Synthesizing a four-dimensional beam particle distribution from multiple two-dimensional views*

A. Friedman and D. P. Grote
Lawrence Livermore National Laboratory, Livermore, California 94550

C. M. Celata and J. W. Staples
Lawrence Berkeley National Laboratory, Berkeley, California 94720

The transverse dynamics of a nearly-monoenergetic particle beam are described by the evolution of the 4D distribution $f(x,y,x',y')$, where x and y are the transverse spatial coordinates and $x' \equiv p_x/p_z$ and $y' \equiv p_y/p_z$ are the corresponding momentum components divided by the longitudinal momentum component. In present-day experimental practice, such beams are often diagnosed by passing them through an axially-separated pair of slits parallel to the *y* axis. This selects for x and x′ and integrates over y and y′. A sequence of pulses (with the slits at various x positions) yields a 2D projection of the beam phase space, $f(x,x')$. Another scanner might yield $f(y,y')$ or, using crossed slits, $f(x,y)$. The challenge is that a small set of such 2D scans does not uniquely specify $f(x,y,x',y')$; correlations in planes other than those measured are unknown. We have developed Monte-Carlo methods and formulated physically-motivated constraints to synthesize a "reasonable" set of particles having 2D projectional densities consistent with the experimental data. Such a set may be used to initialize simulations of the downstream beam. The methods and their performance on model problems are described.

PACS Codes: 29.27.-a, 29.27.Fh, 29.85.+c, 52.59.-f, 52.59.Fn, 52.59.Sa, 52.65.-y, 52.65.Rr

I. INTRODUCTION

In a wide variety of particle-beam applications, the beams are not well-described by Gaussian or other simple profiles. In induction accelerator drivers for heavy-ion beam-driven inertial fusion energy (Heavy Ion Fusion, or HIF), and in present-day experiments exploring the beam physics of such drivers, the beams are space-charge-dominated, with spatial profiles that are observed to be far from Gaussian [1]. In other applications as well, *e.g.* radio-frequency quadrupole (RFQ) beam injection [2], the beam emerges from an injector with a distribution that is not well-described by a small number of moments.

Discrete-particle simulations are commonly employed in efforts to better understand the behavior of such particle beams. Integrated simulations, beginning at the source and carried out in full time-dependent 3D or a reduced description when appropriate, offer the promise of the greatest fidelity in describing the long-term beam behavior. However, the source, injector, and beam line upstream of a section of interest in an experiment will in general not be completely characterized. Thus, the beam distribution function in a simulation beginning at the source will in general "drift" away from the experimentally measured beam distribution, and it can be challenging to adjust the beamline description and other parameters in the simulation to bring them back into agreement. Thus, as a complement to integrated simulations and as a tool for routine experimental analysis, we

*This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore and Lawrence Berkeley National Laboratories under Contract Nos. W-7405-Eng-48 and DE-AC03-76SF00098.

are developing the ability to launch particle simulations of "real" experiments, using an initial beam particle distribution derived from experimental measurements at a station partway along the machine.

A longitudinally-cold steady flow is often reasonably well described by a particle distribution in the transverse 4D phase space $f(x,y,x',y')$. Here, x, y, and z are orthogonal Cartesian coordinates (with x and y transverse to the nominal beam propagation direction and z along it), p_x, p_y, and p_z the corresponding mechanical momentum components, and $x' \equiv p_x/p_z$ and $y' \equiv p_y/p_z$. It is customary to work with the experimentally measured quantities x' and y' rather than with the momenta. In reality, beams have a small spread in p_z, and indeed, particle orbits in the transverse 4D space are not independent of p_z. Nonetheless, a longitudinally-cold beam, such as we assume in this paper, is a useful approximation in many applications.

The most common beam phase-space diagnostic for intense beams is the two-slit emittance scan, which is used to measure a "projectional" beam emittance [3] in the direction perpendicular to the slits. Such an apparatus is shown schematically in Fig. 1. In this diagnostic, "paddles" containing the slits are moved in increments, most commonly by stepping motors. Behind the second paddle, a shallow, reverse-biased Faraday cup is commonly employed to generate the signal. A map of the beam distribution is built up over many pulses, between each of which one paddle or the other is moved (multi-slit diagnostics sometimes are employed, to similar effect). The position coordinate, x, at which the intensity is measured during any given beam pulse is determined by the position of the first slit (at the left in the figure). The transverse velocity angle of the measurement, x', is determined by the position of the second slit relative to that of the first. The arrows in the figure denote particle trajectories, and show how each slit admits only a small fraction of the particles reaching that plane. A scan yields a 2D projection of $f(x,y,x',y')$, *e.g.*, $f(x,x')$, with the data most commonly (but not always) known at evenly spaced points on a slanted grid in (x,x') that roughly overlays the beam. In addition to the parallel-slit emittance scan, a crossed-slit intensity map is sometimes used to map out the distribution of beam current density in the transverse (x,y) plane.

Note that some diagnostic techniques can measure the 4D transverse phase space directly, again in a sampled sense. For example, a "gated beam imager" consisting of a multi-hole plate upstream of an imaging plane, microchannel plate, and detector has been used to obtain time-dependent 4D ion beam distributions [4]. A version of this diagnostic is to be developed for the higher-current experiments currently being pursued in the HIF program. Such diagnostics will become far more challenging as the kinetic energy and current are increased. For the present, the principal diagnostics in the HIF research effort (and in many other accelerator applications) yield projectional emittances and transverse density profiles. We consider the problem of synthesizing "reasonable" distributions from such data sets.

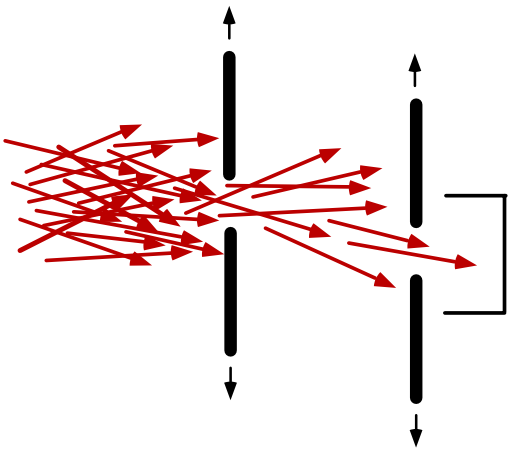

FIG. 1. Schematic depiction of the principle behind a two-slit emittance scanner.

In Section II, below, we state the problem and discuss its nature, and elaborate on the goals of this research. In Section III, we present a number of methods that we have developed for synthesis of 4D distributions. In Section IV, we describe a model problem and summarize the performance of the various methods when applied to it. In Section V we describe a problem taken from actual practice in loading a beam into an RFQ accelerator, and show application of some of the methods to that problem. In Section VI we describe another problem, this time using a simulation of an electrostatic-quadrupole (ESQ) injector. This problem is especially useful as a benchmark, since, in contrast with the experimental data available to us, the simulation data contains the full 4D phase space information. A final discussion and some recommendations are presented in Section VII.

II. STATEMENT AND DESCRIPTION OF THE PROBLEM

We seek to carry out kinetic simulations of beams (using discrete-particle methods, or continuum Vlasov methods wherein f is evolved on a 4D grid) using an initial beam state derived from experimentally measured data. Thus, we are trying to solve the problem:

> Given (x,x′), (y,y′), and optionally (x,y) projections of f(x,y,x′,y′), synthesize an f(x,y,x′,y′) which is consistent with the measured data.

As stated immediately above, the problem itself is underdetermined. The under-determined nature of the problem is discussed below. It is necessary to add additional constraints, which lead us to reformulate the problem as:

> Given (x,x′), (y,y′), and optionally (x,y) projections of f(x,y,x′,y′), synthesize a "reasonable" f(x,y,x′,y′), which is consistent with the measured data and with the general nature of the class of beams under study.

This formulation of the problem is imprecise, but it accurately reflects the goals of the research effort described herein. Even within this revised formulation, the problem is *still* underdetermined because we do not have precise criteria for "reasonableness," and indeed we show two classes of synthesis methods which yield results on model problems consistent with this latter formulation. A more precise formulation of the problem, perhaps one based on maximum entropy considerations or other criteria, would be desirable, but has yet to be realized. Nonetheless, within the framework of this heuristic formulation, considerable benefit can be obtained by use of a "synthesized" distribution for downstream particle simulations.

To better illustrate the underdetermined nature of the 4D synthesis problem, we consider a simpler analogue, the problem of tomographic reconstruction of a 2D distribution from a small number of 1D projectional densities. As shown in Fig. 2, many different 2D distributions can share a common pair of 1D "Cartesian" projections. The shaded areas in figure 2(a)-(c) depict possible 2D density distributions which are consistent with the presumed "measured" 1D densities $f(x)$ and $f(y)$ shown along the lower and left edges of the figure. In general, and roughly speaking, 2D methods analogous to the methods we describe herein would tend to produce distributions resembling the one shown in (c).

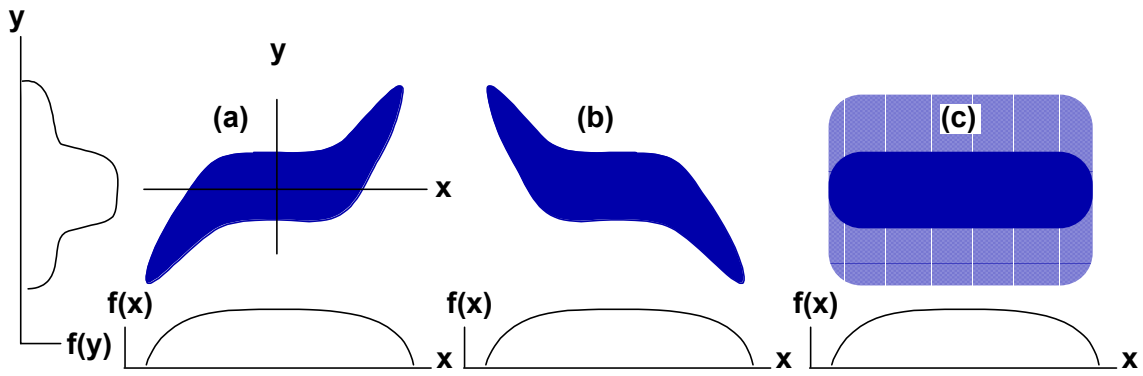

FIG. 2. Depiction of a two-dimensional analogue problem, as described in text.

Another way to view the underdetermined nature of the problem is to consider the amount of information necessary to specify a 4D distribution at any chosen granularity or "resolution." Consider, *e.g.*, $f(x,y,x',y')$ on a 4D grid with 20 points along each Cartesian axis. In such a case we would need $20^4 = 1.6\times10^5$ points to fully specify the 4D phase space density. The measured (x,x') and (y,y') projections each consist of 400 data points, yielding a total of 800 data points, far fewer than is demanded by the complete description. Should the intensity $f(x,y)$ also be available, (highly desirable, as discussed later), then 1200 data points are known. We usually don't have direct information about the other "Cartesian" projections (x,y'), (y,x'), or (x',y'); but even if we knew them, plus a few other projections at "angles" to the principal Cartesian axes, the information available would be far less than that needed to specify the 4D distribution.

Thus, to make the problem well posed, we must add constraints based on considerations separate from the data, *e.g.*, how the beam was made, and/or how it may be presumed to have evolved, and/or "what beams generally look like." There is considerable arbitrariness in the choice of these constraints, and so multiple prescriptions may reasonably be employed. We proceed to describe the methods explored to date.

III. METHODS FOR SYNTHESIZING FOUR-DIMENSIONAL DISTRIBUTIONS

We are exploring a family of methods which yield usable "synthesized" 4D density distributions. All are based on Monte-Carlo sampling techniques, and are intended to satisfy the heuristically-stated version of the problem as noted above.

Naïve sampling — drawing random points from a 4-box and accepting each with a likelihood proportional to the product of the input probabilities, $f(x,x') \times f(y,y')$ — leads to artifacts in the synthesized distribution which are clearly nonphysical. Specifically, "corners" appear in the projectional densities in those Cartesian planes for which there is no direct data; that is, the density contours appear rectangular. This arises because the naive sampling assumes that the likelihood of finding a particle in a particular bin in (x,x') is independent of the bin's location in y and y', which is in general not the case.

The first of the viable methods was developed by one of us (Staples), drawing on discussions with J. Stovall [5]. It has been successfully applied to problems associated with injecting a beam into an RFQ accelerator. This method also draws random points from a 4-box and accepts each with a likelihood proportional to $f(x,x') \times f(y,y')$, but then it applies "clipping," *i.e.*, removal of points outside a prescribed 4-volume. Because of the clipping, the synthesized projectional (x,x') and (y,y') phase spaces differ slightly from the measured ones. The naive sampling result is recovered as the clipping volume is enlarged to include the entire 4-box. This first "sampling" method is denoted "S" in the sections below.

Two newer classes of algorithms exactly reproduce, in the limit of large numbers of particles and extremely fine data grids, the measured (x,x') and (y,y') data; some can reproduce the measured (x,y) data as well when that data is available. These "generalized" methods are denoted "G" in the sections below, followed by a digit identifying the actual method used. The methods in one class resemble the S method in that they also use random sampling from a user-specified sampling (clipping) region within a 4-box; they are, essentially, statistically-correct variants of the S algorithm. Another class of methods is based on the assumption of a certain kind of anticorrelation of (x,x') and (y,y') densities, as described below.

None of the newer methods is "fully virtuous;" that is, different problems seem to benefit from the use of different methods. All can give the "exact" answer (limited only by fluctuations and grid resolution errors) on certain model problems. Thus, in this paper we describe the most useful of the methods we have developed, outline their strengths and weaknesses, and show examples of their use.

We present three statistically-correct sampling-region methods. The first of these, G1, is based on the use of "bin counts" to ensure reproduction of the data in the input planes within resolution-induced errors; the use of "bins" minimizes the fluctuation error but introduces other difficulties, described below. The second, G2, uses corrected probabilities to obtain consistency with the input data; it is an extremely simple and robust method, but does introduce bin-count fluctuations in f which scale inversely as the square root of the number of particles in the bin. The third, G7, is a three-plane variant of G1, using (x,y) data as additional input. We also present one anticorrelation-based method, G3, which requires no user-specified sampling region; it works well when the assumptions upon which it is based are satisfied, but fails in other cases.

Preliminary Steps

For all algorithms, certain preliminary steps are generally carried out. Any known biases in the "raw" experimental data, e.g. offsets in the zero-beam signal, should be removed first. The raw data often suffer from some level of random noise, and so "thresholding" of some type is typically used. Thresholding in our testbed program is controlled by the input parameters THRESHXXP, THRESHYYP and THRESHXY (corresponding to the principal planes for which experimental data may be available). Two prescriptions are available; for, *e.g.*, the (x,x′) plane, the prescription to be used is selected by the sign of THRESHXXP. In the first prescription, a threshold value is subtracted from all data points, with any negative results zeroed; in the second, any points below a threshold value are zeroed:

$$\text{for THRESHXXP} > 0\text{: } f(x,x') = \max\{0,\, f_{raw}(x,x') - \text{THRESHXXP}\, \max[f_{raw}(x,x')]\}\ ; \tag{1}$$

$$\begin{aligned}\text{for THRESHXXP} < 0\text{: } f(x,x') &= f_{raw}(x,x') \quad \text{if } f_{raw}(x,x') > |\text{THRESHXXP}|\, \max[f_{raw}(x,x')] \\ &= 0 \qquad\quad \text{if } f_{raw}(x,x') < |\text{THRESHXXP}|\, \max[f_{raw}(x,x')]\end{aligned} \tag{2}$$

where the usual maximum operator, "max," is employed. The former prescription yields data that tends to approach zero smoothly, but alters all data points; the latter tends to yield a "cliff" at the edge of the distribution, but leaves the retained data unchanged. For "good" data with low noise, they may be expected to perform comparably.

The thresholded signal values (values of f) are then renormalized to range from zero to unity. This yields the "true" experimental data sets: $f(x,x')$, $f(y,y')$, and in some cases $f(x,y)$, that serve as inputs to the actual synthesis process. Further operations do not alter the f values, but rather transform the nodal coordinates with which they are associated.

For computational convenience we translate the experimental data along each of its axes so that the center of mass is at (0,0) in each measured plane, so that (*e.g.*) $\langle x \rangle = 0$ and $\langle x' \rangle = 0$, etc. It is straightforward to shift the distribution back after it has been synthesized. The translation is accomplished by changing the nodal coordinates indexed by the "logical" grid coordinates i and j:

$$x_{ij} = x_{raw\ ij} - \langle x_{raw} \rangle \tag{3}$$

$$x'_{ij} = x'_{raw\ ij} - \langle x'_{raw} \rangle \tag{4}$$

where the averages are computed by multiplying the nodal-coordinate expressions by the measured signal and summing:

$$\langle x_{raw} \rangle = \Sigma_{ij}\ x_{raw\ ij}\ f_{ij}\ /\ \Sigma_{ij}\ f_{ij} \tag{5}$$

$$\langle x'_{raw} \rangle = \Sigma_{ij}\ x'_{raw\ ij}\ f_{ij}\ /\ \Sigma_{ij}\ f_{ij}\ . \tag{6}$$

Note that we have indexed x and x′ each by two subscripts, i and j. In most cases it will suffice to use a single subscript for each, that is, x_i and x'_j. The notation we have used allows for the possibility of a nonuniform mesh wherein, *e.g.*, the set of nodal x′ locations may differ at different x locations (some HIF experimental data is indeed taken that way).

It is inefficient to overlay a Cartesian grid on the projectional phase space density of a strongly converging (or diverging) beam because the range of x′ values associated with the mean convergence can easily exceed that due to the thermal spread. Furthermore, it can be difficult to display the phase space of such a beam because a contour plot can resemble a thick line. Thus, the mean convergence or "tilt" in each plane (*e.g.*, linear correlation of x and x′) is generally removed by subtracting the product of the x values of the data nodes and the mean slope from the x′ values of the nodes at which the data is obtained. This leads to a revised set of nodal coordinates (denoted, here only, by upper case symbols, which we proceed to drop in favor of lower case in the following sections):

$$X'_{ij} = x'_{ij} - \psi_x\ x_{ij} \tag{7}$$

$$X_{ij} = x_{ij} \tag{8}$$

where, assuming that the center of mass has been shifted to the origin as mentioned above, the convergence ψ_x is given by:

$$\psi_x = \langle x\ x' \rangle / \langle x^2 \rangle\ . \tag{9}$$

Here, the averages are computed using the measured distribution functions:

$$\langle x\ x' \rangle = \Sigma_{ij}\ (x\ x')_{ij}\ f_{ij}\ /\ \Sigma_{ij}\ f_{ij} \tag{10}$$

$$\langle x^2 \rangle = \Sigma_{ij}\ (x^2)_{ij}\ f_{ij}\ /\ \Sigma_{ij}\ f_{ij}\ . \tag{11}$$

When the distribution function f_{ij} is plotted using the revised nodal coordinates (X_{ij},X'_{ij}) as the independent variables, in a contour plot, a colored (filled) mesh plot, or a "mountain range" plot, it retains no net "tilt," though it may have much structure.

When the raw data are taken on a regular but "slanted" grid, it may be preferable to remove, not the mean convergence of the measured distribution, but rather the mean tilt of the grid of points on which the measurements are made. Then, one is left with data on a regular Cartesian grid and no further mapping to such a grid (as described below) is necessary; but some net linear correlation between positions and transformed velocities remains. In either case, it is straightforward to restore the net convergence to the synthesized distribution by means of an operation inverse to the original transformation.

A complication arises in practice when, as is often the case, the (x,x′) and (y,y′) slit scans, and perhaps the (x,y) scan as well, are not carried out at the same longitudinal position. Then it is necessary to map the measured data onto new grids at a common value of z. This may be accomplished only approximately, perhaps using a linear transformation based on velocities and applied focusing forces at the measured plane and a simple model for the space charge, or perhaps through a more elaborate procedure, yet to be worked out, more akin to a short simulation run.

Another complication can arise when, as is highly desirable, the spatial density $f(x,y)$ is measured along with $f(x,x')$ and $f(y,y')$: the one-dimensional projectional densities $n_1(x) = \int dx'\ f(x,x')$ and $n_2(x) = \int dy\ f(x,y)$ may disagree with each other. This may arise because the various 2D planes use differing grids for (*e.g.*) x; because the data in the coordinate being integrated over is merely sampled, so that an exact integration is impossible; because the measurements are made at differing values of z; and/or because of noise or other experimental measurement errors. If all three planes are to be used in the synthesis, some reconciliation procedure is necessary. The procedure to be used will depend upon the relative resolution and accuracy of the various measurements. For example, if the spatial density scan is deemed most trustworthy, then the densities on the $f(x,x')$ grid may be renormalized to bring $n_1(x)$ into agreement with $n_2(x)$. Some of the statistical methods may be robust to small errors of this type. We have, as yet, only done three-plane syntheses using simulated data, so we have yet to face this difficulty in practice.

Yet another complication arises if the raw data is accumulated on an irregular grid. In such cases, a remapping onto a (usually somewhat finer) regular mesh is carried out (for some of the methods this may not be strictly necessary). Here care is needed to avoid artificial spreading of the distribution. It is in general preferable to employ a "nearest grid point" remapping, where the projectional phase-space area associated with each point of the original data grid matches (at least approximately), in size and in position, the area of the set of points on the regular grid into which that value is mapped.

In our testbed program, the net "tilt" (convergence) is removed and the remapping to the regular mesh carried out before any thresholding is performed. For the remapping we

employ, and because the experimental data with which we are dealing always uses fixed increments in x and x′ (even when an irregular grid is used, as in some HIF experiments to which we are beginning to apply these methods), there is a one-to-n correspondence of raw nodes to remapped nodes. Thus the order of the remapping and thresholding operations is usually not important. However, the computed tilt does vary as the thresholding is changed; therefore it is in general best to apply thresholding first, before calculating ψ_x and ψ_y and transforming away the tilt. None of the cases described in this paper employ remapping.

First Two-Plane Sampling Method ("S")

Method S:

Carry out the following steps repeatedly in a loop, until the requisite number of particles have been generated:

1) Generate a random point in the 4-box bounded by the extremes of the (thresholded and shifted) measured data , and two random numbers, ρ_1 and ρ_2, on the unit interval

2) With likelihood proportional to $f(x,x') \times f(y,y')$, tentatively accept the point; that is, keep it if $f(x,x') > \rho_1$ and $f(y,y') > \rho_2$ (it is statistically equivalent to keep the point if $f(x,x') \times f(y,y') > \rho_1$, with only a single random number generated)

3) Apply clipping, as controlled by the input parameter CLIP, which is generally set to unity: Accept point as a particle in the synthesized distribution if and only if its coordinates (with each component normalized to the extent in x, y, x′, or y′ of the region where f is nonzero) lie inside all of:

   a) four "4-cylinders," the extrusions of circles of radius CLIP in the (x,y), (x′,y′), (x,y′), and (y,x′) planes, along the two Cartesian axes out of the corresponding plane

   b) a 4D hyperellipsoid of radius $2^{1/2}$ CLIP

To clarify this, in implementing Step 3) above we define a "flag" variable IFIN, which denotes whether each trial point lies in the clipping volume, as the product of a series of logical results (either 0 or 1, denoting false and true, respectively) associated with each volume tested in computing the intersection:

$$\begin{aligned} \text{IFIN} = {} & \{(x/a)^2 + (y/b)^2 < \text{CLIP}\} \times \{(x'/a')^2 + (y'/b')^2 < \text{CLIP}\} \\ & \times \{(x/a)^2 + (y'/b')^2 < \text{CLIP}\} \times \{(x'/a')^2 + (y/b)^2 < \text{CLIP}\} \\ & \times \{[(x/a)^2 + (y/b)^2 + (x'/a')^2 + (y'/b')^2]^{1/2} < 2^{1/2}\ \text{CLIP}\} , \end{aligned} \tag{12}$$

where the quantities a, b, a′, b′ are normalizing factors for the x, y, x′, and y′ coordinates, respectively. That is,

$$a \equiv (x_{lastr} - x_{lastl}) / 2, \tag{13}$$

where

$$x_{lastr}\ (x_{lastl}) \equiv \text{the maximal (minimal) } x_{ij} \text{ such that } f(x_{ij},x'_{ij}) > 0, \tag{14}$$

and similarly for b, a′, and b′. The alternative definition:

$$\begin{aligned} a &\equiv \text{the maximal } |x_{ij}| \text{ such that } f(x_{ij},x'_{ij}) > 0 \\ &= \max(|x_{lastr}|,|\ x_{lastl}|) \end{aligned} \tag{13′}$$

appears more natural and may in fact be superior in many cases. However, the use of the former definition can yield a more compact distribution for any given value of the input parameter CLIP, at the expense of a greater degree of clipping. alternatively, if the clipping volume is centered on $(x_{mid},y_{mid},x'_{mid},y'_{mid})$, where $x_{mid} \equiv (x_{lastr}+x_{lastl}) / 2$, etc., rather than on the center of mass, then the prescriptions (13) and (13′) are equivalent. The centering of the clipping volume is, in our testbed program, controlled by the input variable CENTER, which can be set to either AVG or MID. For beams with a significant displacement of $(x_{mid},y_{mid},x'_{mid},y'_{mid})$ from $(x_{avg},y_{avg},x'_{avg},y'_{avg})$, the use of AVG can lead to a clipping region which is bounded by one or more sides of the 4D data grid.

This method has worked well on the RFQ problem to which we have applied it (as described below). However, it suffers from two shortcomings. Firstly, the clipping leads to some inconsistency with the measured data. Secondly, the clipping volume, and in fact the concept of clipping itself, is *ad hoc* in nature. The methods we proceed to discuss attempt to address these issues. We begin with a discussion of the sampling volumes used in the generalized methods.

Sampling Volumes for the Generalized Methods

For the methods described below, several options for the sampling region (the region retained in the clipping process described above) have been explored. They are described here for convenient reference. In our test program, the input parameter CLIPALG is used to make the selection.

CLIPALG = 0: This first option yields a sampling region similar to that of the S method, that is, the intersection of the interiors of an upright 4-ellipsoid and four 4-cylinders, except for a minor generalization: the radii of the cylinders, CLIP, and of the ellipsoid, CLIPELLIPSOID, are independently set by the code user. We thus have:

$$\begin{aligned}\text{IFIN} = &\{(x/a)^2 + (y/b)^2 < \text{CLIP}\} \times \{(x'/a')^2 + (y'/b')^2 < \text{CLIP}\} \\ &\times \{(x/a)^2 + (y'/b')^2 < \text{CLIP}\} \times \{(x'/a')^2 + (y/b)^2 < \text{CLIP}\} \\ &\times \{[(x/a)^2 + (y/b)^2 + (x'/a')^2 + (y'/b')^2]^{1/2} < \text{CLIPELLIPSOID}\} \ .\end{aligned} \tag{15}$$

CLIPALG = 1: This option uses only a 4-ellipsoid of radius CLIPELLIPSOID, but "tilts" it, shifting x′ of the edges of the sampling region by an amount linear in the x coordinate (and similarly for y′). The tilting is adjusted so that the extremes of the ellipsoid in x have the same x′ as the bin with the largest value of |x| (and similarly for y). We have:

$$\text{IFIN} = \{(x/a)^2 + (y/b)^2 + [(x'-s_x x)/c']^2 + [(y'-s_y y)/d']^2\}^{1/2} < \text{CLIPELLIPSOID} \ , \tag{16}$$

where the "slope" s_x is computed by averaging x′ at the extreme edge of the beam, at x_{lastr} or x_{lastl}, whichever has the larger magnitude, *e.g.*,:

$$s_x \equiv \langle x'(x_{lastr}) \rangle \ / \ x_{lastr} \tag{17}$$

and the normalizing factor c′ is:

$$c' \equiv \text{the maximal } |x'_{ij}-s_x x_{ij}| \text{ such that } f(x_{ij},x'_{ij}) > 0 \ , \tag{18}$$

and similarly for the slope s_y and the normalizing factor d′.

CLIPALG = 2: This option also uses only a 4-ellipsoid, but generalizes the prescription used in CLIPALG = 1 by shifting both x′ and x coordinates of the boundary of the sampling volume for a more tightly-fitted ellipsoid (and similarly for y):

$$\begin{aligned}\text{IFIN} = &\{[x-q_x(x'-s_x x)]^2/h_x^2 + [y-q_y(y'-s_y y)]^2/h_x^2 \\ &+ [(x'-s_x x)/c']^2 + [(y'-s_y y)/c']^2\}^{1/2} < \text{CLIPELLIPSOID} \ ,\end{aligned} \tag{19}$$

where q_x and h_x are given by:

$$q_x \equiv x_{ij*} \ / \ x'_{ij*} \tag{20}$$

$$ij* \equiv \text{indices of the maximal } |x'_{ij}| \text{ such that } f(x_{ij},x'_{ij}) > 0 \tag{21}$$

$$h_x \equiv \text{the maximal } |x_{ij}-q_x(x'_{ij}-s_x x_{ij})| \text{ such that } f(x_{ij},x'_{ij}) > 0 \ , \tag{22}$$

and similarly for q_y and h_y.

CLIPALG = 3: This option was specially developed for the mock problem described below; it effectively wraps the sampling volume precisely around the waterbag boundary of that problem:

$$\text{IFIN} = x^2 + y^2 + (x'-\mu x^3)^2 + y'^2 < 1 \ , \tag{23}$$

where μ is an input parameter generally set to 1.75.

As will be seen, the use of a small "smooth" sampling volume whose "equators" match the boundaries of the measured data in two planes appears highly desirable; but the synthesis of such a volume seems hard to accomplish in general. This in part motivated the development of the "anticorrelation" method, described below.

Two-Plane Method using Bin Counts ("G1")

This first "G" method is a statistically-correct variant of the "S" algorithm which reproduces, in the limit of many particles, the measured (x,x′) and (y,y′) data. It too is based on random sampling from a 4-box.

The method begins by assigning target "counts" N(x,x′) and N(y,y′) of the numbers of particles to be loaded into each "bin" in (x,x′) and into each bin in (y,y′), proportional to the measured f in that bin. Since the bin counts must be integers and yet must sum to the desired total number of simulation particles, a simple scaling and rounding procedure applied to f(x,x′) and f(y,y′) may be insufficient. A first effort to resolve this relied on varying the scaling factor (*i.e.*, the nominal target number of simulation particles) until the desired number of simulation particles was loaded. This too failed when bin counts were obtained via "nearest grid point" weighting of simulation particle coordinates, since multiple cells tended to have the same input f, and their bin counts were incremented as a block when the scaling was adjusted past a threshold. A similar problem would occur when experimental data is digitized using relatively few bits. Other procedures based on adjusting the rounding threshold were observed to fail similarly. A procedure which does seem to work well is to add a small random number to the pre-rounding bin counts; then bins are incremented individually as the scaling is gradually adjusted, and it is possible to obtain exactly a sum of bin counts equal to any desired number of synthesized particles.

Method G1:

Assign target "counts" N(x,x′) and N(y,y′) to the bins as described above.

Carry out the following steps repeatedly in a loop, until the requisite number of particles have been generated and all "bin counts" have been decremented to zero, or until a user-specified maximum number of passes through the loop have been completed, whichever occurs first:

1) Generate a random point in the 4-box bounded by the extremes of the (thresholded and shifted) measured data

2) Accept the randomly-placed point only if it:
   i) falls within the "clipping" (now really "sampling") region, and
   ii) falls in a pair of bins both having nonzero counts

3) If the point has been accepted, decrement the counts in the corresponding bins by one

Use of the bin counts offers the advantage of minimal statistical noise in the generated distribution, since (assuming success of the algorithm) the correct number of particles (within ±1) is loaded into each "cell" in both (x,x′) and (y,y′). However, the procedure is computationally slow since most trial points fail to be accepted, especially toward the end of the process.

In addition, it is possible for the algorithm to fail to successfully place the desired number of particles because of a pitfall, which we refer to as "painting itself into a corner." This can arise when a number of points near the center of the distribution in both (x,x′) and (y,y′) have been loaded, and so the counts in those regions have become depleted. Then, when the algorithm attempts to decrement a bin near the periphery in (x,x′), it cannot pair that bin with one near the periphery in (y,y′) because that combination would not satisfy the clipping constraint. If no bins near the center of the distribution in (y,y′) remain, then the algorithm cannot decrement all bin counts to zero. In our numerical experiments we have observed this to be a minor effect, and only a small fraction of the bin counts are not fully depleted. The error introduced by this effect has in general been smaller than the fluctuation-induced error of method G2, described below.

If it is necessary to improve on this, an iterative procedure might be devised wherein loaded particles near the middle of the distribution in 4-space might be excised and the corresponding bin counts incremented, so that the outliers near the periphery might be able to be placed. Alternatively, a non-random search procedure might be used, whereby bins near the periphery in (alternately) one plane and then the other would be loaded first.

Two-Plane Method using Corrected Probabilities ("G2")

The second "G" method is also a statistically-correct variant of the "S" algorithm, and is implemented via a straightforward modification of that method. It too is based on random sampling from a 4-box, but it does not sample with the input probabilities directly. Rather, it uses "corrected" acceptance probabilities which are obtained by dividing the input probabilities in (x,x′) and (y,y′) by the (normalized) area of the "slice" of the 4D sampling region in the other principal plane at that bin's location. This method yields results similar to those of method 1. It offers gains in speed and simplicity, and is immune to the "painting into a corner" problem, but suffers from larger fluctuations in the number of particles in each 4-bin.

Method G2:

Define the two-volumes $V_{yy'}$ = the total number of data nodes in the (y,y′) plane, and $V_{xx'}$ = the total number of data nodes in the (x,x′) plane.

First consider the (x,x′) plane. For each data node (x_{ij},x'_{ij}), compute V_{ij} = the number of nodes (each indexed by kl) in the (y,y′) plane for which the node location $(x_{ij},x'_{ij},y_{kl},y'_{kl})$ falls inside the clipping region. Compute the modified likelihood via:

$$G(x_{ij},x'_{ij}) = f(x_{ij},x'_{ij})\ V_{yy'} / V_{ij}\ ,\ \text{if } V_{ij} > 0 \qquad (24)$$
$$= 0, \qquad \text{if } V_{ij} = 0\ .$$

The (y,y′) plane is handled similarly. For each data node (y_{kl},y'_{kl}), compute V_{kl} = the number of nodes (each indexed by ij) in the (x,x′) plane for which the node location $(x_{ij},x'_{ij},y_{kl},y'_{kl})$ falls inside the clipping region. Compute the modified likelihood via:

$$G(y_{kl},y'_{kl}) = f(y_{kl},y'_{kl})\ V_{xx'} / V_{kl}\ ,\ \text{if } V_{kl} > 0 \qquad (25)$$
$$= 0, \qquad \text{if } V_{kl} = 0\ .$$

It is desirable at this point to check whether any cells with nonzero $f(x_{ij},x'_{ij})$ have $V_{ij} = 0$ and have thus been excluded by the clipping process; and to check similarly in the (y,y′) plane. If the input distribution is to be matched by the output, this should be avoided by enlarging the clipping region.

The procedure then tracks that of the S method. Carry out the following steps repeatedly in a loop, until the requisite number of particles have been generated:

1) Generate a random point in the 4-box bounded by the extremes of the measured data, and two random numbers, ρ_1 and ρ_2, on the unit interval

2) With likelihood proportional to $G(x,x') \times G(y,y')$, tentatively accept the point; that is, keep it <u>if</u> $G(x,x') > \rho_1$ <u>and</u> $G(y,y') > \rho_2$

3) Accept point as a particle in the synthesized distribution if and only if its coordinates (x,x′,y,y′) lie inside the clipping region, as set by the parameters CLIPALG, CLIP, and CLIPELLIPSOID

It is possible to imagine improvements to this procedure, *e.g.*, carrying out the integrals which yield V_{ij} and V_{ij} more precisely, using not just the node locations (which may be thought of as cell centers) but rather the actual fractions of the 4-cell volumes inside the clipping region. Use of a finer grid of nodes would have a similar effect. However, the main limitations appear to be associated with the somewhat arbitrary clipping volume.

Three-Plane Method using Bin Counts ("G7")

This method is closely related to the two-plane version G1, but takes advantage of the extra information available when a spatial density scan has yielded $f(x,y)$.

Method G7:

Assign target "counts" $N(x,x')$ and $N(y,y')$ to the bins as described above. Also assign $N(x,y)$ by a similar procedure.

Carry out the following steps repeatedly in a loop, until the requisite number of particles have been generated and all "bin counts" have been decremented to zero, or until a user-specified maximum number of passes through the loop have been completed, whichever occurs first:

1) Generate a random point in the 4-box bounded by the extremes of the (thresholded and shifted) measured data

2) Accept the randomly-placed point only if it:
   i) falls within the "clipping" (now really "sampling") region, and
   ii) falls in a triad of bins all having nonzero counts

3) If the point has been accepted, decrement the counts in the corresponding bins by one

Two-Plane Method based on Anticorrelation ("G3")

Another class of statistically-correct algorithms is based on an assumed anticorrelation, of a particular kind, of the (x,x') and (y,y') densities. The assumed anticorrelation can actually be viewed as either a postulate or as the consequence of two other assumptions; here we follow the latter course. The first assumption is:

> *Assume* that the beam distribution was, at an earlier time, well described by a set of nested hyperellipsoidal contours each with uniform phase space density, and with the peak phase space density at the center of the distribution (the simplest case is a uniformly-filled 4-ellipse)

Then, because $f(x,x') = \int \int f(x,y,x',y')\, dy\, dy'$, bins near the "edge" in (x,x') with small $f(x,x')$ are that way largely because there is only a small region in (y,y') with nonzero density at that (x,x').

As an analogy, consider the unit 3-sphere; particles with the largest |x| all have small |y| and |z|. Thus there is only a small area in (y,z) that contributes at large |x|. See Fig. 3.

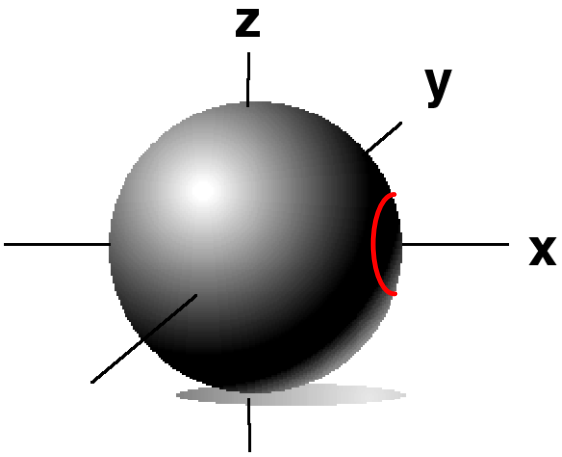

FIG. 3. Illustration of 3-D analogue of the anticorrelation described in the text.

By similar reasoning, in a uniformly-filled (upright) 4-ellipse, particles with small $f(x,x')$ have small |y| and |y′|, and a particular anticorrelation obtains, as is illustrated in Fig. 4.

Particles with <u>small</u> $f(x,x')$ must have <u>large</u> $f(y,y')$ but particles with <u>large</u> $f(x,x')$ may have "<u>any</u>" $f(y,y')$

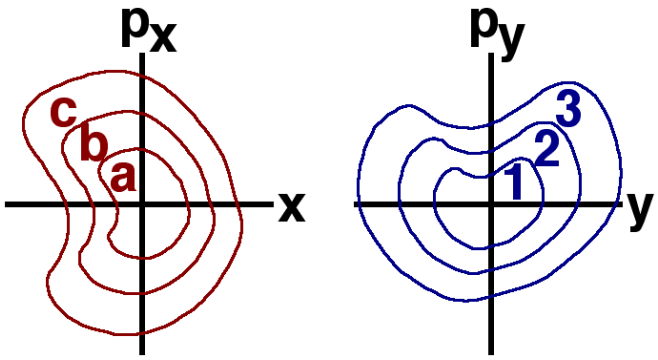

FIG. 4. Illustration of the anticorrelation described in the text. In the figure:
- Particles in region "c" (between contour levels) are all in region "1".
- Particles in region "b" are all in regions "1" or "2".
- Particles in region "a" can be in regions "1", "2", or "3".

The (x,x′) and (y,y′) density contours will have been distorted by the time they reach the measurement station. Thus we add another assumption:

> <u>*Assume*</u> that the distortion mechanisms involved no coupling between motions in the (x,x′) and (y,y′) planes.

Then, Liouville's theorem says that the 2D f's remain constant along orbits, and the anticorrelation described above persists.

Method G3:

Assign target "counts" N(x,x′) and N(y,y′) to the bins as described above for Method 1.

Carry out the following steps repeatedly in a loop, until the requisite number of particles have been generated and all "bin counts" have been decremented to zero:

1) Load a particle into a random location within the (x,x′) bin with the smallest nonzero N(x,x′) and within the (y,y′) bin with the largest N(y,y′). (If multiple bins have the same counts, choose randomly.)

2) Decrement the counts N(x,x′) and N(y,y′) in the appropriate bins.

Toward the end of the process, the distribution of counts in (y,y′) becomes flat. This algorithm is asymmetric with respect to the roles of the input (x,x′) and (y,y′) distributions, but the result appears to be invariant to an interchange of those roles.

IV. MODEL PROBLEM

This idealized "mock" problem was employed as a means of developing our understanding of the methods and their performance. We consider a "waterbag" distribution with uniform phase space density in the distorted 4-sphere:

$$x^2 + y^2 + (x' - \mu x^3)^2 + y'^2 < 1 \ , \tag{26}$$

where μ is usually set to 1.75. This distortion does not change the occupied volume.

In Fig. 5 three views of the (x,x′) and (y,y′) planes of the "true" particle distribution are shown. These planes are used to generate input to the various methods. The views include "scatter plots" of the particle locations, color contour plots, and "cell array" (filled mesh) plots, obtained by nearest-grid-point weighting from the particles. For the most part we rely on scatter plots in this paper, but it is beneficial to view data (especially experimental data) using a variety of graphical representations and color tables.

In Fig. 6 all six principal Cartesian projections of the true input distribution are shown. The structure in the (x′,y′) and (y,x′) planes should be noted.

In Fig. 7 the result of naïve application of the product of the (x,x′) and (y,y′) probabilities is shown. Note the good fidelity in the (x,x′) and (y,y′) projections, and the spurious "squareness" in the other projections.

In Fig. 8 the results of using the S method, with clip = 1, are shown. The spurious "squareness" is improved by the application of clipping, but persists. The (x,x′) and (y,y′) projections are slightly inaccurate, as a result of the clipping; the tips of the former are very slightly thinned (they are less dense, and are slightly truncated), and the density distribution of the latter is slightly distorted. These effects are hard to see graphically but are evident when moments over the "true" and synthesized particle distributions are computed.

In Fig. 9 the results of using the G1 method with clipalg = 0 are shown; parameters clip and clipellipsoid were set so that the sampling volume was identical to that used by the S method. The "equators" of the sampling volume are shown in the plots of the (x,x′) and (y,y′) planes. Those input planes are accurately reproduced; the spurious "squareness" in the other planes such as (x′,y′) is reduced but is not eliminated.

In Fig. 10 the results of using the G1 method with clipalg = 1 are shown; the sampling volume is an ellipsoid which is tilted in the (x,x′) plane, as described earlier. The "equator" of the sampling volume is shown in the plot of the (x,x′) plane. Again, the input planes are accurately reproduced; the spurious "squareness" in the other planes is reduced (to a slightly greater degree than for CLIPALG = 0), but is not eliminated.

In Fig. 11 the results of using the G1 method with CLIPALG = 2 are shown; the sampling volume is a more tightly fitted tilted ellipsoid, as described earlier. The "equator" of the sampling volume is shown in the plot of the (x,x′) plane. Again, the input planes are accurately reproduced; the spurious "squareness" in the other planes is reduced.

In Fig. 12 the results of using the G1 method with CLIPALG = 3 are shown. The sampling volume matches the beam boundary. All planes are accurately reproduced, because the method is exact (within discretization errors) for a waterbag distribution when the boundaries are made to coincide.

In Fig. 13 the results of using the G3 method (which does not require specification of a sampling volume) are shown. All planes are accurately reproduced, because the method is exact (within discretization errors) for a waterbag distribution of this type.

From these results we see the desirability of wrapping the sampling volume boundary tightly around the beam. The use of more tightly-fitted ellipsoids should be advantageous in real problems, but it is difficult in general to synthesize a sampling volume that wraps tightly around a highly distorted beam, especially when one does not know the 4D shape of that beam, and when the beam has a gradual falloff of density rather than a sharp boundary. This motivates the search for parameter-free methods such as the G3 method, which works well on this model problem.

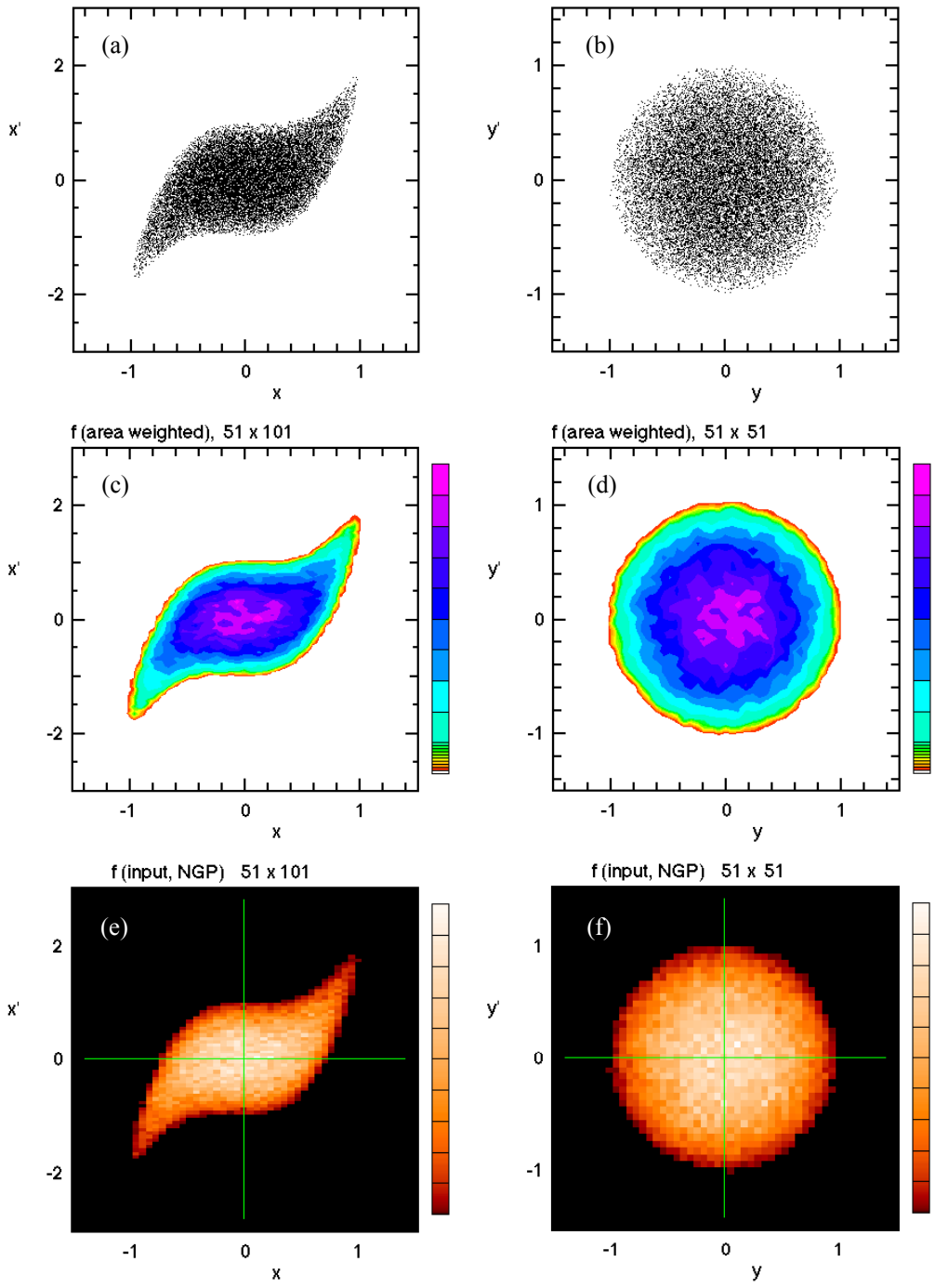

FIG. 5 (color). Projections of "true" particle distribution for "mock" problem. These planes are used to generate input to the various methods. Three representations are shown: in (a) and (b), "scatter plots" of the particle locations, in (c) and (d) color contour plots with 19 discrete contour levels (colors), arranged to provide additional detail at low density; and in (e) and (f) "cell array" (filled mesh) plots with more than 200 colors, obtained by nearest-grid-point weighting from the particles. The color bars at right of (c-f) show a linear scale from zero to peak of the data.

FIG. 6. Projections of “true” particle distribution for “mock” problem used for algorithm development. Planes at top of page are used to generate input to the various methods; these and the other planes are used to assess the success of the methods.

FIG. 7. Projections of “mock” problem particle distribution obtained by naïve application of the product of the (x,x′) and (y,y′) probabilities. Note the good fidelity in the (x,x′) and (y,y′) projections, and the spurious “squareness” in the other projections.

FIG. 8. Projections of “mock” problem particle distribution obtained by application of the S method. The spurious “squareness” is improved by the application of clipping, but persists. The (x,x′) and (y,y′) projections are slightly inaccurate.

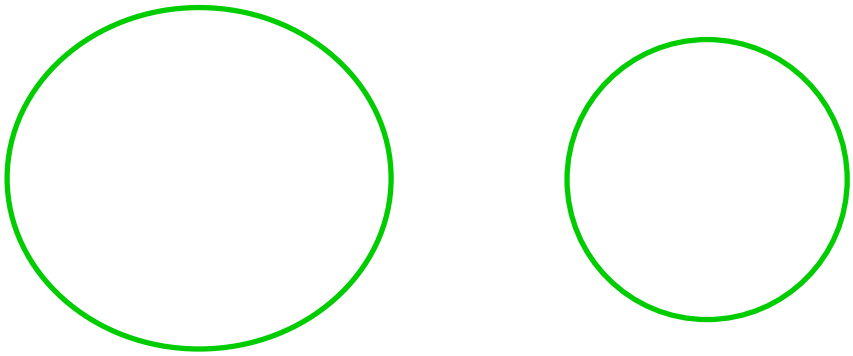

FIG. 9. Projections of “mock” problem particle distribution obtained by application of the G1 method and CLIPALG = 0. The sampling volume was the same as that used for the S method.

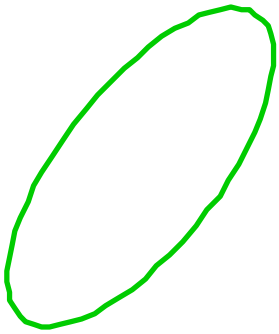

FIG. 10. Projections of “mock” problem particle distribution obtained by application of the G1 method and CLIPALG = 1. The sampling volume is a tilted ellipsoid.

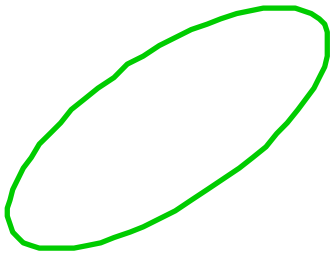

FIG. 11. Projections of “mock” problem particle distribution obtained by application of the G1 method and CLIPALG = 2. The sampling volume is a more tightly fitted tilted ellipsoid.

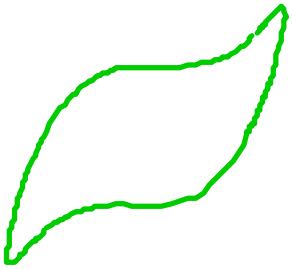

FIG. 12. Projections of “mock” problem particle distribution obtained by application of the G1 method and CLIPALG = 3. The sampling volume matches the beam boundary, and the method gives the “exact” answer for the mock problem, within discretization errors.

FIG. 13. Projections of “mock” problem particle distribution obtained by application of the G3 method. The method gives the “exact” answer for the mock problem, within discretization errors.

V. INJECTION INTO RFQ PROBLEM BASED ON EXPERIMENTAL DATA

We now consider application of the methods to a first "real" problem: a 65-keV H^- source and a two-lens electrostatic Low Energy Beam Transport system feeding a radio-frequency quadrupole accelerator. The layout is depicted in Fig. 14.

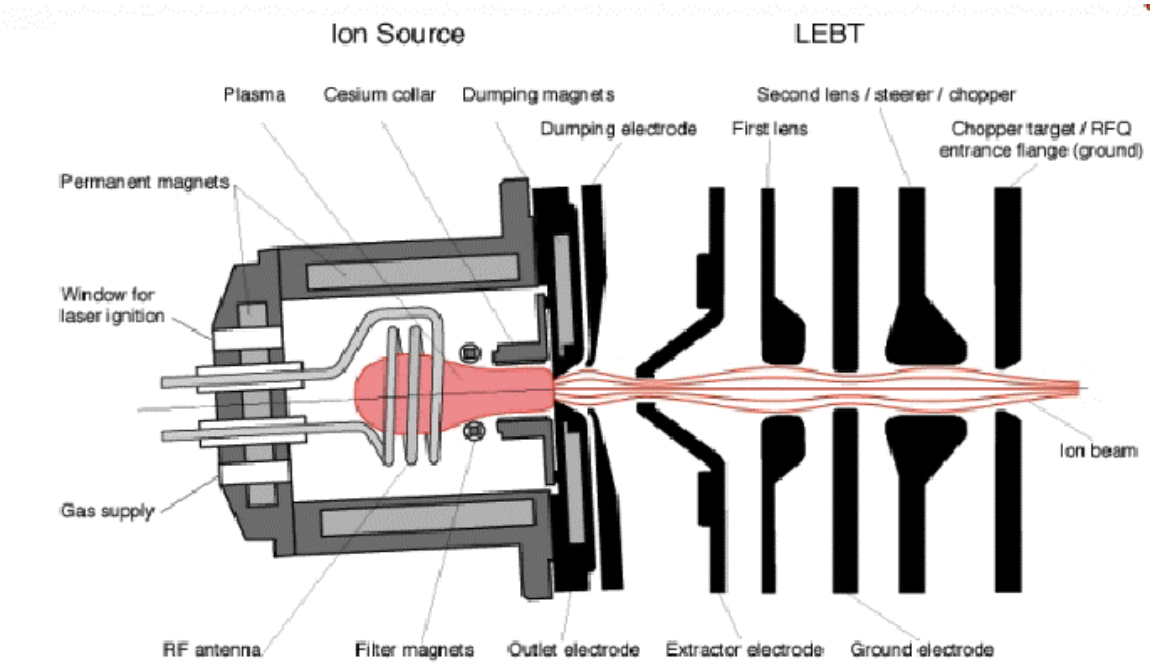

FIG. 14 (color). Schematic of ion source and Low Energy Beam Transport (LEBT) system for injector feeding an RFQ accelerator. The indicated beam particle trajectories are merely suggestive, and exaggerate the degree to which the aperture is filled.

We carried out a scan of both lens voltages, and took two-plane phase-space data for each voltage pair. The scanner was of the Allison type, that is, it used a variable deflection voltage to sweep the beam across the second slit, rather than moving that slit. We then used the S method to synthesize particle distributions, and simulated beam transmission through the RFQ for each data set. We found that the RFQ transmission fraction peaked at exactly the predicted value of lens voltages, within experimental error. The results had greater fidelity to the experiments than those obtained using idealized initial distributions, *e.g.*, a simple 4D waterbag.

An example of the input data is shown in Fig. 15. The beam is distorted due to the aberrations induced by the various beamline elements. The data exhibit some noise, and some points are missing; the algorithms must deal with such effects gracefully. Thresholding was performed by subtracting 0.06 from the data (previously normalized to range from 0 to 1), then dividing by (1-0.06) = 0.94 to renormalize the data. This resulted in retention of 69.4% of the total count in (x,x′) and of 57.8% in (y,y′). A compact, if distorted, beam is the result. No removal of the mean (x,x′) tilt was performed.

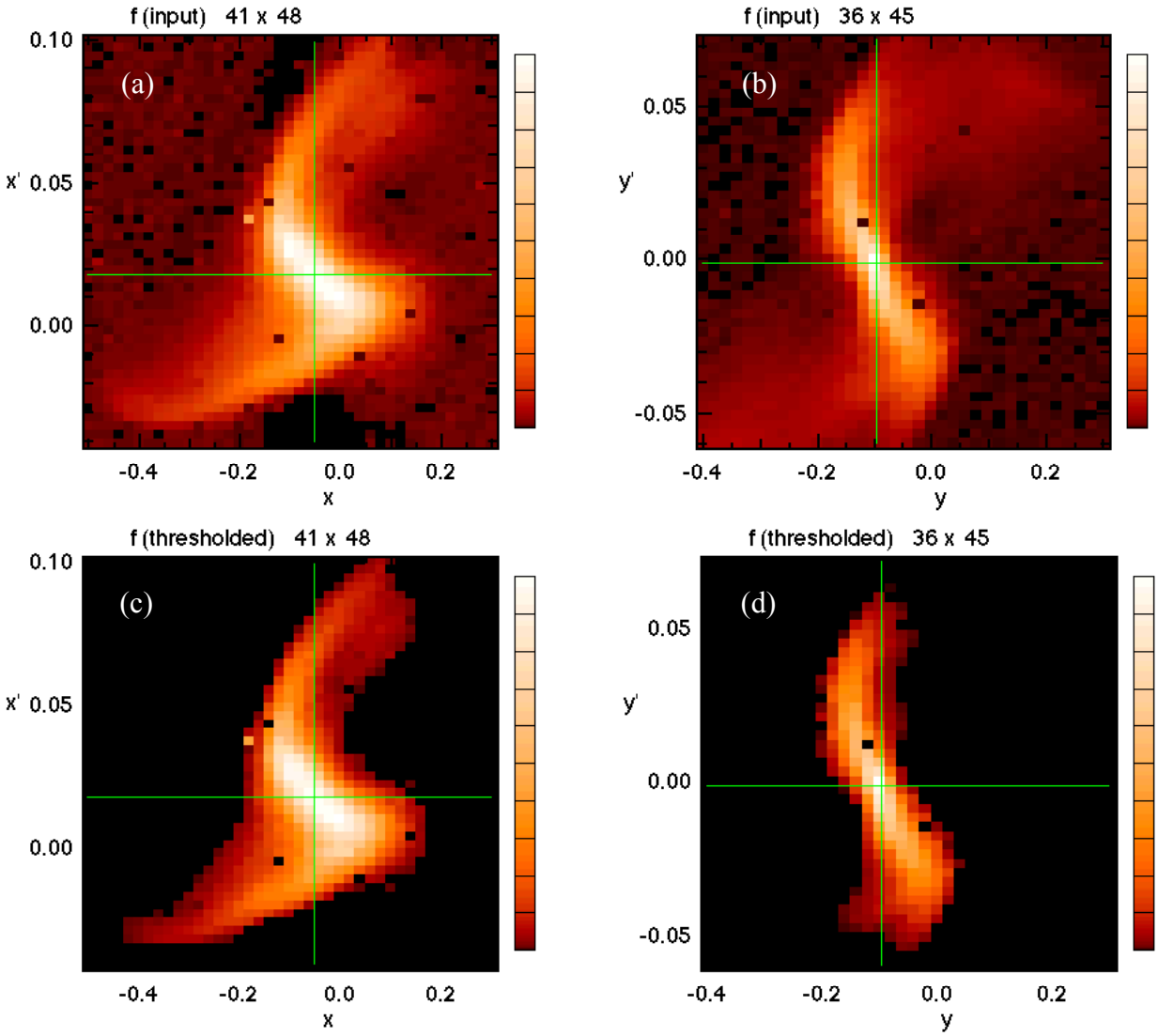

FIG. 15 (color). Raw input data in (a) plane (x,x′)and (b) plane (y,y′), and thresholded input data in (c) plane (x,x′)and (d) plane (y,y′). Color denotes intensity; lines across the color-bars to the right of the images denote 0, 10%, …, 100% of peak intensity.

The results of applying various methods to the thresholded data, synthesizing a set of 50,000 particles, are shown in Figs. 16-23. Figures 16 and 17 show cell-array plots obtained by nearest-grid-point mapping of the synthesized particle set into bins in (x,x′) and (y,y′). Figure 18 shows cell-array plots of the synthesized (x,y) projection for all of the methods tested. Figures 19-23 show "scatter" plots of 5000 of the particles projected onto the six principal Cartesian axes.

In Figs. 16 and 19 the S method was employed. The extremes of the distribution have been slightly truncated and the colors do not match those of the input images exactly;

both of these artifacts are results of the clipping. The (x,y) projection (in Figs. 18 and 19) has a simple shape, with some noise evident at its left edge.

In Fig. 20 the G1 method with CLIPALG = 0 was employed. We do not show the cell-array plots of the (x,x′) and (y,y′) projections because the method is sufficiently "exact" that the plots are indistinguishable from those in Fig. 15. In this case the phenomenon of "painting into a corner" was not an issue; all but one of the desired 50,000 particles was successfully loaded. The (x,y) projection (in Figs. 18 and 20) resembles that of the S method but has smaller cell-to-cell fluctuations.

In Figs. 17 and 21 the G2 method with CLIPALG = 0 was employed. The results resemble those of the G1 method but are noisier; detailed examination of the moments of the synthesized distribution shows that the reproduction of the input data also suffers somewhat because the correction of the probabilities is imperfect, due to the crudeness of the integrals performed using just the node centers. The (x,y) projection (in Figs. 18 and 21) differs in detail from the two immediately above, but in general all three methods using the "S-like" clipping region yield similar results, differing primarily in their noise level and their degree of fidelity to the input data.

In Figure 22 the G2 method with the tilted-ellipsoid CLIPALG = 1, CLIPELLIPSOID = 1.3, CENTER = "mid", was employed. We do not show the cell-array plots of the (x,x′) and (y,y′) projections because they closely resemble those of Fig. 17. Again some additional "noise" associated with the modest number of particles in each bin is evident. This time, qualitative differences are evident in the (x,y) projection (in Figs. 18 and 22), which now exhibits a broader "tail" at left, associated with the shape of the clipping region.

In Fig. 23 the G3 method which requires no user-settable parameters describing a sampling region was employed. Here again we do not show the cell-array plots of the (x,x′) and (y,y′) projections because the method is sufficiently "exact" that the plots are indistinguishable from those in Fig. 15. The (x,y) projection (in Figs. 18 and 23), and indeed the other non-input projections (in Fig. 23) are qualitatively different from those of the other methods, in that the method tends to pull out narrow "tails" as necessary to satisfy the need for consistency with the input data. This may seem a natural approach, since aberrations in applied and self-fields often preferentially affect the particles at the outside of the distribution. A proper assessment requires that we either know the "correct" answer (which we indeed do know for simulated data, as in the following section), or can discern greater fidelity to experiment in simulations of the downstream system using one synthesis method or another.

Table I shows the moments of the input distribution and of the various synthesized distributions. Note the near-exact agreement with the input data for the bin-based methods G1 and G3.

	<x>	<x´>	<y>	<y´>
true	-0.0512702	0.0174406	-0.0956311	-0.00152390
S	-0.0511420	0.0179291	-0.0950784	-0.00160050
G1 clipalg=0	-0.0512550	0.0174326	-0.0956295	-0.00152277
G2 clipalg=0	-0.0477220	0.0154727	-0.0960640	-0.00148848
G2 clipalg=1	-0.0415696	0.0159627	-0.0964068	-0.00130404
G3	-0.0512580	0.0174318	-0.0956304	-0.00152238

	<xx´>	<yy´>	<xy>	<x´y´>
true	-0.000718996	-0.000659834	--	--
S	-0.000883782	-0.000617363	0.00485649	-2.74791e-05
G1 clipalg=0	-0.000718136	-0.000659876	0.00489727	-2.24467e-05
G2 clipalg=0	-0.000294878	-0.000760568	0.00458730	-2.37916e-05
G2 clipalg=1	-0.000512324	-0.000728030	0.00403814	-1.97213e-05
G3	-0.000718042	-0.000659914	0.00493310	-2.83877e-05

	x rms	x´ rms	y rms	y´ rms
true	0.0763887	0.0238792	0.0464589	0.0232978
S	0.0716974	0.0229140	0.0452524	0.0226395
G1 clipalg=0	0.0764222	0.0238761	0.0464525	0.0232985
G2 clipalg=0	0.0854201	0.0256229	0.0491085	0.0247265
G2 clipalg=1	0.0824613	0.0238921	0.0480306	0.0237619
G3	0.0764243	0.0238765	0.0464525	0.0232984

	xx´rms	yy´rms	xy rms	x´y´rms
true	0.00251650	0.00232828	--	--
S	0.00233499	0.00226495	0.00780659	0.000654713
G1 clipalg=0	0.00251687	0.00232825	0.00833170	0.000747419
G2 clipalg=0	0.00273412	0.00244101	0.00923673	0.000717199
G2 clipalg=1	0.00253258	0.00237507	0.00909147	0.000689639
G3	0.00251694	0.00232824	0.00833995	0.000546162

rms emittances:	(x,x´)	(y,y´)
true	0.00181567	0.00072293
S	0.00164254	0.00067631
G1 clipalg=0	0.00181622	0.00072283
G2 clipalg=0	0.00214330	0.00081121
G2 clipalg=1	0.00196436	0.00075741
G3	0.00181629	0.00072283

Table I. Moments of the RFQ injection problem input data and of the various synthesized distributions; for the latter, moments were computed by depositing particle data onto the grid using nearest-grid-point weighting.

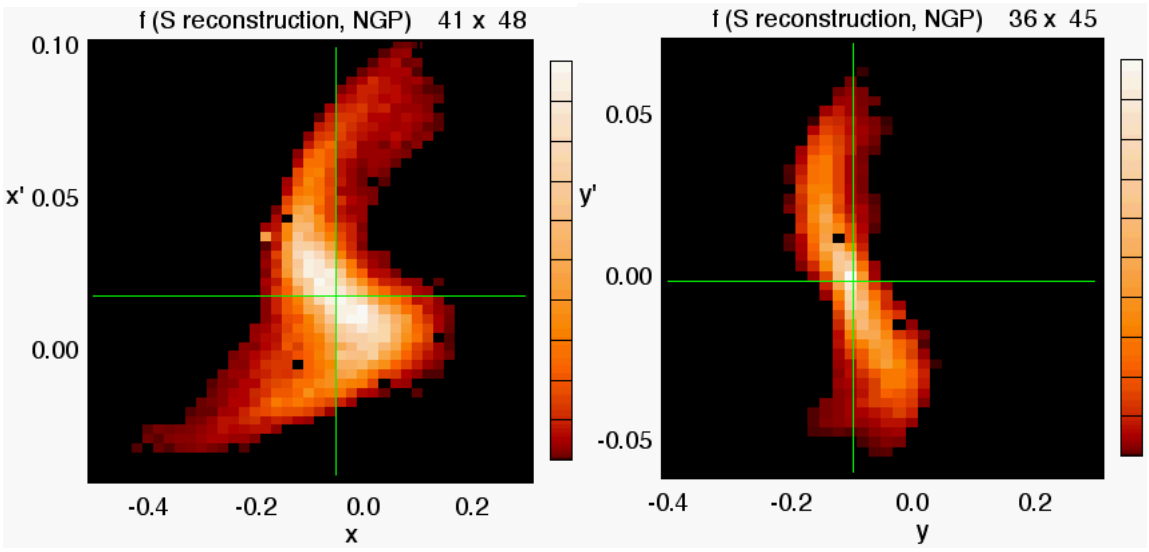

FIG. 16 (color). Cell-array plots obtained by applying S method with CLIP = 1.1, CENTER = "mid", and nearest-grid-point mapping of the set of 50,000 synthesized particles into bins; planes as noted in figure.

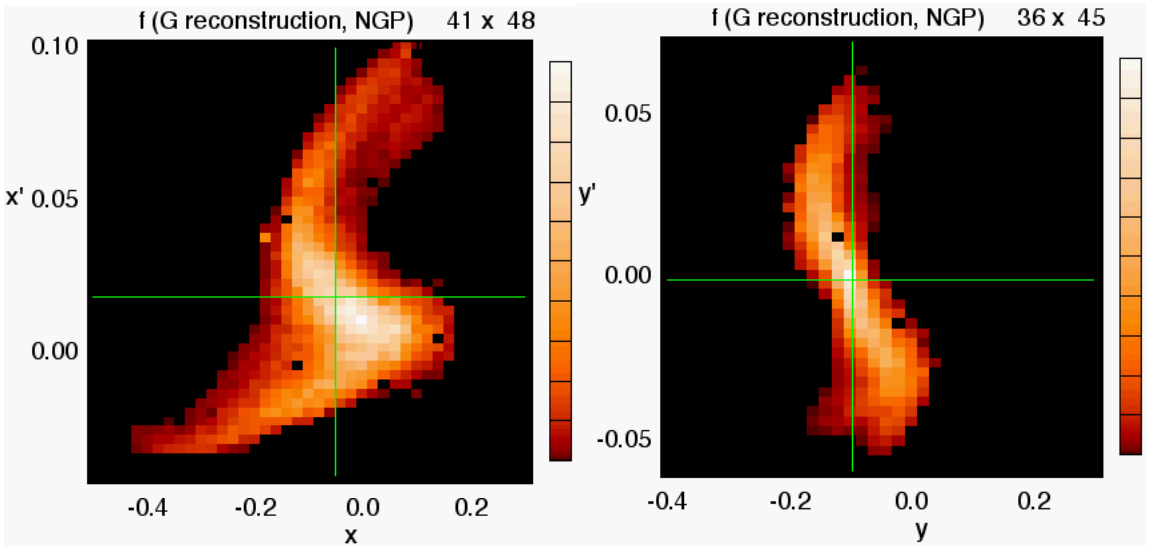

FIG. 17 (color). Cell-array plots obtained by applying G2 method with CLIPALG = 0, CLIP = 1.1, CLIPELLIPSOID = 1.54, CENTER = "mid", and nearest-grid-point mapping of the set of 50,000 synthesized particles into bins; planes as noted in figure.

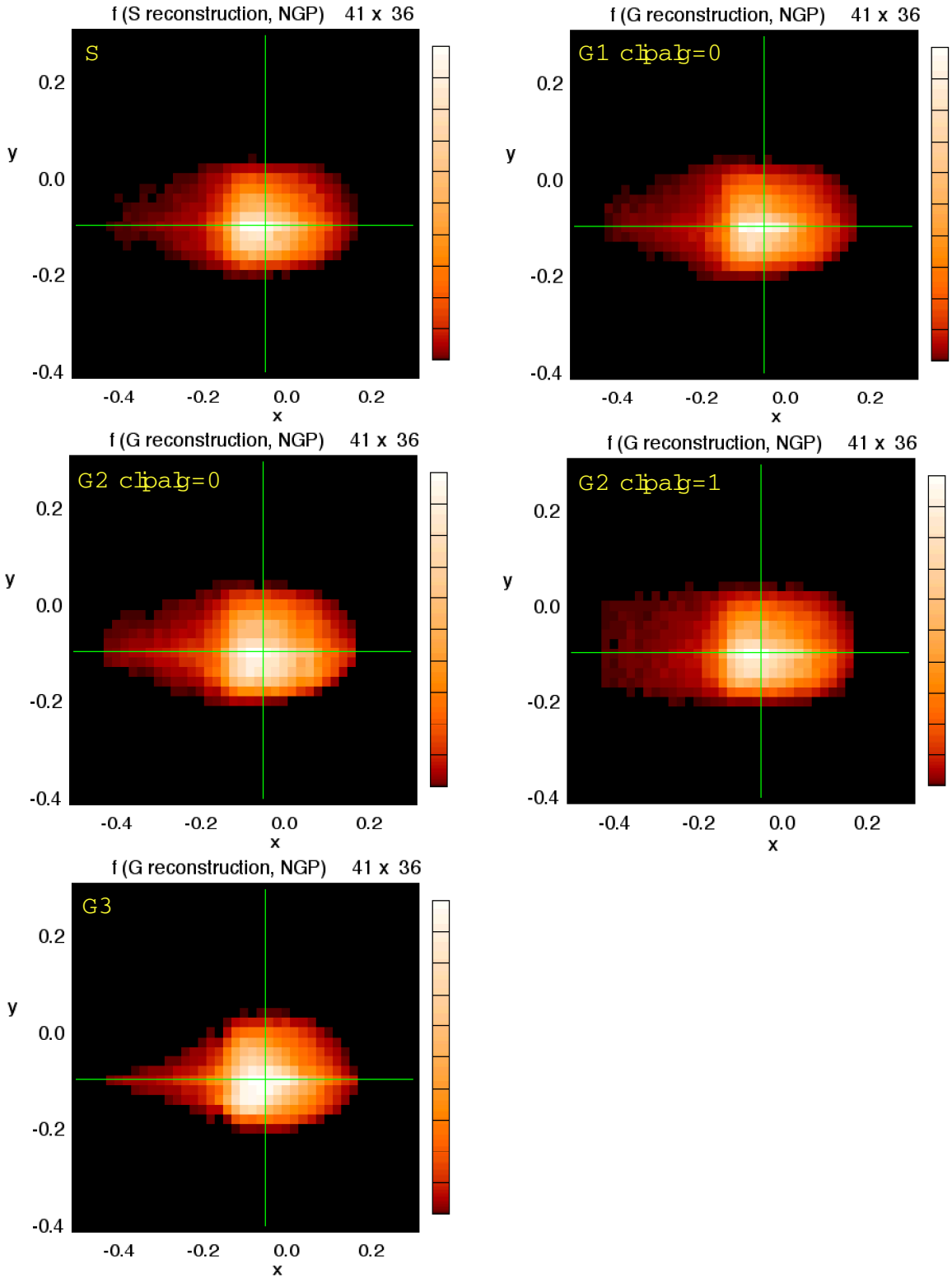

FIG. 18 (color). Cell-array plots of (x,y) projections derived using five methods as noted in figure, obtained by nearest-grid-point binning of 50,000 synthesized particles.

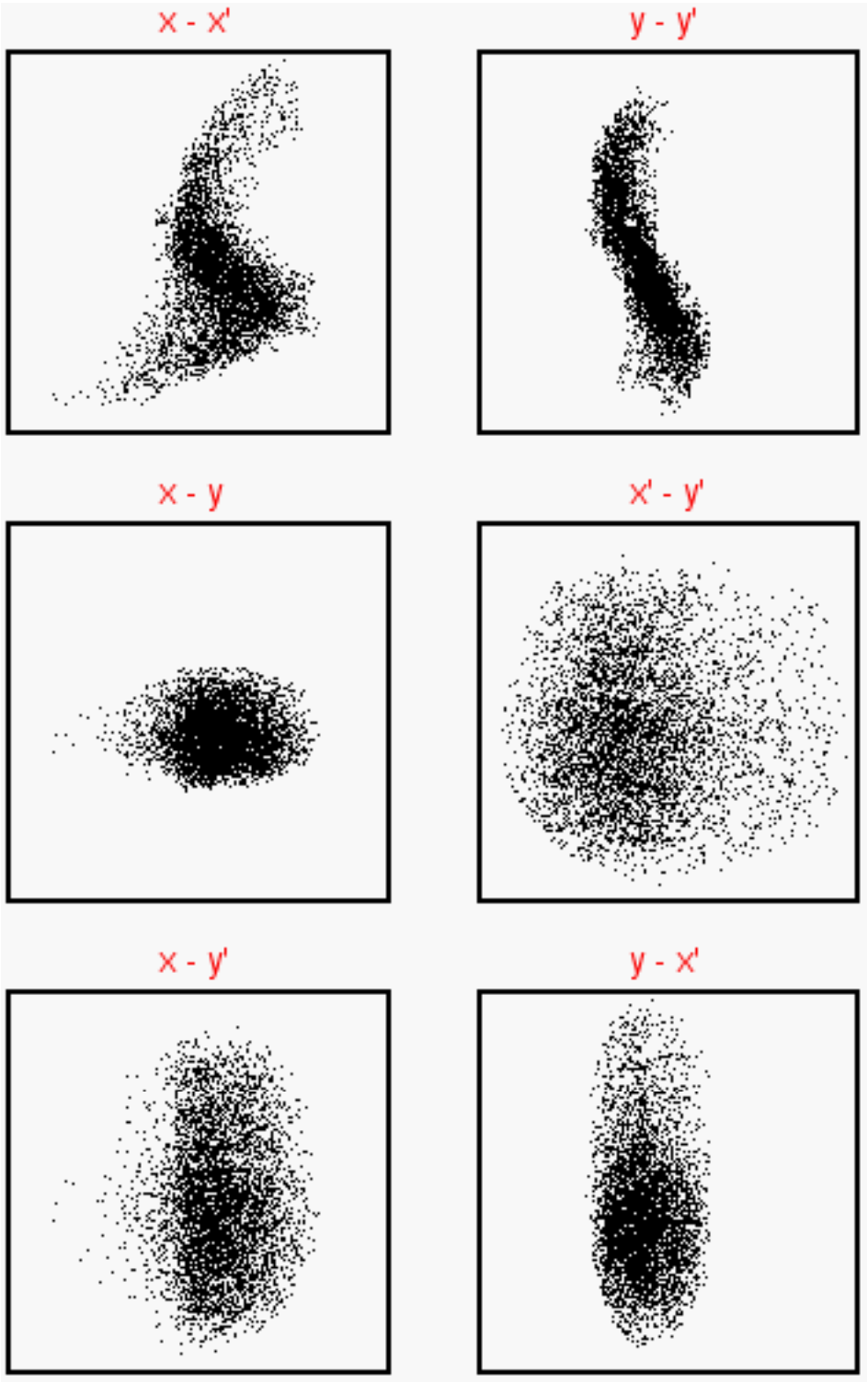

FIG. 19. Particle "scatter" plots obtained by applying S method with CLIP = 1.1, CENTER = "mid", planes as noted in figure.

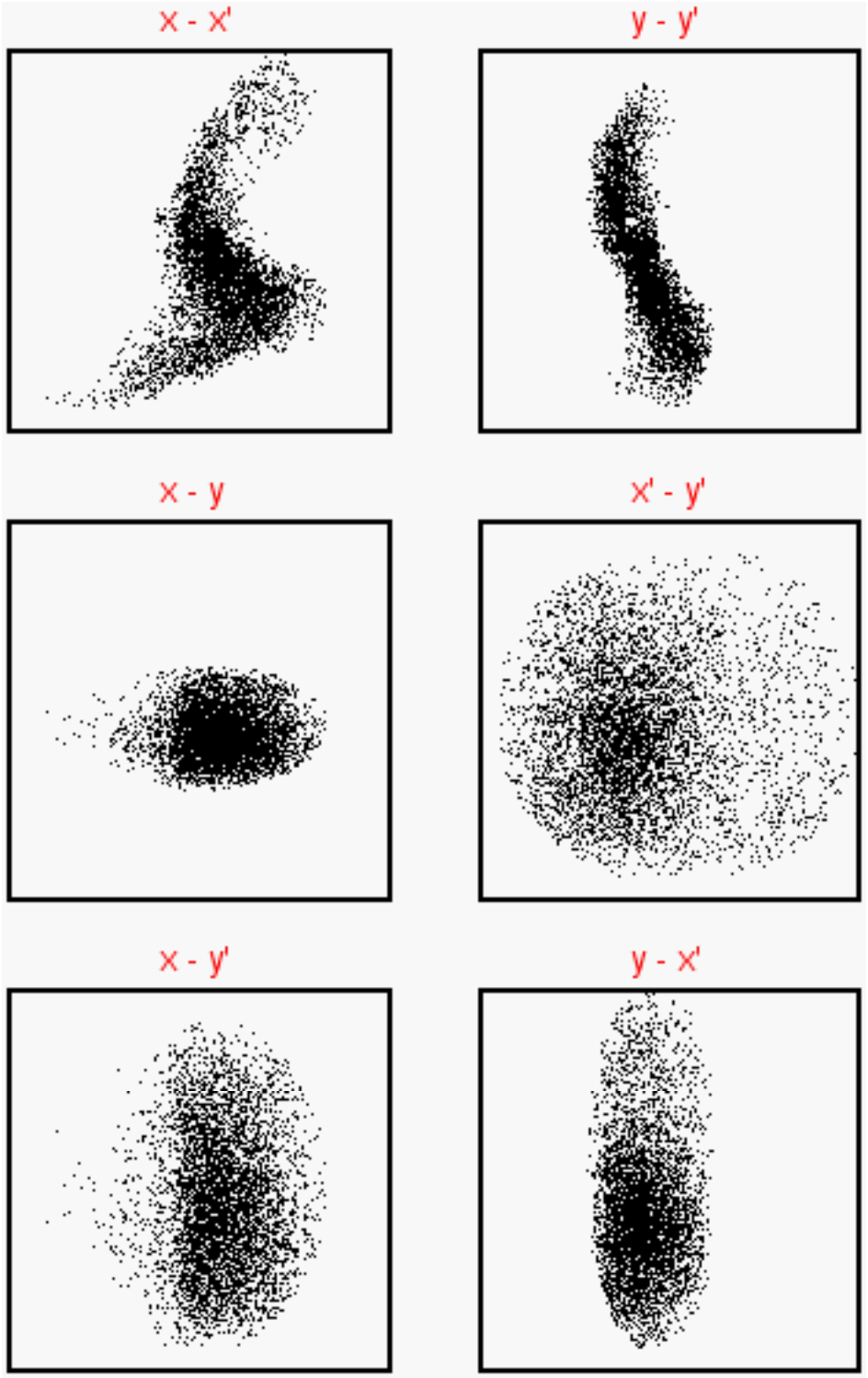

FIG. 20. Particle "scatter" plots obtained by applying G1 method with CLIPALG = 0, CLIP = 1.1, CLIPELLIPSOID = 1.54, CENTER = "mid"; planes as noted in figure.

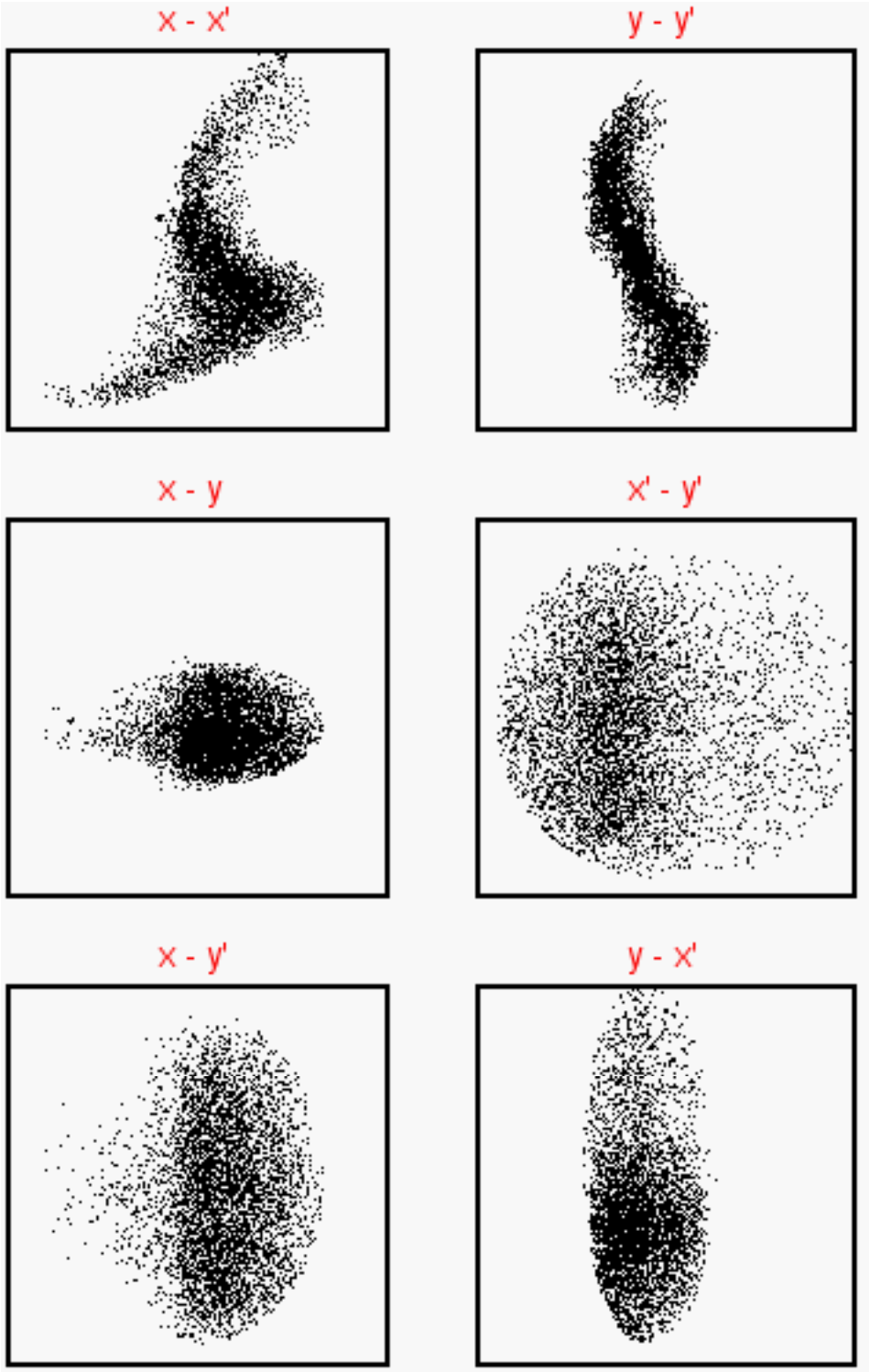

FIG. 21. Particle "scatter" plots obtained by applying G2 method with CLIPALG = 0, CLIP = 1.1, CLIPELLIPSOID = 1.54, CENTER = "mid"; planes as noted in figure.

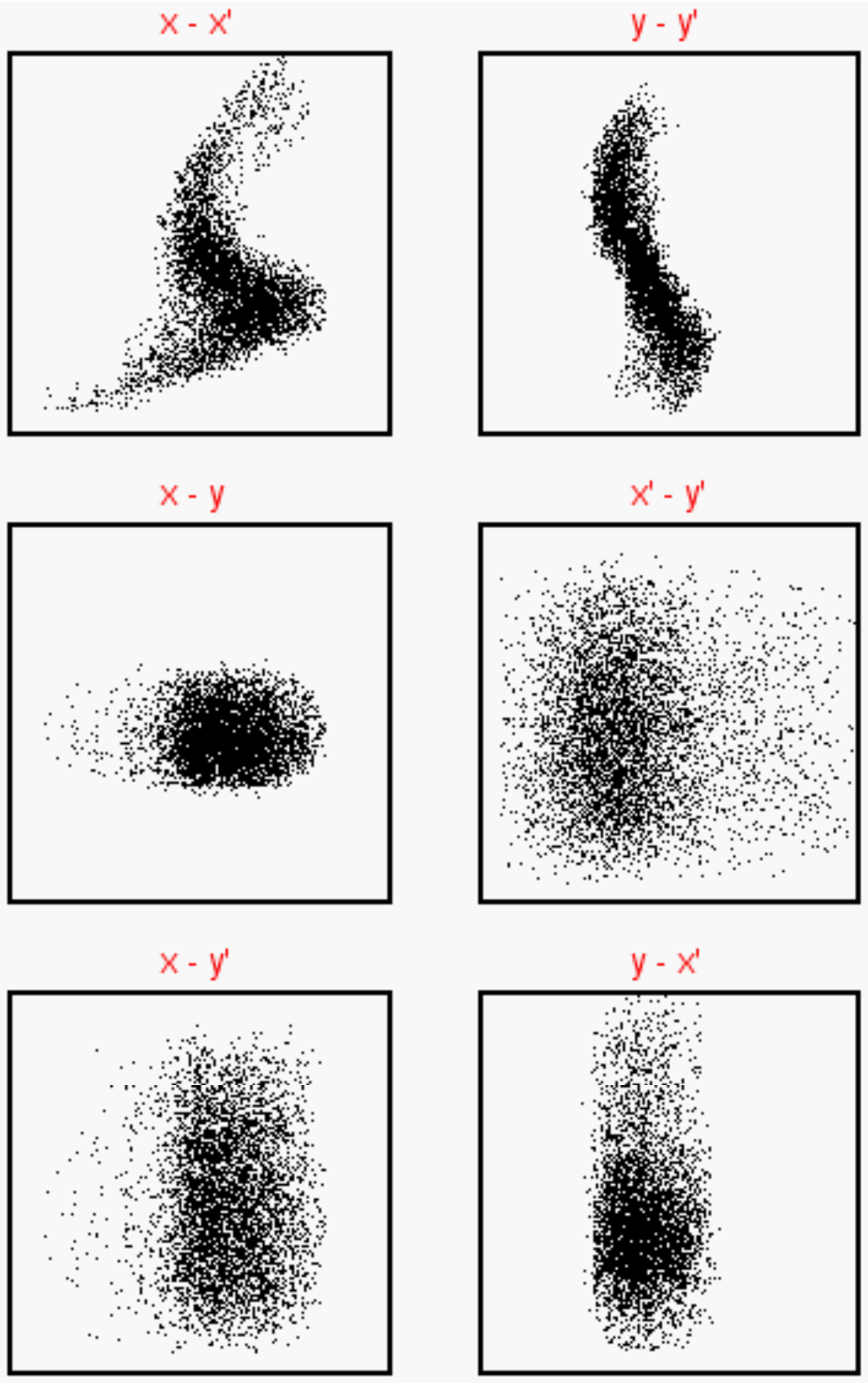

FIG. 22. Particle "scatter" plots obtained by applying G2 method with CLIPALG = 1, CLIPELLIPSOID = 1.3, CENTER = "mid"; planes as noted in figure.

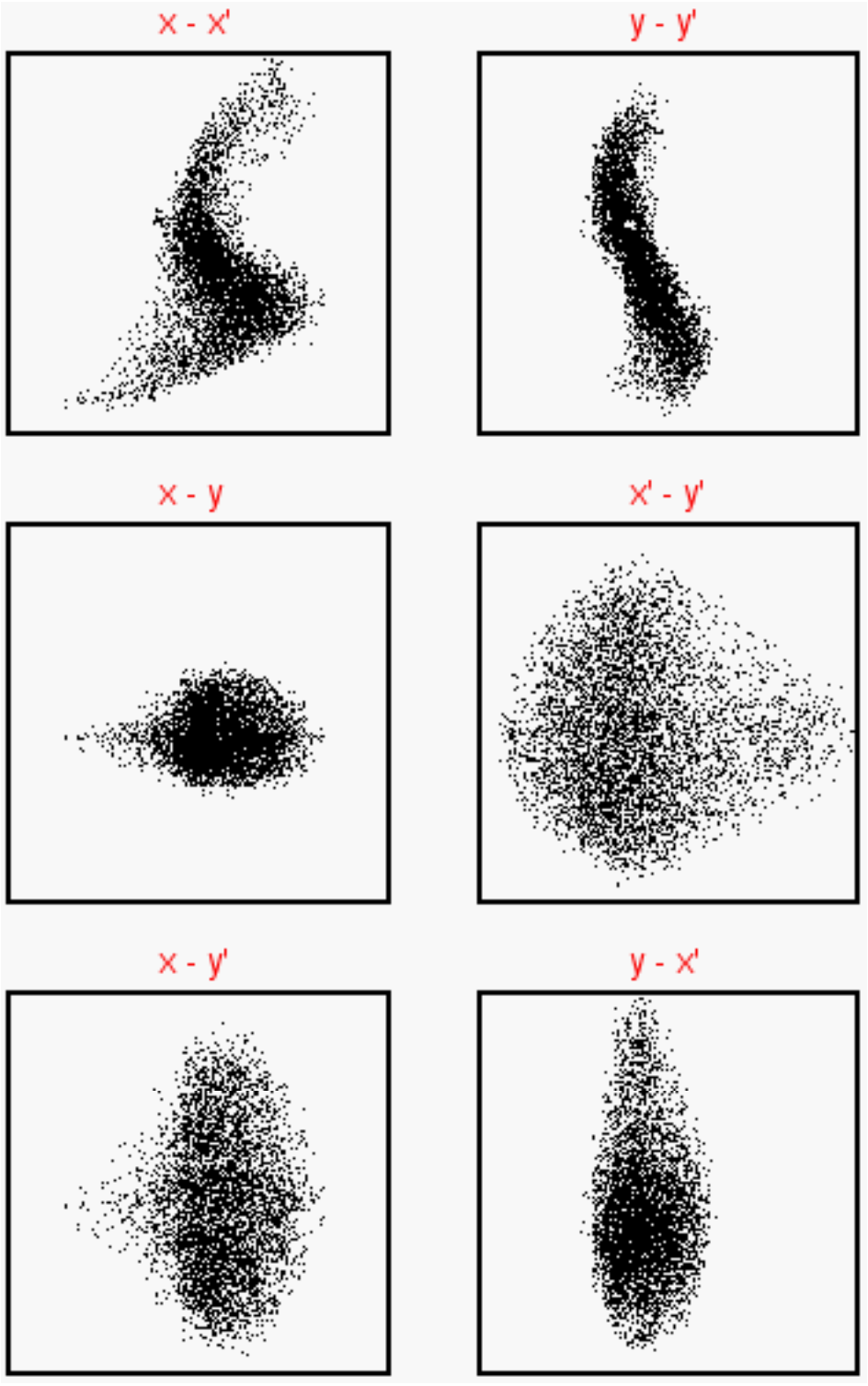

FIG. 23. Particle “scatter” plots obtained by applying G3 method; planes as noted in figure.

VI. HIF INJECTOR PROBLEM BASED ON SIMULATION DATA

In order to better assess the performance of the algorithms under development, it seemed important to study a realistic example where the "correct" answer was known. No data set describing a full 4D distribution is available from experiments. We thus used the output of a 3D WARP simulation of the electrostatic quadrupole (ESQ) injector [6] being used at LBNL as the front end of the High Current Experiment (HCX) [7] being conducted in the HIF research program. The simulated beam is depicted in Fig. 24, which also shows (from left to right) the high-voltage stalk surrounded by voltage-grading rings; the triode (gun) region consisting of the emitting surface (anode), switching electrode (gate), and extraction electrode (cathode); and the set of electrostatic quadrupole lenses which both confine and further accelerate the beam by means of an overall gradient superposed on bipolar quadrupole rod voltages.

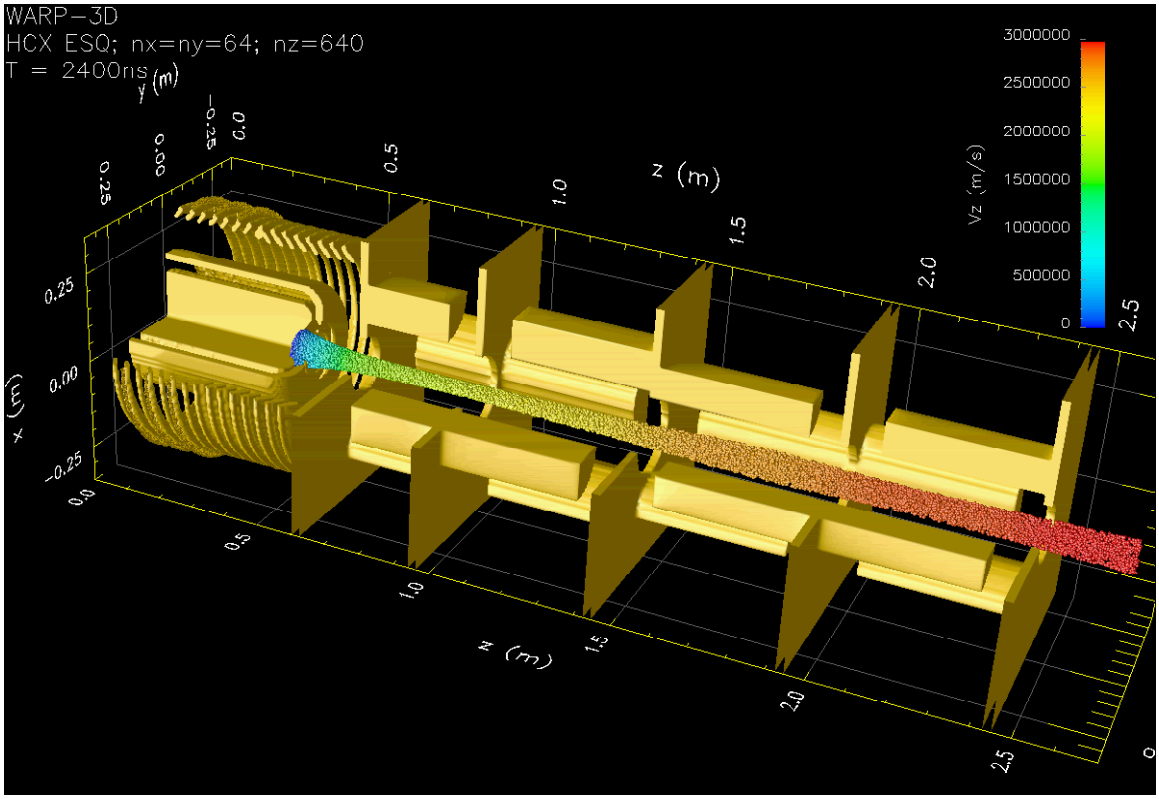

FIG. 24. 3D simulation of ESQ injector using WARP code; the simulated beam at the exit plane is the test problem described in this section.

The WARP3d simulations were carried out in a "steady-state" operating mode, emitting a steady stream of particles from the source and iterating to convergence on the field solution, in an effort to obtain a high-fidelity representation of the mid-pulse of the beam where its properties may reasonably be considered time-invariant. (Other simulations use the code's time-dependent mode to capture the behavior at the beam head and tail.) Thus, it was possible to obtain a reasonably large data set by accumulating the particle data

over many time steps at the same z, once the field solution had converged. The longitudinal coordinates of the simulation particles were examined after each time step, and those particles "landing" within a prescribed small distance of the exit plane of the injector were selected. The transverse coordinates and momenta of those particles were corrected, using an extrapolation over the distance from their individual longitudinal positions to the exit plane. By this means, a set of 102288 particles was obtained. In the syntheses shown, 51 grid points were used along each of the four principal Cartesian axes to define the "true" phase-space density. Table II and Figs. 25-31 summarize the results.

In Fig. 25, projections of the "true" distribution as supplied by WARP3d are shown; 10,000 particles are shown in each frame. The structure is complicated because of aberrations induced by non-paraxial flow in the triode "gun," and by the applied and self-fields in the ESQ accelerating and focusing structure. Note the "rim" in the (x,y) plane, associated with "translaminar" flow induced by the gun optics. By symmetry, the quantities in the first two rows of Table II are essentially zero. Note that, in contrast with the previous example, data is available on such quantities as the "true" RMS of (xy).

In Fig. 26, the results of applying the S method are shown. The method fails to reproduce moments such as $(xx')_{RMS}$ and the (x,x´) emittance faithfully (see the table). As with all methods which do not make explicit use of the (x,y) projection, the method does not synthesize the "rim" in (x,y) that is observed in the "true" data. Also, as with all the methods, some features of the other "unknown" projections are not correctly synthesized.

In Fig. 27 the results of applying the G1 method with CLIPALG = 0 and a sampling region identical to that of the S method are shown. Almost all, 102285, of the requested 102288 particles were generated. In contrast with the S method, a modest rim is suggested by the (x,y) projection and is reflected in a more accurate value of $(xy)_{RMS}$. All of the "known" moments (available from the (x,x´) and (y,y´) projections) are accurately reproduced.

In Fig. 28 the results of applying the G2 method with CLIPALG = 0 and a sampling region identical to that of the S method are shown. A partial rim in (x,y) is evident at only the upper, lower, left, and right edges of the beam; this is an artifact of the method's attempts to correct probabilities using crude integrals. The "known" moments are not accurately reproduced due to the errors in probabilities, and also due to enhanced fluctuations relative to the "bin" based methods.

In Fig. 29 the results of applying the G2 method with CLIPALG = 1 and CLIPELLIPSOID = 1.2 are shown. With this "large" value of CLIPELLIPSOID, the beam appears somewhat "square" in (x,y). However, even at this value some of the populated cells in both the (x,x´) and (y,y´) projections are clipped away; use of a smaller value to eliminate the "squareness" would exclude more of the populated cells.

In Fig. 30 the results of applying the G3 method are shown. Despite excellent reproduction of the "known" moments, the synthesized (x,y) projection is poor.

	<x>	<x´>	<y>	<y´>
true	-5.12358e-18	6.47141e-19	-3.39060e-18	2.77510e-19
S	-1.61739e-05	-1.25606e-06	6.29751e-05	7.71156e-07
G1 clipalg=0	-2.95645e-06	-5.00562e-08	4.92741e-06	-2.12739e-07
G2 clipalg=0	-2.41358e-05	-1.55795e-06	-0.000164664	-1.63929e-06
G2 clipalg=1	-3.18786e-05	-1.90834e-06	-0.000100141	-8.75958e-08
G3	-2.06476e-06	-9.22884e-08	3.98874e-06	-8.91600e-08
G7	6.26125e-08	-3.75675e-08	2.39493e-06	-2.09752e-07

	<xx´>	<yy´>	<xy>	<x´y´>
true	-2.82253e-18	4.99587e-18	4.11489e-07	1.74175e-09
S	1.66828e-06	7.40595e-07	9.79196e-07	3.83169e-10
G1 clipalg=0	6.37966e-09	2.76310e-09	-9.08270e-07	-1.07721e-09
G2 clipalg=0	-2.14564e-06	-9.82017e-07	-1.32745e-07	-1.24236e-09
G2 clipalg=1	3.53987e-08	1.14057e-07	6.33618e-07	-1.16628e-10
G3	2.43266e-09	-5.13061e-10	-1.00873e-06	4.22963e-11
G7	5.73280e-09	-3.38108e-10	2.91098e-07	-7.28810e-10

	x rms	x´ rms	y rms	y´ rms
true	0.0207051	0.000643990	0.0191242	0.000611029
S	0.0187002	0.000600425	0.0174473	0.000572439
G1 clipalg=0	0.0207041	0.000643666	0.0191242	0.000610622
G2 clipalg=0	0.0220313	0.000686674	0.0200806	0.000627863
G2 clipalg=1	0.0218298	0.000670973	0.0195460	0.000623362
G3	0.0207050	0.000643912	0.0191249	0.000610969
G7	0.0207023	0.000643859	0.0191234	0.000610889

	xx´rms	yy´rms	xy rms	x´y´rms
true	1.67396e-05	1.45560e-05	0.000332161	4.41434e-07
S	1.37048e-05	1.27497e-05	0.000279593	3.22351e-07
G1 clipalg=0	1.67235e-05	1.45482e-05	0.000317864	4.62113e-07
G2 clipalg=0	1.94509e-05	1.56537e-05	0.000324187	3.62190e-07
G2 clipalg=1	1.79111e-05	1.49995e-05	0.000355712	3.80647e-07
G3	1.67350e-05	1.45556e-05	0.000362058	2.22061e-07
G7	1.67270e-05	1.45548e-05	0.000332094	4.66602e-07

rms emittances:	(x,x´)	(y,y´)
true	1.33339e-05	1.16854e-05
S	1.11034e-05	9.95998e-06
G1 clipalg=0	1.33265e-05	1.16777e-05
G2 clipalg=0	1.49754e-05	1.25695e-05
G2 clipalg=1	1.46471e-05	1.21837e-05
G3	1.33322e-05	1.16847e-05
G7	1.33294e-05	1.16822e-05

Table II. Moments of the WARP-simulated ESQ problem input data and of the various synthesized distributions.

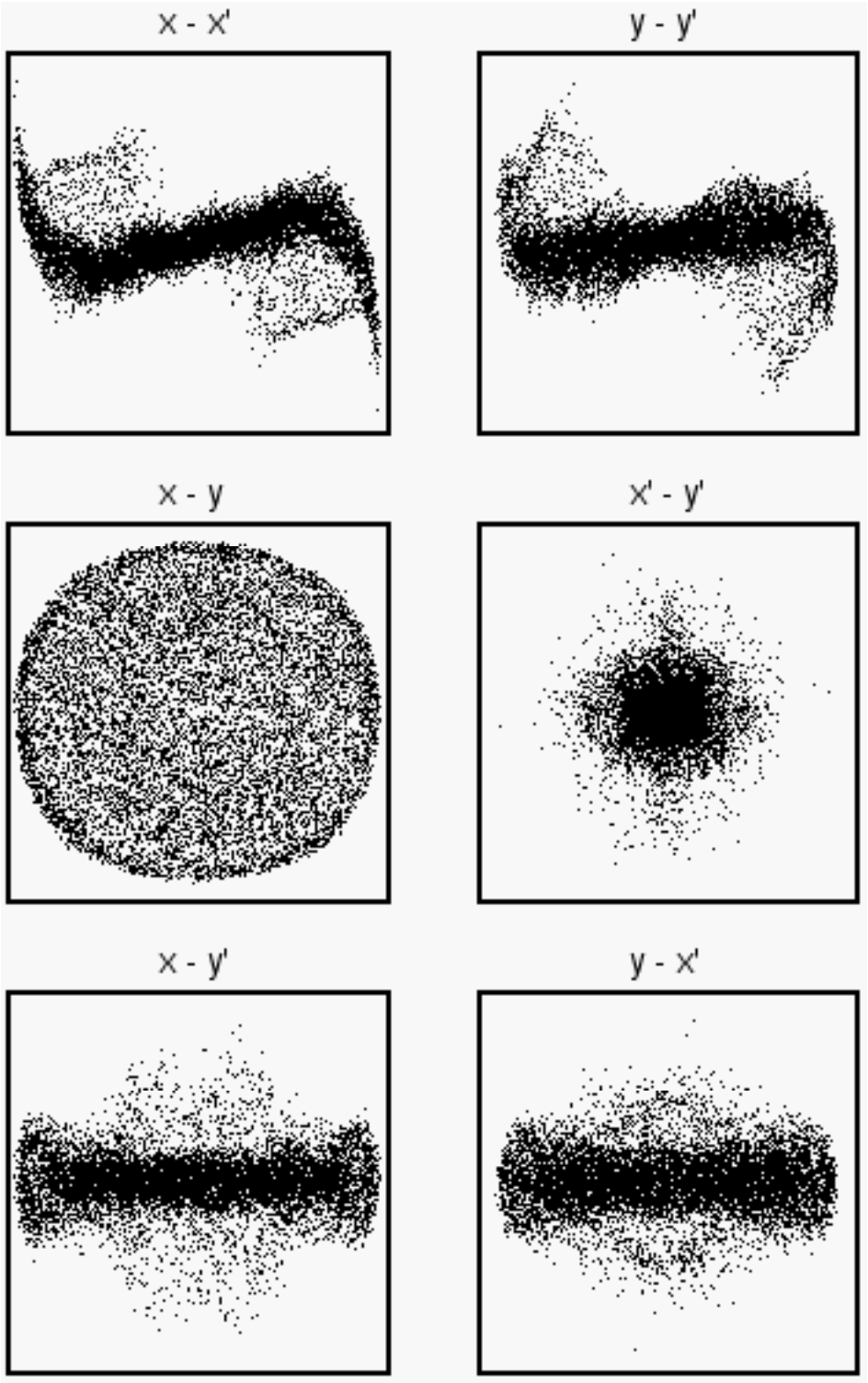

FIG. 25. Particle “scatter” plots showing “true” WARP-generated beam distribution in all six principal Cartesian planes, for comparison to synthesized distributions, as noted in figure. In each plot, 10,000 particles are shown.

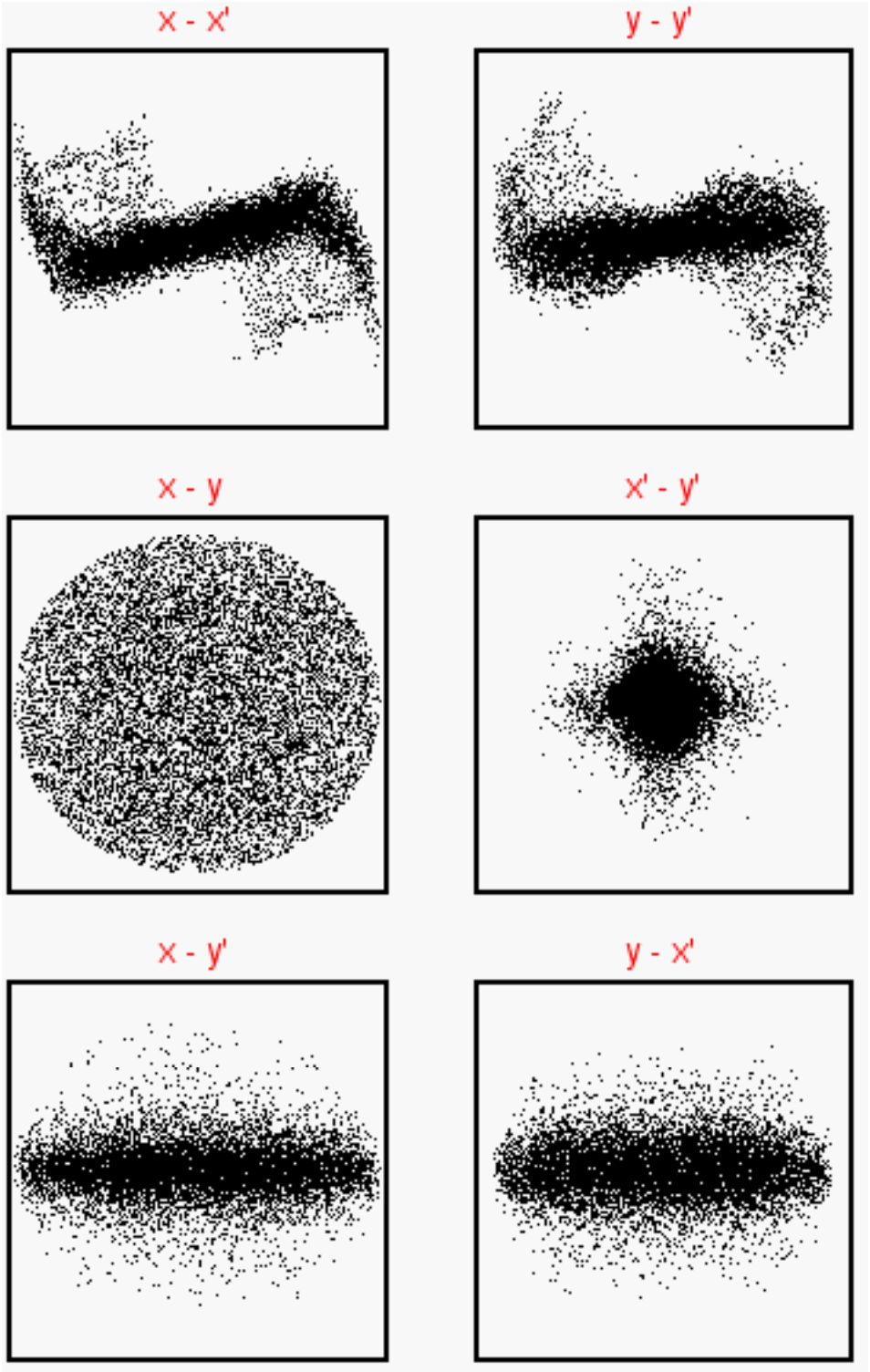

FIG. 26. Particle “scatter” plots of synthesized distributions using S method with CLIP=1.0; planes as noted in figure.

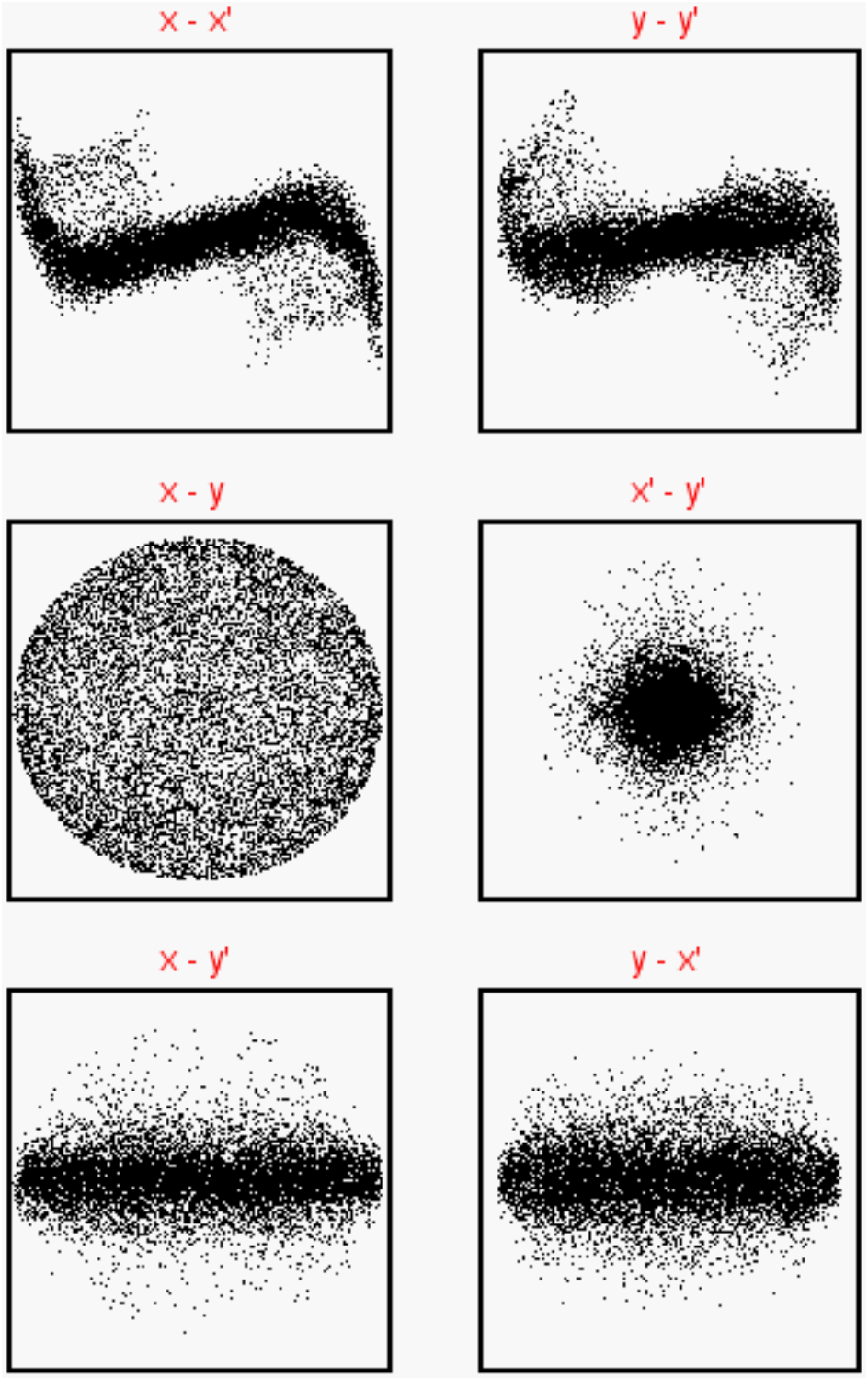

FIG. 27. Particle "scatter" plots of synthesized distributions using 2-plane method G1. with CLIPALG=0, CLIP=1.0, CLIPELLIPSOID=1.4; planes as noted in figure. 102285 of the desired 102288 particles were generated.

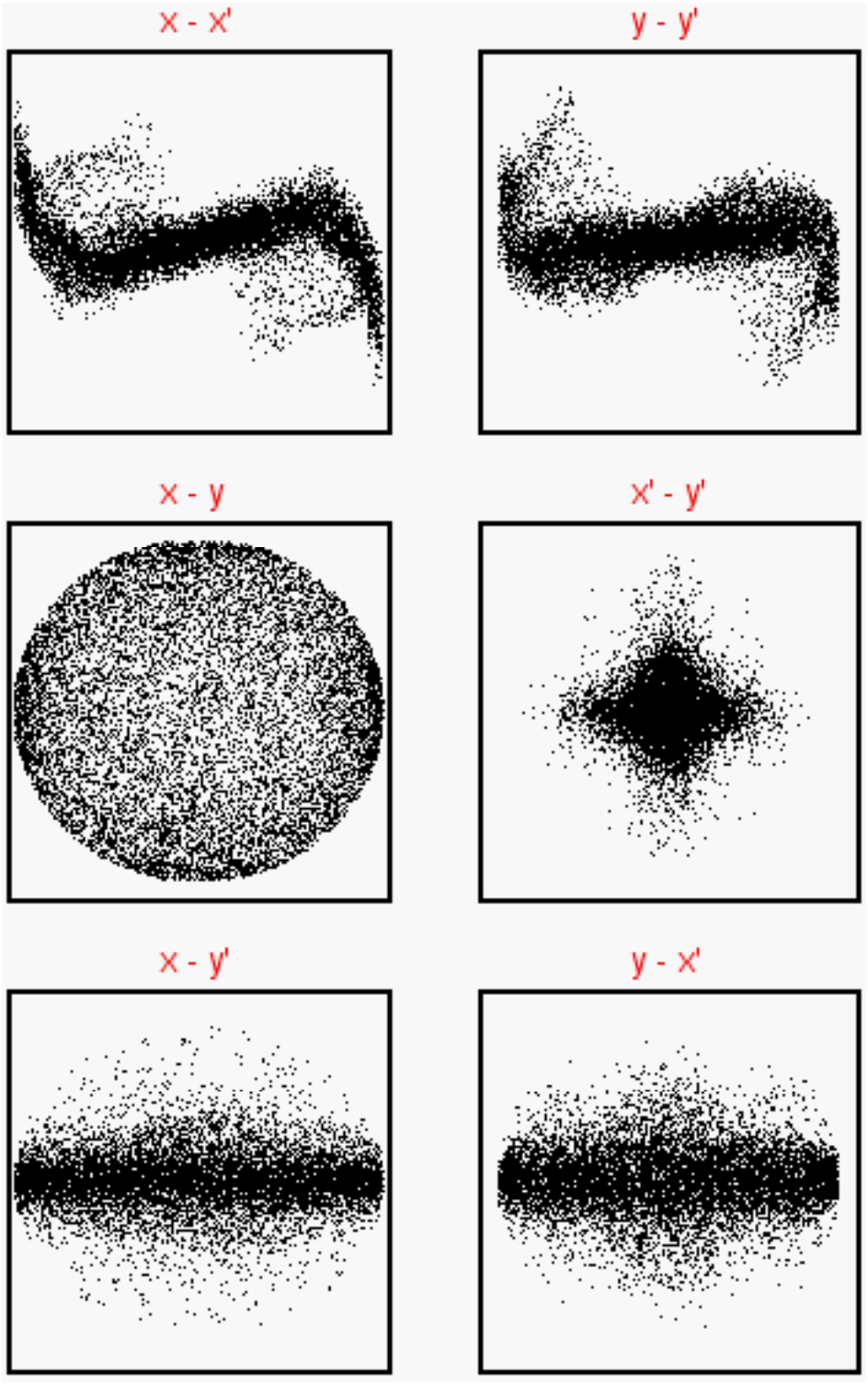

FIG. 28. Particle "scatter" plots of synthesized distributions using 2-plane method G2. with CLIPALG=0, CLIP=1.0, CLIPELLIPSOID=1.4; planes as noted in figure.

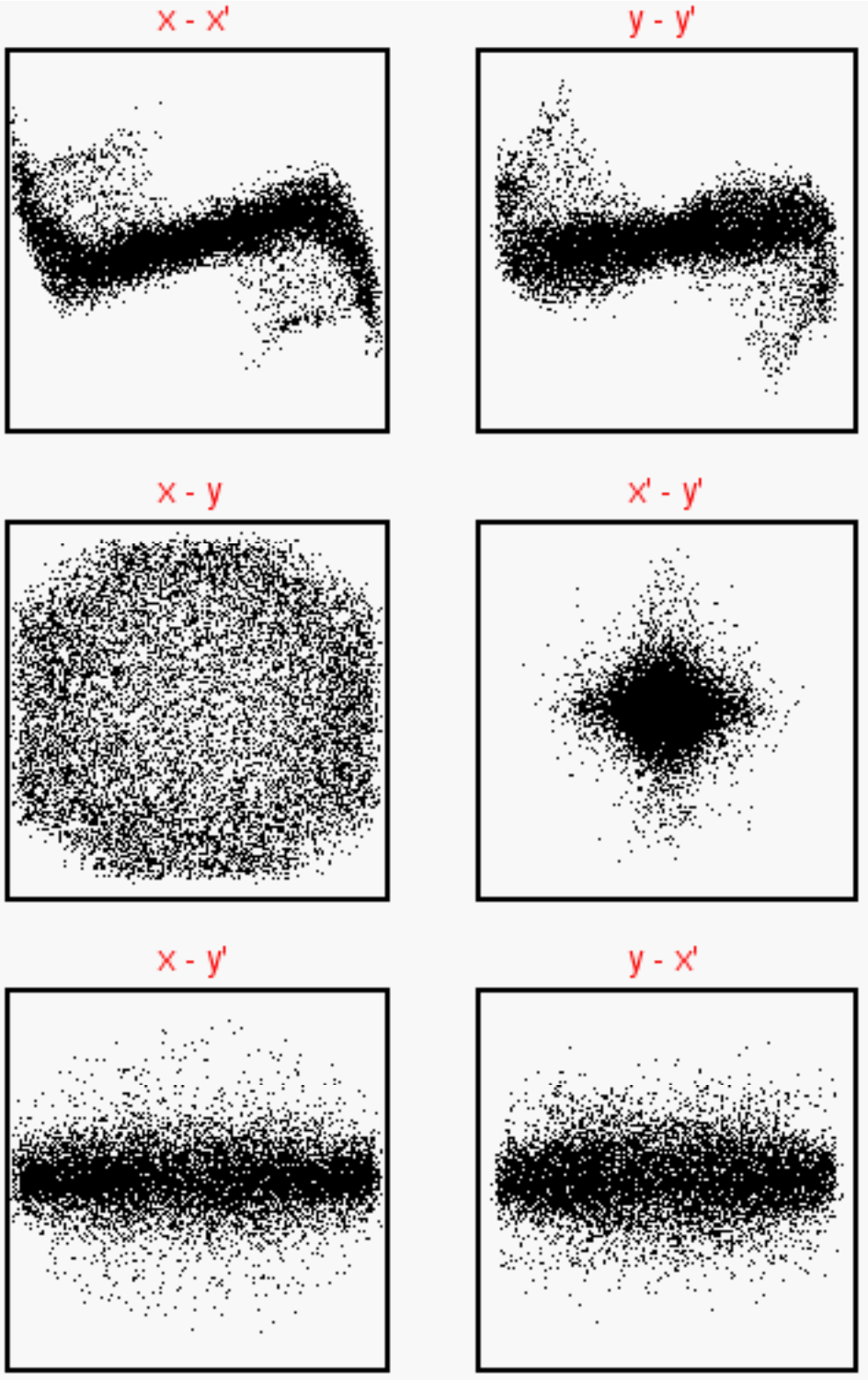

FIG. 29. Particle “scatter” plots of synthesized distributions using 2-plane method G2. with CLIPALG=1, CLIPELLIPSOID=1.2; planes as noted in figure.

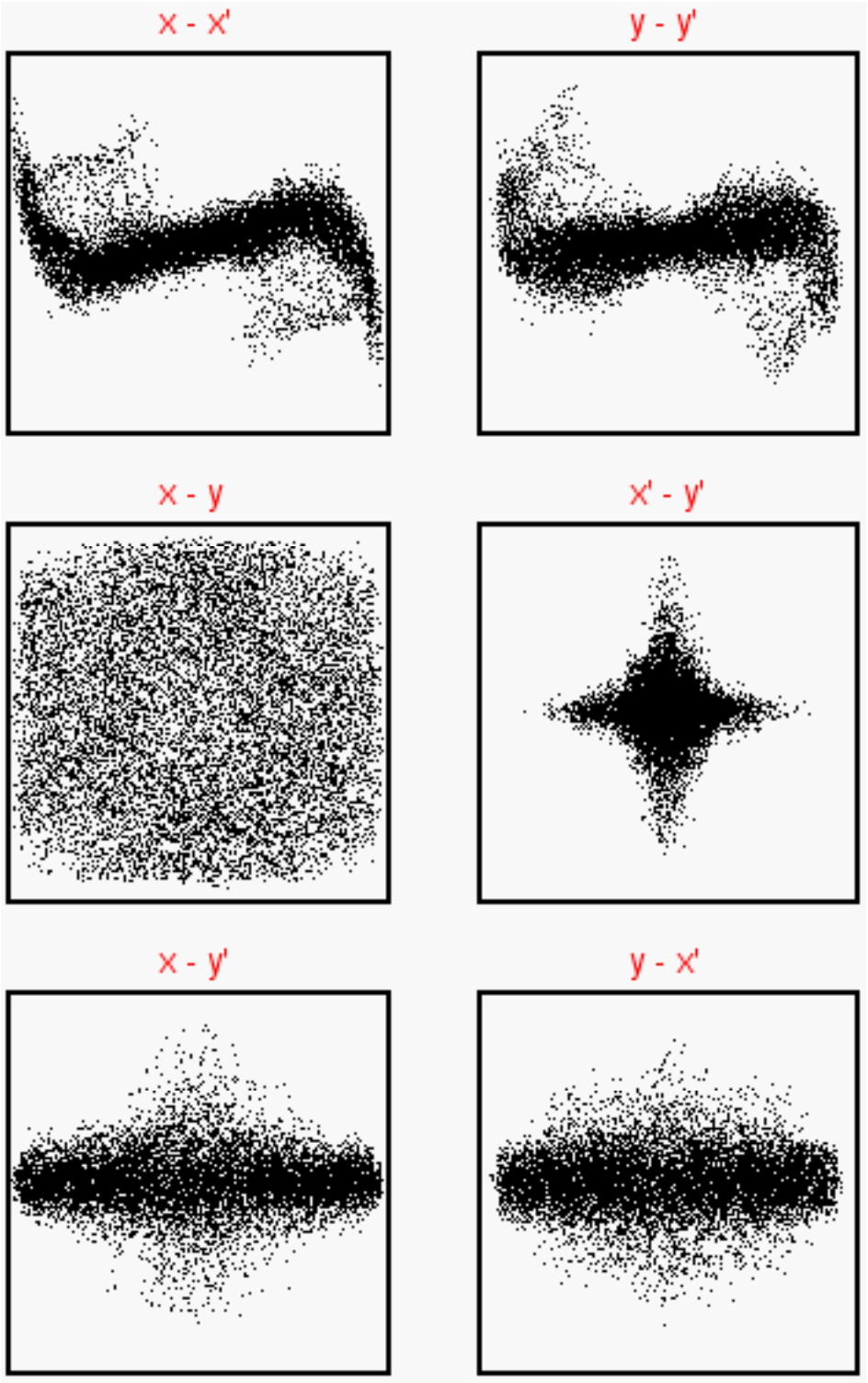

FIG. 30. Particle “scatter” plots of synthesized distributions using 2-plane method G3; planes as noted in figure.

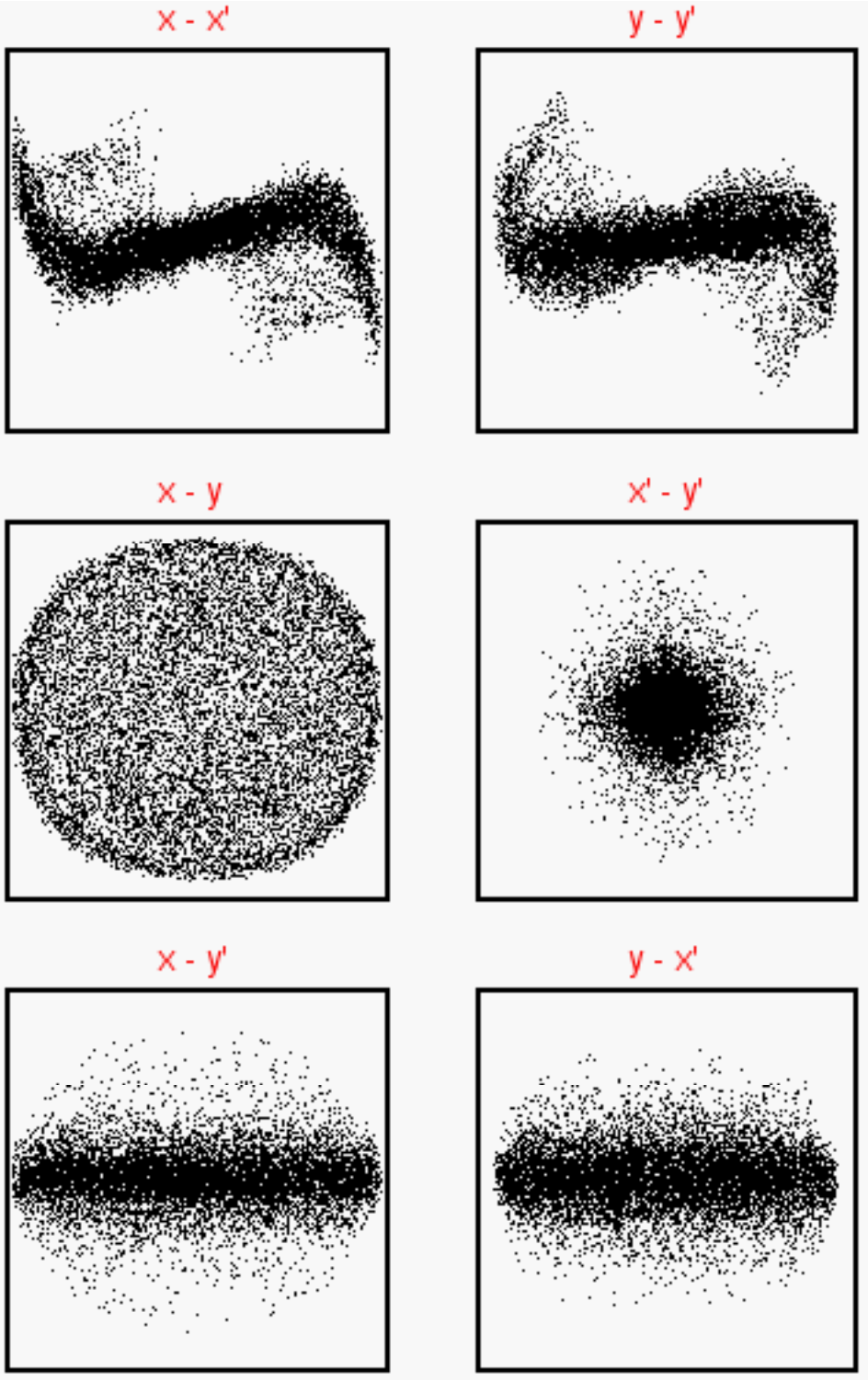

FIG. 31. Particle “scatter” plots of synthesized distributions using 3-plane method G7. with CLIPALG=0, CLIP=1.0, CLIPELLIPSOID=1.4; planes as noted in figure. 102216 of the desired 102288 particles were generated.

In Fig. 31 the results of applying the 3-plane method G7 with CLIPALG=0, CLIP=1.0, CLIPELLIPSOID=1.4 are shown. 102216 of the desired 102288 particles were generated. The "known" moments are accurately reproduced, as is the rim in(x,y). The few "missing" particles are perhaps associated with the "painting into a corner effect." For this problem, the G7 method appears to be the method of choice when it can be used.

To assess the utility of the synthesis concept in general, and of the various methods in detail, we have begun to carry out simulations of the HCX beam line downstream of the ESQ injector. Here, a self-consistent simulation using the "slice" model WARPxy and beginning with the "true" WARP output data at the ESQ exit, as shown above, serves as the reference case. Another simulation used an idealized distribution, the RMS-equivalent "semi-Gaussian" beam (flat-top in its spatial profile and Maxwellian in its velocity coordinates); that is, the beam had the same RMS size, emittance, and "tilt" in each principal plane. We also carried out simulations using the two-plane G1 synthesis shown in Fig. 27, and the three-plane G7 synthesis shown in Fig. 31. The resulting "histories" of the (x,x´) and (y,y´) emittances are shown in Fig. 32.

It is evident that use of either synthesis procedure enables a simulation of the downstream system which is far superior to that which can be obtained using a simple RMS-equivalent beam. The latter fails to capture several of the features seen in the emittance evolution. In addition, the 3-plane synthesis does better than either the 2-plane synthesis or the semi-Gaussian at capturing the initial emittance evolution. It is strikingly accurate for the (y,y´) emittance; in the (x,x´) plane its qualitative behavior is similar to that of the self-consistent distribution, but the size of the jump at the exit of the matching section (z = 0) is incorrect. The short-time behavior is of particular importance in the HCX experiments because the beam will be diagnosed while it is still evolving rapidly; the experiment will not be long enough for the "asymptotic" regime in which the beam has settled down to a near steady-state to be reached. This simulation effort has just begun. More results and details will be presented in a future article.

VII. DISCUSSION AND RECOMMENDATIONS

We anticipate that the methods described here will be of significant utility in experiment analysis. Their performance on the problems described in this paper is encouraging, and the failure of simple idealized distributions to yield acceptable answers provides strong motivation to employ the best available synthesis techniques.

Of the various methods presented here, the G1 and G7 (bin-based) methods seem to consistently offer faithful reproduction of "known" moments and of the phase-space projections in the "known" planes. It should be possible to improve on those methods, *e.g.* by eliminating the "painting into a corner" syndrome as suggested earlier.

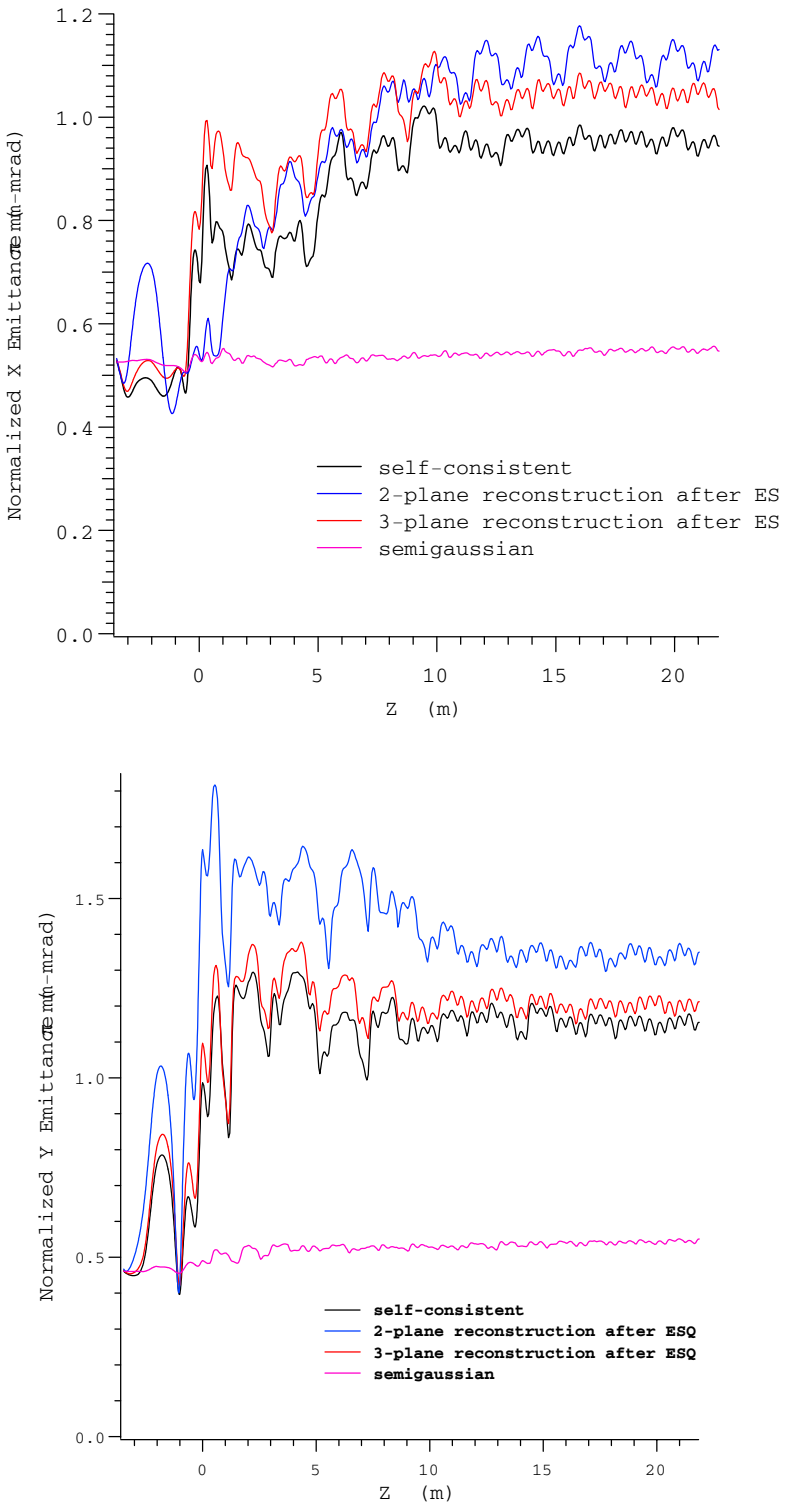

FIG. 32 (color). WARPxy simulations: downstream evolution of (x,x´) and (y,y´) emittances for self-consistent, 2-plane reconstruction, 3-plane reconstruction, and semi-Gaussian beams. The axial location z = 0 corresponds to the end of the matching section.

The superior performance of the 3-plane G7 method leads us to recommend its use whenever possible. The spatial information provided by a spatial density (crossed-slit) scan guarantees that the potential energy will be correctly included in the initialization of a simulation of the downstream system. Because the kinetic energy folds in the various velocity components in quadrature, and all synthesis methods do a good job of reproducing the x and y velocity distributions, inclusion of this plane's information guarantees that the total energy in the simulation will be correct.

An unattractive aspect of the methods based on a sampling volume is the need for the user to specify the clipping parameters. This requires a degree of judgment on the part of the user, and introduces the possibility of subjective bias. Thus, we were initially encouraged by the good performance of the anticorrelation-based G3 method on the model problem and its apparently reasonable performance on the RFQ-injection problem. However, its failure to generate a reasonable spatial distribution in the ESQ problem leads us to believe that the assumptions on which it is based are not satisfied for that problem. In other applications it may be the method of choice.

We believe that it is possible to improve on the anticorrelation-based method. Perhaps an attractive method might result from insisting that all points are chosen from a sampling region. Ideally, we would prefer to develop another "parameter-free" method based on geometrical properties that are actually present in systems such as the ESQ injector. Clearly there is room for invention in this area.

We plan to continue research into other synthesis methods. On the practical side, it will be important to develop a method tuned for synthesizing a beam distribution near the "gun" exit, where the beam is (nearly) axisymmetric and may be well-described by $f(r,p_r,p_\theta)$, where r and θ are the usual polar coordinates. We also would like to develop a means of incorporating any extra available information, *e.g.* localized data at high resolution, into the synthesis. Finally, we would like to establish a firmer theoretical basis for these methods, and to investigate whether there may be other guiding principles that can be employed, *e.g.* maximum entropy principles, in completing the problem specification.

In addition to this "inverse" approach, we are also working on the "forward" problem via integrated simulations starting from the source. To achieve agreement with experimental data at downstream stations, it is necessary to vary (within the experimental error bars) voltages, alignment, and/or source inhomogeneities. While the "forward" approach is more consistent than the one described in this paper, it can be laborious (sometimes prohibitively so). In any event, we anticipate that much of our day-to-day work will be done using the synthesis procedures described here.

ACKNOWLEDGMENTS

We are pleased to acknowledge valuable discussions on these topics with J. J. Barnard, I. Haber, E. P. Lee, S. M. Lund, and J.-L. Vay. The simulation data used in the HIF injector case described in the text was supplied to us by E. Henestroza.

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore and Lawrence Berkeley National Laboratories under Contract Nos. W-7405-Eng-48 and DE-AC03-76SF00098.

---

[1] Martin Reiser, *Theory and Design of Charged Particle Beams* (Wiley, New York, 1994).
[2] R. F. Welton, M. P. Stockli, J. E. Boers, R. Rauniyar, R. Keller, J. W. Staples, and R. Thomae, Rev. Sci. Instrum. **73 No. 2,** 1013 (2002).
[3] C. F. Chan, W. S. Cooper, J. W. Kwan, and W. F. Steele, Nucl. Inst. Meth. in Physics Research **A306,** 112 (1991).
[4] L. Ahle and H. S. Hopkins, in *Proc. Beam Instrumentation Workshop,* 1998, edited by R. O. Hettel, (AIP Conference Proceedings No. 451, College Park, MD, 1998), p. 507.
[5] J. Stovall, Los Alamos National Laboratory, private communication.
[6] F. M. Bieniosek, E. Henestroza, J. W. Kwan, L. R. Prost, and P. A. Seidl, in *Proc. Particle Accelerator Conference,* 2001, edited by P. Lucas and S. Webber, (IEEE, Piscataway, NJ, 2001), p. 2099.
[7] P. A. Seidl, R. O. Bangerter, C. M. Celata, A. Faltens, V. Karpenko, E. P. Lee, S. M. Lund, A. W. Molvik, and I. Haber, in *Proc. Particle Accelerator Conference,* 2001, edited by P. Lucas and S. Webber, (IEEE, Piscataway, NJ, 2001), p. 2932.