SLAC-PUB-9576
November 2002

Two Methods for Simulating the Strong–Strong Beam–Beam Interaction in Hadron Colliders*

Mathias Vogt (UNM), James A. Ellison (UNM),
Tanaji Sen (FNAL), and Robert L. Warnock (SLAC)

Abstract

We present and compare the method of weighted macro particle tracking and the Perron–Frobenius operator technique for simulating the time evolution of two beams coupled via the collective beam–beam interaction in 2–D and 4–D (transverse) phase space. The coherent dipole modes, with and without lattice nonlinearities and external excitation, are studied by means of the Vlasov–Poisson system.

1 INTRODUCTION

Simulations of coherent effects in many particle systems traditionally employ Particle–in–Cell (PIC) methods with an ensemble of macro–particles generated by the Monte Carlo method. We have developed two alternative approaches, the discretized Perron–Frobenius method (PF) and weighted macro–particle tracking (WMPT). We have written several codes, a PF/Fokker–Planck code in one degree of freedom (including diffusion and dissipation) and the hadron codes `BBPF`m`D` (Beam–Beam Perron–Frobenius) and `BBDeMo`m`D` (Beam–Beam Density and Moments, i.e. WMPT) with $m = 1, 2$ in m degrees of freedom. The PF/Fokker–Planck code is described in detail in [1]. Here we will concentrate on the completely symplectic hadron codes.

2 MODELS AND METHODS

2.1 The Ring Model

We assume a ring with one IP at $\theta = 0$ and two counter–rotating bunches. We only treat head–on collisions here and our reference point at which the distribution is studied is directly *before* the IP ($\theta \bmod 2\pi = 0^-$). In what follows we will always use the convention that if some parameter, or dynamical variable X describes one beam, then X^* describes the other beam.

Let $\psi_n(\vec{z})$ and $\psi_n^*(\vec{z})$ ($\vec{z} := (q,p)^{\mathrm{T}}$) denote the normalized phase space densities of the beams at $\theta = 0^- + 2n\pi$ (Here and in the following we use $:=$ or $=:$ if we want to emphasize that the quantity on the left or right respectively is being defined). Then the representations of the one turn map for the unstarred beam from turn n to $n+1$, implemented in `BBPF`m`D` and `BBDeMo`m`D` so far, are

$$\vec{T}[\psi_n^*] = \vec{A} \circ \vec{K}[\psi_n^*] \tag{1}$$

where

$$\vec{A} = \begin{cases} \vec{R} & : \text{linear} \\ \vec{R}^{\frac{1}{2}} \circ \vec{K}_{\mathrm{p}} \circ \vec{R}^{\frac{1}{2}} & : \text{lin. \& pert.} \\ \vec{R}^{-}_{\frac{\pi}{2}} \circ \vec{K}_{\mathrm{s}} \circ \vec{R}' \circ \vec{K}_{\mathrm{s}} \circ \vec{R}^{+}_{\frac{\pi}{2}} & : \text{lin. \& IR-sxt.} \end{cases} . \tag{2}$$

The $\vec{R}$'s represent the linear stable symplectic parts of the lattice ($\vec{R}(\vec{z}) = \underline{R}\,\vec{z}$). At the moment the two degree of freedom versions of our simplified ring models have block diagonal representations of the linear lattice, in other words they do not contain linear coupling. The $\vec{K}$'s are symplectic kicks. In particular, $\vec{K}[\psi_n^*]$ is the collective beam–beam kick on the unstarred beam due to the starred beam, $\vec{K}_{\mathrm{p}}$ is an RF–dipole and/or a multipole kick (up to dodecapole) and $\vec{K}_{\mathrm{s}}$ is an IR–sextupole. In the case of a completely linear lattice, $\underline{R}$ is parameterized by the tune Q and the unperturbed Courant–Snyder parameters β_0 and α_0 at the IP. In the second case, where a perturbation is included in the center of the arc, $\underline{R}^{\frac{1}{2}}$ each have a phase advance of πQ so that $\underline{R} = (\underline{R}^{\frac{1}{2}})^2$. In the third case, $\underline{R}^{-}_{\frac{\pi}{2}}$ and $\underline{R}^{+}_{\frac{\pi}{2}}$ transform the Courant–Snyder parameters at the IR–sextupoles to those at the IP with a phase advance of $\pi/2$ and $\underline{R}'$ is parameterized by the phase advance $2\pi(Q - 1/2)$ and the Courant–Snyder parameters at the IR sextupoles. Note that all three possible ring layouts are mirror symmetric around the axis through the IP and the center of the arc. The different different ring layouts are shown in Figure 1. The collective kick $\vec{K}[\psi]$ is given by

$$\vec{K}[\psi](q,p) = \begin{pmatrix} q \\ p + K[\psi](q) \end{pmatrix} , \tag{3}$$

$$K[\psi](q) = \zeta \int_{\mathbb{R}^m} g(q,q')\rho(q')\, d^m q' \tag{4}$$

with a model dependent kernel function g, a strength parameter ζ and the spatial density $\rho(q) = \int_{\mathbb{R}^m} \psi(q,p)\, d^m p$. The ring models described in Eq. (2) are illustrated in Fig.1, where S.O.D.. . . refers to sextupole, octupole, decapole, etc. and RFD refers to RF–dipole. For the codes in two degrees of freedom, we plan to implement the `MAD`–interpreter and the map generators from the spin code `SPRINT` [2] and the higher order symplectic integrators described in [3].

The particle trajectories are propagated turn by turn via

$$\vec{z}_{n+1} = \vec{T}[\psi_n^*](\vec{z}_n) \ , \ \vec{z}^*_{n+1} = \vec{T}^*[\psi_n](\vec{z}^*_n) \tag{5}$$

and since the maps are measure preserving, the densities evolve via

$$\psi_{n+1}(\vec{z}_{n+1}) = \psi_n(\vec{z}_n) \ , \ \psi^*_{n+1}(\vec{z}^*_{n+1}) = \psi^*_n(\vec{z}^*_n) \ . \tag{6}$$

Presented at Beam-Beam Workshop At Fermilab, 25-27 June 2001, Batavia, IL

*Work supported in part by Department of Energy Contract DE-AC03-76SF00515.
Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94309, USA

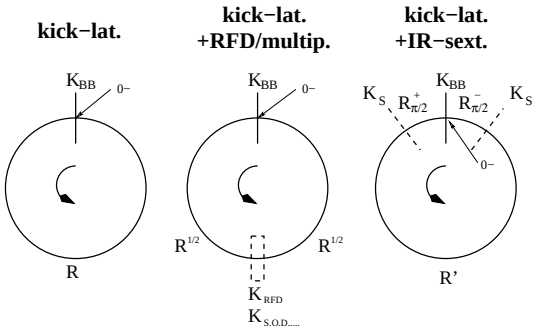

Figure 1: Ring models.

Note that the $\vec{T}$ and $\vec{T}^*$ are explicitly distinguished, allowing for different parameter sets describing the starred and the unstarred lattice. Equations (1–6) define a representation of the beam–beam Vlasov–Poisson system using maps.

In highly relativistic beams the beam–beam force for head–on bunch crossings with zero crossing angle is essentially transverse. In this approximation the collective kick is determined by the solution of a 2–D Poisson problem, $-\Delta u = f$. The Green function in two dimensions for open boundary conditions is given by

$$G(x,y,x',y') = -\frac{1}{2\pi}\log\left(\sqrt{(x-x')^2+(y-y')^2}\right) \quad . \tag{7}$$

We started our study by analyzing three different limits of the beam-beam interaction giving one degree of freedom. Here we shall discuss only the limit studied by Chao and Ruth (CR) [5], which is meant to model beams with large horizontal-to-vertical aspect ratio, a case often found in electron machines. The force on a particle in one beam from the other beam is computed as though it came from infinite planes of charge, perpendicular to the vertical (y) axis, and distributed in y with some density. This force is concentrated in time, however, at the instant of collision. The motion is in the y-direction only. Although not quite appropriate for hadron machines, the model is an attractive starting point because it is the only case in one degree of freedom for which a completely self-consistent calculation can be done with an operation count $\mathcal{O}(N\log N)$ in the WMPT method. Similarly, it is particularly easy to implement in the PF method. The Green function of the corresponding Poisson problem is equal to $-|y-y'|/2$, while its gradient $-\mathrm{sgn}\,(y-y')/2$ is proportional to the kernel of Eq. (4). See [5, 6, 4] for more information on the three limits giving one degree of freedom.

2.2 The Perron–Frobenius Method

The Perron–Frobenius method [1] and weighted macroparticle tracking are both based on the evolution law (6) for the phase space densities given a measure preserving map. We can rewrite Eq. (6) in the form $\psi_{n+1}(\vec{z}) = \psi_n(\vec{T}^{-1}[\psi_n^*](\vec{z}))$ and an analogous equation for ψ_{n+1}^*. This defines the action of the Perron–Frobenius operator associated with (1) on the densities. Now consider a rectangular mesh for $m = 1$: $\{\vec{z}_{ij}\}$, $\vec{z}_{ij} := (i\Delta_q, j\Delta_p)$, where Δ_q and Δ_p are the grid spacings in configuration and momentum space respectively and the integers i and j satisfy $-\frac{n_q}{2} \le i \le \frac{n_q}{2}$ and $-\frac{n_p}{2} \le j \le \frac{n_p}{2}$. In the case of a $2m$ dimensional phase space ($m > 1$), i and j are m–dimensional multi–indices and Δ_q, n_q, etc. are m–dimensional vectors and we have the obvious generalization of the above rectangular grid. Given approximations $\Psi_{ij}(n)$ and $\Psi_{ij}^*(n)$ to $\psi_n(\vec{z}_{ij})$ and $\psi_n^*(\vec{z}_{ij})$ and an l-th order local interpolation scheme ("stencil") $\mathcal{S}_l[f](\vec{z})$ which interpolates values of f at neighboring mesh points of $\vec{z}$, we can update $\Psi(n)$ to $\Psi(n+1)$ by

$$\Psi_{ij}(n+1) = \mathcal{S}_l[\Psi(n)]\left(\vec{T}^{-1}[\Psi^*(n)](\vec{z}_{ij})\right) \quad . \tag{8}$$

For example in the kick–lattice model of Eq. (1) we have $\vec{T}[\psi_n^*] = \vec{R}\circ\vec{K}[\psi_n^*]$ and $\vec{T}^{-1}[\psi_n^*] = \vec{K}^{-1}[\psi_n^*]\circ\vec{R}^{-1}$. To compute this given the $\Psi_{ij}^*(n)$ we first sum over j to get an approximation $\varrho_i^*(n)$ of the spatial density, $\rho_n^*(q_i)$, on the spatial sub–mesh $\{q_i\}$ and then we use $\varrho^*(n)$ to determine the kicks at the mesh points q_i. Spatial interpolation then gives the kicks at all q. Thus we obtain $\vec{T}^{-1}[\psi_n^*](\vec{z}_{ij}) \approx \vec{K}^{-1}[\Psi^*(n)](\underline{R}^{-1}\,\vec{z}_{ij})$ where $\underline{R}$ is the matrix for the linear lattice. These points are not mesh points so we need the interpolant of $\Psi(n)$ in (8). Note that this procedure uses two interpolations where in (8) we only emphasize the latter. This is because the intermediate interpolation is relatively cheap, being an interpolation in configuration space rather than in phase space. The lattice–kick model, $\vec{T}[\Psi^*(n)] = \vec{K}[\Psi^*((n+1)^-)]\circ\vec{R}$ where $\Psi_{ij}^*((n+1)^-) := \mathcal{S}_l[\Psi^*(n)](\underline{R}^{-1}\,\vec{z}_{ij})$, would be more expensive because it would require two interpolations in *phase space* and phase space interpolation is the most expensive part of this calculation.

We want to stress that since $\Psi(n)$ is known, the corresponding spatial density $\varrho(n)$ on the m–dimensional spatial sub–mesh is given by simply summing $\Psi(n)$ over the second, momentum multi–index. If the mesh has $N =: n_g^{2m}$ mesh points in total, then in the *worst case* the $\sqrt{N} = n_g^m$ different kicks on the spatial sub–mesh can be computed by multiplying the $\sqrt{N}$ vector $\varrho(n)$ with a $\sqrt{N}\times\sqrt{N}$ matrix, the discretized kernel, giving a worst case operations count of $O(N)$ for computation of the collective kick. Actually, even if (e.g. in the CR case) the collective kick can be computed less expensively ($O(\sqrt{N}\log N)$), the application of the kick to the N mesh points is, although usually unproblematic, always $O(N)$. In other words a N particle tracking routine with an operations count of less than $O(N)$ is impossible.

We have developed a Vlasov Fokker–Planck code[1] in 2–D phase space for leptons (not restricted to beam–beam interaction) and two hadron codes (without the Fokker–Planck step) in 2–D and 4–D phase space (`BBPF1D` and `BBPF2D`). The order of the interpolation scheme can be chosen, but at least in 2–D phase space quadratic or cubic inter-

polation seems sufficient. In Section 2.5 we will compare PF and WMPT simulations in one degree of freedom and see that with properly chosen mesh parameters the methods are in good agreement, at least in principal aspects.

In 4–D phase space unfortunately the number n_g of mesh points in *each* dimension required to preserve probability to a decent level has to be so large that the 4–D PF algorithm in its serial (single CPU) version is possibly too slow. In particular since the local 4–D interpolation between neighbors along the 4–D mesh axis is not local in the *linear* memory of the computer, it requires accessing array elements with potentially large stride and thus potentially produces a large amount of cache misses (potentially many more than one per updated mesh point). Note that already in a serial code, domain decomposition of the mesh into blocks smaller but comparable with the cache size of the computer would give some (unfortunately hardware dependent, non–portable) relief.

2.3 Weighted Macro Particle Tracking

WMPT [4] is a method for computing time dependent phase space averages of f

$$\langle f\rangle_n := \int_{\mathbb{R}^{2m}} f(\vec{z})\psi_n(\vec{z})\,d^{2m}z \tag{9}$$

$$\langle f\rangle_n^* := \int_{\mathbb{R}^{2m}} f(\vec{z})\psi_n^*(\vec{z})\,d^{2m}z \tag{10}$$

via

$$\begin{aligned}\langle f\rangle_n &= \int_{\mathbb{R}^{2m}} f(\vec{z})\ \psi_0\left(\vec{M}_n^{-1}(\vec{z})\right)\ d^{2m}z \\ &= \int_{\mathbb{R}^{2m}} f\left(\vec{M}_n(\vec{z})\right)\psi_0(\vec{z})\,d^{2m}z \end{aligned}\tag{11}$$

where $\vec{M}_n := \vec{T}[\psi_{n-1}^*]\circ\ldots\circ\vec{T}[\psi_0^*]$ is the symplectic n–turn map containing successive collective kicks. Note that the beam-beam kick function can be written as such an average over the beam–beam kernel with q fixed

$$K[\psi_n^*](q) = \zeta\ \langle g(q,\cdot)\rangle_n^* \quad . \tag{12}$$

Now starting with the initial densities ψ_0 and ψ_0^* defined on an *initial mesh* $\{\vec{z}_{ij}\}$, a quadrature formula with weights w_{ij}, and trajectories $\vec{M}_n(\vec{z}_{ij})$, we can approximate $\langle f\rangle_n \approx \sum_{ij} f(\vec{M}_n(\vec{z}_{ij}))\psi_0(\vec{z}_{ij})w_{ij}$. Of course if we approximate the kicks in Eq. (12) by this method we will only have approximate trajectories $\vec{\eta}_{ij}(n) \approx \vec{M}_n(\vec{z}_{ij})$. Thus our final approximation is

$$\langle f\rangle_n \approx \sum_{ij} f(\vec{\eta}_{ij}(n))\psi_0(\vec{z}_{ij})w_{ij} \quad . \tag{13}$$

Note that this procedure uses only forward tracking of particles with an additional pre–assigned and a constant "total weight" $\mathfrak{W}_{ij} := \psi_0(\vec{z}_{ij})w_{ij}$. Conservation of probability is guaranteed by construction. Also note that the initial mesh structure is lost after the first turn. Thus, in contrast to methods with an explicit mesh (like PF), naive computation of the collective kick is an $O(N^2)$ operation. In the CR case, ordering the trajectories $\vec{\eta}_{ij}(n)$ and $\vec{\eta}_{ij}^*(n)$ with respect to the spatial coordinate at the cost of $O(N\log N)$ makes the remaining part of the computation of the collective kick $O(N)$ (see [4]). In the two degree of freedom case the hybrid fast multipole method (HFMM) [7] allows efficient computation of the kicks ($O(N)$ with a *reasonable* order constant !) as long as the distributions of the trajectories in configuration space are sufficiently regular.

2.4 An Implementation of HFMM for WMPT

HFMM [7] is a hybrid of the fast multipole method (FMM) developed by Greengard and a PIC based reduction of the number of independent particles developed by Jones for space charge and applied the first time to beam–beam simulations by Herr, Zorzano and Jones. FMM is a tree code that allows the computation of the collective force of an ensemble of N charges on themselves to a given accuracy δ with an operations count $O(N)$ given that the distribution of the ensemble in configuration space is not too irregular. It employs the fact that the force on a test charge due to a distant localized "clump" of charge is given by a *finite order* multipole expansion up to precision δ.

The FMM algorithm successively subdivides an outer rectangle in configuration space occupied by the ensemble until, on the finest level of subdivision, no more than a fixed number (typically 40) of particles are in each box. This leads to a tree structure of boxes containing boxes containing boxes and so on, until the boxes on the finest level finally contain a small number of particles. In the non–adaptive version of the scheme all boxes on the same level have the same number of child boxes (weighted tree). We use the adaptive version, which means that a box only branches into child boxes if the box itself still contains too many particles. Then the algorithm computes the multipole (long distance) expansions for all boxes on the finest level explicitly. The next step is to generate multipole expansions around the center of the parent boxes by translating their children's expansions to the center and adding them up so that in the end every box, no matter which level of mesh refinement it belongs to, has its own long distance expansion. Then the far fields inside each box due to all sufficiently well separated boxes are converted to Taylor (local) expansions on each level going down from the coarsest possible level to the finest level. Finally, for each box on the *finest* level, the forces due to charges in a close vicinity have to be computed directly. The 2–D adaptive routines (`DAPIF2`), used in our simulations, were supplied by Greengard. For more details see Greengard in [7].

Unfortunately, FMM needs about 16–18 times N `REAL*8` words of workspace and in addition the order constant of this $O(N)$ algorithm is large enough to be prohibitive for the purpose of multi–turn tracking of many millions of particles.

The original HFMM (Jones) divided the configuration

space into core- and halo domains, superimposed a PIC mesh on the core domain, deposited the core charges on the PIC mesh and passed the joint set of the mesh points and halo particles to FMM. The idea is that if the core is populated densely enough, then the approximation of a PIC type charge collection strongly decreases the required computational resources while not strongly affecting the accuracy of the force computation. On the other hand, the halo particles would not be very well represented on a PIC mesh and in addition would need an unreasonably large mesh to cover the halo.

Our implementation of HFMM in `BBDeMo2D` first determines an outer rectangle in configuration space around the joint starred and unstarred particles. We then divide the rectangle into two parts, a "core" region where the density of particles is high and a "halo" region where the density of particles is small. To do this we divide this rectangle into $n_g \times n_g$ cells. The idea is (as pointed out above) to represent the core by depositing the weights of the particles on the corners of the cells and to represent the halo particles by themselves. In order to find the core and halo regions dynamically (adaptively) we proceed as follows. The population of each cell is determined. Cells with less than a specified minimum number of particles are put into the halo and the remaining cells are put into the core. This determines the number of corners of the core cells, N_c, and the number of halo particles, N_p, and then N_c+N_p is the number of entries in the charge and coordinate lists for the FMM routine. If $N_c + N_p$ is deemed too large, we decrease the minimum number of particles required for a cell to be part of the core and evaluate the cells anew. In general this increases N_c but decreases N_p significantly. Once we are satisfied with the size of $N_c + N_p$, the total weights $\mathfrak{W}^*_{ij}$ of the trajectories of the starred beam are either assigned to the corners of the finally chosen core cells by some PIC scheme (currently "cloud in cell") or, if the trajectories are inside halo cells their weights are kept as single entries in the charge list. In this step, the weights $\mathfrak{W}_{ij}$ of the unstarred beam are not used but the unstarred trajectories inside halo cells are passed to the coordinate list. Then the FMM routine computes the forces due to the starred charges at all the N_c+N_p coordinates. The kicks are either directly applied to the unstarred trajectories in the halo cells or distributed among the unstarred trajectories in the core cells by an interpolation scheme related to the original PIC deposition scheme. Finally, the process is repeated with the role of the starred and the unstarred trajectories interchanged. Note that only the charge list has to be reassigned. The coordinate lists stay the same for both calls to `DAPIF2`.

This method performs the most time consuming step, the FMM, in $O(N_c + N_p)$ operations and only needs 16–18 times (N_c+N_p) `REAL*8` words of workspace. Thus if $N_c+N_p \ll N$, a significant decrease in computation time and workspace is achieved. For example, in a typical `BBDeMo2D` run with $45^4 \approx 4 \cdot 10^6$ macro-particles per beam, we use a PIC mesh of size $150 \times 150 = 22,500$. If we allow $N_c + N_p \leq 5 \times 22,500$, the reduction of entries in the FMM lists is about 70. For a more extreme example with $61^4 \approx 14 \cdot 10^6$ particles per beam the reduction is actually close to 250.

2.5 *Comparison of PF and WMPT*

Both the PF and WMPT methods, in their current implementation in `BBPF`m`D` and `BBDeMo`m`D`, use a uniform (in the case of WMPT *initial*) mesh, but, in principle, both methods allow more general non–uniform meshes. The uniform (initial) mesh which covers a finite rectangular domain in phase space, typically ± 5 initial beam widths (σ_0) for WMPT or ± 6–$7\sigma_0$ for PF, treats its inner (core) and outer ("halo") regions equally. Thus in contrast to conventional macro-particle methods, where the initial beam distribution is typically represented by a Monte Carlo generated ensemble of particles of equal weight and concentrated around the core, the two methods used here are expected to simulate the evolution of the higher order beam moments more accurately. A round–Gaussian ensemble in 2–D phase space with 40,000 particles has about 39,600 (99%) of its particles contained in the $\pm 3\sigma_0$ square. A WMPT or PF ensemble with 200×200 particles or mesh points on a uniform square mesh of size $\pm 5\sigma_0$ has only $120 \times 120 = 14,400$ (36%) of the particles/mesh points in the $\pm 3\sigma_0$ square and 25,600 particles outside. In the current implementation, the `BBPF`m`D` codes use local interpolation in the tensor–product space spanned by the quadratic or cubic polynomials over the mesh axes. All 4 codes use the simplest possible quadrature formula, namely the Gaussian midpoint formula with $w_{ij} = \prod_{i=1}^{2m} \Delta_{z_i}$ independent of i and j.

In several one degree of freedom examples, we have checked agreement between the `BBPF1D` and `BBDeMo1D` codes. Here we study the the turn by turn evolution of the centroids $\bar{q}_n^{\sigma,\pi} := \langle q \rangle_n \pm \langle q \rangle_n^*$, as well as the evolution of the beam emittance $\epsilon_n := \sqrt{C_n^{2,0} C_n^{0,2} - (C_n^{1,1})^2}$, where $C_n^{i,j} := \langle (q - \langle q \rangle_n)^i (p - \langle p \rangle_n)^j \rangle_n$.

Let us summarize the approximations of the time dependent phase space averages used in the two cases

$$\langle f \rangle_n \approx \begin{cases} \sum_{ij} f(\vec{z}_{ij})\ \Psi_{ij}(n)\ w_{ij} & : \text{PF} \\ \sum_{ij} f(\vec{\eta}_{ij}(n))\ \psi_0(\vec{z}_{ij})\ w_{ij} & : \text{WMPT} \end{cases} . \tag{14}$$

Figure 2 shows excellent agreement between the π mode frequencies and the spectra of the σ modes obtained for $\xi = 0.003$ and almost identical linear tunes $Q_0 = \sqrt{5} - 2$ in the CR limit, giving some confidence in the methods. The difference between the two π mode spectra needs further study, but we will comment briefly on this later. Both beams were initially round Gaussians in phase space with the unstarred beam having an initial coherent betatron amplitude of $0.1\sigma_0$. The initial density was represented by a 201×201 square grid over $\pm 5\sigma_0$ in both directions for WMPT and the grid for the PF simulation used 241×241 points over $\pm 6\sigma_0$. Note that the initial mesh used for WMPT only needs to cover the domain in

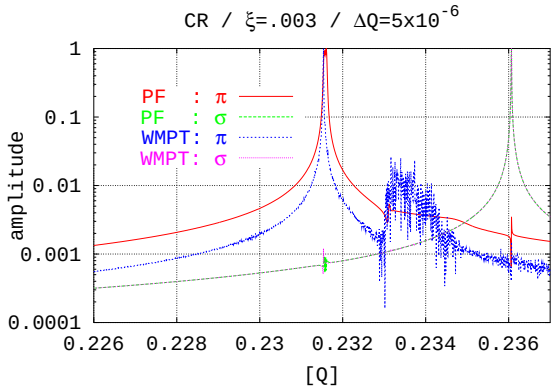

Figure 2: (color) Comparison between PF and WMPT : coherent dipole modes.

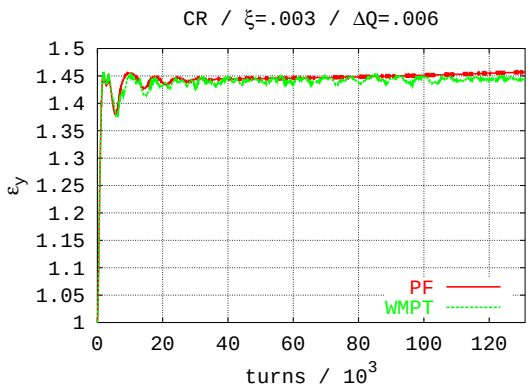

Figure 3: (color) Comparison between PF and WMPT : emittances.

phase space where the *initial* density contributes significantly to the phase space averages. If we want to follow up to 4-th order centered moments of an initially round Gaussian ensemble, then $\pm 5\sigma_0$ around $\langle \vec{z} \rangle_0$ is a reasonable choice, as that is the domain where the functions $(z_k - \langle z_k \rangle_0)^4 e^{-(z_k - \langle z_k \rangle_0)^2/(2\sigma_0^2)}$, $k = 1, 2m$ are not negligibly small. Thus for example the *computed* initial kurtosis $C_0^{4,0}$ with the chosen cut off is close enough (typically 10^{-4}) to its exact value $3\sigma_0^2$. PF on the other hand requires a mesh that is large enough to take care of centroid oscillations and emittance blow up *during* the simulation. The FFT was performed over data from 2^{17} turns. The two σ mode spectra (peak on the right) are indistinguishable and the two π mode spectra have nearly the same tune. The Yokoya factor $(Q_\sigma - Q_\pi)/\xi$ for the Chao–Ruth model comes out higher (≈ 1.51), than in the round beam case [4]. In [8] we have introduced an averaged Vlasov equation and linearization around a Gaussian equilibrium of the averaged system yields exactly this value for the Yokoya factor. We don't understand as yet why the agreement is so good. The incoherent continuum due to the single particle motion is more pronounced in the WMPT spectrum, because it keeps track of N actual trajectories, whereas PF smoothes out the density in each interpolation step. This needs further study. Figure 3 shows the initial emittance growth due to filamentation in an example with a tune split $\Delta Q_0 = 2\xi = 0.006$. The unstarred beam had an initial coherent betatron amplitude of $1\sigma_0$. All other beam and simulation parameters were the same as in Fig. 2. Both simulations agree up to the 1% level, but the general impression is that the time evolutions obtained with WMPT are a little more noisy ("wiggly") then those obtained with PF. We conclude here that with properly chosen mesh sizes both methods agree very well qualitatively and also, to a large extent, quantitatively. Nevertheless, in situations where there are neither large amplitude coherent oscillations nor large emittance growths present the PF method in one degree of freedom is slightly more efficient and stable than WMPT.

Simulations in 4–D phase space (2 d.o.f.) are much more computationally expensive. In conventional macro-particle simulations the phase space ensemble is generated by the Monte Carlo method. There one is tempted to use a relatively small number of macro-particles to gain computational speed, because the actual accuracy of the representation of the density in phase space is somewhat hidden at first sight. But looking at the sampling from the point of a uniform (initial) mesh, suggests that following the evolution of a phase space density over a large number of turns in the presence of a collective force requires a large number of macro-particles or mesh points. Let us assume we want a decent representation of the density on a rectangular domain with $\pm 5\sigma_0$ in 4–D phase space. Then even with moderate beam–beam parameters around $3 \cdot 10^{-3}$, our studies using WMPT with $N = 11^4$ to $N = 61^4$ have shown that one should have at least 30–40 particles *per phase space dimension* and per bunch in order avoid instabilities over 16,000 turns. It is clear that with this large number of macro–particles / mesh points any method for the computation of the collective kicks which has an operations count of more than $O(N)$ is prohibitive, but even with an operations count of $O(N)$ the restrictions imposed by computation time and memory requirements are hard to meet. In the case of `BBDeMo2D`, we employ HFMM as explained in Section 2.4, and a typical (serial) run with 45^4 macro–particles over 2^{14} turns on a SUN ULTRA–80 dual UltraSparc–II workstation with 450 MHz clock, 3GB RAM and 4MB cache takes about 145h (6 days) CPU time.

In the PF case, the interpolation and not the calculation of the collective force determines the performance. With a local cubic 4–D interpolation, the updating of each mesh point touches 256 neighboring mesh points in phase space. As pointed out in Section 2.2, the 4–D structure of the mesh leads to a large amount of cache misses and thus increases the execution time even more. In addition, even with cubic 4–D interpolation and with $N = 51^4 \approx 6.8 \cdot 10^6$ mesh points the conservation of probability is relatively poor. As an example, a `BBPF2D` run with $N = 51^4$ and using cubic 4–D interpolation took over 300h (12 days) for 5000 turns on our ULTRA–80. During this run the density in

the outer mesh region degraded so badly that the computed kurtosis actually became negative. Note that the code had been completely inlined and pre–optimized *by hand* which increased the performance (with the SUN compiler) by a factor of 2. We are working on improved interpolation schemes and other enhancements to speed up the code.

Last but not least, the memory needed to store the main array (the mesh table (PF) or the particle table (WMPT)) for two bunches in m degrees of freedom and in `REAL*8` is

$$\text{Memory}/\texttt{bytes} = \begin{cases} 16 \cdot n_g^{2m} \cdot 2 & : \text{PF} \\ 16 \cdot n_g^{2m} \cdot (2m+1) & : \text{WMPT} \end{cases} . \tag{15}$$

Here n_g is the number of mesh points/particles per dimension, the factor of 16 is 2 bunches times 8 `bytes`, the factor of 2 for PF is due to the fact that we have to store 2 instantiations of the mesh for the purpose of interpolation ("old/new") and the factor of $2m+1$ for WMPT comes from $2m$ phase space coordinates plus the total weight $\mathfrak{W}_{ij}$ of the trajectory.

We will discuss some of our first results with WMPT in 4–D phase space in Section 3.2.

3 SIMULATIONS

3.1 Some Results in One d.o.f.

We have studied extensively the dependence of the coherent dipole modes (π/σ modes) on the split of the bare machine tunes (ΔQ_0) and on the ratio of the beam–beam parameters in one degree of freedom. These studies were reported in [4]. In [8] we report on the the existence of quasi-equilibria for the Chao–Ruth limit. Here we only report some new results based on the implementation of RF–dipoles and higher order multipoles in the lattice and their interaction with the beam–beam kick.

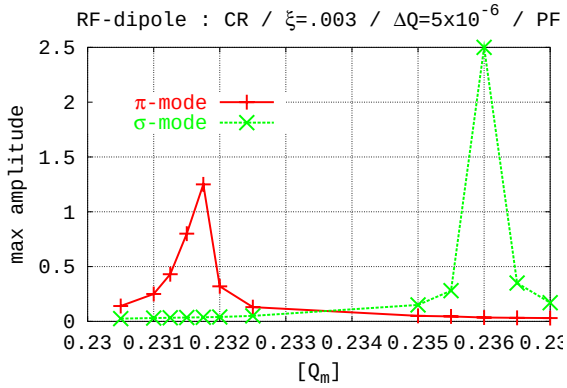

Figure 4: (color) Maximum amplitudes with RF–dipole excitation.

Figure 4 shows the response of the coherent modes to an external excitation with an RF–dipole located at the center of the arc. The simulations were performed with `BBPF1D` and with a mesh of 241×241 points over $\pm 6\sigma_0$. The beam–beam parameter is 0.003 and the tune split is $5 \cdot 10^{-6}$ with $Q_0 \approx Q_0^* \approx 0.236$ (as in Fig. 2) and the normalized kick strength of the RF–dipole was $1 \cdot 10^{-3}$. Both beams had no initial coherent betatron amplitude and the simulation was performed over 2^{17} turns. When the RF-dipole operates at a constant modulation tune $Q_m := f_{\text{dip.}}/f_0$ both modes are excited and their amplitudes are modulated with a tune comparable to $|Q_m - Q_{\pi,\sigma}|$. Figure 4 shows the maximum amplitude of the π mode (red) and the σ mode (green) in the CR limit as a function Q_m. The maximum amplitude is given in units of σ_0. For both modes we find a resonance excitation peak at the frequency predicted in Fig. 2.

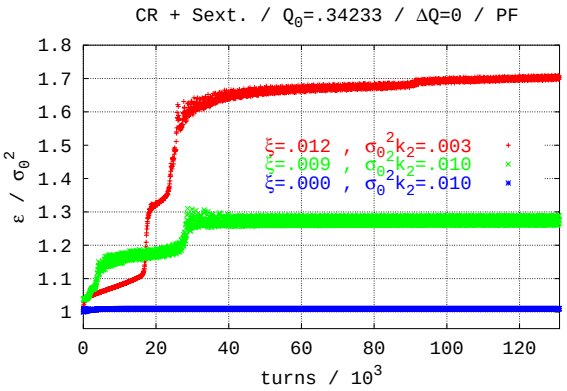

Figure 5: (color) Beam–beam with sextupole: emittance growth.

Figures 5 to 7 show the emittance growth and the instability of the centroid motion induced by the interaction of a strong sextupole kick in the center of the arc and a strong beam–beam interaction in the CR limit close to the third–integer resonance. Both beams were initially round in phase space and one had a $0.1\sigma_0$ offset. The sextupole alone (blue curve) (or with ξ up to 0.006) leads to hardly any emittance growth. However, when the incoherent tune spread reaches $1/3$ ($\xi = 0.009$, green) the emittance is significantly increased. Moreover, in the latter case the π mode amplitude is enhanced from about $0.1\sigma_0$ to about $1.5\sigma_0$ whereas the σ mode amplitude stays small (Fig. 6). Finally when the $1/3$ resonance is well inside the incoherent tune spread ($\xi = 0.012$, red), the emittance grows strongly and the amplitudes of both modes are significantly enhanced (Fig. 7). These observations seem to be consistent with the 4–D PIC simulations presented by Shi and Jin [9] at this workshop.

3.2 First Results in Two d.o.f.

We have simulated the coherent dipole modes in 4–D phase space with `BBDeMo2D` using the HFMM representation of the collective kick and a linear lattice very close to the difference resonance ($Q_x = Q_y =: Q_0$, $Q_x^* = Q_y^* =: Q_0^*$) with $Q_0 = \sqrt{5} - 2$ and $\Delta Q_0 = 5 \cdot 10^{-6}$ (Fig. 8, 9) and $\Delta Q_0 = 6 \cdot 10^{-3}$ (Fig. 10, 11). The beam–beam tune shift

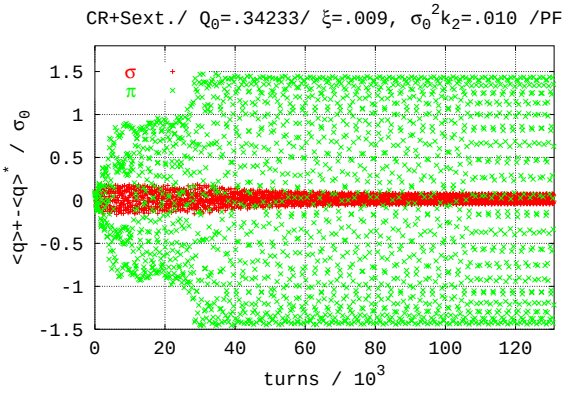

Figure 6: (color) Beam–beam with sextupole: dipole mode oscillations for $\xi = 0.009$ and $k_2 = 0.1\sigma_0^{-2}$.

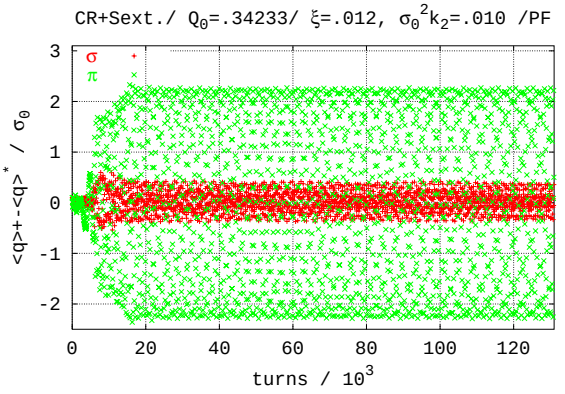

Figure 7: (color) Same as Fig. 6 but with $\xi = 0.012$.

in both cases, given by the extent of the incoherent continuum, is $\xi_x = \xi_y = 0.0036$. Figure 8 shows the spectra of

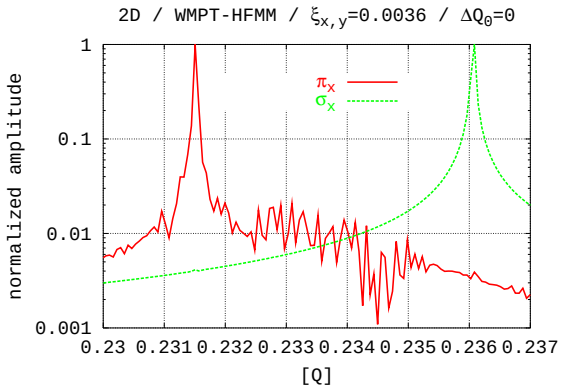

Figure 8: (color) The π and σ mode spectra without tune split.

the π_x and the σ_x mode, $\langle x\rangle_n \mp \langle x\rangle_n^*$ obtained from an FFT over 2^{14} turns. Both beams were initially round Gaussians in their four dimensional phase space and in each plane one of them had an initial coherent betatron amplitude of $0.1\sigma_0$ whereas the other was at rest. Both modes can clearly be resolved. The separation of the π_x mode tune from the σ_x mode tune is $(1.28 \pm 0.02)\xi_x$ with ξ_x being the extent of the incoherent continuum. The spectra of the dipole modes for the vertical motion (not shown) are the same. Figure

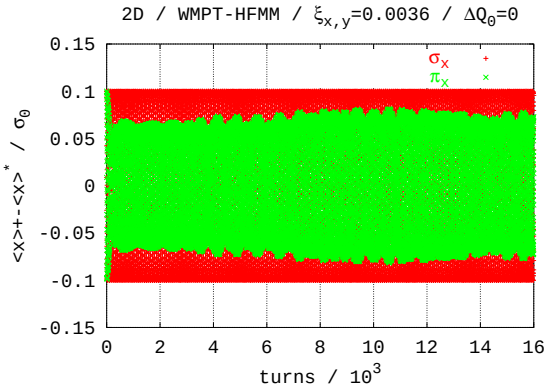

Figure 9: (color) The π and σ modes in time domain without tune split.

9 shows the turn by turn evolution of the dipole modes for the above parameters. The σ mode (red) has a completely stationary amplitude. The π mode (green) amplitude drops slightly to about 80% of its initial value but then stays almost constant. The weak but visible low–frequency modulation of the π mode amplitude seems to be an artifact of the discretization. It is reduced when the number of particles is increased (not shown). The emittances (also not shown) stay constant to the 1% level. This, in combination with our earlier results in one degree of freedom [4, 8], indicates that, for a linear lattice, moderate beam–beam parameters and in the absence of external excitation, the dipole modes are neutrally stable.

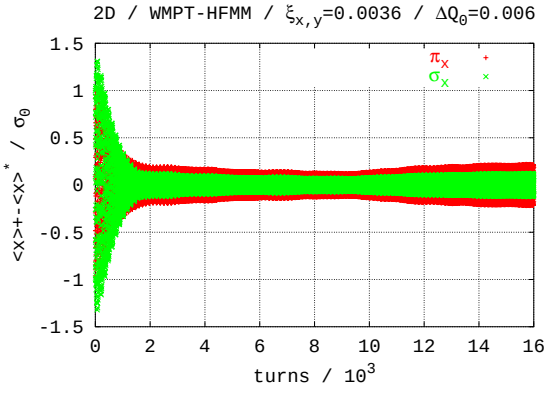

Figure 10: (color) Filamentation of the π and σ modes with large tune split.

Figure 10 and 11 show an example with $\Delta Q_0 = 0.006$, i.e. almost twice the beam–beam tune shift parameter so that the coherent dipole modes should be Landau damped. The initial coherent betatron amplitudes of the beams are $1\sigma_0$, 0 for the horizontal and vertical planes respectively of

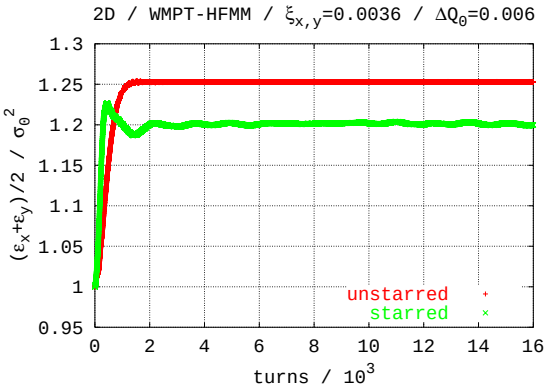

Figure 11: (color)Emittance blow up with with large tune split.

the unstarred beam and 0, $1\sigma_0$ for starred beam. Figure 10 shows the decay of the modes in the horizontal plane. Figure 11 shows the blow up of the emittances because of the filamentation of the dipole modes. The mean $(\epsilon_x + \epsilon_y)/2$ is printed for the unstarred and the starred beam. Both beams stay round up to the 10% level. The unequal emittance blow up of the two beams can most likely be explained by their different tunes ($Q_0 \approx 0.236$, $Q_0^* \approx 0.230$) and thus different sensitivity to the nonlinear perturbation of the beam–beam force.

4 SUMMARY AND OUTLOOK

We have developed two methods, PF and WMPT, for the simulation of nonlinear collective effects in beams described by Vlasov–Poisson systems. Both methods are based on the symplecticity of the one turn map, but PF easily allows extension to Vlasov Fokker–Planck systems by means of operator splitting. We have written codes for simulating the beam–beam interaction in the strong–strong model. WMPT and PF show good agreement in the one degree of freedom limits of the beam–beam and we have extended both methods to the more important two degree of freedom case. At present WMPT is more efficient in higher dimensions. We will try to improve on the efficiency of PF in 4–D phase space and we have already started developing a parallel (distributed memory) version of the WMPT code. Moreover, we will include a `MAD`–reader, higher order maps for real beam line elements, and interpolated higher order generating functions (see [3]) for composed IP–to–IP maps. We will pursue the idea of speeding up of both methods (PF and WMPT) by incorporating our results on averaging [8].

In one degree of freedom, we have studied the dipole modes and Landau damping [4]. In addition, we have studied the response to external excitations (RF–dipole) and have observed large emittance growth together with a strong increase of the amplitudes of the centroid motion in the presence of lattice nonlinearities (e.g. sextupole) combined with a sufficiently large beam–beam parameter.

We have just begun analyzing the dynamics of the beam–beam in 4–D phase space and have begun determining optimal parameter settings for the codes, e.g. n_g, N, etc. as functions of ξ, ξ^*, the distance of the bare tunes to orbital resonances, the total turn number, etc. First tests clearly resolve the dipole modes and their neutral stability in a linear lattice and with moderate tune shift parameters. They also indicate the possibility of introducing Landau damping via a tune split.

5 ACKNOWLEDGEMENTS

We thank L. Greengard for supplying the adaptive FMM routines. Our work was supported by DOE contracts DE-FG03-99ER41104, DE-AC02-76CH03000, DE-AC03-76SF00515 and the US-LHC project.

6 REFERENCES

[1] R. L. Warnock and J. A. Ellison, *Proc. 2nd ICFA Advanced Accelerator Workshop on the Physics of High Brightness Beams, UCLA, November 9-12,1999*, (World Scientific, Singapore, 2001);
Proc. 2nd ICFA Workshop on Quantum Aspects of Beam Physics, Capri, October, 2000, to be published by World Scientific.

[2] G. H. Hoffstaetter and M. Vogt, The `SPRINT` Manual, to appear in 2001.

[3] R. L. Warnock and J. A. Ellison, *Appl.Num.Math.* **29**, 89-98 (1999); Also published in
Beam Stability and Nonlinear Dynamics, ed. Z. Parsa, AIP Conf. Proc. **405**, p.41 (Amer. Inst. Phys., Woodbury, NY, 1997).

[4] M. Vogt, T. Sen, J. A. Ellison, FNAL Pub-01/096-T, submitted to Phys. Rev. ST–AB, (2001).

[5] A. W. Chao and R. D. Ruth, *Particle Accelerators* **16**, 201–216 (1985).

[6] K. Yokoya and H. Koiso, *Particle Accelerators* **27**, 181–186 (1990).

[7] L. Greengard, *Thesis*, The MIT Press, Cambridge, MA (1988);
F. W. Jones, *Proc. Workshop on Space Charge Physics in High Intensity Hadron Rings*, Ed. A. U. Luccio and W. T. Weng, AIP Conf. Proc. **448**, (1998);
W. Herr, M.–P. Zorzano and F. W. Jones, Phys. Rev. ST–AB **4** 054402, (2001);
W. Herr, in these proceedings.

[8] J. A. Ellison and M. Vogt, in these proceedings.

[9] J. Shi and L. Jin, in these proceedings.