Journal of Scientific Computing (2020) 83:42
https://doi.org/10.1007/s10915-020-01226-9

# Accurate and Efficient Spectral Methods for Elliptic PDEs in Complex Domains

**Yiqi Gu**[1] · **Jie Shen**[1]

Received: 30 December 2019 / Accepted: 21 April 2020


**Abstract**
We develop accurate and efficient spectral methods for elliptic PDEs in complex domains using a fictitious domain approach. Two types of Petrov–Galerkin formulations with special trial and test functions are constructed, one is suitable only for the Poisson equation but with a rigorous error analysis, the other works for general elliptic equations but its analysis is not yet available. Our numerical examples demonstrate that our methods can achieve spectral convergence, i.e., the convergence rate only depends on the smoothness of the solution.

**Keywords** Spectral method · Petrov–Galerkin · Fictitious domain · Elliptic PDE · Error analysis

**Mathematics Subject Classification** 65N15 · 65N35 · 65N85

## 1 Introduction

We consider in this paper spectral methods for solving the following PDE:

$$\begin{aligned} Lu &= f \ \text{in}\,\Omega, \\ u &= h \ \text{on}\,\partial\Omega, \end{aligned} \tag{1.1}$$

where $\Omega \in \mathbb{R}^d$ is a simply connected domain, $Lu(x) := -\nabla \cdot (\beta(x)\nabla u(x)) + \alpha(x)u(x)$ is a strictly elliptic operator with $\alpha, \beta \in C(\overline{\Omega})$, $\alpha \geq 0$, $\beta \geq \beta_0 > 0$.

If $\Omega$ is a regular separable domain, spectral methods can solve the above problem in high accuracy with a computational cost comparable to the finite-elements or finite-difference methods [20,21]. However, it is still a challenge to solve the above problem in general

This work is supported in part by NSF Grant DMS-1720442 and AFOSR Grant FA9550-16-1-0102.

✉ Jie Shen
shen7@purdue.edu

Yiqi Gu
gu129@purdue.edu

[1] Department of Mathematics, Purdue University, 150 N. University Street, West Lafayette, IN 47907-2067, USA

Published online: 22 May 2020

complex domains with spectral methods, and only very limited attempts have been made in this regard. In [18], Orszag proposed the first spectral method for a class of complex domains that can be mapped to a regular domain with an explicit mapping. The idea is to transform the original PDE, usually with constant coefficients, in a complex domain to a transformed PDE with variable coefficients on a regular domain, then use an iterative method to solve the resulting dense linear system. For domains that can not be easily mapped to a regular domain, it appears that the only option for a one-domain approach using spectral methods is through a domain embedding or fictitious domain approach, which embeds the original domain into a regular one so that classical spectral methods can be applied. More precisely, one needs to choose a suitable regular domain $\widetilde{\Omega} \in \mathbb{R}^d$ s.t. $\Omega \subset \widetilde{\Omega}$, find extensions $\tilde{\alpha}$, $\tilde{\beta} \in L^\infty(\widetilde{\Omega})$ and $\tilde{f} \in L^2(\widetilde{\Omega})$ such that

$$\tilde{\alpha}(x) = \alpha(x), \ \tilde{\beta}(x) = \beta(x), \ \tilde{f}(x) = f(x) \quad \text{if } x \in \Omega,$$

and then solve the following extended problem:

$$\begin{aligned} \tilde{L}\tilde{u} &= \tilde{f} \text{ in } \widetilde{\Omega}, \\ \tilde{u} &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{1.2}$$

where $\tilde{L}\tilde{u}(x) := -\nabla \cdot \Big(\tilde{\beta}(x)\nabla\tilde{u}(x)\Big) + \tilde{\alpha}(x)\tilde{u}(x)$.

The fictitious domain approach has been well studied in the context of finite-element methods [9,13] or finite-difference methods [4,19,24] in which the data, the coefficients and the forcing function, are simply set to zero in the extended domain, but its accuracy is limited to first- or second-order due to the low regularity of the extended problem.

In order to achieve higher accuracy, there are two essential requirements. The first is to smoothly extend the coefficient and data functions from the original domain $\Omega$ to the enlarged one $\widetilde{\Omega}$. The smooth extension (or continuation) of a given function by using truncated Fourier series in 1D is well studied [1,6,15], and in higher dimensional cases, the Fourier extension is usually implemented by performing 1D extension on a fixed direction [2,3,7,16]. Note that (1.2) is not a classical boundary value problem since the solution value is prescribed on a $(d-1)$-dimensional manifold inside $\widetilde{\Omega}$. Thus, the second requirement is to setup a suitable variational formulation for the extended problem so that the extended solution is as smooth as the solution in the original domain. A first attempt in this direction is a spectral-collocation method proposed in [14], where the usual boundary condition on $\partial\widetilde{\Omega}$ is replaced by setting $\tilde{u} = 0$ at a fixed number of nodes on $\partial\Omega$, which leads to a dense linear system with constraints that are very ill conditioned so it can only be used with a small number of unknowns. In [8], a spectral-Galerkin formulation with Lagrange multipliers is presented, and the boundary conditions are manipulated by using internal forcing functions which are compactly supported inside the fictitious domain. This method is improved in [17] by replacing the Dirac delta function basis for the Lagrange multipliers in the physical space with Fourier basis functions in the frequency space with improved accuracy.

The aim of this paper is to construct accurate and efficient spectral methods for solving the extended problem (1.2). We assume a smooth extension for a given function is always available, through for instance Fourier-extension [15], and concentrate on developing proper variational formulations and corresponding spectral methods for (1.2). More precisely, we propose two spectral-Petrov–Galerkin approaches with proper test and trial spaces, investigate their well posedness and error analysis, and develop effective algorithms for solving the ill-conditioned linear systems resulting from the spectral-Petrov–Galerkin approaches.

The organization of this paper is as follows. In Sect. 2, we present the first spectral-Petrov–Galerkin method for the extended problem (1.2), and carry out rigorous analysis and

error estimates for the special case of Poisson equation. In Sect. 3, we present the second spectral-Petrov–Galerkin method which is suitable for general elliptic equations. In Sect. 4, we develop a fast and stable algorithm for solving the linear systems resulting from the two spectral-Petrov–Galerkin methods. We present ample numerical results in Sect. 5 to validate our algorithms, followed by some concluding remarks in Sect. 6. In the following, we will simply denote $\tilde{\alpha}$, $\tilde{\beta}$, $\tilde{u}$ and $\tilde{f}$ by $\alpha$, $\beta$, $u$ and $f$ without ambiguity.

## 2 The First Method

We restrict our attention to the case of $\alpha = 0$, i.e. $Lu(x) = -\nabla \cdot (\beta(x)\nabla u(x))$. Let the trial space $X$ and test space $Y$ be defined as

$$X := \{u \in H^2(\widetilde{\Omega}) : u = 0 \text{ on } \partial\Omega\}, \quad Y := L^2(\widetilde{\Omega}). \tag{2.1}$$

$X$ and $Y$ are both Banach spaces with

$$\|u\|_X := \left(\int_{\widetilde{\Omega}} |\Delta u|^2\right)^{\frac{1}{2}}, \forall u \in X, \tag{2.2}$$

$$\|v\|_Y := \left(\int_{\widetilde{\Omega}} |v|^2\right)^{\frac{1}{2}}, \forall v \in Y. \tag{2.3}$$

It is clear that the norm defined in (2.2) is indeed a norm, since $\|u\|_X = 0$ implies $u$ is harmonic, so by the maximum principle, we have $u = 0 \in \Omega$, and by unique continuation of harmonic function, we have $u = 0 \in \widetilde{\Omega}$ .

Then the weak formulation of problem (1.2) is to find $u \in X$ s.t.

$$a_1(u, v) := -\int_{\widetilde{\Omega}} \nabla \cdot (\beta \nabla u)\, v = \int_{\widetilde{\Omega}} f v, \quad \forall v \in Y. \tag{2.4}$$

### 2.1 Well-Posedness

The well-posedness of (2.4) can be shown when $\beta(x)$ is constant, namely, the Poisson problem. Without of loss of generality, we suppose $\beta(x) = 1$, then it is trivial to see $a_1(\cdot, \cdot)$ is a continuous bilinear form on $X \times Y$. Also we need the following lemma.

**Lemma 2.1** *Suppose $\Omega$ satisfies an interior cone condition [12, p.27]. Then under the definition in (2.2), (2.3) and (2.4) with $\beta(x) = 1$, we have*

$$\inf_{u\in X} \sup_{v\in Y} \frac{a_1(u, v)}{\|u\|_X \|v\|_Y} \geq 1; \tag{2.5}$$

*and*

$$\sup_{0\neq u\in X} a_1(u, v) > 0, \quad \forall 0 \neq v \in Y. \tag{2.6}$$

*Specifically, (2.5) and (2.6) holds if $\Omega$ is a $C^1$ domain or a polygon.*

***Proof*** Given $u \in X$, we have $\Delta u \in Y$, so

$$\sup_{v\in Y} \frac{a_1(u, v)}{\|u\|_X \|v\|_Y} \geq \frac{a_1(u, \Delta u)}{\|u\|_X \|\Delta u\|_Y} = 1. \tag{2.7}$$

Next, for any $0 \neq v \in Y$, the Dirichlet problem

$$\begin{aligned} \Delta u &= v \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{2.8}$$

admits a solution $u \in H^2(\Omega)$, denoted by $u_1$. On the other hand, since $\widetilde{\Omega}\backslash\overline{\Omega}$ satisfies an exterior cone condition, the Dirichlet problem

$$\begin{aligned} \Delta u &= v \text{ in } \widetilde{\Omega}\backslash\overline{\Omega}, \\ u &= 0 \text{ on } \partial\Omega \cup \partial\widetilde{\Omega}, \end{aligned} \tag{2.9}$$

admits a solution in $H^2(\widetilde{\Omega}\backslash\overline{\Omega})$, denoted by $u_2$ [12, Theorem 2.14]. Let

$$u = \begin{cases} u_1 & \text{in } \Omega \\ u_2 & \text{in } \widetilde{\Omega}\backslash\overline{\Omega} \\ 0 & \text{on } \partial\Omega \cup \partial\widetilde{\Omega} \end{cases}, \tag{2.10}$$

then $u \in X$, and

$$\begin{aligned} a_1(u, v) = \int_{\widetilde{\Omega}} (\Delta u)v &= \int_{\Omega} (\Delta u_1)v + \int_{\widetilde{\Omega}\backslash\overline{\Omega}} (\Delta u_2)v \\ &= \int_{\Omega} v^2 + \int_{\widetilde{\Omega}\backslash\overline{\Omega}} v^2 = \|v\|_Y^2 > 0. \end{aligned} \tag{2.11}$$

□

We then derive from the Banach-Necăs-Babuška theorem [5, p.112] that

**Theorem 2.2** *Under the hypothesis of Lemma 2.1, the problem (2.4) admits a unique solution $u$ satisfying*

$$\|u\|_X \leq \|f\|_Y, \ \forall f \in Y. \tag{2.12}$$

### 2.2 A Non-Conforming Petrov–Galerkin Spectral Method

Let $N$ be an odd integer, and $P_N$ the polynomial space of degree no greater than $N$. Let $\xi_i : C(\partial\Omega) \to \mathbb{R}$ with $i = 1, \cdots, 2N+2$ represents $2N+2$ independent constraints placed on $u$ to approximate the original boundary condition $u = 0$ on $\partial\Omega$ in (2.1). This is similar to the boundary element used in boundary integral method ([10]). For example, one simple choice for $\xi_i$ is

$$\xi_i(u_N) := u_N(z_i), \ i = 1, \cdots, 2N+2, \tag{2.13}$$

where $\{z_i\}$ are a set of prescribed points on $\partial\Omega$. Another choice is

$$\xi_i(u_N) := \int_{\partial\Omega} u_N \chi_i \mathrm{d}s, \ i = 1, \cdots, 2N+2, \tag{2.14}$$

where $\{\chi_i\}$ are a set of linearly independent functions defined on $\partial\Omega$, and they play a similar role to the Lagrange multipliers (see [8]).

### 2.2.1 Weak Formulation and Wellposedness

To simplify the presentation, we shall consider only the 2-D case althought extension to 3D is straightforward. We also assume that the problem domain $\Omega$ in (2.4) is scaled so that it can be enclosed in $\widetilde{\Omega} = (-1, 1) \times (-1, 1)$. We define

$$X_N := \{u_N \in P_N \times P_N,\ \xi_i(u_N) = 0,\ i = 1, \cdots, 2N + 2\}, \tag{2.15}$$

and

$$Y_N := \text{span}\{\Delta(x^i y^j)\}_{i,j=0}^{N}. \tag{2.16}$$

Note that $X_N$ is not a subspace of $X$. It is clear that

$$\dim(X_N) = (N + 1)^2 - (2N + 2) = N^2 - 1. \tag{2.17}$$

**Lemma 2.3** $\dim(Y_N) = N^2 - 1$ *if* $N$ *is odd and* $\dim(Y_N) = N^2$ *if* $N$ *is even.*

***Proof*** We use the following table $T$ to describe $\{\Delta(x^i y^j)\}_{i,j=0}^{N}$:

| | 0 | 1 | 2 | 3 | $\cdots$ | N |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | $x$ | $\cdots$ | $x^{N-2}$ |
| 1 | 0 | 0 | $y$ | $xy$ | $\cdots$ | $x^{N-2}y$ |
| 2 | 1 | $x$ | $(x^2, y^2)$ | $(x^3, xy^2)$ | $\cdots$ | $(x^N, x^{N-2}y^2)$ |
| 3 | $y$ | $xy$ | $(x^2y, y^3)$ | $(x^3y, xy^3)$ | $\cdots$ | $(x^Ny, x^{N-2}y^3)$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| N | $y^{N-2}$ | $xy^{N-2}$ | $(x^2y^{N-2}, y^N)$ | $(x^3y^{N-2}, xy^N)$ | $\cdots$ | $(x^Ny^{N-2}, x^{N-2}y^N)$ |

In the above table, $T(j, i)$ is filled by $\Delta(x^i y^j)$ without coefficients, and the parenthesis $(\cdot, \cdot)$ means the linear combination of the two terms with nonzero coefficients. From the table it is straightforward to see that, if $N$ is odd,

$$T(0, i) \in \text{span}\{T(2, i - 2), T(4, i - 4), \cdots, T(i - 1, 1)\} \tag{2.18}$$

and

$$T(1, i) \in \text{span}\{T(3, i - 2), T(5, i - 4), \cdots, T(i, 1)\} \tag{2.19}$$

for $i = 2, \cdots, N$. Hence by removing the first two rows of $T$, the reduced table $\{T(i, j)\}_{i=0,j=2}^{N}$ is still a spanning set of $Y_N$.

Next, we show that $\{T(i, j)\}_{i=0,j=2}^{N}$ is linearly independent. To this end, note for $i = -(N-2), -(N-1), \cdots, 2N-2$, each anti-diagonal $\{T(N, i), T(N-1, i+1), \cdots, T(3, i+N-3)\}$ consists of all the entries of order $N - 2 + i$ in the reduced table (ignore the entries with indices which are negative or greater than $N + 1$), so distinct anti-diagonals are linearly independent. Also, every anti-diagonal itself is linearly independent since each entry in it has a special term that cannot be obtained by linear combination of other entries. Therefore, $\dim(Y_N)$ is equal to the number of entries in $\{T(i, j)\}_{i=0,j=2}^{N}$, which is $(N + 1)(N - 1) = N^2 - 1$.

The case of $N$ even is essentially the same as the odd case except for one entry in (2.19), that is

$$T(1, N) \notin \text{span}\{T(3, N - 2), T(5, N - 4), \cdots, T(N - 1, 2)\}. \tag{2.20}$$

Hence $T(1, N) \cup \{T(i, j)\}_{i=3,j=1}^{N}$ form a basis for $Y_N$ and $\dim(Y_N) = N^2$. □

Note that $\dim(X_N) = \dim(Y_N)$ for odd $N$. Since $X_N$ is not a subspace of $X$, we define

$$\|u_N\|_{X_N} := \left(\int_{\widetilde{\Omega}} |\Delta u_N|^2\right)^{\frac{1}{2}}, \tag{2.21}$$

which is consistent with (2.2), and is indeed a norm, as long as $\{\xi_i\}_{i=1}^{2N+2}$ in (2.15) are specifically chosen s.t. $\Delta : X_N \to Y_N$ has a trivial nullspace (this can always be satisfied in numerical implementation, and we assume this hypothesis holds in the remaining context).

Let $I_N : L^2(\widetilde{\Omega}) \to P_N \times P_N$ be the 2D tensorial polynomial interpolation operator at the Legendre-Gauss-Lobatto points. Our Petrov–Galerkin spectral method for (2.4) is: find $u_N \in X_N$ s.t.

$$a_1(u_N, v_N) = \int_{\widetilde{\Omega}} I_N f v_N, \quad \forall v_N \in Y_N. \tag{2.22}$$

To study the well-posedness of (2.22), we need

**Lemma 2.4** *Under the definition in (2.15),(2.16) and (2.4) with $\beta(x) = 1$, we have*

$$\inf_{u_N \in X_N} \sup_{v_N \in Y_N} \frac{a_1(u_N, v_N)}{\|u_N\|_{X_N}\|v_N\|_{Y_N}} \geq 1, \tag{2.23}$$

*and*

$$\sup_{u_N \in X_N} |a_1(u_N, v_N)| > 0, \quad \forall 0 \neq v_N \in Y_N. \tag{2.24}$$

***Proof*** (2.23) can be proven by the exactly same argument as in the proof of Lemma 2.1. And (2.24) follows the fact $\dim(X_N) = \dim(Y_N)$ and [11, Proposition 2.21]. □

Finally, by Lemma 2.4 we obtain

**Theorem 2.5** *The approximate problem (2.22) admits a unique solution $u_N$, which satisfies the a priori estimate*

$$\|u_N\|_{X_N} \leq \|I_N f\|_{L^2(\widetilde{\Omega})}. \tag{2.25}$$

### 2.2.2 Error Estimates

We first consider the approximation property of $X_N$ to $X$.

**Lemma 2.6** *For any odd integer $N$,*

$$P_{\frac{N-3}{2}} \times P_{\frac{N-3}{2}} \subset Y_N. \tag{2.26}$$

***Proof*** By virtue of the proof of Theorem 3.1, the following reduced table consists of a basis for $Y_N$ if $N$ is odd.

| | 0 | 1 | 2 | 3 | ⋯ | N |
|---|---|---|---|---|---|---|
| 2 | 1 | $x$ | $(x^2, y^2)$ | $(x^3, xy^2)$ | ⋯ | $(x^N, x^{N-2}y^2)$ |
| 3 | $y$ | $xy$ | $(x^2y, y^3)$ | $(x^3y, xy^3)$ | ⋯ | $(x^Ny, x^{N-2}y^3)$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| N | $y^{N-2}$ | $xy^{N-2}$ | $(x^2y^{N-2}, y^N)$ | $(x^3y^{N-2}, xy^N)$ | ⋯ | $(x^Ny^{N-2}, x^{N-2}y^N)$ |

Denote $T_k := \{T(k+2,0), T(k+1,1), T(k,2), \cdots, T(2,k)\}$, for $k = 2, \cdots, N-3$, which consists exactly of $k+1$ independent entries of order $k$. Hence $T_k$ spans the space of 2D monomial of degree $k$. Therefore for $i \le \frac{N-3}{2}$, $j \le \frac{N-3}{2}$, $x^i y^j \in \text{span} T_{i+j}$, which implies $P_{\frac{N-3}{2}} \times P_{\frac{N-3}{2}} \subset Y_N$. □

Next we recall the error estimate for 2D tensorial polynomial interpolation, which is given by

**Lemma 2.7** (cf. [22]) *Suppose the interpolation nodes for $I_N : L^2(\widetilde{\Omega}) \to P_N \times P_N$ are the roots of the Legendre polynomial of degree $N$ for each variable, and let $u \in H^r(\widetilde{\Omega})$ with $2 \le r \le N+1$, then*

$$\|I_N u - u\|_{L^2(\widetilde{\Omega})} \le c\sqrt{\frac{(N-r+1)!}{N!}}(N+r)^{-\frac{r+1}{2}}|u|_{H^r(\widetilde{\Omega})} \tag{2.27}$$

*with a constant $c$. In particular, for fixed $r$, we have that for $N$ sufficiently large,*

$$\|I_N u - u\|_{L^2(\widetilde{\Omega})} \le cN^{-r}|u|_{H^r(\widetilde{\Omega})}. \tag{2.28}$$

We can then derive the following result:

**Theorem 2.8** *Assuming $u \in X \cap H^r(\widetilde{\Omega})$ with $r \ge 4$, we have*

$$\inf_{u_N \in X_N} \|\Delta(u - u_N)\|_{L^2(\widetilde{\Omega})} \le \left(\frac{N-3}{2}\right)^{-(r-2)} |u|_{H^r(\widetilde{\Omega})}. \tag{2.29}$$

***Proof*** Let $q := I_{\frac{N-3}{2}}(\Delta u) \in P_{\frac{N-3}{2}} \times P_{\frac{N-3}{2}} \subset Y_N$ by Lemma 2.6. Note the linear problem

$$\text{find } w_N \in X_N \text{ s.t. } \Delta w_N = q, \tag{2.30}$$

admits a unique solution since $\dim(X_N) = \dim(Y_N)$ and $\Delta$ has a trivial nullspace. Therefore

$$\begin{aligned}\inf_{u_N \in X_N} \|\Delta(u - u_N)\|_{L^2(\widetilde{\Omega})} &\le \|\Delta u - \Delta w_N\|_{L^2(\widetilde{\Omega})} = \|\Delta u - I_{\frac{N-3}{2}}(\Delta u)\|_{L^2(\widetilde{\Omega})} \\ &\le \left(\frac{N-3}{2}\right)^{-(r-2)} |\Delta u|_{H^{-(r-2)}(\widetilde{\Omega})} \le \left(\frac{N-3}{2}\right)^{-(r-2)} |u|_{H^r(\widetilde{\Omega})}.\end{aligned} \tag{2.31}$$

□

Finally, we have the following error estimate for (2.22):

**Theorem 2.9** *Let $\beta(x) = 1$ and $f \in H^s(\widetilde{\Omega})$ for some $s \ge 2$. Suppose the solution $u$ of (2.4) satisfies the regularity hypothesis $u \in X \cap H^r(\widetilde{\Omega})$ for some $r \ge 4$, then the solution $u_N$ of (2.22) satisfies*

$$\|u - u_N\|_X \le c\left(\left(\frac{N-3}{2}\right)^{-(r-2)} |u|_{H^r(\widetilde{\Omega})} + N^{-s}|f|_{H^s(\widetilde{\Omega})}\right), \tag{2.32}$$

*for some constant $c > 0$.*

***Proof*** Thanks to the discrete inf-sup condition (2.23) and the continuity of $a(\cdot,\cdot)$ on $(X + X_N) \times Y$, the problem (2.22) satisfies the hypothesis of the Second Strang Lemma ([23]), which gives

$$\begin{aligned}\|\Delta(u - u_N)\|_{L^2(\widetilde{\Omega})} \le &(1 + \|a\|) \inf_{u_N \in X_N} \|\Delta u - \Delta u_N\|_{L^2(\widetilde{\Omega})} \\ &+ \sup_{v_N \in Y_N} \frac{|\int_{\widetilde{\Omega}} I_N f v_N - a(u, v_N)|}{\|v_N\|_{Y_N}}.\end{aligned} \tag{2.33}$$

For $f \in H^s(\widetilde{\Omega})$, we have by (2.28) that

$$\begin{aligned} |\int_{\widetilde{\Omega}} I_N f v_N - a(u, v_N)| &= |\int_{\widetilde{\Omega}} I_N f v_N - \int_{\widetilde{\Omega}} f v_N| \\ &\le \|I_N f - f\|_{L^2(\widetilde{\Omega})} \|v_N\|_{L^2(\widetilde{\Omega})} \le c N^{-s} |f|_{H^s(\widetilde{\Omega})} \|v_N\|_{L^2(\widetilde{\Omega})}, \end{aligned} \tag{2.34}$$

for some constant $c > 0$. Therefore, the inequality (2.32) follows from (2.29) and (2.33). □

## 3 The Second Method

Although the method presented in the last section can be applied to more general elliptic equations with non-constant coefficients, it is only mathematically justified for $\alpha(x) \equiv 0$ and $\beta(x) \equiv 1$. In fact, numerical evidence indicates that the convergence rate deteriorates if the method is applied to the problem (1.1) with $\alpha \neq 0$. Therefore, we shall present another Petrov–Galerkin method which does not have this drawback.

### 3.1 Weak Formulation

In this method, we set the trial and test spaces to be

$$X := \{u \in H^1(\widetilde{\Omega}),\ \mathrm{tr}(u) = 0 \text{ on } \partial\Omega\},\ \ \|u\|_X := \left(\int_{\widetilde{\Omega}} u^2 + |\nabla u|^2\right)^{\frac{1}{2}}, \tag{3.1}$$

$$Y := H_0^1(\widetilde{\Omega}),\ \ \|v\|_Y := \left(\int_{\widetilde{\Omega}} v^2 + |\nabla v|^2\right)^{\frac{1}{2}}. \tag{3.2}$$

Here $X$ differs from $Y$, as the functions in $X$ vanish on the interior boundary $\partial\Omega$ rather than the outer boundary $\partial\widetilde{\Omega}$.

Then a weak formulation of problem (1.2) is: find $u \in X$ s.t.

$$a_2(u, v) := \int_{\widetilde{\Omega}} \beta \nabla u \cdot \nabla v + \alpha(x) u v = \int_{\widetilde{\Omega}} f v,\ \ \forall v \in Y. \tag{3.3}$$

### 3.2 Spectral Approximation

We set

$$X_N := \{u_N \in P_N \times P_N,\ \xi_i(u_N) = 0,\ i = 1, \cdots, 4N\}, \tag{3.4}$$

and

$$Y_N := P_N^0 \times P_N^0, \tag{3.5}$$

where $P_N^0 := \{p \in P_N,\ p(\pm 1) = 0\}$. The sampling points $\{\xi_i\}$ are still distributed on $\partial\Omega$ as in the first method but the number here is increased to $4N$ to force $\dim(X_N) = \dim(Y_N) = (N-1)^2$.

Our second Petrov–Galerkin method is: find $u_N \in X_N$ s.t.

$$a_2(u_N, v_N) = \int_{\widetilde{\Omega}} I_N f v_N,\ \forall v_N \in Y_N, \tag{3.6}$$

where $a_2(\cdot, \cdot)$ is defined in (3.3).

Unfortunately, we are unable to provide an analysis for the above method, but our numerical experiments show theabove method (3.6) works better than the first method in Sect. 2 for problem (1.1) with a nonzero $\alpha(x)$ (see Sect. 5).

## 4 Efficient Numerical Implementation

We describe in this section how the two spectral methods presented in previous sections can be efficiently implemented.

### 4.1 Derivation of the Linear System

We shall use Legendre polynomials to construct basis functions for $X_N$ and $Y_N$. Recall the Legendre polynomials $\{L_k\}_{k=0}^N$ form an orthogonal basis for $P_N$ satisfying

$$\int_{-1}^{1} L_n(x)L_m(x)\mathrm{d}x = \frac{2}{2n+1}\delta_{mn}. \tag{4.1}$$

Hence, we define $\overline{L}_n(x)$ be the polynomial that has a second derivative equal to $L_{n-2}(x)$ for $n \geq 2$, namely

$$\overline{L}_0(x) = 1, \quad \overline{L}_1(x) = x, \quad \overline{L}_2(x) = x^2/2, \quad \overline{L}_3(x) = x^3/6, \tag{4.2}$$

and

$$\begin{aligned}\overline{L}_n(x) &:= \int_{-1}^{x}\int_{-1}^{t} L_n(s)\mathrm{d}s\mathrm{d}t \\ &= \frac{1}{(2n-3)(2n-5)}L_{n-4}(x) - \frac{2}{(2n-1)(2n-5)}L_{n-2}(x) + \frac{1}{(2n-1)(2n-3)}L_n(x),\end{aligned} \tag{4.3}$$

for $n \geq 4$. It can be verified that $\{\overline{L}_n(x)\}_{n=0}^N$ form a basis for $P_N$ and

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\overline{L}_n(x) = L_{n-2}(x) \text{ for } n \geq 2. \tag{4.4}$$

We start with basis functions for $Y_N$. For the first spectral method described in Sect. 2,

$$Y_N = \mathrm{span}\{\widetilde{L}_{mn}\}_{m=0,n=2}^N, \text{ with } \widetilde{L}_{mn} = L_{m-2}(x)L_n(y) + L_m(x)L_{n-2}(y), \tag{4.5}$$

where $L_{-2} = L_{-1} := 0$. And for the second method described in Sect. 3,

$$Y_N = \mathrm{span}\{\widetilde{L}_{mn}\}_{m,n=0}^{N-2}, \text{ with } \widetilde{L}_{mn} = \widetilde{L}_m(x)\widetilde{L}_n(y), \tag{4.6}$$

where $\widetilde{L}_m(t) := L_{m+2}(t) - L_m(t) \in P_N^0$. Generally, we denote $Y_N = \mathrm{span}\{\psi_j\}_{j=1}^{M'}$, where $M' = N^2 - 1$ for the first one and $M' = (N-1)^2$ for the second one is the dimension of $Y_N$ and $X_N$.

Next we consider how to construct basis functions $\{\phi_i\}_{i=1}^{M'}$ for $X_N$. Due to complexity of domain boundary $\partial\Omega$ and the prescribed constraints $\{\xi_k(u_N) = 0\}$ in the definition of $X_N$, it is not possible to write these basis functions in a closed form, so we write

$$\phi_i = \sum_{s,t=0}^{N} d_{st}^i \overline{L}_s(x)\overline{L}_t(y) \text{ such that } \xi_k(\phi_i) = 0 \quad \forall k = 1, \cdots, M, \tag{4.7}$$

where $M = 2N + 2$ for the first one and $M = 4N$ for the second one is the number of sampling points on the boundary of $\Omega$.

For each $\phi_i$, the $M$ constraints $\{\xi_k(\phi_i) = 0\}_{k=1}^{M}$ defined in (2.15) can be written in a matrix-vector form:

$$\boldsymbol{B}\boldsymbol{d}^i = 0, \tag{4.8}$$

where $\boldsymbol{B} \in \mathbb{R}^{M\times(M+M')}$, independent of $i$, with the $k$-th row corresponding to the $k$-th constraint $\{\xi_k(\phi_j) = 0\}_{j=1}^{M'}$, and

$$\boldsymbol{d}^i := \left[d_{00}^i\, d_{01}^i\, \cdots\, d_{NN}^i\right]^T, \tag{4.9}$$

which is a long vector consisting all the coefficients of $\phi_i$ in (4.7) lexicographically. We observe that $\boldsymbol{B}$ is determined by $\Omega$, $\widetilde{\Omega}$ and the choice for $\xi_k$, and is independent of the PDE operator $L$ and the data $f$.

It is now evident that $\{\phi_i\}_{i=1}^{M'}$ can be constructed by finding a basis for null($\boldsymbol{B}$), since the basis contains exactly $M'$ vectors, each of which corresponds to one element of $\{\phi_i\}_{i=1}^{M'}$. More precisely, let

$$\boldsymbol{D} := \left[\boldsymbol{d}^1\, \boldsymbol{d}^2\, \cdots\, \boldsymbol{d}^{M'}\right] \in \mathbb{R}^{(M+M')\times M'} \tag{4.10}$$

with linearly independent columns such that $\boldsymbol{B}\boldsymbol{D} = 0$, and denote

$$\overline{\boldsymbol{L}}(x, y) := \left[\overline{L}_0(x)\overline{L}_0(y)\, \overline{L}_0(x)\overline{L}_1(y)\, \cdots\, \overline{L}_N(x)\overline{L}_N(y)\right], \tag{4.11}$$

then formally we have

$$[\phi_1\, \phi_2\, \cdots\, \phi_{M'}] = \overline{\boldsymbol{L}}(x, y)\boldsymbol{D}. \tag{4.12}$$

Writing $u_N = \sum_{i=1}^{M'} \tilde{u}_i\phi_i$, then (2.22) (or (3.6)) leads to the following linear system,

$$\sum_{i=1}^{M'} a(\phi_i, \psi_j)\tilde{u}_i = \int_{\widetilde{\Omega}} I_N f\psi_j := f_j, \text{ for } j = 1, \cdots, M'. \tag{4.13}$$

Denoting

$$\boldsymbol{A} := \left[a\left(\overline{L}_s(x)\overline{L}_t(y), \psi_j\right)\right] \in \mathbb{R}^{M'\times(M+M')} \tag{4.14}$$

with row indices $j = 1, \cdots, M'$ and column indices $s, t = 0, \cdots, N$, and with the notation in (4.12), we can rewrite (4.13) in the matrix form as

$$\boldsymbol{A}\boldsymbol{D}\boldsymbol{u} = \boldsymbol{f}, \tag{4.15}$$

where

$$\boldsymbol{u} := \left[\tilde{u}_1\, \tilde{u}_2\, \cdots\, \tilde{u}_{M'}\right]^T, \quad \boldsymbol{f} := \left[\tilde{f}_1\, \tilde{f}_2\, \cdots\, \tilde{f}_{M'}\right]^T.$$

Note that for any given point $(x_p, y_p)$ at which the solution is evaluated,

$$u_N(x_p, y_p) = [\phi_1\, \phi_2\, \cdots\, \phi_{M'}]\,|_{(x_p,y_p)}\boldsymbol{u} = \widetilde{\boldsymbol{L}}(x_p, y_p)\boldsymbol{D}\boldsymbol{u} := \widetilde{\boldsymbol{L}}(x_p, y_p)\boldsymbol{y}, \tag{4.16}$$

which means the evaluation of $u_N$ only depends on $\widetilde{\boldsymbol{L}}(x_p, y_p)$ and $\boldsymbol{y}$, Hence, instead of solving (4.15) for $\boldsymbol{u}$ explicitly, we can solve

$$\boldsymbol{A}\boldsymbol{y} = \boldsymbol{f}, \tag{4.17}$$

directly.

Note that (4.17) has $M'$ equations for $M + M'$ unknowns. The remaining equations are from the boundary constraints

$$\boldsymbol{B}\boldsymbol{y} = 0. \tag{4.18}$$

Hence, the final linear system to be solved is

$$\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{y} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix}. \tag{4.19}$$

For problems with non-homogeneous boundary condition $u|_{\partial\Omega} = h$, it suffices to let

$$\boldsymbol{h} = \left[ \int_{\partial\Omega} h\chi_1 \mathrm{d}s \int_{\partial\Omega} h\chi_2 \mathrm{d}s \cdots \int_{\partial\Omega} h\chi_M \mathrm{d}s \right]^T, \tag{4.20}$$

and replace the right vector in (4.19) by

$$\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{y} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{h} \end{bmatrix}. \tag{4.21}$$

### 4.2 Fast and Robust Algorithm for Solving the Linear System

Unfortunately, it is numerically observed that (4.21) is very ill-conditioned so a direct solver is not feasible. Note that the upper part $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{f}$ is the approximation to the PDE $Lu = f$, while the lower part $\boldsymbol{B}\boldsymbol{y} = h$ describes the boundary constraints. The idea is to solve the upper part accurately and relax the accuracy requirement for the lower part. More precisely, we aim to reduce the residue of $\boldsymbol{B}\boldsymbol{y} = \boldsymbol{h}$ as much as possible subject to $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{f}$. A straightforward approach is to solve the least square problem

$$\min_{\boldsymbol{y} \in \boldsymbol{y}_s + \boldsymbol{Y}_K} \|\boldsymbol{h} - \boldsymbol{B}\boldsymbol{y}\|_2, \tag{4.22}$$

where $\boldsymbol{y}_s$ is a particular solution of $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{f}$ and $\boldsymbol{Y}_K$ is a $K$-dimensional subspace of null($\boldsymbol{A}$) with $K \leq M$. Note that if $K = M$, (4.22) is equivalent to (4.21). Hence, to avoid the ill-conditioning, $K$ should not be too close to $M$ in practical computation.

For a fixed $K < M$, we first find a particular solution $\boldsymbol{y}_s$ of $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{f}$ by letting $\boldsymbol{y}_s$ being in the row space of $\boldsymbol{A}$, i.e.

$$\boldsymbol{y}_s = \boldsymbol{A}^T \boldsymbol{x}. \tag{4.23}$$

Hence if follows

$$\left(\boldsymbol{A}\boldsymbol{A}^T\right) \boldsymbol{x} = \boldsymbol{f}, \tag{4.24}$$

where $\boldsymbol{A}\boldsymbol{A}^T$ is symmetric positive-definite, so the above can be easily solved.

Thanks to the orthogonality of the Legendre polynomials, $\boldsymbol{A}$ is a sparse block band matrix with 4 block bands (the structure of $\boldsymbol{A}$ for $N = 15$ is shown in Fig. 1). So we can find easily an orthonormal set $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_K$ in null($\boldsymbol{A}$). Denote

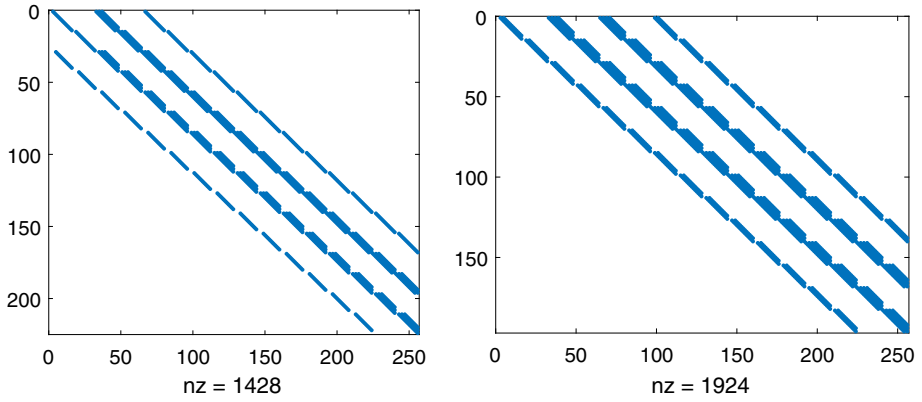
$$\boldsymbol{Y}_K = [\boldsymbol{y}_1\ \boldsymbol{y}_2\ \cdots\ \boldsymbol{y}_K] \in \mathbb{R}^{(M+M') \times K}, \tag{4.25}$$

then (4.22) can be rewritten as

$$\min_{z_K \in \mathbb{R}^K} \|\boldsymbol{h} - \boldsymbol{B}(\boldsymbol{Y}_K z_K + \boldsymbol{y}_s))\|_2. \tag{4.26}$$

Therefore it suffices to compute the least square solution $z_K$ to (4.26) so that the solution to (4.22) is given by

$$\boldsymbol{y} = \boldsymbol{Y}_K z_K + \boldsymbol{y}_s. \tag{4.27}$$




**(a)** the first method **(b)** the second method

**Fig. 1** The structure of $\boldsymbol{A}$ for N=15 (around 50,000 total entries)

The choice of $K$ is of critical importance, since large $K$ may cause a large condition number, and small $K$ may lead to large errors for the boundary constraints $\boldsymbol{By} = 0$. Therefore, we employ an adaptive procedure to choose $K$ which better balances the ill-conditioning and the errors for the boundary constraints $\boldsymbol{By} = 0$.

We now describe how to solve the problem (4.26). We first rewrite it as the following over-determined linear system

$$\boldsymbol{BY}_K \boldsymbol{z}_K = \boldsymbol{g} := \boldsymbol{h} - \boldsymbol{By}_s. \tag{4.28}$$

We start by using the QR factorization with Householder transformation to (4.28). In the $(k-1)$-th iteration, we have the following form

$$\widetilde{\boldsymbol{Q}}_{k-1}\widetilde{\boldsymbol{Q}}_{k-2}\cdots\widetilde{\boldsymbol{Q}}_1\boldsymbol{BY}_{k-1} = \boldsymbol{R}_{k-1}, \tag{4.29}$$

where $\widetilde{\boldsymbol{Q}}_{k-1}, \widetilde{\boldsymbol{Q}}_{k-2}, \cdots, \widetilde{\boldsymbol{Q}}_1 \in \mathbb{R}^{M\times M}$ is an orthogonal matrix and $\boldsymbol{R}_{k-1} \in \mathbb{R}^{M\times(k-1)}$ is upper-triangular. Note in the $k$-th iteration,

$$\boldsymbol{BY}_k = \boldsymbol{B}\left[\boldsymbol{Y}_{k-1}\ \boldsymbol{y}_k\right] = \left[\boldsymbol{BY}_{k-1}\ \boldsymbol{By}_k\right], \tag{4.30}$$

which is obtained by adding a new column $\boldsymbol{By}_k$ to $\boldsymbol{BY}_{k-1}$. Hence

$$\widetilde{\boldsymbol{Q}}_{k-1}\widetilde{\boldsymbol{Q}}_{k-2}\cdots\widetilde{\boldsymbol{Q}}_1\boldsymbol{BY}_k = \left[\boldsymbol{R}_{k-1}\ \boldsymbol{r}_k\right], \tag{4.31}$$

where $\boldsymbol{r}_k = \widetilde{\boldsymbol{Q}}_{k-1}\widetilde{\boldsymbol{Q}}_{k-2}\cdots\widetilde{\boldsymbol{Q}}_1\boldsymbol{By}_k$. Write $\boldsymbol{r}_k = \begin{bmatrix}\boldsymbol{r}_k^t \\ \boldsymbol{r}_k^b\end{bmatrix}$ with $\boldsymbol{r}_k^t \in \mathbb{R}^{k-1}$ and $\boldsymbol{r}_k^b \in \mathbb{R}^{M-k+1}$, and let $\boldsymbol{H}_k$ be the Householder reflector associated with $\boldsymbol{r}_k^b$, then $\widetilde{\boldsymbol{Q}}_k := \begin{bmatrix}\boldsymbol{I} & \\ & \boldsymbol{H}_k\end{bmatrix}$ will make

$$\widetilde{\boldsymbol{Q}}_k\widetilde{\boldsymbol{Q}}_{k-1}\cdots\widetilde{\boldsymbol{Q}}_1\boldsymbol{BY}_k = \boldsymbol{R}_k, \tag{4.32}$$

which is upper-triangular. So far we can estimate the condition number of the $k$-step least square system (4.28) by estimating the condition number $\kappa(\boldsymbol{R}_k)$ (it suffices to consider $\kappa(\widetilde{\boldsymbol{R}}_k)$, where $\widetilde{\boldsymbol{R}}_k := \boldsymbol{R}_k(1:k,:)$ is the top square part of $\boldsymbol{R}_k$) and decide whether to continue the iteration or not. Given a threshold $\epsilon > 0$, the $k$-th iteration stops if $\kappa(\widetilde{\boldsymbol{R}}_k) > \epsilon^{-1}$. Actually, $\kappa(\widetilde{\boldsymbol{R}}_k)$ can also be computed iteratively, that is, we can update $\kappa(\widetilde{\boldsymbol{R}}_k)$ by the information

of $\widetilde{\boldsymbol{R}}_{k-1}$. For example, one simple approach is to use 1 or $\infty$-condition number $\kappa_*(\widetilde{\boldsymbol{R}}_k)$ for $* = 1$ or $\infty$. Suppose we have evaluated $\widetilde{\boldsymbol{R}}_{k-1}^{-1}$ by the $(k-1)$-th iteration, and obtained $\widetilde{\boldsymbol{R}}_k$ in the $k$-th iteration as following form

$$\widetilde{\boldsymbol{R}}_k = \begin{bmatrix} \widetilde{\boldsymbol{R}}_{k-1} & \boldsymbol{r}_k \\ 0 & \sigma_k \end{bmatrix}, \tag{4.33}$$

then $\widetilde{\boldsymbol{R}}_k^{-1}$ can be evaluated by

$$\widetilde{\boldsymbol{R}}_k^{-1} = \begin{bmatrix} \widetilde{\boldsymbol{R}}_{k-1}^{-1} & -\sigma_k^{-1}\widetilde{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{r}_k \\ 0 & \sigma_k^{-1} \end{bmatrix}, \tag{4.34}$$

which only costs $O(k^2)$ flops. Next $\kappa_*(\widetilde{\boldsymbol{R}}_k) = \|\widetilde{\boldsymbol{R}}_k\|_*\|\widetilde{\boldsymbol{R}}_k^{-1}\|_*$ can be updated from the information of $\widetilde{\boldsymbol{R}}_{k-1}$ and $\widetilde{\boldsymbol{R}}_{k-1}^{-1}$ by $O(k)$ flops. Hence the total flops for computing $\kappa_*(\boldsymbol{R}_k)$ in all iterations will be no greater than $O(K^3)$ flops, where $K$ is the total number of iterations.

After the QR factorization, (4.28) can be rewritten as

$$\boldsymbol{Q}_K \boldsymbol{R}_K z_K \approx \boldsymbol{g}, \tag{4.35}$$

and then the least square solution $z_K$ is computed by applying back-substitution to

$$\widetilde{\boldsymbol{R}}_K z_K = \left(\boldsymbol{Q}_K^T \boldsymbol{g}\right)(1:K). \tag{4.36}$$

All in all, the whole algorithm for solving (4.21) can be depicted as follows.

**Algorithm SOLVE**.

1. find a particular solution $\boldsymbol{y}_s$ by (4.23) and (4.24), and let $\boldsymbol{g} := \boldsymbol{h} - \boldsymbol{B}\boldsymbol{y}_s$;
2. define $\boldsymbol{R} = [\,]$ which is an empty matrix in the beginning;
3. for $k = 1 : M$
4. find $\boldsymbol{y}_k \in \text{null}(\boldsymbol{A})$ which is orthonormal to $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{k-1}$;
5. $\boldsymbol{r}_k = \boldsymbol{B}\boldsymbol{y}_k$;
6. $\boldsymbol{r}_k = \widetilde{\boldsymbol{Q}}_{k-1} \cdots \widetilde{\boldsymbol{Q}}_1 \boldsymbol{r}_k$; (if $k = 1$, skip this line)
7. define $\boldsymbol{r}_k^t = \boldsymbol{r}_k(1 : k-1)$, $\boldsymbol{r}_k^b = \boldsymbol{r}_k(k : M)$;
8. $\boldsymbol{s}_k = -\text{sign}((\boldsymbol{r}_k^b)_1)\|\boldsymbol{r}_k^b\|\boldsymbol{e}_1$,
9. $\boldsymbol{v}_k = (\boldsymbol{s}_k - \boldsymbol{r}_k^b)/\|\boldsymbol{s}_k - \boldsymbol{r}_k^b\|$;
10. $\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R} & \begin{bmatrix} \boldsymbol{r}_k^t \\ \boldsymbol{s}_k \end{bmatrix} \end{bmatrix}$;
11. if $\kappa(\boldsymbol{R}(1:k,:)) > \epsilon^{-1}$, break;
12. end for
13. $\boldsymbol{g} = \widetilde{\boldsymbol{Q}}_k \cdots \widetilde{\boldsymbol{Q}}_1 \boldsymbol{g}$;
14. solve $\boldsymbol{R}(1:k,:)\boldsymbol{z} = \boldsymbol{g}(1:k)$ for $\boldsymbol{z}$ by back-substitution;
15. $\boldsymbol{y} = [\boldsymbol{y}_1 \cdots \boldsymbol{y}_k]\,\boldsymbol{z} + \boldsymbol{y}_s$.

Note that in Line 6, $\boldsymbol{r}_k = \widetilde{\boldsymbol{Q}}_{k-1} \cdots \widetilde{\boldsymbol{Q}}_1 \boldsymbol{r}_k$ can be computed by

- for $i = 1 : k-1$
- $\boldsymbol{r}_k(i : M) = \boldsymbol{r}_k(i : M) - 2\boldsymbol{v}_i\left(\boldsymbol{v}_i^T \boldsymbol{r}_k(i : M)\right)$;
- end for

and in Line 13, $\boldsymbol{g}$ can be computed by the same way.

**Fast matrix-vector multiplication**.

Most of computational time in the above algorithm is spent by Line 5, namely, computing $\boldsymbol{B}\boldsymbol{y}_k$. Since $\boldsymbol{B}$ is of size $O(N) \times O(N^2)$, a direct matrix-vector multiplication $\boldsymbol{B}\boldsymbol{y}$ costs $O(N^3)$

arithmetic operations. Fortunately, the specific data array of $\boldsymbol{B}$ allows a fast multiplication. Note that the adjacent rows of $\boldsymbol{B}$ are highly linearly dependent, and each row varies smoothly from previous ones. We first consider the boundary constraints (2.13), where $\boldsymbol{B}$ has the following form

$$\boldsymbol{B} = \begin{bmatrix} \overline{L}_0(x_1)\overline{L}_0(y_1) & \overline{L}_0(x_1)\overline{L}_1(y_1) & \cdots & \overline{L}_N(x_1)\overline{L}_N(y_1) \\ \overline{L}_0(x_2)\overline{L}_0(y_2) & \overline{L}_0(x_2)\overline{L}_1(y_2) & \cdots & \overline{L}_N(x_2)\overline{L}_N(y_2) \\ \cdots & \cdots & \cdots & \cdots \\ \overline{L}_0(x_M)\overline{L}_0(y_M) & \overline{L}_0(x_M)\overline{L}_1(y_M) & \cdots & \overline{L}_N(x_M)\overline{L}_N(y_M) \end{bmatrix}, \tag{4.37}$$

where $(x_i, y_i) = \boldsymbol{z}_i$, $i = 1, \cdots, M$ are the points spaced on $\partial\Omega$. Given

$$\boldsymbol{y} = [y_{00}\ y_{01}\ \cdots\ y_{N+1,N+1}]^T \in \mathbb{R}^{(N+1)^2}, \tag{4.38}$$

then

$$\boldsymbol{B}(i,:)\boldsymbol{y} = \sum_{j,k} \overline{L}_j(x_i)\overline{L}_k(y_i)y_{jk} \tag{4.39}$$

evaluates the expansion with base functions $\overline{L}_j\overline{L}_k$ and coefficients $y_{jk}$ at point $\boldsymbol{z}_i$. Hence, the plot of $\boldsymbol{B}\boldsymbol{y}$ shows the profile of $\sum \overline{L}_j(x)\overline{L}_k(y)y_{jk}$ defined on $\partial\Omega$, which is usually (piecewise) smooth as long as $\partial\Omega$ is (piecewise) smooth.

Due to its smoothness, instead of evaluating the whole product $\boldsymbol{B}\boldsymbol{y}$, it suffices to choose several sampling nodes on $\partial\Omega$ (namely, several rows of $\boldsymbol{B}$) and do multiplication on them. After that, the value at non-sampling points on $\partial\Omega$ can be interpolated based on the data at sampling nodes. Fortunately, the complexity of evaluation at a point by usual interpolation techniques is much less than doing a direct vector multiplication. Therefore, when computing the product $\boldsymbol{B}\boldsymbol{y}$, we can only multiply a fixed number $N_0$ rows of $\boldsymbol{B}$ by $\boldsymbol{y}$, and estimate other part of $\boldsymbol{B}\boldsymbol{y}$ by interpolation, for instance, the cubic spline interpolation which costs $O(N_0)$ for a solo entry and $O(N_0N)$ for all entries. By this method, the total complexity for computing $\boldsymbol{B}\boldsymbol{y}$ is $O(N_0N^2)$. In practical implementation, $N_0$ is determined by the accuracy requirement and is independent of $N$. We demonstrate it by the following example, in which $\partial\Omega$ is set by $r = 0.65 + 0.25\sin(3\theta)$ and $\boldsymbol{B}$ is multiplied by an all-one vector $\boldsymbol{e}$. In Fig. 2, the $l^2$ errors of computing $\boldsymbol{B}\boldsymbol{e}$ by our interpolation method versus $N$ are shown, and it is observed the errors only depend on the number of sampling nodes, rather than the size of $\boldsymbol{B}$. Furthermore, the numbers of operations for different $N$ and $N_0$ are estimated and presented in Fig. 3, from which we see the complexity for the matrix-vector multiplication on $\boldsymbol{B}$ is indeed about $O(N^2)$.

For boundary constraints (2.14), we suppose the number of test functions $\{\chi_i\}$ and the quadrature nodes are set by $O(N)$, then in this case $\boldsymbol{B}$ is formed as the product of a $O(N) \times O(N)$ matrix related to $\{\chi_i\}$ and another $O(N) \times O(N^2)$ matrix of the form in (4.37). Hence $\boldsymbol{B}\boldsymbol{y}$ is computed by first applying the preceding fast multiplication technique, and then doing a usual $O(N)$ by $O(N)$ matrix-vector multiplication. Thus the total number of operations is also $O(N^2)$.

Now we can determine the complexity of **Algorithm SOLVE**. First, Line 4 can be precomputed since it does not depend on the domain and the data. Due to the orthogonality, $\boldsymbol{A} \in \mathbb{R}^{O(N^2)\times O(N^2)}$ is sparsely structured with $O(1)$ nonzero entries in each row. So sparse solvers can be applied to compute an orthonormal set of null($\boldsymbol{A}$) in advance. Then for other lines relate to computation, Line 1 costs $O(sN^2)$ if an iterative solver is used for (4.24) with $s$ iterations. Inside the for-loop, Line 5 costs $O(N^2)$ by fast computation and Line 6,8,9,11 and 13 each costs no more than $O(N^2)$, hence the whole for-loop costs at most $O(N^3)$ due

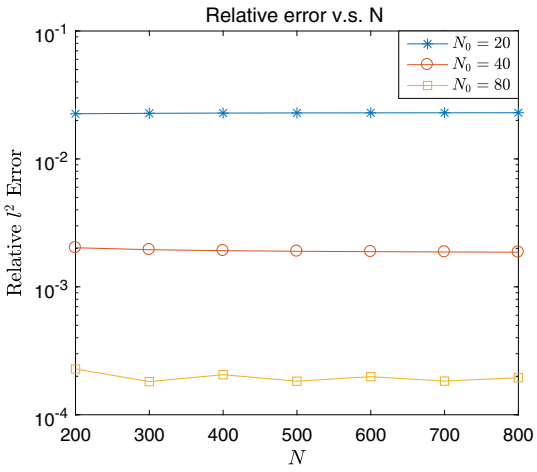



**Fig. 2** $l^2$ error for computing $\boldsymbol{Be}$ by interpolation method for different $N_0$ and $N$

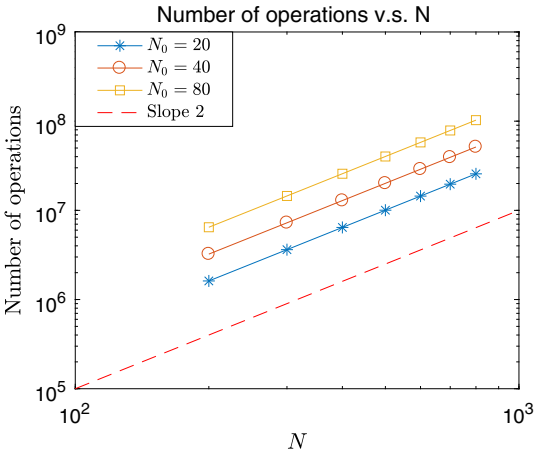



**Fig. 3** Number of arithmetic operations for different $N$

to $M = O(N)$. Finally the cost of Line 14 and 15 is within $O(N^3)$. Therefore, the total cost for solving (4.21) is $O(N^3) + O(sN^2)$ operations.

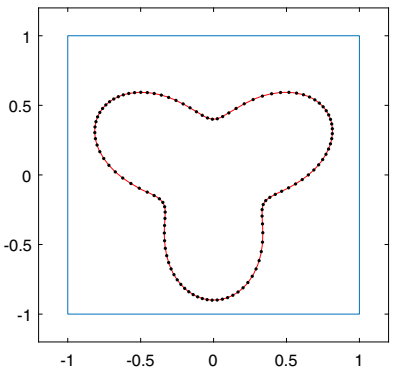
## 5 Numerical Results

We present in this section several numerical results using the two proposed methods. For all examples below, (2.13) is chosen as the approximate boundary condition.

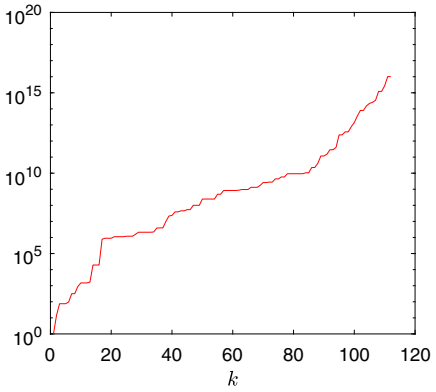
### 5.1 Poisson Type Equations with Smooth Solutions

In the first example, we use the first method to solve the following Poisson-type equation

$$\begin{aligned} -\Delta u + \alpha u &= f \text{ in } \Omega, \\ u &= h \text{ on } \partial\Omega, \end{aligned} \tag{5.1}$$




**(a)** the problem domain, extended domain and boundary nodes




**(b)** $\kappa_1(\boldsymbol{R}_k)$

**Fig. 4** The first example: the first method applied to (5.1) with $N = 51$

where $\partial\Omega$ (see Fig. 4a) is characterized by the polar expression

$$r = r_0 + \delta \sin(n\theta), \tag{5.2}$$

and the exact solution is set by

$$u = r^3(r_0 + \delta \sin(n\theta) - r), \tag{5.3}$$

with $r_0 = 0.65$, $\delta = 0.25$, $n = 3$. Note that the exact solution satisfies the homogeneous Dirichlet boundary condition.

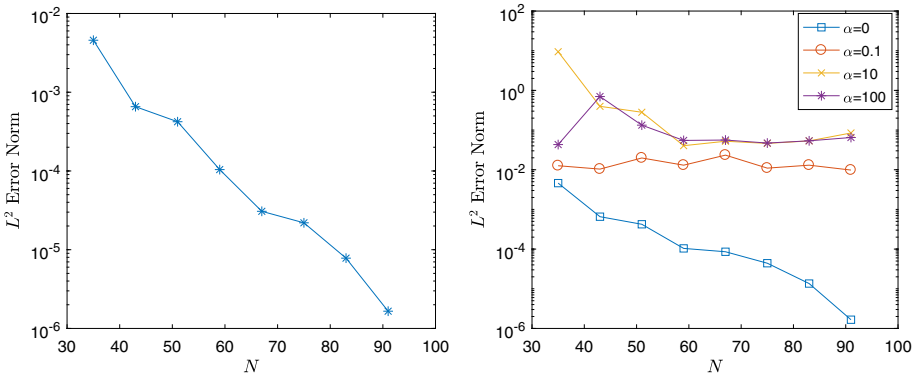
First, we let $\alpha = 0$, and choose $N = 51$ ($M = 104$) where $N$ is the degree of tensorial polynomial space specified in (2.15)-(2.16). The original domain $\Omega$, extended domain $\widetilde{\Omega}$ and the sampling nodes $\{\mathbf{z}_i\}$ defined in (2.13) for $N = 51$ are shown in Fig. 4a. We plot the condition number of $\boldsymbol{R}_k$ in Fig. 4b for all $k \leq M$, and observe that $\kappa_1(\boldsymbol{R}_k)$ increases rapidly from the beginning, and reaches an acceptable level of $10^6$ when $k$ is around $\frac{1}{3}M$. therefore in this example we choose $K = \lfloor \frac{1}{3}M \rfloor = \lfloor \frac{2N+2}{3} \rfloor$ as a prescribed number of iterations in the for-loop in **Algorithm SOLVE**, that means the solution to the least square problem (4.26) is searched in a $K$-dimensional subspace of null($\boldsymbol{A}$).

In Fig. 5a, we plot the $L^2$-error for the numerical solution of (5.1) with $\alpha = 0$ for various $N$, and observe that the error converges exponentially as predicted by Theorem 2.9. On the other hand, we plot in Fig. 5b the $L^2$-error for the numerical solution of (5.1) with various $\alpha \neq 0$. We observe that the convergence rate deteriorates as $\alpha$ increases, which explains why we were only able to prove the results in Theorem 2.9 for $\alpha = 0$.

In the second example, we use the second method to solve the problem (5.1) where $\Omega$ is a pentagon with vertices $(0, 0.9)$, $(-0.9, 0.2)$, $(-0.7, -0.8)$, $(0.7, -0.8)$, and $(0.9, 0.2)$. The exact solution is chosen to be

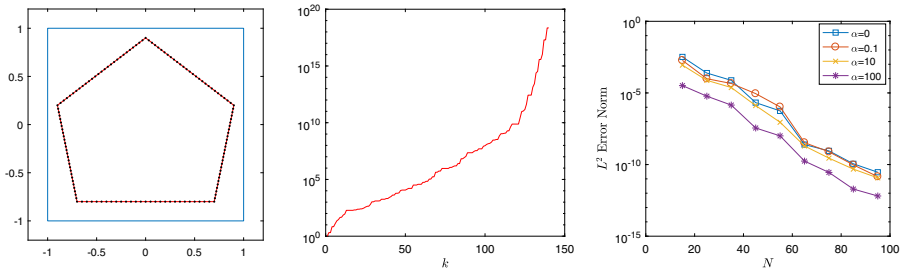
$$u = \exp\left(-\frac{x_1^2 + x_2^2}{2}\right). \tag{5.4}$$

First we take $\alpha = 10$, $N = 35$, and plot $\kappa_1(\boldsymbol{R}_k)$ for different $k$ in Fig. 6b, together with the original domain $\Omega$, extended domain $\widetilde{\Omega}$ and the sampling nodes $\{\mathbf{z}_i\}$ shown in Fig. 6a. We observe that $\kappa_1(\boldsymbol{R}_k)$ behaviors similarly as with the first method.




**(a)** $\alpha = 0$ **(b)** various $\alpha$

**Fig. 5** $\|u - u_N\|_{L^2(\Omega)}$ versus $N$ for the first example




**(a)** the problem domain, extended domain and boundary nodes **(b)** $\kappa_1(\boldsymbol{R}_k)$ **(c)** $L^2$ error with various $\alpha$

**Fig. 6** The second example: the second method applied to (5.1) with $N = 35$

Then, we set $\epsilon^{-1} = 10^{1+N/10}$ in the **Algorithm SOLVE**, and plot $L^2$ error with various $\alpha$ in Fig. 6c. We observe that exponential convergence is achieved for all $\alpha$.

### 5.2 Problems with Corner Singularities

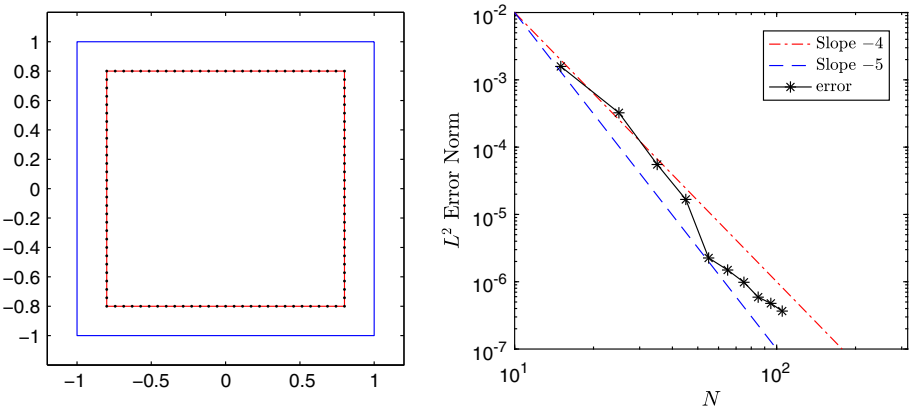
In the third example, we apply the first method to the Poisson equation

$$\begin{aligned} -\Delta u &= 1 \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{5.5}$$

where $\Omega$ is a square with vertices $(T, T)$, $(T, -T)$, $(-T, -T)$, $(-T, T)$ with $T = 0.8$. The exact solution is given by

$$u(x_1, x_2) = -\frac{64T^2}{\pi^4} \sum_{\substack{n,m=1 \\ n,m \text{ odd}}}^{\infty} (-1)^{\frac{n+m}{2}} \frac{cos\left(\frac{n\pi x_1}{2T}\right) cos\left(\frac{m\pi x_2}{2T}\right)}{nm\left(n^2+m^2\right)}, \tag{5.6}$$

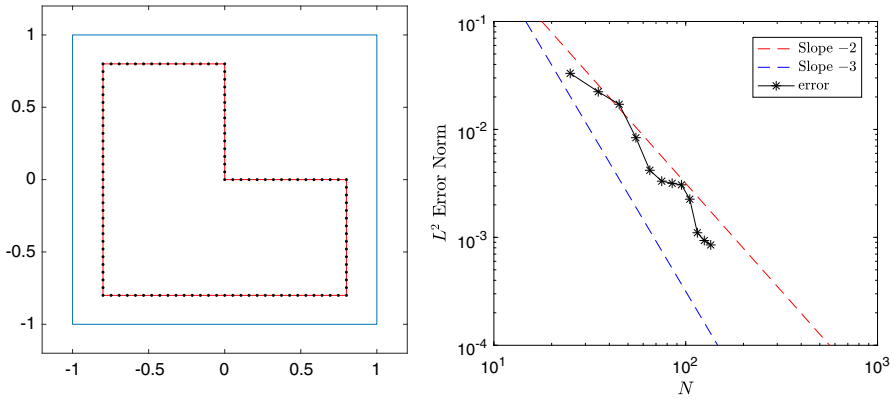
which are weakly singular at the four corners. The original domain $\Omega$, extended domain $\widetilde{\Omega}$ and the boundary nodes are shown in Fig. 7a, together with the $L^2$-error vs. $N$ in Fig. 7b.




**(a)** the problem domain, extended domain and boundary nodes

**(b)** $L^2$-error vs. $N$

**Fig. 7** The third example: Poisson equation with corner singularity using the first method



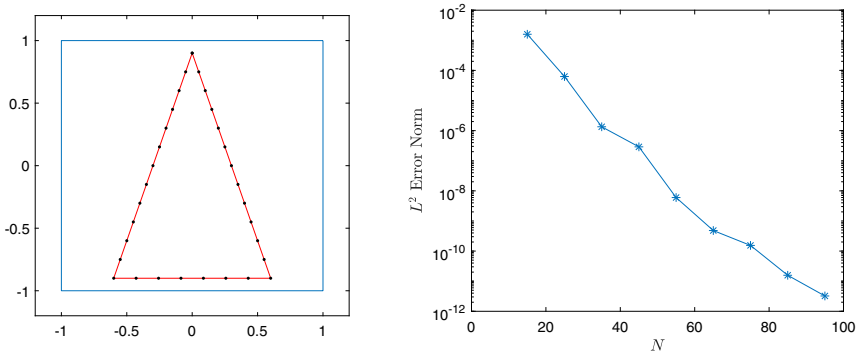

**(a)** the problem domain, extended domain and boundary nodes for the fourth example (N=15)

**(b)** $\|u - u_N\|_{L^2(\Omega)}$ vs. $N$

**Fig. 8** The fourth example: Poisson equation in L-shaped domain using the second method

We observe that the convergence rate is between 4th and 5th order which is similar to the rate by the spectral Galerkin method [20] and about twice the rate by finite differences with a uniform grid.

For the fourth example, we consider (5.5) in a L-shaped domain with vertices $(0, 0)$, $(T, 0)$, $(T, -T)$, $(-T, -T)$, $(-T, T)$ and $(0, T)$ with $T = 0.8$. The solution of the PDE is weakly singular at all corners but with the strongest singularity at the reentry corner. We apply the second method with stopping criteria $\epsilon^{-1} = 10^{N/20}$ to this problem. The original domain $\Omega$, extended domain $\widetilde{\Omega}$ and the boundary nodes $\{\mathbf{z}_i\}$ for $N = 15$ is shown in Fig. 8a. Since an exact solution is not available, we use the approximate solution obtained with $N = 255$ as the reference solution. We plot the $L^2$-errors for variour $N$ in Fig. 8b, and observe that the convergence rate is between 2nd and 3rd order, which is also much better than the rate by finite differences or finite elements with a uniform grid.




**(a)** the problem domains and boundary nodes with N=15

**(b)** $\|u - u_N\|_{L^2(\Omega)}$ vs. $N$ with $\alpha(x) = (\sin x_1 + 1)(\cos x_2 + 1)$ by the second method

**Fig. 9** A problem with variable coefficients

### 5.3 A Problem with Variable Coefficients

As the last example, we apply the second method to the Dirichlet problem (1.1) with non-constant coefficients on a triangle with vertices $(0, 0.9)$, $(0.6, -0.9)$, $(-0.6, -0.9)$. We set $\beta(x) = \exp(x_1 + x_2)$ with $\alpha(x) = 0$ and $\alpha(x) = (\sin x_1 + 1)(\cos x_2 + 1)$ with the exact solution given by (5.4). In Fig. 9a and b, we plot the problem domains and boundary nodes with $N = 15$, the $L^2$ errors with $\alpha(x) = (\sin x_1 + 1)(\cos x_2 + 1)$, respectively.

## 6 Concluding Remarks

We developed in this paper two novel spectral methods for solving two-dimensional elliptic PDEs in complex domains using a fictitious domain approach. One is specifically designed for the Poisson equation with the trial space $H^2(\tilde{\Omega})$ satisfying the original boundary condition and the test space $L^2(\tilde{\Omega})$, where $\tilde{\Omega}$ is the extended domain. This method is proved to be well-posed with spectral accuracy in the sense that the convergence rate increases with the smoothness of the solution. However, the error deteriorates if the method is applied to more general elliptic equations. On the other hand, the second method can achieve spectral accuracy for general elliptic equations with trial space $H^1(\tilde{\Omega})$ satisfying the original boundary condition and the test space $H_0^1(\tilde{\Omega})$. However, its well-posedness and error estimate are still elusive.

Both methods lead to ill-conditioned linear systems which can not be efficiently solved by a direct methods. We developed a tailored least square algorithm which allows us to solve these ill-conditioned linear systems with a $O(N^3)$ computational complexity (where $N$ being the number of points in each direction), which is comparable to the fast spectral elliptic solver in rectangular domains. We presented ample numerical results to show that the new methods are very effective for problems with smooth as well as weakly singular exact solutions.

While the two fictitious domain formulations can be essentially applied to elliptic problems in three dimensional complex domains, their implementations are much more involved and will be left for a future endever.

## References


1. Adcock, B., Huybrechs, D., Martin-Vaquero, J.: On the numerical stability of fourier extensions. Found. Comput. Math. **14**, 635–687 (2014)
2. Albin, N., Bruno, O.P.: A spectral fc solver for the compressible navier-stokes equationsin general domains i: explicit time-stepping. J. Comput. Phys. **230**, 6248–6270 (2011)
3. Albin, N., Bruno, O.P., Cheung, T.Y., Cleveland, R.O.: Fourier continuation methods for high-fidelity simulation of nonlinear acoustic beams. J. Acoust. Soc. Am. **132**, 2371–2387 (2012)
4. Angot, P., Pan, C.-H.B., Fabrie, P.: A penalization method to take into account obstacles in incompressible viscous flows. Numer. Math. **81**, 497–520 (1999)
5. Babuska, I., Aziz, A.K.: Survey lectures on the mathematical foundation of the finite element method. In: Aziz, A.K. (ed.) The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations. Academic Press, New York (1972)
6. Boyd, J.P.: A comparison of numerical algorithms for fourier extension of the first, second, and third kinds. J. Comput. Phys. **178**, 118–160 (2002)
7. Bruno, O.P., Lyon, M.: High-order unconditionally stable fc-ad solvers for general smooth domains i. basic elements. J. Comput. Phys. **229**, 2009–2033 (2010)
8. Buffat, M., Le Penven, L.: A spectral fictitious domain method with internal forcing for solving elliptic pdes. J. Comput. Appl. Math. **230**, 2433–2450 (2011)
9. Dinh, Q.V., Glowinski, R., He, J., Kwock, V., Pan, T.W., Periaux, J.: Lagrange multiplier approach to fictitious domain methods: application to fluid dynamics and electro-magnetics. In: Keyes, D.E., Chan, T.F., Meurant, G., Scroggs, J.S., Voigt, R.G. (eds.) Domain Decomposition Methods for Partial Differential Equations. SIAM, Philadelphia (1992)
10. Elghaoui, M., Pasquetti, R.: A spectral embedding method applied to the advection–diffusion equation. J. Comput. Phys. **125**, 464–476 (1996)
11. Ern, A., Guermond, J.-L.: Theory and Practice of Finite Elements. Springer, Berlin (2004)
12. Gilbarg, D., Trudinger, N.S.: Elliptic Partial Differential Equations of Second Order. Springer, Berlin (2001)
13. Glowinski, R., Pan, T.-W., Periaux, J.: A fictitious domain method for dirichlet problem and applications. Comput. Methods Appl. Mech. Eng. **111**, 283–303 (1994)
14. Lui, S.H.: Spectral domain embedding for elliptic pdes in complex domains. J. Comput. Appl. Math. **225**, 541–557 (2009)
15. Lyon, M.: A fast algorithm for fourier continuation. SIAM J. Sci. Comput. **33**, 3241–3260 (2011)
16. Lyon, M., Bruno, O.P.: High-order unconditionally stable fc-ad solvers for general smooth domains ii. elliptic, parabolic and hyperbolic pdes; theoretical considerations. J. Comput. Phys. **229**, 3358–3381 (2010)
17. Le Penven, L., Buffat, M.: On the spectral accuracy of a fictitious domain method for elliptic operators in multi-dimensions. J. Comput. Phys. **231**, 7893–7906 (2012)
18. Orszag, S.A.: Spectral methods for complex geometries. J. Comput. Phys. **37**, 70–92 (1980)
19. Schneider, K.: Numerical simulation of the transient flow behaviour in chemical reactors using a penalisation method. Comput. Fluids **34**, 1223–1238 (2005)
20. Shen, J.: Efficient spectral-Galerkin method I. direct solvers for second- and fourth-order equations by using Legendre polynomials. SIAM J. Sci. Comput. **15**, 1489–1505 (1994)
21. Shen, J.: Efficient spectral-Galerkin method II. direct solvers for second- and fourth-order equations by using Chebyshev polynomialS. SIAM J. Sci. Comput. **16**, 74–87 (1995)
22. Shen, J., Tang, T., Wang, L.: Spectral Methods: Algorithms, Analysis and Applications. Springer, Berlin (2011)
23. Strang, G.: Variational crimes in the finite element method. In: Aziz, A.K. (ed.) The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations. Academic Press, New York (1972)
24. van yen, R.N., Kolomenskiy, D., Schneider, K.: Approximation of the laplace and stokes operators with dirichlet boundary conditions through volume penalization: a spectral viewpoint. Numer. Math. **128**, 301–338 (2014)