Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 216 (2006) 153–178
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Event-driven, hybrid particle-in-cell simulation: A new paradigm for multi-scale plasma modeling

Y.A. Omelchenko, H. Karimabadi *

SciberQuest, Inc., Computational Physics, 777 South Highway 101, Suite 108, Solana Beach, CA 92075-2623, United States

Received 11 July 2005; received in revised form 10 October 2005; accepted 25 November 2005
Available online 18 January 2006

Abstract

Particle-in-cell models have become standard computational tools for studying complex nonlinear phenomena in space and laboratory plasmas. These simulations are normally very compute-intensive since they require time integration of strongly coupled equations governing the field and particle dynamics. As a result, despite a significant progress in hardware technology, particle-in-cell codes are rarely used to simulate long-time evolution of large-scale systems with strongly varying temporal and spatial scales. We propose an alternative paradigm to time stepping, which is traditionally used for time integration of such systems. This new approach is based on explicit discrete-event simulation technology. It offers distinct advantages over synchronous time stepping: (i) updates of individual macro-particles and discrete field elements are performed *asynchronously*, (ii) local time increments are determined and *self-adaptively* adjusted in time through scheduling and execution of physically meaningful local updates (''events''). The event-driven time advance is accurate, free of the global Courant condition, stable, parallelizable, extendable to multiple dimensions and well suited for nonuniform spatial meshes. We demonstrate the new method on a one-dimensional hybrid particle-in-cell model with applications to several plasma discontinuities, including a high-Mach-number fast magnetosonic shock and the associated plasma turbulence.

Keywords: Hybrid; Particle-in-cell; PIC; Multi-scale; Asynchronous; Discrete-event simulation; Magnetosonic shocks

1. Introduction

The collective kinetic properties of laboratory and space plasmas are commonly simulated by the particle-in-cell (PIC) method [1–3], where the self-consistent (coupled) equations of particle motion and field evolution are usually solved on a spatial mesh. The forces acting on individual macro-particles as well as the mesh-defined current and charge densities are obtained via mesh-particle interpolation (gather–scatter) operations. The full-PIC models follow the motion of both electron and ion macro-particles in self-consistent fields governed by the Maxwell equations. In this paper, we will focus on the hybrid-PIC approximation [4,5], where the ion species are represented by macro-particles, whereas the electrons are described by quasineutral fluid

* Corresponding author. Tel.: +1 8587937063; fax: +1 8587775684.
E-mail address: homak@sciberquest.com (H. Karimabadi).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.11.029

equations (considered to be part of the field solve). The traditional time-dependent PIC models employ time stepping for updating macro-particles and fields in time. Normally, in explicit hybrid schemes, the particle orbits are leap-frog integrated, and the electric and magnetic fields are obtained with proper predictor–corrector schemes [4,5]. The maximum time-step sizes allowed for stable synchronous advances of particle and field quantities on a spatial mesh are known to be restricted by relevant Courant conditions. These conditions may become especially prohibitive for computational problems with strongly varying time scales [6]. In addition, in situations where physical processes are characterized by irregular temporal dynamics (e.g. turbulence), synchronous updates may have difficulty adjusting time steps in a timely fashion to correctly capture phase correlations among fast-evolving variables, which may give rise to numerical instabilities. In this paper, we address these issues by describing a self-adaptive, event-driven approach to time integration of such complex, multi-scale systems. This paper extends and generalizes our works on asynchronous electrostatic PIC simulations [7] and flux-conservative partial differential equations (PDEs) [8]. It is organized as follows. In Section 2, we introduce a general discrete-event simulation (DES) paradigm for the time-dependent PIC method. Section 3 is devoted to the description of hybrid plasma approximation. In Sections 4 and 5, we describe self-adaptive DES algorithms for time integration of the hybrid model. In Section 6, we consider an additional (ad hoc) computational algorithm that enables ''on-the-fly'' activation of mesh cells in accordance with evolving solution. In Section 7, we validate the new DES hybrid code against two different time-stepping codes. In Section 8, we present results from simulations of high-Mach-number kinetic plasma turbulence. Finally, Section 9 concludes this paper with a summary of the most important results obtained.

2. PIC-DES paradigm

Discrete-event (event-driven) simulations have their origin in operations research and management science, war games and telecommunications [9,10]. This powerful methodology has recently been applied to simulate electrostatic PIC effects [7] and advection–diffusion-reaction equations [8].

In discrete-event simulations, the temporal evolution of a global system is modeled by executing discrete *events*, which force finite transitions of the system from one global state to another [9]. At minimum, each discrete-event object is characterized by a method (*process function*) for modifying the corresponding local state of the system and a point in simulated time, t_e (*time stamp*), at which the process function is *scheduled* to be executed by the DES control flow executive (*engine*). A physics-based discrete-event simulation normally begins by scheduling events (i.e., predicting future updates) for all micro-states (e.g., particles, mesh-defined variables), based on the rates of change derived from the governing equations for these quantities [7,8]. All *pending* (scheduled but not yet executed) events are sorted into the *event queue* by their time stamps in non-decreasing order so that the time stamp of the top event always corresponds to the earliest event in the simulation. The DES engine repeatedly removes the earliest (top) event from the event list and *processes* the local state corresponding to that event. The global system progress in simulated time is monitored by the simulation clock, which is advanced upon processing each event by setting its current time, t_{clock} to the time stamp of the event being processed, $t_{\text{clock}} = t_e$ (Fig. 1).

In time-dependent mesh-based models, each spatial element (''cell'') may be assigned a number of discrete micro-states (further simply referred to as ''states'') characterized by unique simulation variables. For instance, in PIC simulations, for cell k we introduce states, $\hat{P}_k$ (PIC-state) and $\hat{F}_k$ (F-state), which correspond to the local particle and field descriptions, respectively. From a programming standpoint, these states encapsulate data and evolution rules associated with their physical representations (note that in a multi-species PIC model, each particle species, s in cell k may be represented by a separate state, $\hat{P}_k^s$). Having been initialized and scheduled at start-up, the PIC and field states evolve and interact in time in accordance with predefined causality and communication rules very much like processes in a preemptive operating system environment. Their *process* functions are composed of three common algorithmic phases (Fig. 1): (i) the time *advance* algorithm; (ii) the *synchronization* (inter-state communication) phase and (iii) the event *scheduling* procedure.

The basic PIC-DES programming structures have been described previously [7]. Here we repeat and generalize the most essential provisions of that work. $\hat{P}_k$ encapsulates a time-sorted queue of *wrapper* objects corresponding to macro-particles located inside cell k. Each particle wrapper encapsulates an index to the

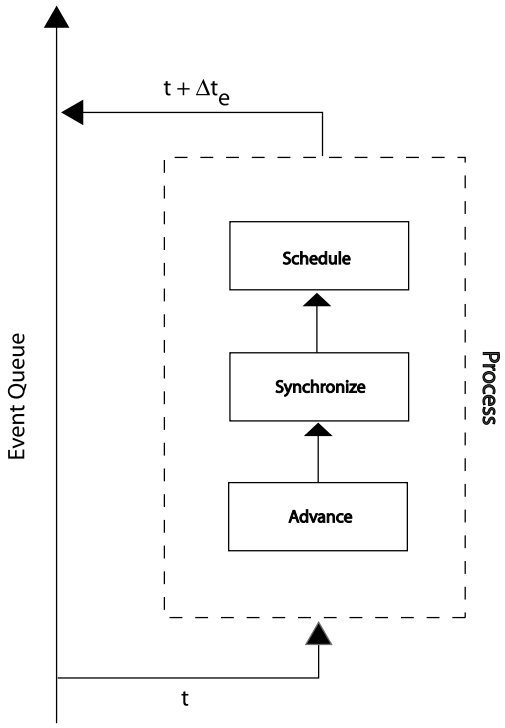


Fig. 1. The typical phases of event-driven time integration. Upon executing the process function of an event with a timestamp, t_e, the global simulation time $t_{clock} = t$ is advanced by an irregular time increment, $\Delta t_e = t_e - t$. Note that the event synchronization phase may result in retracting pending events and scheduling new events for nonlocal micro-states.

location of the corresponding macro-particle in memory storage and an estimated move time of that particle. To facilitate operations related to dynamic memory reallocation, we use indirect integer array indices instead of direct pointers to particle memory locations. The process time of $\hat{P}_k$ coincides with the earliest move time in its PIC queue (or infinity, if the latter is empty). Depending on the PIC interpolation scheme, particles in $\hat{P}_k$ may interact (through gather–scatter operations) with mesh-defined quantities defined in a stencil-wide neighborhood of cell k. Therefore, in principle, particle-field ($\hat{P} \to \hat{F}$) and field-particle ($\hat{F} \to \hat{P}$) synchronization operations may be nonlocal. In this paper, for the sake of simplicity, we adopt the nearest grid point (NGP) interpolation [1,2], which results in local gather–scatter and synchronization updates (Fig. 2). Note that the priority queue-based implementation of PIC-state automatically ensures proper particle–particle ($\hat{P} \to \hat{P}$) synchronization when the next move time of the particle being updated happens to be earlier than the process time of its new PIC-state [7]. Should this PIC-state have a pending event, it is *retracted* and a new one is scheduled accordingly. The algorithmic details of event-driven and synchronization operations are provided below (see Sections 4 and 5).

In general, an F-state event may correspond to the time advance of a single solution variable or a group of synchronously updated variables defined at each cell of the spatial mesh. Electromagnetic PIC models describe the time evolution of magnetic and electric fluxes and plasma sources. A general event-driven approach to solution of flux-conservative equations was developed elsewhere [8]. In essence, this algorithm advances field quantities based on pre-defined "target" (threshold) increment values, rather than prescribed time increments. This enables truly asynchronous, *self-adaptive* integration that strictly preserves local flux conservation laws. Note that in the case of time-dependent field equations, field update events are scheduled and processed asynchronously with respect to particle push events. This introduces a new degree of complexity compared to the previously considered electrostatic PIC-DES model [7], where local field updates were automatically triggered by particle displacements. In addition, self-adaptive field–field ($\hat{F} \to \hat{F}$) flux synchronization updates may also lead to indirect (i.e., nonlocal) interactions between field and particle states.

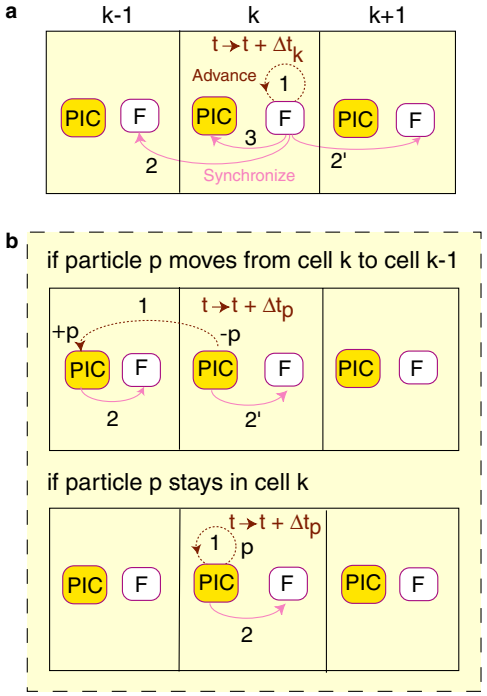


Fig. 2. Schematics of event synchronization updates executed upon the processing of field and PIC events in cell k: (a) F-state $(\hat{F}_k)$ is advanced with a time increment, Δt_k; (b) particle p of PIC-state $(\hat{P}_k)$ is pushed with an individual time increment, Δt_p. The numbers next to the arrowed lines indicate the sequential order of the corresponding event-driven (1) and synchronization (2 and 3) updates.

The most significant property of DES is its *self-adaptivity*. Indeed, contrary to the spirit of traditional time-stepping schemes, event-driven computation follows *causal* (interactive) rather than *temporal* (parametric) dependencies existing among different components of the system (e.g., particles and fields). Consequently, an actual simulation model is able to control its numerical behavior by adjusting local update rates of physical quantities in accordance with physical process rates [8]. DES integration is *explicit* and therefore, in principle, time reversible with the exception of extraneous effects such as boundary damping conditions, particles losses, etc.

The temporal accuracy of a time-stepping scheme depends on the order of its time integrator and the global time-step size. The numerical errors are usually controlled by observing convergence of results obtained with different time increments. Likewise, the temporal resolution of a DES model depends on the order of temporal discretization of the governing equations and the threshold values for physical quantities of interest. However, self-adaptive DES solutions are able to *asynchronously* increase local computation accuracy by reducing local time increments. This results in automated mitigation of the contribution of large time derivatives to numerical approximation errors, leading to enhanced numerical stability. Therefore, multi-scale DES models prove to be more robust and CPU efficient than their time-stepping counterparts.

3. Hybrid-PIC model

The hybrid-PIC description [4,5] may be considered as an intermediary plasma approximation between (one-fluid or two-fluid) magnetohydrodynamics (MHD) and fully kinetic models. The hybrid simulation paradigm provides a full particle-in-cell description for the ions and treats the electrons as a fluid. The relevant scales are the ion gyroradius, $r_{ci} \equiv v_i/\Omega_{ci}$ and the inertial length, $\delta_i \equiv c/\omega_{pi}$ (v_i is the characteristic ion velocity; ω_{pi}, Ω_{ci} are the ion plasma and cyclotron frequencies, respectively). Consequently, to fully resolve ion dynamics in spatial regions of interest, the grid spacing in hybrid models has to be smaller than these characteristic lengths.

3.1. Hybrid plasma approximation

The traditional hybrid models assume the charge quasi-neutrality condition:

$$en_e = q_i n_i, \tag{1}$$

where $-e$, q_i are the electron and ion charges and n_e, n_i are the electron and ion number densities, respectively (we assume a single ion species in this paper). The electrons are usually treated as a collisional, inertialess ($m_e d\mathbf{v}_e/dt = 0$) fluid. This results in the following form of Ohm's law for the electric field $\mathbf{E}$:

$$\mathbf{E} = \frac{\mathbf{j}_e \times \mathbf{B}^{tot}}{en_e c} - \frac{\nabla p_e}{en_e} + \eta \mathbf{j}, \quad \mathbf{j} = \mathbf{j}_e + \mathbf{j}_i, \tag{2}$$

where $\mathbf{j}_e$, p_e are the electron current and (scalar) pressure respectively, η is the plasma resistivity, $\mathbf{B}^{tot}$ is the total magnetic field, and $\mathbf{j}_i$ is the ion current density collected at mesh cells by summing up relevant macro-particle contributions (V_{cell} is the cell volume, q_p and $S(r)$ are the macro-particle charge and form-factor, respectively):

$$\mathbf{j}_i = \sum_p q_p S(|\mathbf{r} - \mathbf{r}_p|)\mathbf{v}_p / V_{cell}. \tag{3}$$

The electron pressure is normally excluded from Eq. (2) through an adiabatic equation-of-state:

$$p_e = n_e T_e = C_p n_e^{\gamma}, \quad \gamma = 5/3, \quad C_p = \text{const}. \tag{4}$$

Ions with the charge q_i and the mass m_i move in the electric and magnetic fields under the influence of electron–ion collisions (conserving the total plasma momentum) in accordance with the non-relativistic equations of motion,

$$\frac{d\mathbf{v}_p}{dt} = \frac{q_i}{m_i}\left(\mathbf{E}'_p + \frac{\mathbf{v}_p \times \mathbf{B}_p^{tot}}{c}\right), \quad \frac{d\mathbf{r}_p}{dt} = \mathbf{v}_p, \tag{5}$$

where we introduced the effective electric field, $\mathbf{E}'$:

$$\mathbf{E}' = \mathbf{E} - \eta \mathbf{j}. \tag{6}$$

In Eq. (5) $\mathbf{E}'_p$, $\mathbf{B}_p^{tot}$ are evaluated at the particle positions $\mathbf{r}_p$ by summing up the contributions from neighboring mesh cells, k:

$$\mathbf{E}'_p = \sum_k \mathbf{E}'_k S(|\mathbf{r}_k - \mathbf{r}_p|), \quad \mathbf{B}_p^{tot} = \sum_k \mathbf{B}_k^{tot} S(|\mathbf{r}_k - \mathbf{r}_p|). \tag{7}$$

Ampere's law is written in the low-frequency Darwin (radiation-free) limit for the self-generated magnetic field, $\mathbf{B}$:

$$\nabla \times \mathbf{B} = \frac{4\pi}{c}\mathbf{j}. \tag{8}$$

Finally, the magnetic field is advanced in time via Faraday's law:

$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}. \tag{9}$$

Eqs. (1)–(9) constitute a closed set of equations governing plasma evolution in the quasi-neutral hybrid approximation. More detailed discussions of underlying physical assumptions and further extensions to this formulation can be found in [4,5].

3.2. Numerical challenges of hybrid codes

There are two main factors that determine the stability and accuracy of *local* time integration of the hybrid equations: (i) the particle push time, Δt_{PIC}, which needs to be the smallest of the local minimum particle cell transit time and a fraction (say, $\lambda_{PIC} \leqslant 0.1$) of the local inverse ion gyrofrequency, and (ii) the field update

time, Δt_w restricted by the local CFL condition on plasma wave propagation. Mathematically, these two conditions can be expressed as follows:

$$\Delta t_{\mathrm{PIC}} < \min(\Delta x/|v_{\mathrm{PIC}}|^{\max}, \lambda_{\mathrm{PIC}}\Omega_{\mathrm{ci}}^{-1}), \tag{10a}$$

$$\Delta t_w < \frac{(\Delta x/\delta_i)^2}{\pi\sqrt{D}}\Omega_{\mathrm{ci}}^{-1}. \tag{10b}$$

Here D is the dimensionality of the problem, Δx is the mesh size, and $|v_{\mathrm{PIC}}|^{\max}$ is the maximum absolute particle velocity component in a given cell. Note that condition (10b) is based on the whistler mode [11], which is supported due to the inclusion of the Hall term in Eq. (2). Whistlers are usually the fastest propagating waves in hybrid applications. When electron inertia is not neglected, the whistler spectrum has a cutoff at the electron cyclotron frequency, Ω_{ce} due to Landau damping. However, in the inertialess hybrid approximation ($m_e \to 0$) the whistler frequency spectrum is unbounded. As a result, the CFL constraint on field integration (Eq. (10b)) may become very restrictive. To avoid generation of unphysical oscillations in the low-density regions, one usually imposes a minimum value on the electron density, $n_{\min}$ ($n_e \geqslant n_{\min}$). This cutoff value is assumed to be a small fraction of the characteristic plasma density, which is physically equivalent to the existence of a small cold background ion population.

In order to keep the hybrid computation stable, restrictions (10) have to be satisfied globally at all times. This poses severe challenges for simulations of multi-scale systems such as hybrid simulations of the solar wind interaction with the Earth's magnetosphere (see Appendix A.1). Standard hybrid predictor–corrector field solvers [4] may have difficulty capturing these variations in time. Iterative solvers [12,13] are based on error estimation and usually more effective in detecting a developing instability since they dynamically track the number of iterations required for convergence and adjust the global time-step size accordingly. However, this inevitably results in a proportional slow-down of the simulation.

3.3. Spatial and temporal discretization

We consider a one-dimensional hybrid model based on Eqs. (1)–(9). We apply a staggered mesh approach [13] to the spatial discretization of the field equations and define time integrators for the field and particle solvers in accordance with event-driven methodology.

3.3.1. Field solve

We assume that the global computation domain, $x \in [0, L]$, is partitioned into a uniform mesh composed of N cells ($k = 0, N-1$), with the cell size $\Delta x = L/N$, cell centers $x_k = (k + 1/2)\Delta x$, and cell "face" positions $x_{k-1/2} = k\Delta x$. In addition, ghost cells $k = -1, N$ (with centers at x_{-1} and x_N) are added to the left and right boundaries of the domain. All mesh-based quantities, except the self-generated magnetic field $\mathbf{B}_{k-1/2}$, are defined at cell centers x_k. $\mathbf{B}_{k-1/2}$ is defined at cell faces $x_{k-1/2}$ in order to preserve the numerical analogue of $\mathrm{div}\,\mathbf{B} = 0$ in accordance with Faraday's law (Eq. (9)). This monopole-free property of the magnetic field in one-dimension is automatically reduced to the following requirement:

$$B_k^{x,\mathrm{tot}} = \mathrm{const} = B_0^x, \tag{11}$$

where $\mathbf{B}_0 = (B_0^x, 0, B_0^z)$ is the stationary applied magnetic field. To enable convenient asynchronous computation on staggered meshes, we introduce the vector-potential function, $\mathbf{A}$:

$$\mathbf{B} = \nabla \times \mathbf{A}. \tag{12}$$

By defining $\mathbf{A}$ at the cell centers and using second-order approximations for the spatial derivatives, we reduce Eq. (12) to the following finite-difference relations:

$$B_{k-1/2}^y = -(A_k^z - A_{k-1}^z)/\Delta x, \quad B_{k-1/2}^z = (A_k^y - A_{k-1}^y)/\Delta x. \tag{13}$$

Applying the forward-differencing scheme to the time derivative in Eq. (9) and taking into account Eq. (12) results in a local time evolution equation for the cell-centered $\mathbf{A}$:

$$\frac{\mathbf{A}_k(t_k+\Delta t_k)-\mathbf{A}_k(t_k)}{\Delta t_k}=-c\mathbf{E}_k(t_k), \tag{14}$$

where t_k is the last update time of $\hat{F}_k$ and $\Delta t_k = t_{\text{clock}} - t_k$. The cell-centered electric field $\mathbf{E}_k$ is computed by combining discrete analogues of Eqs. (2) and (8):

$$\mathbf{E}_k=\frac{\mathbf{j}_{ek}\times\mathbf{B}_k^{\text{tot}}}{en_{ek}c}-\frac{\nabla p_{e,k}}{en_{ek}}+\eta_k\mathbf{j}_k, \tag{15}$$

$$\nabla p_{e,k}=\frac{1}{2}(\nabla p_{e,k-1/2}+\nabla p_{e,k+1/2}),\quad \nabla p_{e,k+1/2}=\gamma C_{\text{p}} n_{e,k+1/2}^{\gamma-1}\frac{n_{e,k+1}-n_{e,k}}{\Delta x}, \tag{16}$$

$$n_{e,k+1/2}=\frac{1}{2}(n_{e,k}+n_{e,k+1}),\quad \mathbf{B}_k^{\text{tot}}=\frac{1}{2}(\mathbf{B}_{k-1/2}+\mathbf{B}_{k+1/2})+\mathbf{B}_{0k}, \tag{17}$$

$$\mathbf{j}_{ek}=\mathbf{j}_k-\mathbf{j}_{ik},\quad j_k^x=0,\quad j_k^y=-\frac{c}{4\pi\Delta x}(B_{k+1/2}^z-B_{k-1/2}^z),\quad j_k^z=\frac{c}{4\pi\Delta x}(B_{k+1/2}^y-B_{k-1/2}^y). \tag{18}$$

To reduce the potentially harmful effect of plasma pressure (density) fluctuations on ion dynamics [14], we smooth the electron pressure gradient $\nabla p_{e,k}$ before using it in Eq. (15) by applying two-point temporal and three-point (binomial) spatial filters. Expressions for magnetic field components (Eq. (13)) require that appropriate boundary conditions be defined at the ghost cell centers (for **A**) or faces (for **B**). In this paper we use a linear combination of Neumann and Dirichlet conditions for the vector-potential, **A**:

$$D_g^\zeta A_g^\zeta+N_g^\zeta\left[\frac{\mathrm{d}A^\zeta}{\mathrm{d}x}\right]_g=R_g^\zeta,\quad g=-1,N,\ \ \zeta=y,z. \tag{19}$$

Here D_g^ζ and N_g^ζ and N_g^ζ are arbitrary constant parameters. In discrete form, Eq. (19) is readily inverted to express values A_g^ζ in ghost cells g in terms of values A_s^ζ in adjacent (boundary) cells s:

$$A_g^\zeta=\frac{R_g^\zeta+A_s^\zeta(\alpha_g^\zeta-D_g^\zeta/2)}{\alpha_g^\zeta+D_g^\zeta/2},\quad \alpha_g^\zeta=(g-s)N_g^\zeta/\Delta x. \tag{20}$$

Finally, Eq. (15) needs to be complemented by a model for the phenomenological resistivity η. In this paper we consider only collisionless plasmas (with negligible physical dissipation) and apply small artificial resistivity for the purpose of filtering spurious whistler oscillations generated by noisy particle motion. We use the following dissipative filter based on detection of short-wavelength variations of the total current:

$$\eta_k=\eta_{\text{eff}}\prod_{\varsigma=y,z}r_k^\varsigma,\quad r_k^\zeta=\left(\frac{|\Delta_k^+ j^\zeta-\Delta_k^- j^\zeta|+\varepsilon^2}{|\Delta_k^+ j^\zeta|+|\Delta_k^- j^\zeta|+\varepsilon}\right)^m, \tag{21}$$

$$\Delta_k^+ j^\zeta=j_{k+1}^\zeta-j_k^\zeta,\quad \Delta_k^- j^\zeta=j_k^\zeta-j_{k-1}^\zeta. \tag{22}$$

In Eq. (21) η_{eff} is the maximum value of artificial resistivity (considered to be a constant in this study), $m \geqslant 1$ is an arbitrary exponent, which makes the resistivity decay away from regions with spurious oscillations (we use $m=2$), and $\varepsilon \ll 1$ is a small number of order the precision round-off error that guarantees $r_k^\zeta \approx 0$ in regions where $\Delta_k^+ j^\zeta \approx \Delta_k^- j^\zeta \approx 0$. Note that in general, $r_k^\zeta \approx 0$, if the local solution is smooth ($\Delta_k^+ j^\zeta \approx \Delta_k^- j^\zeta$) and $r_k^\zeta \approx 1$, if a local maximum or minimum develops in either component of the total current ($\Delta_k^+ j^\zeta \approx -\Delta_k^- j^\zeta$). A suitable alternative expression to Eq. (21) could also be obtained by taking the arithmetic average of factors, r_k^ζ.

Given the mesh-defined ion moments, n_{ik}, $\mathbf{j}_{ik}$, Eqs. (1), (11), (13)–(18), (20)–(22) compose a complete 1-D discrete hybrid approximation. It should be noted that we make use of the vector-potential **A** only for the purpose of calculating the magnetic field **B** in Eq. (13). Therefore, our model does not require the usual calibration condition, $\operatorname{div}\mathbf{A}=0$. As a result, it is extendable to multiple spatial dimensions, unlike some special **A**-based formulations [15,16].

3.3.2. PIC push

The ion macro-particles are updated in time (“pushed”) with Eq. (5). For particle p, its new velocity $\tilde{\mathbf{v}}_{\text{p}} \equiv \mathbf{v}_{\text{p}}(t_{\text{p}}+\Delta t_{\text{p}})$ can be easily expressed in terms of its old velocity $\mathbf{v}_{\text{p}} \equiv \mathbf{v}_{\text{p}}(t_{\text{p}})$ and the field $\{\mathbf{E}'_{\text{p}}, \mathbf{B}_{\text{p}}^{\text{tot}}\}$ evaluated at its position x_{p} by inverting the following discrete equation [1]:

$$\frac{\tilde{\mathbf{v}}_\mathrm{p}-\mathbf{v}_\mathrm{p}}{\Delta t_\mathrm{p}}=\frac{q_\mathrm{i}}{m_\mathrm{i}}\left(\mathbf{E}'_\mathrm{p}+\frac{\mathbf{v}_\mathrm{p}+\tilde{\mathbf{v}}_\mathrm{p}}{2c}\times\mathbf{B}^{\mathrm{tot}}_\mathrm{p}\right). \tag{23}$$

The particle positions are advanced using estimates of their time-centered velocities, $\bar{v}^x_\mathrm{p}$:

$$\tilde{x}_\mathrm{p}=x_\mathrm{p}+\Delta t_\mathrm{p}\bar{v}^x_\mathrm{p},\quad \bar{v}^x_\mathrm{p}=v^x_\mathrm{p}+\frac{a^x_\mathrm{p}\Delta t_\mathrm{p}}{2}, \tag{24}$$

where a^x_p is a predicted particle acceleration in the x-direction:

$$a^x_\mathrm{p}=\frac{q_\mathrm{i}}{m_\mathrm{i}}\left(\mathbf{E}'_\mathrm{p}+\frac{\mathbf{v}_\mathrm{p}\times\mathbf{B}^{\mathrm{tot}}_\mathrm{p}}{c}\right)^x. \tag{25}$$

Note that in Eqs. (23)–(25) Δt_p is the individual time increment for particle p. In DES, Δt_p may be dynamically adjusted in accordance with a particle's trajectory and resolution requirements. In this paper, Δt_p is selected based on a stipulation that a particle displacement in a single update (push) cannot exceed a given fraction of the cell size, $0<\sigma_{\mathrm{PIC}}\leqslant 1$ (note that in principle, σ_{PIC} can be defined as a function of cell position):

$$\tilde{x}_\mathrm{p}=x_\mathrm{p}\pm\sigma_{\mathrm{PIC}}\Delta x. \tag{26}$$

Combining Eqs. (24) and (26) results in a quadratic equation for Δt_p:

$$\left(v^x_\mathrm{p}+\frac{a^x_\mathrm{p}\Delta t_\mathrm{p}}{2}\right)\Delta t_\mathrm{p}=\pm\sigma_{\mathrm{PIC}}\Delta x, \tag{27}$$

which is easily solved for the minimum positive solution, Δt^E_p. Note that neglecting a^x_p both in Eqs. (24) and (27) would result in a faster but less accurate scheme. In addition, as mentioned above, to guarantee an adequate numerical accuracy of Eq. (23), we limit the particle time increment Δt_p to a fraction of the inverse *local* ion gyrofrequency, $\Omega^{-1}_{\mathrm{ci}}(x)$:

$$\Delta t_\mathrm{p}=\min(\Delta t^E_\mathrm{p},\lambda_{\mathrm{PIC}}/\Omega_{\mathrm{ci}}(x)). \tag{28}$$

The previous 1-D PIC-DES models [7,16] required that macro-particles switch cells in each update. They employed analytical PIC pushers to compute particle cell exit times. This algorithm is difficult to implement in higher dimensions. In addition, it is critically dependent on the NGP scheme for gather–scatter operations. The PIC push algorithm described in this paper (Eqs. (23)–(28)) is extendable to multiple spatial dimensions and arbitrary interpolation schemes.

3.4. Event-driven time integration

The comparison of algorithmic flows in typical event-driven and time-driven (time-stepping) simulations has been illustrated elsewhere [7,8]. At start-up ($t_{\mathrm{clock}}=0$), all field and particle states in the hybrid DES model are properly initialized and scheduled for execution in accordance with their estimated process times (see Sections 4.4 and 5.3). As mentioned above, the DES loop proceeds by continuously processing the earliest event (regardless of its type) stored in the dynamically evolving global event queue (Fig. 1), until the global clock time reaches the specified simulation finish time. The processing of events associated with local PIC-states and F-states is carried out asynchronously. This is done in a self-adaptive (recurrent) fashion via three event processing phases (Figs. 1 and 2) defined for each type of event (see Sections 4 and 5). Note that self-adaptation of the hybrid DES model manifests itself in modifying local particle and field states (and possibly pending events associated with them) upon processing each event stored in the global event queue. The accuracy of DES is controlled through limiting local incremental changes both to the particle sources (current and number densities) and the magnetic field. This supersedes the conventional time-stepping hybrid paradigm, where synchronous particle and field computations are usually time centered with respect to each other in order to *minimize* temporal discretization errors. However, time centering does not *limit* the magnitude of residual error terms (of higher order in Δt), which may become large due to spurious (unphysical) fluctuations in stochastic and discontinuous solutions. On the contrary, DES mitigates the effect of such fluctuations by limiting one-time changes to simulation variables and following causality rules. As a

result, DES automatically increases update rates in regions where the solution exhibits fast variation. For simplicity, in Eqs. (14), (23)–(25) we use forward-time integration for advancing the field and particle quantities. This assumes that all physical micro-states evolve at predetermined rates approximated to be *piecewise constant* in time. These rates are adaptively modified via event-driven or synchronization updates when changes to physical quantities are found to exceed specified threshold values.

4. Event-driven field solve

Asynchronous field advance is performed by applying a variation of the flux-conserving algorithm described in [8]. For these purposes, we augment each state, $\hat{F}_k$ by a number of "internal" DES variables, namely the field "flux capacitor", $\delta\mathbf{A}_k$, the ion current and charge density "register" variables, $\delta\mathbf{j}_{ik}$ and $\delta\rho_{ik}$, the field "target increment", $\Delta\mathbf{A}_k^{\mathrm{tr}}$, and the F-state update time, t_k. The field flux capacitor $\delta\mathbf{A}_k$ is one of the most essential elements of asynchronous DES integration [8]. It represents the net change to $\mathbf{A}_k$ due to inter-state ($\hat{F} \to \hat{F}$ and $\hat{P} \to \hat{F}$) synchronization updates occurred since last invocation of the process function of $\hat{F}_k$. The target increment $\Delta\mathbf{A}_k^{\mathrm{tr}}$ is a threshold quantity used to predict the next event for $\hat{F}_k$ and, if necessary, pre-empt its execution. The variables $\delta\mathbf{j}_{ik}$ and $\delta\rho_{ik}$ store the net increments to the ion current ($\mathbf{j}_{ik}$) and charge ($q_i n_{ik}$) densities accumulated since last evaluation of Ohm's law (Eq. (15)). The local time counter t_k "remembers" the most recent update time of state $\hat{F}_k$ caused by its processing or synchronization with other states. Let t_e be the timestamp of event e corresponding to state $\hat{F}_k$, being currently processed. The DES field solve can be described as follows (also see Fig. 1).

4.1. F-state processing

1. Let $t_{\mathrm{clock}} = t_e$, $\Delta t_k = t_{\mathrm{clock}} - t_k$. Integrate $\hat{F}_k$ in time to obtain $\mathbf{A}_k(t_{\mathrm{clock}})$, using Eq. (14).
2. Let $\delta\mathbf{A}_k = 0$, $\delta\mathbf{j}_{ik} = 0$, $\delta\rho_{ik} = 0$.
3. Call the field flux ($\hat{F} \to \hat{F}$) synchronization procedure for all neighboring states, $\hat{F}_{k\pm1}$ (Section 4.2).
4. Call the field-PIC ($\hat{F} \to \hat{P}$) synchronization procedure for state $\hat{P}_k$ (Section 5.2).
5. Schedule a new event for state $\hat{F}_k$ (Section 4.4).

4.2. Field flux synchronization

1. If $\hat{F}_s$ (the F-state being synchronized) is a boundary state, then apply an appropriate boundary condition, execute step 5 of this procedure and return.
2. Let $\Delta\mathbf{A} \equiv -c\mathbf{E}_s(t_{\mathrm{clock}} - t_s)$. Update the flux capacitor $\delta\mathbf{A}_s$ (Eq. (14)): $\delta\mathbf{A}_s(t_{\mathrm{clock}}) = \delta\mathbf{A}_s(t_s) + \Delta\mathbf{A}$.
3. Update the local vector potential $\mathbf{A}_s$ (Eq. (14)): $\mathbf{A}_s(t_{\mathrm{clock}}) = \mathbf{A}_s(t_s) + \Delta\mathbf{A}$. Let $t_s = t_{\mathrm{clock}}$.
4. If $|\delta\mathbf{A}_s| \geqslant |\Delta\mathbf{A}_s^{\mathrm{tr}}|$ then retract the pending event (corresponding to $\hat{F}_s$), execute steps 2–5 of the process function for $\hat{F}_s$ (Section 4.1, with k being replaced by s) and return.
5. Update the face-centered magnetic field components, $B^y_{(s+k)/2}$, $B^z_{(s+k)/2}$ (k denotes the cell that initiated the synchronization call) via Eq. (13) using the latest approximations of the vector-potential variables $\mathbf{A}_k(t_{\mathrm{clock}})$, $\mathbf{A}_s(t_{\mathrm{clock}})$.
6. Update the electric field $\mathbf{E}_s(t_{\mathrm{clock}})$ via Ohm's law (15) using the latest approximations of all physical quantities. Let $\delta\mathbf{j}_{is} = 0$, $\delta\rho_{is} = 0$.

4.3. PIC-field synchronization

This procedure is called from within the PIC-state process function (Section 5.1). Its purpose is to prevent the local ion current and density fluctuations, $|\delta\mathbf{j}_{is}|$ and $|\delta\rho_{is}|$ from exceeding their given maximum values ($\Delta j^{\mathrm{tr}}_{\mathrm{PIC}}$ and $\Delta\rho^{\mathrm{tr}}_{\mathrm{PIC}}$, respectively) during the time period between successive updates of $\hat{F}_s$. The $\hat{P} \to \hat{F}$ synchronization procedure for state $\hat{F}_s$ can be summarized as follows.

1. If $|\delta \mathbf{j}_{is}| < \Delta j^{tr}_{PIC}$ and $|\delta \rho_{is}| < \Delta \rho^{tr}_{PIC}$, then return.
2. Let $\Delta \mathbf{A} \equiv -c\mathbf{E}_s(t_{clock} - t_s)$. Update the flux capacitor $\delta \mathbf{A}_s$ (Eq. (14)): $\delta \mathbf{A}_s(t_{clock}) = \delta \mathbf{A}_s(t_s) + \Delta \mathbf{A}$.
3. Update the vector-potential $\mathbf{A}_s$ (Eq. (14)): $\mathbf{A}_s(t_{clock}) = \mathbf{A}_s(t_s) + \Delta \mathbf{A}$. Let $t_s = t_{clock}$.
4. If $|\delta \mathbf{A}_s| \geqslant |\Delta \mathbf{A}^{tr}_s|$, then retract the corresponding pending event, execute steps 2–5 of the process function for $\hat{F}_s$ (Section 4.1, with k being replaced by s) and return.
5. Update the electric field $\mathbf{E}_s(t_{clock})$ via Ohm's law (15) using the latest approximations of all physical quantities. Let $\delta \mathbf{j}_{is} = 0$, $\delta \rho_{is} = 0$.

4.4. F-state event scheduling

1. Update the electric field $\mathbf{E}_k(t_{clock})$ via Ohm's law (15) using the latest approximations of all physical quantities.
2. Find the maximum rate-of-change of the magnetic field, $(dB/dt)^{max}_k = \max_{k\pm1/2}|(\nabla \times \mathbf{E})^{\zeta=y,z}(t_{clock})|$.
3. Compute the time delay $\Delta t_k = \Delta B^{tr}/(dB/dt)^{max}_k$ (ΔB^{tr} is the given threshold increment for the magnetic field) and estimate the target value $\Delta \mathbf{A}^{tr}_k = \Delta t_k c\mathbf{E}_k$.
4. Schedule a new event for state $\hat{F}_k$ at the new process time, $\tilde{t}_e = t_{clock} + \Delta t_k$.

Note that Step 5 in Section 4.2 may result in a redundant computation of the electric field if an F-state being synchronized was already processed at time $t = t_{clock}$ (i.e., it initiated the original synchronization sequence). This is due to the fact that the electric field is also computed in Step 1 of this section (following the completion of all synchronization calls initiated by this F-state). This redundancy can be avoided by adding straightforward logic. However, in our current practice it does not seem to significantly affect CPU timings, as the number of events being retracted and reactivated via synchronization updates is usually small compared to the total number of events being executed at initially scheduled process times.

5. Event-driven PIC push

In addition to the local priority queue of particle wrappers, each PIC-state ($\hat{P}_k$) is characterized by two "wake-up" phase variables: $\delta x_k = \int v^x_k \, d\tau$ and $\delta \phi_k = \int \Omega_{ci}(x_k)\, d\tau - \phi_{k0}$, where $v^x_k = (q_i/m_i)\int E^x_k \, d\tau$, $\phi_{k0} = \tau \Omega_{ci}(x, \tau = 0)$ and τ is the simulation time elapsed since last call to the field-PIC synchronization routine (see Section 5.2). These variables keep track of the effective displacements of particles in space and phase angle due to their motion in the electric field and rotation in the magnetic field, respectively. The field-PIC synchronization occurs when either $|\delta x_k| \geqslant \sigma^w_{PIC}\Delta x$ or $|\delta \phi_k| \geqslant \lambda^w_{PIC}$ ($0 < \sigma^w_{PIC} < \sigma_{PIC}$ and $0 < \lambda^w_{PIC} < \lambda_{PIC}$ are control parameters). This mechanism guarantees that particle move times are properly recalculated in accordance with dynamic changes in the local electromagnetic field. Let t_e be the timestamp of event e, corresponding to state $\hat{P}_k$, being currently processed. The event-driven PIC push algorithm proceeds as follows.

5.1. PIC-state process function

1. Let $t_{clock} = t_e$. Remove the top particle wrapper from the local PIC priority queue and identify the corresponding particle p (with the previous push time t_p) to advance.
2. Subtract the particle charge q_p and current $\mathbf{J}_p = q_p\mathbf{v}_p$ from the net ion charge Q_k and current $\mathbf{J}_{ik}$ in cell k.
3. Advance the particle position (Eq. (24)) and velocity (Eq. (23)) with the time increment $\Delta t_p = t_{clock} - t_p$. If the particle has crossed the domain boundary, then apply an appropriate boundary condition. Determine the particle's destination cell, d. If the particle has been absorbed, then proceed to step 6.
4. Add the particle charge q_p and updated current $\mathbf{J}_p = q_p\tilde{\mathbf{v}}_p$ to the net ion charge Q_d and current $\mathbf{J}_{id}$ in cell d.
5. Compute the next particle push time, $\tilde{t}_p = t_p + \Delta t_p$ (Eq. (27)) using the latest field approximations and add a new particle wrapper with this time stamp to the priority queue in cell d.
6. Reschedule state $\hat{P}_k$ and, if needed, state $\hat{P}_{d\neq k}$ (Section 5.3).
7. Call the PIC-field synchronization procedure (Section 4.3) for state $\hat{F}_k$ and non-ghost state $\hat{F}_{d\neq k}$.
8. If cell k is the injection cell, then inject new particles (if necessary).

5.2. Field-PIC synchronization

1. Update δx_s and $\delta\phi_s$ in $\hat{P}_s$ (the PIC-state being synchronized). If $|\delta x_s| < \sigma^w_{\mathrm{PIC}}\Delta x$ and $|\delta\phi_s| < \lambda^w_{\mathrm{PIC}}$, then return.
2. Advance all particles in cell s to the current simulation time t_{clock} by applying steps 2–5 of Section 5.1 (with k being replaced by s) for $\hat{P}_s$.
3. If needed, reschedule updated PIC-states, $\hat{P}_d$ (Section 5.3).
4. Call the PIC-field synchronization procedure (Section 4.3) for non-ghost states $\hat{F}_d$.

5.3. PIC-state scheduling

1. If $\hat{P}_k$ is empty, then return.
2. Schedule an event for $\hat{P}_k$ with a time stamp equal to the estimated move time of the top particle in the local priority queue.

6. Field and PIC induced cell activation

The event-driven field and PIC algorithms described in Section 5 provide a powerful paradigm for building flexible simulation models of multi-scale (spatially inhomogeneous) plasma systems. By default, they assume that all system states are properly activated (initialized and scheduled) at the simulation start-up time. In many problems of interest, however, certain parts of the system may remain in initial states for long periods of time. For example, the upstream plasma in hybrid shock simulations [17–19] may be assumed to be unperturbed as long as plasma oscillations or reflected particles do not enter that region. If all these cells were activated at start-up, the simulation would spend unnecessary CPU time computing field noise oscillations and free-streaming particle motion. To avoid this situation, we implemented an ad hoc cell "activation" mechanism that keeps track of the moving "active" region boundary. In particular, we assume that cell activation may be "field" or "particle" induced. The field-induced activation of an inactive cell takes place when the absolute value of the local magnetic field in the "active boundary" cell deviates from the equilibrium value by a given threshold value, ΔB_a. The particle-induced activation occurs when any particle enters an inactive cell [7]. In either case, the previously "dormant" cell sets its internal field and PIC time counters to the current clock time and schedules its field and particle states for execution. The "active boundary" cell is also assumed to serve as the PIC injection cell.

7. DES algorithm validation

A number of time-stepping algorithms have been developed for the numerical solution of the hybrid plasma equations [4,5]. Given the fact that different algorithms are known to lead to certain differences in simulation results, we have carried out a detailed comparison of our DES model with two different time-stepping (time-driven) schemes. The first time-driven simulation (TDS) code has an implicit field solve [15] and the second one is based on a modified predictor–corrector scheme [18,19]. Both TDS codes employ linear gather/scatter particle interpolation. In case of discontinuities, plasma dynamics critically depend on the ratio of ion to electron plasma beta, $\beta_{\mathrm{i}}/\beta_{\mathrm{e}}$, the Alfvén Mach number of the shock, M_{A}, and the shock normal angle (angle formed by the incident flow velocity with respect to the ambient magnetic field), θ_{BN} [5]. Accordingly, in our validation studies we consider three different physical regimes, so that the combination of electromagnetic, thermal and geometry effects produces three well-known types of plasma discontinuities: the low-Mach-number fast shock (LFS) [5,17], the weak, intermediate shock (IS) [20,21], and the rotational discontinuity (RD) [22,23]. These problems span a broad range of nonlinear wave–particle interactions and can be considered as stringent validation tests of DES methodology. Table 1 summarizes the important simulation parameters for these runs. We have also made additional runs with different combinations of time-step size and resistivity values and included some of those results in Appendix A.2. The fourth case listed in Table 1 (HFS) corresponds to our simulations of high-Mach-number quasi-parallel shock-driven turbulence. It is separately discussed in Section 8.

Table 1
Summary of simulation run parameters

Run	M_A	θ_{BN}	β_i	β_e	Δx	x_{max}	t_{max}	$N_{PIC}/cell$
LFS	2.0	30°	0.1	0.1	0.2	400	200	100
RD	0.707	45°	0.06	0.4	0.2	400	200	100
IS	1.05	20°	0.06	0.4	0.2	400	200	100
HFS	6.0	20°	0.5	0.5	0.5	1000	300	1000

The mesh size Δx and the domain length x_{max} are normalized to the upstream ion inertial length, $\bar{\delta}_i = c/\bar{\omega}_{pi}$. The run time is normalized to the inverse upstream ion cyclotron frequency, Ω^{-1}.

In all simulations we consider proton plasmas with the same ion plasma to gyrofrequency ratio, $\bar{\omega}_{pi}/\bar{\Omega}_{ci} = 2000$ ($\bar{\omega}_{pi} \equiv \omega_{pi}(n_0)$, $\bar{\Omega}_{ci} \equiv \Omega_{ci}(B_0) \equiv \Omega$, n_0 is the upstream plasma equilibrium number density, B_0 is the magnitude of the ambient magnetic field). Unless stated otherwise, TDS runs are typically performed with

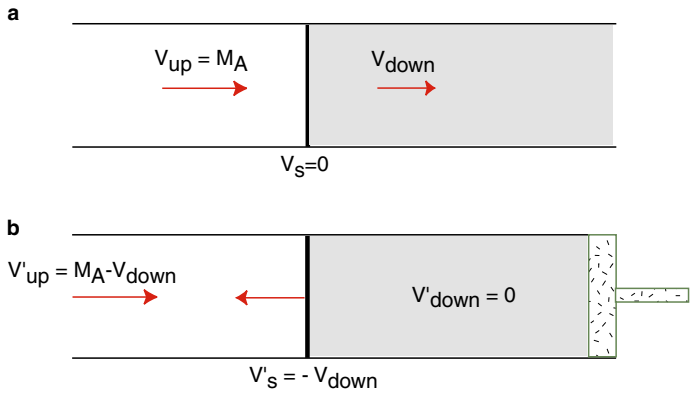


Fig. 3. The piston method (all velocities are normalized to the upstream Alfvén velocity): (a) the shock rest frame of reference, where the discontinuity propagation velocity is zero ($v_s = 0$) and the Rankine–Hugoniot relationship is obtained, (b) the piston (simulation) frame of reference, where the downstream flow velocity is zero ($v'_{down} = 0$) and ions are assumed to be reflected off the stationary piston (right) boundary.

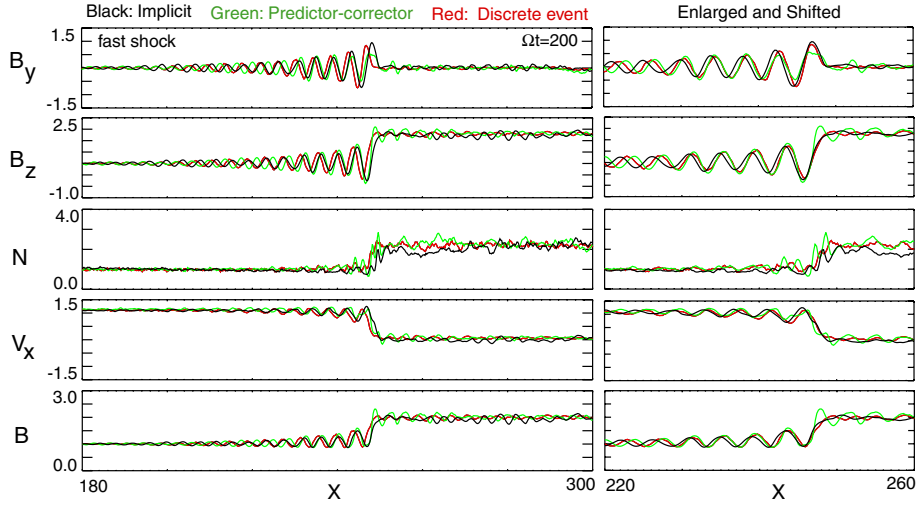


Fig. 4. Comparison of low-Mach-number fast shock (LFS) profiles obtained with the time-stepping and DES codes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the normalized resistivity, $\tilde{\eta}_{\mathrm{TDS}} = \eta\bar{\omega}_{\mathrm{pi}}/4\pi = 10^{-5}$ and the time-step size, $\bar{\Omega}_{\mathrm{ci}}\Delta t = 5 \times 10^{-3}$. The DES runs (except the HFS case) were conducted with $\lambda_{\mathrm{PIC}} = 0.02$, $\tilde{\eta}_{\mathrm{eff}} = 2 \times 10^{-5}$, $\Delta B^{\mathrm{tr}}/B_0 = 10^{-3}$. For simplicity, the DES target displacement fraction, σ_{PIC} was calculated as a function of λ_{PIC} (see Section 3.3.2):

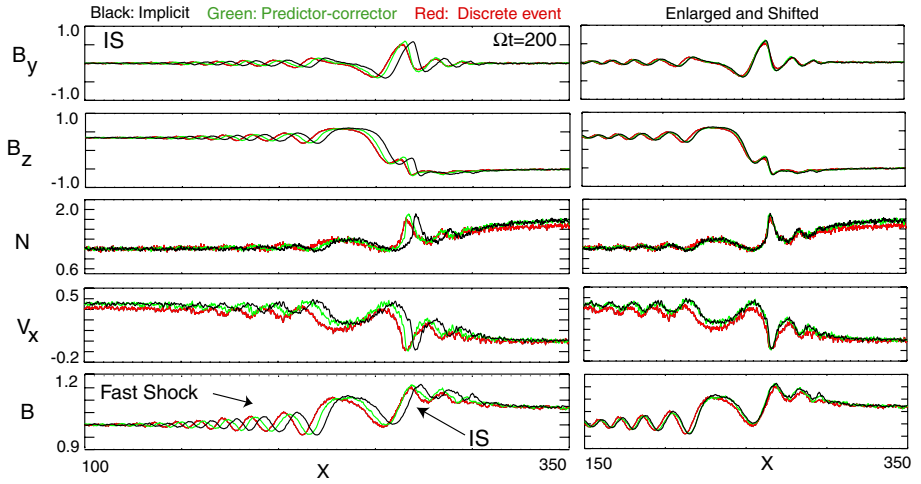


Fig. 5. Comparison of intermediate shock (IS) profiles obtained with the time-stepping and DES codes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

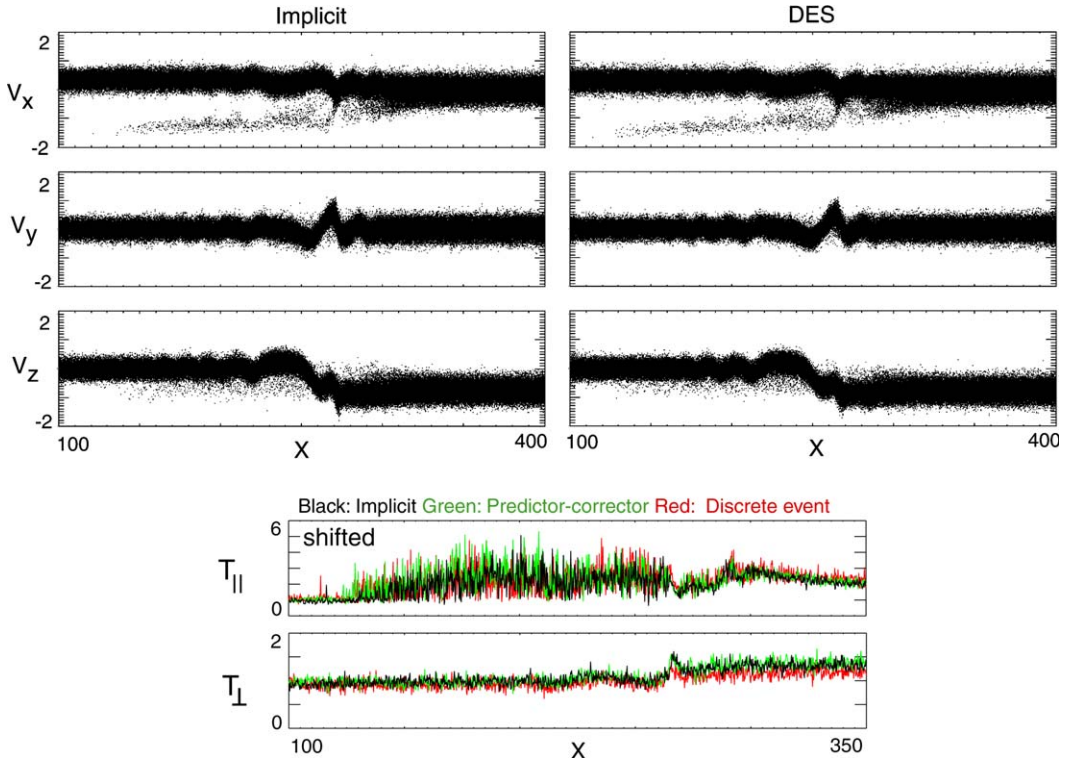


Fig. 6. Comparison of IS phase space diagrams (the top panel) and temperature profiles (the bottom panel) obtained at $\Omega t = 200$ in the TDS and DES runs. Particle velocities are normalized by the upstream Alfvén velocity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\sigma_{\mathrm{PIC}} = \lambda_{\mathrm{PIC}} \frac{v_{\mathrm{PIC}}/\bar{\Omega}_{\mathrm{ci}}}{\Delta x},$$

where v_{PIC} is the characteristic PIC velocity, equal to either the injection (I
(RD). In addition, all $\hat{P} \to \hat{F}$ and $\hat{F} \to \hat{P}$ wake-up parameters were ch
appropriate characteristic quantities: $\sigma^{w}_{\mathrm{PIC}}/\sigma_{\mathrm{PIC}} = 0.05$, $\lambda^{w}_{\mathrm{PIC}}/\lambda_{\mathrm{PIC}}$
5×10^{-2}.

In all cases we form plasma discontinuities using the r
domain is initialized with a preexisting thermal plasma no
equal to the background Alfven velocity times the de
minus the downstream plasma velocity (as obtaine
are injected at the left boundary with the same
incoming plasma is reflected off the stationary

formation of a discontinuity which propagates to the left, away from the piston. Backstreaming particles are absorbed at the left boundary.

In the special case of rotational discontinuity (RD) [22,23], the upstream and downstream plasma velocities are equal, and there is no initial plasma flow in the simulation frame of reference. In this case, macro-particles exiting the computation domain through the left boundary are simply re-injected back with thermal velocities to maintain the constant plasma density in the upstream region. It should be noted that in all cases subtle differences in the particle initialization and injection schemes adopted in the three codes cause small variations in the solutions. In order to emphasize the quantitative closeness of simulation results from the different codes, we also find it useful to provide additional profiles obtained by applying appropriate (visually determined) shifting transforms in the x-direction.

7.1. Low-Mach-number fast shock (LFS)

Fig. 4 (left panels) shows an overlay of fast shock profiles observed in the event-driven and two time-stepping simulations. As expected, in this low-Mach-number case the upstream shock structure is composed of a well-defined whistler wavetrain. The DES code is seen to capture the structure of this shock very accurately. All three codes produce minor differences in the speed and amplitude of downstream oscillations. Shifting the solutions (right panels) proves that the wavetrain and shock transition regions in these simulations are almost identical. The small discrepancies in the shock speed were found to be due to the different amounts of effective (numerical plus artificially imposed) resistivity in the three different codes (the time-stepping codes use constant artificial resistivity). We have verified that increasing the resistivity in the predictor–corrector and implicit codes results in slower shock speeds (see Appendix A.2). Therefore, this case is characteristic of strongly dominant short-wavelength whistler effects, which are sensitive to resistivity. For comparison, neither the intermediate shock nor the rotational discontinuity simulation (considered below) shows as much dependency on resistivity. Further discussion of the role of artificial resistivity in different hybrid applications and their convergence studies is beyond the scope of this paper. Note that the simulation period is very long, and the fact that the DES code captures the shock structure and position so accurately is truly remarkable, especially given its NGP interpolation model and larger particle push time increments. Interestingly, replacing "fat" particles with the NGP scheme in the *time-stepping* codes was found to lead to numerical instabilities, which provides indirect evidence for superior stability properties of DES.

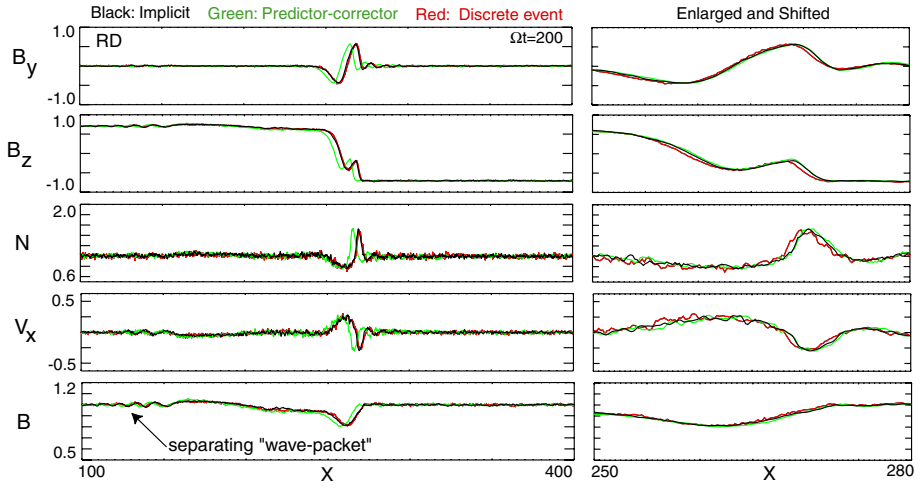


Fig. 8. Comparison of rotational discontinuity (RD) profiles obtained with the time-stepping and DES codes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

7.2. Intermediate sh

The physics of the
shock profiles obtaine
modify the fluid Rank
found to be rather inde
is strongly dominated by

The closeness of the D
phase space diagrams and
suppresses negative effects
plasma heating due to spur

Furthermore, Fig. 7 (prot
with which DES reproduces

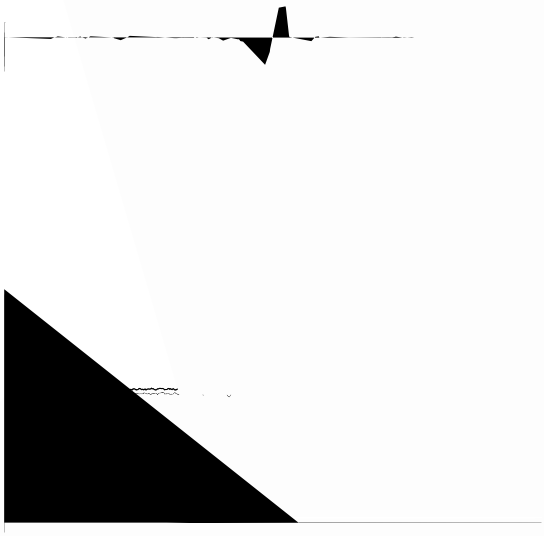


Fig. 9. Time-stacked plots of B_y obtained in the RD case with the DES and predictor–corrector codes.

7.3. Rotational discontinuity (RD)

Our final test problem deals with an isotropic rotational discontinuity generated by flipping around the z-component of the magnetic field at the right boundary. Observe (Fig. 8) that shortly after the RD structure is formed, a separate wave structure is generated in the upstream region. This wave packet eventually detaches itself from the RD.

In the downstream region, however, the RD launches a small amplitude wave train, which remains inseparable of the steady-state RD structure. Fig. 8 shows that DES captures the separating wave packet as well as the attached wave train very accurately. Indeed, aside from minor differences in the RD speed (Fig. 8, left panels), the solution profiles produced by the three codes are almost identical (Fig. 8, right panels). As in the previous case, Fig. 9 illustrates the temporal dynamics of the RD wave structure through a series of ''stack'' plots.

8. Simulation of high-Mach-number kinetic turbulence

The previous test problems have proved the ability of DES to efficiently capture the essential physics of weak-to-moderate plasma discontinuities. Now we turn our attention to a problem of great interest to heliospheric and magnetospheric physics, namely the self-consistent generation of plasma turbulence by strong fast magnetosonic shocks (run HFS in Table 1). This problem is known to be very difficult to simulate with traditional codes because of its multi-scale nature, i.e., a considerable variation in the spatial and temporal scales of self-generated turbulence [5,17]. For this case we have conducted a number of DES runs with the following

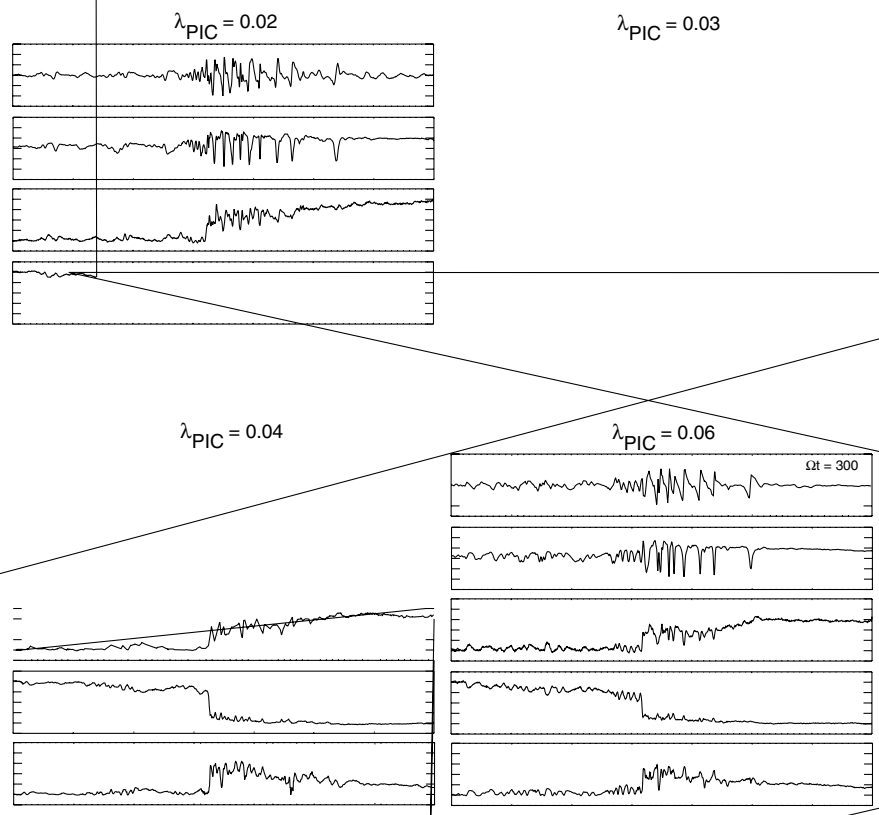
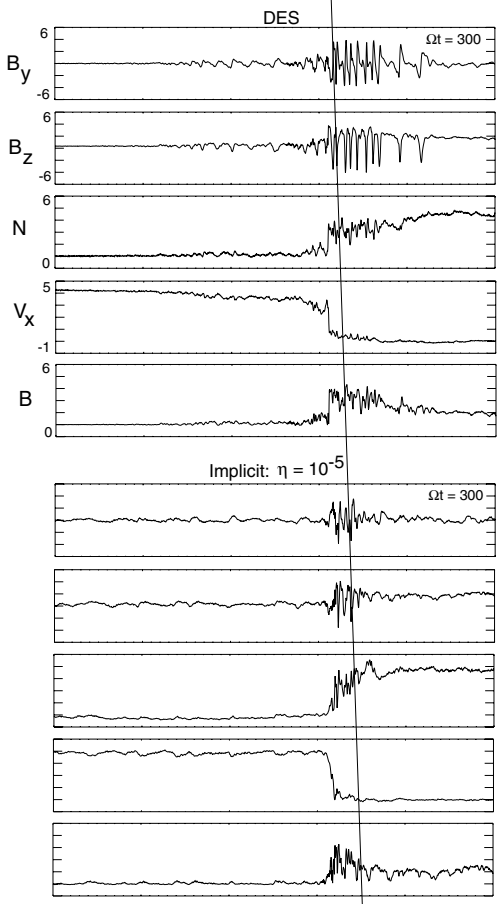


Fig. 10. High-Mach-number fast shock (HFS) turbulence profiles obtained in DES runs with different values of λ_{PIC} (and proportional values of η_{eff}). Note that in each run actual time increments used for local particle updates are adaptively selected in accordance with a given value of λ_{PIC}.

parameters: $\lambda_{\rm PIC} = 0.02$–0.06, $\tilde{\eta}_{\rm eff} = (5\text{–}15) \times 10^{-5}$, $\Delta B^{\rm tr}/B_0 = 2 \times 10^{-3}$. In order to suppress the generation of high-frequency oscillations, the magnitude of artificial resistivity in these runs was adjusted proportionally to the maximum PIC time increment allowed, $\tilde{\eta}_{\rm eff} \sim \sigma_{\rm PIC} \sim \lambda_{\rm PIC}$.

These runs are found to converge to physically close solutions (Fig. 10). Note that all DES runs are stable and produce quantitatively close shock profiles and speeds. The differences observed in these runs are found to



be comparable to statistical differences observed in runs conducted with the same input parameters but different random generator seeds. As in the time-stepping simulations, this is caused by the highly turbulent nature of high-Mach-number shock dynamics developing over long simulation periods.

Fig. 11 compares the DES solution profiles ($\lambda_{\mathrm{PIC}} = 0.03$, $\tilde{\eta}_{\mathrm{eff}} = 7 \times 10^{-5}$) with those obtained in the corresponding time-stepping simulations ($\bar{\Omega}_{\mathrm{ci}}\Delta t = 0.005$). Clearly, the event-driven simulation shows four well-resolved zones of shock-driven turbulence: (i) the low-frequency steepened oscillations observed far upstream (''shocklets''), (ii) the short-wavelength oscillations in the near upstream region (driven by the reflecting ions), (iii) the coherent shock transition region (where the upstream oscillations get compressed and amplified), and (iv) the long-wavelength (rarefaction) waves in the downstream region.

On the other hand, the TDS solutions are found to drastically differ from the DES solution as they achieve poorer resolution in all regions of the computation domain (with or without resistivity, see also Section A.2).

The differences between the DES and TDS profiles are further accentuated by the concomitant differences in phase space (Fig. 12).

Clearly, in the TDS runs, the coherent oscillations in the shock transition layer are poorly resolved and the upstream turbulence is either damped or transformed into incoherent noise (with zero resistivity). The abnormally rapid decay of coherent oscillations in the time-stepping simulations was noted previously [17]. Comparisons with the DES model indicate that this is caused by numerical errors, which lead to phase de-synchronization between particle and field dynamics. Furthermore, we have shown elsewhere [24] that increasing the temporal or spatial resolution in time-stepping simulations tends to make their results (including energetic ion spectra) evolve towards those observed in the DES model. Fig. 13 illustrates the formation and evolution of the high-Mach-number shock structure in the DES run with $\lambda_{\mathrm{PIC}} = 0.03$.

Fig. 14 shows two DES shock profiles computed at two different times. Also shown are the corresponding spatial distributions of instantaneous time increments for local field and particle updates (note that the particle time increments are cell-averaged). Based on this figure, we can make several important observations pertaining to the nature of the hybrid DES model. First, early in the simulation a large fraction of the computation

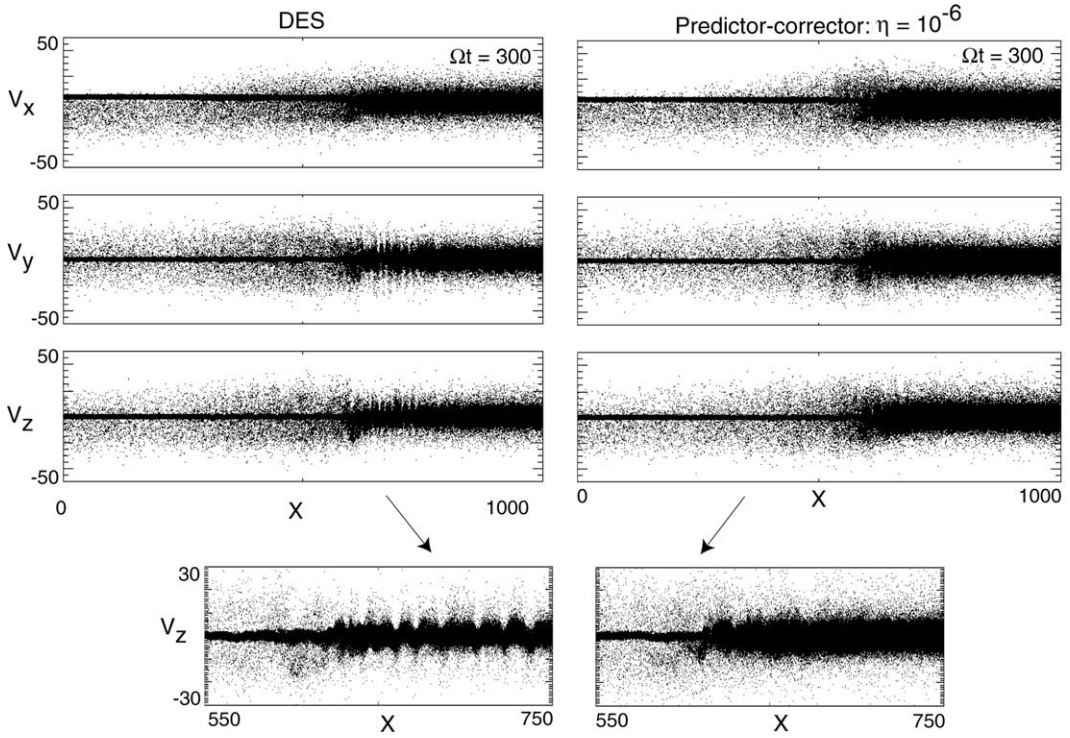


Fig. 12. Phase space diagrams obtained with the DES ($\lambda_{\mathrm{PIC}} = 0.03$) and predictor–corrector codes for the high-Mach-number shock turbulence case. Particle velocities are normalized by the upstream Alfvén velocity.

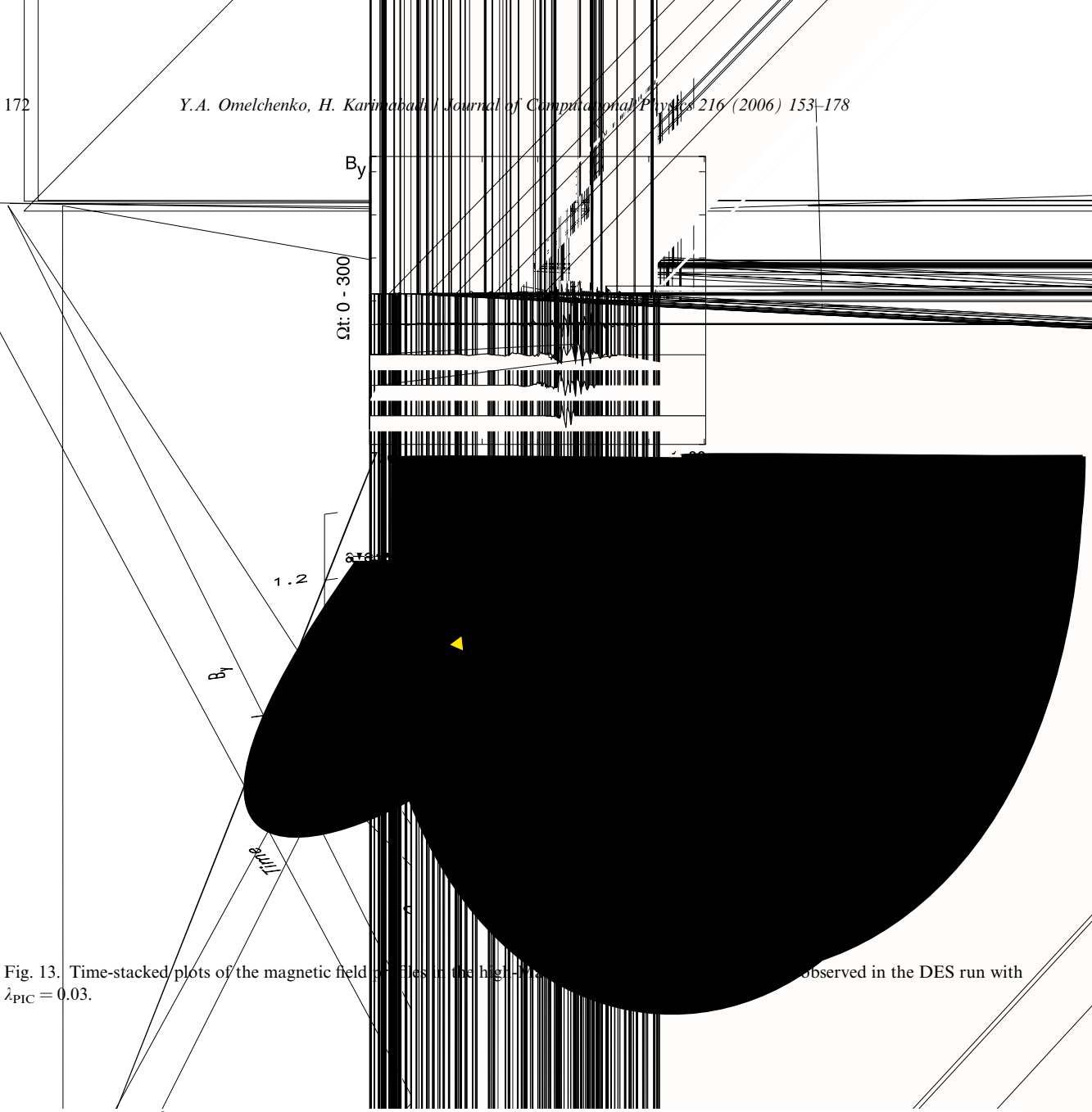


Fig. 13. Time-stacked plots of the magnetic field profiles in the high-Mach-number shock observed in the DES run with $\lambda_{PIC} = 0.03$.

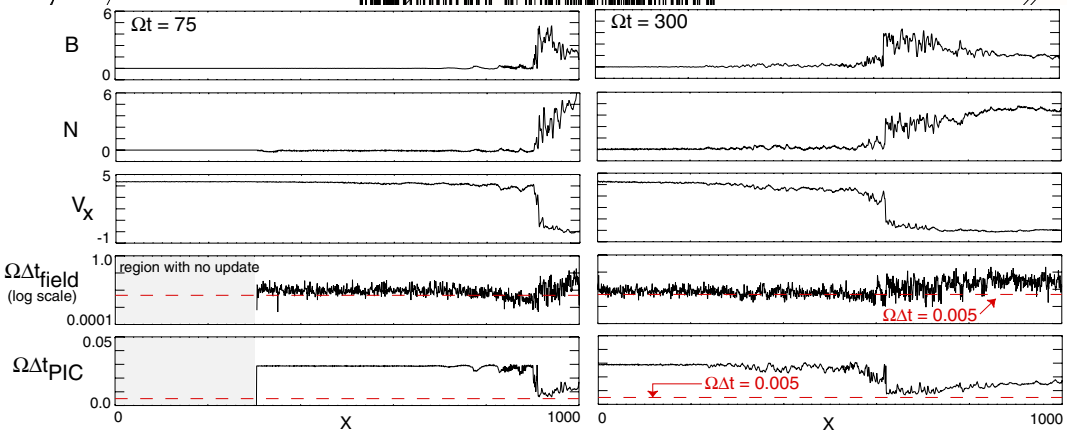


Fig. 14. High-Mach-number shock profiles and characteristic time increments observed in the DES run with $\lambda_{PIC} = 0.03$.

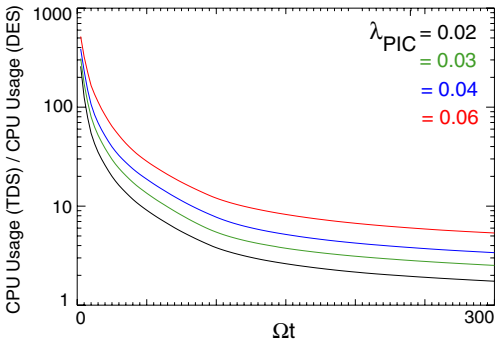


Fig. 15. Relative speed-ups of DES (compared to the predictor–corrector run), as functions of time for different values of λ_{PIC} in the HFS case.

domain remains computationally inactive as it takes some time for the backstreaming ions and whistler waves to reach the unperturbed plasma regions. DES naturally removes unnecessary computation in those regions. Second, the characteristic field update time increments vary by several orders of magnitude across the shock structure, being self-consistently adjusted to match the local wave frequencies.

One can also observe that the stability-based time-step size used in the TDS codes overresolves local frequencies in some regions and underresolves them in the near upstream zone, where the oscillations are generated. In higher spatial dimensions, the need to choose spatially uniform time-step sizes suitable for all regions makes multi-scale time-stepping simulations even more difficult. The characteristic particle update time increments naturally span a much smaller range in magnitude than those required for the field updates, being mostly dependent on the local magnitude of the magnetic field. Note, however, that a relatively few number of very energetic ions require time increments, which are much smaller than the depicted cell-averaged quantities. These time increments are automatically adapted to match individual velocities and accelerations of energetic particles, without affecting the slower update rates for the bulk ions. Aside from improving CPU efficiency, this enhances the DES model robustness by preventing particle push time increments from accidentally exceeding the PIC Courant limit (Eq. (10a)). In contrast, the TDS codes have to choose the global particle time-step size based on the fastest particle velocity in the simulation. This requirement severely degrades their performance for higher Mach numbers and finer spatial resolutions.

We would also like to emphasize that in addition to achieving superior accuracy in the HFS case, the DES code offers a significant advantage in CPU speed over its TDS counterparts even for the fairly homogeneous problems considered in this paper. In general, DES speed-ups can be infinite since actual numbers depend on both details of a given application (such as the range of temporal scales and presence of inactive regions) and desired accuracy (see Appendix A.1 for a realistic example of the Earth's magnetosphere). For all cases shown in the paper, the DES code remains stable for a broad range of input parameters and runs faster than its (stable) time-stepping counterparts despite its presently unoptimized form. For the high-Mach-number case, we find that even using a small time step of 0.005 in the time-stepping codes does not result in producing physically satisfactory solutions. Nevertheless, comparing CPU usage in the DES and TDS runs in this case (Fig. 15) is very instructive. It is evident that early in the simulation (when the shock has not yet propagated far from the piston) larger fractions of the computation domain remain inactive and the effective DES speed-ups are over a hundred. Less than half way through the run, all domain cells become activated by the backstreaming ions, which reduces the DES speed advantage. Nevertheless, at the end of the simulation, the DES runs still demonstrate speed-ups up to a factor of 5.

9. Conclusion

We have presented a self-adaptive hybrid-PIC model based on a newly developed discrete-event simulation methodology [7,8]. The traditional hybrid models [4] are based on time-stepping techniques and limited in

their abilities to effectively describe systems with strong inhomogeneities or varying time scales. On the contrary, the DES approach is inherently multi-scale. DES treats all spatially distributed system micro-states (particles, field components) as asynchronous entities with individual time increments, which are automatically selected to match local physical frequencies. Time synchronization among coupled micro-states is achieved through causality relationships, which guarantee timely updates of all variables of interest. In the future, we deem it possible to increase the accuracy of the current implementation of DES by applying the well-known Runge–Kutta approach. This will require introducing backward-time biased rules that will apply local flux-conserving corrections to simulation quantities based on their values found via initial forward-time integration.

We successfully validated the new hybrid-DES code against two different time-stepping codes on a number of realistic problems describing weak magnetosonic discontinuities. These tests clearly show that even over long simulation periods DES reproduces solutions *quantitatively* close to those obtained in the standard time-stepping simulations. We also applied this code to simulate the physics of high-Mach-number kinetic turbulence and showed that the self-adaptive DES model is more robust, more accurate and faster than its time-stepping counterparts. Note that with the introduction of a nonuniform applied magnetic field and more complex (multi-dimensional) plasma systems (such as the Earth's magnetosphere) the advantages of DES are expected to become much more pronounced due to the distinct ability of event-driven simulation to efficiently handle "stiff" and "idle" computations in highly inhomogeneous systems (see Appendix A.1).

The event-driven algorithm described in this paper is fully extendable to multiple dimensions and nonuniform meshes (currently we are developing a uni-dimensional infrastructure with adaptive logical mapping capabilities) and conducive to effective parallelization via several techniques [16,25], including a novel conservative (causality preserving) preemptive event processing (PEP) method, which enables synchronous processing of events with close timestamps. The parallel PEP technique and its application to hybrid event-driven simulations will be described elsewhere.

Finally, we would like to emphasize another distinction of DES from traditional time-stepping methodology. In the time-stepping simulations, numerical accuracy is controlled indirectly through a choice of characteristic global time-step sizes, which are often difficult to optimally determine a priori in many complex problems of interest. In DES, however, one directly deals with accuracy constraints through specifying a number of physical input parameters (threshold values), while letting the algorithm self-adaptively determine

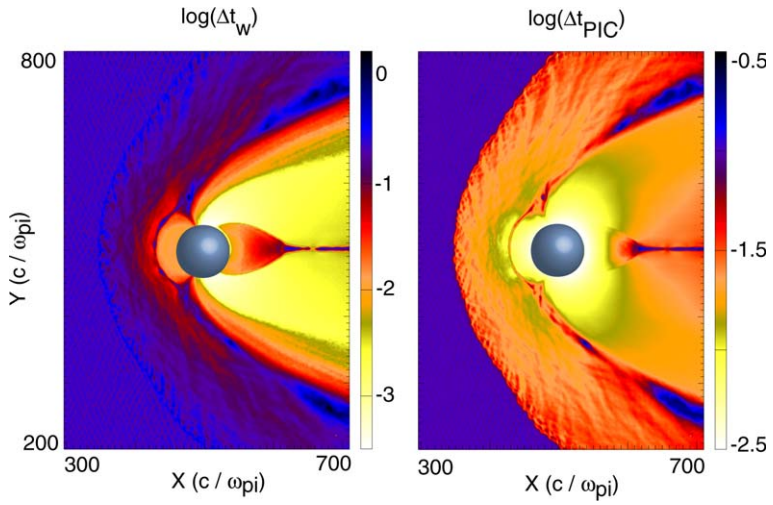
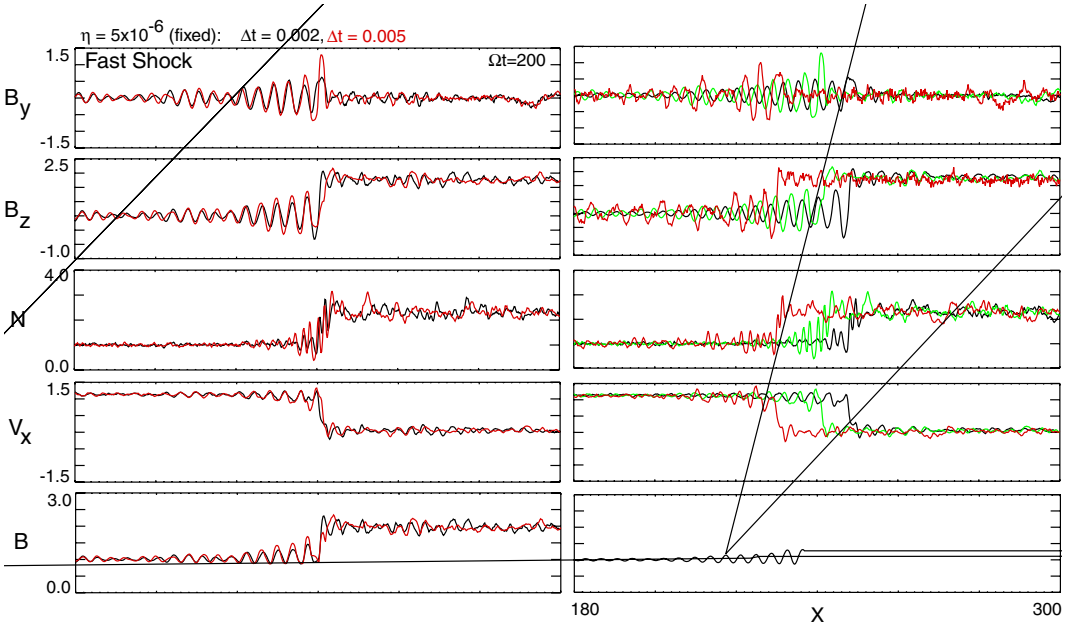


Fig. A.1. Global time-stepping hybrid simulation of the Earth's magnetosphere. The intensity log-plots show spatial distributions of characteristic time-step sizes normalized to the inverse upstream ion gyrofrequency. The left panel represents the whistler-based time-step size, Δt_w (Eq. (10b)). The right panel corresponds to the particle update time increment, Δt_{PIC} (Eq. (10a)). No computation is done within the blue circle, which represents the Earth. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

proper local time increments. Benchmarking the performance of event-driven hybrid simulations with respect to various control parameters requires additional methodical work and will be addressed elsewhere.

Acknowledgments

This work was supported by the National Science Information Technology Research (ITR) Grant No. 0539106 at SciberQuest, Inc. The authors gratefully acknowledge useful discussions with Richard Fujimoto, Jonathan Driscoll, and Dietmar Krauss-Varban.



Appendix A

A.1. Time-step size restrictions in the global magnetospheric hybrid simulations

The interaction of the solar wind with the Earth's magnetosphere leads to a complex set of boundaries and discontinuities, each possessing its own temporal and spatial scales. We use results from recent 2-D hybrid time-stepping simulation of the Earth's magnetosphere [26] to demonstrate some of the difficulties associated with simulations of such multi-scale problems within time-stepping methodology. The simulation is set up as follows. The solar wind plasma ($M_A = 8$, $\beta_i = 0.5$, $T_i = T_e$) is initialized with a uniform density distribution and continuously injected from the left boundary in the x-direction toward the Earth. The plasma is allowed to escape through the other boundaries. The uniform mesh cell size Δx is equal to one ion inertial length (computed with respect to the upstream density) in each direction, $\Delta x = \Delta y = \delta_i$. The applied magnetic field is obtained by superposing a uniform southward interplanetary magnetic field (aligned with the y-axis), and the magnetic field of the Earth's line dipole. Fig. A.1 shows the intensity plots of the CFL-restricted time increments (Eqs. (10a) and (10b)) at an early time in the simulation. The time increments clearly exhibit a highly inhomogeneous structure, spanning a wide range in magnitude. Such systems are difficult to simulate (espe-

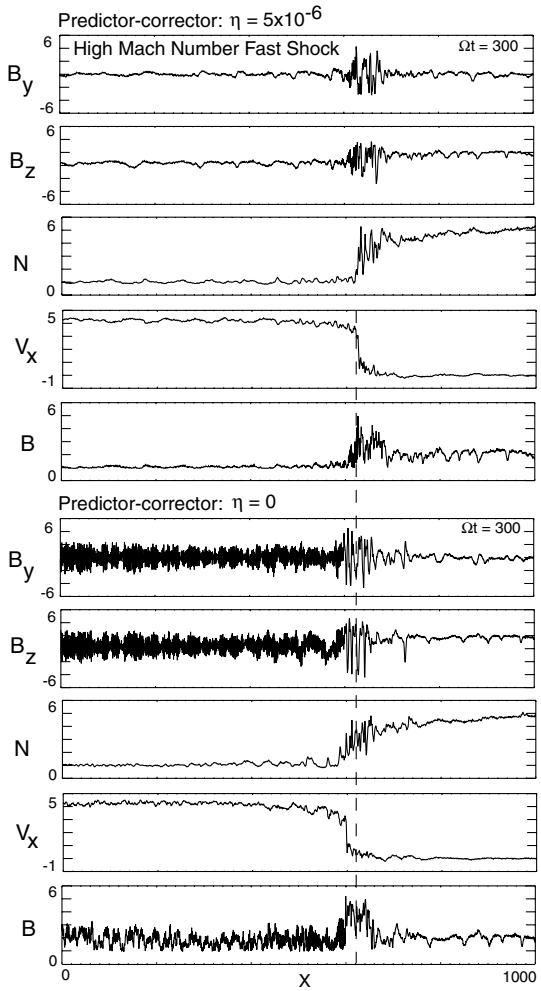


Fig. A.4. High-Mach-number (HFS) turbulence profiles obtained in the predictor–corrector TDS run with small and zero values of the resistivity η.

cially in 3-D), using traditional time-stepping codes. Indeed, the need to globally satisfy Eqs. (10a) and (10b) leads to an inefficient computation, unless regions with different time increments are updated based on their own time scales (i.e., asynchronously as in DES). In addition, these time increments can vary significantly during the course of the simulation as the magnetosphere forms and evolves towards its asymptotic state. Since the minimum time-step sizes are not known a priori, the simulation can easily become numerically unstable if the pre-chosen time steps are not small enough. DES, as outlined in this paper, is an ideal approach to such problems due to *self-adaptive* determination of local time increments. We are currently developing a multi-dimensional DES infrastructure for modeling the magnetosphere and other multi-scale systems.

A.2. Resistivity and timestep variation in the hybrid TDS codes

Fig. A.2 shows low-Mach-number (LFS) fast shock solutions obtained with the explicit predictor–corrector code for different values of the time-step size Δt and the applied (artificial) resistivity η. The LFS fast shock phase is shown to be affected by short-wavelength whistler noise, which is sensitive to the amount of applied resistivity. Therefore, variations in Δt and η lead to long-time shock profiles with somewhat different amplitudes and shock front positions. A thorough convergence study of these time-stepping simulations as a function of the mesh resolution, the number of particles per cell, the time-step size and the applied resistivity deserves special attention in the future.

To further corroborate the above point on the differences in the shock phases, Fig. A.3 provides intermediate shock (IS) profiles similarly obtained with the same predictor–corrector code by varying η and Δt. In this case (as well as in the RD problem), pressure-driven effects dominate over whistler-driven physics. As a result, the speeds and amplitudes of the discontinuities are found to be rather insensitive to the values of η and Δt.

Finally, Fig. A.4 demonstrates the effect of artificial resistivity on the HFS turbulence. In the absence of artificial resistivity, one clearly observes the generation of incoherent noise in the upstream region. These high-frequency oscillations are driven by noisy PIC fluctuations present in any hybrid code. If not damped through artificial smoothing, they quickly grow to large amplitudes and fill the whole upstream region. On the other hand, applying small levels of resistivity results in "cleaning" the upstream solution of unphysical oscillations. However, since in this study the time-stepping codes use the spatially uniform (constant) resistivity model, the resistive smoothing procedure also results in damping some profile features. Interestingly, this can be avoided (as shown by our preliminary studies to be published elsewhere) by applying the same form of artificial resistivity as used in the DES code (Eqs. (21) and (22)). It should also be noted that the resistive smoothing (frequency filtering) brings extra benefits to DES compared to time stepping, as it results in enabling larger time increments for local field computations.

References

[1] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulation, McGraw-Hill, NewYork, 1985.
[2] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, McGraw-Hill, New York, 1981.
[3] J.M. Dawson, Particle simulation of plasmas, Rev. Mod. Phys. 55 (1962) 403.
[4] D. Winske, L. Yin, N. Omidi, H. Karimabadi, K. Quest, Hybrid Simulation Codes: Past, Present and Future – A Tutorial, Lecture Notes in Physics, vol. 615, Springer, Berlin, 2003, p. 136.
[5] A.S. Lipatov, The Hybrid Multiscale Simulation Technology. An Introduction with Application to Astrophysical and Laboratory Plasmas, Springer, Berlin, Heidelberg, 2002.
[6] J. Brackbill, B. Cohen (Eds.), Multiple Time Scales, Academic Press, Orlando, 1985.
[7] H. Karimabadi, J. Driscoll, Y.A. Omelchenko, N. Omidi, A new asynchronous methodology for modeling of physical systems: breaking the curse of Courant condition, J. Comput. Phys. 205 (2) (2005) 755.
[8] Y.A. Omelchenko, H. Karimabadi, Self-adaptive solution of flux-conservative equations with sources, J. Comput. Phys. (in press), doi:10.1016/j.jcp.2005.12.008.
[9] J. Banks (Ed.), Handbook of Simulation, Wiley, New York, 1998.
[10] R.M. Fujimoto, Parallel and Distributed Simulation Systems, Wiley–Interscience, New York, 2000.
[11] P.L. Pritchett, Particle-in-cell simulations of magnetosphere electrodynamics, IEEE Trans. Plasma Sci. 28 (6) (2000) 1976.
[12] A. Mankofsky, R.N. Sudan, J. Denavit, Hybrid simulation of ion beams in background plasma, J. Comput. Phys. 70 (1987) 89.
[13] Y.A. Omelchenko, R.N. Sudan, A 3-D Darwin-EM hybrid code for ion ring studies, J. Comput. Phys. 133 (1997) 146.
[14] P.W. Rambo, Numerical heating in hybrid plasma simulations, J. Comput. Phys. 133 (1997) 173.

[15] D. Winske, N. Omidi, Hybrid codes: methods and applications, in: H. Matsumoto, Y. Omura (Eds.), Computer Space Plasma Physics: Simulation Techniques and Software, Terra Scientific Publishing Company, Tokyo, 1993, p. 103.
[16] H. Karimabadi, J. Driscoll, J. Dave, Y. Omelchenko, K. Perumalla, R. Fujimoto, N. Omidi, Parallel discrete event simulations of grid-based models: asynchronous electromagnetic hybrid code, in: Workshop on State-of-the-Art in Scientific Computing, Springer LNCS Proceedings, 2005, p. 580.
[17] K. Quest, Theory and simulations of collisionless parallel shocks, J. Geophys. Res. 93 (A9) (1988) 9649.
[18] H. Karimabadi, D. Krauss-Varban, J.D. Huba, H.X. Vu, On magnetic reconnection regimes and associated three-dimensional asymmetries: hybrid, Hall-less hybrid and Hall-MHD simulations, J. Geophys. Res. 109 (2004) A09205.
[19] D. Krauss-Varban, From theoretical foundation to invaluable research tool: modern hybrid simulations, in: Proceedings of ISSS-7, March, 2005, pp. 15–18.
[20] H. Karimabadi, D. Krauss-Varban, N. Omidi, Kinetic structure of intermediate shocks: implications for the magnetopause, J. Geophys. Res. 100 (A7) (1995) 11957.
[21] H. Karimabadi, Physics of intermediate shocks: a review, Adv. Space Res. 15 (8–9) (1995) 507.
[22] H. Karimabadi, D. Krauss-Varban, N. Omidi, On the stability of rotational discontinuities – one-dimensional and 2-dimensional hybrid simulations, Geophys. Res. Lett. 22 (21) (1995) 2989.
[23] D. Krauss-Varban, H. Karimabadi, N. Omidi, Kinetic structure of rotational discontinuities – implications for the magnetopause, J. Geophys. Res. 100 (A7) (1995) 11981.
[24] H. Karimabadi, Y. Omelchenko, J. Driscoll, R. Fujimoto, K. Perumalla, D. Krauss-Varban, A new simulation technique for study of collisionless shocks: self-adaptive simulations, in: Proceedings of 4th IGPP Annual International Astrophysics Conference, The Physics of Collisionless Shocks, Palm Springs, CA, USA, February 26–March 3, 2005.
[25] Y. Tang, K. Perumalla, R. Fujimoto, H. Karimabadi, J. Driscoll, Y. Omelchenko, Parallel discrete event simulations of physical systems using reverse computation, in: ACM/IEEE Workshop on Principles of Advanced and Distributed Simulation, vol. 19, 2005, p. 26.
[26] H. Karimabadi, N. Omidi, H.X. Vu, Formation of Plasmoids and Magnetic Flux Ropes during Southward IMF, American Geophysical Union, Fall Meeting 2004, abstract #SM11A-1171.