November 2007

A High-Order Solver for the Heat Equation in 1d Domains with Moving Boundaries

Shravan K. Veerapaneni
University of Pennsylvania, shravan@seas.upenn.edu

George Biros
University of Pennsylvania, biros@seas.upenn.edu

Recommended Citation

Veerapaneni, Shravan K. and Biros, George, "A High-Order Solver for the Heat Equation in 1d Domains with Moving Boundaries" (2007). *Departmental Papers (MEAM)*. 132.
http://repository.upenn.edu/meam_papers/132

© SIAM 2007. Reprinted from *SIAM Journal on Scientific Computing*, Volume 29, Issue 6, November 2007, pages 2581-2606.
Publisher URL: http://dx.doi.org/10.1137/060677896

A High-Order Solver for the Heat Equation in 1d Domains with Moving Boundaries

Abstract

We describe a fast high-order accurate method for the solution of the heat equation in domains with moving Dirichlet or Neumann boundaries and distributed forces. We assume that the motion of the boundary is prescribed. Our method extends the work of Greengard and Strain [*Comm. Pure Appl. Math.*, XLIII (1990), pp. 949–963]. Our scheme is based on a time-space Chebyshev pseudo-spectral collocation discretization, which is combined with a recursive product quadrature rule to accurately and efficiently approximate convolutions with Green's function for the heat equation. We present numerical results that exhibit up to eighth-order convergence rates. Assuming N time steps and M spatial discretization points, the evaluation of the solution of the heat equation at the same number of points in space-time requires $O(NM \log M)$ work. Thus, our scheme can be characterized as "fast"; that is, it is work-optimal up to a logarithmic factor.

Keywords

integral equations, spectral methods, Chebyshev polynomials, moving boundaries, heat equation, quadratures, Nyström's method, collocation methods, potential theory

Comments

© SIAM 2007. Reprinted from *SIAM Journal on Scientific Computing*, Volume 29, Issue 6, November 2007, pages 2581-2606.
Publisher URL: http://dx.doi.org/10.1137/060677896

SIAM J. SCI. COMPUT.
Vol. 29, No. 6, pp. 2581–2606

A HIGH-ORDER SOLVER FOR THE HEAT EQUATION IN 1D DOMAINS WITH MOVING BOUNDARIES[*]

SHRAVAN K. VEERAPANENI[†] AND GEORGE BIROS[‡]

Abstract. We describe a fast high-order accurate method for the solution of the heat equation in domains with moving Dirichlet or Neumann boundaries and distributed forces. We assume that the motion of the boundary is prescribed. Our method extends the work of Greengard and Strain [*Comm. Pure Appl. Math.*, XLIII (1990), pp. 949–963]. Our scheme is based on a time-space Chebyshev pseudo-spectral collocation discretization, which is combined with a recursive product quadrature rule to accurately and efficiently approximate convolutions with Green's function for the heat equation. We present numerical results that exhibit up to eighth-order convergence rates. Assuming N time steps and M spatial discretization points, the evaluation of the solution of the heat equation at the same number of points in space-time requires $\mathcal{O}(NM \log M)$ work. Thus, our scheme can be characterized as "fast"; that is, it is work-optimal up to a logarithmic factor.

Key words. integral equations, spectral methods, Chebyshev polynomials, moving boundaries, heat equation, quadratures, Nyström's method, collocation methods, potential theory

AMS subject classifications. 35K05, 31A10, 65D30, 65N35, 65N38

DOI. 10.1137/060677896

1. Introduction.

We present a fast and high-order method for the solution of the one-dimensional (1D) heat equation in domains with moving boundaries. We assume that the boundary motion is prescribed. Given smooth functions f, g, and w we seek to compute $u(x,t)$ such that

$$\begin{gathered}\frac{\partial u}{\partial t} = \triangle u(x,t) + f(x,t) \quad \text{in} \quad \omega(t), \qquad t > 0,\\ u(x,0) = w(x) \quad \text{in} \quad \omega(0), \quad u(x,t) = g(x,t) \quad \text{on} \quad \gamma(t).\end{gathered} \tag{1.1}$$

All of our algorithmic choices can be extended to 2D and 3D. The details, however, become quite involved, and additional algorithmic components are necessary. For clarity and due to space limitations, we present the core ideas of our method for the one-dimensional case. We will report the extensions to higher dimensions in a future article.

Formulation. The proposed algorithm is based on potential theory [13]. Without loss of generality, let $\omega(t)$ be contained in the unit box $\Omega = [0,1]$. Let Γ denote the boundary of the unit box. By linearity, we decompose the problem (1.1) into the initial condition component u_i, the distributed force component u_f, and the boundary contribution u_b:

$$\frac{\partial u_i}{\partial t} = \triangle u_i \quad \text{in} \quad \Omega, \quad u_i(x,0) = w(x), \quad u_i(\Gamma,t) = 0, \quad t > 0, \tag{1.2}$$

[*] Received by the editors December 17, 2006; accepted for publication (in revised form) June 20, 2007; published electronically October 24, 2007. This work is partially supported by the U.S. Department of Energy under grant DE-FG02-04ER25646 and by the U.S. National Science Foundation grants CCF-0427985 and DMS-0612578.

http://www.siam.org/journals/sisc/29-6/67789.html

[†] Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104 (shravan@seas.upenn.edu).

[‡] Department of Mechanical Engineering and Applied Mechanics and Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 (biros@seas.upenn.edu.)

$$\frac{\partial u_f}{\partial t} = \triangle u_f + f \quad \text{in} \quad \Omega, \quad u_f(x,0) = 0, \quad u_f(\Gamma,t) = 0, \quad t > 0, \tag{1.3}$$

$$\frac{\partial u_b}{\partial t} = \triangle u_b \quad \text{in} \quad \omega(t), \quad u_b(x,0) = 0, \quad u_b(\gamma(t),t) = g - u_i - u_f, \quad t > 0. \tag{1.4}$$

We solve (1.2), (1.3), and (1.4) using an integral equation formulation; we compute u_i and u_f by

$$\begin{aligned} u_i(x,t) &= \mathcal{V}_0[w](x,t) = \int_\Omega G(x,t;y)w(y)dy, \\ u_f(x,t) &= \mathcal{V}[f](x,t) = \int_0^t \int_\Omega G(x,t;y,\tau)f(y,\tau)dy\,d\tau, \end{aligned} \tag{1.5}$$

where $G(x,t;y,\tau)$ is Green's function for the unit box Ω with periodic boundary conditions.[1] The solution of (1.4) is obtained by the solution of a double layer indirect integral equation formulation, which is given by

$$u_b(x,t) = \mathcal{D}[\phi](x,t) = \int_0^t \int_{\gamma(\tau)} \frac{\partial G(x,t;y,\tau)}{\partial n(y,\tau)}\phi(y,\tau)ds(y)d\tau, \; x \in \omega \times (0,T]. \tag{1.6}$$

Here n is the outward normal to γ. The double layer potential denoted by $\mathcal{D}[\phi]$ is the solution for problem (1.4) provided the boundary density ϕ solves the following Fredholm second-kind integral equation for $t > 0$:

$$-\frac{1}{2}\phi(x,t) + \mathcal{D}[\phi](x,t) = u_b(x,t) \quad \forall x \in \gamma(t). \tag{1.7}$$

Computational complexity. The solution of (1.1) through potential theory requires evaluation of double layer and volume potentials. A direct evaluation of $\mathcal{V}[f]$ using $M \times N$ quadrature points at $M \times N$ locations in space and time requires $\mathcal{O}(M^2N^2)$ work. The history dependence of this convolution can be overcome by using the fast algorithm of Greengard and Strain [6]. The key idea is to use two equivalent expansions for the kernel $G(x,t)$: one that converges fast at distant times and the other that converges fast locally. The volume potential $\mathcal{V}$ is split into a local or near part $\mathcal{V}_L$ and a far part $\mathcal{V}_F$. For the evaluation of the local part, a method-of-images expansion of G is used; for the far part, a Fourier expansion is used. A recurrence relation to update the Fourier coefficients in the far part eliminates the need to integrate over the entire history. As pointed out in [22], discrete sums of the form $\sum_{n=1}^{p} C_n \sin(n\pi x)$ and $\sum_{k=1}^{M} \sin(n\pi y_k) f_k$ that arise in the computation of $\mathcal{V}_F$ can be computed in optimal time by using the nonuniform FFT (NUFFT) [4]. The local part can be computed optimally using the fast Gauss transform [7]. Using these methods the overall complexity of computing the volume potential can be reduced to $\mathcal{O}(MN \log M)$; the overall scheme is second-order accurate.

Synopsis of the new method. Here we extend the Greengard and Strain algorithm to a high-order accurate scheme. We develop special product integration rules to compute the local and far parts of the heat potentials. The basic idea is to approximate the boundary density using Chebyshev polynomials and then compute the resulting moments exactly. To achieve optimal complexity, we propose a scheme

[1] Here we describe the decomposition into three subproblems for the case of a Dirichlet boundary condition. The extension to the Neumann case is analogous.

that makes use of fast summation methods [4, 6, 7]. We then build a solver for (1.1) based on a fast and high-order accurate evaluation of volume and layer potentials.

Following Greengard and Strain, this is the outline of our method:

- We use a double layer integral equation formulation for (1.1), which results in a well-conditioned linear system;
- we discretize using a pseudo-spectral collocation method using Chebyshev polynomials;
- we develop high-order quadratures for the heat (and other sharply peaked) kernels using recurrences;
- we derive a high-order scheme for the heat equation in domains with moving boundaries (with prescribed motion) and distributed forces.

The main contributions of this work are the high-order time-marching scheme and the extension to problems with moving boundaries and distributed sources.

Related work. Much of the research on solving the heat equation on moving boundaries has been concentrated on Stefan's problem for which the motion of the boundary is unknown. In this paper, we consider only problems in which the motion of the boundary is *prescribed.*[2] Fast algorithms introduced recently dramatically improved the computational complexity of solving integral equation formulations for parabolic PDEs. The fast Gauss transform [7] can be used to accelerate the solution of the free-space initial value problem for the heat equation. The Greengard and Strain algorithm [6] can be used for the efficient evaluation of single and double layer heat potentials in bounded domains. Fast algorithms for unbounded domains are discussed in [5, 14]. The work of [2, 20] is one of the first attempts to design fast methods for the heat equation. A direct formulation was used in [10, 28] for the 1D Stefan problem. In the case of prescribed Dirichlet data, a direct integral equation formulation leads to a Volterra system of equations that is ill-conditioned.[3] Ill-conditioning can be avoided by using an indirect double layer formulation, which we describe in detail in section 5. Recently, epitaxial step flow growth in 1D was simulated in [9] using layer potentials. Despite these remarkable advances, however, none of the existing methods achieves both high accuracy and optimal complexity.

Indeed, fast summation algorithms and high-order schemes are necessary to build work-efficient solvers for (1.1). The time integrals in all variants of the heat potentials have kernels that are sharply peaked. A generic quadrature rule can be used, but it will not achieve its order of convergence for reasonable discretization sizes. Uniform second-order convergence in computing the single layer was achieved in [6, 20] using asymptotic expansions. To obtain high-order convergence through asymptotic expansions, one needs high-order derivatives of the boundary density ϕ, something that can be computationally expensive. In [22, 23] a uniform second-order convergence was achieved by using a piecewise linear approximation for the boundary density and computing the convolution with the heat kernel exactly.

An alternative is the design of special quadrature rules: Given $I(\alpha) = \int_{-1}^{1} K(\alpha, \theta)\,\phi(\theta) d\theta$, with K being a singular, sharply peaked, or oscillatory kernel, the goal is to design high-order accurate integration schemes. There exists a substantial body

[2] There is a large (and significant) body of work regarding local stencil-based discretizations of (1.1); we do not attempt to review this literature. We believe that, for certain problems, there are significant advantages in using integral equations. Since this work is in 1D, we postpone the discussion to a future paper.

[3] In 1D ill-conditioning has little impact in practical computations. The argument is important for problems in higher dimensions.

of work on quadrature rules for such problems. The main challenge is to ensure optimal algorithmic complexity and compatibility with fast summation methods. In Kapur and Rokhlin [12] and Rokhlin [19], corrections to the trapezoidal rule have been suggested for integrating singular functions. In Ma and Wandzura [11] and in Yarvin and Rokhlin [26], numerical tools to obtain generalized Gaussian quadratures have been devised. In Alpert [1] hybrid Gauss-trapezoidal rules for regular functions and singular kernels were developed. These approaches, however, are not directly applicable to the heat kernel due to the spatial and temporal components of the kernel. Although it may be the case that they can be extended to the heat kernel, we have opted for an alternative approach based on recurrence relationships.

Such recurrences have been established by Piessens and Branders [15] for a wide variety of kernels. In Hasegawa and Torii [8], Cauchy principal value integrals were evaluated using quadrature rules combined with Chebyshev-polynomial approximations. Here we concentrate on heat potentials, but this strategy can be applied to solving other PDEs via integral equations. For example, in Piessens [16] integral equations like the Abel integral equation were solved. The main difference between this method and quadrature rules is that we discretize the density but not the integral operator. Given a density $\phi(\theta)$ and a kernel $K(\alpha,\theta)$ we write:

$$\phi(\theta) = \sum_{k=0}^{q-1} \phi(k) T_k(\theta) \ \Rightarrow I(\alpha) = \sum_{k=0}^{q} \phi(k) I_k(\alpha), \quad \text{where } I_k(\alpha) = \int_{-1}^{1} K(\alpha,\theta) T_k(\theta) d\theta, \tag{1.8}$$

based on the Chebyshev-polynomial expansion of ϕ. The integrals $I_k(\alpha)$ are computed to machine accuracy using recurrence relations (section 2). Using the FFT, the Chebyshev coefficients $\{\phi(n)\}_{n=0}^{q}$ can be computed in $\mathcal{O}(q \log q)$ work, and all $I_k(\alpha)$ can be computed using recurrence relations in $\mathcal{O}(q)$ work. Hence the overall complexity of computing the integral $I(\alpha)$ is $\mathcal{O}(q \log q)$. Using this method we compute heat potentials by approximating the potential density function using piecewise $(q-1)$th-order Chebyshev polynomials and thus obtain a qth-order accurate method.

Contents. In section 2, we discuss algorithms for the fast evaluation of heat potentials. In section 3, we derive recurrence relations for computing the moments $I_k(\alpha)$ defined in (1.8) for heat kernels. The details of implementing the boundary integral solver for (1.1) with static boundaries are given in section 4. The case of boundaries with prescribed motion is discussed in section 5. We report numerical results that verify the efficiency and accuracy of our scheme in section 6.

2. Fast summation.

The direct evaluation of u_b by (1.6) at M spatial locations and N time levels requires $\mathcal{O}(N^2 M)$ work in 1D. The Greengard–Strain algorithm reduces this work to $\mathcal{O}(NM)$. The key idea is to use two different representations of Green's function (for a square box with periodic boundary conditions) in different time intervals: one for the history part that represents the influence of the sources located temporally away from the current evaluation time and one for the local part that represents the influence of the sources located at times closer to the current evaluation time.

Green's function for the unit box is the solution to the adjoint problem with homogeneous boundary conditions:

$$\frac{\partial G}{\partial \tau} + \triangle_y G = \delta(x-y, t-\tau) \quad \text{in } \Omega \quad \text{such that (s.t.) } G(\Gamma, t) = 0. \tag{2.1}$$

The solution of this problem by Fourier series and by Kelvin's method of images give us two equivalent expansions:

$$G(x,y;t,\tau) = \sum_{n=1}^{\infty} e^{-n^2\pi^2(t-\tau)} 2\sin(n\pi x)\sin(n\pi y) \tag{2.2}$$

$$= \sum_{n=-\infty}^{\infty} \left(\frac{e^{-(x-y-2n)^2/4(t-\tau)}}{\sqrt{4\pi(t-\tau)}} - \frac{e^{-(x+y-2n)^2/4(t-\tau)}}{\sqrt{4\pi(t-\tau)}} \right). \tag{2.3}$$

These expansions converge with different rates in different time intervals. The Fourier representation converges faster at distant times (from the current evaluation time), whereas the method of images representation converges faster at closer times. This motivates splitting of heat potentials into two parts:

$$\mathcal{V}[f] = \mathcal{V}_F[f] + \mathcal{V}_L[f] = \int_0^{t-\delta}\int_\Omega Gf + \int_{t-\delta}^{t}\int_\Omega Gf. \tag{2.4}$$

Here δ is a parameter that determines the error in truncating both of the series in (2.3). The operator $\mathcal{V}_F$ will be termed as the far part, and $\mathcal{V}_L$ will be termed as the local part. The exact same decomposition is also valid for single and double layer potentials, with f being replaced by the boundary density ϕ.

2.1. Truncating the series. In 1D the boundary γ of the domain ω corresponds to two points b_1 and b_2. Then the far part of the double layer potential simplifies to

$$\mathcal{D}_F[\phi] = \sum_{n=1}^{\infty}\sum_{k=1}^{2}(-1)^k\, 2n\pi\sin(n\pi x)\int_0^{t-\delta}\cos(n\pi b_k)e^{-n^2\pi^2(t-\tau)}\phi(b_k,\tau)d\tau. \tag{2.5}$$

The error in truncating the series (2.5) after p terms can be bounded from above as follows:

$$\begin{aligned}
\mathcal{E}_F(p) &= \left|\mathcal{D}_F[\phi] - \sum_{n=1}^{p}\sum_{k=1}^{2}(-1)^k\, 2n\pi\sin(n\pi x)\int_0^{t-\delta}\cos(n\pi b_k)e^{-n^2\pi^2(t-\tau)}\phi(b_k,\tau)d\tau\right| \\
&= \left|\sum_{n=p+1}^{\infty}\sum_{k=1}^{2}(-1)^k\, 2n\pi\sin(n\pi x)\int_0^{t-\delta}\cos(n\pi b_k)e^{-n^2\pi^2(t-\tau)}\phi(b_k,\tau)d\tau\right| \\
&\le 4|\phi|_\infty \sum_{n=p+1}^{\infty}\left|\int_0^{t-\delta} n\pi e^{-n^2\pi^2(t-\tau)}d\tau\right| = 4|\phi|_\infty \sum_{n=p+1}^{\infty}\left|\frac{e^{-n^2\pi^2\delta} - e^{-n^2\pi^2 t}}{n\pi}\right| \\
&\le \frac{4|\phi|_\infty}{\pi(p+1)}\sum_{n=p+1}^{\infty} e^{-n^2\pi^2\delta} \le \frac{4|\phi|_\infty}{\pi(p+1)} e^{-(p+1)^2\pi^2\delta}\int_0^\infty e^{-\pi^2x^2\delta}dx \\
&= \frac{2|\phi|_\infty}{\pi\sqrt{\pi\delta}(p+1)} e^{-(p+1)^2\pi^2\delta}.
\end{aligned}$$

The local part expansion represents the influence of Gaussian pulses initiated at $t=0$ and located at $\{x-2n, 2n-x\}_{n=-\infty}^{\infty}$. The value of a Gaussian pulse evaluated at a distance $2r\sqrt{t}$ from its center would be of the order $\frac{e^{-r^2}}{\sqrt{t}}$. Hence, if the boundary γ is a distance d away from the boundary of the unit box Γ, then the error in approximating Green's function by just the first term ($n=0$) is of the order $\frac{e^{-d^2/\delta}}{\sqrt{\delta}}$.

Then the local part of the double layer is

$$\mathcal{D}_L[\phi] = \int_{t-\delta}^{t} \int_{\gamma(\tau)} \frac{x-y}{4[\pi(t-\tau)]^{3/2}} e^{-\frac{(x-y)^2}{4(t-\tau)}} \phi(y,\tau) d\tau + \mathcal{O}\left(\frac{e^{-d^2/\delta}}{\delta^{3/2}}\right). \tag{2.6}$$

2.2. Fast evaluation of the far part. A direct evaluation of $\mathcal{D}_F[\phi]$ at $M \times N$ points in space and time requires $\mathcal{O}(N^2Mp)$ computations. Optimal complexity of $\mathcal{O}((M+p)N)$ can be achieved by using the fast algorithms in [4, 6]. We briefly summarize these algorithms here. Let

$$\hat{\phi}_n(\tau) = \int_{\gamma(\tau)} \frac{\partial \sin(n\pi y)}{\partial n(y)} \phi(y,\tau) ds(y) = n\pi \left(\cos(n\pi b_2)\phi(b_2,\tau) - \cos(n\pi b_1)\phi(b_1,\tau)\right), \tag{2.7}$$

$$C_n(t,\delta) = 2\int_0^{t-\delta} e^{-n^2\pi^2(t-\tau)} \hat{\phi}_n(\tau)\, d\tau. \tag{2.8}$$

Then

$$\mathcal{D}_F[\phi](x,t) = \sum_{n=1}^{p} C_n(t,\delta) \sin(n\pi x). \tag{2.9}$$

The Fourier coefficients can be updated with constant work per time step using the following recurrence:

$$C_n(t+\triangle t,\delta) = e^{-n^2\pi^2\triangle t} C_n(t,\delta) + 2\int_{t-\delta}^{t+\triangle t-\delta} e^{-n^2\pi^2(t+\triangle t-\tau)} \hat{\phi}_n(\tau) d\tau. \tag{2.10}$$

This recurrence is the key step that eliminates the history dependence of the far part. Once we have computed the C_n coefficients, we can evaluate (2.9) using the inverse NUFFT.

2.3. Optimal splitting of the heat potentials. In this section we briefly comment on the selection[4] of the parameters associated with the splitting of the heat potentials to a far and local part. If we retain p terms in the far part expansion and if $\delta = l\triangle t$, the overall complexity of computing $\mathcal{D}[\phi]$ is $W = \mathcal{O}(N(M+p)\log p + MNl)$. For each target point, the local part computation involves integrating over l time steps, hence the local part evaluation contributes $\mathcal{O}(MNl)$ to W. For fixed constants (c_1, c_2), we set $p = c_1\sqrt{M}$ and $l = \left\lceil \frac{c_2 \log M}{N\triangle t} \right\rceil$ to obtain $W = \mathcal{O}(MN\log M)$, which is off from being optimal only by a logarithmic factor. For this choice of parameters (assuming exact evaluation of the far and near integrals) the error estimates due to the splitting are given by

$$\begin{aligned} \mathcal{E}_F(M,N) &= \mathcal{O}\left(\left(\frac{N}{M\log M}\right)^{1/2} e^{-\frac{\pi^2 c_1^2 c_2 M \log M}{N}}\right), \\ \mathcal{E}_L(M,N) &= \mathcal{O}\left(\left(\frac{N}{\log M}\right)^{3/2} e^{-\frac{d^2 N}{c_2 \log M}}\right). \end{aligned} \tag{2.11}$$

[4] The discussion here is important only in the asymptotic limit, i.e., when the size of the time step is small. But for significantly large time-step discretizations, we can just fix the parameters p, δ such that $\delta < \triangle t$ and such that the truncation errors are within the required accuracy. This would suffice to obtain optimal complexity.

Therefore, both errors are exponentially converging as M and N are increased. For instance, if $N = M = 256$, by choosing $c_1 = 2.5, c_2 = 0.1$, both $\mathcal{E}_F, \mathcal{E}_L$ are less than 10^{-15}, and for this choice we get $p = 40, \delta \approx 0.0022$.

Next we describe our construction of recursive product integration rules for the local and far parts.

3. Quadratures.

In this section, we compute the moments $I_k(\alpha)$ as defined by (1.8) for the kernels: $e^{-\alpha\theta}$, $\frac{e^{-\frac{\alpha}{\theta}}}{\sqrt{\theta}}$, $\frac{e^{-\frac{\alpha}{\theta}}}{\sqrt{\theta^3}}$ and then show how they can be used to compute the time integrals in $\mathcal{D}_L$ and $\mathcal{D}_F$. First, we review some basic properties of Chebyshev polynomials. For $x \in [-1, 1]$, the closed-form expression for the nth-order Chebyshev polynomial $T_n(x)$ is given by:

$$T_n(x) = \cos(n \cos^{-1} x). \tag{3.1}$$

The Chebyshev polynomials $T_n(x)$ are orthogonal w.r.t. the inner product defined by:

$$\langle f(x), g(x) \rangle = \int_{-1}^{1} \frac{f(x)g(x)}{\sqrt{1-x^2}} dx. \tag{3.2}$$

The nth-order Chebyshev coefficient of $f(x)$ denoted by $f(n)$ can be computed by taking the inner product with $T_n(x)$:

$$f(n) = \int_{-1}^{1} \frac{f(x)T_n(x)}{\sqrt{1-x^2}}\, dx = \frac{c_n}{\pi} \int_0^{\pi} f(\cos\theta) \cos n\theta\, d\theta = \frac{c_n}{2\pi} \int_0^{2\pi} f(\cos\theta) \cos n\theta\, d\theta, \tag{3.3}$$

$$\text{where} \quad c_0 = 1 \quad \text{and} \quad c_k = 2 \quad \text{for} \quad k \geq 1. \tag{3.4}$$

Now as the function $f(\cos\theta)$ is 2π-periodic, we can use FFT or the fast cosine transform to compute $\{f(n)\}_{n=0}^{q-1}$ in $\mathcal{O}(q \log q)$ time. Also, we will use the following properties of Chebyshev polynomials:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad \text{and} \quad \int T_n = \frac{1}{2}\left(\frac{T_{n+1}}{n+1} - \frac{T_{n-1}}{n-1}\right) + C.$$

For more details on the properties of Chebyshev polynomials, we refer to [18]; a review on spectral methods for solving PDEs using the Chebyshev polynomials can be found in [24].

3.1. Recurrence relations.

The idea of computing the integrals through the approximation of the integrand by Chebyshev polynomials and then computing the moments of smooth functions by recurrences was first used by Clenshaw and Curtis [3]. Consider integrating a smooth function $f(x)$ in the interval $(0, 1)$:

$$I = \int_0^1 f(x)dx = \sum_{n=0}^{q-1} f(n) \int_0^1 T_n(2x-1)dx = \sum_{n=0}^{q-1} f(n) I_n,$$

$$\text{where} \quad I_n = \int_0^1 T_n(2x-1)dx = \left[\frac{T_{n+1}(2x-1)}{n+1} - \frac{T_{n-1}(2x-1)}{n-1}\right]_0^1.$$

One can easily show that evaluation of this will yield:

$$I_n = \begin{cases} 0 & \text{if } n \text{ is odd,} \\ -\frac{1}{n^2-1} & \text{otherwise.} \end{cases} \tag{3.5}$$

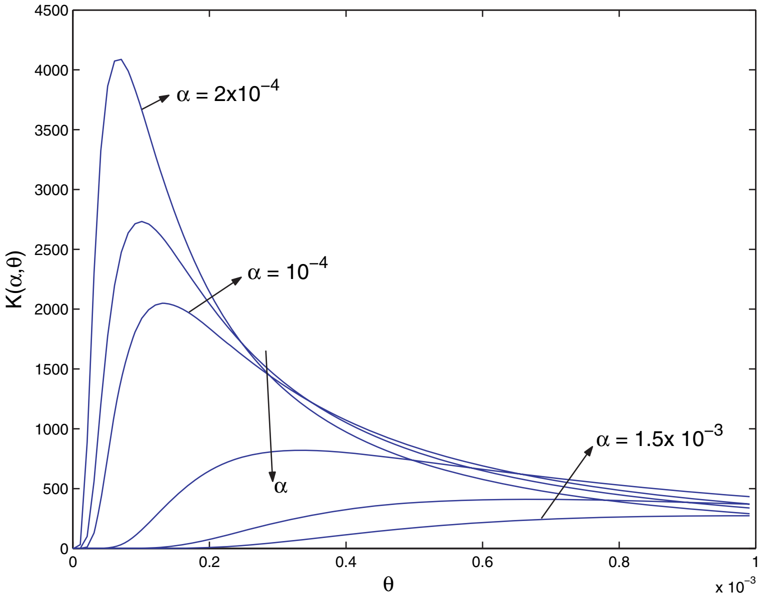


FIG. 3.1. *Kernel* $K(\alpha,\theta) = \frac{\sqrt{\alpha}}{\sqrt{\theta^3}}e^{-\frac{\alpha}{\theta}}$ *for different values of* α. *The kernel becomes sharply peaked as* α *decreases, so an accurate evaluation of* (3.6) *by generic quadratures requires an excessive number of points.*

Similarly we can derive the recurrences for computing the moments $I_k(\alpha) = \int_0^1 K(\alpha,\theta)$ $T_k(\theta)$ for a variety of kernels K. In Appendix A we give recurrences for the singular kernels $\theta^{-\gamma}$ for $\gamma \in (-\infty, 1)$ and $\log\theta$. As the approximation of $f(\theta)$ by Chebyshev polynomials is superalgebraically convergent, the error in the computation of the integral $I(\alpha)$ is approaching zero exponentially (numerical examples can be found in section 6). Now we discuss the computation of integral operators that will arise while evaluating the heat potentials. Let us consider

$$I(\alpha) = \int_0^1 \frac{\sqrt{\alpha}}{\sqrt{\theta^3}}e^{-\frac{\alpha}{\theta}} f(\theta)\,d\theta, \quad \text{and} \quad J(\alpha) = \int_0^1 \frac{1}{\sqrt{\theta}}e^{-\frac{\alpha}{\theta}} f(\theta)\,d\theta. \tag{3.6}$$

In the limit any generic quadrature rule will converge to the correct value of I and J with its optimal convergence rate. The quality of the approximation, however, will be nonuniform in the value of α. In fact, the constants deteriorate dramatically with small α. As a result, I and J become very difficult to compute accurately; a generic rule will require thousands of points—even for a few digits of accuracy. This is because the kernel becomes sharply peaked for smaller values of α; see Figure 3.1.

This poses a challenge for Nyström-type methods as one would need to develop specialized quadratures for different values of α. Instead, we are using a Chebyshev approximation of f and derive recurrence relations to compute the moments I_n. In this way, given α, we can compute all I_n with $\mathcal{O}(q)$ work.

First we compute the base condition for the recurrence:

$$I_0(\alpha) = \int_0^1 \frac{\sqrt{\alpha}}{\sqrt{\theta^3}}e^{-\frac{\alpha}{\theta}}\,d\theta = 2\int_{\sqrt{\alpha}}^{\infty} e^{-z^2}dz = \sqrt{\pi}\operatorname{erfc}(\sqrt{\alpha}).$$

Then we compute higher moments by

$$I_n(\alpha) = \int_0^1 \frac{\sqrt{\alpha}}{\sqrt{\theta^3}} e^{-\frac{\alpha}{\theta}} T_n \, d\theta = \int_0^1 \frac{\sqrt{\alpha}}{\sqrt{\theta^3}} e^{-\frac{\alpha}{\theta}} \left[2(\lambda\theta + \eta)T_{n-1} - T_{n-2} \right] d\theta,$$

$$\Rightarrow I_n(\alpha) = 2\lambda\sqrt{\alpha} J_{n-1} + \eta I_{n-1} - I_{n-2}, \tag{3.7}$$

$$\text{where} \quad J_{n-1}(\alpha) = \int_0^1 \frac{1}{\sqrt{\theta}} e^{-\frac{\alpha}{\theta}} T_{n-1} \, d\theta = 2\lambda \int_0^1 \sqrt{\theta} e^{-\frac{\alpha}{\theta}} T_{n-2} \, d\theta + 2\eta J_{n-2} - J_{n-3}.$$

Integrating by parts, we get:

$$2\lambda \int_0^1 \sqrt{\theta} e^{-\frac{\alpha}{\theta}} T_{n-2} \, d\theta = C_n - \int_0^1 \left(\frac{1}{2\sqrt{\theta}} + \frac{\alpha}{\sqrt{\theta^3}} \right) e^{-\frac{\alpha}{\theta}} \left(\frac{T_{n-1}}{n-1} - \frac{T_{n-3}}{n-3} \right) d\theta.$$

Substituting this back in the expression for J_{n-1}, we obtain the following recurrence:

$$J_{n-1}(\alpha) = \frac{2n-2}{2n-1} \left(C_n - \frac{2n-7}{2n-6} J_{n-3} + 2\eta J_{n-2} - \sqrt{\alpha} \left(\frac{I_{n-1}}{n-1} - \frac{I_{n-3}}{n-3} \right) \right), \tag{3.8}$$

$$\text{where} \quad C_n = e^{-\alpha} \left(\frac{T_{n-1}(\lambda+\eta)}{n-1} - \frac{T_{n-3}(\lambda+\eta)}{n-3} \right). \tag{3.9}$$

Using (3.8) we can compute $J_{n-1}(\alpha)$, from which $I_n(\alpha)$ can be computed using (3.7). Now we establish the recurrences for the kernel $e^{-\alpha\theta}$:

$$\begin{aligned} I_n &= \int_0^1 e^{-\alpha x} T_n(\lambda x + \eta) \, dx, \\ &= \left[\frac{e^{-\alpha x}}{2\lambda} \left(\frac{T_{n+1}}{n+1} - \frac{T_{n-1}}{n-1} \right) \right]_0^1 + \int_0^1 \left[\frac{e^{-\alpha x}}{2\lambda} \left(\frac{T_{n+1}}{n+1} - \frac{T_{n-1}}{n-1} \right) \right], \end{aligned}$$

$$\begin{aligned} &\Rightarrow I_{n+1} = c_{n+1} + (n+1) \left(\frac{2\lambda}{\alpha} I_n + \frac{I_{n-1}}{n-1} \right), \\ &\text{where } c_{n+1} = \frac{n+1}{\alpha} \left[e^{-\alpha x} \left(\frac{T_{n+1}}{n+1} - \frac{T_{n-1}}{n-1} \right) \right]_0^1. \end{aligned} \tag{3.10}$$

The recurrences (3.8) and (3.10) are numerically unstable[5] in the forward direction. More specifically, we get a dominant solution rather than a minimal[6] solution for these recurrences, which prohibits us from computing higher-order moments (see [17]

[5] There are algorithms that can be used for stabilizing these recurrences; see, for example, [25]. The length of the domain ω (which is less than 1) sets an upper bound on the parameter α in the recurrences (3.7), (3.8). Based on this upper bound, we have obtained a global eighth-order accurate method without resorting to any such recurrence-stabilizing algorithms. In general, it is possible to get *arbitrary* order of accuracy by stabilizing the recurrences.

[6] If we take two linearly independent solutions f_n and g_n of a three term recurrence $y_{n+1} + a_n y_n + b_n y_{n-1} = 0$, then f_n is called a minimal solution if $f_n/g_n \to 0$ as $n \to \infty$; g_n is then termed as the dominant solution. Adding any multiple of f_n to g_n would give us another dominant solution, but the minimal solution is always unique.

for more details). Since we use a qth-order piecewise polynomial approximation of $\mu(\theta)$ we need to ensure that the recurrences are stable up to $n = q$.

Remark. The more sharply peaked the integration kernel is, the more stable the recurrences are and thus the higher the order of moments that we can compute accurately. For example, the kernel $e^{-\alpha\theta}$ is sharply peaked for higher values of α.

For $\alpha \ll 1$, instead of using the recurrence (3.10), we use $e^{-\alpha x} = \sum_{k=0}^{p}(-1)^k \frac{(\alpha x)^k}{k!}$ and the recurrences given in Appendix A to compute

$$I_n = \int_0^1 e^{-\alpha x} T_n \, dx = \sum_{k=0}^{p} \frac{(-\alpha)^k}{k!} M_{nk}, \quad \text{where } M_{nk} = \int_0^1 x^k T_n \, dx; \tag{3.11}$$

here p is chosen such that $\frac{\alpha^p}{p!} < \epsilon$, the required accuracy.

The numerical instability is a property of the recurrence relation; different recurrences such as (3.5) and the ones given in Appendix A for the kernels $\frac{1}{\theta^\gamma}$ for $\gamma \in (-\infty, 1)$ and $\log\theta$ are unconditionally stable.

3.2. Heat kernels. Using the recurrences derived here, we now develop product integration rules for the time integrals of far and near parts of the double layer.

(a) *Far part.* To compute $C_n(t + \triangle t, \delta)$ using the recurrence (2.10), we have to compute

$$U_n[\hat{\phi}] = \int_{t-\delta}^{t+\triangle t-\delta} e^{-n^2\pi^2(t+\triangle t-\tau)} \hat{\phi}_n(\tau) \, d\tau. \tag{3.12}$$

As n increases, the kernel becomes sharply peaked, prohibiting the use of smooth quadrature rules to compute (3.12). Instead, we use the recurrences. We compute the Chebyshev coefficients of $\hat{\phi}_n(\tau)$ in the interval[7] $(t - \delta, t + \triangle t - \delta)$, then we have $\hat{\phi}_n(\tau) = \sum_{k=0}^{q-1} \hat{\phi}_n(k) T_k(\frac{2}{\triangle t}(\tau - t + \delta) - 1)$. By setting $\tau = t + \triangle t - \theta\triangle t - \delta$ we obtain

$$U_n[\hat{\phi}] = \triangle t e^{-n^2\pi^2\delta} \sum_{k=0}^{q-1} \hat{\phi}_n(k) \int_0^1 e^{-(n^2\pi^2\triangle t)\theta} T_k(-2\theta + 1) \, d\theta = \sum_{k=0}^{q-1} \hat{\phi}_n(k) E_{nk}. \tag{3.13}$$

For a fixed step size $\triangle t$ we can precompute

$$E_{nk} = \triangle t e^{-n^2\pi^2\delta} \int_0^1 e^{-(n^2\pi^2\triangle t)\theta} T_k(-2\theta + 1), \quad n = 1, \ldots, p, \, k = 1, \ldots, q. \tag{3.14}$$

This precomputation is done in an optimal $\mathcal{O}(pq)$ time using the recurrence (3.10) with $\alpha = n^2\pi^2\triangle t$ and the scaling parameters $\lambda = -2, \eta = 1$.

(b) *Local part.* The local part approximation of the double layer potential is given by

$$\mathcal{D}_L[\phi] = \sum_{k=1}^{2} (-1)^k \int_{t-\delta}^{t} \frac{x - b_k}{4[\pi(t-\tau)]^{3/2}} e^{-\frac{(x-b_k)^2}{4(t-\tau)}} \phi(b_k, \tau) d\tau. \tag{3.15}$$

[7] Here we assume $\delta = l\triangle t$. We have the Chebyshev coefficients of $\hat{\phi}_n(\tau)$ at each time interval Δt (see section 4). If $\delta < \triangle t$, we use the recurrence (3.10) with modified scaling factors (λ, η) instead of computing the Chebyshev coefficients in the interval $(t - \delta, t + \triangle t - \delta)$.

Let

$$\mathcal{D}_{L_{b_k}}[\phi] = (-1)^k \int_{t-\delta}^{t} \frac{x-b_k}{4[\pi(t-\tau)]^{3/2}} e^{-\frac{(x-b_k)^2}{4(t-\tau)}} \phi(b_k,\tau) d\tau. \tag{3.16}$$

Then

$$\mathcal{D}_L[\phi] = \mathcal{D}_{L_{b_1}}[\phi] + \mathcal{D}_{L_{b_2}}[\phi]. \tag{3.17}$$

Setting $\theta = \frac{t-\tau}{\delta}$ and substituting $\alpha = \frac{(x-b_k)^2}{4\delta}$ we get

$$\mathcal{D}_{L_{b_k}}[\phi] = \frac{1}{2\sqrt{\pi}} \int_0^1 \frac{\sqrt{\alpha}}{\sqrt{\theta^3}} e^{-\frac{\alpha}{\theta}} \phi(b_k, t-\delta\theta)\, d\theta. \tag{3.18}$$

Similarly to the far part case, we compute the Chebyshev coefficients[8] of $\phi(b_k,\tau)$ for $\tau \in (t-\delta, t)$. Then we use the recurrence relations (3.7) together with (3.8) to compute (3.18).

4. The overall algorithm for a stationary boundary.

So far we have presented (1) an integral equation formulation for (1.1); (2) a near-far decomposition in time of the volume and layer potentials; and (3) quadratures for convolutions of functions (defined in terms of Chebyshev polynomials) for singular or nearly singular kernels and in particular heat kernels. Next we describe how we can combine these techniques to efficiently solve (1.1) on stationary boundaries. In a nutshell, we compute the volume potentials $u_i(x,t)$ and $u_f(x,t)$ using the formulas (1.5), solve the integral equation (1.7) for the double layer density $\phi(\gamma,t)$, and use (1.6) to compute $u_b(x,t)$.

4.1. Evaluation of the volume potentials.

We will need two additional algorithms.

Computing Fourier coefficients for the far part. We need a fast algorithm for computing the sine transforms $\hat{w}$ and $\hat{f}(\tau)$ used in $\mathcal{V}_{0,F}[w]$ and $\mathcal{V}_F[f]$:

$$\hat{w}_n = \int_\Omega w(y)\sin(n\pi y)\, dy; \quad \hat{f}_n(\tau) = \int_\Omega f(y,\tau)\sin(n\pi y)\, dy, \qquad n = 1,\dots,p. \tag{4.1}$$

We assume that both f and w are given in regular grid points in space time. If f and w are periodic, then (4.1) can be computed using FFTs. Otherwise, we use the high-order hybrid Gauss trapezoidal rules developed in [1]. The advantage of using those rules is that, except for a few Gaussian nodes, all of the other quadrature nodes are on a regular grid. Then if n_g is the number of Gaussian nodes and M is the number of trapezoidal nodes, the complexity of computing the discretization of (4.1) can be reduced from $\mathcal{O}((M+n_g)p)$ to $\mathcal{O}((M+p)\log M + pn_g)$ using the FFT.

Fast Gauss transform. We need an accurate and fast evaluation of the Gauss transform defined by

$$G_\sigma[\mu](x) = \frac{1}{\sqrt{\pi\sigma}} \int_\Omega e^{-\frac{(x-y)^2}{\sigma}} \mu(y)\, dy. \tag{4.2}$$

[8] If $\delta = l\triangle t$, we need to loop through the l time intervals, each of which has a Chebyshev polynomial representation of $\phi(b_k,\tau)$. On the other hand, if $\delta < \triangle t$, we compute (3.18) from the polynomial representation of $\phi(b_k,\tau)$ in the interval $(t-\triangle t, t)$.

Using quadrature rules to discretize (4.2) and applying the fast Gauss transform (FGT) [7] on the resulting discrete sum, the complexity of computing G_σ at M points in space using M quadrature points is reduced from $\mathcal{O}(M^2)$ to $\mathcal{O}(M)$. But through this approach, uniform high-order convergence would not be possible for smaller values of σ, since the Gaussian becomes sharply peaked. So, to get the expected convergence with optimal complexity, we proceed as follows:

- We divide Ω into M/q_s uniform cells, each of size l_C; q_s is defined below.
- In each cell we use a fixed number of quadrature nodes q_s. One way to choose q_s is to ensure that a Gaussian of support equal to the size of cell can be resolved using q_s points.
- An approximate support of a Gaussian is given by $2r\sqrt{\sigma}$, where r is such that $e^{-r^2} < \epsilon$ and ϵ is the desired accuracy. If $2r\sqrt{\sigma} > l_C$, we use the FGT to compute G_σ at the M target points. Since there are M sources the complexity is $\mathcal{O}(M)$.
- If $2r\sqrt{\sigma} < l_C$ we compute G_σ directly: We truncate the domain of integration from Ω to the support $(x - r\sqrt{\sigma}, x + r\sqrt{\sigma})$ and use q_s quadrature points to discretize the integral. The forcing term at these points is computed using an appropriate high-order interpolation. The complexity is $\mathcal{O}(M)$.

4.1.1. Initial condition volume potential. The kernel $G(x, y, t)$ is approximated by the truncated Fourier series expansion if $t > \delta$ and by the Kelvin expansion otherwise. Hence,

$$u_i(x,t) = \begin{cases} G_{2t}[w](x), & \text{if } t \le \delta, \\ \sum_{n=1}^{p}(2e^{-n^2\pi^2 t}\hat{w}_n)\sin(n\pi x), & \text{otherwise.} \end{cases} \tag{4.3}$$

The transform $G_{2t}[w]$ is evaluated using the aforementioned strategy with a quadrature rule of order q. The inverse NUFFT is used to compute the discrete sum in (4.3) for $t > \delta$.

4.1.2. Distributed forcing volume potential. The evaluation of (1.5) is split into a far part $(0, t - \delta)$ and a local part $(t - \delta, t)$. At time interval $(t, t + \triangle t)$ the evaluation of the far part $\mathcal{V}_F[f]$ involves the following steps:

- Computation of the Chebyshev coefficients for each of $\{\hat{f}_n(\tau)\}_{n=1}^{p}$ by evaluating (4.1) (using quadrature rules in space) at the Chebyshev nodes belonging to $(t - \delta, t + \triangle t - \delta)$.
- For $n = 1, \dots, p$, computation of the update $U_n[\hat{f}]$ as defined in (3.12) using Chebyshev recurrences developed in section 3.
- Update of $\{C_n\}_{n=1}^{p}$ using $C_n(t + \triangle t, \delta) = C_n(t, \delta) + 2U_n[\hat{f}]$.
- Evaluation of the discrete sum $\mathcal{V}_F[f](x, t+\triangle t) = \sum_{n=1}^{p} C_n(t+\triangle t, \delta)\sin(n\pi x)$ at the target points using the inverse NUFFT.

The local part of $u_f(x,t)$ denoted by $\mathcal{V}_L[f]$ is given by

$$\mathcal{V}_L[f](x,t) = \int_{t-\delta}^{t}\int_{\Omega} \frac{e^{-\frac{(x-y)^2}{4(t-\tau)}}}{\sqrt{4\pi(t-\tau)}} f(y,\tau)dy\,d\tau. \tag{4.4}$$

Setting $\sigma^2 = t - \tau$ and noting that, outside the interval $(x - 2r\sigma, x + 2r\sigma)$, the Gaussian decays exponentially fast to zero, we obtain

$$\mathcal{V}_L[f](x,t) \approx \int_0^{\sqrt{\delta}} g(\sigma)\,d\sigma, \quad \text{where } g(\sigma) = \frac{1}{\sqrt{\pi}}\int_{x-2r\sigma}^{x+2r\sigma} e^{-\frac{(x-y)^2}{4\sigma^2}} f(y, t-\sigma^2)dy. \tag{4.5}$$

By substituting $y = x + 2\xi\sigma$, we get $g(\sigma) = \frac{2\sigma}{\sqrt{\pi}} \int_{-r}^{r} e^{-\xi^2} f(x + 2\xi\sigma, t - \sigma^2) d\xi$; the integrand is smooth, and hence, $g(\sigma)$ is a smooth function in σ. Then, we use a high-order quadrature rule of order q to integrate $g(\sigma)$, i.e., $\mathcal{V}_L[f](x,t)] = \sum_{k=1}^{q} w_k g(\sigma_k)$, where w_k, σ_k are the quadrature weights and nodes, respectively. Notice that $g(\sigma_k) = 2\sigma_k G_{2\sigma_k}[f](t - \sigma_k^2)$ at each σ_k. Hence, the optimal strategy for computing the Gauss transform discussed previously can be used for the accurate computation of $g(\sigma_k)$.

4.2. Computation and evaluation of the boundary double layer potential. To evaluate $u_b(x,t)$, we first need to compute the density $\phi(\gamma, t)$ by solving (1.7). At each time step, we incrementally solve (1.7) for the Chebyshev coefficients of $\phi(b_1, t)$ and $\phi(b_2, t)$. Let $\phi(b_1, n), \phi(b_2, n)$ denote the corresponding nth-order Chebyshev coefficients. Then, for any $\tau \in [t, t + \triangle t]$, we have

$$\phi(b_k, \tau) = \sum_{n=0}^{q-1} \phi(b_k, n) T_n\left(\frac{2}{\triangle t}(\tau - t) - 1\right), \quad k = 1, 2.$$

Since we require q Chebyshev coefficients at each point on the boundary, we need to solve (1.7) at q collocation nodes $\{t + t_i\}_{i=1}^{q}$. The natural choice of the collocation nodes are the zeros of qth-order Chebyshev polynomial defined in the interval $[t, t + \triangle t]$. Then the transformation between physical and spectral space can be performed using the FFT. In addition, interpolation at the Chebyshev nodes gives rise to a stable approximation of the interpolated function.

The Chebyshev nodes are given by

$$t_i = \frac{\triangle t}{2}\left(1 + \cos\frac{\pi(i-1)}{q}\right), \qquad i = 1, \ldots, q. \tag{4.6}$$

The integral equation (1.7) can be rewritten as

$$\begin{aligned} &-\frac{1}{2}\phi(x, t + t_i) + \int_{t}^{t+t_i} \int_{\gamma} \frac{\partial G(x, t + t_i; y, \tau)}{\partial n(y)} \phi(y, \tau) ds(y) d\tau \\ &= u_b(x, t + t_i) - \int_{0}^{t} \int_{\gamma} \frac{\partial G(x, t + t_i; y, \tau)}{\partial n(y)} \phi(y, \tau) ds(y)\, d\tau \quad \forall x \in \gamma. \end{aligned} \tag{4.7}$$

The resulting finite-dimensional linear equation is $K\phi_h = r$, where

$$\phi_h = \left[\phi(b_1, 0), \ldots, \phi(b_1, q-1),\ \phi(b_2, 0), \ldots, \phi(b_2, q-1)\right]^T. \tag{4.8}$$

Assuming $\delta \geq \triangle t$, the entries of K are given by $K = \left[\begin{smallmatrix} A & B \\ B & A \end{smallmatrix}\right]$,

$$\text{where} \quad A_{ij} = -\frac{1}{2} T_{j-1}\left(\frac{2}{\triangle t} t_i - 1\right), \quad i, j = 1, \ldots, q, \tag{4.9}$$

$$B_{ij} = \int_{t}^{t+t_i} \frac{b_2 - b_1}{4[\pi(t-\tau)]^{3/2}} e^{-\frac{(b_2 - b_1)^2}{4(t-\tau)}} T_{j-1}\left(\frac{2}{\triangle t}(\tau - t) - 1\right) d\tau. \tag{4.10}$$

The entries of B can be computed using recurrences on $I_n(\alpha) = \int_0^1 \frac{\sqrt{\alpha}}{\sqrt{\theta^3}} e^{-\frac{\alpha}{\theta}} T_n(\lambda\theta + \eta) d\theta$. The operator K is well-conditioned and can by "inverted" using a matrix-free GMRES.[9] The integration of the history part of double layer on the right-hand side

[9] Given a fixed time step $\triangle t$, we can precompute the inverse K^{-1} once and subsequently solve for ϕ_h by just one matrix vector multiplication $K^{-1}r$. This is can be done efficiently only in 1D and for static boundaries.

of (4.7), denoted by $\mathcal{D}_h$, is split into $(0, t+t_i-\delta)$ and $(t+t_i-\delta, t)$. The first interval integral is

$$\sum_{n=1}^{p}\left(e^{-n^2\pi^2 t_i}C_n(t,\delta)+2\int_{t-\delta}^{t+t_i-\delta}e^{-n^2\pi^2(t+t_i-\tau)}\hat{\phi}(\tau)\,d\tau\right)\sin(n\pi x). \tag{4.11}$$

We use the local part expansion of the kernel for the second interval. The time integration in both intervals is performed using the product integration rules discussed in section 3. Since the coefficients C_n are updated at each time step, r can be computed with constant work per time step. Then $u_b(x,t)$ can be evaluated at the target M spatial locations using (1.6) and the fast summation method for the double layer described in section 2.

4.3. Summary of the overall scheme for stationary boundaries. The overall algorithm is summarized in Algorithm 1. The output of the algorithm is $u(x,t)$ at the target points. The inputs are

- **parameters:** A fixed time step Δt and an error tolerance ϵ.
- **boundary conditions:** In each time interval $(t, t+\triangle t)$, the boundary data at the collocation nodes $g(b_k, t+t_i), i = 1, \ldots, q$.
- **distributed force:** The forcing f at points belonging to a regular grid of size $M \times N$ imposed on $\Omega \times [0,T]$.
- **initial condition:** w at M regular points on Ω.

Since we use special quadrature points to compute the volume integrals (4.1), f and w are evaluated using high-order interpolation schemes; for example, FFT combined with B-splines can be used [27].

The overall complexity of the algorithm is $\mathcal{O}(MN\log M)$. Below, we discuss the different sources for errors in the final solution.

- Truncating the series expansion for Green's function in the near and far parts. These have been shown to be exponentially convergent with increasing M, N.
- The order of the Chebyshev polynomials used for approximating the density ϕ in each time interval.
- The quadrature rule used for the computing the spatial integrals (4.1) and time integral (4.5).

By approximating ϕ with $(q-1)$th-order Chebyshev polynomials and picking a quadrature rule of order q to perform the space and time integration for the volume potentials, we get a qth-order method. In section 6, we present results that validate our scheme.

5. Moving boundary.

Now we will describe the methodology for solving (1.1) for boundaries with a prescribed motion. By linearity, we decompose the solution into three parts: $u = u_i + u_f + u_b$. The three parts correspond to the solutions of an initial-value problem (1.2), an inhomogeneous problem with homogeneous boundary conditions (1.3), and a homogeneous problem with inhomogeneous boundary conditions (1.4). The motion of the boundary $\omega(t)$ is assumed to be contained within the unit box Ω; hence, the solutions $u_i(x,t)$ and $u_f(x,t)$ are given by (1.5). At any point x inside the domain $\omega(t)$ we assume that $u_b(x,t) = \mathcal{D}[\phi](x,t)$. As $G(x,t;y,\tau)$ satisfies the adjoint equation (2.1) and the potential u_b satisfies the heat equation, the initial condition $u_b(x,0) = 0$ is satisfied by construction. To satisfy the prescribed boundary conditions on $\gamma(t)$, we need to compute the jumps in the double layer as $x \to \gamma(t)$. The conditions have been derived before; for example, see [21, 9]. We have included their derivation for completeness. We first discuss the computation of the double layer potential at a point away from the boundary, and then we give the jump conditions.

Algorithm 1. Overall algorithm for stationary boundaries.

```
INITIALIZATION
Choose $p$ and $l$ using (2.11)
Compute $E$ (3.14) using (3.10)
Compute $\hat{w}_n$ and set $C_n = 2e^{-n^2\pi^2(l-1)\triangle t}\hat{w}_n$, $n = 1 \dots p$        $\mathcal{O}((M+p)\log M)$
Compute $t_i$ using (4.6)
Compute $K^{-1}$                                                                  $\mathcal{O}(q^3)$

TIME MARCHING
for $j = 1 : N$ do
  If $j < l$, $l = j$                                        first $l$ time steps: $\delta = j\triangle t$

  SOLVE FOR THE BOUNDARY DENSITY
  loop over boundary nodes
  for $k = 1 : 2$ do
    Loop over time-collocation nodes
    for $i = 1 : q$ do
      $t_c = (j-1)\triangle t + t_i$
      $r_k(i) = g(b_k, t_c) - \mathcal{V}[f](b_k, t_c) - \mathcal{V}_0[w](b_k, t_c)$          $\mathcal{O}(M)$, using FGT
      $r_k(i) = r_k(i) - \mathcal{D}_h[\phi](b_k, t_c)$
    end for
  end for
  $\phi_h = K^{-1}r$

  EVALUATION AND UPDATE
  Local part
  $u(x_k, j\triangle t) = \mathcal{V}_L[f](x_k, j\triangle t)$, $k = 1 \dots M$                         $\mathcal{O}(M)$
  $u(x_k, j\triangle t) = u(x_k, j\triangle t) + \mathcal{D}_L[\phi](x_k, j\triangle t)$, $k = 1 \dots M$              $\mathcal{O}(Mq)$
  if $j < l$ then
    Initial Condition
    $u(x_k, j\triangle t) = u(x_k, j\triangle t) + G_{j\triangle t}[w](x_k)$, $k = 1 \dots M$               $\mathcal{O}(M)$
  else
    Far part
    for n = 1:p do
      Compute $U_n[\hat{\phi} + \hat{f}]((j-1)\triangle t)$ using (3.12)
      $C_n = e^{-n^2\pi^2\triangle t}C_n + 2U_n[\hat{f} + \hat{\phi}]$
    end for                                                                   $\mathcal{O}(pq+p)$
    $u(x_k, j\triangle t) += \sum_{n=1}^{p} C_n \sin(n\pi x_k)$, $k = 1 \dots M$                 $\mathcal{O}((M+p)\log M)$
  end if
end for
```

To evaluate the double layer, we split it into a local and a far part: $\mathcal{D}[\phi] = \mathcal{D}_L[\phi] + \mathcal{D}_F[\phi]$. We first discuss the local part computation. We rewrite the local part as $\mathcal{D}_L[\phi] = \mathcal{D}_{L_{b_1}}[\phi] + \mathcal{D}_{L_{b_2}}[\phi]$, with each term being defined as in (3.16) but using the time-dependent boundaries. Let's look at computing $D_{L_{b_2}}[\phi](x,t)$ defined by

$$\mathcal{D}_{L_{b_2}}[\phi] = \int_{t-\delta}^{t} \frac{x - b_2(\tau)}{4[\pi(t-\tau)]^{3/2}} e^{-\frac{(x-b_2(\tau))^2}{4(t-\tau)}} \phi(b_2(\tau), \tau) d\tau. \tag{5.1}$$

Setting $\tau = t - \delta\theta$, $\frac{x - b_2(\tau)}{2\sqrt{\delta}} = \beta(\theta)$, and $\phi(b_2, t - \delta\theta) = \psi(\theta)$ we get

$$\mathcal{D}_{L_{b_2}}[\phi] = \frac{1}{2\sqrt{\pi}} \int_0^1 \frac{\beta(\theta)}{\sqrt{\theta^3}} e^{-\frac{\beta^2(\theta)}{\theta}} \psi(\theta)\, d\theta. \tag{5.2}$$

Unlike the static boundary case, we cannot directly use the recurrences (3.7), (3.8) to compute (5.2) as β is not a constant. Instead, we rewrite (5.2) as

$$\mathcal{D}_{L_{b2}} = \frac{1}{2\sqrt{\pi}} \int_0^1 \frac{e^{-\frac{\beta_0^2}{\theta}}}{\sqrt{\theta^3}} \{\beta(\theta) E_\beta(\theta) \psi(\theta)\}\, d\theta, \quad \text{where } E_\beta(\theta) = e^{-\frac{\beta^2 - \beta_0^2}{\theta}}, \tag{5.3}$$

and $\beta_0 = \beta(0)$. The kernel in this expression is similar to the static boundaries case (3.18). Therefore, we first compute the Chebyshev coefficients of the function $\beta E_\beta \psi$ and then use the recurrences (3.7), (3.8) with $\alpha = \beta_0^2$ to compute (5.3).

Remark. Assuming $\beta(\theta)$ is sufficiently smooth, it is easy to see that E_β is also smooth: By expanding $\beta(\theta)$ about $\theta = 0$ using Taylor's series as $\beta(\theta) = \sum_{k=0}^{\infty} \beta_k \theta^k$, we can see that $E_\beta(\theta)$ is of the form $e^{-\sum_{k=0}^{\infty} c_k \theta^k}$.

Similarly, the single layer potential is calculated by rewriting it as

$$\mathcal{S}_{L_{b_2}}[\phi] = \frac{1}{2\sqrt{\pi}} \int_0^1 \frac{e^{-\frac{\beta^2(\theta)}{\theta}}}{\sqrt{\theta}} \psi(\theta)\, d\theta = \frac{1}{2\sqrt{\pi}} \int_0^1 \frac{e^{-\frac{\beta_0^2}{\theta}}}{\sqrt{\theta}} \{E_\beta \psi(\theta)\}\, d\theta \tag{5.4}$$

and using the recurrence (3.8). Now let us examine the limit as the evaluation point x approaches the boundary. As $x \to b_2(t)$, the constant term in Taylor's expansion of $\beta(\theta)$ given by $\beta_0 = \frac{x - b_2(t)}{2\sqrt{\delta}}$ approaches zero.

PROPOSITION 5.1. *We define the single layer potential as* $\mathcal{S}_L[f](\beta_0) = \int_0^1 \frac{e^{-\frac{\beta_0^2}{\theta}}}{\sqrt{\theta}} f(\theta)\, d\theta$ *and the double layer potential as* $\mathcal{D}_L[f](\beta_0) = \int_0^1 \beta_0 \frac{e^{-\frac{\beta_0^2}{\theta}}}{\sqrt{\theta^3}} f(\theta)\, d\theta$. *Then* $\mathcal{S}_L[f]$ *is continuous as* $\beta_0 \to 0$*, whereas* $\mathcal{D}_L[f]$ *sustains a jump equal to* $f(0)$*.*

$$\lim_{\beta_0 \to 0^-} \mathcal{S}_L[f] = \lim_{\beta_0 \to 0^+} \mathcal{S}_L[f] = \int_0^1 \frac{f(\theta)}{\sqrt{\theta}}\, d\theta, \tag{5.5}$$

$$\lim_{\beta_0 \to 0^-} \mathcal{D}_L[f] = -\frac{1}{2} f(0), \quad \lim_{\beta_0 \to 0^+} \mathcal{D}_L[f] = \frac{1}{2} f(0). \tag{5.6}$$

We give a proof of this proposition in Appendix B. In the case of static boundaries, using this proposition, we arrived at the Volterra system (1.7). For moving boundaries, clearly the proposition cannot be applied directly, since β is not constant. Here we write the double layer potential defined in (5.2) as follows:

(5.7)
$$\mathcal{D}_{L_{b_2}} = \frac{1}{2\sqrt{\pi}} \int_0^1 \frac{e^{-\frac{\beta_0^2}{\theta}}}{\sqrt{\theta}} \left\{ \left(\frac{\beta(\theta) - \beta_0}{\theta} \right) E_\beta \psi(\theta) \right\} d\theta + \frac{1}{2\sqrt{\pi}} \int_0^1 \frac{\beta_0 e^{-\frac{\beta_0^2}{\theta}}}{\sqrt{\theta^3}} \{E_\beta \psi(\theta)\}\, d\theta.$$

Let us define $F_\beta(\theta) = \frac{\beta(\theta) - \beta_0}{\theta}$. Using Taylor's series (and assuming sufficient smoothness), we have $F_\beta(\theta) = \sum_{k=1}^{\infty} \beta_k \theta^{k-1}$. Therefore, $F_\beta(\theta)$ is smooth and independent of β_0 and $\lim_{\theta \to 0} F_\beta(\theta) = \beta_1$, where $\beta_1 = -\frac{\dot{b}(t)}{2\sqrt{\delta}}$. We denote the first integral in (5.7) by $\tilde{\mathcal{S}}_{L_{b_1}}$; it is a single layer potential with constant β_0, and hence it is continuous as $\beta_0 \to 0$. By Proposition 5.1, the second integral approaches $-\frac{1}{2} E_\beta(0) \psi(0)$

as $\beta \to 0^-$. We can show that $\lim_{\theta \to 0_{\beta_0 \to 0}} E_\beta(\theta) = 1$. The Volterra system that we obtain for moving boundaries is given by

$$(5.8) \qquad -\frac{1}{2}\phi(x,t) + \int_0^t \int_{\gamma(\tau)} \frac{\partial G(x,t;y,\tau)}{\partial n(y)} \phi(y,\tau)\, ds(y)\, d\tau = u_b(x,t)\, \forall x \in \gamma(\tau).$$

In the case of static boundaries, the Volterra equation (1.7) at the boundary point b_1 at time t can be written as

$$(5.9) \qquad \left(-\frac{1}{2}I + \mathcal{D}_{F_{b_1}} + \mathcal{D}_{b_2}\right)[\phi](b_1,t) = u_b(b_1,t),$$

since the local part of the double layer $D_{L_{b_1}}[\phi](b_1,t) \to -\frac{1}{2}\phi(b_1,t)$. The far part does not contribute to the jump in the double layer. From (5.7), we can show that the Volterra equation in the case of moving boundaries at $(b_1(t),t)$ is given by

$$(5.10) \qquad \left(-\frac{1}{2}I + \mathcal{D}_{F_{b_1}} + \tilde{\mathcal{S}}_{L_{b_1}} + \mathcal{D}_{b_2}\right)[\phi](b_1(t),t) = u_b(b_1(t),t).$$

We can verify that, as the boundary motion becomes negligible, (5.10) leads to (5.9): $\beta(\theta) \to \beta_0$ and $\lim_{\beta(\theta)\to\beta_0} \tilde{\mathcal{S}}_{L_b}[\phi](b(t),t) = 0$ as the function $F_\beta(\theta) \to 0$ since $\beta_k \to 0$ for all $\{\beta_k\}_{k=1}^{\infty}$.

We now summarize the changes in the algorithm described in section 4, which are necessary for moving boundary problems. At each time step, we solve (5.8) for the Chebyshev coefficients of $\phi(b_1(t),t)$ and $\phi(b_2(t),t)$ using GMRES. Assuming $\phi(b_2,t)$ is given, our aim is to solve the resulting algebraic system after discretizing the operator equation (5.10) by collocation at nodes t_i. Starting with an initial guess $\{\phi_n(b_1)\}_{n=0}^{q-1}$, we first compute $\phi_0(b_1,t_i)$ at the collocation nodes in time t_i by inverse FFT. Computing the left-hand side of (5.10) involves computing three operators at (b_1,t_i) : the far part $\mathcal{D}_{F_{b_1}}[\phi]$, the local part of the single layer $\tilde{\mathcal{S}}_{L_{b_1}}[\phi]$, and the double layer $\mathcal{D}_{b_2}[\phi]$.

- As with the static case, the far part $\mathcal{D}_{F_{b_1}}[\phi](b_1,t)$ is computed by storing and updating $\{C_n(t,\delta)\}_{n=1}^{p}$ and evaluating (2.9). The only difference is that the boundaries are time-dependent in the definition of $\hat{\phi}_n(\tau)$.
- To compute $\tilde{\mathcal{S}}_{L_{b_1}}[\phi](b_1,t_i)$, we need to evaluate the function $F_\beta(\theta)E_\beta(\theta)\psi(\theta)$ in (5.7). Let

$$(5.11) \qquad \phi_{\tilde{\mathcal{S}}}(t,\tau) = \frac{b_1(t) - b_1(\tau)}{2(t-\tau)} e^{-\frac{(b_1(t)-b_1(\tau))^2}{4(t-\tau)}} \phi(b_1(\tau),\tau),$$

and then

$$\begin{aligned} \tilde{\mathcal{S}}_{L_{b_1}}[\phi](b_1(t_i),t_i) &= \int_{t_i-\delta}^{t_i} G\left(b_1(t_i), b_1(t_i); t_i, \tau\right) \phi_{\tilde{\mathcal{S}}}(t_i,\tau)\, d\tau \\ &= \int_{t_i-\delta}^{t_i} \frac{\phi_{\tilde{\mathcal{S}}}(t_i,\tau)}{\sqrt{4\pi(t_i-\tau)}}\, d\tau. \end{aligned}$$

To compute the Chebyshev coefficients $\{\phi_{\tilde{\mathcal{S}}}(t_i,n)\}_{n=0}^{q-1}$, we evaluate $\phi_{\tilde{\mathcal{S}}}(t_i,\tau)$ at the collocation nodes $\{t_j\}_{j=1}^{q}$ (which also are the Chebyshev nodes). Note that, when $\tau \to t$, we have $\lim_{\tau\to t} \phi_{\tilde{\mathcal{S}}}(t,\tau) = -\dot{b}_1(t)$. Then we evaluate $\tilde{\mathcal{S}}_{L_{b_1}}[\phi](b_1(t_i),t_i)$ using the recurrence (3.8) with appropriate scaling factors and $\alpha = 0$.

- To compute $\mathcal{D}_{b_2}[\phi]$, we split it into a far and a local part. The local part is computed by rewriting it as in (5.3) with $\beta_0 = \frac{b_1(t_i)-b_2(t_i)}{2\sqrt{\delta}}$.

Once we solve for the Chebyshev coefficients of $\phi(b_1,t)$ and $\phi(b_2,t)$ the potential u_b is evaluated at all target locations using (1.6). The asymptotic complexity and accuracy of the algorithm are the same with the stationary case.

6. Numerical results. In this section we illustrate the accuracy of the quadrature scheme described in section 3 for different kernels. Then we present numerical results for solving the heat equation (1.1).

6.1. Quadratures. We compare the hybrid quadratures of Alpert [1] with the Chebyshev spectral method. First we compute the following integral:

$$\int_0^1 \cos(200x)s(x) + \cos(200x+0.5)\,dx \tag{6.1}$$

for the functions $s(x) = 0, s(x) = \frac{1}{\sqrt{x}}, s(x) = \log x$. The recurrence relations for computing the moments $I_k = \int_0^1 s(x)T_k(2x-1)dx$ are given in Appendix A. We report relative errors for both the Chebyshev method and the quadrature methods in [1] for convergence orders 4, 8, and 16. See Figure 6.1.

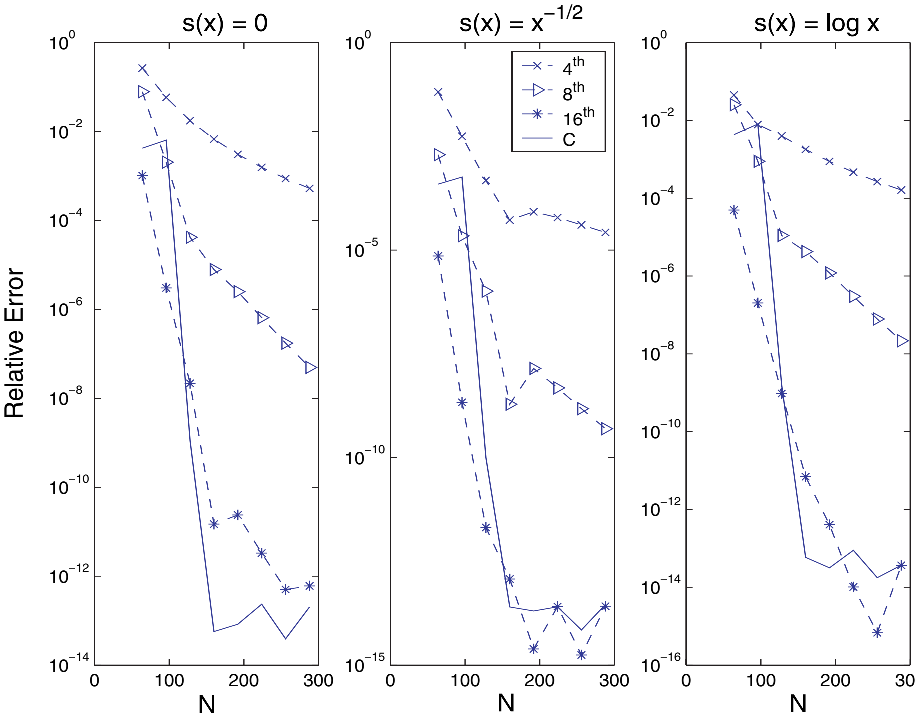


FIG. 6.1. *Plot of the relative errors given in Tables* 6.1, 6.2, *and* 6.3 *w.r.t. p, the number of Chebyshev modes used. As the number of quadrature points is increased, we observe that the quadrature error gets reduced by a constant multiple, whereas in the case of the Chebyshev approximation, the error decays exponentially.*

Table 6.1

In this table we report a numerical comparison between our method and the quadrature rules proposed in [1] *for convergence orders* 4, 8, *and* 16. *Relative errors in computing the integral* (6.1) *for the case* $s(x) = 0$, p *indicates the number of Chebyshev coefficients or the total number of points used in the quadrature rule.*

p	4	8	16	Chebyshev
64	2.66e-01	7.85e-02	1.01e-03	4.21e-03
96	5.83e-02	2.03e-03	3.04e-06	6.39e-03
128	1.75e-02	4.18e-05	2.16e-08	1.11e-09
160	6.72e-03	7.87e-06	1.48e-11	5.66e-14
192	3.05e-03	2.50e-06	2.37e-11	8.32e-14
224	1.56e-03	6.50e-07	3.28e-12	2.33e-13
256	8.75e-04	1.74e-07	5.00e-13	3.88e-14
288	5.25e-04	4.95e-08	6.00e-13	2.04e-13

Table 6.2

Relative errors in computing the integral (6.1) *for the case* $s(x) = \frac{1}{\sqrt{x}}$.

p	4	8	16	Chebyshev
64	6.42e-02	1.98e-03	7.17e-06	3.87e-04
96	5.53e-03	2.19e-05	2.14e-09	5.73e-04
128	4.74e-04	1.02e-06	2.06e-12	1.00e-10
160	5.27e-05	1.92e-09	1.18e-13	2.50e-14
192	8.35e-05	1.40e-08	2.44e-15	2.01e-14
224	6.09e-05	4.78e-09	2.57e-14	2.53e-14
256	4.04e-05	1.50e-09	1.74e-15	6.99e-15
288	2.66e-05	4.94e-10	2.62e-14	2.88e-14

Table 6.3

Relative errors in computing the integral (6.1) *with* $s(x) = \log x$.

p	4	8	16	Chebyshev
64	4.46e-02	2.52e-02	4.96e-05	4.31e-03
96	7.84e-03	8.83e-04	2.03e-07	8.07e-03
128	3.97e-03	1.09e-05	9.53e-10	1.53e-09
160	1.79e-03	4.25e-06	6.93e-12	5.88e-14
192	8.76e-04	1.20e-06	4.04e-13	3.16e-14
224	4.66e-04	2.99e-07	1.01e-14	8.93e-14
256	2.67e-04	7.82e-08	6.79e-16	1.76e-14
288	1.62e-04	2.16e-08	3.65e-14	3.68e-14

We now present numerical results for evaluating the following integral for two different functions $f(x)$:

$$J(\alpha) = \int_0^1 \frac{e^{-\frac{\alpha}{x}}}{\sqrt{x}} f(x)\, dx. \tag{6.2}$$

The relative errors in computing (6.2) using two different methods are presented in Tables 6.4 and 6.5. One is to use *Nyström's method* with the quadrature nodes and weights of the singular kernel $x^{-1/2}$, and the other is to approximate the function $f(x)$ using Chebyshev polynomials and then using the recurrence relations (3.8). In Table 6.4, we can see that the relative error is high in the case of Nyström's method even if we take as many as 512 points. This can be expected as the behavior of the heat kernels is different from $x^{-1/2}$. So we need to develop separate quadrature rules. Developing such quadratures for each value α is not practical. Instead, we can compute $\{I_n(\alpha)\}_{n=1}^{p}$ for each α in $\mathcal{O}(p)$ work using the recurrence relations (3.8). For

Table 6.4

Relative errors in computing the double layer heat potential with density $f_1(x) = 1 + x^{10}$ and $\alpha = 2 \times 10^{-4}$. Here N indicates the number of quadrature nodes or the order of Chebyshev polynomials used to approximate the function $f(x)$; q denotes the order of singular quadrature rule used. Observe that the error in the case of quadrature doesn't reduce as $\mathcal{O}(N^{-q})$, whereas in the second case the error reaches machine precision for $N > 10$. Recall that the smaller the α, the more difficult the computation of the heat kernel becomes.

N	4	8	12	16	20	512
$q = 4$	1.87e-01	1.09e-02	8.31e-03	6.36e-03	4.78e-03	3.64e-04
Chebyshev	1.38e-03	8.59e-06	2.17e-16	2.17e-16	2.17e-16	2.17e-16

Table 6.5

Relative errors in computing (6.2) *with the function $f_2(x) = \cos(20x)$ and $\alpha = 2 \times 10^{-4}$. The* 16*th-order singular quadrature rule from* [1] *is used.*

N	16	20	24	28	32	36
$q = 16$	2.94e-01	7.75e-02	7.64e-02	7.53e-02	7.46e-02	7.39e-02
Chebyshev	3.34e-04	2.16e-06	5.75e-09	7.51e-12	4.00e-15	1.20e-15

Table 6.6

Relative errors in computing $I(\alpha)$ defined by (6.3) *using recurrences for the functions $f_1(x) = 1 + x^{10}$ and $f_2(x) = \cos(20x)$, for $\alpha = 2 \times 10^{-8}$.*

N	4	8	16	32	64
$f_1(x)$	7.38e-02	3.61e-04	3.75e-16	1.25e-16	1.25e-16
$f_2(x)$	2.62e-01	5.32e-01	2.61e-03	6.84e-14	2.50e-16

Table 6.7

We report l_∞–norm errors for the solution of the heat equation with an exact solution given by $u(x,t) = e^{-10^2 t}\sin(10x)$. In this example we solve a problem with Neumann boundary conditions and zero distributed forces. We compute the numerical solution using a direct integral equation formulation that results in a well-conditioned Volterra equation.

$N = M$	2	4	8	16	32	64
$q = 4$	4.62e-02	1.01e-03	5.34e-05	3.03e-06	2.66e-07	2.11e-08
$q = 6$	1.72e-03	7.09e-05	4.15e-06	1.62e-07	1.39e-09	4.94e-11
$q = 8$	4.36e-04	7.86e-06	3.33e-07	1.05e-08	1.71e-11	5.36e-13

the same functions $f_1(x)$ and $f_2(x)$, we evaluate the integrals

$$I(\alpha) = \int_0^1 \frac{\sqrt{\alpha} e^{-\frac{\alpha}{x}}}{\sqrt{x^3}} f(x)\, dx. \tag{6.3}$$

See Table 6.6. In Figure 6.2, we plot the relative errors in computing aforementioned integral operators by the Chebsyshev method.

6.2. The heat equation: Stationary boundaries. In this section we report results for the diffusion problems in domains with static boundaries. In all of the examples, the domain ω is taken to be $[0.4, 0.6]$, $\delta = 10^{-3}$, and the total time $T = 0.1$; N indicates the number of time steps and q the number of Chebyshev polynomials used to approximate the functions in the boundary integrals at each time step. We report the l_∞ error evaluated at M spatial points and N time levels in Tables 6.7 and 6.8 for two different analytic solutions.

6.3. The heat equation on domains with moving boundaries. Here we present numerical results for the solution of the heat equation on domains with prescribed boundary motion. The boundary position, as a function of time, is given by

TABLE 6.8

In this table we report l_∞-norm errors for the heat equation (1.1) *problem with an analytic solution given by $u(x,t) = \cos(100x)\cos(200t)$ and a corresponding nonzero distributed force. We consider Dirichlet boundary conditions and an indirect double layer formulation. We use eight quadrature points in computing the volume integrals in* (4.1)*. See Table* 6.9 *for running times.*

N	2	4	8	16	32	64
$q = 4$	7.16e-01	1.28e-04	3.54e-05	2.41e-06	1.51e-07	1.17e-08
$q = 6$	8.27e-01	8.10e-06	7.80e-07	6.72e-08	1.08e-09	2.92e-11
$q = 8$	7.75e-01	3.74e-07	8.38e-08	3.40e-09	2.46e-11	3.34e-13

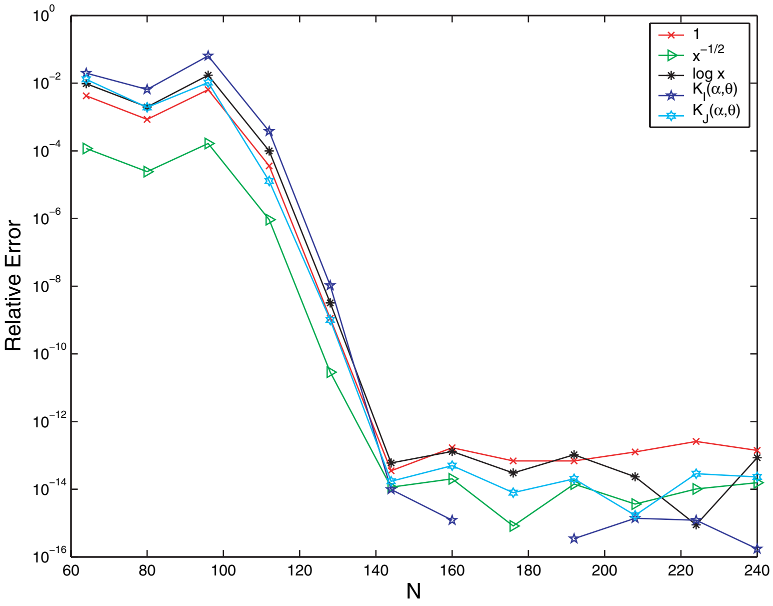


FIG. 6.2. *Relative error plot (logarithmic scale) versus the number of Chebyshev modes N used for approximating the function $f(x) = \cos(200x)$ in computing the integral $I(\alpha) = \int_0^1 K(\alpha,x)f(x)\,dx$ for the kernels $1, x^{-1/2}, \log x$ and for the heat kernels $\frac{\sqrt{\alpha}}{\sqrt{x^3}}e^{-\frac{\alpha}{x}}, \frac{e^{-\frac{\alpha}{x}}}{\sqrt{x}}$, with $\alpha = 2 \times 10^{-4}$. The integration errors follow the approximation error of the function $f(x)$. We can observe that, once $f(x)$ is resolved, the integration is accurate to machine precision—independently of the kernel in which we used in the integration.*

TABLE 6.9

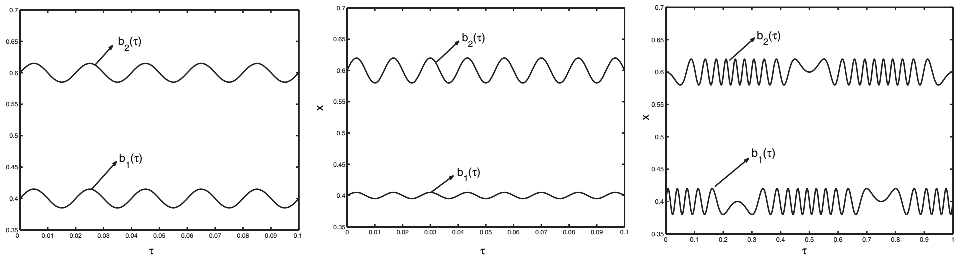
Representative running times for the example in Table 6.8*. Notice that only the relative increase is important here, as the code is implemented in* `MATLAB` *and has not been optimized. The following observations can be made: First, the computation time grows linearly with N; second, for a fixed N, the computation time increases by a constant factor as the order q is increased.*

N	2	4	8	16	32	64
$q = 4$	0.8	1.3	2.6	5.3	10.3	21.1
$q = 6$	1.0	1.7	3.3	6.0	12.9	26.3
$q = 8$	1.3	2.3	4.0	8.0	16.0	32.6

$b_1(t) = 0.4 + s_1(t)$ and $b_2(t) = 0.6 + s_2(t)$. We present results for three different motions of the boundary for which the functions $s_1(t)$ and $s_2(t)$ are defined in Table 6.10. The motion of the boundary is depicted in Figure 6.3.

Table 6.10
Test cases for the boundary motion.

Case	$s_1(t)$	$s_2(t)$
(i)	$0.01\sin(100\pi t)$	$0.01\sin(100\pi t)$
(ii)	$0.005\sin(150\pi t)$	$0.02\sin(150\pi t)$
(iii)	$0.02\sin[10\pi\sin(2\pi t)]$	$0.02\sin[10\pi\cos(2\pi t)]$



(a) case (i) (b) case (ii) (c) case (iii)

Fig. 6.3. *Plot of test cases for the boundary motion.*

Table 6.11
l_∞–norm errors for the solution of Example 1 *with the $\omega(t)$ varying in time as given in case* (i). *See Table* 6.12 *for CPU times.*

$N = M$	2	4	8	16	32	64
$q = 4$	1.85e-01	1.00e-01	8.59e-03	3.28e-04	1.72e-05	9.42e-07
$q = 6$	1.37e-01	9.85e-03	2.08e-04	4.90e-06	3.80e-08	1.47e-09
$q = 8$	6.36e-02	6.13e-04	9.96e-06	3.39e-08	2.62e-10	9.01e-13

Table 6.12
CPU times for the example in Table 6.11.

$N = M$	2	4	8	16	32	64
$q = 4$	0.80	1.27	2.60	4.94	10.34	21.47
$q = 6$	1.19	2.32	4.49	8.83	17.84	36.86
$q = 8$	1.92	3.58	7.14	13.71	27.81	56.25

Example 1 (prescribed Neumann conditions). We solve the homogeneous heat equation with Neumann boundary conditions given by $\frac{\partial u}{\partial x}(b_k, t) = 2e^{-4t}\cos(2b_k(t))$ and an initial condition given by $u(x, 0) = \sin(2x)$. For this problem the exact solution is given by $u(x, t) = e^{-4t}\sin(2x)$. We report l_∞-norm errors of our numerical approximation scheme in Tables 6.11, 6.13, and 6.14. We have used a direct formulation that results in a well-conditioned integral equation. We solve this problem for the three cases of boundary motion. At each time step, we sample the numerical and exact solution at a uniformly spaced spatial grid. We report the maximum of the absolute error values over all time steps and spatial points. We observe optimal convergence rates for all three cases.

Example 2 (prescribed Dirichlet data). In this second example we solve a heat equation problem with distributed forces and prescribed Dirichlet boundary condi-

Table 6.13
l_∞–norm errors for the solution of Example 1 *with the $\omega(t)$ varying in time as given in case* (ii)*.*

$N = M$	2	4	8	16	32	64
$q = 4$	1.31e+00	1.00e+00	9.83e-02	4.81e-03	2.50e-04	1.35e-05
$q = 6$	3.00e+00	1.42e-01	7.39e-03	2.13e-04	2.37e-06	9.67e-08
$q = 8$	6.23e-01	3.11e-02	7.22e-04	7.20e-06	7.37e-08	2.06e-10

Table 6.14
l_∞–norm errors for the solution of Example 1 *with the $\omega(t)$ varying in time as given in case* (iii)*, with $T = 1$.*

$N = M$	16	32	64	128	256	512
$q = 4$	1.49e+000	4.09e-02	7.65e-03	2.44e-04	1.42e-05	5.98e-07
$q = 6$	6.93e-02	3.71e-03	2.02e-04	2.21e-06	1.64e-08	6.48e-10
$q = 8$	8.73e-03	8.07e-04	2.70e-06	5.76e-08	2.79e-10	5.51e-13

Table 6.15
l_∞–norm errors for the solution of Example 2 *with the $\omega(t)$ varying in time as given in case* (i)*.*

N	2	4	8	16	32	64
$q = 4$	2.19e-02	5.24e-03	2.20e-03	5.55e-05	4.54e-06	4.15e-07
$q = 6$	2.29e-03	1.54e-03	1.05e-04	2.50e-06	5.39e-08	4.33e-10
$q = 8$	3.27e-03	5.04e-04	8.82e-06	9.07e-08	1.28e-10	7.83e-13

Table 6.16
l_∞–norm errors for the solution of Example 2 *with the $\omega(t)$ varying in time as given in case* (ii)*.*

N	4	8	16	32	64	128
$q = 4$	2.04e-02	1.58e-02	4.94e-04	3.47e-05	2.71e-06	8.48e-08
$q = 6$	1.29e-02	3.36e-03	2.56e-05	8.92e-07	1.00e-08	1.64e-10
$q = 8$	2.30e-03	4.54e-04	2.49e-06	6.40e-09	3.85e-11	2.54e-13

Table 6.17
l_∞–norm errors for the solution of Example 2 *with the $\omega(t)$ varying in time as given in case* (iii)*; the total time here is $T = 1$.*

N	32	64	128	256	512	1024
$q = 4$	2.54e-01	4.85e-03	5.52e-04	6.69e-05	6.41e-06	4.21e-07
$q = 6$	1.01e-01	7.57e-04	3.54e-05	7.14e-07	1.23e-08	2.10e-10
$q = 8$	2.07e-02	3.67e-05	1.85e-06	2.09e-09	1.24e-11	4.28e-14

tions:

$$\frac{\partial u}{\partial t} = \triangle u + 4\cos(2x)\cos(100t) - 100\cos(2x)\sin(100t) \quad \text{in } \omega(t), \tag{6.4}$$

$$\text{and} \quad u(x,0) = \cos(2x), \;\; u(b_k(t),t) = \cos(2b_k(t))\cos(100t). \tag{6.5}$$

The exact solution for this problem is given by $u(x,t) = \cos(2x)\cos(100t)$. We report l_∞ errors, computed as in the case of the other examples, in Tables 6.15, 6.16, and 6.17. Again, we observe optimal convergence rates.

7. Conclusions. We have presented an extension of the Greengard and Strain algorithm [6] for the fast solution of the heat equation with moving boundaries. Our main contribution is the introduction of the product integration quadrature and a scheme for moving boundaries. Our approach results in a high-order scheme in both space and time. We presented results for the Neumann and Dirichlet problems with distributed forces. In all cases the accuracy and efficiency of the proposed scheme is verified—even in the case of highly oscillatory boundary motions. Optimal complexity has been achieved by using the fast Gauss transform and nonuniform FFT. Also, our method can be extended to variable-coefficient and nonlinear problems by introducing a volume density for which we have to solve.

A shortcoming of our method is that it has a complex implementation. A second, more important drawback is that high-order accuracy can be achieved for sufficiently smooth data only (forces, boundary conditions, and boundary motion). For certain applications, in particular industrial ones, this can be a quite restrictive assumption. For general nonsmooth data the high-order properties of our scheme will be lost.

We are currently working on extending this work to higher dimensions. As we mentioned in the introduction, all algorithmic choices extend to higher dimensions: the product quadrature rules, the Chebyshev approximation (using adaptive, tensor products), the kernel splitting for the local and far parts, and the approximations in the case of prescribed boundaries. The most significant complication is the fact that the boundary of the target domain becomes an infinite-dimensional manifold. Boundary convolution with the heat kernel requires additional algorithmic machinery. Although one can outline the extensions of the method in higher dimensions, such a task is difficult when one wants to preserve work optimality and high accuracy. Finally, the assumption that f and w are defined everywhere is restrictive; in most practical applications the data will be defined only on a regular grid or at random points. Smooth extensions and design of special quadrature rules for such data are not trivial. We are currently working on addressing some of these issues, and we will report our work in a future paper.

Appendix A. Here we give recurrence relations to compute $I_k = \int_0^1 \sigma(\theta) T_k(2\theta - 1) d\theta$ for the singular kernels $\sigma_1(\theta) = \theta^{-\gamma}$ for any $\gamma \in (-\infty, 1)$ and $\sigma_2(\theta) = \log(\theta)$. These recurrences are unconditionally stable.

$$I_n = -\frac{n}{n+1-\gamma}\left(\frac{2}{n(n-2)} + 2I_{n-1} + \frac{n-3+\gamma}{n-2} I_{n-2}\right) \quad n > 2 \quad \text{for } \sigma_1(\theta). \tag{A.1}$$

$$I_n = \frac{n}{n+1}\left(C_n - 2I_{n-1} - \frac{n-3}{n-2} I_{n-2}\right) \quad n > 2 \quad \text{for } \sigma_2(\theta), \tag{A.2}$$

$$\text{where} \quad C_n = \begin{cases} 0 & n \text{ is odd,} \\ -\frac{6}{n^4 - 4n^3 + n^2 + 6n} & \text{otherwise.} \end{cases}$$

Appendix B.

Proof of the direct formulation. The Leibniz rule for differentiating the volume integral, also called Reynold's transport theorem, states that, for any function $q(x,t)$ defined in $\Omega(t)$,

$$\frac{d}{dt}\int_{\Omega(t)} q d\Omega = \int_{\Omega(t)} \frac{\partial q}{\partial t} d\Omega + \int_{\gamma(t)} q(v.n) d\gamma,$$

where v is the velocity of the boundary. Substituting $q = Gu$, we get

$$\begin{aligned}\frac{d}{dt}\int_{\Omega(t)} Gud\Omega &= \int_{\Omega(t)} \frac{\partial G}{\partial t}u + G\frac{\partial u}{\partial t} + \int_{\gamma(t)} Gu(v\cdot n)d\gamma\\ &= \int_{\Omega(t)} \frac{\partial G}{\partial t}u + G(\triangle u + b) + \int_{\gamma(t)} Gu(v\cdot n)\\ &= \int_{\Omega(t)} \left(\frac{\partial G}{\partial t} + \triangle G\right)u + Gb + \int_{\gamma(t)} \left(G\frac{\partial u}{\partial n} - u\frac{\partial G}{\partial n}\right) + Gu(v\cdot n).\end{aligned}$$

Integrating both sides from 0 to t and using $G(x, y, t, t) = 0$ for any t, we get

$$-\int_{\Omega_0} Gw = \int_0^t \int_{\Omega(\tau)} [-\delta(x,t)u + Gb] + \int_0^t \int_{\gamma(\tau)} \left(G\frac{\partial u}{\partial n} - u\frac{\partial G}{\partial n}\right) + Gu(v\cdot n),$$

$$\Rightarrow u(x,t) = \int_{\Omega_0} Gw + \int_0^t \int_{\Omega(\tau)} Gb\; + \int_0^t \int_{\gamma(\tau)} \left(G\frac{\partial u}{\partial n} - u\frac{\partial G}{\partial n}\right) + Gu(v\cdot n). \tag{B.1}$$

Jump conditions: (a) Defining $I(\beta_0) = \int_0^1 \frac{e^{-\frac{\beta_0^2}{\theta}}}{\sqrt{\theta}} f(\theta)\,d\theta$, we have to prove:

$$\lim_{\beta_0\to 0^-} I(\beta_0) = \lim_{\beta_0\to 0^+} I(\beta_0) = \int_0^1 \frac{1}{\sqrt{\theta}} f(\theta)\,d\theta. \tag{B.2}$$

Proof. Consider the sequence of functions g_n defined as $g_n = e^{-\frac{\beta_n^2}{\theta}}$ with the sequence β_n chosen such that $\lim_{n\to\infty}\beta_n = 0$. Clearly $|g_n| \leq 1$ and $g_n \to 1$ pointwise almost everywhere. The result follows by Lebesgue's dominated convergence theorem.

(b) Assuming that $f(\theta)$ is uniformly continuous in $\theta \in (0, 1)$, we need to prove:

$$\begin{gathered}\lim_{\beta_0\to 0^+} I(\beta_0) = \frac{1}{2}f(0);\\ \lim_{\beta_0\to 0^-} I(\beta_0) = -\frac{1}{2}f(0), \quad \text{where } I(\beta_0) = \frac{\beta_0}{2\sqrt{\pi}}\int_0^1 \frac{e^{-\frac{\beta_0^2}{\theta}}}{\sqrt{\theta^3}} f(\theta)\,d\theta.\end{gathered} \tag{B.3}$$ □

Proof. We split the integral into two parts: $I(\beta_0) = I_1(\beta_0)+I_2(\beta_0)$. The integrals I_1 and I_2 are defined as follows:

$$I_1 = \frac{\beta_0 f(0)}{2\sqrt{\pi}}\int_0^1 \frac{e^{-\frac{\beta_0^2}{\theta}}}{\sqrt{\theta^3}}\,d\theta = \frac{\beta_0}{2\,|\beta_0|}\mathrm{Erfc}(\beta_0)f(0). \tag{B.4}$$

$$\therefore \lim_{\beta_0\to 0^-} I_1(\beta_0) = -\frac{1}{2}f(0) \quad \text{and} \quad \lim_{\beta_0\to 0^+} I(\beta_0) = \frac{1}{2}f(0). \tag{B.5}$$

$$I_2 = \frac{1}{2\sqrt{\pi}}\int_0^1 \frac{e^{-\frac{\beta_0^2}{\theta}}}{\sqrt{\theta^3}}\left(f(\theta) - f(0)\right)d\theta = \frac{1}{\sqrt{\pi}}\int_{\beta_0}^{\infty} e^{-\sigma^2}\left(f\left(\frac{\beta_0^2}{\sigma^2}\right) - f(0)\right), \tag{B.6}$$

by the change of variables $\sigma = \frac{\beta_0}{\sqrt{\theta}}$. Now breaking the interval of integration into two parts $\left(\beta_0, \sqrt{\beta_0}\right)$ and $\left(\sqrt{\beta_0}, \infty\right)$, we see that as $\beta_0 \to 0$ the integral of the first interval vanishes. The integral of the second interval $\sigma \geq \sqrt{\beta_0} \Rightarrow |\frac{\beta_0^2}{\sigma^2}| \leq |\beta_0|$. Since $f(\theta)$ is uniformly continuous, we can choose δ such that $|f(\frac{\beta_0^2}{\sigma^2}) - f(0)| < \epsilon$ whenever $|\beta_0| < \delta$. Thus,

$$\lim_{\beta_0\to 0} |I_2(\beta_0)| < \lim_{\epsilon\to 0} \frac{\epsilon}{2\sqrt{\pi}}\int_{\sqrt{\beta_0}}^{\infty} e^{-\sigma^2}\,d\sigma = 0. \tag{B.7}$$ □

REFERENCES

[1] B. K. Alpert, *Hybrid Gauss-trapezoidal quadrature rules*, SIAM J. Sci. Comput., 20 (1999), pp. 1551–1584.

[2] K. Brattkus and D. I. Meiron, *Numerical simulations of unsteady crystal growth*, SIAM J. Appl. Math., 52 (1992), pp. 1303–1320.

[3] C. W. Clenshaw and A. R. Curtis, *A method for numerical integration on an automatic computer*, Numer. Math., 2 (1960), pp. 197–205.

[4] A. Dutt and V. Rokhlin, *Fast Fourier transforms for nonequispaced data*, SIAM J. Sci. Comput., 14 (1993), pp. 1368–1393.

[5] L. Greengard and P. Lin, *Spectral approximation of the free–space heat kernel*, Appl. Comput. Harmon. Anal., 9 (2000), pp. 83–97.

[6] L. Greengard and J. Strain, *A fast algorithm for the evaluation of heat potentials*, Comm. Pure Appl. Math., XLIII (1990), pp. 949–963.

[7] L. Greengard and J. Strain, *The fast Gauss transform*, SIAM J. Sci. Comput., 12 (1991), pp. 79–94.

[8] T. Hasegawa and T. Torii, *An automatic quadrature for cauchy principal value integrals*, Math. Comp., 194 (1991), pp. 741–754.

[9] J. Huang, M.-C. Lai, and Y. Xiang, *An integral equation method for epitaxial step-flow growth simulations*, J. Comput. Phys., 216 (2006), pp. 724–743.

[10] M. T. Ibanez and H. Power, *An efficient direct BEM numerical scheme for phase change problems using Fourier series*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 2371–2402.

[11] J. Ma, V. Rokhlin, and S. Wandzura, *Generalized Gaussian quadrature rules for systems of arbitrary functions*, SIAM J. Numer. Anal., 33 (1996), pp. 971–996.

[12] S. Kapur and V. Rokhlin, *High-order corrected trapezoidal quadrature rules for singular functions*, SIAM J. Numer. Anal., 34 (1997), pp. 1331–1356.

[13] R. Kress, *Linear Integral Equations*, Appl. Math. Sci., Springer, New York, 1999.

[14] P. P.-Y. Lin, *On the Numerical Solution of the Heat Equation in Unbounded Domains*, Ph.D. thesis, New York, NY, 1993.

[15] R. Piessens and M. Branders, *Numerical-solution of integral-equations of mathematical physics, using Tschebyscheff polynomials*, J. Comput. Phys., 21 (1976), pp. 178–196.

[16] R. Piessens, *Computing integral transforms and solving integral equations using Chebyshev polynomial approximations*, J. Comput. Appl. Math., 121 (2000), pp. 113–124.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, New York, NY, 1992.

[18] T. J. Rivlin, *Chebyshev Polynomials*, Wiley–Interscience, New York, 1990.

[19] V. Rokhlin, *End-point corrected trapezoidal quadrature rules for singular functions*, Comput. Math. Appl., 20 (1990), pp. 51–62.

[20] J. A. Sethian and J. Strain, *Crystal-growth and dendritic solidification*, J. Comput. Phys., 98 (1992), pp. 231–253.

[21] J. Strain, *Linear-stability of planar solidification fronts*, Phys. D, 30 (1988), pp. 297–320.

[22] J. Strain, *Fast potential theory.* II. *Layer potentials and discrete sums*, J. Comput. Phys., 99 (1992), pp. 251–270.

[23] J. Strain, *Fast adaptive methods for the free–space heat equation*, SIAM J. Sci. Comput., 15 (1994), pp. 185–206.

[24] L. N. Trefethen, *Spectral Methods in Matlab*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.

[25] J. Wimp, *Computation with Recurrence Relations*, Pitman, Boston, 1984.

[26] N. Yarvin and V. Rokhlin, *Generalized Gaussian quadratures and singular value decompositions of integral operators*, SIAM J. Sci. Comput., 20 (1998), pp. 699–718.

[27] L. Ying, G. Biros, and D. Zorin, *A high-order 3D boundary integral equation solver for elliptic PDEs in smooth domains*, J. Comput. Phys., 219 (2006), pp. 247–275.

[28] M. Zerroukat and L. C. Wrobel, *A boundary element method for multiple moving boundary problems*, J. Comput. Phys., 138 (1997), pp. 501–519.