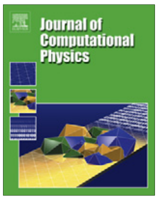
Journal of Computational Physics 230 (2011) 7488–7501

Contents lists available at ScienceDirect

Journal of Computational Physics

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



Second kind integral equations for the first kind Dirichlet problem of the biharmonic equation in three dimensions

Shidong Jiang [a], Bo Ren [a], Paul Tsuji [b], Lexing Ying [c,*]

[a] Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ 07102, United States
[b] ICES, University of Texas at Austin, Austin, TX 78712, United States
[c] Department of Mathematics and ICES, University of Texas at Austin, Austin, TX 78712, United States

ARTICLE INFO

Article history:
Received 17 June 2010
Received in revised form 29 March 2011
Accepted 14 June 2011
Available online 28 June 2011

Keywords:
Second kind integral equation
Dirichlet problem
Biharmonic equation

ABSTRACT

A Fredholm second kind integral equation (SKIE) formulation is constructed for the Dirichlet problem of the biharmonic equation in three dimensions. A fast numerical algorithm is developed based on the constructed SKIE. Its performance is illustrated via several numerical examples.

1. Introduction

The biharmonic equations are an important class of equations in both physics and engineering. In fluid dynamics, the so-called stream function satisfies the biharmonic equation. Many problems in elasticity can also be formulated in terms of the biharmonic equation where the fundamental physical quantities such as displacement, stress, and strain all satisfy the biharmonic equation (see, for example, [30]). There have been extensive research activities on the biharmonic equation both theoretically and computationally (see, for example, [1,9,31,32,35]). Solving the biharmonic equation numerically is a nontrivial task. Since the biharmonic equation is a fourth order differential equation, standard finite difference approximation results in a linear system with condition number proportional to $O(N^4)$, where N is the number of the discretization point in each dimension. Thus even a relatively small N causes a catastrophic loss of precision, rendering the numerical computation meaningless. For three dimensional problems, the computational cost is prohibitive for finite difference schemes since one has to discretize the whole domain and the stencil for the 4th order differential operator requires more than twenty points to achieve a good accuracy (see, for example, [2]).

In this paper, we consider the first kind Dirichlet problem of the biharmonic equation as follows. Given continuous functions f_1, f_2 on a sufficiently smooth surface S in $\mathbb{R}^3$, find a function $u \in C^4(D) \cap C(\overline{D})$ such that

$$\begin{cases} \Delta^2 u = 0 & \text{on } D, \\ u = f_1 & \text{on } S, \\ \frac{\partial u}{\partial n} = f_2 & \text{on } S, \end{cases} \tag{1}$$

where D is the domain enclosed by S. We construct a Fredholm second kind integral equation formulation for this problem. The advantages of the SKIE formulation are obvious. First, it reduces the dimension of the problem by one since we now only

[*] Corresponding author.
E-mail addresses: shidong.jiang@njit.edu (S. Jiang), br39@njit.edu (B. Ren), ptsuji@ices.utexas.edu (P. Tsuji), lexing@math.utexas.edu (L. Ying).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.06.015

need to solve integral equations with unknowns on the boundary. Second, being a boundary integral equation formulation, it can easily handle problems involving complex geometry. Third, for second kind integral equations the condition number of the resulting linear system remains bounded when the number of unknowns increases. Finally, there are fast numerical algorithms for this kind of boundary integral equations such as tree codes and fast multipole algorithms (see [6,17,18]) where the computational cost is $O(N)$ or $O(N\log N)$ for problems with N the total number of discretization points on the boundary. We would like to remark here that the currently existing fast multipole algorithms (see for example, [6,13,14,17,18,27,36–38]) either are not directly applicable to the kernels in our SKIE formulation or involve a large prefactor. Hence, we have developed a fast matrix–vector multiplication scheme based on randomized factorization for low rank matrices. The scheme is easy to implement and applicable to a broad of kernels.

There has been a great amount of work in developing numerical methods for biharmonic equations in two dimensions. In [15], the two dimensional problem is solved by a decomposition into two Poisson problems with an iteration for the trace of the right hand side. Since one needs to solve second order elliptic PDEs in the whole domain, the approach is not as efficient as the boundary integral equation method complexity-wise. The two dimensional problem has been reduced to second kind integral equations by many researchers (see for example [1,7,8,22–24,26,28,29,34]). There are also several attempts to reduce the three dimensional problem to integral equations. For example in [35], the 3D problem is reduced to an integral operator that is a sum of an invertible operator and a compact one. However, it assumes the knowledge of the Laplace Green's function of a general compact domain, thus making it less convenient for the purpose of numerical computation.

The paper is organized as follows. First, we summarize our main result in Section 2. We then present the detailed derivation of the SKIE formulation in Section 3. A numerical algorithm based on the SKIE formulation is followed in Section 4. In Section 5, we illustrate the performance (accuracy, efficiency, and robustness) of the numerical algorithm via several numerical examples. Finally, we conclude our paper with a short discussion on the extension and applications of our method.

2. The main result

We use D to denote a bounded domain in $\mathbb{R}^3$. Its boundary, denoted by S, is assumed to be a sufficiently smooth regular surface. We use x and y to denote points in $\mathbb{R}^3$. The outward unit normal vector at a point x on S is denoted by n_x. However, when the point is y, we will simply denote it by n. We use $\boldsymbol{r}$ to denote the vector $x-y$ and r to denote its length $|x-y|$. We use $\boldsymbol{r}\cdot\boldsymbol{n}$ to denote the dot product between $\boldsymbol{r}$ and $\boldsymbol{n}$ (note that n is boldfaced in the dot product). Finally, with a slight abuse of notation, we always use the same symbol to denote both the kernel of an integral operator and the operator itself.

We now summarize our main result in the following theorem.

Theorem 1. *Let u be defined by the formula*

$$u(x)=\int_S [K_1(x,y)\sigma_1(y)+K_2(x,y)\sigma_2(y)]ds_y, \tag{2}$$

where K_1 and K_2 are defined by the formulas

$$K_1(x,y)=-2G_{nnn}(x,y)+3(\Delta G)_n(x,y)=-\frac{3(\boldsymbol{r}\cdot\boldsymbol{n})^3}{4\pi r^5}, \tag{3}$$

$$K_2(x,y)=-\Delta G(x,y)=\frac{1}{4\pi r} \tag{4}$$

with $G(x,y)=-\frac{1}{8\pi}r$ the fundamental solution of the biharmonic equation in $\mathbb{R}^3$. Then u is a solution of the first (interior) Dirichlet problem of the biharmonic Eq. (1) *if the density σ satisfies the following system of second kind integral equations:*

$$D(x)\sigma(x)+\int_S K(x,y)\sigma(y)ds_y=f(x), \tag{5}$$

where

$$D(x)=\begin{pmatrix}\frac{1}{2} & 0\\ -H(x) & \frac{1}{2}\end{pmatrix},\quad \sigma(x)=\begin{pmatrix}\sigma_1(x)\\ \sigma_2(x)\end{pmatrix},\quad f(x)=\begin{pmatrix}f_1(x)\\ f_2(x)\end{pmatrix}, \tag{6}$$

$$K(x,y)=\begin{pmatrix}K_{11}(x,y) & K_{12}(x,y)\\ K_{21}(x,y) & K_{22}(x,y)\end{pmatrix}=\begin{pmatrix}-\frac{3(\boldsymbol{r}\cdot\boldsymbol{n})^3}{4\pi r^5} & \frac{1}{4\pi r}\\ -\frac{9(\boldsymbol{r}\cdot\boldsymbol{n})^2(\boldsymbol{n}\cdot\boldsymbol{n}_x)}{4\pi r^5}+\frac{15(\boldsymbol{r}\cdot\boldsymbol{n})^3(\boldsymbol{r}\cdot\boldsymbol{n}_x)}{4\pi r^7} & -\frac{\boldsymbol{r}\cdot\boldsymbol{n}_x}{4\pi r^3}\end{pmatrix} \tag{7}$$

and H(x) is the sum of principal curvatures of S at x.

Remark 2. For exterior problems, the diagonal part D will change sign.

Remark 3. The discretized matrix D is block diagonal but non-symmetric. For certain problems, it may be advantageous that we precondition the whole system by $D^{-1}=\begin{pmatrix}2I & 0\\ 4H & 2I\end{pmatrix}$.

3. Construction of the SKIE formulation

3.1. A heuristic reasoning

To reduce the first Dirichlet problem (1) to SKIEs, our strategy follows the line along Peter Farkas' thesis [12] (we would like to remark here that [12] deals only with the two dimensional case). We try to represent the solution via a sum of two multiple layer potentials

$$u(x)=\int_S [K_1(x,y)\sigma_1(y)+K_2(x,y)\sigma_2(y)]ds_y, \tag{8}$$

where K_i and $\sigma_i(i=1,2)$ are integral kernels (to be determined) and unknown densities, respectively. Since u has to satisfy the biharmonic equation for $x\in D$, the kernels K_i $(i=1,2)$ have to be a linear combination of G and its partial derivatives. u also needs to satisfy two boundary conditions. It is well known that layer potentials may experience certain jumps across the boundary. Thus, we denote

$$\begin{aligned} K_{11}(x,y)&=K_1(x,y),\\ K_{12}(x,y)&=K_2(x,y),\\ K_{21}(x,y)&=\tfrac{\partial K_1(x,y)}{\partial n_x},\\ K_{22}(x,y)&=\tfrac{\partial K_2(x,y)}{\partial n_x}\end{aligned} \tag{9}$$

and assume that the jump relation for each associated layer potential is as follows:

$$\lim_{\varepsilon\to 0+}\int_S K_{ij}(x-\varepsilon n_x,y)\sigma_j(y)ds_y = D_{ij}\sigma_j(x)+\int_S K_{ij}(x,y)\sigma_j(y)ds_y,\quad x\in S,\quad i,j=1,2, \tag{10}$$

where D_{ij} are to be determined. With this assumption, the boundary conditions lead to the following system of integral equations in $\sigma_i(i=1,2)$

$$\begin{pmatrix} D_{11} & D_{12}\\ D_{21} & D_{22}\end{pmatrix}\begin{pmatrix}\sigma_1(x)\\ \sigma_2(x)\end{pmatrix}+\int_S\begin{pmatrix}K_{11}(x,y) & K_{12}(x,y)\\ K_{21}(x,y) & K_{22}(x,y)\end{pmatrix}\begin{pmatrix}\sigma_1(y)\\ \sigma_2(y)\end{pmatrix}ds_y=\begin{pmatrix}f_1(x)\\ f_2(x)\end{pmatrix}. \tag{11}$$

To make the above system second kind, we must require that the diagonal matrix $D=\begin{pmatrix} D_{11} & D_{12}\\ D_{21} & D_{22}\end{pmatrix}$ has nonzero determinant and the integral operators K_{ij} are all compact.

3.2. Choices of K_1 and K_2

To simplify the discussion and fix the notation, we require further that $D_{12}=0$. Note that this can always be achieved without really changing the system by exchanging K_1 and K_2 and/or forming proper linear combinations of them. Now it is easy to choose K_2 such that $D_{12}=0$, $D_{22}\neq 0$ and K_{12}, K_{22} are both compact. Indeed, we may simply choose

$$K_2(x,y)=-\Delta G(x,y)=\frac{1}{4\pi r}. \tag{12}$$

That is, K_2 is the single layer potential operator for the Laplace equation. It is well known (see, for example, [21]) that the single layer potential of the Laplace equation is continuous across the boundary (i.e., $D_{12}=0$); its normal derivative has the following jump relation:

$$\lim_{\varepsilon\to 0+}\int_S\frac{\partial}{\partial n_x}\frac{1}{4\pi|x\pm\varepsilon n_x-y|}\sigma(y)ds_y=\int_S\frac{\partial}{\partial n_x}\frac{1}{4\pi|x-y|}\sigma(y)ds_y\mp\frac{1}{2}\sigma(x),\quad x\in S, \tag{13}$$

i.e., $D_{22}=\frac{1}{2}$; $\frac{1}{r}$ is only weakly singular; and $\frac{\partial}{\partial n_x}\frac{1}{r}$ is in fact continuous for sufficiently smooth boundary. Thus both K_{12}, K_{22} are compact.

Remark 4. ΔG is not the only choice for K_2. In fact, G_{nn} is another perfectly reasonable choice which satisfies all the requirements on K_2. A straightforward calculation shows that G_{nn} satisfies exactly the same jump relations as ΔG (one may also obtain this result via the standard decomposition of the Laplacian $\Delta=\Delta^S+H\frac{\partial}{\partial n}+\frac{\partial^2}{\partial n^2}$ with Δ^S the surface Laplacian). Indeed, one may set $K_2=-\alpha\Delta G-(1-\alpha)G_{nn}$ for any real number α without changing the diagonal matrix D or the singular properties of K_{12} and K_{22}.

Let us now try to choose K_1. Since $D_{12}=0$ and $\det D\neq 0$, we must have $D_{11}\neq 0$. We also require that K_{11} and K_{21} be both compact. This is a nontrivial task. In fact, it can never be done for the Laplace equation where its double layer potential has a jump (i.e., $D_{11}\neq 0$) and the double layer potential operator itself is compact (i.e., K_{11} is compact), but the normal derivative of the double layer potential is hypersingular (i.e., K_{21} is NOT compact). The situation becomes worse if one tries to use higher order layer potentials for the Laplace equation.

Fortunately, this is not the case for the biharmonic equation. We claim that the following choice

$$K_1(x,y) = -2G_{nnn}(x,y) + 3(\Delta G)_n(x,y) \tag{14}$$

would meet all requirements. The key is that K_{21} is also compact due to a cancellation of the hypersingular parts. Before we continue, let us write down the exact expressions for these kernels. We first note that by normal derivatives we really mean that $\frac{\partial f}{\partial n} = \nabla f \cdot \boldsymbol{n}$. Straightforward calculation then shows that

$$\begin{aligned} &\tfrac{\partial r^2}{\partial n} = -2\boldsymbol{r}\cdot\boldsymbol{n}, \quad \tfrac{\partial r^2}{\partial n_x} = 2\boldsymbol{r}\cdot\boldsymbol{n}_x, \\ &\tfrac{\partial \boldsymbol{r}\cdot\boldsymbol{n}}{\partial n} = -1, \quad \tfrac{\partial \boldsymbol{r}\cdot\boldsymbol{n}}{\partial n_x} = \boldsymbol{n}\cdot\boldsymbol{n}_x. \end{aligned} \tag{15}$$

We readily have

$$\begin{aligned} G_n(x,y) &= \frac{\boldsymbol{r}\cdot\boldsymbol{n}}{8\pi r}, \quad G_{nn}(x,y) = -\frac{1}{8\pi r} + \frac{(\boldsymbol{r}\cdot\boldsymbol{n})^2}{8\pi r^3}, \\ G_{nnn}(x,y) &= -\frac{3(\boldsymbol{r}\cdot\boldsymbol{n})}{8\pi r^3} + \frac{3(\boldsymbol{r}\cdot\boldsymbol{n})^3}{8\pi r^5}, \\ G_{nnnn_x}(x,y) &= -\frac{9(\boldsymbol{r}\cdot\boldsymbol{n})(\boldsymbol{r}\cdot\boldsymbol{n}_x)}{8\pi r^5} - \frac{3(\boldsymbol{n}\cdot\boldsymbol{n}_x)}{8\pi r^3} \\ &\quad + \frac{9(\boldsymbol{r}\cdot\boldsymbol{n})^2(\boldsymbol{n}\cdot\boldsymbol{n}_x)}{8\pi r^5} - \frac{15(\boldsymbol{r}\cdot\boldsymbol{n})^3(\boldsymbol{r}\cdot\boldsymbol{n}_x)}{8\pi r^7} \end{aligned} \tag{16}$$

and

$$\begin{aligned} \Delta G(x,y) &= -\frac{1}{4\pi r}, \quad (\Delta G)_n(x,y) = -\frac{\boldsymbol{r}\cdot\boldsymbol{n}}{4\pi r^3}, \\ (\Delta G)_{nn_x}(x,y) &= -\frac{3(\boldsymbol{r}\cdot\boldsymbol{n})(\boldsymbol{r}\cdot\boldsymbol{n}_x)}{4\pi r^5} - \frac{(\boldsymbol{n}\cdot\boldsymbol{n}_x)}{4\pi r^3}. \end{aligned} \tag{17}$$

Thus

$$\begin{aligned} K_1(x,y) &= -\frac{3(\boldsymbol{r}\cdot\boldsymbol{n})^3}{4\pi r^5}, \\ K_{11}(x,y) &= K_1(x,y) = -\frac{3(\boldsymbol{r}\cdot\boldsymbol{n})^3}{4\pi r^5}, \\ K_{21}(x,y) &= \frac{\partial K_1(x,y)}{\partial n_x} = -\frac{9(\boldsymbol{r}\cdot\boldsymbol{n})^2(\boldsymbol{n}\cdot\boldsymbol{n}_x)}{4\pi r^5} + \frac{15(\boldsymbol{r}\cdot\boldsymbol{n})^3(\boldsymbol{r}\cdot\boldsymbol{n}_x)}{4\pi r^7} \end{aligned} \tag{18}$$

and

$$\begin{aligned} K_2(x,y) &= \frac{1}{4\pi r}, \\ K_{12}(x,y) &= K_2(x,y) = \frac{1}{4\pi r}, \\ K_{22}(x,y) &= \frac{\partial K_2(x,y)}{\partial n_x} = -\frac{\boldsymbol{r}\cdot\boldsymbol{n}_x}{4\pi r^3}. \end{aligned} \tag{19}$$

We have the following lemma.

Lemma 5. *All functions $K_{ij}(i,j = 1,2)$ are at most weakly singular. Hence all four associated integral operators are of order -1 and thus compact as maps from $H^s(S)$ to itself for any real number s.*

Proof. This simply follows from the well-known fact in potential theory (see, for example, [21]) that

$$|\boldsymbol{r}\cdot\boldsymbol{n}| \leqslant Mr^2, \quad |\boldsymbol{r}\cdot\boldsymbol{n}_x| \leqslant Mr^2, \quad x,y \in S \tag{20}$$

and thus

$$|K_{ij}(x,y)| \leqslant \frac{M}{r}, \quad i,j = 1,2 \quad x,y \in S. \qquad \square \tag{21}$$

3.3. Derivation of the jump relations and diagonal terms

We now derive the jump relations of layer potentials associated with K_{11} and K_{21}. We assume that the density is at least twice continuously differentiable. The conventional method for deriving such jump relations relies on the global integral formulas like Gauss' formula (see, for example, [21]). Here we will apply a local analysis to obtain the jump relations. The basic idea is that we will study the asymptotic expansion of the given kernel as one point, say x, approaches the boundary. The asymptotic expansion will single out the hypersingular terms, singular terms, terms that are approximation to the identity, and weakly singular terms. Terms that are approximation to the identity, or more intuitively approximation to the delta function, will induce jumps across the boundary. We think this local analysis approach is more natural since the jump relation is actually a local property. We remark here that similar techniques have already been used in [20].

For a fixed point $x \in S$, by translation and rotation of the coordinate system, we can assume that $x = (0,0,0)$ and $n_x = (0,0,-1)$. It is well known that locally S is the graph of a smooth function defined on the tangent space T_xS. Moreover, at a small neighborhood of x, the points on S have the following parameterization:

$$y = (u, \phi(u)) = \left(u_1, u_2, \frac{1}{2}\sum_{i=1}^{n} k_i u_i^2 + O\left(|u|^3\right)\right), \tag{22}$$

where k_1, k_2 are the principle curvatures of S at $x = 0$. We now consider the asymptotic expansions of $K_{11}(x + \varepsilon n_x, y)$ and $K_{21}(x + \varepsilon n_x, y)$ as $\varepsilon \to 0$ and y is close to $x = 0$. We denote $\tilde{\boldsymbol{r}} = x + \varepsilon n_x - y$, $\tilde{r} = |\tilde{\boldsymbol{r}}|$, and $d = \sqrt{u_1^2 + u_2^2 + \varepsilon^2}$. We have

$$\begin{aligned}
\tilde{r}^2 &= |x + \varepsilon n_x - y|^2 = |u|^2 + (\phi + \varepsilon)^2 \\
&= d^2\left[1 + \frac{\varepsilon\sum_{i=1}^2 k_i u_i^2}{d^2} + O\left(d^2\right)\right], \\
\frac{1}{\tilde{r}^j} &= \frac{1}{d^j}\left[1 - \frac{j}{2}\frac{\varepsilon\sum\limits_{i=1}^{2} k_i u_i^2}{d^2} + O\left(d^2\right)\right], \\
n = n_y &= \frac{(\nabla\phi, -1)}{\sqrt{1 + |\nabla\phi|^2}} = (k_1u_1, k_2u_2, -1)(1 + O(d^2)), \\
(\tilde{\boldsymbol{r}} \cdot \boldsymbol{n})^j &= \varepsilon^j - \frac{j}{2}\varepsilon^{j-1}\sum_{i=1}^{2} k_i u_i^2 + O(d^{j+2}), \\
\tilde{\boldsymbol{r}} \cdot \boldsymbol{n}_x &= \varepsilon + \frac{1}{2}\sum_{i=1}^{2} k_i u_i^2 + O(d^3), \\
\boldsymbol{n} \cdot \boldsymbol{n}_x &= 1 + O(d^2).
\end{aligned} \tag{23}$$

Using these formulas, we obtain

$$K_{11}(x + \varepsilon n_x, y) = \frac{3\varepsilon^3}{4\pi d^5} + O\left(\frac{1}{d}\right), \tag{24}$$

and

$$K_{21}(x + \varepsilon n_x, y) = -\frac{9\varepsilon^2}{4\pi d^5} + \frac{15\varepsilon^4}{4\pi d^7} + \frac{9\varepsilon\sum_{i=1}^2 k_i u_i^2}{4\pi d^5} + \frac{15\varepsilon^3\sum_{i=1}^2 k_i u_i^2}{8\pi d^7} - \frac{105\varepsilon^5\sum_{i=1}^2 k_i u_i^2}{8\pi d^9} + O\left(\frac{1}{d}\right). \tag{25}$$

A simple calculation using polar coordinates shows that

$$\begin{aligned}
&\int_{\mathbb{R}^2} \frac{1}{\left(u_1^2+u_2^2+1\right)^{5/2}} du_1 du_2 = \frac{2\pi}{3}, \\
&\int_{\mathbb{R}^2} \frac{1}{\left(u_1^2+u_2^2+1\right)^{7/2}} du_1 du_2 = \frac{2\pi}{5}, \\
&\int_{\mathbb{R}^2} \frac{u_i^2}{\left(u_1^2+u_2^2+1\right)^{5/2}} du_1 du_2 = \frac{2\pi}{3}, \quad i = 1,2, \\
&\int_{\mathbb{R}^2} \frac{u_i^2}{\left(u_1^2+u_2^2+1\right)^{7/2}} du_1 du_2 = \frac{2\pi}{15}, \quad i = 1,2, \\
&\int_{\mathbb{R}^2} \frac{u_i^2}{\left(u_1^2+u_2^2+1\right)^{9/2}} du_1 du_2 = \frac{2\pi}{35}, \quad i = 1,2.
\end{aligned} \tag{26}$$

We observe now that the kernels $\frac{3\varepsilon^3}{2\pi d^5}$, $\frac{3\varepsilon u_i^2}{2\pi d^5}$, $\frac{15\varepsilon^3 u_i^2}{2\pi d^7}$, and $\frac{35\varepsilon^5 u_i^2}{2\pi d^9}$ on the right sides of (24) and (25) are all functions of the form $\frac{1}{\varepsilon^2}\psi\left(\frac{u}{\varepsilon}\right)$ with ψ a positive integrable function which integrates to 1 in the parameter plane. By Theorem 1.25 on page 13 of [33], the corresponding integral operators are all approximations to the identity operator (see p.13 in [33]). Thus, we have

$$
\begin{aligned}
&\lim_{\varepsilon\to 0+} \int_{S\cap B_\delta(x)} \tfrac{3\varepsilon^3}{4\pi d^5}\sigma(y)ds_y = \tfrac{1}{2}\sigma(x),\\
&\lim_{\varepsilon\to 0+} \int_{S\cap B_\delta(x)} \tfrac{9\varepsilon u_i^2}{4\pi d^5}\sigma(y)ds_y = \tfrac{3}{2}\sigma(x),\\
&\lim_{\varepsilon\to 0+} \int_{S\cap B_\delta(x)} \tfrac{15\varepsilon^3 u_i^2}{8\pi d^7}\sigma(y)ds_y = \tfrac{1}{4}\sigma(x),\\
&\lim_{\varepsilon\to 0+} \int_{S\cap B_\delta(x)} -\tfrac{105\varepsilon^5 u_i^2}{8\pi d^9}\sigma(y)ds_y = -\tfrac{3}{4}\sigma(x),
\end{aligned} \tag{27}
$$

where $B_\delta(x)$ is a ball centered around x of sufficiently small radius δ.

Second, the first two terms on the right side of (25) are hypersingular. Using (26) again we see that

$$\lim_{\varepsilon\to 0} \int_{S\cap B_\delta(x)} \left(-\frac{9\varepsilon^2}{4\pi d^5} + \frac{15\varepsilon^4}{4\pi d^7}\right)\sigma_0\, du = 0, \tag{28}$$

that is, the two terms cancel out for a constant density when taking the limit of $\varepsilon \to 0$. Now for a twice continuously differentiable density, we may use Taylor's theorem to expand the density (multiplying with the Jacobian) as the sum of a constant term, a linear term, and a second order term. The constant term vanishes in the limit process due to cancellation. The linear term vanishes due to symmetry. And the second order term vanishes in the limit process by the dominated convergence theorem since the whole integrand is only weakly singular and thus integrable. Hence, the first two hypersingular terms exactly cancel out in the limit process, and we have

$$\lim_{\varepsilon\to 0} \int_{S\cap B_\delta(x)} \left(-\frac{9\varepsilon^2}{4\pi d^5} + \frac{15\varepsilon^4}{4\pi d^7}\right)\sigma(y)ds_y = 0. \tag{29}$$

Finally, combining (24)–(29), we obtain the following jump relations:

$$
\begin{aligned}
&\lim_{\varepsilon\to 0+} \int_S K_{11}(x \pm \varepsilon n_x, y)\sigma(y)ds_y = \int_S K_{11}(x,y)\sigma(y)ds_y \mp \tfrac{1}{2}\sigma(x),\\
&\lim_{\varepsilon\to 0+} \int_S K_{21}(x \pm \varepsilon n_x, y)\sigma(y)ds_y = \int_S K_{21}(x,y)\sigma(y)ds_y \pm H(x)\sigma(x),
\end{aligned} \tag{30}
$$

where $H = k_1 + k_2$ is the mean curvature at $x \in S$.

4. Numerical algorithm

4.1. Discretization

For a given mesh size h, we assume that the surface S is discretized with a curvilinear triangle mesh with patches $\{P_j\}$ with diameters of order $O(h)$, i.e. $S = \bigcup_{j\in J} P_j$ where J serves as the index set of the patches. Each patch P_j is parameterized by a C^2 function $m_j : T \to P_j$ from the standard flat triangle T. Let us define the three vertices of T by t^0, t^1 and t^2. We denote the vertices of the mesh by $\{x_i\}_{i\in I}$ where I is the index set of vertices and write $x_i \in P_j$ when x_i is a vertex of the patch P_j. The value of h controls the resolution of this mesh of S.

The approximation space for $\sigma_1(y)$ and $\sigma_2(y)$ is the span of the basis functions $\{\phi_i(x)\}_{i\in I}$ that are local, continuous, and piecewise linear. More precisely, the function $\phi_i(x) : S \to \mathbb{R}$ satisfies the following conditions: (1) $\phi_i(x)$ is continuous on S, (2) the support of $\phi_i(x)$ is contained in $\bigcup_{j:x_i\in P_j} P_j$, i.e., the union of the patches that contain x_i, (3) $\phi_i(x_{i'}) = \delta_{ii'}$, and (4) for any j such that $x_i \in P_j$, $\phi_i(m_j(t)) : T \to \mathbb{R}$ is linear in T. It is clear from these requirements that the support of each $\phi_i(x)$ on S has a diameter of order $O(h)$.

A standard collocation method looks for approximations

$$\sigma_1(x) = \sum_{i\in I} \sigma_{1,i}\phi_i(x) \quad \sigma_2(x) = \sum_{i\in I} \sigma_{2,i}\phi_i(x), \tag{31}$$

so that when we plug them into (5) the integral equation is satisfied exactly at the vertices $\{x_i\}_{i\in I}$:

$$\begin{pmatrix} \frac{1}{2} & 0 \\ -H(x_i) & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \sigma_{1,i} \\ \sigma_{2,i} \end{pmatrix} + \int_S \begin{pmatrix} K_{11}(x_i,y) & K_{12}(x_i,y) \\ K_{21}(x_i,y) & K_{22}(x_i,y) \end{pmatrix} \begin{pmatrix} \sum_j \sigma_{1,j}\phi_j(y) \\ \sum_j \sigma_{2,j}\phi_j(y) \end{pmatrix} ds_y = \begin{pmatrix} f_{1,i} \\ f_{2,i} \end{pmatrix} \tag{32}$$

with $f_{1,i} = f_1(x_i)$ and $f_{2,i} = f_2(x_i)$. Since the mesh size is $O(h)$ and piecewise linear basis functions are used in the collocation scheme, we expect formally a quadratic convergence rate $O(h^2)$. The integrals in the collocation Eq. (32) cannot be evaluated explicitly in general and numerical quadrature is required for their calculation. As the kernels $K_{11}(x,y)$, $K_{12}(x,y)$, $K_{21}(x,y)$, and $K_{22}(x,y)$ have similar behavior at the singularity, we treat them in the same way by considering a general form

$$\int_S K(x,y)\sigma(y)ds_y \tag{33}$$

with $K(x,y) = K_{11}(x,y)$, $K_{12}(x,y)$, $K_{21}(x,y)$, or $K_{22}(x,y)$, and $\sigma(y) = \sum_j \sigma_{1,j}\phi_j(y)$ or $\sum_j \sigma_{2,j}\phi_j(y)$. The definition of the surface mesh gives

$$\int_S K(x_i,y)\sigma(y)ds_y = \sum_{j\in J}\int_{P_j} K(x_i,y)\sigma(y)ds_y = \sum_{j\in J}\int_T K(x_i,m_j(t))J_j(t)\sigma(t)dt, \tag{34}$$

where $J_j(t)$ is the Jacobian factor of the map $m_j : T \to P_j$ at t. The last integral can decomposed into two parts:

$$\sum_{j:x_i\notin P_j}\int_T K(x_i,m_j(t))J_j(t)\sigma(t)dt + \sum_{j:x_i\in P_j}\int_T K(x_i,m_j(t))J_j(t)\sigma(t)dt. \tag{35}$$

In the first sum of (35), since $x_i \notin P_j$, the kernel $K(x_i,m_j(t))$ is never singular. These integrals over T are discretized using the standard symmetric Gaussian quadrature weights over the triangle T. Denote $\{t_\alpha\}$ and $\{w_\alpha\}$ to be the quadrature points and weights, respectively. The first non-singular integral then becomes

$$\sum_{j:x_i\notin P_j}\sum_\alpha K(x_i,m_j(t_\alpha))J_j(t_\alpha)\sigma(t_\alpha)w_\alpha. \tag{36}$$

Next, let us consider next the second sum in (35) and fix a j with $x_i \in P_j$. Since $x_i \in P_j$, there exists c_{ij} with value 0, 1, or 2 such that $x_i = m_j(t^{c_{ij}})$. As a result, the integral $\int_T K(m_j(t^{c_{ij}}),\ m_j(t))J_j(t)\sigma(t)dt$ can have a $1/r$ type singularity at $t^{c_{ij}}$. The Duffy quadrature [10] is used to integrate this singularity accurately. Suppose that $\{t^c_\beta\}$ and $\{w^c_\beta\}$ are the Duffy quadrature points and weights, respectively, for the vertex t^c. Then the numerical quadrature form of the second sum in (35) is

$$\sum_{j:x_i\in P_j}\sum_\beta K\left(m_j(t^{c_{ij}}), m_j(t^{c_{ij}}_\beta)\right)J_j\left(t^{c_{ij}}_\beta\right)\sigma\left(t^{c_{ij}}_\beta\right)w^{c_{ij}}_\beta. \tag{37}$$

Once the numerical quadrature is ready, we can solve the integral Eq. (32). The standard method is the GMRES algorithm, which requires evaluating the integral at each iteration. Therefore, from a computational viewpoint, the main task is to evaluate the two sums (36) and (37) efficiently. Let N be the number of vertices $\{x_i\}_{i\in I}$. For the sum (37), since $x_i \in P_j$ for a small number of patches P_j and the number of quadrature points $\{t^c_\beta\}$ in the Duffy quadrature is constant, the cost of computing the second sum is proportional to $O(N)$. On the other hand, for the sum (36), one needs to iterate over all P_j such that $x_i \notin P_j$ for each x_i. Since there are $O(N)$ such P_j, the direct evaluation takes $O(N^2)$ steps. In order to speed up this calculation, we write it (36) as

$$\sum_j\sum_\alpha K(x_i,m_j(t_\alpha))J_j(t_\alpha)\sigma(t_\alpha)w_\alpha - \sum_{j:x_i\in P_j}\sum_\alpha K(x_i,m_j(t_\alpha))J_j(t_\alpha)\sigma(t_\alpha)w_\alpha. \tag{38}$$

The first sum in (38) is in fact an N-body problem with sources $\{J_j(t_\alpha)\sigma(t_\alpha)w_\alpha\}_{j,\alpha}$ at $\{m_j(t_\alpha)\}_{j,\alpha}$, targets at $\{x_i\}_{i\in I}$, and kernel $K(x,y)$. An algorithm for evaluating this N-body problem rapidly will be given in Section 4.2. As to the second sum in (38), since it is over $j : x_i \in P_j$ for each x_i, the overall cost is again proportional to $O(N)$. Therefore, the algorithm for evaluating the integral $\int_S K(x_i,y)\sigma(y)dy$ takes the following form:

Algorithm 1. Numerical quadrature for $u_i = \int K(x_i,y)\sigma(y)ds_y$.

```
1: Apply Algorithms 3 and 4 in Section 4.2 to compute
   $u^1_i = \sum_j \sum_\alpha K(x_i,m_j(t_\alpha))J_j(t_\alpha)\sigma(t_\alpha)w_\alpha.$
2: For each $x_i$, compute
   $u^2_i = \sum_{j:x_i\in P_j} \sum_\alpha K(x_i,m_j(t_\alpha))J_j(t_\alpha)\sigma(t_\alpha)w_\alpha.$
3: For each $x_i$, compute
   $u^3_i = \sum_{j:x_i\in P_j} \sum_\beta K(m_j(t^{c_{ij}}), m_j(t^{c_{ij}}_\beta))J_j(t^{c_{ij}}_\beta)\sigma(t^{c_{ij}}_\beta)w^{c_{ij}}_\beta.$
4: Finally, for each $x_i$,
   $u_i = u^1_i - u^2_i + u^3_i.$
```

This algorithm should be repeated for four different cases for our integral equation

$$\int_S K_{11}(x_i,y)\left(\sum_j \sigma_{1,j}\phi_j(y)\right)ds_y, \quad \int_S K_{12}(x_i,y)\left(\sum_j \sigma_{2,j}\phi_j(y)\right)ds_y, \tag{39}$$

$$\int_S K_{21}(x_i,y)\left(\sum_j \sigma_{1,j}\phi_j(y)\right)ds_y, \quad \int_S K_{22}(x_i,y)\left(\sum_j \sigma_{2,j}\phi_j(y)\right)ds_y. \tag{40}$$

Our discretization is not optimal in terms of convergence rate. A spectrally accurate scheme for discretizing surface integrals can be found in [4]. However, the reason for adopting the current collocation scheme is its simplicity and flexibility to work with surfaces with singularities with edges and corners.

4.2. Fast matrix–vector multiplication

In this section, we describe the algorithm of evaluating the first integral in the previous algorithm. In a slightly more general form, given the points $\{p_i\}$, the normal directions $\{n_i\}$ at $\{p_i\}$, and the weights $\{f_i\}$, the goal is to compute for each i

$$u_i = \sum_j K(p_i, n_i; p_j, n_j) f_j. \tag{41}$$

The normal directions n_i are assume to depend smoothly on the points p_i, i.e., $p_i \approx p_j$ implies $n_i \approx n_j$. Here we use K to denote the four kernels K_{11}, K_{12}, K_{21}, and K_{22}, and make their dependence on the normal direction explicit. Also, there is no distinction between the source points and target points. In order to compute the summations in (39) and (40), we take the point sets to be the union of the sources $\{m_j(t_\alpha)\}_{j,\alpha}$ and the targets $\{x_i\}_{i \in I}$, set the f_j associated to the targets to be zero, and extract the potentials u_i only at targets.

We define the ϵ-rank r_ϵ of a matrix A of size $m \times n$ to be the number of the singular values that are greater than ϵ. The matrix A is called numerically low-rank if the ϵ-rank r_ϵ is much smaller compared to the dimensions of the matrix A and grows extremely slowly when ϵ decreases. The essential idea of the algorithm for evaluating (41) rapidly is that, given two sets B_1 and B_2 well-separated (the precise definition of well-separatedness will be given later), the matrix that represents the interaction restricted to B_1 and B_2,

$$(K(p_i, n_i; p_j, n_j))_{p_i \in B_1, p_j \in B_2} \tag{42}$$

is numerically low-rank. The following randomized algorithm, first introduced in [11], constructs a low rank approximation of such a matrix in a cost that is essentially linear to the dimension of the matrix.

Algorithm 2. Randomized factorization of a numerical low-rank matrix A.

1: Select randomly a set Ω_1 of $\gamma \cdot r_\epsilon$ rows from the matrix A. Perform the pivoted QR factorization to $A(\Omega_1,:)$. Define Π_1 to be the index set that contains the first r_ϵ pivoted columns of $A(\Omega_1,:)$. The columns of the submatrix $A(:,\Pi_1)$ then serve as a stable approximate basis for the column space of A.

2: Select randomly a set Ω_2 of $\gamma \cdot r_\epsilon$ columns from the matrix A. Perform the pivoted QR factorization to $A(:,\Omega_2)^T$. Define Π_2 to be the index set that contains the first r_ϵ pivoted columns of $A(:,\Omega_2)^T$. The rows of the submatrix $A(\Pi_2,:)$ then serve as a stable approximate basis for the row space of A.

3: Define the matrix $M = (A(\Pi_2, \Pi_1))^+$ where $(\cdot)^+$ stands for the pseudo-inverse. Then

$$A(:,\Pi_1) \cdot M \cdot A(\Pi_2,:) \tag{43}$$

is a rank-r_ϵ approximation of the matrix A.

This method works quite well in practice. Typically the constant γ is chosen to be 3 or 4. Since r_ϵ is typically a small constant, it is not difficult to see that the overall cost of this algorithm is of order $O(r_\epsilon^2 \cdot (m+n)) = O(m+n)$. Once this factorization is constructed, the product of A with any vector f can be computed with

$$A(:,\Pi_1)(M(A(\Pi_2,:)f)) \tag{44}$$

in $O(m+n)$ steps.

The algorithm for evaluating (41) consists of two stages: setup and evaluation. In the setup step, Algorithm 2 is invoked to generate the low rank factorization for each submatrix of form (42).

Algorithm 3: Setup stage for $u_i = \sum_j K(p_i, n_i; p_j, n_j) f_j$.

```
Construct the octree structure. Starting from a square box that contains all points $\{p_i\}$, we subdivided the boxes
  recursively into eight children boxes until the number of points in each leaf box is bounded by a prescribed constant.
Let $B$ be the top level box of the octree. Insert the pair $(B,B)$ into a queue $Q$.
Let $L$ be an empty list.
while $Q$ is not empty do
  Pop the top entry from $Q$, say $(B_1,B_2)$.
  if both $B_1$ and $B_2$ contain points then
    if none of $B_1$ and $B_2$ are leaf boxes then
      if $B_1$ and $B_2$ are well-separated then
```

(continued on next page)

Algorithm 3 (*continued*)

```
                Compress the matrix $\left(K(p_i, n_i; p_j, n_j)\right)_{p_i \in B_1, p_j \in B_2}$ using the randomized procedure. Insert the triple
                  $(B_1, B_2,$"compressed"$)$ into the list $L$.
            else
                Let $B_{1,\alpha}$ be the children of $B_1$ and $B_{2,\beta}$ be the children of $B_2$. Insert all pairs $(B_{1,\alpha}, B_{2,\beta})$ into $Q$.
            end if
        else
            Insert the triple $(B_1, B_2,$"dense"$)$ into the list $L$.
        end if
    end if
end while
```

A missing piece of the above algorithm is the definition of *well-separated*. In our implementation, B_1 and B_2 are well-separated if the intersection of B_1 and B_2 is either empty or a single vertex. This criteria works quite well in practice.

Once the setup stage is done, the evaluation stage speeds up to the matrix vector multiplication by using the constructed factorizations whenever possible.

Algorithm 4: Evaluation stage for $u_i = \sum_j K(p_i, n_i; p_j, n_j) f_j$

```
for each triple of the list $L$ do
    if the triple is of form $(B_1, B_2,$"compressed"$)$ then
        Evaluate the interaction between boxes $B_1$ and $B_2$ using the compressed form
    else
        Evaluate the interaction between boxes $B_1$ and $B_2$ densely.
    end if
end for
```

Obviously, the kernels K_{12} and K_{22} can be directly handled with the adaptive new version FMM in [6]. It is also possible that with some modifications the FMM for the biharmonic equations [19] can be used for kernels K_{11} and K_{21}. Furthermore, there have been many kernel-independent FMMs developed recently (see, for example, [13,14,27,36–38]). The algorithm outlined above is essentially a tree algorithm (see, for example, [3]) with multipole expansion replaced by randomized skeletonization (see, for example, [5]). We have chosen to implement our algorithm because the algorithm is easier to implement as compared with the FMMs, and the algorithm is directly applicable to a broader class of kernels.

5. Numerical examples

A C++ code has been written implementing the algorithm described in the preceding section. In this section, we demonstrate the performance of the scheme with several numerical examples. We test the accuracy of our procedure by constructing a function which is biharmonic in D and setting the boundary data equal to the function value and its normal derivative on S. Since Green's function and its partial derivatives satisfy the biharmonic equation in D when the source is outside D, such function can be constructed, say, by forming a linear combination of Green's functions (and its partial derivatives) with the sources randomly distributed outside the computational domain D. The accuracy and speed of our algorithm depend on several parameters. In the tables below, we set the precision of the fast matrix–vector algorithm to 10^{-6}, the error tolerance of GMRES to 10^{-6}. We choose the degree of Gaussian quadrature for smooth integrals to be 5 and the degree of the Duffy quadrature for the weakly singular integral to be 5, so that the quadrature error from the triangular Gaussian quadrature and the Duffy quadrature is always smaller than the discretization error of the collocation method (32) in the following tests.

In each of those tables, the first column contains the total number of unknowns N of the whole linear system, which is twice the total number of vertices in the triangulation of the boundary surface. The second column contains the setup time of the randomized matrix compression. The third column contains the total time for solving the linear system. The fourth column contains the number of iterations of GMRES. The last column contains the relative L^2 error of the numerical solution as compared with the analytic solution of the original BVP where the checking points lie on a smaller surface inside the boundary S.

The purpose of the numerical tests is three-fold. Firstly, the numerical discretization scheme presented in Section 4.1 is formally second order since piecewise linear basis functions are used. Therefore, if the size of the triangles is halved (or equivalently the number of unknowns N roughly quadruples), the accuracy should decrease by a factor of four. Secondly, the fast algorithm described in Section 4.2 has essentially a linear complexity. Hence, each time N quadruples, the solution time should roughly grows by the same factor. Finally, since our second kind integral equation is well-conditioned, the number of iterations of the iterative solver should essentially be independent of the number of unknowns. In each numerical test, we start from a fairly coarse discretization to a fairly dense one. We would like to point out that, for the coarsest discretization, standard dense solvers are often faster than the iterative solvers used here. However, since the goal is to demonstrate the two scaling behaviors mentioned above, all linear systems are solved using the iterative method.

Example 1. In this example, the boundary is a unit sphere centered at the origin:

$$x_1^2 + x_2^2 + x_3^2 = 1. \tag{45}$$

The atlas consists of six charts with each one corresponding to one six principal axes: $\pm x_1, \pm x_2$, and $\pm x_3$.The parameter domain of each chart is discretized with a Cartesian grid where each rectangle is further split into two triangles. The triangulated surface is shown in Fig. 1. The numerical results are summarized in Table 1.

Example 2. In this example, the boundary is an ellipsoid centered at the origin:

$$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} + \frac{x_3^2}{c^2} = 1, \tag{46}$$

where $a = 2$, $b = 1$, $c = 1$. The parameterization and triangulation are essentially the same as those in Example 1 with proper scalings. The triangulated surface is shown in Fig. 2. The numerical results are summarized in Table 2.

Example 3. In this example, the boundary is a "twisted" ellipsoid given by the formula

$$\frac{x_1^2}{a^2} + \frac{\left(x_2 - \frac{d}{c} \cdot \cos(\pi x_3)\right)^2}{b^2} + \frac{x_3^2}{c^2} = 1, \tag{47}$$

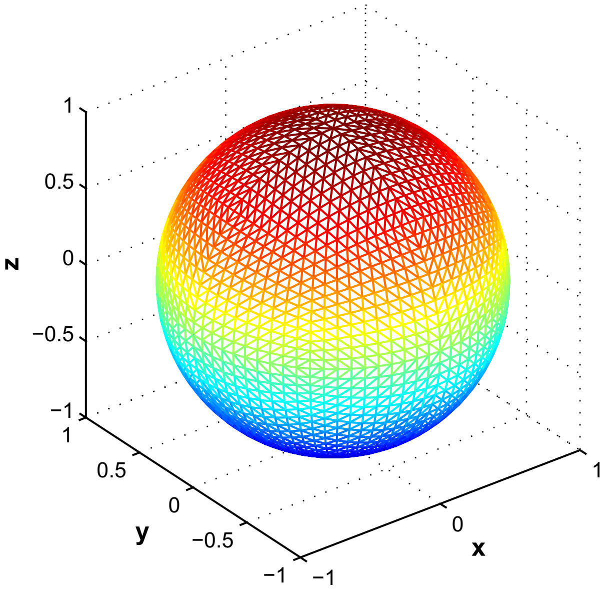


Fig. 1. Triangulated boundary surface for Example 1.

Table 1
Numerical results for Example 1.

N	T_{setup}	T_{solve}	N_{iter}	$E(L^2)$
196	2.80*e*−01	1.44*e*+00	16	3.27*e*−02
772	5.98*e*+00	1.10*e*+01	18	1.05*e*−02
3076	8.54*e*+01	4.81*e*+01	18	2.87*e*−03
12292	7.97*e*+02	2.20*e*+02	16	7.38*e*−04

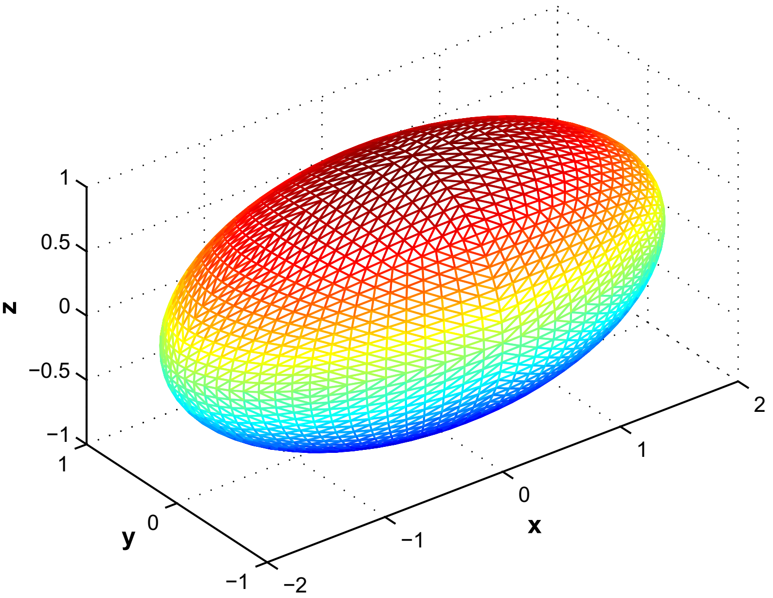


Fig. 2. Triangulated boundary surface for Example 2.

Table 2
Numerical results for Example 2.

N	T_{setup}	T_{solve}	N_{iter}	$E(L^2)$
196	1.35*e*+00	3.56*e*+00	17	6.17*e*−02
772	1.92*e*+01	2.40*e*+01	21	1.99*e*−02
3076	2.07*e*+02	1.12*e*+02	20	5.42*e*−03
12292	1.84*e*+03	4.81*e*+02	20	1.40*e*−03

Table 3
Numerical results for Example 3.

N	T_{setup}	T_{solve}	N_{iter}	$E(L^2)$
196	3.60*e*−01	2.05*e*+00	20	1.44*e*−01
772	1.07*e*+01	1.42*e*+01	22	2.22*e*−02
3076	1.03*e*+02	1.13*e*+02	24	8.03*e*−03
12292	1.11*e*+03	4.36*e*+02	24	2.16*e*−03

where $a = 1$, $b = 1$, $c = 2$, and $d = 0.2$. The parameterization and triangulation are essentially the same as those in Example 2 with the term $\cos(\pi x_3)$ treated as a small perturbation. The triangulated surface is shown in Fig. 3. The numerical results are summarized in Table 3.

Example 4. In this example, the boundary is a torus given by the formula

$$\left(\sqrt{x_1^2 + x_2^2} - R\right)^2 + x_3^2 = r^2, \tag{48}$$

where $R = 3$, $r = 1$. The atlas consists of a single chart with a parameterization given by the formulas

$$x_1 = (R + r \cdot \cos(\theta))\cos(\psi), \quad x_2 = (R + r \cdot \cos(\theta))\sin(\psi), \quad x_3 = r \cdot \sin(\theta), \tag{49}$$

with $(\theta, \psi) \in [-\pi, \pi)^2$. We then use a rectangular grid to discretize (θ, ψ) and each small rectangle is further split into two triangles. The triangulated surface is shown in Fig. 4. The numerical results are summarized in Table 4.

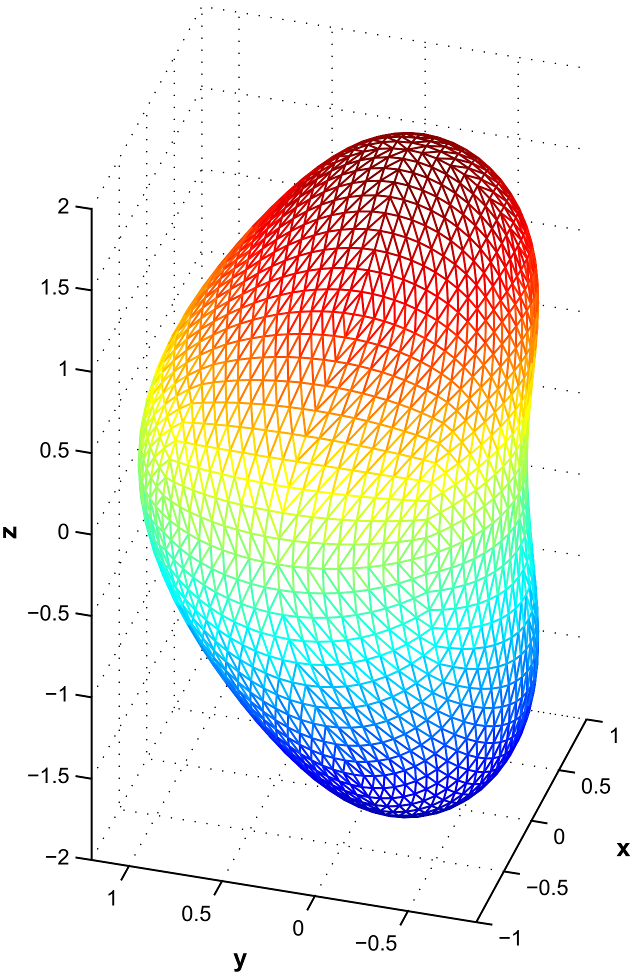


Fig. 3. Triangulated boundary surface for Example 3.

Table 4
Numerical results for Example 4.

N	T_{setup}	T_{solve}	N_{iter}	$E(L^2)$
216	3.70*e*−01	1.59*e*+00	18	2.67*e*−01
864	1.11*e*+01	1.61*e*+01	26	6.40*e*−02
3456	1.40*e*+02	8.78*e*+01	26	2.20*e*−02
13824	1.41*e*+03	3.96*e*+02	26	6.06*e*−03

In the above examples, we observe that the convergence rate is roughly equal to 2 on average, which is consistent with the convergence order of our discretization scheme. The solution time T_{solve} grows by a factor of 4 to 7 each time N quadruples. This is roughly consistent with the $N\log N$ theoretical estimate. The fluctuations in the growth factor of T_{solve} are due to various factors, such as the sudden change in the depth of the octree and the changes in the number of iterations of the GMRES algorithm. Most importantly, we observe that the number of iterations is very low and roughly independent of total number of unknowns, which indicates that our SKIE formulation is well-conditioned.

In our numerical examples, we triangulate the surfaces in a straightforward manner and compute the mean curvature at each vertex analytically. Obviously, the quality of triangulation affects the accuracy greatly and in general the triangulation and the computation of the mean curvature should be handled via more general and robust algorithms available in the community of finite (or boundary) elements.

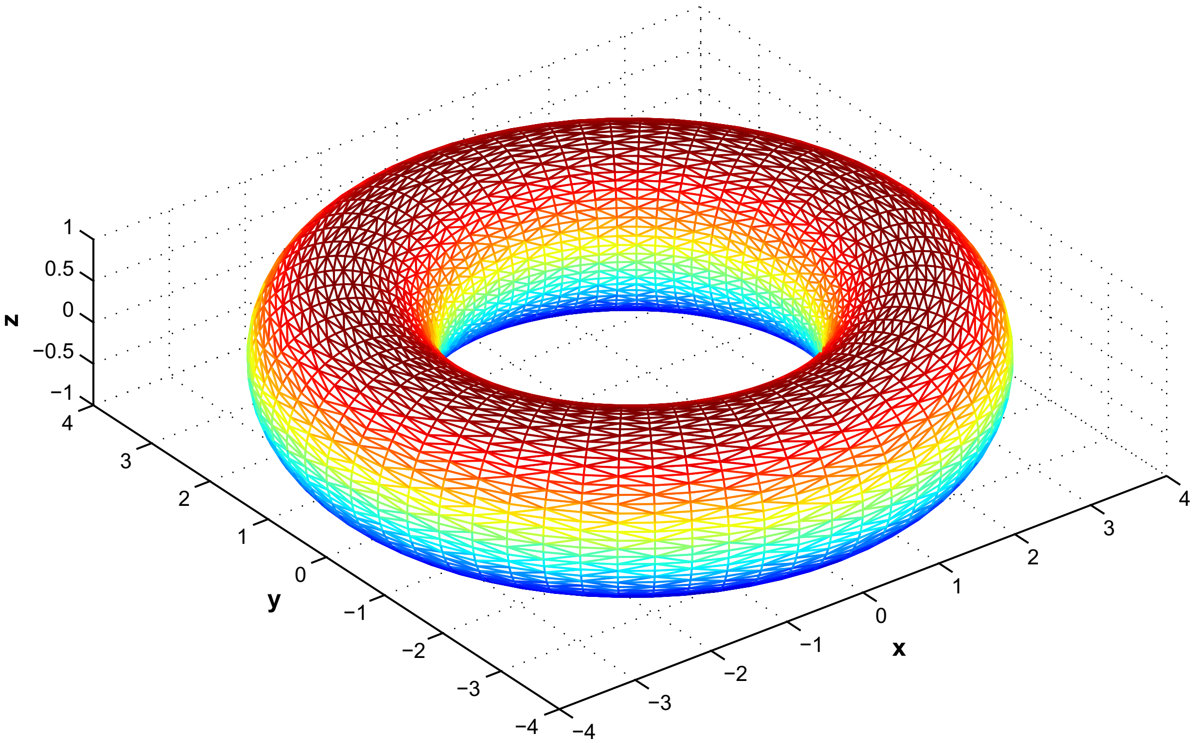


Fig. 4. Triangulated boundary surface for Example 4.

6. Conclusions and further discussions

We have constructed a second kind integral equation formulation for the first Dirichlet problem for the biharmonic equation in three dimensions. A fast algorithm and a second order discretization scheme have also been developed for solving the boundary integral equations. One may try to use this approach to construct SKIEs for other boundary value problems for the biharmonic equation. For example, the problem of finding u which satisfies

$$\begin{aligned}&\Delta^2 u = 0 \quad \text{on } D\\ &\frac{\partial u}{\partial n} = f_1 \quad \text{on } S\\ &\frac{\partial^2 u}{\partial n^2} = f_2 \quad \text{on } S\end{aligned}$$

for given f_1 and f_2, can be solved by looking for u as

$$u(x) = \int_S [(-2G_{nn} + 3\Delta G)(x,y)\sigma_1(y) + G_n(x,y)\sigma_2(y)]ds_y$$

for which the matrix of diagonal terms is

$$\begin{pmatrix} \frac{1}{2} & 0 \\ \frac{3}{2}H(x) & \frac{1}{2} \end{pmatrix}.$$

Our construction can also be easily extended to the first Dirichlet problem for the biharmonic equation in higher dimensions ($\geqslant 4$) and may be applicable for solving certain boundary value problems for polyharmonic equations.

The biharmonic equation has many applications in mathematical physics. Some direct applications of the first Dirichlet problem include solving the Cahn-Hillard equations for phase-transition in material science, and the phase-field models in two-phase flows (see, for example, [25]). These problems involve time-dependent nonlinear PDEs whose principal part is a biharmonic operator. The first Dirichlet conditions arise naturally in such problems since they are the natural conditions from variational principles. We would also like to remark that our method can be adapted to construct the SKIEs for the modified biharmonic equation, which is the governing equation of the stream function for unsteady Stokes flow. And an efficient numerical algorithm based on the SKIE formulation for the modified biharmonic equation will eventually lead to a numerical scheme for solving the Navier-Stokes equation (see, for example, [16]). These problems are currently under investigation and the results will be reported in a later date.

Acknowledgements

S.J. was supported in part by NSF under Grant CCF-0905395 and would like to thank Prof. Leslie Greengard at Courant Institute for stimulating discussions. L.Y. was supported in part by the Sloan Foundation under a Sloan Research Fellowship and by NSF under grant DMS-0708014 and CAREER Grant DMS-0846501.

References

[1] S. Agmon, Multiple layer potentials and the Dirichlet problem for higher order elliptic equations in the plane, Commun. Pure & Appl. Math. 10 (1957) 179–239.

[2] I. Altas, J. Erhel, M.M. Gupta, High accuracy solution of three-dimensional biharmonic equations, matrix iterative analysis and biorthogonality (Luminy, 2000), Numer. Algorithms 29 (1-3) (2002) 1–19.

[3] J. Barnes, P. Hut, A hierarchical O(NlogN) force-calculation algorithm, Nature 324 (4) (1986).

[4] O.P. Bruno, L.A. Kunyansky, A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests, and applications, J. Comput. Phys. 169 (1) (2001) 80–110.

[5] H. Cheng, Z. Gimbutas, P.G. Martinsson, V. Rokhlin, On the compression of low rank matrices, SIAM J. Sci. Comput. 26 (4) (2005) 1389–1404.

[6] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, J. Comput. Phys. 155 (2) (1999) 468–498.

[7] J. Cohen, J. Gosselin, The Dirichelt problem for the biharmonic equation in a C^1 domain in the plane, Indiana Univ. Math. J. 32 (5) (1983) 635–685.

[8] J. Cohen, J. Gosselin, Adjoint boundary value problems for the biharmonic equation on C^1 domains in the plane, Arkiv for Matematik 23 (2) (1985) 217–240.

[9] M. Costabel, I. Lusikka, J. Saranen, Comparison of three boundary element approaches for the solution of the clamped plate problem, Boundary Elements IX 2 (1987) 19–34.

[10] M.G. Duffy, Quadrature over a pyramid or cube of integrands with a singularity at a vertex, SIAM J. Numer. Anal. 19 (1982) 1260–1262.

[11] B. Engquist, L. Ying, A fast directional algorithm for high frequency acoustic scattering in two dimensions, Commun. Math. Sci. 7 (2) (2009) 327–345.

[12] P. Farkas, Mathematical foundations for fast algorithms for the biharmonic equations, Ph.D. Thesis, University of Chicago, 1989.

[13] W. Fong, E. Darve, The black-box fast multipole method, J. Comput. Phys. 228 (23) (2009) 8712–8725.

[14] Z. Gimbutas, V. Rokhlin, A generalized fast multipole method for nonoscillatory kernels, SIAM J. Sci. Comput. 24 (3) (2002) 796–817.

[15] R. Glowinski, O. Pironneau, Numerical methods for the first biharmonic equation and the two-dimensional Stokes problem, SIAM Rev. 21 (2) (1979) 167–212.

[16] L. Greengard, M.C. Kropinski, An integral equation approach to the incompressible Navier–Stokes equations in two dimensions, SIAM J. Sci. Comput. 20 (1) (1998) 318–336.

[17] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (2) (1987) 325–348.

[18] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the Laplace equation in three dimensions, Acta Numerica 6 (1997) 229–269.

[19] N.A. Gumerov, R. Duraiswami, Fast multipole method for the biharmonic equation in three dimensions, J. Comput. Phys. 215 (1) (2006) 363–383.

[20] S. Jiang, Jump relations of the quadruple layer potential on a regular surface in three dimensions, Appl. Comput. Harmon. Anal. 21 (2006) 395–403.

[21] R. Kress, Linear Integral Equations, second ed., Springer, 1999.

[22] G. Lauricella, Sull'integrazione dell'equazione $\Delta^2 U = 0$, Rendiconti Lincei, Serie V XVI (1907) 373–383. 2 sem.

[23] G. Lauricella, Sur l'integration de l'equation relative a l'equilibre des plaques elastiques encastrees, Acta Mathematica XXXII (1909) 201–256.

[24] G. Lauricella, Sull'integrazione dell'equazione $\Delta^{2i} U = 0$ per le aree piane, Rendiconti Lincei, Serie V XVIII (1909) 5–15. 2 sem.

[25] C. Liu, J. Shen, A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method, Physica D 179 (2003) 211–228.

[26] R. Marcolongo, La teoria delle equazioni integrali e le sue appliquazioni alla Fisica-matematica, Rendiconti Lincei, Serie V XVI (1907) 742–749. 1 sem.

[27] P.G. Martinsson, V. Rokhlin, An accelerated kernel-independent fast multipole method in one dimension, SIAM J. Sci. Comput. 29 (3) (2007) 1160–1178.

[28] S.G. Mikhlin, Integral Equations, Pergamon Press, New York, 1957.

[29] M. Nicolesco, Les Fonctions Polyharmoniques, Hermann et Co, Paris, 1936.

[30] A.P.S. Selvadurai, Partial Differential Equations in Mechanics, vol. 2, Springer, 2000.

[31] J. Shen, Efficient spectral-Galerkin method. I. Direct solvers of second- and fourth-order equations using Legendre polynomials, SIAM J. Sci. Comput. 15 (61) (1994) 1489–1505.

[32] J. Shen, Efficient spectral-Galerkin method. II. Direct solvers of second- and fourth-order equations using Chebyshev polynomials, SIAM J. Sci. Comput. 16 (1) (1995) 74–87.

[33] E.M. Stein, G. Weiss, Introduction to Fourier Analysis on Euclidean Spaces, Princeton University Press, New Jersey, 1971.

[34] I.N. Vekua, New Methods for Solving Elliptic Equations, John Wiley and Sons Inc.,, New York, 1967.

[35] G. Verchota, The Dirichlet problem for the biharmonic equation in C^1 domains, Indiana Univ. Math. J. 36 (1987) 867–895.

[36] L. Ying, A kernel independent fast multipole algorithm for radial basis functions, J. Comput. Phys. 213 (2) (2006) 451–457.

[37] L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole algorithm in two and three dimensions, J. Comput. Phys. 196 (2) (2004) 591–626.

[38] B. Zhang, Integral-equation-based fast algorithms and graph-theoretic methods for large-scale simulations, Ph.D. Thesis, University of North Carolina, Chapel Hill, 2010.