Sparse grid-based Adaptive Noise Reduction strategy for Particle-In-Cell schemes

Sriramkrishnan Muralikrishnan[a], Antoine J. Cerfon[b], Matthias Frey[d], Lee F. Ricketson[c], Andreas Adelmann[a]

[a] *Paul Scherrer Institut, Forschungsstrasse 111, 5232 Villigen, Switzerland.*
[b] *Courant Institute of Mathematical Sciences, New York University, New York NY 10012, USA.*
[c] *Lawrence Livermore National Laboratory, Livermore, USA.*
[d] *Paul Scherrer Institut, Forschungsstrasse 111, 5232 Villigen, Switzerland.*
Current affiliation: Mathematical Institute, University of St Andrews, St Andrews KY16 9SS, UK.

Abstract

We propose a sparse grid-based adaptive noise reduction strategy for electrostatic particle-in-cell (PIC) simulations. By projecting the charge density onto sparse grids we reduce the high-frequency particle noise. Thus, we exploit the ability of sparse grids to act as a multidimensional low-pass filter in our approach. Thanks to the truncated combination technique [1, 2, 3], we can reduce the larger grid-based error of the standard sparse grid approach for non-aligned and non-smooth functions. The truncated approach also provides a natural framework for minimizing the sum of grid-based and particle-based errors in the charge density. We show that our approach is, in fact, a filtering perspective for the noise reduction obtained with the sparse PIC schemes first introduced in [4]. This enables us to propose a heuristic based on the formal error analysis in [4] for selecting the optimal truncation parameter that minimizes the total error in charge density at each time step. Hence, unlike the physical and Fourier domain filters typically used in PIC codes for noise reduction, our approach automatically adapts to the mesh size, number of particles per cell, smoothness of the density profile and the initial sampling technique. It can also be easily integrated into high performance large-scale PIC code bases, because we only use sparse grids for filtering the charge density. All other operations remain on the regular grid, as in typical PIC codes. We demonstrate the efficiency and performance of our approach with two test cases: the diocotron instability in two dimensions and the three-dimensional electron dynamics in a Penning trap. Our run-time performance studies indicate that our approach can provide significant speedup and memory reduction to PIC simulations for achieving comparable accuracy in the charge density.

Keywords: PIC, Sparse grids, Filters, Adaptive noise reduction, Penning trap, Diocotron instability

1. Introduction

Particle-in-cell (PIC) schemes have been a popular and effective method for the simulation of kinetic plasmas for a long period of time [5, 6, 7]. Compared to continuum kinetic codes, PIC schemes effectively reduce the dimension from six to three for kinetic simulations requiring three spatial dimensions and three velocity dimensions (3D3V). On the other hand, compared to pure particle codes with direct summation, PIC reduces the computation of self-consistent forces from $\mathcal{O}(N_p^2)$ to $\mathcal{O}(N_p + N_c)$ where N_p is the total number of particles and $N_c \ll N_p$ is the number of mesh points. Even though the fast multipole method [8] reduces the complexity of pure particle schemes to $\mathcal{O}(N_p)$, such an approach has other limitations, such as the need for overly restrictive small time steps. Other attractive features of PIC schemes include simplicity, ease of parallelization and robustness for a wide variety of physical scenarios [4].

The main drawback of PIC schemes as compared to deterministic continuum kinetic schemes is the numerical error associated with particle noise [6, 9], which decreases slowly as one increases the number of particles. Specifically, the noise in PIC schemes decreases as $1/\sqrt{P_c}$ [6, 4] where $P_c = N_p/N_c$ is the number of particles per cell[1]. High fidelity large-scale 3D PIC simulations thus often require at least $\mathcal{O}(10^9)$ grid points and $\mathcal{O}(10^{12})$ particles to get the desired accuracy level [10]. These simulations require hours to complete even on large-scale state-of-the-art supercomputers available today. Thus, noise reduction approaches are of great interest to the PIC community to improve accuracy and also to speed up computations and reduce memory requirements.

There have been several efforts in this area in the past and a brief overview is given in section 3. Some of the strategies, such as the δf technique [11, 12, 13], are applicable for certain classes of plasma physics problems and give great computational savings. Their utility, however, is limited to these specific classes of problems. Filtering is a common noise reduction technique which finds applications in many production-level PIC codes such as TRISTAN-MP [14, 15], ORB5 [16], IMPACT-T [17] and Warp-X [18], to name a few. One of the primary reasons for this is its simplicity and ease of implementation in these frameworks. The stencil width and number of passes in case of digital filters and the cut-off wavenumber in case of Fourier domain filters is typically selected based on experience and knowledge about the physical problem at hand. Thus, these could result in scenarios where either too much signal is smoothed or the high-frequency noise is not removed sufficiently. Even if we managed to choose the parameters in the filter so that they are optimal for a particular mesh size, number of particles per cell, point in time and the initial sampling technique, they may no longer be optimal once we change any of the above and require tuning once again.

Our objective in this work is to develop a noise reduction strategy, or fil-

[1] In this paper, we define the number of particles per cell only with respect to the regular grid.

tering scheme for the charge density, that automatically adapts itself to the aforementioned parameters. As with other filtering techniques, we require it to be easily integrated into existing production-level PIC codes. Our starting point towards that goal is the recent work [4] which combined sparse grids with the PIC scheme. In that article, the authors showed that owing to the large cell sizes involved in sparse grids compared to regular grids, the PIC scheme combined with sparse grids has many more particles per cell than its regular counterpart. This led to significant noise reduction and enormous speedups for certain classes of problems which have smooth or axis-aligned density profiles.

Now, let us give a brief overview of the present work. We revisit and reinterpret the noise reduction component of the scheme introduced in [4] from a filtering perspective, to construct a sparse grid-based noise reduction strategy for electrostatic PIC simulations. Unlike [4], where all the operations occur on sparse grids, in our approach the sparse grids come into play only for noise reduction of the charge density. Hence, for a user of PIC (who may not be familiar with sparse grids) it exactly resembles a filtering routine - i.e., it takes as input unfiltered charge density on the regular grid, and returns as output the filtered charge density on the same grid. Compared to existing filtering approaches, this sparse grid-based approach is superior for functions which are smooth or aligned with an axis. In simple terms, this can be understood as follows: with any filtering technique the reduction in noise comes with a price, which is an increase in the grid-based error. The unique aspect of our sparse grid filtering is that the resulting noise reduction can also be viewed from a Monte-Carlo perspective. Thanks to this property, we have maximal noise reduction, since the sparse grid approximation involves cells with maximal size, which in the context of PIC, for a given total number of particles, translates to a maximal number of particles per cell. At the same time the increase in grid-based error for smooth or axis-aligned functions is minimal. However, the same cannot be said for all functions in general, and for these general cases the increase in grid-based error associated with sparse grids may be high. In order to tackle that issue, we use the so-called truncated combination technique [1, 2, 3], which reduces the large grid-based error of standard sparse grid technique for non-aligned and non-smooth functions. This is because the truncated combination technique uses a different choice of coarse grids with finer mesh sizes than those used in the standard sparse grid combination. The truncation parameter involved in the combination technique is crucial for minimizing the sum of grid-based error and particle noise. Hence, we propose a heuristic based on formal error analysis to calculate the optimal truncation parameter on the fly which minimizes the total error.

This paper is organized as follows. Section 2 introduces the PIC method in the context of electrostatic Vlasov-Poisson equations. Section 3 briefly reviews the existing noise reduction strategies in PIC and provides motivation and objectives for this article. Section 4 explains in detail the components and algorithm for a sparse grid-based adaptive noise reduction strategy. Numerical results for the 2D diocotron test case and 3D penning trap are presented in section 5 and section 6 presents conclusions and proposes future work.

2. Particle-in-cell method

In this work, without loss of generality, we consider the non-relativistic electrostatic Vlasov-Poisson system with a fixed magnetic field, and introduce the PIC method in that setting. The electrons are immersed in a uniform, immobile, neutralizing background ion population and the system is given by

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{q_e}{m_e} \left(\mathbf{E} + \mathbf{v} \times \mathbf{B}_{ext} \right) \cdot \nabla_{\mathbf{v}} f = 0, \tag{1}$$

where $\mathbf{E} = \mathbf{E}_{sc} + \mathbf{E}_{ext}$, and the self-consistent field due to space charge is given by

$$\mathbf{E}_{sc} = -\nabla \phi, \quad -\Delta \phi = \rho = \rho_e - \rho_i.$$

In the above equation $f(\mathbf{x}, \mathbf{v}, t)$ is the electron phase-space distribution, q_e and m_e are the electron charge and mass respectively. The total electron charge in the system is given by $Q_e = q_e \int \int f d\mathbf{x} d\mathbf{v}$, the electron charge density by $\rho_e(\mathbf{x}) = q_e \int f d\mathbf{v}$ and the constant ion density by $\rho_i = \frac{Q_e}{\int d\mathbf{x}}$. Throughout this paper we use bold letters for vectors and non-bold ones for scalars.

The particle-in-cell method discretizes the phase space distribution $f(\mathbf{x}, \mathbf{v}, t)$ in a Lagrangian way by means of macro-particles (hereafter referred to as "particles" for simplicity). At time $t = 0$, the distribution f is sampled to get the particles and after that a typical computational cycle in PIC consists of the following steps:

1. Assign a shape function - e.g., cloud-in-cell [6] - to each particle p and deposit the electron charge onto an underlying mesh.
2. Use a grid-based Poisson solver to compute ϕ by solving $-\Delta \phi = \rho$ and differentiate ϕ to get the electric field $\mathbf{E} = -\nabla \phi$ on the mesh.
3. Interpolate $\mathbf{E}$ from the grid points to particle locations $\mathbf{x}_p$ using an interpolation function. This is typically known as field gathering.
4. By means of a time integrator advance the particle positions and velocities using

$$\frac{d\mathbf{v}_p}{dt} = \frac{q_e}{m_e} \left(\mathbf{E} + \mathbf{v} \times \mathbf{B}_{ext} \right) |_{\mathbf{x}=\mathbf{x}_p},$$
$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p.$$

The sources of different errors in the PIC simulations and their orders of accuracy for typical choices are as follows. For simplicity, if we consider a uniform mesh with spacing h in all the directions then for the shape functions used in typical PIC schemes (B-splines), the grid-based error scales as $\mathcal{O}(h^2)$ [19, 20]. This is a result of approximating Dirac-δ functions in the configuration space by shape functions of compact support. The Poisson equation is typically solved by means of FFT solvers or by multigrid methods. In case of multigrid solvers the equation is discretized by second-order finite difference or finite element schemes. The field solves together with the interpolation (typically linear)

accounts for an additional $\mathcal{O}(h^2)$ [21]. The particle noise is the result of approximating the expected value of the shape function by an arithmetic mean over a finite number of discrete particles. It scales as $(N_p h^d)^{-1/2}$ [4], where d is the spatial dimension of the problem. The initial distribution is sampled using one of the standard sampling techniques such as the naive Monte-Carlo strategy [12], importance sampling [12] or by means of the quiet start [20, 22, 21]. The choice of initial sampling plays an important role in determining the constant associated with the particle noise. Finally, for time integration, typical choices are the second-order leap-frog scheme [6] and Runge-Kutta schemes of order 2 and higher. If we consider the leap-frog scheme then the error in the time discretization scales as $\mathcal{O}(\Delta t^2)$. The mesh size h, time step Δt and the number of particles N_p in most PIC simulations are such that the dominant error comes from the particle noise. Hence, high fidelity simulations typically require a large number of particles to minimize it. The high noise associated with PIC simulations has motivated researchers to develop several noise reduction strategies, which we discuss next.

3. Noise reduction strategies in PIC

Noise reduction can be achieved in several ways in the context of PIC simulations, categorized as: (i) variance reduction techniques such as the δf method [11, 12, 13] and quiet start [13]; (ii) phase space remapping [20, 22, 21]; (iii) filtering in physical domain [6, 23, 14, 15, 24], Fourier domain [6, 16] and wavelet domain [25, 17, 26]. This list is not exhaustive and there are many other contributions in this area. In addition, recently a noise reduction strategy using kernel density estimation algorithm has been proposed in [27], where the authors adaptively select the shape functions in PIC which minimize the sum of bias squared and variance of the error in the density. Also, in [4] sparse grid techniques are used to achieve noise reduction in PIC. We discuss this method in detail in section 4.7, since this approach has the most in common with the present work. In this section, we focus on the filtering strategies.

The goal of filtering in PIC simulations is to smooth high frequency oscillations usually associated with noise. Filtering can be done in any field quantity, although the most common one in electrostatic PIC is the charge density [23] as it is the origin of noise and the potential and electric field are smoother because of the integration inherent in solving Poisson's equation. In case of filtering in the physical domain, one typically selects a filter of certain stencil width - e.g., binomial filter - and does a few passes on the field quantity. On the other hand, for filters in the Fourier domain, a maximum wavenumber is specified by the user and the filter eliminates all the wavenumbers higher than the specified cut-off wavenumber [6]. In almost all the filtering strategies, the number of passes/stencil width in the physical domain or the cut-off wavenumber in the Fourier domain has to be chosen *a priori* such that the total error, which is the sum of grid-based error (bias) and particle noise (variance), is minimized. However, in practice there are not many constructive strategies available to pick these parameters and in many cases the values are chosen based on a rule of

thumb and previous experience [28]. Even if one manages to choose these parameters so that they are optimal for a particular point in time, mesh, number of particles per cell and sampling technique, they are unlikely to remain optimal as the simulation evolves. Indeed, due to non-linear space-charge effects, fine scale structures appear in the density and this changes the smoothness of the profile continuously with time. Hence, an ideal filter should be adaptive with respect to all aforementioned parameters to minimize the total error. Towards this goal, we propose a sparse grid-based adaptive noise reduction strategy in the following section.

4. Sparse grid-based noise reduction

4.1. Sparse grid combination technique

The sparse grid combination technique was first introduced in [29] as a way to approximate smooth functions on rectangular grids efficiently by using a specific linear combination of their approximations on different coarse grids. If we consider linear interpolation as an example, then for a regular grid of mesh size h we need $\mathcal{O}(h^{-d})$ grid points to get an accuracy of $\mathcal{O}(h^2)$. The sparse grid combination technique on the other hand uses only $\mathcal{O}(h^{-1}|log(h)|^{(d-1)})$ total grid points to get an accuracy of $\mathcal{O}(h^2|log(h)|^{(d-1)})$ for smooth functions, which is only slightly deteriorated compared to the regular grids. More precisely, the requirement for realizing this accuracy is the existence of an error expansion of the form $C_1(h_i)h_i^2+C_2(h_j)h_j^2+D_1(h_i,h_j)h_i^2h_j^2$ in 2D (and similar expressions in higher dimensions), where C_1, C_2 and D_1 are appropriate coefficient functions with a uniform upper bound independent of the mesh sizes [29, 30, 4]. Thus, we can clearly see the advantages of sparse grids in high dimensions, where they have found many applications [31]. The key idea is the cancellations that happen between the error expansions in the different coarse grids, which are called component grids in the sparse grid terminology. Also, the scalar values that multiply each component grid involved in the combination are called the combination coefficients. In Figure 1 an illustration is shown, where we can see the different component grids and their combination coefficients involved in approximating a $2^8 \times 2^8$ regular grid. The literature on the sparse grid combination technique and sparse grids in general is vast and the readers can refer to [31, 29, 32, 33, 34] and the references therein for more details. We will now show how sparse grid combination can be used to achieve noise reduction in the context of PIC.

4.2. Sparse grid filter

Let us consider a domain of size $[0, L]^d$, where d is the dimension (typically $d = 2$ or 3[2]), and for simplicity a regular grid of mesh size $h = \frac{L}{2^n}$ in all the

[2]For $d = 1$, sparse grids are same as the regular grids, and our noise reduction will thus not be applicable for 1D1V PIC.

directions. In our noise reduction strategy, after step 1 in the PIC algorithm shown in section 2 we perform a sparse grid projection of the charge density as follows

$$\varrho_e = G\tilde{\rho}_e = \left(\sum_{l=1}^{nc} c_l P_l R_l\right)\tilde{\rho}_e. \tag{2}$$

Here, $\tilde{\rho}_e$ and ϱ_e are the charge densities on the regular grid before and after the sparse grid transformation. R_l and P_l are the transfer operators[3] which transfer the density from the regular grid to the lth component grid in the sparse grid combination technique and vice versa, respectively. c_l is the combination coefficient for the lth component grid which is a scalar value and nc is the number of component grids involved in the combination technique. We also denote the transfer operators and combination coefficients simply as R, P and c in places where the subscript l is not needed.

One requirement for the transfer operators P_l and R_l is to ensure global charge conservation. In our approach, we use the cloud-in-cell or linear interpolation function, which is given by

$$W_l(\mathbf{x} - \tilde{\mathbf{x}}) = \prod_{m=1}^{d} \max\left\{0, 1 - \frac{|x_m - \tilde{x}_m|}{h_m}\right\} \tag{3}$$

where $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are the locations of the grid points in the lth component grid and regular grid, respectively, and h_m is the mesh size of the lth component grid along the mth coordinate axis. The operators R_l and P_l in terms of this function are given by

$$R_l(i,j) = \frac{h^d}{V_l} W_l\left(\mathbf{x}_i - \tilde{\mathbf{x}}_j\right), \tag{4}$$

$$P_l(j,i) = W_l\left(\mathbf{x}_i - \tilde{\mathbf{x}}_j\right) \quad \text{for} \quad i = 1, \ldots, N_l \quad j = 1, \ldots, N_c \tag{5}$$

where V_l is the volume of each cell in the lth component grid and N_c, N_l are the number of points in the regular grid and lth component grid respectively.

Upon considering the standard sparse grid combination technique in [29], one sees that the sparse grid projection or interpolation in equation (2) essentially removes high frequency components which are coupled between the axes. This is because the sparse grid combination corresponding to a regular grid of mesh size h does not have the fine resolution h in all the directions. In this sense, the sparse grid combination acts as a multi-dimensional low pass filter and keeps only certain wavenumbers resolved by a regular grid of mesh size h. This is the filtering point of view for the noise reduction obtained from the sparse grids. It can also be understood from a Monte Carlo point of view as shown in [4]

[3] We call these operators as R and P simply because they resemble restriction and prolongation operators in multigrid methods. However, we would like to note that the analogy ends there and the requirements for the transfer operators in the current context and the multigrid methods are different.

by means of increased particles per cell in the sparse grids compared to the regular grid for the same total number of particles. However, in the sparse PIC presented in [4] the particles deposit directly onto the component grids, unlike the strategy pursued here. These two approaches are related as stated in the following proposition, and hence the noise reduction obtained with the sparse grids can be understood from a Monte Carlo point of view *or* from a filtering perspective. In later sections, we will leverage this equivalence to explain the noise reduction with sparse grids depending on the context.

Proposition 1. *For node-centered grids and linear interpolation shape functions, the direct charge density deposition onto the component grids in the sparse PIC approach [4] is equivalent to first depositing the charge density onto the regular grid and then transferring it to the component grids by means of the operator R in equation (4)[4]. That is, the two approaches result in identical charge densities. In the case of cell-centered grids, an exact equivalence between the two approaches does not hold. There, the two-step approach can be viewed as direct charge deposition onto the component grids with a different shape function than the standard hat function, which is also second-order accurate.*

Proof. The proof is given in appendix A. □

The advantage of the Monte Carlo point of view is that we can estimate the grid-based error and particle noise with explicit dependence on the number of particles and mesh size as we show in the section 4.4. From a pure filtering perspective, this may be very difficult or not possible.

Now, we are interested in knowing how much grid-based error and particle noise are increased and decreased, respectively, by the sparse grid filter. To answer this, we observe that for interpolation the sparse grid combination technique is equivalent to the sparse grids based on hierarchical bases [32]. The latter is identified based on an optimization process [31] which guarantees for smooth functions, the fewest degrees of freedom for maximal accuracy of $\mathcal{O}\left(|log(h)|^{d-1}h^2\right)$ based on the L^2 or L^∞ norm. Thanks to this, in the context of PIC, the sparse grid transformation in equation (2) gives maximal noise reduction (because of the minimal number of grid points and hence maximum particles per cell) and at the same time the increase in grid-based error is minimal for smooth functions. Thus, compared to other filters, the one based on the standard sparse grid combination technique is optimal in the sense of minimizing the total error for functions which are either smooth or aligned with an axis.

4.3. Truncated combination technique to handle non-aligned and non-smooth functions

The optimality mentioned in the previous section for sparse grid filtering is no longer applicable in case of non-smooth functions or functions which are not

[4] Let us refer this as two-step approach for simplicity.

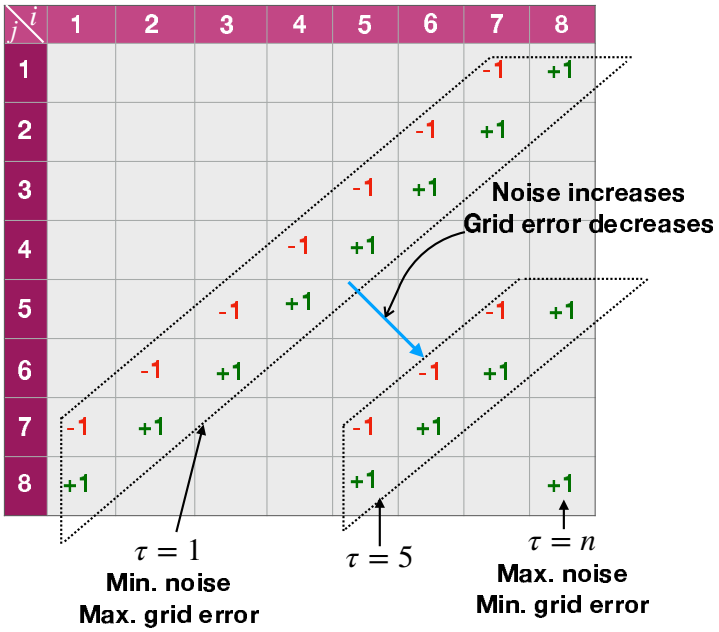


Figure 1: Schematic explaining the sparse grid combination technique and how the truncated combination can be used to minimize the total error. Here, $\tau = 1$ corresponds to the standard sparse grid combination technique and $\tau = n$ corresponds to the regular grid. The $+1$ and -1 are the combination coefficients c_l in equation (2) corresponding to the component grids.

aligned with either of the axes. Here the grid-based error is significantly larger than the regular grid because of large mixed derivatives [35], which leads the coefficient D_1 in the error expansion given in section 4.1 being much larger than other coefficients. While the sparse grid scaling remains optimal, the coefficient in front of that scaling can be so large as to eliminate its benefits at practical grid resolutions. This is why in [4], the authors reported poor performance of sparse PIC for the diocotron instability test case as it falls into the non-aligned category when simulated with a Cartesian grid. There are a few ways to tackle this problem, as mentioned in [4, 36]. Options include optimized coordinate systems which evolve with the charge density, and the use of spatially adaptive sparse grids. These strategies, which are perhaps more elegant from a mathematical point of view and more efficient, have the drawback of requiring significant changes to existing regular PIC code bases. Also, no detailed, robust algorithm is known at present.

Here, we pursue another direction using the truncated combination technique [1, 2, 3], which is much simpler and can be easily implemented in existing codes. The truncated combination technique was originally proposed as a modification to the standard sparse grid combination technique to tackle convergence issues in certain types of PDEs in financial applications caused by the presence of extremely anisotropic grids in the standard sparse grid technique.

In Figure 1, we show the different combination strategies for a 2D problem with a regular mesh of size $2^8 \times 2^8$. The indices i and j on the row and column headers in Figure 1 indicate the mesh sizes of the component grids involved in the combination technique such that the (i, j)th component grid has mesh sizes

$h_i = \frac{L}{2^i}$ and $h_j = \frac{L}{2^j}$, where L is the length of the domain in each direction. The truncated combination technique [1, 2, 3] introduces a truncation parameter τ[5], which is a positive integer that determines the component grids involved in the combination. Precisely, the component grids corresponding to a truncation parameter τ has indices $(i, j) \geq \tau$ as shown in Figure 1. Moreover, except for $\tau = n$, there are two sets of component grids: one with $i + j = n + \tau$ and combination coefficient $c = 1$, and the other with $i + j = n + \tau - 1$ and $c = -1$. If we consider a $2^n \times 2^n$ regular grid, then the value of $\tau = 1$ corresponds to the standard combination technique in [29] and $\tau = n$ corresponds to the regular grid. By increasing τ, fewer component grids are used in the combination technique, but each with finer resolution than the previous τ. This alleviates the issue of non-aligned and non-smooth functions by controlling the error term associated with the mixed fourth derivatives. Thus, the truncated combination technique provides a unified framework to transition from standard sparse grid to regular grid in terms of approximation capability by increasing τ.

Let us consider a PIC simulation with N_p total particles and a $2^n \times 2^n$ regular grid with mesh size $h = \frac{L}{2^n}$. The regular grid with $\tau = n$ will have the minimal grid-based error and maximal noise because it has the mesh size h in all the directions. The standard sparse grid technique with $\tau = 1$, at the other extreme, has maximal grid-based error and minimal noise as it has the mesh size h in directions aligned with x or y axis but not in others. As we increase τ from 1 to n as shown in Figure 1, we decrease the grid-based error because of the inclusion of finer mesh sizes in the component grids but at the same time increase the particle noise due to decreased particles per cell or, from the filtering perspective, the inclusion of higher wavenumbers in the filtering process of equation (2). Thus depending on the smoothness and the orientation of the function there is an optimal τ at which the total error, which is the sum of grid-based error and particle noise, is minimized. Hence, the truncated combination technique provides a natural way to minimize the total error within the framework of sparse grid-based noise reduction without much modification to the standard sparse grid combination technique. In the following we will present a formal error analysis and propose a heuristic approach to estimate the optimal τ.

4.4. Formal error analysis

In [4], a formal error analysis is presented for sparse PIC quantifying the grid-based error and particle noise. Proposition 1 states the exact equivalence between the direct charge deposition in [4] and our new filtering approach for the case of node-centered grids. Thus, for PIC codes based on node-centered grids[6], the formal error analysis in [4] is directly applicable. In contrast, our

[5] For the time being we consider the same truncation parameter τ in all the directions for the clarity of the exposition. We refer the readers to Remark 1 for more general cases.

[6] We highlight the fact that for the scheme we present in this article, only the centering scheme of the charge density matters. The other fields do not play a role in our noise reduction algorithm, and the analysis is therefore independent of their centerings.

codes are based on cell-centered grids (as is the default choice in many plasma PIC codes [37, 18]). According to Proposition 1, the direct charge deposition in [4] and the current approach are not exactly equivalent for cell-centered grids because of the differences in the shape functions. Nevertheless, the order of accuracy is the same for both approaches and they differ only by constants. Hence, we will largely follow the steps in [4] and generalize it to include the truncated combination technique.

As shown in [4] and appendix B, approximating ρ_e in PIC simulations consists of two parts: namely, grid-based error and particle noise. In what follows we will quantify these two components to get an estimate of the total error.

4.4.1. Grid-based error

Let us recall the different notations for charge density which will be of use here. ρ_e is the exact electron charge density given by

$$\rho_e(\mathbf{x}) = q_e \int f(\mathbf{x}, \mathbf{v}) d\mathbf{v} = \int \int f(\xi, \mathbf{v}) \delta(\mathbf{x} - \xi) d\xi d\mathbf{v}.$$

The density on the regular grid before the sparse grid transformation is $\tilde{\rho}_e$ and it is obtained from ρ_e by first approximating delta-functions in configuration space by shape functions of compact support (see equation (28) in appendix B) and then approximating the expected value of the shape function by an arithmetic mean over a finite number of discrete particles (see equation (38) in appendix B). The density on the regular grid after the sparse grid transformation in equation (2) is ϱ_e. We will denote the grid error component of the total error as $||\rho_e - \varrho_e||_{grid}$, where for simplicity we have denoted the L^∞ norm $||\cdot||_{L^\infty}$ by $||\cdot||$ (equivalently, we can also use the L^2-norm). In our approach, the grid-based error comes from the approximation of delta-functions in configuration space by shape functions of compact support as well as from the transfer operators R and P.

Towards quantifying the grid-based error, for simplicity, let us consider a 2D PIC simulation in a periodic domain $[0, L]^2$ and a regular mesh of size $2^n \times 2^n$. Let the mesh size of the regular grid be $h_n = \frac{L}{2^n}$ and the mesh sizes of the component grids be $h_i = \frac{L}{2^i}$ and $h_j = \frac{L}{2^j}$ for the (i, j)th component grid in Figure 1. In our approach, we use the cloud-in-cell or linear interpolation operators for all the grid transfer operations. Hence, from Proposition 1 and the grid-based error derived in equation (36) of appendix B, we use an error expansion of the form $C_1(h_i)h_i^2 + C_2(h_j)h_j^2 + D_1(h_i, h_j)h_i^2 h_j^2$ similar to [4, 29, 1, 38], where C_1, C_2 and D_1 are appropriate coefficient functions with a uniform upper bound. The summation over the component grids in equation (2) leads to pair-wise cancellations both in the standard sparse grid combination technique as well as in the truncated combination technique as shown in Figure 1. After multiplying by the combination coefficients and summing across all the component grids,

we get

$$(\rho_e - \varrho_e)_{grid} = C_1(h_n)h_n^2 + C_2(h_n)h_n^2 + \frac{4h_n^2L^2}{2^{2\tau}}\left[\frac{1}{4}\sum_{\substack{i+j=n+\tau\\ i,j\geq\tau}} D_1(h_i,h_j) - \sum_{\substack{i+j=n+\tau-1\\ i,j\geq\tau}} D_1(h_i,h_j)\right], \quad (6)$$

where we used the fact that $h_ih_j = \frac{h_nL}{2^\tau}$ when $i+j = n+\tau$ and $h_ih_j = \frac{h_nL}{2^{(\tau-1)}}$ when $i+j = n+\tau-1$. Taking the norm of both sides of the above equation and noting that there are $n-(\tau-1)$ component grids with $i+j = n+\tau$ and $(n-1)-(\tau-1)$ component grids with $i+j = n+\tau-1$, we obtain

$$\begin{aligned}||\rho_e - \varrho_e||_{grid} &\leq \kappa_1 h_n^2 + \kappa_2 h_n^2 + \frac{4\beta_1 h_n^2 L^2}{2^{2\tau}}\left[\frac{n-(\tau-1)}{4} + \{(n-1)-(\tau-1)\}\right]\\ &\leq h_n^2\left(\kappa_1+\kappa_2+\beta_1 L^2 2^{-2\tau}\left[5(n-\tau)+1\right]\right). \end{aligned} \quad (7)$$

Here, κ_1, κ_2 and β_1 are constants corresponding to the upper bounds such that $||C_1(h_n)|| \leq \kappa_1$, $||C_2(h_n)|| \leq \kappa_2$ and $||D_1(h_i,h_j)|| \leq \beta_1$, $\forall h_i, h_j$. The same expression for the error is also obtained in [1] for the truncated combination in 2D. Similarly one can derive the estimates in 3D and the grid-based error in that case is given by

$$\begin{aligned}||\rho_e - \varrho_e||_{grid} \leq h_n^2\Big(&\kappa_1+\kappa_2+\kappa_3+(\beta_1+\beta_2+\beta_3)L^2 2^{-2\tau}\left[5(n-\tau)+1\right]\\ &+\gamma L^4 2^{-(4\tau+1)}\left\{25(n-\tau)^2 - 5(n-\tau)+2\right\}\Big), \end{aligned} \quad (8)$$

where the upper bounds for the coefficient functions in 3D are such that $||C_d(h_n)|| \leq \kappa_d$, $||D_d(h_i,h_j)|| \leq \beta_d$ and $||F(h_i,h_j,h_k)|| \leq \gamma$ for $d = 1,2,3$ and $\forall h_i, h_j, h_k$. By plugging in $\tau = 1$ and $\tau = n$ in (7) and (8) we recover the estimates for the standard sparse grid combination in [29] and for regular grids respectively.

4.4.2. Particle noise

Now, we will derive estimates for the particle noise component of the total error. The particle noise is the result of approximating the expected value of the shape function by an arithmetic mean over a finite number of discrete particles. As per the error analysis in [4], in 2D the particle noise in each component grid is $\mathcal{O}\left(1/\sqrt{N_p h_i h_j}\right)$ and as stated in the grid error estimates we have $n-(\tau-1)$ component grids each with $h_ih_j = \frac{h_nL}{2^\tau}$ and $(n-1)-(\tau-1)$ component grids with $h_ih_j = \frac{h_nL}{2^{(\tau-1)}}$. Thus we can write an estimate for the particle noise as

$$\begin{aligned}||\rho_e - \varrho_e||_{noise} &= \mathcal{O}\left(\sigma\left[\frac{n-(\tau-1)}{\sqrt{\frac{N_p h_n L}{2^\tau}}} + \frac{(n-1)-(\tau-1)}{\sqrt{\frac{N_p h_n L}{2^{(\tau-1)}}}}\right]\right)\\ &= \mathcal{O}\left(\sigma\left\{\frac{2^{0.5(\tau-1)}\left[(n-\tau)(1+\sqrt{2})+\sqrt{2}\right]}{\sqrt{N_p h_n L}}\right\}\right), \end{aligned} \quad (9)$$

where σ is a particle noise constant. Following the same procedure, the noise estimate in 3D is given by

$$||\rho_e - \varrho_e||_{noise} = \mathcal{O}\left(\sigma\left\{\frac{2^{(\tau-2)}\left[(3+\sqrt{2})(n-\tau)^2+(5+\sqrt{2})(n-\tau)+4\right]}{\sqrt{N_p h_n L^2}}\right\}\right). \tag{10}$$

Again, by plugging in $\tau = 1$ and $\tau = n$ in equations (9), (10) we recover the estimates shown in [4] for the standard sparse grid technique and regular grids respectively. With the grid and particle error estimates in hand, we will show how these can be used in practice to adaptively select the optimal τ.

4.4.3. Heuristic approach for the quantitative estimation of the coefficients in the error analysis

In order to use the grid and particle error estimates derived in the previous section we need to have a quantitative estimate of the coefficients. To that end, we note that a rigorous derivation of coefficients for the current approach in the case of cell-centered grids depends on the ratio of the mesh sizes of the component grids to the regular grid and is more involved. Instead, in this section we approximate the grid and particle coefficients based on heuristic arguments and empirical observations and intend to improve these choices in the future iterations of our algorithm. Let us first consider the grid-based error. As explained in [4, 36] and equations (36) and (37) in appendix B, the coefficient functions in the grid error estimates are proportional to the derivatives of the charge density ρ_e such that

$$C_1 \propto \frac{\partial^2 \rho_e}{\partial x^2}, C_2 \propto \frac{\partial^2 \rho_e}{\partial y^2}, C_3 \propto \frac{\partial^2 \rho_e}{\partial z^2}, D_1 \propto \frac{\partial^4 \rho_e}{\partial x^2 \partial y^2}$$
$$D_2 \propto \frac{\partial^4 \rho_e}{\partial y^2 \partial z^2}, D_3 \propto \frac{\partial^4 \rho_e}{\partial z^2 \partial x^2}, F \propto \frac{\partial^6 \rho_e}{\partial x^2 \partial y^2 \partial z^2}.$$

In PIC, we only have an approximation of ρ_e on the regular grid, which we call $\tilde{\rho}_e$ as defined in equation (38), and this also contains the particle noise. In order to have a realistic approximation of the derivatives of the charge density from the noisy regular PIC data $\tilde{\rho}_e$, we perform a denoising by thresholding in the Fourier domain. Specifically, we first take the Fourier transform of the density on the regular grid $\hat{\rho}_e = \mathcal{F}(\tilde{\rho}_e)$ and perform a hard thresholding such that

$$\chi_\epsilon(\hat{\rho}_e) := \begin{cases} \hat{\rho}_e & |\hat{\rho}_e| \geq \epsilon, \\ 0 & |\hat{\rho}_e| < \epsilon, \end{cases} \tag{11}$$

where $\hat{\rho}_e$ is a vector and the operator $\chi_\epsilon(\cdot)$ acts on it component wise. Here, ϵ is the threshold for denoising and $|\hat{\rho}_e|$ denotes the magnitude of the Fourier transform $\hat{\rho}_e$. This type of denoising is common in signal processing as well as wavelet denoising [39] techniques.

The threshold parameter ϵ is a function of the number of particles per cell P_c, the initial sampling method and also the distribution f. It determines how

much noise and signal is removed by the denoising process. Too low a value will not remove much noise and too high a value may remove a significant portion of the signal along with the noise. However, in contrast to denoising techniques in signal processing where after applying this threshold one performs an inverse transform to get the signal in the physical domain, we emphasize the fact that for our scheme we only use it for selecting the truncation parameter τ (which performs the final filtering). Hence the threshold ϵ does not need to be optimal, and we only need to ensure that we do not pick up excessive noise.

At present, we use an ad-hoc strategy to select the value of ϵ as a certain percentage of the maximum value of $|\hat{\rho}_e|$, namely $\epsilon = \alpha \max(|\hat{\rho}_e|)$, where α denotes the percentage. To determine α in our algorithm, for a certain number of particles per cell $(P_c)_{ref}$ (e.g., 5) we run the PIC simulation for a few different values of α and pick the minimum value necessary for denoising. To reduce the run time we use the coarsest mesh possible for the problem in these simulations. Once we pick the value of α for a reference number of particles per cell $(P_c)_{ref}$, we run simulations with other values of P_c by multiplying α by $\sqrt{(P_c)_{ref}/P_c}$, as we know the noise in PIC methods scales as $1/\sqrt{P_c}$.

To give an idea of how one can execute this process, in our numerical experiments in section 5 we typically start with $\alpha = 0.01$ (ϵ is one percent of the maximum value of $|\hat{\rho}_e|$) as we found it to be a good initial guess through many experiments. In order to examine whether the selected value of α is sufficient for denoising, we examine the theoretical error curves from the τ estimator as shown in the right columns of Figures 3-5 and 9-10. From these figures we can see that when the grid based error is dominant (which is the case for low τ values) there is a specific shape to these curves which is dictated by the physical evolution of the density. If on the other hand the particle noise is dominant (high τ values), then these curves are almost flat as the noise is insensitive to the time evolution of the density. If the selected value of α is not large enough for denoising, then even the theoretical error curves for low τ cases are insensitive to the density evolution with visible anomalies. In such a case, we increase the value of α until we do not see this behavior any more. On the other hand, if the selected value of α is too high, then we decrease it until we see the anomalies, and select the value just before this behavior is observed. In addition to the theoretical error curves, we also use the time history of optimal τ as shown in Figures 6 and 11 to help in the detection of anomalies and guide us in the process of whether to increase or decrease the initial value of α selected. Using this process we found that anomalies start to occur for the values of $\alpha = 0.005, 0.025, 0.004$ for the 2D diocotron instability with Gaussian sampling, uniform sampling and 3D Penning trap respectively in section 5. We thus chose the values of $\alpha = 0.01, 0.03$ and 0.005 for these three cases respectively to provide enough denoising.

Currently the selection of α is intrusive and performed manually, although it needs to be done only once for a test case. In future work, we will develop a more systematic way to pick the threshold directly from the density data, based on techniques similar to the ones used in wavelet denoising [39]. Machine learning techniques can also be used for this purpose, and this is another direction we

will pursue.

After denoising the charge density, we compute the derivatives in the Fourier domain and perform inverse transforms. Next, in order to find the constants in front of these derivatives in appendix B we derive the grid-based error for regular PIC schemes. Since each component grid in the sparse grid combination technique is a regular grid with mesh sizes h_i, h_j and h_k, equations (36) and (37) can be used for determining the constants involved in the upper bounds. To that end, we note that the grid transfer operators R and P incur twice the grid-based error of similar magnitude given in equations (36) and (37). Moreover, the charge density $\tilde{\rho}_e$ in the regular grid adds another $1/12$ in front of the second derivative terms. Summing all these contributions we get an estimate for the coefficients in equations (7) and (8) as

$$\kappa_1 = \frac{1}{4}\left\|\frac{\partial^2 \bar{\rho}_e}{\partial x^2}\right\|, \kappa_2 = \frac{1}{4}\left\|\frac{\partial^2 \bar{\rho}_e}{\partial y^2}\right\|, \kappa_3 = \frac{1}{4}\left\|\frac{\partial^2 \bar{\rho}_e}{\partial z^2}\right\|, \beta_1 = \frac{1}{72}\left\|\frac{\partial^4 \bar{\rho}_e}{\partial x^2 \partial y^2}\right\|$$
$$\beta_2 = \frac{1}{72}\left\|\frac{\partial^4 \bar{\rho}_e}{\partial y^2 \partial z^2}\right\|, \beta_3 = \frac{1}{72}\left\|\frac{\partial^4 \bar{\rho}_e}{\partial z^2 \partial x^2}\right\|, \gamma = \frac{1}{864}\left\|\frac{\partial^6 \bar{\rho}_e}{\partial x^2 \partial y^2 \partial z^2}\right\|, \tag{12}$$

where $\bar{\rho}_e$ is the denoised charge density defined in equation (28).

Finally, following the particle noise estimates in equations (52) and (53) as well as [4, 17], for our algorithm we take

$$\sigma = \sqrt{(2/3)^d \left\|Q_e \tilde{\rho}_e\right\|} \tag{13}$$

in equations (9) and (10), where d is the dimension and $\tilde{\rho}_e$ is the charge density on the regular grid before denoising as defined in equation (38). Here, we use the density $\tilde{\rho}_e$ instead of the denoised density $\bar{\rho}_e$ as it helps in adjusting the particle constant with respect to different sampling techniques.

Through numerical experiments we also found another choice for the coefficients in the grid-based error and particle noise as

$$\kappa_1 = \left\|k_x^2 \hat{\rho}_e\right\|, \kappa_2 = \left\|k_y^2 \hat{\rho}_e\right\|, \kappa_3 = \left\|k_z^2 \hat{\rho}_e\right\|, \beta_1 = \left\|k_x^2 k_y^2 \hat{\rho}_e\right\|$$
$$\beta_2 = \left\|k_y^2 k_z^2 \hat{\rho}_e\right\|, \beta_3 = \left\|k_x^2 k_z^2 \hat{\rho}_e\right\|, \gamma = \left\|k_x^2 k_y^2 k_z^2 \hat{\rho}_e\right\|, \sigma = \sqrt{\left\|Q_e \tilde{\rho}_e\right\|} \tag{14}$$

where k_x, k_y and k_z are the wavenumbers in x, y and z respectively. We do not present detailed results, but for the numerical experiments in section 5 as well as for other synthetic examples in the context of interpolation we found this choice yields similar optimal τ values as that of the constants in equations (12) and (13). It has an added advantage that we do not need to take inverse transform of the derivatives, which is three in 2D and seven in 3D. Thus it may be of interest from a practical point of view, and for the numerical experiments in section 5 we observed up to 7 times speedup in the τ estimation part with this choice compared to the ones in equations (12) and (13).

In Algorithm 1 we consolidate the steps in the optimal τ estimator algorithm. For the range of τ, we consider $[1, n-3]$ for 2D and $[1, n-2]$ for 3D where 2^n is the number of points in the regular grid in each dimension. We do not

include the extreme values of τ ($[n-2, n]$ for 2D and n for 3D[7]) because we observed consistent false optima in the τ estimation due to these cases in our numerical experiments. These false optima can be explained by the fact that the high τ cases are less penalized by the inaccurate upper bounds of the triangle inequality than the low τ ones, because fewer components grids are involved in the combination. Currently, unless we take the specific properties of a given simulation into account, we do not know of a general strategy which can resolve this problem. Hence, we plan to improve this in our future work.

Remark 1. *So far, for the sake of the clarity and simplicity of our presentation, we have used the same number of grid points in all the directions to explain the steps of the noise reduction strategy. Here, for completeness, we will briefly outline the procedure needed for the general case of different grid resolutions in each direction. To that end, we define a few convenient notations. We again consider the two-dimensional case for simplicity, with the extension to three dimensions left as a straightforward task for the reader. Let us define* $\mathbf{n} = \{n_1, n_2\}$ *as the extension of its scalar counterpart. Since we want the target level of the sparse grid approximation space [33] to be the same as the underlying regular grid, we also need to use different truncation parameters in each direction. Let us denote these by* $\boldsymbol{\tau} = \{\tau_1, \tau_2\}$. *Let* $n_{max} = \max(\mathbf{n})$ *and* $n_{min} = \min(\mathbf{n})$. *The parameter* τ *can now take the values* $1 \leq \tau \leq n_{min}$, *and for each value of* τ *we calculate the final truncation parameter* $\boldsymbol{\tau}$ *(which is only used in the error analysis) according to [33]:*

$$a = \min(\mathbf{n} - \tau \cdot \mathbf{1}), \tag{15}$$

$$\boldsymbol{\tau} = \mathbf{n} - a \cdot \mathbf{1}, \tag{16}$$

where $\mathbf{1} = \{1, 1\}$. *The component grids corresponding to parameter* τ *now will have* $i \geq \tau_1$, $j \geq \tau_2$ *and again there are two sets of component grids: one with* $i + j = n_{max} + \tau$, $c = 1$ *and the other with* $i + j = n_{max} + \tau - 1$, $c = -1$. *The grid and particle errors can then be derived in a similar fashion as in sections 4.4.1 and 4.4.2.*

4.5. Implementation in a HPC PIC code base.

Once the optimal τ is obtained from Algorithm 1 we need to perform sparse grid noise reduction. In Algorithm 2 we present a matrix-free implementation of the sparse grid filtering in equation (2). This implementation is more suitable for large-scale high performance PIC code bases like OPAL (which are mostly matrix-free) than the matrix version in equation (2). In these codes, the density in the regular grid is domain-decomposed between different processors and in Algorithm 2 each processor holds the entire component grid in the combination technique. For moderate values of τ, each component grid has very few degrees of freedom compared to the regular grid and this is not very expensive in terms of

[7] In the current sparse grids setup $\tau = n - 1$ is not possible for 3D.

memory. However, for high τ, the component grids involved in the combination have a considerable number of degrees of freedom (especially in 3D) and hence both memory as well as the MPI_Allreduce step in Algorithm 2 could present a bottleneck. In our future work we will also split up the component grids between processors which would require a more complicated parallelization strategy as shown in [40].

If the parallelization of the code base uses MPI for inter-node parallelism and OpenMP, GPU or any other accelerator for intra-node parallelism then the for-loop over component grids in Algorithm 2 can also be done in parallel with the available intra-node shared memory parallelism. Algorithms 1 and 2 are performed in between steps 1 and 2 in the regular PIC procedure outlined in section 2. Ingredients such as the FFT, which are required for the tauEstimator algorithm, are already available in many large-scale PIC code bases and hence these two algorithms can be incorporated inside them very easily without any modification to the other parts.

Algorithm 1 tauEstimator: An algorithm for estimating optimal τ.

1: Compute Fourier transform of the charge density $\hat{\rho}_e = \mathcal{F}(\tilde{\rho}_e)$.
2: Perform denoising by hard thresholding according to equation (11).
3: Compute the constants for the grid-based error with (12) and the particle error constant (13).
4: **for** $\tau = 1$ **to** $n - 3$ for 2D and $n - 2$ for 3D **do**
5: Evaluate grid-based error and particle noise using equations (7),(9) for 2D and (8),(10) for 3D.
6: **end for**
7: Select the τ with minimum total error.

Algorithm 2 transferToSparse: An algorithm for sparse grid-based noise reduction with a given τ.

1: **for** $l = 1$ **to** nc **do**
2: Each processor deposits their regular grid partition of $\tilde{\rho}_e$ to the lth component grid using the transfer operator R_l in equation (4).
3: MPI_Allreduce to add contributions from all processors on the lth component grid.
4: Each processor interpolates from the lth component grid to their regular grid partition of $\tilde{\rho}_e$ using transfer operator P_l in equation (5).
5: Multiply by combination coefficient c_l and accumulate.
6: **end for**

Remark 2. *In general the charge density ϱ_e after sparse grid transformation is not guaranteed to be positive everywhere. This is not unique to our approach and also happens in other noise reduction strategies such as high-order shape functions [21], compensating filters [6] and wavelet-based density estimation [41]. In our numerical results in section 5 we do not observe any problems caused by*

this. However, we could adopt the density redistribution procedure used in [21] to make the charge density positive everywhere after the sparse grids transformation. This will be studied in future versions of the algorithm. Also, as shown in [28], the filtering procedures used in explicit PIC simulations improve energy conservation but at the loss of momentum conservation. In our future study we will investigate in detail the impact of the noise reduction strategy on energy and momentum conservation and report the results.

4.6. Computational complexity estimates of the noise reduction strategy

Here, we provide the asymptotic serial computational complexity estimates for the tauEstimator (Algorithm 1) and transferToSparse (Algorithm 2) parts of the noise reduction strategy. The dominant computational components of the tauEstimator are the FFT and inverse FFTs, each of which has a complexity of $\mathcal{O}\left(N_c log_2(N_c)\right)$. In the case of the transferToSparse algorithm, we have nc component grids, and for each component grid we deposit the regular grid density onto the component grid and then interpolate it back to the regular grid. The deposition and interpolation both are of complexity $\mathcal{O}(N_c)$, and since we do it for nc component grids it results in $\mathcal{O}(nc \cdot N_c)$. Now the number of component grids in 2D and 3D are $nc = \mathcal{O}\left(log_2(N_c) - \tau\right)$ and $nc = \mathcal{O}\left(\left\{log_2(N_c) - \tau\right\}^2\right)$ respectively. Thus the complexity of the transferToSparse part of the noise reduction is $\mathcal{O}\left(N_c \left(log_2(N_c) - \tau\right)^{d-1}\right)$, where d is the dimension. Hence, summing up the contributions from both parts, the total complexity of the noise reduction algorithm per time step is $\mathcal{O}\left(N_c \left\{log_2(N_c) + \left(log_2(N_c) - \tau\right)^{d-1}\right\}\right)$. The cost of typical physical domain filters such as the binomial filter is $\mathcal{O}(N_c)$. Hence, the asymptotic cost of our approach is slightly more than the usual filters. Taking into account the adaptivity of our approach, this is only a small price to pay. In terms of additional memory requirements, for both the tauEstimator and transferToSparse parts, they are $\mathcal{O}\left(N_c\right)$ which is similar to other filters. In PIC schemes, memory requirements of particles usually dominate as the number of particles is far more than N_c. Additionally, each particle contains many attributes (e.g. position, velocity, charge etc.). Thus, the additional memory requirement caused by the noise reduction strategy is usually not significant.

4.7. Relation between sparse grid-based noise reduction strategy and sparse PIC schemes

In this section we compare and contrast the sparse PIC scheme introduced in [4] with the noise reduction strategy proposed in the current work. The distinctions may be enumerated as follows.

- As mentioned in the introduction, the sparse PIC scheme in [4] performs all the operations - e.g. charge deposition and Poisson solve - on the sparse grids and does not introduce regular grids at all (except for visualization purposes or post-processing). This absence of a regular grid can provide computational and memory savings. By contrast, the current approach is

designed to be an add-on for standard PIC schemes. We use sparse grids only for noise reduction in the charge density, while all the operations such as charge deposition and the Poisson solve happen on the regular grid as in typical PIC codes.

- In [4], the noise reduction obtained from the sparse grids is viewed from a Monte-Carlo perspective. In the current work we construct the strategy based on a filtering perspective and use the Monte Carlo perspective for the error analysis to find the optimal τ. This is possible because of the equivalence between the two perspectives, as shown in Proposition 1.

- The truncated combination technique and the tauEstimator can also be used in the context of the sparse PIC scheme in [4] - although this fact is not noted in that work - at the expense of reintroducing regular grid complexity. However, in the regime where particle operations dominate, this may be a worthwhile trade-off.

- The adaptive noise reduction strategy can also be used offline as a post-processing tool to filter the charge density (or any other grid quantity) from regular PIC simulations.

To summarize, the sparse PIC scheme in [4] can be used as an alternative to regular PIC, whereas the sparse grid-based noise reduction strategy is an accessory to improve the performance of regular PIC.

5. Numerical results

In this section we will test the performance of the adaptive noise reduction strategy on two benchmark problems in plasma physics and beam dynamics; namely two-dimensional diocotron instability, and three-dimensional electron dynamics in a Penning trap with a neutralizing ion background. These test cases produce fine-scale structures during the nonlinear evolution and thus can be used to evaluate the ability of the adaptive τ method to capture them while still reducing noise. Also, they are very relevant to the large-scale accelerator simulations which we intend to perform in our future works.

In all the simulations we consider a periodic box $\Omega = [0, L]^d$, where d is the dimension and L is the length in each dimension. The charge to mass ratio q_e/m_e in all our simulations is -1. In measuring the error in field quantities we use the relative discrete L^2-norm also known as the normalized root mean squared error given by

$$\mathcal{E}(\boldsymbol{\psi}) = \sqrt{\frac{\sum_{i=1}^{N_{points}} \left(\boldsymbol{\psi}(\mathbf{x}_i) - \boldsymbol{\psi}_{ref}(\mathbf{x}_i)\right)^2}{\sum_{i=1}^{N_{points}} (\boldsymbol{\psi}_{ref}(\mathbf{x}_i))^2}}, \tag{17}$$

where $\boldsymbol{\psi}$ is any field quantity, $\boldsymbol{\psi}_{ref}$ is the reference field which is obtained from an ensemble average of high-resolution regular PIC simulations and $\mathbf{x}_i$ are the

locations of points in the domain at which we measure the error. This error is for a particular point in time and we measure the error at few instants in the whole simulation. In both numerical examples, we calculate the error for regular PIC, adaptive τ PIC and fixed τ PIC with the range of τ taken to be the same as the one used in the tauEstimator Algorithm 1. By means of these error curves we can see how well the adaptive τ algorithm performs in terms of picking the optimal τ and also how the errors compare to that of the regular PIC results with different number of particles per cell P_c. We always define the number of particles per cell P_c based on the regular grid. It is given by

$$P_c = \frac{N_p}{N_c} = \frac{N_p}{2^{nd}}.$$

For the time integration we use the leap-frog method and for the Poisson equation we use the second order cell-centered finite difference method as in [42, 43] with single level and without any spatial adaptivity. For solving the linear system arising from the discretized Poisson equation we use the smoothed aggregation algebraic multigrid (SAAMG) from the second generation Trilinos MueLu library [44]. The stopping tolerance for the iterative solver is set as 10^{-10} multiplied by the infinity norm of the right hand side. More details on the solver can be found in [43]. The code is written on top of a C++ miniapp based on the particle accelerator library OPAL [37] and box structured adaptive mesh refinement library AMReX [45]. Even though FFT solver would be the most accurate and fastest option [46] in this context, the reason for the above choice of field solver is in our future work we want to extend the current approach to include adaptive mesh refinement. Also, the conclusions of the present study will not be much affected by this choice and will be applicable for FFT solver too.

All the computations are performed on the Merlin6 HPC cluster at the Paul Scherrer Institut, the details of which are as follows. Each Merlin6 node consists of 2 sockets and each socket in turn has Intel Xeon Gold 6152 processor with 22 cores at 2.1-3.7GHz. There are 2 threads in each core, however in all the present computations we only use single thread. Each node contains 384 GB DDR4 memory in total.

5.1. 2D diocotron instability

5.1.1. Problem description and simulation setup

As a first example, we consider the 2D diocotron instability test case as already described in [4]. In this test case, we have electrons with a hollow density profile immersed in a neutralizing immobile and uniform ion background and confined by a uniform external axial magnetic field. The magnetic field is strong enough that the electron dynamics is dominated by advection in the self-consistent $\mathbf{E}_{sc} \times \mathbf{B}_{ext}$ velocity field [47, 48, 49, 50]. The initial electron density profile is not monotonic in the radial direction, which translates to an $\mathbf{E}_{sc} \times \mathbf{B}_{ext}$ shear flow which is unstable to what is known as the Kelvin-Helmholtz shear layer instability [47, 51, 50] in fluid dynamics, and the diocotron instability in

beam and plasma physics [12, 52, 47]. This instability deforms the initially axisymmetric electron density distribution, leading, in the nonlinear phase, to the formation of a discrete number of vortices, and eventually breakup [50, 52]. This test case has importance both from a fundamental physics point of view [12, 52, 47] as well as in practical applications such as beam collimation [53].

The parameters for this test case are as follows and are very similar to the ones in [4]. We apply a uniform external magnetic field $\mathbf{B}_{ext} = \{0, 0, 5\}$ along the $z-$axis in a domain of length $L = 22$. The external electric field $\mathbf{E}_{ext} = \mathbf{0}$ for this problem. The initial distribution is given by

$$f(t=0) = \frac{C}{2\pi} e^{-|\mathbf{v}|^2/2} \exp\left\{-\frac{(r-L/4)^2}{2(0.03L)^2}\right\},$$
$$r = \sqrt{(x-L/2)^2 + (y-L/2)^2}, \tag{18}$$

and the constant C is chosen such that the total electron charge $Q_e = -400$. We sample the phase space using Gaussian distribution in the velocity variables with mean 0 and standard deviation 1. For the configuration space, we use a uniform distribution for θ in $[0, 2\pi]$, and for r a Gaussian distribution with mean $L/4$ and standard deviation $0.03L$. From (r, θ) we do the polar to Cartesian transformation to get (x_p, y_p) for the particles.

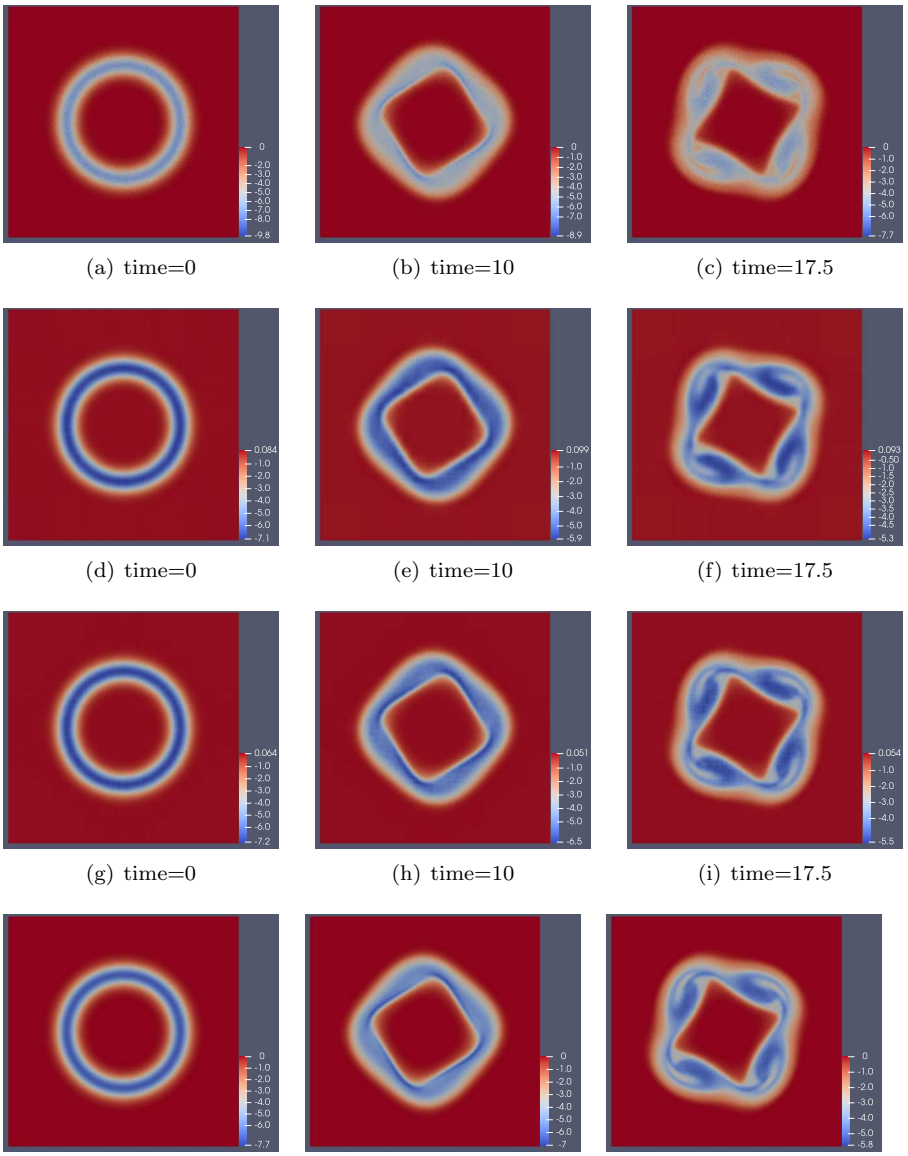
For denoising in equation (11), we take $\epsilon = \alpha\sqrt{(P_c)_{ref}/P_c}\max(|\hat{\rho}_e|)$ as explained in section 4.4.3, where $(P_c)_{ref} = 5$ and $\alpha = 0.01$. This means that with 5 particles per cell, charge densities with Fourier amplitude less than 1 percent of the maximum amplitude will be set to 0 and for other P_c the threshold will be scaled accordingly. The time step of the time integrator is chosen as $\Delta t = 0.02$ and the simulation is run till final time $T = 17.5$.

5.1.2. Qualitative comparison of charge density

Figure 2 shows the evolution of the electron charge density with time for regular, $\tau = 1$ and adaptive τ PIC for a 1024^2 mesh. For the first three rows $P_c = 5$ and for the last row $P_c = 80$. From the first and second rows we can see that while the regular PIC results are dominated by noise, $\tau = 1$ results are dominated by grid error due to the smearing of fine scale structures. This is also noted in [4] in their sparse PIC studies. In contrast, the adaptive τ results in the third row strikes a balance between the grid-based error and noise and are in close agreement (in visual norm) with the regular PIC results with high P_c in the fourth row.

5.1.3. Quantitative comparison of charge density

In order to make a quantitative comparison, in the left columns of Figures 3-5, the error in ρ_e calculated using (17) at 8 different points in time is shown for three different meshes $256^2, 512^2, 1024^2$ and number of particles per cell $P_c = 5, 10, 20$. For regular PIC we also carried out simulations at higher P_c, namely $40, 80, 160$ in order to compare the accuracy level with adaptive τ results. The reference in equation (17) is computed using the average of 8 independent regular PIC simulations each with a 1024^2 mesh and $P_c = 320$. In equation (17),



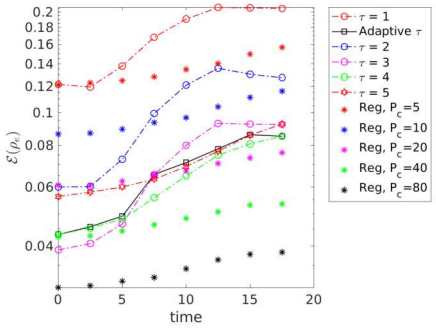
(a) time=0 (b) time=10 (c) time=17.5

(d) time=0 (e) time=10 (f) time=17.5

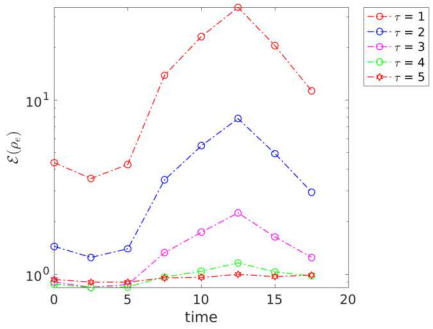
(g) time=0 (h) time=10 (i) time=17.5

(j) time=0 (k) time=10 (l) time=17.5

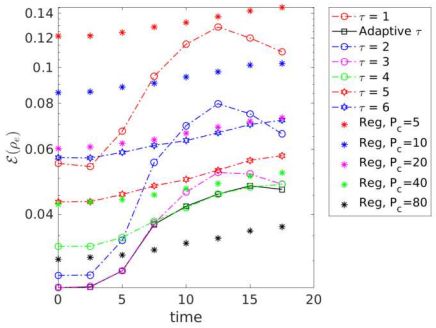
Figure 2: 2D diocotron instability: Evolution of the electron charge density with time for regular PIC, $P_c = 5$ (first row); $\tau = 1$, $P_c = 5$ (second row); adaptive τ, $P_c = 5$ (third row); and regular PIC, $P_c = 80$ (fourth row). The mesh considered here is 1024^2. The minimum and maximum values of the charge densities for each figure are displayed in the color bars itself.



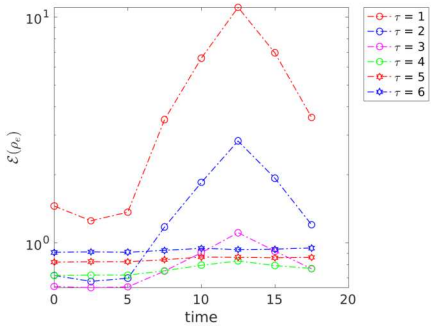
(a) 256^2, $P_c = 5$



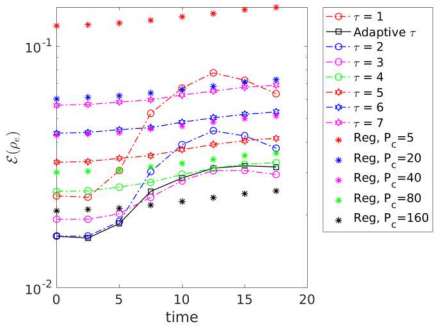
(b) 256^2, $P_c = 5$



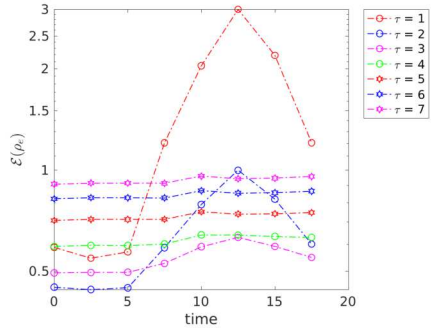
(c) 512^2, $P_c = 5$



(d) 512^2, $P_c = 5$

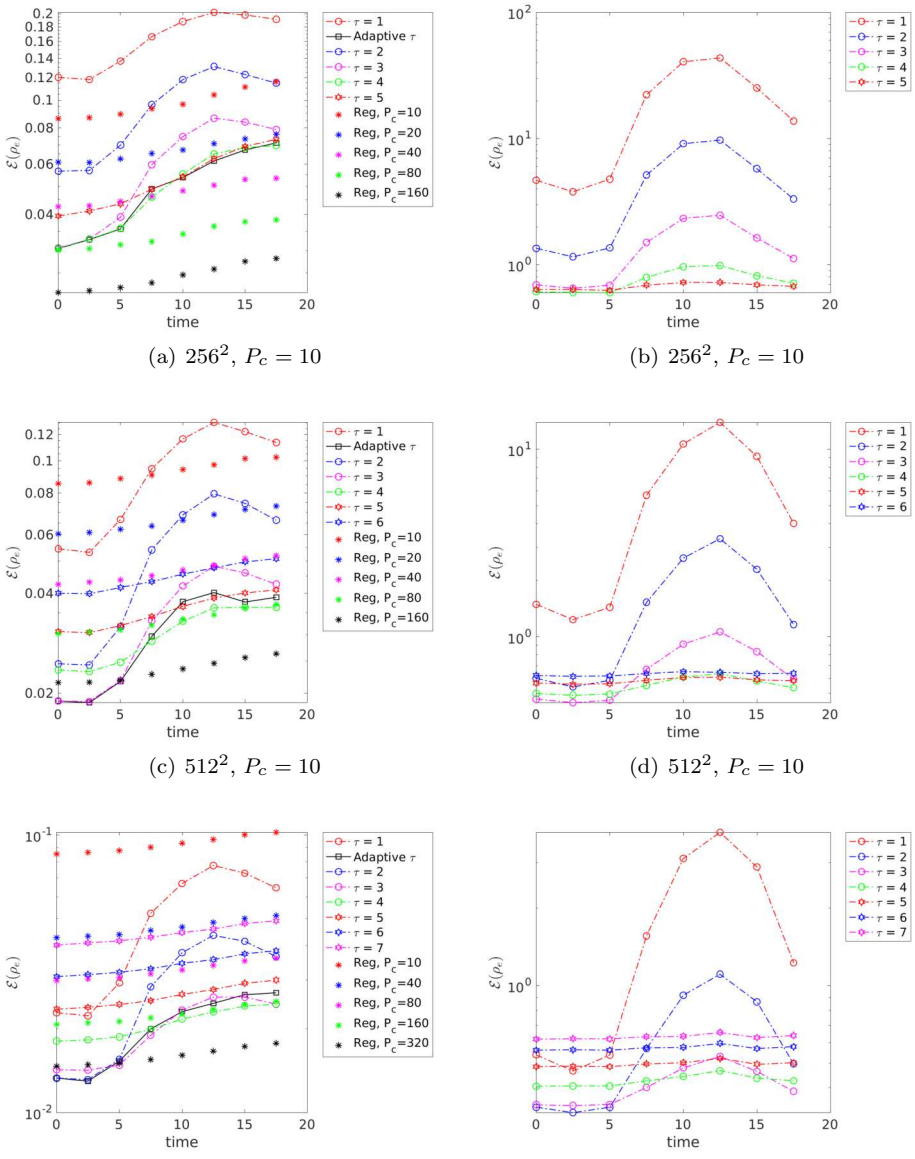


(e) 1024^2, $P_c = 5$



(f) 1024^2, $P_c = 5$

Figure 3: 2D diocotron instability: Electron charge density error comparison between regular (Reg), fixed τ and adaptive τ PIC. The left column is the actual error calculated using equation (17) and the right column is the estimations from the τ estimator based on which the optimal τ is selected. The fixed as well as adaptive τ has the number of particles per cell $P_c = 5$.

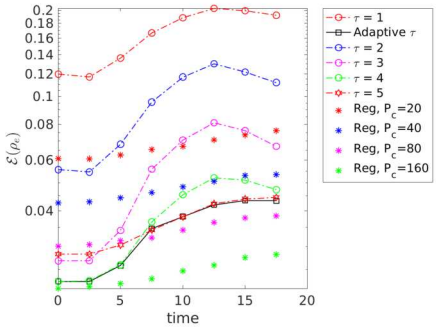


(a) 256^2, $P_c = 10$ (b) 256^2, $P_c = 10$

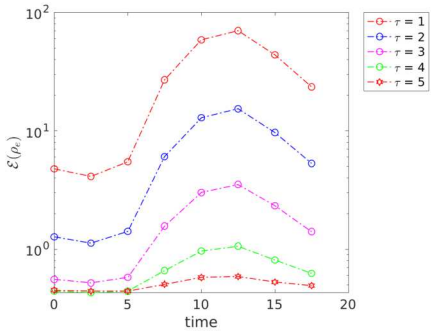
(c) 512^2, $P_c = 10$ (d) 512^2, $P_c = 10$

(e) 1024^2, $P_c = 10$ (f) 1024^2, $P_c = 10$

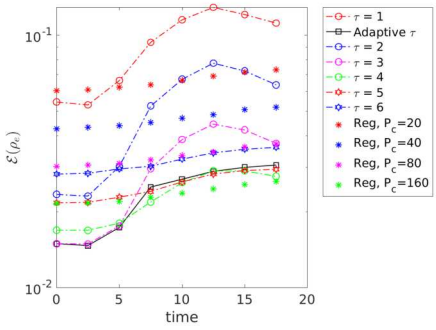
Figure 4: 2D diocotron instability: Electron charge density error comparison between regular (Reg), fixed τ and adaptive τ PIC. The left column is the actual error calculated using equation (17) and the right column is the estimations from the τ estimator based on which the optimal τ is selected. The fixed as well as adaptive τ has the number of particles per cell $P_c = 10$.



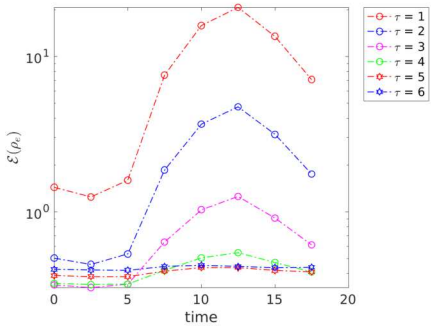
(a) 256^2, $P_c = 20$



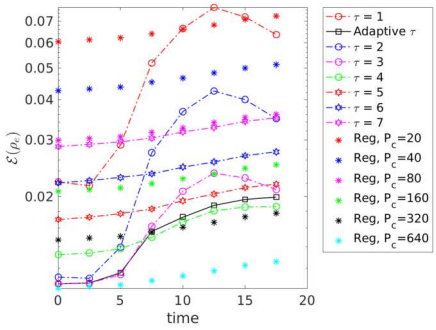
(b) 256^2, $P_c = 20$



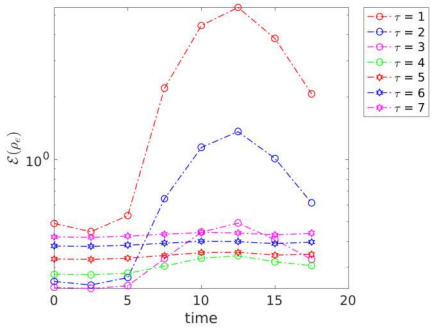
(c) 512^2, $P_c = 20$



(d) 512^2, $P_c = 20$



(e) 1024^2, $P_c = 20$



(f) 1024^2, $P_c = 20$

Figure 5: 2D diocotron instability: Electron charge density error comparison between regular (Reg), fixed τ and adaptive τ PIC. The left column is the actual error calculated using equation (17) and the right column is the estimations from the τ estimator based on which the optimal τ is selected. The fixed as well as adaptive τ has the number of particles per cell $P_c = 20$. The errors for regular PIC with $P_c = 320$ and 640 are calculated from that of $P_c = 160$ based on the theoretical particle error scaling $1/\sqrt{Pc}$. This is based on the observation that the errors for the regular PIC are in the noise dominated regime.

the N_{points} are taken as the cell-centered points in the mesh under consideration and the reference ρ_e is interpolated to these points for calculating error. In Figure 5(e), for calculating the error with regular PIC at $P_c = 320,640$ we divided the error for $P_c = 160$ by $\sqrt{2}$ and $\sqrt{4}$ respectively as we observed the errors are already in the noise dominated regime and follow the scaling $1/\sqrt{P_c}$. On the right columns of Figures 3-5 are the estimations of the error for different τ values from the τ estimator divided by the root mean squared value of the reference ρ_e. It is based on these curves that the optimal τ - i.e., the one with minimum error - is selected at each time step during the simulation.

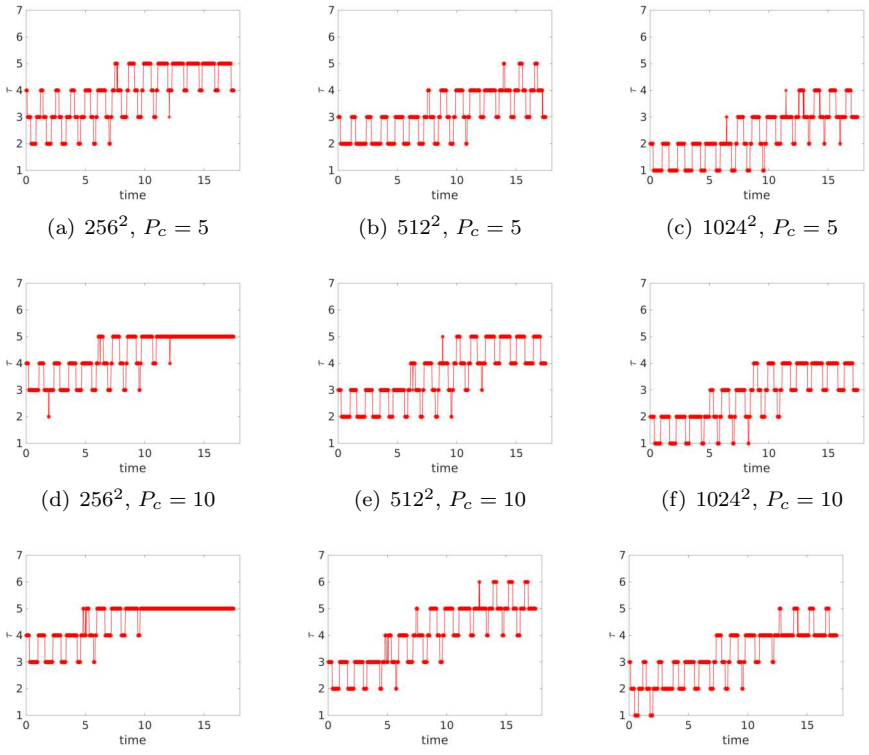
From the left columns of Figures 3-5, we can see that in general the adaptive τ performs well in terms of picking one of the τ values with the lowest error (if not the optimal τ at all points in time). The shapes of the error curves for individual τ values are also similar for the estimated and actual ones. It demonstrates the ability of our estimator to predict correct error dynamics for different τ cases. While we do not have to worry about the magnitude of the errors in the estimator, the ordering of the error curves between different τ values is of importance as it decides the optimal τ, and we want it to be close to the actual scenario on the left columns. To that extent, we make an observation that in the time interval $t \in [7.5, 17.5]$ the difference in the magnitude of errors between different τ values in the estimator differs more from the actual scenario than in the time interval $t \in [0, 7.5)$. More specifically, for low τ values ($\tau = 1, 2, 3$) the estimator predicts a significantly higher error compared to the other τ values in that regime.

One of the reasons for this behavior is for low τ cases - e.g., $\tau = 1, 2$ and 3 - the number of component grids in the combination technique is higher than that for the high τ cases. Since we use the triangle inequality to bound the errors, both the grid and particle errors tend to be more over-estimated for the low τ cases than those for the high τ ones. Another reason is, in the estimates for the grid error we use the derivatives based on the regular grid. While this is a sharper upper bound for high τ, the derivatives seen in reality by the low τ cases for functions with fine scale structures will be smaller because of the larger mesh sizes. Indeed, fine scale structures form in the time interval $t \in [7.5, 17.5]$ and hence grid error dominated for the simulations with sparse grid noise reduction.

In spite of these differences, in all the cases even with the predicted sub-optimal τ the error values of the adaptive τ PIC is significantly lower than that of the regular PIC with same P_c. If we use some problem specific information, then it may be possible to reduce the over-estimations in the grid and particle errors by introducing a correction factor for different τ values.

5.1.4. Evolution of τ with time

In Figure 6, the time history of τ is shown for the meshes and P_c considered in Figures 3-5. Here we can see that for the same P_c, when we decrease the mesh size - i.e., going from left to right in Figure 6 - the τ values decrease. This is because we are moving from the grid error dominated regime to the particle error dominated regime. On the other hand, for the same mesh size and increasing P_c - i.e., moving from top to bottom in Figure 6 - the τ values



(a) 256^2, $P_c = 5$ (b) 512^2, $P_c = 5$ (c) 1024^2, $P_c = 5$

(d) 256^2, $P_c = 10$ (e) 512^2, $P_c = 10$ (f) 1024^2, $P_c = 10$

(g) 256^2, $P_c = 20$ (h) 512^2, $P_c = 20$ (i) 1024^2, $P_c = 20$

Figure 6: 2D diocotron instability: Time history of τ for different mesh sizes and number of particles per cell P_c.

increase as we are moving from the particle error dominated regime to the grid error dominated regime. Also, for a particular mesh size and given P_c the later points in time have higher τ compared to the earlier ones. This is due to the formation of fine scale structures in the problem and resolving them require a higher τ.

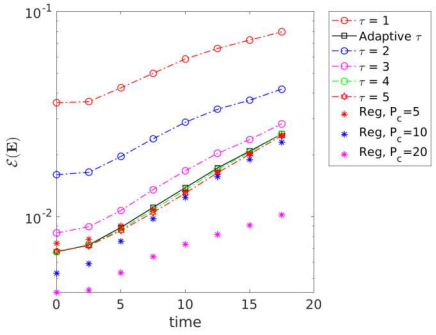
5.1.5. Quantitative comparison of electric field

In Figure 7, the error in the electric field $\mathbf{E}$ calculated using equation (17) is shown for the meshes[8] and P_c considered. We can see that the adaptive τ errors at the best are similar to the regular PIC and in some cases it is higher than regular PIC error for the same P_c. We also notice that none of the fixed τ error levels are better than the regular PIC errors. The reason for this is as follows: the electric field is obtained by integrating the charge density, and integration is a smoothing operation which reduces the particle noise. Since in our adaptive τ noise reduction algorithm we increase the grid-based error to reduce the particle noise and minimize the total error in the density, this can result in either similar or even an increase in the electric field error as compared to the regular PIC if the integration itself is sufficient enough to reduce the noise. High-order shape functions are a promising option to address this limitation as depending on the distribution they may reduce the particle noise without increasing the grid-based error. We will investigate the combination of high-order shape functions with our algorithm in future work.

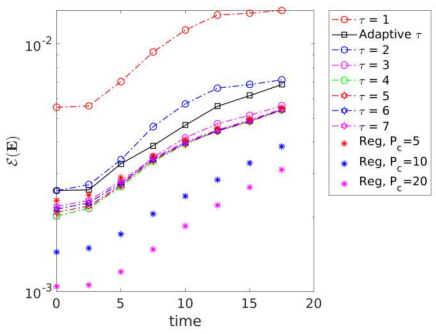
5.1.6. Adaptivity with initial sampling

Having studied the adaptivity of the algorithm with respect to mesh size, P_c and time, we also considered a different initial sampling technique, and evaluated the performance of our scheme. We do not show the results here in order to limit the already fairly large number of tables and figures in the article, but we briefly summarize our main observations. We used a uniform distribution in all the variables to sample f in equation (18). The range for the velocity variables was chosen as $[-6, 6]$ while for the configuration space it was $[0, L]$. Note that unlike the Gaussian sampling described earlier, with this sampling each particle will have a different constant charge q_e [12] to match the distribution. Still, the charge to mass ratio is the same for all the particles. Similar to [20], we ignored particles with weights less than 1.0×10^{-9}. For this particular example, uniform sampling is not a particularly good idea as it results in sampling particles which have very small computational weights. Hence, for the same total number of particles we found that this sampling has higher noise levels than the Gaussian sampling. Uniform sampling can however be useful in scenarios where we do not know of an importance sampling technique to sample the distribution at hand. Due to higher noise levels, we needed a higher value of $\alpha = 0.03$ for the calculation of the denoising threshold. Except for the coarsest mesh size 256^2,

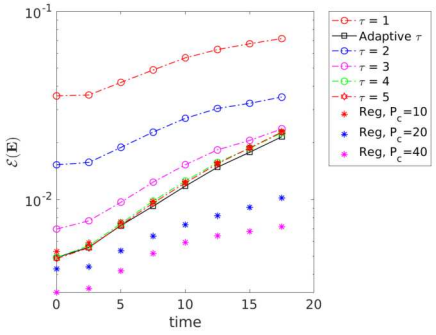
[8] For brevity we do not show results for a 512^2 mesh, as it does not contain much new and valuable information.



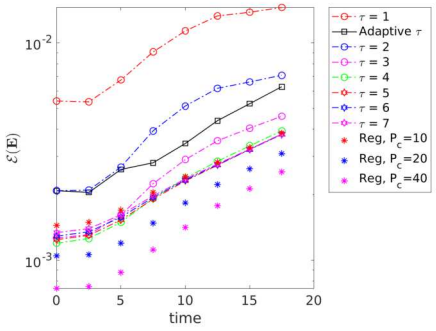
(a) 256^2, $P_c = 5$



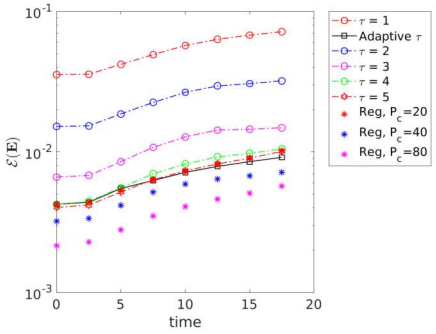
(b) 1024^2, $P_c = 5$



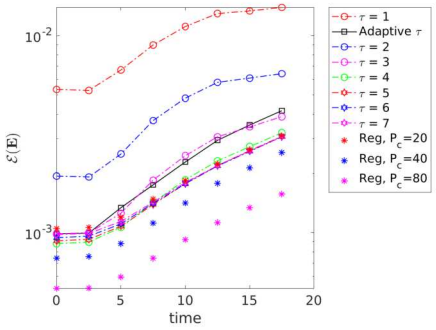
(c) 256^2, $P_c = 10$



(d) 1024^2, $P_c = 10$



(e) 256^2, $P_c = 20$



(f) 1024^2, $P_c = 20$

Figure 7: 2D diocotron instability: Electric field error comparison between regular (Reg), fixed τ and adaptive τ PIC.

the adaptive τ algorithm performed well in this sampling - i.e., the scheme picked a nearly optimal τ for most cases. The optimal τ values, as expected, are lower than that for the Gaussian sampling, owing to higher noise levels.

5.1.7. Run time performance

Mesh	P_c		
	5	10	20
256^2	43.7	47.5	55.4
512^2	91.9	110.1	145.8
1024^2	435	501.5	653.8

Table 1: 2D diocotron instability. Adaptive τ PIC: Total run time in seconds on 64 cores for different mesh sizes and number of particles per cell.

Mesh	Regular PIC			Reg/adaptive τ		
256^2	50.9 (20)	50.9 (20)	104.4 (80)	1.2	1.1	1.9
512^2	201.7 (40)	364.2 (80)	708.6 (160)	2.2	3.3	4.9
1024^2	1544.5 (80)	2911.6 (160)	5857.3 (320)	3.5	5.8	9.0

Table 2: 2D diocotron instability: Columns $2-4$ are the total run time in seconds taken by the regular PIC on 64 cores for different mesh sizes and number of particles per cell (within parentheses) to reach a comparable accuracy (based on visual norm from the left columns of Figures 3-5) in charge density that of the adaptive τ results in Table 1 at time $T = 17.5$. Columns $5-7$ are the ratio of time taken by regular PIC to the values in columns $2-4$ of Table 1 for adaptive τ PIC.

Mesh	tauEstimator			transferToSparse		
	P_c			P_c		
	5	10	20	5	10	20
256^2	10.1	10.7	11.4	5.9	5.7	4.9
512^2	21	20.2	18.6	5.4	4.5	3.6
1024^2	40.1	36.1	30.2	2.7	2.3	1.8

Table 3: 2D diocotron instability. Adaptive τ PIC: Percentage of total time (which is shown in Table 1) taken by the tauEstimator and transferToSparse parts of the noise reduction strategy for different mesh sizes and number of particles per cell.

Finally, we perform a preliminary run time performance study to see the effectiveness of the current approach in comparison to the regular PIC. To that extent, we note that we did not perform any optimization to both the regular PIC as well as the adaptive τ PIC routines. Optimization of different components involved in the algorithm as well as a thorough parallel performance study is left for future work. In Table 1 the total run time in seconds is shown for the adaptive τ PIC on 64 cores for the mesh sizes and P_c considered before. All the timings reported are the average of three runs performed. In Table 2, we compare the adaptive τ PIC timings with the timings for the regular PIC with the

P_c value required to reach a comparable accuracy in charge density as that of the adaptive τ results at final time $T = 17.5$. The approximate P_c values within parentheses are obtained from Figures 3-5 based on visual examination. Even in this preliminary performance study, we can see that the adaptive τ strategy can provide significant speedups close to an order of magnitude compared to the regular PIC for similar accuracy in charge density. In terms of memory storage, the benefits are even more pronounced. Using the number of particles N_p as a measure of the dominant memory cost (for PIC methods this is usually the case) we see $\approx 2 - 16$ times memory reduction with adaptive τ PIC compared to regular PIC. In Table 3, we present timings for the components of the noise reduction only, expressed as percentage of the total time given in Table 1. Even though the percentage of time taken by the transferToSparse part is small, the tauEstimator represents a significant fraction of the total time. One of the reasons for this is that for the FFT parts of the tauEstimator algorithm (Algorithm 1) we use the OPAL library. Since our other data structures are based on the AMReX library, we have to copy between them. Since the parallel decomposition is different for these two libraries, it can result in excessive communication, especially for large numbers of grid points and for high core counts. We are currently resolving this problem in the ongoing implementation of our noise reduction strategy in OPAL, using only OPAL's native data structures and thereby avoiding the copy and excessive communication.

5.2. 3D Penning trap

5.2.1. Problem description and simulation setup

In this section we will consider a 3D Penning trap problem as the test case. Penning traps are storage devices for charged particles, which uses a quadrupole electric field to confine the particles axially and a homogeneous axial magnetic field to confine the particles in the radial direction [54, 55]. The evolution of the density in this problem (see Figure 8) is very similar to that observed in cyclotrons [56, 57]. Thus this test case is very relevant to our ultimate goal of high precision large-scale simulation of cyclotrons. The fine scale structures developed in this problem pose challenges for the sparse grids similar to the diocotron case in the previous section.

The parameters for this test case are as follows. The length of the periodic box is $L = 20$. The external magnetic field is given by $\mathbf{B}_{ext} = \{0, 0, 5\}$ and the quadrupole external electric field by

$$\mathbf{E}_{ext} = \left\{ -\frac{15}{L}\left(x - \frac{L}{2}\right), -\frac{15}{L}\left(y - \frac{L}{2}\right), \frac{30}{L}\left(z - \frac{L}{2}\right) \right\}.$$

For the initial conditions, we sample the phase space using a Gaussian distribution in all the variables. The mean and standard deviation for all the velocity variables is 0 and 1 respectively. While the mean for all the configuration space variables is $L/2$ the standard deviations are $0.15L$, $0.05L$ and $0.2L$ for x, y and z respectively. The total electron charge is $Q_e = -1562.5$, and the charge of each particle is $q_e = \frac{Q_e}{N_p}$.

The denoising parameters are taken as $(P_c)_{ref} = 1$ and $\alpha = 0.005$ for this problem with the above mentioned sampling. The time step is chosen as $\Delta t = 0.05$ and the simulations are run till final time $T = 15$.

5.2.2. Qualitative comparison of charge density

Figure 8 shows the evolution of the electron charge density with time for regular, $\tau = 1$ and adaptive τ PIC. The mesh used is 256^3 and $P_c = 1$ for the first three rows and 20 for the last row. As we had seen in Figure 2 for the diocotron test case, the adaptive τ results, in the third row are better than both the regular PIC and $\tau = 1$ results and are comparable to the results of the regular PIC with higher P_c in the last row.

5.2.3. Quantitative comparison of charge density and time history of τ

In a way analogous to Figures 3-5 for the diocotron instability, in Figures 9-10 we show the errors calculated using equation (17) and the estimations from the τ estimator for meshes $64^3, 128^3, 256^3$ and $P_c = 1, 5$. The reference in equation (17) is the average of 5 independent computations of regular PIC with 256^3 mesh and $P_c = 40$. For the N_{points} in equation (17), we select approximately 4096 random points throughout the domain and interpolate both the reference density as well as the density under consideration at these points to measure the error. The errors are measured at 7 different points in time in the simulation.

In general, as before, the adaptive τ predictions are close to optimal and most of the conclusions from the diocotron test case are applicable in this case too. Figure 11 shows the time history of τ for the meshes and P_c considered and the high values of τ indicate that the total error is dominated by the grid-based error in these cases.

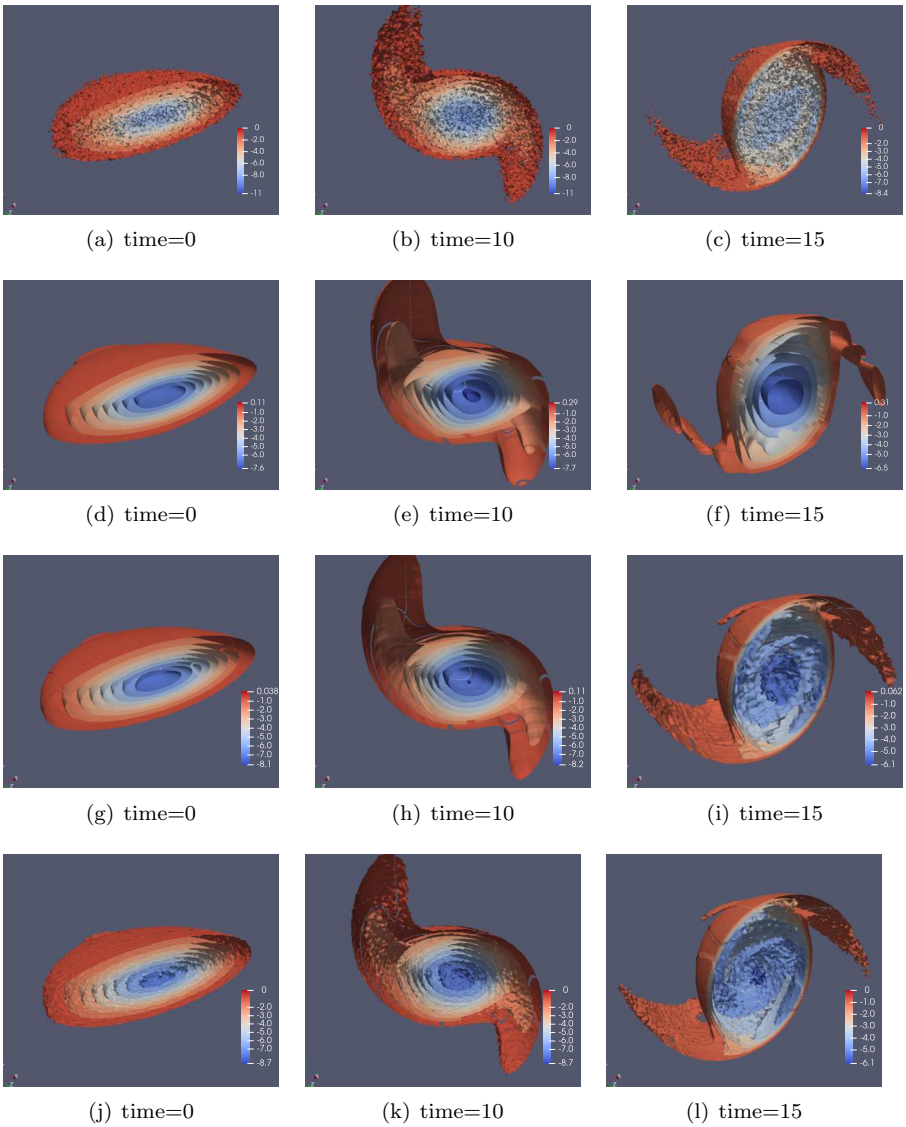
5.2.4. Run time performance

In terms of run time performance comparisons, we ran the $64^3, 128^3$ mesh cases on 64 cores and the 256^3 test cases on 512 cores for both the regular and adaptive τ PIC. For 64^3 mesh, at the last point in time we can see that the regular PIC is more accurate than the adaptive τ or any other fixed τ PIC.

	Adaptive τ		Regular		Reg/adaptive τ	
128^3	360.4 (1)	475.4 (5)	274.8 (5)	443.7 (10)	0.8	0.9
256^3	825.5 (1)	1196.4 (5)	2352.8 (15)	3080.8 (20)	2.8	2.6

Table 4: 3D Penning Trap: Total run time in seconds on 64 cores for 128^3 mesh and 512 cores for 256^3 mesh in case of the regular and adaptive τ PIC. The values within the parentheses represent the different number of particles per cell required to reach a comparable accuracy (based on visual norm from the left columns of Figures 9-10) in the charge density for both the schemes at final time $T = 15$. Columns $6 - 7$ are the ratio of time taken by the regular PIC to that for adaptive τ PIC.

For 128^3 and 256^3 meshes, from Table 4 we can see a maximum speedup of 2.8 with adaptive τ PIC over the regular PIC for the finest mesh size. Again considering the number of particles as a measure for the memory cost adaptive



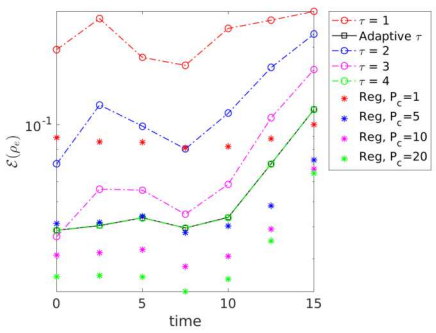
(a) time=0 (b) time=10 (c) time=15

(d) time=0 (e) time=10 (f) time=15

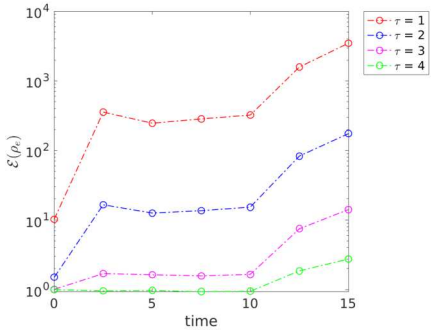
(g) time=0 (h) time=10 (i) time=15

(j) time=0 (k) time=10 (l) time=15

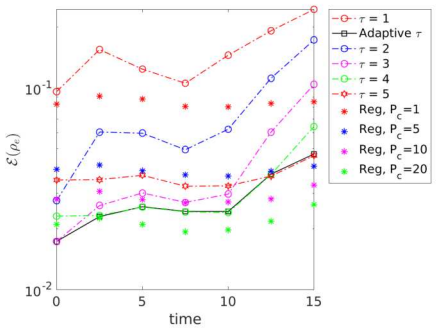
Figure 8: 3D Penning trap: Evolution of the electron charge density with time for regular PIC, $P_c = 1$ (first row); $\tau = 1$, $P_c = 1$ (second row); adaptive τ, $P_c = 1$ (third row); and regular PIC, $P_c = 20$ (fourth row). The mesh considered here is 256^3. The minimum and maximum values of the charge densities for each figure are displayed in the color bars itself.



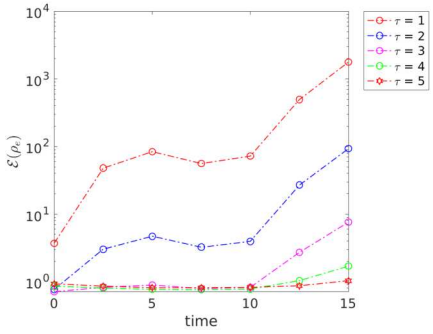
(a) 64^3, $P_c = 1$



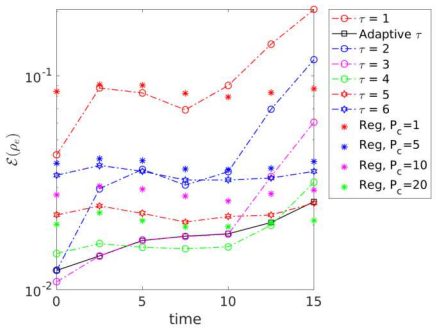
(b) 64^3, $P_c = 1$



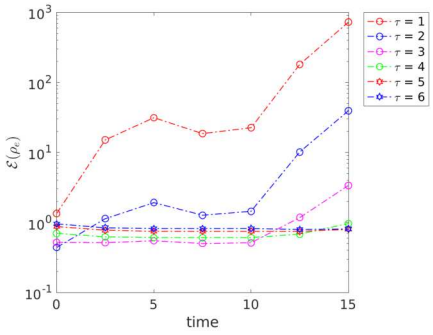
(c) 128^3, $P_c = 1$



(d) 128^3, $P_c = 1$

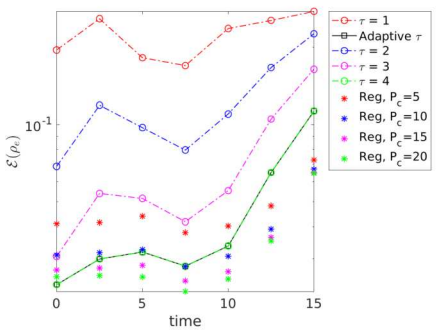


(e) 256^3, $P_c = 1$

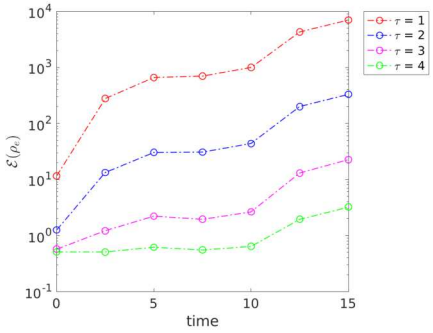


(f) 256^3, $P_c = 1$

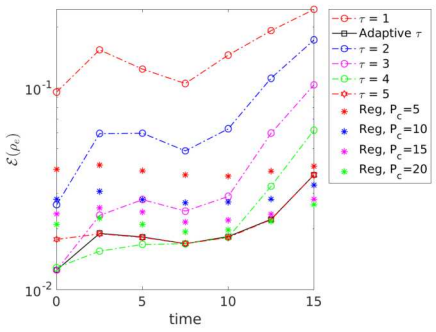
Figure 9: 3D Penning trap: Electron charge density error comparison between regular (Reg), fixed τ and adaptive τ PIC. The left column is the actual error calculated using equation (17) and the right column is the estimations from the τ estimator based on which the optimal τ is selected. The fixed as well as adaptive τ has the number of particles per cell $P_c = 1$.



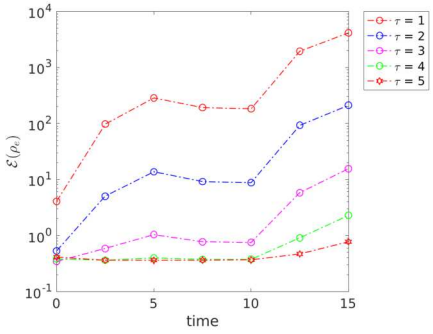
(a) 64^3, $P_c = 5$



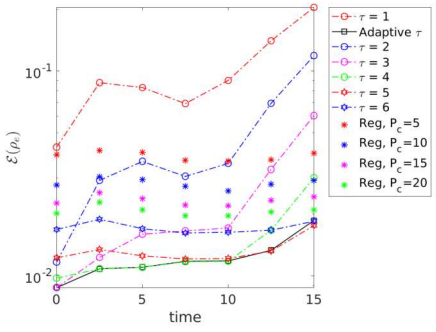
(b) 64^3, $P_c = 5$



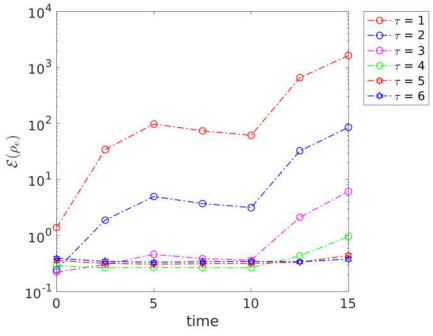
(c) 128^3, $P_c = 5$



(d) 128^3, $P_c = 5$

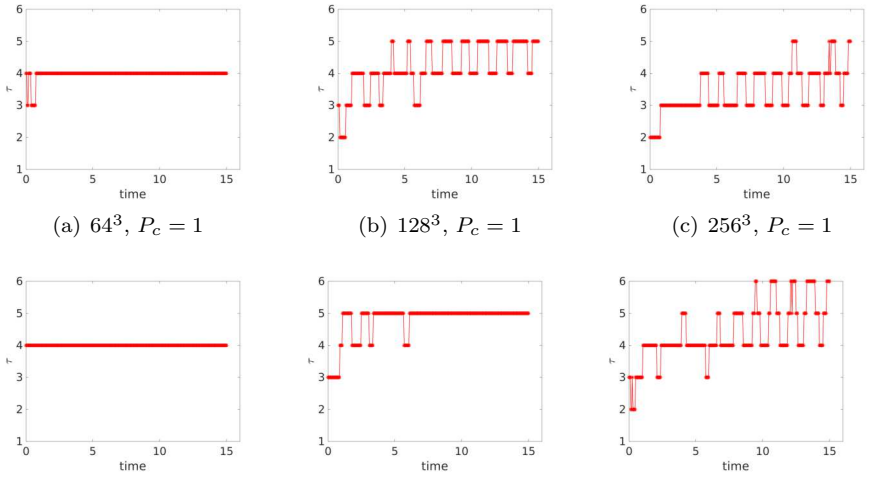


(e) 256^3, $P_c = 5$



(f) 256^3, $P_c = 5$

Figure 10: 3D Penning trap: Electron charge density error comparison between regular (Reg), fixed τ and adaptive τ PIC. The left column is the actual error calculated using equation (17) and the right column is the estimations from the τ estimator based on which the optimal τ is selected. The fixed as well as adaptive τ has the number of particles per cell $P_c = 5$.



(a) 64^3, $P_c = 1$ (b) 128^3, $P_c = 1$ (c) 256^3, $P_c = 1$

(d) 64^3, $P_c = 5$ (e) 128^3, $P_c = 5$ (f) 256^3, $P_c = 5$

Figure 11: 3D Penning trap: Time history of τ for different mesh sizes and number of particles per cell P_c.

Mesh	tauEstimator		transferToSparse	
	P_c		P_c	
	1	5	1	5
128^3	55.4	39.9	4.6	3.3
256^3	41.8	29.5	15.3	9.3

Table 5: 3D Penning Trap: Percentage of total time (which is shown in columns $2-3$ of Table 4) taken by the tauEstimator and transferToSparse parts of the noise reduction strategy for different mesh sizes and number of particles per cell.

τ PIC is $2-15$ times cheaper than the regular PIC. In order to see more computational benefits with the adaptive τ PIC for this problem we need to perform runs with finer meshes and more particles per cell. These 3D large-scale simulations are part of our future work and the results will be reported elsewhere.

In Table 5, we show the percentage of the total time taken by the components of the noise reduction algorithm. Similar to the diocotron instability example, we can see that the dominant portion comes from the tauEstimator, for the same reasons as in the two-dimensional example. In addition, transferToSparse also exhibits an increase in percentage compared to the previous example. This is due to the bottleneck with MPI_Allreduce for high τ values in 3D as described in section 4.5. In future work, we will adopt an improved parallelization strategy as in [40], which can mitigate this problem. Furthermore, the optimal τ does not need to be calculated for each time step. If the time-step is small, the charge density will not change much in a single time-step. The optimal τ, being only

dependent on ρ_e, is therefore also unlikely to change much. One could thus get speed-up by only recomputing τ every 5th or 10th time-step, for instance, while still accurately estimating the optimal τ. This is borne out in Figures 6 and 11, where τ stays fixed for many consecutive time-steps. We will also investigate this aspect in detail in future work.

6. Conclusions

We have proposed a sparse grid-based adaptive noise reduction strategy for particle-in-cell (PIC) simulations. Unlike the typical physical or Fourier domain filters used in PIC methods, the strategy adapts to mesh size, number of particles per cell, smoothness of the charge density and the initial sampling technique. In order to construct the strategy we use the key idea of increased particles per cell in sparse grids compared to the regular grid for the same total number of particles as proposed in [4]. The current work extends that concept in several directions. Specifically, we present a filtering perspective for the sparse grid-based noise reduction which helps to incorporate it with ease in existing high performance large-scale PIC code bases and also opens the door for sparse grid based filtering approaches. We tackle the problem of large grid-based error of sparse grid for non-aligned and non-smooth functions by means of the truncated combination technique [1, 2, 3]. We show in the context of PIC simulations that the truncated combination technique provides a natural framework to minimize the sum of grid-based error and particle noise. This allows us to propose a heuristic based on formal error analysis to select the optimal truncation parameter on the fly that minimizes the total error in the charge density.

We show the performance and applicability of our strategy on two benchmark problems; namely the 2D diocotron instability and electron dynamics in a 3D Penning trap. In both test cases the adaptive noise reduction strategy picks a truncation parameter which is close to optimal for all times. To achieve comparable accuracy for the charge density we obtain significant speedups and memory savings close to an order of magnitude with the noise reduction technique compared to regular PIC in the 2D diocotron test case. For the 3D Penning trap test case a maximum speedup of 2.8 and 15 times memory reduction is obtained for the finest mesh size tested. Further speedups and memory reduction in the 3D test case require us to test the strategy for even finer resolutions and that is part of future work.

Our strategy can be very easily integrated into existing high performance large-scale PIC code bases and ongoing work is to integrate it into the open source particle accelerator library OPAL [37]. In terms of future work, we plan on investigating the applicability and performance of the noise reduction strategy on large-scale high intensity particle accelerator simulations such as the IsoDAR project [58, 59] with a particular focus on understanding the dynamics of halo particles and efficient collimation strategies. Filtering strategies have much more impact on the electromagnetic PIC simulations as reported in [24].

Hence we would like to extend the current approach for Vlasov-Maxwell equations and investigate the performance in that context. Use of machine learning approaches to tune denoising threshold in our strategy is also of interest. Currently, we are unable to use the full range of truncation parameter τ due to the false optima obtained when the extreme values are included. We will work on strategies in the τ estimation to resolve this problem. Finally, we also intend to compare the current strategy with other filtering approaches and denoising techniques.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 701647 and from the United States National Science Foundation under Grant No. PHY-1820852. L.F. Ricketson's work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC. We are grateful for the support. The first author would like to thank Dr. Weiqun Zhang for help with the AMReX related queries. We thank the anonymous referees for their useful comments which improved this paper substantially.

Appendix A. Proof of Proposition 1 relating the direct charge density deposition onto the component grids and the two-step approach

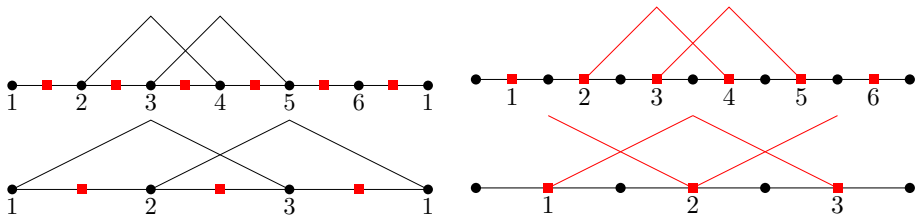
Proof. Even though sparse grids make sense only for dimensions 2 and higher we can still understand the essence of the proof in 1D. Also, since the shape functions and transfer operators in 2D and 3D are obtained by the tensor product of 1D linear interpolation functions the proof extends naturally to those cases.

Consider a periodic 1D domain $[0, L]$ and two grids with mesh sizes h_f and h_l. The grid with mesh size h_l is coarser than the one with h_f and assume h_l is an integer multiple of h_f. Let us first consider the node-centered grids where all the coarse grid points are also grid points in the fine grid as shown in Figure 12(a).

The particles deposit onto the fine grid with mesh size h_f and the charge density $\tilde{\rho}_e$ is given by

$$\tilde{\rho}_e(\tilde{x}_j) = \frac{Q_e}{N_p h_f} \sum_{p=1}^{N_p} W_f(\tilde{x}_j - x_p), \tag{19}$$

where $W_f(\zeta) = \max\left\{0, 1 - \frac{|\zeta|}{h_f}\right\}$ is the cloud-in-cell shape function and x_p and $\tilde{x}_j$ are the locations of the particles and the grid points in the fine grid respectively. Now, we transfer the density $\tilde{\rho}_e$ to the coarse grid by means of the



(a) Node-centered grid (b) Cell-centered grid

Figure 12: Schematic showing the node-centered and cell-centered grids and the corresponding shape functions. The nodes are marked with black circles and the cell-centers with red squares. The domain is periodic. The shape functions W_l corresponding to the coarse grid are linear between the nodes in the fine grid in case of node-centered grids. For cell-centered grids W_l has discontinuity in derivative between some of the cell-centers in the fine grid whereas between nodes of the fine grid it is always linear.

transfer operator R in equation (4) which gives

$$\varrho_e(x_k) = \frac{h_f}{h_l} \sum_{j=1}^{N_c} \tilde{\rho}_e(\tilde{x}_j) W_l(x_k - \tilde{x}_j), \tag{20}$$

where $W_l(\zeta) = \max\left\{0, 1 - \frac{|\zeta|}{h_l}\right\}$, x_k are the locations of the grid points in the coarse grid and N_c is the total number of cells in the fine grid. Substituting for $\tilde{\rho}_e$ from equation (19) and switching the order of sums we get

$$\varrho_e(x_k) = \frac{Q_e}{N_p h_l} \sum_{p=1}^{N_p} \sum_{j=1}^{N_c} W_l(x_k - \tilde{x}_j) W_f(\tilde{x}_j - x_p). \tag{21}$$

Now, for a given particle, $W_f(\tilde{x}_j - x_p)$ is non-zero for exactly two values of j: the floor of x_p/h_f and the ceiling of that same quantity. Let us call these values J and $J+1$ and assume the grid points are ordered such that x_J is to the left of x_{J+1}. We have

$$\begin{aligned}\sum_{j=1}^{N_c} W_l(x_k - \tilde{x}_j) W_f(\tilde{x}_j - x_p) = W_l(x_k - \tilde{x}_J) W_f(\tilde{x}_J - x_p) \\ + W_l(x_k - \tilde{x}_{J+1}) W_f(\tilde{x}_{J+1} - x_p).\end{aligned} \tag{22}$$

Now we note that because of the way the two grids are related (mesh sizes are integer multiples, coincident grid points), we are guaranteed that $W_l(x_k - \tilde{x})$ is linear on the interval $\tilde{x} \in [\tilde{x}_J, \tilde{x}_{J+1}]$. This is because the places where W_l has a discontinuity in its derivative are guaranteed to be fine grid points as shown in Figure 12(a). So, linear interpolation is exact for W_l on the interval $[\tilde{x}_J, \tilde{x}_{J+1}]$. Since x_p is in this interval, we have

$$W_l(x_k - x_p) = W_l(x_k - \tilde{x}_J) \left[\frac{\tilde{x}_{J+1} - x_p}{\tilde{x}_{J+1} - \tilde{x}_J}\right] + W_l(x_k - \tilde{x}_{J+1}) \left[1 - \frac{\tilde{x}_{J+1} - x_p}{\tilde{x}_{J+1} - \tilde{x}_J}\right].$$

Now we notice that

$$\left[\frac{\tilde{x}_{J+1}-x_p}{\tilde{x}_{J+1}-\tilde{x}_J}\right] = \left[\frac{\tilde{x}_J+h_f-x_p}{h_f}\right] = 1+\frac{\tilde{x}_J-x_p}{h_f} = 1-\frac{|\tilde{x}_J-x_p|}{h_f} = W_f(\tilde{x}_J-x_p),$$

and a nearly identical reasoning gives

$$\left[1-\frac{\tilde{x}_{J+1}-x_p}{\tilde{x}_{J+1}-\tilde{x}_J}\right] = W_f(\tilde{x}_{J+1}-x_p).$$

Combining these with equation (22) we get

$$\sum_{j=1}^{N_c} W_l(x_k-\tilde{x}_j)W_f(\tilde{x}_j-x_p) = W_l(x_k-x_p). \tag{23}$$

Substituting this into equation (21) we get the density on the coarse grid as

$$\varrho_e(x_k) = \frac{Q_e}{N_p h_l}\sum_{p=1}^{N_p} W_l(x_k-x_p). \tag{24}$$

Comparing equation (24) with equation (19) we see this is exactly the expression we would obtain if the particles were to deposit directly onto the coarse grid with mesh size h_l.

Now we will consider the cell-centered grids. In this case the coarse grid points are also not the grid points in the fine grid and W_l will have a discontinuity in the derivative for some of the intervals $[\tilde{x}_j, \tilde{x}_{j+1}]$ as shown in Figure 12(b) depending on the ratio h_l/h_f. Hence an exact equivalence between the two approaches does not hold. However, we will now show that

$$\mathcal{W}_l(x_k-x) = \sum_{j=1}^{N_c} W_l(x_k-\tilde{x}_j)W_f(\tilde{x}_j-x)$$

can be considered as a shape function by itself. To that end, we will show that it satisfies the three conditions for any shape function as given in [9]. These are listed as follows

1. $\mathcal{W}_l(\zeta) = \mathcal{W}_l(-\zeta)$,
2. $\frac{1}{h_l}\int \mathcal{W}_l(\zeta)d\zeta = 1$,
3. $\sum_k \mathcal{W}_l(x_k-x) = 1$.

The first condition is manifestly true as W_f which is the standard hat function is even. For the second condition we observe that

$$\frac{1}{h_l}\int \mathcal{W}_l(\zeta)d\zeta = \frac{h_f}{h_l}\sum_{j=1}^{N_c} W_l(x_k-\tilde{x}_j),$$

as W_f is a shape function and by definition integrates to h_f. Now, $h_f \sum_{j=1}^{N_c} W_l(x_k - \tilde{x}_j)$ is the midpoint rule applied for the integration $\int W_l(x_k - \tilde{x})$ over the fine grid. From Figure 12(b) it is clear that W_l is linear on each integration cell and the midpoint rule is exact. Thus,

$$\frac{h_f}{h_l} \sum_{j=1}^{N_c} W_l(x_k - \tilde{x}_j) = \frac{1}{h_l} \int W_l(x_k - \tilde{x}) d\tilde{x} = 1,$$

where the last step comes from the fact that W_l which is also a standard hat function integrates to h_l by definition. Finally, the third condition is related to global charge conservation and we note that since W_l and W_f are standard hat functions they satisfy the partition of unity and hence $\mathcal{W}_l$ also satisfies it when we carry out the summation.

Now, using conditions 1 and 2 and noting that $\mathcal{W}_l$ is bounded in $[0, L]$ we can carry out the same set of steps shown in appendix B for a standard hat function. We can then see the grid-based error for $\mathcal{W}_l$ is of $\mathcal{O}(\left|\partial_x^2 \rho_e\right| h_l^2)$ and the particle noise is $\mathcal{O}(\sqrt{|Q_e \rho_e| / N_p h_l})$ as in equations (35) and (51) but with the constants depending on the ratio of h_l to h_f. □

Appendix B. Grid-based and particle errors in the charge density deposition for regular PIC schemes

In this section, we follow the analysis in [4] and derive in details the grid-based error and noise estimates for the charge density deposition in regular PIC schemes explicitly revealing the constants. For simplicity, let us consider a 1D PIC scheme and extensions to 2D and 3D are relatively straightforward. In the following, we consider a particular point in time and hence suppress the dependence of the different quantities with respect to time.

Let $f(x, v)$ be the electron phase-space distribution under consideration and let us define $\bar{f}$ as

$$\bar{f} = \frac{f}{\int \int f dx dv}.$$

Since, $\bar{f}$ is non-negative and its phase-space integral is unity it can be interpreted as probability density. The exact charge density $\rho_e(x)$ is given by

$$\rho_e(x) = q_e \int \int f(\xi, v) \delta(x - \xi) d\xi dv, \tag{25}$$

$$= q_e \left(\int \int f dx dv \right) \int \int \bar{f}(\xi, v) \delta(x - \xi) d\xi dv, \tag{26}$$

$$= Q_e \int \int \bar{f}(\xi, v) \delta(x - \xi) d\xi dv, \tag{27}$$

where $Q_e = q_e \int \int f dx dv$ is the total electron charge in the system and $\delta(x - \xi)$ is the Dirac-delta function.

In PIC, we approximate $\delta(x-\xi)$ with the shape function $S(x-\xi)$ which for our discussion here consider it to be the cloud-in-cell or linear interpolation function. The approximate charge density $\bar{\rho}_e$ with the shape function $S(x-\xi)$ is given by

$$\bar{\rho}_e(x) = Q_e \int\int \bar{f}(\xi, v) S(x-\xi) d\xi dv, \tag{28}$$

$$= Q_e \mathbb{E}_{\bar{f}(\xi,v)} \left[S(x-\xi) \right], \tag{29}$$

where $\mathbb{E}$ is the expected value over the probability density $\bar{f}$.

B.1 Grid-based error estimate

This is the error due to approximating $\delta(x-\xi)$ with the shape function $S(x-\xi)$

$$e_g = \left| \rho_e - \bar{\rho}_e \right|. \tag{30}$$

Towards estimating this error, let us expand $\bar{f}(\xi, v)$ in equation (28) in terms of Taylor's series about x,

$$\begin{aligned} \bar{\rho}_e = Q_e \int\int \Big(& \bar{f}(x,v) + (\xi - x)\partial_x \bar{f}(x,v) \\ & + \frac{(\xi-x)^2}{2} \partial_x^2 \bar{f}(x,v) + \cdots \Big) S(x-\xi) d\xi dv, \end{aligned} \tag{31}$$

$$= \underbrace{Q_e \int \bar{f} dv}_{\rho_e} \underbrace{\int S(x-\xi) d\xi}_{1} + Q_e \int \partial_x \bar{f} dv \int (\xi - x) S(x-\xi) d\xi$$

$$+ Q_e \int \partial_x^2 \bar{f} dv \int \frac{(\xi-x)^2}{2} S(x-\xi) d\xi + \cdots, \tag{32}$$

where we have used the fact that the integral of the shape function $S(x-\xi)$ is unity. In the above equations we have used the short hand notations $\partial_x = \frac{\partial(.)}{\partial x}$ and $\partial_x^2 = \frac{\partial^2(.)}{\partial x^2}$. Taking outside the partial derivatives with respect to x in the $\int dv$ integrals we get

$$\bar{\rho}_e = \rho_e + \partial_x \rho_e \int (\xi - x) S(x-\xi) d\xi + \partial_x^2 \rho_e \int \frac{(\xi-x)^2}{2} S(x-\xi) d\xi + \cdots. \tag{33}$$

The cloud-in-cell shape function is given by

$$S(\zeta) = \frac{1}{h_x} \max \left\{ 0, 1 - \frac{|\zeta|}{h_x} \right\}. \tag{34}$$

Performing a change of variables with $\zeta = \xi - x$ in equation (33) and noting that $S(\zeta)$ has a compact support and is zero outside $|\zeta| \leq h_x$ all the integrals has to be carried only in $-h_x \leq \zeta \leq h_x$.

Also, $S(\zeta)$ is an even function and hence $\int_{-h_x}^{h_x} \zeta S(\zeta) d\zeta$ which is the second term in equation (33) is 0. However, the integrand in the third term of the equation (33) is an even function and it evaluates to

$$\int \frac{(\xi - x)^2}{2} S(x - \xi) d\xi = \frac{1}{h_x} \int_0^{h_x} \zeta^2 \left(1 - \frac{\zeta}{h_x}\right) d\zeta = \frac{h_x^2}{12}.$$

Thus equation (30) becomes

$$\begin{aligned} e_g(x) &\leq \frac{h_x^2}{12} \left|\partial_x^2 \rho_e(x)\right| + \cdots, \\ e_g &= \mathcal{O}\left(\frac{h_x^2}{12} \left|\partial_x^2 \rho_e(x)\right|\right). \end{aligned} \tag{35}$$

Since, the cloud-in-cell shape functions in 2D and 3D are obtained by the tensor product of 1D shape functions the analysis extends easily to these cases. Carrying out similar steps we obtain the grid-based error for 2D and 3D as

$$e_g = \mathcal{O}\left(\frac{1}{12}\left\{\left|\frac{\partial^2 \rho_e}{\partial x^2}\right| h_x^2 + \left|\frac{\partial^2 \rho_e}{\partial y^2}\right| h_y^2\right\} + \frac{1}{144}\left|\frac{\partial^4 \rho_e}{\partial x^2 \partial y^2}\right| h_x^2 h_y^2\right) \quad \text{in 2D,} \tag{36}$$

$$\begin{aligned} e_g = \mathcal{O}\Bigg(&\frac{1}{12}\left\{\left|\frac{\partial^2 \rho_e}{\partial x^2}\right| h_x^2 + \left|\frac{\partial^2 \rho_e}{\partial y^2}\right| h_y^2 + \left|\frac{\partial^2 \rho_e}{\partial z^2}\right| h_z^2\right\} \\ &+ \frac{1}{144}\left\{\left|\frac{\partial^4 \rho_e}{\partial x^2 \partial y^2}\right| h_x^2 h_y^2 + \left|\frac{\partial^4 \rho_e}{\partial y^2 \partial z^2}\right| h_y^2 h_z^2 + \left|\frac{\partial^4 \rho_e}{\partial z^2 \partial x^2}\right| h_z^2 h_x^2\right\} \\ &+ \frac{1}{1728}\left|\frac{\partial^6 \rho_e}{\partial x^2 \partial y^2 \partial z^2}\right| h_x^2 h_y^2 h_z^2\Bigg) \quad \text{in 3D.} \end{aligned} \tag{37}$$

Note in the above equations the reason for including the only higher order terms proportional to the mixed derivatives is because these terms will contribute to the dominant error for the sparse grid combination. Hence, the constants in front of these terms are of interest for estimating the coefficients of the grid-based error in section 4.4.3.

B.2 Noise estimate

This is the error which occurs when we approximate the expected value of the shape function by means of an arithmetic mean over the number of discrete particles. Thus equation (29) becomes

$$\bar{\rho}_e(x) \approx \tilde{\rho}_e(x) = \frac{Q_e}{N_p} \sum_p S(x - x_p). \tag{38}$$

The error incurred by this approximation $\eta(x)$ is a random variable with mean 0 and variance given by

$$Var_{\bar{f}}\left[\eta(x)\right] = \mathbb{E}_{\bar{f}}\left[\left(\bar{\rho}_e - \tilde{\rho}_e\right)^2\right], \tag{39}$$

$$= \bar{\rho}_e^2 - 2\bar{\rho}_e\mathbb{E}_{\bar{f}}[\tilde{\rho}_e] + \mathbb{E}_{\bar{f}}[\tilde{\rho}_e^2], \tag{40}$$

$$= \mathbb{E}_{\bar{f}}[\tilde{\rho}_e^2] - \bar{\rho}_e^2. \tag{41}$$

Here, in equation (41) we used the fact that $\mathbb{E}_{\bar{f}}[\tilde{\rho}_e] = \mathbb{E}_{\bar{f}}[\bar{\rho}_e] = \bar{\rho}_e$. Let us compute $\mathbb{E}_{\bar{f}}\left[\tilde{\rho}_e^2\right]$

$$\mathbb{E}_{\bar{f}}\left[\tilde{\rho}_e^2\right] = \mathbb{E}_{\bar{f}}\left[\frac{Q_e^2}{{N_p}^2}\left(\sum_p S(x - x_p)\right)^2\right]. \tag{42}$$

Similar to [4] we assume that the initial particle states have been chosen by independent sampling from $\bar{f}(t=0)$ and also they remain approximately independent for $N_p \gg 1$. Then $\mathbb{E}_{\bar{f}}\left[S(x-x_p)S(x-x_q)\right] = 0$ if $p \neq q$ and all the cross terms vanish giving

$$\mathbb{E}_{\bar{f}}\left[\tilde{\rho}_e^2\right] = \frac{Q_e^2}{{N_p}^2}\sum_p \mathbb{E}_{\bar{f}}\left[\left(S(x-x_p)\right)^2\right], \tag{43}$$

$$= \frac{Q_e^2}{N_p}\mathbb{E}_{\bar{f}}\left[\left(S(x-x_p)\right)^2\right], \tag{44}$$

where, we have used the fact that each particle has same $\mathbb{E}_{\bar{f}}\left[\left(S(x-x_p)\right)^2\right]$. Now,

$$\frac{Q_e^2}{N_p}\mathbb{E}_{\bar{f}}\left[\left(S(x-x_p)\right)^2\right] = \frac{Q_e^2}{N_p}\int\int \bar{f}(x_p, v)\left(S(x-x_p)\right)^2 dx_p dv, \tag{45}$$

$$= \frac{Q_e^2}{N_p}\int\int \Big(\bar{f}(x,v) + (x_p - x)\partial_x\bar{f}(x,v) + \frac{(x_p-x)^2}{2}\partial_x^2\bar{f}(x,v) + \cdots\Big)\left(S(x-x_p)\right)^2 dx_p dv, \tag{46}$$

and similar to the previous exercise for grid-based error the term associated with $(x_p - x)\partial_x\bar{f}(x,v)$ vanishes and the third term evaluates to $\mathcal{O}(h_x)$. Hence evaluating the leading order term gives

$$\frac{Q_e^2}{N_p}\int\int \bar{f}(x,v)\left(S(x-x_p)\right)^2 dx_p dv = \frac{Q_e}{N_p}\underbrace{\int Q_e\bar{f}dv}_{\rho_e}\int\left(S(x-x_p)\right)^2 dx_p, \tag{47}$$

$$= \frac{Q_e\rho_e}{N_p}\frac{2}{h_x^2}\int_0^{h_x}\left(1 - \frac{\zeta}{h_x}\right)^2 d\zeta, \tag{48}$$

$$= \frac{2}{3}\frac{Q_e\rho_e}{N_p h_x}. \tag{49}$$

Plugging the above term in equation (44) gives

$$\mathbb{E}_{\bar{f}}\left[\tilde{\rho}_e^2\right] = \frac{2}{3}\frac{Q_e\rho_e}{N_p h_x} + \mathcal{O}(h_x) + \cdots . \tag{50}$$

Omitting the $\bar{\rho}_e^2$ term in equation (41) as it is small compared to equation (50) and substituting the above expression gives

$$Var_{\bar{f}}\left[\eta(x)\right] \approx \frac{2}{3}\frac{Q_e\rho_e}{N_p h_x}.$$

Defining the particle noise error e_n as the standard deviation of the random variable η we get

$$e_n(x) = \mathcal{O}\left(\sqrt{\frac{2}{3}\frac{|Q_e\rho_e(x)|}{N_p h_x}}\right). \tag{51}$$

Similarly, carrying out the same set of steps in 2D and 3D we get the estimates for the particle noise as

$$e_n = \mathcal{O}\left(\sqrt{\frac{4}{9}\frac{|Q_e\rho_e|}{N_p h_x h_y}}\right) \quad \text{in 2D}, \tag{52}$$

$$e_n = \mathcal{O}\left(\sqrt{\frac{8}{27}\frac{|Q_e\rho_e|}{N_p h_x h_y h_z}}\right) \quad \text{in 3D}. \tag{53}$$

References

[1] C. C. W. Leentvaar, Pricing multi-asset options with sparse grids, Ph.D. thesis, Technische Universiteit Delft (2008).

[2] J. Benk, D. Pflüger, Hybrid parallel solutions of the Black-Scholes PDE with the truncated combination technique, in: 2012 International Conference on High Performance Computing & Simulation (HPCS), IEEE, 2012, pp. 678–683.

[3] J. Benk, H.-J. Bungartz, A.-E. Nagy, S. Schraufstetter, Variants of the combination technique for multi-dimensional option pricing, in: Progress in Industrial Mathematics at ECMI 2010, Springer, 2012, pp. 231–237.

[4] L. F. Ricketson, A. J. Cerfon, Sparse grid techniques for particle-in-cell schemes, Plasma Physics and Controlled Fusion 59 (2) (2016) 024002.

[5] R. W. Hockney, J. W. Eastwood, Computer simulation using particles, CRC Press, 1988.

[6] C. K. Birdsall, A. B. Langdon, Plasma physics via computer simulation, CRC press, 2004.

[7] J. M. Dawson, Particle simulation of plasmas, Reviews of modern physics 55 (2) (1983) 403.

[8] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, Journal of computational physics 73 (2) (1987) 325–348.

[9] F. Filbet, E. Sonnendrücker, Numerical methods for the Vlasov equation, in: Numerical mathematics and advanced applications, Springer, 2003, pp. 459–468.

[10] H. Nakashima, Y. Summura, K. Kikura, Y. Miyake, Large scale manycore-aware PIC simulation with efficient particle binning, in: 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2017, pp. 202–212.

[11] R. E. Denton, M. Kotschenreuther, δf Algorithm, Journal of Computational Physics 119 (2) (1995) 283–294.

[12] A. Y. Aydemir, A unified monte carlo interpretation of particle simulations and applications to non-neutral plasmas, Physics of Plasmas 1 (4) (1994) 822–831.

[13] R. Sydora, Low-noise electromagnetic and relativistic particle-in-cell plasma simulation models, Journal of computational and applied mathematics 109 (1-2) (1999) 243–259.

[14] A. Spitkovsky, Simulations of relativistic collisionless shocks: shock structure and particle acceleration, in: AIP Conference Proceedings, Vol. 801, American Institute of Physics, 2005, pp. 345–350.

[15] O. Buneman, Computer space plasma physics, simulation techniques and softwares, ed, H. Matsumoto and Y. Omura (Terra Scientific, Tokyo, 1993) p 67.

[16] S. Jolliet, A. Bottino, P. Angelino, R. Hatzky, T.-M. Tran, B. Mcmillan, O. Sauter, K. Appert, Y. Idomura, L. Villard, A global collisionless PIC code in magnetic coordinates, Computer Physics Communications 177 (5) (2007) 409–425.

[17] B. Terzić, I. V. Pogorelov, C. L. Bohn, Particle-in-cell beam dynamics simulations with a wavelet-based poisson solver, Physical Review Special Topics-Accelerators and Beams 10 (3) (2007) 034201.

[18] J.-L. Vay, A. Almgren, J. Bell, L. Ge, D. Grote, M. Hogan, O. Kononenko, R. Lehe, A. Myers, C. Ng, et al., Warp-x: A new exascale computing platform for beam–plasma simulations, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 909 (2018) 476–479.

[19] G.-H. Cottet, P.-A. Raviart, Particle methods for the one-dimensional Vlasov–Poisson equations, SIAM journal on numerical analysis 21 (1) (1984) 52–76.

[20] B. Wang, G. H. Miller, P. Colella, A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas, SIAM Journal on Scientific Computing 33 (6) (2011) 3509–3537.

[21] A. Myers, P. Colella, B. V. Straalen, A 4th-order particle-in-cell method with phase-space remapping for the Vlasov–Poisson equation, SIAM Journal on Scientific Computing 39 (3) (2017) B467–B485.

[22] B. Wang, G. Miller, P. Colella, An adaptive, high-order phase-space remapping for the two dimensional Vlasov–Poisson equations, SIAM Journal on Scientific Computing 34 (6) (2012) B909–B924.

[23] J. P. Verboncoeur, Particle simulation of plasmas: review and advances, Plasma Physics and Controlled Fusion 47 (5A) (2005) A231.

[24] J.-L. Vay, C. G. Geddes, E. Cormier-Michel, D. P. Grote, Numerical methods for instability mitigation in the modeling of laser wakefield accelerators in a lorentz-boosted frame, Journal of Computational Physics 230 (15) (2011) 5908–5929.

[25] S. Gassama, É. Sonnendrücker, K. Schneider, M. Farge, M. O. Domingues, Wavelet denoising for postprocessing of a 2d particle-in-cell code, in: ESAIM: Proceedings, Vol. 16, EDP Sciences, 2007, pp. 195–210.

[26] B. Terzić, G. Bassi, New density estimation methods for charged particle beams with applications to microbunching instability, Physical Review Special Topics-Accelerators and Beams 14 (7) (2011) 070701.

[27] W. Wu, H. Qin, Reducing noise for PIC simulations using kernel density estimation algorithm, Physics of Plasmas 25 (10) (2018) 102107.

[28] M. Shalaby, A. E. Broderick, P. Chang, C. Pfrommer, A. Lamberts, E. Puchwein, SHARP: A spatially higher-order, relativistic particle-in-cell code, The Astrophysical Journal 841 (1) (2017) 52.

[29] M. Griebel, M. Schneider, C. Zenger, A combination technique for the solution of sparse grid problems, in: Iterative Methods in Linear Algebra, eds. R. Bequwens and P. de Groen (Amsterdam: Elsevier), 1990, pp. 263–281.

[30] H.-J. Bungartz, M. Griebel, D. Röschke, C. Zenger, Pointwise convergence of the combination technique for Laplace's equation, East-West J. Numer. Math 2 (1994) 21–45.

[31] H.-J. Bungartz, M. Griebel, Sparse grids, Acta numerica 13 (2004) 147–269.

[32] D. M. Pflüger, Spatially adaptive sparse grids for high-dimensional problems, Ph.D. thesis, Technische Universität München (2010).

[33] M. Heene, A massively parallel combination technique for the solution of high-dimensional PDEs, Ph.D. thesis, Universitat Stuttgart (2018).

[34] M. Griebel, V. Thurner, The efficient solution of fluid dynamics problems by the combination technique, International Journal of Numerical Methods for Heat & Fluid Flow 5 (3) (1995) 251–269.

[35] L. N. Trefethen, Cubature, approximation, and isotropy in the hypercube, SIAM Review 59 (3) (2017) 469–491.

[36] A. Cerfon, L. Ricketson, Sparse grid particle-in-cell scheme for noise reduction in beam simulations, in: 13th Int. Computational Accelerator Physics Conf.(ICAP'18), Key West, FL, USA, 20-24 October 2018, JACOW Publishing, Geneva, Switzerland, 2019, pp. 71–75.

[37] A. Adelmann, P. Calvo, M. Frey, A. Gsell, U. Locans, C. Metzger-Kraus, N. Neveu, C. Rogers, S. Russell, S. Sheehy, J. Snuvernik, D. Winklehner, OPAL a versatile tool for charged particle accelerator simulations, arXiv preprint arXiv:1905.06654.

[38] B. Lastdrager, B. Koren, Error analysis for function representation by the sparse-grid combination technique, Journal of Computational and Applied Mathematics.

[39] D. L. Donoho, I. M. Johnstone, Adapting to unknown smoothness via wavelet shrinkage, Journal of the american statistical association 90 (432) (1995) 1200–1224.

[40] P. E. Strazdins, M. M. Ali, B. Harding, Highly scalable algorithms for the sparse grid combination technique, in: 2015 IEEE International Parallel and Distributed Processing Symposium Workshop, IEEE, 2015, pp. 941–950.

[41] D. del Castillo-Negrete, K. Schneider, M. Farge, G. Chen, et al., Wavelet-based density estimation for noise reduction in plasma simulations using particles, Journal of Computational Physics 229 (8) (2010) 2821–2839.

[42] D. F. Martin, P. Colella, A cell-centered adaptive projection method for the incompressible Euler equations, Journal of computational Physics 163 (2) (2000) 271–312.

[43] M. Frey, A. Adelmann, U. Locans, On architecture and performance of adaptive mesh refinement in an electrostatics particle-in-cell code, Computer Physics Communications (2019) 106912.

[44] L. Berger-Vergiat, C. A. Glusa, J. J. Hu, C. Siefert, R. S. Tuminaro, M. Matthias, P. Andrey, W. Tobias, Muelu user's guide., Tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States) (2019).

[45] W. Zhang, A. Almgren, V. Beckner, J. Bell, J. Blaschke, C. Chan, M. Day, B. Friesen, K. Gott, D. Graves, M. Katz, A. Myers, T. Nguyen, A. Nonaka, M. Rosso, S. Williams, M. Zingale, AMReX: a framework for block-structured adaptive mesh refinement, Journal of Open Source Software 4 (37) (2019) 1370.

[46] A. Gholami, D. Malhotra, H. Sundar, G. Biros, FFT, FMM, or multigrid? A comparative study of state-of-the-art poisson solvers for uniform and nonuniform grids in the unit cube, SIAM Journal on Scientific Computing 38 (3) (2016) C280–C306.

[47] C. Driscoll, K. Fine, Experiments on vortex dynamics in pure electron plasmas, Physics of Fluids B: Plasma Physics 2 (6) (1990) 1359–1366.

[48] K. S. Fine, C. F. Driscoll, J. H. Malmberg, T. B. Mitchell, Measurements of symmetric vortex merger, Phys. Rev. Lett. 67 (1991) 588–591.

[49] A. J. Cerfon, J. P. Freidberg, F. I. Parra, T. A. Antaya, Analytic fluid theory of beam spiraling in high-intensity cyclotrons, Phys. Rev. ST Accel. Beams 16 (2013) 024202.

[50] A. J. Cerfon, Vortex dynamics and shear-layer instability in high-intensity cyclotrons, Phys. Rev. Lett. 116 (2016) 174801.

[51] P. G. Drazin, W. H. Reid, Hydrodynamic Stability, Cambridge University Press, 2004.

[52] R. C. Davidson, Physics of nonneutral plasmas, Imperial College Press London, 2001.

[53] Y. Jo, J. Kim, G. Stancari, M. Chung, H. J. Lee, Control of the diocotron instability of a hollow electron beam with periodic dipole magnets, Physics of Plasmas 25 (1) (2018) 011607.

[54] L. S. Brown, G. Gabrielse, Geonium theory: Physics of a single electron or ion in a penning trap, Reviews of Modern Physics 58 (1) (1986) 233.

[55] K. Blaum, Y. N. Novikov, G. Werth, Penning traps as a versatile tool for precise experiments in fundamental physics, Contemporary Physics 51 (2) (2010) 149–175.

[56] S. Adam, Space charge effects in cyclotrons-from simulations to insights, in: Proc. of the 14th Int. Conf. on Cyclotrons and their Applications,(World Scientific, Singapore, 1996), Vol. 446, 1995.

[57] J. Yang, A. Adelmann, M. Humbel, M. Seidel, T. Zhang, et al., Beam dynamics in high intensity cyclotrons including neighboring bunch effects: Model, implementation, and application, Physical Review Special Topics-Accelerators and Beams 13 (6) (2010) 064201.

[58] A. Bungau, A. Adelmann, J. Alonso, W. Barletta, R. Barlow, L. Bartoszek, L. Calabretta, A. Calanna, D. Campo, J. Conrad, et al., Proposal for an electron antineutrino disappearance search using high-rate li 8 production and decay, Physical Review Letters 109 (14) (2012) 141802.

[59] J. Yang, A. Adelmann, W. Barletta, L. Calabretta, A. Calanna, D. Campo, J. Conrad, Beam dynamics simulation for the high intensity daeδalus cyclotrons, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 704 (2013) 84–91.