary point is not a grid node, from the grid values.

Assume that the side CD (Figure 1) is specified with a Neumann boundary condition. Consider line dd'. This line and the boundary $\partial\Omega$ intersect at two points, namely (x_{b1}, y_{b1}) and (x_{b2}, y_{b2}) with $x_{b1} = x_{b2}$, which have Dirichlet and Neumann boundary conditions, respectively. In the following discussion, attention will be given to the implementation of the latter.

Using ICS2, the conversion system can be formed as

$$\begin{pmatrix} \widehat{u} \\ \frac{\partial^2 u_{b1}}{\partial y^2} \\ \frac{\partial u_{b2}}{\partial y} \end{pmatrix} = \begin{bmatrix} \mathcal{I}^{(0)}_{[2]} \\ \mathcal{B} \end{bmatrix} \begin{pmatrix} \widehat{a} \\ c_1 \\ c_2 \end{pmatrix}, \tag{32}$$

where

$$\mathcal{B} = \begin{bmatrix} T_0(y_{b1}), & T_1(y_{b1}), & \cdots, & T_{N_y}(y_{b1}), & 0, & 0 \\ I^{(1)}_0(y_{b2}), & I^{(1)}_1(y_{b2}), & \cdots, & I^{(1)}_{N_y}(y_{b2}), & 1, & 0 \end{bmatrix}_{[2]}.$$

Vectors of values of $\partial u/\partial y$ and $\partial^2 u/\partial y^2$ become

$$\widehat{\frac{\partial u}{\partial y}} = \mathcal{I}^{(1)}_{[2]} \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \frac{\partial^2 u_{b1}}{\partial y^2} \\ \frac{\partial u_{b2}}{\partial y} \end{pmatrix}, \tag{33}$$

$$\widehat{\frac{\partial^2 u}{\partial y^2}} = \mathcal{I}^{(2)}_{[2]} \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \frac{\partial^2 u_{b1}}{\partial y^2} \\ \frac{\partial u_{b2}}{\partial y} \end{pmatrix}. \tag{34}$$

The problem here is that the value of $\partial u_{b2}/\partial y$ on the right-hand side of (33) and (34) is not known. The method thus requires some additional manipulations. Those Neumann conditions along the boundary CD are taken into account by replacing $\partial u_{b2}/\partial y$ with

$$\frac{1}{n_y}\left(\frac{\partial u_{b2}}{\partial n} - n_x \frac{\partial u_{b2}}{\partial x}\right), \tag{35}$$

where n_x and n_y are the $x-$ and $y-$components of the unit vector normal to the boundary. The term $\partial u_{b2}/\partial x$ in (35) is still unknown; however, one can express it in terms of the grid values of u. Since line dd' passes through one of the G-L points $\{x_i\}_{i=0}^{N_x}$ (e.g. $x_{b2} = x_p$), this derivative value can be easily computed as

$$\frac{\partial u_{b2}}{\partial x} = \mathbf{D}^{(1)}(p,:)\widehat{u}_s, \tag{36}$$

where $\mathbf{D}^{(1)}(p,:)$ is the pth row of the differentiation matrix $\mathbf{D}^{(1)}$ (the entries of $\mathbf{D}^{(1)}$ are defined by (6)-(8)), and

$$\widehat{u}_s = \left(u(x_0, y_{b2}), u(x_1, y_{b2}), \cdots, u(x_{N_x}, y_{b2})\right)^T. \tag{37}$$

Each component of $\widehat{u}_s$ is computed from the interpolation of the nodal point function u along the line that runs parallel to the y axis and goes through that point

$$u(x_i, y_{b2}) = \left[T_0(y_{b2}), T_1(y_{b2}), \cdots, T_{N_y}(y_{b2})\right] \left[\mathbf{D}^{(0)}\right]^{-1} \widehat{u}, \tag{38}$$

where

$$\widehat{u} = \left(u(x_i, y_0), u(x_i, y_1), \cdots, u(x_i, y_{N_y})\right)^T$$

and $\left[\mathbf{D}^{(0)}\right]^{-1}$ is the inverse of $\mathbf{D}^{(0)}$

$$\begin{aligned} D_{ij}^{(0)} &= T_j(x_i), \\ \left[D^{(0)}\right]_{ij}^{-1} &= \frac{2}{N_y} \frac{1}{\bar{c}_i} \frac{1}{\bar{c}_j} T_i(x_j), \end{aligned}$$

with $0 \leq i, j \leq N_y$.

It can be seen that the values of derivatives of u with respect to y ((33) and (34)) are written in terms of nodal variable values, and they take account of derivative boundary conditions.

Similarly, one can construct the Chebyshev approximations for $\partial u/\partial x$ and $\partial^2 u/\partial x^2$ at the grid points along horizontal lines, which cross the boundary CD, in terms of the grid values of u.

The proposed technique imposes the boundary conditions prior to the assembly process. For Dirichlet boundary conditions, the approximation of derivatives at a grid point involves the grid values along the lines that go through that point, while for Neumann boundary conditions, it involves all the grid values.

4 NUMERICAL RESULTS

The accuracy of an approximation scheme is measured by means of the discrete relative L_2 error of the solution defined as

$$N_e = \frac{\sqrt{\sum_{i=0}^{M-1} \left(u_e(x_i, y_i) - u(x_i, y_i)\right)^2}}{\sqrt{\sum_{i=0}^{M-1} \left(u_e(x_i, y_i)\right)^2}}, \tag{39}$$

where M is the number of test points, and u_e and u are the exact and computed solutions, respectively. The proposed technique is verified through the solution of Poisson equations. The error (39) is computed at the interior points of the actual domain. It is assumed that a driving function b can be extended along the grid lines to a fictitious part of the domain in a smooth manner. This is achievable and such an extension can be constructed explicitly ([32] and references therein). Several smooth and singular test problems are considered. For the former, an exact solution

to the problem is chosen in advance, and the appropriate boundary conditions (u or $\partial u/\partial n$) and function b are then derived from the exact solution. Examples chosen involve single domains, multi-domains, Dirichlet boundary conditions and Neumann boundary conditions. We also look at an example of a singular solution.

Before considering the solution of PDEs in the following examples, the integral collocation formulation for fictitious domains is first tested with the case of function interpolations. Consider a function $f = \sin(\pi x)$ with $-1 \leq x \leq 1$. Apart from the values of f at the G-L points, other information on f is also given. Three examples corresponding to cases of lines bb', aa' and cc' are considered. Extra information is given at $x_{b1} = -1/3$ and/or $x_{b2} = 1/3$ which do not coincide with any grid points. The ICS2 scheme is employed here to evaluate the grid values of derivatives of f. Table 1 shows that an exponential rate of convergence is achieved for all examples. Another example, where the integral and differential collocation approaches use the same sets of collocation points, is also studied. The values of df/dx and d^2f/dx^2 (extra information) are given at $x = 0$ and the discretizations are chosen such that $x = 0$ is a grid point. Figure 2 shows that the integral approach yields a higher degree of accuracy than the differential approach. It is noted that the latter does not take account of extra information in the process of determining expansion coefficients (its nodal derivative values are simply computed using (3)-(5)).

4.1 Example 1 (Simply-connected domain)

Consider a simply-connected domain as shown in Figure 1. Here, points A, B, C, D and E are chosen to be $(-1/2, -1)$, $(3/4, -1)$, $(3/4, 0)$, $(0, 1)$ and $(-1/2, 1)$, respectively, and the centre and radius of the arc EA are taken as (1/4,0) and 5/4. As mentioned earlier, the driving function and boundary conditions are provided by the exact solution. This example uses

$$u_e(x, y) = \frac{1}{\pi^2} \sin(\pi x) \sin(\pi y), \tag{40}$$

from which it is easy to deduce the driving function

$$b(x, y) = -2 \sin(\pi x) \sin(\pi y). \tag{41}$$

Dirichlet boundary conditions, obtained from (40), are specified on the boundary. The problem domain is embedded in a regular quadrilateral of 2×2 centred at the origin. Figure 3 shows the plot of u over the extended domain.

There are two versions to be employed here. The first version uses ICS0, ICS1 and ICS2. At the boundary grid-points, only the boundary conditions are imposed. In the second version, ICS0 and ICS2 are employed. This version forces both the boundary conditions and the governing equation to be satisfied at the boundary grid-points. Numerical results show that the two versions yield spectral accuracy and they have similar degrees of accuracy. Unlike the case of rectangular domains

[25], satisfaction of the governing equation at the boundary grid-points here does not result in a significant improvement in accuracy. Table 2 presents errors $N_e(u)$ obtained by the second version.

4.2 Example 2 (domain with holes)

This problem takes the exact solution and driving function as

$$u_e(x,y) = \sin(2\pi x)\cosh(2y) - \cos(2\pi x)\sinh(2y), \tag{42}$$

$$b(x,y) = 4(1-\pi^2)\left[\sin(2\pi x)\cosh(2y) - \cos(2\pi x)\sinh(2y)\right]. \tag{43}$$

A domain with several holes is employed (Figure 4). The domain and the square hole are chosen as $[-1,-1]\times[1,1]$ and $[1/10,-9/10]\times[9/10,-1/10]$, respectively. The circular hole has its centre at (-1/2,1/2) and a radius of 2/5. Dirichlet boundary conditions, obtained from (42), are specified along the boundaries. Figure 4 also shows the variation of the function u over this extended domain.

The ICS2 scheme is employed to solve the problem. Results concerning $N_e(u)$ are given in Table 3, which indicate that the approximate solution converges exponentially to the exact solution as the grid is refined.

4.3 Example 3 (Neumann boundary condition)

Consider an irregularly shaped domain as shown in Figure 5. The edge CD takes a Neumann boundary condition, while the others are specified with Dirichlet boundary conditions. Positions of points A, B, C, D, E and F are (0,-1), (-1,1), (1,0), (0,1), (-1,1) and (-1,0), respectively. EF is an arc centered at (-1,-1). The exact solution and driving function used here are given below

$$u_e(x,y) = x\left[\sin(2x)\cosh(2y) - \cos(2x)\sinh(2y)\right], \tag{44}$$

$$b(x,y) = 4\left[\cos(2x)\cosh(2y) + \sin(2x)\sinh(2y)\right]. \tag{45}$$

The variation of u_e over the extended domain defined by $[-1,-1]\times[1,1]$ is also shown in Figure 5. Discretizations are carried out using ICS0 and ICS2. The former is applied for lines AB, BC, DE and EF. Table 4 shows that the proposed fictitious-domain technique yields an exponential rate of convergence when the grid is refined.

4.4 Example 4 (domain decomposition)

The use of domain decompositions is necessary to deal with complicated/large-scale engineering problems. The purpose of giving this example here is to investigate

whether the rapid convergence of the solution with respect to grid refinement is preserved when the proposed fictitious-domain technique is employed in conjunction with domain decompositions.

Consider a Dirichlet problem with the domain of interest being composed of 2 octagons of unit inradius (Figure 6). The exact solution and driving function are

$$u_e(x,y) = \cos\left(\frac{\pi}{2}x\right)\sinh(y), \tag{46}$$

$$b(x,y) = \left[1 - \left(\frac{\pi}{2}\right)^2\right]\cos\left(\frac{\pi}{2}x\right)\sinh(y). \tag{47}$$

The present multi-domain scheme is based on the substructuring technique (cf. [33]). The problem domain is divided into two non-overlapping subdomains (Figure 6). The solution procedure consists of two stages. In the first stage, one deals with the interface solution, while the second stage involves finding the solution of subdomains. Each subdomain is embedded in the reference square domain, and ICS0 and ICS2 are employed to discretize the governing equation. Continuity of the function u and its normal derivative is imposed pointwise along the interface. Results obtained are displayed in Table 5, showing that the fictitious-multidomain technique also provides a very fast convergence as the grids on subdomains are refined.

4.5 Example 5 (singular solution)

This example is concerned with the case of singular solutions. A non-rectangular domain with curved and straight boundaries is considered here (Figure 7). The curve is an arc having its centre at (-1,-1) and a radius of 2. The present singularity of the solution is due to incompatibility of the differential equation ($\nabla^2 u = 1$) with the boundary conditions (u=0) at the lower left corner (mild singularity). The approximate solution u is represented by means of ICS1. Table 6 shows the values of u at point (0,0). Since the exact solution is not known, the values of u obtained with coarse grids are compared with the value of u with the fine grid (25×25). Like conventional pseudospectral techniques, in this case (singular solution), the proposed technique is capable of yielding an algebraic convergence rate only. However, it can be seen that the obtained convergence rate is fast, up to $O(h^{8.7})$.

We have also employed a finite-element method (FEM) to solve this problem. The present FEM results (Table 7) were obtained using the PDE tool in MATLAB. To have the solution converged to 5 significant digits, the FEM requires a mesh that is finer than that of 69,120 linear triangular elements and 34,849 nodes. It is noted that the present technique is able to provide a solution with 9 significant digits using a relatively coarse grid of 19×19 (361 nodal points) (CPU time < 1 sec, Pentium 4–2.4GHz, MATLAB environment). Thus, it appears that the proposed technique is more efficient than the FEM. However, for a solution with low regularity (strong singularity), the accuracy of the Chebyshev approximations deteriorates and

there is no significant advantage over low-order approximation methods. In such a circumstance, a suitable treatment of the singularities is needed in order to obtain a high level of accuracy. This issue has been dealt with and reported in the literature, see, e.g., [14].

5 CONCLUDING REMARKS

This paper reports a new global fictitious-domain/integral-collocation method for the numerical solution of second-order elliptic PDEs in irregularly shaped domains. The construction of the Chebyshev approximations representing the dependent variable and its derivatives is based on integration rather than conventional differentiation. Information about the actual boundary is taken into account through the transformation of the spectral space into the physical space. Different types of domains (simply-connected domains, multiply-connected domains and multi-domains) and of boundary conditions (Dirichlet and Neumann boundary conditions) are considered. Numerical results obtained show that, for smooth (analytic) solutions, the technique yields an exponential rate of convergence as the grid is refined. With this very high order accuracy (comparing with second-order accuracy of the Lagrange-multiplier-based fictitious-domain technique), the present technique is particularly attractive for solving problems where high accuracy is required.

ACKNOWLEDGEMENTS

This work is supported by the Australian Research Council. The authors are grateful to Prof. R.I. Tanner and Prof. X.-J. Fan for fruitful discussions throughout this study.

REFERENCES

1. Glowinski R, Kuznetsov Yu. Distributed Lagrange multipliers based on fictitious domain method for second order elliptic problems. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**: 1498–1506.

2. Young DP, Melvin RG, Bieterman MB, Johnson FT, Samant SS, Bussoletti JE. A locally refined rectangular grid finite element method: Application to computational fluid dynamics and computational physics. *Journal of Computational Physics* 1991; **92**: 1–66.

3. Glowinski R, Pan T-W, Hesla TI, Joseph DD, Priaux J. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. *Journal of Computational Physics* 2001; **169**: 363–426.

4. Fletcher CAJ. *Computational Techniques for Fluid Dynamics 2.* Springer-Verlag: Berlin, 1988.

5. Glowinski R, Pan T-W, Priaux J. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**: 283–303.

6. Glowinski R, Pan T-W, Priaux J. A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1994; **112**: 133–148.

7. Bertrand F, Tanguy PA, Thibault F. A three-dimensional fictitious domain method for incompressible fluid flow problems. *International Journal for Numerical Methods in Fluids* 1997; **25**: 719–736.

8. Patankar NA, Singh P, Joseph DD, Glowinski R, Pan T-W. A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow* 2000; **26**: 1509–1524.

9. Yu Z, Phan-Thien N, Fan Y, Tanner RI. Viscoelastic mobility problem of a system of particles. *Journal of Non-Newtonian Fluid Mechanics* 2002; **104**: 87–124.

10. Yu Z. A DLM/FD method for fluid/flexible-body interactions. *Journal of Computational Physics* 2005; **207**: 1–27.

11. Gottlieb D, Orszag SA. *Numerical Analysis of Spectral Methods: Theory and Applications*. SIAM: Philadelphia, 1977.

12. Canuto C, Hussaini MY, Quarteroni A, Zang TA. *Spectral Methods in Fluid Dynamics*. Springer-Verlag: New York, 1988.

13. Trefethen LN. *Spectral Methods in MATLAB*. SIAM: Philadelphia, 2000.

14. Peyret R. *Spectral Methods for Incompressible Viscous Flow*. Springer-Verlag: New York, 2002.

15. Owens RG, Phillips TN. *Computational Rheology*. Imperial College Press: London, 2002.

16. Orszag SA. Spectral methods for problems in complex geometries. *Journal of Computational Physics* 1980; **37**: 70–92.

17. Karniadakis GE, Sherwin SJ. *Spectral/hp Element Methods for CFD*. Oxford University Press: Oxford, 1999.

18. Brebbia CA, Walker S. *Boundary Element Techniques in Engineering*. Newnes-Butterworths: London, 1980.

19. Mai-Duy N, Tran-Cong T. Approximation of function and its derivatives using radial basis function networks. *Applied Mathematical Modelling* 2003; **27**: 197–220.

20. Mai-Duy N, Tran-Cong T. Numerical solution of differential equations using multiquadric radial basis function networks. *Neural Networks* 2001; **14**: 185–199.

21. Kansa EJ, Power H, Fasshauer GE, Ling L. A volumetric integral radial basis function method for time-dependent partial differential equations: I. Formulation. *Engineering Analysis with Boundary Elements* 2004; **28**: 1191–1206.

22. Madych WR, Nelson SA. Multivariate interpolation and conditionally positive definite functions, II. *Mathematics of Computation* 1990; **54**: 211-230.

23. Mai-Duy N. Solving high order ordinary differential equations with radial basis function networks. *International Journal for Numerical Methods in Engineering* 2005; **62**: 824–852.

24. Mai-Duy N. An effective spectral collocation method for the direct solution of high-order ODEs. *Communications in Numerical Methods in Engineering* 2006; **22**: 627–642.

25. Mai-Duy N, Tanner RI. A spectral collocation method based on integrated Chebyshev polynomials for biharmonic boundary-value problems. *Journal of Computational and Applied Mathematics* 2007; **201**: 30–47.

26. Mai-Duy N, Tran-Cong T. An efficient domain-decomposition pseudo-spectral method for solving elliptic differential equations. *Communications in Numerical Methods in Engineering*, in press.

27. Greengard L. Spectral integration and two-point boundary value problems. *SIAM Journal on Numerical Analysis* 1991; **28**: 1071–1080.

28. Viswanathan RV. Solution of Poisson's equation by relaxation method–normal gradient specified on curved boundaries. *Mathematical Tables and Other Aids to Computation* 1957; **11**: 67–78.

29. Thuraisamy V. Approximate solutions for mixed boundary value problems by finite-difference methods. *Mathematics of Computation* 1969; **23**: 373–386.

30. Thuraisamy V. Monotone type discrete analogue for the mixed boundary value problem. *Mathematics of Computation* 1969; **23**: 387–394.

31. Sanmiguel-Rojas E, Ortega-Casanova J, del Pino C, Fernandez-Feria R. A Cartesian grid finite-difference method for 2D incompressible viscous flows in irregular geometries. *Journal of Computational Physics* 2005; **204**: 302–318.

32. Atkinson KE. The numerical evaluation of particular solutions for Poisson's equation. *IMA Journal of Numerical Analysis* 1985; **5**: 319–338.

33. Smith BF, Bjorstad PE, Gropp WD. *Domain Decomposition Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press: New York, 1996.

Table 1: $f = \sin(\pi x)$, $-1 \leq x \leq 1$: Relative L_2 errors of df/dx and d^2f/dx^2 by the integral collocation approach. Apart from nodal function values, extra information given at $x_{b1} = -1/3$ and/or $x_{b2} = 1/3$, which do not coincide with any grid points, is also imposed. An exponential rate of convergence is achieved for all cases.

N	(f_{b1}, f_{b2})		$(df_{b1}/dx, d^2f_{b1}/dx^2)$		$(d^2f_{b1}/dx^2, d^2f_{b2}/dx^2)$	
	$N_e(df/dx)$	$N_e(d^2f/dx^2)$	$N_e(df/dx)$	$N_e(d^2f/dx^2)$	$N_e(df/dx)$	$N_e(d^2f/dx^2)$
4	8.8460e-02	7.0451e-01	1.5343e-01	1.0797e+00	1.0419e-01	7.9100e-01
6	4.5988e-03	5.5131e-02	6.7822e-03	7.6128e-02	4.1839e-03	5.1067e-02
8	1.3393e-04	2.4939e-03	2.0038e-04	3.5285e-03	8.0820e-05	1.5281e-03
10	2.5850e-06	6.9033e-05	4.0471e-06	1.0215e-04	4.0866e-06	1.0173e-04
12	3.5690e-08	1.2885e-06	5.7271e-08	1.9513e-06	3.9559e-08	1.4085e-06
14	3.7087e-10	1.7326e-08	5.7189e-10	2.5296e-08	3.7210e-10	1.7376e-08
16	3.0199e-12	1.7638e-10	4.5415e-12	2.5175e-10	2.7687e-12	1.6322e-10
18	2.4876e-14	1.6541e-12	3.0433e-14	1.7612e-12	1.4664e-14	4.0458e-13

Table 2: Example 1: Relative L_2 errors of the solution u. An exponential rate of convergence is achieved.

$N_x \times N_y$	$Ne(u)$
4×4	7.2627e-02
6×6	2.2341e-03
8×8	3.5313e-05
10×10	1.4536e-06
12×12	2.4660e-08
14×14	8.4682e-10
16×16	3.6123e-12
18×18	2.2815e-13

Table 3: Example 2: Relative L_2 errors of the solution u. An exponential rate of convergence is achieved.

$N_x \times N_y$	$Ne(u)$
6×6	3.1388e-01
8×8	1.5516e-02
10×10	7.9954e-04
12×12	3.2827e-05
14×14	7.8000e-06
16×16	2.5851e-08
18×18	5.5654e-10
20×20	9.4352e-12
22×22	1.6256e-13
24×24	6.6625e-14

Table 4: Example 3: Relative L_2 errors of the solution u. An exponential rate of convergence is achieved.

$N_x \times N_y$	$Ne(u)$
3×3	5.1493e-01
5×5	4.2323e-03
7×7	9.5121e-05
9×9	8.8346e-06
11×11	1.9794e-08
13×13	7.2909e-11
15×15	5.6142e-12
17×17	2.2127e-13

Table 5: Example 4: Relative L_2 errors of the solution u. An exponential rate of convergence is achieved.

$N_x \times N_y$/subdomain	$Ne(u)$
3×3	1.4926e-003
5×5	7.7614e-005
7×7	4.0172e-007
9×9	1.6790e-009
11×11	4.5440e-012
13×13	1.1983e-013

Table 6: Example 5: Computed values of u at point (0,0). For singular problems, the proposed method, like conventional pseudospectral methods, is only capable of yielding an algebraic rate of convergence. The values of $u(0,0)$ obtained with coarse grids are compared with the value of $u(0,0)$ with the fine grid (25×25).

$N_x \times N_y$	u	$\lvert u - u_{25\times 25}\rvert$
5×5	-0.2047754128	3.4e-004
7×7	-0.2044510201	2.4e-005
9×9	-0.2044239912	2.8e-006
11×11	-0.2044278480	9.7e-007
13×13	-0.2044267185	1.5e-007
15×15	-0.2044268922	1.9e-008
17×17	-0.2044268784	6.1e-009
19×19	-0.2044268720	2.2e-010
25×25	-0.2044268722	
		$O(h^{8.7})$

Table 7: Example 5: Computed values of u at point (0,0) by FEM. The linear-FEM results are obtained using the PDE toolbox in MATLAB.

No of nodes	No of $\triangle$ elements	u
154	270	-0.2025125196
577	1080	-0.2037602559
2233	4320	-0.2043066030
8785	17280	-0.2044072175
34849	69120	-0.2044199008
138817	276480	-0.2044248592

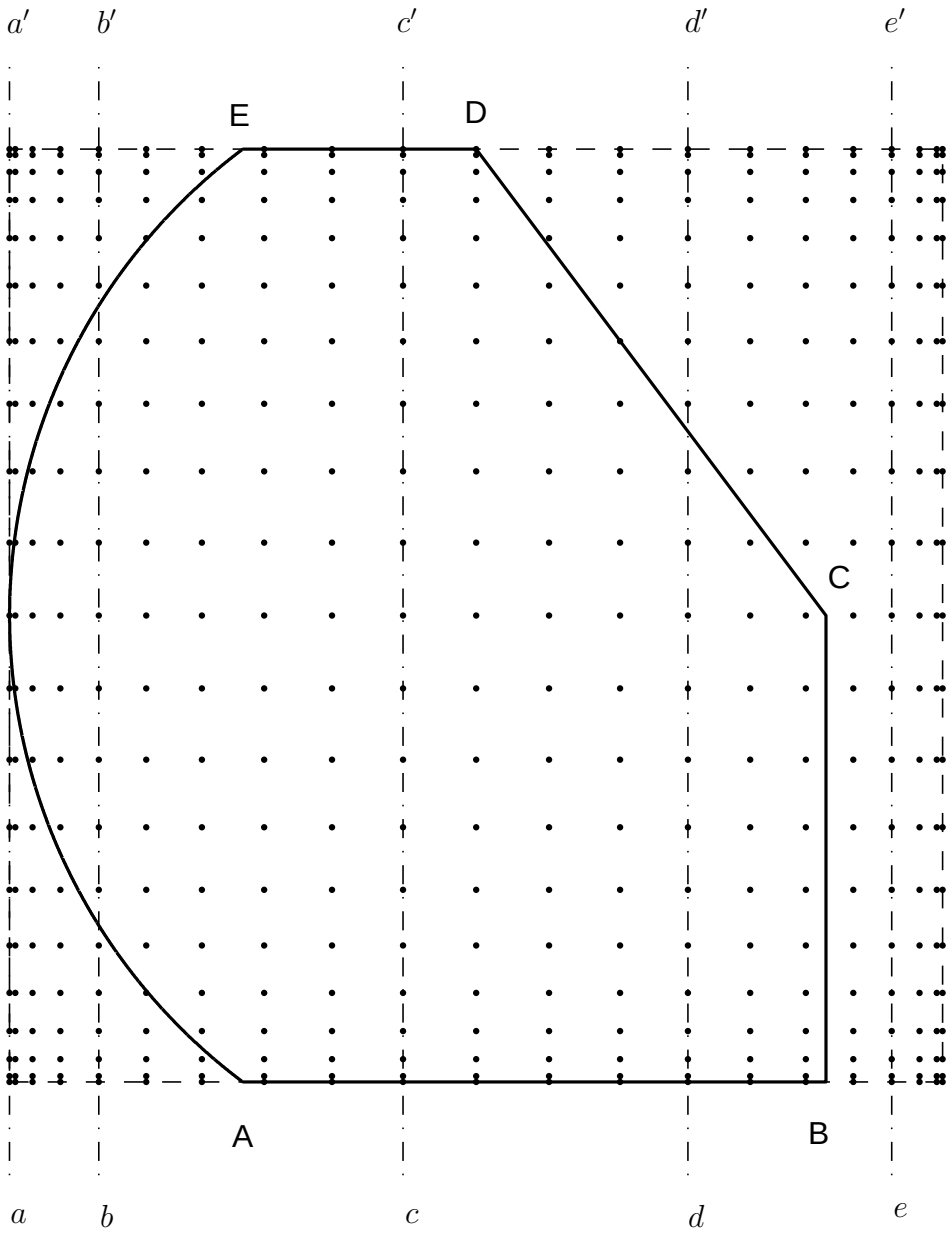

Figure 1: Extended domain. An irregular domain is embedded in a rectangular domain which is then discretized using a tensor product grid. Lines aa', bb', cc', dd' and ee' present typical cases for the approximation of derivatives of the field variable with respect to y.

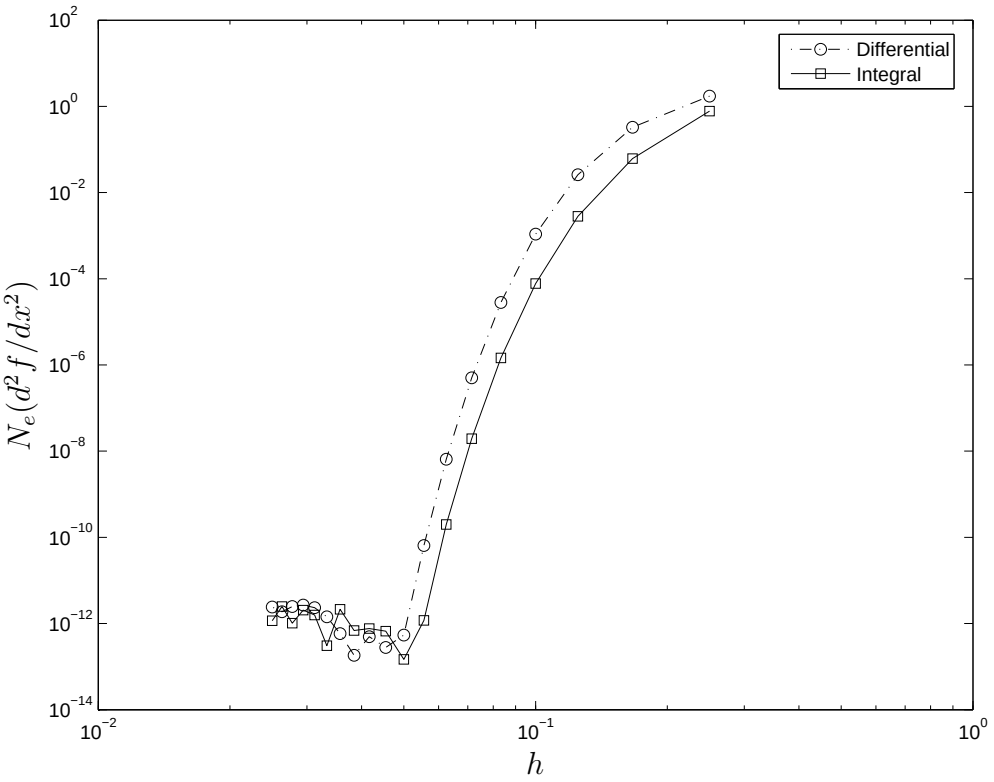

Figure 2: $f = \sin(\pi x)$, $-1 \leq x \leq 1$: Relative L_2 errors (N_e) of d^2f/dx^2. The integral approximation scheme takes into account not only the nodal function values but also the values of df/dx and d^2f/dx^2 at $x = 0$. Since the discretizations used here are chosen such that $x = 0$ is a grid point, the integral and differential formulations use the same grids. The former yields a higher level of accuracy than the latter. It is noted that h is the average spacing defined as $h = 2/N$.

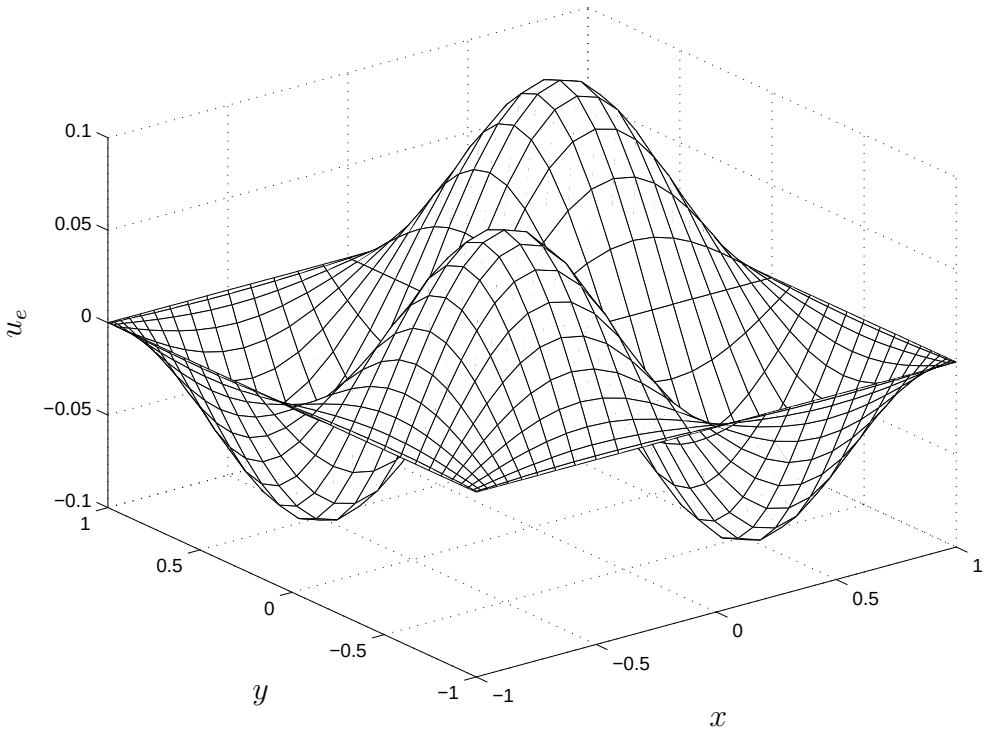

Figure 3: Example 1: Exact solution.

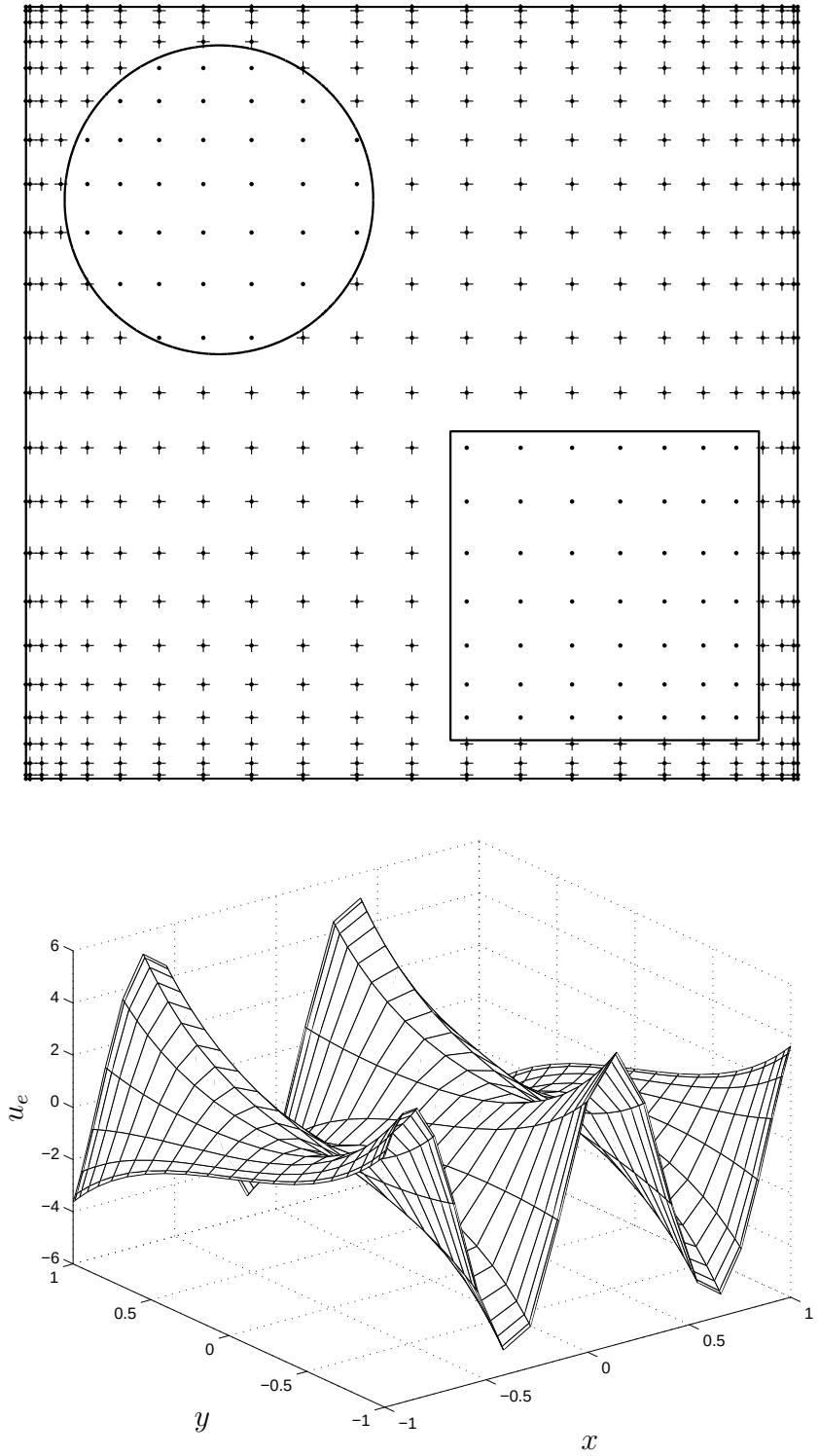

Figure 4: Example 2 (Dirichlet problem): A domain with holes and exact solution. The mark + is used to denote interior points of the actual domain Ω.

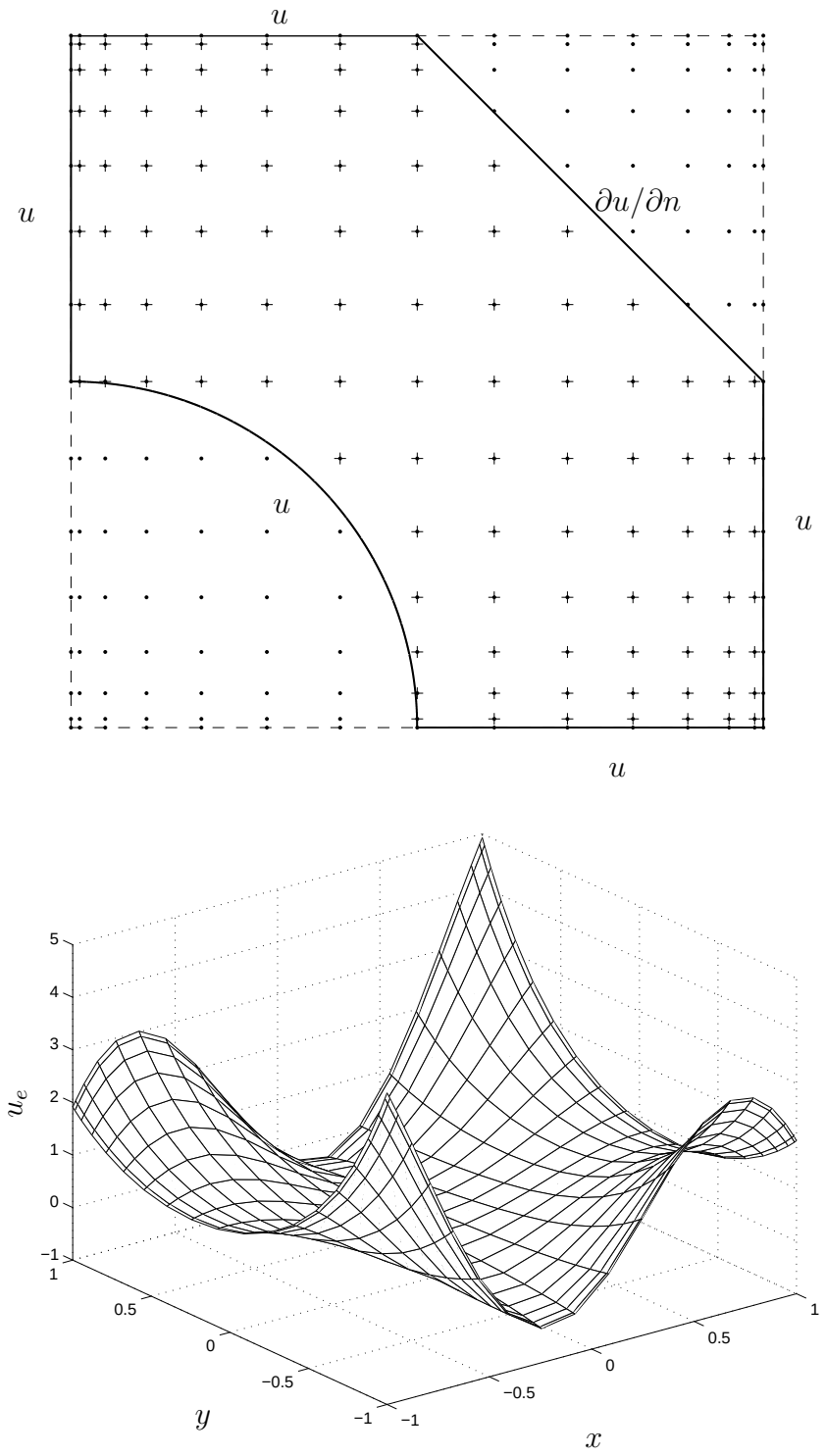

Figure 5: Example 3: Extended domain and exact solution. The side CD is specified with a Neumann boundary condition. The mark + is used to denote interior points of the actual domain Ω.

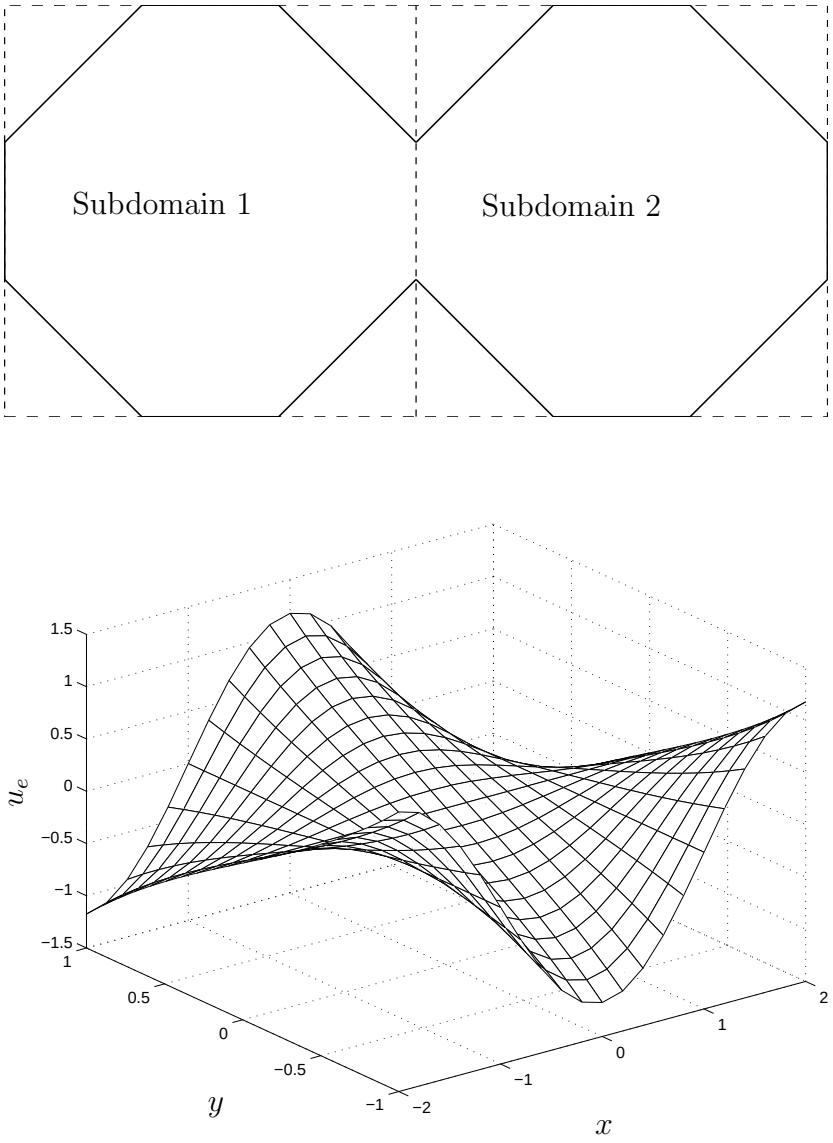

Figure 6: Example 4 (Dirichlet problem): Extended subdomains and exact solution. The problem domain is divided into 2 subdomains.

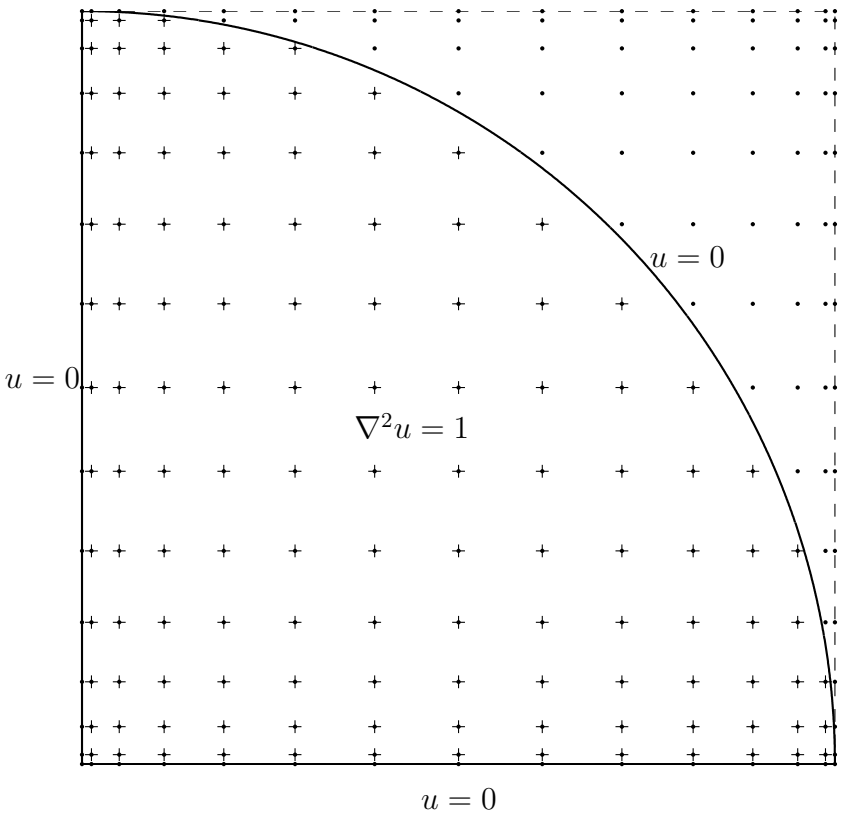

Figure 7: Example 5 (singular problem): Extended domain. The mark + is used to denote interior points of the actual domain Ω.