

High-Order Interpolation Algorithms for Charge Conservation in Particle-in-Cell Simulations

Jinqing Yu^{1,2}, Xiaolin Jin¹, Weimin Zhou², Bin Li^{1,*} and Yuqiu Gu^{2,*}

¹ Vacuum Electronics National Laboratory, University of Electronic Science and Technology of China, Chengdu 610054, China.

² Research Center of Laser Fusion, China Academy of Engineering Physics, Mianyang 621900, China.

Received 29 August 2011; Accepted (in revised version) 5 April 2012

Communicated by Song Jiang

Available online 21 September 2012

Abstract. High-order interpolation algorithms for charge conservation in Particle-in-Cell (PIC) simulations are presented. The methods are valid for the case that a particle trajectory is a zigzag line. The second-order and third-order algorithms which can be applied to any even-order and odd-order are discussed in this paper, respectively. Several test simulations are performed to demonstrate their validity in two-dimensional PIC code. Compared with the simulation results of one-order, high-order algorithms have advantages in computation precision and enlarging the grid scales which reduces the CPU time.

PACS: 02.60.-x, 52.65.Rr, 52.65.-y

Key words: High-order algorithms, charge conservation, PIC code, CPU time.

1 Introduction

Particle-in-cell (PIC) codes are widely used in plasma physics and astrophysics because it is simple and straightforward. It is well known that PIC method can be carried out by solving continuity equation instead of Poisson equation [1].

There are several techniques for satisfying the continuity equation [1–7], which are called "charge conservation methods". In references [4–6], the authors introduced a charge conservation method for simple shapes of quasi-particles. As described in references [2, 3], the particle trajectories were divided into straight line segments between

*Corresponding author. *Email addresses:* yujinqing5480@sina.com (J. Q. Yu), jinxiaolin@uestc.edu.cn (X. L. Jin), zhouweimin@gmail.com (W. M. Zhou), libin@uestc.edu.cn (B. Li), yqgu@caep.ac.cn (Y. Q. Gu)

the start and end points. The current density was assigned to each segment and then charge conservation could be achieved for each particle trajectories. In references [1], the author developed a method of density decomposition in Cartesian geometry, which was a new charge conservation method. The method was valid for arbitrary form-factor of particles.

Umeda developed methods for first-order [7] and second-order [8] spline interpolation. In Umeda's methods, a particle trajectory was assumed to be a zigzag line in one time step. The methods could be used without any "IF" statements, which enhanced the speed of computation without any substantial distortion of physics. The method for first-order is widely used in PIC codes because of its simple and straightforward. The method used in second-order spline interpolation is not as simple and straightforward as the method used in first-order. It is well known that higher-order algorithms can reduce the numerical noises and increase the sizes of grid scales [9]. In this paper, we develop two new methods of higher-order algorithm for the condition of a particle trajectory assumed to be a zigzag line in one time step and can also be used without any "IF" statements to enhance the speed of computation, which are simple and straightforward. The methods can be applied to any even-order and odd-order, respectively. And the validity is checked by comparing the results of the two methods with one-order method.

This paper is organized as follows: in Section 2, the zigzag scheme for second-order which can be expanded to any even-order is presented. In Section 3, the zigzag scheme for third-order which can be expanded to any odd-order is considered. In order to check the usability of our algorithms, we compare the simulation results of high-order with the case of one-order in Section 4. The conclusions are summarized in Section 5.

2 Zigzag scheme for second-order spline interpolation

Let us consider the continuity equation in finite differences [1] and reduce it to two dimensions, which can be written as

$$\frac{\rho^{t+\Delta t}(j,k) - \rho^t(j,k)}{dt} + \frac{J_x^{t+\frac{\Delta t}{2}}(j+\frac{1}{2},k) + J_x^{t+\frac{\Delta t}{2}}(j-\frac{1}{2},k)}{dx} + \frac{J_y^{t+\frac{\Delta t}{2}}(j,k+\frac{1}{2}) + J_y^{t+\frac{\Delta t}{2}}(j,k-\frac{1}{2})}{dy} = 0. \tag{2.1}$$

Here dx and dy are the grid spaces and dt stands for one time step. The charge density ρ is made up of form-factors of particles

$$\rho(j,k) = \sum_i q_i S_{j,k}(x_i, y_i). \tag{2.2}$$

Here $q_i, S_{j,k}(x_i, y_i)$ are the charge and form-factor of the i th particle. When a particle move from a location of (x^t, y^t) to another, which can be written as $(x^{t+\Delta t}, y^{t+\Delta t})$ and they are

related as

$$x^{t+\Delta t} = x^t + v_x^{t+\frac{\Delta t}{2}} dt, \quad y^{t+\Delta t} = y^t + v_y^{t+\frac{\Delta t}{2}} dt. \quad (2.3)$$

x^t , $x^{t+\Delta t}$, y^t and $y^{t+\Delta t}$ are scaled to dx and dy . Then we can get the normalized values:

$$x_{old} = \frac{x^t}{dx}, \quad y_{old} = \frac{y^t}{dy}, \quad (2.4a)$$

$$x_{new} = \frac{x^{t+\Delta t}}{dx}, \quad y_{new} = \frac{y^{t+\Delta t}}{dy}. \quad (2.4b)$$

We define

$$j_1 = [x_{old} + 1.5], \quad k_1 = [y_{old} + 1.5], \quad (2.5a)$$

$$j_2 = [x_{new} + 1.5], \quad k_2 = [y_{new} + 1.5], \quad (2.5b)$$

where j_1 , k_1 , j_2 and k_2 are the largest integer values not greater than $x_{old} + 1.5$, $y_{old} + 1.5$, $x_{new} + 1.5$ and $y_{new} + 1.5$ and denote the nearest grid space numbers away from x_{old} , y_{old} , x_{new} and y_{new} . We assume that the particle does not move more than one grid space in one time step, which means $v_x dt < dx$ and $v_y dt < dy$. A point (x_r, y_r) has been defined, which locates between the locations of (x_{old}, y_{old}) and (x_{new}, y_{new}) . The move progress can be divided into the moves in x -direction and y -direction. As a matter of convenience, we only discuss the move progress in x -direction, by which way the move progress in y -direction can be considered. In x -direction, we need think of the following cases:

- (1) A particle moves less than a grid space and does not across the midpoint of j_1 and $j_1 + 1$, which can be seen in Fig. 1. Here x_{old} and x_{new} denote the locations at the time t and $t + dt$, respectively. Under this condition, the point x_r can be expressed as $x_r = (x_{old} + x_{new}) / 2.0$. The move progress can be separated into two parts. The first progress is from x_{old} and x_r and the other is from x_r to x_{new} . The charge flux only contributes to the points $j_1 - 1$, j_1 and $j_1 + 1$ in the first progress, while in the second the charge flux contributes to the points $j_2 - 1$, j_2 and $j_2 + 1$.
- (2) A particle moves across the midpoint of the grid shown in Fig. 2. The locations of the particle at t and $t + dt$ locate the two side of the midpoint. The point x_r can be expressed as $x_r = j_1 - 0.5$ or $x_r = j_2 - 1.5$. The move progress can be separated into the part of from x_{old} and x_r and the part of from x_r to x_{new} . The charge flux only contributes to the points $j_1 - 1$, j_1 and $j_1 + 1$ in the first part, while in the second part, the charge flux contributes to the points $j_2 - 1$, j_2 and $j_2 + 1$.

In y -direction, the move progress can be considered by the same methods above. The particle trajectories of zigzag method in two dimensional for any even-order interpolation can be described in Fig. 3. While in the second-order interpolation method of Umeda,

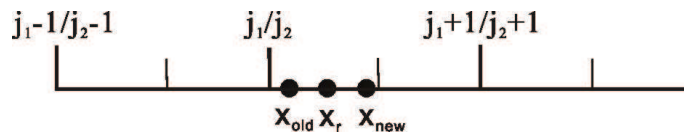


Figure 1: Location of the particle at t and $t+dt$ of Case (1).

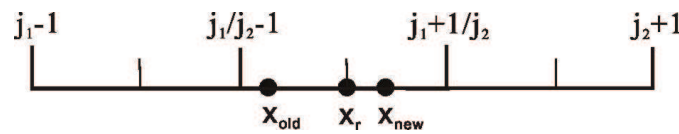


Figure 2: Location of the particle at t and $t+dt$ of Case (2).

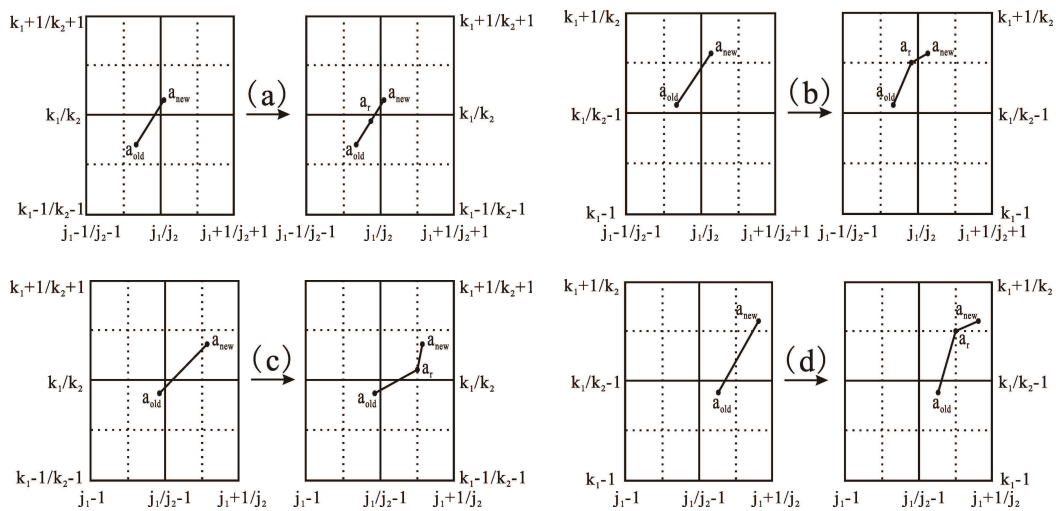


Figure 3: The particle trajectories of zigzag method in two dimensional under the condition of any even-order interpolation.

the point (x_r, y_r) can be expressed as follows [8]:

$$x_r = \begin{cases} \frac{x_1 + x_2}{2}, & (j_1 = j_2), \\ \frac{(j_1 + j_2)\Delta x}{2}, & (j_1 \neq j_2), \end{cases} \quad y_r = \begin{cases} \frac{y_1 + y_2}{2}, & (k_1 = k_2), \\ \frac{(k_1 + k_2)\Delta y}{2}, & (k_1 \neq k_2). \end{cases} \quad (2.6)$$

From the above argumentations, we can get the equations, which can be used to compute the point (x_r, y_r) ,

$$x_r = \min[\min(j_1 - 0.5, j_2 - 0.5), \max(\max(j_1 - 1.5, j_2 - 1.5), (x_{old} + x_{new}) / 2.0)], \quad (2.7a)$$

$$y_r = \min[\min(k_1 - 0.5, k_2 - 0.5), \max(\max(k_1 - 1.5, k_2 - 1.5), (y_{old} + y_{new}) / 2.0)]. \quad (2.7b)$$

After getting the point (x_r, y_r) , a charge flux $q_i(v_x, v_y)$ can be separated into $F_1 = (F_{x1}, F_{y1})$ and $F_2 = (F_{x2}, F_{y2})$. Here F_{x1}, F_{y1}, F_{x2} and F_{y2} can be expressed as:

$$F_{x1} = q_i(x_r - x_{old})dx/dt, \quad F_{x2} = q_i(x_{new} - x_r)dx/dt = q_i v_x - F_{x1}, \quad (2.8a)$$

$$F_{y1} = q_i(y_r - y_{old})dy/dt, \quad F_{y2} = q_i(y_{new} - y_r)dy/dt = q_i v_y - F_{y1}. \quad (2.8b)$$

Here q_i is the quantity of electric charge of the i th particle. We apply shape-factors defined at the midpoints of (x_{old}, y_{old}) and (x_r, y_r) and (x_r, y_r) and (x_{new}, y_{new}) , respectively. Under the case of second-order spline interpolation, we define weighting functions $w_{x1}, w_{x2}, w_{x3}, w_{y1}, w_{y2}$ and w_{y3} for the midpoint of (x_{old}, y_{old}) and (x_r, y_r) and $w_{x4}, w_{x5}, w_{x6}, w_{y4}, w_{y5}$ and w_{y6} for the midpoint of (x_r, y_r) and (x_{new}, y_{new}) , which can be expressed as follows [9,10]:

$$w_{x1} = 0.125(2xx_{11} - 3.0)^2, \quad w_{x2} = 0.75 - xx_{12}^2, \quad w_{x3} = 0.125(2xx_{13} - 3.0)^2, \quad (2.9a)$$

$$w_{y1} = 0.125(2yy_{11} - 3.0)^2, \quad w_{y2} = 0.75 - yy_{12}^2, \quad w_{y3} = 0.125(2yy_{13} - 3.0)^2, \quad (2.9b)$$

and

$$w_{x4} = 0.125(2xx_{21} - 3.0)^2, \quad w_{x5} = 0.75 - xx_{22}^2, \quad w_{x6} = 0.125(2xx_{23} - 3.0)^2, \quad (2.10a)$$

$$w_{y4} = 0.125(2yy_{21} - 3.0)^2, \quad w_{y5} = 0.75 - yy_{22}^2, \quad w_{y6} = 0.125(2yy_{23} - 3.0)^2. \quad (2.10b)$$

Here $xx_{11}, xx_{12}, xx_{13}, xx_{21}, xx_{22}, xx_{23}, yy_{11}, yy_{12}, yy_{13}, yy_{21}, yy_{22}, yy_{23}$ are denoted as:

$$xx_{11} = (x_{old} + x_r)/2.0 - (j_1 - 2), \quad (2.11a)$$

$$xx_{12} = |(x_{old} + x_r)/2.0 - (j_1 - 1)|, \quad (0 \leq xx_{12} \leq 0.5), \quad (2.11b)$$

$$xx_{13} = j_1 - (x_{old} + x_r)/2.0, \quad (2.11c)$$

$$xx_{21} = (x_{new} + x_r)/2.0 - (j_2 - 2), \quad (2.11d)$$

$$xx_{22} = |(x_{new} + x_r)/2.0 - (j_2 - 1)|, \quad (0 \leq xx_{22} \leq 0.5), \quad (2.11e)$$

$$xx_{23} = j_2 - (x_{new} + x_r)/2.0, \quad (2.11f)$$

$$yy_{11} = (y_{old} + y_r)/2.0 - (k_1 - 2), \quad (2.11g)$$

$$yy_{12} = |(y_{old} + y_r)/2.0 - (k_1 - 1)|, \quad (0 \leq yy_{12} \leq 0.5), \quad (2.11h)$$

$$yy_{13} = k_1 - (y_{old} + y_r)/2.0, \quad (2.11i)$$

$$yy_{21} = (y_{new} + y_r)/2.0 - (k_2 - 2), \quad (2.11j)$$

$$yy_{22} = |(y_{new} + y_r)/2.0 - (k_2 - 1)|, \quad (0 \leq yy_{22} \leq 0.5), \quad (2.11k)$$

$$yy_{23} = k_2 - (y_{new} + y_r)/2.0. \quad (2.11l)$$

We define

$$j_{11} = [x_{old} + 1.0], \quad j_{22} = [x_{new} + 1.0], \quad (2.12a)$$

$$k_{11} = [y_{old} + 1.0], \quad k_{22} = [y_{new} + 1.0]. \quad (2.12b)$$

The equations remark that j_{11} , j_{22} , k_{11} and k_{22} are the grid numbers where x_{old} , x_{new} , y_{old} and y_{new} locate. Then we can get the parts of the charge flux contributed to the grid points by the following expressions

$$j_x(k_1 - 1, j_{11} + 0.5) = \frac{1}{\Delta x \Delta y} F_{x1} w_{y1}, \quad j_x(k_2 - 1, j_{22} + 0.5) = \frac{1}{\Delta x \Delta y} F_{x2} w_{y4}, \quad (2.13a)$$

$$j_x(k_1, j_{11} + 0.5) = \frac{1}{\Delta x \Delta y} F_{x1} w_{y2}, \quad j_x(k_2, j_{22} + 0.5) = \frac{1}{\Delta x \Delta y} F_{x2} w_{y5}, \quad (2.13b)$$

$$j_x(k_1 + 1, j_{11} + 0.5) = \frac{1}{\Delta x \Delta y} F_{x1} w_{y3}, \quad j_x(k_2 + 1, j_{22} + 0.5) = \frac{1}{\Delta x \Delta y} F_{x2} w_{y6}, \quad (2.13c)$$

$$j_y(k_{11} + 0.5, j_1 - 1) = \frac{1}{\Delta x \Delta y} F_{y1} w_{x1}, \quad j_y(k_{22} + 0.5, j_2 - 1) = \frac{1}{\Delta x \Delta y} F_{y2} w_{x4}, \quad (2.13d)$$

$$j_y(k_{11} + 0.5, j_1) = \frac{1}{\Delta x \Delta y} F_{y1} w_{x2}, \quad j_y(k_{22} + 0.5, j_2) = \frac{1}{\Delta x \Delta y} F_{y2} w_{x5}, \quad (2.13e)$$

$$j_y(k_{11} + 0.5, j_1 + 1) = \frac{1}{\Delta x \Delta y} F_{y1} w_{x3}, \quad j_y(k_{22} + 0.5, j_2 + 1) = \frac{1}{\Delta x \Delta y} F_{y2} w_{x6}. \quad (2.13f)$$

After getting the charge flux contributed by each particle, the total current densities can be obtained. In any even-order spline interpolation, using the particle trajectories of zigzag method used in the second-order spline interpolation can receive expression similarly to Eq. (2.11) and then any even-order spline interpolation can be expanded to by adopting correspondence weighting function (e.g., the method expanded to fourth-order spline interpolation can be seen in Appendix A), but cannot be used in the case of odd-order spline interpolation.

3 The case of third-order spline interpolation

We consider the condition of odd-order spline interpolation in this section. Third-order condition is introduced as an example. The method used here can be expanded to any odd-order cases. The relation between the initial and new location can be written as Eq. (2.3) and the expressions of x_{old} , x_{new} , y_{old} and y_{new} are the same as Eq. (2.4). We define

$$j_1 = [x_{old} + 1.0], \quad k_1 = [y_{old} + 1.0], \quad (3.1a)$$

$$j_2 = [x_{new} + 1.0], \quad k_2 = [y_{new} + 1.0]. \quad (3.1b)$$

Here j_1 , k_1 , j_2 and k_2 denote the grid space number where x_{old} , y_{old} , x_{new} and y_{new} locate. In the method of third-order spline interpolation, when a particle moves across the cell mesh, the weighting will change. The same as described above, we assume a point (x_r, y_r) and consider the move progress in x -dimension. The following cases should be considered:

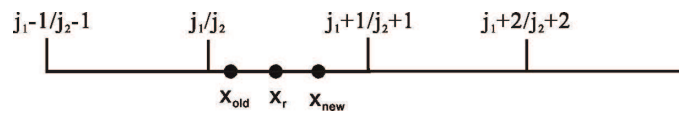


Figure 4: Location of the particle at t and $t+dt$ of Case (1).

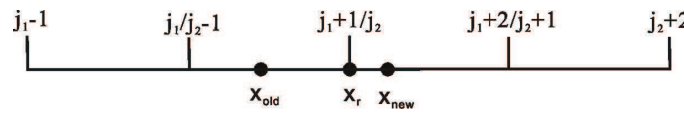


Figure 5: Location of the particle at t and $t+dt$ of Case (2).

- (1) A particle moves less than a grid space and does not move across the cell mesh, which can be seen in Fig. 4. Under this case, the point x_r can be written as $x_r = (x_{old} + x_{new})/2.0$. The progress can be separated into two parts. One is from x_{old} to x_r and the other is from x_r to x_{new} . The charge flux only contributes to the points $j_1 - 1, j_1, j_1 + 1$ and $j_1 + 2$ in the first progress, while the second contributes to the points $j_2 - 1, j_2, j_2 + 1$ and $j_2 + 2$.
- (2) A particle moves across the cell mesh as shown in Fig. 5. At this time, x_{old} and x_{new} locate at the two side of the grid. The point x_r can be expressed as $x_r = (j_1 + 1) - 1.0$ or $x_r = j_2 - 1.0$. We can separate the progress into two parts. One is from x_{old} to x_r and the other is from x_r to x_{new} . The charge flux only contributes to the points of $j_1 - 1, j_1, j_1 + 1$ and $j_1 + 2$ in the first progress, while the second contributes to the points of $j_2 - 1, j_2, j_2 + 1$ and $j_2 + 2$.

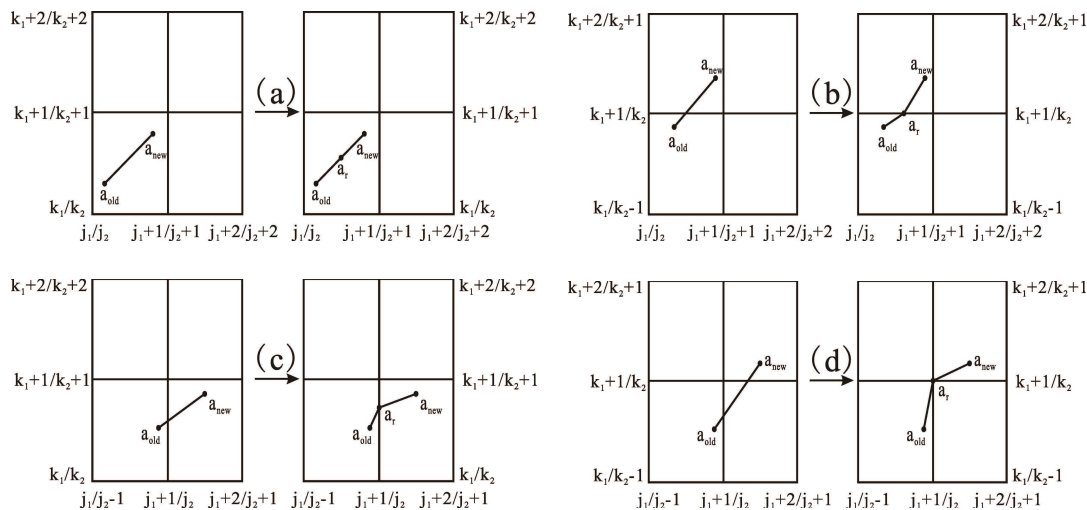


Figure 6: The particle trajectories of zigzag method in two dimensional under the condition of any odd-order interpolation.

After getting the move progress in x -direction, we can obtain the condition in y -direction by the same way. The particle trajectories of zigzag method in two dimensional for any odd-order interpolation can be described in Fig. 6.

Then the point (x_{ar}, y_{ar}) can be computed from the following equations

$$x_r = \min[\min(j_1, j_2), \max(\max(j_1 - 1.0, j_2 - 1.0), (x_{old} + x_{new}) / 2.0)], \tag{3.2a}$$

$$y_r = \min[\min(k_1, k_2), \max(\max(k_1 - 1.0, k_2 - 1.0), (y_{old} + y_{new}) / 2.0)]. \tag{3.2b}$$

Here F_{x1} , F_{y1} , F_{x2} and F_{y2} are the same as Eq. (2.7). Under the case of third-order spline interpolation, we define weighting functions w_{x1} , w_{x2} , w_{x3} , w_{x4} , w_{y1} , w_{y2} , w_{y3} and w_{y4} for the midpoint of (x_{old}, y_{old}) and (x_r, y_r) and w_{x5} , w_{x6} , w_{x7} , w_{x8} , w_{y5} , w_{y6} , w_{y7} and w_{y8} for the midpoint of (x_r, y_r) and (x_{new}, y_{new}) , which can be expressed as follows [9, 10]:

$$w_{x1} = (2.0 - xx_{11})^3 / 6.0, \quad w_{x2} = (4.0 - 6xx_{12}^2 + 3xx_{12}^3) / 6.0, \tag{3.3a}$$

$$w_{x3} = (4.0 - 6xx_{13}^2 + 3xx_{13}^3) / 6.0, \quad w_{x4} = (2.0 - xx_{14})^3 / 6.0, \tag{3.3b}$$

$$w_{y1} = (2.0 - yy_{11})^3 / 6.0, \quad w_{y2} = (4.0 - 6yy_{12}^2 + 3yy_{12}^3) / 6.0, \tag{3.3c}$$

$$w_{y3} = (4.0 - 6yy_{13}^2 + 3yy_{13}^3) / 6.0, \quad w_{y4} = (2.0 - yy_{14})^3 / 6.0, \tag{3.3d}$$

$$w_{x5} = (2.0 - xx_{21})^3 / 6.0, \quad w_{x6} = (4.0 - 6xx_{22}^2 + 3xx_{22}^3) / 6.0, \tag{3.3e}$$

$$w_{x7} = (4.0 - 6xx_{23}^2 + 3xx_{23}^3) / 6.0, \quad w_{x8} = (2.0 - xx_{24})^3 / 6.0, \tag{3.3f}$$

$$w_{y5} = (2.0 - yy_{21})^3 / 6.0, \quad w_{y6} = (4.0 - 6yy_{22}^2 + 3yy_{22}^3) / 6.0, \tag{3.3g}$$

$$w_{y7} = (4.0 - 6yy_{23}^2 + 3yy_{23}^3) / 6.0, \quad w_{y8} = (2.0 - yy_{24})^3 / 6.0. \tag{3.3h}$$

xx_{11} , xx_{12} , xx_{13} , xx_{14} , xx_{21} , xx_{22} , xx_{23} , xx_{24} , yy_{11} , yy_{12} , yy_{13} , yy_{14} , yy_{21} , yy_{22} , yy_{23} and yy_{24} are denoted as:

$$xx_{11} = 1.0 + xx_{12}, \quad xx_{12} = (x_{old} + x_r) / 2.0 - (j_1 - 1), \quad (0 \leq xx_{12} \leq 1), \tag{3.4a}$$

$$xx_{13} = 1.0 - xx_{12}, \quad xx_{14} = 2.0 - xx_{12}, \tag{3.4b}$$

$$xx_{21} = 1.0 + xx_{22}, \quad xx_{22} = (x_{old} + x_r) / 2.0 - (j_2 - 1), \quad (0 \leq xx_{22} \leq 1), \tag{3.4c}$$

$$xx_{23} = 1.0 - xx_{22}, \quad xx_{24} = 2.0 - xx_{22}, \tag{3.4d}$$

$$yy_{11} = 1.0 + yy_{12}, \quad yy_{12} = (y_{old} + y_r) / 2.0 - (k_1 - 1), \quad (0 \leq yy_{12} \leq 1), \tag{3.4e}$$

$$yy_{13} = 1.0 - yy_{12}, \quad yy_{14} = 2.0 - yy_{12}, \tag{3.4f}$$

$$yy_{21} = 1.0 + yy_{22}, \quad yy_{22} = (y_{new} + y_r) / 2.0 - (k_2 - 1), \quad (0 \leq yy_{22} \leq 1), \tag{3.4g}$$

$$yy_{23} = 1.0 - yy_{22}, \quad yy_{24} = 2.0 - yy_{22}. \tag{3.4h}$$

Then we can get the parts of the charge flux contributed to the grid points by the following expressions:

$$j_x(k_1 - 1, j_1 + 1.5) = \frac{1}{\Delta x \Delta y} F_{x1} w_{y1}, \quad j_x(k_2 - 1, j_2 + 1.5) = \frac{1}{\Delta x \Delta y} F_{x2} w_{y5}, \tag{3.5a}$$

$$j_x(k_1, j_1 + 1.5) = \frac{1}{\Delta x \Delta y} F_{x1} w_{y2}, \quad j_x(k_2, j_2 + 1.5) = \frac{1}{\Delta x \Delta y} F_{x2} w_{y6}, \tag{3.5b}$$

$$j_x(k_1+1, j_1+1.5) = \frac{1}{\Delta x \Delta y} F_{x1} w_{y3}, \quad j_x(k_2+1, j_2+1.5) = \frac{1}{\Delta x \Delta y} F_{x2} w_{y7}, \quad (3.5c)$$

$$j_x(k_1+2, j_1+1.5) = \frac{1}{\Delta x \Delta y} F_{x1} w_{y4}, \quad j_x(k_2+2, j_2+1.5) = \frac{1}{\Delta x \Delta y} F_{x2} w_{y8}, \quad (3.5d)$$

$$j_y(k_1+1.5, j_1-1) = \frac{1}{\Delta x \Delta y} F_{y1} w_{x1}, \quad j_y(k_2+1.5, j_2-1) = \frac{1}{\Delta x \Delta y} F_{y2} w_{x5}, \quad (3.5e)$$

$$j_y(k_1+1.5, j_1) = \frac{1}{\Delta x \Delta y} F_{y1} w_{x2}, \quad j_y(k_2+1.5, j_2) = \frac{1}{\Delta x \Delta y} F_{y2} w_{x6}, \quad (3.5f)$$

$$j_y(k_1+1.5, j_1+1) = \frac{1}{\Delta x \Delta y} F_{y1} w_{x3}, \quad j_y(k_2+1.5, j_2+1) = \frac{1}{\Delta x \Delta y} F_{y2} w_{x7}, \quad (3.5g)$$

$$j_y(k_1+1.5, j_1+2) = \frac{1}{\Delta x \Delta y} F_{y1} w_{x4}, \quad j_y(k_2+1.5, j_2+2) = \frac{1}{\Delta x \Delta y} F_{y2} w_{x8}. \quad (3.5h)$$

If add up the charge flux contributed by each particle, we can obtain the total current densities under the condition of third-order spline interpolation. In any odd-order spline interpolation, using the particle trajectories of zigzag method used in the third-order spline interpolation can receive expression similarly to Eq. (3.5) and then any odd-order spline interpolation can be expanded to by adopting correspondence weighting function (e.g., the method expanded to fifth-order spline interpolation can be seen in Appendix B).

4 Application test

In order to test the algorithms, some simulations are performed. The application tests are completed using fully relativistic electromagnetic PIC code, one-order code of which has been successfully used in the interactions of laser and plasma [11]. The order for the fields is the same as used for charge and charge flux in this paper. The tests are performed on a PC with Dual Pentium(R) 2.20GHz processor. The velocity of hole boring and numerical error under the condition of different order are tested, respectively.

The simulation condition of hole boring can be described as follows. The scales of simulation box used here are $X_L \times Y_L = 10\lambda_0 \times 10\lambda_0 = 2823\lambda_{De} \times 2823\lambda_{De}$ with the time step of 0.0125τ , the simulation duration of 70τ and the grid size of $\Delta X = \Delta Y = 0.03\lambda_0 = 8.47\lambda_{De}$, where τ and $\lambda_0 = 1.06\mu\text{m}$ are the period and wavelength of laser pulse, λ_{De} is the Debye length of electron, respectively. 100 electrons (ions) are used in one cell. with the total particles of 3.55×10^6 . We considered a Gaussian p-polarized laser pulse with duration of 10τ and focal spot of $4\lambda_0$. The laser introduces along the axis from the left. The peak intensity is $I_0 = 3.5 \times 10^{19} \text{W/cm}^2$. The initial temperatures of electrons and ions are 1.0keV and 0.1keV, respectively. A planar target with $4\lambda_0 * 8\lambda_0$ and density of $4n_c$ is considered. Here n_c is the critical density, which is related to the frequency of the laser as $n_c = m_e \omega_0^2 / 4\pi e^2$.

Hole boring [12] of laser pulse into plasma is a very important feature of laser-plasma interaction. We perform the velocities of hole boring in the cases of first-, second- and

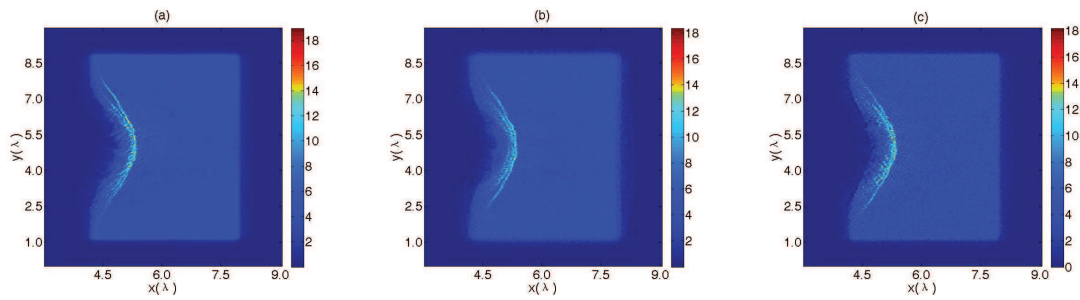


Figure 7: Ion density distribution at the time of $t = 70\tau$, (a) the simulation result of first-order case, (b) the result of second-order case, (c) the result of third-order case.

third-order algorithms, while fixing other parameters. From Fig. 7, we can see the ion density distributions under different-orders at the time of $t = 70\tau$. The depths of the holes are almost the same at the same time, which means using higher-order methods do not affect the physics.

In order to study the advantages of high-order algorithm, we calculate the total energy vs. time without external-field. A $40n_c$ with $2\mu\text{m}$ wide and $5\mu\text{m}$ thick plane target is considered. First-order method under the condition of $\Delta x = \Delta y = 15.3\lambda_{De}$, $\Delta x = \Delta y = 18.3\lambda_{De}$ and second-order with $\Delta x = \Delta y = 15.3\lambda_{De}$, $\Delta x = \Delta y = 24.5\lambda_{De}$ and $\Delta x = \Delta y = 45.9\lambda_{De}$ are simulated. The time steps are 0.025τ under the cases of $\Delta x = \Delta y = 15.3\lambda_{De}$ and $\Delta x = \Delta y = 18.3\lambda_{De}$, 0.04τ under the cases of $\Delta x = \Delta y = 24.5\lambda_{De}$ and 0.05τ under the cases of $\Delta x = \Delta y = 45.9\lambda_{De}$. 900 electrons (ions) are used in one cell. The temperatures of electrons and ions are 7.6keV. Fig. 8 shows the total energy vs. time. From Fig. 8, we can find second-order algorithm can suppress numerical grid heating and allow increasing grid size in PIC codes. Using second-order algorithm the grid sizes can be extended up to 46 Debye lengths without significant numerical heating over the time of 400fs. The second-

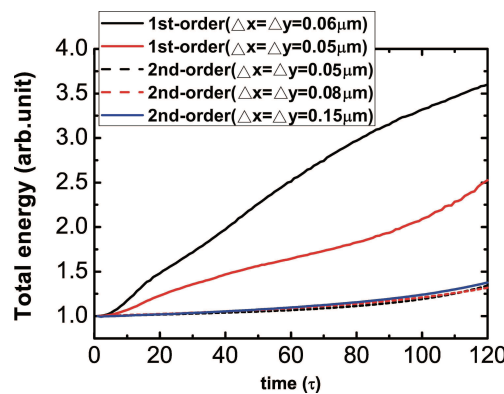


Figure 8: Total energy vs. time of $2\lambda_0$ wide and $5\lambda_0$ thick plane target, the grid scales are $\Delta x = \Delta y = 15.3\lambda_{De} = 0.05\mu\text{m}$, $\Delta x = \Delta y = 18.3\lambda_{De} = 0.06\mu\text{m}$ under the condition of first-order algorithm and $\Delta x = \Delta y = 15.3\lambda_{De} = 0.05\mu\text{m}$, $\Delta x = \Delta y = 24.5\lambda_{De} = 0.08\mu\text{m}$, $\Delta x = \Delta y = 45.9\lambda_{De} = 0.15\mu\text{m}$ under the condition of second-order algorithm.

order method is very useful in calculating the efficiency of laser to plasma. Generally speaking, numerical error mostly comes from the precision and the steps of computation. In the case of larger grid size, larger time step is adopted and smaller number of computational steps is received and then smaller numerical error can be obtained; meanwhile, the precision of computation reduces as the expanding of the grid size will enlarge the numerical error. On the other hand, reducing the grid size, the precision can be improved. Meanwhile, larger number of computational steps will be consumed, which will enhance the numerical error of repeated addition. In summary, under the condition of second-order case, the result that the total energy of $\Delta x = \Delta y = 15.3\lambda_{De}$ is larger than that of $\Delta x = \Delta y = 24.5\lambda_{De}$ after the time of 110τ shown in Fig. 8 is the outcomes of the above two reasons.

Fig. 9(a) shows the total energy vs. different particles per cell of $2\lambda_0$ wide and $5\lambda_0$ thick plane target at the time of 400fs, the grid scales are $\Delta x = \Delta y = 30.7\lambda_{De}$ for second-order and $\Delta x = \Delta y = 15.3\lambda_{De}$ for first-order, respectively. Under the condition of first-order, 900 electrons (ions) are used in one cell. In the case of second-order, 900, 1600, 2500, 3600 electrons (ions) are considered in one cell, respectively. From the figure, we find that the energy conservation is better under the condition of more number of particles per cell. When the field is added, the energy conservation is better than the case of without external-field. Fig. 9(b) shows the CPU time as a function of the number of particle under the conditions of first-order and second-order, from which we can find high-order algorithms have advantages in reducing the CPU time.

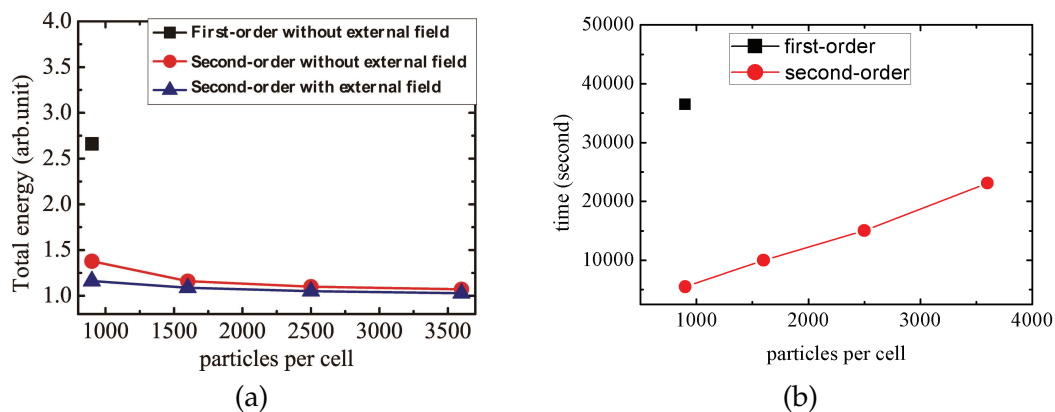


Figure 9: (a) Total energy vs. different particles per cell of $2\lambda_0$ wide and $5\lambda_0$ thick plane target at the time of 400fs, the grid scales are $\Delta x = \Delta y = 30.7\lambda_{De} = 0.1\mu\text{m}$ for second-order and $\Delta x = \Delta y = 15.3\lambda_{De} = 0.05\mu\text{m}$ for first-order; (b) CPU time as a function of particles per cell at the time of 400fs.

Under the condition of third-order method, a $40n_c$ with $20\mu\text{m}$ wide and $25\mu\text{m}$ thick plane target is considered. The temperatures of electrons and ions are both 7.6keV. 10000 electrons (ions) are used in one cell. The grid sizes are extended to $\Delta x = \Delta y = 0.96\mu\text{m} = 293.7\lambda_{De}$. Compared with the result of second-order method with $\Delta x = \Delta y = 0.3\mu\text{m} = 91.7\lambda_{De}$, as shown in Fig. 10, third-order algorithm can extend the grid sizes up to 294

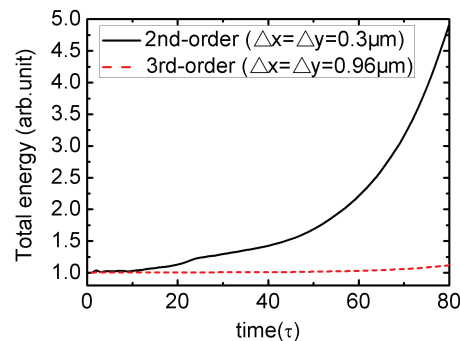


Figure 10: Total energy vs. time under the conditions of different-order method.

Debye lengths without significant numerical heating over the time of 260fs, which indicates that high-order method can enlarge the grid scales and reduce the CPU time.

5 Conclusions

In this paper, high-order algorithms for charge conservation in Particle-in-Cell (PIC) simulations are presented. The algorithms are valid for the case that a particle trajectory is a zigzag line.

The method of second-order algorithm which can be applied to any even-order algorithms is introduced. And the algorithm for third-order form-factor which can be expanded to any even-order algorithms is also presented.

The algorithms in 2D-PIC code are tested and proved the usability of the methods. The simulation results indicate that high-order algorithms can reduce the numerical noises, increase the sizes of space grids and reduce the CPU time.

Acknowledgments

We are very glad to discuss with Dr. K. G. Dong, Dr. W. Fan, Dr. D. X. Liu, Dr. L. Wei, Dr. H. P. Zang and Dr. Y. H. Yan. This work is supported by the National Natural Science Foundation of China (Grant Nos. 10905009, 11174259, 11175165 and 10975121), the Doctorate Foundation of the Ministry of Education of China (Grant No. 200806141034) and the Fundamental Research Funds for the Central Universities (Grant No. ZYGX2010J052).

Appendix

A The method expanded to fourth-order spline interpolation

The weighting functions $w_{x1}, w_{x2}, w_{x3}, w_{x4}, w_{x5}, w_{y1}, w_{y2}, w_{y3}, w_{y4}$ and w_{y5} for the mid-point of (x_{old}, y_{old}) and (x_r, y_r) and $w_{x6}, w_{x7}, w_{x8}, w_{x9}, w_{x10}, w_{y6}, w_{y7}, w_{y8}, w_{y9}$ and w_{y10} for

the midpoint of (x_r, y_r) and (x_{new}, y_{new}) can be expressed as follows:

$$\begin{aligned}
 w_{x1} &= \frac{1}{24} \left(\frac{5}{2} - xx_{11} \right)^4, \\
 w_{x2} &= \frac{1}{120} (51 + 75xx_{12} - 210xx_{12}^2 + 150xx_{12}^3 - 45xx_{12}^4 + 5xx_{12}^5), \\
 w_{x3} &= \frac{1}{192} (115 - 120xx_{13}^2 + 48xx_{13}^4), \\
 w_{x4} &= \frac{1}{120} (51 + 75xx_{14} - 210xx_{14}^2 + 150xx_{14}^3 - 45xx_{14}^4 + 5xx_{14}^5), \\
 w_{x5} &= \frac{1}{24} \left(\frac{5}{2} - xx_{15} \right)^4, \\
 w_{y1} &= \frac{1}{24} \left(\frac{5}{2} - yy_{11} \right)^4, \\
 w_{y2} &= \frac{1}{120} (51 + 75yy_{12} - 210yy_{12}^2 + 150yy_{12}^3 - 45yy_{12}^4 + 5yy_{12}^5), \\
 w_{y3} &= \frac{1}{192} (115 - 120yy_{13}^2 + 48yy_{13}^4), \\
 w_{y4} &= \frac{1}{120} (51 + 75yy_{14} - 210yy_{14}^2 + 150yy_{14}^3 - 45yy_{14}^4 + 5yy_{14}^5), \\
 w_{y5} &= \frac{1}{24} \left(\frac{5}{2} - yy_{15} \right)^4, \\
 w_{x6} &= \frac{1}{24} \left(\frac{5}{2} - xx_{21} \right)^4, \\
 w_{x7} &= \frac{1}{120} (51 + 75xx_{22} - 210xx_{22}^2 + 150xx_{22}^3 - 45xx_{22}^4 + 5xx_{22}^5), \\
 w_{x8} &= \frac{1}{192} (115 - 120xx_{23}^2 + 48xx_{23}^4), \\
 w_{x9} &= \frac{1}{120} (51 + 75xx_{24} - 210xx_{24}^2 + 150xx_{24}^3 - 45xx_{24}^4 + 5xx_{24}^5), \\
 w_{x10} &= \frac{1}{24} \left(\frac{5}{2} - xx_{25} \right)^4, \\
 w_{y6} &= \frac{1}{24} \left(\frac{5}{2} - yy_{21} \right)^4, \\
 w_{y7} &= \frac{1}{120} (51 + 75yy_{22} - 210yy_{22}^2 + 150yy_{22}^3 - 45yy_{22}^4 + 5yy_{22}^5), \\
 w_{y8} &= \frac{1}{192} (115 - 120yy_{23}^2 + 48yy_{23}^4), \\
 w_{y9} &= \frac{1}{120} (51 + 75yy_{24} - 210yy_{24}^2 + 150yy_{24}^3 - 45yy_{24}^4 + 5yy_{24}^5), \\
 w_{y10} &= \frac{1}{24} \left(\frac{5}{2} - yy_{25} \right)^4.
 \end{aligned}$$

Here $xx_{11}, xx_{12}, xx_{13}, xx_{14}, xx_{15}, xx_{21}, xx_{22}, xx_{23}, xx_{24}, xx_{25}, yy_{11}, yy_{12}, yy_{13}, yy_{14}, yy_{15}, yy_{21}, yy_{22}, yy_{23}, yy_{24}, yy_{25}$ are denoted as:

$$\begin{aligned}
 xx_{11} &= (x_{old} + x_r) / 2.0 - (j_1 - 3), & xx_{12} &= (x_{old} + x_r) / 2.0 - (j_1 - 2), \\
 xx_{13} &= |(x_{old} + x_r) / 2.0 - (j_1 - 1)|, & & (0 \leq xx_{13} \leq 0.5), \\
 xx_{14} &= j_1 - (x_{old} + x_r) / 2.0, & xx_{15} &= j_1 + 1.0 - (x_{old} + x_r) / 2.0, \\
 xx_{21} &= (x_{new} + x_r) / 2.0 - (j_2 - 3), & xx_{22} &= (x_{new} + x_r) / 2.0 - (j_2 - 2), \\
 xx_{23} &= |(x_{new} + x_r) / 2.0 - (j_2 - 1)|, & & (0 \leq xx_{23} \leq 0.5), \\
 xx_{24} &= j_2 - (x_{new} + x_r) / 2.0, & xx_{25} &= j_2 + 1.0 - (x_{new} + x_r) / 2.0, \\
 yy_{11} &= (y_{old} + y_r) / 2.0 - (k_1 - 3), & yy_{12} &= (y_{old} + y_r) / 2.0 - (k_1 - 2), \\
 yy_{13} &= |(y_{old} + y_r) / 2.0 - (k_1 - 1)|, & & (0 \leq yy_{13} \leq 0.5), \\
 yy_{14} &= k_1 - (y_{old} + y_r) / 2.0, & yy_{15} &= k_1 + 1.0 - (y_{old} + y_r) / 2.0, \\
 yy_{21} &= (y_{new} + y_r) / 2.0 - (k_2 - 3), & yy_{22} &= (y_{new} + y_r) / 2.0 - (k_2 - 2), \\
 yy_{23} &= |(y_{new} + y_r) / 2.0 - (k_2 - 1)|, & & (0 \leq yy_{23} \leq 0.5), \\
 yy_{24} &= k_2 - (y_{new} + y_r) / 2.0, & yy_{25} &= k_2 + 1.0 - (y_{new} + y_r) / 2.0.
 \end{aligned}$$

The expressions of $x_{old}, x_r, x_{new}, y_{old}, y_r, y_{new}, j_1, j_2, k_1, k_2$ can be seen in (2.4), (2.5) and (2.7). We define

$$\begin{aligned}
 j_{11} &= [x_{old} + 1.0], & j_{22} &= [x_{new} + 1.0], \\
 k_{11} &= [y_{old} + 1.0], & k_{22} &= [y_{new} + 1.0].
 \end{aligned}$$

Then we can get the parts of the charge flux contributed to the grid points by the following expressions:

$$\begin{aligned}
 j_x(k_1 - 2, j_{11} + 0.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y1}, & j_x(k_2 - 2, j_{22} + 0.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y6}, \\
 j_x(k_1 - 1, j_{11} + 0.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y2}, & j_x(k_2 - 1, j_{22} + 0.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y7}, \\
 j_x(k_1, j_{11} + 0.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y3}, & j_x(k_2, j_{22} + 0.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y8}, \\
 j_x(k_1 + 1, j_{11} + 0.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y4}, & j_x(k_2 + 1, j_{22} + 0.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y9}, \\
 j_x(k_1 + 2, j_{11} + 0.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y5}, & j_x(k_2 + 2, j_{22} + 0.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y10}, \\
 j_y(k_{11} + 0.5, j_1 - 2) &= \frac{1}{\Delta x \Delta y} F_{y1} w_{x1}, & j_y(k_{22} + 0.5, j_2 - 2) &= \frac{1}{\Delta x \Delta y} F_{y2} w_{x6}, \\
 j_y(k_{11} + 0.5, j_1 - 1) &= \frac{1}{\Delta x \Delta y} F_{y1} w_{x2}, & j_y(k_{22} + 0.5, j_2 - 1) &= \frac{1}{\Delta x \Delta y} F_{y2} w_{x7},
 \end{aligned}$$

$$\begin{aligned}
 j_y(k_{11}+0.5, j_1) &= \frac{1}{\Delta x \Delta y} F_{y1} w_{x3}, & j_y(k_{22}+0.5, j_2) &= \frac{1}{\Delta x \Delta y} F_{y2} w_{x8}, \\
 j_y(k_{11}+0.5, j_1+1) &= \frac{1}{\Delta x \Delta y} F_{y1} w_{x4}, & j_y(k_{22}+0.5, j_2+1) &= \frac{1}{\Delta x \Delta y} F_{y2} w_{x9}, \\
 j_y(k_{11}+0.5, j_1+2) &= \frac{1}{\Delta x \Delta y} F_{y1} w_{x5}, & j_y(k_{22}+0.5, j_2+2) &= \frac{1}{\Delta x \Delta y} F_{y2} w_{x10}.
 \end{aligned}$$

B The method expanded to fifth-order spline interpolation

The weighting functions for the midpoint of (x_{old}, y_{old}) and (x_r, y_r) and for the midpoint of (x_r, y_r) and (x_{new}, y_{new}) can be expressed as follows:

$$\begin{aligned}
 w_{x1} &= (3.0 - xx_{11})^5 / 120, \\
 w_{x2} &= (51 + 75xx_{12} - 210xx_{12}^2 + 150xx_{12}^3 - 45xx_{12}^4 + 5xx_{12}^5) / 120, \\
 w_{x3} &= (33 - 30xx_{13}^2 + 15xx_{13}^4 - 5xx_{13}^5) / 60, \\
 w_{x4} &= (33 - 30xx_{14}^2 + 15xx_{14}^4 - 5xx_{14}^5) / 60, \\
 w_{x5} &= (51 + 75xx_{15} - 210xx_{15}^2 + 150xx_{15}^3 - 45xx_{15}^4 + 5xx_{15}^5) / 120, \\
 w_{x6} &= (3.0 - xx_{16})^5 / 120, \\
 w_{y1} &= (3.0 - yy_{11})^5 / 120, \\
 w_{y2} &= (51 + 75yy_{12} - 210yy_{12}^2 + 150yy_{12}^3 - 45yy_{12}^4 + 5yy_{12}^5) / 120, \\
 w_{y3} &= (33 - 30yy_{13}^2 + 15yy_{13}^4 - 5yy_{13}^5) / 60, \\
 w_{y4} &= (33 - 30yy_{14}^2 + 15yy_{14}^4 - 5yy_{14}^5) / 60, \\
 w_{y5} &= (51 + 75yy_{15} - 210yy_{15}^2 + 150yy_{15}^3 - 45yy_{15}^4 + 5yy_{15}^5) / 120, \\
 w_{y6} &= (3.0 - yy_{16})^5 / 120, \\
 w_{x7} &= (3.0 - xx_{21})^5 / 120, \\
 w_{x8} &= (51 + 75xx_{22} - 210xx_{22}^2 + 150xx_{22}^3 - 45xx_{22}^4 + 5xx_{22}^5) / 120, \\
 w_{x9} &= (33 - 30xx_{23}^2 + 15xx_{23}^4 - 5xx_{23}^5) / 60, \\
 w_{x10} &= (33 - 30xx_{24}^2 + 15xx_{24}^4 - 5xx_{24}^5) / 60, \\
 w_{x11} &= (51 + 75xx_{25} - 210xx_{25}^2 + 150xx_{25}^3 - 45xx_{25}^4 + 5xx_{25}^5) / 120, \\
 w_{x12} &= (3.0 - xx_{26})^5 / 120, \\
 w_{y7} &= (3.0 - yy_{21})^5 / 120, \\
 w_{y8} &= (51 + 75yy_{22} - 210yy_{22}^2 + 150yy_{22}^3 - 45yy_{22}^4 + 5yy_{22}^5) / 120, \\
 w_{y9} &= (33 - 30yy_{23}^2 + 15yy_{23}^4 - 5yy_{23}^5) / 60,
 \end{aligned}$$

$$\begin{aligned}
 w_{y10} &= (33 - 30yy_{24}^2 + 15yy_{24}^4 - 5yy_{24}^5) / 60, \\
 w_{y11} &= (51 + 75yy_{25}^2 - 210yy_{25}^3 + 150yy_{25}^4 - 45yy_{25}^5) / 120, \\
 w_{y12} &= (3.0 - yy_{26})^5 / 120,
 \end{aligned}$$

where $xx_{11}, xx_{12}, xx_{13}, xx_{14}, xx_{15}, xx_{16}, xx_{21}, xx_{22}, xx_{23}, xx_{24}, xx_{25}, xx_{26}, yy_{11}, yy_{12}, yy_{13}, yy_{14}, yy_{15}, yy_{16}, yy_{21}, yy_{22}, yy_{23}, yy_{24}, yy_{25}$ and yy_{26} are denoted as:

$$\begin{aligned}
 xx_{11} &= 2.0 + xx_{13}, & xx_{12} &= 1.0 + xx_{13}, & xx_{13} &= (x_{old} + x_r) / 2.0 - (j_1 - 1), & (0 \leq xx_{13} \leq 1), \\
 xx_{14} &= 1.0 - xx_{13}, & xx_{15} &= 2.0 - xx_{13}, & xx_{16} &= 3.0 - xx_{13}, \\
 xx_{21} &= 2.0 + xx_{23}, & xx_{22} &= 1.0 + xx_{23}, & xx_{23} &= (x_{old} + x_r) / 2.0 - (j_2 - 1), & (0 \leq xx_{23} \leq 1), \\
 xx_{24} &= 1.0 - xx_{23}, & xx_{25} &= 2.0 - xx_{23}, & xx_{26} &= 3.0 - xx_{23}, \\
 yy_{11} &= 2.0 + yy_{13}, & yy_{12} &= 1.0 + yy_{13}, & yy_{13} &= (y_{old} + y_r) / 2.0 - (k_1 - 1), & (0 \leq yy_{13} \leq 1), \\
 yy_{14} &= 1.0 - yy_{13}, & yy_{15} &= 2.0 - yy_{13}, & yy_{16} &= 3.0 - yy_{13}, \\
 yy_{21} &= 2.0 + yy_{23}, & yy_{22} &= 1.0 + yy_{23}, & yy_{23} &= (y_{old} + y_r) / 2.0 - (k_2 - 1), & (0 \leq yy_{23} \leq 1), \\
 yy_{24} &= 1.0 - yy_{23}, & yy_{25} &= 2.0 - yy_{23}, & yy_{26} &= 3.0 - yy_{23}.
 \end{aligned}$$

The expressions of $x_{old}, x_r, x_{new}, y_{old}, y_r, y_{new}, j_1, j_2, k_1, k_2$ can be seen in (2.4), (3.1) and (3.2). Then we can get the parts of the charge flux contributed to the grid points by the following expressions:

$$\begin{aligned}
 j_x(k_1 - 2, j_1 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y1}, & j_x(k_2 - 2, j_2 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y7}, \\
 j_x(k_1 - 1, j_1 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y2}, & j_x(k_2 - 1, j_2 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y8}, \\
 j_x(k_1, j_1 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y3}, & j_x(k_2, j_2 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y9}, \\
 j_x(k_1 + 1, j_1 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y4}, & j_x(k_2 + 1, j_2 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y10}, \\
 j_x(k_1 + 2, j_1 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y5}, & j_x(k_2 + 2, j_2 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y11}, \\
 j_x(k_1 + 3, j_1 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x1} w_{y6}, & j_x(k_2 + 3, j_2 + 1.5) &= \frac{1}{\Delta x \Delta y} F_{x2} w_{y12}, \\
 j_y(k_1 + 1.5, j_1 - 2) &= \frac{1}{\Delta x \Delta y} F_{y1} w_{x1}, & j_y(k_2 + 1.5, j_2 - 2) &= \frac{1}{\Delta x \Delta y} F_{y2} w_{x7}, \\
 j_y(k_1 + 1.5, j_1 - 1) &= \frac{1}{\Delta x \Delta y} F_{y1} w_{x2}, & j_y(k_2 + 1.5, j_2 - 1) &= \frac{1}{\Delta x \Delta y} F_{y2} w_{x8}, \\
 j_y(k_1 + 1.5, j_1) &= \frac{1}{\Delta x \Delta y} F_{y1} w_{x3}, & j_y(k_2 + 1.5, j_2) &= \frac{1}{\Delta x \Delta y} F_{y2} w_{x9},
 \end{aligned}$$

$$\begin{aligned}
 j_y(k_1 + 1.5, j_1 + 1) &= \frac{1}{\Delta x \Delta y} F_{y1} * w_{x4}, & j_y(k_2 + 1.5, j_2 + 1) &= \frac{1}{\Delta x \Delta y} F_{y2} * w_{x10}, \\
 j_y(k_1 + 1.5, j_1 + 2) &= \frac{1}{\Delta x \Delta y} F_{y1} * w_{x5}, & j_y(k_2 + 1.5, j_2 + 2) &= \frac{1}{\Delta x \Delta y} F_{y2} * w_{x11}, \\
 j_y(k_1 + 1.5, j_1 + 3) &= \frac{1}{\Delta x \Delta y} F_{y1} * w_{x6}, & j_y(k_2 + 1.5, j_2 + 3) &= \frac{1}{\Delta x \Delta y} F_{y2} * w_{x12}.
 \end{aligned}$$

References

- [1] T. Zh. Esirkepov, Exact charge conservation for particle-in-cell simulation with an arbitrary form-factor, *Comput. Phys. Commun.*, 135 (2001), 144–153.
- [2] J. W. Eastwood, The virtual particle electromagnetic particle-mesh method, *Comput. Phys. Commun.*, 64 (1991), 252.
- [3] J. W. Eastwood, W. Arter, N. J. Brealey and R. W. Hockney, Body-fitted electromagnetic PIC software for use on parallel computers, *Comput. Phys. Commun.*, 87 (1995), 155.
- [4] R. L. Morse and C. W. Nielson, Numerical simulation of the Weibel instability in one and two dimensions, *Phys. Fluids*, 14 (1971), 830.
- [5] V. A. Vshivkov, M. A. Kraeva and V. E. Malyshkin, Parallel implementation of the particle-in-cell method, *Program. Comput. Software*, 23(2) (1997), 87–97.
- [6] J. Villasenor and O. Buneman, Rigorous charge conservation for local electromagnetic field solvers, *Comput. Phys. Commun.*, 69 (1992), 306.
- [7] T. Umeda, Y. Omura, T. Tominaga and H. Matsumoto, A new charge conservation method in electromagnetic particle-in-cell simulations, *Comput. Phys. Commun.*, 156 (2003), 73–85.
- [8] T. Umeda, Y. Omura, T. Tominaga and H. Matsumoto, Charge conservation methods for computing current densities in electromagnetic particle-in-cell simulations, *Proceedings of ISSS-7*, 26-31 March, 2005.
- [9] H. Abe, N. Sakairi, R. Itatani and H. Okuda, High-order spline interpolations in the particle simulation, *J. Comput. Phys.*, 63 (1986), 247–267.
- [10] C. K. Birdsall and A. B. Langdon, *Plasma Physics Via Computer Simulation*, Adam-Hilger, 1991.
- [11] W. M. Zhou, *Research on Laser Plasma Acceleration by Particle-in-Cell Simulation*, Osaka University, 2008.
- [12] S. C. Wilks, W. L. Kruer, M. Tabak and A. B. Langdon, Absorption of ultra-intense laser pulses, *Phys. Rev. Lett.*, 69 (1992), 1383–1386.