ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SUPAF09

SPARSE GRID PARTICLE-IN-CELL SCHEME FOR NOISE REDUCTION IN BEAM SIMULATIONS

L. F. Ricketson*, Lawrence Livermore National Laboratory, Livermore, CA, USA
A. J. Cerfon, Courant Institute of Mathematical Sciences, New York University, New York, NY, USA

Abstract

We demonstrate that the sparse grid combination technique, a scheme originally designed for grid based solvers of high-dimensional partial differential equations, can be effectively applied to reduce the noise of Particle-in-Cell (PIC) simulations. This is because the sparse grids used in the combination technique have large cells relative to a comparable regular grid, which, for a fixed overall number of particles, increases the number of particles per cell, and thus improves statistical resolution. In other words, sparse grids can accelerate not only the computation of the electromagnetic fields, but also the particle operations, which typically dominate the computation and storage requirements.

INTRODUCTION

In charged beams in particle accelerators, the Coulomb collision frequency is much smaller than the other frequencies of interest, even at the highest achievable beam intensities. A kinetic description of the beam is therefore required, in which one solves for the beam distribution function giving the number of particles in an infinitesimal six-dimensional phase space volume. Because of the high dimensionality of the phase space volume, intense beam simulations are very computationally intensive; even a modest grid resolution for each of the six dimensions pushes the limits of today's largest supercomputers.

To circumvent this difficulty, particle based approaches to the problem have been widely adopted, usually in the form of the Particle-in-Cell (PIC) algorithm [1]. The PIC method has the advantages of being conceptually intuitive, being well-suited for massive parallelization, and only requiring discretization of configuration space. Detailed PIC simulations of intense charged beams are routinely run and relied upon to explain experimental results and to design new accelerators [2, 3]. Even so, the accuracy of these PIC simulations remains limited due to the probabilistic nature of the PIC scheme, which requires a large number of particles to be simulated in order to reduce statistical noise. We will indeed show in this article that the slow decay of this noise with the number of simulated particles implies that for a given target accuracy, standard PIC simulations may even be more computationally intensive than grid based simulations.

In this work, we present a new algorithm which addresses this unsatisfying state of affairs by reducing the noise in PIC simulations. The algorithm is based on the sparse grid combination technique, a method originally intended to address the poor scaling of grid based PDE solvers with dimension [4]. We will explain how to apply the combination technique in a PIC setting, and demonstrate its promise by using our algorithm to solve standard problems in plasma and beam physics. The structure of the article is as follows. In the first Section, we compare the asymptotic run time complexity of a standard grid based kinetic solver and a standard PIC solver, and arrive at the conclusion that noise reduction strategies need to be implemented for the PIC approach to be desirable from a complexity point of view. In the second Section, we briefly present the sparse grid combination technique in the simple yet enlightening context of linear interpolation. The combination technique is the central idea motivating our new scheme. In the third Section, we explain how the combination technique can indeed be favorably applied in the context of PIC solvers, and numerically demonstrate the significant reduction in noise from doing so in the fourth Section. The fifth Section focuses on the limitations of our new sparse PIC scheme in its current, somewhat naive implementation, and highlights directions for further improvement. We provide a brief summary of our work in the last Section.

THE CURSE OF DIMENSIONALITY VS THE CURSE OF NOISE

It is well known that grid-based solvers for the kinetic equations describing beam evolution scale badly with the dimensionality of the problem. The computational complexity κ of a grid-based code can be expressed, in the best case scenario, as

$$\kappa \sim \frac{h^{-d}}{\Delta t},$$

where h is the grid size, Δt is the time step, and d is the dimension of the problem. If we consider a typical solver which would be second-order accurate in space and time, the numerical error ε scales like h^2 and Δt^2, so numerical error ε and run-time complexity κ can be related through the scaling

$$\kappa \sim \varepsilon^{-\frac{d+1}{2}}.$$

The exponential dependence of κ on d is a major reason why continuum kinetic simulations are computationally intensive. It is often referred to as the *curse of dimensionality*.

Particle based algorithms such as the PIC algorithm address the curse of dimensionality by approximating the distribution function in terms of macro-particles which evolve in configuration space, which is at most three-dimensional. If N_p is the number of particles used in the simulation, we can write

$$\kappa \sim \frac{dN_p}{\Delta t}$$

* ricketson1@llnl.gov

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SUPAF09

At this point, it appears that the dependence of κ on dimension is much more favorable. However, the proper measure is to consider the complexity for a certain level of numerical error ε. The price one pays by adopting a particle based approach is in its inherent statistical noise, which only decreases with the square root of the number of simulated particles. Specifically, in a standard PIC scheme, if N_c is the number of cells used and d_X the number of dimensions of the configuration space, $\varepsilon \sim (N_p/N_c)^{-1/2} \sim N_p^{-(1/2)} h^{-d_X/2}$, $\varepsilon \sim h^2$, and $\varepsilon \sim \Delta t^2$. Using these scalings in the formula for κ, one finds

$$\kappa \sim d\varepsilon^{-(2.5+\frac{d_X}{2})}.$$

One notes that because of the slow convergence of the numerical error with the number of simulated particles, the scaling of κ is not favorable for the PIC scheme for small values of d and d_X. Even for $d = 6$ and $d_X = 3$, which correspond to the largest values one needs to consider in beam physics, κ depends more strongly on ε in a standard PIC scheme than in a standard continuum scheme: ε^{-4} vs $\varepsilon^{-7/2}$. This is what we may call the *curse of noise*.

In the remainder of this article, we will present and analyze the performance of a numerical scheme we propose to address the curse of noise in PIC simulations. The algorithm is directly inspired from a numerical method known as the sparse grid combination technique, which was originally invented to tackle the curse of dimensionality of grid based solvers for partial differential equations [4]. As we will show, it can also be an efficient technique to address the curse of noise in particle based solvers.

THE SPARSE GRID COMBINATION TECHNIQUE: ILLUSTRATION WITH INTERPOLATION

In this section, we present the fundamental idea underpinning the sparse grid combination technique by focusing on its application in a key step of the PIC scheme, namely the interpolation from values on a grid to points off the grid. For the simplicity of the presentation, we will consider a two-dimensional situation. However, the same ideas are applicable - and in fact more valuable - in three-dimensional situations.

We consider a standard case in which we know the values of the function $u(x, y)$ on a Cartesian grid discretizing the square $[-1, 1] \times [-1, 1]$, with grid width and height h_x and h_y, and we wish to approximate u on the entire domain via bilinear interpolation. The error between the exact function u and its approximation u at a particular point off the grid is given by the *exact* formula [4]

$$u(x, y) - \mathrm{u}(x, y) = C_1(h_x)h_x^2 + C_2(h_y)h_y^2 + C_3(h_x, h_y)h_x^2 h_y^2. \quad (1)$$

where the C_i above are *functions* with a uniform upper bound. In the absence of additional information about u, one typically chooses $h_x = h_y = h$ and finds an error of $O(h^2)$. Furthermore, the computational complexity κ of the optimal scheme scales linearly in the number of grid points, so that $\kappa = O(h^{-2})$. We conclude that κ scales with the numerical error ε according to $\kappa \sim \varepsilon^{-1}$. In arbitrary dimension D, the same reasoning leads to the optimal complexity $\kappa \sim \varepsilon^{-D/2}$, where we observe an exponential dependence on D – this is precisely the curse of dimensionality.

The combination technique is designed to reduce the strong dependence of κ on D by using cancellation across different "sparser" grids. Suppose the desired resolution is $h_N = 2^{-N}$ for some positive integer N. Let $h_x^i = 2^{-i}$, $h_y^j = 2^{-j}$ and $\mathrm{u}_{i,j}$ be the approximation of u on the corresponding grid. Then, consider the quantity u_N defined by

$$\mathrm{u}_N = \sum_{i+j=N+1} \mathrm{u}_{i,j} - \sum_{i+j=N} \mathrm{u}_{i,j}. \quad (2)$$

In each of the sums, i and j are strictly positive integers. The combination (2) is depicted graphically in Figure 1.

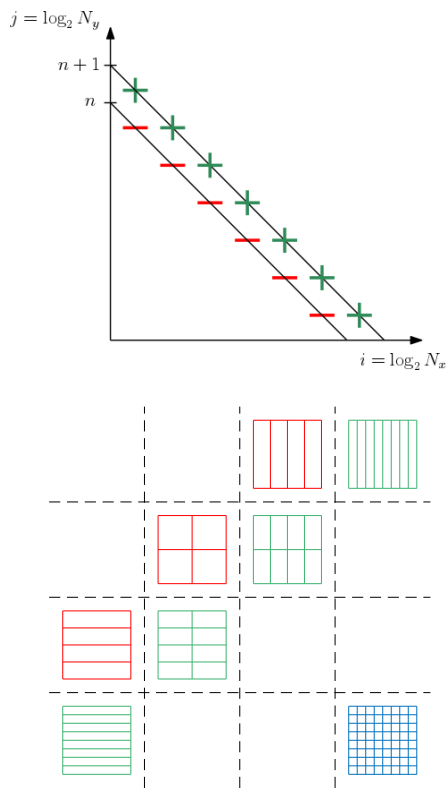

Figure 1: A graphical depiction of the combination of grids used in (2). (Top) The green '+' signs represent grids that give a positive contribution, while red '−' signs are subtracted. Cancellation arises from pairing neighboring grids along vertical and horizontal axes. (Bottom) An illustration of the sparse grids giving a positive contribution (in green), and the sparse grids giving a negative contribution (in red), for the case $N = 4$. The blue grid is the equivalent full grid one would use in a standard PIC simulation

By considering Figure 1 and the error formula (1), we see that a great deal of cancellation occurs in computing the error corresponding to u_N. Specifically, for any particular i be-

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SUPAF09

tween 1 and $N-1$, a grid with horizontal spacing h_x^i appears exactly once in each of the two sums in (2). For those two grids, the term $C_1(h_x)h_x^2$ that appears in (1) cancels *exactly*, because it is independent of h_y. The only contribution from the $O(h_x^2)$ term thus comes from the grid with $h_x = 2^{-N}$. Analogous reasoning for the y-direction leads to the result

$$u - \mathrm{u}_N = C_1(h_N)h_N^2 + C_2(h_N)h_N^2 \tag{3}$$

$$+ h_N^2 \left\{ \frac{1}{4} \sum_{i+j=N+1} C_3(h_x^i, h_y^j) - \sum_{i+j=N} C_3(h_x^i, h_y^j) \right\}, \tag{4}$$

where we have used the fact that $h_x^i h_y^j = h_N/2$ when $i+j = N+1$ and $h_x^i h_y^j = h_N$ when $i+j = N$. The expression in braces contains $2N-1$ terms which are all uniformly bounded by constants, so we find that

$$|u - \mathrm{u}_N| = O(Nh_N^2) = O(h_N^2 |\log h_N|). \tag{5}$$

In other words, u_N approximates u *nearly as well* as an approximate solution using $h_x = h_y = 2^{-N}$. This is illustrated in Figure 2, in which we plot the maximum error when interpolating the function $u(x,y) = \sin(2\pi x)\cos(3\pi y)$ at 50 randomly located off-grid points in the square $[-1,1]\times[-1,1]$. For each value of N, the interpolation error for the 50 points are plotted, with black dots for the sparse grids technique, and red stars for standard linear interpolation. The results confirm our asymptotic scalings. We however note a drawback of the sparse grid combination technique, which is explicit in this figure and which we will return to later in this article: the leading term in the sparse grid error depends on C_3, which is proportional to the fourth-order mixed derivative u_{xxyy}. This is the reason the constant for the purple curve in Figure 2 is larger than the constant for the yellow curve, and highlights the fact that the sparse grid technique requires functions to be quite smooth to be more efficient than the standard scheme.

Now, to understand why this technique leads to a favorable run time complexity, observe that each grid used in the combination technique has $O(h_N^{-1})$ grid points, and there are $O(N)$ grids, implying the scaling $\kappa = O(h_N^{-1}|\log h_N|)$, which we can write in terms of the error ε: $\kappa \sim \varepsilon^{-1/2}|\log\varepsilon|^2$. For three dimensional problems, this scaling generalizes to [5]

$$\kappa \sim \varepsilon^{-1/2}|\log\varepsilon|^4. \tag{6}$$

We see that with sparse grids, the dimensionality of the problem affects the complexity of the algorithm only through a weak logarithmic dependence. At least asymptotically, one can thus achieve the same accuracy considerably faster with the combination technique than with a single regular grid.

COMBINING SPARSE GRIDS WITH PIC

Sparse grids are not limited to interpolation, and can also be applied to another expensive stage in a standard PIC scheme, in which we assign a charge density to each macroparticle, through the introduction of a shape functions S. To see why this is so, note that the numerical error of approximating the true particle density ρ with the approximate

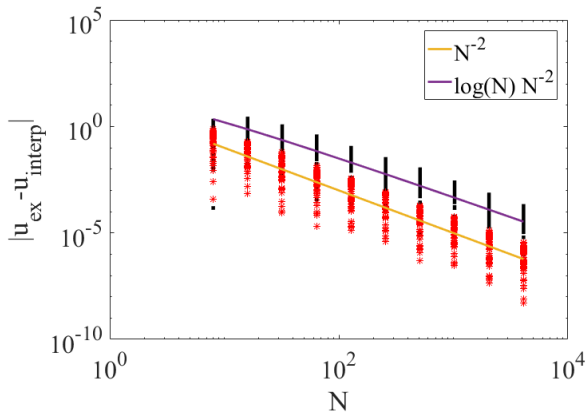

Figure 2: Numerical error as a function of the number of grid points N for the linear interpolation of the $u(x,y) = \sin(2\pi x)\cos(3\pi y)$ at randomly located off-grid points in the square $[-1,1]\times[-1,1]$. The black dots correspond to the error obtained with the sparse grids combination technique, and the red stars correspond to the error obtained with standard interpolation. The asymptotic scalings $O(N^{-2})$ and $O(N^{-2}\log N)$ are shown with continuous lines for comparison.

density ϱ can be written as [5]

$$\rho(\mathbf{x}_k) - \varrho(\mathbf{x}_k) = C_1(h_x)h_x^2 + C_2(h_y)h_y^2 + C_3(h_x,h_y)h_x^2h_y^2 + \xi_k, \tag{7}$$

where ξ_k is a random variable with $\mathbb{E}[\xi_k] = 0$ and $\mathrm{Var}[\xi_k] \approx \frac{4Q\rho(\mathbf{x}_k)}{9}\frac{1}{h_x h_y N_p}$, with Q the total charge of the beam. The first three terms on the right-hand side of Eq. (7) can be interpreted as the grid-based error of the scheme, while the last term is the particle sampling error. The grid based error has exactly the same form as in Eq. (1), so applying the combination technique to the evaluation of ϱ will yield the same benefits as before for this contribution to the total error. Furthermore, we have shown in [5] that the sparse grids combination technique also eliminates the curse of dimensionality for the particle sampling error, which scales as $\varepsilon_{samp} \sim |\log h_N|^{D-1}(N_p h_N)^{-1/2}$ in a sparse grids implementation of a problem with spatial dimension D. These results prompt us to consider the following sparse grids modification to the standard PIC algorithm:

(i) Push particles exactly as in standard PIC.

(ii) Assign to each particle a sequence of shape functions $\mathcal{S}_{i,j}(\mathbf{x}-\mathbf{x}_p) = \tau(2^i(x-x_p))\tau(2^j(y-y_p))/2^{i+j}$, where τ is the "hat" function such that $\tau(x) = 1-|x|$ if $|x| \le 1$ and $\tau(x) = 0$ otherwise, and approximate the overall charge density via

$$\rho \approx \varrho = \sum_{i+j=n+1} \varrho_{i,j} - \sum_{i+j=n} \varrho_{i,j}, \tag{8}$$

where $\varrho_{i,j}$ is defined at grid points $\mathbf{x}_{k,\ell} = (k2^{-i}, \ell 2^{-j})$ by

$$\varrho_{i,j}(\mathbf{x}_{k,\ell}) = \frac{Q}{N_p}\sum_p \mathcal{S}_{i,j}(\mathbf{x}_p - \mathbf{x}_{k,\ell}) \tag{9}$$

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SUPAF09

and extended to the entire domain using bilinear interpolation.

(iii) Use a grid-based Poisson solver to compute $\varphi_{i,j}$ and $\mathbf{E}_{i,j}$ by solving $-\nabla^2\varphi_{i,j} = \varrho_{i,j}$

(iv) Evaluate $\mathbf{E}$ at the particle positions $\mathbf{x}_p$ via

$$\mathbf{E}(\mathbf{x}_p) = \sum_{i+j=n+1} \mathbf{E}_{i,j}(\mathbf{x}_p) - \sum_{i+j=n} \mathbf{E}_{i,j}(\mathbf{x}_p). \qquad (10)$$

Use this to repeat step i.

The run time complexity κ in terms of the error ε of this sparse-PIC algorithm is $\kappa \sim D\frac{N_p}{\Delta t} \sim D\varepsilon^{-3}|\log\varepsilon|^{3(D-1)}$ and depends weakly on the dimension of the problem [5]. The improvement over standard PIC can be intuitively understood in the following way. The figure of merit for the statistical error in a PIC scheme is the number of particles *per cell* $P_c = N_p/N_c$, where N_c is the number of cells. Since N_c scales inversely with the cell volume, $N_c \sim h^{-3}$ on a regular grid in 3-D. However, on a 3-D sparse grid, $N_c \sim (4h)^{-1}$. We thus achieve *many* more particles per cell, even with the total particle number fixed, by using a hierarchy of sparse grids. Observe furthermore that in a typical PIC scheme, the cell size has to be smaller than the DeBye length in all dimensions. In contrast, in sparse PIC only few grids have to resolve the DeBye length, and when they do, they only do so in one cell direction. In the next section, we will show that these theoretical results are confirmed in practice, with significant speed up as compared to a standard PIC scheme.

In addition to the stark algorithmic advantages described above, sparse PIC also holds the promise of tremendous benefit in massively parallel implementations. On modern distributed memory architectures, PIC's spatial grid is typically domain-decomposed across many compute nodes. Thus, at each time step, particle data must be moved onto the node corresponding to its location on the grid. This creates considerable communication and load-balancing overhead, both of which play increasing roles in determining overall computation time as architectures advance. The sparse combination grids being promoted here, however, require *far* less memory to store than their full-grid counterparts - for example, a $2048 \times 2048 \times 2048$ full grid of double precision floats requires 64 gigabytes of storage, while the analogous sparse combination grid requires only 1.44 *mega*bytes. With sparse grids, it thus becomes trivially cheap to replicate the *entire* spatial grid on every compute node. By doing so, particles may remain on a single node for the entirety of the simulation, eliminating data motion and making load balancing for particle operations trivial - one simply initializes each node with exactly the same number of particles!

NUMERICAL EXAMPLE

To illustrate noise reduction through the sparse grids combination technique, we consider the time evolution of a Gaussian electron distribution in a periodic simulation domain. Two things are expected to happen: 1) the electron distribution is subject to periodic Langmuir oscillations at the plasma frequency; 2) the amplitude of the density distribution decreases due to nonlinear Landau damping. This is indeed what we observed. The measure of interest for this article is the level of statistical noise in the density distribution after one Langmuir oscillation, which can be visualized in Figure 3. This figure demonstrates the remarkable savings in terms of computer time and memory usage sparse grids bring about: with 100 times fewer particles, the sparse-PIC scheme yields comparable accuracy as the standard PIC scheme.

Since sparse grids are designed to combat the curse of dimensionality, the savings are even more significant for 3-dimensional simulations, as we have shown for Landau damping in three dimensions in [5]. We chose not to plot the results here, because they are not as visually striking. However, we will highlight the central conclusions. We found that sparse PIC consistently uses less memory than standard PIC for a given target accuracy, often by an order of magnitude. For a target error in the electron density, sparse PIC also consistently uses less computation time than standard PIC; for the electric field, which is less affected by sampling noise, the computation times can be comparable.

A WORD OF CAUTION

This article summarizes very recent progress for a project that is still in its infancy. We therefore would like to emphasize drawbacks of the sparse grids technique as presented here, which one will need to address in order to obtain a scheme which is indeed superior to the standard PIC scheme in most situations of physical interest. First, it is clear from the presentation in the third Section ("The Sparse Grid Combination Technique: Illustration with Interpolation") that the combination technique requires a structured grid. This may not be too stringent a constraint for the simulation of the acceleration phase in cyclotrons, but may be more challenging in other situations, such as axial injection into the cyclotron [3]. Furthermore, as we also highlighted in the third Section, the combination technique is not well suited for functions which have fine structure in all directions, which can be understood from the fact that the sparse grids never have fine resolution in all directions. Put differently, sparse grids perform best when the structure of the solution is aligned with the grid in a tensor product-like structure [6]. We have verified this empirically in [5] by considering a physical problem for which the solution is more efficiently represented in cylindrical coordinates. We found that in this situation the performance of the standard PIC scheme was superior to that of the sparse PIC scheme in the simple Cartesian implementation presented in this article.

Because our work is still at a very early stage, we do not see these drawbacks as condemning the sparse PIC algorithm to only be used for the very specific cases considered here. Instead, it is a motivation to improve the young version of our scheme we presented in this article, in order to make it more robust and versatile. A promising avenue for improvement is to rely on higher order interpolation and higher order shape

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-ICAP2018-SUPAF09

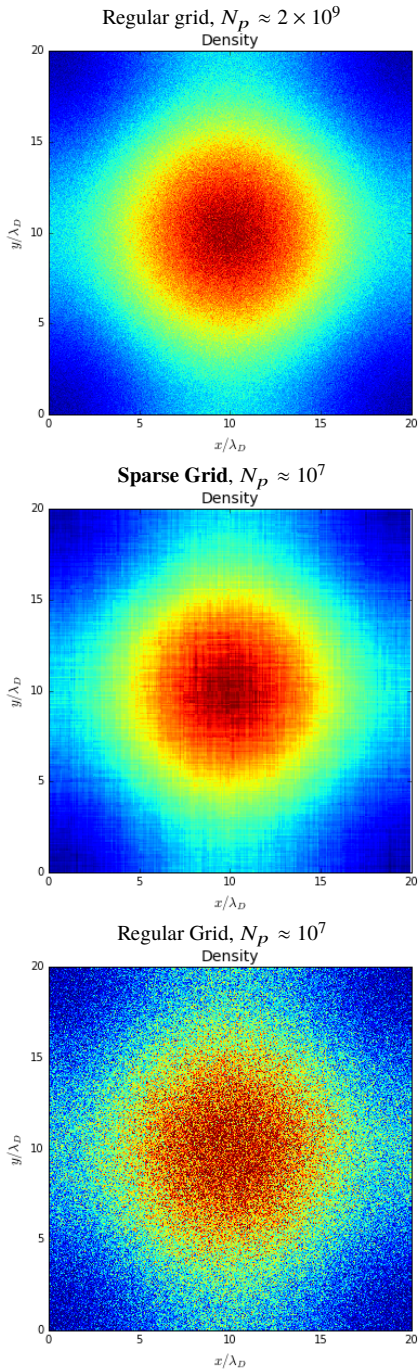

Figure 3: Time snapshots of the electron density from three simulations of the same nonlinear Landau damping problem. All have 1024×1024 effective resolution. Compared to the top figure, the sparse grid solution (middle) has comparable statistical resolution but runs 30 times faster. Compared to the bottom figure, the sparse scheme runs in comparable time, but with considerably improved statistics.

functions, in order to reduce the grid-based error due to the sparse grids combination technique. Combining the sparse grids combination technique with adaptive mesh refinement [7] may also have a strong potential, as it would allow us to better resolve the fine scale structures the sparse grids are missing. Finally, one could numerically construct optimized coordinate systems to be used by the sparse-PIC solver as the solution evolves, which would be designed to optimally align with the solution at each time step.

CONCLUSION

We have presented a new strategy for reducing statistical noise in PIC simulations based on the sparse grid combination technique, a numerical method which had previously only been considered for grid based solvers. We found that our algorithm could lead to major savings in memory and computation time because the number of particles required to reach a certain level of accuracy is drastically reduced as compared to the requirements for the standard PIC scheme. This is because the sparse grids used in our numerical scheme all have larger cells than in a standard PIC scheme, thus increasing the number of particle per cells for a given total number of particles. And by virtue of the sparse grid combination technique, the price we pay for this is only a slight increase in the grid-based error.

Our straightforward algorithm underperforms in situations in which the solution is far from aligning with the directions of the grid and has fine structure in all dimensions. We are currently considering improvements to our scheme to tackle these issues.

ACKNOWLEDGEMENTS

The authors would like to thank the organizers of ICAP'18 and the scientific program committee for inviting one of us and giving us the opportunity to present our results.

REFERENCES

[1] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*, Hilger, New York, 1988.

[2] Y. J. Bi, A. Adelmann, R. Dölling, M. Humbel, W. Joho, M. Seidel, and T. J. Zhang, *Phys. Rev. ST Accel. Beams*, vol. 14, pp. 054402, 2011.

[3] D. Winklehner, A. Adelmann, A. Gsell, T. Kaman, and D. Campo, *Phys. Rev. Accel. Beams*, vol. 20, pp. 124201, 2017.

[4] M. Griebel, M. Schneider, and C. Zenger, "A combination technique for the solution of sparse grid problems", in *Iterative Methods in Linear Algebra*, ed. R. Bequwens and P. de Groen, Amsterdam: Elsevier, pp. 263–81, 1990.

[5] L. F. Ricketson and A. J. Cerfon, *Plasma Phys. Control. Fusion* vol. 59, pp. 024002, 2017.

[6] L. N. Trefethen, "Cubature, Approximation, and Isotropy in the Hypercube", *SIAM Review*, vol. 59, no. 3, pp. 469–491, 2017. doi:10.1137/16M1066312

[7] H.-J. Bungartz and M. Griebel, "Sparse grids", *Acta Numerica*, vol. 13, pp. pp. 147–269, 2004. doi:10.1017/S0962492904000182