EUROPEAN LABORATORY FOR PARTICLE PHYSICS

CERN/PS 92-31 (RF)

User's Guide to ESME version 7.3 at CERN

S. Hancock J. MacLachlan S. Stahl

April 1992

Abstract

ESME is a FORTRAN program which may be used to calculate the evolution of a distribution of particles in azimuth and energy while this is acted upon by the rf system of a proton synchrotron. It enables multiple rf systems to be modelled, as well as feedback control, space-charge and many longitudinal coupling impedance effects. The capabilities of the program are described and input data requirements are specified in sufficient detail to permit significant calculations to be made by an uninitiated user. Also described is a VAX-based code management convention which has been established with a view to maintaining functional equivalence in versions used on different computers. The program currently stands at version 7.3 and has been extensively modified since the previous release was obtained from Fermilab.

Geneva, Switzerland

Preface

ESME has been at CERN in various guises since it was first imported from Fermilab in 1984 by Jim MacLachlan, one of its principal architects. The program evolved considerably in the intervening years with contributions from many authors, but these developments were not followed here and, apart from the efforts of the occasional visitor, ESME stagnated at CERN until version 7.1 was acquired in 1990. More recently, Steve Stahl visited CERN in order to convert the graphics to GKS (via HIGZ) and version 7.3 was born. Now, after a year's experience with jobs routinely tracking more than 10^7 particle.turns and after the elimination of a serious (but little-publicised) bug which affected vector execution in the IBM batch system, it is felt opportune to launch ESME v. 7.3 upon a wider CERN public.

This user's guide was originally prepared by my Fermilab co-authors and has simply been edited by me with a particular regard for ESME in the CERN environment. Chapter 1 includes an historical overview which is retained for the references it contains, while chapters 4 and 5 are unlikely to interest new users and I have not altered them. The remainder has been thoroughly revised, not merely to respect notation conventions on this side of the water or to appease my pedantry, but to correct or clarify those points which were deemed less than obvious by this "initiated" user. However, not all the multifarious options provided by the program have been tried at CERN and this documentation (which resides on the PSRF 191 disk on CERNVM as ESMEDOC TEX and, in PostScript format, as ESMEDOC PS) may be subject to amendment.

Steve Hancock

Contents

Chapter 1

Introduction

1.1 Version 7.1, February 1990

The program ESME has been developed to model those aspects of beam behaviour in a proton synchrotron that are governed by the rf systems. It follows, turn by turn, the evolution of a distribution in azimuth-energy coordinates by iterating a map corresponding to the single-particle equations of motion. The map parameters may be updated each turn to reflect the action of the beam current on the individual particles through feedback loops, space-charge, coupling impedance, etc. The code was originally developed during the years 1981–82 for the design of the Tevatron I Antiproton Source[1] and it was documented for general use in 1984[4]. That documentation pertains to ESME v. N, which was the last major revision of the code for use on Cyber machines. In 1986, provisions were made for longitudinal coupling and space-charge[5] to investigate the usefulness of a γ_{tr}-jump in the Fermilab Booster[2]. This version was adapted from the Cyber machines to the FPS-164 by Peter Lucas who, in the process, made a number of the improvements which entered into what was designated v. 6, the version that is in most common use at Fermilab. A somewhat modified VAX version, v. 6.05, was sent to Brookhaven.

The improvements made in 1987 were more or less incremental changes to the 1981 code. However, for several years there has been interest in capabilities for simultaneous operation of more than two independent rf systems, for explicit control over the separation between the accelerator reference orbit and the synchronous trajectory and for input using time units rather than beam turn number. In view of this interest and because the manner in which the various rf phase angles were defined and controlled had proved confusing to some users, a major revision of the code was started in 1987. Although nearly all functions have been retained and data requirements are similar, the new code will not work with data prepared for earlier versions. The conceptual basis for the new version is discussed elsewhere[10]. The new version, dubbed v. 7.1, will probably be less stable and more subject to bugs than v. 6, at least for the time it takes to accumulate experience from a reasonable variety of applications. Nonetheless, the new version is fundamentally improved and there is no intention to maintain both.

[1] Instructive examples of the capabilities of ESME can be found in reports relating to TeV I. Some of these are not cited directly in the text but are included in the references. See for example refs. [1], [2] and [3].

[2] There are examples of the calculation of collective effects resulting from the Booster studies. See refs. [6], [7], [8] and [9].

1.2 Version 7.3, April 1991

In the course of the past year ESME has seen significant use. A number of bugs have been removed and some refinements made. The improvements and corrections described in ref. [14] bring ESME up to version 7.2 and the relevant changes have been incorporated in this edition of the user's guide.

The portability of the graphics has continued to be a concern of the authors. Version 7.2 of ESME used DI-3000 and GRAFMAKER[3]. The current incarnation of ESME uses HIGZ[15], a high-level graphics interface. HIGZ was developed at CERN with GKS in mind, but has also been interfaced to DI-3000. Thus, sites with either graphics package need only to obtain HIGZ to be able to run ESME without substantial programming effort. The current version of ESME, which uses HIGZ, is v. 7.3.

1.3 Applications

ESME is frequently used to assess the efficiency of a given rf beam manipulation or to optimize certain system parameters. For this task, the user must specify fully the technical details of various subsystems and derive numerical measures of system performance from the particle distribution. Thus, many input data are required and the program must include numerical analysis features. Equally useful, however, are qualitative calculations designed to illustrate a concept or explore the feasibility of a novel approach. For such applications, the code should require a minimum of system-specific data and provide easy access to a variety of graphics output. When a qualitative investigation has been fruitful, it is natural to proceed in stages of increasing realism and detail to a thorough modelling of the process. ESME is intended to serve effectively over a wide range of problem specificity by providing distinct functions so that each may be invoked only as required and by establishing reasonable defaults so that generic systems can be represented by few data. Thus, a few lines of data may permit a first glimpse at a system which can then be studied in greater detail by overriding defaults with specific input and by invoking additional functions such as, for example, those related to collective behaviour or those related to the numerical evaluation of the properties of the distribution.

1.4 The Coordinate System

The basic coordinates used by ESME are the azimuthal position of the particles in the machine and the difference between their energy and the synchronous energy at the time when the rf is passing through its synchronous phase. The origin in azimuth is at the location where the rf voltage is applied:

$$-180^\circ \leq \Theta_{i,n} \leq 180^\circ$$

where i is the particle index and n is the turn number. Θ is measured positive clockwise, whereas the modelled beam circulates in the opposite, negative Θ-direction. Typically, a well-behaved bunch will be centred near $\Theta = 0^\circ \bmod 180^\circ/h$. When there are h equivalent bunches, it is generally efficient to process a single bunch by setting periodic boundaries at $\Theta = \pm 180^\circ/h$.

The energy unit is MeV, while azimuth is expressed in radians internally but in degrees for input and output. The basic time unit of the difference equations is turn number, but all input specification of parameter time dependence is made using seconds.

[3] DI-3000 and GRAFMAKER are trademarks of Precision Visuals, Inc.

One of the difficulties when using a general program for a wide range of problems is being able to specify the phases of all the rf systems and their desired dependence on time or on bunch centroid position. In version 7.3 of ESME, where several systems with different harmonic numbers may be active simultaneously, the phase of each system is defined in terms of its own frequency at the synchronous time, i.e., the time when a particle acted upon by all rf systems receives the energy matched to the specified energy and/or radial position change for the next turn. Options are available which cause the phases to be controlled by an input program, to follow the time derivative of the magnetic field, or to maintain a system as a Landau cavity, etc. Many common options have been provided and the program structure would easily accommodate any other scheme which one is willing to specify in a FORTRAN subroutine.

1.5 Difference Equations

The basis of the program is the pair of single-particle difference equations

$$\begin{aligned} \vartheta_{i,n} &= \left[\frac{\tau_{s,n-1}}{\tau_{s,n}} \vartheta_{i,n-1} + 2\pi \frac{\tau_{i,n}}{\tau_{s,n}} - \pi \right]_{mod(2\pi)} + \pi \\ E_{i,n} &= E_{i,n-1} + eV(\varphi_{s,n} + h\vartheta_{i,n}) - eV(\varphi_{s,n}) \end{aligned}$$

giving the azimuth and energy, respectively, of the i-th particle at the end of the n-th turn of the synchronous particle. The variable $E_{i,n}$ is the energy difference with respect to the synchronous particle. The program ignores a slightly subtle distinction between $\vartheta_{i,n}$, the azimuthal variable in the mapping, and the periodic spatial variable $-\pi \leq \Theta_{i,n} \leq \pi$, which differ by an amount which is generally of no practical importance[4]. The conversion from the variable of the mapping to the true azimuth, $\Theta_{i,n}$, of the i-th particle is

$$\Theta_{i,n} = \frac{\tau_{s,n}}{\tau_{i,n}} \vartheta_{i,n} \approx \left(1 + \eta \frac{\Delta p_{i,n}}{p_{s,n}} \right) \vartheta_{i,n}$$

In most applications for a high-energy synchrotron, $\eta = \gamma_{tr}^{-2} - \gamma_{s,n}^{-2}$ is small compared to unity. However, given that there might be some case for which the distinction between ϑ and Θ is important, there would be no difficulty in modifying ESME to use $\Theta_{i,n}$. The change is not made in the general release because it adds to both cpu and memory requirements without practical benefit for any anticipated use.

The relation between the revolution period, $\tau_{s,n}$, of the synchronous particle when making its n-th turn and that, $\tau_{i,n}$, of the i-th particle is treated exactly. Hence the kinematic non-linearity is treated exactly. This can be very important if the synchronous energy is close to the transition energy. The lattice non-linearity is expressed as the dependence of γ_{tr} upon the relative momentum difference, $\Delta p_{i,n}/p_{s,n}$, between a general particle and the synchronous one. The rf potential is the sum of one or more sinusoidal terms so that the dynamic non-linearity of a simple waveform is treated exactly, while other forms of potential may be treated as a Fourier expansion of up to ten terms[5]. The equation employed in the program is generalized somewhat to permit the option of

[4] The difference equations given here are not precisely those given in ref. [10] firstly, because the sense of particle motion is in the negative ϑ-direction here and in the positive Θ-direction in ref. [10] and, secondly, because the equations have been slightly modified in order to conserve rigorously longitudinal phase space density for all beam energies. A detailed derivation of the difference equations will be given in a separate Fermilab note.

[5] A little-used option mentioned in ref. [4] which provided a perfectly linear sawtooth waveform has been dropped because of a severe conflict with the mechanism by which the new code finds the synchronous phase.

multiple, evenly-spaced cavities and to allow the drift between cavities to be subdivided for more frequent application of the space-charge kicks if the dynamics require it.

The treatment starts from the specification of a reference orbit of average radius R_{eq} on which the mean dipole magnetic field $<B_z>$ is known. A particle which would follow the reference orbit has the reference momentum $p_o = 299.792458 <B_z> R_{eq}$, with p_o expressed in MeV/c, R_{eq} in metres and $<B_z>$ in Tesla. The angular frequency of beam circulation on this orbit is $\Omega_o = \beta_o c/R_{eq}$, where $\beta_o c$ is the speed of a particle with the reference momentum. The variation of the guide field away from the reference orbit is, for the purpose of describing longitudinal motion, completely characterized by the momentum dependence (at fixed $<B_z>$) of the transition energy. The minimum one need specify is γ_{tr} = constant, an adequate choice in many instances. In many cases, it is also correct to identify the radius of the synchronous trajectory, R_s, with the reference orbit, R_{eq}. However, in applications like stacking or displacement acceleration, where the synchronous trajectory may be radially offset or even outside the beampipe, it is necessary to take explicit account of the difference between them. ESME calculates all motion relative to a hypothetical synchronous particle. The synchronous frequency is calculated from the synchronous momentum and R_s. The momentum compaction factor, α, and the magnetic field at the synchronous radius are determined from the values provided on the reference orbit.

Two versions of the difference equations are available in the main tracking routine. One which calculates $\Delta\vartheta$ per turn as described above and another which uses the approximation $2\pi\eta\Delta p_{i,n}/p_{s,n}$. Substantial time can be saved by using the simpler algorithm when this is sufficient. Significant savings in execution time could be expected if the main tracking loop were rewritten to use only this simpler form of the difference equations while eliminating branches to unused portions of the code.

1.6 Multiparticle Calculations

The particle distribution at the end of each turn is calculated from that at the end of the previous one by applying a single-turn map to each particle independently. The program then optionally performs three general types of calculation using the properties of the distribution as input. The most common calculations are those which quantify the distribution so that its properties can be plotted as functions of time. Examples of such properties are first and second moments, rms emittance, equivalent matched emittance and the Fourier spectrum of the beam current, etc. Another type of calculation is the computation of feedback contributions to rf system parameters. ESME provides options for phase feedback and for feedback to the rf amplitude. The third general class of collective calculation is the evaluation of beam-induced voltages due to space-charge and longitudinal coupling impedance. The space-charge calculation is based on a constant geometric factor relating average beam size and average beampipe radius. The longitudinal impedance can be characterized by an arbitrary table of real and imaginary parts vs. frequency and/or by a table of resonances, each defined by its resonant frequency, quality factor, Q, and real part at resonance. This last type of calculation was introduced to investigate the effect of beam intensity upon single-bunch behaviour at transition. Formerly, the harmonic response of a resonator was treated in the steady-state approximation only. However, since such an approach does not realistically represent the driving terms for certain types of collective motion, the response of a high-Q resonator may now be modelled in the time domain as well[10]. Ref. [12] describes the use of this facility to model coupled-bunch instabilities.

Chapter 2

Data Requirements

Data for ESME are generally acquired by NAMELIST reads dispersed among subroutines which, as much as possible, segregate different program functions. The subroutine that reads a particular class of data will only be called if the input data include the single-character command invoking the related program function. All ESME commands and their associated NAMELIST data are listed in this chapter and the function of each input datum is briefly described.

The data pertaining to each program function or accelerator subsystem are stored in different FORTRAN COMMON blocks. These blocks are initialized with values and switches which allow the program to run using few input data. Default values exist even for some quantities which are nearly always problem-specific, so that program execution may proceed far enough to expose more than one data error in a single test run. Input values are retained unaltered[1], so that a command may be re-issued without re-entering data that remain the same.

2.1 Input Structure

In this section the required format of the various input files is described. First and foremost is the data file that is assigned to FORTRAN logical unit 5. This is the principal, and often the only input to ESME. In addition, ESME may read in a table of values of the rf voltage, frequency or phase as a function of time, of the wall impedance as a function of frequency, or of the parameters of a number of high-Q resonators.

2.1.1 The Data File

A command is issued to ESME by entering the appropriate single character in the first column of a line in the data file. The next four columns are ignored and the rest of the line is read[2] and is echoed in the printed output as a comment associated with the command. Such comments are optional and also serve to annotate the dataset itself. Many commands are followed by a NAMELIST read. The NAMELIST input is specified by entering a dollar sign in the second column followed immediately by the NAMELIST identifier and a blank space, after which any variable which is a member of the corresponding NAMELIST block may be given a value by entering its name followed by an equals sign and the value to be assigned. All such data items should be separated by commas and the NAMELIST input terminated by "$END". Since NAMELIST is a FORTRAN extension, the required syntax may vary.

[1] There are two minor exceptions; see the O- and T-commands.

[2] FORTRAN format `(A1,4X,A68)`.

Exceptions to the general rule that NAMELIST reads procure the input from the data file can occur if a command (see section 2.2.16) is issued to call a user-written subroutine which employs a different method. Also, the title for graphics output, if supplied, follows the NAMELIST block of an O-command as a plain character string.

2.1.2 The RF Tables

When the time dependence of, for example, the rf voltage is rather involved, it may be simplest to provide ESME with a table from which the value of the voltage at a particular time can be interpolated. Such a table is read in from the file specified in the line immediately following the NAMELIST input of an A-command in which the voltage, frequency and/or phase variation of any of the rf sources is declared to be of type 4 (see section 2.2.3). The file is assigned to logical unit 10 and is read[3] to find a line containing "SOURCE i", where i is the index of the rf source to which the subsequent (list-directed) data pertain. The required format of the tables is illustrated by the following:

SOURCE i
VOLTAGE
n_v^i
t_1^i, v_1^i
t_2^i, v_2^i
...
$t_{n_v^i}^i, v_{n_v^i}^i$
FREQUENCY
n_f^i
t_1^i, f_1^i
t_2^i, f_2^i
...
$t_{n_f^i}^i, f_{n_f^i}^i$
PHASE
n_p^i
t_1^i, ϕ_1^i
t_2^i, ϕ_2^i
...
$t_{n_p^i}^i, \phi_{n_p^i}^i$
SOURCE j
...

In the case of more than one rf source, their order of appearance in the file is arbitrary. However, the data in each table must be entered in order of increasing time (in seconds), i.e., $t_1^i < t_2^i < \ldots < t_{n^i}^i$. It is not necessary to supply all three tables for each source, nor is it necessary to enter them in one place under a single "SOURCE" heading. However, if they do appear in one place, they must do so in the order voltage, frequency, phase. Only one file of rf tables may be specified in a given program run and its entire contents are read and stored at one time. Consequently, even those tables which are not used until later in the run must be included in the data at the time when the file is read.

It is important to note that the data required in the tables are <u>not</u> absolute values of voltage, frequency, or phase, but are dimensionless fractions of the variation of those quantities between the

[3]FORTRAN format (A6,I4).

initial and final values that are specified in the A-command. Thus, in general, a table should run from 0.0 to 1.0. For example, if the rf voltage of the i-th source is specified to change from VI(i) at time TVBEG(i) to VF(i) at time TVEND(i) with KURVE(i) = 4, then, at some general time t such that TVBEG(i) $\leq t \leq$ TVEND(i), the voltage would be calculated as VI(i) + (VF(i) − VI(i)) $*\, \Delta V$, where ΔV is the value obtained by cubic spline interpolation in the appropriate "voltage" table. At times $t <$ TVBEG(i) or $t >$ TVEND(i), no interpolation would be performed and the voltage would be taken as VI(i) or VF(i), respectively.

2.1.3 The Impedance Table

The space-charge command (see section 2.2.14) provides the option to enter a table of cubic polynomial coefficients for the wall impedance. Such a table is read in from the file specified in the line immediately following the NAMELIST input of a B-command in which NZ is declared to be greater than zero. The file is assigned to logical unit 11 and NZ frequency values are read, each of these together with eight polynomial coefficients. The required format is illustrated by the following:

f_1
$x_{1,1}, x_{1,2}, x_{1,3}, x_{1,4}$
$y_{1,1}, y_{1,2}, y_{1,3}, y_{1,4}$
f_2
...
f_{NZ}
$x_{\mathrm{NZ},1}, x_{\mathrm{NZ},2}, x_{\mathrm{NZ},3}, x_{\mathrm{NZ},4}$
$y_{\mathrm{NZ},1}, y_{\mathrm{NZ},2}, y_{\mathrm{NZ},3}, y_{\mathrm{NZ},4}$

The coefficients x and y are the values of the real and imaginary parts of the impedance, respectively. Since the corresponding FORTRAN READ statement is list-directed, it is sufficient simply to list the values separated by commas or spaces in the file. However, the entries must be made in order of increasing frequency (in MHz), i.e., $f_1 < f_2 < \ldots < f_{\mathrm{NZ}}$. The coefficients are used to compute the value of the impedance, Z, at some general frequency, f, according to:

$$Z = \begin{cases} x_{1,1} + iy_{1,1}\,, & f < f_1 \\ x_{n,1} + x_{n,2}\Delta f + x_{n,3}\Delta f^2 + x_{n,4}\Delta f^3 & \\ \quad + i(y_{n,1} + y_{n,2}\Delta f + y_{n,3}\Delta f^2 + y_{n,4}\Delta f^3)\,, & f_1 \leq f < f_{\mathrm{NZ}} \\ x_{\mathrm{NZ},1} + iy_{\mathrm{NZ},1}\,, & f \geq f_{\mathrm{NZ}} \end{cases}$$

where $\Delta f = f - f_n$ with $f_n = \max f_i \in f_i < f$ for $1 \leq i <$ NZ.

2.1.4 The Resonance Table

Narrow (high-Q) resonances may also be included in the impedance. The parameters of each resonator are read in from the file specified in the line immediately following the NAMELIST input of a B-command in which NR is declared to be greater than zero (or, if NZ > 0, immediately following the specification of the impedance file). The file is assigned to logical unit 12 and NR resonant frequency values (in MHz) are read, each of these together with the corresponding resonant impedance (in Ω), multiplicity and Q-value. The multiplicity value simply multiplies the impedance that is calculated at some general frequency. Each of the NR entries in the resonance file should occupy a single line. The corresponding READ statement is list-directed with only real values expected.

2.2 Commands

ESME is a so-called "data-driven" program; the course of program execution is governed by the order of the commands in the input data. The data acquired, or the process initiated by each command pertain to a separate subsystem of the accelerator or to a distinct phase of the calculation. The following commands will appear in almost every input dataset:

R Read in the lattice (**R**ing) parameters, magnetic field, energies, etc. of the NAMELIST block /RING/.

A Read in the rf (**A**cceleration) parameters of the NAMELIST block /RF/.

P **P**opulate phase space with the initial distribution that is described by parameters read into the NAMELIST block /POPUL8/.

O Select graphics **O**utput options using parameters in the NAMELIST block /GRAPH/.

T **T**rack the distribution according to parameters read into the NAMELIST block /CYCLE/.

Q Stop program execution, i.e., **Q**uit.

These are listed in the order in which they are generally used and commands whose order is somewhat less than arbitrary are underlined. Significant calculations may be made using these commands alone. Additional output options are provided by:

D Output (**D**isplay) graphics.

H Select the quantities to be plotted from **H**istory records using parameters in the NAMELIST block /HISTRY/.

M Request that histograms of the azimuthal distribution be saved for construction of a **M**ountain-range plot.

N Plot mountain-range data.

W **W**rite a comment into the printed output.

Special calculations, including ones prepared by the user, are initiated by the following commands:

L Activate **L**ow-level beam control (feedback, transition phase jump, etc.) according to parameters read into the NAMELIST block /LLRF/.

B Read in the **B**eam parameters of the NAMELIST block /SCHG/ and set up the space-charge calculation.

F Set up a fast **F**ourier transform of the azimuthal distribution according to parameters read into the NAMELIST block /FFT/.

0-9 Call the subroutine SHAZAM at the entry point SHAZAM, SHAZAM**1**, SHAZAM**2**,... to manipulate any quantity in COMMON storage.

Finally, there exists a pair of commands to save and restore the entire state of a calculation:

S **S**ave tracking and control data.

G **G**et the tracking and control data of a previous run.

These enable a long calculation to be performed in stages, or several calculations which share a common intermediate point all to proceed from the same conditions at that point.

The command names could arguably have been chosen according to a better set of mnemonics than that illustrated by boldface type in the above. The names are retained for consistency with earlier versions of the program.

2.2.1 Command Ordering

The R-command, which initiates the input of the basic lattice parameters and energy scale, should generally appear before any other because several commands require this information to perform their own functions. The A-command, which brings in the rf parameters, is usually second. The P-command, which establishes the initial phase space distribution, normally needs to be preceded by both. While the B-command, which sets up the machinery for longitudinal impedance calculations, needs the initial distribution in order to obtain the initial beam current distribution.

Although it is possible to create valid datasets which do not respect the typical command ordering, it is generally safer to adhere to the order R, A, P, B for the first appearance of these commands. Other commands can usually be ordered arbitrarily.

2.2.2 The R-command – Lattice Parameters

The members of NAMELIST /RING/ are stored in COMMON /RINGP/. This NAMELIST read is made in the subroutine RINGPAR, which derives quantities like η, γ_{tr}, etc. which depend only on the lattice parameters and the reference orbit. ESME distinguishes the synchronous trajectory from the reference one and it is possible to separate them physically by specifying an energy difference between them.

R-command, NAMELIST /RING/

Variable	Default Value	Default Unit	Description
REQ	None	m	Radius of the reference (central) orbit.
GAMMAT	None	-	Transition value of γ on the reference orbit.
ALPHA1	0.0	-	Coefficient of $(\Delta p/p)^2$ in series expansion for $\Delta R/R$ about reference orbit.
ALPHA2	0.0	-	Coefficient of $(\Delta p/p)^3$ in series expansion for $\Delta R/R$ about reference orbit.
ALPHA3	0.0	-	Coefficient of $(\Delta p/p)^4$ in series expansion for $\Delta R/R$ about reference orbit.
EK0I	None	MeV	Kinetic energy on the central orbit at time TI.
EK0F	0.0	MeV	Kinetic energy on the central orbit at time TF.
TI	0.0	s	Time at which magnetic field change begins.
TF	0.0	s	Time at which magnetic field change ends.
TSTART	0.0	s	Time at which tracking begins.
FRAC	1.0	-	Determines azimuthal periodicity by restricting calculation to $-180°/\text{FRAC} \leq \vartheta \leq 180°/\text{FRAC}$.
PIPRAD	1.0	m	Radius of beampipe.
EBDRY	.F.	-	If .TRUE., puts absorbing beampipe "walls" at REQ ± PIPRAD.
DES	0.0	MeV	Energy offset of synchronous orbit relative to reference orbit.

R-command, NAMELIST /RING/ (continued)			
Variable	Default Value	Unit	Description
KURVEB	1	-	Magnetic field ramp from EK0I to EK0F: 1 – Linear 2 – Increasing parabolic 3 – Biased sinusoidal 4 – Decreasing parabolic 5 – Parabolic with gradient from EKIDOT to EKFDOT.
EKIDOT	0.0	MeV/s	Slope of parabolic ramp at time TI.
EKFDOT	0.0	MeV/s	Slope of parabolic ramp at time TF.
JNRAMP	.F.	-	If .TRUE., sets starting point of ramp to be the current value, hence smoothly piecing ramp segments together.
GMAJMP	.F.	-	If .TRUE., activates γ_{tr}-jump.
KINDG	1	-	Type of γ_{tr} variation[a]: 1 – Linear; γ_{tr} = GAMPAR(1) + GAMPAR(2) * T 2 – Decreasing exponential; γ_{tr} = GAMPAR(1) + GAMPAR(3) * $(1 - e^{-T/GAMPAR(2)})$.
GAMPAR(1:3)	0.0	-	Coefficients for γ_{tr} variation.

[a]T = 0 corresponds to the time at which the R-command is invoked with GMAJMP = .TRUE.

An example of a "minimal" R-command might be:

```
$RING REQ=100., GAMMAT=6.1, EK0I=25079., FRAC=20. $END
```

Here, the lattice is characterized simply by a radius, γ_{tr} value and a magnetic field[4] which corresponds to a reference momentum of 26 GeV/c. FRAC is useful for restricting the azimuthal range considered by ESME to a suitable period of the ring, such as a single bucket. Periodic boundaries are established at the limits of this range. Since EK0F is not specified in the above example, the default, EK0F = 0.0, indicates that the field is static. In addition, the "start" time for this run is taken to be zero.

At times $t \leq$ TI, the kinetic energy of the reference orbit is taken to be EK0I and, for $t \geq$ TF, it is EK0F. Note that PIPRAD has no effect unless EBDRY = .TRUE. and, if the γ_{tr}-jump option has been enabled, γ_{tr} is varied until another R-command is issued with GMAJMP = .FALSE.

2.2.3 The A-command – RF Parameters

The members of NAMELIST /RF/ are read in subroutine RFPROG and are stored in COMMON /RFP/. Up to ten independent rf sources may be specified.

A-command, NAMELIST /RF/			
Variable	Default Value	Unit	Description
NRF	1	-	Number of rf sources.
H(i), i = 1:10	1	-	Harmonic number of source i. (Integer.)

[4]An explicit consideration of magnetic fields is not part of the program; rather, an average guide field is assumed.

A-command, NAMELIST /RF/ (continued)			
Variable	Default Value	Unit	Description
HW(i)	1	-	Limits source i to be "active" for $-180°/HW \leq \vartheta \leq 180°/HW$. (Integer.)
ISYNC	0	-	Synchronism condition to be imposed: 0 – None; voltages and phases remain as programmed 1 – Phase of rf waveform shifted to synchronous, stable point 2 – Magnitude of rf waveform scaled to give correct synchronous energy gain 3 – Source 2 acts as a Landau cavity to source 1[a].
EXCHRF	.T.	-	If .TRUE., sources are interchanged so that source 1 is always the greatest contributor to the bucket height.
VI(i)	0.0	MV	(Peak) voltage of source i at time TVBEG(i).
VF(i)	0.0	MV	(Peak) voltage of source i at time TVEND(i).
TVBEG(i)	0.0	s	Time at which rf voltage change begins.
TVEND(i)	0.0	s	Time at which rf voltage change ends.
KURVE(i)	0	-	Type of rf voltage variation between times TVBEG(i) and TVEND(i) for source i: 0 – None; voltage maintained at VI(i) 1 – Linear 2 – Isoadiabatic 3 – Sigmoid 4 – Cubic spline interpolation[b].
VKON	.T.	-	If .TRUE., activates voltage curves for all sources.
PSII(i)	0.0	deg	Phase of source i at time TPBEG(i).
PSIF(i)	0.0	deg	Phase of source i at time TPEND(i).
TPBEG(i)	0.0	s	Time at which rf phase change begins.
TPEND(i)	0.0	s	Time at which rf phase change ends.
KURVP(i)	0	-	Type of rf phase variation between times TPBEG(i) and TPEND(i) for source i: 0 – None; phase maintained at PSII(i) 1 – Linear 2 – Quadratic 4 – Cubic spline interpolation[b].
PHKON	.F.	-	If .TRUE., activates phase curves for all sources.
FRI(i)	0.0	MHz	Frequency of source i at time TFBEG(i).
FRF(i)	0.0	MHz	Frequency of source i at time TFEND(i).
TFBEG(i)	0.0	s	Time at which rf frequency change begins.
TFEND(i)	0.0	s	Time at which rf frequency change ends.

[a]Synchronism is only assured for sources 1 and 2.
[b]Fit to values read from file. See section 2.1.2.

A-command, NAMELIST /RF/ (continued)			
Variable	Default Value	Unit	Description
KURVF(i)	0	-	Type of rf frequency variation between times TFBEG(i) and TFEND(i) for source i: 0 – None; frequency maintained at FRI(i) 1 – Linear 2 – Quadratic 4 – Cubic spline interpolation[a].
FRKON	.F.	-	If .TRUE., activates frequency curves for all sources.
CNTINU(i)	.F.	-	If .TRUE., sets starting point(s) of voltage, phase and/or frequency curve(s) for source i to be the current value(s), hence smoothly piecing curve segments together.
VMATCH(i)	.F.	-	If .TRUE., sets VI(i) to match the current distribution emittance[b].
HOLDBH	.F.	-	If .TRUE., the voltage of source 1[c] is varied such that the corresponding bucket height is multiplied by HDECR on successive turns.
HDECR	1.0	-	Factor by which (source 1) bucket height is adjusted on successive turns if HOLDBH = .TRUE.
HOLDBA	.F.	-	If .TRUE., the voltage of source 1[c] is varied such that the corresponding bucket area is multiplied by SDECR on successive turns.
SDECR	1.0	-	Factor by which (source 1) bucket area is adjusted on successive turns if HOLDBA = .TRUE.
PHISLIM	.95	-	The HOLDBH or HOLDBA option cannot reduce the (source 1) voltage such that $\sin\phi_s >$ PHISLIM.
DELTRF(i)	0.0	-	Momentum offset ($\Delta p/p$ with respect to the synchronous particle) at which source i is operated (by introducing the corresponding phase offset).
PHSLIP	.F.	-	If .TRUE., activates momentum offsets for all sources.

[a]Fit to values read from file. See section 2.1.2.

[b]Consequently, in this instance, the P-command (or its equivalent) should precede the A-command.

[c]The algorithms used to maintain the bucket height and area consider only a single source.

An example of a "minimal" A-command might be:

```
$RF H(1)=20, VI(1)=.200, PSII(1)=180. $END
```

Since rf manipulations are at the heart of ESME, this command can become rather lengthy and involved; the above example is exceptionally brief.

At times prior to the start of a voltage, phase, or frequency variation, the relevant quantity is maintained at its initial value. At times after the programmed end of a curve, the quantity is maintained at the final value. This makes it easier to specify multiple curves in unequal time steps. It is important to note that the values supplied for phase curves are the relative phases of the sources

(each expressed in terms of its own frequency) and are to be distinguished from the absolute phases of all the harmonics in the cavity at the passage of the synchronous particle, although they may be the same. Thus, any periodic waveform may be represented by simply specifying the correct relative phases and amplitudes of the voltages in a Fourier expansion of up to ten terms. The resultant waveform may be modified over time by specifying the evolution of the voltages, phases, and/or frequencies, but note that, in cases in which both a phase and a frequency variation are specified for the same source, the phase variation takes precedence.

In addition, the user may request that ESME search for a synchronous point on the accelerating waveform. Setting ISYNC = 1 causes the program to search for a stable value, PHIS, of the phase; this is stored internally in units of radians of azimuth and output as degrees of "shift" of the resultant voltage waveform. The option ISYNC = 2 causes the amplitude of the resultant waveform to be scaled to provide the correct voltage at the passage of the synchronous particle, although not necessarily with a stable slope since the phases of the rf sources remain as programmed. The Landau cavity option, ISYNC = 3, results in the phases of sources 1 and 2 being altered, together with the voltage of source 2, such that the first and second derivatives of the combined voltage vanish at the passage of the synchronous particle. If NRF > 2, the other sources are not affected and will be included in the iteration of the difference equations.

2.2.4 The P-command – Initial Distribution Parameters

The members of NAMELIST /POPL8/ are read in subroutine POPUL8 and are stored in COMMON /POPLATE/. Phase space points are generated in azimuth, ϑ, vs. energy (relative to that of the synchronous particle).

P-command, NAMELIST /POPL8/

Variable	Default Value	Default Unit	Description
KIND	1	-	Type of distribution to be generated: 1 – Rectangular outline of NTH points between THMIN, THMAX in ϑ and NE points between REMIN, REMAX in energy 2 – Rectangular NTH by NE (regular) grid with limits as for KIND = 1 3 – Rectangular uniform (random) distribution of NPOINT particles with limits as for KIND = 1 4 – NPOINT particles uniform in ϑ between THMIN, THMAX, Gaussian in energy with $\pm 2\sigma$ limits REMIN, REMAX 5 – NPOINT particles Gaussian in ϑ with $\pm 2\sigma$ limits THMIN, THMAX, uniform in energy between REMIN, REMAX 6 – Rectangular NTH by NE grid, regular spacing in ϑ, Gaussian spacing in energy, and limits as for KIND = 1 7 – Bunch contour outline of NPOINT particles 8 – Rectangular NTH by NE grid inside contour 9 – Uniform distribution of NPOINT particles inside contour

P-command, NAMELIST /POPL8/ (continued)			
Variable	Default Value	Unit	Description
KIND	1	-	10 – Bi-Gaussian distribution of NPOINT particles, 95% within contour 12 – NPOINT particles parabolic in ϑ between THMIN, THMAX, uniform in energy between REMIN, REMAX 13 – Bi-parabolic distribution of NPOINT particles inside contour 14 – Matched elliptical distribution of NPOINT particles 15 – NPOINT particles uniform in ϑ between THMIN, THMAX, parabolic in energy between REMIN, REMAX.
THMIN	-90.0	deg	Lower limit of ϑ for rectangular distributions.
THMAX	90.0	deg	Upper limit of ϑ for rectangular distributions.
REMIN	None	MeV	Lower limit of energy (relative to that of the synchronous particle) for rectangular distributions.
REMAX	None	MeV	Upper limit of energy (relative to that of the synchronous particle) for rectangular distributions.
NTH	2	-	Number of grid points in ϑ.
NE	2	-	Number of grid points in energy.
SBNCH	0.1	eVs	Area within matching contour.
IPOP	1	-	Source used to generate matching contour: 0 – All active rf sources > 0 – Source IPOP ($1 \leq$ IPOP $\leq$ NRF).
THOFF	0.0	deg	Displacement in azimuth of distribution generated in the current P-command.
EOFF	0.0	MeV	Displacement from synchronous energy of distribution generated in the current P-command.
THTRAN	0.0	deg	Displacement in azimuth of all particles generated in this and previous P-commands.
ETRAN	0.0	MeV	Displacement from synchronous energy of all particles generated in this and previous P-commands.
NPOINT	1	-	Number of particles in distribution (except for the grids, KIND = 1, 2, 6, 8).
PARTION	.F.	-	If .TRUE., partitions distribution into separate classes[a]; each use of the P-command with PARTION = .TRUE. introduces a new partition.
ISEED	314159	-	Seed for internal random number generator.

[a]Different classes of particles are plotted with distinct symbols.

For example, a bi-Gaussian distribution of 1000 particles with a 95% emittance of 0.5 eV.s would be generated by the following P-command:

```
$POPL8 KIND=10, NPOINT=1000, SBNCH=.5 $END
```

Since IPOP is not specified, the aspect ratio of this bunch is matched to rf source 1 by default. Likewise, the bunch is centred at zero degrees of azimuth and the synchronous energy.

For multi-bunch simulations in which the same distribution is repeated with the same parameters but in a different position, it is sufficient to re-issue the P-command specifying only the desired offsets, THOFF and EOFF. This is because NAMELIST members which do not appear in the input retain their previous values. Consequently, expeditious use of THTRAN and ETRAN can make it unnecessary to declare explicitly the position of each group of particles. The total number of particles in all distributions must not exceed 50,000.

2.2.5 The O-command – Graphics Output Options

The members of NAMELIST /GRAPH/ are read in subroutine GRAFSET and are stored in COMMON /GRAFIX/. Phase space plots of azimuth, ϑ, vs. energy (relative to that determined by IEREF) are produced at regular intervals during tracking unless the POSTP flag is set to .TRUE., when the raw data are written (every MPLOT turns or each time a D-command is issued) to FORTRAN logical unit 18 instead. This allows the user to produce his own plots in a separate job. The post-processor program ESMEPLOT is described in chapter 4.

O-command, NAMELIST /GRAPH/

Variable	Default Value	Default Unit	Description
MPLOT[a]	1000	turn	Graphics output interval; plots are produced every MPLOT turns (or each time a D-command is issued).
IDEV	0	-	Virtual device number[b] for graphics output.
IMETA	-1	-	Directs graphics output: < 0 – Output is written to a metafile 0 – Output is directed to the device specified by IDEV > 0 – Output is written both to a metafile and to the device specified by IDEV.
POSTP	.F.	-	If .TRUE., all data in COMMON blocks are written to logical unit 18; the plotting routine is not called.
PLTSW			Selects plot options (when array element is .TRUE.):
(1)	.T.	-	Draw phase space plot
(2)	.T.	-	Plot phase space points (distinct symbol for each class)
(3)	.F.	-	Interconnect points within each class
(4)	.F.	-	Draw lines at centroid and $\pm 1\sigma$
(5)	.F.	-	Draw voltage waveform
(6)	.F.	-	Set plot boundaries to turning points of contour
(7)	.F.	-	Suppress captions, axis labels, etc.
(8)	.T.	-	Plot ϑ histogram
(9)	.F.	-	Set ϑ histogram limits to turning points of contour
(10)	.T.	-	Plot energy histogram
(11)	.F.	-	Set energy histogram limits to turning points of contour
(12)	.F.	-	Plot Fourier amplitudes
(13)	.F.	-	Include phases in plot of Fourier spectrum
(14)	.F.	-	Plot space-charge energy loss (per turn) vs. ϑ
(15)	.F.	-	Include distribution histogram in space-charge plot
(16)	.F.	-	Plot high-Q resonator voltage.

[a]MPLOT = 0 results in plots being produced only when a D-command is issued.
[b]See refs. [15] and [16].

O-command, NAMELIST /GRAPH/ (continued)			
Variable	Default Value	Default Unit	Description
THPMIN	0.0[a]	deg	Lower limit of ϑ for phase space plot.
THPMAX	0.0[a]	deg	Upper limit of ϑ for phase space plot.
DEPMIN	0.0[b]	MeV	Lower limit of energy for phase space plot.
DEPMAX	0.0[b]	MeV	Upper limit of energy for phase space plot.
TITLE[c]	.F.	-	If .TRUE., use the line following NAMELIST input as a plot title.
KLPLOT	0	-	Determines which class of particles is plotted and histogrammed. 0 – All classes are plotted > 0 – Only class KLPLOT is plotted.
NPJMP	1	-	Plot only every NPJMP-th point in phase space.
NBPFFT	0[d]	-	Number of bins in FFT plot.
IOPT	-1	-	Source for bucket (or matched contour) plotting: < 0 – None; no contour plotted 0 – All active rf sources > 0 – Source IOPT ($1 \leq$ IOPT $\leq$ NRF).
MTCHSI	.F.	-	If .TRUE., plot contour (matched to source IOPT) whose area is equal to the initial distribution emittance.
MTCH95	.F.	-	If .TRUE., plot contour (matched to source IOPT) containing 95% of the present distribution emittance.
IEREF	1	-	Determines energy origin for phase space plots: 1 – E0, energy on the reference orbit 2 – ES, synchronous energy 3 – EBAR, mean particle energy 4 – EREF, energy of the "reference particle"[e].
NBINTH	50	-	Number of bins in ϑ for histogram.
THBMIN	0.0[f]	deg	Lower limit of ϑ for histogram.
THBMAX	0.0[f]	deg	Upper limit of ϑ for histogram.
NBINE	50	-	Number of bins in energy for histogram.
EBMIN	0.0[g]	MeV	Lower limit of energy for histogram.
EBMAX	0.0[g]	MeV	Upper limit of energy for histogram.
IFBMIN	1	-	Lower limit for FFT plot.
IFBMAX	0[h]	-	Upper limit for FFT plot.

[a] THPMIN and THPMAX both zero results in the plot range being $-180°/\text{FRAC} \leq \vartheta \leq 180°/\text{FRAC}$.

[b] DEPMIN and DEPMAX both zero results in approximately the entire range of particle energies being plotted.

[c] This is an exception to the usual maintenance of NAMELIST data; TITLE is set to .FALSE. after every O-command.

[d] NBPFFT = 0 results in the value of NBINFFT (see the F-command) being used.

[e] A particle which ESME tracks from the origin, (0,ES).

[f] THBMIN and THBMAX both zero results in the histogram range being the same as for the phase space plot.

[g] EBMIN and EBMAX both zero results in the histogram range being the same as for the phase space plot.

[h] IFBMAX = 0 results in the upper limit being the greatest Fourier harmonic computed.

O-command, NAMELIST /GRAPH/ (continued)			
Variable	Default Value	Unit	Description
SCBMIN	0.0[a]	-	Lower limit of ϑ for space-charge plot.
SCBMAX	0.0[a]	-	Upper limit of ϑ for space-charge plot.
RBMIN	0.0[b]	-	Lower limit of ϑ for resonator voltage plot.
RBMAX	0.0[b]	-	Upper limit of ϑ for resonator voltage plot.
XCRNR	135.0	0.001	Fraction of the full width of the plot frame between the left-hand edge of the frame and the bottom left-hand corner of the plot.
YCRNR	69.0	0.001	Fraction of the full height of the plot frame between the bottom edge of the frame and the bottom left-hand corner of the plot.
XAXISL	750.0	0.001	Plot width as a fraction of the full width of the frame.
YAXISL	750.0	0.001	Plot height as a fraction of the full height of the frame.
DTHCURV	0.0	deg	Displacement of contour in azimuth.
DECURV	0.0	MeV	Displacement of contour in energy.
DELCON	0.01	-	Determine rf bucket to within DELCON x 360° of rf waveform.
KNTLIM	500000	-	Maximum number of iterations of difference equations which will be attempted to close contour.

[a]SCBMIN and SCBMAX both zero results in the plot range being ±180°/FRAC.
[b]RBMIN and RBMAX both zero results in the plot range being ±180°/FRAC.

The default limits for many of the plots serve as flags to the plotting routine to choose reasonable limits, so that it may be sufficient simply to supply MPLOT in an O-command. Note that MPLOT = 0 specifies that graphics output is only to be generated when explicitly requested. The default PLTSW settings produce a phase space plot of the distribution without a bucket contour, together with plots of the projections of the distribution in ϑ and in energy. In addition to the graphics output, the first and second moments of the distribution are computed and output as well as a number of other system parameters. The moments included in the data on the projections are calculated for those particles which are in the class(es) being plotted and within the plot limits. Whereas, the moments printed in the standard output are those for the entire distribution.

The parameter DELCON allows the user either to determine the separatrix arbitrarily closely or simply to save processing time, since the ESME subroutine CONTOUR which determines the bucket may have to perform many iterations of the difference equations in order to determine the separatrix to the specified accuracy. In cases when this routine is unacceptably slow (or is unable) to find the contour, it may be useful to set KNTLIM to some lower value.

2.2.6 The T-command – Track Distribution

The members of NAMELIST /CYCLE/ are stored in COMMON /CYCLP/. This NAMELIST read is made in subroutine CYCPROG, which iterates the difference equations.

T-command, NAMELIST /CYCLE/			
Variable	Default Value	Unit	Description
TSTOP[a]	0.0	s	Time at which to stop tracking.
TTRACK[b]	0.0	s	Duration of tracking.
MSTEP	100	-	(Min.) number of tracking steps per synchrotron period.
ACCEL0[c]	1.0	-	(Max.) number of beam turns per tracking step.
NCAV	1	-	Number of rf "cavities" in the ring, i.e., the number of iterations of the difference equations per turn.
LGRTHM	1	-	Selects difference equations used in tracking: 1 - Complete kinematics; expand α to maximum order using the coefficients ALPHAn[d] 2 - Use the simplified form, $\vartheta_{i,n} = \frac{\tau_{s,n-1}}{\tau_{s,n}}\vartheta_{i,n-1} + 2\pi\eta\frac{\Delta p_{i,n}}{p_{s,n}}$.
ITRAP(1:4)	0	-	Condition which causes tracking to be interrupted before the time indicated by TTRACK or TSTOP: 0 - No trap 1 - Trap on minimum bunch width 2 - Trap on minimum bunch height 3 - Trap on η = ETATRP (tolerance = $\pm 0.01\eta$) 4 - Trap on $\lvert \sin\phi_s \rvert$ = PHISTRP (tolerance = ± 0.005) 5 - Trap on $\eta > 0$, i.e., transition crossing 10-19 - Call subroutine SHAZAM at entry point SHAZAM, SHAZAM1,...,SHAZAM9 after every iteration of the difference equations. Tracking is not stopped.
ETATRP	.001	-	See ITRAP = 3.
PHISTRP	.95	-	See ITRAP = 4.
MGRACE	0	-	Allow a "grace period" of MGRACE turns before checking trapping conditions.
HISTRY	.F.	-	If .TRUE., write a history record to logical unit 9 after every iteration of the difference equations.
MOMNTS	.F.	-	If .TRUE., compute the first and second moments of the distribution after every iteration of the difference equations.
BBDRY	.F.	-	If .TRUE., remove particles outside the region $-180°/\text{FRAC} \leq \vartheta \leq 180°/\text{FRAC}$.

[a]This is an exception to the usual maintenance of NAMELIST data; TSTOP is set to zero when tracking is completed or is interrupted by an ITRAP option.

[b]TSTOP takes precedence unless it is zero, then TTRACK determines the tracking duration.

[c]Tracking proceeds according to the integer, ACCEL (≥ 1), nearest ACCEL0 which satisfies the MSTEP constraint.

[d]See the R-command.

A typical T-command line might be:

```
$CYCLE TTRACK=.001, LGRTHM=2, HISTRY=.T., MOMNTS=.T. $END
```

Here, particles are tracked for a simulated millisecond, with a calculation of distribution moments and a write of this and other data to the so-called "history" file (required by the H-command) on every iteration of the difference equations.

Note that either TSTOP or TTRACK may be used to specify the duration of tracking, but that TSTOP takes precedence. Also, since an interruption in tracking by an ITRAP option (for ITRAP ≤ 5) causes TSTOP to be set to zero, the stop time must be respecified in a subsequent T-command. Any condition which halts or interrupts tracking is tested for at most once per turn, so that the actual tracking duration may be longer than that specified by TSTOP or TTRACK. Four ITRAP variables are provided to allow for multiple traps and/or calls to SHAZAM routines.

2.2.7 The Q-command – Quit

The Q-command directs ESME to stop processing. No more commands are read.

2.2.8 The D-command – Display Distribution

The D-command directs ESME to generate graphics output at the point in the calculation where the command is issued. The form of the output is determined by the most recent O-command. The D-command complements the O-command since the latter only provides graphics at fixed intervals.

2.2.9 The H-command – History Output

The members of NAMELIST /HISTRY/ are read in subroutine HISTORY. Several variables are shared with the O-command but may be respecified for history output.

H-command, NAMELIST /HISTRY/			
Variable	Default Value	Unit	Description
IDEV	0	-	Virtual device number[a] for graphics output.
IMETA	-1	-	Directs graphics output: < 0 – Output is written to a metafile 0 – Output is directed to the device specified by IDEV > 0 – Output is written both to a metafile and to the device specified by IDEV.
NPLT(1:2,m)	0	-	Parameters to be plotted vs. each other from history records; NPLT(1,m) is the independent and NPLT(2,m) the dependent variable of the m-th history plot (m = 1:50).
NWRT(1:2,m)	0	-	Same as NPLT, except that the data pairs are written to logical unit 19 rather than being plotted.
			Real records: 1 – Time 2 – PHIS[b], "synchronous phase" 3 – PDOT, time derivative of synchronous momentum 4 – THBAR, mean value of ϑ of the distribution 5 – EBAR, mean energy of the distribution 6 – THRMS, rms value of ϑ of the distribution 7 – ERMS, rms energy spread of the distribution 8 – ES, synchronous energy 9 – E0, energy on the reference orbit

[a] See refs. [15] and [16].
[b] See section 2.2.3.

H-command, NAMELIST /HISTRY/ (continued)			
Variable	Default Value	Unit	Description
NPLT or NWRT	0	-	10 – ES – E0 11 – THREF, azimuth of the "reference particle"[a] 12 – EREF, energy of the "reference particle"[a] 13 – EPSILON, statistical "emittance" about barycentre of the distribution 14 – NUS, synchrotron frequency 15 – SBCKT, rf bucket area 16 – HBCKT, rf bucket height 17 – ETA, synchronous value of η 18 – ACCEL, number of beam turns per tracking step[b] 19 – TAU, synchronous revolution period 20 – PSIADD, phase feedback[c] 21 – DAMPL, voltage feedback factor[c] 22 – DELR, synchronous mean radial postion (relative to the reference orbit) 23 – RFFREQ, frequency of rf source 1.
			Integer records: 31 – Turn number 32 – KOUNT, number of surviving particles.
			Array records: 51-60 – SPARE(1-10), available for SHAZAM routines 101-110 – EV(1-10), voltage of rf source 1-10 111-120 – PSI(1-10), phase of rf source 1-10 121-130 – FREQ(1-10) – FRI(1-10), change in frequency of rf source 1-10 201-250 – FAMPL(1-50), Fourier amplitudes[d] 251-300 – FAZE(1-50), Fourier phases[d].
XCRNR	135.0	0.001	Fraction of the full width of the plot frame between the left-hand edge of the frame and the bottom left-hand corner of the plot.
YCRNR	69.0	0.001	Fraction of the full height of the plot frame between the bottom edge of the frame and the bottom left-hand corner of the plot.
XAXISL	750.0	0.001	Plot width as a fraction of the full width of the frame.
YAXISL	750.0	0.001	Plot height as a fraction of the full height of the frame.

[a] A particle which ESME tracks from the origin, (0,ES).
[b] See the T-command.
[c] See section 2.2.13.
[d] See section 2.2.15.

The H-command causes the history data saved during tracking (by setting HISTRY = .TRUE. in a T-command) to be processed and output in graphics form. The output is not presently as flexible as that produced by the O-command. The user simply selects pairs of parameters to be plotted by specifying the indices above as elements of the array NPLT. There is no choice of range

for any axis; the program produces a plot for each pair of parameters assembled sequentially from the entire history file. For example, suppose that a simulation has run for 10,000 turns and that the desired output is of the distribution moments vs. time over that period. The H-command might appear as:

```
$HISTRY NPLT=1,4,1,5,1,6,1,7,4,5 $END
```

This command will generate plots of THBAR, EBAR, THRMS, and ERMS vs. time, as well as a plot of EBAR vs. THBAR (provided that both HISTRY and MOMNTS were set to .TRUE. in the original T-command). Advantage is taken here of the way in which FORTRAN arrays are constructed; NPLT(1,m) and NPLT(2,m) are consecutive elements so that explicit reference to each array index is not necessary.

For simulations exceeding several thousand turns, not every turn will be included in the plot(s). This is due to the fact that the frequency of write operations to the history file may be modified by the program[5] and to a culling procedure which is applied to the data points by the plotting routine. The number of plots in a single H-command must not exceed 50.

2.2.10 The M-command – Save Mountain Range Data

The members of NAMELIST /MRANGE/ are read in subroutine MRINIT and are stored in COMMON /MRANGE/.

M-command, NAMELIST /MRANGE/

Variable	Default Value	Default Unit	Description
TMBEGIN	0.0	s	Time at which to start saving mountain range data.
TMEND	0.0	s	Time at which to stop saving mountain range data.
MRMPLOT	1	turn	Interval between mountain range data records; data are saved every MRMPLOT turns.
MRNBIN	50	-	Number of bins for mountain range histogram.
MRTHBMIN	0.0[a]	deg	Lower limit of ϑ for mountain range histogram.
MRTHBMAX	0.0[a]	deg	Upper limit of ϑ for mountain range histogram.

[a]MRTHBMIN and MRTHBMAX both zero results in the histogram range being ±180°/FRAC.

The M- and N-commands are together intended to produce plots which are similar to those provided by an oscilloscope displaying successive traces of a longitudinal pick-up, each trace being vertically displaced from the previous one. (The resultant display of the time evolution of the azimuthal projection of a distribution of particles somewhat resembles a "mountain range".) The M-command causes the data to be histogrammed and written to logical unit 20, while the N-command directs the program to process the resultant file and produce a mountain range plot.

2.2.11 The N-command – Plot Mountain Range Data

The members of NAMELIST /MRPLOT/ are read in subroutine MRPLT and are stored in COMMON /MRANGE/. Several variables are shared with the O-command but may be respecified for mountain range plots.

[5]See chapter 4

N-command, NAMELIST /MRPLOT/			
Variable	Default Value	Default Unit	Description
IDEV	0	-	Virtual device number[a] for graphics output.
IMETA	-1	-	Directs graphics output: < 0 – Output is written to a metafile 0 – Output is directed to the device specified by IDEV > 0 – Output is written both to a metafile and to the device specified by IDEV.
MRTHPMIN	0.0[b]	deg	Lower limit of ϑ for mountain range plot.
MRTHPMAX	0.0[b]	deg	Upper limit of ϑ for mountain range plot.
NTRACE	100	-	Number of traces on each page of mountain range plot.
NSKIP	0	-	Number of data records to be "skipped" between each trace.
TOPTOB	0.7	-	Fraction of YAXISL in which NTRACE traces are plotted (approximate if TBASE = .TRUE.).
SCALE	0.3	-	Height of the first trace (as a fraction of YAXISL).
MSTART	0[c]	-	Turn number at which to start plotting mountain range data.
MSTOP	0[c]	-	Turn number at which to stop plotting mountain range data.
TMSTART	0.0[d]	s	Time at which to start plotting mountain range data.
TMSTOP	0.0[d]	s	Time at which to stop plotting mountain range data.
TBASE	.F.	-	If .TRUE., trace separation is proportional to time.
LIM	.F.	-	If .TRUE., dotted lines are drawn connecting the leftmost and rightmost non-zero points of consecutive traces.
SMOOTH	.F.	-	If .TRUE., a smoothing algorithm is applied to the data before plotting.
XCRNR	135.0	0.001	Fraction of the full width of the plot frame between the left-hand edge of the frame and the bottom left-hand corner of the plot.
YCRNR	69.0	0.001	Fraction of the full height of the plot frame between the bottom edge of the frame and the bottom left-hand corner of the plot.
XAXISL	750.0	0.001	Plot width as a fraction of the full width of the frame.
YAXISL	750.0	0.001	Plot height as a fraction of the full height of the frame.

[a]See refs. [15] and [16].

[b]MRTHPMIN and MRTHPMAX both zero results in the data being plotted over its entire range, MRTHBMIN $\leq \vartheta \leq$ MRTHBMAX (see the M-command).

[c]MSTART, MSTOP, TMSTART and TMSTOP all zero results in all data, from TMBEGIN to TMEND (see the M-command), being plotted.

[d]MSTART and MSTOP take precedence unless both are zero, then TMSTART and TMSTOP determine the data plotted.

Typical M- and N-commands might be:

```
$MRANGE TMBEGIN=1.0, TMEND=1.2, MRMPLOT=100 $END
```

which causes mountain range records to be written every 100 turns from 1.0 to 1.2 seconds in the

simulation and

```
$MRPLOT SMOOTH=.T. $END
```

which causes all the mountain range data accumulated thus far in the calculation to be plotted with smoothing.

The default TBASE = .FALSE. produces consecutive traces which are vertically equidistant. However, even in this case, the values of TOPTOB and/or SCALE may have to be adjusted since the scaling of the plot is based on the characteristics of the first two traces alone.

2.2.12 The W-command – Write Comment

The W-command simply directs ESME to echo any characters which follow (on the same line and starting at the sixth column) in the printed output. It serves as a means to add comments to an input dataset.

2.2.13 The L-command – Low Level Feedback Parameters

The members of NAMELIST /LLRF/ are read in subroutine LOWLVL and are stored in COMMON /FEEDS/.

L-command, NAMELIST /LLRF/

Variable	Default Value	Default Unit	Description
PHFBON	.F.	-	If .TRUE., activates phase feedback.
VFBON	.F.	-	If .TRUE., activates voltage feedback.
NTUAVG	1[a]	-	Number of previous turns over which to average in the calculation of the phase "signal".
NTURES	1	turn	Delay in phase feedback response; the present "signal" is compared with that obtained NTURES turns previously.
IFTB	0	-	Form of the phase feedback: 0 – Critical damping 1 – Fixed.
USEWT	.F.	-	If .TRUE., the array W is used in averaging the individual "measurements" of mean phase over NTUAVG turns.
W(1:NTUAVG)	0.0	-	Weights which multiply the turn-by-turn mean phases if the phase "signal" is computed with USEWT = .TRUE.
FBFACT	1.0	-	Gain applied to the phase feedback.
DLIMIT	5.7296	deg	Upper limit of the magnitude of the phase feedback on a single turn.
VFBFCTR	1.0	-	Gain applied to the voltage feedback.
VLIMIT	0.1	-	Upper limit of the magnitude of the voltage feedback fraction on a single turn.
ETAJMP	0.0	-	Value of η at which to "flip" the phase of the rf.

[a]NTUAVG = 1 represents infinite-bandwidth phase feedback.

The phase feedback, which is intended to damp dipolar bunch oscillations, attempts to maintain the mean azimuthal position of the particle distribution the same at the end of each tracking step by adding a phase shift to all active rf sources given by:

$$\mathrm{PSIADD} = \frac{\mathrm{FBFACT}\sum_{i=1}^{\mathrm{NTUAVG}} \mathrm{W}(i)(\langle\vartheta\rangle_{n-i} - \langle\vartheta\rangle_{n-\mathrm{NTURES}-i})}{\pi \mathrm{Q_s} \mathrm{NTUAVG}^2 \sum_{i=1}^{\mathrm{NTUAVG}} \mathrm{W}(i)}$$

where angular brackets denote the mean of the distribution, n is the current turn number and where, if ITFB = 0, $\mathrm{Q_s}$ is the synchrotron tune or, if ITFB = 1, $\mathrm{Q_s}$ is replaced by the fixed value 4.0×10^{-3}. The magnitude of PSIADD is not permitted to exceed DLIMIT.

The simplest invocation of phase feedback would be:

```
$LLRF PHFBON=.T. $END
```

The defaults imply critical damping.

The voltage feedback, which is intended to damp quadrupole bunch oscillations, multiplies the voltage of all active rf sources at the end of each tracking step by the factor:

$$1 + \mathrm{DAMPL} = 1 + \frac{\mathrm{VFBFCTR}\sum_{i=0}^{n-j}(\langle E^2\rangle_{n-i} - \langle E^2\rangle_{n-i-1})}{2\pi \mathrm{Q_s} 100 H_j}$$

where j is the turn number when feedback starts (i.e., when the L-command is issued with VFBON = .TRUE.) and E is particle energy. The factor $100H_j$ in the denominator, where H_j is the height of the bucket due to source 1 when feedback starts, is somewhat arbitrary and the user may have to adjust VFBFCTR to obtain satisfactory results. The magnitude of DAMPL is not permitted to exceed VLIMIT.

2.2.14 The B-command – Space-charge

The members of NAMELIST /SCHG/ are read in subroutine FOURFIT (entry BEAMSC) and are stored in COMMON /SPCHG/.

B-command, NAMELIST /SCHG/

Variable	Default Value	Default Unit	Description
A	0.002	m	Effective beam radius.
B	0.05	m	Effective beampipe radius.
ENQ	2.E10	-	Number of protons to be represented by the distribution.
NZ	0	-	Number of impedance values to be read from file[a].
NR	0	-	Number of resonance values to be read from file[b].
SCON	.F.	-	If .TRUE., activates space-charge calculation.
MSC	1	-	Number of times collective effects are calculated between rf cavities.
QREZON	.F.	-	If .TRUE., activates high-Q resonance calculation.
NBRES	10,000	-	Number of bins to be used in the high-Q resonance calculation.

[a]See section 2.1.3.

[b]See section 2.1.4.

B-command, NAMELIST /SCHG/ (continued)			
Variable	Default Value	Unit	Description
NNF	0	-	Number of Fourier harmonics written to the history file.
NF(1:NNF)	0	-	Harmonic numbers of the NNF Fourier components.
MAXFFTB	1024	-	Max. number of bins to be used in FFT.
MFFT	1	turn	Interval between calculations of the Fourier transform; an FFT is performed every MFFT turns.

The B-command controls facilities in ESME for modelling the interactions of the beam particles with each other both through the direct particle-particle force and through wakefields excited as a consequence of the interaction of the beam with its environment (vacuum chamber, rf cavities, etc.). The routines that are activated by setting SCON = .TRUE. calculate these effects using an equivalent impedance, which may be adequate for many problems. However, implicit in this approach is the assumption of a steady-state solution for the response of a resonator[10]. This is reflected in the absence of any frequencies other than harmonics of the revolution frequency in the calculation, which employs an FFT of the particle distribution. The number of bins used in the FFT is the smaller of MAXFFTB and an upper limit determined by the cutoff frequency of the beampipe.

Transient effects are taken into account using the routines that are activated by setting QREZON = .TRUE. If the space-charge force is calculated when resonances have been specified, the QREZON routines take precedence over the SCON ones in the calculation of the resonator voltage. Implicit in the code are the assumptions that the distribution being tracked is periodic and that the resonator voltage is applied at intervals which are equal to the time duration of the distribution. The simplest way to satisfy both of these constraints is to track a distribution which spans the entire machine circumference and to calculate the resonator voltage once per turn. Otherwise, it may be sufficient to track a distribution whose azimuthal periodicity FRAC > 1 and to calculate the resonator voltage at FRAC equispaced intervals per turn (i.e., MSC * NCAV = FRAC). Note that the number of protons specified by ENQ fill 1/FRAC of the machine circumference.

A B-command to model space-charge, resistive wall impedance and certain parasitic high-Q resonances might be:

```
B
  $SCHG SCON=T, A=.005, B=.075, ENQ=2.E13,
  NR=3, NZ=10, MAXFFTB=512 $END
ifname iftype
rfname rftype
```

Here, cubic polynomial coefficients for the wall impedance are read from file **ifname iftype**, while resonance values are obtained from the file **rfname rftype**. (Under VAX VMS, the two parts of each file identifier must be joined by a full stop). The impedance file is only read if NZ > 0 and the resonance file only if NR > 0. No files need be specified if NZ = NR = 0.

2.2.15 The F-command – Fourier Transform

The members of NAMELIST /FFT/ are read in subroutine FOURFIT and are stored in COMMON /FOURIR/. Since the Fourier transform of the particle distribution is used in the space-charge calculation, several variables are shared with the B-command.

F-command, NAMELIST /FFT/			
Variable	Default Value	Unit	Description
FFTON	F	-	If .TRUE., activates Fourier transform calculation.
NNF	0	-	Number of Fourier harmonics written to the history file.
NF(1:NNF)	0	-	Harmonic numbers of the NNF Fourier components.
NBINFFT	256	-	Number of bins to be used in FFT.
MFFT	1	turn	Interval between calculations of the Fourier transform; an FFT is performed every MFFT turns.
FFTOUT	F	-	If .TRUE., Fourier transform is printed.

The F-command allows the user to examine the Fourier spectrum of the distribution and, in conjunction with the H-command, to follow the development of selected harmonics as a function of time. For example, the F-command to record the turn-by-turn development of the first five odd Fourier components of the distribution would be:

```
$FFT FFTON=.T., NBINFFT=32, NNF=5, NF=1,3,5,7,9 $END
```

Then, to plot the evolution of, say, the amplitude and phase of the seventh harmonic, an H-command should be issued to retrieve FAMPL(4) and FAZE(4) (see section 2.2.9).

Note that the harmonic numbers of the distribution must be multiplied by its azimuthal periodicity, FRAC, in order to obtain the correct harmonics of the revolution frequency. If, for example, FRAC = 4, then the seventh harmonic of the distribution corresponds to harmonic twenty-eight for the entire machine circumference.

2.2.16 Commands 0-9 – Call User-written SHAZAM Routines

The subroutine SHAZAM is a collection of ten independent routines each with its own entry point. All ESME's COMMON blocks are made available to this subroutine. Its purpose is to allow users to integrate their own routines into the code. The different routines may be called after every iteration of the difference equations using the "trapping" options of the T-command (see section 2.2.6), or they may be called explicitly using the commands 0-9. The user is free to choose the method by which input data, if any, are supplied to the routines that he writes. However, he must ensure that all such input records appearing in the data file are read since, upon returning from a SHAZAM routine, ESME reads the next line in the input data and attempts to interpret it as a command.

2.2.17 The S-command – Save Tracking Parameters

The S-command directs subroutine SAVE to write all the data in ESME's COMMON blocks to a file assigned to logical unit 8. It allows the user to suspend tracking at any point in the calculation. Subsequent jobs may then continue from the same point by restoring these data with the G-command. Subroutine SAVE would also be useful to anyone wishing to analyze the tracking data independently of ESME (see chapter 4). The format of the S-command is simply:

```
S
fname ftype
```

where **fname ftype** is the file to which the data are written. (Under VAX VMS, the two parts of the file identifier must be joined by a full stop).

2.2.18 The G-command – Get Tracking Parameters

The G-command directs subroutine GET to read data from a file assigned to logical unit 7 into ESME's COMMON blocks. It allows the user to retrieve data which were saved using the S-command. The format of the G-command is simply:

```
G
fname ftype
```

where **fname ftype** is the file from which the data are read. (Under VAX VMS, the two parts of the file identifier must be joined by a full stop).

Chapter 3

Running the Program

ESME is available on both IBM and VAX machines at CERN, but it should be appreciated that the program had its origins on a VAX cluster at Fermilab. Considerable effort has gone into making the original code acceptable to IBM since this is the recommended machine for production jobs. However, code management is done entirely on VXCERN (see chapter 5) in order that updates may be implemented in a fashion which is consistent with their Fermilab source.

Basically, the process of running ESME consists of associating file names with FORTRAN logical units and then issuing the necessary directives to execute the program. It may be run either interactively or in batch mode. FORTRAN logical unit 5 is expected to supply the main input. An appropriate device driver must be specified if graphics output is to be viewed while running the program interactively on a device capable of such output[1]. However, since the results of mixing text and graphics output on the same device are somewhat unpredictable, it is recommended that the user simply employ GRVIEW after the program has run to examine the GKS metafile which ESME produces by default. If a printer driver were used for graphics output, then a graphics file will be generated which may be printed. If IMETA is set to 1 or -1 (see section 2.2.5), then a metafile is produced which may be used in conjunction with GRPLOT and/or GRVIEW to generate output on physical devices. History records are written to the file that is assigned to logical unit 9 and, if the post-processor option is in effect, ESME writes its phase space data to the file assigned to unit 18. The user is then free to process these two files according to his own methods. A simple post-processor "shell" has been written which processes such files using routines taken directly from ESME. It is described in chapter 4.

For most applications, command procedures have been prepared which automate the process of running the program and relieve the user of file management tasks.

3.1 ESME on CERNVM

The "official" release of ESME on the IBM at CERN resides on the PSRF account as the file ESME TEXT. This file is the product of a vector compilation since this typically gains a factor of two in execution time. Consequently, the user must obtain authorization to run vectorized code before he can use the standard TEXT file. (Alternatively, the file ESME FORTRAN on the PSRF 191 disk can be recompiled without vectorization in the user's own area.)

An exec file has been prepared which simplifies the task of running ESME, either interactively or as a batch job. (This procedure also allows the user to specify that ESME FORTRAN source

[1] Workstation types and numbers can be found in refs. [15] and [16].

code is to be compiled and run instead of the version on the PSRF account.) Having issued the command "GIME PSRF" to obtain access to the procedure, the appropriate syntax is:

```
RUNESME [fname] [ftype] [fmode] [ ( [[TIME [mm]|mm:ss] | NOBATCH]
                                   [GRPLOT [(grplot_options)]]
                                   [VFORT [(vfort_options)]] ]
```

where square brackets enclose optional parameters. There is no default for the **fname** of the input data file, whereas the default **ftype** and **fmode** are ESMEDATA and *, respectively. If no keywords are supplied (after the open parenthesis), RUNESME submits a five-minute batch job to process the input data file; that is, **TIME** is specified to be 5:00 by default. (The upper limit of the short batch queue, class S, is currently five minutes of 168-equivalent CPU time.) All files created by ESME in batch are returned to the user's virtual reader. If **NOBATCH** is specified, ESME runs interactively and all output is written to a temporary 20-cylinder T-disk allocated by RUNESME.

ESME will always generate a file called "**fname** OUTPUT", which is just the text that would also be written to the terminal if ESME were run interactively. Additional files may be produced depending on the commands issued in the input dataset. Any of the following files may be created:

- SAVE file (logical unit 8), name specified in S-command.
- **fname** HISTORY (logical unit 9) – history data records.
- **fname** GKSERR (logical unit 13) – GKS errors.
- **fname** METAFILE (logical unit 14) – GKS metafile.
- **fname** PPDATA (logical unit 18) – post-processor data (generated if POSTP is set to .TRUE. in an O-command).
- **fname** HISTTEXT (logical unit 19) – *en clair* text output of history data (generated by NWRT entries in an H-command).
- **fname** MTRANGE (logical unit 20) – mountain range data (generated by an M-command).

A second interactive run with the same input **fname** will cause the deletion of the first set of files on the T-disk. The metafile, if produced, may be printed automatically upon job completion by specifying **GRPLOT** when RUNESME is invoked. There are no default GRPLOT options in RUNESME.

RUNESME will load and execute the ESME TEXT file that resides on the disk that is linked at the highest mode in the user's hierarchy. In particular, if the user has an ESME TEXT on his own A-disk, this will be used instead of the version on the PSRF account. By specifying **VFORT**, the user can even create such a compiled version when submitting his job. The default VFORT options in RUNESME are "VECTOR FLAG(E) NOPRINT", so that users without the necessary authorization to run vectorized code should type "(OPT(3) FLAG(E) NOPRINT)", for example, after the keyword **VFORT**.

3.2 ESME on VXCERN

A command procedure, USESME.COM, has been prepared which defines a number of logicals and symbols which make it possible to run ESME without an explicit knowledge of where the program and its associated procedures are located. On the VAX cluster at CERN, ESME can be

run by entering:

```
@DISK$PZ:[SBH.ESME.COM]USESME
RESMEVAX
```

The command procedure RESMEVAX.COM performs all the necessary tasks by:

1. Asking the user for an input data file.
2. Creating a subdirectory into which all of ESME's output will go.
3. Copying the input file and any other specified files into this subdirectory.
4. Asking the user whether graphics output is to be printed and where.
5. Submitting the job to a batch queue.

ESME will always generate a .LOG file, which is just the text that would also be written to the terminal if ESME were run interactively. Additional files may be produced depending on the commands issued in the input dataset. Assuming an input dataset with the file name ESMEJOB.DAT, any of the following files may be created:

- SAVE file (logical unit 8), name specified in S-command.
- ESMEJOB.HST (logical unit 9) – history data records.
- ESMEJOB.GKE (logical unit 13) – GKS errors.
- ESMEJOB.MTA (logical unit 14) – GKS metafile.
- ESMEJOB.DPP (logical unit 18) – post-processor data (generated if POSTP is set to .TRUE. in an O-command).
- ESMEJOB.HTX (logical unit 19) – *en clair* text output of history data (generated by NWRT entries in an H-command).
- ESMEJOB.MRG (logical unit 20) – mountain range data (generated by an M-command).

Chapter 4

Post-processing

For those users who wish to process ESME data independently of the program, a post-processor option is provided.[1] When this option is in effect, any plotting routine calls are substituted for by writes of ESME's common blocks (containing essentially all of the information about the current state of the simulation) to FORTRAN unit 18 using subroutine SAVE. Later, this file may be read using subroutine GET. The code for a graphics post-processor is appended here as an example. The plotting routines employed here are the same ones imbedded in ESME. This command "shell" is modelled after that of the main program. As in ESME, one-letter commands initiate various routine calls and namelist reads. Since the plotting routines are those of ESME, for which the graphical output options set in the O Command were specifically intended, the user can construct plots using those options set during the running of the program as retrieved from COMMON. Those employing other graphics routines may wish to implement another sort of interface entirely (e.g., menu-driven), with an entirely different set of output options.

```
C     PROGRAM ESMEPLOT
C  This program "post-processes" ESME output data.  It may read
C  and plot history data or process the output file generated by
C  ESME during a run.
C
C  Possible enhancements include:
C   1) Input to GET command specifying
C      record to be read -- at present GET command only invokes GET
C      to read next record.
C   2) Command invoking CONTOUR interactively. (C is available)
C   3) "Menu" input option; input via menus rather than NAMELISTs.
C      Should be more robust and easier for the first-time user.
C   1 is easy; 2 and 3 are a little tougher.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'GRAFIX.INC'
INCLUDE 'CURVES.INC'
INCLUDE 'VERSION.INC'
      DIMENSION ISVARY(ISIZPH)
      CHARACTER*1 CMND
```

[1] POSTP=T, O Command, Section 2.2.5.

```
      CHARACTER*74 WORDS
      LOGICAL EXFG, DONE, SAVND
      NAMELIST /USEFIL/ GRAFIL, HFIL, MRFIL
      CHARACTER*128 GRAFIL,HFIL,MRFIL,OHFIL,OGRAFIL,OMRFIL
      DATA GRAFIL,HFIL,MRFIL /3*' '/
C
C  READ SINGLE LETTER COMMANDS.
C
      NODRAW=.TRUE.
      POSTP=.FALSE.
      IDEV=0
      IMETA=-1
      PRINT 19,PVER
19    FORMAT(//1X,'ESME RF PLOTTING PROGRAM : VERSION NUMBER IS ',F5.2)
$$IF VAX
      ISTAT=LIB$INIT_TIMER()
C      IF(MOD(ISTAT,2).NE.0) CALL LIB$STOP(%VAL(ISTAT))
      IF(.NOT.ISTAT) CALL LIB$STOP(%VAL(ISTAT))
$$ENDIF
C
C  Return point after execution of a command.
C
   10 CONTINUE
      PRINT *,'ENTER COMMAND:
      READ(5,2000) CMND,WORDS
 2000 FORMAT(A1,4X,A74)
$$IF VAX
      ISTAT=LIB$STAT_TIMER(2,ITIME)
      IF(.NOT.ISTAT) CALL LIB$STOP(%VAL(ISTAT))
      CPTIME=1.E-2*ITIME
$$ENDIF
      PRINT 2010, CPTIME
 2010 FORMAT('0',53('+'),2X,F9.2,' SEC.',2X,54('+'))
       PRINT 2020, CMND,WORDS
 2020 FORMAT(' CMND IS ',A1,'  ',A74)
      GO TO (100,200,300,400,500,600,700,800,900)
     +       INDEX('ODGHZXUNQ',CMND)
      PRINT*, 'THE IMPLEMENTED COMMANDS ARE'
      PRINT*, '     O: SET OUTPUT OPTIONS'
      PRINT*, '     D: DISPLAY PHASE SPACE WITH CURRENT PLOT PARAMETERS'
      PRINT*, '     G: GET A RECORD FROM TAPE 7'
      PRINT*, '     H: RETRIEVE AND PLOT HISTORY DATA FROM TAPE 9'
      PRINT*, '     Z: REWIND TAPE 7'
      PRINT*, '     X: PLOT REMAINDER OF TAPE 7'
      PRINT*, '     U: CHOOSE GRAPHICS AND/OR HISTORY FILES'
      PRINT*, '     N: COMPOSE AND DISPLAY MOUNTAIN RANGE PLOT'
      PRINT*, '     Q: STOP'
C  Try again if the command wasn't recognized.
```

```
      GO TO 10
C
C  O: OPTIONS FOR OUTPUT GRAPHICS
C
  100 CALL GRAFSET
      GO TO 10
C
C  D:  DISPLAY GRAPHICALLY PHASE POINTS AND (OPTIONALLY) BUCKET
C
  200 CONTINUE
      IF(IOPT.GE.0)THEN
        IF(MTCHSI .OR. MTCH95)THEN
C  Draw matched contour.
          CALL REFCONT
        ELSE
C  Draw bucket.
          THETAS=0.0
          CALL BUCKIT(IOPT,THETAS,DELCON)
        ENDIF
      ENDIF
      DO 210 I=1,NC
        CURVE(I,1)=CURVE(I,1)+DTHCURV
        CURVE(I,2)=CURVE(I,2)+DECURV
  210 CONTINUE
C  Plot distribution.
      CALL PHPLT
      GO TO 10
C
C  G: GET COORDINATES AND MACHINE PARAMETERS
C
  300 PRINT *, 'READING COORDINATES AND MACHINE PARAMETERS'
C  Save logical telling us whether we have initialized
C  DI-3000 or not and reset it after restoring COMMONs.
C  Also save phase space plot template.
      SAVND=NODRAW
C      DO 310 I=1,ISIZPH
C        ISVARY(I)=IPHARY(I)
C  310 CONTINUE
      CALL GET(.TRUE.,ITURN,DONE)
      NODRAW=SAVND
C      DO 320 I=1,ISIZPH
C        IPHARY(I)=ISVARY(I)
C  320 CONTINUE
      POSTP=.FALSE.
      PRINT *, 'DATA FOR TURN ',ITURN,' READ'
      GO TO 10
C
C  H: HISTORY OF IMPORTANT PARAMETERS COLLECTED ON TAPE9 EACH STEP
```

```
C
  400 CALL HISTORY(.TRUE.)
      GO TO 10
C
C
C  Z: REWIND GRAPHICAL OUTPUT TAPE TO BEGINNING (PRESENTLY
C     ONLY WAY TO GET AT A RECORD BEFORE THE CURRENT ONE)
C
  500 REWIND(7)
C  Save logical telling us whether we have initialized
C  DI-3000 or not and reset it after restoring COMMONs.
      SAVND=NODRAW
C      DO 510 I=1,ISIZPH
C        ISVARY(I)=IPHARY(I)
C  510 CONTINUE
      CALL GET(.TRUE.,ITURN,DONE)
      NODRAW=SAVND
C      DO 520 I=1,ISIZPH
C        IPHARY(I)=ISVARY(I)
C  520 CONTINUE
      POSTP=.FALSE.
      PRINT *, 'DATA FOR TURN ',ITURN,' READ'
      GO TO 10
C
C  X: JUST PLOT EVERYTHING FROM HERE ON OUT
C
  600 CONTINUE
C  Save logical telling us whether we have initialized
C  DI-3000 or not and reset it after restoring COMMONs.
      SAVND=NODRAW
      CALL GET(.TRUE.,ITURN,DONE)
      NODRAW=SAVND
      IF(DONE)THEN
        PRINT *,'ALL PLOTS COMPLETED'
        GO TO 10
      ENDIF
      PRINT *, 'DATA FOR TURN ',ITURN,' READ'
      IF(IOPT.GE.0)THEN
        IF(MTCHSI .OR. MTCH95)THEN
C  Draw matched contour.
          CALL REFCONT
        ELSE
C  Draw bucket.
          THETAS=0.0
          CALL BUCKIT(IOPT,THETAS,DELCON)
        ENDIF
      ENDIF
      DO 610 I=1,NC
```

```
        CURVE(I,1)=CURVE(I,1)+DTHCURV
        CURVE(I,2)=CURVE(I,2)+DECURV
  610 CONTINUE
      CALL PHPLT
      GO TO 600
C
C  Use specified graphics and history files.
C
  700 CONTINUE
      OGRAFIL=GRAFIL
      OHFIL=HFIL
      OMRFIL=MRFIL

      READ(5,USEFIL)
      IF(GRAFIL .NE. OGRAFIL)THEN
C  First close old file if this is a new one
        IF(OGRAFIL.NE.' ')CLOSE(UNIT=7)
C  See if the requested file exists.
        INQUIRE(FILE=GRAFIL,EXIST=EXFG)
C  Exit if it doesn't
        IF(.NOT.EXFG)THEN
          PRINT *,'FILE NAMED ',GRAFIL,' CANNOT BE FOUND '
          GOTO 710
        ENDIF
C  Print reassurance if it does
        PRINT *,'GRAPHICAL DATA FROM FILE',GRAFIL
C  Open the requested file.  Note that we haven't read
C  anything yet.
        OPEN(UNIT=7,FILE=GRAFIL,ACCESS='SEQUENTIAL',
     1       STATUS='OLD')
      ENDIF
  710 CONTINUE
      IF(HFIL .NE. OHFIL)THEN
C  First close any old history file.
        IF(OHFIL.NE.' ')CLOSE(UNIT=9)
C  See if the requested file exists.
        INQUIRE(FILE=HFIL,EXIST=EXFG)
C  Exit if it doesn't
        IF(.NOT.EXFG)THEN
          PRINT *,'FILE NAMED ',HFIL,' CANNOT BE FOUND '
          GO TO 720
        ENDIF
C  Print reassurance if it does
      PRINT *,'HISTORY DATA FROM FILE',HFIL
C  Open the requested file.
      OPEN(UNIT=9,FILE=HFIL,ACCESS='SEQUENTIAL',
     1     STATUS='OLD')
      ENDIF
```

```
  720 CONTINUE
      IF(MRFIL .NE. OMRFIL)THEN
C  First close old file if this is a new one
        IF(OMRFIL.NE.' ')CLOSE(UNIT=20)
C  See if the requested file exists.
        INQUIRE(FILE=MRFIL,EXIST=EXFG)
C  Exit if it doesn't
        IF(.NOT.EXFG)THEN
          PRINT *,'FILE NAMED ',MRFIL,' CANNOT BE FOUND '
          GOTO 730
        ENDIF
C  Print reassurance if it does
        PRINT *,'GRAPHICAL DATA FROM FILE',MRFIL
C  Open the requested file.  Note that we haven't read
C  anything yet.
        OPEN(UNIT=20,FILE=MRFIL,ACCESS='SEQUENTIAL',
     1       STATUS='OLD')
       ENDIF
  730 CONTINUE
      GO TO 10
C
C  M: DISPLAY MOUNTAIN RANGE(s)
C
  800 CONTINUE
      CALL MRPLT
      GO TO 10
C
C  Q: QUIT PROGRAM ENTIRELY; NOTHING FURTHER TO DO
C
  900 CONTINUE
      PRINT *, 'QUIT COMMAND'
C
C  Terminate HIGZ
C
      IF(.NOT.NODRAW)CALL IGEND
C
      END
```

Chapter 5

Programming

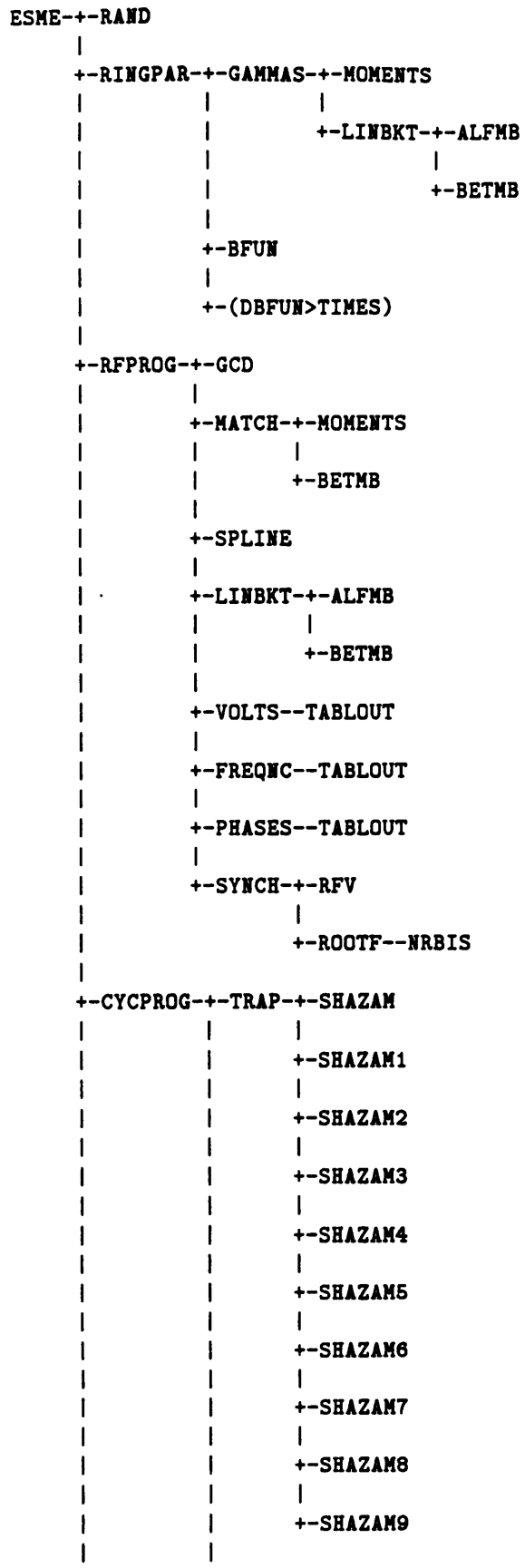
In this chapter some of the basic structure of the program is presented, and the use of certain VAX-based facilities in the development of the code is discussed. Though somewhat specialized, it is felt that the utility of these features to those who might be willing and able to use them is of great enough importance to warrant some description here. This chapter is primarily of interest to programmers. There is no need to make use of the material in this section in order to use the program. Furthermore, this section alone is not likely to make the reader an ESME expert. It is merely the hope of the authors that the material presented here will ease the process of incorporating changes and additions to the code.

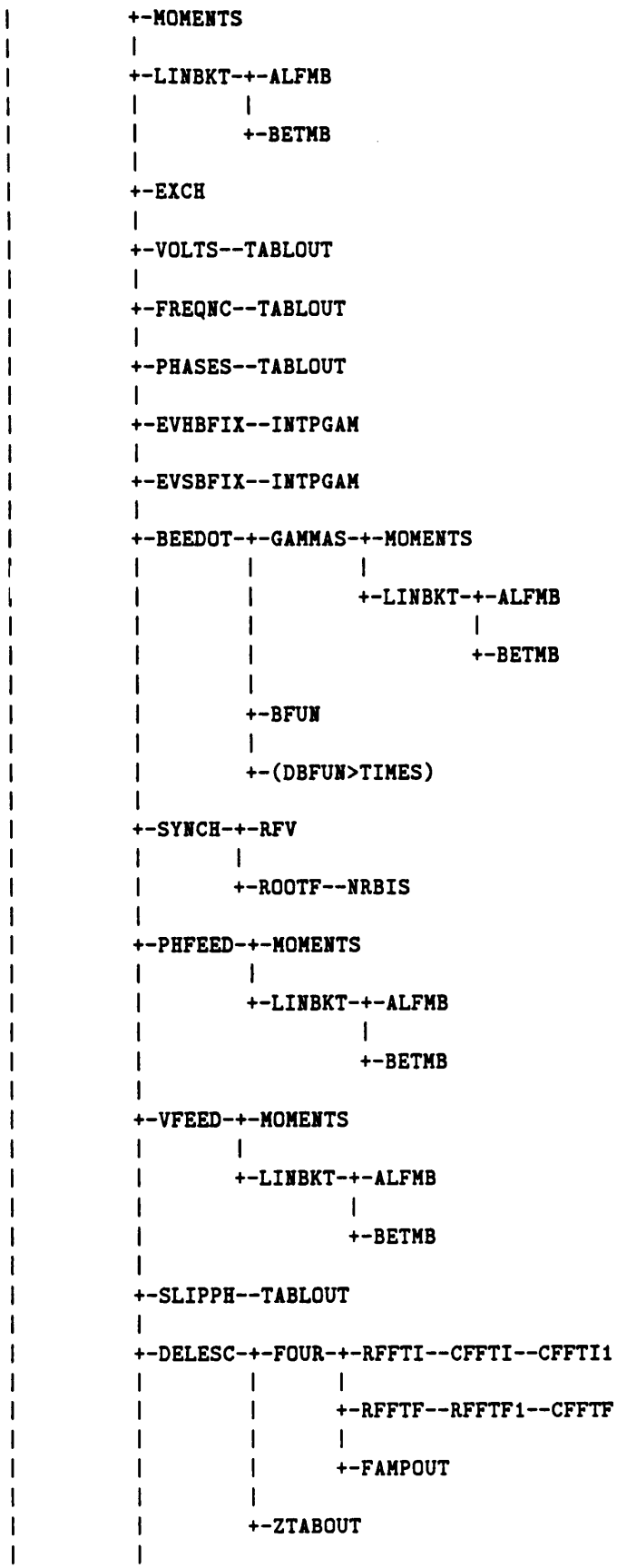
5.1 Program Basics

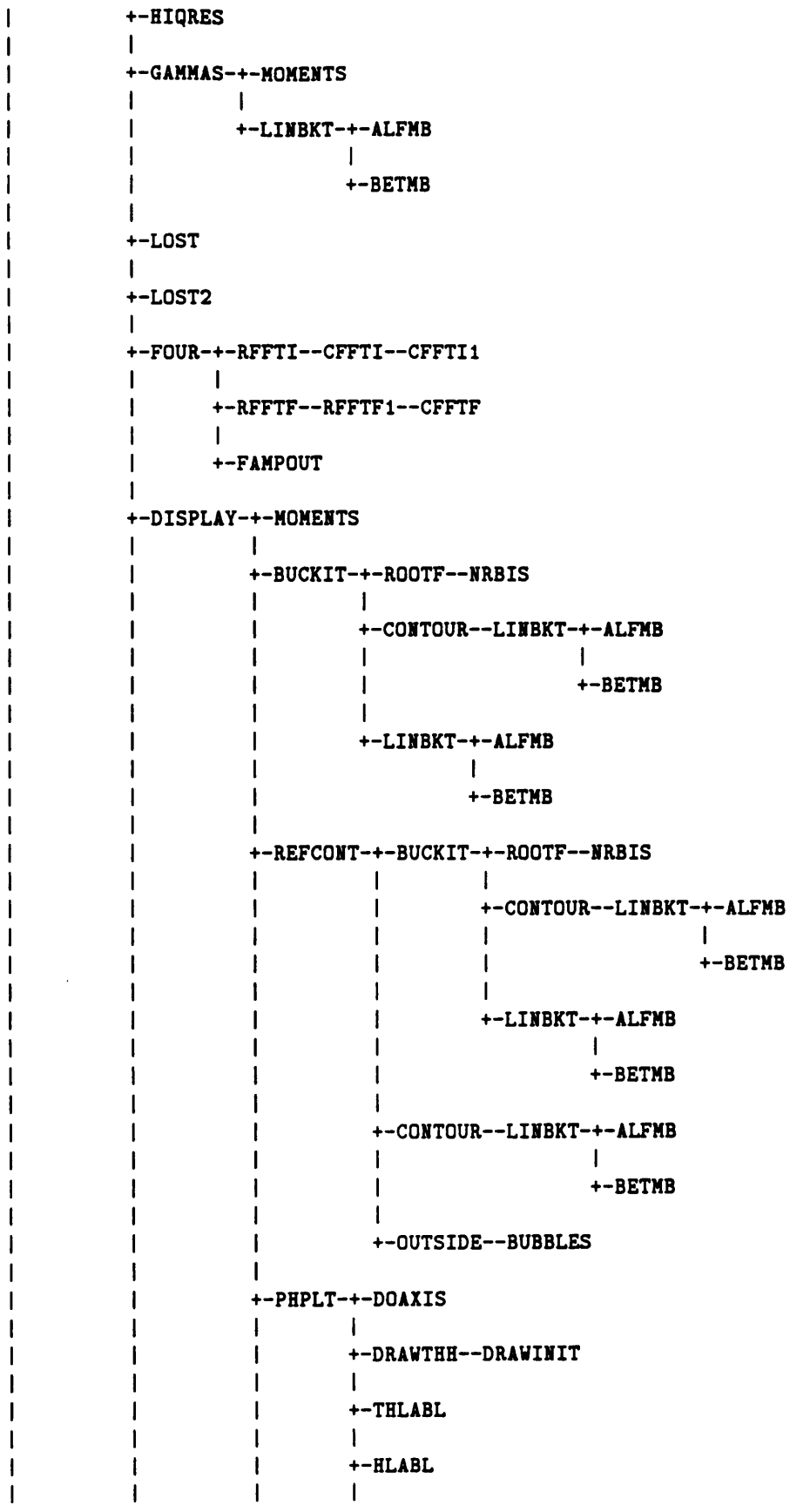
The first part of this section describes the overall structure of ESME. The second section details the main tracking loop. The third section lists the important variables in ESME, which are held in COMMON and thus available in SHAZAM.

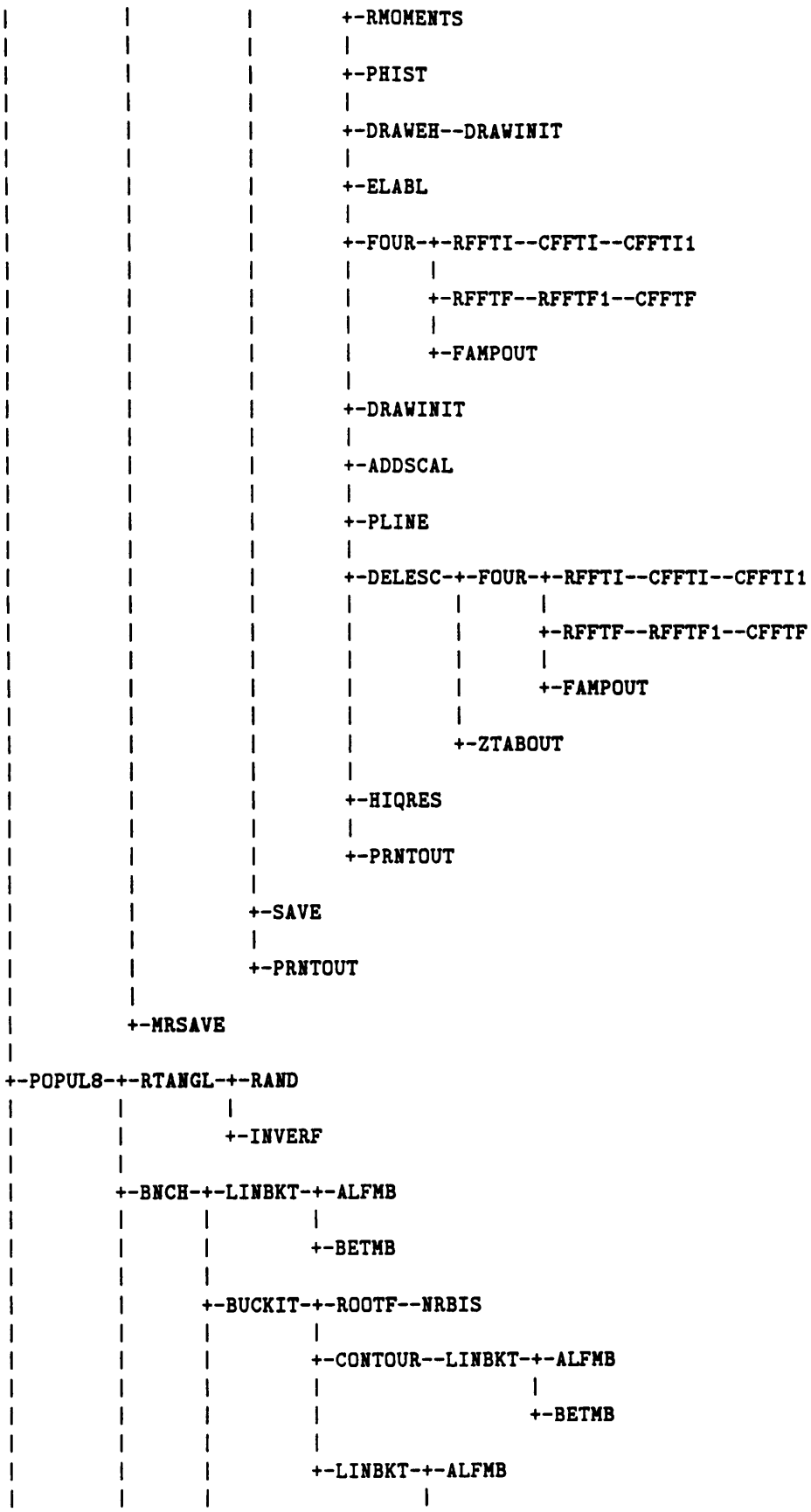
5.1.1 Program Structure

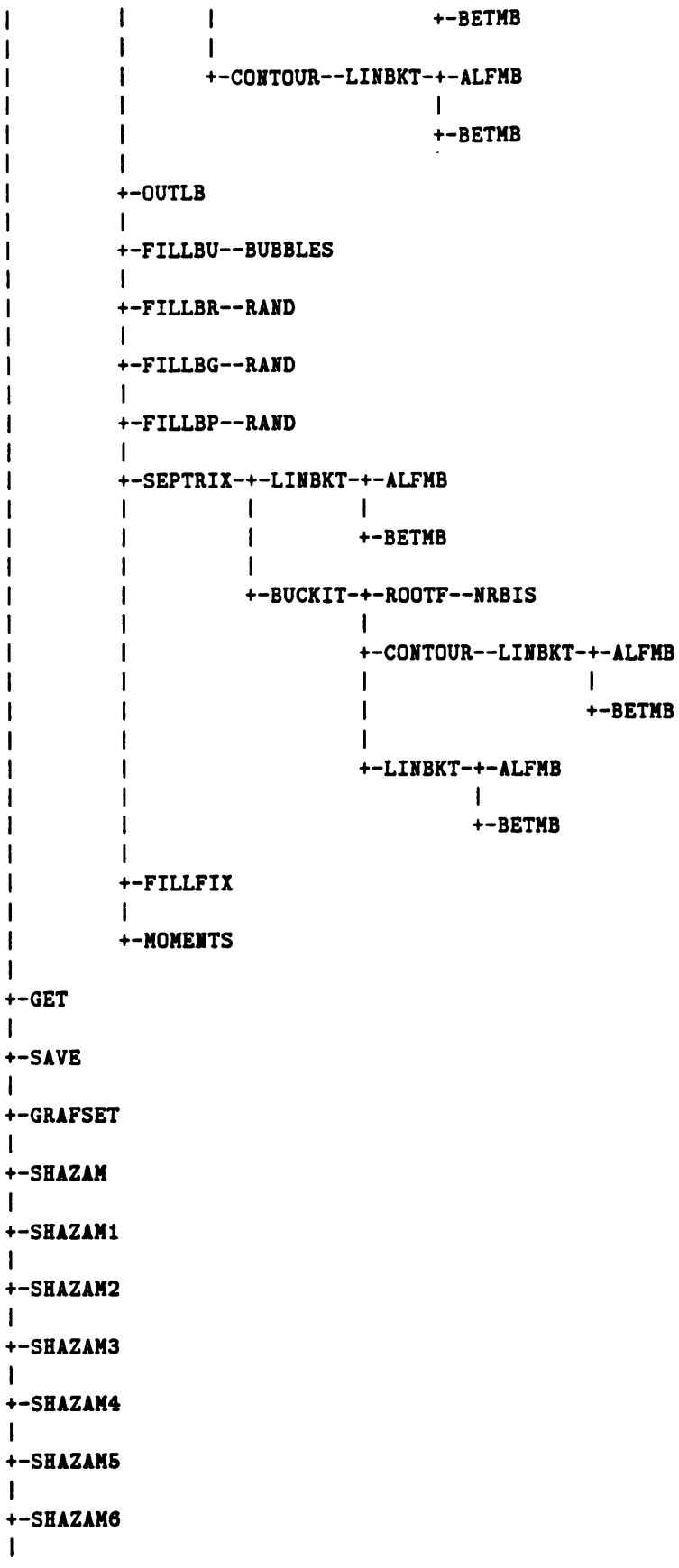
The basic pattern of ESME is a main program which calls subroutines selected by single letter commands in the input stream. The called routine reads in any needed parameters; it, and any dependent subroutines, carry out calculations for a distinct phase of the calculation or for a distinct accelerator subsystem. The program is integrated by putting particle coordinates and system variables into named commons each of which contain a group of closely related quantities. Higher level subroutines communicate through common. Certain lower level routines and some utility routines shared among different functional areas pass data through calling lists. Nearly all system and coordinate variables are stored in common blocks; only loop counters and a few intermediate results are local variables. Thus, the program is divided into numerous functional modules but important variables are global. The large number of parameters often required to specify a distribution and the rf systems which act upon it encourage this structure, as well as making it easier to incorporate changes into the code. A schematic tree diagram indicating the program flow follows.

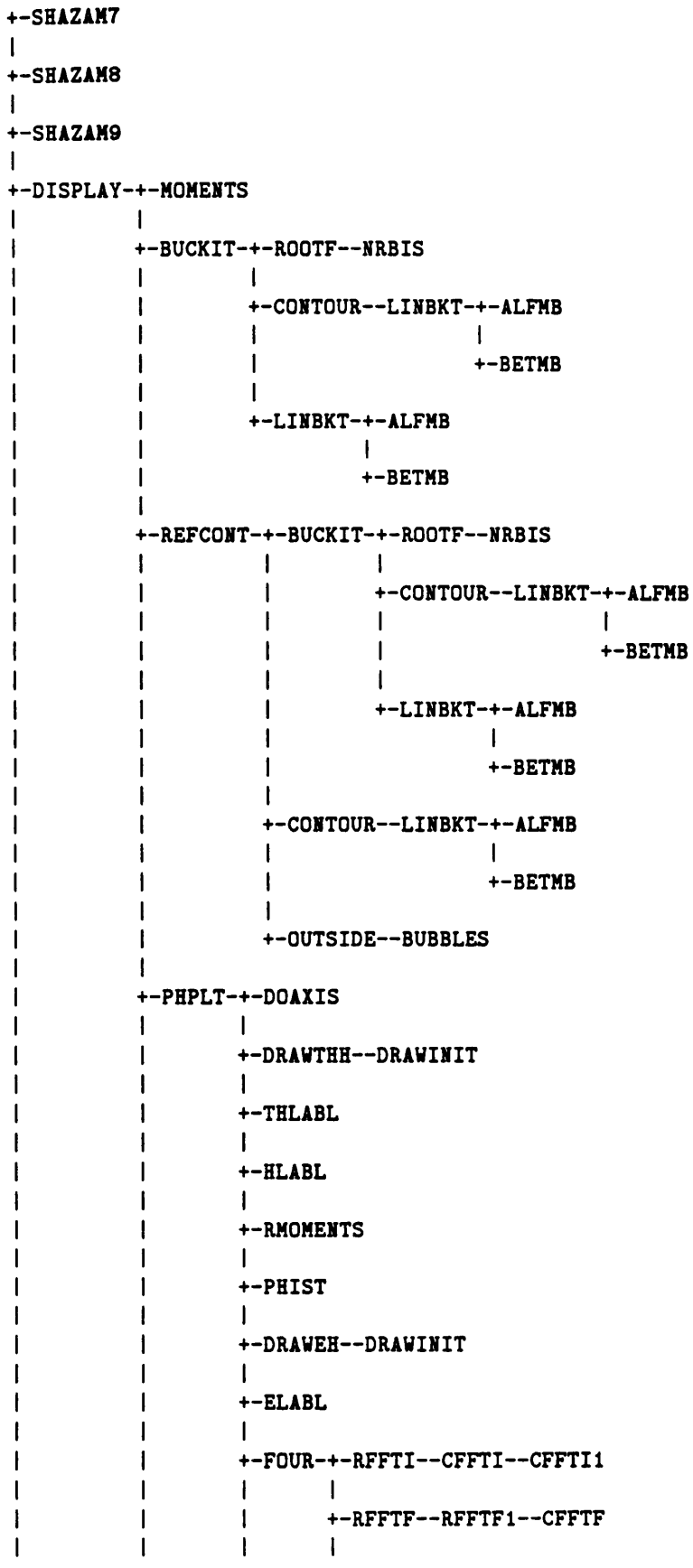

ESME-+-RAND
+-RINGPAR-+-GAMMAS-+-MOMENTS
+-LINBKT-+-ALFMB
+-BETMB
+-BFUN
+-(DBFUN>TIMES)
+-RFPROG-+-GCD
+-MATCH-+-MOMENTS
+-BETMB
+-SPLINE
+-LINBKT-+-ALFMB
+-BETMB
+-VOLTS--TABLOUT
+-FREQNC--TABLOUT
+-PHASES--TABLOUT
+-SYNCH-+-RFV
+-ROOTF--NRBIS
+-CYCPROG-+-TRAP-+-SHAZAM
+-SHAZAM1
+-SHAZAM2
+-SHAZAM3
+-SHAZAM4
+-SHAZAM5
+-SHAZAM6
+-SHAZAM7
+-SHAZAM8
+-SHAZAM9


+-MOMENTS
+-LINBKT-+-ALFMB
+-BETMB
+-EXCH
+-VOLTS--TABLOUT
+-FREQNC--TABLOUT
+-PHASES--TABLOUT
+-EVHBFIX--INTPGAM
+-EVSBFIX--INTPGAM
+-BEEDOT-+-GAMMAS-+-MOMENTS
+-LINBKT-+-ALFMB
+-BETMB
+-BFUN
+-(DBFUN>TIMES)
+-SYNCH-+-RFV
+-ROOTF--NRBIS
+-PHFEED-+-MOMENTS
+-LINBKT-+-ALFMB
+-BETMB
+-VFEED-+-MOMENTS
+-LINBKT-+-ALFMB
+-BETMB
+-SLIPPH--TABLOUT
+-DELESC-+-FOUR-+-RFFTI--CFFTI--CFFTI1
+-RFFTF--RFFTF1--CFFTF
+-FAMPOUT
+-ZTABOUT


+-HIQRES
+-GAMMAS-+-MOMENTS
+-LINBKT-+-ALFMB
+-BETMB
+-LOST
+-LOST2
+-FOUR-+-RFFTI--CFFTI--CFFTI1
+-RFFTF--RFFTF1--CFFTF
+-FAMPOUT
+-DISPLAY-+-MOMENTS
+-BUCKIT-+-ROOTF--NRBIS
+-CONTOUR--LINBKT-+-ALFMB
+-BETMB
+-LINBKT-+-ALFMB
+-BETMB
+-REFCONT-+-BUCKIT-+-ROOTF--NRBIS
+-CONTOUR--LINBKT-+-ALFMB
+-BETMB
+-LINBKT-+-ALFMB
+-BETMB
+-CONTOUR--LINBKT-+-ALFMB
+-BETMB
+-OUTSIDE--BUBBLES
+-PHPLT-+-DOAXIS
+-DRAWTHH--DRAWINIT
+-THLABL
+-HLABL


+-RMOMENTS
+-PHIST
+-DRAWEH--DRAWINIT
+-ELABL
+-FOUR-+-RFFTI--CFFTI--CFFTI1
+-RFFTF--RFFTF1--CFFTF
+-FAMPOUT
+-DRAWINIT
+-ADDSCAL
+-PLINE
+-DELESC-+-FOUR-+-RFFTI--CFFTI--CFFTI1
+-RFFTF--RFFTF1--CFFTF
+-FAMPOUT
+-ZTABOUT
+-HIQRES
+-PRNTOUT
+-SAVE
+-PRNTOUT
+-MRSAVE
+-POPUL8-+-RTANGL-+-RAND
+-INVERF
+-BNCH-+-LINBKT-+-ALFMB
+-BETMB
+-BUCKIT-+-ROOTF--NRBIS
+-CONTOUR--LINBKT-+-ALFMB
+-BETMB
+-LINBKT-+-ALFMB


+-BETMB
+-CONTOUR--LINBKT-+-ALFMB
+-BETMB
+-OUTLB
+-FILLBU--BUBBLES
+-FILLBR--RAND
+-FILLBG--RAND
+-FILLBP--RAND
+-SEPTRIX-+-LINBKT-+-ALFMB
+-BETMB
+-BUCKIT-+-ROOTF--NRBIS
+-CONTOUR--LINBKT-+-ALFMB
+-BETMB
+-LINBKT-+-ALFMB
+-BETMB
+-FILLFIX
+-MOMENTS
+-GET
+-SAVE
+-GRAFSET
+-SHAZAM
+-SHAZAM1
+-SHAZAM2
+-SHAZAM3
+-SHAZAM4
+-SHAZAM5
+-SHAZAM6


+-SHAZAM7
+-SHAZAM8
+-SHAZAM9
+-DISPLAY-+-MOMENTS
+-BUCKIT-+-ROOTF--NRBIS
+-CONTOUR--LINBKT-+-ALFMB
+-BETMB
+-LINBKT-+-ALFMB
+-BETMB
+-REFCONT-+-BUCKIT-+-ROOTF--NRBIS
+-CONTOUR--LINBKT-+-ALFMB
+-BETMB
+-LINBKT-+-ALFMB
+-BETMB
+-CONTOUR--LINBKT-+-ALFMB
+-BETMB
+-OUTSIDE--BUBBLES
+-PHPLT-+-DOAXIS
+-DRAWTHH--DRAWINIT
+-THLABL
+-HLABL
+-RMOMENTS
+-PHIST
+-DRAWEH--DRAWINIT
+-ELABL
+-FOUR-+-RFFTI--CFFTI--CFFTI1
+-RFFTF--RFFTF1--CFFTF

```
|          |        |      +-FAMPOUT
|          |        |
|          |        +-DRAWINIT
|          |        |
|          |        +-ADDSCAL
|          |        |
|          |        +-PLINE
|          |        |
|          |        +-DELESC-+-FOUR-+-RFFTI--CFFTI--CFFTI1
|          |        |        |       |
|          |        |        |       +-RFFTF--RFFTF1--CFFTF
|          |        |        |       |
|          |        |        |       +-FAMPOUT
|          |        |        |
|          |        |        +-ZTABOUT
|          |        |
|          |        +-HIQRES
|          |        |
|          |        +-PRNTOUT
|          |
|          +-SAVE
|          |
|          +-PRNTOUT
|
+-HISTORY-+-GLABL
|         |
|         +-GETDAT
|
+-FFTSET--FOUR-+-RFFTI--CFFTI--CFFTI1
|              |
|              +-RFFTF--RFFTF1--CFFTF
|              |
|              +-FAMPOUT
|
+-BEAMSC
|
+-LOWLVL-+-MOMENTS
|        |
|        +-LINBKT-+-ALFMB
|                 |
|                 +-BETMB
|
+-MRINIT
|
+-MRPLT--PLINE
```

This is a detached tree starting at the subprogram 'CFFTF'

```
CFFTF--CFFTF1-+-PASSF4
              |
              +-PASSF2
              |
              +-PASSF3
              |
```

```
            +-PASSF5
            |
            +-PASSF
```

Other routines (usually passed as arguments):

```
DRAWPH--DRAWINIT
BETPRMB
ALFPRMB
PREFIX
BVOLT
ENGFMT
DRFV
DBVOLT
```

5.1.2 Important Variables

The various named common blocks are listed below. Each entry includes a statement of purpose and a list of all the included variables. If the variables come primarily from one subroutine, that subroutine is named on the last line of the entry; likewise, if a significant number of the variables come directly from the input data, the controlling NAMELIST is identified also.

```
C  Block parameterizing isolated or barrier bucket generation.
      PARAMETER (ISRC3 = 10)
      COMMON /BKTSUP/ THL(ISRC3),THU(ISRC3)
Principal Source: RFPROG                          Controlling NAMELIST: /RF/

C  THE MAIN STORAGE BLOCK. CONTAINS NPHASE PHASE SPACE COORDINATES AND
C  THE POINTERS TO SEPARATE PARTITIONS OF THE DISTRIBUTION
      PARAMETER (NPHASE = 50000, NKLIM=50)
      COMMON /BLANK/ PHASE(0:NPHASE,2)
      COMMON /IBLANK/ KOUNT,KLASSES,KLIMIT(0:NKLIM)
      COMMON /LBLANK/ PARTION
      LOGICAL PARTION
Principal Source: POPUL8                          Controlling NAMELIST: /POPL8/

C  BUCKET PARAMETERS TURN-BY-TURN
      COMMON /BUCKET/ PHISL,PHIUSFP,GNUS,SBCKT,HBCKT
      COMMON /IBUCKET/ NFIRST,NLAST
Principal Source: CYCPROG

C  BUNCH PARAMETERS TURN-BY-TURN
      COMMON /BUNCH/ THBAR,EBAR,THRMS,ERMS,EPSILON,SBUNCH
Principal Source: CYCPROG

C  CONTAINS MATHEMATICAL AND PHYSICAL CONSTANTS, PROGRAM CONSTANTS, AND
C  CONVERSION FACTORS FROM EXTERNAL TO INTERNAL UNITS
      COMMON /CONST/ HALFPIE,PIE,TWOPIE,FOURPIE,RADDEG,DEGRAD,BIG,SMALL,
     1     C,EMCSQ,EMCSQS,RP,QE,ZNAUGHT
```

```
Principal Source: BLOCK DATA

C  MISCELANEOUS PARAMETERS AND INTERMEDIATE RESULTS TURN-BY-TURN
      PARAMETER (KSRC = 10)
      COMMON /CURRENT/ EV(KSRC),PSIO(KSRC),PSI(KSRC),FREQ(KSRC),
     1     ES,PS,RS,BETAS,BETASQ,ETA,EO,PO,BETAO,ACCEL,
     2     THREF,EREF,TAU,TIME,DEBFLD,PDOT,ALFAO,ALFA1,ALFA2,
     3     ELO,EHI,EAVG,TGCURR,GMSQINV,DEO,DEBAR,DELEREF,DELR,
     4     RFFREQ,PHIS,DELES
      COMMON /ICURRENT/ M,MM,NTURNS,MHIST
      COMMON /LCURRENT/ TRPSW,TRANSW
      LOGICAL TRPSW,TRANSW
Principal Source: CYCPROG

C  PHASE SPACE COORDINATES OF BUCKET OR OTHER CONTOUR OF INTEREST
      PARAMETER (ICURVE = 5001)
      COMMON /CURVES/ CURVE(ICURVE,2),TURNPT(4,3),NC
      REAL CURVE,TURNPT
C
Principal Source: BUCKET & REFCONT

C  PARAMETERS GOVERNING DURATION & OPTIONAL FEATURES OF THE TRACKING CALCULATION
      PARAMETER (NTRP = 4)
      COMMON /CYCLP/ TBEGIN,TEND,TSTART,TSTOP,TTRACK,DES,ACCELO,
     1     ETATRP,PHISTRP
      COMMON /ICYCLP/ MSTEP,LGRTHM,ITRAP(NTRP),MGRACE(NTRP),NTRAP,NCAV
      COMMON /LCYCLP/ HISTRY,MOMNTS,BBDRY,EAPROX
      LOGICAL HISTRY,MOMNTS,BBDRY,EAPROX
Principal Source: CYCPROG                          Controlling NAMELIST: /CYCLE/

C  PARAMETERS DEFINING FEEDBACK LOOPS (CURRENTLY PHASE & VOLTAGE - NO RADIAL)
      PARAMETER (IFBPRS = 1001)
      COMMON /FEEDS/ DLIMIT,PLIMIT,FBFACT,VLIMIT,
     1     VFBFCTR,FBPRS(IFBPRS),PSIADD,DAMPL,W(IFBPRS)
      COMMON /IFEEDS/ NTUAVG,NTURES,IFTB
      COMMON /LFEEDS/ PHFBON,VFBON,USEWT
      LOGICAL PHFBON,VFBON,USEWT
Principal Source: LOWLVL                           Controlling NAMELIST: /LLRF/

C  PARAMETERS DEFINING AND RESULTS OF FOURIER TRANSFORM OF CHARGE DISTRIBUTION
      PARAMETER (IFFT = 1024, NFMAX = 50)
      COMMON /FOURIR/ BINWIDT,FAMPL(NFMAX),FAZE(NFMAX),SCFRAC,SCTHLO,
     +   WSAVE(3*IFFT+15),F(IFFT),G(IFFT)
      COMMON /IFOURIR/ NBINFFT,MFFT,MAXFFTB,NF(NFMAX),NNF,NCC
      COMMON /LFOURIR/ FFTON
      LOGICAL FFTON
Principal Source: FOURFIT                          Controlling NAMELIST: /FFT/
```

```
C  Parameters defining transition gamma time dependence and
C  transition phase switch timing.
      COMMON /GAMJMP/ GAMPAR(5),ETAJMP
      COMMON /IGAMJMP/ KINDG
      COMMON /LGMAJMP/ GMAJMP
      LOGICAL GMAJMP
Principal Source: RINGPAR                          Controlling NAMELIST: /RING/

C  PARAMETERS DEFINING DESIRED GRAPHICAL OUTPUT
      PARAMETER (ISIZGM = 10000, ISIZPH = 2000)
      COMMON /GRAFIX/ THPMIN,THPMAX,DEPMIN,DEPMAX,DTHCURV,DECURV,
     1     THBMIN,THBMAX,EBMIN,EBMAX,
     2     HBCKTN,SBCKTN,SREF,
     3     SCBMIN,SCBMAX,RBMIN,RBMAX,DELCON
     4     XCRNR,YCRNR,XAXISL,YAXISL
      COMMON /IGRAFIX/ MPLOT,IOPT,IEREF,NBINTH,NBINE,KNTLIM,
     1     IFBMIN,IFBMAX,IDEV,NPJMP,KLPLOT,NBPFFT,IMETA
      COMMON /LGRAFIX/ PLTSW(20),TITLE,MTCHSI,MTCH95,DRWREF,NODRAW,
     1     POSTP
      LOGICAL PLTSW,TITLE,MTCHSI,MTCH95,DRWREF,NODRAW,POSTP
Principal Source: GRAFSET                          Controlling NAMELIST: /GRAPH/

C  DESCRIPTIVE HEADING FOR GRAPHICAL OUTPUT
      PARAMETER (MAXTTL=50)
      COMMON /HEADING/ TITL
      COMMON /IHEADING/ TITLEN
      CHARACTER TITL*50
      INTEGER TITLEN
Principal Source: GRAFSET

C  Include specifying mountain range parameters.
C  For saving data (plotting parameters are local to MRPLT).
      COMMON /MRANGE/ MRTHMINB,MRTHMAXB,TMBEGIN,TMEND
      REAL MRTHMINB,MRTHMAXB,TMBEGIN,TMEND
      COMMON /IMRANGE/ MRMPLOT,MRNBIN
      INTEGER MRMPLOT,MRNBIN
Principal Source: MRINIT                           Controlling NAMELIST: /MRANGE/

C  THE PARAMETERS DEFINING THE INITIAL PHASESPACE DISTRIBUTION(S)
      COMMON /POPLATE/ THMIN,THMAX,REMIN,REMAX,
     1     SBNCH,THOFF,EOFF,THTRAN,ETRAN
      COMMON /IPOPLATE/ KIND,IPOP,NTH,NE,NPOINT,ISEED
Principal Source: POPUL8                           Controlling NAMELIST: /POPL8/

C  PARAMETERS DEFINING THE RF SYSTEMS
      PARAMETER (NSRC = 10)
      COMMON /RFP/
     1     VI(NSRC),VF(NSRC),TVBEG(NSRC),TVEND(NSRC),VTABL(5,21,NSRC),
```

```
     2     FRI(NSRC),FRF(NSRC),TFBEG(NSRC),TFEND(NSRC),FTABL(5,21,NSRC),
     3     PSII(NSRC),PSIF(NSRC),TPBEG(NSRC),TPEND(NSRC),
     4     PTABL(5,21,NSRC),C1(NSRC),C2(NSRC),DELTRF(NSRC),
     5     HDECR,SDECR,PHISLIM
      COMMON /IRFP/ NRF,H(NSRC),HW(NSRC),HGCD,HMAX,ISYNC,
     1     KURVE(NSRC),NTV(NSRC),NTABV(NSRC),
     2     KURVF(NSRC),NTF(NSRC),NTABF(NSRC),
     3     KURVP(NSRC),NTP(NSRC),NTABP(NSRC)
      COMMON /LRFP/ VKON,FRKON,PHKON,PHSLIP,EXCHRF,HOLDBH,HOLDBA,CNTINU,
     1     VMATCHI(NSRC),VMATCHF(NSRC)
      INTEGER H,HW,HGCD,HMAX
      LOGICAL VKON,FRKON,PHKON,PHSLIP,VMATCHI,VMATCHF
      LOGICAL EXCHRF,HOLDBH,HOLDBA,CNTINU(NSRC)
Principal Source: RFPROG                          Controlling NAMELIST: /RFP/

C  THE LATTICE PARAMETERS INCLUDING TIME DEPENDENCES
      COMMON /RINGP/ REQ,GAMMAT,ALPHA0,ALPHA1,ALPHA2,ALPHA3,
     1     TAUINF,EKOI,EKOF,EKS,TI,TF,PI,PF,EKIDOT,EKFDOT,
     2     PIDOT,PFDOT,THLO,THHI,THRNG,FRAC,PIPRAD
      COMMON /IRINGP/ KURVEB
      COMMON /LRINGP/ JNRAMP,EBDRY
      LOGICAL JNRAMP,EBDRY
Principal Source: RINGPAR                         Controlling NAMELIST: /RING/

C  AREA TO BE USED WITHIN A SINGLE SUBROUTINE ON A GIVEN CALL (MOSTLY GRAPHICS)
      COMMON /SCRATCH/ SCRPAD(4422)

C  Contains "spare" variables which are written to history
C  along with everything else.  These can be whatever the user
C  wishes; these commons are available to CYCPROG and SHAZAM.
C  Also contains labels for spares used in history plots.
      PARAMETER (NSPARE = 10, TITLL=14, UNITL=7)
      COMMON /SPARES/ SPARE(NSPARE)
      COMMON /CSPARES/ SPLABL(NSPARE),SPUNIT(NSPARE)
      CHARACTER*14 SPLABL
      CHARACTER*7 SPUNIT
Principal Source: CYCPROG

C  PARAMETERS DEFINING THE SPACE CHARGE & WALL IMPEDANCE ENERGY/TURN
      PARAMETER (ITBOUT=40, IRLLEN=30)
      COMMON /SPCHG/ A,B,ENQ,ZTABL(ITBOUT,9),EVSC(1024),
     1     RESTBL(IRLLEN,4)
      COMMON /ISPCHG/ NZ,NR,MSC
      COMMON /LSPCHG/ SCON
      LOGICAL SCON
Principal Source: FOURFIT                         Controlling NAMELIST: /SCHG/

C  A BLOCK TO CONTAIN CPU TIME SINCE START
```

```
      COMMON/TIMES/ CPUBEG,CPUNOW
      COMMON/ITIMES/ ITIME,IRCOD1,IRCOD2
      DOUBLE PRECISION CPUBEG, CPUNOW

C  Record the version number for program documentation & debugging.
C  VERNUM is the version of ESME
C  PVER is the post-processor version number.
C  Version 7.10 (beta) 1 DEC 1989
      PARAMETER (VERNUM=7.10, PVER=1.00)

C  Commons for voltage due to resonator.
      PARAMETER (MAXRB=10000, MAXRES=30)
      COMMON /ZRES/ VRES(MAXRB),RVO(MAXRES),RVODOT(MAXRES),
     +              DQDTO(MAXRES)
      COMMON /IZRES/ NBRES
      COMMON /LZRES/ QREZON
      LOGICAL QREZON
Principal Source: FOURFIT
```

5.1.3 Main Tracking Loop

The fundamental mapping or tracking algorithm is contained in the subroutine CYCPROG which is called when a **T** command is encountered in the data stream. The subroutine reads data for the number of turns to track etc. and then tracks successive turns applying the difference equations to each particle on each turn. At the end of each turn the tracking duration is checked, various properties of the distribution are calculated, system parameters are updated, and tests are applied to see if any selected parameter has reached a desired endpoint. The loop structure is schematized in Fig. 5.1. The boxed numbers in the figure are Fortran statement numbers relating the depicted structure to the Fortran listing. These markers should help the intensely curious or thoroughly sceptical find his way through the core code. An IF loop is depicted in the diagram by its head statement number in a box to the outside of a vertical line representing the scope of the loop; a DO loop is represented with its terminating number in a box at the end of the vertical line representing its scope. The names of loop counters are given within ovals. Steps of the calculation in italics are executed optionally according to logical switches and integer conditions set by the input data.

5.2 Code Management

The current version of ESME is largely an outgrowth of efforts to tailor the program to specific applications. The various efforts embarked upon to adapt the code for different purposes revealed that the code had diverged sufficiently enough from its original intent that a systematic overhaul was called for. The aims of this overhaul were to enhance the capabilities of the program somewhat, to clarify both the overall structure of the code and its command structure, and to make the process of adapting the code for specific problems slightly more appealing (less painful). The tools presented here were developed specifically to aid in the process of coding. The following two sections detail these tools and their use. It is not necessary to read the first section to make use of these tools, so the impatient reader may skip immediately to Section 5.2.2

5.2.1 Tools–An Overview

The development process was aided by VAX DEC/Module Management System (MMS) in conjunction with VAX DEC/Code Management System (CMS)[1], and Fermilab's CDF EXPAND[13] utility. CMS was used to maintain a source-code library for ESME. MMS was used to construct the executable version of the code.

MMS is patterned after the UNIX[2] *make* utility. At the heart of MMS is a description file, which MMS processes to determine what actions are necessary to produce an up-to-date executable version of the "target", which in the case of ESME is an executable version of the program. The description file consists of dependency "rules" and action lines. Each dependency rule consists of a target and its sources (e.g., A.EXE and A.OBJ). MMS compares the dates of the targets and sources to determine if the target is older than any of the sources. If so, the target needs to be "updated", and in that case the action specified on the action line is executed. A simple example of a description file for ESME might be:

```
ESME.EXE : ESME.OBJ
           LINK ESME
ESME.OBJ : ESME.FOR
           FORTRAN ESME
```

in which all the source code for ESME is found in ESME.FOR. Of course, the description given here is incomplete. However, the underlying principle itself is very simple– that of updating only those system components which it is necessary to update to build a new target. Using this system, the process of code development is simplified, since MMS can "pick up" any changes to the common CMS library. In addition, individual users can incorporate their own changes into the code very easily.

As mentioned earlier, the Fermilab utility EXPAND was used with MMS to build the program. Had ESME been intended only for use on VAX systems, EXPAND could have been dispensed with. This utility allows the user to build the program for various machine architectures[3] from the same cource code, in addition to offering various pre-processor options. Of course, in cases where the compiler for the machine architecture does not reside in the evironment (i.e., VAX) in which the code is prepared, MMS cannot direct the VAX to compile and link the code. However, presumably, once the source code has been prepared for the destination machine, it can be exported for compilation and linking there.

ESME is also maintained on a SUN workstation under UNIX. SCCS and Unix "make" are used to manage the code. Pre-processing is performed by the C pre-processor. Since all but small sections of the code are standard FORTRAN 77, the program is easily exported to other systems. The translation of pre-processor directives into PATCHY control statements is straightforward, so that anyone wishing to use PATCHY or derivatives of PATCHY to maintain the code should be able to do so.

5.2.2 Using the Tools

The relevant command procedures and MMS files are found in `DISK$PZ:[SBH.ESME.MMS]`. Generally, the user should first SET the appropriate CMS library and FETCH any routines to be modi-

[1] VAX, DEC/MMS, and DEC/CMS are trademarks of Digital Equipment Corporation.

[2] UNIX is a registered trademark of American Telephone and Telegraph Company.

[3] At present, EXPAND accepts the arguments VAX, CYBER, IBM, ACP_NODE, FTN77, and FPS for the /ENVIRONMENT qualifier.

fied. The addition of any routines which are not part of ESME will require the user to modify some of the files in `DISK$PZ:[SBH.ESME.MMS]`. Such operations will not be discussed here, though an examination of the relevant MMS (INC.MMS, ESME.MMS, ESMESRC.MMS) files should provide some illumination as to how to proceed in such a case.

Once the appropriate files have been modified, the VMS user may incorporate them into an executable version of ESME using the prepared command procedure MAKESME.COM. This procedure then prepares either an executable or source-code version of the program using the routines supplied by the user while supplying the remainder of the routines from ESME libraries. If the procedure USESME.COM has been invoked, it should be sufficient to simply enter the command `MAKESME`. The command procedure will then

1. Ask the user if he wishes to make ESME, ESMEPLOT, or both.
2. Ask whether the output is to be VAX executable code, IBM executable code, or source code.
3. Ask for a location in which to place the output; if one is not already available, then it will be created.
4. Query the user for source files (presumably modified).
5. Query the user for pre-processor and/or compile options. If the output is to be compiled on the IBM, the user will be asked for an account and password for this operation to take place.
6. Ask if the job is to be submitted to a batch queue.

The resultant executable (assuming no compile time errors) or source code file will be constructed in the specified directory. For compilation on CERNVM, the source code is copied and compiled on the IBM via a batch procedure submitted from VXCERN. Any other source code will have to be moved to the appropriate destination and compiled by whatever means are appropriate locally.

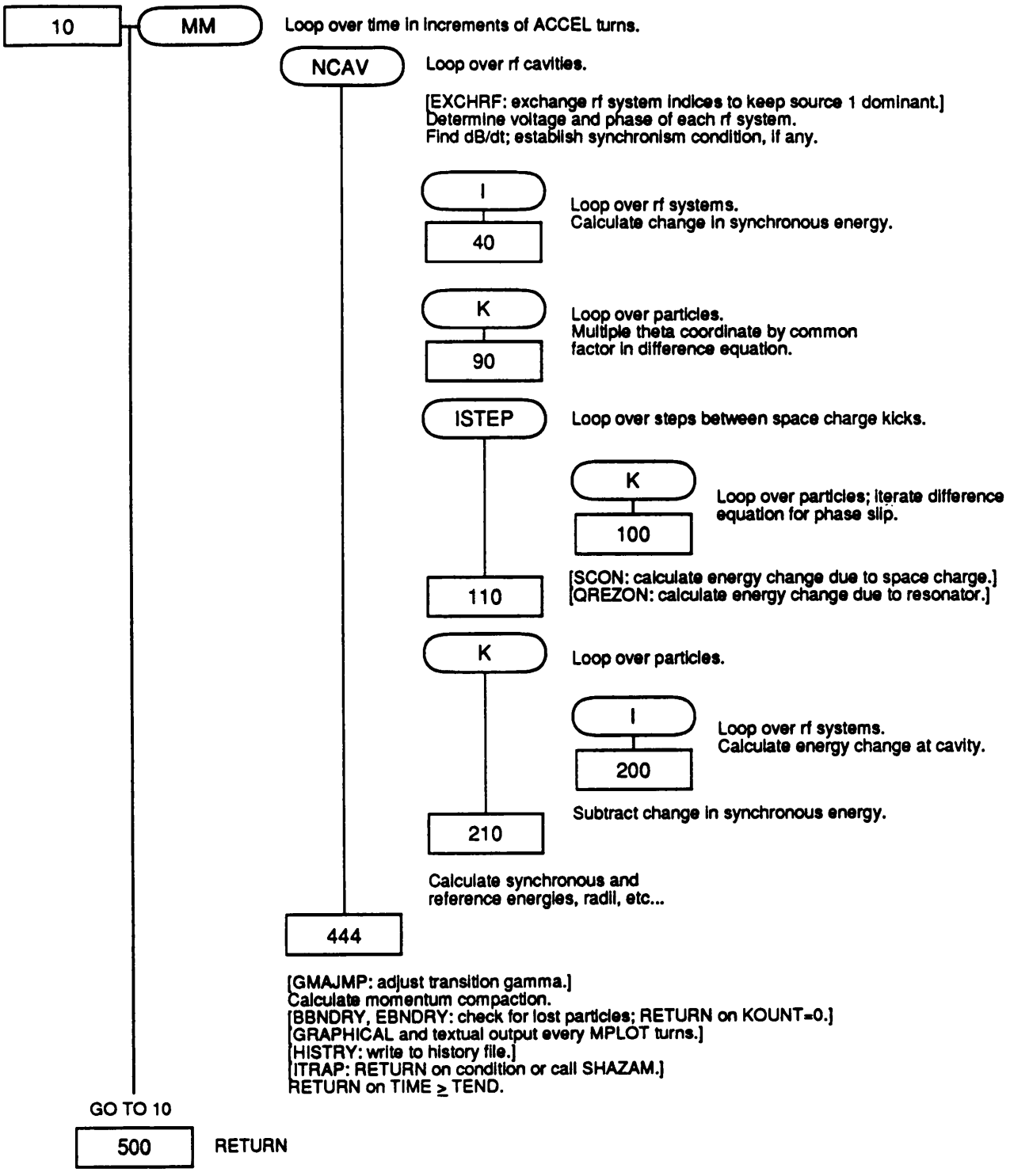


Figure 5.1: Schematic representation of the main tracking loop in ESME

Appendix A

Command NAMELIST Summary

This appendix provides a quick reference in alphabetical order for those who wish to avoid leafing through the entire document.

A-command, NAMELIST /RF/

Variable	Default Value	Default Unit	Description
NRF	1	-	Number of rf sources.
H(i), i = 1:10	1	-	Harmonic number of source i. (Integer.)
HW(i)	1	-	Limits source i to be "active" for $-180°/HW \leq \vartheta \leq 180°/HW$. (Integer.)
ISYNC	0	-	Synchronism condition to be imposed: 0 - None; voltages and phases remain as programmed 1 - Phase of rf waveform shifted to synchronous, stable point 2 - Magnitude of rf waveform scaled to give correct synchronous energy gain 3 - Source 2 acts as a Landau cavity to source 1[a].
EXCHRF	.T.	-	If .TRUE., sources are interchanged so that source 1 is always the greatest contributor to the bucket height.
VI(i)	0.0	MV	(Peak) voltage of source i at time TVBEG(i).
VF(i)	0.0	MV	(Peak) voltage of source i at time TVEND(i).
TVBEG(i)	0.0	s	Time at which rf voltage change begins.
TVEND(i)	0.0	s	Time at which rf voltage change ends.
KURVE(i)	0	-	Type of rf voltage variation between times TVBEG(i) and TVEND(i) for source i: 0 - None; voltage maintained at VI(i) 1 - Linear 2 - Isoadiabatic 3 - Sigmoid 4 - Cubic spline interpolation[b].
VKON	.T.	-	If .TRUE., activates voltage curves for all sources.
PSII(i)	0.0	deg	Phase of source i at time TPBEG(i).
PSIF(i)	0.0	deg	Phase of source i at time TPEND(i).
TPBEG(i)	0.0	s	Time at which rf phase change begins.
TPEND(i)	0.0	s	Time at which rf phase change ends.
KURVP(i)	0	-	Type of rf phase variation between times TPBEG(i) and TPEND(i) for source i: 0 - None; phase maintained at PSII(i) 1 - Linear 2 - Quadratic 4 - Cubic spline interpolation[b].
PHKON	.F.	-	If .TRUE., activates phase curves for all sources.
FRI(i)	0.0	MHz	Frequency of source i at time TFBEG(i).
FRF(i)	0.0	MHz	Frequency of source i at time TFEND(i).
TFBEG(i)	0.0	s	Time at which rf frequency change begins.
TFEND(i)	0.0	s	Time at which rf frequency change ends.

[a]Synchronism is only assured for sources 1 and 2.

[b]Fit to values read from file. See section 2.1.2.

A-command, NAMELIST /RF/ (continued)

Variable	Default Value	Default Unit	Description
KURVF(*i*)	0	-	Type of rf frequency variation between times TFBEG(*i*) and TFEND(*i*) for source *i*: 0 – None; frequency maintained at FRI(*i*) 1 – Linear 2 – Quadratic 4 – Cubic spline interpolation[a].
FRKON	.F.	-	If .TRUE., activates frequency curves for all sources.
CNTINU(*i*)	.F.	-	If .TRUE., sets starting point(s) of voltage, phase and/or frequency curve(s) for source *i* to be the current value(s), hence smoothly piecing curve segments together.
VMATCH(*i*)	.F.	-	If .TRUE., sets VI(*i*) to match the current distribution emittance[b].
HOLDBH	.F.	-	If .TRUE., the voltage of source 1[c] is varied such that the corresponding bucket height is multiplied by HDECR on successive turns.
HDECR	1.0	-	Factor by which (source 1) bucket height is adjusted on successive turns if HOLDBH = .TRUE.
HOLDBA	.F.	-	If .TRUE., the voltage of source 1[c] is varied such that the corresponding bucket area is multiplied by SDECR on successive turns.
SDECR	1.0	-	Factor by which (source 1) bucket area is adjusted on successive turns if HOLDBA = .TRUE.
PHISLIM	.95	-	The HOLDBH or HOLDBA option cannot reduce the (source 1) voltage such that $\sin\phi_s >$ PHISLIM.
DELTRF(*i*)	0.0	-	Momentum offset ($\Delta p/p$ with respect to the synchronous particle) at which source *i* is operated (by introducing the corresponding phase offset).
PHSLIP	.F.	-	If .TRUE., activates momentum offsets for all sources.

[a]Fit to values read from file. See section 2.1.2.

[b]Consequently, in this instance, the P-command (or its equivalent) should precede the A-command.

[c]The algorithms used to maintain the bucket height and area consider only a single source.

B-command, NAMELIST /SCHG/			
Variable	Default Value	Unit	Description
A	0.002	m	Effective beam radius.
B	0.05	m	Effective beampipe radius.
ENQ	2.E10	-	Number of protons to be represented by the distribution.
NZ	0	-	Number of impedance values to be read from file[a].
NR	0	-	Number of resonance values to be read from file[b].
SCON	.F.	-	If .TRUE., activates space-charge calculation.
MSC	1	-	Number of times collective effects are calculated between rf cavities.
QREZON	.F.	-	If .TRUE., activates high-Q resonance calculation.
NBRES	10,000	-	Number of bins to be used in the high-Q resonance calculation.
NNF	0	-	Number of Fourier harmonics written to the history file.
NF(1:NNF)	0	-	Harmonic numbers of the NNF Fourier components.
MAXFFTB	1024	-	Max. number of bins to be used in FFT.
MFFT	1	turn	Interval between calculations of the Fourier transform; an FFT is performed every MFFT turns.

[a]See section 2.1.3.

[b]See section 2.1.4.

F-command, NAMELIST /FFT/			
Variable	Default Value	Unit	Description
FFTON	F	-	If .TRUE., activates Fourier transform calculation.
NNF	0	-	Number of Fourier harmonics written to the history file.
NF(1:NNF)	0	-	Harmonic numbers of the NNF Fourier components.
NBINFFT	256	-	Number of bins to be used in FFT.
MFFT	1	turn	Interval between calculations of the Fourier transform; an FFT is performed every MFFT turns.
FFTOUT	F	-	If .TRUE., Fourier transform is printed.

H-command, NAMELIST /HISTRY/

Variable	Default Value	Default Unit	Description
IDEV	0	-	Virtual device number[a] for graphics output.
IMETA	-1	-	Directs graphics output: < 0 – Output is written to a metafile 0 – Output is directed to the device specified by IDEV > 0 – Output is written both to a metafile and to the device specified by IDEV.
NPLT(1:2,m)	0	-	Parameters to be plotted vs. each other from history records; NPLT(1,m) is the independent and NPLT(2,m) the dependent variable of the m-th history plot (m = 1:50).
NWRT(1:2,m)	0	-	Same as NPLT, except that the data pairs are written to logical unit 19 rather than being plotted.
			Real records: 1 – Time 2 – PHIS[b], "synchronous phase" 3 – PDOT, time derivative of synchronous momentum 4 – THBAR, mean value of ϑ of the distribution 5 – EBAR, mean energy of the distribution 6 – THRMS, rms value of ϑ of the distribution 7 – ERMS, rms energy spread of the distribution 8 – ES, synchronous energy 9 – E0, energy on the reference orbit 10 – ES – E0 11 – THREF, azimuth of the "reference particle"[c] 12 – EREF, energy of the "reference particle"[c] 13 – EPSILON, statistical "emittance" about barycentre of the distribution 14 – NUS, synchrotron frequency 15 – SBCKT, rf bucket area 16 – HBCKT, rf bucket height 17 – ETA, synchronous value of η 18 – ACCEL, number of beam turns per tracking step[d] 19 – TAU, synchronous revolution period 20 – PSIADD, phase feedback[e] 21 – DAMPL, voltage feedback factor[e] 22 – DELR, synchronous mean radial postion (relative to the reference orbit) 23 – RFFREQ, frequency of rf source 1.
			Integer records: 31 – Turn number 32 – KOUNT, number of surviving particles.
			Array records: 51-60 – SPARE(1-10), available for SHAZAM routines 101-110 – EV(1-10), voltage of rf source 1-10 111-120 – PSI(1-10), phase of rf source 1-10 121-130 – FREQ(1-10) – FRI(1-10), change in frequency of rf source 1-10 201-250 – FAMPL(1-50), Fourier amplitudes[f] 251-300 – FAZE(1-50), Fourier phases[f].
XCRNR	135.0	0.001	Fraction of the full width of the plot frame between the left-hand edge of the frame and the bottom left-hand corner of the plot.
YCRNR	69.0	0.001	Fraction of the full height of the plot frame between the bottom edge of the frame and the bottom left-hand corner of the plot.
XAXISL	750.0	0.001	Plot width as a fraction of the full width of the frame.
YAXISL	750.0	0.001	Plot height as a fraction of the full height of the frame.

[a] See refs. [15] and [16].
[b] See section 2.2.3.
[c] A particle which ESME tracks from the origin, (0,ES).
[d] See the T-command.
[e] See section 2.2.13.
[f] See section 2.2.15.

L-command, NAMELIST /LLRF/			
Variable	Default Value	Unit	Description
PHFBON	.F.	-	If .TRUE., activates phase feedback.
VFBON	.F.	-	If .TRUE., activates voltage feedback.
NTUAVG	1[a]	-	Number of previous turns over which to average in the calculation of the phase "signal".
NTURES	1	turn	Delay in phase feedback response; the present "signal" is compared with that obtained NTURES turns previously.
IFTB	0	-	Form of the phase feedback: 0 – Critical damping 1 – Fixed.
USEWT	.F.	-	If .TRUE., the array W is used in averaging the individual "measurements" of mean phase over NTUAVG turns.
W(1:NTUAVG)	0.0	-	Weights which multiply the turn-by-turn mean phases if the phase "signal" is computed with USEWT = .TRUE.
FBFACT	1.0	-	Gain applied to the phase feedback.
DLIMIT	5.7296	deg	Upper limit of the magnitude of the phase feedback on a single turn.
VFBFCTR	1.0	-	Gain applied to the voltage feedback.
VLIMIT	0.1	-	Upper limit of the magnitude of the voltage feedback fraction on a single turn.
ETAJMP	0.0	-	Value of η at which to "flip" the phase of the rf.

[a]NTUAVG = 1 represents infinite-bandwidth phase feedback.

M-command, NAMELIST /MRANGE/

Variable	Default Value	Default Unit	Description
TMBEGIN	0.0	s	Time at which to start saving mountain range data.
TMEND	0.0	s	Time at which to stop saving mountain range data.
MRMPLOT	1	turn	Interval between mountain range data records; data are saved every MRMPLOT turns.
MRNBIN	50	-	Number of bins for mountain range histogram.
MRTHBMIN	0.0[a]	deg	Lower limit of ϑ for mountain range histogram.
MRTHBMAX	0.0[a]	deg	Upper limit of ϑ for mountain range histogram.

[a]MRTHBMIN and MRTHBMAX both zero results in the histogram range being $\pm 180°$/FRAC.

N-command, NAMELIST /MRPLOT/

Variable	Default Value	Default Unit	Description
IDEV	0	-	Virtual device number[a] for graphics output.
IMETA	-1	-	Directs graphics output: < 0 – Output is written to a metafile 0 – Output is directed to the device specified by IDEV > 0 – Output is written both to a metafile and to the device specified by IDEV.
MRTHPMIN	0.0[b]	deg	Lower limit of ϑ for mountain range plot.
MRTHPMAX	0.0[b]	deg	Upper limit of ϑ for mountain range plot.
NTRACE	100	-	Number of traces on each page of mountain range plot.
NSKIP	0	-	Number of data records to be "skipped" between each trace.
TOPTOB	0.7	-	Fraction of YAXISL in which NTRACE traces are plotted (approximate if TBASE = .TRUE.).
SCALE	0.3	-	Height of the first trace (as a fraction of YAXISL).
MSTART	0[c]	-	Turn number at which to start plotting mountain range data.
MSTOP	0[c]	-	Turn number at which to stop plotting mountain range data.
TMSTART	0.0[d]	s	Time at which to start plotting mountain range data.
TMSTOP	0.0[d]	s	Time at which to stop plotting mountain range data.
TBASE	.F.	-	If .TRUE., trace separation is proportional to time.
LIM	.F.	-	If .TRUE., dotted lines are drawn connecting the leftmost and rightmost non-zero points of consecutive traces.
SMOOTH	.F.	-	If .TRUE., a smoothing algorithm is applied to the data before plotting.
XCRNR	135.0	0.001	Fraction of the full width of the plot frame between the left-hand edge of the frame and the bottom left-hand corner of the plot.
YCRNR	69.0	0.001	Fraction of the full height of the plot frame between the bottom edge of the frame and the bottom left-hand corner of the plot.
XAXISL	750.0	0.001	Plot width as a fraction of the full width of the frame.
YAXISL	750.0	0.001	Plot height as a fraction of the full height of the frame.

[a]See refs. [15] and [16].

[b]MRTHPMIN and MRTHPMAX both zero results in the data being plotted over its entire range, MRTHBMIN $\leq \vartheta \leq$ MRTHBMAX (see the M-command).

[c]MSTART, MSTOP, TMSTART and TMSTOP all zero results in all data, from TMBEGIN to TMEND (see the M-command), being plotted.

[d]MSTART and MSTOP take precedence unless both are zero, then TMSTART and TMSTOP determine the data plotted.

O-command, NAMELIST /GRAPH/

Variable	Default Value	Default Unit	Description
MPLOT[a]	1000	turn	Graphics output interval; plots are produced every MPLOT turns (or each time a D-command is issued).
IDEV	0	-	Virtual device number[b] for graphics output.
IMETA	-1	-	Directs graphics output: < 0 – Output is written to a metafile 0 – Output is directed to the device specified by IDEV > 0 – Output is written both to a metafile and to the device specified by IDEV.
POSTP	.F.	-	If .TRUE., all data in COMMON blocks are written to logical unit 18; the plotting routine is not called.
PLTSW			Selects plot options (when array element is .TRUE.):
(1)	.T.	-	Draw phase space plot
(2)	.T.	-	Plot phase space points (distinct symbol for each class)
(3)	.F.	-	Interconnect points within each class
(4)	.F.	-	Draw lines at centroid and $\pm 1\sigma$
(5)	.F.	-	Draw voltage waveform
(6)	.F.	-	Set plot boundaries to turning points of contour
(7)	.F.	-	Suppress captions, axis labels, etc.
(8)	.T.	-	Plot ϑ histogram
(9)	.F.	-	Set ϑ histogram limits to turning points of contour
(10)	.T.	-	Plot energy histogram
(11)	.F.	-	Set energy histogram limits to turning points of contour
(12)	.F.	-	Plot Fourier amplitudes
(13)	.F.	-	Include phases in plot of Fourier spectrum
(14)	.F.	-	Plot space-charge energy loss (per turn) vs. ϑ
(15)	.F.	-	Include distribution histogram in space-charge plot
(16)	.F.	-	Plot high-Q resonator voltage.
THPMIN	0.0[c]	deg	Lower limit of ϑ for phase space plot.
THPMAX	0.0[c]	deg	Upper limit of ϑ for phase space plot.
DEPMIN	0.0[d]	MeV	Lower limit of energy for phase space plot.
DEPMAX	0.0[d]	MeV	Upper limit of energy for phase space plot.
TITLE[e]	.F.	-	If .TRUE., use the line following NAMELIST input as a plot title.
KLPLOT	0	-	Determines which class of particles is plotted and histogrammed. 0 – All classes are plotted > 0 – Only class KLPLOT is plotted.
NPJMP	1	-	Plot only every NPJMP-th point in phase space.
NBPFFT	0[f]	-	Number of bins in FFT plot.
IOPT	-1	-	Source for bucket (or matched contour) plotting: < 0 – None; no contour plotted 0 – All active rf sources > 0 – Source IOPT ($1 \leq$ IOPT $\leq$ NRF).
MTCHSI	.F.	-	If .TRUE., plot contour (matched to source IOPT) whose area is equal to the initial distribution emittance.
MTCH95	.F.	-	If .TRUE., plot contour (matched to source IOPT) containing 95% of the present distribution emittance.
IEREF	1	-	Determines energy origin for phase space plots: 1 – E0, energy on the reference orbit 2 – ES, synchronous energy 3 – EBAR, mean particle energy 4 – EREF, energy of the "reference particle"[g].

[a] MPLOT = 0 results in plots being produced only when a D-command is issued.

[b] See refs. [15] and [16].

[c] THPMIN and THPMAX both zero results in the plot range being $-180°/\text{FRAC} \leq \vartheta \leq 180°/\text{FRAC}$.

[d] DEPMIN and DEPMAX both zero results in approximately the entire range of particle energies being plotted.

[e] This is an exception to the usual maintenance of NAMELIST data; TITLE is set to .FALSE. after every O-command.

[f] NBPFFT = 0 results in the value of NBINFFT (see the F-command) being used.

[g] A particle which ESME tracks from the origin, (0,ES).

O-command, NAMELIST /GRAPH/ (continued)

Variable	Default Value	Default Unit	Description
NBINTH	50	-	Number of bins in ϑ for histogram.
THBMIN	0.0[a]	deg	Lower limit of ϑ for histogram.
THBMAX	0.0[a]	deg	Upper limit of ϑ for histogram.
NBINE	50	-	Number of bins in energy for histogram.
EBMIN	0.0[b]	MeV	Lower limit of energy for histogram.
EBMAX	0.0[b]	MeV	Upper limit of energy for histogram.
IFBMIN	1	-	Lower limit for FFT plot.
IFBMAX	0[c]	-	Upper limit for FFT plot.
SCBMIN	0.0[d]	-	Lower limit of ϑ for space-charge plot.
SCBMAX	0.0[d]	-	Upper limit of ϑ for space-charge plot.
RBMIN	0.0[e]	-	Lower limit of ϑ for resonator voltage plot.
RBMAX	0.0[e]	-	Upper limit of ϑ for resonator voltage plot.
XCRNR	135.0	0.001	Fraction of the full width of the plot frame between the left-hand edge of the frame and the bottom left-hand corner of the plot.
YCRNR	69.0	0.001	Fraction of the full height of the plot frame between the bottom edge of the frame and the bottom left-hand corner of the plot.
XAXISL	750.0	0.001	Plot width as a fraction of the full width of the frame.
YAXISL	750.0	0.001	Plot height as a fraction of the full height of the frame.
DTHCURV	0.0	deg	Displacement of contour in azimuth.
DECURV	0.0	MeV	Displacement of contour in energy.
DELCON	0.01	-	Determine rf bucket to within DELCON x 360° of rf waveform.
KNTLIM	500000	-	Maximum number of iterations of difference equations which will be attempted to close contour.

[a]THBMIN and THBMAX both zero results in the histogram range being the same as for the phase space plot.

[b]EBMIN and EBMAX both zero results in the histogram range being the same as for the phase space plot.

[c]IFBMAX = 0 results in the upper limit being the greatest Fourier harmonic computed.

[d]SCBMIN and SCBMAX both zero results in the plot range being ±180°/FRAC.

[e]RBMIN and RBMAX both zero results in the plot range being ±180°/FRAC.

P-command, NAMELIST /POPL8/

Variable	Default Value	Default Unit	Description
KIND	1	-	Type of distribution to be generated: 1 – Rectangular outline of NTH points between THMIN, THMAX in ϑ and NE points between REMIN, REMAX in energy 2 – Rectangular NTH by NE (regular) grid with limits as for KIND = 1 3 – Rectangular uniform (random) distribution of NPOINT particles with limits as for KIND = 1 4 – NPOINT particles uniform in ϑ between THMIN, THMAX, Gaussian in energy with $\pm 2\sigma$ limits REMIN, REMAX 5 – NPOINT particles Gaussian in ϑ with $\pm 2\sigma$ limits THMIN, THMAX, uniform in energy between REMIN, REMAX 6 – Rectangular NTH by NE grid, regular spacing in ϑ, Gaussian spacing in energy, and limits as for KIND = 1 7 – Bunch contour outline of NPOINT particles 8 – Rectangular NTH by NE grid inside contour 9 – Uniform distribution of NPOINT particles inside contour 10 – Bi-Gaussian distribution of NPOINT particles, 95% within contour 12 – NPOINT particles parabolic in ϑ between THMIN, THMAX, uniform in energy between REMIN, REMAX 13 – Bi-parabolic distribution of NPOINT particles inside contour 14 – Matched elliptical distribution of NPOINT particles 15 – NPOINT particles uniform in ϑ between THMIN, THMAX, parabolic in energy between REMIN, REMAX.
THMIN	-90.0	deg	Lower limit of ϑ for rectangular distributions.
THMAX	90.0	deg	Upper limit of ϑ for rectangular distributions.
REMIN	None	MeV	Lower limit of energy (relative to that of the synchronous particle) for rectangular distributions.
REMAX	None	MeV	Upper limit of energy (relative to that of the synchronous particle) for rectangular distributions.
NTH	2	-	Number of grid points in ϑ.
NE	2	-	Number of grid points in energy.
SBNCH	0.1	eVs	Area within matching contour.
IPOP	1	-	Source used to generate matching contour: 0 – All active rf sources > 0 – Source IPOP ($1 \leq$ IPOP $\leq$ NRF).
THOFF	0.0	deg	Displacement in azimuth of distribution generated in the current P-command.
EOFF	0.0	MeV	Displacement from synchronous energy of distribution generated in the current P-command.
THTRAN	0.0	deg	Displacement in azimuth of all particles generated in this and previous P-commands.
ETRAN	0.0	MeV	Displacement from synchronous energy of all particles generated in this and previous P-commands.
NPOINT	1	-	Number of particles in distribution (except for the grids, KIND = 1,2,6,8).
PARTION	.F.	-	If .TRUE., partitions distribution into separate classes[a]; each use of the P-command with PARTION = .TRUE. introduces a new partition.
ISEED	314159	-	Seed for internal random number generator.

[a]Different classes of particles are plotted with distinct symbols.

R-command, NAMELIST /RING/

Variable	Default Value	Default Unit	Description
REQ	None	m	Radius of the reference (central) orbit.
GAMMAT	None	-	Transition value of γ on the reference orbit.
ALPHA1	0.0	-	Coefficient of $(\Delta p/p)^2$ in series expansion for $\Delta R/R$ about reference orbit.
ALPHA2	0.0	-	Coefficient of $(\Delta p/p)^3$ in series expansion for $\Delta R/R$ about reference orbit.
ALPHA3	0.0	-	Coefficient of $(\Delta p/p)^4$ in series expansion for $\Delta R/R$ about reference orbit.
EK0I	None	MeV	Kinetic energy on the central orbit at time TI.
EK0F	0.0	MeV	Kinetic energy on the central orbit at time TF.
TI	0.0	s	Time at which magnetic field change begins.
TF	0.0	s	Time at which magnetic field change ends.
TSTART	0.0	s	Time at which tracking begins.
FRAC	1.0	-	Determines azimuthal periodicity by restricting calculation to $-180°/\text{FRAC} \leq \vartheta \leq 180°/\text{FRAC}$.
PIPRAD	1.0	m	Radius of beampipe.
EBDRY	.F.	-	If .TRUE., puts absorbing beampipe "walls" at REQ ± PIPRAD.
DES	0.0	MeV	Energy offset of synchronous orbit relative to reference orbit.
KURVEB	1	-	Magnetic field ramp from EK0I to EK0F: 1 – Linear 2 – Increasing parabolic 3 – Biased sinusoidal 4 – Decreasing parabolic 5 – Parabolic with gradient from EKIDOT to EKFDOT.
EKIDOT	0.0	MeV/s	Slope of parabolic ramp at time TI.
EKFDOT	0.0	MeV/s	Slope of parabolic ramp at time TF.
JNRAMP	.F.	-	If .TRUE., sets starting point of ramp to be the current value, hence smoothly piecing ramp segments together.
GMAJMP	.F.	-	If .TRUE., activates γ_{tr}-jump.
KINDG	1	-	Type of γ_{tr} variation[a]: 1 – Linear; γ_{tr} = GAMPAR(1) + GAMPAR(2) * T 2 – Decreasing exponential; γ_{tr} = GAMPAR(1) + GAMPAR(3) * $(1 - e^{-T/\text{GAMPAR(2)}})$.
GAMPAR(1:3)	0.0	-	Coefficients for γ_{tr} variation.

[a] T = 0 corresponds to the time at which the R-command is invoked with GMAJMP = .TRUE.

T-command, NAMELIST /CYCLE/

Variable	Default Value	Default Unit	Description
TSTOP[a]	0.0	s	Time at which to stop tracking.
TTRACK[b]	0.0	s	Duration of tracking.
MSTEP	100	-	(Min.) number of tracking steps per synchrotron period.
ACCEL0[c]	1.0	-	(Max.) number of beam turns per tracking step.
NCAV	1	-	Number of rf "cavities" in the ring, i.e., the number of iterations of the difference equations per turn.
LGRTHM	1	-	Selects difference equations used in tracking: 1 – Complete kinematics; expand α to maximum order using the coefficients ALPHAn[d] 2 – Use the simplified form, $\vartheta_{i,n} = \frac{r_{s,n-1}}{r_{s,n}}\vartheta_{i,n-1} + 2\pi\eta\frac{\Delta p_{i,n}}{p_{s,n}}$.
ITRAP(1:4)	0	-	Condition which causes tracking to be interrupted before the time indicated by TTRACK or TSTOP: 0 – No trap 1 – Trap on minimum bunch width 2 – Trap on minimum bunch height 3 – Trap on η = ETATRP (tolerance = $\pm 0.01\eta$) 4 – Trap on $\lvert\sin\phi_s\rvert$ = PHISTRP (tolerance = ± 0.005) 5 – Trap on $\eta > 0$, i.e., transition crossing 10-19 – Call subroutine SHAZAM at entry point SHAZAM, SHAZAM1,...,SHAZAM9 after every iteration of the difference equations. Tracking is not stopped.
ETATRP	.001	-	See ITRAP = 3.
PHISTRP	.95	-	See ITRAP = 4.
MGRACE	0	-	Allow a "grace period" of MGRACE turns before checking trapping conditions.
HISTRY	.F.	-	If .TRUE., write a history record to logical unit 9 after every iteration of the difference equations.
MOMNTS	.F.	-	If .TRUE., compute the first and second moments of the distribution after every iteration of the difference equations.
BBDRY	.F.	-	If .TRUE., remove particles outside the region $-180°/\text{FRAC} \leq \vartheta \leq 180°/\text{FRAC}$.

[a]This is an exception to the usual maintenance of NAMELIST data; TSTOP is set to zero when tracking is completed or is interrupted by an ITRAP option.

[b]TSTOP takes precedence unless it is zero, then TTRACK determines the tracking duration.

[c]Tracking proceeds according to the integer, ACCEL (≥ 1), nearest ACCEL0 which satisfies the MSTEP constraint.

[d]See the R-command.

Bibliography

[1] *Design Report Tevatron I Project*, Fermi National Accelerator Laboratory (October 1984), Chapters 4–6

[2] J. E. Griffin, J. A. MacLachlan, A. G. Ruggiero, K. Takayama, "Time and Momentum Exchange for the Production and Collection of Intense Antiproton Beams at Fermilab", IEEE Trans. Nucl. Sci. **30**#4(1983) 2630–2632

[3] J. E. Griffin, J. A. MacLachlan, Z. B. Qian, "RF Exercises Associated with Acceleration of Intense Antiproton Bunches at Fermilab", IEEE Trans. Nucl. Sci. **30**#4(1983) 2627–2629

[4] J. A. MacLachlan, "ESME: Longitudinal Phasespace Particle Tracking — Program Documentation", Fermilab TM-1274(May 1984), unpublished (largely obsolete)

[5] J. A. MacLachlan, "Longitudinal Phasespace Tracking with Spacecharge and Wall Coupling Impedance", Fermilab FN-446 (February 1987), unpublished.

[6] P. Lucas and J. MacLachlan, "Simulation of Spacecharge Effects and Transition Crossing in the Fermilab Booster", Proc. 1987 IEEE Particle Accelerator Conference held at Washington, D.C. (16–19 March 1987) p1114.

[7] S. Stahl and C. Ankenbrandt, "Simulation of the Capture Process in the Fermilab Booster", Proc. 1987 IEEE Particle Accelerator Conference held at Washington, D.C. (16–19 March 1987)p1117

[8] P. Lucas and Q. Kerns, "Simulation of a Programmed Frequency Shift Near Extraction from the Fermilab Booster", Proc. IEEE Particle Accelerator Conference held at Washington, D.C. (16–19 March 1987)p1108

[9] S. Stahl and S. A. Bogacz, "Coupled Bunch Instability in a Circular Accelerator and Possible Cures: Longitudinal Phasespace Simulation", Phys. Rev. **D37**#5(1 March 1988)1300–1306

[10] J. A. MacLachlan, "Fundamentals of Particle Tracking for the Longitudinal Projection of Beam Phasespace in Synchrotrons", Fermilab FN-481(15 April 88), unpublished.

[11] J. A. MacLachlan, "Difference Equations for Longitudinal Motion in a Synchrotron", Fermilab FN-529(15 December 1989), unpublished.

[12] S. Stahl and S. A. Bogacz, "Simulation of Coupled Bunch Mode Growth Driven by a High-Q Resonator: A Transient Response Approach", Proc. 1989 IEEE Particle Accelerator Conference held at Chicago, Il. (20–23 March 1989)p1175.

[13] M. W. Eaton, "Expand and Xfort, Macro Expanders for CDF", Fermilab CDF-194(January 1986), unpublished. Longitudinal Projection of Beam Phasespace in Synchrotrons", Fermilab FN-481(15 April 88), unpublished.

[14] J. A. MacLachlan, "An ESME Update (v. 7.2)", Fermilab TM-1716(March 91), unpublished.

[15] R. Bock et. al., "HIGZ–High level Interface to Graphics and Zebra", CERN program library Q120(March 88).

[16] Graphics Section, "Guide to computer graphics at CERN", CERN/DD/US/111 (August 87).