CERN/MPS/LIN. 71-1
September 1971

COMPUTER SIMULATION OF SPACE-CHARGE EFFECTS IN THE LINAC-BOOSTER TRANSFER LINE

J. Morrison[*] , T.R. Sherwood

Note

This report is a revised version of the report CERN/MPS/LIN. 69-19 and replaces it. Users of the code BEAM should consult only this report for instructions on how to use BEAM.

[*] Summer Student at CERN in 1969.

1. INTRODUCTION

The purpose of the program BEAM is to simulate, taking into account space-charge effects, the motion of a proton beam during its transfer from the linac to the PSB. The transport line may consist of any required combinations of magnetic quadrupole lenses, drift spaces and RF cavities.

The program traces a single bunch of protons represented here as a limited number of particles. The space-charge forces are obtained by directly computing the interaction between all particles by an approximate method.

The main technical points of interest are as in the report (1) on the program BUNCH, from which BEAM is derived. Main points of difference are discussed in this report.

2. OUTLINE OF COMPUTATIONAL METHOD

The program simulates the dynamics of a proton beam as it passes through the various sections of a long beam line, the components of which are specified as data.

The initial dimensions of the simulated beam are deduced from measured data obtained at the end of the Linac.

The phase space coordinates of representative particles are assumed to belong to a 6-dimensional Gaussian distribution in x, x' $(= \frac{dx}{dz})$, y, y', ϕ and W at the end of the Linac, and the phase space is filled using this model. x and y are the transverse space coordinates; ϕ the phase angle with respect to the synchronous particles and W is the particle energy.

The space-charge forces are computed as follows :

a) A box is constructed which fits around the bunch in 3-dimensional real space, and this box is divided into a large number of small parallelepiped cells. The number of cells is fixed, but the dimensions of the box are variable. This arrangement is called the space-charge "cage".

3.2. METHOD OF COMPUTING THE PARTICLE MOTION

The method of moving the bunch along the transfer line is that of transferring the particles coordinates over finite distances dz (of order of component length) at a time.

The independent variable is therefore the distance z along the transfer line and all particles, at the end of a computation step, will be located onto a plane perpendicular to the line axis. Clearly, because different particles have difference velocities, they will not arrive at one section all at the same time. One keeps track of this fact by storing in the computer memory the phases of each particle relative to the reference one (centre of the bunch).

When doing a space-charge computation, the longitudinal dimension of the bunch is reconstructed by interpreting the phases as longitudinal coordinates according to :

$$\Delta z = \frac{\beta\lambda}{2\pi} \Delta\phi \quad .$$

This corresponds to switching back the independent variable from z to time (phase). The transverse coordinates are approximately taken as if all particles at the time when space-charge forces are evaluated (i.e. when the centre of the bunch crosses the section under consideration) were on the same plane.

To check this approximation, the program BEAM was run against the program BUNCH, which does an exact time integration of the motion. The results for beam envelopes were identical, although small differences in scatter diagrams were noted.

The advantage of our method of movement is that the logic is much simpler, the limitation being that the change of transverse displacement must be small over a distance of the order of the bunch length.

3.3. METHOD OF DECIDING NEXT TRANSFER LINE COMPONENT

As the program should be able to deal with an arbitrary sequence of elements, it was decided that each component of the transfer line should be read in as data. Therefore, each data card contains the component code and essential parameters. The layout of the data card is explained in § 3.9.

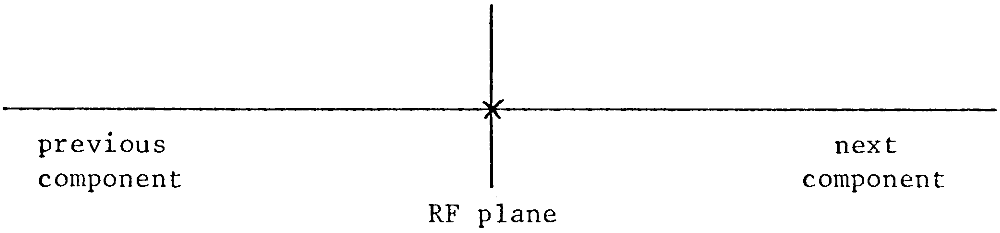
The logical sequence is :

a) The bunch centre is moved forward a distance dz at a time until it passes the end of a component; this step is then adjusted to arrive at the end of the element.

b) The main program then determines the data for the next component from the stored input values and calls the subroutine which calculates the particle motion for that element. Each component can be divided into several steps dz; at each step a space-charge calculation is carried out.

3.4. EQUATIONS OF MOTION IN AN RF CAVITY

The equations of motion are identical as those given in (1). However, the cavity is treated as a plane at which impulsive corrections are made as follows :



a) The bunch is moved to the plane of the cavity.

b) Impulsive corrections to velocities are made according to the following FORTRAN type of replacement expressions *

$$X' = X' + \Delta X'$$
$$Y' = Y' + \Delta Y'$$
$$W = W + \Delta W$$

where $\Delta X'$, $\Delta Y'$ are given as in (1) and ΔW is function of the particle phase and of the cavity voltage.

* Here, and in sects, 3.5 and 3.6 use is made of 'equations' which must be interpreted like FORTRAN replacement expressions. These are numbered F(n) to avoid confusion with genuine algebraic expressions.

c) The bunch then moves to the next component.

The particles are assumed to stay in position during this impulse and hence no recalculation of position and space-charge acceleration is made at the cavity plane.

3.5. EQUATIONS OF MOTION IN A DRIFT SPACE

The formulae used are :

$$\Delta W_{long} = F_{lav.} \cdot dz$$

where F_{lav} is average longitudinal force over the step dz.

$$F_{lav.} = 0.5 \text{ m } (acc(z + dz) + acc(z))$$

and acc(z) is longitudinal space charge acceleration at z etc.

$$\Delta r' = \frac{dz}{mv} (F_r - r'F_z)$$

where r' is the x or y velocity and v is the longitudinal velocity of each particle.

F(i) $VZI = VZ$

F(ii) $VX = VX + \frac{1}{2} . \frac{DZ}{VZI} . (AX - AZ . \frac{VX}{VZI})$

F(iii) $VY = VY + \frac{1}{2} . \frac{DZ}{VZI} . (AY - AZ . \frac{VY}{VZI})$

F(iv) $VZ = SQRT (VZ^2 + AZ.DZ)$

F(v) $X = X + \frac{DZ}{VZI} . VX$

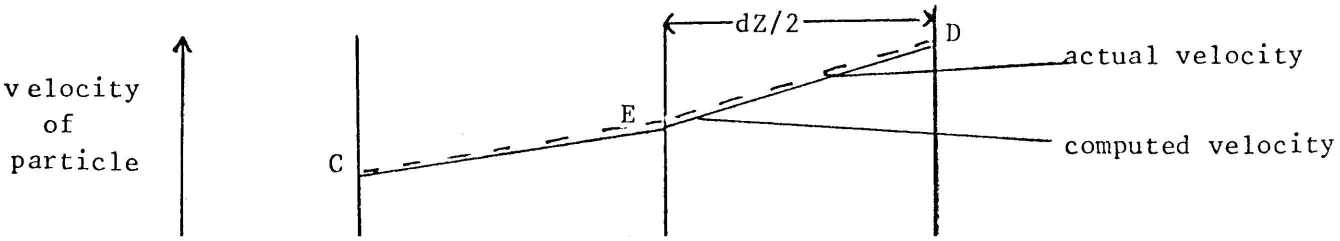
F(vi) $Y = Y + \frac{DZ}{VZI} . VY$

F(vii) $\Phi = \Phi + D\Phi$
where $D\Phi - 2\pi .f.DZ(1./VZI - 1./V)$

V is the velocity of bunch centre and f is the RF frequency of the linac.

Symbols used : VXI or VX is x velocity, AX is x component of space-charge force and similarly for the other coordinates.

The computation is carried out as follows :



(1) The operations (i) to (iv) above are carried out once during which the velocities of all particles are calculated at E using the space charge forces at C.

(2) The coordinates X, Y, ϕ are transferred from C to D using the velocities at E, assumed as average velocities for the step dZ.

(3) New space charge forces are calculated at D using the above new positions.

(4) The velocities are now transferred from E to D using the accelerations computed at D.

Using the above method both position and velocity are calculated to second order of accuracy.

3.6. EQUATIONS OF MOTION IN A QUADRUPOLE LENS

For a quadrupole lens, the formulae used in (1) apply, the independent variable being changed from t to Z.

The transfer of longitudinal coordinates is as in previous sections. Radially one has after a step dZ in a lens focussing in S and defocussing in Y :

F(viii) $XI = X$

F(ix) $YI = Y$

F(x) $X = XI.\cos(k.dZ) + X'.\sin(k.dZ)/k + \alpha_x \left\{(1-\cos(k.dZ)\right\}$

F(xi) $X' = -XI.k \sin(k.dZ) + X'.\cos(k.dZ) + \alpha_x\ k \sin(k.dZ)$

F(xii) $Y = YI. \cosh(k.dZ) + Y'.\sinh(k.dZ)/k + \alpha_y \left\{\cosh(k.dZ)\right\}$

F(xiii) $Y' = YI.k.\sinh(k.dZ) + Y'.\cosh(k.dZ) + \alpha_y\ \sinh(k.dZ)$

where :

$$k^2 = qG/m_o\gamma v \quad , \quad \alpha_x = A_x/(kv)^2 \quad , \quad \alpha_y = A_y/(kv)^2$$

and

q is elementary charge

G is lens gradient

v is (average) longitudinal velocity

A_x is the x-component of the space charge force, assumed constant over the step dZ

A_y is defined similarly

γ is $(W+W_o)/W_o$ where W_o is the rest mass energy.

(II) XOUT

This subroutine decides, depending on the step in the calculation, whether any graphical output should be printed. This is controlled by a data card by KPRNT(NN) (§ 4).

(III) GRAPH

This routine plots scatter-diagrams of 7 types depending on subroutine XOUT.

b) Taped Output

As the program can run for up to 40 minutes for 100 m of beam transport line, it was decided to have the extra option of writing all the relevant res on tape at every step of the calculation. These tapes can then be processed at a later date. (It is possible to recommence a calculation from the last record of such a summary tape).

4. FORMAT OF DATA CARDS

The basic input of data is as in (1). The main differences are :

-1-

```
 999 READ 1000, JOBNUM, IFOC, FREQ, INDEX, NOPT, ZTERM, TSTOP,ZERODE
1000 FORMAT (2I5, F10.0, 2I5, 3F10.0).
```

JOBNUM

If JOBNUM = 0 the job terminates
= 1 , only uses data from the input file, no summary tape is written.
= 2 , uses data from input files and summary tape written.
= 3 , data from summary tape (TAPE6) read as well as data from input file, no summary tape is written.
= 4 , as for 3 but with a summary tape.

IFOC

IFOC = 0 or 1, 1 if run from a FOCUS remote terminal.

NOPT

is the number of the output option the user wishes. These are :

1) Standard output

In addition one can have :

2) Beam envelopes in x,x',y,y'

3) Histograms in ϕ, x space

4) Histograms and envelopes

The option NOPT is fixed for the complete line.

INDEX

is the number of components in the transfer line.

ZREF

this is the starting position and is always zero except when JOBNUM if 3 or 4 and a predetermined start is made from tape.

```
-2-           READ 1005, ((C(I,L),I = 1,5), K = 1, INDEX)
        1005  FORMAT (5F10.4)
```

This is the reading in of the transfer line parameters. Each card contains a complete set of data for each component. The fields are :

KODE	ZEND	RBORE	NSTEP	QMG ECAV

KODE

is the code number of the element namely :

2.0 is drift

3.0 is buncher, debuncher

4.0 is lens

5.0 is bore radius change.

ZEND

is the distance of the end of the component from the origin of the line in metres.

(1) The operations (i) to (iv) above are carried out once during which the velocities of all particles are calculated at E using the space charge forces at C.

(2) The coordinates X,Y,ϕ are transferred from C to D using the velocities at E, assumed as average velocities for the step dZ.

(3) New space charge forces are calculated at D using the above new positions.

(4) The velocities are now transferred from E to D using the accelerations computed at D.

Using the above method both position and velocity are calculated to second order of accuracy.

3.6. EQUATIONS OF MOTION IN A QUADRUPOLE LENS

For a quadrupole lens, the formulae used in (1) apply, the independent variable being changed from t to Z.

The transfer of longitudinal coordinates is as in previous sections. Radially one has after a step dZ in a lens focussing in S and defocussing in Y :

F(viii) $XI = X$

F(ix) $YI = Y$

F(x) $X = XI.\cos(k.dZ) + X'.\sin(k.dZ)/k + \alpha_x\left\{(1-\cos(k.dZ)\right\}$

F(xi) $X' = -XI.k\ \sin(k.dZ) + X'.\cos(k.dZ) + \alpha_x\ k\ \sin(k.dZ)$

F(xii) $Y = YI.\ \cosh(k.dZ) + Y'.\sinh(k.dZ)/k + \alpha_y\left\{\cosh(k.dZ)\right\}$

F(xiii) $Y' = YI.k.\sinh(k.dZ) + Y'.\cosh(k.dZ) + \alpha_y\ \sinh(k.dZ)$

where :

$$k^2 = qG/m_o\gamma v\ ,\quad \alpha_x = A_x/(kv)^2\ ,\quad \alpha_y = A_y/(kv)^2$$

and q is elementary charge

G is lens gradient

v is (average) longitudinal velocity

A_x is the x-component of the space charge force, assumed constant over the step dZ

A_y is defined similarly

γ is $(W+W_o)/W_o$ where W_o is the rest mass energy.

The accuracy of this method of computing space-charge forces depends on how efficiently the cage lattice is used. That is, how many of the cells of the cage have particles in them (current number of cells is 25x25x25). However, if the cage size fits too tightly to the bunch, then as the bunch dimensions grow the cage size has to be increased too often. Each time the cage size is altered the space-charge tables have to be recalculated. If too many changes are made, the advantage of having a pre-calculated table is lost. Some compromise must be made between computing accuracy and speed.

Initially the cage size is fixed at 2.5 standard deviations. The following subroutines are used in determining cage size :

1) PNEW

To avoid excessive number of calculations of the space-charge forces tables, the permissible sizes of the cage are obtained by steps of 1 cm in x,y. ϕ increases are obtained by multiplying the previous limit by the factor 1.5. PNEW stores the previous cage dimensions and compares them with new bunch dimensions returned by COVAR. If the new values are large, of course the cage size must be increased, but if smaller, PNEW decides if it is worthwhile decreasing the dimensions. At present the criterion for altering the previous dimensions is if the new dimensions are less than half the old values. After the cage has been changed a tests is made to determine if all particles are inside. Further increases are made to ensure this. After a cage dimension change, a flag ICHANGE is set for the routine ACCEL which computes the force tables.

Particles can be lost if they fall outside a circle corresponding to the bore,RBORE, of a cylindrical vacuum pipe. The variable RBORE can be altered by inserting a (5. RBORE) card at the required placed in the set of element data cards. The maximum cage size for ϕ is 2π .

2) COVAR

This routine calculates the covariance matrix of the particles in the sample.

3) AV

AV calculates the means of the sample and also limits the range of ϕ to 2π . If a value of ϕ exceeds 2π , then 2π is added or subtracted as required. This has the effect that if ϕ exceeds 2π, the particle appears to belong to a neighbouring bunch.

3.8. OUTPUT ROUTINES

a) General

The output from the program is partly in graphical form and is produced by subroutine GRAPH which gives 2-dimensional plots of projections of the 6-dimensional phase space. In addition, printed output of histograms, beam envelopes, and taped results can be obtained by punching the correct code number on a data card. The various routines in the programs are :

(I) OUTOPT

This routine manages the other routines and is controlled by the code number. (At each step the following statistical analysis if performed. Six standard deviations and the 3 correlation coefficients for the (x,x'), (y,y') and (z,z') projections are printed. The population in the five projections (x,x'), (y,y'), (z,z'), (x,z) and (y,z) are tested against the hypothesis that they are from the bivariate Gaussian population. The test is made by first calculating the second moments, constructing a set of confocal ellipses adjusted so that each annulus would contain 10% of the population if the distributions were normal and then making a χ^2-test for equality of number of particles in each annulus. Of course any other distribution which resulted in a redistribution of particles within a given annulus would also satisfy the test. Visual examination of the scatter-plots would reveal any gross departure from a normal distribution. If the test is passed, the areas of the projections corresponding to the standard deviations are calculated and printed. If the test (at 10% significance level) is failed, the phase-space area, corresponding to 95% of the particles is given instead. Also the fractions falling within the annuli are given, together with the fractions of the population within one and within two standard deviations).

(II) XOUT

This subroutine decides, depending on the step in the calculation, whether any graphical output should be printed. This is controlled by a data card by KPRNT(NN) (§ 4).

(III) GRAPH

This routine plots scatter-diagrams of 7 types depending on subroutine XOUT.

b) Taped Output

As the program can run for up to 40 minutes for 100 m of beam transport line, it was decided to have the extra option of writing all the relevant results on tape at every step of the calculation. These tapes can then be processed at a later date. (It is possible to recommence a calculation from the last record of such a summary tape).

4. FORMAT OF DATA CARDS

The basic input of data is as in (1). The main differences are :

```
-1-        999 READ 1000, JOBNUM, IFOC, FREQ, INDEX, NOPT, ZTERM, TSTOP,ZERODE
          1000 FORMAT (2I5, F10.0, 2I5, 3F10.0).
```

JOBNUM

If JOBNUM = 0 the job terminates

= 1 , only uses data from the input file, no summary tape is written.

= 2 , uses data from input files and summary tape written.

= 3 , data from summary tape (TAPE6) read as well as data from input file, no summary tape is written.

= 4 , as for 3 but with a summary tape.

IFOC

IFOC = 0 or 1, 1 if run from a FOCUS remote terminal.

NOPT

is the number of the output option the user wishes. These are :

1) Standard output

In addition one can have :

2) Beam envelopes in x,x',y,y'

3) Histograms in ϕ, x space

4) Histograms and envelopes

The option NOPT is fixed for the complete line.

INDEX

is the number of components in the transfer line.

ZREF

this is the starting position and is always zero except when JOBNUM if 3 or 4 and a predetermined start is made from tape.

-2-

```
          READ 1005, ((C(I,L),I = 1,5), K = 1, INDEX)
     1005 FORMAT (5F10.4)
```

This is the reading in of the transfer line parameters. Each card contains a complete set of data for each component. The fields are :

KODE	ZEND	RBORE	NSTEP	QMG) ECAV)

KODE

is the code number of the element namely :

2.0 is drift

3.0 is buncher, debuncher

4.0 is lens

5.0 is bore radius change.

ZEND

is the distance of the end of the component from the origin of the line in metres.

RBORE

is the radius of the vacuum pipe of the component in metres (only needed on a 5.0 card).

NSTEP

is the number of space charge computations wanted in that component.

QMG

is the strength of the lens in Tesla/m

ECAV

is the peak energy change, in MeV, possible acrcss the gap at r = 0

For a lens focussing in x and defocussing in y, QMG is positive and negative vice versa. For a drift space this field is blank.

```
-3-         READ 7715, (TITLE1(JI), JI = 1,36)
            READ 7715, (TITLE2(JI), JI = 1,36)
      7715  FORMAT (12A6)
```

These are headings for histograms in y and x respectively.

```
-4-         READ 100, ((MU(I), V(I), I = 1,6)
       100  FORMAT (2F10.6)
```

These are the means and standard deviations for initial values of ϕ , W, x, x', y, y' respectively.

c oordinate	ϕ	W	x	x'	y	y'
units	degrees	MeV	mm	mr	mm	mr

```
-5-         READ 101, ((CORR(I,J), J = 1,6), I = 1,6)
       101  FORMAT (6F10.6)
```

This is the reading of the initial correlations between the 6-variables. The order is as in -4- above.

5. FUTURE IMPROVEMENTS

The improvements which can be made to the program come into 2 classes :

i) Improvements in programming, such as using assembler language routines, faster space charge routines, and a large increase in the number of particles traced.

ii) Due to the flexibility of the program present structure, the user can easily add new subroutines. An obvious example is the inclusion of a routine which will simulate bending magnets.

ACKNOWLEDGEMENTS

We wish to record our debt to Drs. M. Martini and D.J. Warner for the suggestions they offered and for several helpful discussions.

REFERENCES

(1) F. Vermeulen, "Computer Simulation of Beam Bunching with the Inclusion of Space Charge Forces and Quadrupole Lenses". CERN, MPS/Int. LIN 67-5.

(2) W.T. Eadie, M. Martini, "Statistical Population of the Linac Beam Transverse Phase Space". CERN, MPS/Int. LIN 68-9.

Distribution (open)

L. Evans
H. Haseroth
K. Pedersen
C. Poinard
F. Sacherer
C.S. Taylor
D.J. Warner
M. Weiss