SLAC-PUB-13519
January2009

MESHLESS SOLUTION OF THE VLASOV EQUATION USING A LOW-DISCREPANCY SEQUENCE *

R. Warnock, Stanford Linear Accelerator Center, Menlo Park CA 94025, USA †
J. A. Ellison, K. Heinemann, G. Q. Zhang, U. New Mexico, Albuquerque NM 87131 USA‡

Abstract

A good method for solving the nonlinear Vlasov equation is the semi-Lagrangian algorithm, in which the phase space density is represented by its values on a fixed Cartesian grid with interpolation to off-grid points. At each time step, orbits are followed backward from grid points. Since this method is expensive with phase space dimension $D > 2$, we seek a more efficient discretization of the density. Taking a cue from the theory of numerical quadrature in high dimensions, we explore the idea of replacing the grid by scattered data sites from a low-discrepancy (quasi-random) sequence. We hope to see a reduction in the required number of sites, especially for $D > 2$. In our first implementation we follow forward orbits rather than backward, and work only with $D = 2$. We are able to reduce the number of sites by a factor of 8, at least for a limited time of integration. A much bigger reduction is expected in higher dimensions.

INTRODUCTION

Direct solution of the nonlinear Vlasov equation is a valuable alternative to macroparticle simulations of coherent motion, offering lower numerical noise and a rational control of errors. Until now it has been applied most often in problems with phase space dimension $D = 2$, because of heavy computational requirements with higher D. The time consuming step is interpolation of the phase space density to off-grid points. In seeking a more efficient representation of the density we look for inspiration in the theory and practice of numerical quadrature in high dimensions. Conventional Monte-Carlo integration, based on pseudo-random sequences, has a probabilistic error bound that is $\mathcal{O}(n^{-1/2})$ for n samples, which is more favorable in its n-dependence than error bounds for grid-based methods in high dimensions (although the latter are genuine, deterministic bounds)[1]. For instance, the trapezoidal rule in s-dimensions with a total of n nodes has $\mathcal{O}(n^{-2/s})$ convergence for a C^2 integrand, thus poorer for $s > 4$. Moreover, the Quasi - Monte Carlo method, based on deterministic "quasi-random" sequences, can have a genuine error bound that is $\mathcal{O}(N^{-1}(\ln N)^{s-1})$, thus better than the trapezoidal rule for $s > 2$ [1]. Of course, many grid-based rules have faster convergence than the trapezoidal, but the deterioration with s (the so-called "curse of dimensionality") is similar. Note that the dependence on s is very weak for Quasi - Monte Carlo, effectively killing the curse of dimensionality. Quasi-random sequences are also called low-discrepancy sequences (LDS) since they are designed to have minimal deviation from an ideal uniform distribution.

Solution of the Vlasov equation is certainly different from numerical quadrature, but it is reasonable to hope that the efficiency in use of data on a LDS will carry over to the Vlasov problem. This idea grew out of valuable conversations with G. Fasshauer. Such an approach would hardly be possible without a method for smooth interpolation of data at scattered sites, or possibly some kind of smooth fitting without strict interpolation. Opportunely, interpolation and fitting of scattered data has been an active area of numerical analysis in recent years [2, 3, 4, 5]. Intensively studied methods make use of radial basis functions, moving least squares, Shepard interpolation, and other techniques. In contrast to situations contemplated in text books, we need a huge number of interpolations and integrations, requiring close attention to efficiency and ruling out some suggested methods. Ambitions somewhat similar to ours can be found in the LDS literature [6], but a wedding of LDS and meshless interpolation is usually lacking.

FORMULATION

Although we expect a bigger advantage of LDS in higher dimensions we first study a Vlasov equation in 2D, for the case of longitudinal motion in a storage ring; namely,

$$\frac{\partial f}{\partial t} + p\frac{\partial f}{\partial q} + [-q + F_c(q, f(\cdot, t)]\frac{\partial f}{\partial p} = 0 \,, \qquad (1)$$

where $f(z,t)$ is the phase space density, $z = (q,p)$. Here $q = \zeta/\sigma_z,\ p = (E - E_o)/\sigma_E$, where ζ is the arc distance from the reference particle (positive in front), $E - E_o$ the deviation of energy from nominal, and $\sigma_z,\ \sigma_E$ the r.m.s. bunch length and energy spread at low current. The force on a particle consists of the r.f. force $-q$ plus the collective force given by a wake potential W as follows:

$$F_c(q, f(\cdot, t)) = -I_c \int W(q - q')f(q', p, t)dpdq' \,. \quad (2)$$

Here $I_c = e^2N/(2\pi\nu_s\sigma_E)$ is the normalized current, in terms of synchrotron tune ν_s and bunch population N. The independent variable t is referred to as time, but is actually $\omega_s(\mathrm{time})$ in our convention.

We seek to integrate (1) as an initial value problem, with initial density being the equilibrium density

* Work supported in part by U.S. Department of Energy contracts DE-AC02-76SF00515 and DE-FG02-99ER41104.
† warnock@slac.stanford.edu
‡ ellison@math.unm.edu

$f_o(q,p) = \exp(-p^2/2)\rho_o(q)/\sqrt{2\pi}$, where ρ_o is the solution of Haïssinski's equation [7].

Equation (1) expresses invariance of $f(z(t),t)$, where $z(t)$ is any particle orbit. An effective and stable integration algorithm, the semi-Lagrangian method, can be derived directly from the condition of invariance, which for a time step Δt is

$$f(M(z), t+\Delta t) = f(z,t)\ , \tag{3}$$

where $M(z)$ is the evolution map for the time interval $[t, t+\Delta t]$. In fuller notation, $M(z) = \phi(t+\Delta t, t, z, f)$ where $\phi(t', t, z, f)$ is the solution of the single particle equations of motion at time t' with initial value z at time t. Now $M(z)$ depends on the collective force which varies in time, but for a sufficiently small time step the variation can be ignored. This means that M and its inverse L are represented as

$$\begin{aligned} M(z) &\approx \phi(t+\Delta t, t, z, f(t))\ , \\ M^{-1}(z) = L(z) &\approx \phi(t-\Delta t, t, z, f(t))\ . \end{aligned} \tag{4}$$

Thinking of (4) as exact, and replacing z by $L(z)$ in (3), we have a scheme for updating f:

$$f(z, t+\Delta t) = f(L(z), t)\ . \tag{5}$$

In practice we have implemented (5) by representing $f(z,t)$ by its values $f(z_i,t)$ on a Cartesian grid $\{z_i\}$, using local polynomial interpolation (bi-quadratic or bi-cubic) to find values at the off-grid points $L(z_i)$ [7].

We could as well apply (5) with a fixed set of scattered sites $\{z_i\}$ if we had a way of interpolating the values $f(L(z_i),t)$ so as to obtain an approximation to $f(L(z),t)$ at all z.

Given a method to interpolate scattered data, we need not base our method on the backward conservation equation (5), since we could as well use the forward equation (3). Then $f(z, t+\Delta t)$ would be represented as an interpolation of the data $f(M(z_i), t+\Delta t) = f(z_i, t)$. Now the interpolation sites evolve in time and merely follow the particle orbits that begin at the initial sites $z_i(0)$, and the density at an evolved site $z_i(t)$ is equal to its initial value $f(z_i(0), 0)$. The hazard of this proposal is that clustering of sites may occur, so that the suitability of the evolved sites as a basis for interpolation may deteriorate. If that difficulty could be overcome the forward method would have a potential advantage in a certain class of problems; for instance, single-pass problems with energy chirp in which the support of the density is narrow and difficult to treat in a mesh-based backward algorithm.

Note that this forward scheme is rather similar to a weighted macroparticle simulation, in which the i-th macroparticle is assigned weight $f(z_i(0),0)$[8]. The differences between such a simulation and our scheme have to do with (A) the method of computing the charge density; and (B) the possibility of monitoring and controlling error.

For point (A) we propose to use the interpolant to find $f(q,p,t)$ on a grid $\{q_i^g, p_i^g\}$ and then use a grid-based quadrature rule to integrate over p. By contrast, macroparticle schemes usually find a histogram of charge, by collecting the charge in bins of q-space, then use a more or less arbitrary recipe to smooth the histogram. This is without doubt a faster procedure than what we suggest, but should be compared in accuracy as well as speed. A macroparticle scheme usually does not bother to construct the phase space density, but could do so, again with arbitrariness in the matter of smoothing.

For point (B) we note that in our schemes based on discretization of either (3) or (5) we can monitor interpolation error at each time step. Suppose that $f(z,t)$ is known. Then we also know $f(z, t+\Delta t)$ exactly, through (3). (Here we ignore error in the map M due to the freezing of the collective force.) For a useful algorithm we must, however, replace the exact $f(z, t+\Delta t)$ by an approximation based on interpolation. The error of the approximation can be computed directly on some suitable finite set $\{y_i\}$. Writing $\tilde{f}(z, t+\Delta t)$ for the approximation, we have the local errors ϵ_i and η_i for backward and forward methods, respectively, as follows:

$$\begin{aligned} \epsilon_i &= |\tilde{f}(y_i, t+\Delta t) - f(L(y_i), t)|\ , \\ \eta_i &= |\tilde{f}(M(y_i), t+\Delta t) - f(y_i, t)|\ . \end{aligned} \tag{6}$$

The y_i should have a fairly dense distribution, but be different from the sites z_i of the main algorithm. Similar considerations of error were introduced by Iske and collaborators [5].

APPLYING THE FORWARD SCHEME

Perhaps the simplest way of interpolating scattered data is Shepard's method [4, 2]. It is based on a partition of unity formed from singular weight functions $w_i(z) = \|z - z_i\|^{-p}$ with Euclidean norm $\|\ \|$ and positive p, which we take to be an even integer. The generating functions $\psi_i(z)$ satisfy

$$\sum_i \psi_i(z) = 1\ , \quad \psi_i(z) = \frac{w_i(z)}{\sum_j w_j(z)}\ , \quad \psi_i(z_j) = \delta_{ij}\ . \tag{7}$$

The Shepard interpolant of data $\big[z_i, f(z_i), Df(z_i), \cdots, D^{(n)}f(z_i)\big]$, where D denotes differentiation, has the form

$$s(z) = \sum_i Pf^{(n)}(z - z_i)\psi_i(z)\ , \tag{8}$$

where $Pf^{(n)}(z-z_i)$ is the n-th degree Taylor polynomial of f about $z = z_i$. With $n = 0$ the interpolant has flat spots (zero gradient) at the sites, which gives too much noise in our application. Fortunately we can go to higher n, since we know derivatives at time 0 and can update them by formulas derived by differentiating (5). The choice $n = 2$ is quite tractable. It should be mentioned that we actually put a cut-off factor in the $w_i(z)$ to avoid summing over all sites; this does not affect the matter of interpolation [4, 2].

NUMERICAL RESULTS

We take the wake function for the SLAC damping rings, with a current $I_c = 0.0838pC/V$, $N = 3.22 \cdot 10^{10}$, large enough to make the Haïssinski equilibrium unstable [7], and integrate with 1024 time steps per synchrotron period. For the LDS $\{z_i(0)\}$ we choose a 2D Sobol sequence of length about 20000. A plot of this sequence is shown in Fig.2, the points being color coded according to their weights $f(z_i(0), 0)$. The distribution of points looks more uniform than that of a typical pseudo-random sequence which will show noticeable clustering. We use Shepard interpolation with $n = 2$, and find that over 40 synchrotron periods the Sobol points evolve to the set shown in Fig.3, which exhibits noticeable clustering. In Fig.1 we see that over 5 periods the mean local interpolation error for the forward LDS with 20000 points is comparable to that for the backward grid-based method with 160000 points, in spite of the $1 : 8$ ratio in the number of sites. This comparison is encouraging, especially since we think that comparisons will be much more favorable in higher dimensions. The collective force (Fig.4) and charge density agree closely with the grid-based results.

We see the anticipated defect of the forward method, however, in that the calculation goes unstable at about 43 periods. This seems to be due to clustering of sites, so that the evolved data show high gradients and are increasingly hard to interpolate. Other deficiencies of the code may be implicated as well. We are studying schemes for successive redefinitions of the data sites, which may help to maintain low interpolation error. Investigation of a backward scheme with LDS is also on the agenda.

Acknowledgment: Thanks to Greg Fasshauer and Antonio Ferreira for discussions and warm hospitality at the 2nd Eccomas Conference on Meshless Methods in Porto.

REFERENCES

[1] H. Niederreiter, "Random Number Generation and Quasi - Monte Carlo Methods", (SIAM, 1992); R. Caflisch, Acta Numerica **7**, 1-50 (1998).

[2] G. Fasshauer, "Meshfree Approximation Methods with MATLAB", (World Scientific, 2007).

[3] H. Wendland, "Scattered Data Approximation", (Cambridge U. Press, 2005).

[4] M. D. Buhmann, "Radial Basis Functions", (Cambridge U. Press, 2003)

[5] A. Iske, "Multiresolution Methods in Scattered Data Modelling", (Springer, 2004).

[6] See papers cited by Niederreiter, *op. cit.*, p.11 , and S. Wollman, CUNY, unpublished note.

[7] R. L. Warnock and J. A. Ellison, SLAC-PUB-8404 (2000)

[8] M. Vogt, T. Sen, and J. A. Ellison, Phys. Rev. ST AB **5**, 024401 (2002).

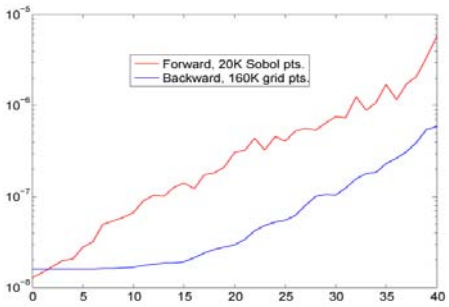

Figure 1: Mean local error for forward method with 20K Sobol sites (red) and backward method with 160K grid sites (blue), vs. time in periods.

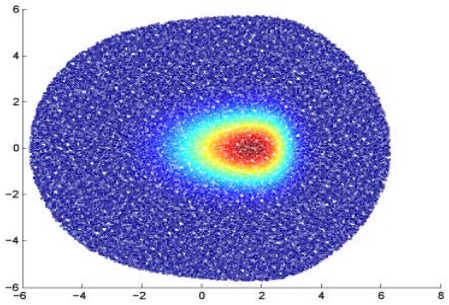

Figure 2: Initial Sobol points with color coded weights.

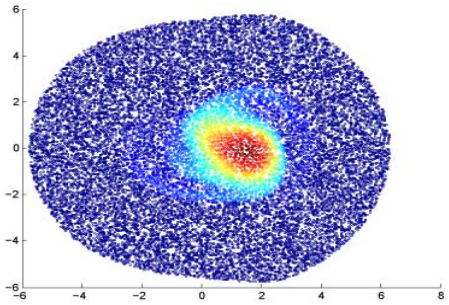

Figure 3: Evolved Sobol points at period 40, with evident clustering.

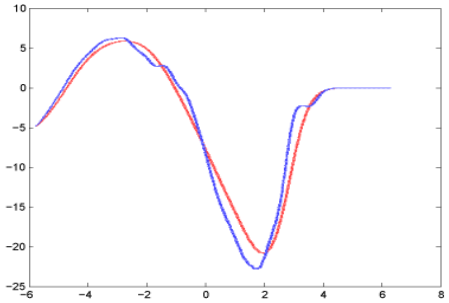

Figure 4: Collective force at start (red) and at 40 periods (blue) as a function of q.