

Fast Fourier-Galerkin Methods for Nonlinear Boundary Integral Equations

Xiangling Chen · Rui Wang · Yuesheng Xu

Received: 6 October 2012 / Revised: 3 January 2013 / Accepted: 15 January 2013
Published online: 1 February 2013
© Springer Science+Business Media New York 2013

Abstract We develop in this paper a fast Fourier-Galerkin method for solving the nonlinear integral equation which is reformulated from a class of nonlinear boundary value problems. By projecting the nonlinear term onto the approximation subspaces, we make the Fourier-Galerkin method more efficient for solving the nonlinear integral equations. A fast algorithm for solving the resulting discrete nonlinear system is designed by integrating together the techniques of matrix compressing, numerical quadrature for oscillatory integrals, and the multilevel augmentation method. We prove that the proposed method enjoys an optimal convergence order and a nearly linear computational complexity. Numerical experiments are presented to confirm the theoretical estimates and to demonstrate the efficiency and accuracy of the proposed method.

Keywords Nonlinear boundary integral equations · Fourier-Galerkin methods · Multilevel augmentation methods

X. Chen

Department of Mathematics and System Science, College of Science, National University of Defense Technology, Changsha 410073, People's Republic of China
e-mail: chenxiangling828@163.com

R. Wang (✉)

School of Mathematics, Jilin University, Changchun 130012, People's Republic of China
e-mail: rwang11@jlu.edu.cn

Y. Xu

Department of Mathematics, Syracuse University, Syracuse, NY 13244, USA

Y. Xu

Guangdong Key Laboratory of Computational Science, Sun Yat-sen University, Guangzhou 510275, People's Republic of China
e-mail: yxu06@syr.edu

1 Introduction

We consider in this paper the Laplace equation with nonlinear boundary conditions, which serves as mathematical models for many important applications, such as acoustics, elasticity, electromagnetics, and fluid dynamics. Suppose that D is a simply connected bounded domain in \mathbb{R}^2 with a C^2 boundary Γ . Specifically, we shall solve the nonlinear boundary value problem

$$\begin{cases} \Delta u(x) = 0, & x \in D, \\ \frac{\partial u(x)}{\partial \mathbf{n}_x} = -g(x, u(x)) + g_0(x), & x \in \Gamma, \end{cases} \tag{1.1}$$

where \mathbf{n}_x denotes the exterior unit normal to Γ at x . Throughout this paper we assume that $g(x, u)$ is continuous with respect to $x \in \Gamma$ and Lipschitz continuous with respect to $u \in \mathbb{R}$, the partial derivative $D_u g$ of g with respect to the variable u exists and is Lipschitz continuous, and for each $u \in C(\Gamma)$, $g(\cdot, u(\cdot))$, $D_u g(\cdot, u(\cdot)) \in C(\Gamma)$.

The boundary integral equation approach is commonly used in solving (1.1). It transforms the boundary problem into an integral equation defined on the boundary Γ . In contrast to the Laplace equation with linear boundary conditions, the reformulation of nonlinear boundary value problem (1.1) leads to a nonlinear boundary integral equation. In the literature, various numerical methods for solving nonlinear integral equations reformulated from a broad range of boundary value problems were developed. Specifically, a degenerate kernel scheme was introduced in [13, 19]. Nyström’s method and the product integration method were discussed in [17, 22]. Projection methods, including the Galerkin scheme and the collocation scheme, were discussed in [2, 3, 6, 12, 16, 20, 21, 24–27, 29]. Recently, a series of schemes based on the projection method were proposed in [4, 7, 8]. Also, an extrapolation algorithm based on Nyström’s method was developed in [11].

In this paper, we shall develop a fast algorithm based on the Fourier-Galerkin method for solving the nonlinear boundary integral equation reformulated from (1.1). The main difficulties of the numerical solution of the integral equations are caused by the nonlinearity. Typically, solving the corresponding discrete nonlinear system by the classical Newton’s method and its variants demands a significantly large amount of computational efforts. Specifically, in each iteration step, we have to evaluate and update the Jacobian matrix in the entire approximate subspace. Moreover, the dimension of the approximate subspace is usually required to be large so that one can get good approximation accuracy.

The resulting boundary integral equation contains both linear and nonlinear components. The nonlinear component involves an integral operator with the logarithmic kernel. This will lead us to compute double singular integrals when we implement Newton’s method. As in the case of solving linear boundary integral equations [30], we decompose the integral operator into the form $\mathcal{A} + \mathcal{B}$, where \mathcal{A} is a singular operator carrying the main singularity characteristic of the integral operator and \mathcal{B} is a compact operator with a smooth kernel. It is worth to point out that the singular operator \mathcal{A} has the Fourier basis functions as its eigenfunctions. Motivated by the features of \mathcal{A} and \mathcal{B} , we choose the subspaces generated by the Fourier basis functions as the approximate subspaces in the Galerkin methods. Moreover, we project the nonlinear term arising in the nonlinear integral operator onto the approximate subspace. This idea was used in [7], where a collocation method was considered. By doing this, we avoid evaluating the double singular integrals which appear in the related Jacobian matrix. Instead, we only need to generate once the matrix representation of operators \mathcal{A} and \mathcal{B} under the Fourier basis and in each iteration step multiply the matrix representation by vectors generated by evaluating single integrals.

Since operator \mathcal{A} has the Fourier basis functions as its eigenfunctions, the matrix representation of \mathcal{A} is diagonal and the entries on the diagonal can be obtained directly. However, the matrix representation of operator \mathcal{B} and the integral operator in the linear components are usually dense. Hence, generating these matrices and multiplying them by vectors will still need large computational costs. Furthermore, the multiplications of vectors and dense matrices have to be carried out in each iteration step of Newton's method because the Jacobian matrices need to be established and updated during the iteration. To overcome these computational challenges so as to develop a fast solver for the Fourier-Galerkin method, we adopt three techniques. A matrix truncation strategy has been proposed in [5] for solving linear boundary integral equations. In the spirit of [5], the matrix representation of smooth kernels under the Fourier basis can be compressed to sparse ones. Here, we also employ the numerical quadrature scheme proposed in [15] for computing all the nonzero entries of the representation matrices. The multilevel augmentation method has been presented in [10] and developed in [7, 8] for solving nonlinear boundary integral equations. In this method, one inverts the nonlinear operator in a much smaller subspace rather than in the whole approximate subspace. It reduces the computational complexity to a nearly linear order. Combining these three techniques together, we develop a multilevel augmentation method for solving the resulting fully discrete truncated nonlinear system.

We organize this paper in four sections. In Sect. 2 we review the boundary integral equation reformulation of nonlinear boundary value problem (1.1), then we describe a Fourier-Galerkin method based on nonlinear term projection for solving the resulting nonlinear boundary integral equation. In Sect. 3 we show that the Fourier-Galerkin method enjoys the exponential convergence order if the exact solution is analytic. For the case that the exact solution has a Sobolev regularity, we develop in Sect. 4 a fully discrete fast Fourier-Galerkin method by employing a matrix truncation strategy, a quadrature formula for oscillatory integrals, and the multilevel augmentation method. To show the proposed method is a fast and efficient solver, we establish optimal estimates of its computational complexity and its convergence order. Moreover, we present numerical examples to show the computational efficiency and the approximate accuracy of the proposed method.

2 The Fourier-Galerkin Method for Nonlinear Boundary Integral Equation

In this section, we describe the Fourier-Galerkin method for solving the nonlinear boundary integral equation reformulated from the nonlinear boundary value problem (1.1).

We begin with reviewing the boundary integral equation reformulation. Following [2, 24], solving (1.1) leads to the following nonlinear integral equation defined on Γ

$$\begin{aligned} u(x) - \frac{1}{\pi} \int_{\Gamma} u(y) \frac{\partial}{\partial \mathbf{n}_y} \log |x - y| ds_y - \frac{1}{\pi} \int_{\Gamma} g(y, u(y)) \log |x - y| ds_y \\ = - \frac{1}{\pi} \int_{\Gamma} g_0(y) \log |x - y| ds_y, \quad x \in \Gamma. \end{aligned} \quad (2.1)$$

Making use of the solution of Eq. (2.1) and the boundary condition in (1.1), we can obtain the value of u at $x \in D$ by the following formula

$$u(x) = \frac{1}{2\pi} \int_{\Gamma} u(y) \frac{\partial}{\partial \mathbf{n}_y} (\log |x - y|) ds_y - \frac{1}{2\pi} \int_{\Gamma} \frac{\partial u(y)}{\partial \mathbf{n}_y} \log |x - y| ds_y.$$

From this viewpoint, in this paper, we aim at solving the nonlinear boundary integral equation (2.1) numerically. Suppose that the boundary Γ has a parametrization

$$x(t) = (\xi(t), \eta(t)), \quad t \in I := [0, 2\pi].$$

Let \mathbb{Z} be the set of all integers. We define three kernels at $t, \tau \in I$ as

$$K(t, \tau) := \begin{cases} \frac{1}{\pi} \frac{\eta'(\tau)(\xi(\tau) - \xi(t)) - \xi'(\tau)(\eta(\tau) - \eta(t))}{(\xi(\tau) - \xi(t))^2 + (\eta(\tau) - \eta(t))^2}, & t - \tau \neq 2k\pi, k \in \mathbb{Z}, \\ \frac{1}{\pi} \frac{\eta'(t)\xi''(t) - \xi'(t)\eta''(t)}{2(\xi'(t)^2 + \eta'(t)^2)}, & \text{otherwise,} \end{cases}$$

$$A(t, \tau) := -\frac{1}{\pi} \log \left| 2e^{-1/2} \sin \frac{t - \tau}{2} \right|,$$

and

$$B(t, \tau) := \begin{cases} -\frac{1}{\pi} \log \left| \frac{e^{1/2} \sqrt{(\xi(t) - \xi(\tau))^2 + (\eta(t) - \eta(\tau))^2}}{2 \sin \frac{t - \tau}{2}} \right|, & t - \tau \neq 2k\pi, k \in \mathbb{Z}, \\ -\frac{1}{\pi} \log \left| e^{1/2} \sqrt{\xi'(\tau)^2 + \eta'(\tau)^2} \right|, & \text{otherwise.} \end{cases}$$

We also introduce functions

$$u(t) := u(x(t)), \quad g(t, u(t)) := g(x(t), u(x(t))), \quad g_0(t) := g_0(x(t)), \quad t \in I.$$

With these notations, Eq. (2.1) can now be rewritten as

$$\begin{aligned} u(t) - \int_I K(t, \tau)u(\tau)d\tau + \int_I A(t, \tau)g(\tau, u(\tau))|x'(\tau)|d\tau + \int_I B(t, \tau)g(\tau, u(\tau))|x'(\tau)|d\tau \\ = \int_I A(t, \tau)g_0(\tau)|x'(\tau)|d\tau + \int_I B(t, \tau)g_0(\tau)|x'(\tau)|d\tau, \quad t \in I. \end{aligned} \tag{2.2}$$

We next represent Eq. (2.2) in an operator form. To this end, we introduce three linear integral operators defined, respectively, by

$$(\mathcal{K}u)(t) := \int_I K(t, \tau)u(\tau)d\tau, \quad t \in I,$$

and

$$(Au)(t) := \int_I A(t, \tau)u(\tau)d\tau, \quad (Bu)(t) := \int_I B(t, \tau)u(\tau)d\tau, \quad t \in I,$$

and define a nonlinear operator by

$$(\Psi u)(t) := g(t, u(t)) |x'(t)|, \quad t \in I.$$

With these operators, Eq. (2.2) is represented as the operator equation

$$u - (\mathcal{K} - \mathcal{A}\Psi - \mathcal{B}\Psi)u = f, \tag{2.3}$$

where $f := (\mathcal{A} + \mathcal{B})(g_0\sqrt{\xi'^2 + \eta'^2})$.

We observe from the definition of the kernels that K is of C^{s-2} and B is of C^{s-1} when the boundary Γ is of C^s , $s \geq 2$. By the smoothness of the two kernels, we have that the operators \mathcal{K} and \mathcal{B} are both compact operators. The operator \mathcal{A} has weak singularity and it has trigonometric monomials as its eigenfunctions, that is, there holds

$$\mathcal{A} : e^{ikt} \rightarrow \frac{1}{\max\{1, |k|\}} e^{ikt}, \quad k \in \mathbb{Z}.$$

We note that the sum $\mathcal{A} + \mathcal{B}$ is the boundary integral operator with logarithmic kernel, which has weak singularity. In particular, \mathcal{A} carries the main singularity characteristic of the boundary integral operator and \mathcal{B} is a compact operator with a smooth kernel. This decomposition has two advantages. One is that \mathcal{A} has good properties that its matrix representation under some basis can be calculated with less efforts. For instance, in [8], the entries of the matrix representation of \mathcal{A} under the multi-scale wavelet basis can be calculated exactly by explicit formulas. In our case, since operator \mathcal{A} has the Fourier basis functions as its eigenfunctions, its matrix representation under the Fourier basis is diagonal and the entries on the diagonal can also be obtained exactly. The other is that the matrix representation of the compact operator \mathcal{B} under the same basis can be compressed to a sparse matrix without loss of critical information encoded in the matrix.

Taking advantages of the operator decomposition, we will establish the Galerkin method using the Fourier basis for solving operator equation (2.3). Let \mathbb{N} denote the set of all positive integers and set $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. To enumerate finite sets, we also introduce index sets $\mathbb{Z}_n^+ := \{1, 2, \dots, n - 1\}$ and $\mathbb{Z}_n := \mathbb{Z}_n^+ \cup \{0\}$. For any $k \in \mathbb{Z}$, for $x \in I$ we set

$$e_k(x) := \begin{cases} \frac{1}{\sqrt{\pi}} \cos kx, & k \in \mathbb{N}, \\ \frac{1}{\sqrt{2\pi}}, & k = 0, \\ \frac{1}{\sqrt{\pi}} \sin(-kx), & k \in \mathbb{Z} \setminus \mathbb{N}_0. \end{cases}$$

For each $n \in \mathbb{N}$, we define a finite dimensional subspace \mathbb{X}_n by

$$\mathbb{X}_n := \text{span}\{e_k : |k| \in \mathbb{Z}_n\}.$$

Let \mathcal{P}_n be the orthogonal projection operator from $L^2(I)$ to \mathbb{X}_n . The Fourier-Galerkin method for solving Eq. (2.3) is to seek u_n in \mathbb{X}_n such that

$$u_n - \mathcal{P}_n(\mathcal{K} - \mathcal{A}\Psi - \mathcal{B}\Psi)u_n = \mathcal{P}_n f. \tag{2.4}$$

The following theorem gives the unique solvability of Eq. (2.4), which can be proved by making use of Theorem 2 of [27, 28].

Theorem 2.1 *Suppose that $u \in L^2(I)$ is an isolated solution of (2.3) and 1 is not an eigenvalue of the linear operator $\mathcal{K} - ((\mathcal{A} + \mathcal{B})\Psi)'(u)$. Then for sufficiently large n , Eq. (2.4) has a unique solution $u_n \in B(u, \delta)$ for some $\delta > 0$ and there exist positive constants c_1 and c_2 such that*

$$c_1 \|u - \mathcal{P}_n u\| \leq \|u - u_n\| \leq c_2 \|u - \mathcal{P}_n u\|.$$

To describe the regular property of the solution and then estimate the convergence order of the Fourier-Galerkin method, we introduce appropriate Sobolev spaces. For each $\mu \geq 0$, we denote by $H^\mu(I)$ the Sobolev space of functions $\phi \in L^2(I)$ with the property

$$\sum_{k \in \mathbb{Z}} (1 + k^2)^\mu |\phi_k|^2 < \infty,$$

where ϕ_k is the k th Fourier coefficient of ϕ defined by $\phi_k := \int_I \phi(x)e_k(x)dx$. The inner product of the Sobolev space $H^\mu(I)$ is defined for $\phi, \psi \in H^\mu(I)$ by

$$\langle \phi, \psi \rangle_\mu := \sum_{k \in \mathbb{Z}} (1 + k^2)^\mu \phi_k \overline{\psi_k},$$

and its norm is defined by $\|\phi\|_\mu := \langle \phi, \phi \rangle_\mu^{\frac{1}{2}}$. It is well-known [1, 18] that there exists a positive constant c such that for any $v \in H^\mu(I)$,

$$\|v - \mathcal{P}_n v\| \leq cn^{-\mu} \|v\|_\mu. \tag{2.5}$$

The following corollary is a direct result of Theorem 2.1 and estimation (2.5).

Corollary 2.2 *If the solution $u \in H^\mu(I)$, $\mu \geq 0$, there exists a constant $c > 0$ such that*

$$\|u - u_n\| \leq cn^{-\mu} \|u\|_\mu.$$

Based upon the Fourier basis of the subspace \mathbb{X}_n , the Fourier-Galerkin method (2.4) is equivalent to a system of nonlinear equations. Specifically, it suffices to seek a function $u_n \in \mathbb{X}_n$ with the form

$$u_n(t) = \sum_{|l| \in \mathbb{Z}_n} a_l e_l(t), \quad t \in I,$$

which satisfies the nonlinear system

$$\langle u_n - (\mathcal{K} - \mathcal{A}\Psi - \mathcal{B}\Psi) u_n, e_k \rangle = \langle f, e_k \rangle, \quad |k| \in \mathbb{Z}_n. \tag{2.6}$$

Newton’s method is a standard iterative method for solving nonlinear system (2.6). However, at each iteration step of this method we have to establish and invert the Jacobian matrix, which is usually dense and has the size of $(2n - 1) \times (2n - 1)$. Moreover, each entry of the Jacobian matrix is of the form

$$\langle e_l, e_k \rangle - \langle \mathcal{K}e_l, e_k \rangle + \langle (\mathcal{A} + \mathcal{B}) \Psi' (u_n^{(m)}) e_l, e_k \rangle, \quad |k|, |l| \in \mathbb{Z}_n, \tag{2.7}$$

which involves double integrals. Hence, solving (2.6) by Newton’s method demands a large amount of computational efforts.

To develop a fast algorithm for solving (2.6), we first need to consider the computation of the Jacobian matrix. We observe from (2.7) that the first term need not be evaluated because of the orthogonality of e_k ’s. For the second term, we need to compute it only once during the iteration. In addition, we will point out later that the quantities $\langle \mathcal{K}e_l, e_k \rangle$, $|k|, |l| \in \mathbb{Z}_n$, can be evaluated efficiently by compressing the matrix representation of operator \mathcal{K} . In fact, it is the third item that involves the nonlinear operator and carries the main computational difficulties. Motivated by the operator decomposition and the fact that the Fourier basis functions are the eigenfunctions of operator \mathcal{A} , we consider overcoming the difficulties of evaluating the third term by projecting the nonlinear terms $\Psi'(u_n^{(m)})$ into the subspace \mathbb{X}_n . By doing so, we will transform the computation of double singular integrals into setting up the matrix representations of \mathcal{A} and \mathcal{B} and the multiplication of matrices and vectors.

To implement the above idea, we do not solve (2.4) directly and instead, we consider solving the following equation

$$\hat{u}_n - \mathcal{P}_n (\mathcal{K} - \mathcal{A}\mathcal{P}_n \Psi - \mathcal{B}\mathcal{P}_n \Psi) \hat{u}_n = \mathcal{P}_n f \tag{2.8}$$

for a solution

$$\hat{u}_n := \sum_{|l| \in \mathbb{Z}_n} \hat{a}_l e_l \in \mathbb{X}_n.$$

By proofs similar to those of Theorem 2.1 and Corollary 2.2, we obtain in the following theorem regarding the existence and the error estimation of the solution of Eq. (2.8).

Theorem 2.3 *Suppose that $u \in L^2(I)$ is an isolated solution of (2.3) and 1 is not an eigenvalue of the linear operator $\mathcal{K} - ((\mathcal{A} + \mathcal{B})\Psi)'(u)$. Then for sufficiently large n , Eq. (2.8) has a unique solution $\hat{u}_n \in B(u, \delta)$ for some $\delta > 0$ and there exists a constant $c > 0$ such that*

$$\|u - \hat{u}_n\| \leq c \|u - \mathcal{P}_n u\|.$$

Furthermore, if the solution $u \in H^\mu(I)$, $\mu \geq 0$, then there exists a constant $c > 0$ such that

$$\|u - \hat{u}_n\| \leq cn^{-\mu} \|u\|_\mu.$$

In the rest of this section, we establish a matrix form of the nonlinear equation (2.8) and then describe Newton’s method for solving the resulting nonlinear system, from which one will see the advantages of introducing the projection operator. In terms of the Fourier basis of \mathbb{X}_n , Eq. (2.8) is equivalent to the nonlinear system

$$\langle \hat{u}_n - (\mathcal{K} - \mathcal{AP}_n\Psi - \mathcal{BP}_n\Psi) \hat{u}_n, e_k \rangle = \langle f, e_k \rangle, \quad |k| \in \mathbb{Z}_n. \tag{2.9}$$

In order to rewrite (2.9) in a matrix form, we begin with describing the matrix representation of the operators appearing in (2.9) under the Fourier basis. In this paper, we write $\mathbf{v} = [v_k, v_{-k} : k \in \mathbb{Z}_n^+]$ to denote the vector $\mathbf{v} = [v_1, v_{-1}, \dots, v_{n-1}, v_{-(n-1)}]$ and $\mathbf{v} = [v_k, v_{-k} : k \in \mathbb{Z}_n]$ to denote the vector $\mathbf{v} = [v_0, v_1, v_{-1}, \dots, v_{n-1}, v_{-(n-1)}]$. For each $|k|, |l| \in \mathbb{Z}_n$, we let

$$K_{k,l} := \int_I \int_I K(t, \tau) e_k(t) e_l(\tau) dt d\tau$$

and introduce 2×2 matrix

$$\mathbf{K}_{k,l} := \begin{bmatrix} K_{k,l} & K_{k,-l} \\ K_{-k,l} & K_{-k,-l} \end{bmatrix}.$$

By defining the matrix blocks

$$\begin{aligned} \mathbf{K} &:= [\mathbf{K}_{k,l} : k, l \in \mathbb{Z}_n^+], \quad \mathbf{K}' := [K_{0,0}], \quad \mathbf{K}'' := [K_{0,k}, K_{0,-k} : k \in \mathbb{Z}_n^+], \\ \mathbf{K}''' &:= [K_{k,0}, K_{-k,0} : k \in \mathbb{Z}_n^+]^T, \end{aligned}$$

the matrix representation of operator \mathcal{K} can be written as

$$\mathbf{K}_n = \begin{bmatrix} \mathbf{K}' & \mathbf{K}'' \\ \mathbf{K}''' & \mathbf{K} \end{bmatrix}.$$

Likewise, the matrix representation of operators \mathcal{A} and \mathcal{B} can be described by replacing the kernel K by A and B , respectively. Since for each $k \in \mathbb{Z}$, e_k is an eigenfunction of the operator \mathcal{A} , the matrix \mathbf{A}_n is diagonal. Specifically, there holds

$$\mathbf{A}_n := \text{diag} \left(1, 1, 1, \frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{n-1}, \frac{1}{n-1} \right).$$

It follows that

$$\mathcal{P}_n \Psi \hat{u}_n(t) = \sum_{|k| \in \mathbb{Z}_n} \hat{b}_k e_k(t), \quad t \in I,$$

where $\hat{b}_k := \langle \Psi \hat{u}_n, e_k \rangle$, $|k| \in \mathbb{Z}_n$. Associated with $\hat{u}_n = \sum_{|l| \in \mathbb{Z}_n} \hat{a}_l e_l$ we introduce two vectors

$$\hat{\mathbf{u}}_n := [\hat{a}_k, \hat{a}_{-k} : k \in \mathbb{Z}_n]^T \quad \text{and} \quad \hat{\mathbf{v}}_n := [\hat{b}_k, \hat{b}_{-k} : k \in \mathbb{Z}_n]^T.$$

We note that each \hat{b}_k is a nonlinear function of $2n - 1$ variables \hat{a}_l , $|l| \in \mathbb{Z}_n$. For each $|k| \in \mathbb{Z}_n$, we set

$$f_k := \langle f, e_k \rangle \quad \text{and} \quad \mathbf{f}_n := [f_k, f_{-k} : k \in \mathbb{Z}_n]^T.$$

Let \mathbf{E}_n denote the identity matrix which has the same order as matrix \mathbf{K}_n . By using the matrix and vector notation, we have the equivalent matrix form of (2.9)

$$(\mathbf{E}_n - \mathbf{K}_n) \hat{\mathbf{u}}_n + (\mathbf{A}_n + \mathbf{B}_n) \hat{\mathbf{v}}_n = \mathbf{f}_n. \tag{2.10}$$

For solving Eq. (2.10), we describe the classic Newton’s method in the following algorithm. For any $u \in L^2(I)$ and each $k, l \in \mathbb{Z}$, we let $\psi_{k,l}(u) := \langle \Psi'(u) e_k, e_l \rangle$.

Algorithm 1: The Newton Iteration Scheme

Set $m := 0$, $\mathbf{f}_n := [\langle f, e_k \rangle, \langle f, e_{-k} \rangle : k \in \mathbb{Z}_n]^T$ and choose an initial guess $\hat{\mathbf{u}}_n^{(0)}$ and an iteration stopping threshold δ .

Step 1: Let $\hat{u}_n^{(m)} := \sum_{|k| \in \mathbb{Z}_n} (\hat{\mathbf{u}}_n^{(m)})_k e_k$. Compute 2×2 matrices

$$F_{k,l} := \begin{bmatrix} \psi_{k,l}(\hat{u}_n^{(m)}) & \psi_{k,-l}(\hat{u}_n^{(m)}) \\ \psi_{-k,l}(\hat{u}_n^{(m)}) & \psi_{-k,-l}(\hat{u}_n^{(m)}) \end{bmatrix}, \quad k, l \in \mathbb{Z}_n^+,$$

and vectors $F' := [\psi_{0,0}(\hat{u}_n^{(m)})]$ and

$$F'' := [\psi_{0,k}(\hat{u}_n^{(m)}), \psi_{0,-k}(\hat{u}_n^{(m)}) : k \in \mathbb{Z}_n^+],$$

$$F''' := [\psi_{k,0}(\hat{u}_n^{(m)}), \psi_{-k,0}(\hat{u}_n^{(m)}) : k \in \mathbb{Z}_n^+]^T.$$

Set

$$F_1 := [F_{k,l} : k, l \in \mathbb{Z}_n^+], \quad \mathbf{F}_n^{(m)} := \begin{bmatrix} F' & F'' \\ F''' & F_1 \end{bmatrix}$$

and compute the Jacobian matrix

$$\mathbf{J}(\hat{\mathbf{u}}_n^{(m)}) := \mathbf{E}_n - \mathbf{K}_n + (\mathbf{A}_n + \mathbf{B}_n) \mathbf{F}_n^{(m)}.$$

Step 2: Compute

$$\hat{\mathbf{v}}_n^{(m)} := [\langle \Psi(\hat{u}_n^{(m)}), e_k \rangle, \langle \Psi(\hat{u}_n^{(m)}), e_{-k} \rangle : k \in \mathbb{Z}_n]^T$$

and

$$\mathbf{F}(\hat{\mathbf{u}}_n^{(m)}) := (\mathbf{E}_n - \mathbf{K}_n)\hat{\mathbf{u}}_n^{(m)} + (\mathbf{A}_n + \mathbf{B}_n)\hat{\mathbf{v}}_n^{(m)} - \mathbf{f}_n.$$

Step 3: Solve $\Delta_n^{(m)}$ from $\mathbf{J}(\hat{\mathbf{u}}_n^{(m)})\Delta_n^{(m)} = -\mathbf{F}(\hat{\mathbf{u}}_n^{(m)})$ and compute $\hat{\mathbf{u}}_n^{(m+1)} := \hat{\mathbf{u}}_n^{(m)} + \Delta_n^{(m)}$.

Step 4: Set $m \leftarrow m + 1$ and go back to Step 1 until $\|\Delta_n^{(m)}\|_\infty < \delta$.

We observe from Algorithm 1 that through introducing the projection on the nonlinear term $\Psi(u_n)$, we reduce the difficulties of computing the Jacobian matrices $\mathbf{J}(\hat{\mathbf{u}}_n^{(m)})$ and $\mathbf{F}(\hat{\mathbf{u}}_n^{(m)})$ in each iteration step of Newton’s method. In fact, instead of computing double integrals involved in the entries of $\mathbf{J}(\hat{\mathbf{u}}_n^{(m)})$ and $\mathbf{F}(\hat{\mathbf{u}}_n^{(m)})$ in each iteration step, we only need to set up the matrices \mathbf{K}_n , \mathbf{A}_n and \mathbf{B}_n once, evaluate the single integrals $\langle \Psi'(\hat{u}_n^{(m)})e_k, e_l \rangle$ and $\langle \Psi(\hat{u}_n^{(m)}), e_k \rangle$, $|k|, |l| \in \mathbb{Z}_n$ and then compute the multiplications of matrices and vectors. Especially for the operator \mathcal{A} , we even avoid calculating the double singular integrals because of the eigenfunctions property of \mathcal{A} .

In spite of this, we still face large computational costs for solving nonlinear system (2.10) by Algorithm 1. Although matrix \mathbf{A}_n is diagonal and its entries on the diagonal are known, we still have to treat matrices \mathbf{K}_n and \mathbf{B}_n . By the smoothness of the kernel functions K and B , matrices \mathbf{K}_n and \mathbf{B}_n are usually dense. The density will lead to large computational costs for setting up the matrices and in each iteration step, computing the multiplications of the matrices and some vectors. Moreover, since the Jacobian matrix $\mathbf{J}(\hat{\mathbf{u}}_n^{(m)})$ and $\mathbf{F}(\hat{\mathbf{u}}_n^{(m)})$ need to be updated during the iteration, the generating of vectors and the multiplications of vectors and dense matrices have to be carried out in each iteration step. All these demand the availability of a fast solver for the nonlinear system (2.10).

3 Exponential Convergence Order for Analytic Solutions

We consider fast solutions of the nonlinear system (2.10) on two cases according to the regularity of the exact solution of (2.3). We discuss in this section the case of periodic analytic solutions and postpone the case of solutions with general regularity to the next section.

It is known [1, 18] that the projection error of a periodic analytic function decays exponentially. That is, for a periodic analytic function v , there exist positive constants c and s such that

$$\|v - \mathcal{P}_n v\| \leq ce^{-sn}. \tag{3.1}$$

This together with the error estimate given in Theorem 2.3 leads to the following result about the convergence property of the approximate solution of (2.8):

Corollary 3.1 *If the solution u is a periodic analytic function, there exist positive constants c and s such that*

$$\|u - \hat{u}_n\| \leq ce^{-sn}.$$

The above corollary shows that the Fourier-Galerkin method (2.8) enjoys the exponential convergence order if the exact solution of the boundary integral equation is analytic. Therefore, for this special case, there is no need to develop a fast solver for the Fourier-Galerkin method, since for a small n the approximate solution \hat{u}_n will get enough accuracy.

To close this section, we present a numerical example for the case of periodic analytic solutions to illustrate the exponential convergence of the approximate solution of (2.8).

Table 1 Numerical results of Algorithm 1 for analytic solutions

n	d_n	Err
4	7	8.71e-2
6	11	4.60e-3
8	15	1.31e-4
10	19	2.47e-6
12	23	3.49e-8
14	27	3.70e-10
16	31	3.03e-12

All computer programs for the numerical experiments presented in this paper are run on a personal computer with 2.8G CPU and 8G memory.

Example 3.2 We consider solving the nonlinear boundary integral equations (2.2) reformulated from the boundary value problem (1.1) in an elliptical region

$$D : x_1^2 + \frac{x_2^2}{4} < 1.$$

Utilizing the parametrization of the boundary Γ as $x(t) = (\cos t, 2 \sin t)$, $t \in I$, we establish the boundary integral equation (2.2) with the kernel functions K , A and B defined respectively by

$$K(t, \tau) = \frac{1}{2\pi} \frac{2}{\sin^2 \frac{t+\tau}{2} + 4 \cos^2 \frac{t+\tau}{2}}$$

and

$$A(t, \tau) = -\frac{1}{\pi} \log \left| 2e^{-1/2} \sin \frac{t-\tau}{2} \right|,$$

$$B(t, \tau) = -\frac{1}{2\pi} \left[1 + \log \left(\sin^2 \frac{t+\tau}{2} + 4 \cos^2 \frac{t+\tau}{2} \right) \right].$$

We assume that the nonlinear function g is with the form $g(t, u(t)) := u(t) + \sin(u(t))$ and choose g_0 so that the function

$$u(t) = e^{\cos(t)} \cos(2 \sin(t)), \quad t \in I,$$

is the exact solution of (2.2). Here, we note that the solution u is analytic.

In this example, the integral equation is solved directly by Algorithm 1. We report the numerical results in Table 1. The second column in Table 1 lists the dimensions of the approximate subspaces. We present the relative error ‘‘Err’’ of the approximate solution in the third column, which is computed by $\text{Err} := \frac{\|u - \hat{u}_n\|}{\|u\|}$. We observe from the numerical results that the approximation error decays quite fast so that for $n = 16$ the approximate solution \hat{u}_n has already got enough accuracy. Hence, we need not to consider a fast solver for the Fourier-Galerkin method.

4 Fast Fully Discrete Fourier-Galerkin Method

We know from Theorem 2.3 that if $u \in H^\mu(I)$, $\mu \geq 0$, the approximation error of \hat{u}_n can not decay as fast as the error in the case of analytic solutions. Hence, in this section when the

exact solution has a general regularity, we develop a fast solver for the nonlinear system (2.10) which requires only nearly linear number of multiplications to implement it. This will be done by implementing three approximation techniques. They include a matrix truncation strategy, a quadrature formula for oscillatory integrals, and the multilevel augmentation method. We then show theoretically and numerically that the use of these approximation techniques does not ruin the convergence order of the conventional Fourier-Galerkin method.

4.1 A Fast Solver for the Nonlinear System

To develop a fast algorithm, we first compress the matrices \mathbf{K}_n and \mathbf{B}_n to sparse ones by using the matrix truncation strategy proposed in [5]. For each $n \in \mathbb{N}$, we introduce the index set

$$\mathbb{L}_n := \{[k, l] \in \mathbb{Z}_n^2 : kl \leq n\},$$

which will be used to compress the matrices. Specifically, we define the truncation matrix of \mathbf{K}_n by setting

$$\tilde{\mathbf{K}}_{k,l} := \begin{cases} \mathbf{K}_{k,l}, & [k, l] \in \mathbb{L}_n \cap (\mathbb{Z}_n^+)^2, \\ \mathbf{0}_{2 \times 2}, & \text{otherwise,} \end{cases} \quad \tilde{\mathbf{K}} := [\tilde{\mathbf{K}}_{k,l} : k, l \in \mathbb{Z}_n^+]$$

and letting

$$\tilde{\mathbf{K}}_n := \begin{bmatrix} \mathbf{K}' & \mathbf{K}'' \\ \mathbf{K}''' & \tilde{\mathbf{K}} \end{bmatrix}.$$

In the same way, we also define the truncation matrix of \mathbf{B}_n . Clearly, the matrices $\tilde{\mathbf{K}}_n$ and $\tilde{\mathbf{B}}_n$ are sparse. In Eq. (2.10), we replace the dense matrices \mathbf{K}_n and \mathbf{B}_n by $\tilde{\mathbf{K}}_n$ and $\tilde{\mathbf{B}}_n$, respectively. We then obtain the truncated nonlinear system

$$(\mathbf{E}_n - \tilde{\mathbf{K}}_n)\tilde{\mathbf{u}}_n + (\mathbf{A}_n + \tilde{\mathbf{B}}_n)\tilde{\mathbf{v}}_n = \mathbf{f}_n, \tag{4.1}$$

where $\tilde{\mathbf{u}}_n := [\tilde{a}_k, \tilde{a}_{-k} : k \in \mathbb{Z}_n]^T$ is the corresponding solution to be determined and $\tilde{\mathbf{v}}_n := [\tilde{b}_k, \tilde{b}_{-k} : k \in \mathbb{Z}_n]^T$ with

$$\tilde{b}_k := \left\langle \Psi \left(\sum_{|l| \in \mathbb{Z}_n} \tilde{a}_l e_l(t) \right), e_k \right\rangle.$$

Before solving numerically the truncated nonlinear system (4.1), we are required to efficiently evaluate the nonzero entries of matrices $\tilde{\mathbf{K}}_n$ and $\tilde{\mathbf{B}}_n$, defined by the integrals

$$\begin{aligned} K_{k,l} &:= \int_I \int_I K(t, \tau) e_k(t) e_l(\tau) dt d\tau, \\ B_{k,l} &:= \int_I \int_I B(t, \tau) e_k(t) e_l(\tau) dt d\tau, \quad |k|, |l| \in \mathbb{L}_n. \end{aligned} \tag{4.2}$$

It is clear that we have to compute the oscillatory integrals when $k \gg 1$ or $l \gg 1$. An efficient quadrature formula for dealing with the integrals in (4.2) was developed in [14] by using the notion of the refinable set which was initiated in [9]. Following [9, 14], we first construct multi-scale piecewise Lagrange interpolations of kernel functions K and B on sparse grids. The multi-scale piecewise Lagrange interpolation requires the interpolation points used in

the current scale contain those used in the previous scale. To this end, we need the notion of the refinable set. We introduce two contractive mappings from \mathbb{R} to \mathbb{R} by

$$\varphi_0(x) := \frac{x}{2}, \quad \varphi_1(x) := \frac{x + 2\pi}{2}, \quad x \in I,$$

and set $\Phi := \{\varphi_\varrho : \varrho \in \mathbb{Z}_2\}$. It is clear that I is the invariant set related to the mappings Φ . A subset V of I is called a refinable set relative to the mappings Φ if $V \subseteq \Phi(V)$.

To establish the multi-scale piecewise Lagrange interpolation formula, we need to construct the Lagrange polynomials on I . For $m \in \mathbb{N}$, we assume that $V := \{v_r : 0 < v_0 < v_1 < \dots < v_{m-1} < 2\pi, r \in \mathbb{Z}_m\}$ is refinable relative to Φ . These points will be used to construct the Lagrange interpolation polynomial of the initial scale. The set $\Phi^j(V)$ is used to generate the interpolation points of for the Lagrange interpolation polynomials of scale j . Specifically, for any $N \in \mathbb{N}$ and $\mathbf{p}_N := [\varrho_\gamma : \gamma \in \mathbb{Z}_N] \in \mathbb{Z}_2^N$, we let

$$\varphi_{\mathbf{p}_N} := \varphi_{\varrho_{N-1}} \circ \dots \circ \varphi_{\varrho_0} \text{ and } \mu(\mathbf{p}_N) := \sum_{\gamma \in \mathbb{Z}_N} \varrho_\gamma 2^\gamma.$$

For any $N \in \mathbb{N}$ and $r \in \mathbb{Z}_{2^N m}$, we define $v_{N,r} := \varphi_{\mathbf{p}_N}(v_{r'})$, where $\mathbf{p}_N \in \mathbb{Z}_2^N$ and $r' \in \mathbb{Z}_m$ satisfy $r = m\mu(\mathbf{p}_N) + r'$. For any $r \in \mathbb{Z}_m$, we also let $v_{0,r} := v_r$. We set $V_N := \{v_{N,r}, r \in \mathbb{Z}_{2^N m}\}$, $N \in \mathbb{N}_0$, and accordingly define the Lagrange polynomials of degree $m - 1$ on I by

$$l_{0,r}(x) := \prod_{q=0, q \neq r}^{m-1} \frac{x - v_{0,q}}{v_{0,r} - v_{0,q}}, \quad x \in I \text{ and } r \in \mathbb{Z}_m,$$

and

$$l_{N,r}(x) := \begin{cases} \prod_{q=m\mu(\mathbf{p}_N), q \neq r}^{m\mu(\mathbf{p}_N)+m-1} \frac{x - v_{N,q}}{v_{N,r} - v_{N,q}}, & x \in \varphi_{\mathbf{p}_N}(I), \\ 0, & x \in I \setminus \varphi_{\mathbf{p}_N}(I), \end{cases} \quad N \geq 1 \text{ and } r \in \mathbb{Z}_{2^N m},$$

where $\mathbf{p}_N \in \mathbb{Z}_2^N$ and $r' \in \mathbb{Z}_m$ satisfy $r = m\mu(\mathbf{p}_N) + r'$. We also need to define some useful linear functionals on $C(I)$. For any $r \in \mathbb{Z}_m$ and $\kappa \in \mathbb{Z}_{2m}$, let $a_{r,\kappa} := l_{0,r}(v_{1,\kappa})$. We define linear functionals $\eta_{j,r}$, $j \in \mathbb{N}_0$, $r \in \mathbb{Z}_{2^j m}$, by

$$\begin{aligned} \eta_{0,r}(f) &:= f(v_{0,r}), \\ \eta_{j,r}(f) &:= f(v_{j,r}) - \sum_{q \in \mathbb{Z}_m} f(v_{j-1, m\lfloor \frac{r}{2^m} \rfloor + q}) a_{q,r} \text{ mod } 2m, \quad \text{for all } f \in C(I). \end{aligned}$$

Here, we denote by $\lfloor x \rfloor$ the largest integer not greater than x .

Making use of the Lagrange polynomials and the linear functionals defined above, we now describe the multi-scale piecewise Lagrange interpolation of kernel function K on sparse grids. We define index sets $\mathbb{W}_0 := \mathbb{Z}_m$ and for any $N \in \mathbb{N}$, $\mathbb{W}_N := \{r \in \mathbb{Z}_{2^N m} : v_{N,r} \in V_N \setminus V_{N-1}\}$. For any $\mathbf{j} := [j_0, j_1] \in \mathbb{N}_0^2$, we let $\mathbb{W}_{\mathbf{j}} := \mathbb{W}_{j_0} \otimes \mathbb{W}_{j_1}$, and for any $\mathbf{j} := [j_0, j_1] \in \mathbb{N}_0^2$ and $\mathbf{r} := [r_0, r_1] \in \mathbb{W}_{\mathbf{j}}$, we set

$$\eta_{\mathbf{j},\mathbf{r}} := \eta_{j_0,r_0} \otimes \eta_{j_1,r_1} \text{ and } l_{\mathbf{j},\mathbf{r}} := l_{j_0,r_0} \otimes l_{j_1,r_1}.$$

For any $N \in \mathbb{N}$, we introduce the index set by

$$\mathbb{S}_N := \{\mathbf{j} := [j_0, j_1] \in \mathbb{Z}_{N+1}^2 : \max\{j_0, 1\} + \max\{j_1, 1\} \leq N\}.$$

With these notations, we get the multi-scale piecewise Lagrange interpolation of K on sparse grids

$$S_N K = \sum_{\mathbf{j} \in \mathbb{S}_N} \sum_{\mathbf{r} \in \mathbb{W}_{\mathbf{j}}} \eta_{\mathbf{j},\mathbf{r}}(K) l_{\mathbf{j},\mathbf{r}}, \quad N \in \mathbb{N}.$$

Replacing K by $S_N K$ in the second term of (4.2), we establish a numerical quadrature formula for evaluating the coefficients $K_{k,l}, |k|, |l| \in \mathbb{L}_n$, as

$$\tilde{K}_{k,l} := \sum_{\mathbf{j} \in \mathbb{S}_N} \sum_{\mathbf{r} \in \mathbb{W}_{\mathbf{j}}} \eta_{\mathbf{j},\mathbf{r}}(K) K_{\mathbf{j},\mathbf{r}}(k, l), \tag{4.3}$$

where for any $\mathbf{j} \in \mathbb{S}_N, \mathbf{r} \in \mathbb{W}_{\mathbf{j}}$ and $|k|, |l| \in \mathbb{L}_n$,

$$K_{\mathbf{j},\mathbf{r}}(k, l) := \int_I \int_I l_{\mathbf{j},\mathbf{r}}(t, \tau) e_k(t) e_l(\tau) dt d\tau.$$

We note that by rewriting (4.3) in the form of discrete Fourier transforms of matrices $[\eta_{\mathbf{j},\mathbf{r}}(K) : \mathbf{r} \in \mathbb{W}_{\mathbf{j}}], \mathbf{j} \in \mathbb{S}_N$, and then employing the well-known fast Fourier transform, we can design a fast quadrature algorithm for computing $\tilde{K}_{k,l}, |k|, |l| \in \mathbb{L}_n$. For a detailed description of the algorithm, the readers are referred to Appendix in [15]. We denote by $\tilde{\mathbf{K}}_{n,N}$ the matrix $\tilde{\mathbf{K}}_n$ with $K_{k,l}$ being evaluated by the quadrature formula (4.3), where N is the number of partition level of the multiscale Lagrange interpolation for kernel K . Likewise, we also define matrix $\tilde{\mathbf{B}}_{n,N}$. We thus obtain a fully discrete truncated nonlinear system

$$(\mathbf{E}_n - \tilde{\mathbf{K}}_{n,N}) \tilde{\mathbf{u}}_{n,N} + (\mathbf{A}_n + \tilde{\mathbf{B}}_{n,N}) \tilde{\mathbf{v}}_{n,N} = \mathbf{f}_n. \tag{4.4}$$

We next consider solving Eq. (4.4) by the multilevel augmentation method. The key idea of the multilevel augmentation method is to solve the nonlinear system by inverting the nonlinear operator in a subspace of a much lower dimension with a compensation of high frequency by matrix–vector multiplications. To apply this idea to Eq. (4.4), we assume that $n := 2^{k+m}, k, m \in \mathbb{N}$, where k is fixed and denotes the initial level of approximation, and we split the matrices in this equation into “low frequency” and “high frequency” components. Specifically, we write the matrix $\tilde{\mathbf{K}}_{n,N}$ in the block form

$$\tilde{\mathbf{K}}_{n,N} = \begin{bmatrix} \overline{K}_{00} & \overline{K}_{01} & \cdots & \overline{K}_{0m} \\ \overline{K}_{10} & \overline{K}_{11} & \cdots & \overline{K}_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ \overline{K}_{m0} & \overline{K}_{m1} & \cdots & \overline{K}_{mm} \end{bmatrix},$$

where \overline{K}_{00} has order $(2^{k+1} - 1) \times (2^{k+1} - 1)$, \overline{K}_{0q} has order $(2^{k+1} - 1) \times 2^{k+q}$, \overline{K}_{q0} has order $2^{k+q} \times (2^{k+1} - 1)$ for $q = 1, 2, \dots, m$ and \overline{K}_{pq} has order $2^{k+p} \times 2^{k+q}$ for $p, q = 1, 2, \dots, m$. For each $l = 0, 1, \dots, m$, we denote by $\tilde{\mathbf{K}}_{k,l,N}$ the submatrix of $\tilde{\mathbf{K}}_{n,N}$ that has the form

$$\tilde{\mathbf{K}}_{k,l,N} = \begin{bmatrix} \overline{K}_{00} & \overline{K}_{01} & \cdots & \overline{K}_{0l} \\ \overline{K}_{10} & \overline{K}_{11} & \cdots & \overline{K}_{1l} \\ \cdots & \cdots & \cdots & \cdots \\ \overline{K}_{l0} & \overline{K}_{l1} & \cdots & \overline{K}_{ll} \end{bmatrix},$$

and split $\tilde{\mathbf{K}}_{k,l,N}$ into its “low frequency” and “high frequency” components:

$$\tilde{\mathbf{K}}_{k,l,N}^L = [\bar{K}_{00} \ \bar{K}_{01} \ \cdots \ \bar{K}_{0l}] \quad \text{and} \quad \tilde{\mathbf{K}}_{k,l,N}^H = \begin{bmatrix} \bar{K}_{10} & \bar{K}_{11} & \cdots & \bar{K}_{1l} \\ \cdots & \cdots & \cdots & \cdots \\ \bar{K}_{l0} & \bar{K}_{l1} & \cdots & \bar{K}_{ll} \end{bmatrix}.$$

By the same way, we also introduce the submatrices and corresponding “low frequency” components and “high frequency” components for matrices \mathbf{E}_n , \mathbf{A}_n and $\tilde{\mathbf{B}}_{n,N}$. In the light of the idea of the multilevel augmentation methods, we establish the following fast algorithm.

Algorithm 2: The Fast Multilevel Augmentation Method

Let k be a fixed positive integer. Set $\mathbf{f}_{2^{k+m}} := [\langle f, \mathbf{e}_j \rangle, \langle f, \mathbf{e}_{-j} \rangle : j \in \mathbb{Z}_{2^{k+m}}]^T$.

Step 1: According to Algorithm 1, solve the nonlinear system

$$(\mathbf{E}_{2^k} - \tilde{\mathbf{K}}_{k,0,N})\mathbf{u}_{k,0,N} + (\mathbf{A}_{2^k} + \tilde{\mathbf{B}}_{k,0,N})\mathbf{v}_{k,0,N} = \mathbf{f}_{2^k},$$

for the solution $\mathbf{u}_{k,0,N} = [(\mathbf{u}_{k,0,N})_j, (\mathbf{u}_{k,0,N})_{-j} : j \in \mathbb{Z}_{2^k}]^T$, where

$$\mathbf{v}_{k,0,N} := [\langle \Psi(\mathbf{u}_{k,0,N}), \mathbf{e}_j \rangle, \langle \Psi(\mathbf{u}_{k,0,N}), \mathbf{e}_{-j} \rangle : j \in \mathbb{Z}_{2^k}]^T$$

and $u_{k,0,N} := \sum_{|j| \in \mathbb{Z}_{2^k}} (\mathbf{u}_{k,0,N})_j \mathbf{e}_j$. Set $l := 1$.

Step 2: Set $\mathbf{f}_{k,l} := [\langle f, \mathbf{e}_j \rangle, \langle f, \mathbf{e}_{-j} \rangle : j \in \mathbb{Z}_{2^{k+l}} \setminus \mathbb{Z}_{2^k}]^T$.

- Compute $\hat{\mathbf{v}}_{k,l,N} = [\langle \Psi(\mathbf{u}_{k,l-1,N}), \mathbf{e}_j \rangle, \langle \Psi(\mathbf{u}_{k,l-1,N}), \mathbf{e}_{-j} \rangle : j \in \mathbb{Z}_{2^{k+l}}]^T$.
- Augment the matrices $\tilde{\mathbf{K}}_{k,l-1,N}^H$, $\mathbf{A}_{k,l-1}^H$ and $\tilde{\mathbf{B}}_{k,l-1,N}^H$ to form $\tilde{\mathbf{K}}_{k,l,N}^H$, $\mathbf{A}_{k,l}^H$ and $\tilde{\mathbf{B}}_{k,l,N}^H$, respectively.
- Augment the vector $\mathbf{u}_{k,l-1,N}$ by setting $\hat{\mathbf{u}}_{k,l,N} = \begin{bmatrix} \mathbf{u}_{k,l-1,N} \\ \mathbf{0}_{2^{k+l} \times 1} \end{bmatrix}$.
- Compute

$$\mathbf{u}_{k,l,N}^H := \mathbf{f}_{k,l} + \tilde{\mathbf{K}}_{k,l,N}^H \hat{\mathbf{u}}_{k,l,N} - (\mathbf{A}_{k,l}^H + \tilde{\mathbf{B}}_{k,l,N}^H) \hat{\mathbf{v}}_{k,l,N}. \tag{4.5}$$

Let $u_{k,l,N}^H := \sum_{|j| \in \mathbb{Z}_{2^{k+l}} \setminus \mathbb{Z}_{2^k}} (\mathbf{u}_{k,l,N}^H)_j \mathbf{e}_j$.

Step 3: Augment the matrices $\tilde{\mathbf{K}}_{k,l-1,N}^L$, $\mathbf{A}_{k,l-1}^L$ and $\tilde{\mathbf{B}}_{k,l-1,N}^L$ to form $\tilde{\mathbf{K}}_{k,l,N}^L$, $\mathbf{A}_{k,l}^L$ and $\tilde{\mathbf{B}}_{k,l,N}^L$, respectively. According to Algorithm 1, solve the nonlinear system

$$(\mathbf{E}_{k,l}^L - \tilde{\mathbf{K}}_{k,l,N}^L) \begin{bmatrix} \mathbf{u}_{k,l,N}^L \\ \mathbf{u}_{k,l,N}^H \end{bmatrix} + (\mathbf{A}_{k,l}^L + \tilde{\mathbf{B}}_{k,l,N}^L) \bar{\mathbf{v}}_{k,l,N} = \mathbf{f}_{2^k} \tag{4.6}$$

for the solution $\mathbf{u}_{k,l,N}^L = [(\mathbf{u}_{k,l,N}^L)_j, (\mathbf{u}_{k,l,N}^L)_{-j} : j \in \mathbb{Z}_{2^k}]^T$, where

$$\bar{\mathbf{v}}_{k,l,N} := [\langle \Psi(\mathbf{u}_{k,l,N}^L + u_{k,l,N}^H), \mathbf{e}_j \rangle, \langle \Psi(\mathbf{u}_{k,l,N}^L + u_{k,l,N}^H), \mathbf{e}_{-j} \rangle : j \in \mathbb{Z}_{2^{k+l}}]^T$$

and $u_{k,l,N}^L := \sum_{|j| \in \mathbb{Z}_{2^k}} (\mathbf{u}_{k,l,N}^L)_j \mathbf{e}_j$. Define $\mathbf{u}_{k,l,N} := \begin{bmatrix} \mathbf{u}_{k,l,N}^L \\ \mathbf{u}_{k,l,N}^H \end{bmatrix}$ and $u_{k,l,N} := u_{k,l,N}^L + u_{k,l,N}^H$.

Step 4: Set $l \leftarrow l + 1$ and go back to Step 2 until $l = m$.

We remark that in Algorithm 2, we use the fast Fourier transform to evaluate the vectors $\bar{\mathbf{v}}_{k,l,N}$ and $\hat{\mathbf{v}}_{k,l,N}$. The output $\mathbf{u}_{k,m,N}$ of Algorithm 2 is regarded as an approximate solution of the nonlinear system (4.4). The next theorem gives an estimation of the computational cost required for Algorithm 2, which shows that the fully discrete Fourier-Galerkin method developed in this section is indeed a fast algorithm.

Theorem 4.1 *If for each $(t, \tau) \in I^2$, the number of multiplications used to evaluate $K(t, \tau)$ and $B(t, \tau)$ is constant, then the total number of multiplications required for obtaining $\mathbf{u}_{k,m,N}$ in Algorithm 2 is $\mathcal{O}((k + m)^3 2^{k+m})$.*

Proof The total number of multiplications required for obtaining $\mathbf{u}_{k,m,N}$ is composed of two parts. One involves the multiplication for generating the sparse matrices $\tilde{\mathbf{K}}_{2^{k+m},N}$ and $\tilde{\mathbf{B}}_{2^{k+m},N}$. The other is that for carrying out the computing steps listed in Algorithm 2.

According to Theorem A.3 of Appendix in [15], the number of multiplications for generating $\tilde{\mathbf{K}}_{2^{k+m},N}$ and $\tilde{\mathbf{B}}_{2^{k+m},N}$ is $\mathcal{O}((k + m)^3 2^{k+m})$. To carry out the computing steps listed in Algorithm 2, we need to compute only once of step 1 and update for $l = 0, 1, 2, \dots, m$ in steps 2 and 3. The computational cost of step 1, used to provide an initial approximate solution, can be considered as a constant. In step 2, the number of multiplications used for computing $\hat{\mathbf{v}}_{k,l,N}$ by using the fast Fourier transform is $\mathcal{O}((k + l)2^{k+l})$. An estimation of the numbers of nonzero entries of the truncated matrices $\tilde{\mathbf{K}}_{k,l,N}^H$ and $\tilde{\mathbf{B}}_{k,l,N}^H$ has been given in [5], that is, they are $\mathcal{O}((k + l)2^{k+l})$. As a result, the number of multiplications in the matrix-vector multiplication is also $\mathcal{O}((k + l)2^{k+l})$. In each iteration step of Newton’s method for solving (4.6), the number of multiplications for evaluating the Jacobian matrix can be obtained as $\mathcal{O}((k + l)2^{k+l})$ by a similar analysis used in step 2. Since the nonlinear system (4.6) has the same size as (4.1), we consider the computational cost for inverting the Jacobian matrix as a constant. So the total number of multiplications in step 3 is $\mathcal{O}((k + l)2^{k+l})$ since the number of iteration steps is finite.

Adding up the above estimates together, we get the desired result of this theorem. □

4.2 Convergence Analysis

The convergence order of the approximate solution obtained by Algorithm 2 can be estimated by a similar analysis used in [7, 8] for estimating the multilevel augmentation method for the multiscale collocation method.

We first represent Algorithm 2 in an operator form. For $n := 2^{k+m}$ and $N := k + m$, we let $\mathcal{K}_n := \mathcal{P}_n \mathcal{K}|_{\mathbb{X}_n}$ and $\tilde{\mathcal{K}}_{n,N} : \mathbb{X}_n \rightarrow \mathbb{X}_n$ be the linear operator such that its matrix representation under the Fourier basis is $\tilde{\mathbf{K}}_{n,N}$. For each $l \in \mathbb{Z}_{m+1}$, we denote by $\tilde{\mathcal{K}}_{k,l,N}$ the linear operator from $\mathbb{X}_{2^{k+l}}$ to $\mathbb{X}_{2^{k+l}}$, which has $\tilde{\mathbf{K}}_{k,l,N}$ as its matrix representation. Similarly, we also define the operators $\mathcal{B}_n, \tilde{\mathcal{B}}_{n,N}$ and $\tilde{\mathcal{B}}_{k,l,N}$ and $\mathcal{A}_{k,l}, l \in \mathbb{Z}_{m+1}$. For each $l \in \mathbb{Z}_{m+1}$, we split the operator $\tilde{\mathcal{K}}_{k,l,N}$ into its “lower frequency” and “higher frequency” components by letting

$$\tilde{\mathcal{K}}_{k,l,N}^L := \mathcal{P}_{2^k} \tilde{\mathcal{K}}_{k,l,N} \mathcal{P}_{2^{k+l}}, \quad \tilde{\mathcal{K}}_{k,l,N}^H := (\mathcal{P}_{2^{k+l}} - \mathcal{P}_{2^k}) \tilde{\mathcal{K}}_{k,l,N} \mathcal{P}_{2^{k+l}}.$$

It is clear that the matrices $\tilde{\mathbf{K}}_{k,l,N}^L$ and $\tilde{\mathbf{K}}_{k,l,N}^H$ are exactly the matrix representation of the operators $\tilde{\mathcal{K}}_{k,l,N}^L$ and $\tilde{\mathcal{K}}_{k,l,N}^H$, respectively. Likewise, by replacing $\tilde{\mathcal{K}}_{k,l,N}$ with $\tilde{\mathcal{B}}_{k,l,N}$ and $\mathcal{A}_{k,l}$, we also introduce $\tilde{\mathcal{B}}_{k,l,N}^L, \tilde{\mathcal{B}}_{k,l,N}^H, \mathcal{A}_{k,l}^L$ and $\mathcal{A}_{k,l}^H$. We point out that for each $l \in \mathbb{Z}_{m+1}$, Eq. (4.5) is equivalent to

$$u_{k,l,N}^H = (\mathcal{P}_{2^{k+l}} - \mathcal{P}_{2^k}) f + \left[\tilde{\mathcal{K}}_{k,l,N}^H - (\mathcal{A}_{k,l}^H + \tilde{\mathcal{B}}_{k,l,N}^H) \mathcal{P}_{2^{k+l}} \Psi \right] u_{k,l-1,N},$$

and nonlinear system (4.6) is equivalent to

$$\left[\mathcal{P}_{2^k} - \tilde{\mathcal{K}}_{k,l,N}^L + \left(\mathcal{A}_{k,l}^L + \tilde{\mathcal{B}}_{k,l,N}^L \right) \mathcal{P}_{2^{k+l}} \Psi \right] \left(u_{k,l,N}^L + u_{k,l,N}^H \right) = \mathcal{P}_{2^k} f.$$

Moreover, it can be verified that the approximate solution $u_{k,l,N}, l \in \mathbb{Z}_{m+1}$, satisfy the nonlinear operator equations

$$\begin{aligned} \left[\mathcal{I} - \tilde{\mathcal{K}}_{k,l,N}^L + \left(\mathcal{A}_{k,l}^L + \tilde{\mathcal{B}}_{k,l,N}^L \right) \mathcal{P}_{2^{k+l}} \Psi \right] u_{k,l,N} &= \mathcal{P}_{2^{k+l}} f + \tilde{\mathcal{K}}_{k,l,N}^H u_{k,l-1,N} \\ &- \left(\mathcal{A}_{k,l}^H + \tilde{\mathcal{B}}_{k,l,N}^H \right) \mathcal{P}_{2^{k+l}} \Psi u_{k,l-1,N}, \quad l \in \mathbb{Z}_{m+1}. \end{aligned} \tag{4.7}$$

We shall need to estimate the differences between \mathcal{K}_n and $\tilde{\mathcal{K}}_{n,N}$ and also between \mathcal{B}_n and $\tilde{\mathcal{B}}_{n,N}$. To this end, we introduce two appropriate spaces of functions. For any $\alpha := [\alpha_0, \alpha_1] \in \mathbb{N}_0^2$, let $|\alpha|_\infty := \max\{\alpha_0, \alpha_1\}$ and $|\alpha| := |\alpha_0| + |\alpha_1|$. For a function $f \in C^\sigma(I^2)$ and for $\alpha := [\alpha_0, \alpha_1] \in \mathbb{N}_0^2$, we write

$$f^{(\alpha)}(\mathbf{x}) := \left(\frac{\partial^{|\alpha|}}{\partial x^{\alpha_0} \partial y^{\alpha_1}} f \right) (\mathbf{x}), \quad \mathbf{x} := [x, y] \in I^2$$

and define the space

$$X^\sigma(I^2) := \left\{ f : I^2 \rightarrow \mathbb{R} : f^{(\alpha)} \in C(I^2), |\alpha|_\infty \leq \sigma \right\}.$$

For $\mu \geq 0$, we also define the space $H^\mu(I^2)$ of functions $\phi \in L^2(I^2)$ whose Fourier coefficients $\{\phi_{k,l} : k, l \in \mathbb{Z}\}$ satisfy

$$\sum_{k \in \mathbb{Z}} \sum_{l \in \mathbb{Z}} (1 + k^2)^\mu (1 + l^2)^\mu |\phi_{k,l}|^2 < \infty.$$

According to Lemma 3.3 in [15], we have the desired result for the difference between \mathcal{K}_n and $\tilde{\mathcal{K}}_{n,N}$ in the following lemma.

Lemma 4.2 *If $N := \lceil \log_2 n \rceil$ and $K \in X^\sigma(I^2) \cap H^\mu(I^2)$ with $\sigma \geq \mu + 1/2 + \epsilon, \mu \geq 0, \epsilon > 0$, then there exist a positive constant c and $n_0 \in \mathbb{N}$ such that for all $n \in \mathbb{N}$ with $n \geq n_0$ and for all $w \in L^2(I)$,*

$$\|(\mathcal{K}_n - \tilde{\mathcal{K}}_{n,N})\mathcal{P}_n w\| \leq c \|w\| n^{-\mu}. \tag{4.8}$$

For the difference between operators \mathcal{B}_n and $\tilde{\mathcal{B}}_{n,N}$, we also have a similar estimate.

Lemma 4.3 *If $N := \lceil \log_2 n \rceil$ and $B \in X^\sigma(I^2) \cap H^\mu(I^2)$ with $\sigma \geq \mu + 1/2 + \epsilon, \mu \geq 0, \epsilon > 0$, then there exist a positive constant c and $n_0 \in \mathbb{N}$ such that for all $n \in \mathbb{N}$ with $n \geq n_0$ and for all $w \in L^2(I)$,*

$$\|(\mathcal{B}_n - \tilde{\mathcal{B}}_{n,N})\mathcal{P}_n \Psi w\| \leq c \|w\| n^{-\mu}. \tag{4.9}$$

Proof By an estimate similar to that presented in Lemma 4.2, there exist a positive constant c_1 and $n_0 \in \mathbb{N}$ such that for all $n \in \mathbb{N}$ with $n \geq n_0$ and for all $w \in L^2(I)$

$$\|(\mathcal{B}_n - \tilde{\mathcal{B}}_{n,N})\mathcal{P}_n \Psi w\| \leq c_1 \|\Psi w\| n^{-\mu}. \tag{4.10}$$

Since g is continuously differentiable with respect to the second variable, it follows from the standard estimate for nonlinear operator (see, for Example, [23]) that there exists a positive constant c_2 such that for all $w \in L^2(I)$

$$\|\Psi w\| \leq c_2 \|w\|. \tag{4.11}$$

Combining (4.11) and (4.10), we obtain that

$$\|(\mathcal{B}_n - \tilde{\mathcal{B}}_{n,N})\mathcal{P}_n \Psi w\| \leq c_1 c_2 \|w\| n^{-\mu},$$

which prove the estimation (4.9) by letting $c := c_1 c_2$. □

By making use of Lemmas 4.2 and 4.3, we estimate in the following theorem the error of the approximate solution generated by Algorithm 2.

Theorem 4.4 *If kernel functions K and $B \in X^\sigma(I^2) \cap H^\mu(I^2)$ with $\sigma \geq \mu + 1/2 + \epsilon$, $\mu \geq 0$, $\epsilon > 0$ and $u \in H^\mu(I)$, then there exist a positive constant c and a positive integer N_0 , such that for all $k \geq N_0$ and for all $m \in \mathbb{N}_0$,*

$$\|u_{k,m,N} - u\| \leq c 2^{-\mu(k+m)} \|u\|_\mu, \tag{4.12}$$

where $N = k + m$.

Proof We prove this theorem by induction on m . The proof for the case $m = 0$ is trivial. We assume that (4.12) holds for the case of $m - 1$ and consider the case of m .

Let $\hat{u}_{2^{k+m}}$ be the solution of Eq. (2.8) with $n = 2^{k+m}$. By the triangle inequality, we have that

$$\|u_{k,m,N} - u\| \leq \|\hat{u}_{2^{k+m}} - u\| + \|u_{k,m,N} - \hat{u}_{2^{k+m}}\|.$$

The first term in the right hand side of the above inequality can be estimated directly by Theorem 2.3, that is, there holds

$$\|u - \hat{u}_{2^{k+m}}\| \leq c_0 2^{-\mu(k+m)} \|u\|_\mu, \tag{4.13}$$

for some positive constant c_0 . Hence, it remains to estimate the second term. For this purpose, we first present the difference $u_{k,m,N} - \hat{u}_{2^{k+m}}$ according to Eqs. (2.8) and (4.7). Letting

$$\mathcal{T} := ((\mathcal{A}_{k,m} + \tilde{\mathcal{B}}_{k,m,N})\mathcal{P}_{2^{k+m}} \Psi)'(\hat{u}_{2^{k+m}}),$$

we have that

$$(\mathcal{I} - \mathcal{P}_{2^{k+m}}(\tilde{\mathcal{K}}_{k,m,N} - \mathcal{T}))(u_{k,m,N} - \hat{u}_{2^{k+m}}) = u_{\mathcal{K}} + u_{\mathcal{B}} + \mathcal{P}_{2^{k+m}} \mathcal{R}(u_{k,m,N}, \hat{u}_{2^{k+m}})$$

where

$$u_{\mathcal{K}} := \mathcal{P}_{2^{k+m}}(\tilde{\mathcal{K}}_{k,m,N} - \mathcal{K}_{2^{k+m}})\hat{u}_{2^{k+m}} + \tilde{\mathcal{K}}_{k,m,N}^H(u_{k,m-1,N} - u_{k,m,N}),$$

$$\begin{aligned} u_{\mathcal{B}} &:= \mathcal{P}_{2^{k+m}}(\mathcal{B}_{2^{k+m}} - \tilde{\mathcal{B}}_{k,m,N})\mathcal{P}_{2^{k+m}} \Psi(\hat{u}_{2^{k+m}}) \\ &+ \left(\mathcal{A}_{2^{k+m}}^H + \tilde{\mathcal{B}}_{k,m,N}^H\right)\mathcal{P}_{2^{k+m}}(\Psi(u_{k,m,N}) - \Psi(u_{k,m-1,N})) \end{aligned}$$

and

$$\begin{aligned} \mathcal{R}(u_{k,m,N}, \hat{u}_{2^{k+m}}) &:= (\mathcal{A}_{k,m} + \tilde{\mathcal{B}}_{k,m,N})\mathcal{P}_{2^{k+m}}(\Psi(\hat{u}_{2^{k+m}}) - \Psi(u_{k,m,N})) \\ &- \mathcal{T}(\hat{u}_{2^{k+m}} - u_{k,m,N}). \end{aligned}$$

Note that there exist positive constants c_1 and c_2 such that for sufficiently large integer k ,

$$\|(I - \mathcal{P}_{2^{k+m}}(\tilde{\mathcal{K}}_{k,m,N} - \mathcal{T}))^{-1}\| \leq c_1$$

and

$$\|\mathcal{R}(u_{k,m,N}, \hat{u}_{2^{k+m}})\| \leq c_2 \|u_{k,m,N} - \hat{u}_{2^{k+m}}\|^2.$$

By the last inequality and the fact that $\|u_{k,m,N} - \hat{u}_{2^{k+m}}\| \rightarrow 0$ uniformly for all m , as $k \rightarrow \infty$, we conclude that for sufficiently large integer k and for all m there holds

$$\|\mathcal{R}(u_{k,m,N}, \hat{u}_{2^{k+m}})\| \leq \frac{1}{4c_1} \|u_{k,m,N} - \hat{u}_{2^{k+m}}\|. \tag{4.14}$$

According to Lemmas 4.2 and 4.3, there exists a constant c_3 such that

$$\|u_{\mathcal{K}}\| \leq c_3 2^{-\mu(k+m)} \|\hat{u}_{2^{k+m}}\| + \|\tilde{\mathcal{K}}_{k,m,N}^H\| (\|u_{k,m-1,N} - \hat{u}_{2^{k+m}}\| + \|\hat{u}_{2^{k+m}} - u_{k,m,N}\|),$$

and

$$\|u_{\mathcal{B}}\| \leq c_3 2^{-\mu(k+m)} \|\hat{u}_{2^{k+m}}\| + \|\tilde{\mathcal{B}}_{k,m,N}^H\| L (\|u_{k,m-1,N} - \hat{u}_{2^{k+m}}\| + \|\hat{u}_{2^{k+m}} - u_{k,m,N}\|),$$

where L is the Lipschitz constant. Let

$$\bar{\alpha}_{k,m} := \|\tilde{\mathcal{K}}_{k,m,N}^H\| + L \|\tilde{\mathcal{B}}_{k,m,N}^H\|.$$

Since $\|\hat{u}_{2^{k+m}} - u\| \rightarrow 0$ and $\bar{\alpha}_{k,m} \rightarrow 0$ uniformly for all m , as $k \rightarrow \infty$, for sufficiently large integer k and for all m , there exists a constant c_4 such that

$$\|u_{\mathcal{K}}\| + \|u_{\mathcal{B}}\| \leq c_4 2^{-\mu(k+m)} \|u\|_{\mu} + \frac{1}{4c_1} (\|u_{k,m-1,N} - \hat{u}_{2^{k+m}}\| + \|\hat{u}_{2^{k+m}} - u_{k,m,N}\|). \tag{4.15}$$

Combining the inequalities (4.14) and (4.15) with the induction hypothesis, we get that

$$\begin{aligned} \|u_{k,m,N} - \hat{u}_{2^{k+m}}\| &\leq c' (2^{-\mu(k+m)} \|u\|_{\mu} + \|u_{k,m-1,N} - \hat{u}_{2^{k+m}}\|) \\ &\leq c' (2^{-\mu(k+m)} \|u\|_{\mu} + \|u_{k,m-1,N} - u\| + \|u - \hat{u}_{2^{k+m}}\|) \\ &\leq c'' 2^{-\mu(k+m)} \|u\|_{\mu} \end{aligned} \tag{4.16}$$

for some positive constants c' and c'' . By (4.13) and (4.16), we prove (4.12) for m . □

4.3 Numerical examples

In this subsection, we present two numerical examples to demonstrate the approximate accuracy and computational complexity of the proposed method.

We consider solving the same nonlinear boundary integral equation as in Example 3.2 by Algorithm 2. The difference is that we choose g_0 such that the given function u with a specific regularity is the exact solution. To demonstrate the numerical results, we define the relative error ‘‘Err’’ and the approximation order ‘‘AO’’ respectively by

$$\text{Err} := \frac{\|u - u_{k,m,k+m}\|}{\|u\|}, \quad \text{and} \quad \text{AO} := \log_2 \frac{\|u - u_{k,m,k+m}\|}{\|u - u_{k,m+1,k+m+1}\|},$$

where $u_{k,m,k+m}$ is the approximate solution obtained by Algorithm 2. We also denote by ‘‘CT’’ the computing time measured in seconds, which is used for generating the sparse matrices $\tilde{\mathbf{K}}_{n,N}$, $\tilde{\mathbf{B}}_{n,N}$ and solving the nonlinear system (4.4) by Algorithm 2. To confirm the quasi-linear computational complexity, we define

$$\text{CO} := \log_2 \frac{\text{CT}(2^{k+m+1})}{\text{CT}(2^{k+m})}.$$

Table 2 Numerical results of Algorithm 2 for Example 4.5

m	d_{24+m}	Err	AO	CT	CO
1	63	4.86e-2	0.4948	0.062	
2	127	3.44e-2	0.4971	0.174	1.4887
3	255	2.44e-2	0.4984	0.485	1.4789
4	511	1.72e-2	0.4997	1.356	1.4833
5	1023	1.21e-2	0.5040	3.688	1.4435

Table 3 Numerical results of Algorithm 2 for Example 4.6

m	d_{24+m}	Err	AO	CT	CO
1	63	1.11e-3	1.5189	0.062	
2	127	3.83e-4	1.5101	0.175	1.4970
3	255	1.35e-4	1.5000	0.489	1.4825
4	511	4.79e-5	1.4989	1.330	1.4435
5	1023	1.69e-5	1.5002	3.598	1.4358

Example 4.5 We consider solving the nonlinear boundary integral equation by Algorithm 2. Here, the right-hand side function g_0 is chosen such that

$$u(t) = \begin{cases} t, & 0 \leq t < \pi, \\ 2\pi - t, & \pi \leq t < 2\pi, \end{cases}$$

is the exact solution of the equation. We note that $u \in H^{0.5-\epsilon}(I)$ with $\epsilon > 0$ being an arbitrary number.

In this experiment, we solve (4.4) by Algorithm 2 with the initial level $k = 4$. We report the numerical results in Table 2. The third and fourth columns in Table 2 list, respectively, the relative errors and approximation order of the approximate solution of (4.4) obtained by Algorithm 2. In the fifth and sixth columns, we present respectively the computing time for implementing Algorithm 2 and computed order of computational complexity. We observe from these results that the proposed method gives the optimal convergence order $0.5 - \epsilon$ and the computing time is consistent with the quasi-linear computational complexity estimate.

Example 4.6 In this example, we choose the right-hand side function g_0 such that

$$u(t) = \frac{1}{12}(3t^2 - 6\pi t + 2\pi^2), \quad t \in I,$$

is the exact solution of the equation. Here, we have $u \in H^{1.5-\epsilon}(I)$ with $\epsilon > 0$ being an arbitrary number.

In Table 3, we list the numerical results for this example when we solve (4.4) by Algorithm 2 with the initial level $k = 4$. Note that, since the exact solution $u \in H^{1.5-\epsilon}$, the theoretical convergence order is $1.5 - \epsilon$, which is confirmed by the computed order of convergence. At the same time, the computing time is also consistent with the quasi-linear computational complexity estimate.

Acknowledgments This research is partially supported by Guangdong Provincial Government of China through the ‘‘Computational Science Innovative Research Team’’ program, by Guangdong Province Key Lab of Computational Science, by the Natural Science Foundation of China under Grants 11071286, 91130009,

11071250, 11126149 and 11271370. The third author is also supported in part by US Air Force Office of Scientific Research under Grant FA9550-09-1-0511 and by the US National Science Foundation under Grant DMS-1115523.

References

1. Atkinson, K.E.: *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge University Press, Cambridge (1997)
2. Atkinson, K.E., Chandler, G.: Boundary integral equation methods for solving Laplace's equation with nonlinear boundary conditions: the smooth boundary case. *Math. Comput.* **55**, 451–472 (1990)
3. Atkinson, K.E., Potra, F.A.: Projection and iterated projection methods for nonlinear integral equations. *SIAM J. Numer. Anal.* **24**, 1352–1373 (1987)
4. Babolian, E., Mordad, M.: A numerical method for solving systems of linear and nonlinear integral equations of the second kind by hat basis functions. *Comput. Math. Appl.* **62**, 187–198 (2011)
5. Cai, H., Xu, Y.: A fast Fourier-Galerkin method for solving singular boundary integral equations. *SIAM J. Numer. Anal.* **46**, 1965–1984 (2008)
6. Chen, M., Chen, Z., Chen, G.: *Approximate Solutions of Operator Equations*. World Scientific Publishing Co., River Edge (1997)
7. Chen, X., Chen, Z., Wu, B., Xu, Y.: Fast multilevel augmentation methods for nonlinear boundary integral equation. *SIAM J. Numer. Anal.* **49**, 2231–2255 (2011)
8. X. Chen, Z. Chen, B. Wu and Y. Xu: Multilevel augmentation methods for nonlinear boundary integral equations II: accelerated quadratures and Newton iterations. *J. Integral Equ. Appl.* (in press).
9. Chen, Z., Micchelli, C.A., Xu, Y.: A construction of interpolation wavelets on invariant sets. *Math. Comput.* **68**, 1569–1587 (1999)
10. Chen, Z., Wu, B., Xu, Y.: Fast multilevel augmentation methods for solving Hammerstein equations. *SIAM J. Numer. Anal.* **47**, 2321–2346 (2009)
11. Cheng, P., Huang, J.: Extrapolation algorithms for solving nonlinear boundary integral equations by mechanical quadrature methods. *Numer. Algorithm* **58**, 545–554 (2011)
12. Han, G.: Extrapolation of a discrete collocation-type method of Hammerstein equations. *J. Comput. Appl. Math.* **61**, 73–86 (1995)
13. Han, G., Wang, J.: Extrapolation of Nyström solution for two-dimensional nonlinear Fredholm integral equations. *J. Sci. Comput.* **14**, 197–209 (1999)
14. Jiang, Y., Xu, Y.: Fast discrete algorithms for sparse Fourier expansions of high dimensional functions. *J. Complex.* **26**, 51–81 (2010)
15. Jiang, Y., Xu, Y.: Fast Fourier-Galerkin methods for solving singular boundary integral equations: numerical integration and precondition. *J. Comput. Appl. Math.* **234**, 2792–2807 (2010)
16. Kaneko, H., Noren, R., Novaprateep, B.: Wavelet applications to the Petrov-Galerkin method for Hammerstein equations. *Appl. Numer. Math.* **45**, 255–273 (2003)
17. Kaneko, H., Noren, R., Xu, Y.: Numerical solutions for weakly singular Hammerstein equations and their superconvergence. *J. Integral Equ. Appl.* **4**, 391–407 (1992)
18. Kress, R.: *Linear Integral Equations*. Springer, New York (1989)
19. Kaneko, H., Xu, Y.: Degenerate kernel method for Hammerstein equations. *Math. Comput.* **62**, 141–148 (1994)
20. Kumar, S.: A discrete collocation-type method for Hammerstein equation. *SIAM J. Numer. Anal.* **25**, 328–341 (1988)
21. Kumar, S., Sloan, I.H.: A new collocation-type method for Hammerstein integral equations. *Math. Comput.* **48**, 585–593 (1987)
22. Lardy, L.J.: A variation of Nyström's method for Hammerstein equations. *J. Integral Equ.* **3**, 43–60 (1981)
23. Runst, T., Sickel, W.: *Sobolev Spaces of Fractional Order. Nemytskij Operators and Nonlinear Partial Differential Equations*. de Gruyter, Berlin (1996)
24. Ruotsalainen, K., Wendland, W.: On the boundary element method for some nonlinear boundary value problems. *Numer. Math.* **53**, 299–314 (1988)
25. Saranen, J.: Projection methods for a class of Hammerstein equations. *SIAM J. Numer. Anal.* **27**, 1445–1449 (1990)
26. Shahsavaran, A.: Computational method to solve nonlinear integral equations using block pulse functions by collocation method. *Appl. Math. Sci.* **5**, 3211–3220 (2011)
27. Vainikko, G.M.: Perturbed Galerkin method and general theory of approximate methods for nonlinear equations. *Zh. Vychisl. Mat. Fiz.* **7**, 723–751 (1967)

28. Vainikko, G.M.: Galerkin's perturbation method and the general theory of approximate methods (English translation). *USSR Comput. Math. Math. Phys.* **7**, 1–41 (1967)
29. Vainikko, G.M., Uba, P.: A piecewise polynomial approximation to the solution of an integral equation with weakly singular kernel. *J. Aust. Math. Soc. Ser. B* **22**, 431–438 (1981)
30. Yan, Y., Sloan, I.H.: On integral equations of the first kind with logarithmic kernels. *J. Integral Equ. Appl.* **1**, 549–579 (1988)