

*Communications in
Applied
Mathematics and
Computational
Science*

**A NUMERICAL STUDY
OF LANDAU DAMPING WITH PETSC-PIC**

DANIEL S. FINN, MATTHEW G. KNEPLEY,
JOSEPH V. PUSZTAY AND MARK F. ADAMS

vol. 18 no. 1 2023

A NUMERICAL STUDY OF LANDAU DAMPING WITH PETSC-PIC

DANIEL S. FINN, MATTHEW G. KNEPLEY,
JOSEPH V. PUSZTAY AND MARK F. ADAMS

We present a study of the standard plasma physics test, Landau damping, using the particle-in-cell (PIC) algorithm. The Landau damping phenomenon consists of the damping of small oscillations in plasmas without collisions. In the PIC method, a hybrid discretization is constructed with a grid of finitely supported basis functions to represent the electric, magnetic and/or gravitational fields, and a distribution of delta functions to represent the particle field. Approximations to the dispersion relation are found to be inadequate in accurately calculating values for the electric field frequency and damping rate when parameters of the physical system, such as the plasma frequency or thermal velocity, are varied. We present a full derivation and numerical solution for the dispersion relation, and verify the PETSC-PIC numerical solutions to the Vlasov–Poisson system for a large range of wavenumbers and charge densities.

1. Introduction

In 1936, Lev Landau first formulated a simple kinetic model, now referred to as the Fokker–Plank equation in Landau form or simply just the Landau equation, for the description of charged particles in a plasma performing Coulomb collisions [24]. Ten years later, Landau furthered this discovery by predicting the damping of nonrelativistic, collisionless plasma oscillations, or Langmuir waves, for the first time [25]. The basic concept proposed in that paper, that a conservative phenomenon exhibits irreversible behaviors, has since influenced hundreds of papers and become one of the foundational problems in plasma physics. Thus, the phenomenon is now referred to as *Landau damping*. In his seminal paper, Landau used the solution to the Cauchy problem for the linearized Vlasov–Poisson equation around a spatially homogeneous Maxwellian equilibrium. Landau solved the equation analytically using Fourier and Laplace transforms and concluded that the electric field damps exponentially and that the decay is a function of the wavenumber, k , of the perturbation. In [5], Bohm and Gross provide a simple explanation for the damping in

MSC2020: 35Q83, 65-04, 65Y05, 68-04, 82D10.

Keywords: simulation, particle-in-cell, PETSc, plasma, Landau damping.

plasmas. In essence, plasmas exhibit a tendency to remain approximately field free. Therefore, if electric fields are introduced, either by external disturbance or by an incomplete space charge neutralization, the newly introduced fields will be forced out by a reaction from the free charges.

Through the years, numerous others have extensively examined Landau damping [12; 19; 36]. In 2009, a rigorous solution to the nonlinear Vlasov–Poisson equation was given by Villani and Mouhot in [31]. In their paper, the damping phenomenon is reinterpreted in terms of transfer of regularity between kinetic and spatial variables, rather than exchanges of energy, with phase mixing being the driving mechanism.

Developed in parallel to the theory behind Landau damping, numerical methods for approximating solutions to the kinetic plasma system were pioneered by Vlasov [37]. The particle-in-cell (PIC) method has been a popular choice for numerically simulating plasmas since its inception [17; 18], as it can considerably reduce the complexity of the system in comparison to direct N -body methods. The PIC method is a hybrid discretization algorithm comprised of two separate sets of bases for evaluation of different aspects of the problem. These bases are the particle basis, where the particle is represented by some (usually radially symmetric) shape function, and the mesh basis, where a mean field approach may be taken to computing different field quantities from external and self consistent forces. Typically, the continuum field solve is handled by employing the finite element method, although other formulations have used splines [10], finite difference methods [4], etc.

We present a particle-in-cell (PIC) method for solving the Vlasov–Poisson system using the portable extensible toolkit for scientific computing (PETSc) [2; 3]. PETSc-PIC uses symplectic integration schemes [1] for particle pushing while conducting field solves with a finite element method [21; 26]. The goal of PETSc is to provide composable pieces from which optimal simulations can be constructed. PETSc user level APIs allow applications to delay implementation choices, such as solver details, until runtime using dynamic configuration [6]. PETSc-PIC solvers fully conserve the moments, mass, momentum and energy at each time step while also preserving entropy monotonicity. Recent advances in the PETSc-PIC code [33] also include conservative projections between the finite element and particle basis, a key step towards hybrid FEM-particle algorithms.

2. Problem formulation

Consider the Vlasov–Poisson system, a common variation of the more general Vlasov–Maxwell–Landau system of equations in the nonrelativistic case where the magnetic and collisional effects are neglected. It can be an effective model for strongly non-Maxwellian plasmas. The Vlasov equation,

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} - \frac{q_e}{m} \mathbf{E} \cdot \frac{\partial f}{\partial \mathbf{v}} = 0, \quad (1)$$

describes the evolution of the phase space distribution, $f(\mathbf{x}, \mathbf{v}, t)$, defined over the domain $(\mathbf{x}, \mathbf{v}) \in \mathbb{R}^D \times \mathbb{R}^D$ where D is the spatial dimension. The electric field is obtained using Poisson's equation,

$$\Delta\phi(\mathbf{x}, t) = -\frac{\rho}{\epsilon_0}, \quad (2)$$

where ϕ is the electric potential, ρ is the charge density and $\mathbf{E} = -\nabla\phi$. The charge density contains a neutralizing background term, σ , such that

$$\rho(\mathbf{x}, \mathbf{v}, t) = \sigma - q_e \int_{\mathbb{R}^D} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}. \quad (3)$$

This neutralizing background simulates the effect of ions on the electrons in the domain. The use of a stationary, uniform background charge is based on the assumption that the ions are much heavier than the electrons and thus feel little influence from them.

In order to study the linear Landau damping phenomenon, we consider the initial particle distribution

$$f(x, v, t = 0) = \frac{1}{\sqrt{2\pi} v_{th}^2} e^{-v^2/2v_{th}^2} (1 + \alpha \cos(kx)), \quad (4)$$

$$(x, v) = \left[0, \frac{2\pi}{k}\right] \times [-v_{\max}, v_{\max}],$$

where $v_{th} = \sqrt{KT_e/m}$, $\alpha = 0.01$, $k = 0.5$, $v_{\max} = 10$ and the boundaries are periodic. An important piece of PIC methods for the Vlasov–Poisson system is the reduction of noise. Statistical noise in PIC methods is introduced during the initialization phase as continuum representations are discretized into a finite number of macroparticles. As PIC models evolve, the stability error for the electric field contains a term that will grow exponentially [15]. This build up of stability error is often referred to as “particle noise”. To reduce statistical noise in the initialization phase, we mimic a “quiet start” [7; 13; 30] continuum initialization in this work by placing particles at the center of the spatial and velocity cells and weighting them based on the initial distribution function $f(x, v, t = 0)$. This method is also used in [32], where particles are further remapped back to the cell centers every few steps. The remapping step provides enough particle noise reduction to accurately observe nonlinear effects in damping, however, as we are, for now, concerned only with the linear case of Landau damping, we will ignore the remapping phase.

2.1. Linear Landau damping. We seek to first derive a set of equations to understand the damping of plasma oscillations in our system and to calculate expected values for the damping rate and electric field oscillation frequency. These expressions are found by first deriving the dispersion relation for a plasma. The derivation shown follows from [9]. Consider a one-dimensional uniform plasma with an initial

distribution $f_0(v)$ with zero initial electric and magnetic fields, $\mathbf{E}_0 = \mathbf{B}_0 = 0$. To first order, the perturbation in $f(x, v, t)$ is denoted by $f_1(x, v, t)$ such that

$$f(x, v, t) = f_0(v) + f_1(x, v, t). \quad (5)$$

Plugging (5) into (1) gives

$$\frac{\partial f_1}{\partial t} + v \cdot \frac{\partial f_1}{\partial x} - \frac{q_e}{m_e} \mathbf{E}_1 \cdot \frac{\partial f_0}{\partial v} = 0. \quad (6)$$

Assuming that the ions are massive and fixed and that the waves are one-dimensional plane waves $f_1 \propto e^{i(kx - \omega t)}$, (6) becomes

$$f_1 = \frac{i q_e E}{m_e} \frac{\partial f_0 / \partial v}{\omega - kv}. \quad (7)$$

Recall the Poisson equation (2), with the potential ϕ replaced by the divergence of the electric field,

$$\nabla \cdot \mathbf{E} = \nabla \cdot \mathbf{E}_1 = -\frac{\rho}{\epsilon_0} = -\frac{1}{\epsilon_0} \left(\sigma - q_e \int (f_0(v) + f_1(x, v, t)) dv \right). \quad (8)$$

With zero initial electric field, the electric field vector is replaced by the electric perturbation, \mathbf{E}_1 , which takes the form $\mathbf{E}_1 = E e^{i(kx - \omega t)} \hat{\mathbf{x}}$. At equilibrium, the neutralizing background is equal to the total weight of the electron distribution, $\sigma = q_e \int f_0 dv$, leaving only the perturbation term f_1 in the Poisson equation. Thus we are left with

$$ik\epsilon_0 E = -q_e \int f_1 dv. \quad (9)$$

Substituting (7) into (9) and dividing by $ik\epsilon_0 E$, we have

$$1 = \frac{q_e^2}{km_e\epsilon_0} \int \frac{\partial f_0 / \partial v}{kv - \omega} dv. \quad (10)$$

Substituting in the plasma frequency, $\omega_p = (n_e q_e^2 / m_e \epsilon_0)^{1/2}$, and normalizing the electron number, n_e , to 1, leaves the dispersion relation

$$1 = \frac{\omega_p^2}{k^2} \int_{-\infty}^{\infty} \frac{\partial f_0 / \partial v}{v - (\omega/k)} dv. \quad (11)$$

Landau showed that this problem can be solved rigorously by means of the Laplace transform method. Importantly, it is necessary to go around the singularity in the integrand in (11) in the complex plane. The solution to (11) takes the form

$$\omega = \omega_r + i\gamma, \quad (12)$$

where ω_r represents the real oscillations of the plasma and γ the imaginary, which Landau showed to be the part of the solution driving the damping of the oscillations.

Following Landau's method [9], an approximation for the oscillation and damping terms can be derived, given by

$$\omega_r = 1 + \frac{3}{2}\hat{k}^2, \quad \gamma = -\sqrt{\frac{\pi}{8}} \frac{1}{\hat{k}^3} \exp\left[-\frac{1}{2\hat{k}^2}\right]. \quad (13)$$

A normalized form of the wavenumber k has been introduced to simplify the equations going forward. The normalized wavenumber, \hat{k} , is given by

$$\hat{k} = \frac{kv_{th}}{\omega_p}, \quad (14)$$

where $v_{th} = \sqrt{KT/m}$ is the thermal velocity. For all examples, we nondimensionalize so that $v_{th} = 1$. The real part of the solution to (11) was similarly derived by Vlasov in [37], however he did not account for the imaginary damping term.

These approximations are valid for the case where $\hat{k} \ll 1$ but their accuracy degrades considerably as \hat{k} approaches 1 and higher. Even when $\hat{k} = 0.5$, the calculated values for ω_r and γ differ from the numerical results by at least 5%. In [29], McKinstrie draws similar conclusions, electing to derive more accurate forms of (13) by expanding ω_r in powers of \hat{k} :

$$\omega_r = 1 + \frac{3}{2}\hat{k}^2 + \frac{15}{8}\hat{k}^4 + \frac{147}{16}\hat{k}^6, \quad (15)$$

$$\gamma = -\sqrt{\frac{\pi}{8}} \left(\frac{1}{\hat{k}^3} - 6\hat{k} \right) \exp\left[-\frac{1}{2\hat{k}^2} - \frac{3}{2} - 3\hat{k}^2 - 12\hat{k}^4\right]. \quad (16)$$

These new expressions are more accurate for \hat{k} up to 0.4 but still diverge from the correct values as \hat{k} increases further. Shalaby et al. provided further refinements to these equations in [34], using a numerical fitting formula, taking the form

$$\begin{aligned} \omega &= 1 + \frac{3}{2}\hat{k}^2 + \frac{15}{8}\hat{k}^4 + \frac{147}{16}\hat{k}^6 + 736.437\hat{k}^8 - 14729.3\hat{k}^{10} \\ &\quad + 105429\hat{k}^{12} - 370151\hat{k}^{14} + 645538\hat{k}^{16} - 448190\hat{k}^{18}, \\ \gamma &= -\sqrt{\frac{\pi}{8}} \left(\frac{1}{\hat{k}^3} - 6\hat{k} - 40.7173\hat{k}^3 + 3900.23\hat{k}^5 - 2462.25\hat{k}^7 - 274.99\hat{k}^9 \right) \\ &\quad \exp\left[-\frac{1}{2\hat{k}^2} - \frac{3}{2} - 3\hat{k}^2 - 12\hat{k}^4 - 575.516\hat{k}^6 + 3790.16\hat{k}^8 \right. \\ &\quad \left. - 8827.54\hat{k}^{10} + 7266.87\hat{k}^{12}\right]. \end{aligned} \quad (17)$$

These equations give good estimates for ω_r and γ in the case where $\hat{k} = 0.5$, which is of particular interest in this paper. In fact, the values obtained from (17) in the case where $\hat{k} = 0.5$ and all other parameters (ω_p , v_{th} , q_e , etc.) are assumed to be 1.0 match those commonly listed as "analytic solutions" [9; 32; 39]. That being said, the accuracy of the numerical fit still decreases considerably for $\hat{k} > 0.6$.

\hat{k}	ω_r	γ
0.25	1.1056	-0.0021693
0.5	1.4156	-0.15336
0.75	1.7371	-0.46192
1.0	2.0459	-0.85134
1.5	2.6323	-1.7757
2.0	3.1891	-2.8272

Table 1. Values for ω_r and γ for given values of \hat{k} from [8].

An alternate, and as we will show, more accurate way to calculate ω_r and γ for given values of \hat{k} is to find them by computing the zeros of (11). This was done by Canosa in [8] for values of k ranging from 0.25 to 2.0 in increments of 0.05 (see Table 1 for a selection of values). A comparison of the approximations by Landau, McKinstrie and Shalaby to the zero-finding results from Canosa is shown in Section 4.

3. PETSc-PIC

PETSc, the portable extensible toolkit for scientific computation, is a well-known library for numerical methods. It provides parallel data management, structured and unstructured meshes, linear and nonlinear algebraic solvers and preconditioners, optimization algorithms, time integrators and many more functions. The PETSc-PIC algorithm relies on two modules to handle the particle and mesh solves simultaneously. The first, DMPLex [22; 23; 26], is a PETSc module for generic unstructured mesh creation, manipulation, and I/O [16]. It decouples user applications from the implementation details of common mesh and discretization tasks. The other important module for this work, DMSwarm [28], provides a fully parallel solution for pure particle methods (e.g., DEM, SPH, EFG) and for particle-mesh methods (e.g., PIC, FLIP, MPM, GIMP).

We start with discussion of the particle methods in the PETSc-PIC algorithm. A method must first be chosen to represent the particle space and for interpolation between the mesh and particle representations. There are numerous choices in shape functions for this purpose, however, in our case a simple delta function representation of particles is chosen. Thus the approximation of the distribution function is defined in the particle space as

$$f_p = \sum_P \vec{\omega}_p \delta(\mathbf{x} - \mathbf{x}_p), \quad (18)$$

where $\vec{\omega}_p$ is the vector of weights, \mathbf{x} are the configuration space variables and \mathbf{x}_p represents the particle position and velocity, respectively. The finite element

representation, using a function space \mathcal{V} , is given by the weighted sum of basis functions

$$f_{FE} = \sum_i f_i \psi_i(\mathbf{x}), \quad (19)$$

where $\psi_i \in \mathcal{V}$ denotes the basis functions and f_i the associated finite element coefficient.

The Vlasov equation is a linear hyperbolic equation which may be written in a simpler form as

$$\frac{\partial f}{\partial t} + \mathbf{z} \cdot \nabla_{\mathbf{q}} f = 0, \quad (20)$$

where $\mathbf{q} = (\mathbf{x}, \mathbf{v})$ is the phase space variable and $\mathbf{z} = (\mathbf{v}, -q_e \mathbf{E}/m)$ is the combined force. The force term $-q_e \mathbf{E}/m$ is independent of velocity, and therefore (20) may be written in the conservative form

$$\frac{\partial f}{\partial t} + \nabla_{\mathbf{q}} \cdot (\mathbf{z} f) = 0. \quad (21)$$

Given this new advective form of the Vlasov equation, we can rewrite the equation for the characteristics $\mathbf{Q} = (\mathbf{X}, \mathbf{V})$ as

$$\frac{d\mathbf{Q}}{dt} = \mathbf{z}, \quad (22)$$

which reexpressed with the original phase-space variables gives

$$\frac{d\mathbf{X}}{dt} = \mathbf{V}, \quad \frac{d\mathbf{V}}{dt} = -\frac{q_e}{m} \mathbf{E}. \quad (23)$$

Since particles follow characteristics, the Vlasov equation in the particle basis becomes

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p, \quad \frac{d\mathbf{v}_p}{dt} = -\frac{q_e}{m} \mathbf{E}. \quad (24)$$

The equations of motion are stepped forward in time using structure-preserving symplectic integrators which have been well studied [14, pages 179–236]. The electric field is solved concurrently at each step using a finite element solver, discussed in the next section.

3.1. PETSc-FEM. At each step in the simulation, the Poisson equation is solved using the finite element method. The gradient of the potential, i.e., the electric field, is then interpolated across each cell at the particle locations. The interpolated electric field is then applied to the particles in the form of the Coulomb force.

The PETSc-FEM method is abstractly formalized by the *Ciarlet triple* [11; 20], such that a finite element is a triple $(\mathcal{T}, \mathcal{V}, \mathcal{V}')$, where

- the domain \mathcal{T} is a bounded, closed subset of \mathbb{R}^d (for $d = 1, 2, 3, \dots$) with nonempty interior and piecewise smooth boundary;

- the space $\mathcal{V} = \mathcal{V}(\Omega)$ is a finite-dimensional function space on \mathcal{T} of dimension n ;
- the set of degrees of freedom (nodes) $\mathcal{V}' = \{l_1, l_2, \dots, l_n\}$ is a basis for the dual space, that is, the space of bounded linear functionals on \mathcal{V} .

The cell \mathcal{T} together with the local function space \mathcal{V} and the set of rules for describing the functions in \mathcal{V} is the *finite element*. The discretization in PETSc is handled by the PETScFE object, which contains a PetscSpace (\mathcal{V}), PetscDualSpace (\mathcal{V}'), and DMplex (\mathcal{T}). PETScFE supports simplicial elements, tensor cells, and some special cells such as pyramids.

In general, the finite element solve for the Poisson equation can be accomplished using the standard H^1 function space. In the H^1 space, the weak form of the Poisson equation is

$$\int_{\Omega} \nabla \psi_i \cdot \nabla \phi = \int_{\Omega} \psi_i, \quad (25)$$

where $\psi \in V$ and V is the set of basis functions on the cell. The elements are then constructed such that the basis functions are continuous across the cell boundaries.

3.2. Conservative projections. To preserve the conservation laws in a PIC simulation, a method must be constructed to conservatively project between the particle and grid representations. Weak equality of the representations,

$$\int_{\Omega} \psi_i f_{FE} = \int_{\Omega} \psi_i f_P, \quad (26)$$

is enforced on the representations to achieve this [27; 33]. Restricting this equivalence to the finite-dimensional analogues gives the matrix-vector form,

$$M f_{FE} = M_p f_P, \quad (27)$$

where M is the finite element mass matrix,

$$M = \int_{\Omega} \psi_i \psi_j, \quad (28)$$

M_p is the particle mass matrix,

$$M_p = \int_{\omega} \psi_i \delta(\mathbf{x} - \mathbf{x}_p), \quad (29)$$

f_{FE} is a vector containing the finite element coefficients and f_P is the vector of particle weights. The entries of M_p contain evaluations of the finite element basis functions at particle locations with rows being determined by the basis function index, and columns being determined by the particle indices. Moving from the particle basis to the mesh, we must invert the finite element mass matrix, which is easily accomplished with CG/Jacobi [38]. In the other direction, we must invert a rectangular particle mass matrix, usually with LSQR [33].

4. Numerical results

The results of this numerical study are presented. We consider the one-dimensional (1X–1V) case of the Vlasov–Poisson system. By (15), derived in Section 2.1, and the zero-finding data from Canosa [8], the damping rate should be $\gamma = -0.153$ and the frequency of oscillations should be $\omega_r = 1.416$. All runs were conducted on a single 2.4 GHz 8-Core Intel Core i9 processor with 64 GB of memory. The example code and packages/options required to run it are provided in the Appendix.

To begin, we show results from the densest run of the PETSc-PIC simulation with 160 spatial cells and 32, 000 particles per cell and a PIC timestep of $dt = 0.3$. Figure 1 shows the maximum value of the electric field, $E_{\max} = \max_{\Omega} |E|$, over time. The values for γ and ω_r were measured by fitting the peaks of the given data. We ignore the peaks at $t = 0$ and past $t = 25$ as they may represent complex frequency roots other than the desired *Landau root* or data corrupted by particle noise. A more in depth discussion of how the *Landau root* is determined from simulation data is found in [4]. The frequency of oscillations describes the frequency of the electric field completing one full oscillation. Since each oscillation of E_{\max} is the equivalent of one half of the electric field period, we count two E_{\max} oscillations for each plasma oscillation. Values achieved by the PETSc-PIC algorithm, $\gamma = -0.1443$ and $\omega_r = 1.4117$, agree within 5.9% and 0.3%, respectively, of the analytic values from Canosa and Shalaby et al., which are assumed to be the most accurate for the case $\hat{k} = 0.5$. These values for γ and ω_r at $\hat{k} = 0.5$ were also calculated using a zero-finding algorithm, which utilizes Newton’s method, from [35]. The zero-finding algorithm calculates multiple values for ω_r and γ ; however, we select the solution

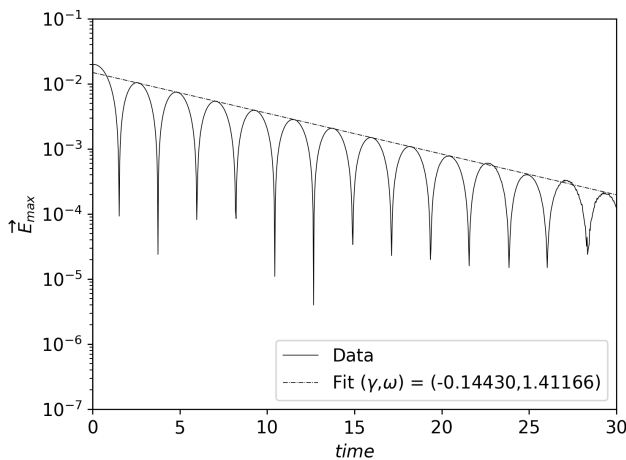


Figure 1. The maximum value of the electric field as a function of time for the one-dimensional linear Landau damping problem.

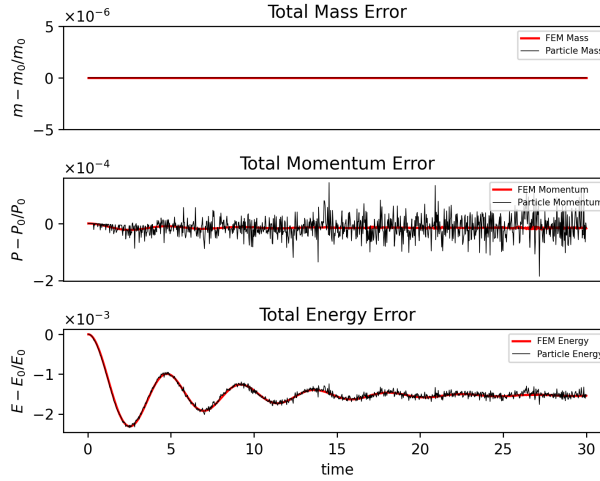


Figure 2. The total mass, momentum and energy error for the particle and finite element solve. The moment errors all converge to zero given a long enough time.

containing the largest γ which corresponds to the smallest ω_r . Other solutions found by the algorithm represent less dominant modes which can be ignored for the purposes of this study. As we discussed in Section 2, particle noise builds over time in the simulation and becomes dominant around $t > 25$. Increasing mesh and particle density has, thus far, been effective in reducing noise build up long enough to achieve time scales reported in similar PIC methods.

The primary solve function, `TSSTep`, takes on average 22.7 s per timestep, while the finite element solve, executed by the `SNESolve` function, requires 0.12 s per time step, or just 0.54% of the total step time. The most expensive operation in the PETSc-PIC algorithm is the conservative projection from the particle basis to the FEM basis. This is due to the currently unoptimized point locator function, `DMPlexLocatePoints`. In future work, this function will be optimized to decrease solver time.

The total error in the moments, shown in Figure 2, was shown to be stable over the entire runtime. At early times, the error in momentum and energy fluctuate but each converges by $t = 20$. This convergence comes from the use of basic symplectic integrator in PETSc-PIC which guarantees the error does not grow over time. We also note that the error in the particle solve and the finite element solve is exactly equal, apart from an increased level of noise in the particle solve. This confirms the effectiveness of the conservative projector used in PETSc-PIC. More detailed tests of the conservative projector can be found in [33].

Convergence studies were conducted for varying mesh and particle densities. We note here that running the densest cases on a single core is costly, particularly if we

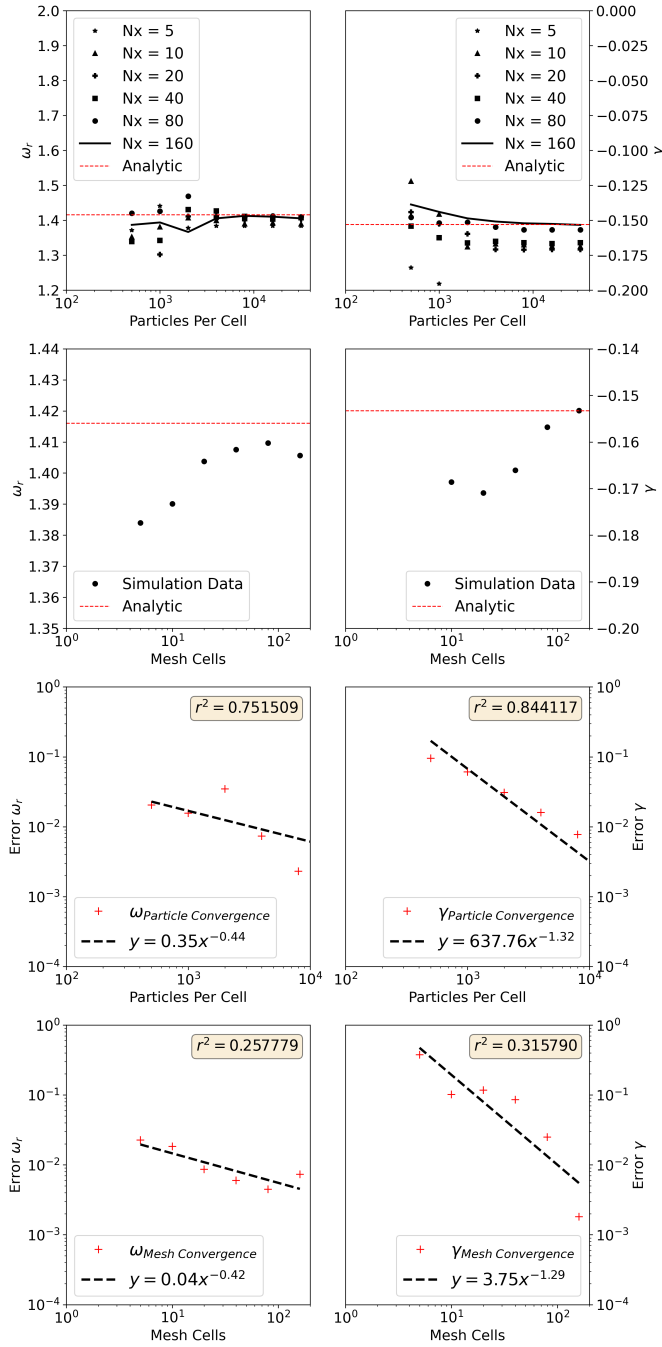


Figure 3. Results from various convergence studies: (first row) particle number convergence results for varying mesh densities, (second row) mesh convergence results using 32, 000 particles per cell, (third row) particle number convergence error using 160 mesh cells, and (fourth row) mesh density convergence error using 32, 000 particles per cell.

consider higher dimensional runs, e.g., $2x \times 2v$ and $3x \times 3v$. Including a remapping stage in our solve would allow for the reduction in particle numbers, however, the scalability of PETSc will similarly ameliorate the computational cost in future work. For now, we rely on scaling up the PETSc-PIC algorithm to increase solve times. Results from the convergence study, seen in Figure 3, show that increasing mesh densities requires increasing particle densities to properly converge towards the analytic values for γ and ω_r . Mesh convergence was measured at the densest particle per cell counts where particle convergence reaches asymptotic behavior. In both cases, as mesh and particle number increases, γ and ω_r approach the analytic values. From this data, we can conclude that given a dense enough population of particles within the domain, the overall error in the system is minimized. Convergence rates are provided in Figure 3, however, these values do not provide clear insight into the convergence behavior of the PETSc-PIC algorithm. Further work will be conducted in future studies to fully capture and explain the convergence behavior of the PETSc-PIC algorithm.

4.1. Variations in wavenumber and charge density. We have thus far shown that the PETSc-PIC algorithm is an accurate and structure-preserving method for modeling plasma systems. We next present results from tests in which the wavenumber k , and consequently the domain size, and the charge density were varied. Varying either of these values impacts the value of the nondimensional wavenumber \hat{k} . In the case of the wavenumber \hat{k} , the calculated values for ω_r and γ were compared to the values obtained with the approximation equations (13) and (15), the numerical fit (17) and the zero-finding results from Table 1. The results from PETSc-PIC, shown in Figure 4, clearly show a strong deviation of the approximation equations and the numerical fit for $\hat{k} > 0.5$ while closely matching the zero-finding data from Canosa and Newton's method for finding zeros. This demonstrates that these approximations quickly break down outside of the small parameter range typically chosen in numerical studies of Landau damping. When considering real plasma systems in which values for \hat{k} , ω_p , etc. are more dynamic, it is far more effective to use zero-finding methods to calculate expected values for ω_r and γ . It may be naively assumed that data from numerical tests with varying charge densities will match the approximation equations (13), (15) and (17) or even the zero-finding data from Canosa, however, these analytic results are based on an assumption of unchanging charge density. More specifically, these results are based on charge densities such that the plasma frequency, ω_p , is always unity. Therefore, to accurately compare analytic results to our data we must resolve the dispersion relation for varying charge densities. The zero-finding algorithm from [35] was employed to calculate new analytic values for ω_r and γ with charge densities ranging from 0.1 to 2.0. Figure 5 contains the results from the zero-finding

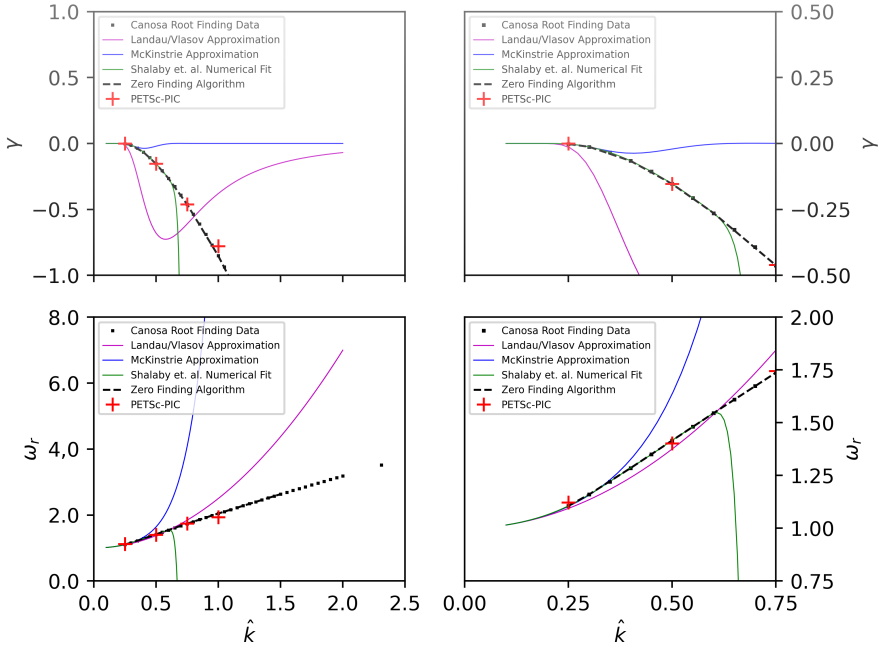


Figure 4. A comparison of various approximations for ω_r and γ to zero-finding algorithms and numerical results from PETSc-PIC. Plots on the right are zoomed in on the region $0.0 \leq \hat{k} \leq 0.75$ to show the accuracy of each approximation before they diverge from the data.

algorithm along with data from numerical tests which agree perfectly. We observe that when the charge density is increased, the frequency of oscillations also increases. This matches the expected physical behavior of an electrically charged plasma.

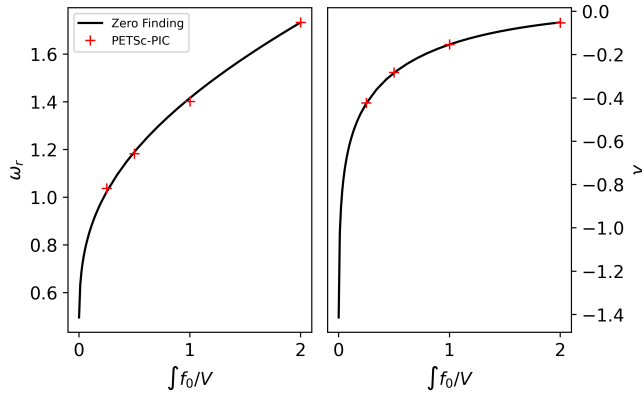


Figure 5. Numerical results for varying charge densities compared to zero-finding data. The charge density is represented on the x -axis as the integral of the initial distribution over the domain volume.

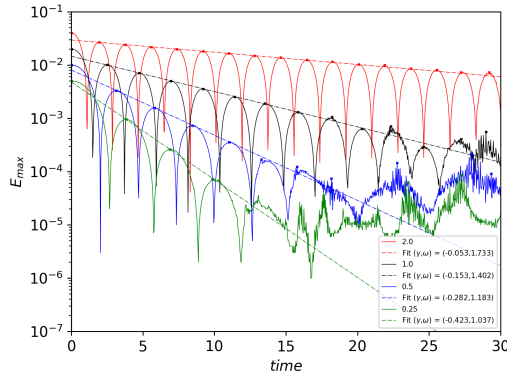


Figure 6. A comparison of electric field oscillations given different charge densities.

We have extended our zero-finding algorithm to the case where the charge density approaches zero ($\hat{k} \rightarrow \infty$) to make note of an interesting phenomenon. At a charge density of zero, the dispersion relation has no solution. We capture this in Figure 5, where we observe that both ω_r and γ are asymptotic at $\int f_0/V = 0$. This can similarly be observed in the numerical results from our PETSc-PIC algorithm. As the charge density is decreased, the rate at which the electric field oscillates becomes too large to resolve numerically. In the case of $\int f_0/V = 0.25$, shown in Figure 6, we can only observe two full oscillations of the electric field before the simulation becomes too noisy. Theoretically, the charge density could be decreased asymptotically in our simulations to observe the damping rate and frequency trends, but in practice there is too much noise to resolve any real processes in the plasma. The implementation of noise reduction techniques, such as the remapping phase used in [32], would allow for longer simulation times at these charge densities. Thus, we will implement these methods in future work.

5. Conclusions and future work

We have presented PETSc-PIC, a structure-preserving particle-in-cell algorithm for solving the electrostatic Vlasov–Poisson systems. The accuracy of our algorithm has been demonstrated by comparing the frequency of electric field oscillations and the damping rate of the oscillations to analytic values. We have also shown that the approximations for the frequency and damping rate break down outside of narrow ranges for the wavenumber and charge density. These approximations are often cited in numerical Landau damping studies without further context or reference to the equations used to compute the parameters, which can lead to complications in reproducing results. We have sought to provide a complete picture of Landau damping and the numerical methods we have used to simulate this phenomenon.

Future work with the PETSc-PIC algorithm will fall into two primary categories: improvements to the algorithm and the extension of the Landau damping test to multidimensional and nonlinear cases. Improvements to the algorithm will focus on reformulation using a mixed form of the Poisson equation and $H(\text{div})$ finite elements. We expect that C^0 electric fields will decrease the noise in our particle representation over time. While we have not observed any major negative impacts from using H^1 finite elements in the test problem chosen for this paper, Landau damping, using a mixed form makes a notable difference in the case of two-stream instability. PETSc currently includes support for the $H(\text{div})$ conforming finite elements Brezzi–Douglas–Marini (BDM) and Raviart–Thomas (RT) on simplicial grids, however RT elements are currently the only element type supported on tensor cells. We will also replicate our tests in parallel, allowing us to increase the number of particles per cell by several orders of magnitude, reducing the largest source of error in the code.

Nonlinear Landau damping is more complex in that nondamping phenomenon, such as plasma echo, are present. Vivally, the linearization of the Vlasov equation, used as the fundamental approximation in the study of linear Landau damping, does not guarantee that the asymptotic behavior of the linear Vlasov equation is an approximation of the asymptotic behavior of the nonlinear Vlasov equation [31]. There are reasons to doubt that the study of the linearized equations gives any hint on the long-time behavior of the nonlinear equations. Therefore if an algorithm is desired that can accurately capture the long-time behavior of a plasma, the nonlinear case of Landau damping must also be considered.

Appendix: Example code

The data presented in this paper can be recreated with PETSc using the DMSwarm example `ex9` (`$PETSC_DIR/src/dm/impls/swarm/tests/ex9.c`). Exact runtimes may vary depending on the architecture and compiler. The DMSwarm example can be run using the options

```
./ex9 -dm_plex_dim 2 -dm_plex_simplex 0 -dm_plex_box_bd periodic:none
      -dm_plex_box_faces 160,1 -dm_view
      -dm_plex_box_lower 0,-0.5 -dm_plex_box_upper 12.5664,0.5
      -dm_swarm_num_species 1 -dm_swarm_num_particles 5120000
      -vdm_plex_dim 1 -vdm_plex_simplex 0 -vdm_plex_box_faces 32000
      -vdm_plex_box_lower -10 -vdm_plex_box_upper 10
      -petscspace_degree 1 -em_type primal
      -em_pc_type svd -em_snes_atol 1.e-12
      -ts_type basicsymplectic -ts_basicsymplectic_type 1
      -ts_max_time 500 -ts_max_steps 1000 -ts_dt 0.3
      -fake_1D -cosine_coefficients 0.01,0.5 -charges -1.0,1.0
      -perturbed_weights -periodic
```

This example uses a 160 square cell mesh on the domain $(x, v) \in [0, 4\pi] \times [-10, 10]$, with 32,000 particles per cell. A first-order basic symplectic integrator is chosen as the time integration method and H^1 finite elements are chosen for the field solves.

References

- [1] S. Abhyankar, J. Brown, E. M. Constantinescu, D. Ghosh, B. F. Smith, and H. Zhang, *PETSc/TS: a modern scalable ODE/DAE solver library*, preprint, 2014. arXiv 1806.01437
- [2] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschel-man, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, and J. Zhang, *PETSc/TAO Users Manual*, 2022.
- [3] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, and J. Zhang, *PETSc Web page*, 2023.
- [4] J. W. Banks, A. G. Odu, R. Berger, T. Chapman, W. Arrighi, and S. Brunner, *High-order accurate conservative finite difference methods for Vlasov equations in 2D + 2V*, *SIAM J. Sci. Comput.* **41** (2019), no. 5, B953–B982. MR Zbl
- [5] D. Bohm and E. P. Gross, *Theory of plasma oscillations, A: Origin of medium-like behavior*, *Physical Review* **75** (1949), no. 12, 1851–1864.
- [6] J. Brown, M. G. Knepley, and B. F. Smith, *Run-time extensibility and librarization of simulation software*, *Computing in Science & Engineering* **17** (2015), no. 1, 38–45.
- [7] J. A. Byers, *Noise suppression techniques in macroparticle models of collisionless plasmas*, Proceedings of the fourth conference of numerical simulation of plasmas (Washington) (J. P. Boris and R. A. Shanny, eds.), Naval Research Laboratory, 1972, pp. 496–510.
- [8] J. Canosa, *Numerical solution of Landau’s dispersion equation*, *Journal of Computational Physics* **13** (1973), no. 1, 158–160. Zbl
- [9] F. F. Chen., *Introduction to plasma physics and controlled fusion, volume 1*, 2nd ed., Plenum press, 1984.
- [10] C. Cheng and G. Knorr, *The integration of the Vlasov equation in configuration space*, *J. Comput. Phys.* **22** (1976), no. 3, 330–351.
- [11] P. G. Ciarlet, *Numerical analysis of the finite element method*, Les Presses de l’Université de Montréal, Montreal, QC, 1976. MR Zbl
- [12] J. Dawson, *On Landau Damping*, *The Physics of Fluids* **4** (1961), no. 7, 869–874.
- [13] J. Denavit and J. M. Walsh, *Nonrandom initializations of particle codes*, *Comments on plasma physics and controlled fusion* **6** (1981), no. 6, 209–223.
- [14] E. Hairer, G. Wanner, and C. Lubich, *Geometric numerical integration*, Springer Series in Computational Mathematics, no. 31, Springer, Berlin, 2006. Zbl
- [15] O. H. Hald, *Convergence of vortex methods for Euler’s equations, II*, *SIAM J. Numer. Anal.* **16** (1979), no. 5, 726–755. MR Zbl
- [16] V. Hapla, M. G. Knepley, M. Afanasiev, C. Boehm, M. van Driel, L. Krischer, and A. Fichtner, *Fully parallel mesh I/O using PETSc DMplex with an application to waveform modeling*, *SIAM J. Sci. Comput.* **43** (2021), no. 2, C127–C153. MR Zbl

- [17] F. Harlow, M. Evans, and R. Richtmyer, *A machine calculation method for hydrodynamic problems*, Tech. report, Los Alamos Scientific Laboratory of the University of California, 1955.
- [18] F. H. Harlow, *The particle-in-cell method for numerical solution of problems in fluid dynamics*, Tech. report, Los Alamos Scientific Laboratory of the University of California, 1962.
- [19] J. D. Jackson, *Longitudinal plasma oscillations*, Journal of Nuclear Energy. Part C, Plasma Physics, Accelerators, Thermonuclear Research **1** (1960), no. 4, 171–189.
- [20] R. C. Kirby, *Algorithm 839: FIAT, a new paradigm for computing finite element basis functions*, ACM Trans. Math. Software **30** (2004), no. 4, 502–516. MR Zbl
- [21] M. G. Knepley, J. Brown, K. Rupp, and B. F. Smith, *Achieving high performance with unified residual evaluation*, preprint, 2013. arXiv 1309.1204
- [22] M. G. Knepley and D. A. Karpeev, *Mesh algorithms for PDE with sieve I: mesh distribution*, Scientific Programming **17** (2009), no. 3, 215–230.
- [23] M. G. Knepley, M. Lange, and G. J. Gorman, *Unstructured overlapping mesh distribution in parallel*, preprint, 2017. arXiv 1506.06194
- [24] L. Landau, *Die kinetische gleichung für den fall Coulombscher wechselwirkung*, Phys. Z. Sowjetunion **10** (1936), 154–164. Zbl
- [25] _____, *On the vibrations of the electronic plasma*, Acad. Sci. USSR. J. Phys. **10** (1946), 25–34. MR
- [26] M. Lange, L. Mitchell, M. G. Knepley, and G. J. Gorman, *Efficient mesh management in Firedrake using PETSc DMplex*, SIAM J. Sci. Comput. **38** (2016), no. 5, S143–S155. MR Zbl
- [27] J. Maddison and P. Farrell, *Directional integration on unstructured meshes via supermesh construction*, Journal of Computational Physics **231** (2012), no. 12, 4422–4432. Zbl
- [28] D. A. May and M. G. Knepley, *DMSwarm: particles in PETSc*, EGU General Assembly Conference Abstracts, 2017, p. 10133.
- [29] C. J. McKinstrie, R. E. Giacone, and E. A. Startsev, *Accurate formulas for the Landau damping rates of electrostatic waves*, Phys. Plasmas **6** (1999), no. 2, 463–466. MR
- [30] R. Morse and C. Nielson, *One-, two-, and three-dimensional numerical simulation of two-beam plasmas*, Phys. Rev. Lett. **23** (1969), no. 19, 1087–1090.
- [31] C. Mouhot and C. Villani, *On Landau damping*, Acta Math. **207** (2011), no. 1, 29–201. MR
- [32] A. Myers, P. Colella, and B. Van Straalen, *A 4th-order particle-in-cell method with phase-space remapping for the Vlasov–Poisson equation*, SIAM J. Sci. Comput. **39** (2017), no. 3, B467–B485. MR Zbl
- [33] J. V. Puszta, M. G. Knepley, and M. F. Adams, *Conservative projection between finite element and particle bases*, SIAM J. Sci. Comput. **44** (2022), no. 4, C310–C319. MR Zbl
- [34] M. Shalaby, A. E. Broderick, P. Chang, C. Pfrommer, A. Lamberts, and E. Puchwein, *SHARP: a spatially higher-order, relativistic particle-in-cell code*, The Astrophysical Journal **841** (2017), no. 1, art. id. 52.
- [35] E. Sonnendrücker, *Numerical Methods for the Vlasov equations*, lecture notes, Technical University of Munich, 2013.
- [36] N. G. van Kampen, *On the theory of stationary waves in plasmas*, Physica **21** (1955), 949–963. MR
- [37] A. Vlasov, *Über die schwingungseigenschaften des elektronengases*, Zh. Èksper. Teor. Fiz. **8** (1938), 291–318. Zbl
- [38] A. J. Wathen, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA J. Numer. Anal. **7** (1987), no. 4, 449–457. MR Zbl

- [39] T. Zhou, Y. Guo, and C.-W. Shu, *Numerical study on Landau damping*, *Physica D: Nonlinear Phenomena* **157** (2001), no. 4, 322–333.

Received February 20, 2023. Revised November 1, 2023.

DANIEL S. FINN: dsfinn@buffalo.edu

Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, United States

MATTHEW G. KNEPLEY: knepley@gmail.com

Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, United States

JOSEPH V. PUSZTAY: josephpu@buffalo.edu

Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, United States

MARK F. ADAMS: mfadams@lbl.gov

Scalable Solvers Group, Lawrence Berkeley National Laboratory, Berkeley, CA, United States

Communications in Applied Mathematics and Computational Science

msp.org/camcos

EDITORS

MANAGING EDITOR

John B. Bell
Lawrence Berkeley National Laboratory, USA
jbbell@lbl.gov

BOARD OF EDITORS

Ann Almgren	Lawrence Berkeley Nat. Lab., USA asalmgren@lbl.gov	Mitchell Luskin	University of Minnesota, USA luskin@umn.edu
Marsha Berger	New York University berger@cs.nyu.edu	Yvon Maday	Université Pierre et Marie Curie, France maday@ann.jussieu.fr
Alexandre Chorin	Univ. of California, Berkeley, USA chorin@math.berkeley.edu	Michael L. Minion	Lawrence Berkeley Nat. Lab., USA mlminion@lbl.gov
Phil Colella	Lawrence Berkeley Nat. Lab., USA pcolella@lbl.gov	Sorin Mitran	Univ. of North Carolina, Chapel Hill, USA mitran@unc.edu
Peter Constantin	Princeton University, USA const@math.princeton.edu	Robert D. Moser	University of Texas, USA rmoser@ices.utexas.edu
Maksymilian Dryja	Warsaw University, Poland dryja@mimuw.edu.pl	Per-Olof Persson	Univ. of California, Berkeley, USA persson@berkeley.edu
M. Gregory Forest	Univ. of North Carolina, Chapel Hill, USA forest@amath.unc.edu	Alfio Quarteroni	Politecnico di Milano, Italy alfio.quarteroni@polimi.it
Leslie Greengard	New York University, USA greengard@cims.nyu.edu	James Sethian	Univ. of California, Berkeley, USA sethian@math.berkeley.edu
Rupert Klein	Freie Universität Berlin, Germany rupert.klein@pik-potsdam.de	Eitan Tadmor	University of Maryland, USA tadmor@umd.edu
Ahmed Ghoniem	Massachusetts Inst. of Technology, USA ghoniem@mit.edu	Denis Talay	INRIA, France denis.talay@inria.fr
Nigel Goldenfeld	University of Illinois, USA nigel@uiuc.edu	Juan Luis Vázquez	Universidad Autónoma de Madrid, Spain juanluis.vazquez@uam.es
Raz Kupferman	The Hebrew University, Israel raz@math.huji.ac.il		

PRODUCTION

production@msp.org

Silvio Levy, Scientific Editor


See inside back cover or msp.org/camcos for submission instructions.

The subscription price for 2023 is US \$125/year for the electronic version, and \$185/year (+\$20, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscriber address should be sent to MSP.

Communications in Applied Mathematics and Computational Science (ISSN 2157-5452 electronic, 1559-3940 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online.

CMCoS peer review and production are managed by EditFLOW[®] from MSP.

PUBLISHED BY

 **mathematical sciences publishers**
nonprofit scientific publishing

<http://msp.org/>

© 2023 Mathematical Sciences Publishers

Communications in Applied Mathematics and Computational Science

vol. 18

no. 1

2023

Design of DIRK schemes with high weak stage order ABHIJIT BISWAS, DAVID I. KETCHESON, BENJAMIN SEIBOLD and DAVID SHIROKOFF	1
A connected component labeling algorithm for implicitly defined domains ROBERT I. SAYE	29
A parallel-in-time collocation method using diagonalization: theory and implementation for linear problems GAYATRI ČAKLOVIĆ, ROBERT SPECK and MARTIN FRANK	55
Hopf bifurcation analysis of a phage therapy model EI EI KYAW, HONGCHAN ZHENG and JINGJING WANG	87
A comparative study of iterative Riemann solvers for the shallow water and Euler equations CARLOS MUÑOZ MONCAYO, MANUEL QUEZADA DE LUNA and DAVID I. KETCHESON	107
A Numerical Study of Landau Damping with PETSc-PIC DANIEL S. FINN, MATTHEW G. KNEPLEY, JOSEPH V. PUSZTAY and MARK F. ADAMS	135
A source term method for Poisson problems with a discontinuous diffusion coefficient JOHN D. TOWERS	153