

*Communications in
Applied
Mathematics and
Computational
Science*

COMPUTATION OF VOLUME POTENTIALS
ON STRUCTURED GRIDS
WITH THE METHOD OF LOCAL CORRECTIONS

CHRIS KAVOUKLIS AND PHILLIP COLELLA

vol. 14 no. 1 2019

COMPUTATION OF VOLUME POTENTIALS ON STRUCTURED GRIDS WITH THE METHOD OF LOCAL CORRECTIONS

CHRIS KAVOUKLIS AND PHILLIP COLELLA

We present a new version of the method of local corrections (MLC) of McCorquodale, Colella, Balls, and Baden (2007), a multilevel, low-communication, noniterative domain decomposition algorithm for the numerical solution of the free space Poisson's equation in three dimensions on locally structured grids. In this method, the field is computed as a linear superposition of local fields induced by charges on rectangular patches of size $O(1)$ mesh points, with the global coupling represented by a coarse-grid solution using a right-hand side computed from the local solutions. In the present method, the local convolutions are further decomposed into a short-range contribution computed by convolution with the discrete Green's function for a Q -th-order accurate finite difference approximation to the Laplacian with the full right-hand side on the patch, combined with a longer-range component that is the field induced by the terms up to order $P - 1$ of the Legendre expansion of the charge over the patch. This leads to a method with a solution error that has an asymptotic bound of $O(h^P) + O(h^Q) + O(\epsilon h^2) + O(\epsilon)$, where h is the mesh spacing and ϵ is the max norm of the charge times a rapidly decaying function of the radius of the support of the local solutions scaled by h . The bound $O(\epsilon)$ is essentially the error of the global potential computed on the coarsest grid in the hierarchy. Thus, we have eliminated the low-order accuracy of the original method (which corresponds to $P = 1$ in the present method) for smooth solutions, while keeping the computational cost per patch nearly the same as that of the original method. Specifically, in addition to the local solves of the original method we only have to compute and communicate the expansion coefficients of local expansions (that is, for instance, 20 scalars per patch for $P = 4$). Several numerical examples are presented to illustrate the new method and demonstrate its convergence properties.

MSC2010: 65N06, 65N12, 65N15, 68W10.

Keywords: Poisson solver, method of local corrections, Mehrstellen stencils, domain decomposition, parallel solvers.

1. Introduction

We are interested in solving Poisson's equation with infinite domain boundary conditions in three dimensions, that is

$$\begin{aligned} \Delta\phi &\equiv \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = f && \text{in } \mathbb{R}^3, \\ \phi(\mathbf{x}) &= -\frac{1}{4\pi\|\mathbf{x}\|} \int_{\mathbb{R}^3} f(\mathbf{y}) d\mathbf{y} + o\left(\frac{1}{\|\mathbf{x}\|}\right), && \|\mathbf{x}\| \rightarrow \infty, \end{aligned} \quad (1)$$

where f is a function with bounded support and by $\|\cdot\|$ we denote the Euclidean norm. It is well known that problem (1) has a solution if f is Hölder continuous and has compact support Ω [12]. Furthermore, the solution of (1) is unique by means of a maximum principle argument for harmonic functions and is given as a convolution of the data with the three-dimensional infinite domain Green's function [10]

$$\phi(\mathbf{x}) = \int_{\Omega} G(\mathbf{x} - \mathbf{y}) f(\mathbf{y}) d\mathbf{y} \equiv (G * f)(\mathbf{x}), \quad G(\mathbf{z}) = -\frac{1}{4\pi\|\mathbf{z}\|}. \quad (2)$$

In addition, if $\Omega \subset B(\mathbf{x}_0, R)$, where $B(\mathbf{x}_0, R)$ is the closed ball of radius R centered at point $\mathbf{x}_0$, then ϕ is harmonic in $\mathbb{R}^3 \setminus B(\mathbf{x}_0, R)$ and hence real analytic. By differentiating (2), we find that the derivatives of the potential are rapidly decaying functions of the form

$$(\nabla^p \phi)(\mathbf{x}) = O\left(\left(\frac{1}{\|\mathbf{x} - \mathbf{x}_0\|}\right)^{\|p\|_1+1} R^3 \|f\|_{\infty}\right). \quad (3)$$

This suggests a domain decomposition strategy, in which the contribution to the fields on each local domain is computed independently and the nonlocal coupling is computed using a reduced number of computational degrees of freedom. This approach has been exploited for particle methods with the right-hand side in (1) given by $f(\mathbf{x}) = \sum_i q_i \delta(\mathbf{x} - \mathbf{x}_i)$. For instance, we mention the Barnes–Hut algorithm [6], the fast multipole method (FMM) [13; 7; 14], and the method of local corrections (MLC) [3; 1; 2]. The aforementioned particle algorithms have been modified to handle gridded data. In [20] the approximate solution of the Poisson problem is given as a discrete convolution of the discrete Green's function with the charge [15] and is computed efficiently by combining the fast Fourier transform (FFT) with interpolation of the kernel. This strategy is substantially accelerated within an FMM setting but has not been extended to support multiresolution calculations. A very attractive kernel-independent method is discussed in [27], for the case of a smooth charge distribution that is represented on a uniform mesh. The kernel is truncated on a sphere that encloses the charge so that its Fourier transform is C^∞ . This allows for a fast and accurate computation of the potential via the trapezoid rule and the FFT transform; however, the method is not readily applicable to adaptive mesh refinement

(AMR) hierarchies. The first kernel-independent, adaptive volume-FMM method has been presented in [19]. The integral in (2) is computed directly with numerical quadrature, and local charges are approximated with polynomials. The analogs of the multipole and local expansions in the original FMM method are convolutions with equivalent source densities defined on auxiliary surfaces that encompass octree boxes. The method can handle nonuniform sources, and a Chebyshev grid may be required to achieve high-order accuracy. A highly optimized parallel implementation is discussed in [21; 22]. For a comprehensive review that includes benchmark studies of the FFT, FMM, and multigrid methods, we refer to [11]. The MLC method [25] relies upon a localization approach that takes advantage of the rapid decay in truncation error of compact finite difference Laplace operators. Further, it is more compatible with traditional AMR. As such, it can be coupled with numerical schemes that require solving Poisson’s equation on nested locally refined grids, for instance adaptive projection methods for computational fluid dynamics. It should also be noted that MLC exhibits a good balance between computation and communication, which is essential for simulations on the emerging exascale platforms.

The present work is based on the extension of the method of local corrections to structured grid data described in [5; 4; 25]. In this approach, the support of the right-hand side is discretized with a rectangular grid, which is decomposed into a set of cubic patches. For two levels the method proceeds in three steps: (i) a loop over the fine disjoint patches and the computation of local potentials induced by the charge restricted to those patches on sufficiently large extensions of their support (downward pass), (ii) a global coarse-grid Poisson solve with a right-hand side computed by applying the coarse-grid Laplacian to the local potentials of step (i), and (iii) a correction of the local solutions computed in step (i) on the boundaries of the fine disjoint patches based on interpolating the global coarse solution from which the contributions from the local solutions have been subtracted (upward pass). These boundary conditions are propagated into the interior of the patches by performing Dirichlet solves on each patch. This can be generalized by replacing the global coarse solution in (ii) by a recursive call to MLC, or by replacing uniform grids at each level covering the entire domain by nested block-structured locally refined grids. The local volume potentials are computed using a high-order finite difference approximation to the Laplacian, combined with an extension to three dimensions of the James–Lackner algorithm [17; 18] for representing infinite domain boundary conditions. Furthermore, in order to make the nested refinement version of this algorithm practical, we require that $R = O(H) = O(h)$, where R is the radius (in max norm) of local patches, H the coarse mesh spacing, and h the fine-mesh spacing (i.e., a fixed number of points per patch and a fixed refinement ratio). In [25], the local field calculation in (i) was split into two contributions: one that represented the field induced by the complete charge distribution on a patch,

and a second corresponding to the monopole component of the charge. By using such a splitting, it is possible to obtain a convergent method by using a relatively large region for computing the monopole component only while keeping the overall computation and communications cost low. However, the convergence properties of the resulting method were erratic, and exhibited a large $O(h)$ solution error for smooth charge distributions that were well resolved on the fine grid.

The starting point for the present work is a new error analysis for the MLC algorithm that suggests a number of generalizations of the method that have better and more predictable convergence properties. For example, we replace the separate treatment of the monopole component of the charge on each patch by a similar treatment of a truncated expansion in Legendre polynomials of the charge distribution on each patch. Our error analysis predicts an $O(h^P) + O(h^Q) + O(\epsilon h^2) + O(\epsilon)$ solution error, where $P - 1$ is the maximum degree of the polynomials in the Legendre expansions, and Q is the order of accuracy of the finite difference discretization used to compute the local potentials. This is consistent with the earlier results in [25] corresponding to $P = 1$. The $O(\epsilon)$ term is a localization error, proportional to the max norm of the charge divided by a localization distance (measured in units of the number of coarse grid points across the patch) raised to the order of accuracy of the discretized Laplacian on harmonic functions. We also change the detailed approach to computing the local potentials, replacing the James–Lackner representation of the infinite domain boundary conditions in the calculation of the local potentials in step (i) with local discrete convolutions computed using FFTs via a variation on Hockney’s domain-doubling method [16]. This leads to a conceptually simpler algorithm, and provides a compact numerical kernel on which to focus the effort of optimization.

In this paper, we focus on the design of the algorithm, including an error analysis of the method and calculations that demonstrate the error properties derived from that analysis. In a second paper [24], we will present performance and parallel scaling results on high-performance computing (HPC) platforms.

2. Mehrstellen discretization and finite difference localization

Notation. We denote by $D^h, \Omega^h, \dots \subset \mathbb{Z}^3$ grids with grid spacing h of discrete points in physical space: $\{\mathbf{g}^h : \mathbf{g} \in D^h\}$. Arrays of values defined over such sets will approximate functions on subsets of $\mathbb{R}^3$; i.e., if $\psi = \psi(\mathbf{x})$ is a function on $D \subset \mathbb{R}^3$, then $\psi^h[\mathbf{g}] \approx \psi(\mathbf{g}h)$. We denote operators on arrays over grids of mesh spacing h by $L^h, \Delta^h, \dots$; $L^h(\phi^h) : D^h \rightarrow \mathbb{R}$. Such operators are also defined on functions of $\mathbf{x} \in \mathbb{R}^3$, and on arrays defined on finer grids $\phi^{h'}$, $h = Nh'$, $N \in \mathbb{N}_+$, by sampling: $L^h(\phi) \equiv L^h(\mathcal{J}^h(\phi))$, $\mathcal{J}^h(\phi)[\mathbf{g}] \equiv \phi(\mathbf{g}h)$, and $L^h(\phi^{h'}) \equiv L^h(\mathcal{J}^h(\phi^{h'}))$, $\mathcal{J}^h(\phi^{h'})[\mathbf{g}] \equiv \phi^{h'}[N\mathbf{g}]$.

For a rectangle $D = [\mathbf{l}, \mathbf{u}]$, defined by its lower-left and upper-right corners $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^3$, we define two operators: a grid extension operator

$$\mathcal{G}(D, r) = [\mathbf{l} - (r, r, r), \mathbf{u} + (r, r, r)], \quad r \in \mathbb{Z},$$

and a grid coarsening operator

$$\mathcal{C}(D) = \left[\left\lfloor \frac{\mathbf{l}}{N_{\text{ref}}} \right\rfloor, \left\lceil \frac{\mathbf{u}}{N_{\text{ref}}} \right\rceil \right].$$

Throughout this paper, we will use $N_{\text{ref}} = 4$ for the refinement ratio between consecutive levels.

We begin our discussion presenting the finite difference discretizations of (1) that we will be using throughout this work and some of their properties that pertain to the method of local corrections. Specifically, we are employing Mehrstellen discretizations [8] (also referred to as compact finite difference discretizations) of the three-dimensional Laplace operator

$$(\Delta^h \phi^h)_g = \sum_{s \in [-s, s]^3} a_s \phi_{g+s}^h, \quad a_s \in \mathbb{R}. \quad (4)$$

If ϕ^h is defined on D^h , then $\Delta^h \phi^h$ is defined on $D^{h,s} \equiv \mathcal{G}(D^h, -s)$. The associated truncation error $\tau^h \equiv (\Delta^h - \Delta)(\phi) = -\Delta^h(\phi^h - \phi)$ for the Mehrstellen discrete Laplace operator is of the form

$$\tau^h(\phi) = C_2 h^2 \Delta(\Delta\phi) + \sum_{q'=2}^{q/2-1} h^{2q'} \mathcal{L}^{2q'}(\Delta\phi) + h^q L^{q+2}(\phi) + O(h^{q+2}), \quad (5)$$

where q is even and $\mathcal{L}^{2q'}$ and L^{q+2} are constant-coefficient differential operators that are homogeneous, i.e., for which all terms are derivatives of orders $2q'$ and $q+2$, respectively. For the two operators we will consider here, $C_2 = \frac{1}{12}$. In general, the truncation error is $O(h^2)$. However, if ϕ is harmonic in a neighborhood of $\mathbf{x}$,

$$\tau^h(\phi)(\mathbf{x}) = \Delta^h(\phi)(\mathbf{x}) = h^q L^{q+2}(\phi)(\mathbf{x}) + O(h^{q+2}). \quad (6)$$

In our numerical test cases we make use of the 19-point (L_{19}^h) and 27-point (L_{27}^h) Mehrstellen stencils [26] that are described in Appendix A, for which $q = 4$ and $q = 6$, respectively. In general, it is possible to define operators for which $s = \lfloor q/4 \rfloor$ for any even q , using higher-order Taylor expansions and repeated applications of the identity

$$\frac{\partial^{2r} \phi}{\partial x_d^{2r}} = \frac{\partial^{2r-2}}{\partial x_d^{2r-2}}(\Delta\phi) - \sum_{d' \neq d} \frac{\partial^{2r}}{\partial x_{d'}^{2r-2} \partial x_d^2}(\phi).$$

Since we are primarily concerned with solving the free-space problem, the corresponding discrete problem can be expressed formally as a discrete convolution

$$(G^h * f^h) = (\Delta^h)^{-1}(f^h), \quad (G^h * f^h)[\mathbf{g}] \equiv \sum_{\mathbf{g}' \in \mathbb{Z}^3} h^3 G^h[\mathbf{g} - \mathbf{g}'] f[\mathbf{g}']^h, \quad (7)$$

where the discrete Green's function $G^h[\mathbf{g}] = h^{-1} G^{h=1}[\mathbf{g}]$ satisfies

$$(\Delta^{h=1} G^{h=1})[\mathbf{g}] = \begin{cases} 1 & \text{if } \mathbf{g} = \mathbf{0}, \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

and

$$G^{h=1}[\mathbf{g}] = \frac{1}{4\pi \|\mathbf{g}\|} + o\left(\frac{1}{\|\mathbf{g}\|}\right), \quad \|\mathbf{g}\| \rightarrow \infty.$$

We use these conditions to construct approximations to G^h numerically; see [Appendix A](#). For any n , we have

$$\sum_{\mathbf{g} \in D} h^3 |G^h[\mathbf{g}]| \leq C, \quad C = C(nh), \quad D \subseteq [-n, \dots, n]^3,$$

from which it follows that convolution with G^h is max norm stable on bounded domains, i.e.,

$$\|G^h * f^h\|_\infty \leq C' \|f^h\|_\infty,$$

$$C' \text{ independent of } f, h, \quad \text{supp}(f^h) \subseteq \left[-\left\lfloor \frac{A}{h} \right\rfloor, \dots, \left\lceil \frac{A}{h} \right\rceil\right]^3, \quad (9)$$

for any fixed $A > 0$.

The form of the truncation error (5) allows us to compute q -th-order accurate solutions to (1) by modifying the right-hand side, i.e.,

$$\Delta^h(\phi) = \tilde{f}^h + O(h^q), \quad (10)$$

$$\tilde{f}^h = f^h + \left(C_2 h^2 (\Delta(f))^h + \sum_{q'=2}^{q/2-1} h^{2q'} \mathcal{L}^{2q'}(f)^h \right), \quad (11)$$

and replacing the differential operators on the right-hand side with finite difference approximations. If only a fourth-order accurate solution is required, it suffices to use the first term, leading to a correction of a particularly simple form:

$$\phi = G^h * f^h + C_2 h^2 f^h + O(h^4). \quad (12)$$

In particular, the solution error $\epsilon^h = G^h * f^h - \phi = O(h^4)$ away from the support of f without any modification of f^h .

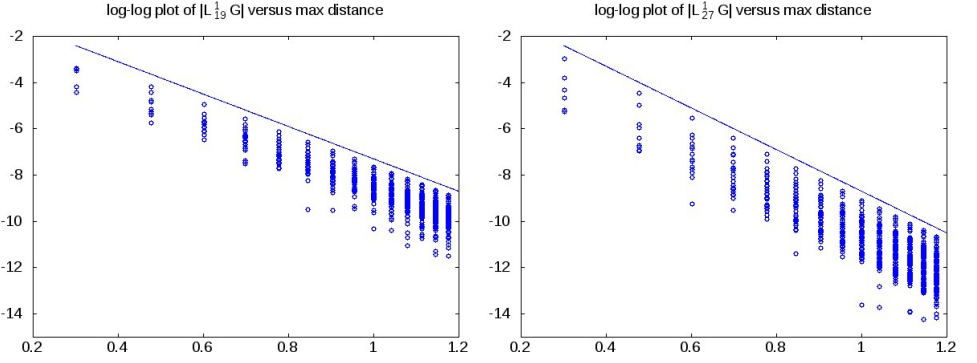


Figure 1. Scatter plots of $\log_{10}(|(\Delta^{h=1} \mathcal{G}^{h=1}(G))[g])|)$ versus $\log_{10}(\|g\|_\infty)$, $g \in \mathbb{Z}^3$, at points away from the singularity of G for the L^h_{19} and L^h_{27} discrete Laplacians. The slopes of the lines depicted are -7 and -9 for the L^h_{19} and L^h_{27} , respectively.

Suppose that $\text{supp}(f) \subset P_c$, where $P_c = c + [-R, R]^3$ is a cube of radius R centered at point c , and that the differential operator L^q is a linear combination of derivatives of order q . By differentiating (2), we have

$$[(L^q G) * f](x) = O\left(\left(\frac{1}{R}\right)^{q-2} \frac{1}{\|x/R - c/R\|_\infty^{q+1}}\right) \|f\|_\infty. \quad (13)$$

In particular, away from the support of f , (5) becomes

$$\tau^h(f) = \Delta^h(G * f)(x) = O\left(\left(\frac{h}{R}\right)^q \frac{1}{\|x/R - c/R\|_\infty^{q+3}}\right) \|f\|_\infty. \quad (14)$$

It is precisely this rapid decay of the truncation error, a consequence of the fact that the local potentials are harmonic away from the supports of the associated charges, that allows us to use a coarse mesh for the global coupling computation. In Figure 1, scatter plots of the truncation error for the case of a point charge located at the origin using the 19-point and 27-point Laplacians are depicted. The rapid decay of the truncation error in the far field and the faster decay with increasing q are evident. Using this localization property of the Mehrstellen operators, we can reduce the cost of computing the potential (2) induced by a localized charged distribution to the cost of computing the potential near the support of the charge, using the finite difference localization approach originally introduced in [23]. We assume that the support of f is contained in cube D of radius R centered at c . First, let $\phi = G * f$ be the exact solution restricted on the extended cube D_β of radius βR , $\beta > 1$. Then we compute $\phi^H = G^H * F^H$ on Ω^H . The coarse right-hand side is defined by

$$F^H = \begin{cases} \Delta^H(\phi) & \text{on } D_\beta^{H,s} = \mathcal{G}(\mathcal{C}(D_\beta^h), -s), \\ 0 & \text{on } \Omega^H \setminus D_\beta^{H,s}. \end{cases} \quad (15)$$

Using (14), we have

$$\Delta^H(\phi^H - G * f) = \begin{cases} 0 & \text{on } D_\beta^{H,s}, \\ O((H/R)^q (1/(k+\beta)^{q+3}) \|f\|_\infty) & \text{on } \{\mathbf{g} : ((k+\beta)+1)R \geq \|\mathbf{g}H\|_1 \geq (k+\beta)R\}, \end{cases} \quad (16)$$

where $\mathbf{g} \in \mathbb{Z}^3$, $k \in \mathbb{N}$. One can decompose the annular region $\{\mathbf{g} : ((k+\beta)+1)R \geq \|\mathbf{g}H\|_1 \geq (k+\beta)R\}$ into $O((k+\beta)^2)$ rectangles, each of which has radius $\leq R$, leading to an analogous decomposition of the right-hand side of (16) into a sum of terms, each of which is supported on one such rectangle. Applying convolution with G^H to both sides of (16) represented in terms of such sums leads to a solution error given by

$$\begin{aligned} \phi^H - G * f &= \sum_{k=0}^{\infty} O\left(\left(\frac{H}{R}\right)^q \frac{1}{(k+\beta)^{q+3}} \|f\|_\infty\right) \\ &= O\left(\left(\frac{H}{R}\right)^q \frac{1}{\beta^q} \|f\|_\infty\right). \end{aligned} \quad (17)$$

Thus, the accuracy of the potential away from the support of the charge can be improved by decreasing the ratio H/R or, for fixed values of that ratio, by adjusting β or q . In any case, the error is only weakly dependent on f . In this context, we will refer to β as a *localization radius*. In addition, (17) is truly independent of whether the right-hand side is modified using the Mehrstellen correction (11). The MLC algorithm combines finite difference localization with domain decomposition into a collection of rectangular patches of size R to obtain a low-communication method for computing volume potentials. This is done in a way that generalizes to nested refinement on an arbitrary number of levels, with the domain at each level decomposed into patches having a fixed number of mesh points, independent of the level of refinement. This implies that H/R remains constant, which leads to (17) being an $O(1)$ error relative to the mesh spacing. Ultimately, that error is controlled by increasing β , combined with choosing a discretization with a larger q . However, the cost of computing the local convolution $G * f$ on $D_\beta^{H,s}$ scales like β^3 . To reduce that cost, we introduce a second localization radius α , $\alpha < \beta$. On $D_\alpha^{H,s}$, we use the full convolution to compute F^H . In the remaining annular region, we use a reduced representation based on the field induced by the first few moments of the Legendre expansion of f , which is much less expensive to compute.

3. Method of local corrections: semidiscrete case

To clarify ideas, we discuss in this section a theoretical proxy for the fully discrete algorithm. We construct a function $\phi^{\text{MLC}} : \Omega \rightarrow \mathbb{R}$ that approximates the potential ϕ by a linear superposition of local potentials, combined with data interpolated from

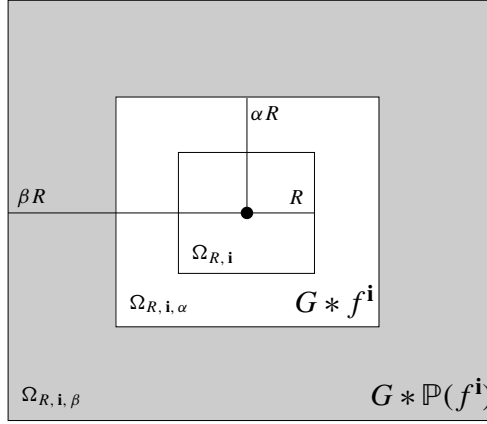


Figure 2. Regions associated with subdomain $\Omega_{R,i}$. The potential in $\Omega_{R,i,\alpha}$ (white region) is given by $G * f^i$. In the ring $\Omega_{R,i,\beta} \setminus \Omega_{R,i,\alpha}$ (shaded region) we use the field induced by a truncated Legendre expansion on $\Omega_{R,i}$ of the local charge f^i to represent the potential.

a discrete global solution. The computational domain is a cube Ω that contains the support of f and is decomposed into a finite union of disjoint cubic subdomains of equal volume that are translations of $[-R, R]^3$, $R > 0$:

$$\text{supp}(f) \subset \Omega = \bigcup_i \Omega_{R,i}, \quad \Omega_{R,i} = c^i + [-R, R]^3, \quad i \in \mathbb{Z}^3, \quad c^i = (2i + (1, 1, 1))R. \quad (18)$$

Then $f = \sum_i f^i$ where $f^i = f \chi^i$, where χ^i is the characteristic function of $\Omega_{R,i}$. As a consequence, the global potential may be written as

$$\phi(x) = (G * f)(x) = \sum_i (G * f^i)(x). \quad (19)$$

In other words, it is the linear superposition of the potentials induced by the local charges f^i which can be computed independently in parallel. The MLC algorithm replaces each of the summands in (19) with a solution truncated to zero outside of a localization radius βR , with the contribution to the solution outside the localization radius represented by interpolation from a single coarse-grid solution ϕ^H obtained by summing contributions of the form (15) over all the patches. At each point in space, the coarse-grid values used to interpolate the global contribution are corrected by subtracting off the contributions of the patches within the localization radius. Finally, to reduce the cost of computing the localized potentials, while keeping β large enough to make the $O(1)$ contribution to the error coming from localization be acceptably small, we introduce an inner radius $\alpha < \beta$ (see Figure 2). Within that inner radius, we compute the full convolution $G * f^i$; in the annular region

$\Omega_{R,i,\beta} \setminus \Omega_{R,i,\alpha}$, the local solution is approximated by $G * \mathbb{P}(f^i)$, where $\mathbb{P}(f^i)$ is the orthogonal projection onto the Legendre polynomials on $\Omega_{R,i}$ of some degree $P - 1$:

$$\begin{aligned} \mathbb{P}(f^i) &= \sum_{\mathbf{p} \in \mathbb{N}^3: \|\mathbf{p}\|_1 < P} \langle Q^{\mathbf{p}}, f^i \rangle Q^{\mathbf{p}}, \\ Q^{\mathbf{p}}(\mathbf{x}) &= R^{-3/2} \prod_{d=1}^3 Q^{p_d} \left(\frac{x_d - c_d^i}{R} \right), \quad \mathbf{x} \in \Omega_{R,i}, \quad \mathbf{q} \in \mathbb{N}^3, \end{aligned} \quad (20)$$

where $\langle \cdot, \cdot \rangle$ is the inner product on $\Omega_{R,i}$, and $Q^p : [-1, 1] \rightarrow \mathbb{R}$ is the classical Legendre polynomial of degree p .

3.1. The semidiscrete MLC algorithm. The semidiscrete MLC algorithm consists of three steps.

Step 1 (local convolutions). We perform local convolutions in regions around each subdomain $\Omega_{R,i}$ that are used to compute local charges at points on the grid:

$$F^{i,H}[\mathbf{g}] = \begin{cases} \Delta^H(G * f^i)[\mathbf{g}] & \text{if } \mathbf{g} \in \Omega_{R,i,\alpha}^H, \\ \Delta^H(G * \mathbb{P}(f^i))[\mathbf{g}] & \text{if } \mathbf{g} \in \Omega_{R,i,\beta}^H \setminus \Omega_{R,i,\alpha}^H, \\ 0 & \text{otherwise.} \end{cases}$$

Step 2 (global coarse solve). The global charge at coarse mesh points is constructed by assembling local contributions

$$F^H[\mathbf{g}] = \sum_i F^{i,H}[\mathbf{g}],$$

and we obtain a global approximation ϕ^H of the potential, represented on the coarse mesh, by computing the discrete convolution over Ω^H :

$$\phi^H = G^H * F^H. \quad (21)$$

Step 3 (local interactions/local corrections). In the final step, we represent the solution on the boundary of each $\Omega_{R,i}$ as the sum of local convolutions induced by charges on nearby patches and values interpolated from the grid calculation, from which the local convolution values have been subtracted:

$$\phi^{B,i}(\mathbf{x}) = \phi^{\text{loc},\mathbf{x}}(\mathbf{x}) + \mathcal{I}^H(\phi^H - \phi^{\text{loc},\mathbf{x}})(\mathbf{x}). \quad (22)$$

Here $\mathcal{I}^H(\psi^H)(\mathbf{x})$ is an interpolation operator that takes as input values of $\psi^H : \mathcal{N}(\mathbf{x}) \rightarrow \mathbb{R}$, where $\mathcal{N}(\mathbf{x}) \subset \{\mathbf{g}^H : \mathbf{g} \in \mathbb{Z}^3\}$, and returns a q_I -th-order accurate polynomial interpolant. In all of the algorithms described here, $\mathbf{x}$ and all of the points in $\mathcal{N}(\mathbf{x})$ are coplanar, so the interpolant is particularly easy to construct. Furthermore, $\phi^{\text{loc},\mathbf{x}}(\mathbf{x})$ is the sum of all local convolutions the support of whose

charges is sufficiently close to $\mathbf{x}$ so that they contributed to the right-hand side for the grid solution near that point:

$$\phi^{\text{loc},\mathbf{x}}(\mathbf{x}') = \sum_{i:\mathbf{x} \in \Omega_{R,i,\alpha}} (G * f^i)(\mathbf{x}') + \sum_{i:\mathbf{x} \in \Omega_{R,i,\beta} \setminus \Omega_{R,i,\alpha}} (G * \mathbb{P}(f^i))(\mathbf{x}'). \quad (23)$$

Equation (22) can be interpreted as the decomposition of the potential at a point $\mathbf{x}$, into the sum of local contributions to the potential given by $\phi^{\text{loc},\mathbf{x}}$ and corrections to include the global coupling by interpolating a corrected form of the coarse-mesh global solution ϕ^H . Specifically, the correction term in (22) is computed by evaluating $\phi^{\text{loc},\mathbf{x}}$ at the points of the interpolation stencil $\mathcal{N}(\mathbf{x})$, subtracting these values from ϕ^H , and interpolating the result to $\mathbf{x}$. The MLC solution ϕ^{MLC} is specified in terms of solutions to Dirichlet problems on each $\Omega_{R,i}$:

$$\Delta \phi^{\text{MLC}} = f^i \quad \text{in } \Omega_{R,i}, \quad \phi^{\text{MLC}} = \phi^{B,i} \quad \text{on } \partial \Omega_{R,i}. \quad (24)$$

3.2. Error analysis. The error of the local corrections step for $\mathbf{x} \in \partial \Omega_{R,i}$ is given by

$$\begin{aligned} (\phi^{B,i} - \phi)(\mathbf{x}) &= \phi^{\text{loc},\mathbf{x}}(\mathbf{x}) - \phi(\mathbf{x}) - \mathcal{I}^H(\phi^{\text{loc},\mathbf{x}} - \phi)(\mathbf{x}) + \mathcal{I}^H(\phi^H - \phi)(\mathbf{x}) \\ &= \epsilon_I^H(\phi^{\text{loc},\mathbf{x}} - \phi)(\mathbf{x}) + \mathcal{I}^H(\phi^H - \phi)(\mathbf{x}) \end{aligned} \quad (25)$$

where $\epsilon_I^H(\psi)(\mathbf{x})$ is the error in applying the interpolation operator $\mathcal{I}^H$ to a smooth function ψ evaluated on the grid and evaluating it at $\mathbf{x}$. There are two sources of error for the semidiscrete algorithm: one from the calculation of ϕ^H in (21), and the other due to interpolation at the local corrections step (22). To estimate the former, i.e., the second term of (25), it suffices to bound the coarse mesh error $\phi^H - \phi$. To do so, we estimate the truncation error of the coarse solve (21) at points $\mathbf{g}$:

$$\begin{aligned} \tau_C^H &= \Delta^H(\phi^H - \phi)[\mathbf{g}] \\ &= -\Delta^H \left(\sum_{i:\mathbf{g}H \notin \Omega_{R,i,\beta}} G * f^i \right) [\mathbf{g}] - \Delta^H \left(\sum_{i:\mathbf{g}H \in \Omega_{R,i,\beta} \setminus \Omega_{R,i,\alpha}} G * ((\mathbb{I} - \mathbb{P})(f^i)) \right) [\mathbf{g}]. \end{aligned} \quad (26)$$

To bound the first term of (26), we use (14) to find that

$$\begin{aligned} \Delta^H \left(\sum_{i:\mathbf{g}H \notin \Omega_{R,i,\beta}} G * f^i \right) [\mathbf{g}] &= O \left(\left(\frac{H}{R} \right)^q \sum_{k=0}^{\infty} \sum_{i:\mathbf{g}H \in \Omega_{R,i,\beta+k+1} \setminus \Omega_{R,i,\beta+k}} \frac{1}{(\beta+k)^{q+3}} \|f^i\|_{\infty} \right) \\ &= O \left(\left(\frac{H}{R} \right)^q \frac{1}{\beta^q} \|f\|_{\infty} \right). \end{aligned} \quad (27)$$

The second term of (26) is bounded in a similar fashion:

$$\begin{aligned} \Delta^H \left(\sum_{i: \mathbf{g}H \in \Omega_{R,i,\beta} \setminus \Omega_{R,i,\alpha}} G * ((\mathbb{I} - \mathbb{P})(f^i)) \right) [\mathbf{g}] &= O \left(\left(\frac{H}{R} \right)^q \frac{1}{\alpha^q} \max_i \|(\mathbb{I} - \mathbb{P})(f^i)\|_\infty \right) \\ &= O \left(\left(\frac{H}{R} \right)^q \frac{1}{\alpha^q} H^P \right), \end{aligned} \quad (28)$$

where we have used

$$\|(\mathbb{I} - \mathbb{P})(f^i)\|_\infty = O(R^P), \quad (29)$$

which follows directly from Taylor's theorem for f^i and the fact that $\pi = \mathbb{P}(\pi)$ for polynomials π of degree less than P . As a result, the estimate

$$\Delta^H(\phi^H - \phi) = O \left(\left(\frac{H}{R} \right)^q \frac{1}{\alpha^q} H^P \right) + O \left(\left(\frac{H}{R} \right)^q \frac{1}{\beta^q} \|f\|_\infty \right) \quad (30)$$

for the coarse mesh error holds uniformly on coarse mesh points. Since convolution with G^H and the interpolation operator $\mathcal{I}^H$ are max norm bounded, $\epsilon_C^H \equiv \phi^H - \phi$ is also bounded by an expression of the form of the right-hand side of (30).

To bound the first term in (25), it follows from the fact that the interpolation method is q_I -th-order accurate that

$$\begin{aligned} \epsilon_I^H(\phi^{\text{loc},\mathbf{x}} - \phi)(\mathbf{x}) &= H^{q_I} L_I^{q_I}(\phi^{\text{loc},\mathbf{x}} - \phi)(\boldsymbol{\xi}) \\ &= -H^{q_I} \left(\sum_{i: \mathbf{x} \in \Omega_{R,i,\beta} \setminus \Omega_{R,i,\alpha}} ((L_I^{q_I} G) * (\mathbb{I} - \mathbb{P})(f^i))(\boldsymbol{\xi}) + \sum_{i: \mathbf{x} \notin \Omega_{R,i,\beta}} ((L_I^{q_I} G) * f^i)(\boldsymbol{\xi}) \right) \end{aligned} \quad (31)$$

where $\boldsymbol{\xi}$ is in an $O(H)$ neighborhood of $\mathcal{N}(\mathbf{x})$ and $L_I^{q_I}$ is a linear differential operator with terms that are derivatives of order q_I . Using (13), a similar argument to that given in the proof of (30) leads to

$$\epsilon_I^H = H^{P+2} O \left(\left(\frac{H}{R} \right)^{q_I-2} \frac{1}{\alpha^{q_I-2}} \right) + H^2 O \left(\left(\frac{H}{R} \right)^{q_I-2} \frac{1}{\beta^{q_I-2}} \|f\|_\infty \right)$$

so that (25) is estimated as

$$\begin{aligned} \epsilon^{\text{SD}} \equiv \phi^{B,i} - \phi &= H^{P+2} O \left(\left(\frac{H}{R} \right)^{q_I-2} \frac{1}{\alpha^{q_I-2}} \right) + H^2 O \left(\left(\frac{H}{R} \right)^{q_I-2} \frac{1}{\beta^{q_I-2}} \|f\|_\infty \right) \\ &\quad + O \left(\left(\frac{H}{R} \right)^q \frac{1}{\alpha^q} H^P \right) + O \left(\left(\frac{H}{R} \right)^q \frac{\|f\|_\infty}{\beta^q} \right). \end{aligned} \quad (32)$$

4. Method of local corrections: fully discrete case

In this section, we describe the two-level algorithm as it is actually implemented. Ω^h is a fine-grid discretization of a bounded domain Ω , the latter containing

the support of f . Ω^h is assumed to be a finite union of rectangles of the form $\Omega_{R,i}^h = n\mathbf{i} + [0, n]^3$, $R = nh/2$. We also define discrete forms of $\Omega_{R,i,\alpha}^h$ and $\Omega_{R,i,\beta}^h$: $\Omega_{R,i,\alpha}^h = \mathcal{G}(\Omega_{R,i}^h, \lceil(\alpha - 1)n/2\rceil)$ and $\Omega_{R,i,\beta}^h = \mathcal{G}(\Omega_{R,i}^h, \lceil(\beta - 1)n/2\rceil)$. The coarse grid Ω^H is assumed to cover all of the fine patch data required for the algorithm described below: $\mathcal{G}(\mathcal{G}(\Omega_{R,i,\beta}^h), b) \subset \Omega^H$ where b is the radius of the stencil for the interpolation function $\mathcal{I}^H$. We also define a discretized form of the characteristic function of a rectangular patch $D \subset \mathbb{Z}^3$:

$$\chi_D(\mathbf{x}) = \begin{cases} \frac{1}{8} & \text{if } \mathbf{g} \text{ is a corner of } D, \\ \frac{1}{4} & \text{if } \mathbf{g} \text{ lies on an edge of } D, \\ \frac{1}{2} & \text{if } \mathbf{g} \text{ lies on a face of } D, \\ 1 & \text{if } \mathbf{g} \text{ lies in the interior of } D, \\ 0 & \text{elsewhere.} \end{cases}$$

In the fully discrete algorithm, we replace the local convolutions with local discrete convolutions, e.g., $G * f^i \rightarrow G^h * f^{i,h}$ and $f^{i,h} = \chi_{\Omega_{R,i}^h} f$, and we take $H = N_{\text{ref}}h$.

4.1. The fully discrete two-level algorithm.

Step 1 (local convolutions). For each $\Omega_{R,i}^h$, we compute the potential induced by $f^{i,h} = \chi_{\Omega_{R,i}^h} f^h$:

$$\phi^{i,h} = G^h * f^{i,h} \quad \text{on } \mathcal{G}(\Omega_{R,i,\alpha}^h, N_{\text{ref}}b). \quad (33)$$

The Legendre expansion coefficients of $f^{i,h}$ required to compute $\mathbb{P}(f^i)$ are computed with composite numerical integration. We employ Boole's rule if f is given only at points of Ω^h or Gauss integration if f is specified analytically. For each $\Omega_{R,i}^h$ we also compute the associated local charges

$$F^{i,H}[\mathbf{g}] = \begin{cases} \Delta^H \phi_i^h[\mathbf{g}], & \mathbf{g} \in \mathcal{G}(\Omega_{R,i,\alpha}^h), \\ \Delta^H (G^h * \mathbb{P}^h(f^{i,h}))[\mathbf{g}], & \mathbf{g} \in \mathcal{G}(\Omega_{R,i,\beta}^h) \setminus \mathcal{G}(\Omega_{R,i,\alpha}^h), \\ 0, & \mathbf{g} \notin \mathcal{G}(\Omega_{R,i,\beta}^h). \end{cases} \quad (34)$$

The values of $\Delta^H (G^h * Q^p)$ can be computed once and stored, reducing the calculation of $\Delta^H (G^h * \mathbb{P}^h(f^{i,h}))$ to computing linear combinations of the appropriate subset of those precomputed values.

Step 2 (global coarse solve). $\phi^H = G^H * F^H \quad \text{on } \Omega^H, \quad F^H = \sum_i F^{i,H}.$

Step 3 (local interactions/local corrections). We define the local potentials at fine boundary points $\mathbf{g} \in \partial\Omega_{R,i}^h$ as combinations of short-range and intermediate-range components

$$\phi^{\text{loc},\mathbf{g}}[\mathbf{g}'] = \sum_{i': \mathbf{g} \in \Omega_{R,i',\alpha}^h} \phi^{i',h}[\mathbf{g}'] + \sum_{i': \mathbf{g} \in \Omega_{R,i',\beta}^h \setminus \Omega_{R,i',\alpha}^h} (G^h * \mathbb{P}^h(f^{i',h}))[\mathbf{g}'], \quad (35)$$

and we correct them by adding the far field effects as in (22):

$$\phi^{B,i,h}[\mathbf{g}] = \phi^{\text{loc},\mathbf{g}}[\mathbf{g}] + \mathcal{J}^H(\phi^H - (\phi^{\text{loc},\mathbf{g}}))(gh), \quad \mathbf{g} \in \partial\Omega_i^h. \quad (36)$$

The interpolation operator on coplanar points $\mathcal{J}^H$ that we are employing is the same as in [25]. Using these boundary conditions, we solve the following local Dirichlet problems on Ω_i^h patches:

$$\begin{aligned} \Delta^h \tilde{\phi}^{\text{MLC},i,h} &= f^{i,h} && \text{on } \Omega_{R,i}^h \setminus \partial\Omega_{R,i}^h, \\ \tilde{\phi}^{\text{MLC},i,h} &= \phi^{B,i,h} && \text{on } \partial\Omega_{R,i}^h. \end{aligned} \quad (37)$$

Finally, the fourth-order Mehrstellen correction (12) is applied to obtain the values of $\phi^{\text{MLC},h}$

$$\phi^{\text{MLC},h}[\mathbf{g}] = \tilde{\phi}^{\text{MLC},i,h}[\mathbf{g}] + C_2 h^2 f^h[\mathbf{g}], \quad \mathbf{g} \in \Omega_{R,i}^h. \quad (38)$$

If we want to go to higher than fourth-order accuracy in h , the algorithm is more complicated—the Mehrstellen correction must be applied earlier in the process. We will not discuss the details in this paper.

4.2. Error analysis. We proceed in this section with estimating the error for the fully discrete MLC algorithm. We want to get some idea of the impact of replacing the analytic continuous convolutions by the discretized convolutions. To do this, we use a modified equation approach, in which we assume that we can approximate the solution error by the action of the operator on the truncation error. In the present setting, this amounts to making the substitution

$$G^h * \psi^h \rightarrow G * (\psi + \delta\tau^h(\psi)) - C_2 h^2 \psi, \quad (39)$$

$$\delta\tau^h(\psi) = \Delta(G^h * \psi^h) - \psi + C_2 h^2 \Delta\psi = O(h^4). \quad (40)$$

As in the semidiscrete case, we want to estimate the error in the boundary conditions

$$\begin{aligned} \phi^{B,i,h}[\mathbf{g}] - \tilde{\phi}(gh) &= \phi^{\text{loc},\mathbf{g}}[\mathbf{g}] - \tilde{\phi}(gh) + \mathcal{J}^H(\phi^H - \phi^{\text{loc},\mathbf{g}})(gh) \\ &= \mathcal{J}^H(\phi^H - \tilde{\phi})(gh) \end{aligned} \quad (41)$$

$$+ \phi^{\text{loc},\mathbf{g}}[\mathbf{g}] - \tilde{\phi}(gh) - \mathcal{J}^H(\phi^{\text{loc},\mathbf{g}} - \tilde{\phi})(gh), \quad \mathbf{g} \in \Omega_{R,i}^h, \quad (42)$$

where

$$\tilde{\phi} \equiv \phi + C_2 h^2 f.$$

An estimate of the contribution from (41) is obtained by bounding $\Delta^H(\phi^H - \tilde{\phi})$, since $\mathcal{J}^H$ and convolution with G^H are both stable in max norm. We have, by (39),

$$\begin{aligned}
\Delta^H(\phi^H - \tilde{\phi})[\mathbf{g}] &= - \sum_{\mathbf{i}': \mathbf{g} \notin \Omega_{R, \mathbf{i}', \beta}^H} \Delta^H(G * f^{\mathbf{i}'})[\mathbf{g}] \\
&- \sum_{\mathbf{i}': \mathbf{g} \in \Omega_{R, \mathbf{i}', \beta}^H \setminus \Omega_{R, \mathbf{i}', \alpha}^H} \Delta^H(G * (\mathbb{I} - \mathbb{P})(f^{\mathbf{i}'}))[\mathbf{g}] - \sum_{\mathbf{i}': \mathbf{g} \notin \Omega_{R, \mathbf{i}', \beta}^H} \Delta^H(G * (\delta \tau^h(f^{\mathbf{i}', h})))[\mathbf{g}] \\
&- \sum_{\mathbf{i}': \mathbf{g} \in \Omega_{R, \mathbf{i}', \beta}^H \setminus \Omega_{R, \mathbf{i}', \alpha}^H} \Delta^H(G * \delta \tau^h((\mathbb{I} - \mathbb{P})(f^{\mathbf{i}'})))[\mathbf{g}] \\
&- \sum_{\mathbf{i}': \mathbf{g} \in \Omega_{R, \mathbf{i}', \beta}^H \setminus \Omega_{R, \mathbf{i}', \alpha}^H} \Delta^H(G^h * ((\mathbb{P}(f^{\mathbf{i}'}))^h - (\mathbb{P}^h(f^{\mathbf{i}', h}))))[\mathbf{g}] + O(h^4). \tag{43}
\end{aligned}$$

The first two terms are identical to the ones that appear in the semidiscrete case, while (39) and the estimate $\|(\mathbb{P} - \mathbb{P}^h)(f^{\mathbf{i}'})\|_\infty = O(h^6)$ (which holds since our quadrature rules for computing the Legendre coefficients are at least sixth-order accurate) guarantee that the remaining terms are $O(h^4)$ or smaller. Using similar arguments to those in (43), we have

$$\phi^{\text{loc}, \mathbf{g}} - \tilde{\phi} = - \sum_{\mathbf{i}': \mathbf{g} \notin \Omega_{R, \mathbf{i}', \beta}^h} G * f^{\mathbf{i}'} - \sum_{\mathbf{i}': \mathbf{g} \in \Omega_{R, \mathbf{i}', \beta}^h \setminus \Omega_{R, \mathbf{i}', \alpha}^h} G * ((\mathbb{I} - \mathbb{P})(f^{\mathbf{i}'})) + O(h^4),$$

and therefore, following (31), we have

$$\begin{aligned}
\epsilon_I^H(\phi^{\text{loc}, \mathbf{g}} - \tilde{\phi})(\mathbf{g}h) &= H^{P+2} O\left(\left(\frac{H}{R}\right)^{q_I-2} \frac{1}{\alpha^{q_I-2}}\right) + H^2 O\left(\left(\frac{H}{R}\right)^{q_I-2} \frac{1}{\beta^{q_I-2}} \|f\|_\infty\right) + O(h^4),
\end{aligned}$$

Thus, we have

$$\phi^{B, \mathbf{i}, h}[\mathbf{g}] - \tilde{\phi}(\mathbf{g}h) = \epsilon^{\text{SD}} + O(h^4).$$

The stability of the discretized boundary value problem implies $\|\phi^{\text{MLC}, h} - \phi\|_\infty = O(\|\phi^{B, h} - \phi\|_\infty) + O(h^4)$, so we finally have the estimate

$$\begin{aligned}
\phi^{\text{MLC}, h} - \phi &= O(h^4) + \epsilon^{\text{SD}} \\
&= O(h^4) + H^{P+2} O\left(\left(\frac{H}{R}\right)^{q_I-2} \frac{1}{\alpha^{q_I-2}}\right) + H^2 O\left(\left(\frac{H}{R}\right)^{q_I-2} \frac{\|f\|_\infty}{\beta^{q_I-2}}\right) \\
&\quad + O\left(\left(\frac{H}{R}\right)^q \frac{1}{\alpha^q} H^P\right) + O\left(\left(\frac{H}{R}\right)^q \frac{\|f\|_\infty}{\beta^q}\right). \tag{44}
\end{aligned}$$

at all fine grid points. This error can be written in the form

$$\phi^{\text{MLC},h} = \phi + O(h^4) + O(h^P) + O\left(h^2 \|f\|_\infty \frac{1}{\beta^{q_I-2}}\right) + O\left(\|f\|_\infty \frac{1}{\beta^q}\right). \quad (45)$$

Thus, MLC differs from classical finite difference methods in that there is a contribution to the error that does not vanish as $h \rightarrow 0$, i.e., the right-most summand in (44). We refer to this contribution to the error as the *barrier error*. Note that, if we take $q_I = q + 2$, we obtain the form of the error given in the Introduction. We have specialized this algorithm to the case of fourth-order accuracy, primarily because it allows us the simplification of applying the Mehrstellen correction (38) at the end of the calculation. However, this analysis suggests that, even with this simplification, there might be an advantage to using discretizations of the Laplacian with larger q , i.e., ones that are higher-order accurate when applied to harmonic functions, since the barrier error is proportional to β^{-q} . We observe this to be the case in the results in Section 7.

5. Multilevel method of local corrections

Following [25], we generalize the method in Section 4 to the case of an arbitrary number of levels $l = 0, \dots, l_{\max}$, where $l_{\max}$ is the finest level on which the solution is sought. We denote the discrete Laplacian with mesh size h_l by Δ^{h_l} , with $h_l = N_{\text{ref}} h_{l+1}$. At each level we discretize the solution on a collection of node-centered cubic patches $\Omega_{R_l, i}$, $R_l = N_{\text{ref}} R_{l+1}$, and the corresponding discretized grids $\Omega_{R_l, i}^{h_l}$; the combined level- l grid is given by $\Omega^{l, h_l} \equiv \bigcup_i \Omega_{R_l, i}^{h_l}$. We also define, for each i , localization regions $\Omega_{R_l, i, \alpha}$ and $\Omega_{R_l, i, \beta}$, and their discretizations $\Omega_{R_l, i, \alpha}^{h_l}$ and $\Omega_{R_l, i, \beta}^{h_l}$, $1 < \alpha < \beta$. At level 0 there is only one patch Ω^{0, h_0} at which the coarse solve of the method is performed, just as in the two-level algorithm. We also impose a proper nesting condition: for $l = 1, \dots, l_{\max}$,

$$\mathcal{G}(\mathcal{G}(\Omega_{R_l, i, \beta}^{h_l}), b) \subset \Omega^{l-1, h_{l-1}}. \quad (46)$$

The multilevel MLC comprises the following steps.

Step 1 (downward pass: initial local convolutions). Local convolutions are computed at levels $l = l_{\max}, \dots, 1$:

$$\phi^{i, h_l} = G^{h_l} * \tilde{f}^{i, h_l} \quad \text{on } \mathcal{G}(\Omega_{R_l, i, \alpha}^{h_l}, N_{\text{ref}} b), \quad (47)$$

where the local right-hand sides are defined as

$$\begin{aligned}\tilde{f}^{i,h_l} &= \sum_{i'} \Delta^{h_l} (\phi^{i',h_{l+1}})|_{\mathcal{C}(\Omega_{R_{l+1},i'},\alpha)} \\ &\quad + \sum_{i'} \Delta^{h_l} (G^{h_{l+1}} * \mathbb{P}(f^{i',h_{l+1}}))|_{\mathcal{C}(\Omega_{R_{l+1},i'},\beta \setminus \Omega_{R_{l+1},i'},\alpha)} + \tilde{\chi}_{\Omega_{R_l,i}^{h_l}} f^{h_l}, \\ \tilde{\chi}_{\Omega_{R_l,i}^{h_l}}[\mathbf{g}] &= \chi_{\Omega_{R_l,i}^{h_l}}[\mathbf{g}] - \sum_{\substack{i'=N_{\text{ref}}i+s \\ 0 \leq s_d \leq N_{\text{ref}}}} \chi_{\Omega_{R_{l+1},i'}^{h_{l+1}}} [N_{\text{ref}}\mathbf{g}].\end{aligned}$$

Step 2 (global coarse solve). $\phi^{h_0} = G^{h_0} * \tilde{f}^{h_0} \quad \text{on } \Omega^{0,h_0}.$

Step 3 (upward pass: local interactions/local corrections for $1, \dots, l_{\text{max}}$). Starting from level 1, the following local Dirichlet problems are solved at levels $l = 1, \dots, l_{\text{max}}$:

$$\begin{aligned}\Delta^{h_l} \tilde{\phi}^{\text{MLC},i,h_l} &= \tilde{f}^{i,h_l} && \text{on } \Omega_{R_l,i}^{h_l} \setminus \partial \Omega_{R_l,i}^{h_l}, \\ \tilde{\phi}^{\text{MLC},i,h_l} &= \phi^{B,i,h_l} && \text{on } \partial \Omega_{R_l,i}^{h_l}, \\ \tilde{\phi}^{\text{MLC},l} &= \tilde{\phi}^{\text{MLC},i,h_l} && \text{on } \Omega_{R_l,i}^{h_l}.\end{aligned}\tag{48}$$

The Dirichlet boundary conditions are given by

$$\phi^{B,i,h_l}[\mathbf{g}] = \phi^{\text{loc},l,\mathbf{g}}[\mathbf{g}] + \mathcal{J}^{h_{l-1}}(\tilde{\phi}^{\text{MLC},l-1} - \phi^{\text{loc},l,\mathbf{g}})(\mathbf{g}h_l).\tag{49}$$

Here the local potentials $\phi^{\text{loc},\mathbf{g},l}$ are given by

$$\phi^{\text{loc},l,\mathbf{g}}[\mathbf{g}'] = \sum_{i': \mathbf{g} \in \Omega_{R_l,i'}^{h_l}} \phi^{i',h_l}[\mathbf{g}'] + \sum_{i': \mathbf{g} \in \Omega_{R_l,i'}^{h_l} \setminus \Omega_{R_l,i'},\alpha} (G^{h_l} * \mathbb{P}(f^{i'}))[\mathbf{g}'].\tag{50}$$

Finally, the Mehrstellen correction at all levels is applied as

$$\phi^{\text{MLC},l}[\mathbf{g}] = \tilde{\phi}^{\text{MLC},l}[\mathbf{g}] + C_2 h_l^2 f^{i,h_l}[\mathbf{g}], \quad \mathbf{g} \in \Omega^{l,h_l}\tag{51}$$

We do not have a complete error analysis for the above algorithm corresponding to that given in the two-level case. However, we can look at error analysis of the two-level algorithm, and determine the change in the error introduced there by replacing the coarse-grid convolution with G^H with an MLC calculation. We denote

- $G^{\text{MLC},S}(r)$ the two-level semidiscrete method of local corrections approximation to $G * r$, with patch radius S ,
- $N_1^S(r)(\mathbf{x}) \equiv \sum_{i: \mathbf{x} \notin \Omega_{S,i,\beta}} h^q L^{q+2}(G * r^i)(\mathbf{x})$,
- $N_2^S(r)(\mathbf{x}) \equiv \sum_{i: \mathbf{x} \in \Omega_{S,i,\beta} \setminus \Omega_{S,i,\alpha}} h^q L^{q+2}(G * ((\mathbb{I} - \mathbb{P})r^i))(\mathbf{x})$, and
- $N^S(r) = N_1^S(r) + N_2^S(r)$.

By (25) and (26), $G^H * (N^R(f))^H = (G * f)^H - \phi^H$ is the only quantity in the error in which convolution with G^H appears. Given that, it is straightforward to assess the impact of replacing the convolution with G^H in this expression with applying the MLC algorithm for a patch size $N_{\text{ref}}R$. To estimate this effect, we use a modified equation approach, in which the difference is approximated by $G * (N^R(f)) - G^{\text{MLC}, N_{\text{ref}}R}(N^R(f))$. Applying the error estimate (26), we obtain

$$\begin{aligned} G * (N^R(f)) - G^{\text{MLC}, N_{\text{ref}}R}(N^R(f)) &= N^{N_{\text{ref}}R}(N^R(f)) \\ &= N_1^{N_{\text{ref}}R}(N_1^R(f)) + N_1^{N_{\text{ref}}R}(N_2^R(f)) + N_2^{N_{\text{ref}}R}(N_1^R(f)) + N_2^{N_{\text{ref}}R}(N_2^R(f)). \end{aligned}$$

For this substitution to have an appropriately small impact, it is sufficient for the error to be comparable to or less than the error in the two-level algorithm. The sum of the first three terms meet this criterion—the sum of the first two terms is bounded by the max norm of the two-level error multiplied by $O(\beta^{-q})$, and the third term is bounded by $O(\alpha^{-q})$ times the max norm of the barrier error of the two-level algorithm. The final term, however, is problematic. In particular, the impact on the error of multiple applications of $\mathbb{I} - \mathbb{P}$ at increasing mesh spacings is far from clear. We will see evidence of this in the numerical results in Section 7.2, and will suggest a remedy that allows the error to be controlled.

6. Computational issues

The analysis and demonstration of the performance of this algorithm will be the subject of a separate paper [24], so we will just make a few high-level observations to justify the pursuit of this line of research. The largest contribution to the floating point operation count in this method comes from the initial local discrete convolutions (33). To compute these convolutions, we use a generalization of Hockney’s domain-doubling algorithm [16], which we describe in Appendix B. The floating point work per unknown for this step is $O(\alpha^3 \log(n))$, $\alpha > 1$, where n^3 is the number of points per patch. The next-largest computation is that of the final Dirichlet solutions (37), performed using sine transforms, which is $O(\log(n))$ per unknown. The floating point work associated with computing the Legendre expansions is small, with the convolutions of Legendre polynomials with the discrete Green’s functions precomputed and stored. The memory overhead for storing these quantities scales like $O(\beta^3 n^3)$. However, there is one copy of these per processor, shared across multiple patches/cores. Furthermore, they are only stored either on a sampled grid coarsened by N_{ref} , or on planar subsets corresponding to boundaries of patches, which reduces the memory overhead further.

The parallel implementation of this algorithm is via domain decomposition, with patches distributed to processors. For the choices of α and β used in the results described below, this corresponds to a floating point operation count about

three times that of a corresponding multigrid algorithm for comparable accuracy. Roughly speaking, the communications cost, in terms of number of messages and overall volume of data moved, corresponds to that of a single multigrid V-cycle, plus the negligible costs of communicating a small number of Legendre expansion coefficients (20 per patch for the case $P = 4$). This is to be compared to the 8 multigrid V-cycles required to obtain a comparable level of accuracy. Current trends in the design of HPC processors based on low-power processor technologies indicate a rapid growth in the number of cores capable of performing floating point operations on a processor, while the communications bandwidth between processors, or between the processor and main memory, is growing much more slowly. In addition, most of the floating point work is performed using FFTs on small patches on a single node, for which there are multiple opportunities for performance optimization. Thus, the present algorithm is well positioned to take advantage of these trends.

7. Numerical test cases

We present in this section several examples that demonstrate the convergence properties of the MLC method described above. In all cases, we use as a measure of the solution error the max norm error of the potential, divided by max norm of the potential

$$\frac{\|\phi^{\text{MLC},h} - \phi\|_\infty}{\|\phi\|_\infty}. \quad (52)$$

For all cases, we set $n = 32$, so that $H/R = \frac{1}{4}$. We refer to the special case $\beta = \alpha$ (i.e., if the long-range potentials induced by the truncated Legendre expansions of local charges are ignored) as the MLC-0 method and to the general case $\alpha < \beta$ as the MLC method. It is not difficult to see that for MLC-0, the estimate (44) reduces to

$$\phi^{\text{MLC},h} - \phi = O(h^4) + O\left(h^2 \left(\frac{H}{R}\right)^{q_l-2} \frac{\|f\|_\infty}{\beta^{q_l-2}}\right) + O\left(\left(\frac{H}{R}\right)^q \frac{\|f\|_\infty}{\beta^q}\right). \quad (53)$$

Increasing β to reduce the barrier error in (53) substantially increases the per patch computational cost of the discrete convolution in the downward pass of the method. This is, in fact, the reason we replaced the local long-range potential values with the convolutions of the local Legendre expansions in [Section 3.1](#).

7.1. A smooth charge distribution. The first test case we are considering involves computing the potential induced by a smooth charge. The computational domain is the unit cube $\Omega = [0, 1]^3$. The charge density is given by

$$f(\mathbf{x}) = \begin{cases} (r - r^2)^4, & r < 1, \\ 0, & r \geq 1, \end{cases} \quad r = \frac{1}{R_o} \|\mathbf{x} - \mathbf{x}_o\|,$$

N	$\beta = 1.5$	$\beta = 3.0$	$\beta = 6.0$
256	1.43756×10^{-5}	6.07186×10^{-7}	5.80288×10^{-8}
512	1.29572×10^{-5}	4.32691×10^{-7}	2.67372×10^{-8}
1024	1.27114×10^{-5}	4.01180×10^{-7}	2.44521×10^{-8}

Table 1. 2-level MLC-0: scaled fine-mesh maximum errors (52) using the L_{19}^h Mehrstellen Laplacian.

N	MLC-0 ($\beta = 1.5$)	$P = 1$	$P = 4$
256	1.43756×10^{-5}	4.35976×10^{-6}	1.63706×10^{-6}
512	1.29572×10^{-5}	1.43414×10^{-6}	4.58615×10^{-7}
1024	1.27114×10^{-5}	5.77475×10^{-7}	3.65246×10^{-7}

Table 2. 2-level MLC: scaled fine-mesh maximum errors (52) using L_{19}^h . For sufficiently small h and $P = 4$, nearly the same errors as the second column of Table 1 are obtained.

N	MLC-0 ($\beta = 1.5$)	$P = 1$	$P = 4$	$P = 5$
256	1.43756×10^{-5}	4.05752×10^{-6}	1.45072×10^{-6}	1.68422×10^{-6}
512	1.29572×10^{-5}	1.12630×10^{-6}	1.04191×10^{-7}	4.49529×10^{-8}
1024	1.27114×10^{-5}	2.37651×10^{-7}	2.55964×10^{-8}	2.44951×10^{-8}

Table 3. 2-level MLC: scaled fine-mesh maximum errors (52) using L_{19}^h . Here $\alpha = 1.5$ and $\beta = 6$. For sufficiently small h and high values of P , nearly the same errors as the third column of Table 1 are obtained.

and the support of the charge is a sphere of radius $R_o = \frac{1}{4}$, centered at the point $\mathbf{x}_o = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. The exact solution for this problem is given by

$$\phi(\mathbf{x}) = R_o^2 \begin{cases} \frac{1}{42}r^6 - \frac{1}{14}r^7 + \frac{1}{12}r^8 - \frac{2}{45}r^9 + \frac{1}{110}r^{10} - \frac{1}{1260}, & r < 1, \\ -1/(2310r), & r \geq 1, \end{cases}$$

and reduces to a pure monopole field for $r \geq 1$.

7.1.1. Two-level results. In Table 1 we present the fine-mesh errors for the MLC-0 algorithm with two levels for mesh sizes $h = \frac{1}{256}, \frac{1}{512}, \frac{1}{1024}$ using the L_{19}^h Mehrstellen Laplacian ($q = 4$). We set $b = 2 \rightarrow q_I = 6$ so that dependence of the interpolation error as a function of α and β matches that of the other error terms. For this problem, the errors in all three cases are so small that they are the barrier errors; each time we double β , the error goes down by roughly a factor of 16, as predicted by (53). In Tables 2 and 3 we present fine-mesh errors for the MLC algorithm, with $\alpha = 1.5$, for $\beta = 3$ and $\beta = 6$, respectively, when refining both h and P . As $h \rightarrow 0$, the error in this case approaches a barrier error for both the $P = 1$ and $P = 4$ cases at a rate of $O(h^2)$ – $O(h^4)$, and those barrier errors correspond to the errors for the MLC-0 calculations with same corresponding values of β . For comparison, we also include

N	$\beta = 2.0$	$\beta = 3.25$
256	1.25208×10^{-7}	4.11121×10^{-8}
512	1.14831×10^{-7}	4.92150×10^{-9}
1024	1.01073×10^{-7}	3.11897×10^{-9}

Table 4. 2-level MLC-0: scaled fine-mesh maximum error (52) using the L_{27}^h Mehrstellen Laplacian. Compare with the second and third columns of Table 1.

N	$P = 1$	$P = 4$
256	1.45293×10^{-6}	1.40270×10^{-6}
512	5.20885×10^{-7}	1.89409×10^{-7}
1024	1.77613×10^{-7}	1.02341×10^{-7}

Table 5. 2-level MLC: scaled fine-mesh maximum errors (52) using L_{27}^h . Here $\alpha = 1.5$ and $\beta = 2$. The $h \rightarrow 0$ errors are the same as the barrier errors in the first column of Table 4.

N	$P = 1$	$P = 4$	$P = 5$
256	1.40367×10^{-6}	1.47261×10^{-6}	1.63939×10^{-6}
512	4.32214×10^{-7}	8.68274×10^{-8}	5.91126×10^{-8}
1024	9.11841×10^{-8}	1.18905×10^{-8}	1.17441×10^{-8}

Table 6. 2-level MLC: scaled fine-mesh maximum errors (52) using L_{27}^h . Here $\alpha = 1.5$ and $\beta = 3.25$. The barrier errors are comparable with those using L_{19}^h with $\beta = 6$ (Table 3).

the values of the error for the MLC-0 calculations with comparable computational costs, i.e., for $\beta = 1.5$. It is clear that for the negligible cost of adding the Legendre expansion, we obtain a decrease in the error by 1–3 orders of magnitude.

Next, we present the errors obtained by performing similar runs using the L_{27}^h Mehrstellen Laplacian, for which $q = 6$. We set $b = 3 \rightarrow q_I = 8$ so that dependence of the interpolation error as a function of α and β matches that of the other error terms. In this case, the barrier error is $O(\beta^{-6})$; hence, we expect that smaller values of the β correction radius are required to obtain errors similar to those obtained with the L_{19}^h difference operator. Since $3^4 \approx 2^6$ and $6^4 \approx 3.25^6$, we set $\beta = 2, 3.25$. First, in order to estimate the barrier values, we present the fine-mesh errors for the MLC-0 method in Table 4 with $\beta = 2, 3.25$ using the L_{27}^h operator. With those values of β , we expect errors comparable to or smaller than those of the MLC-0 method with $\beta = 3, 6$ using the L_{19}^h operator. This is the case, as is evident from a comparison with the error values of Table 1. Furthermore, the barrier error as a function of β decreases by more than the factor of $18.4 = (3.25/2)^6$ predicted by the analysis.

N	$P = 1$	$P = 4$	$P = 6$	$P = 9$
256	1.91470×10^{-7}	1.96490×10^{-7}	6.39837×10^{-8}	4.90745×10^{-8}
512	5.42412×10^{-8}	9.16574×10^{-9}	5.99534×10^{-9}	6.02843×10^{-9}
1024	1.40428×10^{-8}	2.79547×10^{-9}		

Table 7. 2-level MLC: scaled fine-mesh maximum errors (52) using L_{27}^h . Here $\alpha = 1.75$ and $\beta = 3.25$. Compare with the second column of Table 4. A high polynomial degree is required to attain it for $h = \frac{1}{256}$.

N	$\beta = 2.0$	$\beta = 3.25$
512	1.30594×10^{-7}	4.86092×10^{-9}
1024	1.90632×10^{-7}	3.92874×10^{-9}

Table 8. 3-level MLC-0: scaled fine-mesh maximum errors (52) using the L_{27}^h Mehrstellen Laplacian. Compare with Table 4, which contains the two-level results.

In Tables 5 and 6, we present the errors for the MLC algorithm, for the cases $\beta = 2, 3.25$; $\alpha = 1.5$ for both cases. The $\beta = 2$ calculations reach the same barrier errors as h decreases. That is not the case for the $\beta = 3.25$ results in Table 4, but that is not surprising — the reduction of the barrier error by nearly an order of magnitude provides more headroom for h -convergence. However, we see that in Table 7 a slight increase of the inner correction radius to $\alpha = 1.75$ allows us to reach the barrier error more rapidly. This is consistent with the error analysis, in that increasing α reduces the coefficient in front of the $O(h^P)$ error from truncating the Legendre expansion, from which we infer that the error from that source, rather than the error from the inner local convolution, is the dominant h -dependent error for this smooth example.

7.1.2. Three-level results. We next present similar results using the multilevel MLC algorithm of Section 5 with three levels. Since we have demonstrated a clear advantage to using the 27-point stencil, in the remaining studies we will restrict our attention to that operator. In Table 8 we show the barrier fine-mesh errors obtained using the MLC-0 method for $\beta = 2, 3.25$. The errors for $\beta = 3.25$ are more than 18.4 times smaller than the errors for $\beta = 2$ and are nearly the same as the two-level method errors (Table 4). As predicted by the error analysis in Section 5, the error of MLC-0 is insensitive to the number of levels.

In Table 9 the errors obtained with the three-level MLC method are shown using $\alpha = 1.75$ and $\beta = 3.25$. Unlike the two-level results, the $P = 4$ errors are significantly poorer than the MLC-0 errors. For example, we recover the barrier errors only for $N = 4096$, as opposed to the $N = 512$ results for MLC-0. We can improve matters somewhat by increasing P , but even for this very smooth problem,

N	level	$P = 1$	$P = 4$	$P = 6$	$P = 8$
512	$l = 0$	1.4509×10^{-7}	1.1379×10^{-7}	5.8886×10^{-8}	4.2602×10^{-8}
	$l = 1$	4.9396×10^{-7}	1.0594×10^{-6}	1.0990×10^{-6}	3.0059×10^{-7}
	$l = 2$	5.2600×10^{-7}	1.0782×10^{-6}	1.1101×10^{-6}	1.4926×10^{-7}
1024	$l = 0$	1.1143×10^{-7}	1.9032×10^{-8}	9.5197×10^{-9}	4.1018×10^{-9}
	$l = 1$	2.2539×10^{-7}	1.6461×10^{-7}	9.9491×10^{-8}	2.3381×10^{-8}
	$l = 2$	2.3665×10^{-7}	1.6596×10^{-7}	9.9989×10^{-8}	2.3381×10^{-8}
2048	$l = 0$	3.8485×10^{-8}	5.7487×10^{-9}	5.0311×10^{-9}	
	$l = 1$	5.9923×10^{-8}	1.0143×10^{-8}	5.9864×10^{-9}	
	$l = 2$	6.1989×10^{-8}	1.0276×10^{-8}	6.0168×10^{-9}	
4096	$l = 0$	1.3028×10^{-8}	5.1364×10^{-9}		
	$l = 1$	1.6487×10^{-8}	5.2147×10^{-9}		
	$l = 2$	1.6861×10^{-8}	5.2621×10^{-9}		

Table 9. 3-level MLC: scaled maximum errors (52) at all levels using L_{27}^h . Here $\alpha = 1.75$ and $\beta = 3.25$. Compare with the second column of Table 8.

we do not get close to the barrier errors until $N = 2048$. This is consistent with the analysis in Section 5, and indicates that using higher values of P does not solve the problem. We will propose a different solution in Section 7.2.

7.2. An oscillatory charge test case. We further consider a case of three oscillatory charges that has been previously studied in [25]. The computational domain is again the unit cube $\Omega = [0, 1]^3$. Here we define a local charge density, whose support is a sphere of radius R_o centered at point $\mathbf{x}_o$, by

$$f_{\mathbf{x}_o}(\mathbf{x}) = \begin{cases} (1/R_o^3)(r - r^2)^2 \sin^2((\gamma/2)r), & r < 1, \\ 0, & r \geq 1, \end{cases}$$

$$r = \frac{1}{R_o} \|\mathbf{x} - \mathbf{x}_o\|, \quad \gamma = 4\mu\pi, \quad \mu = 7. \quad (54)$$

The exact solution associated with this charge density is given by

$$\phi_{\mathbf{x}_o}(\mathbf{x}) = \frac{1}{R_o} \begin{cases} -\frac{1}{120} - \frac{6}{\gamma^4}, & r = 0, \\ \frac{r^6}{84} - \frac{r^5}{30} + \frac{r^4}{40} + \frac{60}{\gamma^6} - \frac{9}{\gamma^4} - \frac{1}{120} + \frac{120}{\gamma^6 r} \\ + \left(-\frac{120}{\gamma^6 r} - \frac{9}{\gamma^4} + \frac{300}{\gamma^6} + \frac{36r}{\gamma^4} + \frac{r^2}{2\gamma^2} - \frac{30r^2}{\gamma^4} - \frac{r^3}{\gamma^2} + \frac{r^4}{2\gamma^2}\right) \cos(\gamma r) \\ + \left(\frac{12}{\gamma^5 r} - \frac{360}{\gamma^7 r} - \frac{96}{\gamma^5} + \frac{120r}{\gamma^5} - \frac{3r}{\gamma^3} + \frac{8r^2}{\gamma^3} - \frac{5r^3}{\gamma^3}\right) \sin(\gamma r), & r < 1, \\ \left(-\frac{1}{210} - \frac{12}{\gamma^4} + \frac{360}{\gamma^6}\right) \frac{1}{r}, & r \geq 1, \end{cases}$$

and is a pure monopole for $r \geq 1$. For our test case we consider three charges of the form (54), of radius $R_o = \frac{5}{100}$, centered at points $\mathbf{c}_1 = (\frac{3}{16}, \frac{7}{16}, \frac{13}{16})$, $\mathbf{c}_2 = (\frac{7}{16}, \frac{13}{16}, \frac{3}{16})$, and $\mathbf{c}_3 = (\frac{13}{16}, \frac{3}{16}, \frac{7}{16})$. The total charge and total potential are given via linear

N	level	error
2048	$l = 0$	9.59918×10^{-7}
	$l = 1$	1.00600×10^{-6}
	$l = 2$	1.04402×10^{-6}
4096	$l = 0$	5.82005×10^{-8}
	$l = 1$	6.47409×10^{-8}
	$l = 2$	6.71067×10^{-8}
8192	$l = 0$	8.42867×10^{-9}
	$l = 1$	8.42867×10^{-9}
	$l = 2$	8.44657×10^{-9}

Table 10. 3-level MLC-0: scaled maximum errors (52) using the L_{27}^h Mehrstellen Laplacian with $\beta = 3.25$.

N	level	error
2048	$l = 0$	1.03645×10^{-7}
	$l = 1$	9.59723×10^{-7}
	$l = 2$	1.00621×10^{-6}
	$l = 3$	1.04423×10^{-6}
4096	$l = 0$	2.93837×10^{-8}
	$l = 1$	5.84863×10^{-8}
	$l = 2$	6.50247×10^{-8}
	$l = 3$	6.73912×10^{-8}
8192	$l = 0$	7.84890×10^{-9}
	$l = 1$	8.78853×10^{-9}
	$l = 2$	8.78853×10^{-9}
	$l = 3$	8.79911×10^{-9}

Table 11. 4-level MLC-0: scaled maximum errors (52) using L_{27}^h with $\beta = 3.25$.

superposition by

$$f(\mathbf{x}) = f_{c_1}(\mathbf{x}) + f_{c_2}(\mathbf{x}) + f_{c_3}(\mathbf{x}),$$

$$\phi(\mathbf{x}) = \phi_{c_1}(\mathbf{x}) + \phi_{c_2}(\mathbf{x}) + \phi_{c_3}(\mathbf{x}).$$

We first present the results using three levels (Table 10) and four levels (Table 11) using MLC-0. The primary features of the convergence properties of the solution are that the errors are nearly uniform as a function of level, and are the same in both the three- and four-level cases. There is some indication of slowing down of the convergence rate on the finest two levels, but the convergence is still faster than $O(h^2)$.

N	level	$P = 1$	$P = 4$
2048	$l = 0$	1.09448×10^{-7}	1.07739×10^{-7}
	$l = 1$	9.60320×10^{-7}	9.57456×10^{-7}
	$l = 2$	1.00544×10^{-6}	1.00767×10^{-6}
	$l = 3$	1.04414×10^{-6}	1.04686×10^{-6}
4096	$l = 0$	3.58565×10^{-8}	3.03039×10^{-8}
	$l = 1$	5.81436×10^{-8}	7.22707×10^{-8}
	$l = 2$	6.55356×10^{-8}	6.44269×10^{-8}
	$l = 3$	6.74960×10^{-8}	6.95632×10^{-8}
8192	$l = 0$	2.88555×10^{-8}	1.92320×10^{-8}
	$l = 1$	1.61302×10^{-7}	7.46182×10^{-8}
	$l = 2$	1.63846×10^{-7}	7.61412×10^{-8}
	$l = 3$	1.64401×10^{-7}	7.61592×10^{-8}

Table 12. 4-level MLC: scaled maximum errors (52) using L_{27}^h . Here $\alpha = 2.25$ and $\beta = 3.25$.

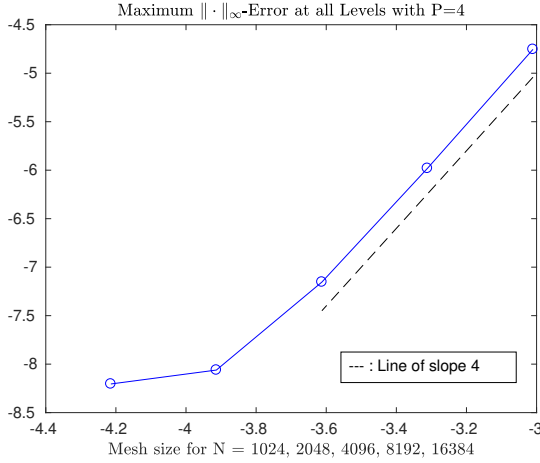


Figure 3. Log-log plot of greatest max norm error at all levels against mesh size using the L_{27}^h Mehrstellen Laplacian. Here fourth-order Legendre polynomials are employed at level 3. For levels 1 and 2, $\alpha = \beta = 3.25$, and $\alpha = 2.25$ at level 3.

In the MLC convergence results in Table 12, we see substantial deviations from the MLC-0 convergence results. The error shows no consistent behavior as a function of resolution, and in fact is worse at the finest resolution ($N = 8192$) in Table 12 than it is at the $N = 4096$ resolution in Table 11. We see no analogous problems in the MLC-0 calculations. Examining the error analysis in Section 5, we identified the terms in a three-level calculation that might lead to problems. Even in the smooth example above, it is clear that the increasing P does not have sufficient impact to solve this problem. A different approach, suggested by the form of the error, is to

N	level	α_l	$P = 1$	$P = 4$
1024	$l = 0$		6.75815×10^{-8}	6.75467×10^{-8}
	$l = 1$	3.25	9.06867×10^{-6}	9.07109×10^{-6}
	$l = 2$	3.25	1.68260×10^{-5}	1.68313×10^{-5}
	$l = 3$	2.25	1.76069×10^{-5}	1.76196×10^{-5}
2048	$l = 0$		1.03740×10^{-7}	1.03725×10^{-7}
	$l = 1$	3.25	9.59547×10^{-7}	9.59794×10^{-7}
	$l = 2$	3.25	1.00638×10^{-6}	1.00564×10^{-6}
	$l = 3$	2.25	1.04443×10^{-6}	1.04435×10^{-6}
4096	$l = 0$		2.96831×10^{-8}	2.95118×10^{-8}
	$l = 1$	3.25	5.81150×10^{-8}	5.83051×10^{-8}
	$l = 2$	3.25	6.45808×10^{-8}	6.48388×10^{-8}
	$l = 3$	2.25	6.86291×10^{-8}	7.02081×10^{-8}
8192	$l = 0$		7.52964×10^{-9}	7.73906×10^{-9}
	$l = 1$	3.25	8.60687×10^{-9}	8.68798×10^{-9}
	$l = 2$	3.25	8.72734×10^{-9}	8.68798×10^{-9}
	$l = 3$	2.25	8.64239×10^{-9}	8.68813×10^{-9}
16384	$l = 0$		5.94183×10^{-9}	5.97157×10^{-9}
	$l = 1$	3.25	6.16822×10^{-9}	6.20416×10^{-9}
	$l = 2$	3.25	6.20010×10^{-9}	6.24059×10^{-9}
	$l = 3$	2.25	6.21301×10^{-9}	6.24325×10^{-9}

Table 13. 4-level MLC: scaled maximum errors (52) using L_{27}^h with higher values of α at intermediate levels. Here $\beta = 3.25$ and $\alpha = \beta$ at levels 1 and 2 and $\alpha = 2.25$ at level 3. Compare with Table 11.

reduce the difference $\beta - \alpha$ at coarser levels. In fact, there is likely a mechanism for defining a systematic strategy for doing this, since $(\mathbb{I} - \mathbb{P})f^i$ is easily computed. We defer that to later work. For the moment, we demonstrate this by setting $\alpha = \beta$ at coarser levels, holding β fixed (Table 12). We see that we can recover exactly the errors in the MLC-0 calculation and moreover there is no appreciable difference in error by increasing P . In addition, the cost of increasing α at coarser levels has a small impact on the overall cost of a multiresolution calculation, since these are applied to calculations at the coarser resolutions, which remain a small fraction of the overall cost of the method, even with the increased values of α . In Figure 3 we present the error behavior for the case of Table 13 with $P = 4$. For $N = 1024$ – 4096 the error is fourth-order accurate as is expected from the error estimate (45) where term $O(h^4)$ dominates at coarser mesh resolutions. For $N \geq 8192$ the error reaches a plateau imposed by the barrier error term $O(\|f\|_\infty/\beta^q)$. This can be reduced

further by employing higher-order Mehrstellen discretizations of the Laplacian or larger values of parameter β .

8. Conclusions

We have presented a domain decomposition method for the numerical solution of Poisson's equation with infinite domain boundary conditions in three dimensions on a nested hierarchy of structured grids. The method is an extension of Anderson's method of local corrections for particles [3] to gridded data and generalizes the scheme of McCorquodale et al. [25]. In the present method, local potentials are computed as volume potentials of local charges up to an inner localization radius, combined with volume potentials induced by order- $(P - 1)$ truncated Legendre expansions of the local charges up to an outer localization radius. The remaining global coupling is represented using a coarse-grid version of the same representation. This generalizes the method in [25], which corresponds to the $P = 1$ special case in the current method. Also, in [25] the local potentials were computed by means of the James–Lackner representation [17; 18] of infinite domain boundary conditions. In the present work, this is replaced by a representation using discrete convolution operators, which can be computed efficiently using FFTs via Hockney's algorithm. This approach eliminates the complicated quadratures that are necessary for the extension of the James–Lackner algorithm to three dimensions, while the FFT-based approach leads to compact compute kernels that can be highly optimized. The resulting algorithm is well suited for high performance on HPC computing platforms made up of multicore processors; in [24], we will present a systematic study of the performance and scaling of the algorithm on such systems.

In this paper, we have focused primarily on the analytical foundations of the MLC method and have provided a detailed error analysis. The errors are of the form $O(h^P) + O(h^4) + O(h^2\beta^{-q}) + O(\beta^{-q})$, where h is the mesh spacing, β is the nondimensionalized outer localization radius which is independent of h , and q is the order of accuracy of the Mehrstellen operator on harmonic functions. Numerical experiments indicate that the observed convergence behavior of the method is consistent with the analysis. For computationally practical values of the localization radius, and using the 27-point Mehrstellen operator (for which $q = 6$), the barrier error corresponds to relative solution error norms of 10^{-8} – 10^{-9} . While the β^{-q} term looks like an $O(1)$ error relative to the mesh spacing h , it is better to think of it as a separate discretization parameter that governs the accuracy of the representation of the nonlocal coupling. Doubling β decreases the error by a factor of 2^{-q} , analogous to the impact of halving h .

For the two-level algorithm, the results indicate that, for a given choice of the Mehrstellen operator, the two localization radii, and $P = 4$, the method converges at

a rate in the range $O(h^4)$ – $O(h^2)$, until the error reaches the barrier, i.e., consistent with the error analysis. We have also defined and implemented the extension to more than two levels, following the approach in [25]. A preliminary analysis of that algorithm indicates the need to control errors at coarser levels coming from the field induced between the inner and outer localization radii by the truncation of the Legendre expansion. The analysis suggests that these might be controlled by increasing the inner localization radius α at coarser levels. The numerical examples indicate that the problem is real, and that the proposed solution represents a viable approach. More generally, an important question that needs to be addressed is turning the error analysis in this work into practical strategies for choosing discretization parameters. For example, what are the tradeoffs between decreasing $\beta - \alpha$ and decreasing h in order to improve the accuracy of a calculation, versus the cost of doing each? We will address these issues in [24].

There are various possible ways to extend the present work. Perhaps most straightforward are extensions to finite volume discretizations and the implementation of other boundary conditions on rectangular domains (including periodic boundary conditions) using a method-of-images approach. Another possibility would be to apply even higher-order Mehrstellen discretizations of the Laplacian to see whether it results in smaller values of the barrier errors than those reported in this work. As was seen in Section 7, the L_{27}^h ($q = 6$) Mehrstellen Laplacian leads to comparable barrier errors to those obtained using the L_{19}^h ($q = 4$) stencil, but using smaller localization radii, in a manner consistent with the $O(\beta^{-q})$ scaling of that error. It is possible to derive Mehrstellen stencils for which $q = 10$, with the stencil contained in a $5 \times 5 \times 5$ block around the evaluation point. This leads to only a modest increase in the computational cost and complexity: for example, the per patch computational cost of the most computationally intensive component of the algorithm—the local discrete convolutions—does not depend on the size of the stencil. Finally, it would be interesting to investigate extensions of this method to other elliptic problems in mathematical physics employing different Green’s functions and high-order discretizations of the associated differential operators. The error analysis of the method as extended to other kernels should be essentially the same as what is discussed in the present study. Moreover, Hockney’s algorithm is kernel-independent and can be readily applied with minor modifications. More generally, the present work uses some detailed analytic tools for understanding the discrete potential theory on locally structured grids associated with the combination of finite difference localization in [23] and the local interactions/local corrections construction underlying [3]. It would be interesting to go back to the original MLC method for particles and to other particle-grid methods, such as particle-in-cell and immersed boundary methods, and apply these tools to better understand the error properties of these methods.

Appendix A: L_{19}^h and L_{27}^h Mehrstellen discretizations of the Laplacian

The stencil coefficients for the L_{19}^h and L_{27}^h Mehrstellen Laplacians are $a_j = (1/h^2)b_{|j|}$, where $|j|$ is the number of nonzero components of j and b_k are defined as

$$b_0 = -4, \quad b_1 = \frac{1}{3}, \quad b_2 = \frac{1}{6}, \quad b_3 = 0 \quad (19\text{-point stencil}),$$

$$b_0 = -\frac{64}{15}, \quad b_1 = \frac{7}{15}, \quad b_2 = \frac{1}{10}, \quad b_3 = \frac{1}{30} \quad (27\text{-point stencil}).$$

The corresponding expressions for the truncation errors τ_{19}^h and τ_{27}^h for L_{19}^h and L_{27}^h , are given by

$$\tau_{19}^h(\phi) = \frac{h^2}{12}(\Delta(\Delta\phi)) + h^4 L^{(6)}(\phi) + O(h^6)$$

and

$$\begin{aligned} \tau_{27}^h(\phi) = \frac{h^2}{12}(\Delta(\Delta\phi)) + \frac{h^4}{360} \left(\left(\Delta^2 + 2 \left(\frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^2 \partial z^2} + \frac{\partial^4}{\partial z^2 \partial x^2} \right) \right) (\Delta\phi) \right) \\ + h^6 L^{(8)}(\phi) + O(h^8) \end{aligned}$$

where $L^{(q)}$ are homogeneous constant-coefficient q -th-order differential operators.

We need to compute an approximation to the discrete Green's function (8) for the 19-point and 27-point operators, restricted to a domain of the form $D = [-n, n]^3$. We do this by solving the following inhomogeneous Dirichlet problem on a larger domain $D_\zeta = [-\zeta n, \zeta n]^3$:

$$(L^{h=1} G^{h=1})[g] = \delta_0[g] \quad \text{for } g \in \mathcal{G}(D_\zeta, -1),$$

$$G^{h=1}[g] = G(g) \quad \text{for } g \in D_\zeta - \mathcal{G}(D_\zeta, -1),$$

where $G = G(x)$ is the Green's function (2) and L^h is either the 19-point or 27-point operator. Then our approximation to $G^{h=1}$ on D is the solution computed on D_ζ , restricted to D . To compute this solution, we put the inhomogeneous boundary condition into residual-correction form and solve the resulting homogeneous Dirichlet problem using the discrete sine transform. The error estimate (12) applied here implies that the error in replacing the correct discrete boundary conditions with those of the exact Green's function scales like $O((\zeta n)^{-4})$ in max norm. In the calculations presented here, we computed $G^{h=1}$ using $n \geq 128$ and $\zeta = 2$, leading to at least 10 digits of accuracy for $G^{h=1}$.

Appendix B: Hockney's method for fast evaluation of discrete convolutions

Hockney [16, pp. 180–181] (see also [9]) observed that discrete convolutions with one of the functions having support on a bounded domain in $\mathbb{Z}^D$, and evaluated on a bounded domain, can be computed exactly in terms of discrete Fourier transforms.

For completeness, we describe that method. We show this first for the case $\mathbf{D} = 1$, and state the general result for any number of dimensions. Given $\Psi, f : \mathbb{Z} \rightarrow \mathbb{R}$, $\text{supp}(f) \subseteq [0, b]$, we want to compute

$$(\Psi * f)[i] = (f * \Psi)[i] = \sum_{j \in \mathbb{Z}} f[i - j] \Psi[j], \quad i \in [0, n]. \quad (55)$$

First, we observe that the infinite sum can be replaced by a finite sum.

$$\sum_{j \in \mathbb{Z}} f[i - j] \Psi[j] = \sum_{j=-b'}^n f[i - j] \Psi[j], \quad i \in [0, n], \quad (56)$$

for any $b' \geq b$. Second, we observe that Ψ and f can be replaced in (56) by periodic extensions of those functions restricted to the interval $[-b', n]$:

$$\sum_{j=-b'}^n f[i - j] \Psi[j] = \sum_{j=-b'}^n \tilde{f}[i - j] \tilde{\Psi}[j], \quad i \in [0, n],$$

$$\tilde{f}[l], \tilde{\Psi}[l] \equiv f[l_{\text{mod}}], \Psi[l_{\text{mod}}], \quad l_{\text{mod}} = \text{mod}(l + b', (n + b' + 1)) - b'. \quad (57)$$

Finally, we express the periodic convolution in (57) in terms of discrete Fourier transforms:

$$\sum_{j=-b'}^n \tilde{f}[i - j] \tilde{\Psi}[j] = \mathcal{F}^{-1}(\mathcal{F}(\tilde{\Psi}) \cdot \mathcal{F}(\tilde{f}))[i], \quad (58)$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the discrete complex Fourier transform and its inverse on the interval $[-b', n] \subset \mathbb{Z}$.

This generalizes to rectangular domains in any number of dimensions. For example, for cubic domains, given $\Psi, f : \mathbb{Z}^{\mathbf{D}} \rightarrow \mathbb{R}^{\mathbf{D}}$, $\text{supp}(f) \subseteq [0, b]^{\mathbf{D}}$,

$$\sum_{j \in \mathbb{Z}^{\mathbf{D}}} \Psi[\mathbf{i} - \mathbf{j}] f[\mathbf{j}] = \mathcal{F}^{-1}(\mathcal{F}(\tilde{\Psi}) \cdot \mathcal{F}(\tilde{f}))[\mathbf{i}], \quad \mathbf{i} \in [0, n]^{\mathbf{D}}, \quad (59)$$

$$\tilde{f}[\mathbf{l}], \tilde{\Psi}[\mathbf{l}] \equiv f[\mathbf{l}_{\text{mod}}], \Psi[\mathbf{l}_{\text{mod}}], \quad (60)$$

$$(\mathbf{l}_{\text{mod}})_d = \text{mod}((\mathbf{l})_d + b', (n + b' + 1)) - b', \quad d = 0, \dots, \mathbf{D} - 1, \quad (61)$$

where $b' \geq b$ and $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the complex discrete Fourier transform and its inverse on the cube $[-b', n]^{\mathbf{D}} \subset \mathbb{Z}^{\mathbf{D}}$. In practice, this is efficient for a broad range of (b, n) since we can choose b' so that the radices of the FFTs are highly composite, with the size of the problem changing by only a small amount. In the case where $b = n$, the length of the domain doubles in each direction; hence, this is often referred to as Hockney's domain-doubling algorithm. However, in the present application, we want to use the more general case, since the size of the support of the localized charge distributions and the size of the grid on which the local fields are defined differ by a significant amount.

Acknowledgments

The authors would like to thank Brian Van Straalen and Peter McCorquodale for a number of helpful discussions. This research was supported at the Lawrence Berkeley National Laboratory by the Office of Advanced Scientific Computing Research of the U.S. Department of Energy (DOE) under Contract Number DE-AC02-05CH11231 and at the National Energy Research Scientific Computing Center by the DOE Petascale Initiative in Computational Science and Engineering.

References

- [1] A. S. Almgren, *A fast adaptive vortex method using local corrections*, Ph.D. thesis, University of California, Berkeley, 1991.
- [2] A. S. Almgren, T. Buttke, and P. Colella, *A fast adaptive vortex method in three dimensions*, J. Comput. Phys. **113** (1994), no. 2, 177–200. [MR](#) [Zbl](#)
- [3] C. R. Anderson, *A method of local corrections for computing the velocity field due to a distribution of vortex blobs*, J. Comput. Phys. **62** (1986), no. 1, 111–123. [MR](#) [Zbl](#)
- [4] G. T. Balls and P. Colella, *A finite difference domain decomposition method using local corrections for the solution of Poisson's equation*, J. Comput. Phys. **180** (2002), no. 1, 25–53. [MR](#) [Zbl](#)
- [5] G. T. Balls, *A finite-difference domain decomposition method using local corrections for the solution of Poisson's equation*, Ph.D. thesis, University of California, Berkeley, 1999. [MR](#) [Zbl](#)
- [6] J. Barnes and P. Hut, *A hierarchical $O(N \log N)$ force-calculation algorithm*, Nature **324** (1986), 446–449.
- [7] J. Carrier, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Statist. Comput. **9** (1988), no. 4, 669–686. [MR](#) [Zbl](#)
- [8] L. Collatz, *The numerical treatment of differential equations*, 3rd ed., Die Grundlehren der mathematischen Wissenschaften, no. 60, Springer, 1960. [MR](#) [Zbl](#)
- [9] J. W. Eastwood and D. R. K. Brownrigg, *Remarks on the solution of Poisson's equation for isolated systems*, J. Comput. Phys. **32** (1979), no. 1, 24–38. [MR](#) [Zbl](#)
- [10] L. C. Evans, *Partial differential equations*, 2nd ed., Graduate Studies in Mathematics, no. 19, American Mathematical Society, 2010. [MR](#) [Zbl](#)
- [11] A. Gholami, D. Malhotra, H. Sundar, and G. Biros, *FFT, FMM, or multigrid? A comparative study of state-of-the-art Poisson solvers for uniform and nonuniform grids in the unit cube*, SIAM J. Sci. Comput. **38** (2016), no. 3, C280–C306. [MR](#) [Zbl](#)
- [12] D. Gilbarg and N. S. Trudinger, *Elliptic partial differential equations of second order*, Springer, 2001. [MR](#) [Zbl](#)
- [13] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys. **73** (1987), no. 2, 325–348. [MR](#) [Zbl](#)
- [14] ———, *A new version of the fast multipole method for the Laplace equation in three dimensions*, Acta Numer. **6** (1997), 229–269. [MR](#) [Zbl](#)
- [15] P. Henrici, *Fast Fourier methods in computational complex analysis*, SIAM Rev. **21** (1979), no. 4, 481–527. [MR](#) [Zbl](#)
- [16] R. Hockney, *The potential calculation and some applications*, Method. Comput. Phys. **9** (1970), 135–211.

- [17] R. A. James, *The solution of Poisson's equation for isolated source distributions*, J. Computational Phys. **25** (1977), no. 2, 71–93. [MR](#) [Zbl](#)
- [18] K. Lackner, *Computation of ideal MHD equilibria*, Comput. Phys. Commun. **12** (1976), no. 1, 33–44.
- [19] M. H. Langston, L. Greengard, and D. Zorin, *A free-space adaptive FMM-based PDE solver in three dimensions*, Commun. Appl. Math. Comput. Sci. **6** (2011), no. 1, 79–122. [MR](#) [Zbl](#)
- [20] S. Liska and T. Colonius, *A parallel fast multipole method for elliptic difference equations*, J. Comput. Phys. **278** (2014), 76–91. [MR](#) [Zbl](#)
- [21] D. Malhotra and G. Biros, *PVFMM: a parallel kernel independent FMM for particle and volume potentials*, Commun. Comput. Phys. **18** (2015), no. 3, 808–830. [MR](#) [Zbl](#)
- [22] ———, *Algorithm 967: a distributed-memory fast multipole method for volume potentials*, ACM Trans. Math. Software **43** (2016), no. 2, 17. [MR](#) [Zbl](#)
- [23] A. Mayo, *Fast high order accurate solution of Laplace's equation on irregular regions*, SIAM J. Sci. Statist. Comput. **6** (1985), no. 1, 144–157. [MR](#) [Zbl](#)
- [24] P. McCorquodale, P. Colella, B. V. Straalen, and C. Kavouklis, *High-performance implementations of the method of local corrections on parallel computers*, preprint, 2018.
- [25] P. McCorquodale, P. Colella, G. T. Balls, and S. B. Baden, *A local corrections algorithm for solving Poisson's equation in three dimensions*, Commun. Appl. Math. Comput. Sci. **2** (2007), 57–81. [MR](#) [Zbl](#)
- [26] W. F. Spitz and G. F. Carey, *A high-order compact formulation for the 3D Poisson equation*, Numer. Methods Partial Differential Equations **12** (1996), no. 2, 235–243. [MR](#) [Zbl](#)
- [27] F. Vico, L. Greengard, and M. Ferrando, *Fast convolution with free-space Green's functions*, J. Comput. Phys. **323** (2016), 191–203. [MR](#)

Received October 5, 2016. Revised July 14, 2018.

CHRIS KAVOUKLIS: kavouklis1@llnl.gov

Computational Engineering Division, Lawrence Livermore National Laboratory, Livermore, CA, United States

PHILLIP COLELLA: pcollella@lbl.gov

Applied Numerical Algorithms Group, Computational Research Division,
Lawrence Berkeley National Laboratory, Berkeley, CA, United States