Nuclear Instruments and Methods in Physics Research A267 (1988) 25–34
North-Holland, Amsterdam

# SIMULATION OF INTENSE PARTICLE BEAMS WITH REGULARLY DISTRIBUTED GAUSSIAN SUBBEAMS

M. BERZ and H. WOLLNIK

*II. Physikalisches Institut, Universität Giessen, D-6300 Giessen, FRG*

Received 23 September 1987

Techniques for the simulation of intense particle beams are investigated with respect to the required number of simulation particles. It is shown that for nonchaotic systems it is advantageous if the particles initially are not distributed in a statistical manner but rather arranged in a regular pattern in phase space. This reduces the number of required simulation particles drastically. In the case of such an initially regular arrangement of particles the algorithm which assigns the charges of the particles to the computation mesh becomes of prime importance. The performances of different commonly used algorithms are investigated. The Gaussian assignment algorithm proved far superior to other more commonly used techniques, allowing simulations even at the theoretical limit of 1 particle per cell. Examples for very accurate simulations of beam dynamics with very few particles using an initially regular mesh of particles and Gaussian assignment are given.

## 1. Introduction

In order to simulate the dynamics of charged particle beams in accelerators, beam guidance systems and particle spectrometers, phase space is often represented by an ensemble of simulation particles which are traced individually through the optical system under consideration [1–3]. The effects of focusing or accelerating fields are taken into account in a step-by-step manner using standard integration techniques such as the Runge–Kutta methods.

In the case of intense charged particle beams not only forces due to external electromagnetic fields act on an individual particle but also the internal space-charge forces. These forces depend on the positions, velocities and charges of all other particles. Thus the computation of space-charge effects is usually rather involved and time consuming. A direct calculation of the internal force on a particle as a sum over the forces created by all other particles requires a computation effort proportional to the square of the number of particles in a beam. Therefore, one often uses a POISSON solver to determine the electromagnetic potentials and fields [1–3] at the points of a regular computation mesh at which charge and current densities are known. The forces on individual particles are then calculated by interpolation.

## 2. The initial distribution of particles in phase space

One way to achieve a representative arrangement of individual particles of a beam in phase space is to distribute them in a random fashion at the beginning. Unfortunately, this requires a large number of particles to achieve on required grid points a charge density that is sufficiently accurate.

To appreciate this, note that most assignment algorithms distribute the charge of a particle among the mesh points closest to the particle. Let $N_m$ be the average number of mesh points to which a significant amount of charge is transferred by the algorithm under consideration. Then the average number $N_c$ of particles contributing to the charge of one of the $N_p$ mesh points is

$$N_c = \frac{N_0 N_m}{N_p}. \tag{1}$$

Here $N_0$ is the total number of simulation particles in the beam. Because the number of contributing particles $N_c$ varies statistically from one mesh point to another, the standard deviation of the charge of an arbitrary mesh point is proportional to $\sqrt{N_c}$.

As an example, consider now as assignment algorithm with $N_m = 4$, then a relative error of 5% requires an average of $N_c = 400$ particles per mesh point totalling $N_0 = 90\,000$ particles in case of a modest 30 by 30 two-dimensional mesh with $N_p = 900$. Choosing algorithms with enlarged values of $N_m$ of course makes $N_c$ larger and thus reduces these statistical errors. However, this entails an undesirable loss of the fine structure of the calculated charge distribution.

Alternatively to statistically selecting the initial phase-space coordinates of the particles one can distribute them in a regular way. This can, for instance, be on a grid where, if desired, each particle can carry a different charge so that it is possible to model an arbitrary charge distribution in a beam. In this case the number of particles in the proximity of a point of the computation mesh does not fluctuate statistically so that one can expect a better performance for small numbers of particles.

An initial regularity in the distribution of particles will more or less persist during the simulation as long as the forces acting on the particles during the whole simulation are continuous and vary only smoothly. In fact, linear forces just enlarge or diminish the whole particle mesh, whereas nonlinear forces cause distortions. In the case of chaotic motion (characterized by the fact that initially neighbouring phase-space points can eventually move further and further apart) this is of course only true for a limited time. After this time the new density distribution in phase space must be determined and then represented by a new regularly arranged ensemble of particles.

## 3. Requirements for assignment algorithms

Algorithms assigning the charge of simulation particles to a computation mesh should have the following three basic properties:

(1) The charge assigned to a computation mesh point A from a simulation particle P should be proportional to the charge of P.
(2) The smaller the distance between A and P becomes, the more charge should be assigned to A.
(3) In case of an infinitely large ensemble of simulation particles of equal charges arranged on a regular mesh the charge assigned to a computation mesh point A should be independent of its position.

The requirements (1) and (2) are obvious. Requirement (3) is necessary since the infinite mesh of simulation particles is the particle in cell (PIC) representation of a homogeneous charge distribution. In this case the charge assigned to a computation mesh point A should be independent of the position of A.

Given a certain algorithm, it turns out that the finer the mesh of particles is chosen, the better property (3) is fulfilled. However, since a finer mesh requires more simulation particles, the objective is to determine an algorithm that satisfies property (3) with as few particles as possible. In sections 4–7 different assignment algorithms will be investigated with respect to this goal.

The above three properties guarantee a smooth charge assignment without unphysical charge fluctuations within one cell also in the case of a distorted particle mesh. To appreciate this, we first consider a distorted mesh of particles of equal charges like the one on the right hand side of fig. 1. Assume here a computation mesh point A located in a certain cell of the distorted particle mesh and construct a regular particle mesh with cells the sizes of all of which are equal to the one A is located in (see left hand side of fig. 1). In both cases the charge assigned to point A from the four closest particles is identical. However, in the distorted case the nine second closest particles contribute slightly more if they are closer to point A as compared to the case of the regular mesh according to the above requirement (2).

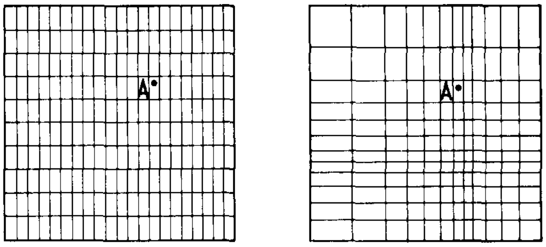

Fig. 1. In the case of a regular mesh of simulation particles of equal charges, the charge assigned to a computation point A should be independent of its position. In the case of a distorted mesh of simulation particles, this implies that the charge assigned to a point A still varies smoothly as this point moves across the particle mesh.

This implies that the charge assigned to A is no longer constant as A moves within one cell of the distorted mesh. In case of the example of fig. 1, the charge assigned to the point A increases if A moves to the lower part of the cell. The most important fact, however, is that the charge assigned to A, though no longer constant, still varies smoothly with the position of A not showing any unphysical charge fluctuations. A similar argument using the above requirement (1) shows that no unphysical charge fluctuations occur even if different particles carry different charges.

In the case of initially statistically distributed particles the smoothness of the final charge distribution across the particle beam depends mainly on the number of particles per cell of the computation grid $N_0/N_p$ and only slightly on the used assignment algorithm. However, in case the particles are originally arranged on a regular grid the assignment algorithm becomes of prime importance. Whereas the above demanded properties (1) and (2) are fulfilled by all frequently used algorithms, property (3) is usually violated for small numbers of particles. In the following sections we therefore will investigate how well different assignment algorithms fulfill property (3) for limited numbers of simulation particles.

## 4. The nearest grid point (NGP) algorithm

Easy to implement and hence frequently used is the nearest grid point algorithm (NGP). In this case the

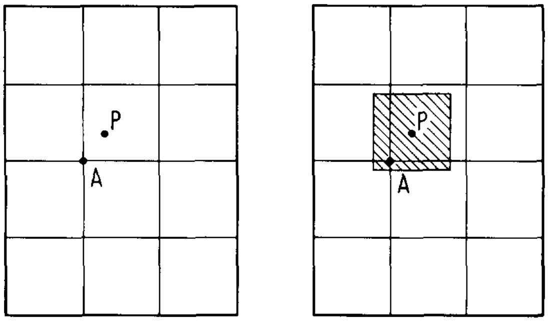

Fig. 2. In the case of the nearest-grid-point (NGP) algorithm, the total charge of a simulation particle P is transferred to the nearest point A of the computation grid. An equivalent interpretation of this method is to assign to each particle a constant, square-shaped charge density distribution extending $d_c/2$ both in $x$- and $y$-direction.

charge $q$ of a particle is transferred to the nearest grid point of the computation mesh of cell size $d_c$. This implies $N_m = 1$. This procedure can also be viewed as every particle with a charge $q$ causing a charge density $\rho_{NGP}$ which vanishes everywhere except in a square area extending to distances $d_c/2$ both in $x$- and $y$-directions around the position of the particle under consideration (see fig. 2). Inside this square the charge density is

$$\rho_{NGP}(x, y) = q/d_c^2. \tag{2}$$

The amount of charge assigned to a computation point A is then proportional to the sum of all charge densities at this point.

Consider now charged particles arranged on a regular mesh of cell size $d_p$. For $d_c = d_p, 2d_p, 3d_p, \cdots$ the requirement that the charge assigned to a computation mesh point A be independent of its position is fulfilled perfectly for the NGP algorithm. But when $d_c$ is no integer multiple of $d_p$, this property fails. Fig. 3 shows the charge $\rho_{NGP}$ assigned to a computation mesh point moving along the $x$-axis of the particle mesh for $d_c = 2.5d_p$ and $d_c = 4.5d_p$. A plot of the relative charge fluctuation $D_{NGP} = (\rho_{NGP}^{(max)} - \rho_{NGP}^{(min)})/\rho_{NGP}^{(max)}$, the difference between the largest and the smallest charge value assigned to a computation point as a function of $r = d_c/d_p$, is shown in fig. 10a. Having chosen a certain value of $D_{NGP}$ as the allowable relative difference, the required $r^2$, i.e. the number of particles per cell of the computation mesh, can be read from this plot. In order for $D_{NGP}$ to stay below 5% it is necessary to choose $r > 10$ which in the two-dimensional case corresponds to a minimum of $r^2 = 100$ particles per cell.

For $d_p \ll d_c$, the fluctuation $D_{NGP}$ becomes smaller and smaller, and this is the reason why the NGP algorithm has actually been used successfully in case of many particles per computation cell, i.e. for $N_c \gg 1$. Another way to achieve a large ratio $r = d_c/d_p$ is to choose $d_c$ in eq. (2) not to be the computation mesh cell size but an arbitrary larger value. This would increase $N_m$ in eq. (1) and thus yield smoother results without an increase in the number of particles. However, as mentioned under eq. (1), this would cause the charge of individual particles to be smeared out over a number of computation cells and thus also entail an undesirable loss of fine structure in the charge distribution of a beam.

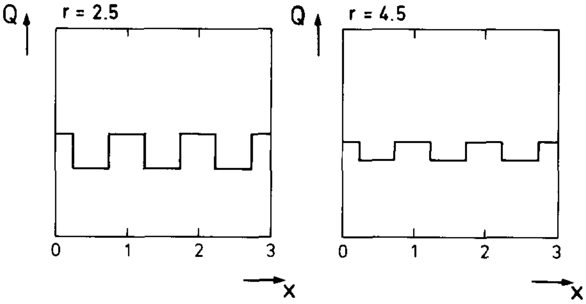

Fig. 3. The charge assigned by the NGP algorithm to a computation mesh point moving along one of the axes of the particle mesh as a function of its position. Shown are results for the cases $r = d_c/d_p = 2.5$ (left) and $r = d_c/d_p = 4.5$ (right). A plot of the corresponding relative differences $D_{NGP} = [\rho_{NGP}^{(max)} - \rho_{NGP}^{(min)}]/\rho_{NGP}^{(max)}$ between maximum and minimum charge values assigned to arbitrary mesh points as function of $r = d_c/d_p$ is shown in fig. 10a.

## 5. The area-weighted method

A refinement of the NGP algorithm is the area-weighted method (AW) [1] where the charge $q$ of a particle is distributed among the four closest computation mesh points in the two-dimensional case. The amount assigned to each mesh point A is proportional to the area of the rectangle R opposite to A as shown in fig. 4. This implies that once a particle lies right on top of a computation mesh point, all its charge is assigned to this point. With increasing distance from this point, an increasing amount of charge will be assigned to the other three points of the cell into which the particle fell. Analogously to eq. (2) this algorithm can be interpreted as attributing to every particle of charge $q$ a charge density $\rho_{AW}$ which vanishes everywhere except for $x < d_c$ and $y < d_c$ where it is

$$\rho_{AW}(x, y) = \frac{q}{4}\left(1 - \frac{x}{d_c}\right)\left(1 - \frac{y}{d_c}\right). \tag{3}$$

Here $x$ and $y$ are the distances between the particle and the mesh point A. Similarly to section 4 we consider now the charged particles to be arranged on an infinite, regular mesh with grid constant $d_p$. Due to the periodicity of the particle mesh only one cell of this

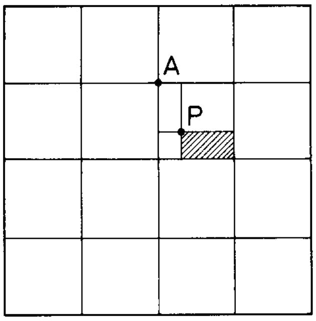

Fig. 4. In case of the area-weighted method the charge $q$ of a particle is distributed among the four closest computation mesh points, such that the amount transferred to point A is proportional to the area of the rectangle opposite to A.

mesh is representative for the whole grid. For such a cell contour lines of the total charge assigned to points at positions $x$, $y$ are shown in the three plots of fig. 5, where $r$ equals $d_c/d_p = 1.25$, 1.5 and 1.75. The contour lines are chosen in 1% steps of the maximum assigned charge. Note that the special form of the charge density distribution in eq. (3) implies that as for the NGP algorithm the assignment is perfectly smooth for $d_p = d_c, 2d_c, 3d_c, \cdots$. A plot of the relative fluctuation $D_{AW}$ as a function of $r = d_c/d_c$ is shown in fig. 10a.

From figs. 3, 5 and 10a it is evident that the nonuniformities of the AW algorithm are smaller than those of the NGP algorithm, though they are still undesirably large. In order to obtain charge fluctuations of 5% or less, one reads from fig. 10a that $r = d_c/d_p$ must be at least 4 and for fluctuations of 1% or less at least 10. This corresponds to an average of about $r^2 = 16$ or $r^2 = 100$ particles per cell, respectively. However, these 16 particles per cell for $D_{AW} < 0.05$ are already considerably less than the 100 in the case of the NGP algorithm of section 4 or the 400 in the case of a statistical distribution of phase-space points with $N_m = 1$.

The area-weighted method of eq. (3) is the two-dimensional generalization $\rho_{AW}(x, y) = \rho_{AW}(x)\rho_{AW}(y)$ of a one-dimensional algorithm with a charge distribution that vanishes everywhere except for $u < d_c$ where it is

$$q_{AW}^{(1)}(u) = \frac{q}{2}\left(1 - \frac{u}{d_c}\right). \tag{4}$$

A different generalization of this one-dimensional algorithm is a charge density $\rho_{RAW}$ which vanishes everywhere except for $(x^2 - y^2) < d_c^2$ where it is

$$\rho_{RAW}(x, y) = \frac{q}{2}\left(1 - \frac{\sqrt{x^2 + y^2}}{d_c}\right). \tag{5}$$

As is illustrated in fig. 6 this radially symmetric (RAW) charge distribution performs slightly better than the area-weighted (AW) method of eq. (3). To obtain charge fluctuations $D_{RAW}$ of 5% or less one reads from fig. 10a that $r = d_c/d_p$ must at least be 3 which corresponds to an average of about $r^2 = 9$ particles per cell.

## 6. The TSC algorithm

The TSC algorithm [4] is a generalization of the AW method having a differentiable and hence smoother charge distribution. For the TSC algorithm the charge density $\rho_{TSC}$ vanishes everywhere except for $|u| < 3/2d_c$ where it is

$$\rho_{TSC}^{(1)}(u) = q\begin{cases} \dfrac{3}{4} - \left(\dfrac{u}{d_c}\right)^2 & \text{for } |u| < \dfrac{d_c}{2}, \\[2ex] \dfrac{1}{2}\left(\dfrac{3}{2} - \dfrac{|u|}{d_c}\right)^2 & \text{for } \dfrac{d_c}{2} \le |u| < \dfrac{3d_c}{2}. \end{cases} \tag{6}$$

This one-dimensional algorithm can be generalized to

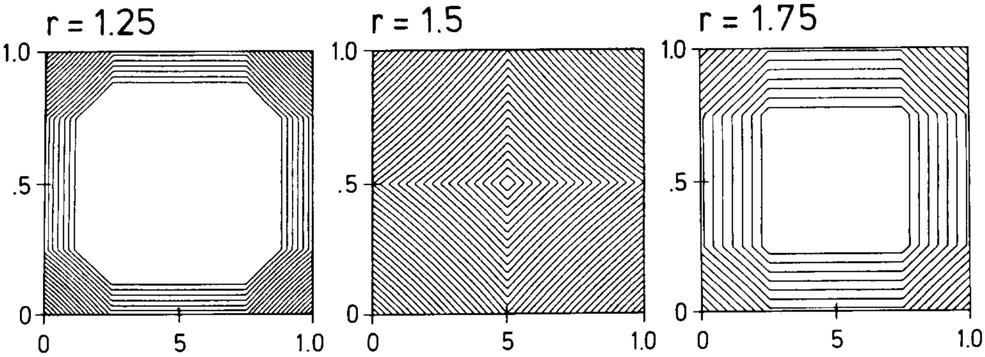

Fig. 5. 1% contour lines are shown of the charge assigned by the AW method to a computation mesh point A($x$, $y$) as a function of its position in a cell of the particle mesh for the cases $r = d_c/d_p = 1.25$, 1.5 and 1.75. A plot of the relative differences $D_{AW} = [\rho_{AW}^{(max)} - \rho_{AW}^{(min)}]/\rho_{AW}^{(max)}$ between maximum and minimum charge values assigned to an arbitrary computation mesh point as a function of $r$ is shown in fig. 10a.

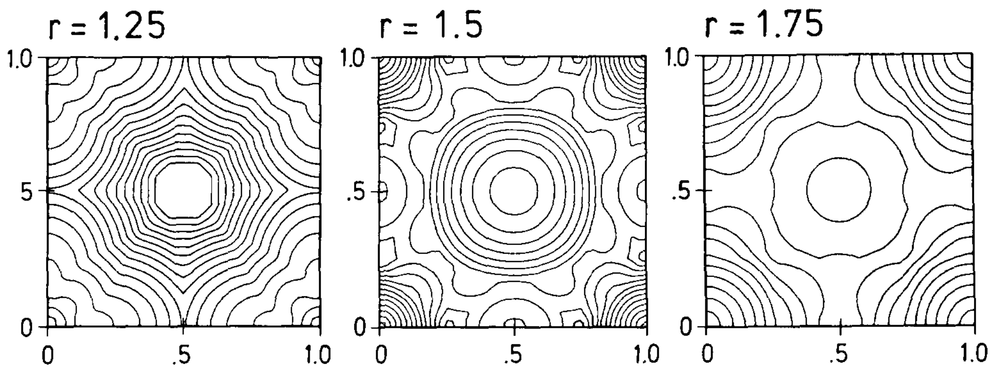

Fig. 6. 1% contour lines are shown of the charge assigned by the RAW method to a computation mesh point A($x$, $y$) as function of its position in a cell of the particle mesh for the cases $r = d_c/d_p = 1.25$, 1.5 and 1.75. A plot of the relative difference $D_{RAW} = [\rho^{(max)}_{RAW} - \rho^{(min)}_{RAW}]/\rho^{(max)}_{RAW}$ between the maximum and minimum charge values assigned to an arbitrary computation mesh point as a function of $r$ is shown in fig. 10a.

two dimensions similarly as the area-weighted method by:

$$\rho_{TSC}(x, y) = \rho^{(1)}_{TSC}(x)\rho^{(1)}_{TSC}(y). \tag{7}$$

As in section 5 we also here tested a new radially symmetric version

$$\rho_{RTSC}(x, y) = q^{(1)}_{TSC}\left(\sqrt{x^2 + y^2}\right). \tag{8}$$

The performance of these two algorithms is shown in figs. 7 and 8. They both display 1% contour lines of the charge assigned to a computation mesh point A($x$, $y$) as function of its position in the particle mesh. In figs. 7a, 7b and 7c, the charge assigned to a computation mesh point is shown for $r = d_c/d_p = 1.25$, 1.5 and 1.75 in the case of the TSC algorithm. Figs. 8a, 8b and 8c show the same assigned charge in the case of the RTSC algorithm. Plots of the relative fluctuations $D_{TSC}$ and $D_{RTSC}$ versus $r = d_c/d_p$ are shown in figs. 10a and 10b. Note that both of these algorithms perform significantly better than those discussed previously. Again the radially symmetric version is better than the normal TSC algorithm. As can be read from figs. 10a and 10b, both the TSC and RTSC assignment algorithms already reach the level of $D = 1\%$ for about $r = d_c/d_p = 3$ which corresponds to about $r^2 = 9$ particles per cell.

## 7. The Gaussian assignment algorithm

In the case of the Gaussian assignment method, the above mentioned charge distribution characterizing the algorithm is a Gaussian distribution of the form

$$\rho_G(x, y) = \frac{q}{\sqrt{2\pi}\, s} e^{-(\frac{x^2+y^2}{s^2})}. \tag{9}$$

As before, $q$ is the total charge of the particle while $s$ is the half-width of the Gaussian.

In order to finally obtain a resolution of the order of that of the computation grid, the standard deviation $s$

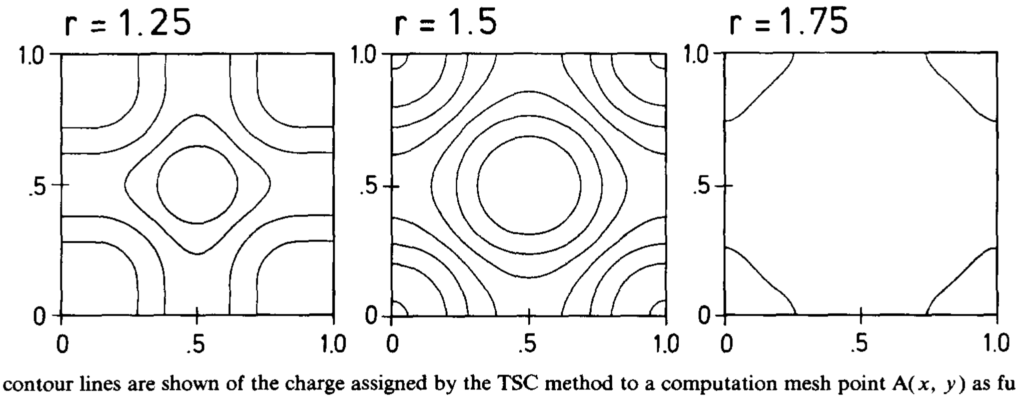

Fig. 7. 1% contour lines are shown of the charge assigned by the TSC method to a computation mesh point A($x$, $y$) as function of its position in a cell of the particle mesh for the cases $r = d_c/d_p = 1.25$, 1.5 and 1.75. A plot of the relative differences $D_{TSC} = [\rho^{(max)}_{TSC} - \rho^{(min)}_{TSC}]/\rho^{(max)}_{TSC}$ between maximum and minimum charge values assigned to an arbitrary computation mesh point as a function of $r$ is shown in fig. 10a.

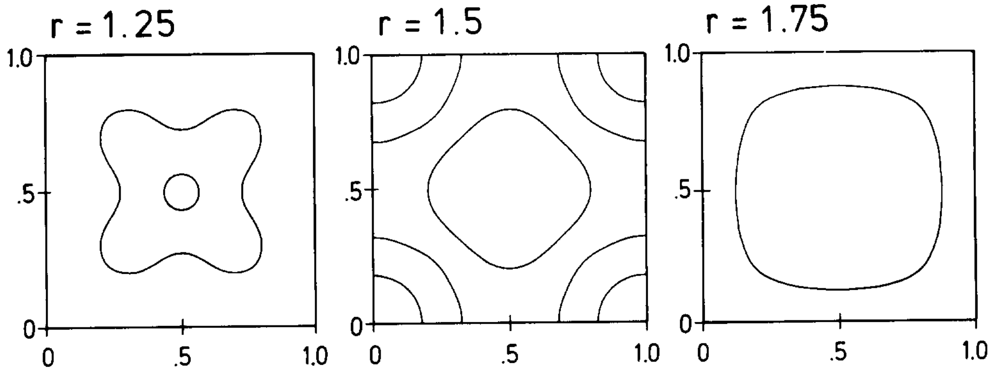

Fig. 8. 1% contour lines are shown of the charge assigned by the RTSC method to a computation mesh point A($x$, $y$) as function of its position in a cell of the particle mesh for the cases $r = d_c/d_p = 1.25$, 1.5 and 1.75. A plot of the relative difference $D_{RTSC} = [\rho^{(max)}_{RTSC} - \rho^{(min)}_{RTSC}]/\rho^{(max)}_{RTSC}$ between maximum and minimum charge value assigned to an arbitrary computation mesh point as function of $r$ is shown in fig. 10b.

should be about $d_c$. However, if the computation mesh is chosen smaller than the required resolution, for instance in order to obtain more precise results in the POISSON solving process using differences instead of derivatives, also large values of the above standard deviation can be taken in order to have less particles.

Note here that even though the Gaussian distribution has infinite range, it is not necessary to calculate the charge assigned to mesh points more than about $3s$ away from the center of the particle due to the fast decrease of the Gaussian. It is also worth mentioning that contrary to the methods discussed in ref. [4], this assignment algorithm is infinitely often differentiable which guarantees a very smooth dependence of the charge assigned to a computation mesh point as the particle moves in the mesh.

To investigate the smoothness of the Gaussian assignment algorithm we have plotted 1% contour lines of the charge assigned to a computation mesh point as a function of its position in a cell of the particle mesh for different values $r = d_c/d_p$ in fig. 9. However, since values of $r = 1.25$, 1.5, 1.75 as used in the previous sections here would show contour lines only if the step size would be 0.01% or less we have here chosen $r = 0.6$, 0.7 and 0.8. Fig. 10b shows a plot of the fluctuations $D_{GAUSS}$, i.e., the difference between maximum and minimum assigned values as a function of $r$ for the case of the Gaussian as well as the RTSC method. The plot shows that in case of the Gaussian algorithm these fluctuations are far below 1% for all values of $r$ greater than 1. The RTSC method, though much better than the NGP, AW, RAW and TSC methods (see figs. 10a and 10b), is by far inferior.

As one finds from figs. 9 and 10b, the fluctuations $D_{GAUSS}$ are negligible as soon as $r$ is greater than 1, i.e. the half-width $s$ is greater than or equal to the particle mesh size $d_p$. From fig. 10b one also reads that comparable values of $D_{RTSC}$ can only be obtained for $r > 3$. Note that in case of a mere statistical distribution of particles, a much larger average fluctuation of, for in-

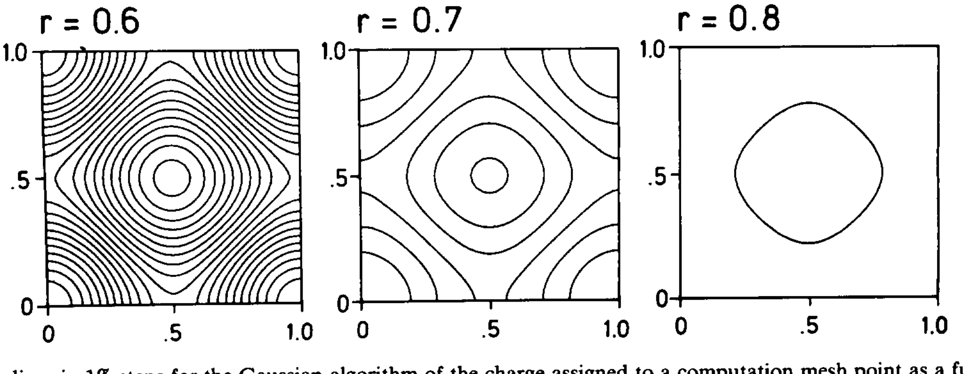

Fig. 9. Contour lines in 1% steps for the Gaussian algorithm of the charge assigned to a computation mesh point as a function of its position for the values $r = d_c/d_p = 0.6$, 0.7 and 0.8. A plot of the relative difference $D_{GAUSS} = [\rho^{(max)}_{GAUSS} - \rho^{(min)}_{GAUSS}]/\rho^{(max)}_{GAUSS}$ between maximum and minimum charge value assigned to an arbitrary computation mesh point as a function of $r$ is shown in fig. 10b.

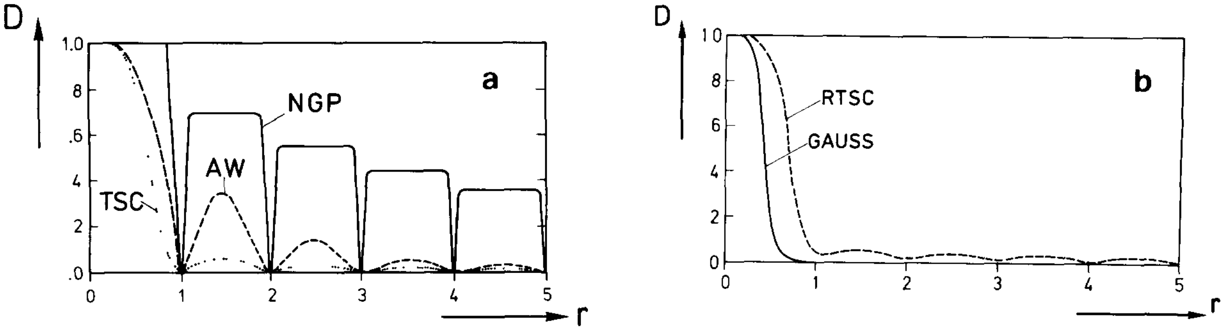

Fig. 10. The relative differences $D = [q^{(\max)} - q^{(\min)}]/q^{(\max)}$ between maximum and minimum charge value assigned to an arbitrary computation mesh point as a function of $r$ are shown for the assignment algorithms in (a): NGP, AW, TSC; in (b): RTSC and GAUSS. Note that for a perfect algorithm, $D$ should be negligible for $r > 1$.

stance, 5% requires about 400 particles per cell. In a 2D example, this entails that the number of particles required in the statistical case is at least 400 times higher and in the case of the RTSC method about $3^2 = 9$ times higher than in the Gaussian case. In a 3D example, the RTSC method would even require at least about 27 times as many particles as the Gaussian method. Note that with $r = 1$ and hence $s = d_p$ the Gaussian method yields a very smooth charge assignment in a case where the width of the particle is of the same magnitude as the distance between the simulation particles.

## 8. Comparison of the above assignment algorithms

As the above sections have shown, the number of particles required in particle-in-cell codes can be reduced drastically by initially arranging the individual particles on a regular mesh and choosing a good assignment algorithm. To compare the performance of the NGP, AW, TSC, RTSC and Gauss methods we have plotted in fig. 10 the relative differences $D$ between maximum and minimum charge values assigned to an arbitrary computation mesh point as function of $r = d_c/d_p$ and $r = s/d_p$ in the Gaussian case.

The performance may also be read from table 1 in which the different $D$-values are given in percent for $r > 1$, $r > 2$ and $r > 3$. The Gaussian algorithm presented here has by far the best performance reaching the level of $D = 1\%$ at about $r = 0.8$, followed by the RTSC method which reaches $D = 1\%$ at about $r = 3$. Note that thus with 1% accuracy the RTSC method requires about $(3/0.8)^2 \approx 14$ times as many particles as the Gaussian in the 2D case and about $(3/0.8)^3 \approx 53$ times as many particles as the Gaussian in the 3D case. Note here also that the RTSC algorithm was already superior to all other investigated algorithms.

Table 1
The maximum relative fluctuations in percent for different assignment functions and $r$-values

| $D$ | NGP | AW | RAW | TSC | RTSC | Gauss |
|---|---|---|---|---|---|---|
| $r > 1$ | 68 | 35 | 17 | 7 | 3 | 0.02 |
| $r > 2$ | 53 | 14 | 6 | 3 | 2 | $< 10^{-17}$ |
| $r > 3$ | 42 | 6 | 3 | 2 | 1 | $< 10^{-17}$ |

## 9. Performance of the above assignment algorithms in PIC simulations

In order to demonstrate the properties of some of the discussed assignment algorithms we simulated a 20 mA beam of 2.8 MeV protons along a field-free drift distance of 80 m. The simulation was done using the program BEAMTRACE [5]. This PIC code uses a two-dimensional POISSON solver and is able to handle arbitrary particle optical elements such as magnetic and electric bending fields and multipoles. It is part of the package GIOS-BEAMTRACE [6,7] of the University of Giessen for the design and study of particle optical systems.

The initial distribution was assumed to be parabolic and radially symmetric with an initial radius of $R = 4$ cm. The parabolic shape of the charge distribution was obtained by assigning different charges to the particles. With $x$ and $y$ denoting the position of the particle and $R$ the initial beam radius, the charge assigned to each particle was $q = \rho_0(1 - (x^2 + y^2)/R^2)$. The computation grid size $d_c$ was readjusted at each new computation of the electromagnetic fields such that the total computation grid was always slightly larger than the beam diameter.

The charge distribution in the particle beam was simulated in three different fashions:

(1) with a statistical initial arrangement of about 1500 simulation particles and the AW assignment method;

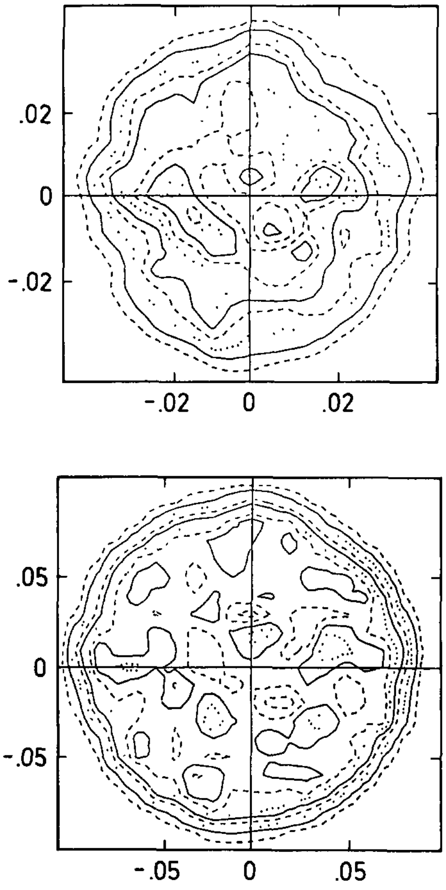

Fig. 11. The simulation of a laminar 20 mA beam of 2.8 MeV protons with initially parabolic charge distribution moving in a field free drift region with a length of 80 m. About 1500 particles were initially arranged statistically. The effect of widening and flattening is apparent only in a qualitative manner. The dashed contour lines mark 10, 40 and 70%, the solid lines 20, 50 and 80% and the dotted lines mark 30, 60 and 90% of the maximum charge density. The scaling is in meters.

(2) with a regular initial arrangement of about 300 simulation particles on a grid and the AW assignment method;

(3) with a regular initial arrangement of about 300 simulation particles on a grid and the Gaussian method.

The initial and final charge distributions for these three cases are shown in fig. 11, 12 and 13. Due to the parabolic structure of the initial charge distribution the resulting space-charge forces are nonlinear and lead to charge distributions which become more and more homogeneous. After a flight distance of about 80 m the charge distribution is almost perfectly homogeneous.

Fig. 11 shows the result of the simulation for a statistical initial distribution and the AW assignment method. The final charge distribution exhibits fluctuations of about 40% in agreement with section 2. The fluctuations are so large that the desired parabolic initial charge distribution is strongly disturbed. After the flight distance of 80 m only a widening and flattening is noticeable from the fact that the contour lines of the charge density accumulate at the edge of the beam. Altogether one can state that the simulation is of poor accuracy and allows a rough estimate only.

The same simulation was then carried out with a regular initial arrangement on a grid using again the AW assignment method. The grid constant of the regular initial particle mesh was chosen as $d_p = d_c$ which required about 300 particles using a 20 by 20 computation mesh. The choice of the initial $d_p = d_c$ implies that the assigned charge distribution should be very smooth according to section 5. This is the case according to fig. 12. However, the nonlinear forces eventually distort the initially regular mesh such that the distances

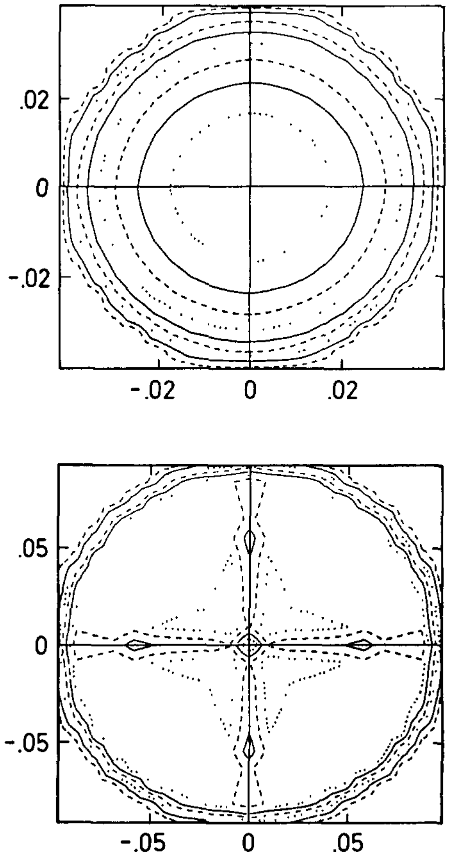

Fig. 12. The simulation of fig. 11 performed with about 300 particles initially arranged on a grid using the AW method. The effect of widening and flattening of the charge distribution can well be seen. The different contour lines mark the same percentages of the maximum charge density as in fig. 11. In the lower figure unphysical charge fluctuations caused by the AW algorithm are clearly visible.

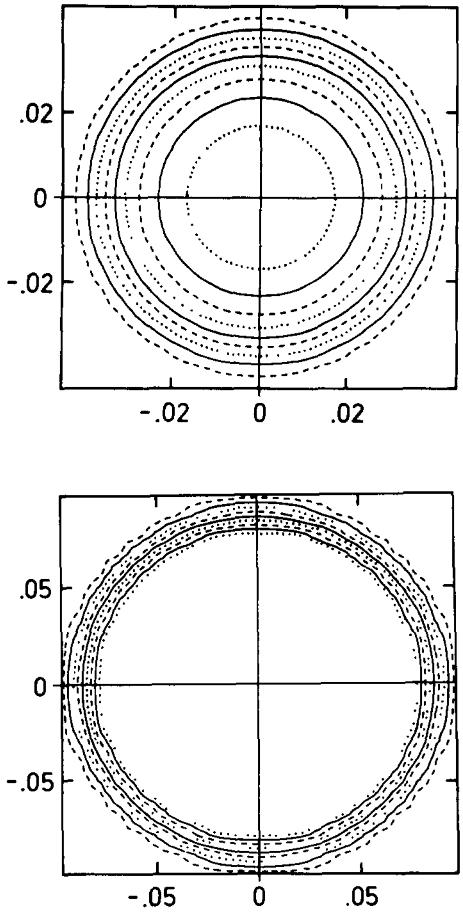

Fig. 13. The simulation of figs. 11 and 12 performed with about 300 particles ınitially arranged on a grıd using the Gaussian assignment algorithm. The different contour lines again mark the same percentages as ın fig. 11. Note that in the lower figure no unphysical charge fluctuations are noticeable.

between two mesh points depend upon their positions. The charge distribution of fig. 12b obtained with 300 particles also exhibits nonphysical fluctuations of up to 40% in agreement with section 5. Finally the same simulation was done with a regular initial particle mesh and the Gaussian assignment algorithm. The results are shown in figs. 13 and 14 obtained with 300 particles as contour plots and as three-dimensional plots, respectively. As predicted in section 7, the unphysical charge fluctuations are much smaller in this case even though the mesh gets distorted and hence $r = s/d_p$ fluctuates. This is true even though it required only about 20% of the number of particles used in the by far less accurate statistical case. Note that about one particle per cell was used – a number far beyond the realms of most PIC codes. Since the computation effort of PIC codes depends mainly upon the number of particles used, the savings in computing time are considerable.

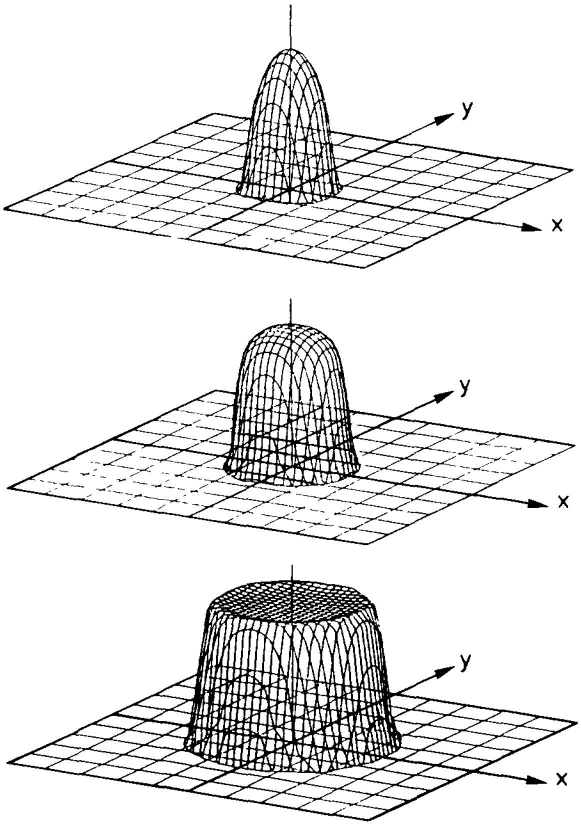

Fig. 14. The sımulation of figs. 13 shown as three-dimensional plots.

## 10. An example with a four-dimensional phase-space distribution

The next example will present results obtained with BEAMTRACE using the Gaussian method for an example with four-dimensional phase space [8,9]. It is a system for the focusing of an intense 10 GeV, 1250 A beam of $^{209}\mathrm{BI}^+$ with an initially waterbag-like charge distribution with a phase-space volume of 56 × 36 × 22 × 34 $\mathrm{mm}^2$ $\mathrm{mrad}^2$ as required by the HIBALL study [9] for an inertia confined fusion reactor.

The final focusing system consists of two quadrupole triplets with two magnetic sector fields of opposite bending direction and bending angles of about 3° each in between. The total length of the system is about 100 m. Its detailed geometric layout was optimızed with the program GIOS under the assumption of linear space-charge forces and can be found in refs. [8,9].

In the BEAMTRACE simulation the widths $s$ of the Gaussians were chosen to be about 1/15 of the respec-

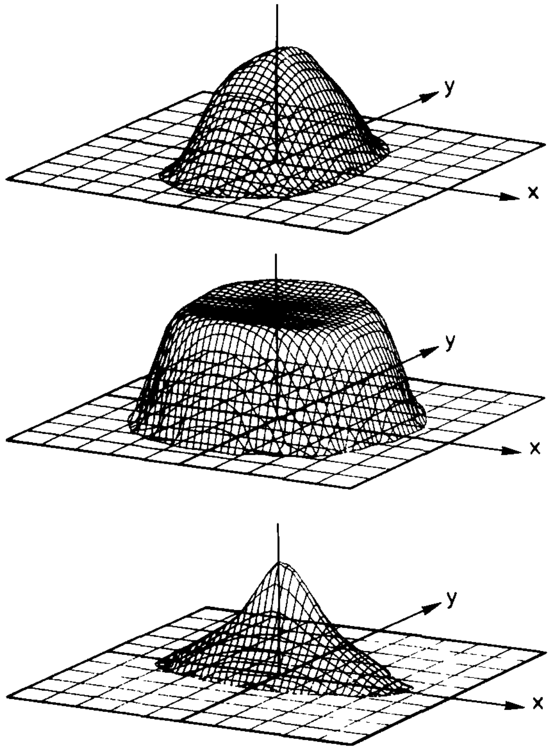

Fig. 15. The charge distribution at the positions $z=0$ m, $z=57.5$ m and $z=101$ m of the final focusing system of the HIBALL study [9–11] generated by BEAMTRACE using the Gaussian assignment algorithm. Note that the scaling of the pictures is readjusted to the actual beam widths which are about ±2.5 cm in the first picture, ±25 cm in the middle picture and ±0.5 cm in the last picture.

tive $x$–$y$ beam diameter and about 1/7 of the respective $v_x$–$v_y$ distribution. Assigning the same charge to each particle as long as its phase-space center point was inside the waterbag ellipsoid and omitting all particles lying outside, the number of particles totalled about 4000.

In fig. 15 the initial, an intermediate and the final charge distributions obtained with BEAMTRACE are plotted where the $x$–$y$ scaling was readjusted to the corresponding beam diameters. As fig. 15 shows, the charge distribution is not stigmatically focused at the final image due to nonlinear space-charge effects which could not be taken into account in the optimization process with GIOS. As fig. 15 shows further, the chosen number of 4000 simulation particles produced rather smooth charge distributions throughout the optical system. This suggests that a few hundred particles could be quite sufficient for a thorough ion optical investigation of a system with non-negligible space charge forces.

## Acknowledgement

For financial support we are grateful to the "Bundesminister für Forschung und Technologie" and the "Studienstiftung des deutschen Volkes".

## References

[1] P. Lapostolle and R. Le Bail, Comp. Phys. Commun. 4 (1972) 333.

[2] P. Tanguy, Proc. Proton Linear Accelerator Conf. (1970) p. 771.

[3] I. Haber, AIP Conf. Proc. no. 136 (1986).

[4] R.W. Hockney and J.W. Eastwood, in: Computer Simulation using Particles (McGraw-Hill, New York, 1981).

[5] M. Berz, Diploma Thesis, University of Giessen (1983) unpublished.

[6] H. Wollnik, J. Brezina, M. Berz and W. Wendel, Proc. AMCO-7, GSI-Rep. THD-26 (1984) p. 679.

[7] H. Wollnik, J. Brezina and M. Berz, Nucl. Instr. and Meth. A258 (1987) 408.

[8] H. Wollnik, J. Brezina, M. Berz and W. Wendel, Proc. Int. Symp. on Heavy Ion Accelerators and their Applications to Inertial Fusion (INS, Tokyo, 1984) p. 433.

[9] B. Badger et al., KfK 3840, FPA 84-4, UWFDM-625 (1985).