Communications in Applied Mathematics and Computational Science



AN ADAPTIVE LOCAL DISCRETE CONVOLUTION METHOD FOR THE NUMERICAL SOLUTION OF MAXWELL'S EQUATIONS

BORIS LO AND PHILLIP COLELLA

vol. 14 no. 1 2019

msp

COMM. APP. MATH. AND COMP. SCI.
Vol. 14, No. 1, 2019
dx.doi.org/10.2140/camcos.2019.14.105

AN ADAPTIVE LOCAL DISCRETE CONVOLUTION METHOD FOR THE NUMERICAL SOLUTION OF MAXWELL'S EQUATIONS

BORIS LO AND PHILLIP COLELLA

We present a numerical method for solving the free-space Maxwell's equations in three dimensions using compact convolution kernels on a rectangular grid. We first rewrite Maxwell's equations as a system of wave equations with auxiliary variables and discretize its solution from the method of spherical means. The algorithm has been extended to be used on a locally refined nested hierarchy of rectangular grids.

1. Introduction

We want to solve the free-space three-dimensional Maxwell's equations

$$\frac{\partial \boldsymbol{E}}{\partial t} = c\nabla \times \boldsymbol{B} - 4\pi \boldsymbol{J}, \tag{1}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = -c\nabla \times \boldsymbol{E}, \tag{2}$$

$$\nabla \cdot \boldsymbol{E} = 4\pi\rho, \tag{3}$$

$$\nabla \cdot \boldsymbol{B} = 0. \tag{4}$$

In our previous work [7], we considered Maxwell's equations in Fourier space, derived a real-space propagator for the system, and discretized the exact solution from Duhamel's formula. This propagator includes Helmholtz decomposition operators. The Helmholtz decomposition operators require global Poisson solves at every time step, which offsets the computational advantages of the local convolution kernel parts of the propagator.

In the present work, we get around this difficulty by applying a similar technique to an auxiliary system of equations instead of directly to Maxwell's equations. This auxiliary system is a system of wave equations for $\boldsymbol{E}, \boldsymbol{B}$ combined with constraints which, if satisfied initially, are satisfied for all time, such that the solutions of the auxiliary system are solutions to Maxwell's equations. We then apply Kirchhoff's formula to this system and discretize the resulting convolution

MSC2010: primary 65M55, 65M80; secondary 78-04.
Keywords: electromagnetics, Green's function, propagator method, adaptive mesh refinement.

equations. The convolution kernels from this propagator are the same as the local kernels for the transverse Maxwell's equations' propagator in [7], and thus, the same discretization techniques and domain decomposition can be applied. The locality of the convolution kernels allows us to naturally incorporate adaptive mesh refinement (AMR), where the domain is divided up into a nested hierarchy of rectangular grids at each refinement level.

In Section 2 we introduce the auxiliary system and show the analytic solution for Maxwell's equations in terms of a propagator with specified charges and currents. In Section 3, we describe the discretization process briefly, and discuss in detail the local discrete convolution method (LDCM) Maxwell solver for a single level and its extension to multiple levels. In Section 4 we present a number of numerical tests that show an implementation of our algorithm. Finally, in Section 5 we make some concluding remarks.

2. Problem statement and derivation of propagators

2.1. ***Maxwell's equations.*** Introducing $\boldsymbol{\Phi} \equiv \nabla \times \boldsymbol{B}$ and $\boldsymbol{\Psi} \equiv \nabla \times \boldsymbol{E}$, we rewrite Maxwell's equations, with ρ, $\boldsymbol{J}$ specified, as the auxiliary system of wave equations

$$\frac{\partial \boldsymbol{E}}{\partial t} = c\boldsymbol{\Phi} - 4\pi \boldsymbol{J}, \tag{5}$$

$$\frac{\partial \boldsymbol{\Phi}}{\partial t} = c\nabla^2 \boldsymbol{E} - 4\pi c \nabla \rho, \tag{6}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = -c\boldsymbol{\Psi}, \tag{7}$$

$$\frac{\partial \boldsymbol{\Psi}}{\partial t} = -c\nabla^2 \boldsymbol{B} - 4\pi \nabla \times \boldsymbol{J}. \tag{8}$$

If the initial conditions satisfy

$$\boldsymbol{\Psi} = \nabla \times \boldsymbol{E}, \tag{9}$$

$$\boldsymbol{\Phi} = \nabla \times \boldsymbol{B}, \tag{10}$$

$$\nabla \cdot \boldsymbol{E} = 4\pi\rho, \tag{11}$$

$$\nabla \cdot \boldsymbol{B} = 0, \tag{12}$$

then the auxiliary system is equivalent to the original Maxwell system. To show this, consider the four error quantities associated with the initial value constraints at $t = 0$:

$$\boldsymbol{K}_B = \boldsymbol{\Phi} - \nabla \times \boldsymbol{B}, \tag{13}$$

$$\boldsymbol{K}_E = \boldsymbol{\Psi} - \nabla \times \boldsymbol{E}, \tag{14}$$

$$D_B = \nabla \cdot \boldsymbol{B}, \tag{15}$$

$$D_E = \nabla \cdot \boldsymbol{E} - 4\pi\rho. \tag{16}$$

Using the auxiliary system (5)–(8), the four evolution equations associated with these quantities are given by

$$\frac{\partial \boldsymbol{K}_B}{\partial t} = c\nabla \times \boldsymbol{K}_E + c\nabla D_E, \tag{17}$$

$$\frac{\partial \boldsymbol{K}_E}{\partial t} = -c\nabla \times \boldsymbol{K}_B - c\nabla D_B, \tag{18}$$

$$\frac{\partial D_B}{\partial t} = -c\nabla \cdot \boldsymbol{K}_E, \tag{19}$$

$$\frac{\partial D_E}{\partial t} = c\nabla \cdot \boldsymbol{K}_B. \tag{20}$$

It is clear that if $\boldsymbol{K}_B, \boldsymbol{K}_E, D_B, D_E$ vanish at $t = 0$, then they remain zero for all time after. In particular, the symbol of the linear operator associated with these eight evolution equations has the eigenvalues $\pm ic|\boldsymbol{k}|$ each with a multiplicity of four. Since errors propagate away with the same wave speed, any error will not accumulate at a fixed location and be a potential source of numerical instability. The initial value problem (5)–(8) is well posed even if the initial-value constraints (13)–(16) are not satisfied. The constraints are required only so that the solution is equivalent to the solution to Maxwell's equations. Since the two systems are equivalent, the solutions for $\boldsymbol{E}, \boldsymbol{B}$ obtained from the auxiliary system will also be the solution to the original Maxwell system.

The solutions to (5)–(8) are given by Kirchhoff's formula using the method of spherical means [13, p. 231]. Defining the kernels $G^{\Delta t}$ and $H^{\Delta t}$ as

$$G^{\Delta t}(\boldsymbol{z}) \equiv \frac{\delta(|\boldsymbol{z}| - c\Delta t)}{4\pi c\Delta t}, \tag{21}$$

$$H^{\Delta t}(\boldsymbol{z}) \equiv \frac{1}{c}\frac{\partial}{\partial s}\left(\frac{\delta(|\boldsymbol{z}| - cs)}{4\pi cs}\right)\bigg|_{s=\Delta t}, \tag{22}$$

$G^{\Delta t}$ is a spherical delta distribution with radius $c\Delta t$. The action of the propagator on an arbitrary state vector $\boldsymbol{h}(\boldsymbol{x}) \equiv [\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{x})]^T$ with $\boldsymbol{f}, \boldsymbol{g} \in \mathbb{R}^3$ is given by

$$\mathcal{P}^{\Delta t}[\boldsymbol{h}] = \begin{bmatrix} H^{\Delta t} * \boldsymbol{f} + G^{\Delta t} * \boldsymbol{g} \\ G^{\Delta t} * \nabla^2 \boldsymbol{f} + H^{\Delta t} * \boldsymbol{g} \end{bmatrix}, \tag{23}$$

where the scalar convolution kernel with vector quantity is defined as convolution with each component and convolutions are defined spatially as

$$(K * f)(\boldsymbol{x}) \equiv \int_{\mathbb{R}^3} K(\boldsymbol{y}) f(\boldsymbol{x} - \boldsymbol{y})\, d\boldsymbol{y}. \tag{24}$$

In particular, the solution to (5)–(6) is then given by

$$\begin{pmatrix} \boldsymbol{E}(\boldsymbol{x}, t+\Delta t) \\ \boldsymbol{\Phi}(\boldsymbol{x}, t+\Delta t) \end{pmatrix} = \mathcal{P}^{\Delta t}\left[\begin{pmatrix} \boldsymbol{E}(\boldsymbol{x}, t) \\ \boldsymbol{\Phi}(\boldsymbol{x}, t) \end{pmatrix}\right] - 4\pi \int_t^{t+\Delta t} \mathcal{P}^{t+\Delta t-s}\left[\begin{pmatrix} \boldsymbol{J}(\boldsymbol{x}, s) \\ c\nabla\rho(\boldsymbol{x}, s) \end{pmatrix}\right] ds. \tag{25}$$

The propagator for (7)–(8) is the same as that for (5)–(6), with the substitution $\Delta t \to -\Delta t$. Thus, the solution is given by

$$\begin{pmatrix} \boldsymbol{B}(\boldsymbol{x}, t+\Delta t) \\ \boldsymbol{\Psi}(\boldsymbol{x}, t+\Delta t) \end{pmatrix} = \mathcal{P}^{-\Delta t}\left[\begin{pmatrix} \boldsymbol{B}(\boldsymbol{x}, t) \\ \boldsymbol{\Psi}(\boldsymbol{x}, t) \end{pmatrix}\right] - 4\pi \int_t^{t+\Delta t} \mathcal{P}^{-(t+\Delta t-s)}\left[\begin{pmatrix} 0 \\ \nabla \times \boldsymbol{J}(\boldsymbol{x}, s) \end{pmatrix}\right] ds. \quad (26)$$

It can be seen from the Fourier transforms of the convolution kernels that

$$G^{-\Delta t} * f = -G^{\Delta t} * f, \quad (27)$$

$$H^{-\Delta t} * f = H^{\Delta t} * f. \quad (28)$$

In addition

$$H^{\Delta t} * f = \frac{1}{ct} G^{\Delta t} * f - \sum_{i=1}^{3} G_i^{\Delta t} * \frac{\partial f}{\partial z_i}, \quad (29)$$

$$G_i^{\Delta t}(\boldsymbol{z}) = \frac{z_i \delta(|\boldsymbol{z}| - c\Delta t)}{4\pi c \Delta t}. \quad (30)$$

With these, we have fully specified the solutions, (25) and (26), in terms of convolution with weighted spherical delta distributions. We note that it can be shown directly that $\boldsymbol{\Psi}(\boldsymbol{x}, t+\Delta t) = \nabla \times \boldsymbol{E}(\boldsymbol{x}, t+\Delta t)$ and $\boldsymbol{\Phi}(\boldsymbol{x}, t+\Delta t) = \nabla \times \boldsymbol{B}(\boldsymbol{x}, t+\Delta t)$ given the constraints are satisfied at t. When ρ, $\boldsymbol{J}$ are not specified but functions of field variables, instead of using Kirchhoff's formula and a quadrature scheme one can use Lawson's method [6] for time integration.

3. Discretization approach

3.1. ***Single-level algorithm.*** We consider a rectangular domain discretized with a Cartesian grid with grid spacing h with open boundary conditions. The convolutions in (25)–(26) are approximated with discrete convolutions on the grid. This requires a discretized representation of the convolution kernels, $G^{\Delta t,h} \approx G^{\Delta t}(\boldsymbol{z})$ and $H^{\Delta t,h} \approx H^{\Delta t}(\boldsymbol{z})$, on the grid. $H^{\Delta t,h}$ is obtained by (29), so that the problem reduces to only creating discrete representations of (weighted) spherical delta distributions. We refer the reader to [7] for a detailed treatment of the discretization of the convolution kernels. The resulting discrete convolution kernels have compact support just like their continuous counterparts. Thus, the discrete convolutions can be computed exactly using Hockney's method [5].

The overall time-stepping algorithm is given in Algorithm 1. This defines the discrete evolution for $\boldsymbol{E}$, $\boldsymbol{B}$, since $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$ are computed at the beginning of every time step. The source term integrals are discretized using a closed Newton–Cotes quadrature scheme with step size $\Delta s = \Delta t/(M-1)$ where M is the number of

Initialize Newton–Cotes quadrature weights $\{w_m\}_{m=0}^{M}$
/* Create the convolution kernels with quadrature step size Δs and spacing h */
Compute $G^{\Delta s,h}$, and $H^{\Delta s,h}$
/* Begin time-stepping loop */
for $n = 1, 2, \ldots$ **do**
  /* Initialize the fields for this time step */
  /* Let $U^{(n),h} \approx U(n\Delta t, \boldsymbol{x})$ */
  $\boldsymbol{E}^{(n),h} \leftarrow \boldsymbol{E}^{(n-1),h}$, $\boldsymbol{B}^{(n),h} \leftarrow \boldsymbol{B}^{(n-1),h}$, $\boldsymbol{\Phi}^{(n),h} \leftarrow \nabla \times \boldsymbol{E}^{(n),h}$, $\boldsymbol{\Psi}^{(n),h} \leftarrow \nabla \times \boldsymbol{B}^{(n),h}$
  /* Begin quadrature loop */
  for $m = 1, 2, \ldots, M$ **do**
    /* Add in source terms evaluated at $t = (n-1)\Delta t + (m-1)\Delta s$ */
    $\boldsymbol{E}^{(n),h} \leftarrow \boldsymbol{E}^{(n),h} - w_m 4\pi \boldsymbol{J}^h$
    $\boldsymbol{\Phi}^{(n),h} \leftarrow \boldsymbol{\Phi}^{(n),h} - w_m 4\pi c \nabla \rho^h$
    $\boldsymbol{\Psi}^{(n),h} \leftarrow \boldsymbol{\Psi}^{(n),h} - w_m 4\pi \nabla \times \boldsymbol{J}^h$
    /* Apply propagator to the fields except final quadrature point */
    if $m < M$ **then**

$$\begin{bmatrix} \boldsymbol{E}^{(n),h} \\ \boldsymbol{\Phi}^{(n),h} \end{bmatrix} \leftarrow \begin{bmatrix} H^{\Delta s,h} * \boldsymbol{E}^{(n),h} + G^{\Delta s,h} * \boldsymbol{\Phi}^{(n),h} \\ (G^{\Delta s,h} * \nabla^2) * \boldsymbol{E}^{(n),h} + H^{\Delta s,h} * \boldsymbol{\Phi}^{(n),h} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{B}^{(n),h} \\ \boldsymbol{\Psi}^{(n),h} \end{bmatrix} \leftarrow \begin{bmatrix} H^{\Delta s,h} * \boldsymbol{B}^{(n),h} - G^{\Delta s,h} * \boldsymbol{\Psi}^{(n),h} \\ -(G^{\Delta s,h} * \nabla^2) * \boldsymbol{B}^{(n),h} + H^{\Delta s,h} * \boldsymbol{\Psi}^{(n),h} \end{bmatrix}$$

    end if
  end for
  /* Enforcing constraints */
  $\boldsymbol{E}^{(n),h} \leftarrow \boldsymbol{E}^{(n),h} + \eta(\mathcal{L}\boldsymbol{E}^{(n),h} - 4\pi \nabla \rho^h)$
  $\boldsymbol{B}^{(n),h} \leftarrow \boldsymbol{B}^{(n),h} + \eta \mathcal{L} \boldsymbol{B}^{(n),h}$
end for

Algorithm 1. Single-level LDCM for Maxwell's equations.

quadrature points. We choose a fixed step size quadrature because $\mathcal{P}^{t_1}[\mathcal{P}^{t_2}[\boldsymbol{U}]] = \mathcal{P}^{t_1+t_2}[\boldsymbol{U}]$, and therefore, we only need to create one propagator with step size Δs during initial setup.

Even though the divergence constraints are preserved by the continuous time evolution, deviations from (11)–(12) may be generated by discretization error. To help remedy this, we apply local filters [8] of the form

$$\boldsymbol{E} := \boldsymbol{E} + \eta(\mathcal{L}\boldsymbol{E} - 4\pi \nabla \rho), \tag{31}$$

$$\boldsymbol{B} := \boldsymbol{B} + \eta \mathcal{L} \boldsymbol{B}, \tag{32}$$

$$\mathcal{L}_{ij} = \partial_{x_i} \partial_{x_j}, \tag{33}$$

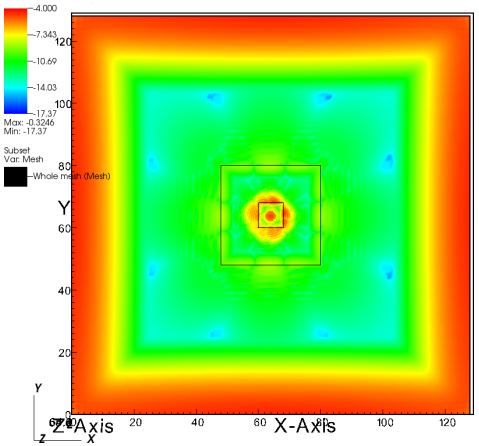


Figure 1. $\log_{10}(|\nabla \cdot \boldsymbol{E} - 4\pi\rho| / \max_{\boldsymbol{x}} 4\pi\rho)$ at $z = 0.5$ for the stopped translating spherical charge distribution problem at $t = \frac{200}{2048}$ for $N = 129$ showing that there are no reflected waves at the refinement boundaries.

where $\eta \sim \mathbb{O}(h^2)$ is a constant and $\mathcal{L}$ is a matrix-valued operator with the diagonal terms discretized with centered-difference approximations to the second derivative while the off-diagonal terms are products of centered-difference approximations to the first derivatives. This filtering step corresponds to applying an explicit diffusion step to the error in the longitudinal fields. Note that we do not have to do this for the curl constraints (9)–(10), since $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$ are reinitialized at the beginning of each time step.

3.2. ***Domain decomposition.*** Since the discretized version of the propagator involves only local operators, we can use standard domain decomposition to parallelize this algorithm. Consider a single-level domain, Ω_h, partitioned into rectangular patches. For each patch,

(1) at the beginning of each quadrature step, copy field values in ghost region from neighboring processors, and

(2) apply propagator to update local field values, invalidating values in ghost region.

The minimum width of the ghost region is determined by the size of the quadrature, Δs, and the order of the method because the size of the support of the spherical delta distributions is dependent on how far in time the fields are to be advanced.

For a point, $\boldsymbol{x}_k$, near the boundary, when applying the discrete convolutions we replace the field values outside the computational domain with the current field value at $\boldsymbol{x}_k$. This approximation leads to waves reflecting back into the computational domain. We could employ standard techniques for simulating infinite domain such as perfectly matched layer (PML) [2]. However, we wanted to focus on the

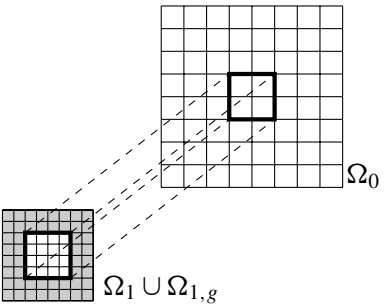


Figure 2. Example schematic of a two-level nested domain with factor of 2 refinement. The unshaded region is Ω_1 and the shaded region is the ghost region $\Omega_{1,g}$.

propagator method and not the boundary conditions. Therefore, in this work, we mitigate this reflection error with mesh refinement, by placing the boundary of the computational domain far away from the sources. This is possible because our method does not generate significant internal reflection at refinement boundaries as shown in Figure 1. The amplitude of the waves reaching the domain boundary will thus be weaker and the reflected error waves will also be smaller.

3.3. ***Multilevel algorithm.*** Consider now a hierarchy of nested rectangular grids, Ω_j, $j = 0, \ldots, J-1$, where the grid spacing for Ω^j is h/r^j for some refinement factor, $r \in \mathbb{Z}^+$, with $\Omega_j \cup \Omega_{j+1} = \Omega_{j+1}$, $j = 0, \ldots, J-2$. We introduce sampling and interpolation operators, $\mathcal{S}$ and $\mathcal{I}$, respectively, to communicate field values with the next immediate lower and upper levels. Similar to the ghost regions for each patch in parallelizing the single-level algorithm, we define a ghost region for each level, $\Omega_{j,g}$, where the width of the ghost region is determined by how far in time the fields are to be advanced. At the beginning of each quadrature step, except on the first level, for all nodes in $\Omega_{j,g}$ we interpolate $\boldsymbol{E}, \boldsymbol{B}$ from level $j-1$. After interpolating, except on the finest level, we replace $\boldsymbol{E}, \boldsymbol{B}$ at level j with field values from level $j+1$ on the nodes that are in $\Omega_j \cap \Omega_{j+1}$. A sample schematic of two levels with $r = 2$ is shown in Figure 2. After interpolating and sampling, each level is evolved independently with the propagator.

Let $f_j^{(n)}$ denote discretized f on level j and at time $t_n = n\Delta t$; the multilevel algorithm is outlined in Algorithm 2. Since (5)–(8) is a system of linear differential equations, we can use linear superposition to generate the overall solution to the problem in this multilevel setup; the solution is given by a composite where it takes the finest level values for any subdomain. For example, in the two-level case, let $\boldsymbol{U} = (\boldsymbol{E}, \boldsymbol{B}, \boldsymbol{\Psi}, \boldsymbol{\Phi})^T$; then the solution is given by

$$\boldsymbol{U}^{(n)} = \begin{cases} \boldsymbol{U}_1^{(n)} & \text{on } \Omega_1, \\ \boldsymbol{U}_0^{(n)} & \text{on } \Omega_0 \setminus \Omega_1. \end{cases} \tag{34}$$

Initialize Newton–Cotes quadrature weights $\{w_m\}_{m=0}^{M}$
for all levels Ω_j, $j = 0, \ldots, J-1$ **do**
    Initialize $\boldsymbol{U}_j^{(0)}$
    Compute $G^{\Delta s, h/r^j}$, and $H^{\Delta s, h/r^j}$
end for
`/* Begin time-stepping loop */`
for all $n = 1, 2, \ldots$ **do**
    for all levels Ω_j, $j = 0, \ldots, J-1$ **do**
        `/* Initialize the fields for this time step */`
        $\boldsymbol{U}_j^{(n)} \leftarrow \boldsymbol{U}_j^{(n-1)}$
        for quadrature step s **do**
            `/* Apply sampling operator except for level 0 */`
            $\boldsymbol{U}_{j-1}^{(n)} \leftarrow \mathcal{I}[\boldsymbol{U}_j^{(n)}]$ on Ω_j
            `/* Apply interpolation operator except for level` $J-1$ `*/`
            $\boldsymbol{U}_{j+1}^{(n)} \leftarrow \mathcal{I}[\boldsymbol{U}_j^{(n)}]$ on $\Omega_{j+1,g}$
            Apply single-level operations (add in source term and apply propagator)
        end for
    end for
    Sample and interpolate $\boldsymbol{E}$, $\boldsymbol{B}$ so that $\mathcal{L}$ can be applied on the refinement levels
    Enforce the constraints independently for each level
end for

Algorithm 2. Multilevel LDCM for Maxwell's equations.

Since we interpolate once every quadrature step, the width of $\Omega_{i,g}$ for level i has the same width as the ghost region required for domain decomposition.

Interpolation. We use high-order B-splines (see the Appendix) to interpolate the fields between levels similar to the ones used to regularize the delta distributions in the propagator. However, the choice of interpolant is more restrictive than the one used to regularize the delta distribution. The convergence of spherical quadrature when regularizing the delta distribution depends on the smoothness of the integrand [1]. However, we are interested in the regularized delta distribution as a discrete convolution kernel with some discretized function f. Numerically, the spherical quadrature and discrete convolution commute, and therefore, we relied on the smoothness of f for the convergence of the spherical quadrature. This allows us to use a C^0 high-order B-spline as a regularizer with the advantage that it has minimal support.

In this method, f is a field component or a component of the source terms. Since the field components must be sufficiently smooth for the spherical quadrature and the accuracy of the high-order finite difference operators applied to the field components also depends on smoothness, these translate into a smoothness requirement for the

for levels Ω_j, $j = 1, \ldots, J-1$ **do**
  if regrid **do**
    `/* Sample down starting from topmost level */`
    for $k = J-1, \ldots, j$ **do**
      $\boldsymbol{U}_{k-1}^{(n)} \leftarrow \mathcal{S}[\boldsymbol{U}_k^{(n)}]$ on $\Omega_k \cap \Omega_{j,\text{discard}}$
      `/* Discard part of domain that has been sampled from */`
      $\Omega_k \leftarrow \Omega_k \setminus (\Omega_k \cap \Omega_{j,\text{discard}})$
    end for
    $\Omega_j \leftarrow \Omega_j \cup \Omega_{j,\text{new}}$
    `/* Interpolate from level` $j-1$ `*/`
    $\boldsymbol{U}_j^{(n)} \leftarrow \mathcal{I}[\boldsymbol{U}_{j-1}^{(n)}]$ on $\Omega_{j,\text{new}}$
    Enforce the constraints
  end if
end for

Algorithm 3. Regridding algorithm.

interpolants. For a q-th-order method, we would need the error from the spherical quadrature to be at least $\mathbb{O}(h^q)$, which requires $f \in C^q$. Therefore, the interpolant must also be at least q-th-order accurate and C^q.

Regridding. For an adaptive version of this method, instead of a fixed hierarchy of rectangular grids, we regrid at the beginning of any time step as needed. Suppose we wish to regrid level j, $j > 0$; let $\Omega_j = \Omega_{j,\text{discard}} \cup \Omega_{j,\text{keep}}$ before regridding and $\Omega_j = \Omega_{j,\text{keep}} \cup \Omega_{j,\text{new}}$ after regridding. First sample down on $\Omega_{j,\text{discard}}$; then interpolate on $\Omega_{j,\text{new}}$ using the same sampling and interpolating operators. The regridding algorithm is outlined in Algorithm 3.

4. Numerical results

We implemented a fourth-order version of our Maxwell solver with $c = 1$; the one-step error for the solver is $\mathbb{O}(h^q)$, but after some number of time steps the total error will be $\mathbb{O}(h^{q-1})$ for a method that has a one-step error of $\mathbb{O}(h^q)$ and $\Delta t = \mathbb{O}(h)$. We used sixth-order centered differences for the spatial derivatives, the fifth-order $\frac{3}{8}$ Simpson's rule for the source integration, $W_{6,0}$ for the discrete delta distribution, and $W_{6,6}$ for the interpolation operator. The discrete convolutions are performed via Hockney's method extending the domain equal to the support of the discrete convolution kernels and using the FFTW library [4]. The domain at the coarsest level is a unit cube and each level is divided into 33^3 node patches with factor of 4 refinement; every level has the same number of nodes, N. The filter parameter at level j is $\eta_j = \frac{45}{544} h_j^2$. For each test, Δt is the same across refinement levels.

4.1. ***Translating spherical charge distribution.*** For the first numerical test, we used a C^6 spherical-support charge distribution with a spatially constant $\boldsymbol{v}(t)$.

$$\rho(\boldsymbol{x},t) = \begin{cases} a(r(t)-r(t)^2)^6, & r<1, \\ 0, & r\geq 1, \end{cases} \qquad r = \frac{1}{R_0}\|\boldsymbol{x}-\boldsymbol{x}_0\|, \tag{35}$$

$$\boldsymbol{J}(\boldsymbol{x},t) = \boldsymbol{v}(t)\rho(\boldsymbol{x},t), \tag{36}$$

$$\boldsymbol{v}(t) = \nu d\pi \tfrac{35}{16}\sin^7(2\pi\nu t)\hat{\boldsymbol{v}}. \tag{37}$$

The electrostatic solution is given by

$$\boldsymbol{E}(\boldsymbol{x}) = 4\pi R_0 a\hat{r}\begin{cases} \frac{1}{9}r^7 - \frac{3}{5}r^8 + \frac{15}{11}r^9 - \frac{5}{3}r^{10} + \frac{15}{13}r^{11} - \frac{3}{7}r^{12} + \frac{1}{15}r^{13}, & r<1 \\ \frac{1}{45045}r^{-2}, & r\geq 1, \end{cases} \tag{38}$$

$$\boldsymbol{B}(\boldsymbol{x}) = 0. \tag{39}$$

Here $\hat{r}$ is with respect to $\boldsymbol{x}_0$, and we use this as the initial condition for this test problem. We perform this test on fixed grids with two refinement levels, $\Omega_1 = \left[\frac{3}{8}, \frac{5}{8}\right]^3$ and $\Omega_2 = \left[\frac{15}{32}, \frac{17}{32}\right]^3$, with parameters $a = 10^4$, $d = \frac{1}{256}$, $\nu = \frac{1024}{80}$, $R_0 = \frac{1}{72}$, $\boldsymbol{x}_0 = \left(\frac{127}{256}, \frac{127}{256}, \frac{127}{256}\right)$, $\hat{\boldsymbol{v}} = \left(\cos\frac{\sqrt{3}}{3}\cos\frac{\sqrt{2}}{3}, \sin\frac{\sqrt{3}}{3}\cos\frac{\sqrt{2}}{3}, \sin\frac{\sqrt{2}}{3}\right)$, and $N = (65, 129, 257)$ with $\Delta t = \left(\frac{1}{1024}, \frac{1}{2048}, \frac{1}{4096}\right)$, respectively; this corresponds to CFL $= 1$ at the finest level, out to $t_{\text{final}} = \frac{200}{1024}$. Figure 3 shows the E_x Richardson convergence rate estimate and the associated ℓ_∞ error as well as the absolute convergence rate and associated ℓ_∞ errors for $\nabla\cdot\boldsymbol{E} - 4\pi\rho$ on the three grids in Ω_2 as a function of time step, and as expected our solution shows fourth-order convergence.

Electrostatic test. We performed another test with same discretization and parameters but stopped the charge distribution after $t = \frac{40}{1024}$ and then ran out to $t_{\text{final}} = \frac{100}{1024}$

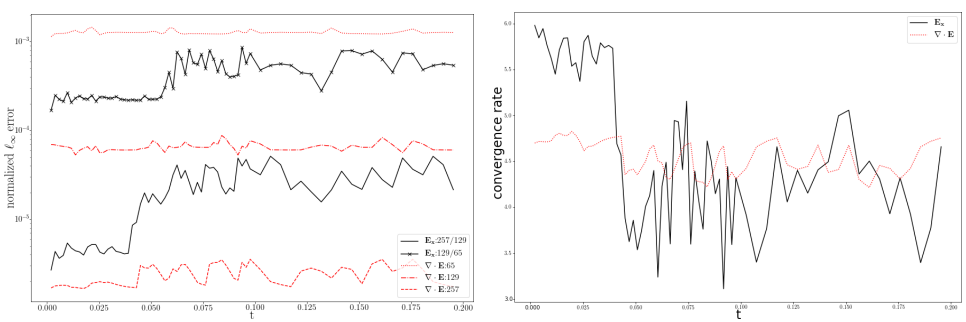


Figure 3. ℓ_∞ error values and convergence results for E_x and $\nabla\cdot\boldsymbol{E} - 4\pi\rho$ for the translating spherical charge distribution problem as a function of time in Ω_2. On the left are the normalized ℓ_∞ errors for E_x and $\nabla\cdot\boldsymbol{E} - 4\pi\rho$. The errors for E_x are obtained from the difference of sampled field values from $N = 257$ with $N = 129$ and also from sampled $N = 129$ with $N = 65$ test case. The E_x error is normalized by the max norm of the electrostatic solution (≈ 0.0694795), and $\nabla\cdot\boldsymbol{E} - 4\pi\rho$ error is normalized by $\max_{\boldsymbol{x}} 4\pi\rho \approx 30.6796$. On the right are the associated convergence rates.

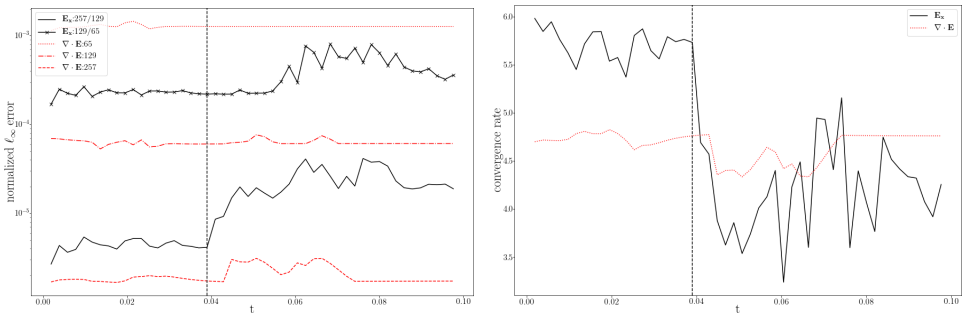


Figure 4. ℓ_∞ error values and convergence results for E_x and $\nabla \cdot \boldsymbol{E} - 4\pi\rho$ for the stopped spherical charge distribution problem as a function of time in Ω_2. On the left are the normalized ℓ_∞ errors for E_x and $\nabla \cdot \boldsymbol{E} - 4\pi\rho$. The errors for E_x are obtained from the difference of sampled field values from $N = 257$ with $N = 129$ and also from sampled $N = 129$ with $N = 65$ test case. The E_x error is normalized by the max norm of the electrostatic solution (≈ 0.0694795) and $\nabla \cdot \boldsymbol{E} - 4\pi\rho$ error is normalized by $\max_{\boldsymbol{x}} 4\pi\rho \approx 30.6796$. On the right are the associated convergence rates. The vertical line indicates the time at which the charge distribution stops moving.

to show that the solver recovers the electrostatic solution. Figure 4 shows the E_x Richardson convergence rate estimate and associated ℓ_∞ error as well as the absolute convergence rate and associated ℓ_∞ errors for $\nabla \cdot \boldsymbol{E} - 4\pi\rho$ on the three grids in Ω_2 as a function of time step, and as expected our solution shows fourth-order convergence.

Regridding test. We tested our regridding algorithm with the translating charge distribution with $\boldsymbol{v} = vd\pi \sin(2vt)\hat{\boldsymbol{x}}$, $a = \frac{1}{160}$, $d = \frac{1}{64}$, $\boldsymbol{x}_0 = \left(\frac{31}{64}, \frac{1}{2}, \frac{1}{2}\right)$, $v = \frac{1024}{80}$, $t_{\text{final}} = \frac{800}{1024}$, and other parameters being the same. We kept Ω_1 the same and fixed, but regridded Ω_2 starts with $\Omega_{2,a}$ and changes between $\Omega_{2,a}$ and $\Omega_{2,b}$ whenever the x coordinate of the center of the charge distribution crosses $\frac{63}{128}$, where $\Omega_{2,a}$ is the rectangular prism defined by the corner points $\left(\frac{29}{64}, \frac{17}{32}, \frac{17}{32}\right)$ and $\left(\frac{33}{64}, \frac{17}{32}, \frac{17}{32}\right)$, and $\Omega_{2,b} = \left[\frac{15}{32}, \frac{17}{32}\right]^3$; effectively Ω_2 oscillates in the x direction with amplitude $\frac{1}{64}$ in the direction of the charge motion. Figure 5 shows E_x and the regridding domains for $N = 129$. Figure 6 shows the E_x Richardson convergence rate estimate and the associated ℓ_∞ error as well as the absolute convergence rate and associated ℓ_∞ errors for $\nabla \cdot \boldsymbol{E} - 4\pi\rho$ on the three grids in Ω_2 as a function of time step and our solution shows fifth-order convergence.

4.2. ***Divergence-free current source.*** We've also tested with a divergence-free current source of the form

$$J_x(x, y, z, t) = -100\frac{y - y_0}{r} \sin\frac{\pi r}{2a} \cos^{10}\frac{\pi r}{2a} \cos^{11}\frac{\pi(z - z_0)}{d} \sin(2\pi vt), \quad (40)$$

$$J_y(x, y, z, t) = \quad 100\frac{x - x_0}{r} \sin\frac{\pi r}{2a} \cos^{10}\frac{\pi r}{2a} \cos^{11}\frac{\pi(z - z_0)}{d} \sin(2\pi vt), \quad (41)$$

$$J_z(x, y, z, t) = 0, \quad (42)$$

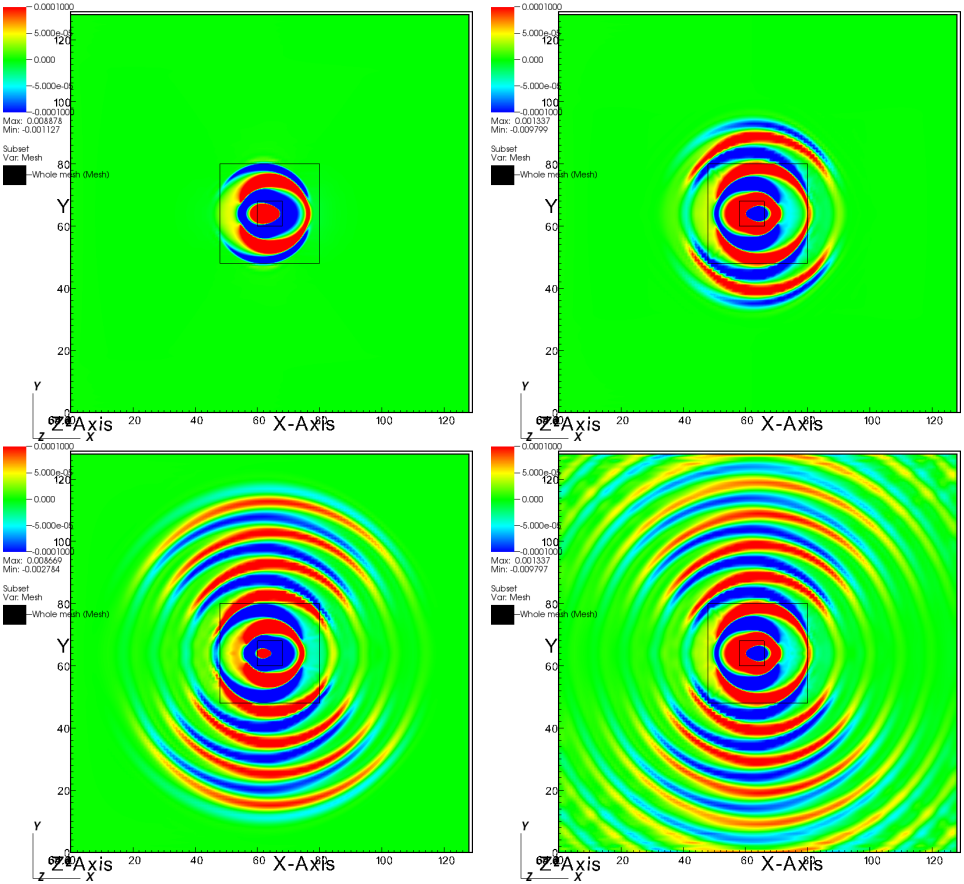


Figure 5. E_x minus the instantaneous electrostatic solution, at $z = \frac{1}{2}$, for the spherical charge distribution problem with regridding for $N = 129$. Top left: $t = \frac{256}{2048}$, charge distribution moving to the right, has almost reached its rightmost position, $\Omega_2 = \Omega_{2,b}$. Top right: $t = \frac{480}{2048}$, charge distribution is at its leftmost position, $\Omega_2 = \Omega_{2,a}$. Bottom left: $t = \frac{864}{2048}$, charge distribution moving to the left, $\Omega_2 = \Omega_{2,b}$. Bottom right: $t = \frac{1600}{2048}$, final time step, $\Omega_2 = \Omega_{2,a}$.

where $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ with parameters $a = \frac{3}{160}$, $d = \frac{13}{320}$, $x_0 = y_0 = z_0 = 0.5$, and $\nu = 20$, and using the same refinement levels, discretization, and t_{final} as the fixed-grids translating-charge problem. Figure 7 shows the E_x Richardson convergence rate estimate and the associated ℓ_∞ error as well as the absolute convergence rate and associated ℓ_∞ errors for $\nabla \cdot \boldsymbol{E}$ on the three grids in Ω_2 as a function of time step, and as expected our solution shows fourth-order convergence.

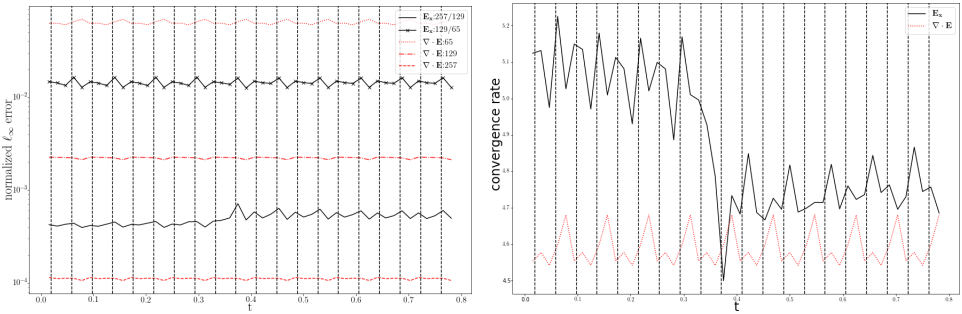


Figure 6. ℓ_∞ error values and convergence results for E_x and $\nabla \cdot \boldsymbol{E} - 4\pi\rho$ for the regridding spherical charge distribution problem as a function of time in Ω_2. On the left are the normalized ℓ_∞ errors for E_x and $\nabla \cdot \boldsymbol{E} - 4\pi\rho$. The errors for E_x are obtained from the difference of sampled field values from $N = 257$ with $N = 129$ and also from sampled $N = 129$ with $N = 65$ test case. The E_x error is normalized by the max norm of the electrostatic solution (≈ 0.0312658), and $\nabla \cdot \boldsymbol{E} - 4\pi\rho$ error is normalized by $\max_{\boldsymbol{x}} 4\pi\rho \approx 30.6796$. On the right are the associated convergence rates. The vertical lines are the times at which regridding occurs.

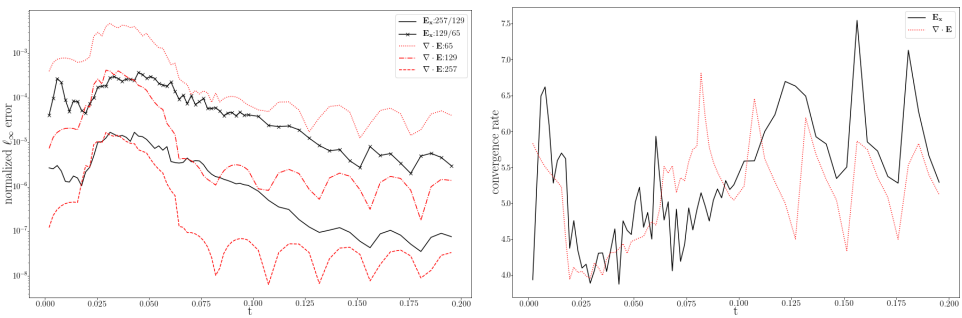


Figure 7. ℓ_∞ error values and convergence results for E_x and $\nabla \cdot \boldsymbol{E} - 4\pi\rho$ for the divergence-free current problem as a function of time in Ω_2. On the left are the normalized ℓ_∞ errors for E_x and $\nabla \cdot \boldsymbol{E} - 4\pi\rho$. The errors for E_x are obtained from the difference of sampled field values from $N = 257$ with $N = 129$ and also from sampled $N = 129$ with $N = 65$ test case. The E_x error is normalized by $|(4\pi/\nu) \max_{r,z} J_x| \approx |10.2341 \sin(2\pi\nu t)|$ and $\nabla \cdot \boldsymbol{E}$ is normalized by $|(4\pi/\nu a) \max_{r,z} J_x| \approx |545.8187 \sin(2\pi\nu t)|$. On the right are the associated convergence rates.

5. Conclusion

We have presented a new version of our Green's function numerical method for Maxwell's equations. This new formulation results in a completely local propagator that does not require Helmholtz decomposition. In principle, the method can choose any CFL but at the cost of larger ghost regions. We have demonstrated a high-order adaptive version of the solver in some test examples. In the future, we are interested in incorporating this method in EM PIC using Lawson's method where the fields and

particles are evolved together with a Runge–Kutta scheme with an extra propagator step for the fields.

Appendix: High-order B-splines

For completeness, we give the B-splines used in our implementation for the delta approximants and interpolants. Detailed discussions on creating high-order B-splines are given in [7; 3; 12; 11; 10; 9]. $W_{q,p}$ denotes a q-th-order accurate, C^p B-spline:

$$W_{6,0}(x) = \begin{cases} -\frac{1}{12}|x|^5+\frac{1}{4}|x|^4+\frac{5}{12}|x|^3-\frac{5}{4}|x|^2-\frac{1}{3}|x|+1, & |x|\in[0,1], \\ \frac{1}{24}|x|^5-\frac{3}{8}|x|^4+\frac{25}{24}|x|^3-\frac{5}{8}|x|^2-\frac{13}{12}|x|+1, & |x|\in[1,2], \\ -\frac{1}{120}|x|^5+\frac{1}{8}|x|^4-\frac{17}{24}|x|^3+\frac{15}{8}|x|^2-\frac{137}{60}|x|+1, & |x|\in[2,3], \\ 0, & |x|>3, \end{cases} \tag{43}$$

$$W_{6,6}(x) = \begin{cases} -\frac{665}{12048}|x|^9+\frac{665}{3012}|x|^8-\frac{2419}{12048}|x|^7-\frac{2437}{12048}|x|^6 \\ \quad +\frac{2723}{3012}|x|^4-\frac{4543}{3012}|x|^2+\frac{19177}{21084}, & |x|\in[0,1], \\ \frac{133}{4016}|x|^9-\frac{399}{1004}|x|^8+\frac{39659}{20080}|x|^7-\frac{104409}{20080}|x|^6+\frac{23443}{3012}|x|^5 \\ \quad -\frac{14175}{2008}|x|^4+\frac{7553}{1506}|x|^3-\frac{32207}{10040}|x|^2+\frac{2933}{15060}|x|+\frac{13081}{14056}, & |x|\in[1,2], \\ -\frac{133}{12048}|x|^9+\frac{665}{3012}|x|^8-\frac{114139}{60240}|x|^7+\frac{109283}{12048}|x|^6-\frac{79303}{3012}|x|^5 \\ \quad +\frac{283423}{6024}|x|^4-\frac{75215}{1506}|x|^3+\frac{170023}{6024}|x|^2-\frac{90923}{15060}|x|-\frac{17653}{42168}, & |x|\in[2,3], \\ \frac{19}{12048}|x|^9-\frac{133}{3012}|x|^8+\frac{225859}{421680}|x|^7-\frac{221003}{60240}|x|^6+\frac{23299}{1506}|x|^5 \\ \quad -\frac{30793}{753}|x|^4+\frac{49184}{753}|x|^3-\frac{208208}{3765}|x|^2\frac{53632}{3765}|x|+\frac{32512}{5271}, & |x|\in[3,4], \\ 0, & |x|>4. \end{cases} \tag{44}$$

Acknowledgments

This research is supported by the Office of Advanced Scientific Computing Research of the U.S. Department of Energy under contract number DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility.

References

[1] K. Atkinson, *Numerical integration on the sphere*, J. Austral. Math. Soc. B **23** (1982), no. 3, 332–347. MR Zbl

[2] J.-P. Berenger, *A perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys. **114** (1994), no. 2, 185–200. MR Zbl

[3] A. K. Chaniotis and D. Poulikakos, *High order interpolation and differentiation using B-splines*, J. Comput. Phys. **197** (2004), no. 1, 253–274. MR Zbl

[4] M. Frigo and S. G. Johnson, *FFTW: an adaptive software architecture for the FFT*, Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (Piscataway, NJ), vol. III: Digital signal processing, IEEE, 1998, pp. 1381–1384.

[5] R. Hockney, *Potential calculation and some applications*, Method. Comput. Phys. **9** (1970), 136–212.

[6] J. D. Lawson, *Generalized Runge–Kutta processes for stable systems with large Lipschitz constants*, SIAM J. Numer. Anal. **4** (1967), 372–380. MR Zbl

[7] B. Lo, V. Minden, and P. Colella, *A real-space Green's function method for the numerical solution of Maxwell's equations*, Commun. Appl. Math. Comput. Sci. **11** (2016), no. 2, 143–170. MR Zbl

[8] B. Marder, *A method for incorporating Gauss' law into electromagnetic PIC codes*, J. Comput. Phys. **68** (1987), no. 1, 48–55. Zbl

[9] J. J. Monaghan, *Extrapolating B-splines for interpolation*, J. Comput. Phys. **60** (1985), no. 2, 253–262. MR Zbl

[10] I. J. Schoenberg, *Contributions to the problem of approximation of equidistant data by analytic functions, A: On the problem of smoothing or graduation, a first class of analytic approximation formulae*, Quart. Appl. Math. **4** (1946), 45–99. MR Zbl

[11] A.-K. Tornberg and B. Engquist, *Numerical approximations of singular source terms in differential equations*, J. Comput. Phys. **200** (2004), no. 2, 462–488. MR Zbl

[12] J. Waldén, *On the approximation of singular source terms in differential equations*, Numer. Methods Partial Differential Equations **15** (1999), no. 4, 503–520. MR Zbl

[13] G. B. Whitham, *Linear and nonlinear waves*, Wiley-Interscience, New York, 1974. MR Zbl

Received April 29, 2018. Revised February 4, 2019.

BORIS LO: bt.lo@berkeley.edu
Applied Science and Technology, University of California, Berkeley, Berkeley, CA, United States
and
Lawrence Berkeley National Laboratory, Berkeley, CA, United States

PHILLIP COLELLA: pcolella@lbl.gov
Applied Numerical Algorithms Group, Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA, United States