Journal of Computational Physics **177,** 233–263 (2002)
doi:10.1006/jcph.2001.6985, available online at http://www.idealibrary.com on IDEAL®

Simulation of Laser Plasma Filamentation Using Adaptive Mesh Refinement[1]

Milo R. Dorr, F. Xabier Garaizar, and Jeffrey A. F. Hittinger

Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, California 94551
E-mail: dorr1@llnl.gov, garaizar1@llnl.gov, hittinger1@llnl.gov

Received June 28, 2001; revised December 10, 2001

We investigate the use of adaptive mesh refinement in the simulation of laser plasma filamentation. A numerical algorithm is constructed to solve model equations consisting of a fluid approximation of a quasineutral plasma combined with a paraxial light propagation model. The algorithm involves high-resolution plasma and light model discretizations on a block-structured, locally refined grid hierarchy, which is dynamically modified during the time integration to follow evolving fine-scale solution features. Comparisons of the efficiency of this approach to that of uniform grid calculations are presented. © 2002 Elsevier Science (USA)

Key Words: laser plasma interaction; numerical algorithms; Godunov method; adaptive mesh refinement; plasma fluid model; paraxial wave equation.

1. INTRODUCTION

In this paper, we investigate the use of adaptive mesh refinement (AMR) in the simulation of laser plasma interaction (LPI). The motivating physical context is that of inertial confinement fusion experiments, in which laser beams are used to drive the implosion of a fuel capsule with the goal of igniting a self-sustained reaction [29]. In such experiments, the rapid ionization of the capsule and surrounding materials generates a dynamic, plasma-filled region through which the laser pulse must continue to propagate without substantial perturbation to achieve the desired distribution of energy at the target. The ability to predict and control LPI is therefore critical in the design of laser-driven fusion experiments. In recent years, computational models have become an increasingly important complement to theoretical analysis and experimentation in LPI research [4, 5, 22]. The goal of this paper is to consider the application of AMR techniques to increasing the efficiency of LPI simulations and the range of problems that can be numerically addressed.

[1] This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory, under Contract W-7405-Eng-48.

0021-9991/02 $35.00

A variety of LPI mechanisms exist. In this paper, we consider filamentation instabilities, which are initiated by the refraction of laser light into regions of low relative plasma density. The resulting localized increase in laser intensity causes a decrease in density through heating and/or ponderomotive forces, further increasing the refraction of light. This instability is eventually limited by diffraction, by heating, or in extreme cases by expulsion of all plasma, but potentially not before the transport of laser energy is significantly altered. Moreover, filamentation interacts with other LPI processes, in particular parametric instabilities, which grow fastest in filaments.

A major challenge of simulating filamentation in actual ignition-scale experimental configurations is the need to accommodate a wide range of physical scales. The "speckles" from which filaments form represent a fine scale requiring mesh cells on the order of a light wavelength. It is known [4] that the speckle dimensions are approximately $f\lambda$ transverse to the light propagation direction and $8f^2\lambda$ along the propagation axis, where f is the f-number of the focusing lens (e.g., $f = 8$) and λ is the light wavelength. In contrast, the diameter of the beams in which filamentation occurs can be much larger. For example, a section of a single beam of the 192-beam laser system being constructed at the National Ignition Facility (NIF) [20] at Lawrence Livermore National Laboratory will span approximately 2000 wavelengths of 351-nm light. Modeling a full NIF beam with a mesh sufficiently small to resolve filament speckles therefore represents a formidable computational problem.

This situation motivates our investigation of AMR to apply computational resources where they are most needed, e.g., to resolve speckles, while using coarser grids in other parts of the simulation region that do not require such high resolution. The gains to be expected from such an approach are problem dependent; AMR is not a technique that should be applied in every circumstance. Local grid refinement is effective when capturing isolated regions of rapid variation, but it is less appropriate when the computational domain is dominated by rapidly varying structures. So if the goal is to resolve a few isolated speckles in a large beam, a significant efficiency increase can be realized from local mesh refinement. On the other hand, if a large portion of the beam is highly filamented, the utility of local refinement within the beam is limited. This can occur, for example, when beam smoothing techniques are used to modify the beam so as to avoid the formation of a few large, catastrophic filaments at the cost of generating many smaller, though more benign, ones. Even in such cases, however, AMR can help expand the simulation region by enabling the use of coarser resolution outside the beam. Simulation of the interaction of multiple beams can also be facilitated by the ability to use coarser meshes in the regions separating the beams.

Since our interest here is in the application of AMR to filamentation problems, not in the development or evaluation of models of the physical processes, we adopt with one modification a mathematical model developed in [4, 22] that we feel is sufficient to demonstrate the advantages of the AMR methodology. This mathematical description combines a two-fluid model of the plasma with a paraxial model of laser light propagation suitable for ponderomotively driven filamentation. We depart from this model by neglecting electron heat flow because of unresolved issues in the modeling of this nonlocal transport effect.

As in [22], we assume quasineutrality of the plasma, which results in a system of equations for conservation of mass, momentum, and energy similar to those for a neutral, ideal gas. These equations are augmented by source terms representing the ponderomotive force exerted by the light field on the plasma. Such a model describes behavior on timescales much slower than the laser frequency, e.g., on the relatively slow ion acoustic timescale. AMR provides the capability to resolve the scales on which the source terms act, so these sources are treated explicitly in an operator split integration with a high-resolution upwind discretization used to advance the fluid equations.

The light propagation model from [4, 22] is obtained from Maxwell's equations using a variety of assumptions. In particular, the assumption of fixed polarization, time-harmonic waves yields a reduction to a Helmholtz equation for the light field in the usual manner. The difficulties encountered in solving Helmholtz equations with variable refractive index in computational domains thousands of wavelengths in diameter motivate the use of paraxial approximations, which assume that the light propagates primarily in an *a priori* known direction and does not scatter at large angles. This allows much of the high-frequency oscillation in the light field to be removed by introducing an appropriate envelope, which is computed by a sweep of the computational domain in the light propagation direction starting from a prescribed incident field on the boundary. On uniform grids, an efficient method to integrate the resulting paraxial equation involves the use of fast Fourier transforms (FFTs) to diagonalize a certain differential operator transverse to the propagation direction [11]. Since the use of FFTs on locally refined grids is problematic, we consider here a discretization based on finite differences, where the transverse differential operator is inverted by a linear system solve.

To introduce local mesh refinement into the solution of the coupled plasma and light systems, we employ a Cartesian, block-structured strategy. In this approach, the computational domain consists of a hierarchy of refinement levels. Each refinement level is a disjoint union of rectangular grids obtained by refining a subregion of the next coarser level by a fixed ratio in each coordinate direction. The location and number of refinement levels can be dynamically modified during the calculation to follow evolving solution features. The integration of the laser plasma system on the locally refined hierarchy is accomplished by a coordinated integration of the plasma and light systems on individual refinement levels, combined with synchronization steps to enforce the mathematically required compatibility conditions across levels. This type of approach has been employed to solve general hyperbolic systems [3], as well as problems in computational fluid dynamics [1, 2], combustion [19], solid mechanics [13, 25], and plasma processing [7, 8]. The new wrinkle in the current context is the use of this approach to solve the plasma fluid model combined with the paraxial light propagation algorithm.

In Section 2, we describe the laser plasma filamentation model. The numerical discretization and solution of the coupled system on a uniform grid is presented in Section 3. The algorithm is generalized to locally refined grids in Section 4, and numerical results are presented in Section 5. Section 6 contains some conclusions and identifies areas for further investigation.

2. LASER PLASMA FILAMENTATION MODEL

Our plasma model is derived from the Poisson–Euler equations, which consist of Euler equations for the evolution of mass, momentum, and energy of both the ions and the electrons coupled by Poisson's equation for the electric potential. We augment this system with a source term representing the ponderomotive force exerted on the plasma by the ambient laser light field. Let n_α, u_α, p_α, T_α, and m_α represent the number density, velocity, pressure, temperature, and mass of the ions ($\alpha = i$) and electrons ($\alpha = e$), respectively. Let e denote the electron charge magnitude, Z the ionization state, and ϵ_0 the permittivity of free space. The Poisson–Euler equations are expressed as

$$\begin{aligned} \partial_t n_i + \nabla \cdot (n_i u_i) &= 0, & (1)\\ \partial_t (m_i n_i u_i) + \nabla \cdot (m_i n_i u_i \otimes u_i) + \nabla p_i &= -Zen_i \nabla \phi, & (2)\\ (\gamma - 1)^{-1}[\partial_t p_i + \nabla \cdot (p_i u_i)] + p_i \nabla \cdot u_i &= 0 & (3) \end{aligned}$$

for the ions,

$$\begin{aligned} \partial_t n_e + \nabla \cdot (n_e u_e) &= 0, & (4)\\ \partial_t (m_e n_e u_e) + \nabla \cdot (m_e n_e u_e \otimes u_e) + \nabla p_e &= en_e \nabla \phi + F_p, & (5)\\ (\gamma - 1)^{-1}[\partial_t p_e + \nabla \cdot (p_e u_e)] + p_e \nabla \cdot u_e &= 0 & (6) \end{aligned}$$

for the electrons, and

$$\epsilon_0 \Delta \phi = e\,(n_e - Zn_i) \tag{7}$$

for the potential ϕ. We have implicitly assumed here that both the ions and the electrons may be treated as ideal gases with the same ratio of specific heats γ. The source term F_p represents the ponderomotive force exerted by the laser light on the plasma (see, e.g., Section 2.8 of [18]), given by

$$F_p \equiv -\frac{\epsilon_0 \omega_p^2}{2\omega^2} \nabla |\mathcal{E}|^2, \tag{8}$$

where $\mathcal{E}$ and ω are the light field and frequency, respectively, and

$$\omega_p \equiv \left(\frac{e^2 n_e}{\epsilon_0 m_e} \right)^{1/2} \tag{9}$$

is the plasma frequency. The model from which $\mathcal{E}$ is obtained is described below.

The Euler–Poisson system can be simplified by the assumption of quasineutrality, which holds in many filamentation problems of interest. Under appropriate nondimensionalization, Eq. (7) becomes

$$\lambda_D^2 \nabla^2 \hat{\phi} = Z\hat{n}_i - \hat{n}_e, \tag{10}$$

where the hats denote dimensionless quantities and λ_D is proportional to the plasma Debye length. When the spatial variation of the potential is large compared to the Debye length ($\lambda_D \to 0$), Eq. (10) reduces to $Z\hat{n}_i \approx \hat{n}_e$, the quasineutral approximation. In this limit, we may define a single fluid density $n \equiv n_i = n_e/Z$ and, further restricting our interest to low-frequency scales, a single fluid velocity $u \equiv u_i = u_e$ (see [14]). These assumptions imply that (1) and (4) reduce to the same equation and furthermore, adding (2) and (5), we obtain a new conservation of momentum equation with a total mass defined by $m \equiv m_i + Zm_e$. Replacing the ion pressure equation (3) by its sum with the electron pressure equation (6), we obtain the system

$$\partial_t n + \nabla \cdot (nu) = 0, \tag{11}$$

$$\partial_t (mnu) + \nabla \cdot (mnu \otimes u) + \nabla p = F_p, \tag{12}$$

$$(\gamma - 1)^{-1}[\partial_t p + \nabla \cdot (pu)] + p\nabla \cdot u = 0, \tag{13}$$

$$(\gamma - 1)^{-1}[\partial_t p_e + \nabla \cdot (p_e u)] + p_e \nabla \cdot u = 0, \tag{14}$$

involving the total pressure $p \equiv p_i + p_e$.

The ideal gas constitutive laws for electrons and ions are $p_i = n_i k T_i$ and $p_e = n_e k T_e$, where k is the Boltzmann constant. The definition of total pressure provides a total constitutive relation $p = nkT$ which defines a total temperature $T = T_i + ZT_e$ and a specific internal energy $\varepsilon \equiv p/(nm(\gamma - 1))$. By replacing the pressure p with the total energy $nm(|u|^2/2 + \varepsilon)$, Eqs. (11)–(13) can be placed in divergence form,

$$\partial_t n + \nabla \cdot (nu) = 0, \tag{15}$$

$$\partial_t (mnu) + \nabla \cdot (mnu \otimes u) + \nabla p = F_p, \tag{16}$$

$$\partial_t \left[nm \left(\frac{1}{2}|u|^2 + \varepsilon \right) \right] + \nabla \cdot \left[nmu \left(\frac{1}{2}|u|^2 + \varepsilon \right) + pu \right] = u \cdot F_p, \tag{17}$$

and

$$(\gamma - 1)^{-1}[\partial_t p_e + \nabla \cdot (p_e u)] + p_e \nabla \cdot u = 0. \tag{18}$$

Except for the ponderomotive source term F_p, which is defined in (8), the equations (15)–(17) are the standard Euler equations of gas dynamics (in terms of the number density n instead of the usual mass density $\rho = mn$).

The full system (15)–(18) expresses the conservation of total mass, momentum, and energy. In this formulation, the electron pressure equation (18) is only required in order to determine the partitioning of energy between the ions and electrons. If this partitioning is for some reason not needed, then (15)–(17) is sufficient to describe the total plasma motion, which is identical to an ordinary neutral fluid under the influence of a body force.

The electric field $\mathcal{E}$ in (8) is obtained as the solution of Maxwell's equations in a plasma (e.g., Section 7.6 of [18]). Assuming that $\nabla \cdot \mathcal{E}$ is small enough to be neglected [4], $\mathcal{E}$ is determined by solving the wave equation

$$-\left(\frac{\partial^2}{\partial t^2} - c^2 \nabla^2 \right) \mathcal{E} = \omega_p^2 \mathcal{E}, \tag{19}$$

where c is the speed of light in a vacuum. The light field $\mathcal{E}$ therefore influences the plasma evolution through the ponderomotive force (8) while the plasma couples to the light propagation through the presence of the electron density in the plasma frequency (9).

We assume that the light field $\mathcal{E}$ satisfying (19) has a fixed polarization $\mathcal{P}$ and is time-harmonic at the light frequency ω; i.e.,

$$\mathcal{E} = \mathcal{P} \operatorname{Re} E e^{-i\omega t}, \tag{20}$$

where E is a complex-valued envelope. Substitution of (20) into (19) yields the Helmholtz equation

$$(\nabla^2 + k^2\eta^2)\,E = 0, \tag{21}$$

where $k \equiv \omega/c$ is the free space wave number, and

$$\eta \equiv \left(1 - \frac{\omega_p^2}{\omega^2}\right)^{1/2} \tag{22}$$

is the refractive index. In this paper, η is real-valued, since we consider only underdense plasmas, i.e., plasmas for which $n_e < n_c$ where $n_c \equiv \omega^2\epsilon_0 m_e/e^2$ is the critical density at which the plasma frequency equals the light frequency.

The Helmholtz equation (21) requires boundary conditions to obtain a well-posed problem. In the context of acoustic wave scattering by a dielectric medium, a standard approach is to consider the amplitude E to be the sum of a prescribed unperturbed incident wave and a scattered wave, with a Sommerfeld radiation boundary condition enforced on the scattered wave [9]. To obtain a numerical discretization, it is necessary to replace the Sommerfeld condition by a condition on the boundary of the finite computational domain. The computational domain boundary must be far enough away from the laser–plasma region of interest to avoid numerical reflections from the approximated far-field boundary conditions. Unfortunately, in the context of ignition-scale inertial confinement fusion problems, the laser–plasma interaction “region of interest” is already large, spanning thousands of wavelengths. The solution of such large systems is not feasible, especially in the context of a time-dependent simulation where the Helmholtz equation must be solved repeatedly as the plasma evolves.

These difficulties motivate the use of paraxial approximations in the solution of large laser–plasma interaction problems. Paraxial approximations assume that the laser light propagates primarily in one direction and does not scatter too far from the propagation axis. These assumptions lead to discrete algorithms that involve a sweep of the computational domain in the propagation direction, starting with incident beam boundary conditions. If periodic boundary conditions are employed in the coordinate directions transverse to the light propagation axis, the need to employ a larger computational domain to enforce far-field boundary conditions is avoided (although larger domains might still be needed to avoid the periodic aliasing of solutions). Another major advantage of paraxial approximations is the opportunity to envelope solutions by factoring out the most highly oscillatory components that would otherwise have to be resolved by the numerical discretization. The obvious downside to paraxial approximations is their limitation to physical problems for which the assumptions are valid, e.g., problems with negligible backscatter.

There are several ways to derive paraxial approximations. Here, we follow the development of [11]. Assume that the light is propagating in the positive $x \equiv x_1$ direction. We formally write (21) as

$$(P^2 + Q^2)\,E = 0, \tag{23}$$

where

$$P \equiv \partial_x, \tag{24}$$

$$Q \equiv (\nabla_\perp^2 + k^2\eta^2)^{1/2}, \tag{25}$$

and $\nabla_\perp^2$ denotes the Laplacian with respect to the coordinates $x_\perp$ transverse to x ($x_\perp \equiv x_2$ in two dimensions and $x_\perp \equiv (x_2, x_3)$ in three dimensions). Assuming that P and Q commute, (23) factors as

$$(P + iQ)(P - iQ)\,E = 0. \tag{26}$$

We now drop the first factor on the left-hand side, eliminating backward propagating waves. Note that the commutativity assumption on P and Q implies that the refractive index η has small x variation, which is consistent with the assumption of negligible backscatter.

An algebraic manipulation verifies that

$$Q = (Q + k\eta)^{-1}\nabla_\perp^2 + k\eta + k(Q + k\eta)^{-1}(\eta Q - Q\eta). \tag{27}$$

If we now assume that η also varies sufficiently slowly in the transverse directions so that it approximately commutes with Q, we may then drop the last term in (27). As suggested in [11], we replace η in the first term of (27) by a reference value η_0 that is solely a function of x (e.g., the transverse average of η). The resulting expression can then be solved for the approximate Q as

$$Q \approx \left(\nabla_\perp^2 + k^2\eta_0^2\right)^{1/2} + k(\eta - \eta_0). \tag{28}$$

Although this expression for Q is the basis for the paraxial algorithm described in [11], the more traditional paraxial approximation is obtained by further assuming that $\nabla_\perp^2$ is dominated by $k^2\eta_0^2$, so that the square root term in (28) can be linearized to obtain

$$Q \approx \frac{1}{2k\eta_0}\nabla_\perp^2 + k\eta. \tag{29}$$

The paraxial solution is therefore the result of integrating

$$\partial_x E = \frac{i}{2k\eta_0}\nabla_\perp^2 E + ik\eta E. \tag{30}$$

To reduce the magnitude of the phase variation resulting from the last term of (30), we take the opportunity to introduce an integrating factor. Specifically, let

$$E \equiv \tilde{E}\exp\left(ik\int_{x_0}^{x}\eta_0(x')\,dx'\right). \tag{31}$$

Substitution in (30) yields an equation for the envelope $\tilde{E}$:

$$\partial_x \tilde{E} = \frac{i}{2k\eta_0}\nabla_\perp^2 \tilde{E} + ik(\eta - \eta_0)\tilde{E}. \tag{32}$$

The ponderomotive force (8) needed for the integration of (15)–(17) can be computed directly from the solution of (32), which has the same modulus as the light field $\mathcal{E}$.

3. UNIFORM GRID ALGORITHM

We employ the method of fractional steps to numerically integrate the system (8), (9), (15)–(18), (19) on a uniform computational grid. Assume that the solution is known at a time t_0, and we wish to compute it at time $t_0 + \Delta t$. The numerical algorithm that performs this update consists of four main steps:

1. Compute the ponderomotive term (8) using values for the plasma and light field at time t_0.
2. Integrate Eqs. (15)–(17) to time $t_0 + \Delta t$.
3. Use the update density as well as some flux quantities computed in Step 2 to integrate (18) to time $t_0 + \Delta t$.
4. Use the updated density to define a new plasma frequency (9), and then solve (19) for the new light field at time $t_0 + \Delta t$.

In the remainder of this section, we describe the above steps in more detail.

3.1. Plasma Model Discretization

We use an explicit, high-resolution Godunov scheme to discretize (15)–(17). The application of such algorithms to the solution of systems of hyperbolic conservation laws is well known, especially in the case of the Euler equations of gas dynamics. The main issue in the current context is the inclusion of source terms, which also affects the approach selected to account for multidimensional effects. We therefore describe these aspects in more detail for the two-dimensional case. The algorithm discussed in this section can be generalized to three spatial dimensions in a straightforward manner. The only essential difference is the formulation of the characteristic tracing and the transverse correction steps, which are extended using the approach of either [6] or [23].

Letting $U \equiv (mn, mnu, mn(\frac{1}{2}|u|^2 + \epsilon))^T$, the system (15)–(17) can be written in vector form,

$$\partial_t U + \nabla \cdot F(U) = G(U, E). \tag{33}$$

To simplify notation, assume two spatial dimensions. Let $U^n_{i,j}$ represent the elementwise average of U in the cell $[i\Delta x_1, (i+1)\Delta x_1] \times [j\Delta x_2, (j+1)\Delta x_2]$ at time $t_n = n\Delta t$. When the source G lagged in time and its cell-averaged value is denoted by $G^n_{i,j}$, the $U^n_{i,j}$ are updated by the conservative difference formula

$$\begin{aligned} U^{n+1}_{i,j} = U^n_{i,j} &- \frac{\Delta t}{\Delta x_1}\left[F_1\left(U^{n+1/2}_{i+1/2,j}\right) - F_1\left(U^{n+1/2}_{i-1/2,j}\right)\right] \\ &- \frac{\Delta t}{\Delta x_2}\left[F_2\left(U^{n+1/2}_{i,j+1/2}\right) - F_2\left(U^{n+1/2}_{i,j-1/2}\right)\right] + \Delta t G^n_{i,j}, \end{aligned} \tag{34}$$

where the values $U^{n+1/2}_{i\pm1/2,j}$ and $U^{n+1/2}_{i,j\pm1/2}$ denote "flux" states at the cell faces at an intermediate time $t = t_n + \Delta t/2$. These values are obtained from solutions of Riemann problems with initial ("left" and "right") states at the respective faces. See Fig. 1.

Our approach to calculation of the initial states is based on a donor cell upwind scheme [6, 24, 26]. The left and right states used to calculate the flux states $U^{n+1/2}_{i+1/2,j}$, $U^{n+1/2}_{i,j+1/2}$ are obtained via characteristic tracing with corrections that account for the source term and for

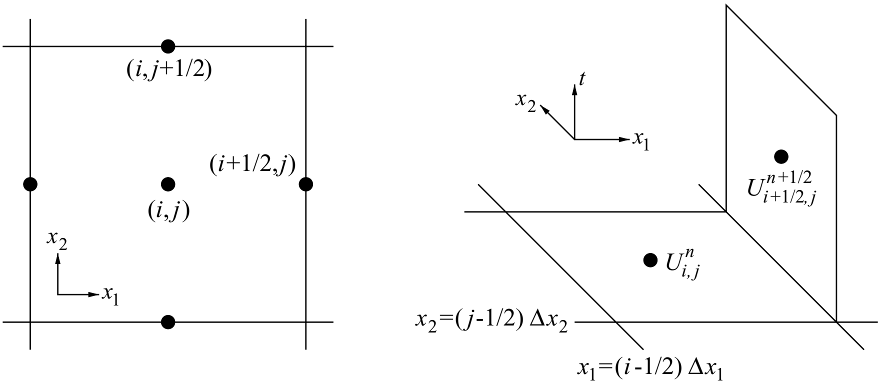


FIG. 1. States associated with the grid.

the flows transverse to the flux direction. For reasons explained below, these calculations are performed on the conserved variables U but using the so-called “flux” or “primitive” variables $V = (mn, u, p)$; in terms of the latter, (33) is

$$\partial_t V + W^{-1}\nabla \cdot \tilde{F}(V) = \tilde{G}(V, E), \tag{35}$$

where $dU = W dV$, $\tilde{F}(V) = F(U)$, and $\tilde{G}(V, E) = W^{-1}G(U, E)$.

We describe the calculation of the initial states on cell faces orthogonal to the x_1 axis; an analogous calculation is valid for the other axis. The calculation of the left state $V^{n+1/2,L}_{i+1/2,j}$ and the right state $V^{n+1/2,R}_{i+1/2,j}$ is based on a Taylor expansion of V, where Eq. (35) is used to eliminate time derivatives in favor of spatial derivatives:

$$\begin{aligned} V\left(x_i \pm \frac{\Delta x_1}{2}, x_j, t^n + \frac{\Delta t}{2}\right) &\approx V^n_{i,j} \pm \frac{\Delta x_1}{2}\left(\partial_{x_1} V\right)^n_{i,j} + \frac{\Delta t}{2}(\partial_t V)^n_{i,j} \\ &= V^n_{i,j} \pm \frac{\Delta x_1}{2}\left(\partial_{x_1} V\right)^n_{i,j} - \frac{\Delta t}{2}(W^{-1}\nabla \cdot \tilde{F})^n_{i,j} + \frac{\Delta t}{2}\tilde{G}^n_{i,j}. \end{aligned} \tag{36}$$

This calculation is performed in two steps in a “predictor–corrector” fashion. First, we predict values $\hat{V}^{n+1/2,L}_{i+1/2,j}$ and $\hat{V}^{n+1/2,R}_{i+1/2,j}$ in a purely one-dimensional fashion by assuming no transverse variation; dropping from (36) the transverse derivatives (i.e., $\partial_{x_2}\tilde{F}_2$), one has

$$\hat{V}^{n+1/2,L}_{i+1/2,j} \equiv V^n_{i,j} + \frac{\Delta x_1}{2}\left(\partial_{x_1} V\right)^n_{i,j} - \frac{\Delta t}{2}\left(W^{-1}\partial_{x_1}\tilde{F}_1\right)^n_{i,j} + \frac{\Delta t}{2}\tilde{G}^n_{i,j}, \tag{37}$$

$$\hat{V}^{n+1/2,R}_{i+1/2,j} \equiv V^n_{i+1,j} - \frac{\Delta x_1}{2}\left(\partial_{x_1} V\right)^n_{i+1,j} - \frac{\Delta t}{2}\left(W^{-1}\partial_{x_1}\tilde{F}_1\right)^n_{i+1,j} + \frac{\Delta t}{2}\tilde{G}^n_{i+1,j}. \tag{38}$$

In this form, derivatives of the flux vector $\tilde{F}_1$ and the primitive state vector V must both be approximated. By assuming continuous solutions, we can further simplify this predictor step by allowing the flux gradient to be written in quasilinear form. All spatial derivatives are then expressed in terms of the state vector V,

$$\hat{V}^{n+1/2,L}_{i+1/2,j} \equiv V^n_{i,j} + \frac{1}{2}\left(I - (\tilde{A}_1)^n_{i,j}\frac{\Delta t}{\Delta x_1}\right)\Delta x_1\left(\partial_{x_1}V\right)_{i,j} + \frac{\Delta t}{2}\tilde{G}^n_{i,j}, \tag{39}$$

$$\hat{V}^{n+1/2,R}_{i+1/2,j} \equiv V^n_{i+1,j} - \frac{1}{2}\left(I + (\tilde{A}_1)^n_{i+1,j}\frac{\Delta t}{\Delta x_1}\right)\Delta x_1\left(\partial_{x_1}V\right)_{i+1,j} + \frac{\Delta t}{2}\tilde{G}^n_{i+1,j}, \tag{40}$$

where $\tilde{A}_1 = W^{-1}(d\tilde{F}_1/dV)$. In a second step (Section 3.1.2), multidimensional effects are included when the left and right states are corrected by reintroducing the approximations of transverse derivatives.

3.1.1. *One-Dimensional Predictor, Characteristic Tracing*

Calculation of the predicted states (39), (40) is performed using characteristic tracing. We approximate the value of the derivative $(\partial_{x_1}V)_{i,j}$ using a local-slope-limiting algorithm that provides second-order accuracy away from extrema and steep gradients while preventing spurious numerical oscillations near rapid spatial transitions [27]. Write

$$\Delta x_1\left(\partial_{x_1}V\right)^n_{i,j} = \Delta V^n_{i,j} \approx \text{dminmod}\left(V^n_{i+1,j} - V^n_{i,j}, V^n_{i,j} - V^n_{i-1,j}\right), \tag{41}$$

where the double minmod function is defined by

$$\text{dminmod}(a, b) \equiv \frac{\text{sgn}(a) + \text{sgn}(b)}{2}\min\left(2|a|, 2|b|, \frac{|a+b|}{2}\right) \tag{42}$$

and the operations are performed componentwise. Slope limiting is done on primitive variables so as to restrict directly nonphysical undershoots in the physically positive quantities mn and p.

Next, we spectrally decompose the middle term in (39), (40). Let X_1 denote a matrix whose columns R_k consist of right eigenvectors of $\tilde{A}_1$ (i.e., $X_1\tilde{A}_1 = \text{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)X_1 = \Lambda X_1$). We then have $\Delta V = X_1(\alpha_k) = \sum \alpha_k R_k$, where $(\alpha_k) = X_1^{-1}\Delta V$. Hence,

$$\begin{aligned}\left(I - \tilde{A}_1\frac{\Delta t}{\Delta x_1}\right)\Delta V &= \left(I - \tilde{A}_1\frac{\Delta t}{\Delta x_1}\right)\sum_k \alpha_k R_k,\\ &= \sum_k\left(I - \lambda_k\frac{\Delta t}{\Delta x_1}\right)\alpha_k R_k.\end{aligned} \tag{43}$$

In the computation of (39), (40), we include only those terms in the sum (43) corresponding to waves that propagate toward the corresponding cell faces: positive eigenvalues λ_k for $\hat{V}^{n+1/2,L}_{i+1/2,j}$ and negative values for $\hat{V}^{n+1/2,R}_{i-1/2,j}$. We write

$$\hat{V}^{n+1/2,L}_{i+1/2,j} \equiv V^n_{i,j} + \frac{1}{2}\left[\sum_{\lambda>0}\left(I - \lambda_k\frac{\Delta t}{\Delta x_1}\right)\alpha_k R_k\right]^n_{i,j} + \frac{\Delta t}{2}\tilde{G}^n_{i,j}, \tag{44}$$

$$\hat{V}^{n+1/2,R}_{i+1/2,j} \equiv V^n_{i+1,j} - \frac{1}{2}\left[\sum_{\lambda<0}\left(I + \lambda_k\frac{\Delta t}{\Delta x_1}\right)\alpha_k R_k\right]^n_{i+1,j} + \frac{\Delta t}{2}\tilde{G}^n_{i+1,j}. \tag{45}$$

3.1.2. *Transverse Correction*

We now use these predicted left and right states to compute a predicted value of V at the cell face from the solution to a Riemann initial value problem, $\hat{V}^{n+1/2}_{i+1/2,j} = \mathrm{RP}(\hat{V}^{n+1/2,L}_{i+1/2,j}, \hat{V}^{n+1/2,R}_{i+1/2,j})$. To restore the transverse flux terms that were omitted in Eqs. (37), (38), first we approximate the derivative term in $W^{-1}\partial_{x_2}\tilde{F}_2(V)$ by a centered difference evaluated with the predicted interface values $\hat{V}^{n+1/2}_{i,j\pm1/2}$. This gives corrected left and right interface states $V^{n+1/2,L}_{i+1/2,j}$ and $V^{n+1/2,R}_{i+1/2,j}$ as follows:

$$V^{n+1/2,L}_{i+1/2,j} \equiv \hat{V}^{n+1/2,L}_{i+1/2,j} - \frac{\Delta t}{2}(W^{-1})^n_{i,j}\frac{\tilde{F}_2\big(\hat{V}^{n+1/2}_{i,j+1/2}\big) - \tilde{F}_2\big(\hat{V}^{n+1/2}_{i,j-1/2}\big)}{\Delta x_2}, \tag{46}$$

$$V^{n+1/2,R}_{i+1/2,j} \equiv \hat{V}^{n+1/2,R}_{i+1/2,j} - \frac{\Delta t}{2}(W^{-1})^n_{i+1,j}\frac{\tilde{F}_2\big(\hat{V}^{n+1/2}_{i+1,j+1/2}\big) - \tilde{F}_2\big(\hat{V}^{n+1/2}_{i+1,j-1/2}\big)}{\Delta x_2}. \tag{47}$$

The final states at the cell faces, which are used to compute the time-average interface fluxes, are obtained by again solving Riemann problems but now with the corrected left and right states as initial data: $V^{n+1/2}_{i+1/2,j} = \mathrm{RP}(V^{n+1/2,L}_{i+1/2,j}, V^{n+1/2,R}_{i+1/2,j})$.

3.1.3. *Electron Pressure*

The last step in the integration of the plasma equations is to update the electron pressure p_e, for which we use a simple upwind scheme. Replacing p_e by $ZnkT_e$ in (18) and dividing by kZ, we have an equivalent equation for the temperature T_e,

$$\partial_t(nT_e) + \gamma\nabla\cdot(nT_e u) - (\gamma-1)u\cdot\nabla(nT_e) = 0, \tag{48}$$

which we discretize as

$$\begin{aligned}(T_e)^{n+1}_{i,j} = \frac{1}{n^{n+1}_{i,j}}\Bigg\{ & n^n_{i,j}(T_e)^n_{i,j} - \Delta t\Bigg[\gamma\Bigg(\frac{(T_e)^{n+1/2}_{i+1/2,j}(nu_1)^{n+1/2}_{i+1/2,j} - (T_e)^{n+1/2}_{i-1/2,j}(nu_1)^{n+1/2}_{i-1/2,j}}{\Delta x_1} \\ & + \frac{(T_e)^{n+1/2}_{i,j+1/2}(nu_2)^{n+1/2}_{i,j+1/2} - (T_e)^{n+1/2}_{i,j-1/2}(nu_2)^{n+1/2}_{i,j-1/2}}{\Delta x_2}\Bigg) \\ & + (\gamma-1)\Bigg((u_1)^n_{i,j}\frac{(T_e)^{n+1/2}_{i+1/2,j}n^{n+1/2}_{i+1/2,j} - (T_e)^{n+1/2}_{i-1/2,j}n^{n+1/2}_{i-1/2,j}}{\Delta x_1} \\ & + (u_2)^n_{i,j}\frac{(T_e)^{n+1/2}_{i,j+1/2}n^{n+1/2}_{i,j+1/2} - (T_e)^{n+1/2}_{i,j-1/2}n^{n+1/2}_{i,j-1/2}}{\Delta x_2}\Bigg)\Bigg]\Bigg\}.\end{aligned} \tag{49}$$

The terms $(nu_1)^{n+1/2}_{i-1/2,j}$ and $(nu_2)^{n+1/2}_{i,j-1/2}$ are flux terms in (15) and therefore have been previously calculated. Similarly, during the calculation of these flux terms, we also keep $n^{n+1/2}_{i-1/2,j}$ and $n^{n+1/2}_{i,j-1/2}$. The only values still unknown in (49) are the values of T_e at the cell faces, e.g., $(T_e)^{n+1/2}_{i-1/2,j}$. These are computed from cell-centered values $(T_e)_{i,j}$ in an upwind fashion.

3.2. Paraxial Light Model Discretization

To discretize (32), consider a uniform grid with cells of size $\Delta x \times \Delta x_\perp$ indexed by (j, m), $1 \le j \le N$, $1 \le m \le N_\perp$. Let $E_{j-1/2,m}$ and $E_{j+1/2,m}$ denote the value of $\tilde{E}$ at the center of the left and right x-faces of cell (j, m), respectively, and let $\eta_{j,m}$ denote the value of the refractive index (22) at the cell center. Define the transverse η average

$$\eta_j^0 \equiv \frac{1}{N_\perp} \sum_{m=1}^{N_\perp} \eta_{j,m}. \tag{50}$$

We discretize (32) as

$$\begin{aligned}\frac{E_{j+1/2,m} - E_{j-1/2,m}}{\Delta x} = {} & \frac{i}{4k\eta_j^0 \Delta x_\perp^2} \big[\big(E_{j-1/2,m-1} + E_{j+1/2,m-1}\big) - 2\big(E_{j-1/2,m} + E_{j+1/2,m}\big) \\ & + \big(E_{j-1/2,m+1} + E_{j+1/2,m+1}\big)\big] + \frac{ik\big(\eta_{j,m} - \eta_j^0\big)}{2} \\ & \times \big(E_{j-1/2,m} + E_{j+1/2,m}\big).\end{aligned} \tag{51}$$

Here, we have applied the standard central differencing of the transverse Laplacian $\nabla_\perp^2$. This implicit discretization in the x direction is similar to a Crank–Nicholson scheme, except for the centering of the coefficients at the step midpoint. The coefficient centering in (51) ensures that energy is algebraically conserved by the algorithm, as can be verified by a spectral analysis (see, e.g., the discussion surrounding (85) in Section 4). The light propagation algorithm can be described in a more compact notation as follows: Given an incident complex amplitude $E_{1/2,m}$, $1 \le m \le N_\perp$, compute for $j = 1, \ldots, N$

$$\begin{pmatrix} E_{j+1/2,1} \\ \vdots \\ E_{j+1/2,N_\perp} \end{pmatrix} = A_j^{-1} A_j^* \begin{pmatrix} E_{j-1/2,1} \\ \vdots \\ E_{j-1/2,N_\perp} \end{pmatrix}, \tag{52}$$

where A_j is the $N_\perp \times N_\perp$ matrix defined by

$$A_j \equiv D_j - \beta_j \nabla_{N_\perp}^2, \tag{53}$$

where

$$\beta_j \equiv \frac{i}{4k\eta_j^0}, \tag{54}$$

D_j is the diagonal matrix with entries defined by

$$(D_j)_{m,m'} \equiv \begin{cases} \frac{1}{\Delta x} - \frac{ik}{2}\big(\eta_{j,m} - \eta_j^0\big), & \text{if } m = m', \\ 0, & \text{otherwise,} \end{cases} \tag{55}$$

and $\nabla_{N_\perp}^2$ denotes the matrix corresponding to the central differencing of $\nabla_\perp^2$. The * superscripts denote complex conjugation. To obtain the amplitudes by "sweeping" the computational grid starting from known boundary values, we therefore solve a linear system with matrix A_j at every Δx step.

4. LOCALLY REFINED MESH ALGORITHM

The algorithm described in the previous section can be generalized to accommodate local mesh refinement. To do so, we adopt a block-structured refinement approach in the style of [2, 3] in which the computational domain is viewed as a nested hierarchy of refinement levels, indexed from coarsest to finest by $\ell = 0, 1, \ldots, \ell_{max}$. Each refinement level is a disjoint union of rectangular grids. On level 0, the coarsest level, this union comprises a uniform meshing of the entire physical domain. The rectangular grids on a given level $\ell > 0$ are obtained by refining rectangular patches of grid cells on the $\ell - 1$ refinement level by the same ratio r_ℓ. The ratio r_ℓ can be generalized to a vector whose components are the refinement ratios in each coordinate dimension. Note that this prescription implies that no cell is partially refined. It is also required that each refinement level by properly contained in the next coarser level, except perhaps at the domain boundary.

A time integration algorithm on a locally refined grid hierarchy is made adaptive by allowing the location and number of refinement levels to change as the computed solution evolves. Using the general approach of [2], individual mesh cells are “tagged” for refinement based on some criterion; then new rectangular grids are generated that contain the tagged cells. The selection of the refinement criteria is an important issue and is often problem-dependent [13]. The main focus of the work in this paper is the AMR integration algorithm, not the selection of the best refinement criteria. Some comments on refinement criteria are made in Section 4.2.

The integration of the plasma and the light equations on a locally refined grid hierarchy such as that described above is accomplished by the integration of the coupled system on individual refinement levels combined with synchronization steps to ensure consistency across levels. The algorithm can be applied recursively on hierarchies with an arbitrary number of refinement levels. We therefore outline the algorithm for the case of two levels (Fig. 2a), where the refinement ratio between the fine and coarse levels is r. The main steps are as follows:

1. Choose a time step Δt_c satisfying a Courant-Friedrichs-Lewy (CFL) stability criterion based on the coarse-level mesh size. Using source terms evaluated at the current time, integrate the plasma model over Δt_c. Using the new plasma density thereby obtained, sweep the light algorithm over the coarse level starting from either physical boundary conditions or amplitudes interpolated from coarser levels (Fig. 2b).

2. Integrate the plasma and light equations on the fine grid up to the same ending time as the coarse-level integration performed in the preceding step (Fig. 2d). Since the CFL stability condition on the fine grid requires a time step smaller than Δt_c, several fine steps are usually necessary. These time steps can be taken to be $\Delta t_c / r$ initially and then modified as needed by the evolving fine-grid solution. Notice that this implies that more than r steps may be needed on the fine grid. Boundary data required during the fine-grid integration steps are obtained using temporal and spatial interpolation of the coarse-level data (Fig. 2c).

3. After the integrations on the coarse and fine levels, synchronize the solution at the new common time across the two levels. For the plasma variables, this step involves averaging and “refluxing” operations to restore conservation (Fig. 2e). For the light amplitude, both refinement levels are reswept in a coupled fashion (Fig. 2f).

4. Using the synchronized solution, apply the refinement criteria and regrid. If necessary, the fine level is replaced by a different one, or perhaps eliminated. Data on the new fine level are initialized using data from the coarse level or the previous fine level. This regridding

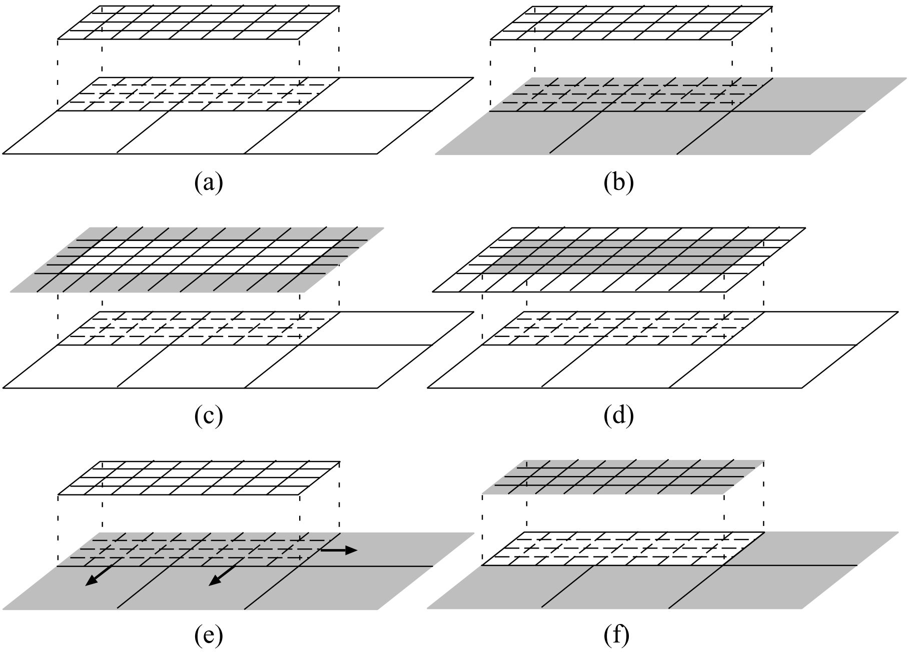


FIG. 2. Steps in the multilevel algorithm.

operation need only be performed periodically, say, after a specified number of coarse-grid time steps.

The specific details of the integration and synchronization of the plasma model across multiple refinement levels closely follow the previously well-documented approach for hyperbolic conservation laws [2, 13, 25]. The primary consideration is the correction of coarse-grid fluxes to maintain conservation in cells bordering the fine grid. The extension of the light algorithm (52) is more problematic, however, due to the need to solve linear systems in the coordinate directions transverse to the propagation direction. We therefore consider this algorithm component separately in more detail.

4.1. Extension of the Light Algorithm to Locally Refined Grids

In the multilevel integration algorithm described above, two types of light equation integrations are performed: (*i*) single-level integrations in which the light sweep (52) is performed on one refinement level using initial and boundary conditions interpolated from coarser levels, and (*ii*) multilevel integrations in which the light is swept across multiple refinement levels to synchronize the levels. In the following description of these operations, we again assume that the light is propagating in the positive x direction.

4.1.1. *Integration of a Single Refinement Level*

Figure 3 illustrates the key steps in the application of the light propagation algorithm on a single refinement level. For simplicity, we assume that the fine level consists of a single rectangular grid obtained by refining a portion of the underlying coarse grid by a factor of 2 in each coordinate direction. The procedure generalizes in a straightforward manner to fine levels obtained by refining a disjoint union of rectangular coarse subgrids by an arbitrary

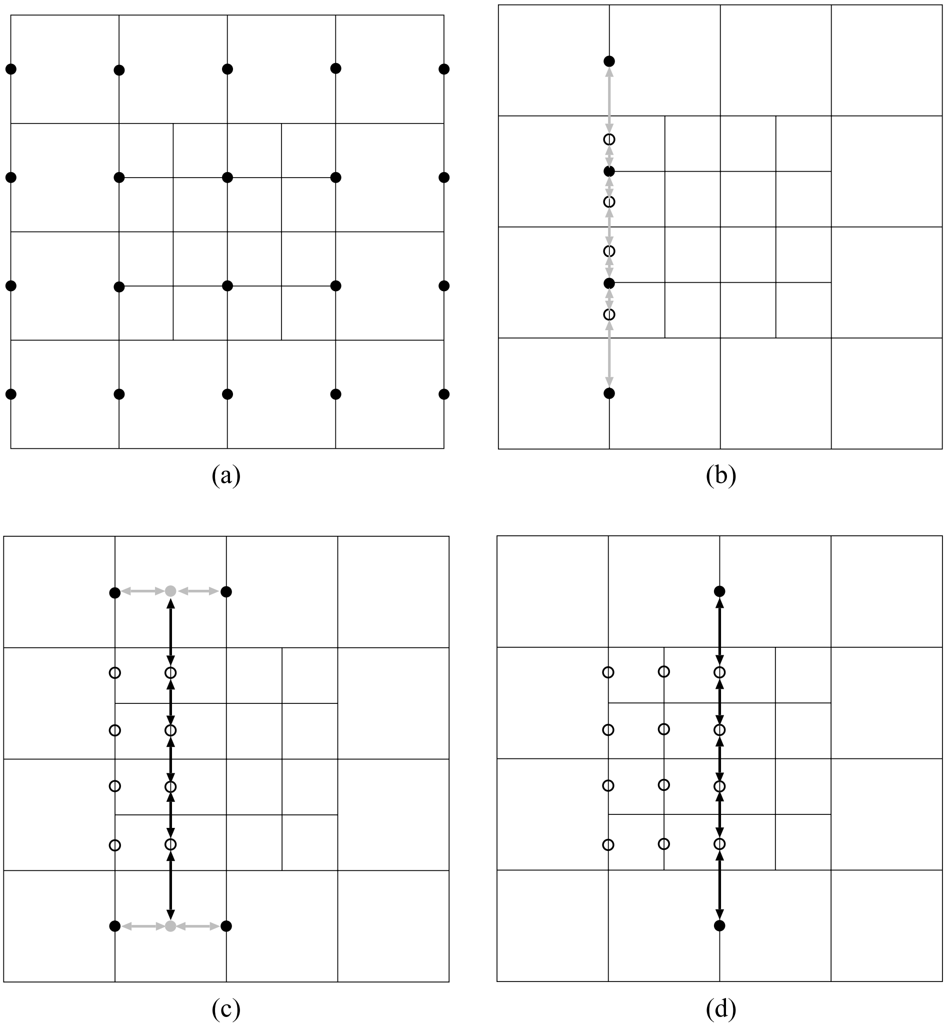


FIG. 3. An illustration of the key steps in the light propagation algorithm on a single level: (a) initial data at coarse x-face centers (black circles); (b) interpolation (gray arrows) of coarse data to fine x-face centers (unfilled circles); (c) interpolation (gray arrows) of coarse data to obtain Dirichlet boundary values (gray circles) for fine-grid linear system solve (black arrows) for next column of fine x-face data (unfilled circles); (d) solution of linear system (black arrows) for next column of fine x-face data (unfilled circles) using coarse-grid data (black circles) as Dirichlet boundary data.

refinement ratio in each coordinate direction. Suppose that amplitudes have been previously computed at the x-face centers of the coarse grid, indicated by the black circles in Fig. 3a, and we now want to obtain a more highly resolved solution on the x-faces of the fine grid. The propagation sweep begins by interpolating coarse data in the transverse direction to obtain amplitudes at the leftmost x-face centers on the fine grid, as shown in Fig. 3b. The interpolated data (denoted by the unfilled circles) are then used as initial data to sweep the remainder of the fine grid by applying (52). The Dirichlet boundary values needed for the inversion of the operators A_j defined by (53) are interpolated from coarse-grid data in the x direction as indicated in Fig. 3c. If the fine x-faces coincide with coarse x-faces, then the coarse data provide boundary values directly without interpolation, as shown in Fig. 3d.

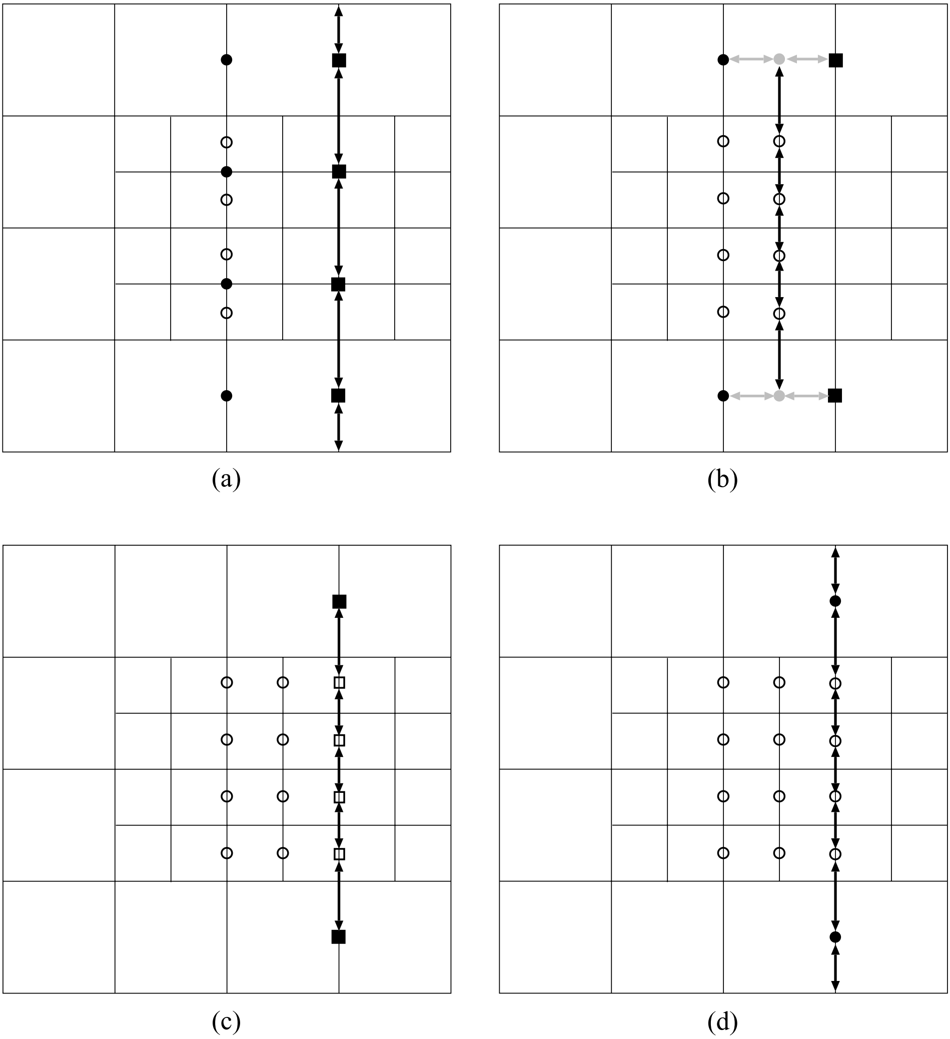


FIG. 4. An illustration of the key steps in the multilevel light propagation algorithm: (a) existing data at coarse x-face centers (filled circles) and fine x-face centers (unfilled circles); data at next plane of coarse x-face centers (filled squares); (b) next plane of data at fine x-face centers (unfilled circles) obtained using interpolated coarse data (gray circles); (c) next plane of data at fine x-face centers (unfilled squares) obtained using coarse data (filled squares); (d) coarse data (filled circles) and fine data (unfilled circles) after composite synchronization.

4.1.2. *Integration of Multiple Refinement Levels*

The integration of the discrete light equations on multiple refinement levels is accomplished using a multilevel sweep. This sweep proceeds in a recursive manner that involves single steps of the algorithm described above on each refinement level, combined with synchronization steps to enforce the multilevel coupling required by the presence of the transverse Laplacian $\nabla^2_{N_\perp}$ in the A_j operator of (53). Since this operator is inverted on a plane of x-faces in the sweep (52), it is natural to perform this synchronization on x-face planes that are aligned on multiple levels.

To describe the algorithm more concretely, consider a two-level, locally refined grid such as that shown in Fig. 4. We again assume that the light propagates from left to right,

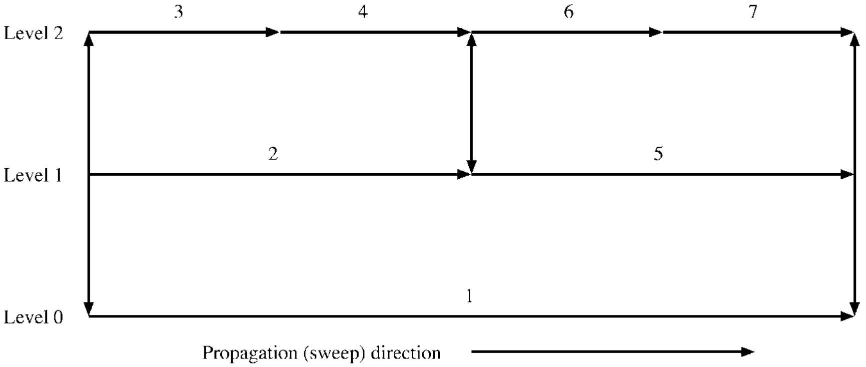


FIG. 5. Recursive light propagation (sweep) schedule. The numerals indicate the order of integration of refinement levels. The vertical arrows denote composite synchronizations.

starting with prescribed incident boundary values on the left boundary. A generic step in the global sweep proceeds as follows. Suppose that the sweep has so far produced amplitudes at the coarse-grid-cell x-faces (filled circles) and the fine-grid-cell x-faces (unfilled circles) in Fig. 4a. Applying a single step of (52) on the coarse grid yields data at the next plane of x-face edges (filled squares) in Fig. 4a. As in the single-level sweeping algorithm described above, the coarse data are interpolated to provide boundary values for the subsequent integration of the fine-grid data up to the same plane as the new coarse-grid data, as indicated in Figs. 4b and 4c. The coarse data (filled squares) and fine data (unfilled squares) are then updated by a composite synchronization step to obtain new values (filled and unfilled circles), as depicted in Fig. 4d. The purpose of the composite synchronization step and the algorithm by which it is performed are described in detail in the next section. Following the synchronization step, the updated fine data are interpolated to obtain data at the coarse x-faces covered by the fine grid (this step is not shown in Fig. 4). This procedure repeats until the entire grid has been swept. The algorithm generalizes to any number of levels via recursion, as illustrated in Fig. 5. In this schedule diagram, the numerals indicate the order of integration of refinement levels. The vertical arrows indicate the composite synchronizations, which occur whenever a refinement level has been integrated to the same point as the next coarser level.

4.1.3. *Composite Synchronization*

The composite synchronization step in the algorithm just described enforces compatibility conditions at coarse–fine boundaries demanded by the presence of the transverse Laplacian $\nabla^2_{N_\perp}$. The use of interpolated coarse-grid data as boundary values for the fine-grid integration in the multilevel sweeping algorithm establishes the continuity of the amplitude across coarse–fine boundaries perpendicular to the transverse direction, but it does not yield continuous normal derivatives there. Consequently, after the fine grid has been integrated to the same plane as the coarse grid, the transverse derivatives of the coarse and fine grid solutions will in general disagree at transverse coarse–fine boundaries. A composite synchronization step is therefore necessary to restore the continuity of the transverse derivatives at coarse–fine boundaries while maintaining the continuity of the amplitudes. This is accomplished by adding a correction obtained by solving a linear system involving both the coarse and fine grids, where the right-hand side represents a residual error due to the discontinuity in the transverse derivatives.

At each composite synchronization point of the light sweeping algorithm, we seek the solution of an equation of the form

$$\left(D - \beta \nabla_{N_\perp}^2\right) E = R \equiv \left(D - \beta \nabla_{N_\perp}^2\right)^* E_{prev} \tag{56}$$

on a locally refined transverse grid hierarchy, where E_{prev} is the vector of amplitudes computed at the previous face plane. The diagonal matrix D is the composition of the corresponding matrices D_j on each level, defined by (55), and the scalar β is the common value of (54) at the current face plane. The solution of (56) corresponds with the application of the operator A_j^{-1} in (52). Since the amplitudes E are centered at x-faces of the original grid, (56) is a cell-centered discretization in the transverse coordinates. The equation (56) can also be expressed in the divergence form

$$\begin{aligned} DE - \beta \nabla \cdot \sigma &= R, \\ \sigma &\equiv \nabla E, \end{aligned} \tag{57}$$

interpreted in a cell-volume-averaged sense; i.e., the divergence term represents the difference of the face-centered fluxes σ, and the gradient operator in the definition of σ denotes the usual first-order differencing of the cell-centered amplitudes E.

The transverse grid hierarchy upon which (56) must be solved is a cross section of the current grid hierarchy. For simplicity, let us assume that this transverse cross section consists of just two refinement levels: a coarse grid Ω_c and a fine grid Ω_f. Let $P(\Omega_f)$ denote the projection of Ω_f into Ω_c, and let Γ denote the interface between $P(\Omega_f)$ and $\Omega_c \backslash P(\Omega_f)$. The solution of (57) on the composite grid $\Omega_f \cup \Omega_c \backslash P(\Omega_f)$ is equivalent to the system

$$D_f E_f - \beta \nabla \cdot \sigma_f = R_f \quad \text{on } \Omega_f, \tag{58}$$

$$\sigma_f \equiv \begin{cases} \nabla E_f, & \text{at edges not on } \Gamma, \\ \nabla(E_f, E_c), & \text{at edges on } \Gamma, \end{cases} \tag{59}$$

$$D_c E_c - \beta \nabla \cdot \sigma_c = \begin{cases} \langle R_f \rangle, & \text{on } P(\Omega_f), \\ R_c + \beta \nabla \cdot J, & \text{on } \Omega_c \backslash P(\Omega_f), \end{cases} \tag{60}$$

$$\sigma_c \equiv \nabla E_c, \tag{61}$$

$$J \equiv \begin{cases} \langle \sigma_f \rangle - \sigma_c, & \text{on } \Gamma, \\ 0, & \text{elsewhere.} \end{cases} \tag{62}$$

Here, a subscript f or c denotes the cell-centered value of the corresponding quantity on the fine or coarse grid, respectively, except for the fluxes σ_f and σ_c, which are defined on cell faces. The quantity $\langle R_f \rangle$ denotes the volume-weighted average of R_f on the underlying coarse grid $P(\Omega_f)$, and similarly $\langle \sigma_f \rangle$ denotes the face-weighted average of σ_f on the coarse faces on Γ. The quantity $\nabla(E_f, E_c)$ denotes the gradient on the fine cell faces contained in Γ, which depends on both the fine amplitude E_f and values interpolated from the coarse amplitude E_c in $\Omega_c \backslash P(\Omega_f)$. The quantity J is the jump between the fluxes on Γ computed using the fine and coarse amplitudes. The inclusion of this quantity as in (60) implies that the amplitude E_c on $\Omega_c \backslash P(\Omega_f)$ is decoupled from the value of E_c on $P(\Omega_f)$ and is coupled instead to the fine-grid amplitude E_f through the flux σ_f. In other words, the fine-grid amplitude provides Neumann boundary conditions for the amplitude on $\Omega_c \backslash P(\Omega_f)$, which in turn provide Dirichlet boundary values for the fine-grid amplitude.

Although we seek the solution of (56), the application of the sweeping algorithm described in Section 4.1.2 yields a solution of a slightly different problem, due to the fact that the separate integration of the coarse and fine levels does not preserve the continuity of the normal derivatives at transverse coarse–fine boundaries. Thus, after the fine grid has been integrated to the same plane as the coarse grid (e.g., Fig. 4d), we have instead solved the system

$$D_f \hat{E}_f - \beta \nabla \cdot \hat{\sigma}_f = R_f \quad \text{on } \Omega_f, \tag{63}$$

$$\hat{\sigma}_f \equiv \begin{cases} \nabla \hat{E}_f, & \text{at edges not on } \Gamma, \\ \nabla(\hat{E}_f, \hat{E}_c), & \text{at edges on } \Gamma, \end{cases} \tag{64}$$

$$D_c \hat{E}_c - \beta \nabla \cdot \hat{\sigma}_c = \begin{cases} \langle R_f \rangle, & \text{on } P(\Omega_f), \\ R_c, & \text{on } \Omega_c \backslash P(\Omega_f), \end{cases} \tag{65}$$

$$\hat{\sigma}_c \equiv \nabla \hat{E}_c. \tag{66}$$

To obtain the solution of (56), we must therefore add to the prediction $(\hat{E}_c, \hat{E}_f)$ a composite correction $(\delta E_c, \delta E_f)$ satisfying the system obtained by subtracting (63)–(66) from (58)–(62):

$$D_f \delta E_f - \beta \Delta \cdot \delta \sigma_f = 0 \quad \text{on } \Omega_f, \tag{67}$$

$$\delta \sigma_f \equiv \begin{cases} \nabla \delta E_f, & \text{at edges not on } \Gamma, \\ \nabla(\delta E_f, \delta E_c), & \text{at edges on } \Gamma, \end{cases} \tag{68}$$

$$D_c \delta E_c - \beta \nabla \cdot \delta \sigma_c = \begin{cases} 0, & \text{on } P(\Omega_f), \\ \beta \nabla \cdot \delta J, & \text{on } \Omega_c \backslash P(\Omega_f), \end{cases} \tag{69}$$

$$\delta \sigma_c \equiv \nabla \delta E_c, \tag{70}$$

$$\delta J \equiv \begin{cases} (\langle \delta \sigma_f \rangle - \delta \sigma_c) + (\langle \hat{\sigma}_f \rangle - \hat{\sigma}_c), & \text{on } \Gamma, \\ 0, & \text{elsewhere.} \end{cases} \tag{71}$$

Assuming homogeneous or periodic boundary conditions, the magnitude of the correction $(\delta E_c, \delta E_f)$ obtained by solving (67)–(71) is proportional to the magnitude of the jump $\hat{J} \equiv \langle \hat{\sigma}_f \rangle - \hat{\sigma}_c$. We therefore attempt to reduce the correction magnitude by anticipating this jump using the value $\hat{J}_{old}$ from the previous sweep plane in the coarse-grid integration step. In other words, instead of solving the system (65), (66) on the coarse grid, we solve the system

$$D_c \hat{E}_c - \beta \nabla \cdot \hat{\sigma}_c = \begin{cases} \langle R_f \rangle, & \text{on } P(\Omega_f), \\ R_c + \beta \nabla \cdot \hat{J}_{old}, & \text{on } \Omega_c \backslash P(\Omega_f), \end{cases} \tag{72}$$

$$\hat{\sigma}_c \equiv \nabla \hat{E}_c. \tag{73}$$

The composite correction is then the solution of (67)–(71), where instead of (71) we have

$$\delta J \equiv \begin{cases} (\langle \delta\sigma_f \rangle - \delta\sigma_c) + \hat{J} - \hat{J}_{old}, & \text{on } \Gamma, \\ 0, & \text{elsewhere}. \end{cases} \tag{74}$$

For a sufficiently slowly varying solution, the correction resulting from the modified jump $\hat{J} - \hat{J}_{old}$ will be smaller than that due to $\hat{J}$. In steady state, the correction will be zero.

To solve the composite system defined by (67)–(70) and (74), we employ a fast adaptive composite (FAC) algorithm [17]. FAC utilizes an iterative procedure to combine solutions of linear systems on individual refinement levels. These "level solves" are easier to perform than the full composite system, since the mesh size is uniform on each refinement level and the stencil nonuniformities at the coarse–fine boundaries only affect the enforcement of boundary values. In two spatial dimensions, we use a tridiagonal solver to perform the transverse one-dimensional level solves; for three-dimensional problems, a multigrid algorithm is used for the corresponding two-dimensional transverse level solves.

4.1.4. *Interpolation and Averaging*

The light propagation algorithm involves a number of interpolation and averaging operations to transfer data between coarse and fine grids. For example, in the integration of a single refinement level, the sweep is initialized using data from the next coarser level interpolated in the transverse direction (i.e., the step depicted in Fig. 3b). Averaging of fine data in the transverse direction to obtain coarse data to initialize the subsequent coarse grid step is required after the composite synchronization step in the multilevel integration algorithm. In both the single- and multilevel algorithms, coarse-grid amplitudes are interpolated in the propagation direction to obtain boundary values for the inversion of the transverse linear operators. In this section, we discuss these issues in more detail.

To interpolate complex amplitudes in the transverse direction, we employ an algorithm in which the amplitudes are first linearly interpolated, then scaled by a real-valued factor so that the squared moduli of the resulting fine-grid amplitudes are equal to those obtained by a conservative linear interpolation of the coarse-grid squared moduli. The motivation for this approach is to conserve energy locally with a higher degree of approximation than that obtained by constant interpolation. To describe the explicit formulas in the case of a one-dimensional transverse coordinate, suppose that we want to interpolate coarse amplitudes $E_{c,i}$ to obtain fine amplitudes $E_{f,j}$ on the next finer level, which has been refined by a factor of r. Assuming that each coarse grid cell i is refined into r fine grid cells, indexed by $j = ri, \ldots, r(i+1) - 1$, the fine grid amplitudes are obtained as

$$E_{f,j} = \frac{e_j \tilde{E}_{f,j}}{|\tilde{E}_{f,j}|}, \tag{75}$$

where

$$\tilde{E}_{f,j} \equiv \begin{cases} -\tau_j E_{c,i-1} + (\tau_j + 1) E_{c,i}, & \text{for } j = ri, \ldots, ri + r/2 - 1, \\ (1 - \tau_j) E_{c,i} + \tau_j E_{c,i+1}, & \text{for } j = ri + r/2, \ldots, r(i+1) - 1, \end{cases} \tag{76}$$

$$e_j^2 \equiv |E_{c,i}|^2 + s_j \tau_j, \tag{77}$$

$$\tau_j \equiv \frac{1}{2r} + \frac{1}{r}(j - ri) - \frac{1}{2}, \tag{78}$$

and the slopes s_j are given by

$$s_j \equiv \text{dminmod}(|E_{c,i+1}|^2 - |E_{c,i}|^2, |E_{c,i}|^2 - |E_{c,i-1}|^2), \tag{79}$$

with the double minmod function defined by (42).

To average fine-grid amplitudes to coarse amplitudes in the transverse direction, we employ

$$E_{c,i} = \frac{e_i \tilde{E}_{c,i}}{|\tilde{E}_{c,i}|}, \tag{80}$$

where

$$\tilde{E}_{c,i} \equiv \frac{1}{r} \sum_{j=0}^{r-1} E_{f,ri+j}, \tag{81}$$

$$e_i^2 \equiv \frac{1}{r} \sum_{j=0}^{r-1} |E_{f,ri+j}|^2. \tag{82}$$

The above interpolation and averaging formulas are extended to two transverse dimensions using tensor products.

To obtain boundary values for the fine-grid linear solves (e.g., Fig. 3c), we use an interpolation formula similar to (75)–(78) that is applied in the propagation direction. Since energy conservation is not required by this interpolation step, we employ the simpler slope formulas

$$s_j \equiv \begin{cases} |E_{c,i}|^2 - |E_{c,i-1}|^2, & \text{for } j = ri, \ldots, ri + r/2 - 1, \\ |E_{c,i+1}|^2 - |E_{c,i}|^2, & \text{for } j = ri + r/2, \ldots, r(i+1) - 1, \end{cases} \tag{83}$$

corresponding to linear interpolation of the coarse-grid squared moduli. This choice is compatible with the Crank–Nicholson discretization of (32), which assumes a linear x variation of the amplitude between x-faces. To demonstrate this more concretely, suppose that $E^{\lambda}_{j-1/2}$ is an eigenvector corresponding to an eigenvalue λ of the transverse operator

$$\frac{1}{2k\eta_j^0} \nabla_\perp^2 + k \,\text{diag}\{\eta_{j,m} - \eta_j^0\}. \tag{84}$$

Assume that η varies slowly in the propagation direction x so that it can be regarded as constant over two consecutive cells in x; that is, $\eta_{j,m} = \eta_{j+1,m}$ for each m, and hence, $\eta_j^0 = \eta_{j+1}^0$. The Crank–Nicholson algorithm applied over the Δx step between the cell edges $j - 1/2$ and $j + 1/2$ yields

$$E^{\lambda}_{j+1/2} = \frac{2 + \Delta x \lambda i}{2 - \Delta x \lambda i} E^{\lambda}_{j-1/2}. \tag{85}$$

Thus, the eigenvector $E^{\lambda}_{j-1/2}$ is multiplied by a factor that preserves its modulus (and hence energy) and changes its phase by the angle

$$\theta \equiv \tan^{-1} \frac{4\lambda \Delta x}{4 - \lambda^2 \Delta x^2} = \lambda \Delta x + O(\lambda^3 \Delta x^3) \quad \text{as} \quad \lambda \Delta x \to 0. \tag{86}$$

The leading order term $\lambda\Delta x$ is the exact phase change that would have resulted from integrating (32) discretized only in the transverse direction. The higher order terms therefore represent the Crank–Nicholson error. Now suppose that we had instead employed Crank–Nicholson over two cells and linearly interpolated the resulting amplitude to approximate $E^{\lambda}_{j+1/2}$ (analogous to the interpolation of coarse-grid amplitudes to obtain boundary values for the fine-grid integration). We have

$$\frac{1}{2}\left(E^{\lambda}_{j-1/2}+E^{\lambda}_{j+3/2}\right)=\frac{1}{2}\left(1+\frac{1+\lambda\Delta xi}{1-\lambda\Delta xi}\right)E^{\lambda}_{j-1/2}=\frac{1}{1-\lambda\Delta xi}E^{\lambda}_{j-1/2}. \tag{87}$$

In this case, the eigenvector $E^{\lambda}_{i-1/2}$ is multiplied by a quantity that changes its phase by the angle

$$\tilde{\theta}\equiv\tan^{-1}\lambda\Delta x=\lambda\Delta x+O(\lambda^3\Delta x^3)\quad\text{as}\quad\lambda\Delta x\to 0, \tag{88}$$

which agrees with the phase angle (86) up to the asymptotic order of the error in the Crank–Nicholson discretization. From (87) we also observe that the modulus of the interpolated amplitude differs from that of (85) by a quantity of order $\lambda^2\Delta x^2$. This difference is eliminated by scaling (87) to have the modulus square obtained by linearly interpolating $|E^{\lambda}_{j-1/2}|^2$ and $|E^{\lambda}_{j+3/2}|^2$.

The fact that the transverse interpolation and averaging algorithms locally conserve energy, together with the fact that single steps of the Crank–Nicholson algorithm conserve energy globally (in the transverse dimension), implies that the multilevel light-propagation algorithm conserves energy if the grid hierarchy is refined only in the transverse dimension. In fact, energy conservation to machine-precision accuracy is obtained, provided that the iterative methods used to solve the transverse linear systems are also converged to this precision. On the other hand, strict conservation of energy is not guaranteed if the grid hierarchy contains a level that has been refined in the propagation direction or if the transverse linear systems are only solved approximately.

4.2. Refinement Criteria

Appropriate refinement criteria are important for an effective AMR algorithm. Nevertheless, we do not consider this issue in this paper, leaving it for future work. Instead, heuristic criteria that work well, but not optimally, and that can be easily formulated are adopted here. The criteria used in the numerical examples of the next section are based on values of the light intensity and/or the gradients of the electron number density and the light intensity.

Heuristic conditions must be tuned to minimize the effects of introducing artificial refinement boundaries in the solution domain, although this process is imperfect. Of concern is the introduction of numerical inaccuracies such as reflections from fine/coarse-grid interfaces or, in underrefined regions, numerical solutions irrevocably corrupted by artificial numerical dispersion and/or dissipation. In fact, the Crank–Nicholson scheme used in the paraxial light algorithm suffers from significant phase errors in regions where the grid is too coarse. Thus, our heuristic criteria for this paper have been designed to be forgiving if not efficient.

A choice of refinement criteria better suited to dealing with numerical inaccuracies is based on local error estimation. Such criteria are more difficult to devise and implement, but they are more robust in detecting numerical errors. We are currently implementing an error-based criterion that uses Richardson extrapolation and in future work, we will analyze this and other heuristic criteria to identify those best suited to LPI problems of interest.

5. NUMERICAL RESULTS

The algorithm described in the preceding sections has been implemented in a research code named ALPS (Adaptive Laser Plasma Simulator). ALPS is built upon the SAMRAI (Structured Adaptive Mesh Refinement Applications Infrastructure) system [16], a C++ class library that supports the development of structured AMR application codes. Parallelization is implemented using the Message Passing Interface (MPI) [12] through SAMRAI. ALPS can be compiled as a spatially two-dimensional or three-dimensional executable.

To demonstrate the efficacy of our algorithm as implemented in ALPS, we present three example calculations. The first result is a convergence study on a crossed plane wave problem that establishes a second-order convergence rate in all variables for the uniform grid discretization. A two-dimensional filamentation problem is presented to demonstrate speedup and memory savings by comparing calculations obtained with both a uniform grid and an adaptively refined mesh. Finally, results are reported for a calculation of crossed beams in a plasma flow that highlight the utility of AMR for problems requiring the simulation of a large hydrodynamic region in which the main LPI subregion of interest is relatively small.

The last two test cases have been selected as representative of the types of LPI problems for which AMR may be useful. In many theoretical and experimental studies of LPI mechanisms, the background plasma flows are relatively smooth, and the regions of high laser intensity variation occur in only a small fraction of the computational domain. We therefore have adopted heuristic refinement criteria that follow areas of high laser intensity, noting that, as with any finite difference algorithm, insufficient resolution of the light field can lead to significant numerical phase errors. Problems with high variation over a large fraction of the computational domain, for example due to large numbers of beams, highly filamented beams, or complicated background flows, clearly will not benefit as much from AMR because the computational savings will be offset by the AMR overhead.

5.1. Rate of Convergence

One of the design goals for the discretizations formulated in Section 3 was to achieve second-order temporal and spatial accuracy on a uniform grid. We indeed observe such convergence rates for the plasma and light algorithms tested separately. To verify that a near second-order rate is maintained when the plasma and light equations are integrated using the operator split approach described earlier, we evaluate the convergence of the coupled system on the following test problem.

Consider a physical domain defined by the rectangle $0 \leq x_1 \leq X_1$, $0 \leq x_2 \leq X_2$, where $X_2 = 21.225$ μm and $X_1 = 2X_2$. The domain is filled with a C_5H_{12} plasma whose initial density is $1/10$ of the critical density for the wavelength $\lambda = 0.351$ μm. The initial velocities are zero, and the initial ion and electron temperatures are 1 keV and 3 keV, respectively. No-flow boundary conditions are prescribed at the boundaries in the x_1 direction, with periodic boundary conditions in the x_2 direction.

TABLE I

L_1 Norm of the Differences between the Solution Computed with N = 1024 and Coarser Calculations at N = 128, 256, and 512

	Relative error in the L_1 norm				
N	Intensity	Density	Pressure	x_1-Velocity	x_2-Velocity
128	1.52e-01	2.94e-02	4.74e-02	6.49e-01	4.01e-01
256	7.87e-02	1.25e-02	2.05e-02	2.21e-01	1.72e-01
512	1.42e-02	3.06e-03	5.04e-03	4.96e-02	4.17e-02

The incident light amplitude (time-harmonically enveloped with respect to the frequency $\omega_0 = 2\pi c/\lambda$) on the left boundary is

$$E(t, x_2) \equiv E_1(t)e^{-i(\omega_\delta t+kx_2)} + E_2 e^{i(\omega_\delta t+kx_2)}, \quad t \geq 0, \quad 0 \leq x_2 \leq X_2, \tag{89}$$

where $\omega_\delta = 8.333 \times 10^{11}$ rad/s and $k = 10\pi/X_2$. The plane wave magnitudes are

$$E_1(t) \equiv \begin{cases} (\alpha I_1)^{1/2}\left[1 - \left(1 - \frac{t^2}{t_{ramp}^2}\right)^2\right], & \text{for } t \leq t_{ramp}, \\ (\alpha I_1)^{1/2} & \text{for } t > t_{ramp}, \end{cases} \tag{90}$$
$$E_2 \equiv (\alpha I_2)^{1/2},$$

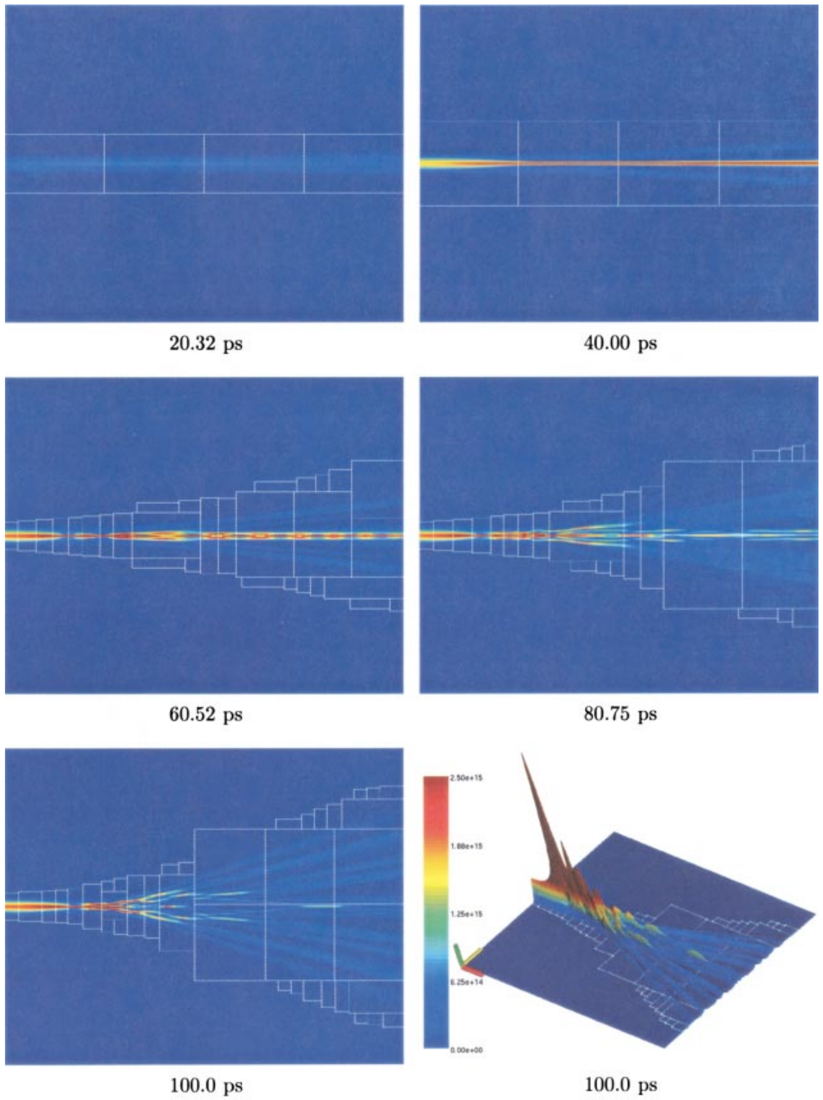
where $I_1 \equiv 10^{16}$ W/cm^2, $I_2 \equiv 10^{12}$ W/cm^2, $t_{ramp} \equiv 7$ ps, and $\alpha \equiv 2 \times 10^4/c\epsilon_0$. For any time t, the prescribed amplitude (89) defines the initial values for light sweeps in the positive x_1 direction, with periodic boundary conditions imposed in the x_2 direction. The interaction of the two plane waves emanating from (89) results in a beat wave that couples to a plasma acoustic wave. In fact, this particular choice of parameters establishes a resonance that transfers energy from the higher frequency plane wave to the lower frequency wave.

The coupled plasma and light system is integrated to a final time of 50 ps on uniform grids of size $2N \times N$, where $N = 128, 256, 512$, and 1024. To estimate the rate of convergence, we regard the solution with $N = 1024$ as the exact solution, and we compute the L_1 norm difference between this solution and the other three approximate solutions, normalized by the L_1 norm of the $N = 1024$ "exact" solution. The relative errors obtained in this manner are displayed in Table I, which shows that the integration of the coupled system is achieving a near second-order convergence rate in all variables.

5.2. AMR Efficiency

We next present a two-dimensional example of the saving of computational resources obtainable from AMR. It should be noted that the maximum potential gains are not fully demonstrated by this problem, since even larger savings are attainable in three dimensions. However, to avoid mixing the issues of AMR speedup and parallel speedup, we are limited by the uniform-grid problem size that we can reasonably run on a single processor.

In this problem, a laser beam is propagated through a uniform, stationary CH plasma filling a 600 μm by 480 μm rectangular domain. The laser, of wavelength 1.06 μm, has a prescribed beam amplitude with a cosine-squared variation over a 40-μm-wide spot centered



20.32 ps

40.00 ps

60.52 ps

80.75 ps

100.0 ps

100.0 ps

FIG. 6. A sequence of intensity plots at various times in the simulation. The white boxes indicate the adaptively refined mesh, and the color scale is the same for all images.

on the left-hand side of the domain. The light source is linearly ramped from a zero initial intensity to a peak intensity of 2×10^{15} W/cm^2 at 50 ps, after which it remains constant. The initial plasma number density is 9.92×10^{19} cm^{-3}, $1/10$ the critical density for the imposed laser light. The initial ion and electron temperatures are 0.5 keV and 1 keV, respectively.

Several light intensity plots from an adaptive simulations are presented in Fig. 6. This computation was run with an initial grid of 512×512 cells covering the entire spatial domain, and an additional refinement level was introduced adaptively where the light intensity exceeded 1.1×10^{12} W/cm^2. The refined level, indicated by the white boxes, is four times finer than the initial coarse grid in each coordinate direction, and regridding was performed every two coarse-grid time steps.

As the input laser intensity increased, a rectangular domain of refined cells was automatically added around the beam; note that the refinement threshold cannot be discerned

TABLE II

Comparison of CPU Times between the 512 × 512 Adaptive Simulation and the Equivalent Uniform Fine-Grid Calculation

	CPU hours		
	Adaptive	Uniform	Speedup
Laser integrator	2.005	3.403	1.70
Plasma integrator	8.446	29.400	3.48
Other	0.054	0.167	3.09
Total run time	10.505	32.970	3.14

because of the color scale required to identify the many orders of magnitude. Due to the ponderomotive force pushing plasma out of the beam, the light focused, eventually leading to the formation of a "hot spot" that caused filamentation and spreading of the beam. The refined level adapted to capture this spreading.

To make a quantitative assessment of the benefits of AMR, it is necessary to consider both the time and memory savings. In Table II CPU timing data are presented for both the 512 × 512 adaptive simulation and an equivalent uniform fine-grid calculation of 2048 × 2048 cells on a single processor of a Compaq AlphaServer 8400. The total speedup for the adaptive computation is 3.14. Breaking down the total time, it is clear that the plasma system integration dominates the computational cost but that it also scales very well with mesh refinement because the adaptive simulation uses on average roughly one-third the number of computational cells. Conversely, the light algorithm does not scale as well, but the current implementation of the light algorithm has not been optimally implemented in the SAMRAI framework.

The evolution of the number of cells in the adaptive simulation relative to the number of cells in an equivalent uniform fine grid is plotted as the solid line in Fig. 7. Assuming a

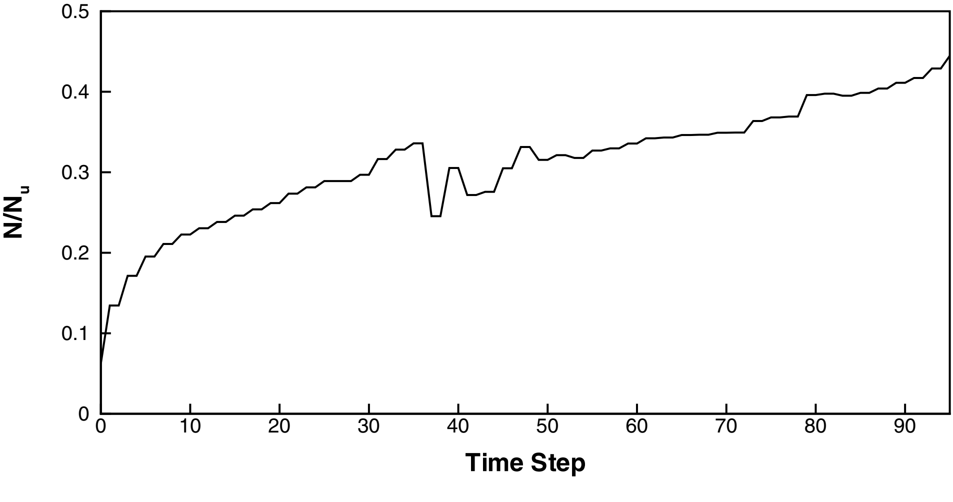


FIG. 7. Evolution of the number of cells at each time step for the 512 × 512, 100 ps adaptive simulation relative to the number of cells in an equivalent uniform fine grid of 2048 × 2048 cells. The number of cells approximates the memory usage during the adaptive run relative to that for an equivalent uniform fine grid.

constant amount of memory per cell, this is equivalent to a measure of the relative memory usage of the adaptive simulation. The simulation begins with 6% of the equivalent uniform fine-grid cells and ends at approximately 45%, verifying that AMR yields a substantial memory savings provided that the solution exhibits sufficiently local features. The variation in the number of cells between time steps 35 and 50 corresponds to mesh adjustments due to the focusing and subsequent spreading of the laser beam.

5.3. Crossed Beams in an Expanding Plasma

Finally, we present results for a problem in which two laser beams cross in an expanding plasma, representing an idealization of a physical experiment in which a foil is exploded by a separate heater beam in the vicinity of the crossing beams. We model the initial conditions as an isothermal, quasineutral CH plasma freely expanding in the x_2-direction in a computational domain of size 640 by 500 μm. The initial number density and velocity distributions are given by

$$\begin{aligned} \frac{n(x_2)}{n_0} &= \exp\left(-\frac{x_2}{L}\right), \\ \frac{u(x_2)}{c_s} &= \left(0, 1+\frac{x_2}{L}\right)^T, \end{aligned} \tag{91}$$

such that, on the vertical centerline ($x_2 = 0$), the plasma velocity is the ion acoustic velocity c_s and the number density is $n_0 = 9.92 \times 10^{19}$ cm^{-3} ($0.1n_c$). The number density at the top of the domain ($x_2 = 250$ μm) is half the centerline density, which fixes the length scale $L = 250/\ln(2)$ μm. The initial ion and electron temperatures are 0.5 and 1 keV, respectively.

Two 1.06-μm-wavelength laser beams instantaneously enter through the left boundary at the initial time. Each beam has a cosine-squared amplitude variation over a 40-μm-wide spot and a maximum intensity of 5.1×10^{14} W/cm^2. The beams are centered at $x_2 = \pm 80$ μm and initially propagate at angles $\mp 15.7^\circ$ to the horizontal so as to intersect, in the absence of plasma, at the center of the domain.

Since little happens outside the immediate neighborhoods of the beams, the lowest level grid is very coarse at 256 × 160 cells. An additional level of refinement by a factor of 8 is initially imposed to provide resolution sufficient to propagate the beams in the correct directions. Subsequently, this level adaptively refines on intensity levels above 5×10^{12} W/cm^2, and a second level of refinement by a factor of 2 resolves regions where either the intensity exceeds 10^{14} W/cm^2 or intensity gradients exceed 10^{19} W/cm^3.

The adaptively computed light intensity is plotted in Fig. 8, where red denotes the highest intensity and blue the lowest. The white boxes indicate the first refined level and the green boxes indicate the second and finest level. Two major effects are present. First, since light will refract due to gradients in the plasma density, both beams can be seen to deflect upward in the direction of lower plasma density. Second, a "grating" is produced in the overlap region where a resonant coupling exists between the beat wave and an ion acoustic wave (see, e.g., [10, 21, 28]). This resonance results in a transfer of energy between the beams, as seen in Fig. 8 where the beam traveling from the lower left to the upper right obtains a much higher intensity. The ability to use local mesh refinement to sufficiently resolve the

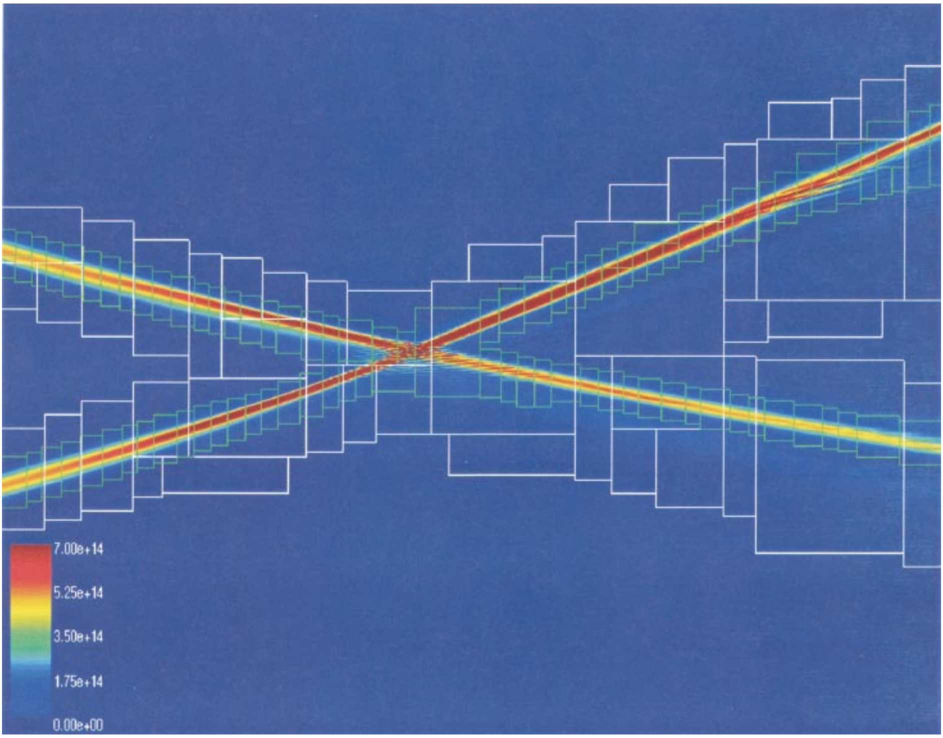


FIG. 8. Adaptive calculation of light intensity for crossed beams in an expanding plasma. The white boxes indicate the first refined level and the green boxes indicate the second and finest level.

overlap region is important to ensure that the resonance is not detuned by dissipation in the numerical scheme.

The memory efficiency of the adaptive algorithm is demonstrated by the data in Table III. The area covered by the first level of refinement actually decreases from its initial specification, with an average coverage of approximately one-third of the domain. The finest grid covers on average 12.69%, and the total number of cells, relative to the total number of cells in an equivalent uniform fine grid, is on average 21.76%.

TABLE III

The Average Area Covered by Each Refinement Level and the Total Number of Cells N, Expressed as a Percentage of the Number of Cells in an Equivalent Uniform Fine Grid, N_u

	Area coverage (%)		
	Level 1	Level 2	N/N_u (%)
First step	49.86	13.13	25.99
Last step	36.94	12.51	22.13
Maximum	49.86	14.42	27.27
Minimum	31.90	11.78	20.62
Average	34.72	12.69	21.76

6. CONCLUSION

We have presented an approach for incorporating AMR into the simulation of laser plasma filamentation. AMR is most appropriate for problems with distinct scale separation, such as problems that are "hydrodynamically large," where the regions of plasma flow that can be resolved with a relatively coarse grid are substantially larger than the LPI regions requiring finer grid resolution. The use of locally refined grids results in significantly reduced execution times and memory utilization. More importantly, given fixed computational resources, the use of AMR enables the consideration of larger problems than could be handled otherwise.

AMR necessarily introduces significant additional complexity in the integration algorithm and the associated code implementation. In particular, the discretizations of the plasma fluid and light models must accommodate essential matching conditions at the interfaces between coarse and fine grids. For the plasma model, which we discretized using a high-resolution Godunov method, we achieved this by enforcing the flux continuity that naturally results from the conservative formulation of the fluid equations. In the paraxial light discretization, we introduced some specialized interpolation and composite synchronization steps to match solutions at coarse–fine boundaries. The latter was formulated in the context of a Crank–Nicholson finite difference discretization of the paraxial wave model, since the use of locally refined grids inhibits the direct use of fast transform methods that can be employed on uniform grids [4]. Since a linear system solve replaces FFTs in our implicit finite difference approach, an efficient solver that can accommodate the coupling of multiple refinement levels is essential.

Some important issues remain for future investigation. Refinement criteria based on estimates of the error in the approximate solutions would be a significant improvement over heuristic criteria that rely on *a priori* knowledge of the solution characteristics. The approach we have taken here for pure filamentation problems could in principle be extended to LPI problems involving parametric instabilities such as stimulated Brillouin and Raman scattering by the inclusion of additional paraxial backscattered light and acoustic wave models following the strategy employed in [5]. Acoustic wave damping is an important physical effect in LPI problems that is challenging to implement on nonuniform grids due to the nonlocal character of configuration space formulations. Landau damping models are obtained by considering additional moments of the Vlasov equation beyond those yielding the fluid model considered here. Since the closure rules proposed to date have been formulated in Fourier space [15], the configuration space representations involve a convolution integral that would be expensive to treat directly. On the other hand, as with the paraxial wave equation solution algorithm, the use of locally refined grids inhibits the use of fast transform methods to apply the damping operator in Fourier space. We are continuing to investigate these and other issues.

ACKNOWLEDGMENTS

We gratefully acknowledge the assistance of R. L. Berger, D. E. Hinkel, A. B. Langdon, C. H. Still, E. A. Williams (Plasma Theory Group, X Division, Defense and Nuclear Technologies Directorate), B. I. Cohen, and L. M. Divol (Physics and Advanced Technologies Directorate) at Lawrence Livermore National Laboratory (LLNL) for many helpful discussions on laser plasma physics and numerical modeling, as well as the use of their uniform grid code pF3D for comparison purposes. We also thank R. D. Hornung, S. G. Smith, and A. M. Wissink at LLNL for providing the SAMRAI software system [16] and adding many enhancements to support the building of the ALPS code.

REFERENCES

1. A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, *J. Comput. Phys.* **142**, 1 (1998), doi:10.1006/jcph.1998.5890.
2. M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* **82**(1), 64 (1989).
3. M. J. Berger and J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* **53**, 484 (1984).
4. R. L. Berger, B. F. Lasinski, T. B. Kaiser, E. A. Williams, A. B. Langdon, and B. I. Cohen, Theory and three-dimensional simulation of light filamentation in laser-produced plasma, *Phys. Fluids B* **5**(7), 2243 (1993).
5. R. L. Berger, C. H. Still, E. A. Williams, and A. B. Langdon, On the dominant and subdominant behavior of stimulated Raman and Brillouin scattering driven by nonuniform laser beams, *Phys. Plasmas* **5**(12), 4337 (1998).
6. P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, *J. Comput. Phys.* **87**, 171 (1990).
7. P. Colella, M. R. Dorr, and D. D. Wake, A conservative finite difference method for the numerical solution of plasma fluid equations, *J. Comput. Phys.* **149**, 168 (1999), doi:10.1006/jcph.1998.6136.
8. P. Colella, M. R. Dorr, and D. D. Wake, Numerical solution of plasma fluid equations using locally refined grids, *J. Comput. Phys.* **152**, 550 (1999), doi:10.1006/jcph.1999.6245.
9. D. Colton and R. Kress, *Inverse Acoustic and Electromagnetic Scattering Theory* (Springer-Verlag, New York, 1998).
10. V. V. Eliseev, W. Rozmus, V. T. Tikhonchuk, and C. E. Capjack, Interaction of crossed laser beams with plasmas, *Phys. Plasmas* **3**(6), 2215 (1996).
11. M. D. Feit and J. A. Fleck, Jr., Beam nonparaxiality, filament formation, and beam breakup in the self-focusing of optical beams, *J. Opt. Soc. Am. B* **5**(3), 633 (1988).
12. Message Passing Interface Forum, available at http://www.mpi-forum.org.
13. F. X. Garaizar and J. A. Trangenstein, Adaptive mesh refinement and front tracking for shear bands in an antiplane shear model, *SIAM J. Sci. Comput.* **20**(2), 750 (1998).
14. V. E. Golant, A. P. Zhilinsky, and I. E. Sakharov, *Fundamentals of Plasma Physics* (Wiley, New York, 1980).
15. G. W. Hammett and F. W. Perkins, Fluid moment models for Landau damping with application to the ion-temperature-gradient instability, *Phys. Rev. Lett.* **64**(25), 3019 (1990).
16. R. D. Hornung and S. R. Kohn, Managing application complexity in the SAMRAI object-oriented framework, Technical Report UCRL-JC-141749, Lawrence Livermore National Laboratory (2001). To appear in Concurrency: Practice and Experience. Available at: http://www.llnl.gov/CASC/SAMRAI.
17. S. F. McCornick, *Multilevel Adaptive Methods for Partial Differential Equations*, Frontiers in Applied Mathematics, Vol. 6 (Soc. for Industr. & Appl. Math., Philadelphia, 1989).
18. D. R. Nicholson, *Introduction to Plasma Theory*, Krieger, Melbourne, FL, 1992.
19. R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, W. A. Fiveland, and J. P. Jesse, An adaptive projection method for unsteady, low-Mach number combustion, *Combust. Sci. and Tech.* **140**, 123 (1998).
20. National Ignition Facility Program, available at http://www.llnl.gov/nif.
21. H. A. Rose and S. Ghosal, Nonlinear theory of power transfer between multiple crossed laser beams in a flowing plasma, *Phys. Plasmas* **5**(5), 1461 (1998).
22. C. H. Still, R. L. Berger, A. B. Langdon, and E. A. Williams, *Three-Dimensional Nonlinear Hydrodynamics Code to Study Laser–Plasma Interactions*, Technical Report UCRL-LR 105821-96-4, Lawrence Livermore National Laboratory (1996).
23. J. A. Trangenstein, personal communication, 1998.
24. J. A. Trangenstein, A second-order Godunov algorithm for two-dimensional solid mechanics, *Comput. Mech.* **13**, 343 (1994).
25. J. A. Trangenstein, Adaptive mesh refinement for wave propagation in nonlinear solids, *SIAM J. Sci. Stat. Comput.* **16**, 819 (1995).

26. J. A. Trangenstein and P. Colella, A higher-order Godunov method for modeling finite deformation in elastic-plastic solids, *Comm. Pure. Appl. Math.* **44**, 41 (1991).

27. B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, *J. Comput. Phys.* **32**, 101 (1979).

28. K. B. Wharton, R. K. Kirkwood, S. H. Glenzer, K. G. Estabrook, B. B. Afeyan, B. I. Cohen, J. D. Moody, B. J. MacGowan, and C. Joshi, Observation of resonant energy transfer between identical-frequency laser beams, *Phys. Plasmas* **6**(5), 2144 (1999).

29. C. Yamanaka, *Introduction to Laser Fusion* (Harwood Academic, Chur, 1991).