ℰ

CLRC

Technical Report
RAL-TR-1998-048

Computer Simulation of the Motion of Charged-Particle Beams Under Space Charge

C R Prior

17th July 1998

COUNCIL FOR THE CENTRAL LABORATORY OF THE RESEARCH COUNCILS

inertial confinement fusion study (HIDIF), the proposals for an upgrade to ISIS using a method of dual harmonic RF acceleration, and a recent investigation to determine the extent to which octupoles and higher order elements might be used to adapt particle distributions and transverse beam shapes into forms suitable for spallation and other problems involving targets.

II PRINCIPLES OF SIMULATION

The basic ideas behind the simulation of particle beams in accelerators are mathematically simple but require care in their implementation. A model beam of typically 10^4 to 10^6 macro-particles is used to represent a real beam, which might have the order of $\sim 10^{14}$ particles in a bunch. The degree to which this is a genuine representation, and predicted effects are not consequences of the reduced number of particles, is related to problems of a statistical nature, and other difficulties, such as rounding and interpolation errors generated in computers, which have finite word-length and work to a fixed precision, also need to be countered.

In the absence of space-charge, recent applications of mathematical theories such as differential and Lie algebras have led to fast and accurate codes based on symplectic transfer maps working to very high order of accuracy. Examples are COSY INFINITY [1] and MARYLIE [2]. Study is underway to include the effects of self-fields, though, since the principles behind the codes are somewhat different to normal tracking techniques, this can probably only be achieved, at least in the first instance, by approximation methods.

Conventional particle tracking codes rely on the generation of an input model beam distribution in phase-space, giving the macro-particles for the subsequent simulation. Packages are readily available for most standard distributions and can quickly be written in other cases. A convenient technique is the "ratio of uniform deviates" method [3], where a density function $f(x)$ is created by uniformly filling the region

$$0 < u < \sqrt{f\left(\frac{v}{u}\right)}$$

of the two-dimensional (u, v)-plane, with a random number generator. The random variable $x = \frac{v}{u}$ has the desired density function.

Progress of the beam through the accelerator is then determined by integrating the appropriate equations of motion. A suitable formalism, based on a standard Serret-Frenet frame of unit vectors moving with the beam, and the ensuing Hamiltonian development, is described in [4]. Various methods of integration may be employed, from linear Euler (forward) difference methods to more complicated Runge-Kutta approaches. However the higher the order of accuracy, the longer the computing time taken to integrate forward in time and it is usually best to adopt the traditional leapfrog method, giving accuracy to second order in the step-length, and use more steps of shorter length. At each stage the external forces (from the

COMPUTER SIMULATION OF THE MOTION OF CHARGED-PARTICLE BEAMS UNDER SPACE CHARGE

C.R. Prior [1]
Rutherford Appleton Laboratory, Chilton, Didcot, U.K.

I INTRODUCTION

A developing involvement in accelerators in which space charge effects play a prominent rôle has prompted studies at the Rutherford Appleton Laboratory into computational tools to treat theoretical problems of beam transport by means of simulation. The main emphasis has been on storage rings and the modelling of injection using stripping foils or septa, but other aspects of beam dynamics have also been investigated. By-products of the tracking codes have been analytical computer programs which generate basic sets of working parameters and, through linear theory, are able to achieve fairly high degrees of optimisation. The simulation codes are then used to confirm these results and explore non-linear effects.

As with all particle tracking codes, the calculation of space charge forces occupies a considerable slice of the programming and, usually, the major part of the CPU run time. Computer speed and core were major restrictions on the kind of modelling which could be achieved 10-15 years ago and many ingenious ideas were developed in order to produce worthwhile results. Nowadays, storage is not a problem, and computing speeds have increased and the cost of CPU time been dramatically reduced so that we think little of running continuously for several days in order to produce results that would have been unthinkable only a few years ago. Earlier ideas for saving on storage and time are, however, not wasted: they now release space and provide opportunities for more complicated simulations using greater numbers of macro-particles, and allow greater accuracy in the type of calculations which, for example, depend on the refinement of a mesh or the size of an integration step length.

In this paper, the general approach to simulation and the fundamental details behind the structure of the RAL codes are described. A range of illustrations is included covering modelling of the H^- injection scheme for the European Spallation Source (ESS), the tilted septum method of injecting heavy ions for the European

Poisson's equation is solved, then the answers are converted back to the laboratory frame for integration forward by one time step. The space charge potential in the beam frame is the function ϕ which gives a stationary value to

$$\pi(\phi) = \frac{1}{2}\int_V |\nabla\phi|^2 \, \mathrm{d}V + \frac{1}{\epsilon_0}\int_V \rho\phi \, \mathrm{d}V - \int_{\partial V} \bar{\phi}_n \phi \, \mathrm{d}S. \tag{1}$$

This variational problem is equivalent to

$$\nabla^2\phi = \frac{\rho}{\epsilon_0} \qquad \text{in } V \tag{2}$$

with

$$\phi = \bar{\phi}, \qquad \frac{\partial\phi}{\partial n} = \bar{\phi}_n \qquad \text{on} \quad \partial V. \tag{3}$$

In two-dimensions, V would be the cross-section of the accelerator channel with ∂V the bounding wall; in three-dimensions, V would be the (topologically) cylindrical region occupied by the beam in the accelerator, with appropriate (say periodic) boundary conditions imposed longitudinally.

Taking two-dimensions as illustration, the variational problem is solved by imposing a triangular mesh across V and fitting to each element a polynomial function for $\phi(x, y)$ of degree n. To avoid extra calculations which do not actually enhance accuracy, one looks for a complete set of functions, which means that ϕ contains $N = \frac{1}{2}(n+1)(n+2)$ unknowns: for example, for $n = 2$ the unknowns are the six coefficients a_i in $a_0 + (a_1 x + a_2 y) + (a_3 x^2 + a_4 xy + a_5 y^2)$. This expression can be restructured in terms of the values of ϕ at specific points of the mesh (the "nodes"). The so-called serendipity method of allocation may be used and an illustration of the nodes for $n = 3$ would then be as shown below.

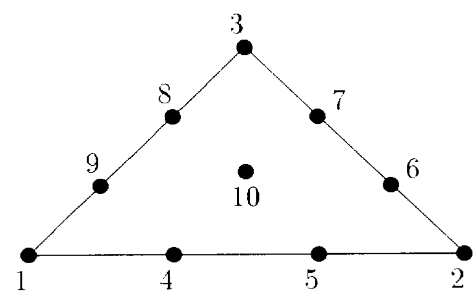

FIGURE 1. Node arrangement for triangular mesh with $n = 3$ potential fit.

Introducing standard areal coordinates ξ_i, $i = 1, 2, 3$, for each triangular element ($\sum \xi_i = 1$), we can write

$$\phi(x,y)=\sum_{i=1}^{N} f_i(\xi_j)\phi_i \tag{4}$$

where f_i are the shape functions and $\{\phi_i\}$ the potentials at the nodes. For $n=1$, $f_i=\xi_i$, and for $n=2$, we would use $f_i=\xi_i(2\xi_i-1)$ for $i=1,2,3$, $f_4=4\xi_1\xi_2$, $f_5=4\xi_2\xi_3$, and $f_6=4\xi_3\xi_1$.

The source terms $\frac{1}{\epsilon_0}\int \rho\phi \, \mathrm{d}V$ depend on a model for the charge density ρ. The simplest is to treat the macro-particles as δ-functions:

$$\rho(\mathbf{x})=\sum_{\text{particles, } i} q_i\delta(\mathbf{x}-\mathbf{x}_i).$$

Alternatively, one could introduce an arbitrary spreading of charge by, for example, treating each macro-particle as a small Gaussian distribution of pre-determined size. The δ-function formalism under (1) is in fact equivalent to assuming a uniform spread of charge over a disc, provided it is contained entirely within a single element of the mesh.

Substituting these formulae into (1), the variational problem reduces to the equivalent condition $\partial\pi/\partial a_i=0$, which results in a set of simultaneous equations for the $\{\phi_i\}$. These are adjusted to account for the boundary conditions (3), giving an equation involving only internal nodes:

$$K_{ij}\phi_j=Q_i. \tag{5}$$

Here the "stiffness" matrix K_{ij} depends only on the mesh and Q_i are the source terms. Equation (5) is solved using techniques such as the conjugate gradient method or the method of triple factoring. The matrix K is extremely sparse and there are well-known methods for handling such linear problems. With a few exceptions, only the non-zero elements are stored and this number can be minimised by appropriate global numbering of the mesh nodes. Space charge forces are then found from

$$\nabla\phi=\sum_{i,j}\frac{\partial f_i}{\partial\xi_j}\phi_i\nabla\xi_j.$$

Programming for any level of polynomial modelling is relatively simple but the degree of accuracy of the answers depends on the refinement of the mesh and the order of fit. Taking 10,000 simulation particles, for $n=1$ (linear potential fit) roughly 3000 triangular mesh elements are needed to give about 90% of space charge forces within a 10% error band for the circular, uniform beam mentioned above. For $n=3$ this can be reduced to about 400 and would give 95% accuracy (the errors being along the axes and actually very small, though greater than 10% in relative terms).

Meshes used in the RAL codes are formed of triangular elements generated by radial and circular lines based on the beam centre, with different (continuous) radial scalings over the beam and the external region. As the beam moves, the mesh adjusts internally, while the boundary wall is fixed (infinite boundaries are simulated by "super-elements" mapping the potentials onto $\ln r$ terms (2D) for large r). An example of an off-centre beam in a square conducting chamber is shown in Figure 2. If the beam varies slowly, the mesh need only be re-generated every 5-10

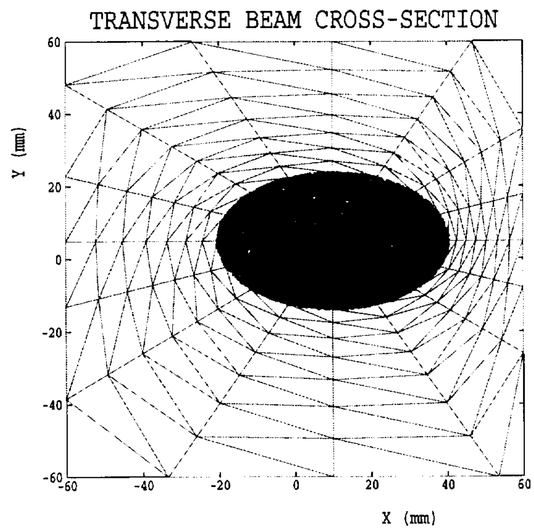

FIGURE 2. Finite element space charge mesh showing an off-centre particle beam in a rectangular conducting chamber.

time steps. In such cases the matrix K_{ij} may be pre-inverted and stored (using the same storage locations), so that only the source terms Q_i need be re-calculated as the particles change their positions; the $\{\phi_i\}$ are then quickly computed from $\phi_i = K_{ij}^{-1} Q_j$.

Extension to three-dimensions is both simple and natural. The region is split longitudinally into planes covered by triangular meshes. Pairs of elements in adjacent planes are joined to form tetrahedra, in each of which volume coordinates ξ_i, $i = 1, 2, 3, 4$, $(\sum \xi_i = 1)$ are used. With four vertices, the simplest polynomial fit is $\phi = a_0 + a_1 x + a_2 y + a_3 z$; the quadratic form requires six extra nodes, which are positioned at the mid-points of each of the edges. The procedure described above applies and equations (1) to (5) remain valid in 3D.

Note that a slight adaptation of the 2D x-y code will give a 3D r-z code with axisymmetry. Starting with a rectangular mesh, a linear potential fit $\phi = a_0 + a_1 r + a_2 z + a_3 rz$ would correspond to nodes at the four vertices, and a 12-node fit (up to cubic terms plus terms $r^3 z$ and $z^3 r$) would include two nodes per mesh edge and give better accuracy. Then all that is needed is the change $\mathrm{d}x\,\mathrm{d}y \to r\,\mathrm{d}r\,\mathrm{d}z$ in the integrals of (1), which are in general performed numerically, and the relevant re-interpretation of the answers.

IV 1D LONGITUDINAL TRACKING CODES

A code which has proved particularly useful in recent years is the RAL program TRACK1D[2], which simulates the injection, acceleration and trapping of bunches in a synchrotron. The basic physics is governed by the equations [6]

$$\frac{\mathrm{d}}{\mathrm{d}t}\Delta\phi = \frac{h\omega_0\eta}{\beta_s^2 E_s}\Delta E \tag{6}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\Delta E}{\omega_0} = \frac{e}{2\pi}[V(\phi) - V(\phi_s) + V_{sc}(\phi)] \tag{7}$$

where variations ΔE and $\Delta\phi$ in energy and phase from synchronous values are used as coordinates. $V(\phi)$ is the applied RF waveform, h is the harmonic number, ω_0 the revolution frequency and $\eta = 1/\gamma_t^2 - 1/\gamma^2$. In this one-dimensional model, the space charge forces are given in terms of the particle line density, λ, by [7]

$$V_{sc}(\phi) = e\frac{h}{R}\frac{\mathrm{d}\lambda}{\mathrm{d}\phi}\left[\frac{Rg_0}{2\epsilon_0\gamma^2} - L\beta^2c^2\right]. \tag{8}$$

L is the total inductance per turn of the reactive wall (usually small and ignored), R is the average machine radius, and for a beam of circular cross-section of mean radius a in a pipe of radius b, $g_0 = 1 + 2\ln(b/a)$. (This formula for g_0 is valid only for long bunches with half-length $z \gtrsim 10a$; for shorter bunches, the linac formula

$$g_0 = \begin{cases} \dfrac{1}{1-\zeta^2} - \dfrac{\zeta\cos^{-1}\zeta}{(1-\zeta^2)^{3/2}} & \text{for} \quad \zeta < 1 \\ \dfrac{\zeta\cosh^{-1}\zeta}{(\zeta^2-1)^{3/2}} - \dfrac{1}{\zeta^2-1} & \text{for} \quad \zeta > 1 \\ \dfrac{1}{3} & \text{for} \quad \zeta = 1 \end{cases}$$

where $\zeta = z/a$, is more appropriate.) From an initial model bunch of $N_P \sim 10^4$ simulation particles in ΔE-$\Delta\phi$ space, a repetitive process of calculating the applied voltages and space charge terms and updating the coordinate in intervals of one time step is applied. For such a simple model, it may be surprising that implementing a suitable code is quite difficult. The main problem is that the line density has first to be determined from the particle distribution and this will inevitably suffer from statistical fluctuations owing to the relatively small N_P. Trying to calculate the derivative, $\frac{\mathrm{d}\lambda}{\mathrm{d}\phi}$, from such a λ will lead to substantial errors which will propagate and most likely grow with time. The solution is to introduce a smoothing process to iron out the statistical effects while at the same time retaining the main features of the distribution. Fourier analysis with up to approximately 10 harmonic terms tends to follow the noise too closely. However a satisfactory approach has been

[2] RAL tracking codes are noted for the complete lack of originality in their names.

found at RAL using cubic splines in which an automatic node allocation identifies the real trends in λ and minimises random errors [8].

TRACK1D contains other unusual features which have been developed from a need to simulate the injection of many turns into longitudinal phase space. A standard approach to injecting N_t turns would be to use N_P/N_t simulation particles per turn, but this number can be unreliably small or forces N_P to be awkwardly large (for $N_t = 1000$, one would need $N_P \geq 10^6$ for example). TRACK1D therefore adopts a process during injection of building up charge at selected points, somewhat akin to the Euler approach to fluid dynamics of identifying changes in properties at fixed locations. After injection is completed the code reverts to a more traditional Lagrangian technique of following particles as the distribution evolves. A third coordinate ("charge" or, more precisely, the number of real particles each macro-particle represents) is introduced. As each beamlet is injected, its particles are allocated to an imaginary rectangular grid of $\lesssim N_P$ nodes in ΔE-$\Delta\phi$ space according to a weighting scheme given by functions of the type $f_i(\xi_j)$ in (4). Beam already injected is similarly allocated to the grid, producing at each stage approximately N_P elements of charge, whether for the first turn or much later in the cycle. Once injected, the nodal charges are regarded as particles to generate the space charge forces for a single time step evolution according to (6) and (7) under which they will move from their grid positions. As the next turn is injected, particles in the machine are weighted back onto the imaginary grid, giving a slightly different arrangement, and the process repeated. The particles used under this procedure, therefore, all carry different amounts of charge, which change as the process is repeated and beam is built up. The specific charge allocation schemes that might be used [9] are (in order of complexity) commonly known as the nearest-grid-point method (NGP), cloud-in-cell scheme (CIC) and the triangular-shaped-cloud (TSC), given mathematically by repeatedly convoluting the "top-hat" function

$$\Pi(x) = \begin{cases} 0 & |x| > \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 1 & |x| < \frac{1}{2} \end{cases}$$

with itself. In fact, TRACK1D uses the next function up in this series (so far unnamed). Charge assignment has not only the benefit of smoothing the distribution but also the unfortunate side-effect of artificially spreading the beam, and TRACK1D includes correction techniques to counteract this effect. The process of assigning charge to the grid points is depicted in Figure 3 and examples of the use of the code are covered in section IX.

V 2D TRANSVERSE TRACKING CODES

Many of the ideas contained in the longitudinal code have recently been incorporated into the two-dimensional transverse program TRACK2D [10]. This code is built in modules, using the same routines for reading in details of the beamline or

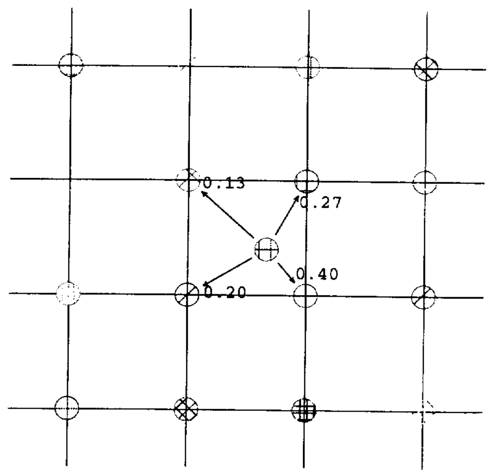

FIGURE 3. Injection mesh for TRACK1D showing the allocation of an incoming particle to the nodes (cloud-in-cell method).

accelerating structure, graphics and optimising facilities as other related codes. By these means, one is sure that elements are being handled in exactly the same way in the tracking calculations as in, say, analytical treatments used for more basic design and comparison. It is also easy to add new modules within the existing structure as new questions arise.

TRACK2D includes the finite element space charge techniques described above. The default routine uses a third order potential fit, but orders up to six are used for verification purposes. Equations of motion are either based on paraxial modelling or include third order effects of the transverse particle angles. Apart from direct simulation of a particle beam through a focusing channel (a storage ring or a beamline, such as the final focusing system for a heavy ion fusion scheme), the code will handle injection and contains an optimiser to determine design parameters for systems under non-linear effects. The optimiser is essentially the analytical code KVBL [11], used for machines with zero current or linear space charge governed by the KV-envelope equations, driving TRACK2D with the space-charge mesh in reduced form. It is best used for slight modifications to a system pre-optimised, so far as is possible, analytically.

In injection mode, the code currently still uses the conventional approach of N_P/N_t simulation particles per injection turn but work is in progress to incorporate the variable-charge scheme described in section IV. This will need to use an irregular 4-dimensional grid in phase space in order to maintain a suitable number of "particles" in real space for proper space charge calculations and beam simulation. The code is able to simulate multiturn injection using septum magnets in either or both planes, or charge-exchange injection by means of stripping foils, as, for example, in ISIS or the ESS [12]. In the latter case, details of the passage of particles through the foil can be used to study its thermal properties and likely peak temperatures [13].

Machine errors such as field strength errors, magnet alignment errors (horizontal, vertical and longitudinal) and bending field errors, may be read in in a simple way or generated internally via a Gaussian random number generator using prescribed rms values. The code also contains models of fringing fields. There was originally an intention that it could determine emittance effects in electron machines, such as synchrotron light sources, where machine errors, leading to coupling via closed orbit and dispersion, need careful control; but since space charge is negligible in these cases, the calculations can be performed much more quickly and effectively by other means [14].

By allowing most major parameters to be read in as tables of varying data, TRACK2D is also able to take into account certain longitudinal effects of bunched beams. Information from TRACK1D, such as bunching factor or changes in momentum spread $\Delta p/p$, can thereby be incorporated into the simulations. Particles carry as an extra coordinate their charge, which may be varied as beam particles move in and out of the transverse slice under consideration. This means that TRACK1D and TRACK2D work in conjunction with each other, each being run successively with appropriate modifications to data until a convergent set of parameters is achieved. Since full three-dimensional simulation under non-linear space charge is extremely slow, this is quite an effective alternative and can be carried out over a realistic timescale.

VI 3D TRACKING CODES

The three-dimensional tracking code, TRACK3D, is an extension of TRACK2D with the inclusion of a third longitudinal (position) coordinate into Maxwell's full equations of motion under the Lorentz force. Space charge is calculated using the 3D version of the finite element codes described in section III. Although written and running some years ago, this code was little used and largely untested because of restrictions on computer storage and, more especially, the availability of CPU time. In consequence, it fell behind TRACK2D in its development as current projects and new questions arose (section IX). It is now being brought up to date with a view to modelling in more detail various aspects of the ESS design [12] and ISIS. In particular, an improved method of global node numbering for the finite element space charge test needs to be discovered in order to reduce the storage space required for the sparse stiffness matrix K_{ij}. Additional use of the method known as static condensation, whereby sub-units of the mesh, rather than the whole mesh, are read into the computer and processed in turn, needs to be introduced.

In parallel, a 3D code to model beam transport through RF quadrupoles is being developed at RAL by Letchford [15]. This can handle quite complicated geometries, though the space charge solver is still based on Coulomb forces and cannot handle the effects of image charges. It is thought that the routines from TRACK3D might eventually be incorporated into this code. It will not only provide a useful test of the finite element approach but almost inevitably be faster than the current method as well as solve the problem of how to deal with conducting boundaries.

VII FORMULATION OF RESULTS

Deciding on the quantities to be output and the form of the results from a tracking code is not obvious and important calculations are often overlooked. It is not always clear how best to describe the behaviour of a highly non-linear beam. Quantities such as rms emittances and rms beam sizes are traditionally used but contain a limited amount of information, and perhaps the best approach is to store as often as possible the particle coordinates so that further calculations can be carried out as required at a later date. This can however lead to storage restrictions. Graphical output probably enables one to see best how a beam is evolving and how, for example, halos are generated. But again, different laboratories use different graphics systems, which can prevent codes being truly portable. History shows that attempts at a universal standard, such as GKS, seem inevitably to fail. Furthermore, without cross referencing, results may be misleading: for example, the RAL codes use colour or shading to represent different levels of charge but in the final plots only the topmost colours in the superposition are visible; and again regions which in terms of colour appear to have the highest charge density may in fact be outweighed by regions with apparently lower charge when the actual number of particles plotted is taken into account. This is not a major problem but emphasises that one needs to think carefully about one's results and not always accept them at face value. Interactive, graphical simulation is perhaps the most informative approach as it not only shows how a beam is evolving but indicates immediately when parameters are unsuitable, allowing termination and avoiding wastage of CPU time.

VIII BENCHMARKS AND VALIDATION

Individual calculations in tracking codes need, of course, to be thoroughly tested for accuracy. In addition, it is important to check that a code actually simulates the motion of a particle beam by agreement with accepted theory. The number of available benchmarks is limited but there are sufficiently many for suitable testing.

In the 1D longitudinal case, there is a stationary distribution found by Hofmann and Pedersen [16] (also Neuffer). This is characterised by the line density being proportional to the potential, whatever the shape of the focusing force, so that self-forces caused by space charge and inductive wall impedances are proportional to the external applied voltages, and the longitudinal particle distribution is preserved. A simulation code of the type described in section IV should show single particles in the bunch describing closed orbits in phase-space without deviation. A tracking test should cover several hundred synchrotron periods.

For 2D transverse tracking codes, the main benchmark is the KV distribution [17], where simulation results can be compared directly with numerical solutions of the KV envelope equations. A simple test might be a matched beam in a periodic focusing channel, but a more stringent challenge could be a final focusing beamline for a heavy ion fusion plant, tracking a space charge dominated beam (current $\sim 1\,\mathrm{kA}$, $\beta \sim 0.3$) through a FODO channel, blowing it up under the effects of emittance and space charge from a radius of about 10 mm to 40 cm, and then focusing this to a target spot size of about 3 mm. Using a model based on the envelope equations, one would expect a 2D tracking code to predict at least 90% of particles hitting the target. If this is not the case, one would have little confidence in results obtained from simulations using non-linear beams, such as input Gaussian or waterbag distributions. Codes should also be able to confirm other theoretical results, such as the rms matching of non-linear beams or the basic properties of two-dimensional stationary waterbag and Gaussian distributions (circular beams in constant, axisymmetric, focusing channels [18]).

Suitable benchmarks for 3D simulations essentially reduce to three-dimensional stationary distributions for spherically symmetric bunches in constant focusing structures [18]. In some cases, results from a 3D code can be directly compared with separate longitudinal and transverse simulations where the coupling is small and good agreement might be expected.

Good working practice demands that accurate, up-to-date records are maintained of all changes made to programs, whether relatively minor corrections or more major developments. At each stage, the altered code should be checked against the standard set of benchmark tests. Where possible, independent validation should also be sought, preferably through codes held at other laboratories. ACCSIM [5] is widely available and is useful in this respect. However, it often turns out that one has specific problems in mind when developing a program or making additions and other codes may not yet have found it necessary to address these issues.

IX ILLUSTRATIONS

By way of illustration, four examples of recent use of the codes can be considered.

The first concerns the injection process for the European Spallation Source, depicted in Figure 4 [19]. 2.34×10^{14} H^- ions, in bunches chopped at 60% of the linac duty cycle, are injected over 1000 revolutions into each of two accumulator rings at an energy of 1.334 GeV. A suitable longitudinal distribution is achieved by painting the phase space both by ramping the momentum of the injected turns in time and steering the RF bucket by varying the RF frequency. Repeated simulations using TRACK1D indicated that the optimum (simple) method of ramping is to increase the momentum linearly from the range $[0-2] \times 10^{-3}$ to $[2-4] \times 10^{-3}$ over the period of injection, 0.6 msec. During this time the RF frequency is held at a value equivalent to a particle orbiting with $\Delta p/p = 2 \times 10^{-3}$ and then reduced to the natural frequency at the design energy of the ring. Thus turns are injected into the lower part of the phase space bucket at the start of injection and into the upper half towards the end. This is visible in Figure 4, as is the phase space grid used to model injection as described in section IV. The resulting distribution has a bunching factor of around 0.46. In the transverse plane, injection is at a point in the ring where the normalised dispersion $\alpha_p/\sqrt{\beta} = 2.0$ and this combines with the momentum ramping to provide horizontal phase space painting and help reduce the number of times recirculating protons pass through the foil. Vertical painting is provided by vertical orbit bump magnets which are programmed to give inversely correlated oscillation orbits, large to small in the horizontal plane, small to large in the vertical plane. The resulting mean number of foil traversals per particle is 6-7 and a good transverse distribution is achieved. In these simulations TRACK1D and TRACK2D were used in alternating runs, results from the longitudinal code being linked via the ring dispersion, RF steering and variable charge method into the transverse plane, where subsequent modifications to reduce foil heating were fed back to the longitudinal plane for further optimisation. The results of the simulations have been confirmed by Ohmori [20] and, more recently, Galambos [21] using ACCSIM and other codes.

The second example, principally of the use of TRACK2D, concerns the design of the multiturn injection scheme for 10 GeV Bismuth ions into accumulator rings for the HIDIF fusion study [22]. A novel method is used with an electrostatic septum tilted at an angle of 44° so that injection is into both transverse planes simultaneously. By optimising vertical and horizontal closed orbit bumps, and carefully selecting the nominal tunes of the rings, it proves theoretically possible to inject 20 turns at 400 mA per turn with losses of less than 3%. The incoming beam is largely debunched at injection to reduce space charge effects, so that tune depressions are of the order of 0.04. Selection of the parameters is by the code MISHIF [23] for zero or small space charge, with TRACK2D exploring the more general case of a non-linear beam as shown in Figure 5. Results of the analytical model are closely confirmed.

An interesting third use of TRACK1D, reported in more detail in [24], is the development of a dual $h = 1/h = 2$ harmonic RF system for injection, acceleration and trapping in the ISIS synchrotron. The voltage waveform used here is

$$V(\phi, t) = V_0(t)[\sin\phi - \delta\sin(2\phi + \theta)]$$

where V_0, δ and θ are functions of time, varying according to a carefully calculated programme. The Hamiltonian for such a system has two stable centres about which particles oscillate (visible in Figure 6); by creating these in a continuous process under acceleration and then merging them at the end of the cycle, it should prove possible to trap at least 20% more particles and possibly as much as 50%. This is borne out by simulation, and the scheme is currently featured in plans for an ISIS upgrade, in collaboration with KEK, Japan.

Finally, the use of the optimising routines in TRACK2D is illustrated in Figure 7, which shows the results of an investigation into the extent to which a system of

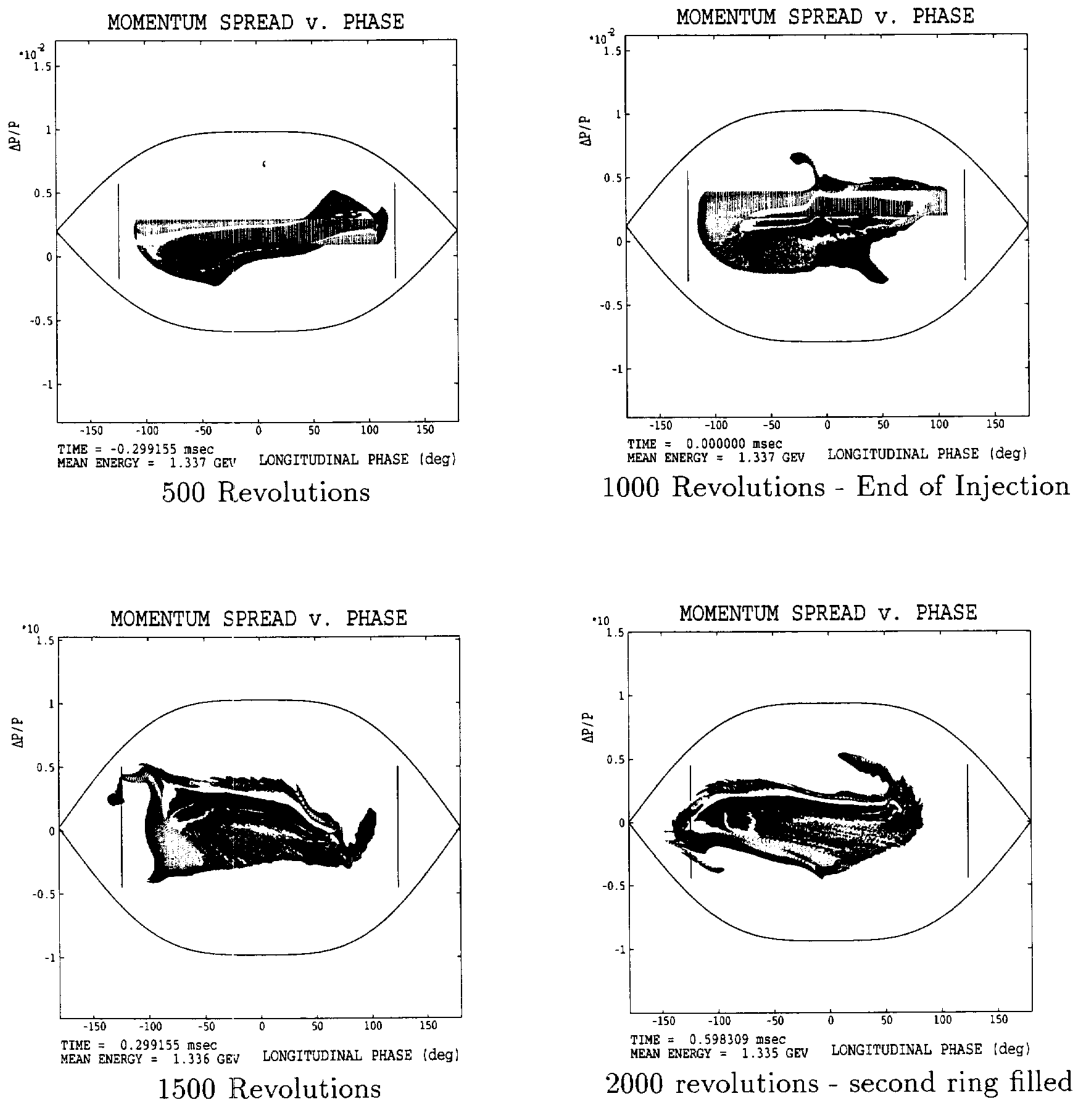

FIGURE 4. Longitudinal phase space plots from TRACK1D for ESS injection.

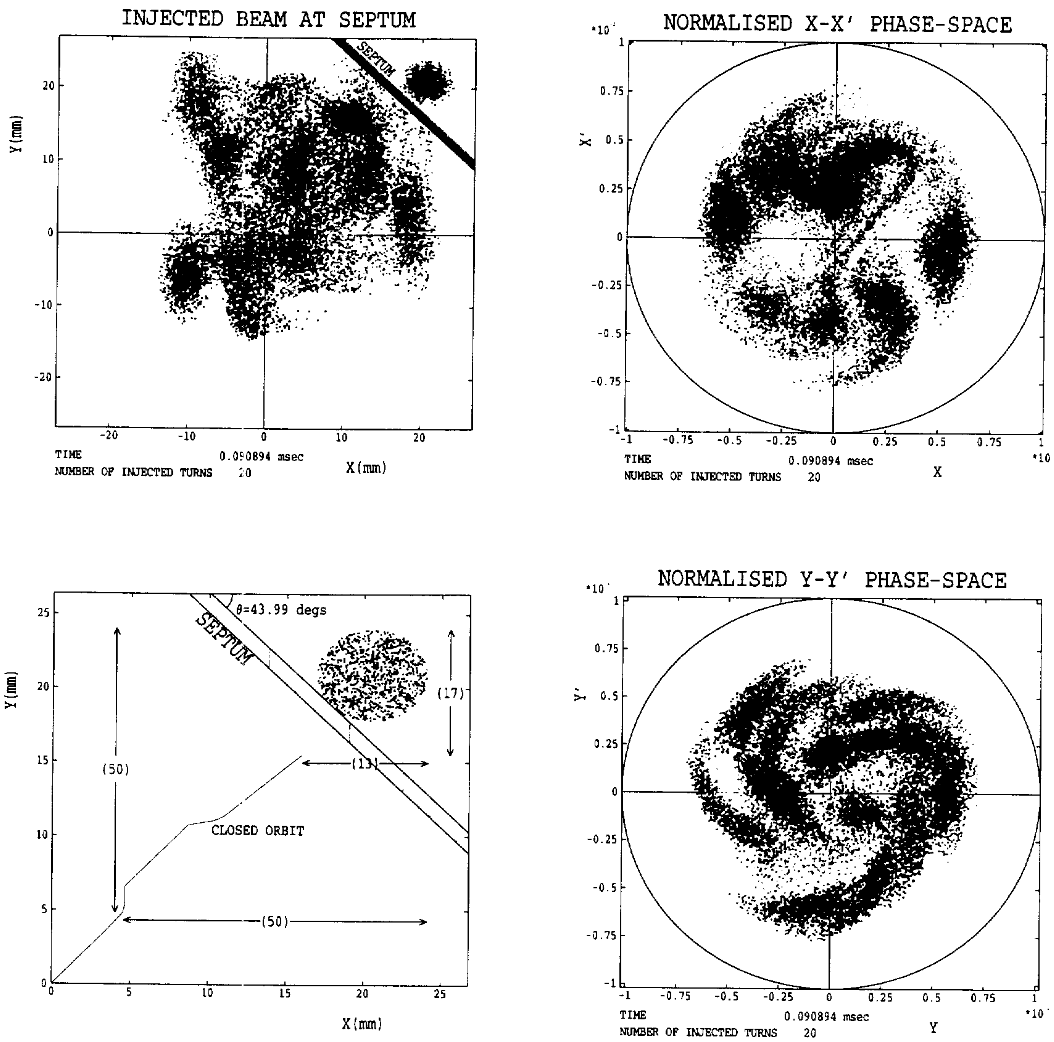

FIGURE 5. Transverse injection using a tilted electrostatic septum into the HIDIF accumulator ring. 20 turns are injected of a beam with a truncated Gaussian distribution and emittances of $4\,\pi\,\mu$ rad m in each plane to achieve a beam with emittances of $50\,\pi\,\mu$ rad m. Losses of less than 3% are predicted.

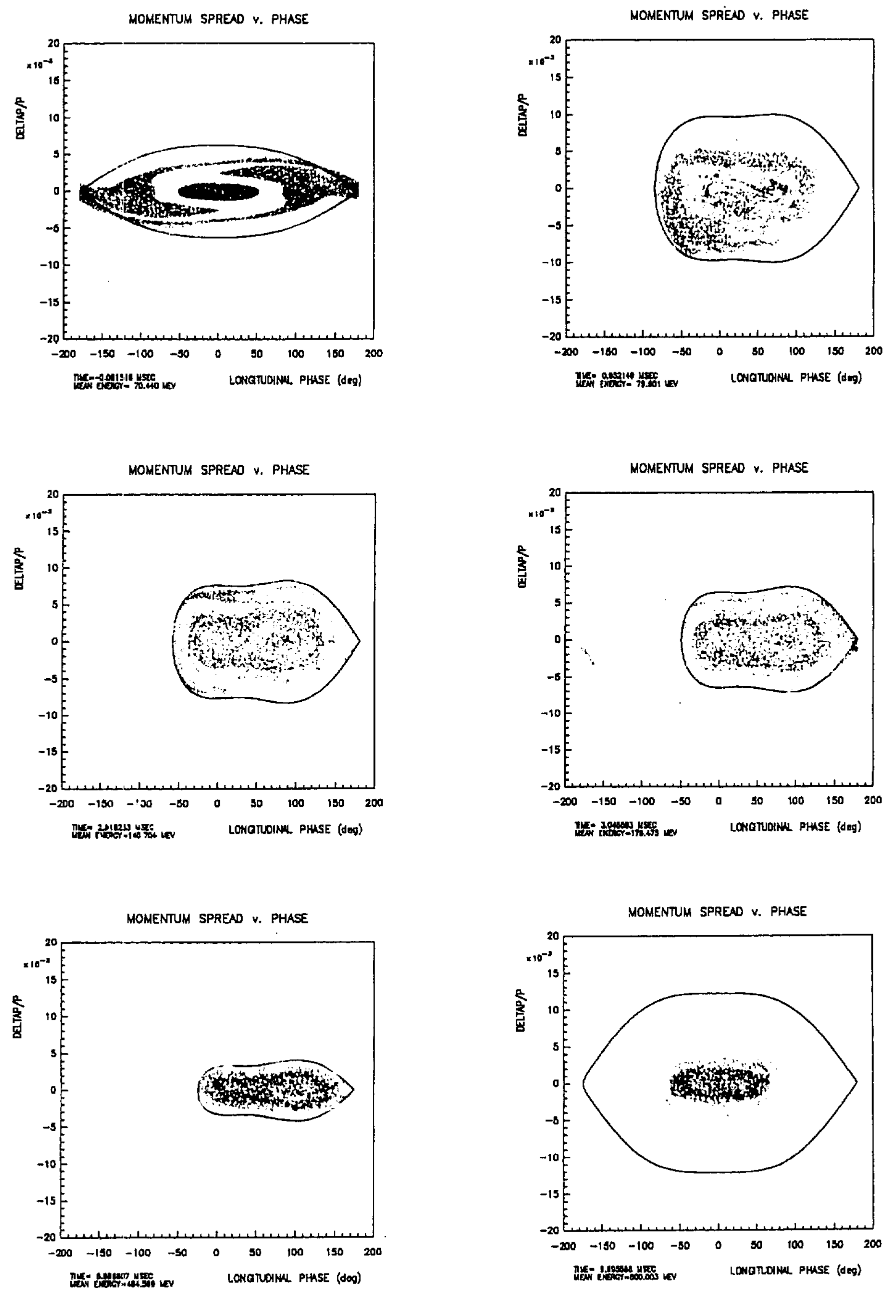

FIGURE 6. Longitudinal simulation of injection, acceleration and trapping for the proposed dual harmonic RF system on ISIS.

octupole and dodecapole magnets might be used to convert a beam with an initially Gaussian distribution into other forms. Repeated runs with a beam of 5000 macro-particles in the absence of space-charge determined in each case the strengths of the higher order elements needed to create a square beam with a uniform distribution, and round beams with two-dimensional parabolic or elliptic distributions. Some minor optimisation was then carried out in the presence of space charge (which was not great for the parameters used in the model). The octupoles have the effect of folding the tail of the Gaussian into the more central parts of the beam and, by adjusting their strengths carefully, this can give a fairly close approximation to each of the distributions considered. The method may conceivably be useful in controlling the development of halos in highly non-linear particle beams.

X SUMMARY

While computers have advanced dramatically and provide far greater scope for research through simulation than in the past, personal experience suggests that progress - at least in Europe - tends to be by individuals in *ad hoc* ways rather than via a co-ordinated approach to a common goal. Different laboratories have different projects and their own particular problems, and codes tend to be developed to meet these individual needs. Seldom is a code found to answer all questions as soon as they arise, and the dream of a set of all-purpose tools is as far away as ever. The manner in which technological advances are made in a highly competitive commercial field can sometimes actually handicap the user, who would prefer not to repeatedly have to update working code to fit in with the requirements of new software packages. This acts against portability and encourages duplication and individuality. Furthermore it does seem to be the case that researchers new to the modelling field or changing direction within the accelerator community often start from base level, unaware until later of what is available, and what is already known.

Simulation codes are notoriously difficult to understand, so that it may actually be the case that one may only trust the predictions when one has carried out the programming oneself. Codes developed by others then become tools for comparison and a means of generating confidence, both of which are vitally important, and too often overlooked. Validation must be carried out with a well-defined set of benchmarks, both theoretical and practical. In this respect the TRACK codes have been used to model ISIS and to predict some of the changes which led to clear improvements being made. It is because of this that there is confidence in the proposals for the dual-harmonic upgrade and in the modelling carried out for ESS.

REFERENCES

1. M. Berz: *COSY INFINITY, An Arbitrary Order General Purpose Optics Code.* Computer Codes and the Linear Accelerator Community, Los Alamos LA-11857-C:137, 1990.

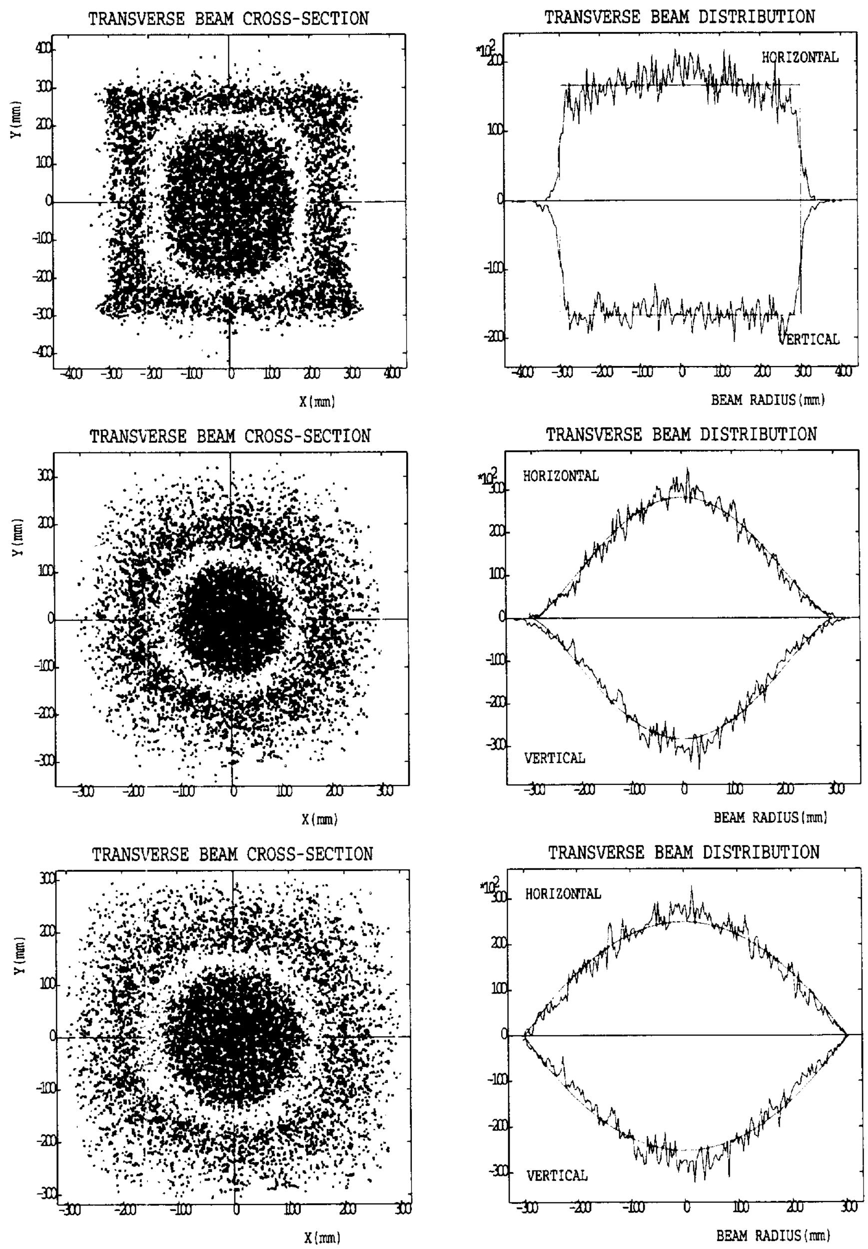

FIGURE 7. Transverse beam plots from TRACK2D showing transformation of an initially Gaussian beam using an octupole-dodecapole focusing system into a uniform, square distribution (top), a 2D parabolic distribution (centre) and a 2D elliptic distribution (bottom).

2. A.J. Dragt: *MARYLIE. A Program for Charged Particle Beam Transport Based on Lie Algebraic Methods.* University of Maryland, USA.
3. A.J. Kinderman and J.F. Monahan: *Computer Generation of Random Variables Using the Ratio of Uniform Deviates.* ACM Transactions on Mathematical Software, Vol. 3, No. 3, 1979.
4. G. Ripken: *Non-linear Canonical Equations of Coupled Synchro-Betatron Motion and their Solution within the Framework of a Non-Linear 6-Dimensional (Symplectic) Tracking Program for Ultra-Relativistic Protons.* DESY Report 85-084, 1985.
5. F.W. Jones, G.H. Mackenzie and H. Schönauer: *ACCSIM - A program to Simulate the Accumulation of Intense Proton Beams.* Proc. 14th International Conference on High Energy Accelerators, Particle Accelerators 31, p 199, 1990.
6. B.W. Montague: *Single-Particle Dynamics - RF Acceleration.* Proceedings of the First Course of the CERN International School of Particle Accelerators, Erice, 1976.
7. A. Hofmann: *Single-beam Collective Phenomena - Longitudinal.* Proceedings of the First Course of the CERN International School of Particle Accelerators, Erice, 1976.
8. M.J.D. Powell: *Curve Fitting by Cubic Splines.* Harwell Report T.P. 307, 1967.
9. R.W. Hockney and J.W. Eastwood: *Computer Simulation Using Particles.* McGraw Hill, 1981.
10. C.R. Prior: *The Multi-Particle Code TRACK2D: A Guide for Users.* CLRC, Rutherford Appleton Laboratory, 1993.
11. C.R. Prior: *The KV-Envelope Code KVBL: A Guide for Users.* CLRC, Rutherford Appleton Laboratory, 1990.
12. G. Bauer, T. Broome, D. Filges, H. Jones, H. Lengeler, A. Letchford, G. Rees, H. Stechemesser. G. Thomas (eds): *ESS: A next Generation Neutron Source for Europe. Vol III: The ESS Technical Study.*, November 1996.
13. J.P. Duke: *Foil Stripping Temperatures in the European Spallation Source.* Proceedings of the 5th European Particle Accelerator Conference EPAC96, Sitges, Spain, 1996.
14. C.R. Prior: *The Two-dimensional Analytical Coupling Code COUPXY.* CLRC, Rutherford Appleton Laboratory, 1993.
15. A.P. Letchford: Minutes of ESS Meeting 1, Rutherford Appleton Laboratory, April 1998.
16. A. Hofmann and F. Pedersen: *Bunches with Local Elliptic Energy Distributions.* IEEE Trans. Nucl. Sci. Vol NS-26, No. 3, 1979.
17. I.M. Kapchinskij and V.V. Vladimirskij: *Limitations of Proton Beam Current in a Strong Focusing Linear Accelerator associated with Beam Space Charge.* Proc. Int. Conf. on High Energy Accelerators, CERN, Geneva, 1959.
18. I. Hofmann and J. Struckmeier: *Generalised Three-Dimensional Equations for the Emittance and Field Energy of High-Current Beams in Periodic Focusing Structures.* Particle Accelerators, Vol 21, pp 69-98. 1987.
19. C.R. Prior: *Longitudinal Injection for the ESS.* ESS-94-7-R, November 1994.
20. C. Ohmori: *Multi-Particle Tracking Study for ESS Ring.* Private communication.
21. J. Galambos, J. Holmes and D. Olsen: *Longitudinal Injection Studies for ESS and SNS.* Private communication.
22. C.R. Prior and G.H. Rees: *Multiturn Injection and Lattice Design for HIDIF.* Pro-

ceedings of the 12th International Symposium on Heavy Ion Inertial Fusion, Heidelberg, Germany. September 1997.

23. C.R. Prior: *The Computer Code MISHIF: Information for Users*, CLRC, Rutherford Appleton Laboratory, 1996.
24. C.R. Prior: *Studies of Dual harmonic Acceleration in ISIS*. Proceedings of the Twelfth Meeting of the International Collaboration on Advanced Neutron Sources, ICANS-XII, Abingdon, UK. May 1993.