UCLA

COMPUTATIONAL AND APPLIED MATHEMATICS

A Recursive Expanding Domain Method for the Solution of Laplace's Equation In Infinite Domains

Christopher R. Anderson

April 2014

CAM Report 14-45

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

Abstract

In this paper we describe a discrete Fourier transform based numerical procedure to evaluate a solution of Laplace's equation in R^2 or R^3 at points in a rectangular computational region. The numerical procedure is a recursive implementation of an "expanding domain" type method where one obtains approximations of increasing accuracy by expanding the computational domain. The method presented here leads to approximations that converge with high order rates of convergence with respect to domain size. Spectrally accurate approximations are used to approximate differential operators and so the method possess very high rates of convergence with respect to mesh size as well. The computational work and memory requirements of the recursive implementation are much lower than the standard implementation of an expanding domain method. The computational cost is $O[\eta K \log(\eta K)]$ in two dimensions and $O[\eta^2 K \log(\eta^2 K)]$ in three dimensions where K is the total number of grid points and η is the non-dimensional expansion factor used in the computation. The memory requirements are reduced to $O[\eta K]$ for both two and three dimensional problems. Computational results are presented that demonstrate the accuracy and computational efficiency of the procedure.

0.1 Introduction

In this paper we present a numerical technique for the evaluation of the solution of Laplace's equation

$$\Delta u = f \qquad x \in \mathrm{R}^{\mathrm{N}} \tag{1}$$

for $\mathrm{N} = 1, 2, 3$ at the grid points associated with a uniform discretization of a rectangular subregion $\Omega_0 \subset \mathrm{R}^{\mathrm{N}}$ with edges of characteristic length L_0. It is assumed that f vanishes outside of Ω_0. The procedure for $\mathrm{N} = 2, 3$ described here is an alternate method of implementing the high order expanding domain procedure described in [2]. The background motivation for the expanding domain procedure and references to other methods for solving Laplace's equation in infinite domains are given in [2].

The procedure in [2] essentially consists of creating an approximate solution of Laplace's equation in an infinite domain by modifying the right hand side of the equation so that moments up to order m vanish, solving the modified problem on an extended periodic domain, and then adding to the modified problem solution an analytically evaluated solution component to compensate for the modification of the right hand side. High order spatial accuracy with respect to the mesh size is obtained because of the implicit use of spectral approximations of the differential operators, and high order convergence rates with respect to the non-dimensional domain expansion factor η is obtained by using the moment canceling modifications of the right hand side. Here $\eta = (L/L_0)$ where L_0 is a measure of the original domain size and L is the size of the expanded domain.

When a domain expansion factor η of modest size provides an acceptably accurate solution, the simplicity of the implementation described in [2] and the use of high performance FFT's allows one to easily construct efficient implementations. However, when the expansion factor η is larger, an alternate implementation is sought because the computational cost and memory footprint of the procedure described in [2] scales as approximately η^N. The procedure described here improves the CPU scaling by a factor of η so that the computational work scales approximately linearly in two dimensions and approximately quadratically in three dimensions. The memory requirements are much improved and they scale linearly in η for *both* two and three dimensional problems. The spatial accuracy of the procedure presented here and the accuracy obtainable with a given value of η can be expected to be either identical or a bit better than that of the implementation in [2]. In addition to better scaling with respect to domain size, implementations can be created that effectively exploit multi-core processors because the bulk of the computational work consists of loops over independent tasks.

The alternate implementation presented here is a recursive procedure because for each dimension N, the computational task is reduced to combining solutions of a collection of infinite domain Helmholtz problems in dimension $N - 1$. What's fortunate about the use of a recursive procedure is that when $N = 1$ the infinite domain problem can be solved "exactly", e.g. the only error is a spatial discretization error and no errors are introduced due to the use of a finite sized domain. In the first section we present the key idea behind our alternate implementation. In the second section we describe the solution procedure for the infinite domain Helmholtz problem when $N = 1$. In the following section the alternate implementations of the procedures of [2] are presented for two and three dimensional problems. In the final section we give computational results that demonstrate the effectiveness of the approach for both reducing the computational cost and the memory footprint of the

method.

0.2 Recursive Expanding Domain Procedure

The expanding domain method presented in [2] is a numerical discretization of the following procedural steps to evaluate the solution of

$$\Delta u = f \qquad x \in \mathrm{R^N} \tag{2}$$

for points in Ω_0.

Step(i). Create a modification function $\tilde{f}(\vec{x})$ so that the moments of $\vec{g}(\vec{x}) = f(\vec{x}) - \tilde{f}(\vec{x})$ vanish up to mth order;

$$\int_{\mathrm{R}^N} g(\vec{x})\, \vec{x}^{\,\alpha} = 0 \qquad |\alpha| = 0, 1, \ldots m \tag{3}$$

where $\alpha = \alpha_1 \alpha_2 \ldots \alpha_N$ is a multi-index, $|\alpha| = \alpha_1 + \alpha_2 + \cdots \alpha_N$, and $\vec{x}^{\,\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_N^{\alpha_N}$.

Step(ii). Evaluate the solution to

$$\Delta v = g \quad \vec{x} \in \mathrm{R}^N \tag{4}$$

for values of $\vec{x} \in \Omega_0$.

Step(iii). Create the solution $u(\vec{x})$ to (2) by setting

$$u(\vec{x}) = v(\vec{x}) + w(\vec{x}) \tag{5}$$

for $\vec{x} \in \Omega_0$, where $w(\vec{x})$ is the solution to $\Delta w = \tilde{f}$ in R^N. It is assumed that the form of $\tilde{f}(\vec{x})$ is chosen so that an analytic representation of w is readily available.

The expanding domain aspect of the method arises in Step (ii), when discrete Fourier transforms (DFT's) on an expanding domain are used to approximate a continuous Fourier transform solution on an infinite domain. As discussed in [2], the modification of the right hand side in Step (i) is implemented so that the values of v obtained with an expanding domain approximation converge rapidly as the domain size increases. The computational tasks required of discrete implementations of Step (i) and Step (iii) only involve data in Ω_0 and thus the increase in computational work and required memory as the domain is expanded is solely due to the increase in work and memory requirements of discrete forward and inverse Fourier transforms. The method presented in [2] thus has computational work that scales as $(\eta^N K)\log(\eta^N K)$ and memory requirements that scale as $\eta^N K$ where K is the total number of grid points in Ω_0 and η is the non-dimensional expansion factor.

The method we are proposing is an alternate Fourier transform based method for evaluating the solution required in Step (ii). Specifically, it is a method for evaluating the solution of

$$\Delta v = g \quad \vec{x} \in \mathrm{R}^N \tag{6}$$

in a region Ω_0 with the assumption that g vanishes outside Ω_0 and that g satisfies the moment conditions of Step (i) up to some order $m > 0$. This alternate procedure is designed so that when the continuous procedure is discretized, the computational work scales as $(\eta^{(N-1)}K)\log(\eta^{(N-1)}K)$ and the memory requirements scale as ηK.

We first describe the general idea behind the Fourier transform based procedure for $N = 2$. If $\hat{g}(\xi_1, \xi_2)$ is the Fourier transform of $g(x, y)$ then a solution to (6) is given by

$$v(x,y) = \frac{1}{4\pi^2}\int_\infty^\infty \int_\infty^\infty -\frac{\hat{g}(\xi_1,\xi_2)}{\xi_1^2+\xi_2^2}\,\mathrm{e}^{i\,y\,\xi_2}\;d\xi_2\;\;\mathrm{e}^{i\,x\,\xi_1}\,d\xi_1 \tag{7}$$

This solution can also be expressed as

$$v(x,y) = \frac{1}{2\pi}\int_\infty^\infty \bar{v}(\xi_1,y)\;\mathrm{e}^{i\,x\,\xi_1}\,d\xi_1 \tag{8}$$

where $\bar{v}(\xi_1, y)$ is given by

$$\bar{v}(\xi_1,y) = \frac{1}{2\pi}\int_\infty^\infty -\frac{\hat{g}(\xi_1,\xi_2)}{\xi_1^2+\xi_2^2}\,\mathrm{e}^{i\,y\,\xi_2}\;d\xi_2 \tag{9}$$

The alternate procedure arises by observing that for any value of ξ_1, $\bar{v}(\xi_1, y)$ is also determined as the solution to an infinite domain Helmholtz problem in y. Specifically,

$$\frac{d^2\bar{v}(\xi_1,y)}{dy^2} - \xi_1^2\,\bar{v}(\xi_1,y) = \bar{g}(\xi_1,y) \quad \text{for } y \in [-\infty,\infty] \tag{10}$$

where $\bar{g}(\xi_1, y)$ is given by

$$\bar{g}(\xi_1,y) = \int_\infty^\infty g(\xi_1,y)\,\mathrm{e}^{-i\,x\,\xi_1}\,d\xi_1 \tag{11}$$

Since $g(x, y)$ is assumed to vanish outside Ω_0, $\bar{g}(\xi_1, y) = 0$ for all $y \notin [-L_0, L_0]$.

The solution of (6) required by Step (ii) can thus be obtained by carrying out the following sub-steps

2D-Step(ii)(a). For each y in $[-L_0, L_0]$ evaluate

$$\bar{g}(\xi_1,y) = \int_\infty^\infty g(\xi_1,y)\,\mathrm{e}^{-i\,x\,\xi_1}\,d\xi_1 \tag{12}$$

2D-Step(ii)(b). For each ξ_1 solve

$$\frac{d^2\bar{v}(\xi_1,y)}{dy^2} - \xi_1^2\,\bar{v}(\xi_1,y) = \bar{g}(\xi_1,y) \quad \text{for } y \in [-\infty,\infty] \tag{13}$$

2D-Step(ii)(c). Evaluate

$$v(x,y) = \frac{1}{2\pi}\int_\infty^\infty \bar{v}(\xi_1,y)\;\mathrm{e}^{i\,x\,\xi_1}\,d\xi_1 \tag{14}$$

for $(x, y) \in \Omega_0$.

When discretized, this alternate procedure for evaluating (6) leads to a reduction in computational work and memory allocation because only approximations to one dimensional continuous Fourier transforms are required. The approximation of the Fourier transform in the other direction of the method in [2] is replaced by the solution of an infinite domain Helmholtz problem in one dimension.

When $N = 3$, the alternate procedure for evaluating (6) in Step (ii) consists of

3D-Step(ii)(a). For each $(y, z) \in [-L_0, L_0] \times [-L_0, L_0]$ evaluate

$$\bar{g}(\xi_1, y, z) = \int_{\infty}^{\infty} g(\xi_1, y, z)\, \mathrm{e}^{-i\,x\,\xi_1}\, d\xi_1 \tag{15}$$

3D-Step(ii)(b). For each ξ_1 solve

$$\frac{d^2\bar{v}(\xi_1, y, z)}{dy^2} + \frac{d^2\bar{v}(\xi_1, y, z)}{dz^2} - \xi_1^2\, \bar{v}(\xi_1, y, z) = \bar{g}(\xi_1, y, z) \quad \text{for } (y, z) \in [-\infty, \infty] \times [-\infty, \infty] \tag{16}$$

3D-Step(ii)(c). Evaluate

$$v(x, y, z) = \frac{1}{2\pi} \int_{\infty}^{\infty} \bar{v}(\xi_1, y)\ \mathrm{e}^{i\,x\,\xi_1}\, d\xi_1 \tag{17}$$

for $(x, y, z) \in \Omega_0$.

As with the two dimensional procedure, when discretized, a reduction in computational work and memory allocation occurs because only approximations to one dimensional continuous Fourier transforms are required. The procedure for obtaining the solutions of the two dimensional infinite domain Helmholtz problems is a minor modification of the expanding domain procedure for the two dimensional Laplace equation.

In the discrete approximation of these alternate evaluation procedures, the one dimensional transforms are approximated by discrete one dimensional transforms on an expanding domain. The rate of convergence of such approximations with respect to domain size is dictated by the differentiability of $\bar{v}(\xi_1, y)$ or $\bar{v}(\xi_1, y, z)$ with respect to ξ_1. Of particular concern is the differentiability in the neighborhood of $\xi_1 = 0$. As can be inferred from (9) in two dimensions and the corresponding formula in three dimensions, the number of bounded derivatives of these functions with respect to ξ_1 at $\xi_1 = 0$ is determined by the number of moments of g that vanish. Thus the satisfaction of the moment condition in Step (i) gives rise to the rapid convergence of the expanding domain approximation as m increases.

The recursive nature of the procedure arises because for dimension N it is necessary to evaluate the solution of an infinite domain Helmholtz equation in dimension $N - 1$. We therefore start with a description of a approximation procedure for the infinite domain Helmholtz equation in $N = 1$ and then follow this with the descriptions of the approximation procedure for $N = 2$ and $N = 3$.

0.3 The Infinite Domain Problem in One Dimension

For one dimensional problems of the form

$$\frac{d^2u}{dx^2} - \gamma^2\, u = f(x) \qquad x \in \mathrm{R} \tag{18}$$

where $\gamma \geq 0$, and where the support of f is contained the interval $[-L_0, L_0]$ there is a solution procedure which has the property that the change in solution values when computed using any expanded domain ($L > L_0$) is of the order of errors introduced by a spectral discretization of the solution. Thus, a spectrally accurate solution approximation can be created using the original computational domain. This procedure is essentially a one-dimensional version of the procedure in [5].

The computational procedure is based upon an analytical procedure in which the solution is obtained by solving two problems in sequence and then combining the results. The first step consists of creating v, a function that solves the following problem

$$\begin{aligned} \frac{d^2v}{dx^2} - \gamma^2 v &= f(x) \qquad x \in [-L, L], \\ v(-L) &= v(L) = 0 \end{aligned} \tag{19}$$

If v is extended by zero to $[\infty, \infty]$, the resulting function satisfies Laplace's equation for all points except at the endpoints of the interval, where it can possess a discontinuous derivative. The second step is to add to v a correction function u_c so that the sum of the two is the desired solution. The required correction is a solution of (18) with $f \equiv 0$ and one that possesses a discontinuous derivatives at the endpoints of the interval that are the opposite of those associated with the extension of v. This correction component is just a linear combination of the Green's functions centered at the endpoints e.g. when $\gamma \neq 0$,

$$u_c = \left(\frac{dv}{dx}|_{x=-L}\right) \frac{1}{2\gamma} \mathrm{e}^{-\gamma|x-L|} - \left(\frac{dv}{dx}|_{x=L}\right) \frac{1}{2\gamma} \mathrm{e}^{-\gamma|x+L|} \tag{20}$$

and when $\gamma = 0$

$$u_c = -\left(\frac{dv}{dx}|_{x=-L}\right) |x-L| + \left(\frac{dv}{dx}|_{x=L}\right) |x+L| \tag{21}$$

Setting $u = v + u_c$ provides the desired solution of (18).

The computational procedure consists of first solving (19) using discrete sine transforms followed by the addition of the harmonic correction in which the required derivatives are evaluated using spectral differentiation. We assume that the interval $[-L, L]$ has been discretized using a grid with M panels and associated grid points $x_j = -L + jh$, $h = \frac{2L}{M}$ and $j = 0 \ldots M$. It is also assumed that $f(-L) = f(L) = 0$.

The discrete sine transform of an approximate solution to (19) is given by

$$\hat{v}(k) = -\frac{\hat{f}(k)}{\left(\frac{\pi^2 k^2}{4L^2} + \gamma^2\right)} \quad k = 1 \ldots M-1 \tag{22}$$

where $\hat{f}(k)$ is the discrete sine transform of the values $f(m)$, $m = 1 \ldots M-1$. The application of the inverse discrete sine transform to these coefficients yields a spectrally accurate approximation to the values $v(m)$ for $m = 1 \ldots M-1$.

To the values $v(m)$, the correction solution of the form (20) or (21) is added. When $\gamma \neq 0$ the cost of evaluating the exponential functions at each grid function can be avoided by just evaluating $\mathrm{e}^{-\gamma h}$ and then accumulating the correction terms recursively using the fact that $\mathrm{e}^{-\gamma x_{i+1}} = \mathrm{e}^{-\gamma h} \mathrm{e}^{-\gamma x_i}$.

The evaluation of the required derivative of v at the endpoints of the interval are computed by evaluating

$$\frac{dv}{dx}|_{x_j} \approx \sum_{k=1}^{M-1} \hat{v}(k) \left(\frac{k\pi}{2L}\right) \cos(\frac{k\pi(x_j+L)}{2L}) \tag{23}$$

$x_0 = -L$ and $x_M = L$. At the endpoints of the interval the cosine function is ± 1, so no trigonometric functions evaluations are required in the evaluation of the sums.

0.4 Recursive Expanding Domain Procedure in Two and Three Dimensions

We next consider the task of evaluating a discrete approximation to the values of the solution of the infinite domain problem

$$\Delta u + \gamma^2 u = f \qquad \vec{x} \in \mathrm{R}^2 \tag{24}$$

at points in $\Omega_0 = [-L_x^0, L_x^0] \times [-L_y^0, L_y^0]$. Let M_x^0 and M_y^0 be the number of panels in a discretization of Ω_0 in the x and y directions respectively. The mesh widths in each direction are then given by $h_x = \dfrac{M_x^0}{2L_x^0}$ and $h_y = \dfrac{M_y^0}{2L_y^0}$. In the case when $\gamma = 0$, we assume that a value, m, of the maximal order of the moment matching condition is specified.

For a given value of the non-dimensional expansion factor η, let Ω_L be the expanded domain of approximate width $L_x = \eta L_x^0$ in the x-direction obtained by adding panels in the x-direction of width h_x on both sides of the domain and increasing L_x as needed to insure an integral number of panels are added. For simplicity of exposition, we assume that an equal number of panels have been added in each direction so that $\Omega_L = [-L_x, L_x] \times [-\frac{L_y^0}{2}, \frac{L_y^0}{2}]$ with the total number of panels in the x-direction given by $M_x = M_x^0 + 2\frac{(L_x - L_x^0)}{hx}$. In practice the number of panels added to each side need not be equal; all that is required is that Ω_0 be approximated centered within Ω_L.

If $u(m,n)$ is used to designate the array of values associated with the discretization of Ω_L, the computational task is to determine approximate values $u(m,n) \approx u(x_m, y_n)$ for all $(x_m, y_n) \in \Omega_0$. In the following description the one dimensional discrete forward and inverse Fourier transform of the values in the x-direction over $[-L_x, L_x]$ are designated as DFT_{D_x} and $\mathrm{DFT}_{D_x}^{-1}$ where $D_x = M_x h_x$. The discrete operator described in Section 2 that approximates the solution of Helmholtz equation in the y-direction, $\dfrac{d^2\bar{v}}{dy^2} - \beta\bar{v} = \bar{g}$, with "infinite" boundary conditions, is designated as $\mathrm{H}_{\beta,y}^{-1}$.

The computational procedure is a discrete implementation of the Steps (i)-(iii) with a minor modification when $\gamma \neq 0$ and the use of (12)-(14) to approximate the solution required in Step (ii);

2D-(i) If $\gamma \neq 0$ set $g(m,n) = f(m,n)$ for (m,n) such that $(x_m, y_n) \in \Omega_0$. If $\gamma = 0$ use the procedure described in [2] to create $\tilde{f}(m,n)$ so that moments of $g = f - \tilde{f}$ vanish up to order m. Extend $g(m,n)$ to have zero value on all grid points outside of Ω_0.

2D-(ii)(a) Compute the forward one-dimensional discrete Fourier transform in the x-direction of $g(m,n)$ for each value transverse grid point y_n,

$$\begin{aligned} &\text{for } n = 0 \dots M_y^0 \\ &\quad \bar{g}(m,n) = \left[\mathrm{DFT}_{L_x}(g(:,n))\right]_m \qquad \text{for} \qquad m = 0 \dots M_x \end{aligned}$$

2D-(ii)(b) Evaluate the solution of the one-dimensional Helmholtz equation with "infinite" boundary conditions for each x-direction discrete Fourier coefficient, $\bar{v}(p,:)$,

$$\begin{aligned} &\text{for } p = -[\tfrac{M_x}{2}] \dots [\tfrac{(M_x-1)}{2}] \\ &\quad D_x = M_x h_x, \quad \beta = \left(\tfrac{4\pi^2 p^2}{D_x^2} + \gamma^2\right) \\ &\quad \bar{v}(p,n) = \left[\mathrm{H}_{\beta,y}^{-1}\left(\bar{g}(p,:)\right)\right]_n \qquad \text{for} \qquad n = 0 \dots M_y^0 \end{aligned} \tag{25}$$

2D-(ii)(c) Compute the inverse one-dimensional discrete Fourier transform in the x-direction of $\bar{v}(p,n)$ for each value transverse grid point y_n,

$$\begin{aligned} &\text{for } n = 0 \dots M_y^0 \\ &\quad v(m,n) = \left[\mathrm{DFT}_{L_x}^{-1}(\bar{v}(:,n))\right]_m \qquad \text{for} \qquad m = 0 \dots M_x \end{aligned}$$

2D-(iii) If $\gamma \neq 0$ set $u(m,n) = v(m,n)$ for $(x_m, y_n) \in \Omega_0$. If $\gamma = 0$ set $u(m,n) = v(m,n) + w(m,n)$ for $(x_m, y_n) \in \Omega_0$ where w is the analytically evaluated solution to $\Delta w = \tilde{f}$ with "infinite" boundary conditions.

The reason one need not alter the right hand side when $\gamma \neq 0$ is due to the fact that when $\gamma \neq 0$ the integrand of the inverse transform is non-singular at $||\vec{\xi}_1|| = 0$, and thus one gets rapid convergence of the discrete inverse transform approximation as the domain size increases. If $K = M_x^0 M_y^0$ is the total number of grid points in Ω_0, then the computational work of the above procedure is $\mathrm{O}[\eta K \log(\eta K)]$ and the memory required is $\mathrm{O}(\eta K)$, since the domain is expanded in only one direction.

The discrete approximation procedure for three dimensional problems is entirely analogous to the two dimensional procedure. The required forward and inverse discrete Fourier transforms are still one-dimensional and are performed in the x-direction for each transverse grid point (y_n, z_r). However, instead of a solution to the one-dimensional Helmholtz equation in (25), a solution of the two-dimensional Helmholtz equation is required:

3D-(ii)(b) Evaluate the solution of the two-dimensional Helmholtz equation with "infinite" boundary conditions for each x-direction discrete Fourier coefficient, $\bar{v}(p,:,:)$,

$$\begin{aligned} &\text{for } p = -[\tfrac{M_x}{2}] \dots [\tfrac{(M_x-1)}{2}] \\ &\quad D_x = M_x h_x, \quad \beta = \left(\tfrac{4\pi^2 p^2}{D_x^2} + \gamma^2\right) \\ &\quad \bar{v}(p,n,r) = \left[\mathrm{H}_{\beta,y,z}^{-1}\left(\bar{g}(p,:,:)\right)\right]_{(n,r)} \qquad\qquad n = 0 \dots M_y^0 \quad r = 0 \dots M_z^0 \end{aligned} \tag{26}$$

$H^{-1}_{\beta,y,z}$ designates the solution of Helmholtz equation with "infinite" boundary conditions in the (y,z)-directions,

$$\frac{d^2\bar{v}}{dy^2} + \frac{d^2\bar{v}}{dz^2} - \beta\bar{v} = \bar{g} \tag{27}$$

If $K = M_x^0 M_y^0 M_z^0$ is the total number of points in Ω_0 for the three dimensional domain, the computational work is $O[\eta^2 K \log(\eta^2 K)]$. Even though the domain is only expanded in one direction, the extra factor of η arises from the use of an expanded domain procedure to solve the required two-dimensional Helmholtz equations. However, the extra memory required to obtain any of the two-dimensional solutions is $O[\eta \frac{K}{M_z^0} \log(\eta \frac{K}{M_z^0})]$, and thus contributes a negligible amount to the total memory requirements. The total memory requirements therefore scale as $O(\eta K)$.

Unlike the the two-dimensional case, the accuracy of the solution of the required lower dimensional Helmholtz equations will depend on the expansion factor used for it's computation. For simplicity of implementation, we've chosen the expansion factor used for the two-dimensional problem to be identical to that used for the three-dimensional problem. We expect that further computational efficiency improvements could be made by altering this choice for the Helmholtz problems for high wave numbers.

0.5 Computational Results

To enable the comparison of the recursive expanding domain procedure, with the expanding domain procedure in [2], the test problem used was identical to that in [2]. This test problem consists of determining the values of the solution to

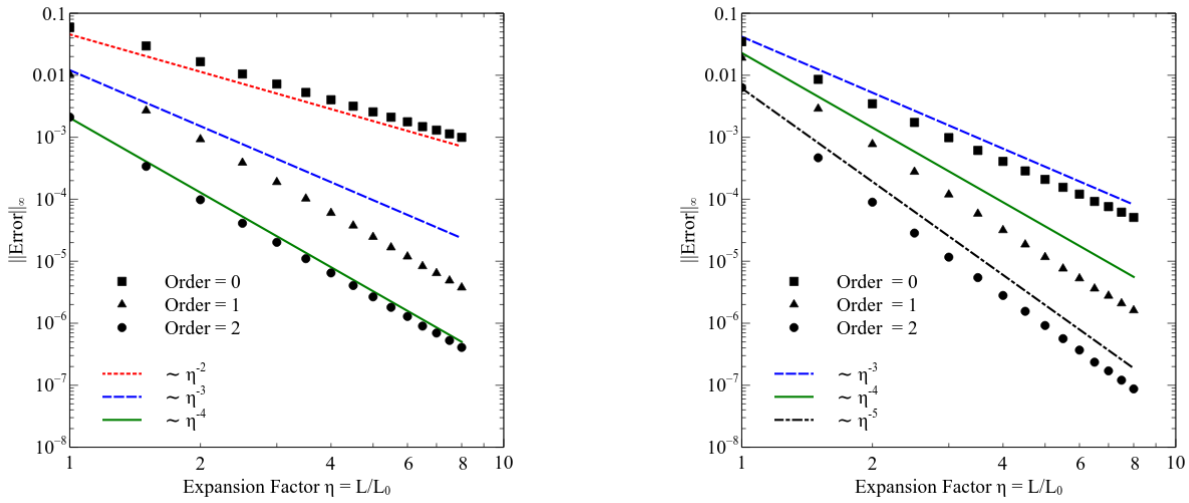
$$\Delta u = f \quad \vec{x} \in \mathrm{R}^N \tag{28}$$

in the region $\vec{x} \in \Omega_0 = [-L_0, L_0]^N$ with $L_0 = 1$ and $N = 2, 3$. The computational grid used in Ω_0 was taken to be a uniform grid with M panels in each direction. The function f was chosen to be a linear combination of two mollifiers $B_\delta(r)$ as described in [1] with width $\delta = 0.6$, specifically

$$f(\vec{x}) = B_{\delta=.6}(\vec{x} - \vec{x}_A) + B_{\delta=.6}(\vec{x} - \vec{x}_B) \tag{29}$$

where $\vec{x}_A = (.11, .22, (.33))$ and $\vec{x}_B = (-.33, -.22, -(.11))$. A mollifier exponent $q = 9$ was specified. This choice of exponent leads to a potential u that is 10 times continuously differentiable. In the construction of the moment matching function $\tilde{f}$ of (3) required for Step (i), the location and width of the mollifiers must be chosen. It is advantageous to use mollifiers with as large a width as possible so $\tilde{f}$ and it's transform can be accurately represented with a coarse mesh. In all of the computational results, the moment matching function was constructed using mollifiers and their derivatives centered at the origin and of width $\delta = 0.9$.

The discrete Fourier transform computations were carried out using FFTW3 routines [3] [4]. A reduction by a factor of two in the computational work was obtained by exploiting the fact that the procedure involves the transform of real data, and thus one need only solve for the coefficients in (25) and (26) for non-positive induces and obtain the positive indexed coefficients by conjugation.

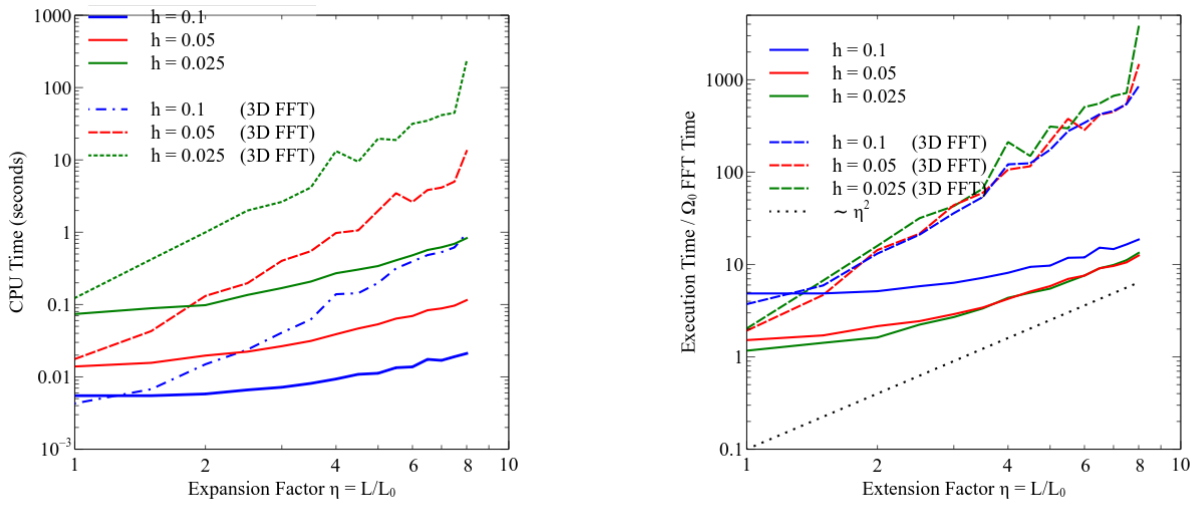


(a) (b)

Figure 1: The behavior of the error in $\Omega_0 = [-L, L] \times [-L_0, L_0]^{N-1}$ as the domain is expanded for different orders, $m = 0, 1, 2$, of the moment matching modification. A mesh width of $\frac{1}{40}$ was used in each direction. (a) N= 2 (b) N = 3.

The first set of computational results concerns the behavior of the error in the potential as the computational domain used, $[-L, L] \times [-L_0, L_0]^{N-1}$, is increased in size (e.g. as $\eta = \frac{L}{L_0}$ is increased). In this computation, the number of panels M used in each direction of Ω_0 was fixed at 80, e.g. the mesh width was fixed at $\frac{1}{40}$. This mesh width was sufficiently small to insure that the solution values were essentially converged with respect to mesh size. A value of $q = 7$ was used as the mollifier exponent for the construction of the moment correction function. In Figure 1 the relative errors in the potential evaluated in the maximum norm are presented for values of $\eta = 1 \ldots 8$ for two and three dimensional computations. The results demonstrate a well defined rate of convergence, a rate that increases by one with each increase in the maximal order of the moments used. In particular, the observed rate of convergence approximately $\eta^{-(m+2)}$ in two dimensions and $\eta^{-(m+3)}$ in three dimensions.

In Figure 2, we show the computational time for the three dimensional test problem with different mesh sizes as the domain is expanded. Second order moment matching, $m = 2$, was used and the exponent of the mollifier used in the moment matching function was taken to be $q = 7$. The results are given in units of CPU seconds (a) and in FFT units (b). The CPU seconds are those of a desktop machine with an AMD FX-8120 eight-core processor with multi-threading execution obtained using OpenMP. The FFT unit of time is the time required for one forward and one inverse transform of the data values in Ω_0. In these plots, the CPU time for the method presented here and the expanding domain method of [2] (3D FFT) are both given. The benefits of using the recursive procedure are quite clear — at the finest resolution the computational time is reduced by almost three orders of magnitude.



(a) (b)

Figure 2: Three dimensional test problem computational time. Time in CPU seconds (a) and Time in FFT units (b). Ω_0 FFT Time is the time for one forward and one inverse discrete Fourier Transform applied to the values in Ω_0. Timing data for the procedure of [2] is designated as 3D FFT.

The reduced memory requirements of the recursive expanding domain method are revealed by the results given in Figure 3. In this figure we show the memory required as the domain is expanded for different mesh widths for the three dimensional test problem. As expected, there is linear growth in the memory required for the recursive method and cubic growth in the memory required for the procedure in [2]. For modest extension factors, a cubic growth rate of required memory may be acceptable, but as the domain is expanded further, the memory requirements quickly become unacceptable and the use of a recursive procedure becomes a necessity.

The last set of computational results concerns the behavior of the error in the potential with respect to the use of a finite mesh size, e.g. the discretization error. In Figure 4 we show the behavior of the errors in the potential of the three dimensional test problem with respect to decreasing mesh size for several values of the mollifier exponent q. Second order moment matching $m = 2$ and an expanded domain size corresponding to $\eta = 6$ were used. For this value η, the finite domain size contribution to the total error is $O(10^{-7})$, so that over the range of mesh widths where the mesh width error dominates, the results clearly indicate a rapid convergence with respect to mesh size. The rate of convergence increases as the differentiability of the moment matching function increases. In fact for $q = 9$, the rate is approximately $O(h^{-12})$. For many problems, as well as the particular test problem considered here, one finds little difference between the convergence for rates when $q \geq 7$.

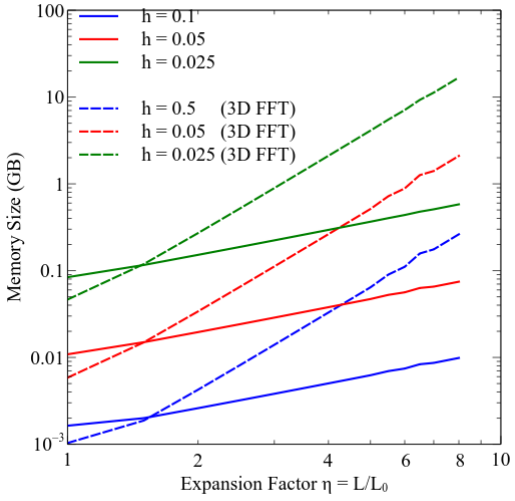


Figure 3: Three dimensional test problem memory size. Data for the procedure of [2] is designated as 3D FFT.

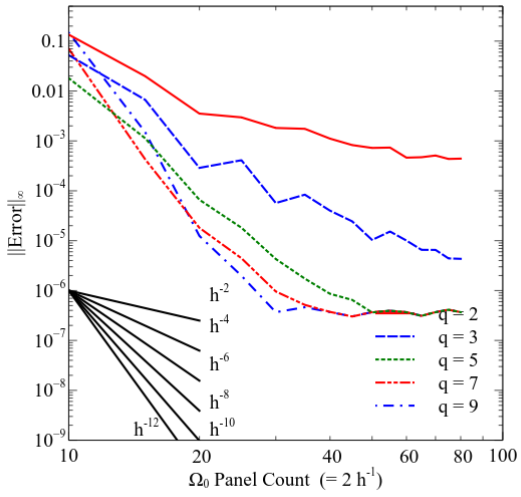


Figure 4: Maximal relative potential error in $\Omega_0 = [-1, 1]^3$ for different computational mesh sizes $h = \frac{2}{\Omega_0 \text{ Panel Count}}$ and expansion factor $\eta = 6$.

0.6 Conclusion

In the simple expanding domain procedure for computing approximate solutions of Laplace's equation in infinite domains, one computes the solution by modifying the right hand side to have zero average value, solves a periodic problem on an extended computational domain, and then adds to that solution a component to account for the modification of the right hand side. The accuracy of the approximation is increased by both refining the computational mesh and expanding the size of the computational domain. In [2], an improved version of this simple procedure was presented that ostensibly consists of adding a specially constructed modification function to the right hand side before one solved a periodic problem. In this paper we present an alternate implementation of the procedure in [2] where the solution of the infinite domain Laplace equation in R^N is reduced to a collection of computational tasks consisting of one dimensional discrete Fourier transforms and the solution of infinite domain Helmholtz equations in R^{N-1}. This fact is exploited recursively so that, for example, the three dimensional problem is reduced to solving infinite domain two dimensional problems, which in turn are solved using infinite domain one dimensional problems. The consequences of using a recursive strategy is a computational cost that is $\mathrm{O}[\eta K \log(\eta K)]$ in two dimensions and $\mathrm{O}[\eta^2 K \log(\eta^2 K)]$ in three dimensions where K is the total number of grid points and η is the non-dimensional expansion factor used in the computation. Perhaps more importantly, the memory requirements of the procedure are reduced to $\mathrm{O}[\eta K]$ for both two and three dimensional problems.

The method presented here shares the same convergence behavior with respect to exten-

sion factor and mesh size as that of [2]. Specifically, the use of the moment modification function leads to high order rates of reduction of the finite domain size errors as the domain size is increased. Our computational results demonstrate that if one uses a modification function that matches the moments of the right hand side to second order the rate of convergence with respect to domain size is $O(\eta^{-4})$ for two dimensional problems and $O(\eta^{-5})$ for three dimensional problems where η is the non-dimensional expansion factor. The use of highly differentiable moment matching functions leads to discretizations which do not adversely effect the high order rates of convergence that are obtainable with spectral differential operator approximations. The method presented here readily extends to the computation of any derivatives of the solution.

Bibliography

[1] Christopher R. Anderson. Compact Polynomial Mollifiers For Poisson's Equation. Technical Report CAM-14-43, Department of Mathematics, UCLA, Los Angeles, California, May 2014.

[2] Christopher R. Anderson. High Order Expanding Domain Methods for the Solution of Laplace's Equation In Infinite Domains. Technical Report CAM-14-44, Department of Mathematics, UCLA, Los Angeles, California, May 2014.

[3] Matteo Frigo and Steven G. Johnson. The Design and Implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[4] Matteo Frigo and Steven G. Johnson. FFTW 3.3.4. http://www.fftw.org/, 2014.

[5] R.A James. The solution of poisson's equation for isolated source distributions. *Journal of Computational Physics*, 25(2):71–93, 1977.