RAPID INTEGRATION OVER HISTORY IN SELF-CONSISTENT 2D CSR MODELING*

Klaus Heinemann, David Bizzozero, James A. Ellison, Stephen R. Lau
Mathematics and Statistics, University of New Mexico, Albuquerque, NM, USA
Gabriele Bassi, Brookhaven National Laboratory, Upton, NY, USA

VLASOV-MAXWELL APPROACH TO BUNCH COMPRESSORS

This paper discusses three strategies in our work on the Vlasov-Maxwell (VM) system for a bunch compressor: the current paradigm, a modified paradigm, and a future paradigm. Each strategy only requires knowledge of the fields in the bunch. In the current paradigm we reduce the field calculation to a 2D integral over the 2D charge and current densities of the bunch and their time history. The current paradigm is implemented in our code VM3@A (**V**lasov-**M**axwell **M**onte-Carlo **M**ethod at **A**lbuquerque) using a time stepping algorithm. Here a major expense is in the integration over history at each time step. The modified paradigm relies on spatial Fourier transformations to reduce the 2D integral to 1D convolutions (one convolution per mode) over history, where we approximate the convolution kernel by a sum of exponentials. As a result the history dependence is effectively localized in time, and each time step then relies only on field information at the present and previous time steps. We present a toy model for the modified paradigm and results are encouraging. We then briefly discuss a future paradigm which involves a 3D Maxwell simulation, using a domain $\mathcal{D}$ which snuggly surrounds the bunch. Such a simulation will require radiation boundary conditions set on $\partial\mathcal{D}$.

The exact problem we deal with in all three strategies is the IVP for the coupled system consisting of the 6D Vlasov equation for the phase space density f and the Maxwell equations for the self field $\mathbf{E}, \mathbf{B}$ with boundary conditions on two perfectly conducting shielding plates. The initial self field is zero for the current paradigm and its modification. Details of the VM approach and the current paradigm are found in [1, 2, 3].

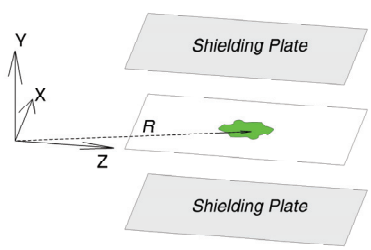

Figure 1: Geometry of sheet bunch model

* Work supported by US DOE grant DE-FG02-99ER41104

CURRENT PARADIGM

The current paradigm and its modification are based on a special case of the VM system, the *sheet bunch model*, where the bunch is confined to the mid-plane between the two shielding plates; whence

$$f(\mathbf{R}, Y, \mathbf{P}, P_Y; u) = \delta(Y)\delta(P_Y) f_{sheet}(\mathbf{R}, \mathbf{P}; u)$$
$$\rho_f(\mathbf{R}, Y; u) = \delta(Y)\rho_{sheet}(\mathbf{R}; u)$$
$$\mathbf{j}_f(\mathbf{R}, Y; u) = \delta(Y)\mathbf{j}_{sheet}(\mathbf{R}; u) \ ,$$

where $u = ct$, $\mathbf{R} = (Z, X), \mathbf{P} = (P_Z, P_X)$, and $\rho_f, \mathbf{j}_f$ are the charge and current densities associated with f. Our setup is shown in Fig. 1.

For the sheet bunch model $\mathbf{E}, \mathbf{B}$ are only needed in the part of the $Y = 0$ plane occupied by the bunch. Moreover, E_Y, B_X, B_Z vanish in the $Y = 0$ plane, and so we only need E_Z, E_X, B_Y. We define

$$\mathbf{F}(\mathbf{R}; u) := (E_Z(\mathbf{R}, 0; u), E_X(\mathbf{R}, 0; u), B_Y(\mathbf{R}, 0; u)) \ ,$$

and the exact solution in the $Y = 0$ plane is

$$\mathbf{F} = \mathbf{F}_0 + \sum_{m=1}^{\infty} \mathbf{F}_m \ , \quad \text{where} \tag{1}$$

$$\mathbf{F}_0(\mathbf{R}; u) = \frac{-1}{4\pi} \int_0^u \int_{-\pi}^{\pi} dv d\theta \mathbf{S}(\mathbf{R} + \mathbf{e}(\theta)(u - v), v) \tag{2}$$

is the nonshielding term and the $\mathbf{F}_m$ are image charge contributions needed to satisfy the boundary conditions. Here, $\mathbf{e}(\theta) = (\cos\theta, \sin\theta)$ and the source $\mathbf{S}$ is determined by $\rho_{sheet}, \mathbf{j}_{sheet}$. The nonshielding term alone is sufficient for some applications.

For the v-integration we use an adaptive Gauss-Kronrod integrator. The θ-integration is done with the trapezoidal rule. The domain of θ-integration is small for most v-values, as shown in Fig. 2 of [1]a.

The evolution of the source is governed by the Vlasov equation. We approximate this by a Monte Carlo particle method in accelerator coordinates [1]. At time u we know the particle phase space positions for $0 \le v \le u$. $\mathbf{F}(\mathbf{R}, u)$ is then computed as an integral over history and used to evolve the particles phase space positions to $u + \Delta$, where Δ is small enough so the fields are constant to good approximation. The new source at $u + \Delta$ is then determined using a density estimation procedure (e.g. kernel density estimation with a product of Epanechnikov kernels [3]) and the process is repeated.

ISBN 978-3-95450-116-8

The self field computation in (2) is the most time consuming part because of the v-integral over the bunch history. The flop count per time step is $O(N_X N_Z N_v N_\theta)$ where $N_X N_Z$ is the number of field grid points and $N_v N_\theta$ is the number of integration grid points. The whole v-integral over the history of the bunch must be done at each time step since $\mathbf{F}(R;u)$ cannot be used to calculate $\mathbf{F}(R;u+\Delta)$ and thus N_v increases with time. Therefore the process $\mathbf{F}(R;u)$ is not Markovian; the future depends on the past not only through the present.

MODIFIED PARADIGM

The main goal of this paper is to propose a modification of the current paradigm. This entails a method to calculate $\mathbf{F}_0(\mathbf{R};u)$ using its spatial Fourier transform $\widetilde{\mathbf{F}}_0(\mathbf{k};u)$ which we hope will speed up the computation. It can be shown that the spatial transforms $\widetilde{\mathbf{S}}(\mathbf{k};u)$ and $\widetilde{\mathbf{F}}_0(\mathbf{k};u)$ of $\mathbf{S}(\mathbf{R};u)$ and $\mathbf{F}_0(\mathbf{R};u)$ are related by

$$\widetilde{\mathbf{F}}_0(\mathbf{k};u) = -\frac{1}{2}\int_0^u dv \widetilde{\mathbf{S}}(\mathbf{k};v) J_0((u-v)|\mathbf{k}|)\ , \quad (3)$$

where J_0 is the zeroth order Bessel function of first kind and $|\mathbf{k}|$ is the Euclidean norm of $\mathbf{k}$. Eq. (3) is still history dependent, but this dependence now resides in J_0 rather than the source.

The key to our approach is the fact that, unlike J_0, an exponential kernel effectively removes the history dependence. To illustrate, consider

$$F(u) = \int_0^u dv g(v)\exp(\beta(u-v))\ , \quad (4)$$

for function g and complex constant β. Then for $\Delta > 0$

$$F(u) = \quad (5)$$
$$\exp(\beta\Delta)F(u-\Delta) + \int_{u-\Delta}^u dv g(v)\exp(\beta(u-v))\ .$$

Although the integral (4) remains history dependent, it solves the initial value problem $F' = \beta F + g(u)$ with $F(0)=0$, and thus is a Markov process.

The above suggests that the integral over the history of the bunch can be essentially localized through an approximation of J_0 by a sum of exponentials:

$$J_0(v) \approx \sum_{n=1}^{N_E} \alpha_n \exp(\beta_n v)\ . \quad (6)$$

Combination of this approximation with Eq. (3) yields

$$\widetilde{\mathbf{F}}_0(\mathbf{k};u) \approx \sum_{n=1}^{N_E} \alpha_n \widetilde{\mathbf{F}}_n(\mathbf{k};u)\ , \quad \text{where}$$
$$\widetilde{\mathbf{F}}_n(\mathbf{k};u) = -\frac{1}{2}\int_0^u dv \widetilde{\mathbf{S}}(\mathbf{k};v)\exp(\beta_n(u-v)|\mathbf{k}|)\ .$$

By analogy with (5) we obtain

$$\widetilde{\mathbf{F}}_n(\mathbf{k};u) = \exp(\beta_n\Delta|\mathbf{k}|)\widetilde{\mathbf{F}}_n(\mathbf{k};u-\Delta)$$
$$-\frac{1}{2}\int_{u-\Delta}^u dv \widetilde{\mathbf{S}}(\mathbf{k};v)\exp(\beta_n(u-v)|\mathbf{k}|)\ . \quad (7)$$

Formula (7) is the center piece of the modified paradigm, and it effectively removes the v-integration over bunch history. At the point in the code where we compute $\mathbf{F}(\mathbf{R},u)$, we know the particle phase space positions. In addition in the modified paradigm we will also know the $\widetilde{\mathbf{F}}_n(\mathbf{k},u-\Delta)$ and $\widetilde{\mathbf{S}}(\mathbf{k},u-\Delta)$. We first compute $\mathbf{S}(\mathbf{R},u)$ from the particle positions, $\widetilde{\mathbf{S}}(\mathbf{k},u)$ by FFT and then $\widetilde{\mathbf{F}}_n(\mathbf{k},u)$ from (7). Since $\widetilde{\mathbf{S}}(\mathbf{k},v)$ is slowly varying, the integral in (7) can be done after a linear interpolation of $\widetilde{\mathbf{S}}(\mathbf{k},v)$ using $\widetilde{\mathbf{S}}(\mathbf{k},\cdot)$ at $u-\Delta$ and u. From the $\widetilde{\mathbf{F}}_n(\mathbf{k},u)$ we find $\mathbf{F}(\mathbf{R},u)$ by an IFFT and then evolve the particles to $u+\Delta$. The FFT is key here, and a flop count per time step gives $O(N_X N_Z N_E \log(N_X N_Z))$ in contrast to $O(N_X N_Z N_v N_\theta)$. So $N_E \log(N_X N_Z)$ replaces the typically larger $N_v N_\theta$. In the case of a nonuniform grid an NFFT will be used [4].

Let us briefly comment on the construction of the approximation (6). Due to the asymptotic behavior of J_0, its sum-of-exponentials approximation (6) must break down for very late times. Therefore, we focus on finite-time approximation of J_0, and its Laplace convolution

$$(J_0 * g)(u) = \int_0^u dv J_0(u-v)g(v). \quad (8)$$

If A (here the exponential sum in (6)) approximates J_0, then we have the estimate [5]

$$\|A*g - J_0*g\|_{L_2(0,T)} \leq$$
$$e^{\eta T}\sup_{s\in\eta+\mathrm{i}\mathbb{R}}\left|\frac{\widehat{A}(s)-\widehat{J}_0(s)}{\widehat{J}_0(s)}\right| \|J_0*g\|_{L_2(0,T)}, \quad (9)$$

where $[0,T]$ is the approximation window in time. Since the Laplace transform $\widehat{J}_0(s) = (s^2+1)^{-1/2}$ of $J_0(v)$ is singular for $s = \pm\mathrm{i}$, the shift $\eta > 0$ above must be included. We have chosen $\eta = 10^{-6}$, and then focused on finding $\widehat{A}(s) = \sum_{n=1}^{N_E} \alpha_n/(s-\beta_n)$ to achieve

$$\sup_{s\in\eta+\mathrm{i}\mathbb{R}}\left|\frac{\widehat{A}(s)-\widehat{J}_0(s)}{\widehat{J}_0(s)}\right| < \varepsilon, \quad (10)$$

where ε is a prescribed tolerance.

Our algorithm for constructing the desired rational approximation $\widehat{A}(s)$ to $\widehat{J}_0(s)$ is essentially Xu and Jiang's (XJ) [6] adaptive version of the Alpert, Greengard, Hagstrom (AGH) compression algorithm [5] used in the context of radiation boundary conditions. We further modify the XJ algorithm to incorporate the parity conditions (see Fig. 2), obeyed by the real and imaginary parts of $\widehat{J}_0(s)$ for $s \in \eta + \mathrm{i}\mathbb{R}$, and we have also used quadruple precision arithmetic. Embedded within the XJ algorithm

ISBN 978-3-95450-116-8

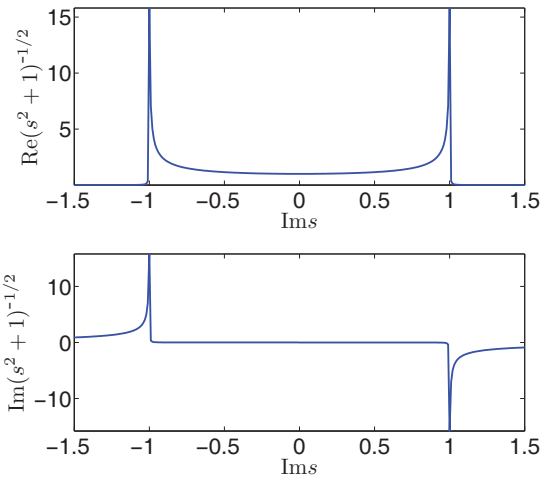

Figure 2: Profiles for $(s^2+1)^{-1/2}$. The real and imaginary profiles are shown along $s = \eta + \mathrm{i}(-\frac{3}{2}, \frac{3}{2})$. For the sake of the figure, here $\eta = 10^{-3}$ rather than $\eta = 10^{-6}$.

is the AGH algorithm for solving the following nonlinear least squares problem (here with $h(s) = \widehat{J}_0(s)$):

$$\min_{P,Q} \int_{\xi_1}^{\xi_2} \left| \frac{P(\eta+\mathrm{i}\xi)}{Q(\eta+\mathrm{i}\xi)} - h(\eta+\mathrm{i}\xi) \right|^2 \mathrm{d}\xi \,, \qquad (11)$$

that is minimization of the integral over the space of polynomials $P(s)$ and $Q(s)$ such that $\deg(Q) = \deg(P) + 1$. The XJ algorithm starts with an adaptive interval refinement of the inversion contour $\eta + \mathrm{i}\mathbb{R}$, one based on the smoothness of $h(\eta + \mathrm{i}\xi)$. The AGH algorithm is then applied recursively from the finest to the coarsest levels, with the goal of first isolating those pole locations β_n (and their corresponding strengths α_n) which lie closest to the contour (thereby somewhat allievating the ill-conditioning of the problem). Once a sum-of-poles approximation has been constructed via the XJ algorithm, we then verify (10). If it is not achieved, then we increment N_E and try again.

We studied the toy model

$$F(\mathbf{R}; u) = \int_0^u dv \int_{-\pi}^{\pi} d\theta G(\mathbf{R} + (u-v)\mathbf{e}(\theta); v) \,,$$

where "the source" is given by

$$G(\mathbf{R}; u) = \exp\Big(-\nu|\mathbf{R} - \mathbf{R}_c(u)|^2\Big) \,,$$
$$\mathbf{R}_c(u) = (a\cos(\omega u), b\sin(\omega u)) \,.$$

The source G is essentially a Gaussian moving on an ellipse. We take $\nu = 5, \omega = 2\pi, a = 1.2, b = 0.8$. In the spirit of the current paradigm and its modification we calculate $F(\mathbf{R}; n\Delta)$ on a $2D$ grid with $64 \cdot 48$ grid points in $Z \cdot X$. Figure 3 displays the expected quadratic growth in CPU time for the current paradigm where we use the quad2D integrator from Matlab. In contrast, in the modified

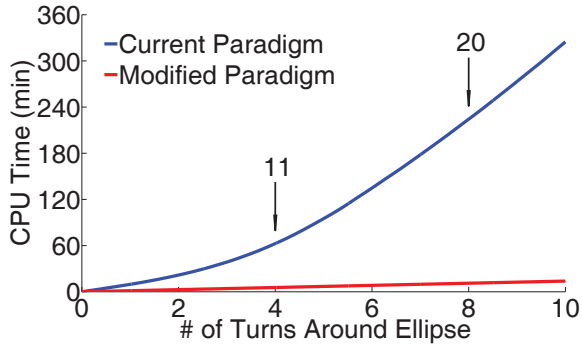

Figure 3: Comparison of CPU times between current paradigm and its modification. The numbers 11 and 20 are ratios of CPU times.

paradigm the CPU time grows linearly. Moreover the CPU times are consistent with flop counts and function evaluation counts. We used $64 \cdot 48$ grid points in $k_Z \cdot k_X$ with maximal $|\mathbf{k}|$-value around 26 so that the needed domain of J_0 is $[0, 260]$. We used $N_E = 56$ where the relative error (with respect to the envelope) of J_0 on the interval $[0, 10^6]$ is ≤ 0.01. The relative error only blows up beyond the interval $[0, 10^7]$.

In summary we have found in the toy model that the CPU time at large times u is considerably less for the modified paradigm and this is consistent with a function evaluation and flop count. For the bunch compressor problem there appears to be a considerable decrease in flop count for the modified paradigm, as discussed after (7). Thus the cpu times and function evaluation counts in the toy model and the flop counts in the current and modified paradigms for the bunch compressor give us some optimism that we can decrease the CPU time in our VM3@A code. Our next step is to revise the toy model to make it more realistic. If this looks good, we'll test the idea in our VM3@A code.

FUTURE PARADIGM

In the approaches discussed above we have replaced the VM system by an integro-differential system which involves the integral over history but only requires the evaluation of fields in the bunch. We now turn our attention to a direct time integration of the VM system of PDEs. This eliminates the integral over history but requires the fields outside the bunch. To deal with the latter we will consider radiation boundary conditions, e.g. as developed in [5]. For spatial discretization we will begin by investigating Discontinuous Galerkin methods [7] and their implementation in the code HEDGE (Hybrid Easy Discontinuous Galerkin Environment, see http://wiki.tiker.net/Hedge). This work will be part of a Ph.D. dissertation project by one of us (D.B.).

ISBN 978-3-95450-116-8

REFERENCES

[1] G. Bassi, J.A. Ellison, K. Heinemann, and R. Warnock, *Microbunching instability in a chicane: Two-dimensional mean field treatment*, Phys. Rev. ST Accel. Beams, **12**, 080704 (2009), and *Transformation of phase space densities under the coordinate changes of accelerator physics*, Phys. Rev. ST Accel. Beams, **13**, 104403 (2010).

[2] G. Bassi, J.A. Ellison, and K. Heinemann, *Self Field of Sheet Bunch: A Search for Improved Methods*. TH2IOPK01 in ICAP09 Proceedings, San Francisco, California. See JACoW.org website.

[3] K. Heinemann, *Two topics in particle accelerator beams*, Ph.D. Dissertation, Math&Stat, University of New Mexico, May, 2010.

[4] J.Keiner, S. Kunis, D. Potts, NFFT, Softwarepackage. http://www.tu-chemnitz.de/∼potts/nfft (20022005).

[5] B. Alpert, L. Greengard, and T. Hagstrom, SIAM J. Numer. Anal. 37, 1138-1164 (2000).
T. Hagstrom, T. Warburton, and D. Givoli, Journal of Computational and Applied Mathematics 234 (2010) 1988.

[6] K. Xu and S. Jiang, *A Bootstrap Method for Sum-of-Poles Approximations*, Journal of Scientific Computing, 2012, DOI: 10.1007/s10915-012-9620-9.

[7] J.S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods* (Springer, 2008).

ISBN 978-3-95450-116-8