Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 198 (2004) 278–294
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A parallel particle-in-cell model for beam–beam interaction in high energy ring colliders

Ji Qiang *, Miguel A. Furman, Robert D. Ryne

Lawrence Berkeley National Laboratory, Accelerator and Fusion Research Division, Ms 71 J-100, 1 Cyclotron Road, Berkeley, CA 94720, USA

Received 30 October 2003; received in revised form 10 January 2004; accepted 13 January 2004
Available online 27 February 2004

Abstract

In this paper we present a self-consistent simulation model of colliding beams in high energy ring colliders. The model, which is based on a particle-in-cell method, uses a new developed shifted effective Green function algorithm for the efficient calculation of the beam–beam interaction with arbitrary separation and large aspect ratio. The model uses transfer maps to treat the external focusing elements and a stochastic map to treat radiation damping and quantum excitation of the beams. In the parallel implementation we studied various strategies to deal with the particular nature of the colliding beam system – a system in which there can be significant particle movement between beam–beam collisions. We chose a particle-field decomposition approach instead of the conventional domain decomposition or particle decomposition approach. The particle-field approach leads to good load balance, reduced communication cost, and shows the best scalability on an IBM SP3 among the three parallel implementations we studied. A performance test of the beam–beam model on a Cray T3E, IBM SP3, and a PC cluster is presented. As an application, we studied the flip–flop instability in an electron–positron collider.

1. Introduction

High energy colliders are essential to study the inner structure of nuclear and elementary particles in modern physics. In a high energy collider, two counter-rotating charged particle beams moving at speeds close to the speed of light collide at one or more interaction points where detectors are located. The rate at which particle production and other events occur inside the detectors depends on a proportionality factor called the *luminosity*. Maximizing luminosity is therefore a key issue in high energy colliders. The electromagnetic interaction between two beams, i.e. the beam–beam interaction, places a strong limit on the luminosity. An accurate simulation of the beam–beam interaction is needed to help optimize the luminosity in high energy accelerators.

* Corresponding author. Tel.: +1-5104952608; fax: +1-5104952323.
E-mail address: jqiang@lbl.gov (J. Qiang).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.01.008

The beam–beam interaction has been studied for many years. However, the extreme computational cost required to accurately and self-consistently model the beam–beam interaction as the beams circulate for many (typically 10^4–10^6) turns has caused most previous studies to use simplified models. Examples include "weak-strong" models, in which only the "weak" beam is affected by the higher intensity "strong" beam [1–4], soft Gaussian models, where one beam is assumed a priori to have a Gaussian shape, or the pancake model, where each beam is assumed to be a single two-dimensional disk perpendicular to the direction of motion of the beam [5–12]. To study the beam–beam interaction fully self-consistently for both beams (i.e. a "strong–strong" formulation), and to include all the physical processes of long range off-centroid interactions, finite beam bunch length effects, and crossing angle collisions, requires computation resources far beyond the capability of current serial computers. As far as we know, there is no previously developed code that can simultaneously handle all of these physical processes accurately. In this paper we present a parallel beam–beam simulation model, with both weak–strong and strong–strong capabilities, that can simulate these physical processes accurately using high performance computers.

The organization of the paper is as follows: The physical model and computational methods are described in Section 2. The parallel implementation is given in Section 3. An application to the study of the flip–flop instability in an electron–positron collider is given in Section 4. We summarize our results in Section 5.

2. Physical model and computational methods

In our model of beam dynamics in an accelerator, each charged particle is characterized by its charge, mass, and phase space coordinates (x, p_x, y, p_y, Δz, $\Delta p_z/p_0$). Here, the independent variable, s, is the arc length along a reference trajectory inside the accelerator, $p_{x,y}$ is the transverse momentum normalized by the total momentum of a reference particle ($p_0 = E_0/c$), $\Delta z = s - ct(s)$ with c the speed of light, $\Delta p_z = |p| - p_0$ with p_0 the absolute momentum value of the reference particle. The motion of particles will be determined by several factors, all of which must be included in the model. Externally applied electromagnetic fields guide the beam and provide transverse and longitudinal focusing. Particles will also lose its energy through synchrotron radiation, a process that involves radiation damping and quantum excitation. The Coulomb interaction among the charged particles within a bunch is negligible due to the cancellation of the electric and magnetic forces at relativistic speeds. However, in the collisions with the oppositely moving beams, the electric and magnetic forces add up. The resulting beam–beam force is a strongly nonlinear interaction that can significantly affect the motion of the charged particles. Fig. 1 gives a schematic plot of two colliding

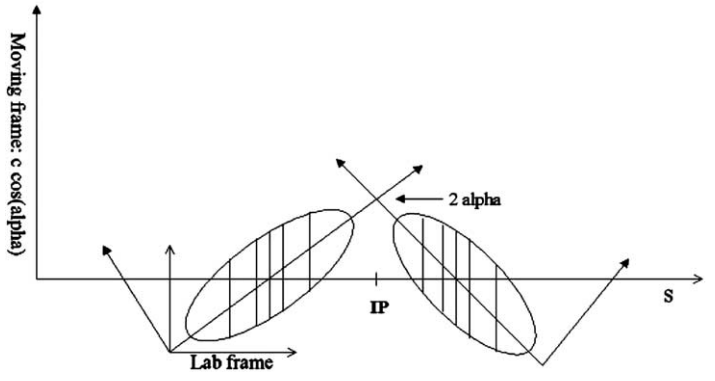

Fig. 1. A schematic plot of the two colliding beams with finite crossing angle.

beams with a finite crossing angle. In the figure, α is the collision crossing angle, and IP is the interaction point.

To calculate the electromagnetic forces from the beam–beam interaction, we have to solve the Maxwell equations in the laboratory frame. Using the Lorentz gauge, the Maxwell equations can be written as

$$\nabla^2\phi - \frac{1}{c^2}\frac{\partial^2\phi}{\partial t^2} = -\rho(x,y,s,t)/\epsilon_0, \tag{1}$$

$$\nabla^2\mathbf{A} - \frac{1}{c^2}\frac{\partial^2\mathbf{A}}{\partial t^2} = -\mathbf{J}(x,y,s,t)\mu_0, \tag{2}$$

where ϕ is the electric scalar potential, $\mathbf{A}$ is the magnetic vector potential, c is the speed of light, ϵ_0 is the vacuum permittivity, μ_0 is the vacuum permeability, ρ is the charge density, and $\mathbf{J}$ is the current density. In the high energy accelerators, the colliding beams moving with a relativistic speed have very small transverse and longitudinal momentum spread. Therefore, we can use a paraxial approximation by assuming all particles moving along the s with speed v_0, $\rho = \rho(x,y,s-v_0t)$, and $J_s = v_0\rho$. Defining a new variable $z = s - v_0t$, Maxwell's equations can be rewritten as:

$$\nabla_\perp^2\phi + \frac{1}{\gamma^2}\frac{\partial^2\phi}{\partial z^2} = -\rho(x,y,z)/\epsilon_0, \tag{3}$$

$$\nabla_\perp^2 A_s + \frac{1}{\gamma^2}\frac{\partial^2 A_s}{\partial z^2} = -J_s(x,y,z)\mu_0. \tag{4}$$

In above equations, the vector potential $\mathbf{A}$ has only the s component since we have neglected the transverse current due to the small momentum spread. From the relationship between the current density J_s and the charge density ρ, we have

$$A_s(x,y,s-v_0t) = \frac{\beta}{c}\phi(x,y,s-v_0t). \tag{5}$$

The electromagnetic fields are obtained from:

$$\mathbf{E} = -\nabla\phi - \frac{\partial\mathbf{A}}{\partial t}, \tag{6}$$

$$\mathbf{B} = \nabla\times\mathbf{A}. \tag{7}$$

Writing in transverse and longitudinal components, we obtain,

$$\mathbf{E}_\perp = -\nabla_\perp\phi, \tag{8}$$

$$E_s = -(1-\beta^2)\frac{\partial\phi}{\partial z}, \tag{9}$$

$$\mathbf{B}_\perp = \nabla\times A_s, \tag{10}$$

$$B_s = 0. \tag{11}$$

For the relativistic colliding beams, $\beta\sim 1$, the longitudinal component of the electric field will vanish, $E_s = 0$. The Lorentz forces are given by

$$\mathbf{F} = q\mathbf{E} + q\mathbf{v} \times \mathbf{B}. \tag{12}$$

Under the paraxial approximation, for the relativistic charged beam, the electric forces and the magnetic forces will cancel each other within the bunch. However, for the colliding beams, moving in the opposite directions, the electric forces and the magnetic forces add up. The resultant electromagnetic beam–beam forces on beam 2 from beam 1 are

$$F_{2x} = q_2 E_{1x}(1 + |\beta_1\beta_2|), \tag{13}$$

$$F_{2y} = q_2 E_{1y}(1 + |\beta_1\beta_2|), \tag{14}$$

$$F_{2s} = 0, \tag{15}$$

where $\beta_1 = v_{0_1}/c$ and $\beta_2 = v_{0_2}/c$. By exchanging the subscript 2 with 1, we can obtain the beam–beam forces on the beam 1 from the beam 2. For the relativistic beams, the γ is very large (typically > 1000), Eq. (3) is reduced to a two-dimensional Poisson equation:

$$\nabla_\perp^2 \phi = -\rho(x, y, z)/\epsilon_0. \tag{16}$$

The equations of motion for the particles are:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v_x, \tag{17}$$

$$\frac{\mathrm{d}p_x}{\mathrm{d}t} = \frac{F_{2x}}{p_0}, \tag{18}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = v_y, \tag{19}$$

$$\frac{\mathrm{d}p_y}{\mathrm{d}t} = \frac{F_{2y}}{p_0}. \tag{20}$$

After each collision, the changes of the transverse momenta are

$$\Delta p_x = \int F_{2x}(x, y, s, t)/p_0 \,\mathrm{d}t, \tag{21}$$

$$\Delta p_y = \int F_{2y}(x, y, s, t)/p_0 \,\mathrm{d}t. \tag{22}$$

The interacting time between two beams is $t = z/c(|\beta_1 - \beta_2|)$, the momentum changes are

$$\Delta p_x = \frac{1}{c(|\beta_1| + |\beta_2|)p_0} \int F_{2x}(x, y, z)\,\mathrm{d}z, \tag{23}$$

$$\Delta p_y = \frac{1}{c(|\beta_1| + |\beta_2|)p_0} \int F_{2y}(x, y, z)\,\mathrm{d}z. \tag{24}$$

Defining an accumulated force $\bar{F} = \int F(x, y, z)\,\mathrm{d}z$ along z, the momentum changes after the beam–beam interaction are:

$$\Delta p_x = \frac{1}{c(|\beta_1| + |\beta_2|)p_0}\bar{F}_{2x}, \tag{25}$$

$$\Delta p_y = \frac{1}{c(|\beta_1| + |\beta_2|)p_0}\bar{F}_{2y}, \tag{26}$$

where

$$\bar{F}_{2x} = q_2\bar{E}_{1x}(1 + |\beta_1\beta_2|), \tag{27}$$

$$\bar{F}_{2y} = q_2\bar{E}_{1y}(1 + |\beta_1\beta_2|), \tag{28}$$

$$\bar{\mathbf{E}}_\perp = \nabla_\perp\bar{\phi}, \tag{29}$$

and

$$\nabla_\perp^2\bar{\phi} = -\bar{\rho}(x,y)/\epsilon_0, \tag{30}$$

where the accumulated transverse charge density is given by

$$\bar{\rho}(x,y) = \int \rho(x,y,z)\,\mathrm{d}z. \tag{31}$$

In Eqs. (21)–(31), we have assumed that the beam–beam interaction is weak and the beam bunch length is short so that transverse positions (x,y) will not change significantly during the collision. In some applications, the beam–beam forces can be strong and the bunch length may not be negligible. We have used a multiple slice model, to calculate the electromagnetic forces from the beam–beam interaction. In this model, each beam bunch is divided into a number of slices along the longitudinal direction in the moving frame as shown in Fig. 1. Each slice contains nearly the same number of particles at different longitudinal locations z. The collision point between two opposite slices i and j is determined by

$$s_c = \frac{1}{2}(z_i^+ - z_j^-), \tag{32}$$

where z_i^+ is the longitudinal centroid location of slice i in one beam, z_j^- is the longitudinal centroid location of slice j in the opposite beam. The transverse coordinates of the particles in the slice at the collision point are given by

$$x^c = x + s_c p_x, \tag{33}$$

$$y^c = y + s_c p_y, \tag{34}$$

where (x,y) are the transverse coordinates of the particles at the interaction point (IP) with $z = 0$. The slopes of the particles are updated using the beam–beam electromagnetic forces at the collision point following

$$p_{x_{\mathrm{new}}} = p_x + \Delta p_x, \tag{35}$$

$$p_{y_{\mathrm{new}}} = p_y + \Delta p_y, \tag{36}$$

where

$$\Delta p_{x_2} = \frac{q_2}{\gamma_2 m_2 c^2} \bar{E}_{x_1}, \tag{37}$$

$$\Delta p_{y_2} = \frac{q_2}{\gamma_2 m_2 c^2} \bar{E}_{y_1}. \tag{38}$$

In the above equations, the subscripts 1 and 2 pertain to each of the two beams, the corresponding equations for the other beam are obtained from the above by the exchange $1 \leftrightarrow 2$, $\gamma = 1/\sqrt{1-\beta^2}$, $\beta_i = v_i/c$, $i = x, y, z$, q is the charge of the particle, m is the rest mass of particle, and $\bar{E}_x$ and $\bar{E}_y$ are the accumulated transverse electric fields generated by the charged particle slice of the opposite moving beam. After the collision, the particles of each slice drift back to their original locations according to

$$x = x^c - s_c p_{x_{\text{new}}}, \tag{39}$$

$$y = y^c - s_c p_{y_{\text{new}}}. \tag{40}$$

The maps described in Eqs. (32)–(40) are not symplectic since they have not included the change of energy deviation during the beam–beam interaction. This is given as [1]:

$$\Delta z_2^{\text{new}} = \Delta z_2, \tag{41}$$

$$(\Delta p_z/p_0)_2^{\text{new}} = (\Delta p_z/p_0)_2 + \frac{q_2}{2\gamma_2 m_2 c^2}\left(\bar{E}_{x_1}\left(p_{x_2} + \frac{1}{2}\Delta p_{x_2}\right) + \bar{E}_{y_1}\left(p_{y_2} + \frac{1}{2}\Delta p_{y_2}\right) + \bar{E}_{z_1}\right), \tag{42}$$

where the longitudinal electrical field $\bar{E}_z$ due to small finite transverse momentum of individual particle is

$$\bar{E}_z = \int \left(p_x \frac{\partial G}{\partial x} + p_y \frac{\partial G}{\partial y}\right) f(\bar{x}, \bar{y}, p_x, p_y)\, \mathrm{d}\bar{x}\, \mathrm{d}\bar{y}\, \mathrm{d}p_x\, \mathrm{d}p_y, \tag{43}$$

where G is the Green's function of Poisson's equation which will be defined in the following and f is the particle distribution in a slice. The calculation of E_z requires two extra convolutions and is computationally expensive. Normally, the transverse momentum spread is small and the beam–beam kick is weak, the change of longitudinal momentum can be negligible. We have implemented it as an option in our computer program and have done a comparison using LHC parameters. There is no visible difference of emittance growth after 50,000 turns with/without longitudinal momentum update.

The electric fields generated by one slice of the opposite moving beam can be obtained from the solution of Poisson's equation. The solution of Poisson's equation can be written as

$$\bar{\phi}(x, y) = \frac{1}{2\pi\epsilon_0} \int G(x, \bar{x}, y, \bar{y}) \bar{\rho}(\bar{x}, \bar{y})\, \mathrm{d}\bar{x}\, \mathrm{d}\bar{y}, \tag{44}$$

where G is the Green's function, $\bar{\rho}$ is the accumulated charge density distribution within a slice, $(\bar{x}, \bar{y})$ is the spatial location of the charged particles. For the case of transverse open boundary conditions, the Green's function is given by:

$$G(x, \bar{x}, y, \bar{y}) = -\frac{1}{2} \ln((x - \bar{x})^2 + (y - \bar{y})^2). \tag{45}$$

Now consider a simulation of an open system where the computational domain containing the particles has a range of $(0, L_x)$ and $(0, L_y)$, and where each dimension has been discretized using N_x and N_y points. From Eq. (44), the electric potentials on the grid can be approximated as

$$\bar{\phi}(x_i, y_j) = \frac{h_x h_y}{2\pi\epsilon_0} \sum_{i'=1}^{N_x} \sum_{j'=1}^{N_y} G(x_i - x_{i'}, y_j - y_{j'}) \bar{\rho}(x_{i'}, y_{j'}), \tag{46}$$

where $x_i = (i-1)h_x$ and $y_j = (j-1)h_y$. This convolution can be replaced by a cyclic convolution in a double-gridded computational domain [13]:

$$\bar{\phi}_c(x_i, y_j) = \frac{h_x h_y}{2\pi\epsilon_0} \sum_{i=1}^{2N_x} \sum_{j=1}^{2N_y} G_c(x_i - x_{i'}, y_j - y_{j'}) \bar{\rho}_c(x_{i'}, y_{j'}), \tag{47}$$

where $i = 1, \ldots, 2N_x$, $j = 1, \ldots, 2N_y$, and

$$\bar{\rho}_c(x_i, y_j) = \begin{cases} \bar{\rho}(x_i, y_j) & : 1 \leqslant i \leqslant N_x; 1 \leqslant j \leqslant N_y, \\ 0 & : N_x < i \leqslant 2N_x \text{ or } N_y < j \leqslant 2N_y, \end{cases} \tag{48}$$

$$G_c(x_i, y_j) = \begin{cases} G(x_i, y_j) & : 1 \leqslant i \leqslant N_x + 1;\ \ 1 \leqslant j \leqslant N_y + 1, \\ G(x_{2N_x - i + 2}, y_j) & : N_x + 1 < i \leqslant 2N_x;\ \ 1 \leqslant j \leqslant N_y + 1, \\ G(x_i, y_{2N_y - j + 2}) & : 1 \leqslant i \leqslant N_x + 1;\ \ N_y + 1 < j \leqslant 2N_y, \\ G(x_{2N_x - i + 2}, y_{2N_y - j + 2}) & : N_x + 1 < i \leqslant 2N_x;\ \ N_y + 1 < j \leqslant 2N_y, \end{cases} \tag{49}$$

$$\bar{\rho}_c(x_i, y_j) = \bar{\rho}_c(x_i + 2(L_x + h_x), \quad y_j + 2(L_y + h_y)), \tag{50}$$

$$G_c(x_i, y_j) = G_c(x_i + 2(L_x + h_x), \quad y_j + 2(L_y + h_y)). \tag{51}$$

These equations make use of the symmetry of the Green function in Eq. (45). From the above definition, one can show that the cyclic convolution will give the same electric potential as the convolution Eq. (46) within the original domain, i.e.

$$\bar{\phi}(x_i, y_j) = \bar{\phi}_c(x_i, y_j) \quad \text{for} \quad i = 1,\ N_x; \quad j = 1,\ N_y. \tag{52}$$

The potential outside the original domain is incorrect but is irrelevant to the physical domain. Since now both G_c and $\bar{\rho}_c$ are periodic functions, the convolution for $\bar{\phi}_c$ in Eq. (47) can be computed efficiently using an FFT as described by Hockney and Eastwood [13].

In the above FFT-based algorithm, the particle domain and the electric field domain are contained in the same computational domain. Here, the particle domain is the configuration space containing the charged particles, and the field domain is the space where the electric field is generated by the charged particles. In the beam–beam interaction, the two opposite moving beams might not overlap with each other. For example, in the long-range interaction, the two colliding beams could be separated by more than several σ, where σ is the rms size of the beam. Thus the field domain where the electric field is generated by one beam can be different from the particle domain containing the beam. Fig. 2 gives a schematic plot of the two separated domains. In this figure, the particle domain has a range from $-R$ to R for x and y, and the field domain has a range from 0 to $2R$ for x and y, where R is maximum extent of the beam. The origin of the field domain in this figure is $x_c = R$, $y_c = R$, where the origin is chosen to be at the beam centroid. In the beam–beam simulation, the origin of the field domain can be at an arbitrary location and varies from turn to turn. To apply Hockney's algorithm directly will require the computational domain to contain both the particle domain and the field domain, i.e. both beams. Since there is a large empty space between two beams, containing both beams in one computational domain will result in a poor spatial resolution of the beams. This is also computationally inefficient because the electric fields in the empty space between two beams are not used.

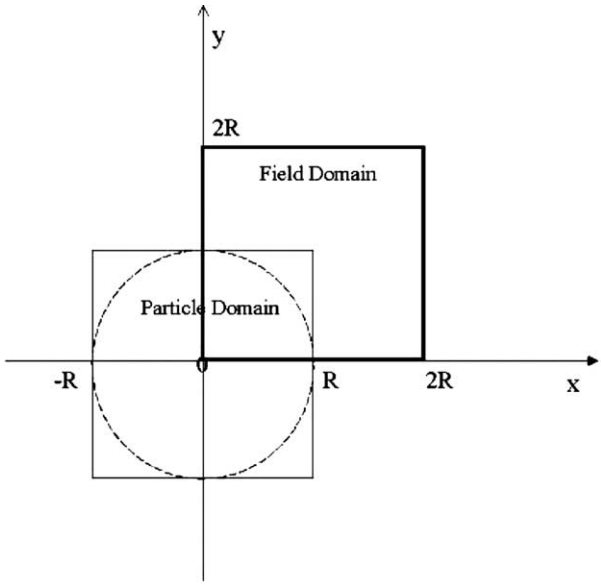

Fig. 2. A schematic plot of the particle domain and the field domain.

To avoid this problem, we have defined a *shifted Green function* as

$$G_s(x,\bar{x},y,\bar{y}) = -\frac{1}{2}\ln((x_c + x - \bar{x})^2 + (y_c + y - \bar{y})^2), \tag{53}$$

where x_c and y_c are the center coordinates of the field domain. The electric potential in the field domain is written as

$$\bar{\phi}(x + x_c, y + y_c) = \frac{1}{2\pi\epsilon_0}\int G_s(x,\bar{x},y,\bar{y})\bar{\rho}(\bar{x},\bar{y})\,d\bar{x}\,d\bar{y}. \tag{54}$$

Using the shifted Green function, the center of the field domain is shifted to the center of the particle domain. The range of x and y cover both the particle domain and the field domain in one computational domain. The FFT can be used to calculate the cyclic convolution in Eq. (47) using the new Green function. Here, on the doubled grids, the Green function is given as

$$G_c(x_i,y_j) = -\frac{1}{2}\begin{cases} \ln((x_c + x_i)^2 + (y_c + y_j)^2) & : 1 \leqslant i \leqslant N_x;\ \ 1 \leqslant j \leqslant N_y, \\ \ln((x_c - x_{2N_x-i+2})^2 + (y_c + y_j)^2) & : N_x < i \leqslant 2N_x;\ \ 1 \leqslant j \leqslant N_y, \\ \ln((x_c + x_i)^2 + (y_c - y_{2N_y-j+2})^2) & : 1 \leqslant i \leqslant N_x;\ \ N_y < j \leqslant 2N_y, \\ \ln((x_c - x_{2N_x-i+2})^2 + (y_c - y_{2N_y-j+2})^2) & : N_x < i \leqslant 2N_x;\ \ N_y < j \leqslant 2N_y. \end{cases} \tag{55}$$

To summarize, using the shifted Green function:

- avoids the requirement that the particle domain and the field domain be contained in one big computational domain,
- leads to better numerical resolution for the charge densities and the resulting electric fields than the conventional method, because the empty space between the beams is not included in the calculation,
- is far more efficient, in terms of computational effort and storage, than the traditional approach of gridding the entire problem domain.

As an example of the above FFT-based algorithms, we have computed the radial electric field distribution generated by a round beam with a Gaussian density distribution using the particle domain and the field domain shown in Fig. 2. Fig. 3 shows the radial electric field E_r as a function of distance along the diagonal line of the field domain using the the shifted Green function, and 128×128 grid. The electric field from the analytical calculation is also given in the same figure for comparison. It is seen that the agreement between the numerical solution and the analytical calculation is excellent.

When the colliding beams have large horizontal (x) to vertical (y) aspect ratio, the straightforward use of the Green function at each mesh point is not efficient since it requires a large number of mesh points along the longer direction in order to get sufficient resolution for the Green function along that direction. An alternative way is to define an effective Green function [14]

$$\bar{\phi}(x_i + x_c, y_j + y_c) = \frac{h_x h_y}{2\pi\epsilon_0} \sum_{i'=1}^{N_x} \sum_{j'=1}^{N_y} \bar{G}_s(x_i - x_{i'}, y_j - y_{j'}) \bar{\rho}(x_{i'}, y_{j'}), \tag{56}$$

where

$$\bar{G}_s(x_i - x_{i'}, y_j - y_{j'}) = \int_{x_{i'}-h_x/2}^{x_{i'}+h_x/2} dx' \int_{y_{j'}-h_y/2}^{y_{j'}+h_y/2} dy' G_s(x_i - x', y_j - y'). \tag{57}$$

This integration can be done analytically using the indefinite integral:

$$\int\int \ln(x^2 + y^2)\, dx\, dy = -3xy + x^2 \arctan(y/x) + y^2 \arctan(x/y) + xy \ln(x^2 + y^2). \tag{58}$$

As a comparison of this effective Green function and the original Green function, we show in Fig. 4 the electric field E_y as function of x along the diagonal line of a field domain from the using of the effective Green function on the mesh, the exact Green function on the mesh, and the analytical error function calculation. The field domain has been shifted one σ along the x axis from the particle domain which has a Gaussian density distribution and horizontal-to-vertical aspect ratio 30. Here, we have used 128×128 mesh points. We see that the fields based on using the effective Green function agree well with the analytical

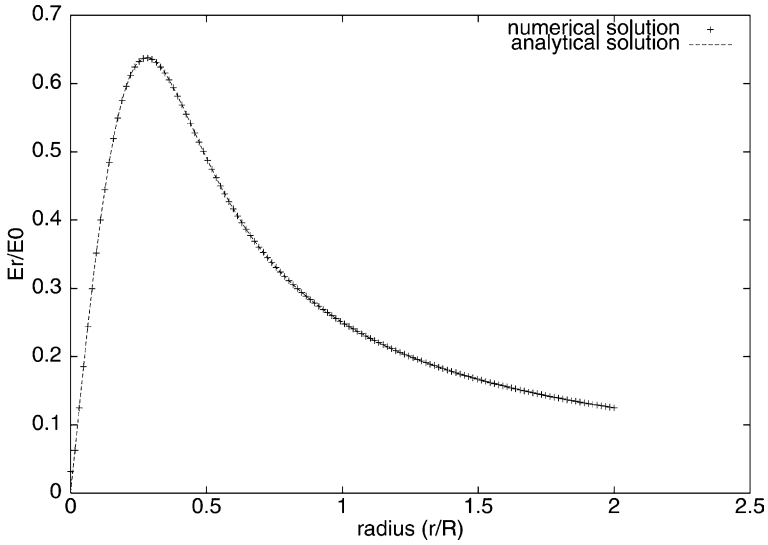

Fig. 3. Radial electric field as a function of distance along the diagonal line of the particle domain.

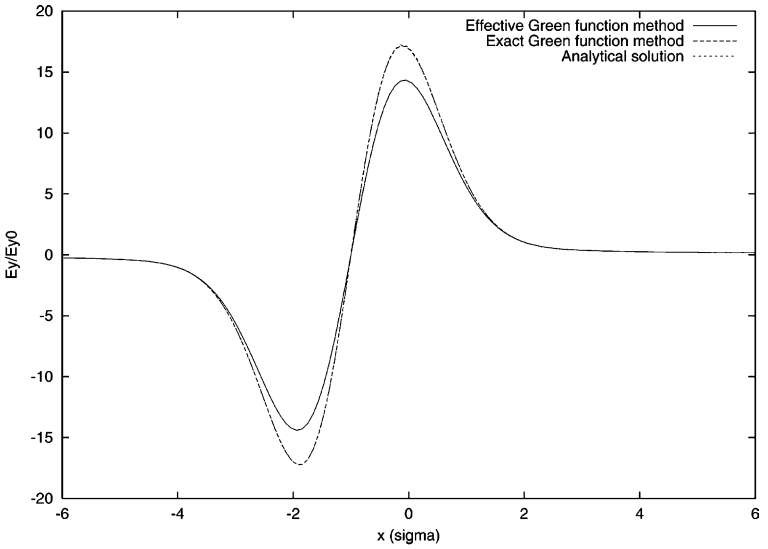

Fig. 4. Electric field E_y as a function of distance along the diagonal line of the field domain. Here, the field domain has one σ separation with the particle domain along x axis.

solution. There is significant discrepancy between the solution using the original Green function and the analytical calculation. The effective Green function defined in above equation is based on the assumption of a uniform charge density distribution within a cell. More accurate expression can be obtained using a linear interpolation of the density distribution within the cell [15].

The effects of external focusing fields can be represented, in the small-amplitude approximation, by a one-turn linear map, i.e.

$$x_{n+1} = (\cos(2\pi\nu_{0x}) + \alpha_x \sin(2\pi\nu_{0x}))x_n + \beta_x \sin(2\pi\nu_{0x})p_{x_n}, \tag{59}$$

$$p_{x_{n+1}} = -\gamma_x \sin(2\pi\nu_{0x})x_n + (\cos(2\pi\nu_{0x}) - \alpha_x \sin(2\pi\nu_{0x}))p_{x_n}, \tag{60}$$

where α_x, β_x and γ_x are lattice functions at the interaction point, and ν_{0x} is horizontal lattice tune. A similar map applies to the vertical phase space y and p_y by replacing $x \to y$ in above equations. For the longitudinal phase space, the one-turn map is defined by

$$\begin{pmatrix} \Delta z/\sigma_z \\ \Delta p_z/\sigma_{p_z} \end{pmatrix}_{n+1} = \begin{pmatrix} \cos(2\pi\nu_s) & \sin(2\pi\nu_s) \\ -\sin(2\pi\nu_s) & \cos(2\pi\nu_s) \end{pmatrix} \begin{pmatrix} \Delta z/\sigma_z \\ \Delta p_z/\sigma_{p_z} \end{pmatrix}_n, \tag{61}$$

where σ_z and σ_{p_z} are the rms beam sizes in z and p_z, and ν_s is the synchrotron tune.

The effects of radiation damping and quantum excitation can be represented using a localized stochastic map. For each particle, with lattice function $\alpha_x = \alpha_y = 0$, the map consists of the following transformations [5]:

$$x_{n+1} = \lambda_x x_n + r_1 \sigma_x \sqrt{1 - \lambda_x^2}, \tag{62}$$

$$p_{x_{n+1}} = \lambda_{p_x} p_{x_n} + r_2 \sigma_{p_x} \sqrt{1 - \lambda_{p_x}^2}, \tag{63}$$

$$y_{n+1} = \lambda_y y_n + r_3 \sigma_y \sqrt{1-\lambda_y^2}, \tag{64}$$

$$p_{y_{n+1}} = \lambda_{p_y} p_{y_n} + r_4 \sigma_{p_y} \sqrt{1-\lambda_{p_y}^2}, \tag{65}$$

$$\Delta z_{n+1} = \lambda_z \Delta z_n + r_5 \sigma_z \sqrt{1-\lambda_z^2}, \tag{66}$$

$$\Delta p_{z_{n+1}} = \lambda_{p_z} \Delta p_{z_n} + r_6 \sigma_{p_z} \sqrt{1-\lambda_{p_z}^2}, \tag{67}$$

where the σ's are the nominal rms equilibrium beam sizes in each dimension, the λ's are given in terms of the damping time τ (measured in units of turns) by $\lambda_i = \exp(-1/\tau_i)$ where i denotes x, y, or z, and the r's are independent random numbers satisfying

$$\langle r_i \rangle = 0, \tag{68}$$

$$\langle r_i r_j \rangle = \delta_{ij}. \tag{69}$$

The first term in the above transformation represents the radiation damping, and the second term represents the quantum excitation.

When two beams approach each other with a finite angle, a transformation is used to change the crossing angle collision in the laboratory frame into a head-on collision in the boosted moving frame. The transformation is given by [4]

$$x^* = x(1 + h_x^* \cos(\psi)\sin(\alpha)) + y h_x^* \sin(\psi)\sin(\alpha) + z\cos(\psi)\tan(\alpha), \tag{70}$$

$$y^* = x h_y^* \cos(\psi)\sin(\alpha) + y(1 + h_y^* \sin(\psi)\sin(\alpha)) + z\sin(\psi)\tan(\alpha), \tag{71}$$

$$z^* = x h_z^* \cos(\psi)\sin(\alpha) + y h_z^* \sin(\psi)\sin(\alpha) + z/\cos(\alpha), \tag{72}$$

$$p_x^* = p_x/\cos(\alpha) - h\cos(\psi)\tan(\alpha)/\cos(\alpha), \tag{73}$$

$$p_y^* = p_y/\cos(\alpha) - h\sin(\psi)\tan(\alpha)/\cos(\alpha), \tag{74}$$

$$p_z^* = p_z - p_x\cos(\psi)\tan(\alpha) - p_y\sin(\psi)\tan(\alpha) + h\tan^2(\alpha), \tag{75}$$

where ψ is the crossing plane angle in the $x-y$ plane, α is the half crossing angle in the $\tilde{x}-z$ plane, $h = p_z + 1 - \sqrt{(p_z+1)^2 - p_x^2 - p_y^2}$, $h_i^* = \frac{\partial}{\partial p_i^*} h^*(p_x^*, p_y^*, p_z^*)$, and $h^*(p_x^*, p_y^*, p_z^*) = h(p_x^*, p_y^*, p_z^*)$.

3. Parallel implementation

Following the above physical model, we have used a particle-in-cell method to calculate the electromagnetic fields at the beam–beam interaction point. Outside the interaction point, the particles are transported through the accelerator using the one-turn lattice map and the radiation damping/quantum excitation map. The fact that the lattice map can cause significant particle movement has important ramifications for the parallelization strategy. During the development of the code we have studied the

performance of three methods: domain decomposition, particle decomposition, and particle-field decomposition.

In the domain decomposition approach the spatial domain is divided into a number of subdomains, with each subdomain mapped onto a single processor [16,17]. The particles with their spatial positions inside the subdomain are assigned to that processor. When particles move out of their spatial subdomain, they are sent to the processor containing the corresponding spatial subdomain. After all particles are local to each processor, the Poisson equation is solved on the grid and the particles are advanced using the electromagnetic fields. To improve the efficiency, a load balance scheme can be used to ensure that each processor contains about the same number of particles. The domain decomposition works well when the particles do not move too far from their positions during each time step. This means that only neighboring processor communication is required. However, in the simulation of colliding beams, after each turn the particles can move a long distance due to the action of external maps. (The colliding beam code is a quasi-static particle-in-cell code, not a fully electromagnetic code, there is no Courant condition. The large movement of particles is not an artifact of the numerical implementation, but instead is a physical effect associated with the fact that particles undergo many oscillations as they are transported around the collider between beam–beam collision points.) A lot of communication is required to move these particles to their local processors. Meanwhile, even though the domain decomposition approach can achieve a load balance of particles, the solution of the Poisson equation is not balanced since each processor has a different number of computational grids, i.e. a different size of subdomain.

Perfect load balance can be achieved, and particle movement avoided, by using a particle decomposition approach [18]. In this approach, the particles are uniformly distributed among processors. Each processor contains the whole spatial domain. To solve the Poisson equation, the particles are deposited onto the global computational grid, collected and broadcast to all processors. Each processor now owns the charge density distribution of the whole domain, and the Poisson equation is solved within this domain. Unfortunately this implementation does not take advantage of the parallelism in the solution of the Poisson equation. To overcome this drawback, in this paper we have proposed a particle-field decomposition approach as the strategy that is best suited to the parallel implementation of the particle-in-cell method for modeling colliding beams.

In the particle-field decomposition approach, each processor possesses the same number of particles and the same number of computational grid points, i.e, the same size of spatial subdomain. Fig. 5 shows a schematic plot of the particle-field decomposition among three processors. We see that the global computational mesh has been uniformly distributed among three processors. Each processor also has the same number of particles. The spatial coordinates of the particles on each processor may not stay within the spatial mesh domain of that processor. In the process of solving the Poisson equation, the particles are deposited onto the computational grid to obtain the charge density distribution. For the particles with spatial positions outside the local subdomain, an auxiliary computational grid is used to store the charge density. After the deposition, the charge density stored on the auxiliary grid will be sent to the processor containing that subdomain. With charge density local to each processor, the Poisson equation is solved in parallel on a local subdomain using the shifted Green function method. Since each processor contains the same number of computational grid points, the work load is well balanced among all processors. The solution of the electric potential on the local subdomain is sent to all processors. With the electric potential on each processor, the electric field is calculated on the grid and interpolated onto individual particles of the opposite beam. The particles are advanced using the electromagnetic field and the external maps. Since each processor contains the same number of particles, the work of this process is also well balanced among processors. The volume of communication in the particle-field decomposition approach is proportional to the number of computational grid points instead of the number of moving particles in the domain decomposition approach. Since, in the study of beam–beam interactions, the number of particles is much larger than the number of computational grid points, e.g. 10^6 vs. 10^4, the particle-field decomposition

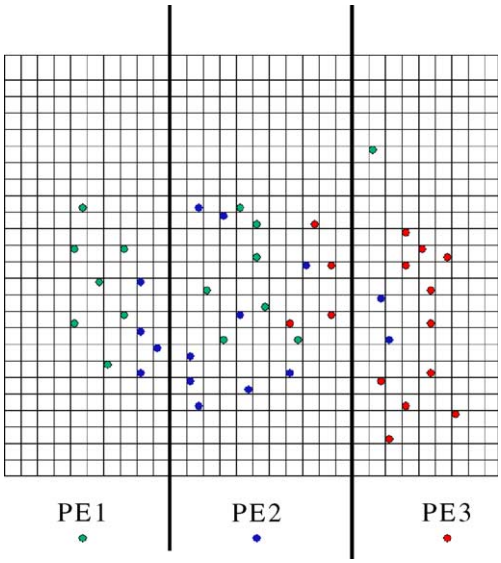

Fig. 5. A schematic plot of the particle-field decomposition among three processors.

approach can significantly reduce the communication cost in the simulation. Fig. 6 shows a comparison of the speedup as a function of number of processors on an IBM SP3 computer using above three parallel implementation approaches for a single slice beam–beam model with one million particles and 128×128 grid points. We see that the particle-field decomposition method has the best scalability among three implementations. The speedup of the domain-decomposition approach saturates at 16 processors due to the large amount of time spent moving the particles among the processors. The particle decomposition

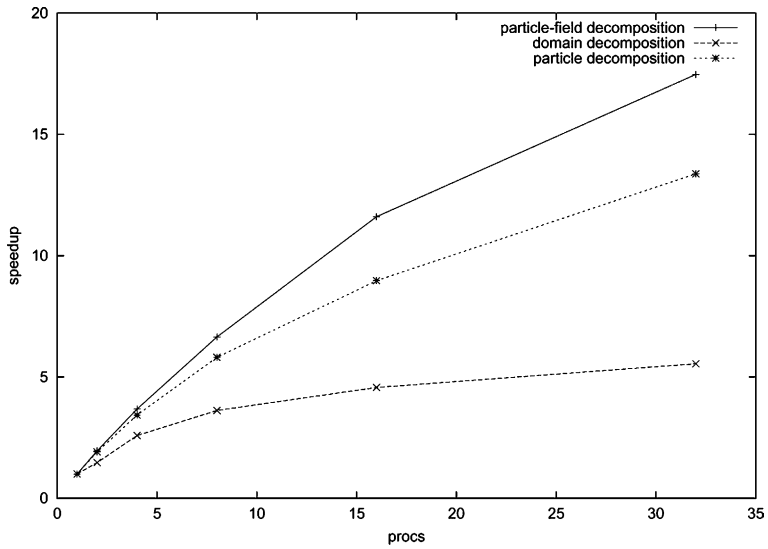

Fig. 6. Speedup as a function of number of processors on IBM SP3 with particle-field decomposition, particle decomposition and domain decomposition.

approach does not take advantage of the parallelism in the field calculation and has a poorer performance than the particle-field decomposition. For example, on 32 processors, the domain-decomposition approach spends 86 s within the particle manager function moving the particles around the processors, while the total beam–beam simulation time is 114 s. In the calculation of electromagnetic fields, it takes about 16 s on 32 processors using the particle decomposition approach. It takes about 6 s using the particle-field decomposition approach. Even though the scalability of the field calculation in the particle-field decomposition approach is not perfect due to the significant communication cost associated with data collecting and transpose, it does show better performance than the particle decomposition approach, where the field calculation is done in serial.

Having adopted the particle-field decomposition approach, we next divide the total number of processors into two groups, with each group responsible for one beam, and each processor in a group containing the same number of particles. We furthermore divide each beam longitudinally into a specified number of slices. The processors in each group are arranged logically into a two-dimensional array with each column of the array containing a number of slices which are assigned to this column of processors cyclically along the row direction. This gives a good load balance of slices among different column processors. Within each column, the computational grid associated with each slice is decomposed uniformly among all the column processors. This provides the parallelization in the solution of the Poisson equation.

As a test of the parallel performance, we have measured the speedup as a function of number of processors on a Cray T3E, IBM SP, and a PC cluster at Lawrence Berkeley National Laboratory. The results are given in Fig. 7. Here, we have used five slices for each beam with two million particles and a computational grid of size 64×64. We see that the program scales up to 128 processors with an efficiency of 65–80% on all three machines. The speedup on the less expensive PC cluster appears to be better than that on the high performance, more expensive parallel machines, which is against general belief. To clarify this issue, we also show in Fig. 8 the total computing time as a function of number of processors on all three machines. We see that the dedicated parallel computers IBM SP3 and Cray T3E are still a factor of 2–3 faster than the PC cluster. A check of the time spent on the communication and computation indicates that in this example, the relative fraction of the communication time in the total computing time on the PC cluster is less than that on the other two parallel computers. For example, on 128 processors, the com-

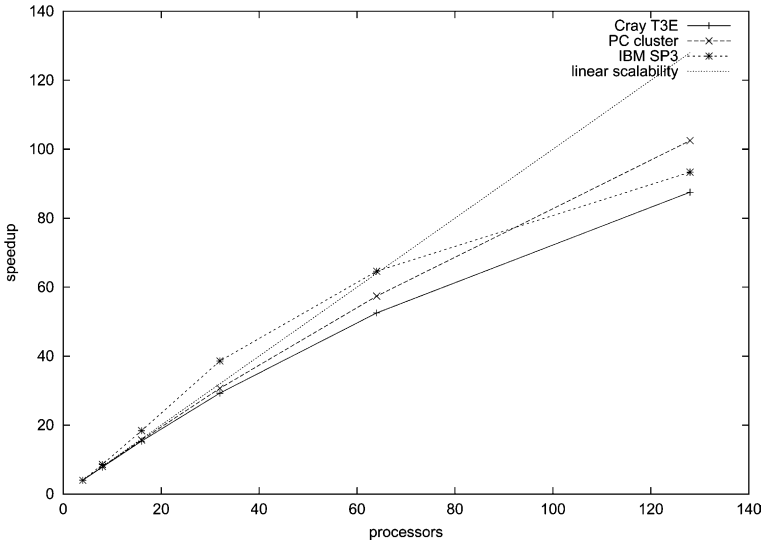

Fig. 7. Speedup as a function of number of processors on Cray T3E, IBM SP3, and PC cluster.

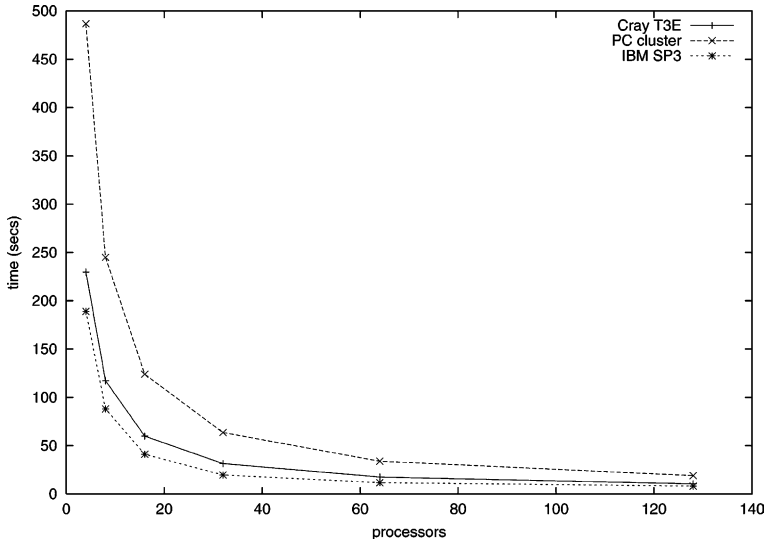

Fig. 8. Total computing time as a function of number of processors on Cray T3E, IBM SP3, and PC cluster.

munication time is 3.4 s out of 19.0 s of total computing time on the PC cluster and 2.9 s out 10.5 s on the Cray T3E. While the communication time is slightly less on the T3E than that on the PC cluster, the computation speed on the T3E is about a factor of two faster than the PC cluster. (Here, the single processor double precision peak performance on the 450 MHz T3E is 900 Mflops while the 866 MHz Pentium III on the PC cluster takes a few cycles to do one double precision operation.) This results in a relatively larger fraction of communication cost on the T3E than that on the PC cluster, which causes the lesser scalability of the T3E for this test example. The speedup on IBM SP3 shows initial superlinear behavior below 64 processors. This superlinear behavior might be due to the finite cache size effects on a small number of processors. The IBM SP3 processor has about 200 KB L1 cache and about 8 MB L2 cache. This corresponds to about 166 K particles in data storage, where each particle occupies 48 bytes. In a case using small number of processors, each processor will hold more particle and field data than the case using a larger number of processors. This leads to more memory access time in the case involving a small number of processors. For example, on four processors, for two million particles, each processor holds 500,000 particles which cannot be stored in the L2 cache. On 32 processors, each processor holds 62,500 particles which can fit in the L2 cache. However, with continuing increase of processor number, the communication time among the processors increases and the speedup falls below linear scalability eventually. For example, on 32 processors of SP3, the communication time is 2.8 s out of the 19.6 s of total computing time. On 128 processors, the communication time is 3.9 s out of 8.1 s of total computing time. The increased communication cost on large number of processors is due to the fact that each node of IBM SP3 has 16 processors but with only two switches to communicate among the nodes.

4. Application

As an application of the parallel particle-in-cell beam–beam model, we have studied the flip–flop instability which has been observed in operating e^+–e^- colliders. During the flip–flop instability, the shape of the beams are distorted. The two beams start out with the same sizes and end up in a state with unequal sizes. The parameters used in this study are similar to those used by Krishnagopal [19]. In that study, the

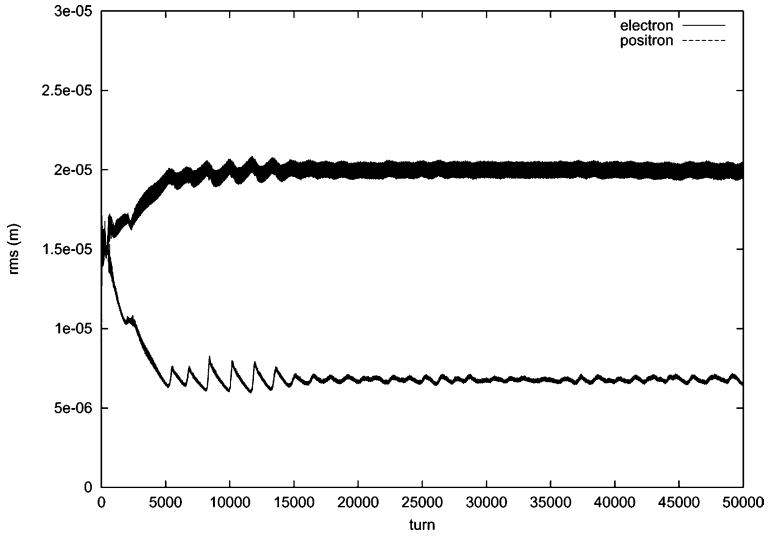

Fig. 9. Vertical rms beam size as a function of turn number in an electron–positron collider with transverse damping time 1000 turn.

aspect ratio was limited to 8 in order to reduce the computational time. However, e^+–e^- colliders can operate with aspect ratios much greater than this. The aspect ratio in real e^+–e^- colliders ranges from 10 to 100, depending on the specific machine parameters. In our study, using the parallel particle-in-cell model with the shifted effective Green function, we can model such high aspect ratios. As an example, Fig. 9 shows the transverse rms sizes of two-colliding beams as a function of turns in an electron–positron ring collider with an aspect ratio of 100. We see that the two beams have the same initial rms sizes. After 50,000 turns, the rms sizes of the beams end up with unequal sizes in the vertical plane. This imbalance results in a reduction of the overlap integral and consequently, in a reduction of the luminosity. Maximizing the luminosity is a key issue in modern colliders, since it affects the number of events observed in the detectors.

5. Summary

In this paper, we have presented a parallel simulation model to study beam–beam effects in high energy colliders. The electromagnetic fields between the two colliding beams are calculated using a parallel particle-in-cell approach with a new developed shifted effective Green function algorithm. Using the shifted effective Green-function, this algorithm allows to calculate the beam–beam forces with large horizontal to vertical aspect ratio and arbitrary separation efficiently. In the parallel implementation, a particle-field decomposition approach has been proposed. This approach shows better scalability than either the domain decomposition approach or the particle decomposition approach due to the nature of the particle movement in beam–beam collision dynamics. A performance test of the multi-slice beam–beam model shows reasonable scalability up to 128 processors on a Cray T3E, IBM SP3 and a PC cluster. As an application, we have studied the flip–flop instability in an electron–positron collider. The appearance of flip–flop instability will reduce the luminosity and should be avoided in the collider design. In conclusion, the development of a parallel code that simultaneously treats the effects of multiple beam–beam phenomena (head-on collisions, long-range collisions, crossing-angle effects, finite bunch length effects, etc.), represents a new and powerful capability that will be useful for understanding and improving the operational characteristics of present and future colliders.

Acknowledgements

We would like to thank Drs. V. Decyk and R. Gerber for helpful discussions about the parallel implementations and parallel performance. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the US Department of Energy (US DOE/SC) under Contract No. DE-AC03-76SF00098, and the resources of the Center for Computational Sciences at Oak Ridge National Laboratory. Some of the computational work for this project was done on the LBL/NERSC Alvarez Cluster, an 80 node Pentium III Myrinet cluster. This work was performed under the auspices of a Scientific Discovery through Advanced Computing project, "Advanced Computing for 21st Century Accelerator Science and Technology," which is supported by the US DOE/SC Office of High Energy and Nuclear Physics and the Office of Advanced Scientific Computing Research.

References

[1] K. Hirata, H. Moshammer, F. Ruggiero, A symplectic beam–beam interaction with energy change, Particle Accel. 40 (1993) 205–228.
[2] K. Hirata, Analysis of beam–beam interactions with a large crossing angle, Phys. Rev. Lett. 74 (12) (1995) 2228–2231.
[3] Y. Papaphilippou, F. Zimmermann, Weak-strong beam–beam simulations for the large hadron collider, Phys. Rev. Special Topics – Accel. Beams 2 (1999) 104001.
[4] L.H.A. Leunissen, F. Schmidt, G. Ripken, Six-dimensional beam–beam kick including coupled motion, Phys. Rev. Special Topics – Accel. Beams 3 (2000) 124002.
[5] M.A. Furman, A. Zholents, T. Chen, D. Shatilov, Comparisons of beam–beam code simulations, CBP Tech Note-59, 1996.
[6] S. Krishnagopal, M.A. Furman, W.C. Turner, Studies of the Beam–Beam Interaction for the LHC, LBNL-43061, CBP Note 308, 1999.
[7] T. Koyama, Single kick approximations for beam–beam deflections, Phys. Rev. Special Topics – Accel. Beams 2 (1999) 024001.
[8] M.A. Furman, Beam–beam simulations with the gaussian code TRS, LBNL-42669, CBP Note 272, 1999.
[9] M.P. Zorzano, F. Zimmermann, Simulations of coherent beam–beam modes at the large hadron collider, Phys. Rev. Special Topics – Accel. Beams 3 (2000) 044401.
[10] M.A. Furman, W.C. Turner, Beam–Beam Simulations for Separated Beams in the LHC, LBNL-46223, CBP Note 350, 2000.
[11] W. Herr, M.P. Zorzano, F. Jones, Hybrid fast multipole method applied to beam–beam collisions in the strong-strong regime, Phys. Rev. Special Topics – Accel. Beams 4 (2001) 054402.
[12] Y. Cai, A.W. Chao, S.I. Tzenov, T. Tajima, Simulation of the beam–beam effects in e^+e^- storage rings with a method of reduced region of mesh, Phys. Rev. Special Topics – Accel. Beams 4 (2000) 011001.
[13] R.W. Hockney, J.E. Eastwood, Computer Simulation Using Particles, McGraw-Hill Book Company, New York, 1985.
[14] K. Ohmi, Simulation of beam–beam effects in a circular e^+e^- collider, Phys. Rev. E 62 (2000) 7287–7294.
[15] R.D. Ryne, A new technique for solving Poisson's equation with high accuracy on domains of any aspect ratio, in: ICFA Beam Dynamics Mini Workshop on Space Charge Simulation, Trinity College, Oxford, 2–4 April 2003.
[16] P.C. Liewer, V.K. Decyk, A general concurrent algorithm for plasma particle-in-cell codes, J. Comput. Phys. 85 (1989) 302.
[17] J. Qiang, R.D. Ryne, S. Habib, V. Decyk, An object-oriented parallel particle-in-cell code for beam dynamics simulation in linear accelerators, J. Comput. Phys. 163 (2001) 434–451.
[18] J.M. Dawson, V.K. Decyk, R.D. Sydora, P.C. Liewer, High performance computing and plasma physics, Phys. Today 46 (3) (1993) 64.
[19] S. Krishnagopal, Energy transparency and symmetrices in the beam–beam interaction, Phys. Rev. Special Topics – Accel. Beams 3 (2000) 024401.