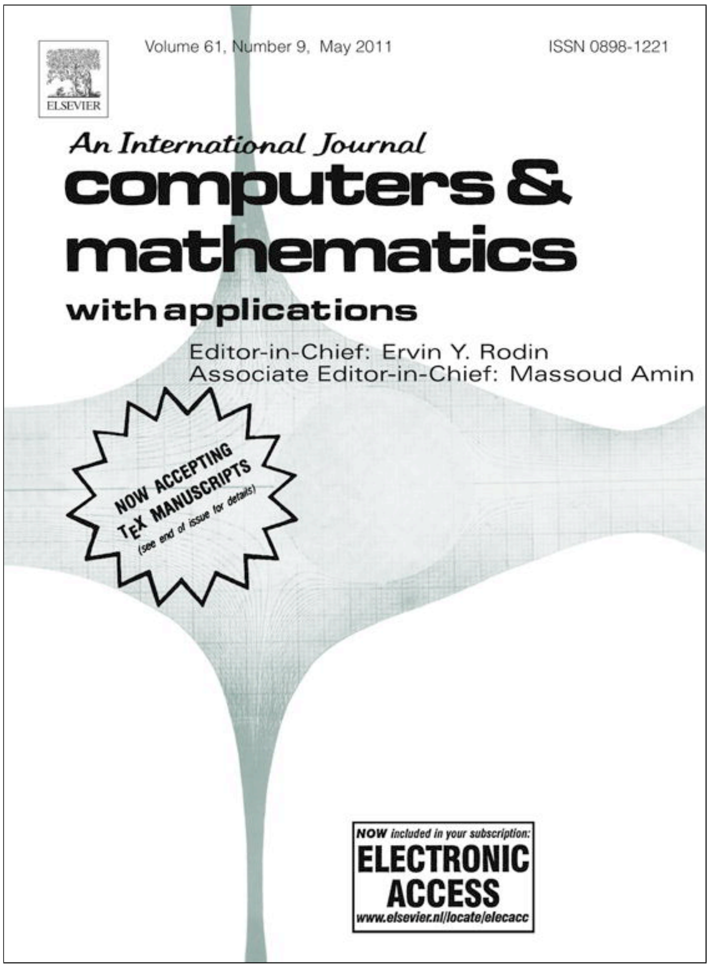

Volume 61, Number 9, May 2011 ISSN 0898-1221

ELSEVIER

An International Journal
computers & mathematics
with applications

Editor-in-Chief: Ervin Y. Rodin
Associate Editor-in-Chief: Massoud Amin

Computers and Mathematics with Applications 61 (2011) 2436–2446

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Fast integral equation methods for Rothe's method applied to the isotropic heat equation

Mary Catherine A. Kropinski *, Bryan D. Quaife

Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, Canada V5A 1S6

ARTICLE INFO

Article history:
Received 15 October 2010
Received in revised form 18 February 2011
Accepted 19 February 2011

Keywords:
Fast multipole method
Gaussian quadrature
Modified Helmholtz equation
Integral equations
The heat equation
Rothe's method
Allen–Cahn equation

ABSTRACT

We present an efficient integral equation approach to solve the forced heat equation, $u_t(\mathbf{x}) - \Delta u(\mathbf{x}) = F(\mathbf{x}, u, t)$, in a two-dimensional, multiply-connected domain, with Dirichlet boundary conditions. Instead of using an integral equation formulation based on the heat kernel, we discretize in time, first. This approach, known as Rothe's method, leads to a non-homogeneous modified Helmholtz equation that is solved at each time step. We formulate the solution to this equation as a volume potential plus a double layer potential, and both of these potentials are calculated with available tools accelerated by the fast multipole method. For a total of N points in the discretization of the boundary and the domain, the total computational cost per time step is $O(N)$. We demonstrate our approach on the heat equation and the Allen–Cahn equation.

1. Introduction

Integral equation methods offer an attractive alternative to conventional finite difference and finite elements for solving partial differential equations that arise in science and engineering. They offer several advantages: complex physical boundaries are easy to incorporate, the ill-conditioning associated with directly discretizing the governing equation is avoided, high-order accuracy is easier to attain, and far-field boundary conditions are handled naturally. Nevertheless, integral equation methods have been slow to be adopted for larger-scale problems, and the reason for this is clear: on an $N \times N$ mesh, integral equation formulations lead to systems with dense $N^2 \times N^2$ matrices. This situation is further exacerbated if the problem is time dependent.

In recent years, fast algorithms have been developed for integral equations for a variety of elliptic boundary value problems that are closely related to the methods we will discuss here. An early example is the work by Greenbaum et al. in [1] for Laplace's equation in multiply-connected domains. Subsequently, similar tools were developed for Poisson's equation [2], the Stokes equations [3], and the modified Helmholtz equation [4,5]. All of these approaches are characterized by the following: well-conditioned integral equations, if needed, are formulated, integrals are discretized using high-order quadrature, and the resulting linear systems are solved or evaluated using a fast multipole accelerated iterative solver. As a result, these tools are highly accurate and have optimal efficiency: for N points in the discretization of the boundary or domain, the solution procedure requires only $O(N)$ operations.

The development of corresponding algorithms for time-dependent problems, however, is considerably more complex. There is a substantial literature on integral equation methods for the heat equation. However, standard discretizations of these integral equations lead to very expensive methods, as the solution depends on the full space–time history. For N points in the discretization of the domain and M time steps, the computational cost requires $O(N^2M + N^2M^2)$ operations [6].

* Corresponding author.
E-mail address: mkropins@cs.sfu.ca (M.C.A. Kropinski).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.02.024

There is recent work by Li and Greengard in [6,7] on developing fast solvers for the heat equation. Their methods use an integral equation formulation based on the heat kernel and rely on a non-uniform FFT. While robust, this approach is highly complicated. Methods that rely on a fast Gauss transform are discussed in [8–10]. (For a more thorough overview of fast multipole accelerated integral equation methods, see [11].)

We present an alternative approach which relies on available fast algorithms for elliptic problems. A semi-implicit temporal discretization of a convection–diffusion-type equation yields the modified Helmholtz, or linearized Poisson–Boltzmann, equation

$$u(\mathbf{x}) - \alpha^2 \Delta u(\mathbf{x}) = B(\mathbf{x}, u, t), \tag{1}$$

where $\alpha^2 = O(\Delta t)$. Thus the parabolic heat equation is solved as a sequence of elliptic boundary value problems, and this approach is known as *Rothe's method*. In general, an integral equation formulation for solutions to (1) will require both a layer and volume potential. In [12], Chapko and Kress present a boundary-only integral equation formulation for Rothe's method applied to the heat equation. Chapko applies a similar approach to the unsteady Stokes equations in [13]. However, their boundary integral formulation applies only to homogeneous equations with homogeneous initial conditions. We aim to solve problems in a more general setting for which including a volume potential is unavoidable.

We base our solver on coupling two fast computational tools for the operator $(1 - \alpha^2 \Delta)$. In [4], Cheng et al. present a fast direct solver for (1) in two dimensions on the unit square. The solution is expressed as a volume potential, and the direct solver is accelerated using a new version of the fast multipole method [14,15]. The solver is fully adaptive and the computational costs are comparable to those of FFT-based methods. In [5], we present fast, well-conditioned integral equation methods for solving the homogeneous equation in unbounded and bounded multiply-connected domains, with either Dirichlet or Neumann boundary conditions. In this paper, we present a preliminary study on coupling the methods discussed in [4] to those discussed in [5] in order to solve nonlinear diffusion-type problems in complicated domains. In order to simplify our focus, we consider problems in bounded multiply-connected domains with constant Dirichlet boundary conditions. The purpose of this paper is twofold: to demonstrate that our approach has the potential to efficiently and accurately solve nonlinear diffusion-type problems in complicated two-dimensional domains, and to identify the key issues that will require more investigation in order to develop a more general solver. Once fully realized, our approach will have the following features:

- High-order temporal schemes will be easily implemented.
- The difficulty and expense associated with convolutions involving the heat kernel will be avoided.
- Spatial adaptivity for resolving features on the boundary and in the domain will be easily incorporated.
- The expense associated with generating an unstructured mesh for complicated geometry will be avoided.
- The solution will be based entirely on an integral formulation, resulting in a well-conditioned method.
- High-order quadrature will be used, resulting in a scheme that has high spatial accuracy.
- The FMM-accelerated solution procedure will have optimal or near-optimal efficiency.

There are integral equation based methods that are also well suited for the problem at hand. Our approach closely resembles the methods presented by Biros et al. in [16] for the Stokes equations and which were extended to the Navier–Stokes equations in [17]. Their method is called an *embedded boundary integral* method (EBI) which is an extension of the techniques developed by Mayo in [18] for efficiently evaluating the volume potentials for the Laplace and biharmonic equations. Instead of integrating in order to evaluate the volume potential, these approaches are based on a finite difference or finite element method with locally corrected stencils to account for jumps in the solution values across the interface or physical boundaries. In [17,16,18] a second-order method is used, although this approach has been extended to 4th order and recently to three dimensions [19,20]. The EBI method has the advantage of being easier to implement—an advantage which becomes even more critical when moving to three dimensions. However, in the case of the Navier–Stokes equations, there is some indication that the differencing of the convection term leads to more severe time-step constraints as the Reynolds number increases (cf. [17]). This is less problematic when using a purely integral formulation (cf. [21]), although this point warrants closer examination. Another method would be to generate an unstructured mesh in order to calculate the volume potential. However, we wish to avoid the difficulties associated with mesh generation and work with an approach that is naturally adaptive and will extend readily to three dimensions.

The paper is organized as follows: In Section 2, we discuss the temporal discretization of the heat equation and how this gives rise to the modified Helmholtz equation. We outline the corresponding potential theory for the modified Helmholtz equation: the solution is decoupled into the sum of a volume potential and double layer potential with an unknown density, and an integral equation for the unknown layer density is formulated. In Section 3, numerical methods for the evaluation of the volume potential and the integral equation are discussed. Numerical examples are presented in Section 4.

2. Potential theory and the integral equation formulation

To fix notation, let us consider a $(M+1)$-ply connected bounded domain Ω with boundary Γ which is comprised of individual smooth component curves Γ_k (see Fig. 1). Consider the isotropic inhomogeneous heat equation in $\mathbb{R}^2$:

$$u_t(\mathbf{x}, t) - \Delta u(\mathbf{x}, t) = F(\mathbf{x}, u, t), \quad \mathbf{x} \in \Omega, \ t > 0, \tag{2}$$

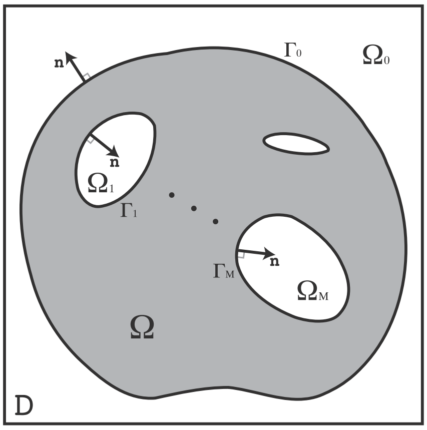

Fig. 1. A bounded $(M+1)$-multiply-connected domain Ω (in gray) is embedded in the unit square D. The outer boundary is denoted by Γ_0, the interior component curves by $\Gamma_1, \dots, \Gamma_M$, Γ is the union of all such curves, $\Omega_k,\ k = 1, \dots, M$ is the region in D bounded by Γ_k, and $\Omega_0 = D \setminus \Omega_1 \cup \dots \cup \Omega_M$. The unit normal $\mathbf{n}$ points out of Ω (and into $D \setminus \Omega$) on each component curve.

with Dirichlet boundary conditions:

$$u(\mathbf{x}, t) = f(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma,\ t > 0,$$

and initial condition

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega.$$

In order to prevent a severe time-step restriction, linearly implicit or IMEX schemes [22] are generally used for marching in time. In such schemes, the diffusive term is treated implicitly, while the remainder is treated explicitly. Regardless of the details of the particular choice of IMEX scheme, the temporal discretization of (2) yields the modified Helmholtz equation:

$$\begin{aligned} &u^{N+1} - \alpha^2 \Delta u^{N+1} = B(\mathbf{x}, t^N, t^{N-1}, \dots, u^N, u^{N-1}, \dots), && \mathbf{x} \in \Omega, \\ &u^{N+1} = f(\mathbf{x}, t), && \mathbf{x} \in \Gamma, \\ &u^0 = u_0(\mathbf{x}), && \text{at } t = 0, \end{aligned} \tag{3}$$

where $t = (N+1)\Delta t$ is the current time. The simplest such scheme is the first-order backward Euler method, which is of the form (3) with

$$\alpha^2 = \Delta t, \qquad B = u^N + \Delta t F^N.$$

A second-order method is the extrapolated Gear method (herein referred to as SBDF), for which

$$\alpha^2 = \frac{2}{3}\Delta t, \qquad B = \frac{4}{3}u^N - \frac{1}{3}u^{N-1} + \frac{4}{3}\Delta t F^N - \frac{2}{3}\Delta t F^{N-1}.$$

At each time step, we represent the solution to (3) as

$$U(\mathbf{x}) = U^p(\mathbf{x}) + U^h(\mathbf{x}).$$

Here, U^p satisfies

$$U^p - \alpha^2 \Delta U^p = B, \quad \mathbf{x} \in \Omega, \tag{4}$$

and U^h satisfies

$$\begin{aligned} &U^h - \alpha^2 \Delta U^h = 0, && \mathbf{x} \in \Omega, \\ &U^h = g(\mathbf{x}), && \mathbf{x} \in \Gamma, \end{aligned} \tag{5}$$

where $g(\mathbf{x}) = f(\mathbf{x}) - U^p(\mathbf{x})$.

The fundamental solution to the modified Helmholtz equation is

$$G(\mathbf{x}) = \frac{1}{2\pi\alpha^2} K_0\left(\frac{|\mathbf{x}|}{\alpha}\right),$$

where K_0 is the zeroth-order modified Bessel function of the second kind. In other contexts, $G(\mathbf{x})$ is also referred to as the Yukawa or screened Coulomb potential. We represent the solution to (4) as a volume potential:

$$U^p(\mathbf{x}) = \frac{1}{2\pi\alpha^2} \int_{\Omega} B(\mathbf{y}) K_0\left(\frac{|\mathbf{y} - \mathbf{x}|}{\alpha}\right) \mathrm{d}A_{\mathbf{y}}. \tag{6}$$

We seek the solution to (5) in the form of a double layer potential

$$U^h(\mathbf{x}) = \frac{1}{2\pi\alpha^2}\int_\Gamma \frac{\partial}{\partial n_{\mathbf{y}}} K_0\left(\frac{|\mathbf{y}-\mathbf{x}|}{\alpha}\right)\sigma(\mathbf{y})\mathrm{d}s(\mathbf{y}), \quad \mathbf{x}\in\Omega, \tag{7}$$

where $\sigma(\mathbf{y})$ is the value of an unknown density at the boundary point $\mathbf{y}$, and $\partial/\partial n_{\mathbf{y}}$ represents the outward normal derivative at the point $\mathbf{y}$. The unknown density σ is found by solving an integral equation derived to ensure that the boundary condition in (5) is satisfied.

In [5], we derive integral equation formulations to solve (5) in bounded and unbounded multiply-connected domains for both Neumann and Dirichlet boundary data. Here, we summarize the results for the bounded case with Dirichlet boundary conditions. Observing that $K_0(z) \sim -\log(z)$ as $z \to 0$ [23], we can determine the jump relations of the double layer potential as these are well known for the logarithmic kernel. Thus for any point $\mathbf{x}$ on the smooth boundary Γ,

$$\lim_{\substack{\mathbf{x}'\to\mathbf{x}\\ \mathbf{x}'\in\Omega}} U^h(\mathbf{x}') = -\frac{1}{2\alpha^2}\sigma(\mathbf{x}) + \frac{1}{2\pi\alpha^2}\int_\Gamma \frac{\partial}{\partial n_{\mathbf{y}}} K_0\left(\frac{|\mathbf{y}-\mathbf{x}|}{\alpha}\right)\sigma(\mathbf{y})\mathrm{d}s(\mathbf{y}).$$

Substituting the above into the boundary condition in (5) results in the integral equation

$$-\frac{1}{2\alpha^2}\sigma(\mathbf{x}) + \frac{1}{2\pi\alpha^2}\int_\Gamma \frac{\partial}{\partial n_{\mathbf{y}}} K_0\left(\frac{|\mathbf{y}-\mathbf{x}|}{\alpha}\right)\sigma(\mathbf{y})\,\mathrm{d}s(\mathbf{y}) = g(\mathbf{x}). \tag{8}$$

As we discuss in [5], the kernel in the above is continuous along Γ,

$$\lim_{\substack{\mathbf{y}\to\mathbf{x}\\ \mathbf{x},\mathbf{y}\in\Gamma}} \frac{\partial}{\partial n_{\mathbf{y}}} K_0\left(\frac{|\mathbf{y}-\mathbf{x}|}{\alpha}\right) = -\frac{1}{2}\kappa(\mathbf{x}),$$

where $\kappa(\mathbf{x})$ denotes the curvature of Γ at the point $\mathbf{x}$. Summarizing, the kernel in the integral equation is bounded and continuous, and the integral operator is therefore compact. In addition, there are no nontrivial homogeneous solutions. By the Fredholm alternative, (8) has a unique solution for any integrable data $g(\mathbf{x})$.

3. Numerical methods

We now discuss our numerical procedure. In Section 3.1, we first outline the methods involved in evaluating the volume potential (6). Next, in Section 3.2, we discuss the coupling of the volume potential to the double layer potential through the boundary conditions and solving the corresponding integral equation (8). While the FMM plays an integral role in our numerical methods, its implementation is standard and has been discussed at length in other works. For details on the FMM, we refer the reader to the work in [24,25] for the original FMM, [15] for the new FMM algorithm based on exponential expansions to represent the far-field interactions, and [4] for the FMM applied to the modified Helmholtz equation in two dimensions. In Section 3.3, the entire solution procedure taken per time step is summarized.

3.1. The volume potential

In [4], methods for the rapid evaluation of the volume potential for the modified Helmholtz equation on the unit square were discussed. These methods use adaptive mesh refinement, and the evaluation is accelerated using the new version of the FMM. In order to employ these methods, we must first extend the right-hand side of (2), which is defined only in Ω to be defined throughout D. One naive way to define an extension $\tilde{B}(\mathbf{x},t,u)$ of $B(\mathbf{x},t,u)$ is the following:

$$\tilde{B}(\mathbf{x},t,u) = \begin{cases} B(\mathbf{x},t,u), & \mathbf{x}\in\Omega\\ 0, & \mathbf{x}\in D\setminus\Omega.\end{cases}$$

However, the discontinuity across Γ would likely cause excessive grid refinement in the vicinity of the boundary, and consequently slow down the evaluation (cf. Example 4.4 in [2]; up to $O(10^5)$ grid points are needed to solve $\Delta u = 1$ inside a disk and $\Delta u = 0$ outside). For multiple-component boundaries, this would simply become too expensive.

For our purposes here, we consider problems in which the boundary conditions on each Γ_k are constant, and in addition, F does not depend explicitly on t or $\mathbf{x}$, i.e. $F = F(u)$. Consequently, B also does not depend explicitly on t or $\mathbf{x}$ and a continuous $\tilde{B}$ can easily be constructed. We save the important consideration of an extension of B in the case of general boundary conditions for future work. Define $\Omega_k,\ k = 1,\ldots,M$ to be the region in D bounded by Γ_k, and $\Omega_0 = D\setminus\Omega_1\cup\cdots\cup\Omega_M$ (see Fig. 1). Assume the Dirichlet boundary conditions are given by $u = C_k$ on Γ_k for $k = 0,\ldots,M$. Then, we define $\tilde{B}$ by:

$$\tilde{B}(u) = \begin{cases} B(u), & \mathbf{x}\in\Omega,\\ B(C_k), & \mathbf{x}\in\Omega_k,\\ B(C_0), & \mathbf{x}\in\Omega_0.\end{cases}$$

We nowoutline the methods presented in [4] to evaluate

$$\tilde{U}^p(\mathbf{x}) = \int_D \tilde{B}(\mathbf{x},t)G(\mathbf{y}-\mathbf{x})\,\mathrm{d}A_{\mathbf{y}}. \tag{9}$$

The FMM uses an adaptive quadtree structure in order to superimpose a hierarchy of refinement on the computational domain. The unit square D is considered to be grid level 0. Grid level $l+1$ is obtained recursively by subdividing each square (or node) s at level l into four equal parts; these are called the "children" of s. Adaptivity is achieved by allowing different levels of refinement throughout the tree. The square s is refined if the error of a third-order polynomial interpolating $\tilde{B}$ is larger than some preset tolerance, τ (the interpolation is checked at 64 points within s). We denote the childless nodes in the quadtree as $D_i,\ i=1,\dots,P$, where P is the total number of such nodes. We assume we are given $\tilde{B}$ on a cell-centered 4×4 grid for each D_i. Thus, $N_D = 16\times P$ is the total number of grid points in D. To obtain fourth-order accuracy, these 16 points are used to construct a third-order polynomial to $\tilde{B}$ of the form

$$\tilde{B}(\mathbf{x}) \approx \sum_{j=1}^{10} c_j^i p_j(\mathbf{x}-\mathbf{x}^i), \quad \mathbf{x}\in D_i,$$

where $\mathbf{x}^i$ is the center of D_i and $\{p_j\}$ are the standard basis functions for polynomials up to order three (see [4,2] for details on the approximating polynomial). Therefore, the evaluation of (9) is approximated by

$$\tilde{U}^p(\mathbf{x}) \approx \sum_{i=1}^{P}\int_{D_i} G(\mathbf{y}-\mathbf{x})\sum_{j=1}^{10} c_j^i p_j(\mathbf{y}-\mathbf{x}^i)\,\mathrm{d}A_{\mathbf{y}}.$$

Direct evaluation of this potential at all N_D grid points would require $O(N_D^2)$ work. This expense is reduced to $O(N_D)$ work, by using the FMM.

3.2. The integral equation

We now discuss the numerical methods to solve (8). We assume each component curve $\Gamma_k,\ k=0,\dots,M$ is parametrized by $\mathbf{y}^k(\alpha)$, where $\alpha \in [0,2\pi)$. Similarly, $\sigma^k(\alpha)$ refers to the restriction of the density σ on Γ_k. We are given N points equi-spaced with respect to α. Thus the mesh spacing is $h = 2\pi/N$, and the total number of discretization points is $N_\Gamma = (M+1)N$. Associated with each such point, denoted by $\mathbf{y}_j^k$, is an unknown density σ_j^k.

In order to approximate the integral operator in (8), we use hybrid gauss-trapezoidal quadrature rules developed by Alpert [26] which are tailored for integrands with logarithmic singularities. These quadratures are of order $h^p\log h$. The order p determines the nodes v_n and weights $u_n,\ n=1,\dots,l$, which are used for the quadrature within the interval $\alpha\in[\alpha_j - ha, \alpha_j + ha]$, on Γ_k (l and a are also determined by p). Outside of this interval, the quadrature is essentially the trapezoid rule. Applying this quadrature to (8) yields

$$\sigma_j^k - \frac{h}{\pi}\left\{\sum_{\substack{m=0\\ m\neq k}}^{M}\sum_{n=1}^{N} K(\mathbf{y}_n^m,\mathbf{y}_j^k)\sigma_n^m + \sum_{n=j+a}^{N+j-a} K(\mathbf{y}_n^k,\mathbf{y}_j^k)\sigma_n^k\right\} - \frac{1}{\pi}\sum_{\substack{n=-l\\ n\neq 0}}^{l} u_{|n|} K\left(\mathbf{y}^k_{j+\frac{n}{|n|}v_{|n|}},\mathbf{y}_j^k\right)\sigma^k_{j+\frac{n}{|n|}v_{|n|}} = -2\alpha^2 g(\mathbf{y}_j^k), \tag{10}$$

where

$$K(\mathbf{y},\mathbf{x}) = \frac{1}{\alpha}K_1\left(\frac{|\mathbf{y}-\mathbf{x}|}{\alpha}\right)\frac{\mathbf{y}-\mathbf{x}}{|\mathbf{y}-\mathbf{x}|}\cdot\mathbf{n}_{\mathbf{y}}.$$

In the middle sum, we invoke periodicity of all functions on Γ_k, or equivalently, $j+N=j$. In the final sum, we are required to know σ at intermediate values to the nodal values. These are found through Fourier interpolation via an FFT.

Eq. (10) is a linear system that is solved iteratively using the generalized minimum residual method GMRES [27]. The bulk of the work at each iteration lies in evaluating (10) at the current solution update. If this was done directly, it would require $O(N_\Gamma^2)$ work. This evaluation can be reduced to $O(N_\Gamma)$, again using the FMM (cf. [5] for more discussion on the details of implementation). Since the number of iterations needed to solve a Fredholm equation of the second kind to a fixed precision ϵ is bounded independent of N_Γ, we can estimate the total cost of solving (10) by

$$I(\epsilon)C(\epsilon)N_\Gamma,$$

where $I(\epsilon)$ is the number of GMRES iterations needed to reduce the residual error to ϵ, and $C(\epsilon)$ is the constant of proportionality in the FMM.

3.3. The solution procedure per time step

Below is an outline of the solution procedure:

STEP 1. Construct $\tilde{B}$ based on $u_0(\mathbf{x})$ and create an initial quadtree according to the specified precision τ. If the initial solution is homogeneous, we build the initial quadtree based on u^1, where $(1-\alpha^2\Delta)u^1 = 0$, $u^1 = C_k$ on Γ_k.

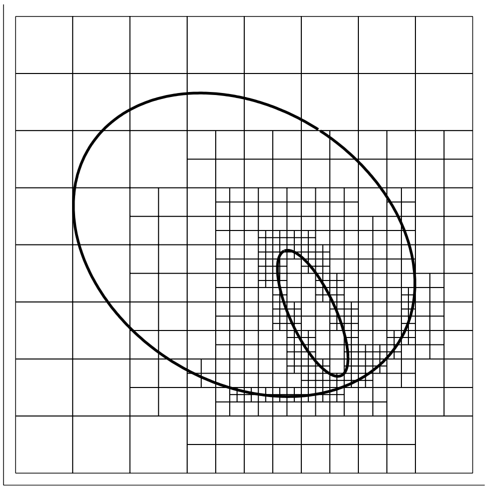

Fig. 2. The domain for Examples 1 and 4, including the quadtree structure for Example 4.

STEP 2. Calculate $\tilde{U}^p$ at all N_D points using the method outlined in Section 3.1. Use these values to construct the approximating polynomial to $\tilde{U}^p$ on each D_i,

$$\tilde{U}^p(\mathbf{x}) \approx \sum_{j=1}^{10} C_j^i \, p_j(\mathbf{x} - \mathbf{x}^i), \quad \mathbf{x} \in D_i. \tag{11}$$

STEP 3. For each $\mathbf{y}_j^k \in \Gamma$, determine in which childless node D_i this point resides, and evaluate $\tilde{U}^p(\mathbf{y}_j^k)$ according to (11). Calculate $g(\mathbf{y}_j^k) = C_k - \tilde{U}^p(\mathbf{y}_j^k)$.

STEP 4. Using the methods outlined in Section 3.2, solve the integral equation for the double layer density σ_j^k to a specified precision ϵ.

STEP 5. Using the FMM and the N_Γ density values, evaluate (7) at all N_D points in the quadtree.

STEP 6. At all points in the quadtree that reside in Ω, calculate $u^{N+1} = \tilde{U}^p + U^h$ and use these values to construct $\tilde{B}$ for the next time step.

STEP 7. Repeat, starting at step 2.

4. Numerical results

The algorithms described above have been implemented in Fortran. Here, we illustrate their performance on a variety of examples. In all examples involving the solution of an integral equation, the tolerance for the residual error in GMRES is set to 10^{-11} and the right-hand side of Eq. (10) is used as the initial guess. The maximum level of refinement in the quadtree is level 7, and the expansions in the FMM include enough terms so that the error is guaranteed to be less than 10^{-12}. All timings cited are for a single processor on a Mac Pro 2.1 with two 3 GHz Quad-Core Intel Xeon processors.

Example 1. The performance of the integral equation solver for (8) is carefully analyzed in [5]. In this example, we demonstrate some of the key features. Here, we solve $u - \alpha^2 \Delta u = 0$, with $\alpha^2 = 0.1$ in an elliptical region containing an off-center elliptical hole (see Fig. 2). The boundaries of the ellipses are described by $\mathbf{y} = \mathbf{c} + (a\cos\phi\cos\alpha - b\sin\phi\sin\alpha, a\sin\phi\cos\alpha + b\cos\phi\sin\alpha)$. On Γ_0, $\mathbf{c} = \mathbf{0}$, $a = 0.3$, $b = 0.4$, $\phi = 1$, and on Γ_1, $\mathbf{c} = (0.15, 0.15)$, $a = 0.15$, $b = 0.05$, $\phi = 2$. We generate the boundary conditions from an exact solution,

$$u(\mathbf{x}) = K_0\left(\frac{|\mathbf{x} - \mathbf{x}_1|}{\alpha}\right),$$

where $\mathbf{x}_1$ is a point inside Ω_1. The performance on this example is shown in Table 1. Note that the number of GMRES iterations remains constant and the CPU time depends linearly on N_Γ.

Example 2. In this example, we test the spatial convergence for the volume potential solver (more details on its performance appear in [4]). We calculate solutions to $u - \alpha^2 \Delta u = B(\mathbf{x})$, on the domain Ω bounded between Γ_0, a circle of radius 0.4, and Γ_1, a circle of radius 0.1. Here, we construct $B(r)$, $r = |\mathbf{x}|$, and the boundary conditions to be chosen according to the exact solution

$$u(r) = (r - 0.4)^4 + r - 0.4.$$

Table 1
Performance of the integral equation solver on Example 1, with an $O(h^8 \log h)$ quadrature rule. CPU time is measured in seconds. The absolute error is evaluated at a single point in the domain well away from the boundaries.

N_Γ	# Iterations	Error	CPU time
128	15	4.87×10^{-3}	0.81
256	15	1.23×10^{-5}	1.80
512	15	5.65×10^{-11}	4.22
1024	15	7.70×10^{-13}	6.37
2048	15	8.18×10^{-13}	11.40
4096	15	8.30×10^{-13}	21.57

Table 2
Performance of the volume potential solver on Example 2. CPU_{vol} is the time taken for the evaluation of the volume potential, measured in seconds. Error_1 is the absolute error at $r = 0.101$, Error_2 at $r = 0.25$ and Error_3 at 0.399. A linear fit determines the decay rate of the error, $\text{Error}_k = O(h^p)$, where $p = 2.8$ for $k = 1$, $p = 4.1$ for $k = 2$ and $p = 3.0$ for $k = 3$.

Max level	h	Error_1	Error_2	Error_3	CPU_{vol}
2	1/4	1.74×10^{-4}	5.80×10^{-3}	1.03×10^{-4}	0.11
3	1/8	3.51×10^{-5}	58.77×10^{-5}	8.26×10^{-6}	0.17
4	1/16	5.75×10^{-6}	53.08×10^{-6}	2.43×10^{-6}	0.27
5	1/32	1.89×10^{-7}	51.91×10^{-7}	7.62×10^{-8}	0.50
6	1/64	1.08×10^{-7}	55.34×10^{-8}	4.17×10^{-9}	1.35
7	1/128	1.55×10^{-8}	52.35×10^{-9}	8.20×10^{-9}	4.49

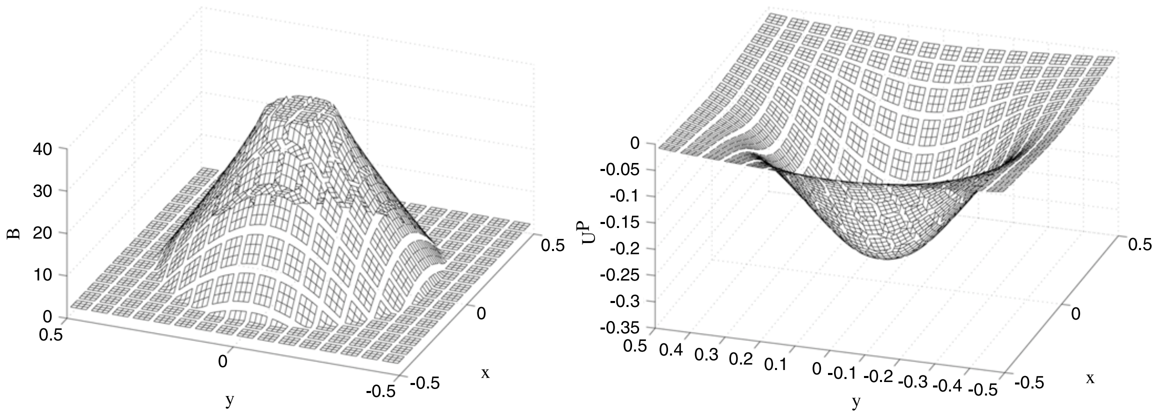

Fig. 3. The left plot shows $\tilde{B}$ for Example 2 and the right is the corresponding volume potential. Here, the domain has been discretized with 4 levels of refinement.

We define a continuous extension of $B(r)$ defined throughout D by

$$\tilde{B}(r) = \begin{cases} B(r), & \mathbf{x} \in \Omega, \\ B(0.4), & \mathbf{x} \in \Omega_0 \\ B(0.1), & \mathbf{x} \in \Omega_1. \end{cases}$$

A picture of this function throughout D is shown in left-hand plot of Fig. 3. Each contour is discretized with 16 384 points, which is large enough to ensure that the error from the evaluation of the double layer potential is negligible. We discretize the domain D with a uniform grid at varying levels of refinement and we calculate the maximum error at three different values of r, including 2 values that are 0.001 units away from the inner and outer boundaries. The results are shown in Table 2 and a picture of the volume potential is shown in right-hand plot of Fig. 3. As expected, we achieve 4th order convergence of the solution at the point that is well away from the physical boundaries. Close to Γ_0 and Γ_1, where $\tilde{B}$ has a discontinuity in its first derivative, the order of convergence drops closer to 3. This is to be expected, and in order to achieve 4th order convergence throughout the domain, we would have to extend B throughout D to be at least $\mathcal{C}^1$ everywhere.

Example 3. In this example, we solve the forced heat equation on a domain for which an analytical solution is known. The domain Ω is bounded between Γ_0, a circle of radius 0.4, and Γ_1, a circle of radius 0.1. Both Γ_0 and Γ_1 are centered at the origin. The forcing term is

$$F(\mathbf{x}) = 400\cos(20|\mathbf{x}|) + 20\frac{\sin(20|\mathbf{x}|)}{|\mathbf{x}|}.$$

Table 3
The $\mathcal{L}_\infty$ error at $t = 0.01$ using IMEX Euler (Error_1) and SBDF (Error_2) for Example 2. (The error is measured at grid points inside the domain that are sufficiently far away from either boundary.) In this simulation, $N_\Gamma = 2048$, and volume grid is obtained by refining the unit square uniformly up to level 7 (i.e. the childless nodes are of size $1/128 \times 1/128$. This spatial resolution ensures that the spatial error is negligible and the above errors are from the temporal discretization.

Δt	Error_1	Error_2
2.0×10^{-3}	1.37×10^{-2}	1.65×10^{-3}
1.0×10^{-3}	7.10×10^{-3}	4.74×10^{-4}
5.0×10^{-4}	3.61×10^{-3}	1.24×10^{-4}
2.5×10^{-4}	1.85×10^{-3}	3.23×10^{-5}

Table 4
A break down in the CPU time (in seconds) for the 100 time steps taken in the solution to Example 4. $N_D = 16P$, where P is the total number of childless boxes as determined by the interpolation error tolerance τ. The total number of points on the boundary is $N_\Gamma = 1024$. CPU_{tree} is the time taken for the generation of the initial quadtree, CPU_{lay} is the total time taken for all evaluations of the layer potential at the N_D grid points, CPU_{int} is the total time taken for the solution to all integral equations (12 GMRES iterations were required at each time step to solve the integral equation).

τ	N_D	CPU_{tree}	CPU_{lay}	CPU_{int}	$\text{CPU}_{\text{total}}$
10^{-4}	6 064	19.27	38.80	329.93	410.51
10^{-6}	13 696	56.22	57.10	330.10	491.32
10^{-8}	33 712	140.38	86.14	329.55	657.04
10^{-10}	67 840	203.54	125.96	329.80	834.83
10^{-12}	94 912	238.79	148.29	328.36	945.66

The following is the exact solution to the heat equation:

$$u(x, t) = \mathrm{e}^{-\lambda^2 t}\,[Y_0(0.1\lambda)J_0(\lambda|\mathbf{x}|) - J_0(0.1\lambda)Y_0(\lambda|\mathbf{x}|)] + \cos(20|\mathbf{x}|),$$

where J_0 is the zeroth-order Bessel function of the first kind, and Y_0 is the zeroth-order Bessel function of the second kind. We have chosen λ so that the time-dependent term vanishes on both boundaries $\lambda \approx 10.244$. The boundary conditions, then, are

$$f(\mathbf{x}) = \begin{cases} \cos(8), & \mathbf{x} \in \Gamma_0, \\ \cos(2), & \mathbf{x} \in \Gamma_1. \end{cases}$$

We calculate solutions to the heat equation using the IMEX Euler and SBDF methods, up to $t = 0.01$. The spatial resolution is high enough that the spatial error is negligible in comparison to the temporal error. The results are summarized in Table 3. As shown in this table, we achieve first and second-order convergence for the Euler and SBDF methods, respectively.

Example 4. In this example, we solve the homogeneous heat equation in an elliptical region containing an off-center elliptical hole (the same geometry as in Example 1). The initial condition is homogeneous, and the boundary conditions are

$$f(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \Gamma_0, \\ 1, & \mathbf{x} \in \Gamma_1. \end{cases}$$

The quadtree structure for the solution at the first time step is shown in Fig. 2 and a break down of the computational cost is shown in Table 4. The bulk of the computation costs comes from the generation of the quadtree, the evaluation of the layer potential at the volume grid points and the solution to the integral equation. The total computational cost is $O(N_D)$, as expected. Plots of the solution at three different times are shown in Fig. 4. At the final time, the solution is near steady state.

Example 5. In this example, we demonstrate that our methods can be applied to much more complex equations. Here, we solve the Allen–Cahn equation

$$\begin{aligned} &u_t - \epsilon \Delta u = u(1 - u^2), \quad \mathbf{x} \in \Omega,\ t > 0, \\ &u(\mathbf{x}, t) = 0, \quad \mathbf{x} \in \Gamma_0, \\ &u(\mathbf{x}, t) = 0, \quad \mathbf{x} \in \Gamma_1, \end{aligned}$$

where $\epsilon = 10^{-5}$. We initialize the solution with random values uniformly distributed on $[-1/2, 1/2]$. The nonlinear forcing term is evaluated explicitly in the time stepping and is incorporated into the right-hand side of Eq. (3). The general behavior of solutions to the Allen–Cahn equation is well known: the stable stationary solutions are $u = 1$ and $u = -1$ and the solution exhibits coarsening towards these values. The presence of physical boundaries can create more complex patterns, as seen in Figs. 5 and 6. In computing the solutions to processes involving coarsening or pattern formation behavior, it would

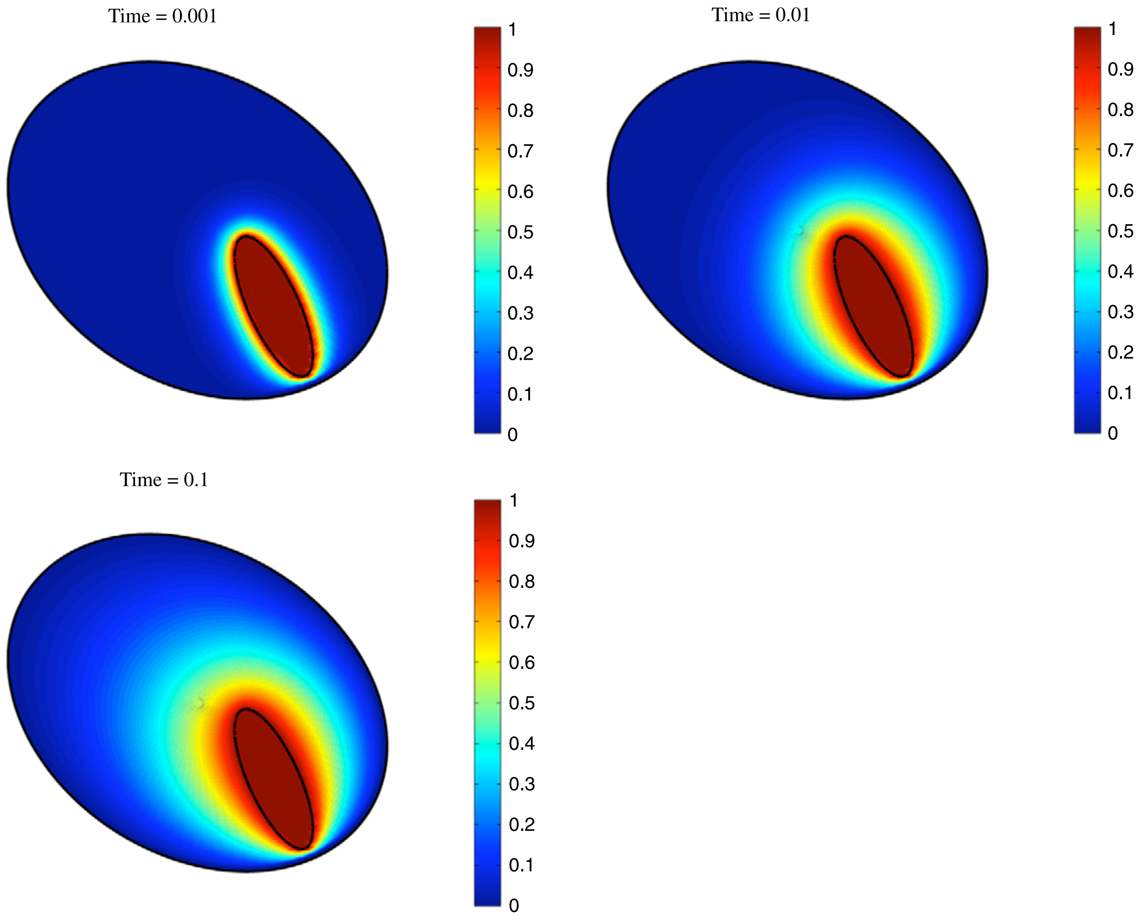

Fig. 4. The solution to Example 4 at three different times.

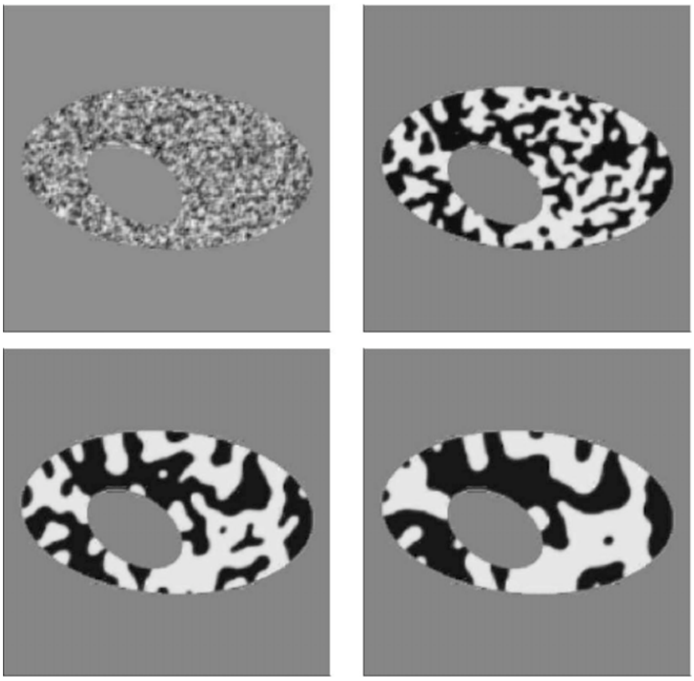

Fig. 5. The solution to Example 5. Here, $\Delta t = 1$ and IMEX Euler's method is used. The unit square is uniformly refined to level 6 (i.e. the finest boxes are of size $1/64 \times 1/64$. The upper left plot is at $t = 0$, and proceeding clockwise, the remaining plots are at $t = 13$, $t = 43$, and $t = 100$, respectively.

be desirable to adapt the quadtree on the fly, so as to ensure adequate resolution for changing solution features within the domain. This will be considered in the future.

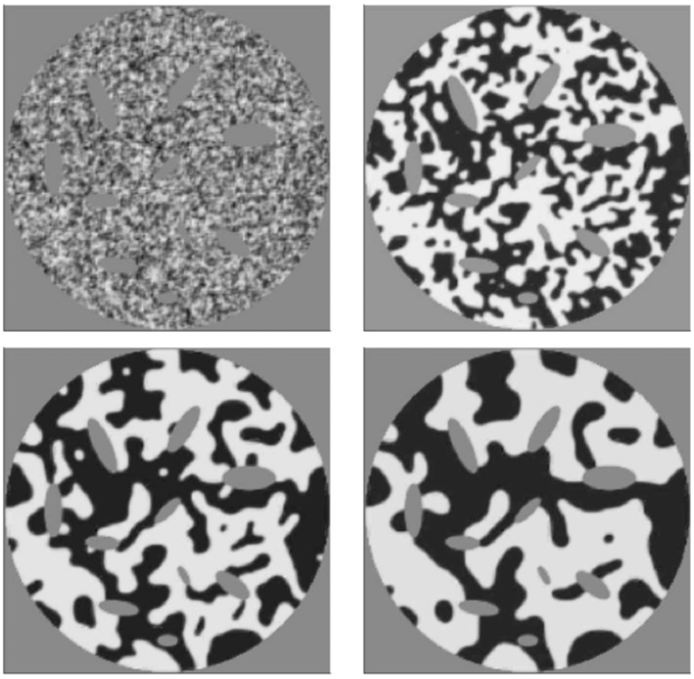

Fig. 6. The solution to Example 5 in a domain with 10 interior component curves. Here, $N_\Gamma = 2816$, $N_D = 65,536$, $\Delta t = 1$, and 100 time steps are taken. Total CPU time of approximately 19.5 min. The upper left plot is at $t = 0$, and proceeding clockwise, the remaining plots are at $t = 13$, $t = 43$, and $t = 100$, respectively.

5. Conclusions

We have presented an investigation on coupling available fast algorithms for the modified Helmholtz equation for the purposes of solving the isotropic, nonlinear heat equation. We base our approach on Rothe's method: we discretize in time first and then use integral equation methods to solve a sequence of elliptic boundary value problems. We have demonstrated the methods on a number of examples and have shown that this approach has the potential to efficiently and accurately solve the heat equation in complicated geometry. One important aim of this paper was to identify and elucidate the remaining issues that need to be resolved in order to develop this approach into a fully general solver. These issues are listed below:

- The extension of $B(\mathbf{x}, u, t)$ throughout the computational domain D for general boundary conditions $u = f(\mathbf{x})$ and forcing term $F(\mathbf{x}, u, t)$ remains an open problem. Based on our results, it may be sufficient to do a $\mathcal{C}^0$ extension only, although a $\mathcal{C}^1$ is desirable. This may be achievable by local interpolation, or by solving a suitable integral equation inside each Ω_k and outside Ω_0.
- In order to appropriately resolve solution features that appear/disappear within the domain, it may be necessary to dynamically generate the quadtree throughout the simulation. However, the cost of generating the quadtree is nontrivial and this would have to be done carefully in order to maintain efficiency.
- The integral operator of the double layer potential becomes singular at grid points in D close to Γ. Our approach here was to simply over-resolve the boundary grid. A more desirable approach would be to incorporate special quadratures, possibly using the ones discussed in [28,29].

Once these issues have been addressed and a fully general solver has been developed, we plan on using this solver to study complex reaction–diffusion systems. We also plan to develop similar integral equation methods for the incompressible Navier–Stokes equations. Future work will also include examining alternative temporal discretization methods. For instance, one promising avenue would be to combine the fast solver for the modified Helmholtz equation with a spectral-deferred correction, time-integration scheme [30,31] in order to achieve both a high degree of accuracy and unconditional stability. Future work will also include moving into three dimensions. This will require careful consideration of an appropriate quadrature and a suitable representation for surfaces. However, a fast multipole method already exists for the three-dimensional screened Coulomb potential in three dimensions [14].

Acknowledgements

The authors wish to thank Rustum Choksi and Yves Van Gennip for their help with Example 5, and Jingfang Huang for his considerable help with his volume potential solver. The first author was supported in part by the Natural Sciences and Engineering Research Council of Canada Grant RGPIN 203326.

References

[1] A. Greenbaum, L. Greengard, G.B. McFadden, Laplace's equation and the Dirichlet–Neumann map in multiply connected domains, J. Comput. Phys. 105 (1993) 267–278.
[2] F. Ethridge, L. Greengard, A new fast-multipole accelerated Poisson solver in two dimensions, SIAM J. Sci. Comput. 23 (3) (2001) 741–760.
[3] L. Greengard, M.C. Kropinski, A. Mayo, Integral equation methods for Stokes flow and isotropic elasticity in the plane, J. Comput. Phys. 125 (2) (1996) 403–414.
[4] H. Cheng, J. Huang, T. Leiterman, An adaptive fast solver for the modified Helmholtz equation in two dimensions, J. Comput. Phys. 211 (2006) 616–637.
[5] M.C. Kropinski, B. Quaife, Integral equation methods for the modified Helmholtz equation, J. Comput. Phys. 230 (2011) 425–434.
[6] Jing-Rebecca Li, Leslie Greengard, On the numerical solution of the heat equation I: fast solvers in free space, J. Comput. Phys. 226 (2007) 1891–1901.
[7] Jing-Rebecca Li, Leslie Greengard, High order accurate methods for the evaluation of layer heat potentials, SIAM J. Sci. Comput. 31 (5) (2009) 3847–3860.
[8] Johannes Tausch, A fast method for solving the heat equation by layer potentials, J. Comput. Phys. 224 (2007) 956–969.
[9] Shravan K. Verrapaneni, George Biros, A high-order solver for the heat equation in 1D domains with moving boundaries, SIAM J. Sci. Stat. Comput. 29 (6) (2007) 2581–2606.
[10] Shravan K. Verrapaneni, George Biros, The Chebyshev fast Gauss and nonuniform fast Fourier transforms and their application to the evaluation of distributed heat potentials, J. Comput. Phys. 227 (2008) 7768–7790.
[11] N. Nishimura, Fast multipole accelerated boundary integral equation methods, Appl. Mech. Rev. 55 (4) (2002) 299–324.
[12] Roman Chapko, Rainer Kress, Rothe's method for the heat equation and boundary integral equations, J. Integral Equations Appl. 9 (1) (1997) 47–69.
[13] Roman Chapko, On the combination of Rothe's method and boundary integral equations for the nonstationary Stokes equations, J. Integral Equations Appl. 13 (2) (2001).
[14] L. Greengard, J. Huang, A new version of the fast multipole method for screened Coulomb interactions in three dimensions, J. Comput. Phys. 180 (2002) 642–658.
[15] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the Laplace equation in three dimensions, Acta Numer. 6 (1997) 229–269.
[16] George Biros, Lexing Ying, Denis Zorin, A fast solver for the Stokes equations with distributed forces in complex geometries, J. Comput. Phys. (2003) 317–348.
[17] George Biros, Lexing Ying, Denis Zorin, An embedded boundary integral solver for the unsteady incompressible Navier–Stokes equations, Preprint, 2002.
[18] Anita Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numer. Anal. 21 (2) (1984) 285–289.
[19] Anita Mayo, Rapid, fourth order accurate solution of the steady Navier–Stokes equations on general regions, Dyn. Contin. Discrete Impuls. Syst. Ser. B Appl. Algorithms 12 (2005) 59–72.
[20] Anita Mayo, Anne Greenbaum, Fourth order accurate evaluation of integrals in potential theory on exterior 3D regions, J. Comput. Phys. 220 (2007) 900–914.
[21] L. Greengard, M.C. Kropinski, An integral equation approach to the incompressible Navier–Stokes equations in two-dimensions, SIAM J. Sci. Comput. 20 (1998) 318–336.
[22] U.M. Ascher, S.J. Ruuth, B.M. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (1995) 797–823.
[23] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions, Dover, New York, NY, 1964.
[24] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325–348.
[25] J. Carrier, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, SIAM J. Sci. Stat. Comput. 9 (1988) 669–686.
[26] Bradley Alpert, Hybrid Gauss-trapezoidal quadrature rules, SIAM J. Sci. Comput. 20 (1999) 1551–1584.
[27] Y. Saad, M.H. Schultz, GMRES: a generalized minimum residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[28] J.Thomas Beale, Ming-Chih Lai, A method for computing nearly singular integrals, SIAM J. Numer. Anal. 38 (6) (2001) 1902–1925.
[29] J. Strain, Locally-corrected multidimensional quadrature rules for singular functions, SIAM J. Sci. Comput. 6 (4) (1995) 992–1017.
[30] S. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT 40 (2) (2000) 241–266.
[31] J. Jia, J. Huang, Krylov deferred correction accelerated method of lines transpose for parabolic systems, J. Comput. Phys. 227 (3) (2008) 1739–1753.