FAST INTEGRAL EQUATION METHODS FOR THE MODIFIED HELMHOLTZ EQUATION

Bryan D. Quaife[†,*]**,Mary-Catherine A. Kropinski**[†]

[†]Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada
[*]Email: bquaife@math.sfu.ca

Talk Abstract

We present an efficient integral equation method 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. We first discretize in time, which is known as Rothe's method, resulting in a non-homogeneous modified Helmholtz equation that is solved at each time step. We formulate the solution to this equation as the sum of a volume potential and a double layer potential. Both potentials are solved using the Fast Multipole Method (FMM) resulting in a $\mathcal{O}(N)$ method where N is the total number of discretization points on the boundary and in the domain. We demonstrate our approach on the heat equation and the Allen-Cahn equation.

Introduction

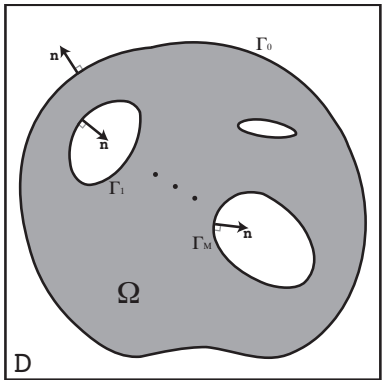

Figure 1: *A bounded $(M+1)$-ply connected domain Ω is embedded in the unit square D. The outer boundary is denoted by Γ_0, the interior components by $\Gamma_1, \ldots, \Gamma_M$, and Γ is the union of all such curves. The unit normal $\mathbf{n}$ points out of Ω on each component of the curve.*

Consider a $(M+1)$-ply connected bounded domain Ω with boundary Γ which is comprised of individual smooth component curves Γ_k (see Figure 1). Many problems in physics and engineering require solving the forced heat equation

$$u_t(\mathbf{x}) - \Delta u(\mathbf{x}) = F(\mathbf{x}, u, t), \qquad \mathbf{x} \in \Omega, \qquad (1a)$$
$$u(\mathbf{x}, t) = f(\mathbf{x}, t), \qquad \mathbf{x} \in \Gamma, \qquad (1b)$$
$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \qquad \mathbf{x} \in \Omega. \qquad (1c)$$

Possible applications include the homogeneous heat equation ($F(\mathbf{x}, u, t) = 0$), reaction-diffusion problems such as the brusselator, or energy minimization problems such as the Allen-Cahn equation ($F(\mathbf{x}, u, t) = u(1-u^2)$).

The long-term research goal is to develop a general solver for (1). The desired features include:

- High-order temporal schemes.
- General multiply-connected geometries will be easily handled.
- Solutions will be based entirely on integral formulations, resulting in a well-conditioned method.
- High-order quadrature will be used for achieving high spatial accuracy.
- A FMM-accelerated solution procedure is used to achieve optimal or near-optimal efficiency.

Solution Procedure

Contrary to the method of lines, Rothe's method first discretizes in time. In order to prevent a severe time-step restriction, Implicit-Explicit (IMEX) methods [1] are used to march in time. Such schemes treat the diffusive term implicitly, while the remaining terms are treated explicitly. Regardless of the choice of IMEX scheme, the temporal discretization of (1) yields the modified Helmholtz equation:

$$(1 - \alpha^2 \Delta) u^{N+1} = B, \qquad \mathbf{x} \in \Omega, \qquad (2a)$$
$$u^{N+1} = f, \qquad \mathbf{x} \in \Gamma, \qquad (2b)$$
$$u^0 = u_0, \qquad \mathbf{x} \in \Omega, \qquad (2c)$$

where $B = B(\mathbf{x}, t^N, \ldots, t^{N-p+1}, u^N, \ldots, u^{N-p+1})$ depends only on the p previous time steps. The simplest such scheme is the first-order backward Euler method

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

A second-order method is extrapolated Gear

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

At each time step, we represent the solution of (2) as

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

Here, U^P is any solution of

$$(1 - \alpha^2\Delta)U^p = B, \quad \mathbf{x} \in \Omega, \tag{3}$$

and U^h satisfies

$$(1 - \alpha^2\Delta)U^h = 0, \qquad \mathbf{x} \in \Omega, \tag{4a}$$

$$U^h = g, \qquad \mathbf{x} \in \Gamma, \tag{4b}$$

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

Volume Potential

The fundamental solution of 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. We can now form a solution of (3) as a volume integral:

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

A Fast Multipole Solver is discussed in [2], but it requires the domain to be the unit square $D := [1/2, 1/2]^2$. In order to use this solver, we embed Ω into D and define

$$\widetilde{B}(\mathbf{x}) = \begin{cases} B(\mathbf{x}), & \mathbf{x} \in \Omega, \\ B_{ext}(\mathbf{x}), & \mathbf{x} \in \Omega \backslash D. \end{cases}$$

We then solve

$$\widetilde{U^p}(\mathbf{x}) = \frac{1}{2\pi\alpha^2} \int_\Omega \widetilde{B}(\mathbf{y}) K_0\left(\frac{|\mathbf{y}-\mathbf{x}|}{\alpha}\right) dA_\mathbf{y}, \tag{5}$$

and let U^p be the restriction of $\widetilde{U^p}$ to Ω. Different methods for defining $B_{ext}(\mathbf{x})$ have been considered. The most naive manner is to let $B_{ext}(\mathbf{x}) = 0$. However, this leads to a severe loss in accuracy of U^p. What is required is a method to smoothly extend a function from Ω to D. This research is ongoing.

Layer Potential

The solution of (4) is written as the double layer potential

$$U^h(\mathbf{x}) = \frac{1}{2\pi\alpha^2} \int_\Gamma \frac{\partial}{\partial \mathbf{n}_\mathbf{y}} K_0\left(\frac{|\mathbf{y}-\mathbf{x}|}{\alpha}\right) \sigma(\mathbf{y}) ds_\mathbf{y},$$

where $\sigma(\mathbf{y})$ is the value of an unknown density function at the boundary point $\mathbf{y}$, and $\partial/\partial\mathbf{n}_\mathbf{y}$ is the outward normal derivative at $\mathbf{y}$. The density function σ is found by solving an integral equation derived to ensure the boundary condition (4b) is satisfied.

In [3], we derive the necessary integral equation

$$g(\mathbf{x}) = -\frac{1}{2\alpha^2}\sigma(\mathbf{x}) + \frac{1}{2\pi\alpha^2} \int_\Gamma \frac{\partial}{\partial \mathbf{n}_\mathbf{y}} K_0\left(\frac{|\mathbf{y}-\mathbf{x}|}{\alpha}\right) \sigma(\mathbf{y}) ds_\mathbf{y}. \tag{6}$$

The above kernel is continuous along Γ,

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

where $\kappa(\mathbf{x})$ denotes the curvature of Γ at the point $\mathbf{x}$. As a note, many of the results concerning the formulations and solutions of the integral equation (6) follow directly from standard potential theory because of the series expansion [4]

$$K_0(z) = p(z)\log(z) + q(z),$$

where p and q are polynomials.

Summarizing, the kernel of the integral equation (6) is bounded and continuous, and the integral operator is therefore compact. In addition, there are nontrivial homogeneous solutions. By the Fredholm alternative, (6) has a unique solution for any integrable data $g(\mathbf{x})$.

Numerical Methods

In [2], methods for rapid evaluation of (5) are discussed. Here, we only outline the method. Once a suitable extension $\widetilde{B}(\mathbf{x})$ is constructed, we construct an adaptive quad-tree 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 by subdividing a box (or node) s at level l into four equal parts; these are called the children of s and s is called the parent. Adaptivity is achieved by allowing different levels of refinement throughout the tree. A node s is subdivided if the error of a third-order polynomial interpolating $\widetilde{B}$ is larger than a preset tolerance.

We denote the childless boxes in the quad-tree as $D_i, i = 1, \ldots, P$, where P is the total number of such nodes. We assume that we are given $\widetilde{B}$ on a cell-centered 4×4 grid for each D_i. To obtain fourth-order accuracy,

these 16 function values are used to construct the third-order polynomial

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

where $\mathbf{x}^i$ is the center of D_i and $\{p_j\}$ are the standard basis functions for polynomials of order three. Then, we approximate (5) by

$$\widetilde{U}^p(\mathbf{x}) \approx \sum_{i=1}^{P} \int_{D_i} G(\mathbf{y}-\mathbf{x}) \sum_{j=1}^{10} c_j^i p_j(\mathbf{y}-\mathbf{x}^i) dA_{\mathbf{y}}.$$

By precomputing convolutions of the fundamental solution against polynomials, and using the FMM, $\widetilde{U}^p(\mathbf{x})$ is computed with $\mathcal{O}(N_D)$ operations where $N_D = 16 \times P$. A direct method would require $\mathcal{O}(N_D^2)$ operations.

We now discuss numerical methods to solve (6). We assume that each component curve $\Gamma_k, k = 0, \ldots, M$ is parameterized by $\mathbf{y}^k(\alpha)$, where $\alpha \in [0, 2\pi)$. Similarly, $\sigma^k(\alpha)$ refers to the restriction of the density function σ to Γ_k. We discretize Γ_k at N points equispaced with respect to α resulting in a total number of discretization points $N_\Gamma = (M+1)N$. Associated with each point, denoted by $\mathbf{y}_j^k$, is the unknown density function σ_j^k.

In order to approximate the integral in (6), we use hybrid Gauss-trapezoidal quadrature rules developed by Alpert [5] which are tailored for integrands with logarithmic singularities. These quadrature rules are of the order $h^p \log h$. The order p determines the weights u_n and nodes $v_n, n = 1, \ldots, l$, 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 this interval, the quadrature rule is exactly the trapezoid rule. Applying the quadrature rule to (6) yields

$$\begin{aligned} -2\alpha^2 g(\mathbf{y}_j^k) &= \sigma_j^k \\ &- \frac{h}{\pi} \left\{ \sum_{\substack{m=0 \\ m \neq k}}^{M} \sum_{n=1}^{N} K(\mathbf{y}_n^m, \mathbf{y}_j^k) \sigma_n^m \right\} \\ &- \frac{h}{\pi} \left\{ \sum_{n=j+a}^{N+j-a} K(\mathbf{y}_n^k, \mathbf{y}_j^k) \sigma_n^k \right\} \\ &- \frac{1}{\pi} \sum_{\substack{n=-l \\ n \neq 0}}^{l} u_{|n|} K(\mathbf{y}^k_{j+\frac{n}{|n|}v_{|\mathbf{n}|}}, \mathbf{y}_j^k) \sigma^k_{j+\frac{n}{|n|}v_{|\mathbf{n}|}}, \end{aligned} \tag{7}$$

where

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

We invoke periodicity of all functions on Γ_k by defining $j + N = j$. In the final sum, we require values of σ intermediate to the nodal values. These are found using Fourier interpolation which requires performing $2l$ Fast Fourier Transforms.

Equation (7) is a dense linear system that is solved iteratively using the generalized minimum residual method (GMRES). The bulk of the work at each iteration lies in evaluating (7) at the current solution update. Directly, this would require $\mathcal{O}(N_\Gamma^2)$ operations. This evaluation can be reduced to $\mathcal{O}(N_\Gamma)$ operations by using the FMM [2], [3], [6].

Numerical Results

The algorithms described above have been implemented in Fortran. The tolerance for the residual error in GMRES is 10^{-11} and the FMM expansions guarantee 12 digits accuracy.

EXAMPLE 1. In this example, we solve the forced heat equation on a domain for which an analytic 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. The forcing term is chosen so that the exact solution is

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

where J_0 and Y_0 are the zeroth-order Bessel functions of the first and second kind. We have chosen λ so that the time-dependent term vanishes on both boundaries $\lambda \approx 10.244$. We march through time using the IMEX Euler and extrapolated Gear methods, up to $t = 0.01$. The results are summarized in Table 1.

Δt	Error_1	Error_2
2.0×10^{-3}	1.37×10^{-2}	1.65×10^{-3}
1.0×10^{-3}	7.09×10^{-3}	4.74×10^{-4}
5.0×10^{-4}	3.59×10^{-3}	1.23×10^{-4}
2.5×10^{-4}	1.78×10^{-4}	3.25×10^{-5}

Table 1: *Temporal Error using IMEX Euler (Error_1) and extrapolated Gear (Error_2); we achieve first and second order convergence, respectively.*

EXAMPLE 2. In this example, we demonstrate that our methods can be applied to much more complicated equations. Here, we solve the Allen-Cahn equation

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

where $\epsilon = 10^{-5}$. We initialize the solution with random variables uniformly on $[-\frac{1}{2}, \frac{1}{2}]$. The general behaviour of solutions of the Allen-Cahn equation is well known: the stable stationary solutions are $u = 1$ and $u = -1$ and the solutions exhibits coarsening towards these values. The physical boundaries can create more complex patterns as can be seen in Figure 2.

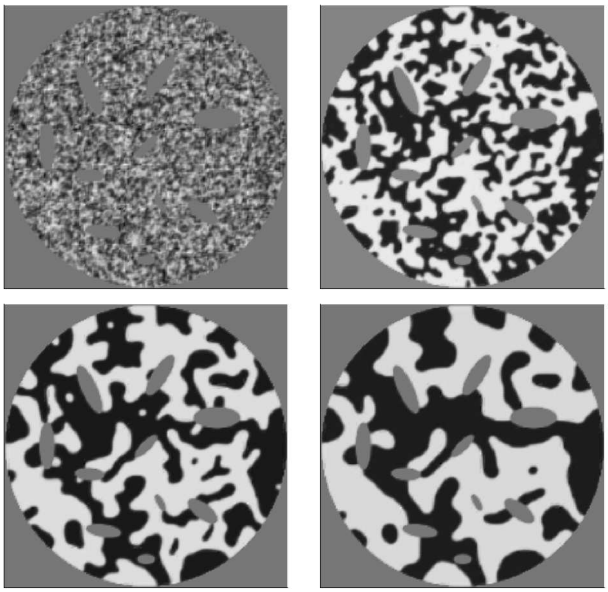

Figure 2: *The solution to the Allen-Cahn equation in a domain with 10 interior component curves. Here,* $N_\Gamma = 2816$, $N_D = 65,536$, *100 time steps of size one are taken for a total CPU time of 19.5 minutes. The first picture shows the initial conditions, and the final picture is at* $t = 100$.

Conclusions

We have presented a coupling of fast algorithms with integral equation formulations for the modified Helmholtz equation. The purpose is to develop general solvers for the isotropic, nonlinear heat equation. We have demonstrated the methods on a forced linear heat equation as well as the Allen-Cahn equation.

The long-term goal is to extend this work to more complicated problems including multiple variables (brusselator) and higher order partial differential equations (incompressible Navier-Stokes). However, the issues that must be addressed before this can be investigated include:

- The extension $\widetilde{B}$ of B throughout the computational domain D for general domains remains an open problem. Under investigation are methods that use a local interpolation and methods that solve a suitable integral equation inside each Ω_k.
- In order to appropriately resolve solution features that appear or disappear within the domain, it may be necessary to dynamically generate the quad-tree throughout the simulation.
- The integral operator of the double layer potential becomes singular at grid points close to Γ. Currently a method discussed by Biros et. al. [7] is being implemented.

References

[1] U.M. Ascher, S.J. Ruuth, and B.M. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32:797–823, 1995.

[2] H. Cheng, J. Huang, and T. Leiterman. An Adaptive Fast Solver for the Modified Helmholtz Equation in Two Dimensions. *Journal of Computational Physics*, 211:616–637, 2005.

[3] M.C. Kropinski and B.D. Quaife. Fast integral equation methods for the modified Helmholtz equation. *Journal of Computational Physics*, 230:425–434, 2011.

[4] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. Dover, New York, 1964.

[5] B. Alpert. Hybrid Gauss-Trapezoidal Quadrature Rules. *SIAM Journal on Scientific Computing*, 20:1551–1584, 1999.

[6] L. Greengard and V. Rokhlin. A Fast Algorithm for Particle Simulations. *Journal of Computational Physics*, 73:325–348, 1987.

[7] G. Biros, L. Ying, and D. Zorin. A high-order 3D boundary integral equation solver for elliptic PDEs in smooth domains. *Journal of Computational Physics*, 219:247–275, 2006.