User’s Guide to ESME esmF95 (esme2007.1)

James A. MacLachlan          Jean-François Ostiguy

December 2007
Abstract

ESME is a computer program to calculate the evolution of a distribution of particles in energy and phase as it is acted upon by the radio frequency system of a synchrotron or storage ring. It provides for the modeling of multiple rf systems, feedback control, space charge, and many of the effects of longitudinal coupling impedance. Optional command line arguments control the volume of output, interactive mode, etc. Memory allocation can be controlled at run time for large arrays; unused arrays are not allocated. The capabilities of the program are described, and the requirements for input data are specified in sufficient detail to permit significant calculations by an uninitiated user.

ESME has been extensively modified since the prior user documentation. This note describes esmF95 (esme2007.1) in particular. A few of the differences from earlier versions have resulted from the translation of the code to Fortran 95. A rationalization of rf phase determination adds little new capability but greatly simplifies use of rf curves, feedback, and synchronization. Curve representation has been changed for greater flexibility in using the internally generated curves and to obtain closer equivalence between internal curves and tabular curve files. Optional scaling of the time per iteration has been introduced either to reduce execution time or to improve the precision of the dynamics in some circumstances. An option for calculating an appropriate time step each iteration introduced in esmF95 (esme2007.1) makes the scaling feature more convenient than the fixed choice in previous versions.

This guide undergoes continual revision and at any time may be a little behind the latest developmental code or contain residual errors. Criticism or corrections from the reader are always welcome.
# Contents

1 Introduction 2
   1.1 Coordinate System .......................................................... 2
   1.2 Difference Equations ....................................................... 4
   1.3 Multiparticle Calculations .............................................. 5
   1.4 Emittance(s) ................................................................. 5
   1.5 Flow Lines in Phase Space .............................................. 6

2 Program Organization and Data Requirements 8
   2.1 Command Ordering .......................................................... 8
   2.2 Commands ................................................................. 9
      2.2.1 R Command - Lattice parameters .................................. 11
      2.2.2 A Command - RF parameters ...................................... 14
      2.2.3 P Command - Initial distribution parameters .................. 19
      2.2.4 O Command - Graphical output options .......................... 22
      2.2.5 T Command - Track distribution .................................. 25
      2.2.6 Q Command - Quit .................................................... 27
      2.2.7 D Command - Display .............................................. 27
      2.2.8 W Command - Write comment ...................................... 27
      2.2.9 H Command - History output ...................................... 28
      2.2.10 M & N Commands - Save mountain range data & Plot mountain range .............................................. 31
      2.2.11 L Command - Low level feedback parameters .................. 34
      2.2.12 B Command - Beam-derived potential (self-force and Z_{||}) .... 36
      2.2.13 F Command - Fourier transform .................................. 39
      2.2.14 C Command - Flow contours ...................................... 40
      2.2.15 K Command - Kill all or parts of the distribution ............ 41
      2.2.16 0 – 9 Commands - User-written SHAZAM routines ............. 41
      2.2.17 Y Command - Memory allocation for .............................. 42
      2.2.18 S Command - Save tracking parameters .......................... 42
      2.2.19 G Command - Get tracking parameters .......................... 42

3 Using The Program 44
   3.1 Running the Program ..................................................... 44
   3.2 Input Structure ........................................................... 45
      3.2.1 Command file ......................................................... 45
      3.2.2 RF curves ............................................................. 46
      3.2.3 Magnet ramp table .................................................. 47
      3.2.4 Impedance table ..................................................... 48
      3.2.5 Resonance table ..................................................... 48
      3.2.6 Time domain basis table .......................................... 49
      3.2.7 Bunch width table ................................................... 49
Chapter 1

Introduction

The program ESME has been developed to model those aspects of beam behavior in a synchrotron that are governed by the radio frequency systems. It follows the evolution of a distribution in energy-phase coordinates by iterating a map corresponding to the single-particle equations of motion. A provision for scaling the map for faster calculations is available: only when the scaling factor is one does an iteration correspond to a beam turn. The scaling has uses other than accelerating the calculation. For example, besides scaling for longer time step one can scale for shorter to approximate distributed rf or to avoid differences between extreme bunches and the central bunches in a long batch. Note that the scaling parameter can take on any positive value; thus an integer or the reciprocal of an integer are intuitively multiple turns per time step and multiple time steps per turn respectively, but an arbitrary value is valid. The map parameters may be updated each iteration to reflect the action of the beam current on the individual particles through feedback loops, space charge, coupling impedance, etc. The code was initially developed during the years 1981–82 for the design of the Tevatron I Antiproton Source and documented for general use in 1984. This manual documents a substantial revision of the program. The previous supported version was esmey2k v. 2003. Data prepared for this and earlier versions are likely to need some modification to run under ESME esmF95 (esme2007.1). The conceptual basis of ESME is discussed at length elsewhere.

ESME is used frequently to assess the efficiency of a given rf beam manipulation or to optimize system parameters. For this the user needs to specify technical details of the various subsystems and derive various numerical measures of system performance from the particle distribution. In this sort of calculation many data are required, and the program may employ several numerical analysis features. Equally useful, however, are qualitative calculations designed to illustrate a concept or explore the feasibility of a novel approach. For such use the code requires a minimum of data and provides easy access to a variety of graphical output. When a preliminary investigation has been fruitful it is natural to proceed to a thorough modeling. ESME is intended to serve effectively over a wide range of detail by separating functions so that only the relevant ones need be considered and by establishing reasonable defaults to reduce the data required for typical cases. Thus, a few lines of data may serve to get a first look at a system which can be studied in greater detail by overriding defaults with specific input and by using additional functions like, for example, those related to collective behavior or those related to numerical evaluation of the properties of the distribution.

1.1 Coordinate System

The macroparticle coordinates in ESME are energy difference [MeV] from the synchronous energy and phase difference from the synchronous phase. Because the program accommodates simultaneous operation of rf systems at different harmonic number $h$, the phase variable is taken at $h = 1$. This means, for example, that the unstable fixed points for a stationary bucket occur at $\pm \pi/h$. Internally the phases are in radians, but input data are in degrees. The single-particle longitudinal dynamics depend on the mean radius of the ring $R_{\text{eq}} = C/2\pi$, the momentum $p_0$ (energy $E_0$) of a particle which follows the central orbit, and the momentum dependence of path length for particles with momenta near $p_0$. When rf is present, the energy $E_s$ of a particle that arrives at location of the rf when the

---

1Instructive examples of the capabilities of ESME can be found in reports relating to TeV I. Some of these will not be cited directly in the following text but are included in the references. See for example refs. [2], [3], and [4].
phase is \( \varphi_s = h \vartheta_s \) such that \( \Delta E_s = \Delta E_0 \) is called a synchronous particle, and \( \varphi_s \) is the synchronous phase. \( E_s \) is usually equal to \( E_0 \) so that the synchronous orbit and the central orbit are the same. Because this is typical, data for an rf system requires a harmonic number but not a frequency, because the rf frequency defaults to \( h \) times the the circulation frequency of a particle with momentum \( p_0 \). The program can calculate a \( \vartheta \) value such that the particle having that phase will remain on the central orbit. When there is a single rf system, \( \vartheta_s \) is the familiar synchronous phase except for the \( h \) factor: \( \vartheta_s = \varphi_s / h \). The synchronous phase is chosen as the origin for \( \vartheta \); thus, \( \vartheta_s \) is the amount the waveform must be shifted so that the particle at \( \vartheta = 0 \) gets the synchronous energy increment. When there are several systems active, each may have its independent phase and harmonic number. However, if there is sufficient total amplitude, there is a phase by which the composite waveform can be shifted to give the synchronous energy increment at a phase stable point. This shift may not be the phase of any of the individual rf systems at the time the synchronous particle passes. This approach to setting the phases of multiple rf systems is usually convenient, but there are times when it interferes with some atypical process. Synchronization which is determined by and applied to a single system can also be specified. Incidentally, the code generally proceeds as though all of the rf gaps are at one location. Usually the synchrotron tune is low enough, say \( \nu_s < 0.005 \), that this will be a good approximation. However, if the cavities are in fact distributed and the synchrotron tune high, one can get more realistic modeling by taking multiple time steps per turn.

Because the quantities \( \vartheta_i \) have the range \( \pm \pi \), it is natural to identify them with the azimuth of the macroparticles at the synchronous time, i.e., the time when the synchronous particle is at the rf location. However, because \( \vartheta \) is actually rf phase, the conversion to azimuth depends on particle momentum. For particles close to the synchronous energy, the difference is qualitatively negligible, and useful intuition can be derived from picturing the distribution in \( \vartheta \) as being distributed around the ring in the same way. Note, however, that in this picture smaller \( \vartheta \) particles are leading particles; beam circulates in the ring in the \( -\vartheta \) direction. Although it is not strictly correct, \( \vartheta \) will occasionally be referred to as the particle azimuth in the following. The conversion from the phase variable to the true azimuth \( \Theta_{i,n} \) of the particle is

\[
\Theta_{i,n} = \frac{\tau_{s,n}}{\tau_{i,n}} \vartheta_{i,n} \approx \left( 1 - \eta \frac{\Delta p}{p} \right) \vartheta_{i,n}.
\]

For most applications to a high energy synchrotron the \( \eta \)-term is small compared to one. There are cases for which the distinction between \( \vartheta \) and \( \Theta \) are important.

An important distinction between the \( h = 1 \) rf phase and the azimuth of a particle is that azimuth is modular with a range of \( 2\pi \) whereas the rf phase is unbounded. In ESME, rf phase is taken mod \( 2\pi \), but a particle which gains or loses a full turn on the synchronous particle gets the appropriate number of energy increments. As explained in the section on the \( \mathbf{R} \) command, it can be be convenient or computationally efficient to invoke a periodicity of the harmonic number \( h \) or some sub harmonic.

The treatment starts from the specification of a reference orbit of average radius \( R_{eq} \) on which the mean normal magnetic field \( <B_y> \) is known. A particle following the reference orbit has the reference momentum

\[
p_0 = 2.997925 \cdot 10^2 <B_y> R_{eq}
\]

with \( p_0 \) in MeV/c, \( R_{eq} \) in m, and \( <B_y> \) in Tesla. The angular frequency of beam circulation on this orbit is \( \Omega_z = \beta_0 c / R_{eq} \) where \( \beta_0 \) is the Lorentz \( \beta \). The variation of the guide field away from the reference orbit is completely characterized for the purposes of describing longitudinal motion by the momentum dependence of the path length for fixed guide field. At a minimum one need only specify \( \gamma_T = \text{const.} \), a choice adequate in many instances. In many problems it is also correct to identify the radius of the synchronous trajectory \( R_s \) with the reference radius \( R_{eq} \). However, in applications like stacking or displacement acceleration where the synchronous trajectory may be offset radially, or even out of the beampipe entirely, 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 momentum and radius. The momentum dependence of path length at the synchronous radius is determined from that given for the reference orbit.

It was mentioned before that the the synchronous energy \( E_s \) need not be equal to the reference energy \( E_0 \) on the central orbit. This is a minor generalization of what has been described. The default mapping period is still the
synchronous period. The path length dependence on momentum must be transformed from the reference momentum as an origin to the synchronous momentum. The synchronous energy can be separated from the reference energy by requesting an energy offset (see 2.2.1). The rf data can include momentum offset programs, frequency programs, and phase programs which will move $E_n$ away from $E_0$ (see 2.2.2).

The initial particle distributions can have any central energy. Naturally if a distribution is not centered on $E_n$, it will appear in different parts of the phase plane from turn to turn. Because the concept of a synchronous particle is fundamental to the program design, such a particle is usually tracked as a check on the program and the input data. A difference between the coordinates of this particle (EREF, THREF) and the presumed synchronous point (ES, 0) is an indicator of non-adiabatic change of parameters. Such difference could of course be intentional, but usually it is a sign that there is an error in the input data. It is also possible to initialize (EREF, THREF) at an arbitrary value for monitoring other aspects of the calculation.

1.2 Difference Equations

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

$$\vartheta_{i,n} = \left[ \frac{\tau_{s,n-1} - 1}{\tau_{s,n}} \vartheta_{i,n-1} + 2\pi \left( \frac{\tau_{i,n}}{\tau_{s,n}} - 1 \right) \right] \mod(\pi)$$

$$E_{i,n} = E_{i,n-1} + eV(\varphi_{s,n} + h\vartheta_{i,n}) - eV(\varphi_{s,n})$$

giving the change in the phase and energy of particle $i$ during the $n$-th turn of the synchronous particle. The $n$-th energy increment comes at the end of the $n$-th turn. The relation between the synchronous beam circulation period $\tau_{s,n}$ and that of the $i$-th particle $\tau_{i,n}$ is treated exactly. Thus, the kinematic nonlinearity is treated exactly; this feature can be very important if the synchronous energy is close to the transition energy. The lattice nonlinearity is expressed by expanding the trajectory length in powers of $\Delta p_i / p_0 = (p_i - p_0) / p_0$ where $p_i$ is the particle momentum and $p_0$ is the momentum on the reference trajectory. The rf potential is the sum of one or more sinusoidal terms so that the dynamic nonlinearity of a simple waveform is treated exactly and other waveforms are treated in a Fourier expansion of ten or fewer terms.\(^2\) The intercavity drift can be subdivided for more frequent application of the beam-induced force when it is necessary to more closely approximate its continuous presence.

In the section on parameters for the tracking routine there is mention of a parameter for choosing alternative versions of the difference equations. There are two choices, one which calculates $\Delta \vartheta$ as indicated above and one that equates it to the familiar approximation $-2\pi \eta \Delta p / p$, where $\eta$ is the same for all macroparticles. Substantial time can be saved by using the simple difference equation if it is sufficient.

All calculation in ESME is performed in double precision. When a very large number of iterations is taken, it is possible in some circumstances to notice effects of finite numerical precision. The greatest effect observed so far has been a spurious energy kick for a stationary bucket above transition arising from the truncation error in the synchronous phase of $\pi$. For fewer than $10^6$ iterations it is very unlikely that any significant effect will be observed. A good way to monitor a calculation for significant effects is to make a HISTORY plot of average energy for the distribution (EBAR) vs. average phase (THBAR) and look for the development of a systematic (linear) growth of the center of charge oscillation. Further discussion of numerical precision, including a cure for the spurious energy kick above transition, can be found in ref. [7]. In some versions of the MAKEFILE one can find a compile option MODDEQ. This option was used in the study leading to the cited reference. There should be no need to use this option in normal circumstances; if one thinks it might be relevant to some special case, read [7] and perhaps some basic material in numerical methods before plunging in.

---

\(^2\)A little-used option mentioned in ref. [5] to provide a perfectly linear sawtooth waveform has been dropped because of severe conflict with the mechanism by which the more recent versions find the synchronous phase.
1.3 Multiparticle Calculations

The particle distribution at each iteration is mapped from the previous by the difference equations. The program provides three general types of optional calculation on the properties of the distribution as a whole. The most common calculations are those which quantify properties of the distribution so that one can plot them as functions of time. Examples of such properties are first and second moments, emittance, Fourier spectrum of beam current, etc. Another type of calculation involves calculating feedback contributions to rf system parameters. ESME provides for phase feedback and feedback to rf amplitude. The third general class of collective calculation is the evaluation of beam induced voltages from space charge and longitudinal coupling impedance. The space charge calculation uses a geometric factor depending on average beam radius and average beampipe radius. The beam radius is continually scaled from its input value according to the current central momentum to account for so called adiabatic damping. The geometric factor is rolled off at very high frequencies.[8] The longitudinal impedance can be characterized by an arbitrary table of real and imaginary part vs. frequency and/or a table of resonances defined by resonant frequency, Q, and real part at resonance. The frequency domain treatment of bunch self force and wall impedance is described in ref. [9]. Several interesting applications were made to the Fermilab Booster, which are described in refs. [10, 11, 12, 13].

When a resonance is represented by an impedance, the transient part of the excitation is neglected. However, this treatment does not realistically represent the driving terms for certain types of collective bunch motion. Therefore, response of a high-Q resonator may also be modeled in the time domain (see ref. [14]). Several resonances can be treated simultaneously with some (or all) treated in time domain and some (or all) treated in frequency domain. Reference [15] describes the use of this facility in modeling the coupled bunch instability. The code also provides a more general time domain calculation of the effect of wake fields. The calculation is based on a basis function giving the voltage response [MV] to a unit charge [C] distributed in a triangular pulse.[16] The basis function is calculated at an interval small enough to give good linear interpolation on the charge distribution and carried out to times for which the total wake field will be significant but in no case for more than one turn. This option may be used along with both the frequency domain calculation and the time domain calculation for simple resonances. The bunch self-force, i.e. the perfectly conducting wall term, can be calculated using either the FFT machinery used in the impedance calculation or in the time domain with a locally determined numerical derivative of the azimuthal charge distribution. Note that an irregular fill pattern for the ring does not necessarily, or indeed generally, mean that the transient beam excitation is important for the dynamics. After the filling time of any high-Q impedance, the current spectrum remains little changed from turn to turn; therefore, the frequency domain treatment will be appropriate. The effect of the irregular fill appears, of course, in the presence of circulation frequency harmonics in the beam current which are not harmonics of the rf frequency. The synchrotron oscillation side bands will be present because of the phase modulation of the beam current distribution by the synchrotron motion.

Calculations of collective effects often require many particles to avoid spurious instabilities driven by numerical noise. If the aim is to model only low frequency effects like bucket distortion or low frequency parasitic modes, it may be acceptable to smooth the charge distribution using either local smoothing or the global (Bernstein polynomial) options provided. However, an arguably more general and rigorous approach, which does not arbitrarily sacrifice high frequency information, is to scale the map[1] with the effects that both fewer iterations and fewer macroparticles are required. The code is supposed to scale all impedances automatically, but a soupçon of scepticism is salutary.

1.4 Emittance(s)

The same macroparticle model can yield several different numbers for emittance. Doubtless this variety has potential to confuse, but it results from useful alternative definitions of emittance and differing uses for the calculated measures. The first distinction is between true emittance, that is a phase space area, and so called rms or statistical emittance. The only way to get a true emittance from ESME is to use the ICONTUR = 4 choice in the O command. This will result in the generation of a sequence of matched contours which will usually terminate successfully with one containing 95 % of the macroparticles. This procedure is time consuming and does not always arrive at an
answer. The rms emittance, however, is uniquely defined by the distribution; there will always be a value available, and the calculation is relatively simple. However, an rms emittance is not a phase space area; it may decrease as well as increase during a process. To the degree that the difference equations are linear over the bunch width, the rms emittance is guaranteed to be invariant because a quadratic form mapped by a linear transformation retains its value. However, not only is the rf potential notably nonlinear near the peak voltage, but the rest of the map can be significantly nonlinear for some parameters. The rms emittance can be related to an area through the assumption of a phase space distribution function. The usual assumption is a Gaussian, but, for heavy particles in general and for many manipulations of the beam, this assumption is approximate at best and frequently useless. Nonetheless, the rms emittance is quick to calculate and often representative, so it is a very valuable number to record on each turn.

The emittance quantities that ESME produces routinely are the rms emittance for the particles within the \( \theta, E \) limits on the phase space scatter plot (labeled \( S_b \) on the plot) and the the rms emittance for the entire distribution (labeled “EPSILON” in the log file). The expression for the rms emittance is

\[
\varepsilon = \sqrt{\frac{1}{2N} \sum (\theta_i - \bar{\theta})^2 \sum (E_i - \bar{E})^2 - \left( \sum (\theta_i - \bar{\theta})(E_i - \bar{E}) \right)^2 \frac{\tau}{2N}} \quad \text{[eVs]}
\]

The divisor is \( 2N \) rather than \( 2N \pi \) so that \( \varepsilon \) is the area of the ellipse whose semi-axes are \( \sigma_E \cdot \sigma_t \) in the uncorrelated case. Thus, for a Gaussian bunch, the area containing 95% of the beam is just six times \( \varepsilon \). When the beam is not cleanly bunched, the correlation term in the emittance may make the quantity within the square root negative. In such case the term is ignored.

If the P command parameter RENORM has been set to .TRUE., the rms emittances are multiplied by a factor which makes the initial scaled rms bunch emittance equal to the initial bunch area, the P command parameter SBNCH. The calculated normalization is retained until changed by another P. Although it is not a feature of the standard code, recording the separate rms emittances of several bunches simultaneously requires only a very simple SHAZAM routine (see 2.2.16) to exploit existing code.

### 1.5 Flow Lines in Phase Space

Although the emphasis during the development of ESME has been on the evolution of distributions, the plotting of flow lines is a complementary approach to visualizing phase space motion which can either replace or enhance scatter plots of the distribution. There are several variants on the contour plotting approach.

1. At selected intervals through the acceleration cycle one can produce a flow line map. If a distribution is tracked, it may be plotted along with the flow lines or its plotting can be suppressed. If no distribution is tracked between the times at which the flow line displays are generated, the flow lines can not reflect the action of a collective potential.

2. An initial set of flow lines can be generated and then tracked as a distribution itself, either with or without an accompanying finite emittance distribution. If a collective potential is calculated with such a composite distribution, the flow line component is ignored, but the evolution of the flow line component will include the effect of the collective potential.

3. A fine grain visualization of the result of non-adiabatic motion can be obtained by tracking a flow line type of distribution and plotting it along with flow lines representing the final system parameter values. The regions of phase space where non-adiabatic motion has been significant will show up as areas where the tracked and the freshly generated contours differ.

4. A specialized variant of the tracking of a contour-like distribution is available as an option to the command which sets up the initial phase space distribution. A distribution of particles just inside and just outside the bucket is available. It can be helpful, for example, in understanding the effects of small perturbations on particle loss.

5. Flow lines can be generated from several arbitrary starting points.
6. Flow lines may be either at fixed parameters, representing the Hamiltonian flow at a particular time or with parameters that vary during the time the particle generates the trajectory to illustrate, for example, the effects of a parametric resonance.

The difference equations are used to generate lines of flow in phase space. If the flowlines are to reflect the influence of the collective potential derived from the charge distribution, there is an intrinsic contradiction when the longitudinal coupling impedance $Z_{||}$ has a real part because the trajectory generating particle is continually decelerated by the voltage arising from the real part. Such a spiraling flow line does not represent what is happening to particles in the distribution because, in the usual case, the synchronous phase is adjusted slightly to keep the bunch on the reference orbit. Therefore, when a single particle trajectory is calculated for a fixed time, the collective potential is reduced by its average value so that bounded trajectories will close. In the absence of a real component of impedance, the flow lines at fixed time are exactly lines of Hamiltonian flow.
Chapter 2

Program Organization and Data Requirements

Data for ESME are generally acquired by NAMELIST reads dispersed among subroutines which segregate different program functions as much as practical. The subroutine reading a particular class of data will be called only if there is in the input stream a single-character command requesting the related program function. The data relating to different program functions or accelerator subsystems are also stored in different fortran modules. These modules are initialized with values or switches to allow the code to proceed on the basis of a few input data. The initial values, or those read in over them, generally are retained unaltered\(^1\) so that a command may be repeated without reentering data that remain *apropos*. This feature is a great convenience when a command must be repeated many times with few changes in parameters; only the changes need be given explicitly. There are default values provided for many parameters to increase the likelihood of completing execution even if the input data fail to fully define the problem. Blank lines in the data do not cause problems.

The following subsection calls attention to requirements on command ordering arising from data dependencies. Next is a subsection listing all of the ESME command characters with a short description of their functions. They are listed in the order that would be likely to appear in an input data set, although some are not affected by order of appearance. The subsection following the listing of the command codes contains NAMELIST’s associated with the commands and a brief description of the function of each input datum. The same tabular information is collected in more compact form in Appendix B where the commands are arranged in alphabetical order for convenient reference.

2.1 Command Ordering

The order in which the commands may appear in the data will generally be logically apparent. For example the \texttt{R} command, which initiates the input of the basic lattice parameters and energy scale, usually will appear before any other command. Several commands require this information to perform their own functions. The \texttt{A} command, which brings in the rf parameters, is generally the second command to appear. The command \texttt{P}, which establishes the initial phase space distribution, generally needs to be preceded by both. If memory allocation differing from the default is used, the \texttt{Y} command must be called before the initial distribution is stored. The command \texttt{B}, which sets up the machinery for longitudinal impedance calculations, needs the initial distribution to get the initial beam current distribution. Although by detailed knowledge of the internal organization of the program or by exploiting some specialized facilities it is possible to create correct data sets that do not observe a typical command ordering, it is generally safer to adhere to the order \texttt{Y,R,A,P,B} of first appearance for these commands. The \texttt{C} command, which generates phase space flow lines, must follow \texttt{R} and \texttt{A}; if the flow lines are to take account of the collective potential produced by a bunch, it must follow \texttt{P} and \texttt{B} also. Other commands can usually be ordered arbitrarily.

---

\(^1\)There are exceptions where a parameter is reset after it has taken effect to avoid undesirable repetitions or altered to maintain desired continuity.
2.2 Commands

The principal functions of the program are activated by single-character commands appearing in the first position of a data record. Starting in the fifth position is an optional character field which serves to annotate the dataset itself and appears in the program output as a useful indication of the motivation for each step. The commands initiate a problem data read and/or an action upon available data. If data are requested by a subroutine invoked by the command, data in NAMELIST form will follow in the next record(s). Character strings like the plot title and names of auxiliary files are read as string variables. Exceptions are the file names for the save/restore commands (S and G), which are read with an A format, and possibly data requested by user-written SHAZAM routines (0 – 9 commands).

The data requested or process initiated by a command is related to a separable subsystem of the accelerator or a distinct phase of the calculation. The course of program execution is governed by the order of the commands in the data stream; that is, the program is what is sometimes called “data-driven”. Since the data are stored in separate modules, a subroutine that USEs an appropriate module has access to the related subset of the data.

Commands are listed below in four groups. The first group consists of the commands that will appear in almost every problem data set. They are listed in the order in which they are generally used. Interesting calculations may be made using these commands only. The second group contains commands for additional output options. Those in the third group invoke special calculations, including, if desired, ones written by the user. The fourth group consists of a save/restore pair for the entire state of the calculation and a dynamic memory allocation command. Commands with strong order dependence are underlined. In the description of a command the first letter of a keyword is indicated in bold face to call attention to an association between the command character and its function. Because of evolution of the code over more than twenty years, some of the mnemonic connections are a bit weak, but because it is useful to resurrect old files occasionally, continuity is a more important consideration than tidy mnemonics.

List of Commands

R read in the lattice (Ring) parameters, magnetic field ramp, problem energies, etc. according to NAMELIST /RING/.

A read in the radio frequency (Acceleration) parameters according to NAMELIST /RF/.

P Populate the phase space with the initial distribution described by parameters read according to NAMELIST /POPUL8/.

O Select graphical Output options using parameters in NAMELIST /GRAPH/.

T Track distribution using parameters in NAMELIST /CYCLE/.

Q Quit the program.

.................................................................

D Display graphical output.

W Write comment into printed output.

H Select quantities to be plotted from History records according to NAMELIST /HISTORY/.

M Save azimuthal histograms for composition of a Mountain-range plot.

N Plot mountain-range data.

.................................................................

L Set Low-level parameters controlling feedback, transition phase change, etc.) according to values in NAMELIST /LLRF/.
**B** Setup collective potential calculation; read in **Beam** parameters in NAMELIST /SCHG/.

**F** Setup the fast **Fourier** transform of the beam current from NAMELIST /FFT/.

**C** Form flow line **Contours** at time intervals or as an initial distribution using control parameters in NAMELIST /FLOW/.

**K** Kill all or selected parts of the phase space distribution according to control parameters in the NAMELIST /KUTS/.

0-9 Enter subroutines SHAZAM, SHAZAM1, SHAZAM2, . . . to manipulate any quantity stored globally.

.......................................................... ..................

**Y** Allocate memory to certain arrays according to input data in NAMELIST /MEMORY/.

**S** Save tracking and control data.

**G** Get tracking and control data from a previous run.
### 2.2.1 R Command - Lattice parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>REQ</strong></td>
<td>None</td>
<td>m</td>
<td>The reference radius for the central orbit</td>
</tr>
<tr>
<td><strong>GAMTSQ</strong></td>
<td>None</td>
<td></td>
<td>Square of transition $\gamma$ (&lt;0 accepted)</td>
</tr>
<tr>
<td><strong>ALPHA1</strong></td>
<td>0.0</td>
<td></td>
<td>Coefficient of $(\Delta p/p)^2$ in series expansion for length difference between particle trajectory and reference orbit</td>
</tr>
<tr>
<td><strong>ALPHA2</strong></td>
<td>0.0</td>
<td></td>
<td>Coefficient of $(\Delta p/p)^3$ in series for path length difference</td>
</tr>
<tr>
<td><strong>ALPHA3</strong></td>
<td>0.0</td>
<td></td>
<td>Coefficient of $(\Delta p/p)^4$ in series for path length difference</td>
</tr>
<tr>
<td><strong>W0I</strong></td>
<td>None</td>
<td>MeV</td>
<td>Kinetic energy on the central orbit at $t = T_I$</td>
</tr>
<tr>
<td><strong>W0F</strong></td>
<td>0.0</td>
<td>MeV</td>
<td>Kinetic energy on the central orbit at $t = T_F$ (see text)</td>
</tr>
<tr>
<td><strong>P0I</strong></td>
<td>None</td>
<td>MeV/c</td>
<td>Momentum on the central orbit at $T = T_I$</td>
</tr>
<tr>
<td><strong>P0F</strong></td>
<td>0.0</td>
<td>MeV/c</td>
<td>Momentum on the central orbit at $t = T_F$ (see text)</td>
</tr>
<tr>
<td><strong>TI</strong></td>
<td>0.0</td>
<td>s</td>
<td>Start time of magnetic field change$^a$</td>
</tr>
<tr>
<td><strong>TF</strong></td>
<td>0.0</td>
<td>s</td>
<td>End time of magnetic field change</td>
</tr>
<tr>
<td><strong>T0</strong></td>
<td>0.0</td>
<td>s</td>
<td>Time at which to set curve value to $W_0I$ ($P_0I$)</td>
</tr>
<tr>
<td><strong>TM</strong></td>
<td>0.0</td>
<td>s</td>
<td>Time of ramp maximum $W_0F$ ($P_0F$) if different from end time $T_F$</td>
</tr>
<tr>
<td><strong>TSTART</strong></td>
<td>0.0</td>
<td>s</td>
<td>Time at which tracking begins</td>
</tr>
<tr>
<td><strong>FRAC</strong></td>
<td>1.0</td>
<td></td>
<td>Determines azimuthal periodicity, calculation restricted to $-180^\circ/FRAC \leq \vartheta \leq 180^\circ/FRAC$</td>
</tr>
<tr>
<td><strong>PIPRAD</strong></td>
<td>1.0</td>
<td>m</td>
<td>Radius of beam pipe for particle loss</td>
</tr>
<tr>
<td><strong>EBDRY</strong></td>
<td>F</td>
<td></td>
<td>Switch to set absorbing beam pipe walls at $REQ \pm PIPRAD$</td>
</tr>
<tr>
<td><strong>DES</strong></td>
<td>0.0</td>
<td>MeV</td>
<td>Constant energy offset of synchronous orbit relative to reference orbit</td>
</tr>
<tr>
<td><strong>KURVEB</strong></td>
<td>1</td>
<td></td>
<td>Magnetic field ramp from $W_0I$ to $W_0F$ or $P_0I$ to $P_0F$:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 – Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 – Increasing parabolic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 – Biased sinusoidal (See text for definitions CRA, CRB, CRC, TR3.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 – Arbitrary ramp table read from file given by FILRMP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 – Parabolic with final slope $W_0FDOT$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6 – Parabolic with final slope $P_0FDOT$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7 – Cubic with zero initial slope and curvature</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 – Decreasing parabolic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9 – Ramp table from file FILRMP - polynomial representation</td>
</tr>
<tr>
<td><strong>CRA</strong></td>
<td>0.0</td>
<td></td>
<td>Time shift of first harmonic of $KURVEB=3$ ramp</td>
</tr>
<tr>
<td><strong>CRB</strong></td>
<td>0.0</td>
<td></td>
<td>Relative amplitude of second harmonic in $KURVEB=3$</td>
</tr>
<tr>
<td><strong>CRC</strong></td>
<td>0.0</td>
<td></td>
<td>Time shift of second harmonic for $KURVEB=3$</td>
</tr>
<tr>
<td><strong>TR3</strong></td>
<td>0.0</td>
<td>s</td>
<td>Half-period of fundamental for $KURVEB=3$; $TR3=0. \Rightarrow$ half-period $= TF - TI$</td>
</tr>
<tr>
<td><strong>W0IDOT</strong></td>
<td>0.0</td>
<td>MeV/s</td>
<td>Slope of parabolic ramp at $T_I$ ($KURVEB=5$)</td>
</tr>
<tr>
<td><strong>W0FDOT</strong></td>
<td>0.0</td>
<td>MeV/s</td>
<td>Slope of parabolic ramp at $T_F$ ($KURVEB=5$)</td>
</tr>
<tr>
<td><strong>P0IDOT</strong></td>
<td>0.0</td>
<td>MeV/c/s</td>
<td>Slope of parabolic ramp at $T_I$ ($KURVEB=6$)</td>
</tr>
<tr>
<td><strong>P0FDOT</strong></td>
<td>0.0</td>
<td>MeV/c/s</td>
<td>Slope of parabolic ramp at $T_F$ ($KURVEB=6$)</td>
</tr>
<tr>
<td><strong>JNRAMP</strong></td>
<td>F</td>
<td></td>
<td>Establishes starting point of ramp as point at which program finds itself—for smoothly piecing ramp segments together</td>
</tr>
<tr>
<td><strong>FILRMP</strong></td>
<td>'DUMMY'</td>
<td></td>
<td>Name for file containing ramp table ($KURVEB=4,9$) (see 3.2.3)</td>
</tr>
</tbody>
</table>

$^a$TI will override the initial time value in ramp table ($KURVEB=4$), if it differs, and shift all time values consistently. (see 3.2.3)
**R Command, Namelist /RING/, continued**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMAJMP</td>
<td>F</td>
<td>-</td>
<td>Set $\gamma_T$-jump on (See text for compatibility.)</td>
</tr>
</tbody>
</table>
| KINDG      | 1             | -      | Type of $\gamma_T$ variation:  
1 - Linear ($\gamma_T = \text{GAMPAR}(1) + \text{GAMPAR}(2) \cdot t$)  
2 - Decreasing exponential ($\gamma_T = \text{GAMPAR}(1) + \text{GAMPAR}(3) \cdot (1 - e^{-t/\text{GAMPAR}(2)})$) |
| GAMPAR(1:3) | 0.0           | -      | Coefficients for $\gamma_T$ variation (real $\gamma_T$ only)               |
| CHGNO      | 1.0           | -      | Charge of beam particle in units $|e|$                                     |
| EM0CSQ     | 938.27231 MeV |        | Rest energy of beam particle                                                |

$^aT = 0$ corresponds to time at which R command is invoked with GMAJMP = .TRUE.

The members of NAMELIST /RING/ are stored in COMMON /RINGP/ by SUBROUTINE RINGPAR, which also derives quantities depending only on lattice parameters. It is possible to distinguish the reference trajectory, of energy $E_0$, from the synchronous trajectory, of energy $E_s$, by specifying an offset DES from the reference energy.

An example of a simple R command is

$$\text{SRING } \text{REQ}=1000. \text{ GAMTSQ}=351.56 \text{ W0I}=150000. \text{ FRAC}=159. \text{ $END}$$

in which the lattice is characterized simply by a radius, $\gamma_T$ value, and a field which corresponds to a reference kinetic energy of 150 GeV. FRAC is useful for restricting the range of consideration of ESME to a suitable period of the ring. In this case, since W0F is not specified, the default, W0F = 0.0, indicates that the guide field does not change. In addition, the start time for this run is taken to be zero. Notice that it is possible to treat a lattice with imaginary $\gamma_T$ simply by entering a negative value for GAMTSQ.

There is some subtlety associated with the use of the FRAC parameter. It is a real quantity but meaningful only for positive integer values. It has as one effect the setting the fundamental frequency for FFT’s to FRAC × the beam circulation frequency. (See Secs. 2.2.12 and 2.2.13 on the B and F commands.) A typical use is to calculate the results for a full ring by following a single bunch with FRAC equal to the harmonic number.

For time $\leq T_I$ the kinetic energy on the reference orbit is taken to be $W_{0I}$, while for time $\geq T_F$ it is $W_{0F}$. Either or both quantities $W_{0I},W_{0F}$ (kinetic energy) may optionally be replaced by $P_{0I},P_{0F}$ (momentum).

Resonant fast cycling synchrotrons like the Fermilab booster have a practically sinusoidal ramp

$$B(t) = B_i + (B_f - B_i) \left[ 1 - \cos \left( \frac{\pi}{2} \frac{t - t_i}{T - t_i} \right) \right] / 2 .$$

This is the curve one gets for KURVEB=3 if the four constants CRA, CRB, CRC, TR3 in the R command are all left at the default value of zero. Take a time variable $x = (t - t_i)/(T - t_i)$ in which the half period $T$ is TR3 for TR3 $\neq 0$ and $t_f - t_i$ otherwise. The full expression for KURVEB=3 is

$$B(t) = B_i + (B_f - B_i) \frac{- \cos \pi(x + a) - b \cos 2\pi(x + c) + \cos \pi a + b \cos 2\pi c}{2 \cos \pi a} .$$

The fraction is zero for $x = 0$ and one for $x = 1$. The data item CRB is the constant $b$ in the above expression; it allows the addition of second harmonic for such purposes as reducing rf power requirements for the cycle. The constant $a$ (CRA) permits starting the ramp with a slope as one might want simulating injection starting before $\dot{B} = 0$. CRC could be used to move the second harmonic along with the first or, perhaps, to turn the $\cos$ into a $\sin$. The half-period value TR3, if specified, allows the use of sinusoidal ramps with a time scale independent of the times $T_I$ and $T_F$ bounding the period in which the curve is used. Notice that if the half-period is not $t_f - t_i$, $B_f$ is not the magnetic field at the end of the ramp; it is rather the maximum value, the value the field will have at $t=TM$; note that TM is TF by default; if the ramp is wanted beyond the time of its maximum, TM should be specified.
Perhaps the KURVEB=7 option seems a little obscure. This curve segment has been used to represent what is called the voltage parabola for the Fermilab main ring power supplies, which did not like to turn on with a finite voltage slope; presumably other high current supplies have similar properties.

If the $\gamma_T$-jump option has been invoked, then $\gamma_T$ is varied until the R command is issued with GMAJMP = F; it applies only to real $\gamma_T$. It is a special purpose option (euphemism for kluge) which is incompatible with other uses of the R command. If it is necessary to change other lattice parameters at the time when a $\gamma_T$-jump is also to be initialized, the ordinary lattice parameters should be entered in a first R command with GMAJMP = F; the jump should then be set up in a second command with GMAJMP = T.

Occasionally it is useful to model the steps immediately preceding and following a transfer between accelerators in a single ESME run. If all of the parameters for the second machine are included in new R, A, B, etc., the simulation will proceed from machine to machine. The distribution will be scaled in $\vartheta$ to account for change in the reference orbit radius $R_{eq}$. 


### 2.2.2 A Command - RF parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRF</td>
<td>1</td>
<td></td>
<td>Number of active RF sources&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>H(1:10)</td>
<td>1</td>
<td></td>
<td>Harmonic numbers of sources (integers); a negative value forces the principal system.</td>
</tr>
<tr>
<td>HW(1:10)</td>
<td>1</td>
<td></td>
<td>Voltage sources will be expressed over the $\theta$ range $-\frac{180^\circ}{\text{HW}} \leq \theta \leq \frac{180^\circ}{\text{HW}}$ to produce isolated buckets or barriers</td>
</tr>
<tr>
<td>ISYNC</td>
<td>0</td>
<td></td>
<td>Indicates synchronism condition to be imposed on RF:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 – None, voltages and phases remain as programmed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 – Phase of RF waveform shifted to synchronous stable point</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 – RF amplitude scaled to give synchronous energy with no change of phase</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 – Source 2 Landau cavity to source 1, synchronism assured only for sources 1 and 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 – Determines synchronous phase for source 1 only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 – Determines synchronous phase for the principal source&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6 – Determines synchronous phase for source 1 wrt B-dot only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7 – Determines synchronous phase for principal source wrt B-dot only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 – Like ISYNC=1 except energy loss to real impedance accounted for</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9 – Like ISYNC=1 except voltages scaled to account for loss to real $Z_{</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10 – Like ISYNC=2 except energy loss to real impedance accounted for</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11 – Like ISYNC=3 except energy loss to real impedance accounted for</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12 – Like ISYNC=4 except energy loss to real impedance accounted for</td>
</tr>
<tr>
<td>VI(1:10)</td>
<td>0.0</td>
<td>MV</td>
<td>Voltage ($&gt; 0$) of source I at time TVBEG(I)</td>
</tr>
<tr>
<td>VF(1:10)</td>
<td>0.0</td>
<td>MV</td>
<td>Voltage ($&gt; 0$) of source I at time TVEND(I)</td>
</tr>
<tr>
<td>TVBEG(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to beginning of RF voltage change</td>
</tr>
<tr>
<td>TVEND(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to end of RF voltage change</td>
</tr>
<tr>
<td>TV0(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to VI if different from TVBEG</td>
</tr>
<tr>
<td>TVM(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time of voltage curve maximum if different from TVEND</td>
</tr>
<tr>
<td>KURVE(1:10)</td>
<td>0</td>
<td></td>
<td>Specifies type of RF voltage variation between times TVBEG and TVEND:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 – None, voltage maintained at VI(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 – Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 – Iso-adiabatic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 – Sigmoid</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 – Fit and linear or cubic spline interpolation of voltage table&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 – Parabolic, initial slope zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6 – Fit and Bernstein polynomial interpolation of voltage table&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
<tr>
<td>NTV(1:10)</td>
<td>0</td>
<td></td>
<td>Number of points to be taken from voltage table (KURVE=4,6)</td>
</tr>
<tr>
<td>VKON</td>
<td>T</td>
<td></td>
<td>Indicates whether programmed voltage curves are to be active</td>
</tr>
</tbody>
</table>

<sup>a</sup>The permissable number of systems is set by the parameter NSRC in the parameters.f module.

<sup>b</sup>that is, the source that taken alone give the greatest bucket height

<sup>c</sup>Fit to values read from file. See Section 3.2.2.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Default Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSII(1:10)</td>
<td>0</td>
<td>deg</td>
<td>Phase of source I at time TPBEG(I)</td>
</tr>
<tr>
<td>PSIF(1:10)</td>
<td>0</td>
<td>deg</td>
<td>Phase of source I at time TPEND(I)</td>
</tr>
<tr>
<td>TPBEG(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to beginning of RF phase change</td>
</tr>
<tr>
<td>TPEND(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to end of RF phase change</td>
</tr>
<tr>
<td>TP0(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to PSII if different from TPBEG</td>
</tr>
<tr>
<td>TPM(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to PSIF if different from TPEND</td>
</tr>
<tr>
<td>KURVP(1:10)</td>
<td>0</td>
<td>-</td>
<td>Specifies type of RF phase variation between times TPBEG and TPEND:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - None, phase maintained at PSII(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Quadratic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 - Fit and cubic spline interpolation of phase table(c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 - Fit and Bernstein polynomial interpolation of phase table(c)</td>
</tr>
<tr>
<td>NTP(1:10)</td>
<td>0</td>
<td></td>
<td>Number of points to be taken from phase table (KURVP=4,5)</td>
</tr>
<tr>
<td>PHKON</td>
<td>F</td>
<td>-</td>
<td>Indicates whether or not phase curves are to be active</td>
</tr>
<tr>
<td>FRI(1:10)</td>
<td>0.0</td>
<td>MHz</td>
<td>Frequency of source I at time TFBEG(I)</td>
</tr>
<tr>
<td>FRF(1:10)</td>
<td>0.0</td>
<td>MHz</td>
<td>Frequency of source I at time TFEND(I)</td>
</tr>
<tr>
<td>TFBEG(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to beginning of frequency change</td>
</tr>
<tr>
<td>TFEND(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to end of frequency change</td>
</tr>
<tr>
<td>TF0(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to FRI if different TFBEG</td>
</tr>
<tr>
<td>TFM(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to FRF if different TFEND</td>
</tr>
<tr>
<td>KURVF(1:10)</td>
<td>0</td>
<td>-</td>
<td>Specifies type of frequency variation between times TFBEG and TFEND:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - None, frequency maintained at FRI(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Quadratic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 - Sigmoid</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 - Fit and cubic spline interpolation of frequency table(c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 - Fit and Bernstein polynomial interpolation of frequency table(c)</td>
</tr>
<tr>
<td>NTF(1:10)</td>
<td>0</td>
<td></td>
<td>Number of points to be taken from frequency table (KURVF=4,5)</td>
</tr>
<tr>
<td>FRKON</td>
<td>F</td>
<td>-</td>
<td>Indicates whether frequency curves are to be active</td>
</tr>
<tr>
<td>DELTRFI(1:10)</td>
<td>0.0</td>
<td>MeV/c</td>
<td>Energy offset of source I at time TDBEG(I)</td>
</tr>
<tr>
<td>DELTRFF(1:10)</td>
<td>0.0</td>
<td>MeV/c</td>
<td>Energy offset of source I at time TDEND(I)</td>
</tr>
<tr>
<td>TDBEG(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to beginning of momentum offset change</td>
</tr>
<tr>
<td>TDEND(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to end of momentum offset change</td>
</tr>
<tr>
<td>TD0(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to DELTRFI if different from TDBEG</td>
</tr>
<tr>
<td>TDM(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to DELTRFF if different from TDEND</td>
</tr>
<tr>
<td>KURVD(1:10)</td>
<td>0</td>
<td>-</td>
<td>Specifies type of momentum offset variation between times TDBEG and TDEND:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - None, frequency maintained at DELTRFI(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Quadratic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 - Sigmoid</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 - Fit and cubic spline interpolation of momentum offset table(c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 - Fit and Bernstein polynomial interpolation of momentum offset table(c)</td>
</tr>
<tr>
<td>NTD(1:10)</td>
<td>0</td>
<td></td>
<td>Number of points to be taken from momentum offset table (KURVD=4,5)</td>
</tr>
<tr>
<td>DKON</td>
<td>F</td>
<td>-</td>
<td>Indicates whether momentum offset curves are to be active</td>
</tr>
<tr>
<td>FILCRV</td>
<td>'DUMMY'</td>
<td>-</td>
<td>Full path for voltage, phase, and frequency curves file</td>
</tr>
</tbody>
</table>
A Command, Namelist /RF/, continued

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTINU(1:10)</td>
<td>F</td>
<td>-</td>
<td>Sets the starting voltage, phase and/or frequency for the corresponding sources to the current values — for smoothly piecing curve segments together.</td>
</tr>
<tr>
<td>VMATCH(1:10)</td>
<td>F</td>
<td>-</td>
<td>VMATCH(I) = T sets VI(I) so that source I matches the emittance of the current distribution.</td>
</tr>
<tr>
<td>HOLDBH</td>
<td>F</td>
<td>-</td>
<td>Switch to calculate the voltage of principal source to hold the current bucket height. (see also HDECR)</td>
</tr>
<tr>
<td>HDECR</td>
<td>1.0</td>
<td>-</td>
<td>Factor by which bucket height for principal source is to be adjusted on successive iterations if HOLDBH = T.</td>
</tr>
<tr>
<td>BH HOLD</td>
<td>0. MeV</td>
<td></td>
<td>If not zero, calculation triggered by HOLDBH will aim for BH HOLD.</td>
</tr>
<tr>
<td>HOLDBA</td>
<td>F</td>
<td>-</td>
<td>Switch to calculate the voltage of principal source to hold the current bucket area. (see also SDECR)</td>
</tr>
<tr>
<td>SDECR</td>
<td>1.0</td>
<td>-</td>
<td>Factor by which bucket area for principal source is to be adjusted on successive iterations if HOLDBA = T.</td>
</tr>
<tr>
<td>BAHOLD</td>
<td>0. eVs</td>
<td></td>
<td>If not zero, calculation triggered by HOLDBA will aim for BAHOLD.</td>
</tr>
<tr>
<td>PHISLIM</td>
<td>.95</td>
<td>-</td>
<td>Voltage may not be reduced such that $\sin \phi_s &gt; \text{PHISLIM}$ using options HOLDBH and HOLDBA.</td>
</tr>
<tr>
<td>PHSLIP</td>
<td>F</td>
<td>-</td>
<td>Switch indicating that the phase of at least one source is to be varied to correspond to an energy offset from the synchronous value (see DELTRFI/F).</td>
</tr>
</tbody>
</table>

*Which means, in this instance, that a P command, or its equivalent, should precede the A command.  
*The constant bucket area or bucket height is calculated from the rf source which gives the greatest bucket height.

The members of NAMELIST /RF/ are read in SUBROUTINE RFPROG, and stored in module RFP. Up to 10 independent voltage sources may be specified. An example of a simple A command is

```
SRF H=1113 VI=.100 SEND
```

Here, one voltage is specified by a minimum set of parameters. H(I) is an integer (as is HW(I)). This example illustrates that a constant rf amplitude requires that only VI be given. The amplitude is always taken > 0. The default condition in ESME is that rf frequency is determined by the synchronous particle circulation frequency.

Since RF manipulations are at the heart of ESME, this command can become rather lengthy and involved; the example given above is exceptionally brief. As in the case of the magnetic field ramp, specified in the R command, a time dependent parameter is maintained at its initial value at times prior to the start of a programmed variation, while at times after the end of a program curve, the quantity is maintained at its final value. This makes it easier to specify multiple curves, perhaps having unequal time steps. Particular attention is called to the CNTINU flags which help join segments smoothly by setting initial values to current values. Versions of ESME later than August 2006 differ from the earlier in the treatment of programmed parameter curves. The only parameters which can be effective with an initial value set without the corresponding variation switch set .TRUE. are VI(1:NRF). Therefore, it will frequently be necessary to have curves which maintain some constant value. For example, a constant energy offset of 10 MeV can be introduced as

```
DKON=T TDEND=1. KURVD=1 DELTRFI=10. DELTRFF=10.
```

where TDBEG defaults to zero. The more lengthy specification has resulted from an effort to make the parameter curves more adaptable and less mysterious. A very common case is above transition where it is now necessary to have a phase curve with PSII = PSIF = 180 deg. Although this novelty seems petty and irritating, the change has multiple benefits.
With suitable caution the curves that define rf phase can be used in combination. A frequency curve and an energy offset curve are two ways of obtaining the same result. If both FRKON and DKON are true and KURVF and KURVD are non-zero for the same system, the frequency curve takes precedence. The frequency and offset curves govern the phase slip contribution to the rf phases, but if the beam is changing energy, there will be induced synchrotron oscillation unless the phase is set correctly by a phase curve or a SHAZAM subroutine.

The values of the programmed phase curves are to be distinguished from the phase of a given harmonic at the synchronous particle, though they may be the same. For example, a user could represent any periodic waveform (in a Fourier expansion of up to 10 terms) simply by specifying the correct relative phases and amplitudes of the voltages. The waveform could be modified over time by specifying the variations of the voltages, phases, and/or frequencies.

Generally, the user will specify that the program search for a synchronous point on the rf waveform. Option ISYNC = 1 will result in the program searching for a stable value of the phase, an h = 1 phase shift of the sum voltage waveform. Option ISYNC = 2 will cause the voltages to be scaled to give only the correct magnitude of total voltage at the synchronous particle, not necessarily at a stable slope, because the phases of the voltage sources remain at their programmed values. The so called Landau cavity option, ISYNC = 3, will vary the phases of voltage sources 1 and 2, and the magnitude of voltage source 2, so that the first and second derivatives of the voltage vanish at the synchronous particle. The phases and magnitudes of other voltages are not altered, although they will be included in the iteration of the difference equation if NRF > 2. ISYNC = 4 results in the calculation of the synchronous phase from the single system which gives the greatest bucket height. This system is called the principal system. ISYNC = 5 is very similar to ISYNC = 4 except that system 1 is used regardless of which is the principal system. These two options do not effect the parameters of any other systems. Options ISYNC = 6 and ISYNC = 7 are like ISYNC = 4 and ISYNC = 5 respectively except that they synchronize to the effect of B only.

ISYNC = 8 – 12 introduce an entirely different sort of behavior; these options are fundamentally multiparticle, whereas historically ESME has been a single particle dynamics code with several multiparticle accessories added. In each case the energy lost to a real part of the longitudinal impedance is automatically accounted in calculating the synchronous phase and voltage. This uses information available to the program that is not as directly available in practice. Unfortunately, one is often rather ignorant of the realistic longitudinal impedance. One way to connect with practice is to use the voltage and phase curves generated by the program as low level rf curves for the accelerator being modeled (the ultimate analog computer). In any case, the results of these latter options should be interpreted carefully. A meaningful use of ISYNC = 8 – 12 almost surely will require a well bunched beam without very much uncaptured beam present. If this condition is not observed, the significance of any results should be checked in some way. Phase and frequency curves can be used simultaneously with or without ISYNC > 0, but the interaction takes some consideration. A few combinations have been tested, but the checking certainly has not exhausted the possibilities. The simplest possible model containing the essentials is the safest; elaboration to more nearly mimic the accelerator to be modeled can be waste effort and could introduce unanticipated complication arising either from the structure of the code or from the beam dynamics.

Although ISYNC options 1, 8, and 9 adjust the phase of multiple rf systems, they will not achieve the intended effect for many of the purposes for which the additional systems have been intended. For example, acceleration with a waveform flattened by a second or third harmonic system is correctly managed with ISYNC = 3 or 11. All of the ISYNC options include synchronization to an energy change resulting from frequency or phase curves.

As noted in the table, the options activated by HOLDBH and VOLDBA are calculated from the bucket height or bucket area respectively of the principal rf system. These options provide an extremely convenient way to automatically generate a reasonable voltage program for any ramp curve. The appropriate synchronization options are ISYNC = 1, ISYNC = 8, ISYNC = 4, ISYNC = 5, ISYNC = 6, ISYNC = 7, or ISYNC = 12. Because there will generally be only one source when the HOLD options are used, the choices are most naturally between ISYNC = 1, ISYNC = 8, ISYNC = 6, and ISYNC = 12. Notice that the bucket area or height can be held to the value calculated from the rf voltage specified in the A command or to a target value BAHOLD or BHHOLD.

The feedback options are useful but require some trial and error on gain etc. for optimum performance. For example, even though the gain for ITFB = 0 or 1 is set for critical damping according to the linearized EOM, the best value of FBFACT may be even something like 5 or 10 depending on emittance, initial error, and perhaps other parameters to a lesser degree. The default gain for voltage feedback is frankly arbitrary. Experience has shown
that the optimum VFBFCTR may be larger than one in rather routine cases. One may expect that the two types of feedback can only be independent in cases of linear rf potential and very small initial error; that is, a linear approximation should be appropriate. A cautious approach is to avoid simultaneous phase and voltage feedback, damping the dominant oscillation first and then the other. However, simultaneous damping often works rather well.
### P Command - Initial distribution parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KIND</td>
<td>1</td>
<td>-</td>
<td>Chose...</td>
</tr>
<tr>
<td>NPOINT</td>
<td>1</td>
<td>-</td>
<td>Number of particles generated for all distributions except KIND = 1, 2, 6, 8, in which NTH and NE are used</td>
</tr>
<tr>
<td>NTH</td>
<td>2</td>
<td>-</td>
<td>Number of grid points in ( \theta ) direction</td>
</tr>
<tr>
<td>NE</td>
<td>2</td>
<td>-</td>
<td>Number of grid points in E direction</td>
</tr>
</tbody>
</table>

- **KIND**: 
  1–Rectangular outline, NTH by NE points, limited by THMIN, THMAX, REMIN, REMAX (see text) 
  2–Uniform rectangular grid NTH by NE, limits as in KIND = 1 
  3–Random uniform distribution of NPOINT points within rectangular limits as in KIND = 1 
  4–Random uniform in \( \theta \), limits THMIN, THMAX; Gaussian in E, REMIN, REMAX = \( \pm 2\sigma \), NPOINT points 
  5–Gaussian in \( \theta \), THMIN,THMAX = \( \pm 2\sigma \); random uniform in E, limits REMIN, REMAX, NPOINT points 
  6–Rectangular grid, regular in \( \theta \), Gaussian in E, NTH by NE points 
  7–Bunch outline of NPOINT particles (see text) 
  8–Bunch filled with regular grid of approximately NTH by NE particles 
  9–Random uniform bunch of NPOINT particles within contour 
  10–Bi-Gaussian distribution of NPOINT particles, 95% within contour 
  11–NPOINT uniformly spaced particles on flow lines just above and below bucket boundary (see text) 
  12–NPOINT particles, random uniform in E, limits REMIN,REMAX, parabolic in \( \theta \), limits THMIN,THMAX 
  13–Parabolic bunch of NPOINT particles 
  14–Elliptical (aka Hoffman-Pedersen) bunch of NPOINT particles 
  15–NPOINT particles, random uniform in \( \theta \), limits THMIN,THMAX; parabolic in E, limits at REMIN,REMAX 
  16–NPOINT particles within matched contour, low-noise uniform quasi-random 
  17–NPOINT particles within matched contour, low-noise Gaussian quasi-random 
  18–NPOINT particles in matched contour, low-noise uniform pseudo-random (Sobel sequence) 
  19–NPOINT particles in matched contour, low-noise Gaussian pseudo-random (Sobel sequence) 
  20–NPOINT particles, pseudo-random within limits REMIN,REMAX, THMIN,THMAX 
  21–Read particle coordinates from file FILDST into a single partition 
  22–Place a pointer particle at THMIN,REMIN and one at THMAX,REMAX, in separate partitions (see text) 

- **NPOINT**: 
  - distribution will not contribute to collective potential 
- **distribution will contribute to collective potential**

---

<p>| 20 |</p>
<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Default Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>THMIN</td>
<td>-90.0</td>
<td>deg</td>
<td>Lower $\theta$ limit on rectangular distributions</td>
</tr>
<tr>
<td>THMAX</td>
<td>90.0</td>
<td>deg</td>
<td>Upper $\theta$ limit on rectangular distributions</td>
</tr>
<tr>
<td>REMIN</td>
<td>None</td>
<td>MeV</td>
<td>Lower energy limit wrt synchronous energy for rectangular distributions</td>
</tr>
<tr>
<td>REMAX</td>
<td>None</td>
<td>MeV</td>
<td>Upper energy limit on rectangular distributions</td>
</tr>
<tr>
<td>SBNCH</td>
<td>0.1</td>
<td>eVs</td>
<td>Area within limiting contour</td>
</tr>
<tr>
<td>IPOP</td>
<td>1</td>
<td>-</td>
<td>Specifies RF source for generating bunch limiting contour:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0–All active (NRF) sources</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1–Source I (1 $\leq$ I $\leq$ NRF)</td>
</tr>
<tr>
<td>THOFF</td>
<td>0.0</td>
<td>deg</td>
<td>Displacement of currently generated distribution in $\theta$ direction</td>
</tr>
<tr>
<td>EOFF</td>
<td>0.0</td>
<td>MeV</td>
<td>Amount to displace currently generated distribution in E direction</td>
</tr>
<tr>
<td>THTRAN</td>
<td>0.0</td>
<td>deg</td>
<td>Displacement of currently and previously generated distributions in $\theta$ direction</td>
</tr>
<tr>
<td>ETRAN</td>
<td>0.0</td>
<td>MeV</td>
<td>Amount to displace all particles in E direction</td>
</tr>
<tr>
<td>WINJ</td>
<td>0.0</td>
<td>MeV</td>
<td>Kinetic energy of injection $^a$</td>
</tr>
<tr>
<td>PINJ</td>
<td>0.0</td>
<td>MeV/c</td>
<td>Injection momentum, alternative to WINJ above ($q$. v.)</td>
</tr>
<tr>
<td>NREPT</td>
<td>0</td>
<td>-</td>
<td>Number of times a matched bunch distribution should be replicated $^b$</td>
</tr>
<tr>
<td>HREPT</td>
<td>1</td>
<td>-</td>
<td>The harmonic number at which replications are to be made</td>
</tr>
<tr>
<td>DITHTH</td>
<td>0.0</td>
<td>deg</td>
<td>An optional rms azimuth for Gaussian centroid scatter</td>
</tr>
<tr>
<td>DITHE</td>
<td>0.0</td>
<td>MeV</td>
<td>An optional rms energy for Gaussian centroid scatter</td>
</tr>
<tr>
<td>IBCKT(1:750)</td>
<td>0</td>
<td>-</td>
<td>The number(s) of buckets relative to original bunch ($\pm$)</td>
</tr>
<tr>
<td>D0TH(1:750)</td>
<td>0.0</td>
<td>deg</td>
<td>Displacement of bunch in azimuth relative to its own bucket center</td>
</tr>
<tr>
<td>D0EN(1:750)</td>
<td>0.0</td>
<td>MeV</td>
<td>Displacement of bunch in energy relative to its own bucket center</td>
</tr>
<tr>
<td>PARTION</td>
<td>T</td>
<td>-</td>
<td>Partition distribution into separate classes; $^c$</td>
</tr>
<tr>
<td>RENORM</td>
<td>T</td>
<td>-</td>
<td>Calculate ANORM for matched bunch $\leftarrow$ EPSILON $\equiv$ SBNCH. $^d$</td>
</tr>
<tr>
<td>FILDST</td>
<td>'DUMMY'</td>
<td>-</td>
<td>File name for coordinate pairs (KIND=21)</td>
</tr>
</tbody>
</table>

$^a$If both WINJ and PINJ are at the 0.0 default value, injection energy is $E_\circ$ of the central orbit.

$^b$The bunch replication feature requires PARTION = T (the default). The number of replications is limited by LTABLE from the parameters.f module.

$^c$Different classes of particles may be plotted with distinct symbols. Each separate use of the P command with PARTION = T introduces at least one new class.

$^d$RENNORM is reset to .FALSE. after each use.

The members of NAMELIST /POPL8/ are read in SUBROUTINE POPUL8, and stored in the module POPLATE. To populate a bi-Gaussian distribution of 100 particles, with an emittance (95%) matched to RF source 1 of .02 eV-s, the P command would be

```
$POPL8 KIND=10 NPOINT=100 IPOP=1 SBNCH=.02 $END
```

For multi-bunch simulations, in which several distributions are needed with the same parameters but in different positions, it is sufficient to simply re-issue the P command with only the desired position offset. For example

```
$POPL8 THTRAN=45.0 ETRAN=0.0 $END
```

Since NAMELIST members which do not appear in the input remain unchanged, one can limit the input by giving only the parameters with changed values. In particular, THTRAN and ETRAN make it unnecessary to explicitly declare the position of each group of particles.

The variables HREPT, NREPT, DITHTH, DITHE, IBCKT, D0TH, and D0EN implement the features of a long popular SHAZAM for repeating a bunch in different buckets with the addition of an optional random Gaussian uncertainty in centroid location. The number of repeated bunches is limited by the compilation parameter LTABLE.
Setting RENORM .TRUE. will result in the inclusion of a factor in the emittance which scales the rms emittance of the initial bunch to its SBNCH value. So long as the bunch form remains similar, the emittance so calculated gives an approximate value for the total bunch area. Generally it seems safer to use the unscaled rms emittance.

An arbitrary distribution can be read from an external file named by the string variable FILDST by using KIND=21. The file has the particle count in the first record and coordinate pairs in succeeding records. Another somewhat special distribution is indicated by KIND=22, a pair of points each given a separate partition. These points may be placed arbitrarily at coordinates given by THMIN,REMIN and THMAX,REMAX to mark special locations. A SHAZAM test on the alignment of these pointers can be a simple way to achieve precise timing.

There are four KINDs (1, 7, 11, 22) which distribute particles that do not contribute to the collective potential calculation; these macroparticles are usually used as witnesses to the effects of more physically plausible distributions. Another way to generate such distributions is via the C command, which places macroparticles along single particle phase space trajectories. The phase space coordinates of such witness particles are stored in a partition immediately above particles which can contribute to the collective potential. To make this convenient separation work, none of the other distributions should be used after the four cited. If only single particle motion is to be tracked, the order of use of the different KINDs is of no consequence.
### 2.2.4 O Command - Graphical output options

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLOT</td>
<td>1000</td>
<td></td>
<td></td>
<td>Output every MPLOT iterations</td>
</tr>
<tr>
<td>POSTP</td>
<td>F</td>
<td></td>
<td></td>
<td>Write all data in COMMON blocks to unit 18; do not call plotting routine.</td>
</tr>
<tr>
<td>TITL</td>
<td>exists</td>
<td></td>
<td></td>
<td>String of up to 50 characters for plot headings</td>
</tr>
<tr>
<td>PLTSW</td>
<td></td>
<td></td>
<td>Select plot options:</td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td>T</td>
<td></td>
<td></td>
<td>Draw phase space plot</td>
</tr>
<tr>
<td>(2)</td>
<td>T</td>
<td></td>
<td></td>
<td>Plot phase space points (different symbol for each class)</td>
</tr>
<tr>
<td>(3)</td>
<td>F</td>
<td></td>
<td></td>
<td>Interconnect points within each class</td>
</tr>
<tr>
<td>(4)</td>
<td>F</td>
<td></td>
<td></td>
<td>Draw lines at centroid and $\pm \sigma$</td>
</tr>
<tr>
<td>(5)</td>
<td>F</td>
<td></td>
<td></td>
<td>Draw voltage waveform</td>
</tr>
<tr>
<td>(6)</td>
<td>F</td>
<td></td>
<td></td>
<td>Set plot boundaries to turning points of contour</td>
</tr>
<tr>
<td>(7)</td>
<td>F</td>
<td></td>
<td></td>
<td>Suppress captions, axis labels, etc.</td>
</tr>
<tr>
<td>(8)</td>
<td>T</td>
<td></td>
<td></td>
<td>Plot $\theta$ histogram</td>
</tr>
<tr>
<td>(9)</td>
<td>F</td>
<td></td>
<td></td>
<td>Set $\theta$ histogram limits to turning points of contour</td>
</tr>
<tr>
<td>(10)</td>
<td>T</td>
<td></td>
<td></td>
<td>Plot $E$ histogram</td>
</tr>
<tr>
<td>(11)</td>
<td>F</td>
<td></td>
<td></td>
<td>Set $E$ histogram limits to turning points of contour</td>
</tr>
<tr>
<td>(12)</td>
<td>F</td>
<td></td>
<td></td>
<td>Plot Fourier amplitudes</td>
</tr>
<tr>
<td>(13)</td>
<td>F</td>
<td></td>
<td></td>
<td>Include phases in plot of Fourier spectrum</td>
</tr>
<tr>
<td>(14)</td>
<td>F</td>
<td></td>
<td></td>
<td>Plot a curve calculated in a shazam routine in place of rf waveform</td>
</tr>
<tr>
<td>(15)</td>
<td>F</td>
<td></td>
<td></td>
<td>Plot the real and imaginary impedance from the input</td>
</tr>
<tr>
<td>(17)</td>
<td>T</td>
<td></td>
<td></td>
<td>Start bucket contour at unstable fixed point $^a$</td>
</tr>
<tr>
<td>(18)</td>
<td>T</td>
<td></td>
<td></td>
<td>Start bucket contour above stable fixed point at $E = E_s + H_{bckt}$</td>
</tr>
<tr>
<td>(19)</td>
<td>T</td>
<td></td>
<td></td>
<td>Plot flow line points (different symbol for each class)</td>
</tr>
<tr>
<td>(20)</td>
<td>T</td>
<td></td>
<td></td>
<td>Interconnect flow line points within each class</td>
</tr>
<tr>
<td>(21)</td>
<td>F</td>
<td></td>
<td></td>
<td>Bunch perf. cond. wall voltage vs. $\vartheta$</td>
</tr>
<tr>
<td>(22)</td>
<td>F</td>
<td></td>
<td></td>
<td>Include distribution in previous plot</td>
</tr>
<tr>
<td>(23)</td>
<td>F</td>
<td></td>
<td></td>
<td>Wake field (time domain) voltage vs. $\vartheta$</td>
</tr>
<tr>
<td>(24)</td>
<td>F</td>
<td></td>
<td></td>
<td>Include distribution in previous plot</td>
</tr>
<tr>
<td>(25)</td>
<td>F</td>
<td></td>
<td></td>
<td>Frequency domain voltage vs. $\vartheta$ (inc. perf. cond. wall if FDSCON = T)</td>
</tr>
<tr>
<td>(26)</td>
<td>F</td>
<td></td>
<td></td>
<td>Include distribution in previous plot</td>
</tr>
<tr>
<td>(27)</td>
<td>F</td>
<td></td>
<td></td>
<td>Voltage from special time domain calculation for resonators vs. $\vartheta$</td>
</tr>
<tr>
<td>(28)</td>
<td>F</td>
<td></td>
<td></td>
<td>Include distribution in previous plot</td>
</tr>
<tr>
<td>(29)</td>
<td>F</td>
<td></td>
<td></td>
<td>Create don’t-plot area in phase plane defined by four points$^b$</td>
</tr>
<tr>
<td>(30)</td>
<td>F</td>
<td></td>
<td></td>
<td>Create don’t-plot area in phase plane defined by matched contour$^c$</td>
</tr>
<tr>
<td>(31)</td>
<td>F</td>
<td></td>
<td></td>
<td>PLTSW(31)=T defines don’t-plot area outside closed figure</td>
</tr>
<tr>
<td>(32)</td>
<td>F</td>
<td></td>
<td></td>
<td>Add collective voltage to voltage waveform (PLTSW(5)=T)</td>
</tr>
<tr>
<td>(34)</td>
<td>F</td>
<td></td>
<td></td>
<td>Replace Harmonic Number abcissa with Harmonic Frequency in Fourier amplitude plot (PLTSW(12)=T)</td>
</tr>
</tbody>
</table>

| NPJMP    | 1     |      | In phase space plot, plot only every NPJMPth point |
| KLPLLOT  | 0     |      | Select classes in phase space plot and projections (see Sec. 2.2.3) |
| IRF      | 1     |      | Selects voltage source for contour plotting: |

$^a$At least one of PLTSW(17) or PLTSW(18) must be true for ICONTUR=1 else program sets both .TRUE.
$^b$see definitions of THEXCM, THEXCPL, DEEXCM, DEEXCP, DEEXCPL
$^c$see definition of SEXCL
### O Command, Namelist /GRAPH/, continued

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
</table>
| ICONTUR  | 1             | -    | Select the type of reference contour to plot on phase space plot\(^a\)  
|          | 0 — No contour  
|          | 1 — Bucket contour  
|          | 2 — Contour of initial bunch area SBNCH  
|          | 3 — Contour of the specified area REFAREA  
|          | 4 — Contour containing 95% of the particles  
|          | 5 — Flow lines chosen by LINES and STVAL  
| REFAREA  | 0.1           | eVs  | Area of reference contour for ICONTUR = 3  
| LINES    | 1             | -    | Number of flow lines for ICONTUR = 5  
| STVAL    | 0,0.          |      | Starting values for contours, LINES coordinate pairs in \(\theta, E\)  
| THPMIN   | 0.0\(^b\)     | deg  | Lower \(\theta\) limit for phase space plot  
| THPMAX   | 0.0\(^b\)     | deg  | Upper \(\theta\) limit for phase space plot  
| DEPMIN   | 0.0\(^d\)     | MeV  | Lower \(E\) limit for phase space plot  
| DEPMAX   | 0.0\(^d\)     | MeV  | Upper \(E\) limit for phase space plot  
| RFVMIN   | 0.0\(^f\)     | MV   | Lower limit for optional rf waveform plot (PLTSW(5))  
| RFVMAX   | 0.0\(^f\)     | MV   | Upper limit for optional rf waveform plot (PLTSW(5))  
| IREF     | 1             | -    | Determines energy origin for phase space:  
|          | 1 — \(E_0\), the reference energy (often = \(E_\text{S}\))  
|          | 2 — \(E_\text{S}\), the synchronous energy  
|          | 3 — \(E_{\text{BAR}}\), the average particle energy  
|          | 4 — \(E_{\text{REF}}\), the reference particle energy\(^g\)  
| NBINTH   | 50            |      | The number of bins for the \(\theta\) histogram  
| THBMIN   | 0.0\(^g\)     | deg  | Lower limit for \(\theta\) histogram  
| THBMAX   | 0.0\(^g\)     | deg  | Upper limit for \(\theta\) histogram  
| NBINE    | 50            |      | The number of bins for the \(E\) histogram  
| EBMIN    | 0.0\(^h\)     | MeV  | Lower limit for \(E\) histogram  
| EBMAX    | 0.0\(^h\)     | MeV  | Upper limit for \(E\) histogram  
| IFBMIN   | 1             | -    | Lower limit for FFT plot - If specified, specify IFBMAX also  
| IFBMAX   | 0\(^i\)       |      | Upper limit for FFT plot  
| CVBMIN   | 0.0\(^k\)     | deg  | Lower \(\theta\) limit for all collective voltage plots  
| CVBMAX   | 0.0\(^k\)     | deg  | Upper limit for all collective voltage plots  
| THEXCI   | 0.            | deg  | Lower theta value defining rectangular don't plot area  
| THEXCLI  | 0.            | deg  | Upper theta value defining rectangular don't plot area  
| DEEXCMI  | 0.           | MeV  | Lower energy value defining rectangular don't plot area  
| DEEXCPL  | 0.           | MeV  | Upper energy value defining rectangular don't plot area  
| SEXCL    | 0.            | eVs  | Area of matched contour defining a don't-plot area  
| DTHCURV  | 0.0          | deg  | Amount by which contour will be moved in \(\theta\) direction  
| DECURV   | 0.0          | MeV  | Amount by which contour will be moved in \(E\) direction  

\(^a\)For ICONTUR = 0 or 5, the SBCKT and HBCKT values are calculated from the principal rf system only.

\(^b\)Starting values for contours have units degrees,MeV.

\(^c\)THPMIN and THPMAX both 0.0 results in a plotting range \(-180^\circ/\text{FRAC} \leq \theta \leq 180^\circ/\text{FRAC}\).

\(^d\)DEPMIN and DEPMAX both 0.0 results in a plotting range approximately the range of particle energies

\(^e\)Default 0.0 results in autoscaling of minimum of voltage axis

\(^f\)Default 0.0 results in autoscaling of maximum of voltage axis

\(^g\)A particle which ESME tracks as a reference. See \(T\) for optional initial values

\(^h\)Limits of 0.0 for both THBMIN and THBMAX result in the plot range being the same as for the phase space plot.

\(^i\)Limits of 0.0 for both EBMIN and EBMAX result in the plot range being the same as for the phase space plot.

\(^j\)IFBMAX = 0 results in the upper limit being the greatest Fourier harmonic computed.

\(^k\)Limits of 0.0 for both CVBMIN and CVBMAX result in the range for the plot being \(\pm 180^\circ/\text{FRAC}\).
The members of NAMELIST /GRAPH/ are read in SUBROUTINE GRAFSET, and stored in module GRAFIX. The actual graphical output can be generated either during or after processing. The standard distribution of the plotting routines written specifically for ESME v. esmF95 (esme2007.1) use pgplot.[17] A sample post-processor is described later in Appendix A. Users wishing to process ESME data independently or with different graphics routines may find it of some use.

The options enabled by the O command are largely self-explanatory; the default PLTSW settings result in the output of a phase space plot of the distribution and the bucket due to RF source 1, as well as plots of the projections of the distribution along the $\theta$ and E directions. An example of an O command requesting such output every 100 turns could be

```
$GRAPH MPlot=100 THPMIN=-10.0 THPMAX=10.0
  DEPMIN=-25.0 DEPMAX=25.0 $END
```

in which the limits are appropriate ones chosen by the user. In addition to the graphical output generated as a result of the O command, the first and second moments of the distribution are computed and output, as well as a number of other system parameters. The moments included on the plots are derived from the particles which are in the class(es) being plotted and within the plot limits. The moments printed in the standard output are those for the entire distribution. The default limits for many of the plots serve as flags to the plotting routine to choose reasonable limits. Note that the limits of the phase space projection plots (THBMIN, EBMIN, etc.) are necessary only if they are different from the phase space plot limits. The parameter DELCON is included to allow the user to either determine the separatrix arbitrarily closely or to save processing time, since in certain situations the routine which determines the bucket in ESME is required to perform many iterations of the difference equations in order to determine the separatrix to the specified accuracy. In those instances in which the contour-drawing routine is unacceptably slow (or unable) to find a contour, it may be useful to set KNTLIM to some lower number. The default values (XCRNR, YCRNR, XAXISL, and YAXISL) give a satisfactory result for landscape orientation on an 8.5 \times 11 sheet, but may need to be adjusted for other devices.

The PLTSW(14) option plots a curve calculated by a shazam routine on the phase plane plot where the rf waveform would be plotted if PLTSW(5) were set .TRUE.. The PLTSW(32) option is operative only if PLTSW(5)=.TRUE.; if it is set .TRUE., any collective voltage is summed with the rf waveform for the plot.
2.2.5  **T Command - Track distribution**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSTOP(^{\dagger})</td>
<td>0.0</td>
<td>s</td>
<td>Time at which to stop tracking</td>
</tr>
<tr>
<td>TTRACK(^{\dagger})</td>
<td>0.0</td>
<td>s</td>
<td>Duration of time to track</td>
</tr>
<tr>
<td>RSCALE0</td>
<td>1.0</td>
<td>-</td>
<td>Sets time scaling if AUTOSCL=F; multiplies range of scaling for AUTOSCL=T</td>
</tr>
<tr>
<td>AUTOSCL</td>
<td>.FALSE.</td>
<td></td>
<td>Switch to activate auto scaling of time step</td>
</tr>
<tr>
<td>RAILU</td>
<td>2.0</td>
<td></td>
<td>Upper limit for RSCALE</td>
</tr>
<tr>
<td>RAILL</td>
<td>0.5</td>
<td></td>
<td>Lower limit for RSSCALE</td>
</tr>
<tr>
<td>GNUSCAL</td>
<td>0.01</td>
<td></td>
<td>Proportionality factor between RSSCALE and GNUS(^{-1}) in auto-scaling</td>
</tr>
<tr>
<td>LGRTHM</td>
<td>1</td>
<td></td>
<td>Select difference equations used in tracking</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1–Complete kinematics, expand path length to maximum order using input</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>coefficients ALPHAn(^{\dagger})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2–Use the simplified difference equation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(\theta_{i,n} = \frac{\tau_{s,n} \theta_{i,n-1}}{\tau_{s,n-1}} + 2\pi \eta \tau_{p} )</td>
</tr>
<tr>
<td>ITRAP(1:4)</td>
<td>0</td>
<td></td>
<td>Indicates a condition for which tracking should be interrupted before time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>indicated by TTRACK or TSTOP:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0–No trap</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1–Trap on minimum bunch width</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2–Trap on minimum bunch height</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3–Trap for (\eta = \text{ETATRP})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4–Trap for (</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5–Trap for (\eta &gt; 0) (transition crossing)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10 – 19–Call SHAZAM, SHAZAM1, SHAZAM2, …, SHAZAM9 following every iteration of the difference equations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-1 – -10–Call SHAZAM, SHAZAM1, SHAZAM2, …, SHAZAM9 before every iteration of the difference equations</td>
</tr>
<tr>
<td>ETATRP</td>
<td>.001</td>
<td></td>
<td>For ITRAP = 3; tracking stopped when (\eta = \text{ETATRP})</td>
</tr>
<tr>
<td>PHISTRP</td>
<td>.95</td>
<td></td>
<td>For ITRAP = 4; tracking stopped when (</td>
</tr>
<tr>
<td>MGRACE</td>
<td>0</td>
<td></td>
<td>Allow a “grace period”(^{\dagger}) of MGRACE turns before trapping</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>conditions are checked</td>
</tr>
<tr>
<td>HISTRY</td>
<td>F</td>
<td></td>
<td>Write history records to logical unit 9 during tracking(^{\dagger})</td>
</tr>
<tr>
<td>HISTSIZ</td>
<td>(10^4)</td>
<td></td>
<td>Approximate number of history records for one execution of T</td>
</tr>
<tr>
<td>BBDRY</td>
<td>F</td>
<td></td>
<td>Remove particles tracked outside of region (-180^\circ/\text{FRAC} \leq \theta \leq 180^\circ/\text{FRAC})</td>
</tr>
<tr>
<td>THREF0(^{\dagger})</td>
<td>0.0</td>
<td>degree</td>
<td>Choose special azimuth for reference particle</td>
</tr>
<tr>
<td>EREF0</td>
<td>0.0</td>
<td>MeV</td>
<td>Choose special energy difference from (E_s) for reference particle</td>
</tr>
</tbody>
</table>

\(^{\dagger}\) TSTOP is set to 0.0 when tracking is completed, or interrupted by an ITRAP option.

\(^{\ddagger}\) TSTOP takes precedence; if TSTOP=0.0, then TTRACK determines duration of tracking.

\(\dagger\) See R command

\(^{\ddagger}\) See Section 3.2

\(^{\dagger}\) Default value for THREF0 or EREF0 give synchronous values; THREF0 = -EREF0 gives THBAR and EBAR at each turn.

The members of NAMELIST /CYCLE/ are read in SUBROUTINE CYCPROG and stored in module CYCLP.

The options in the T command control SUBROUTINE CYCPROG, which iterates the difference equations for the particles. For the default RSCALE0 = 1.0, an iteration corresponds to one beam turn. A typical T command line might be
in which tracking with accumulation of a history file is specified for .001 seconds of simulated time using the simplified form of the difference equations. Note that either TSTOP or TTRACK may be used to specify the duration of tracking, though TSTOP takes precedence. Also, since an interruption in tracking by an ITRAP option results in TSTOP being set equal to 0.0, the stop time must be re-specified in a subsequent T command. Any condition which halts or interrupts tracking is checked at most once per iteration, so tracking duration may be slightly longer than specified by TSTOP or TTRACK. Four ITRAP variables are provided to allow for multiple traps and/or calls to subroutines SHAZAM1 . . . SHAZAM9 during tracking. If $10 \leq ITRAP \leq 19$, then a call is made to the corresponding SHAZAM following every turn. Similarly, for $-19 \leq ITRAP \leq -10$, a SHAZAM call is made at the beginning of an iteration, immediately preceding the loop over individual macroparticles. Tracking may be interrupted by these calls as an option; to trigger the interruption the SHAZAM routine returns a .TRUE. value in its TOTRAP argument. The ITRAP values $1 – 7$ may also be of either positive or negative sign with the effect that the trap condition is checked respectively after or before the iteration of the map.

Calculation time can be reduced dramatically for large scale multiparticle models under many circumstances by scaling the time step.[1] When RSCALE is an integer, it gives the number of turns per time step; when it is the reciprocal of an integer, the integer is the number of iterations per turn. However, the scaling functions correctly for any positive real RSCALE. The effect is a factor of $\text{RSCALE}^{-4}$ in time used for a rather wide class of problems. See the reference cited for the concept and limitations. The amount of scaling can be controlled either by giving a non-unity value to the input parameter RSCALE0 or by using an autoscaling feature (AUTOSCL=.TRUE.) which adjusts RSCALE to be proportional to GNUS. Additional optional parameters can be used to set the upper and lower limits for RSCALE and the proportionality constant. RSCALE0 is an overall multiplier when autoscaling is used. RSCALE is a positive real number. When it is less than one, it is helpful to think of its reciprocal as iterations of the difference equations per turn. When RSCALE is greater than one, it is conveniently thought of as the turns per iteration. If nothing is done but setting RSCALE, the calculation time is proportional to $1/\text{RSCALE}$. Further gains for RSCALE greater than one require reducing the number of macroparticles, a subject which is treated in the reference.

RSCALE is also important in another context; when RSCALE $< 1$, it reduces the time step between mappings. In cases where the ring is loaded with many bunches, for example, the proper mapping for earlier bunches ($\vartheta < 0.$) and later bunches ($\vartheta > 0.$) may differ by a practically significant amount. The program employs the same mapping parameters on a given turn for all bunches, so the way to avoid inaccurate mapping of the outlying bunches is to scale to a smaller time step. As RSCALE is reduced, the calculating time increases approximately in inverse proportion and the solution tends toward that given by the differential EOM with the same integration time step. Another appropriate application of RSCALE $< 0$ is to approximate the distribution of rf around the ring. If the cavities are separated by approximately one quarter of the circumference, RSCALE=.25 will result in a more accurate representation of the dynamics. When $\nu_s$ is small, the refinement is not useful.

There are three parameters which affect how much of the azimuth is actually mapped and what happens if a macroparticle tries to leave the mapped region. If FRAC in the R command is greater than 1, the map is applied to $1/\text{FRAC}$ of the ring centered on $\vartheta = 0^\circ$. The boundary (including the boundary at $\pm 180^\circ$ for FRAC=1) is normally treated as periodic so that a macroparticle exiting on one side reappears on the other boundary. This nicely mimics a full ring of identical bunches by tracking a single bunch when FRAC is taken equal to the harmonic number H. Another choice is to dispose of macroparticles reaching the boundaries; this option is selected by setting BBDRY to .TRUE.

The reference particle, which is tracked along with the distribution, is by default the particle at the initial synchronous point. Another phase space point can be followed by explicitly giving either or both THREF0 and EREF0. Setting THREF0 = -EREF0 results in THREF and EREF equal to THBAR and EBAR each turn.
2.2.6 Q Command - Quit

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

2.2.7 D Command - Display

The D command directs ESME to generate graphical output at the point at which the command is issued. The form of the output is specified by the most recent O command. The D command is useful for generating output at a particular point in a calculation, since the O command itself only provides output every MPLOT turns.

2.2.8 W Command - Write comment

The W command simply directs the program to echo the characters following W (on the same line and after four spaces) to the standard output. It is intended to provide the user with the ability to insert comments into an input dataset which appear as comments in the output file also.
### 2.2.9 H Command - History output

**H Command, Namelist /HISTORY/**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPLT(1:2,1:50)</td>
<td>0(^a)</td>
<td>-</td>
<td>Index of element in history records; NPLT(1,1) is independent variable, NPLT(2,1) is dependent variable.</td>
</tr>
</tbody>
</table>

**Real records:**

1. Time
2. DP0DT, the rate of momentum change proportional to \( \dot{B} \)
3. PDOT, \( dp_s/dt \)
4. THBAR, the mean value of \( \theta \) for the distribution
5. EBAR, the average energy of the distribution
6. THRMS, the rms spread in \( \theta \) of the particles
7. ERMS, the rms energy spread
8. ES, the synchronous energy
9. E0, the energy on the reference orbit
10. ES-E0
11. THREF, the azimuth of a particle tracked from (0,ES)
12. EREF, the energy of a particle tracked from (0,ES)
13. EPSILON, the emittance\(^b\)
14. NUS, the synchrotron tune
15. SBCKT, the RF “bucket” area for principal system \(^c\)
16. HBCKT, the RF “bucket” height for principal system
17. ETA, \( \gamma^2 - \frac{\gamma^2 - 2}{\gamma^2} \)
18. RSCALE; see T command
19. TAU, synchronous revolution period
20. PSIADD, phase feedback; see L command
21. DAMPL, voltage feedback factor; see L command
22. DELR, synchronous orbit radius - reference orbit radius
23. VPKFD, peak voltage from collective potential in frequency domain
24. VPKTD, peak voltage from collective potential in time domain
25. VPKHQ, peak voltage from resonances in time domain
26. BNCHFCT, bunching factor \( \langle I_b \rangle/\bar{I}_b \)
27. DELEIMP, energy loss per turn to real part of longitudinal impedance
28. PDOT-DP0DT
29. DPOP, fractional momentum spread of distribution

**Integer records:**

31. M, iteration number
32. Number of macroparticles remaining in the distribution
33. H(MOST), harmonic number of principal rf system
34. KNTSC, number of macroparticles that contribute to the collective potential
35. NRF, the number of rf systems being used
36. NNF, the number of Fourier amplitudes saved for history
37. KWEERKNT, number of macroparticles lost

---

\(^a\)The default value of 0 indicates to SUBROUTINE HISTORY that all of the desired history plots have been generated, so only the first set of consecutive nonzero entries to array NPLT or NWRT will select data.

\(^b\)EPSILON = ANORM \( \tau \left( \sqrt{\sum (\theta_i - \bar{\theta})^2 \sum (E_i - \bar{E})^2 - \sum (\theta_i - \bar{\theta})(E_i - \bar{E})^2} \right) / \sqrt{N} \) eVs; see text for ANORM.

\(^c\)The principal system is the one which gives the greatest bucket height.
### H Command, Namelist /HISTORY/, continued

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array records:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51-100 – SPARE(1-50)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101-110 – EV(1-10) [MV]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111-120 – PSI(1-10) [deg]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>121-130 – FREQ(1-10), frequency of rf systems [MHz]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201-250 – FAMPL(1-50), selected Fourier amplitudes; see F command</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>251-300 – FAZE(1-50), selected Fourier phases; see F command</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NWRT(1:2,1:50)</td>
<td>0⁰</td>
<td>-</td>
<td>Indices of elements in history records; works like NPLT but writes records to FORTRAN UNIT 19</td>
</tr>
<tr>
<td>XCRNR</td>
<td>135.</td>
<td>0.001</td>
<td>Fraction of full width of plot frame between lefthand edge and left side of plot</td>
</tr>
<tr>
<td>YCRNR</td>
<td>69.0</td>
<td>0.001</td>
<td>Fraction of full height of plot frame between bottom edge and bottom of plot</td>
</tr>
<tr>
<td>XAXISL</td>
<td>750.</td>
<td>0.001</td>
<td>Plot width as a fraction of frame width</td>
</tr>
<tr>
<td>YAXISL</td>
<td>475.</td>
<td>0.001</td>
<td>Plot height as a fraction frame height</td>
</tr>
</tbody>
</table>

The members of NAMELIST /HISTORY/ are read in subroutine HISTORY. The emittance EPSILON logged by the H command is the rms area in eVs. There is a switch RENORM for the P command which controls whether a factor ANORM is included in the emittance. If RENORM is .TRUE., ANORM is included. It is calculated by the P command to scale the rms emittance of the initial bunch to the area SBNCH of the curve used to generate it. If the initial distribution is not a bunch, setting RENORM has no effect on the calculated emittance. Such scaling is not a good idea in general because the factor is calculated only for the initial distribution and may become misleading as the distribution evolves.

The output produced by the H command, as implied by the simplicity of the /HISTORY/ NAMELIST, is not as flexible as the output produced by the O command. The user specifies only pairs of values to be plotted using the indices above in array NPLT, and the program produces a history plot for each pair of indices assembled sequentially from the entire history record (*i.e.*, there is no choice of a range for any axis). For example, suppose that a simulation has run for 10,000 turns, and the desired output is a record of the distribution moments vs. time over that period. Then a suitable H command is

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

which will generate plots of THBAR, EBAR, THRMS, and ERMS vs. time, as well as a plot of EBAR vs. THBAR. The fortran array index convention is exploited to avoid explicit reference to the indices, the recommended manner of input.

A very helpful use of H is to keep track of debugging output or the results of SHAZAM’s. Either in a routine to be debugged or in a SHAZAM, quantities of interest can be entered into SPARES(1 : 50) to be plotted with NPLT = 51 : 100. The unit of SPARE(n) can be given as a string of 7 characters or less in SPUNIT(n), and likewise a label for the plot of up to 14 characters can be given in SPLABL(n).

Because the frequency of write operations to the history tape may be modified by the program and because of a culling procedure applied to the data points by the plotting routine, not every turn will be included in the plot for very long simulations. The frequency of writing records to file is controlled by the HISTSIZ parameter (default $10^4$) of the T command. With the default value, the tracking routine will write a record every turn if there are fewer than $10^4$ iterations between the time limits given in the T command. Because there can be a few hundred four byte words per record, this file can easily be a few MB per T command. Whether HISTORY plots every record depends on the space available for the plot. This can be increased by the KMAXPTS parameter of the Y command. For more elaborate plots or, perhaps, for analysis, anything which can be plotted according to NPLT values can be written to logical unit 19 using NWRT. Another handy application of unit 19 files is as a source of input curves for related
calculations.
2.2.10  M & N Commands - Save mountain range data & Plot mountain range

<table>
<thead>
<tr>
<th>M Command, Namelist /MRANGE/</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable</strong></td>
</tr>
<tr>
<td>IPU</td>
</tr>
<tr>
<td>FASPEC</td>
</tr>
<tr>
<td>FPSPEC</td>
</tr>
<tr>
<td>EVTDSPEC</td>
</tr>
<tr>
<td>EVRESPEC</td>
</tr>
<tr>
<td>EVFDSPEC</td>
</tr>
<tr>
<td>TMBEGIN</td>
</tr>
<tr>
<td>TMEND</td>
</tr>
<tr>
<td>MRMPLOT</td>
</tr>
<tr>
<td>MRNBIN</td>
</tr>
<tr>
<td>MRBMIN</td>
</tr>
<tr>
<td>MRBMAX</td>
</tr>
</tbody>
</table>

$^a$Defaults to $-\frac{180}{\text{FRAC}}$ for current pickup data and to one for Fourier amplitude or power spectrum.

$^b$Defaults to $\frac{180}{\text{FRAC}}$ for current pickup data and NBINFFT/2 for Fourier amplitude or power spectrum.

The members of NAMESPACE /MRANGE/ are read in subroutine MRINIT and stored in the module MTNRANGE in response to the M command. The members of NAMESPACE /MRPLOT/ are read in the subroutine MRPLT in response to the N command and also stored in MTNRANGE.

The M and N commands are intended to provide plots similar to the display provided by an oscilloscope recording successive traces from a beam current pickup, each trace vertically displaced from the previous one. The default display depicts the time evolution of the phase projection of a distribution in a manner which somewhat resembles a mountain range, hence the name. The M command directs the program to save the data, while the N command directs the program to process the data which have been saved in a file and produce mountain range plots. The M-N combination can also be used to plot the evolution of the Fourier amplitude or power spectrum of the beam current. Only one type of plot can be produced at a time, so only one of the switches IPU, FASPEC, FPSPEC, EVTDSPEC, EVRESPEC, EVFDSPEC should be set .TRUE.. The setup of the desired range of Fourier components is handled through either the F or B commands. If there is no need for the Fourier analysis other than the mountain range plot, for example no calculation of the voltage arising from the beam image current or bunch self field, then the value of MFFT in the FFT namelist should be set to the number of turns between MRMPLOT to avoid unnecessary evaluations of the fft. A typical pair of M and N commands might be

```
$\text{MRANGE TMBEGIN}=0.0 \text{ TMEND}=1.0 \text{ MRMPLOT}=1000$ END
```

in which mountain range records are recorded every 1000 turns of tracking from 0.0 to 1.0 seconds, and

```
$\text{MRPLOT SMOOTH}=1$ END
```

which directs that all of the data accumulated thus far be plotted in mountain range format with Bernstein polynomial smoothing. The default for TBASE (F) results in a fixed vertical separation between consecutive traces. If TBASE = T, the vertical separation between consecutive traces will be proportional to the time separating their records, better simulating the mountain ranges normally depicted on an oscilloscope.\(^3\) The values for TOPTOB and/or SCALE may have to be adjusted to achieve a satisfactory effect.

If MRNBIN $\neq$ NRNBIN, the saved data is transformed to the correct number of bins for plotting by cubic spline interpolation. This feature is independent of whether the stored distribution is first smoothed. Either Bernstein polynomial smoothing[18] (SMOOTH = 1) or 1-2-1 averaging of adjacent bins (SMOOTH = -1) may be used. The spline

\(^3\)In relativistic situations, no difference will be discerned between the plots generated with TBASE either T or F.
interpolation can produce some smoothing depending on the particular NRNBIN and MRNBIN values. The defaults for MRNBIN and NRNBIN will usually give very smooth plots with SMOOTH = 1, but the default value of 100 for MRNBIN may be too large for a sparsely populated distribution if SMOOTH = 0. Either type of smoothing may be iterated by raising ITNO. The two techniques are quite different, however, and the meaning of iteration is rather different. The Bernstein polynomial smoothing is a global procedure which seeks to minimize an object function with contributions from both a smoothness measure and a squared fitting error. Iteration attempts repeated minimizations of this object function. The relative weight given to the fit is increased by increasing OBJWGT; the default of three gives considerable weight to fitting. The 1-2-1 smoothing is a local weighted averaging. Iterating it broadens the span of the averaging giving 1-4-6-4-1, 1-6-15-20-15-6-1, . . . smoothing for ITNO = 2,3, . . . respectively. (The relative weights are \( C_i^{2\cdot ITNO} \) for \( i = 1,\ldots, 2 \cdot ITNO \).) Generally some smoothing will be required to make mountain range plots that look like those made from an accelerator beam current pickup using an oscilloscope. Typical choices are SMOOTH=-1,ITNO=2 and SMOOTH=1,ITNO=1.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRPMIN</td>
<td>0.0°</td>
<td>Minimum abscissa value for mountain range plot</td>
</tr>
<tr>
<td>MRPMAX</td>
<td>0.0°</td>
<td>Maximum abscissa value for mountain range plot</td>
</tr>
<tr>
<td>NTRACE</td>
<td>100</td>
<td>Number of traces on page</td>
</tr>
<tr>
<td>NSKIP</td>
<td>0</td>
<td>Number of records to be skipped between each trace</td>
</tr>
<tr>
<td>TOPTOB</td>
<td>0.7</td>
<td>The fraction of the vertical range over which NTRACE traces are to be plotted</td>
</tr>
<tr>
<td>SCALE</td>
<td>0.3</td>
<td>The height of the first trace, in units in which the entire vertical range</td>
</tr>
<tr>
<td>MSTART</td>
<td>0</td>
<td>Iteration number at which to start plots</td>
</tr>
<tr>
<td>MSTOP</td>
<td>0</td>
<td>Turn number at which to stop plots</td>
</tr>
<tr>
<td>TMSTART</td>
<td>0.0</td>
<td>Time at which to start plots</td>
</tr>
<tr>
<td>TMSTOP</td>
<td>0.0</td>
<td>Time at which to stop plots</td>
</tr>
<tr>
<td>TBASE</td>
<td>F</td>
<td>Switch causing plot trace separation to be proportional to time</td>
</tr>
<tr>
<td>NRNBIN</td>
<td>100</td>
<td>Number of points to plot on a trace</td>
</tr>
<tr>
<td>SMOOTH</td>
<td>0</td>
<td>Smoothing option</td>
</tr>
<tr>
<td>OBJWGT</td>
<td>3.0</td>
<td>Weight of fitting term of object function w/ smoothing term for polynomial</td>
</tr>
<tr>
<td>ITNO</td>
<td>1</td>
<td>Number of iterations for either 1-2-1 or Bernstein smoothing</td>
</tr>
<tr>
<td>LIM</td>
<td>F</td>
<td>Switch for plotting dotted lines connecting left-most and rightmost non-zero</td>
</tr>
<tr>
<td>XCRNR</td>
<td>135. 0.001</td>
<td>Fraction of full width of plot frame between lefthand edge and left side of</td>
</tr>
<tr>
<td>YCRNR</td>
<td>69.0 0.001</td>
<td>Fraction of full height of plot frame between bottom edge and bottom of plot</td>
</tr>
<tr>
<td>XAXISL</td>
<td>750. 0.001</td>
<td>Plot width as a fraction of frame width</td>
</tr>
<tr>
<td>YAXISL</td>
<td>475. 0.001</td>
<td>Plot height as a fraction frame height</td>
</tr>
</tbody>
</table>

defaults of 0.0 for MRPMIN and MRPMAX imply data is to be plotted over its entire range; abscissa units are degrees for IPU=T and harmonic number for spectra.

The defaults of 0 for MSTART and MSTOP, or 0.0 for TMSTART and TMSTOP, imply that all mountain range records are to be plotted.

SMOOTH = 1 also required.
### 2.2.11 L Command - Low level feedback parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Default Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHFBON</td>
<td>F</td>
<td>-</td>
<td>Activates phase feedback</td>
</tr>
<tr>
<td>NTUAVG</td>
<td>1</td>
<td>-</td>
<td>The number of past turns to average in computing phase feedback; the default NTUAVG = 1 corresponds to infinite-bandwidth.</td>
</tr>
<tr>
<td>NTURES</td>
<td>1</td>
<td>-</td>
<td>The number of turns for the feedback to respond; the present signal is compared to the signal of NTURES turns ago.</td>
</tr>
<tr>
<td>ITFB</td>
<td>0</td>
<td>-</td>
<td>The form of phase feedback:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 – Critical damping</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 – Critical damping; gain reduced proportionally to $\nu^2$ within non-adiabatic interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 – Fixed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 – Fixed; gain reduced proportionally to $\nu^2$ within non-adiabatic interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 – Fixed; turns to average $5/\nu_n$</td>
</tr>
<tr>
<td>FBFACT</td>
<td>1.0</td>
<td>-</td>
<td>The gain applied to the phase feedback</td>
</tr>
<tr>
<td>USEWT</td>
<td>F</td>
<td>-</td>
<td>Applies weight function $W$ to phase signal over NTUAVG turns</td>
</tr>
<tr>
<td>W(1:NTUAVG)</td>
<td>0.0</td>
<td>-</td>
<td>Weight function multiplying phase signal</td>
</tr>
<tr>
<td>DLIMIT</td>
<td>5.7296</td>
<td>deg</td>
<td>The upper limit on the magnitude of the phase feedback on a given turn</td>
</tr>
<tr>
<td>VFCON</td>
<td>F</td>
<td>-</td>
<td>Activates voltage feedback</td>
</tr>
<tr>
<td>VFBFCTR</td>
<td>1.0</td>
<td>-</td>
<td>The gain applied to the voltage feedback</td>
</tr>
<tr>
<td>VLIMIT</td>
<td>.1</td>
<td>-</td>
<td>Limit on the fraction of voltage feedback (DAMPL)</td>
</tr>
<tr>
<td>FILBWD</td>
<td>'DUMMY'</td>
<td>-</td>
<td>Name of time/bunch length curve for voltage feedback (see text)</td>
</tr>
<tr>
<td>ETAJMP</td>
<td>0.0</td>
<td>-</td>
<td>The value of $\eta$ at which to jump the phase of the RF</td>
</tr>
</tbody>
</table>

The members of NAMELIST /LLRF/ are read in SUBROUTINE LOWLVL and stored in module FEEDS. As in the accelerator itself, one would prefer that some clever person has taken care of the low level system so that the beam behaves in normal circumstances. The ESME defaults serve this function rather well for a considerable range of applications. There are times when it is necessary to take the lid off the black box and deal with grubby details. Most of this sort of thing is handled in the LOWLVL routine.

The phase feedback, intended primarily to damp dipole bunch oscillations, is computed according to the following formula for $\text{ITFB} = 0$:

$$\text{PSIADD} = \frac{\text{FBFACT} \sum_{i=1}^{\text{NTUAVG}} (W_i(\dot{\theta}_{n-i} - \dot{\theta}_{n-\text{NTURES}-i})}{\pi \nu_s \text{NTUAVG}^2 \sum_{i=1}^{\text{NTUAVG}} W_i}$$

where $n$ is the current turn number, and $\nu_s$ is the synchrotron tune. For the option $\text{ITFB} = 1$ the gain is reduced near transition proportionally to $\nu^2$. For $\text{ITFB} = 2$, $\nu_s$ is removed from the denominator of the feedback formula above giving a constant-gain feedback, and FBFFACT should be reduced to some appropriate value like $4.0 \cdot 10^{-3}$ for example. For $\text{ITFB} = 3$ the constant-gain formula is is reduced in the non-adiabatic interval. IFTB=4 results in a constant gain feedback that averages over five(!) synchrotron periods, so it should act as sluggish radial position feedback. (Not well worked out)

A simple invocation of phase feedback would appear as

```sh
$LLRF PHFBON=T $END
```

The defaults imply critical damping. The routine automatically reduces the gain within the non-adiabatic time around transition proportionally to $\nu_s^2$; if the feedback gain is an object of study, this function of the routine LOWLVL can be managed by a user-written SHAZAM.
The voltage feedback, intended to damp quadrupole bunch oscillations, operates according to

\[ V_n' = (1 + (\text{VFBFCTR} \times (\sigma_{\vartheta,n} - \sigma_{\vartheta,n-1} - \varsigma_{\vartheta,n} + \varsigma_{\vartheta,n-1})/\nu_s)) \times V_{n-1} \]

where \( n \) is the current turn number, \( \sigma \) is rms bunch width, \( \varsigma \) is the program value for bunch width, \( \nu_s \) is the synchrotron tune, and \( \text{VFBFCTR} \) is an arbitrary gain. \( \text{VLIMIT} \) is the maximum permitted for the ratio \( V_n'/V_n \). This ratio is the variable \( \text{DAMPL} \) in the module feeds.f, and it may be plotted by using the \textbf{History} command. If \( \text{VFBFCTR} \) is \( < 0 \), the feedback may be used to produce a controlled emittance dilution. If the \( \text{FILBWD} \) variable is not read in to point to a bunch length curve, the \( \varsigma \) values are taken as \( \sigma_{\vartheta,1} \) for all time. This default works fine for a constant fixed bucket or over a period in which there is little secular change in the bunch width. The use of a target curve for the bunch width is a way of avoiding the difficulty of representing the correct bandpass frequency response in a time domain calculation. It is in a certain sense an idealized feedback algorithm, but in a programming sense it is something of a kluge. If one has a curve generating program to produce voltage curves, it can be used to produce the corresponding bunch length curve for a chosen longitudinal emittance. The format for a bunch width curve is a first line containing an integer specifying the number of entries to follow with the following entries consisting of two numbers per line giving a time in seconds and a bunch width at that time. The program normalizes the curve so that the first time slot has the value of \( \sigma_{\vartheta} \) at the time of the \textbf{L} command. Obviously if the curve is not read in at the time corresponding to the initial point in the table, this could cause a scale error. The default feedback gain is arbitrary; it may be necessary to adjust \( \text{VFBFCTR} \) to obtain satisfactory results. All of this may seem a little awkward and \textit{ad hoc}, but it really works very well for reducing shape oscillations after transition, for example.

The phase feedback gain is proportional to the input parameter \( \text{FBFACT} \), which defaults to one. The feedback gain is then at a value which gives critical damping according to the linear differential EOM’s. However, optimum damping for finite emittance and nonlinear potential may be obtained with \( \text{FBFACT} \) as large as five. The voltage feedback gain is rather arbitrary, and optimum damping of shape oscillation may require \( \text{VFBFCTR} \) of forty or fifty.

In ESME2000 the \textbf{L} command contained beam loading compensation features. These have been removed because of a near impossibility in attaining the kind of generality intended for ESME. The voltages developed by the beam in one or more resonators can be studied using the facilities of the \textbf{B} command; feedback or feedforward compensation can be applied through one of the shazam routines (commands \textbf{0} - \textbf{9}). Those wishing to recover the deleted features or develop similar code may want to look at the example of SHAZAM code.
### 2.2.12 **B Command - Beam-derived potential (self-force and $Z_{||}$)**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.005</td>
<td>m</td>
<td>Effective beam radius</td>
</tr>
<tr>
<td>B</td>
<td>0.05</td>
<td>m</td>
<td>Effective beampipe radius</td>
</tr>
<tr>
<td>ENQ</td>
<td>$2 \cdot 10^{10}$</td>
<td></td>
<td>Number of ions (of $Z=\text{CHGNO}$, see R command) to be represented by the distribution</td>
</tr>
<tr>
<td>NBINSC</td>
<td>1024</td>
<td></td>
<td>Number of bins for histogram of charge distribution</td>
</tr>
<tr>
<td>MSC</td>
<td>1</td>
<td></td>
<td>Collective effects are to be calculated MSC times between rf cavities$^a$</td>
</tr>
<tr>
<td>TCHGON</td>
<td>0, s</td>
<td></td>
<td>$&gt; 0$: Time starting from 0 in which charge is ramped from 0 to ENQ $&lt; 0$: Time starting from 0 in which charge is ramped from ENQ to 0</td>
</tr>
<tr>
<td>SCON</td>
<td>F</td>
<td></td>
<td>Activate time domain perf. cond. wall calculation</td>
</tr>
<tr>
<td>TDON</td>
<td>F</td>
<td></td>
<td>Activate time domain wake field calculation with supplied response function$^b$</td>
</tr>
<tr>
<td>FDON</td>
<td>F</td>
<td></td>
<td>Activate frequency domain wall impedance calculation</td>
</tr>
<tr>
<td>FDSCON</td>
<td>F</td>
<td></td>
<td>Activate frequency domain calculation of perf. cond. wall voltage. Requires FDON = T also.</td>
</tr>
<tr>
<td>QREZON</td>
<td>F</td>
<td></td>
<td>Activate time domain calculation for high-Q resonance</td>
</tr>
<tr>
<td>NBINFFT</td>
<td>256</td>
<td></td>
<td>Number of bins to be used in Fourier transform</td>
</tr>
<tr>
<td>MFFT</td>
<td>1</td>
<td></td>
<td>Interval (in turns) between Fourier transforms</td>
</tr>
<tr>
<td>NNF$^c$</td>
<td>0</td>
<td></td>
<td>Number of Fourier harmonics to be stored in history</td>
</tr>
<tr>
<td>NF(1:NNF)$^c$</td>
<td>0</td>
<td></td>
<td>Harmonic numbers of Fourier spectrum components to be stored</td>
</tr>
<tr>
<td>NBRES</td>
<td>10000</td>
<td></td>
<td>Number of time slices for time domain solution of high-Q resonator</td>
</tr>
<tr>
<td>NIXNOIS</td>
<td>0</td>
<td></td>
<td>Three-way switch to control smoothing of charge distribution $-1 =&gt; 1$-2-1 averaging of adjacent bins $0 =&gt;$ no smoothing measures $1 =&gt;$ Bernstein polynomial smoothing</td>
</tr>
<tr>
<td>ITKNT</td>
<td>1</td>
<td></td>
<td>Number of iterations for either 1-2-1 or Bernstein smoothing</td>
</tr>
<tr>
<td>OBWT</td>
<td>3.0</td>
<td></td>
<td>Weight of fitting vs. smoothing in object function for Bernstein smoothing</td>
</tr>
<tr>
<td>FILIMP</td>
<td>'DUMMY'</td>
<td></td>
<td>Full path for file containing the impedance table</td>
</tr>
<tr>
<td>FILRES</td>
<td>'DUMMY'</td>
<td></td>
<td>Full path for file containing the resonance list</td>
</tr>
<tr>
<td>FILTDB</td>
<td>'DUMMY'</td>
<td></td>
<td>Full path for file containing time domain response basis</td>
</tr>
</tbody>
</table>

$^a$The number of such calculations per turn will be MSC/RSCALE; RSCALE0 is a parameter of the T command.

$^b$see Appendix 3.2.6

$^c$For instruction in the use of NF(1:NNF), see the description following the F command.

The members of NAMELIST /SCHG/ are read in subroutine BEAMSC and stored in the module SPCHARGE. The B command controls facilities in ESME for modeling the interactions of the beam particles with each other through both the direct particle-particle force and through wake fields excited as a consequence of the interaction of the beam with its environment (vacuum chamber, rf cavities, etc . . .). Both types of collective voltage can be calculated in either time domain or frequency domain. For the perfectly conducting wall term, either is convenient and the difference is whether the gradient of the charge distribution is calculated from a local derivative or a Fourier fit to the entire distribution. If frequency domain is used for the wall external impedance, it is much faster to use the fft machinery for the perfectly conducting wall also; set FDON=T for the impedance calculation and FDSCON for the perfectly conducting wall. If a response to a unit charge triangular pulse is available, one can use TDON=T; the time domain perfectly conducting wall calculation is selected by SCON=T. One difference between the perfectly conducting wall term in time and frequency domain is that the geometric factor $g$ is rolled off for wavelengths comparable and shorter than the beam pipe radius in the frequency domain calculations. This refinement is not
easily introduced into the time domain. If the longitudinal impedance is representable by a few resonances, a time domain calculation can be made by setting QREZON=T, which is most accurate for high-Q resonances. Because time domain and frequency domain can be used together, low-Q resonances can be treated in frequency domain. Since the transients from low-Q impedances decay rather quickly, the impedance approach is likely to be appropriate for them.

Please note that when the longitudinal impedance contains a real part, i.e., for any physical wake field, the synchronous energy gain per turn is diminished by the energy loss to the real part. This loss is calculated and is now included in the energy gain calculation (new as of February 2004). The value of the loss (DELEIMP) can be plotted in a history plot with index 27. The loss is calculated regardless of the mode in which the wakefield is represented, that is, which of the options FDON, TDON, QREZON are used. It can be used to modify the synchronous phase calculation according to the choice of the ISYNC parameter in the A command.

Resonance data is read from a file designated by the string FILRES; the format is described Sec. 3.2.5. Special values of the basic parameters frequency, \( R_{\text{shunt}} \), multiplier, and Q are used to control how the effects of the resonance are calculated. In particular, the magnitude of a negative frequency parameter is interpreted as a generalized harmonic number (not necessarily an integer) of a resonance that follows the synchronous circulation frequency. A multiplier of zero flags a resonance that is to be treated in time domain.

When QREZON=T, the voltage due to those resonators for which the multipliers (third parameter in the resonance file record) are zero is calculated in time domain. If there is one physical resonator in the ring, the periodicity in the problem is one. Thus, the FRAC parameter for the R command should be one. If the resonance is associated with the rf cavities, and those are distributed in \( n \) groups (more or less) uniformly around the ring, then FRAC = \( n \) is the proper periodicity. Clearly, for an accelerator with high harmonic number, the calculation may be impractically slow. The parameter NBRES must be large enough to give several bins per bunch width even to get the lowest mode of bunch-to-bunch coupling accurately. However, the time domain machinery is efficient, and it possible to run calculations with large NBRES, like \( 10^5 \) for a somewhat extreme example.

The more general version of the time domain calculation of the voltage arising from the beam image current flowing in \( Z_0 \) requires that the user provide a file containing the values of a basis function at a regular tabular interval. The basis function is the response of the impedance to a unit current moving with the beam velocity distributed in a triangular pulse with base width twice the tabular interval.\(^3\) Except for special cases like a simple resonance, it will be necessary to calculate this response using a time domain EM program like TCBI. The first record in the basis function file FILTDB is the time length of the interval between evaluations. Following are the tabular values of the basis function continuing to times as long as the wakefield is significant, or up to \( \tau/\text{FRAC} \) if that is shorter. This calculation does not share with the high-Q resonator calculation the facility for accumulating excitation over multiple turns. If there is appreciable wake field extending beyond \( \tau/\text{FRAC} \), its effect is lost. However, something with so high a Q is probably correctly represented as a resonance or a few resonances and can be properly treated with the time domain resonance calculation (QREZON = .TRUE.). All three types of collective voltage calculation can be carried on simultaneously for different parts of the impedance.

One of the frustrations in calculating emittance growth caused by the collective potential is spurious emittance growth of an initial bunch matched to a single particle trajectory. It is generally difficult to produce a self-consistent distribution directly. Two ways are provided to avoid the initial match problem. The parameter TCHGON can be set to a value of a few synchrotron periods. The strength of the beam charge will then be ramped linearly to its target ENQ by the time TCHGON, thereby approximating an adiabatic introduction of the perturbation. Another technique which may require less computing is to generate a collective potential using the B command and an approximate initial distribution then to use the K command to remove the distribution before using the P again to provide a bunch matched to the new potential. This cycle could be iterated for an exacting application to obtain an adequate approximation to self-consistency.

Under some conditions, most likely at low energies, the collective potential is very significant, and applying it only once per turn is not realistic because there is too much force-free drift between energy increments. A typical symptom of this problem is spurious bunch breakup. The parameter MSC can be set to apply the collective potential MSC times per iteration. However, the collective potential itself changes very little in one turn, so it is evaluated

\(^3\)This approach is developed from the ideas of G. Sabbi,[16] but is not based on his code TRISIM.
once per iteration independently of MSC.

An example of B command which includes the self-force, resistive wall impedance, and some resonant elements is

```
$SCHG FDON=T FDSCON=T QREZON=T A=.01 B=.03 ENQ=2.4E10 NBINSC=1000 NBINFFT=512
NBRES=1000 FILIMP='IMPEDANCE.DAT' FILRES='RESONANCE.DAT'$END
```

in which the real and imaginary parts of the wall impedance are read from IMPEDANCE.DAT while the resonance values are obtained from the file RESONANCE.DAT. Note that the number ENQ of beam particles specified applies to the number of macroparticles in the distribution, not multiplied by the periodicity parameter FRAC. The impedance values apply to the entire circumference; they are not per unit length. The number of bins used in the FFT will be the lesser of NBINFFT or the storage allocation KFFT. The parameter NBINSC should be chosen so that there are an adequate number of bins over the width of the bunch(es) to provide good shape information. With these switches, the bunch self-field is calculated from the FFT of the charge distribution, the general longitudinal impedance is treated in frequency domain, and the resonant contributions are evaluated in frequency domain or time domain according to the value of the multiplier given for each resonance.

There is no necessary numerical relation between NBINSC and NBINFFT; however taking them equal saves some interpolation. If the two are unequal, spline interpolation on a charge distribution of NBINSC bins is used to make a distribution of NBINFFT bins for the FFT. Two types of smoothing may be applied to the distribution of NBINSC bins. SMOOTH=-1 causes a local averaging with weights 1/4, 1/2, 1/4. If ITKNT is greater than one, the smoothing is repeated, producing (less local) averaging with, for example, weights 1/16, 1/4, 3/8, 1/4, 1/16 for ITKNT=2. If SMOOTH=1, Bernstein polynomial smoothing[18] is employed. This is a global technique in which an object function containing a least squares fitting criterion and a smoothness measure is minimized. For this case ITKNT controls the maximum number of attempts to minimize the object function. This method is limited by a requirement that NBINSC be small enough that each bunch is spanned by fewer than 500 bins (default value of the parameter IFLSIZ) and, if there are multiple bunches, that there be three or more empty bins between them.

Another major hurdle in modeling high intensity beams is dealing with a sufficient number of macroparticles to keep local fluctuations in the space charge force from causing a spurious disruption of the distribution. When only the low frequency effects of the beam charge are of interest, it may be satisfactory to take a limited bandwidth in frequency domain and/or smooth the charge distribution using one of options of the SMOOTH control variable. The validity of these expedients is not always clear. A more rigorous alternative[1] is to set the parameter RSCALE0 of the T command to something larger than one; this option will reduce the number of iterations required by a factor $(RSCALE)^{-1}$ and permit reducing the number of bins by the same factor. For most problems, if the number of bins is so reduced, the number of macroparticles can be reduced by $(RSCALE)^{-3}$. 
2.2.13 F Command - Fourier transform

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFTON</td>
<td>F</td>
<td>Unit</td>
<td>Activate Fourier transform calculation</td>
</tr>
<tr>
<td>FFTOUT</td>
<td>F</td>
<td>Unit</td>
<td>If TRUE, Fourier transform is printed(^a)</td>
</tr>
<tr>
<td>FFTWRT</td>
<td>F</td>
<td>Unit</td>
<td>If TRUE, Fourier transform is written to Fortran UNIT=21 (^b)</td>
</tr>
<tr>
<td>NBINFFT</td>
<td>256</td>
<td>Unit</td>
<td>Number of bins to be used in FFT</td>
</tr>
<tr>
<td>NNF</td>
<td>0</td>
<td>Unit</td>
<td>Number of Fourier harmonics to be stored in history</td>
</tr>
<tr>
<td>NF(1:NNF)</td>
<td>0</td>
<td>Unit</td>
<td>Harmonic numbers of Fourier spectrum components to be stored</td>
</tr>
<tr>
<td>MFFT</td>
<td>1</td>
<td>Unit</td>
<td>Frequency of Fourier transform calculation</td>
</tr>
<tr>
<td>NIXNOIS</td>
<td>0</td>
<td>Unit</td>
<td>Three-way switch to control smoothing of azimuthal histogram</td>
</tr>
<tr>
<td></td>
<td>-1 =&gt; 1-2-1</td>
<td></td>
<td>averaging of adjacent bins</td>
</tr>
<tr>
<td></td>
<td>0 =&gt; no</td>
<td></td>
<td>smoothing measures</td>
</tr>
<tr>
<td></td>
<td>1 =&gt; Bernstein polynomial smoothing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITKNT</td>
<td>1</td>
<td>Unit</td>
<td>Number of iterations for either 1-2-1 or Bernstein smoothing</td>
</tr>
<tr>
<td>OBWT</td>
<td>3.0</td>
<td>Unit</td>
<td>Weight of fitting vs. smoothing in object function for Bernstein smoothing</td>
</tr>
</tbody>
</table>

\(^a\)Careful! If MFFT is small, the print volume will be large.

\(^b\)This parameter can result in excessively large file if MFFT is too small.

The members of NAMELIST /FFT/ are read in subroutine FFTSET and stored in module FOURIR. Many of the variables in NAMELIST /FFT/ are shared with NAMELIST /SCHG/, because the Fourier transform may be utilized in the collective potential calculation. The F command is intended to allow the user to examine the Fourier transform of the distribution and to follow the development of selected Fourier components. If, for example, one wants to track the turn-by-turn development of the first five odd Fourier harmonics of the distribution, then the appropriate F command would be

\[\text{FFT FFTON=T NBINFFT=32 NNF=5 NF=1,3,5,7,9 END}\]

If at some point one should wish to plot the record for the amplitude of the third harmonic, then the H command should be issued to retrieve FAMPL(2). \(^4\) If NNF is given a negative value, the power spectrum averaged over the intervals between NF values is stored. There is a limit of 50 on NNF (set by parameter NFMAX). The full array of amplitudes and phases can be written out to the log file or to Fortran UNIT=21. This output can get out of hand if the FFTOUT or FFTWRT parameters are .TRUE. for MFFT too small.

The harmonics of the distribution should be multiplied by the periodicity in determining harmonics of the revolution frequency. If, for example, FRAC = 7, the fifth harmonic of the distribution corresponds to harmonic thirty-five of the beam circulation frequency. It is also possible to plot a time sequence of amplitude or power spectra as described in section 2.2.10.

The histogram can be smoothed before the FFT is performed. NIXNOIS = -1 and ITKNT = 1 results in a straightforward averaging of bins with their immediate neighbors using relative weights of 1-2-1. Increasing ITKNT results in higher order averaging. When NIXNOIS = 1, Bernstein polynomial smoothing is employed.\(^[18]\) The relative importance attached to fitting of the data values and smoothness is governed by OBWT; ITKNT determines how many trials are made to reduce the object function. An ITKNT of zero is valid; it results in a well-smoothed representation of the distribution but may remove detail which is meaningful. There are other remarks on smoothing in secs. 2.2.10 and 2.2.12 where mountain range plots and collective potential calculations are discussed.

\(^4\)See Section 2.2.9, History command.
### 2.2.14 **C Command - Flow contours**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES</td>
<td>2</td>
<td>-</td>
<td>The number of flow lines to be drawn</td>
</tr>
<tr>
<td>STVAL</td>
<td>0.,0.</td>
<td></td>
<td>The ((\vartheta, E)) starting values for LINES different contours</td>
</tr>
<tr>
<td>STATIC</td>
<td>.TRUE.</td>
<td></td>
<td>Contours drawn with parameters fixed at initial values for STATIC = .TRUE.; parameters vary according to their programs for STATIC = .FALSE.</td>
</tr>
<tr>
<td>ACCUMUL8</td>
<td>.FALSE.</td>
<td></td>
<td>Switch to retain earlier contours on the current contour plot</td>
</tr>
<tr>
<td>PLTBKT</td>
<td>.TRUE.</td>
<td></td>
<td>Option switch to include bucket contour on flow line plot</td>
</tr>
<tr>
<td>TSTOP(^b)</td>
<td>0.</td>
<td>s</td>
<td>End of interval over which maps are generated</td>
</tr>
<tr>
<td>TTRACK(^c)</td>
<td>0.</td>
<td>s</td>
<td>Time over which periodic flow maps will be generated</td>
</tr>
<tr>
<td>ACCEL0</td>
<td>1.0</td>
<td></td>
<td>Number of beam turns per step between mappings</td>
</tr>
<tr>
<td>PARTION</td>
<td>T</td>
<td></td>
<td>Switch indicates if points on separate flow lines belong to different partitions of the phase points</td>
</tr>
<tr>
<td>ITRAP(1:4)</td>
<td>1</td>
<td></td>
<td>Flags a condition for which the C command should be interrupted; see T command for values</td>
</tr>
<tr>
<td>ETATRP</td>
<td>.001</td>
<td></td>
<td>Trapping parameter; see T command</td>
</tr>
<tr>
<td>PHISTRP</td>
<td>.95</td>
<td></td>
<td>Trapping parameter; see T command</td>
</tr>
<tr>
<td>MGRACE</td>
<td>0</td>
<td></td>
<td>Trapping parameter; see T command</td>
</tr>
</tbody>
</table>

\(^a\)Starting values for contours have units degrees,MeV.
\(^b\)TSTOP set to 0. when calculation complete or interrupted by an ITRAP option.
\(^c\)TSTOP takes precedence; if TSTOP=0., TTRACK determines duration.

The **C** command is used to generate flow line maps at the plotting intervals MPLOT, which is set by the **O** command along with the other parameters controlling the graphics. It may also be used to establish a (zero-emittance) distribution to be tracked. The subroutine FLOWPRG reads the namelist /FLOW/ and stores some of the parameters in the module FLOWP. However, FLOWPRG is a driver for many ESME subroutines so it distributes data among several modules.

When the **C** command is used to generate an initial distribution for tracking, TTRACK and TSTOP are set to zero so that only one set of points on flow lines is generated. It can be combined with a non-zero emittance distribution (i.e., a bunch) if desired. Such a bunch can be generated before the flow lines and the resulting collective potential will then be taken into account in generating the flow lines if the **B** command has been called first. In any case, a zero-emittance type of distribution is stored in the array PHASE above a partition KNTSC so that only the bunch(es) affect the collective potential when that calculation is active. Furthermore, the flow line points are not used in calculating beam moments and emittance.

The **C** command differs from the **T** command in not tracking any distribution turn-by-turn and not calculating bunch properties or storing quantities for history plots. Because the **C** command works without a bunch-like \((\varepsilon \neq 0)\) distribution, it does not ordinarily incorporate the effect of a collective potential. It is possible to use **C** to generate such flow lines by first populating a bunch (\(P\)) then invoking **B** and **C** in that order. Using the **K** command with KNTSET = -1 will remove the bunch if desired.

When the parameter STATIC is set false, the trajectories from the starting points STVAL(\(\vartheta_i, E_i\)) will be generated over the time indicated by TTRACK or TSTOP. In this case quantities like rf voltage which have a time program will change as the trajectory advances. These dynamic trajectories are not lines of Hamiltonian flow, but they provide an alternative to fixed-time phase space macroparticle distributions or fixed-time contours for visualizing the dynamics.

The static contours produced by the **C** command can also be produced by the ICONTUR = 5 option of the **O** command.
2.2.15 K Command - Kill all or parts of the distribution

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KUT</td>
<td>0</td>
<td>Cut the last KUT particles from the distribution</td>
</tr>
<tr>
<td>KNTSET</td>
<td>0</td>
<td>Reset particle count</td>
</tr>
<tr>
<td></td>
<td>&lt; 0 → KNTSC → 0; removes macroparticles which generate the collective potential</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt; 0 → KOUNT-KWEERKNT → 0; removes entire distribution</td>
<td></td>
</tr>
<tr>
<td>K1</td>
<td>0</td>
<td>Starting index for a partial removal of the distribution</td>
</tr>
<tr>
<td>K2</td>
<td>a</td>
<td>Ending index for partial removal of distribution</td>
</tr>
<tr>
<td>KLASS</td>
<td>0</td>
<td>Selects a partition of the distribution to be removed&lt;sup&gt;6&lt;/sup&gt;</td>
</tr>
<tr>
<td>KICKER</td>
<td>.FALSE.</td>
<td>Kicker is to remove beam between TH1 and TH2</td>
</tr>
<tr>
<td>NOTCHER</td>
<td>.FALSE.</td>
<td>Notcher is to remove beam below TH1 and above TH2</td>
</tr>
<tr>
<td>TH1</td>
<td>0. degree</td>
<td>Lower limit for KICKER or NOTCHER</td>
</tr>
<tr>
<td>TH2</td>
<td>0. degree</td>
<td>Upper limit for KICKER or NOTCHER</td>
</tr>
<tr>
<td>FRACTION</td>
<td>1.0</td>
<td>retains a randomly selected FRACTION of the macroparticles</td>
</tr>
</tbody>
</table>

<sup>6</sup>Defaults to KOUNT-KWEERKNT

The K command calls KARVE which reads the namelist /KUTS/ for integers indicating what part of the current phase space distribution to kill. It is really a pastiche of several related functions which have surfaced at one or another time in SHAZAM routines (see 0 – 9 commands). Only one option can be exercised per call, and parameters are always reset to harmless before the command returns. The counters referred to in the input parameter table above are KOUNT, the number of macroparticles in the initial distribution, KNTSC, the number that participate in the calculation of the collective potential, and KWEERKNT, the number of either sort that have been lost. When a macroparticle is classified as lost, it no longer is tracked or used in the collective potential. However, it still has a storage location. The P command sets KOUNT to KOUNT-KWEERKNT; thus at the time of introducing an additional distribution, the lost macroparticles are lost indeed.

2.2.16 0 – 9 Commands - User-written SHAZAM routines

The SHAZAM facility is intended to allow users to integrate their own routines into the code. The numerical commands initiate calls to the subroutines SHAZAM – SHAZAM9. As indicated in section 2.2.5, SHAZAM’s may also be called before or following every iteration of the difference equations using the TRAP option provided through NAMELIST /CYCLE/. The arguments of the SHAZAM routines are the logicals FROMTRAP and TOTRAP. FROMTRAP is .FALSE. when the subroutine is called by command and .TRUE. when it is called from the tracking routine. When a SHAZAM routine has been called from the tracking routine, it may initiate an interruption of the tracking by returning its TOTRAP argument as .TRUE.. All modules in ESME are available to SHAZAM’s. Any input is up to the author of the SHAZAM routine. It may be NAMELIST directed, as ESME largely is, or there may be no input at all. The user is cautioned to ensure that any input data is read; otherwise ESME will read the line where it finds itself after returning from the SHAZAM and attempt to interpret it as a command. A sample of SHAZAM coding is given in Chapter 4.
2.2.17 **Y Command - Memory allocation for**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNPHASE</td>
<td>25000</td>
<td>-</td>
<td>The storage allocation for each phase coordinate — max. value for KOUNT</td>
</tr>
<tr>
<td>KMAXCVB</td>
<td>10000</td>
<td>-</td>
<td>The storage allocation for large tables of data or intermediate results in time domain collective voltage calculations ₋</td>
</tr>
<tr>
<td>KIFFT</td>
<td>IFFT=1024</td>
<td>-</td>
<td>Allocates memory for FFT’s of length KIFFT</td>
</tr>
<tr>
<td>KLTABL</td>
<td>LTABL</td>
<td>-</td>
<td>Sets the maximum length of several parameter and auxiliary variable tables; others fixed at LTABL</td>
</tr>
<tr>
<td>KNSRC</td>
<td>NSRC=10</td>
<td>-</td>
<td>Limits the number of auxiliary tables related to rf curves</td>
</tr>
</tbody>
</table>

* KMAXCVB sets the maximum allowable value for input parameters NBINSC and NBRES.

The **Y** command allows one to allocate memory for arrays whose sizes vary over a wide range depending on the problem. There are moderate default levels (as indicated in the table above), so it may not be necessary to allocate for a problem with a few thousand macroparticles. The **Y** command calls the subroutine MEMALLOC which reads the namelist /MEMORY/ for integers which set the size of phase space coordinate storage, the plot storage for histories, and beam induced voltage distributions, etc. Please be aware that many tables are not dynamically allocated. If a table length is set by LTABL it can be changed only in the module parameters.f. Then the program must be recompiled. The dynamically allocatable tables will default to length LTABL, currently 750, but their length may be specified at execution time by KLTABL.

2.2.18 **S Command - Save tracking parameters**

The **S** command directs subroutine SAVE to write all global variables to an external file. It allows the user to suspend tracking at any point in the program. Once all the tracking data is SAVE’d, it may be restored simply by issuing the **G** command (see next section). Subroutine SAVE is also useful for those who might want to analyze ESME tracking data independently of the program.5 The format of the **S** command is as follows:

```
S filename
```

where *filename* is the name of the external file to which the data is to be written.

2.2.19 **G Command - Get tracking parameters**

The **G** command directs subroutine GET to read data from an external file into ESME’s global data. It is intended to be used to retrieve data written using the **S** command. The **G** command format is very similar to that for the **S** command:

```
G filename
```

where *filename* is the name of the external file from which the data is to be read. Because **G** recovers all parameters, input data for a new run could consist of only **G**, **T**, and **Q** commands for example. It is useful for restarting a long calculation from a checkpoint or for making alternative conclusions to a common intermediate result. The **S** - **G** combination does not give true checkpointing because local counters, etc. are not saved. Thus, restarts may require preparatory **R** or **A**, for example, to establish starting conditions.

---

5See, for example, Appendix A.
The contents of the saved file are specific to a particular version of the program. A G command is likely provide spurious data if it is accessing a save file written by an earlier program version.
Chapter 3

Using The Program

3.1 Running the Program

Although the graphics can display to the screen during execution, ESME is not usually run as an interactive program. It reads data from standard input and writes numerical results and error messages to standard output. Errors detected by the system or arithmetical library go to the standard error file. However, with command line switches input may be taken from an alternate file and pauses set after each display. Additional output files are used with names containing the fortran unit number as an extension. The names for auxiliary input files are read in with the data for the commands that call for the auxiliary input. The fortran units used are:

- unit 07 – input to restore previous running condition; input to GET
- unit 08 – SAVE output for check pointing or interrupted calculation
- unit 09 – history data (output)
- unit 10 – RF voltage, frequency, and phase curves
- unit 11 – longitudinal impedance table
- unit 12 – table of resonances for longitudinal impedance calculation
- unit 13 – magnet ramp table
- unit 14 – graphical output in postscript — esme.ps
- unit 15 – bunch length table for voltage feedback control
- unit 16 – basis function for time domain collective voltage calculation
- unit 18 – output for graphics post-processor
- unit 19 – tabular output from HISTORY (user plots and analysis)
- unit 20 – mountain range plot output data
- unit 21 – optional Fourier spectra at times given MFFT
- unit 23 – optional debugging output

ESME can be run directly from a command line under UNIX-like systems:

```
esmF95 [optional switches] <input 1 > output 2 > errout
```

The available command line switches and their functions are:

```
USAGE: esme [args [parameters]]
-----
where [args] is one or more of the following

-A : write intermediate files in ASCII
-a4 : output on a4 paper
-bw : turn color output off
```
-d driver : select the device driver for PGPLOT output
-D lvl : debugging message level
-f filename : use filename for input instead of standard input
-h : help
-i : interactive output (pauses between plots)
-o filename : use filename for output instead of standard output
-p filename : use filename for postscript graphics instead of esme.ps
-V : version number

If the -f switch is used to designate the input data file, files generated by the program will be named with whatever precedes a “.” (if any) in the input filename. Otherwise the produced filenames will be “UNDEFINED.un”, where “un” is the fortran unit number. The exception is unit 23 which has the fortran default fort.23 and is always character regardless of the -A switch.

It is possible to run ESME interactively with the commands and namelists entered by the user directly from the keyboard. One can also specify input and output files from the command line and request pauses between plots to the screen, called interactive output mode.

If the post-processor option\(^1\) is in effect, then ESME writes its data to the file associated with FORTRAN logical unit 18 (every MPLOT turns, or upon issuance of the D command). History records are written to the file associated with unit logical unit 9. These files may be processed later according to methods of the user’s choosing. However, it is necessary to use the command line argument -A to preserve the intermediate files. Otherwise the files for logical units 9, 18, and 20 are written to the /tmp file system and will be unavailable after ESME stops. A simple post-processor has been written which processes these files using routines from ESME. It is described in Appendix A.

### 3.2 Input Structure

In this section the format required of various input files is described. First and foremost is FORTRAN logical unit 5 (*i.e.* standard input), which is the main input file for ESME and often the only input file. In addition, ESME may read tables of values for the RF voltage, frequency, or phase as functions of time, the magnet ramp, the wall impedance as a function of frequency, a table of resonant impedances, a table of nominal bunch width vs. time for controlling feedback to rf amplitude, and a tabulation of a basis function for time domain calculations. These auxiliary files are read fully, so the tables can cover any time during the calculation. In each case the name of the auxiliary file (complete path) is passed via a namelist character variable FILxxx. New tables can be introduced by naming a new file. Some tables may be of arbitrary length; the memory is allocated dynamically as needed with upper limit KLTABL (default 750 set by parameter LTABLE).

#### 3.2.1 Command file

The commands which ESME accepts have already been described in Chapter Two. Each command is indicated by a single uppercase letter or a numeral in the first column of a line. The following four places in the line are discarded, and the rest of the line is read\(^2\) and echoed in the output along with the command as a comment. Following many of the commands is namelist data. The namelist input is specified by entering an ampersand (&) in the second column of the input file, followed immediately by the namelist identifier, after which any variable which is a member of the namelist may be entered by specifying its name followed by an equal sign followed by the value to be assigned. The assignment of logical variables may depend on the particular machine (e.g. .TRUE., .T., or T). The namelist is terminated by a /END (or simply /). All of the command examples given previously are valid f77 namelist entries using a widely recognized non-standard $ and $END syntax in place of the more widely accepted & /END convention.

---

\(^1\)POSTP=T in the O command.
\(^2\)FORTRAN format (A1,4X,A74).
3.2.2 RF curves

Each of the NRF rf systems can have voltage, phase, frequency, and radial offset (energy offset DELTRF) programs. If phase and frequency or radial offset programs are provided, the frequency or offset program will increment the values set by the phase program. Generally, if both frequency and offset curves are used, they apply to different rf systems. There are options for simple programs provided with the A command (see 2.2.2), but if these are not sufficient, the desired curves can be provided in tables read from an external file identified by the A parameter string FILCRV. The curves file will be read if FILCRV is given. If other curve tables will be used in a later A command, they can be included in the first reading of the file. This subsection describes the content and format of the curves file.

The numerical tables representing the curves are identified by character string headers which are explained later. They may be of any length up to LTABLE (default 750) without further action. Should the default length be insufficient, an arbitrary length KLTABL may be set in the Y (memory) command. All input records are free-format (list-directed). The first numerical record for each table is an integer giving the number of records to follow. The table entries follow with a time and a curve value per record. Just as for the simple curve options available from the program, the scale of the programmed quantities are set by the A command parameters VI(i), VF(i), PSII(i), PSIF(i), FRI(i), FRF(i), DELTRFI(i), DELTRFF(i). This means it may not be necessary to re-compute a complete curve to change the starting point or overall magnitude. Likewise the time base starts at the initial value TxBEG(i); thus one curve can be used at different times in the cycle by changing TxBEG(i) only. All curves, ramp or rf, internally generated or provided in a table, are normalized in the same way:

\[ f_i(t_n) = \frac{[T(t_n) - T(t_o)]}{[T(t_{\text{max}}) - T(t_o)]}, \]

where the \( T \)'s are table values, \( t_i \) are the tabulated times, \( t_o \) is initial time, and \( t_{\text{max}} \) is the time at which the tabulated function is maximum. In the built in functions \( t_{\text{max}} \) is the same as the parameter TxEND(i), but this limitation does not apply to tables from FILCRV. In built in functions \( t_o \) is TxBEG(i), but for the tabulated functions \( t_o \) can optionally be set later by the T0x(i). The "x" represents "V", "P", "F", or "D" identifying the programmed quantity, the "i" label the different rf systems, and "n" is the tabular index. The continuous function \( f(t) \) is obtained by spline interpolation from the normalized table \( f_i(t_n) \). Linear interpolation rather than cubic is available for any of the rf curves by making the entry count, the first number in the table, negative. For example, to make voltage values used by the program identical to those appearing in the table, set VI(i) equal to the program value for \( t=\)TVBEG(i) and VF(i) to the program value for \( t=\)TVEND(i). There is no rigid relationship between the table times and the begin and end times in the A command data, but \( t_o \) should not be earlier than TxBEG(i) and \( t_{\text{max}} \) can not be later than TxEND(i).

The spline scheme assumes zero derivatives outside the range of the table when interpolating in the end intervals. Usually the assumption is satisfactory, but it is conservative to extend the tables a couple of intervals at each end to ensure the correct derivatives over the full time range needed. The parameters T0x(i) and TMx(i) are available to set the start of the intended function later than the first table values and the end later than the maximum, thereby avoiding the difficulty of having the function scaling determined from the leading or trailing guard values. The discussion at the end of 3.2.3 below is applicable to the rf curves also. The parameters TV0, TP0, TF0, TD0, JVM, TPM, TFM, and TDM are used to choose the times for the values VI, PHII, FI, DELTRFI, VF, PHIF, FRF, and DELTRFF in the same way as T0 and TM are used for a ramp table.

For the rf curve tables, the number of table points to be used must be included in the data as NTV(i), NTP(i), or NTF(i) where \( i \) ranges from 1 to NRF. The NTx may be less than or equal to the number of entries in the table. If the number of entries is given as negative in the file, the curve is interpolated linearly between the given values.

Rather than the spline or linear interpolation scheme, there is also the choice of a Bernstein polynomial representation. This option will give extremely smooth high order polynomial representations, but unless the number of points is adequate, the curve may evidence a disconcerting undulation. However, polynomials of order several hundred can be employed.

Curve files may be generated by independent custom programs of course, but ESME can do almost all of the work by using the history command to write the desired function to a file. Simple data can be used to establish parameter curves wanted for complicated problems.
When the program opens the curves file it looks for a record with a character string “SOURce i”, where “i” is a required integer identifying the rf system to which the following table(s) apply. All character headers are checked for the first four characters only; these characters must be caps. Thus, SOUR, SOURCE, SOURce are fine, but source or Source will not be recognized. These source headers can be in any numerical order, and they can be repeated for entering separate voltage, frequency, and phase programs for the same system if desired. It is also acceptable to read in more than one curve under a single source heading, but when this is done the tables must appear in the order voltage first followed by frequency, if present, phase, if present, and finally frequency offset, if present. The individual tables are identified by the character headers VOLTage, FREQuency, PHASe, and OFFSet, where again only the first four characters are significant. Schematically, a curves file might look like the following:

```
SOURce 1
VOLTS
  5
  0.   1.0
  .001 1.12
  .002 1.36
  .003 1.84
  .006 2.00
FREQ
  25
  0.  37.1101
  .001 37.1175
  .002 37.1203
  
SOUR 2
VOLTAGE
  5
  .01  2.00
  .011 1.8
  .013 1.7
  .014 1.1
  .016 1.0
SOURCE 2
PHAS
  150
  
```

etc.

### 3.2.3 Magnet ramp table

If either the KURVEB=4 or KURVEB=9 option is taken in the R command, one is to provide the name (complete path) of a file containing the ramp function by setting the character variable FILRMP in the `/RING/` namelist. The file is opened and assigned to FORTRAN unit 13. The first line of the table in list directed format is an integer; the absolute value is the number of table lines to follow. If the integer is negative, the interpolation in the table is linear; otherwise, cubic spline interpolation (KURVEB=4) or Bernstein polynomial fitting (KURVEB=9) is used.
Each succeeding line contains a time value and a ramp value. The scaling of the ramp values is arbitrary; the ramp is normalized. The scaling is determined by the /RING/ variables W0I or P0I and W0F or P0F. The first time entry should be at or before TI, and the final entry should be at or after TF. Having values outside the TI – TF range may provide a smoother $\dot{B}$ by properly representing the slope of the tabulated function at the ends. The use of a ramp table is more flexible, but also a bit more complicated than using the installed curve options. One should have a couple of guard values at times preceding and following the part of the curve to be referenced to ensure that the slope of the ramp does not misbehave in the end intervals. Because $\text{TI} \ (\text{which may be } < 0)$ should correspond to the first value in the table, which may be a guard value, it may not be desirable to have the ramp value be $\text{P0I} \ (\text{W0I})$ at $\text{TI}$. The parameter $\text{T0}$ is provided to choose the time at which $\text{P0} \ (\text{W0})$ is equal to $\text{P0I} \ (\text{W0I})$. The parameter $\text{P0F} \ (\text{W0F})$ sets the value of the curve at its maximum, not at its endpoint, so that the scaling of table values by $\text{P0} + (\text{P0F} - \text{P0I}) \times (\text{val(T)} - \text{val(T0)})$ gives the desired ramp values regardless of whether $\text{val(TF)}$ is the largest value in the table or not.

The KURVEB=9 option is similar to KURVEB=4 except that the table is used to construct a Bernstein polynomial representation of the curve. The result is very smooth, and there will be no contribution of the discrete character of the data to the smoothness of the derivative. A sufficient number of table values are needed to restrain the natural tendency of polynomial fits to undulate unacceptably; however, several hundred points may be used without problems.

### 3.2.4 Impedance table

The $\text{B}$ command allows the user to enter a table for the wall impedance as a function of frequency. The file name (complete path) is given as the value of the character variable $\text{FILIMP}$ read by the /SCHG/ namelist. The file will be opened and assigned to FORTRAN unit 11. The first record is an integer giving the number of table entries to follow. The maximum number of entries is set by the compiled parameter $\text{LTABL}$ which is established by the module parameters.f. Each following record consists of frequency (in MHz) accompanied by a real and imaginary part. The format of the input is as follows:

\[
\begin{align*}
\text{NZ} \\
 f^1 & \text{ } x^1 \text{ } y^1 \\
& \cdot \text{ } \cdot \text{ } \cdot \\
 f^{\text{NZ}} & \text{ } x^{\text{NZ}} \text{ } y^{\text{NZ}}
\end{align*}
\]

The frequency is in MHz; the impedance $Z^n = x^n + iy^n$ is in Ohms. The FFT used requires a negative sign for a capacitive reactance, i.e., $Z = R - i(X_C - X_L)$. The impedance should represent the value for the entire circumference, not the value per unit length or period. The FORTRAN format for each line is list-directed, so it is sufficient to list the values separated by commas or spaces in the file. The entries should be made in order of increasing frequency ($f^1 < f^2 < \ldots < f^{NZ}$). Impedance values will be calculated by cubic spline interpolation from the table. As for the rf tables, it may give a better curve if the function is continued two tabular intervals below and above the frequencies that will be needed for the problem. These guard values will ensure that the derivatives in the end intervals are correctly represented and avoid possible wild interpolation near the ends of the curve.

### 3.2.5 Resonance table

Resonant impedances may be specified in a simple form. The resonance values are read from the file whose name is specified by the $\text{FILRES}$ variable in the /SCHG/ namelist. This file is assigned to input unit 12. The first record is an integer giving the number of resonances to follow. The maximum number of entries is set by the compiled parameter $\text{LTABL}$ which is established by the module parameters.f. Each resonance is given in a single record. The format is list-directed, where the values to be read in are:

\[\text{see Section 2.2.12}\]
frequency [MHz], real shunt impedance at resonance [Ω], multiplier, Q

where the multiplier, which is usually 1., multiplies the impedance which is calculated from the frequency, shunt impedance, and Q values for the resonance. However note that, as mentioned in Section 2.2.12, a multiplier of zero has a special purpose. When the time domain calculation of resonator response has been activated by setting QREZON true, those resonances with a zero multiplier will be treated with the time domain technique. A frequency parameter < 0 is used to signal a resonance that tunes in direct proportion to the circulation frequency of the beam. The magnitude of a negative frequency is used as the proportionality constant; an integer results in a synchronous resonance. Thus, the magnitude of a negative frequency is like a harmonic number, but it need not be an integer. The maximum number of resonances is set to ten in the standard distribution by the parameter IRLLEN.

3.2.6 Time domain basis table

If the FILTDB string in the B command is not ’DUMMY’, a basis function or characteristic solution for the time domain response of some longitudinal impedance is read from the file specified by FILTDB. The time interval is the first record and one number from each following record is the value of the basis function for each successive time step. The appropriate length for the table is either the length of time for which the function has a significant amplitude or the circulation period divided by the periodicity, i.e., τ/FRAC. Taking FRAC > 1 corresponds to the physical situation where the impedance occurs multiple times in the ring. The basis function is the time response of the impedance to a triangular pulse of unit charge and length 2 × Δ, where Δ is the tabular interval. The interval should be short enough that a binning of the charge distribution at the same interval gives an adequate representation. Table size is limited by the compiled value of MAXCVB, 1000 in the standard compilation. However, the Y command can be used to allocate a larger table space with KMAXCVB (see section 2.2.17).

3.2.7 Bunch width table

One can damp quadrupole mode bunch oscillation (i.e., shape oscillation) by feeding back from bunch length measurement to the rf voltage. However, bunch length usually has a secular variation arising from voltage change or change in η with energy. Therefore, one needs to specify a transfer function for the feedback which does not confuse the desired secular change with the undesired oscillation. Since this transfer function will be problem dependent, it can not be built into the code. However, when one specifies the parameter time variation for a problem one knows enough to calculate the corresponding nominal bunch length. With a table of the expected bunch length vs. time, the feedback can be applied to just the error between actual and nominal. The actual length need not be correct so long as the functional form is right because the program normalizes the curve to the initial length. Thus, one table will usually work quite well over a substantial range of initial emittance. The times and bunch lengths are read from the file whose name is specified by the FILBWD parameter in the /LLRF/ namelist.⁴ This file is assigned to input unit 15. The first line is an integer giving the number of lines to follow. Each succeeding line contains a time [s] and a bunch length in any convenient units separated by a comma or space in the usual list-directed convention. Cubic spline interpolation is used between table values, so it is not likely that many entries will be required. Note that the table is not required. If FILBWD is not given, the initial bunch length is used as a target throughout the tracking. When the maximum rate of change of the bunch width is considerably faster than the rate of change of accelerator parameters, the bunch width table is not needed.

3.2.8 Particle distribution table

Occasionally it can be helpful to read in phase space coordinates from an external file for subsequent tracking. This can be done by using KIND=21 in the P command, giving the complete path to the table file as the string variable FILDST. The table is organized like almost all of the others with a particle count for the first record and θ, E − Es pairs as succeeding records. The maximum length of the distribution plus any others wanted is set by the

⁴See subsection 2.2.11
parameter KNPHASE read by the Y command. If KNPHASE is not read in, it defaults to the compiled parameter NPHASE, currently 25000, from the module parameters.f.

### 3.2.9 Examples

Below are some complete data sets for reasonably simple and conventional cases. Between the included comments, the information in section 2.2, and the output produced by running with the sample data, it should be possible to understand and use most program features. If one is trying to use the program to study multiparticle dynamics issues, it may be necessary to do some trial runs to understand what the program will do; ESME is not an expert system for collective effects, but it does provide powerful tools for a careful user.

```
===================== Bunch Pair Merging in CERN PS =======================
W Test of adiabatic bunch pair merging in the PS
Y This memory allocation reduces the default allocation.
 &MEMORY KNPHASE=2001 /END
R CERN PS at 3.57 GeV/c
 &RING REQ=100., GAMTSQ=37.21, W0I=2753., FRAC=5. /END
A h=20 and h=10 RF systems: turn down h=10 and turn up h=20 linearly.
 &RF NRF=2 H=20,10 VI=40.E-3,4.E-3 VF=4.E-3,40.E-3 TVEND=0.020,0.040
  KURVE=1,1 PSII=0.,-90. /END
P Parabolic bunch 1; center it in h=20 bucket. Distinguish it from bunch 2.
 &POPL8 KIND=13 SBNCH=0.4 NPOINT=1000 THOFF=-9. PARTION=.T. /END
P Parabolic bunch 2. Center in next h=20 bucket.
 &POPL8 THOFF=9. /END
A RF for tracking. Only the phases have changed; everything else is the same.
 &RF PSII=180.,0. /END
O Output format: Plot Delta E, Delta Theta scatter plot every 5000 turns.
 &GRAPH MPLOT=5000 PLTSW(8)=.F. PLTSW(10)=.F.
  TITL='BUNCH PAIR MERGING IN THE PS' /END
D Plot at start.
M Set up Mountain Range for a trace every 200 turns.
 &MRANGE MRMPLOT=200 IPU=T /END
T Tracking conditions: just go for 0.035 s.
 &CYCLE TTRACK=0.035 /END
N Plot mountain range with two iterations of local smoothing
 &MRPLOT SMOOTH=-2 NTRACE=80 SCALE=0.2 MRPMIN=-18. MRPMAX=18. /END
Q ESME stop.
```

```
===================== Basic Data for Crossing Transition ========================
W Fermilab MR: Normal transition crossing SIMPLIFIED (but still realistic)
W pdot @trans about 83 GeV/c/s
R MR: Start part way into cycle.
 &RING REQ=1000 p0i=9750. p0idot=25.83E3 p0f=26900 p0fdot =120.0E3
  KURVEB=6 GAMTSQ=355.32 ALPHA1=.0023237 FRAC=1113 TI=0.23035 TF=.46556
  TSTART=0.23035 /END
A Let program calculate voltage for constant area bucket
 &RF H=1113 ISYNC=1 vkon=f holdba=t VI=2. /END
O Plot bucket and rf waveform every 3000 turns.
 &GRAPH DEPMIN=-200. DEPMAX=200. MPLOT=3000
```

THPMIN=-0.097 THPMAX=0.097 PLTSW(8)=F PLTSW(10)=F PLTSW(5)=T
TITL='''MR(normal tr. x’ing) Constant area Bucket’’ /END
P 0.1 eVs parabolic bunch
&POPL8 KIND=14 SBNCH=.1 NPOINT=200 /END
D Plot initial distribution
T
&CYCLE TSTOP=0.46556 HISTRY=T /END
D Plot at very end
R M R Ramp II: Linear portion of ramp
&RING JNRAMP=T TF=.79139 p0f=66000. KURVEB=1 /END
T
&CYCLE TSTOP=0.645 /END
D
H
&HISTORY
NPLT=1,2,1,3,1,4,1,5,1,6,1,7,1,13,1,14,1,15,1,101,1,32,1,17,1,17 /END
Q

 ===================== As it were Realistic Transition Crossing ======================

W Fermilab MR : Normal transition crossing
W N_q=2.5e10 Z_||/n=8.6 PIPRAD = 2.54cm
W This example involves a couple of external files.
W pdot @trans about 83 GeV/c/s
R MR: Start part way into cycle; discard particles > 2.54 cm off central orbit
&RING REQ=1000 p0i=9750. p0idot=25.83E3 p0f=26900 p0fdot=120.0E3
    KURVEB=6, GAMTSQ=355.32, ALPHAL=.0023237 FRAC=1113 TI=0.23035 TF=.46556
    TSTART=0.23035 PIPRAD=0.0254 EBDRY=T /END
A Use a voltage curve from an external file
&RF H=1113 ISYNC=1 VI=2. VF=3.699412 TVBEG=.23035 TVEND=.6455522 KURVE=4
    FILCRV='mrtrramp.crv' NTV=138 /END
O
&GRAPH DEPMIN=-200. DEPMAX=200. MPLIT=3000
    THPMIN=-0.097 THPMAX=0.097 NBINTH=100 DELCON=0.005 NPJMP=10
    PLTSW(8)=F PLTSW(10)=F PLTSW(5)=T
    TITL=''MR(normal tr. x’ing): N_q=2.5e10, Z_||/n=8.6’’ /END
P Going to look at collective volts up to 3 GHz; need AT LEAST this many.
&POPL8 KIND=14 SBNCH=.1 IPO=1 NPOINT=20000 /END
B Q=1 resonance (described on external file) to represent broad band Z_||/n
&SCHG SCON=T FDON=T ENQ=2.5e10 A=4.E-3 B=2.5E-2
    NBINSC=128 TCHGON=0.0 NBINFFT=128, NIXNOIS=1 FILIMP='mrzon_8pt6ohm.dat' /END
L Use phase feedback to reduce dipole oscillations.
&LLRF PHFBON=T /END
D
T
&CYCLE TSTOP=0.46556 HISTRY=T /END
D
R M R Ramp II: Linear portion of ramp
&RING JNRAMP=T, p0fdot=120.0E3 p0idot=120.0E3
    TF=.79139 p0f=66000. TI = 0.46556 KURVEB=1 /END
D
T
&CYCLE TSTOP=0.645 /END

D

H

&HISTORY

NPLT=1,2,1,3,1,4,1,5,1,6,1,7,1,13,1,14,1,15,1,101,1,32,1,17,1,17 /END

Q

---------------------------------------------------

FOR MAIN RING WITH Z/N=8.6 OHMS AND CUT OFF OF 1.7 GHZ (File mrz8pt6ohm.dat)

---------------------------------------------------

SOURCE 1

VOLTS

<table>
<thead>
<tr>
<th>VOLTS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.303500E-01</td>
<td>2.000000E+00</td>
<td>2.353401E-01</td>
</tr>
<tr>
<td>2.453406E-01</td>
<td>2.088300E+00</td>
<td>2.503482E-01</td>
</tr>
<tr>
<td>2.603424E-01</td>
<td>2.142347E+00</td>
<td>2.653290E-01</td>
</tr>
<tr>
<td>2.753433E-01</td>
<td>2.170156E+00</td>
<td>2.803308E-01</td>
</tr>
<tr>
<td>2.903373E-01</td>
<td>2.180647E+00</td>
<td>2.953378E-01</td>
</tr>
<tr>
<td>3.053389E-01</td>
<td>2.181473E+00</td>
<td>3.103394E-01</td>
</tr>
<tr>
<td>3.203405E-01</td>
<td>2.176977E+00</td>
<td>3.253410E-01</td>
</tr>
<tr>
<td>3.353421E-01</td>
<td>2.164204E+00</td>
<td>3.403426E-01</td>
</tr>
<tr>
<td>3.503435E-01</td>
<td>2.133029E+00</td>
<td>3.553370E-01</td>
</tr>
<tr>
<td>3.608339E-01</td>
<td>2.081888E+00</td>
<td>3.613375E-01</td>
</tr>
<tr>
<td>3.623446E-01</td>
<td>2.073459E+00</td>
<td>3.628481E-01</td>
</tr>
<tr>
<td>3.638342E-01</td>
<td>2.061674E+00</td>
<td>3.643377E-01</td>
</tr>
<tr>
<td>3.653448E-01</td>
<td>2.047943E+00</td>
<td>3.658274E-01</td>
</tr>
<tr>
<td>3.668454E-01</td>
<td>2.032339E+00</td>
<td>3.673380E-01</td>
</tr>
<tr>
<td>3.683451E-01</td>
<td>2.013773E+00</td>
<td>3.688276E-01</td>
</tr>
<tr>
<td>3.698347E-01</td>
<td>1.991851E+00</td>
<td>3.703383E-01</td>
</tr>
<tr>
<td>3.713453E-01</td>
<td>1.963920E+00</td>
<td>3.718279E-01</td>
</tr>
<tr>
<td>3.728350E-01</td>
<td>1.926325E+00</td>
<td>3.733385E-01</td>
</tr>
<tr>
<td>3.743456E-01</td>
<td>1.858132E+00</td>
<td>3.748282E-01</td>
</tr>
<tr>
<td>3.758226E-01</td>
<td>1.961538E+00</td>
<td>3.763261E-01</td>
</tr>
<tr>
<td>3.773322E-01</td>
<td>2.073507E+00</td>
<td>3.778367E-01</td>
</tr>
<tr>
<td>3.788228E-01</td>
<td>2.154578E+00</td>
<td>3.793264E-01</td>
</tr>
<tr>
<td>3.803335E-01</td>
<td>2.232394E+00</td>
<td>3.808370E-01</td>
</tr>
<tr>
<td>3.818231E-01</td>
<td>2.233351E+00</td>
<td>3.823266E-01</td>
</tr>
<tr>
<td>3.833337E-01</td>
<td>2.381088E+00</td>
<td>3.838373E-01</td>
</tr>
<tr>
<td>3.848234E-01</td>
<td>2.388211E+00</td>
<td>3.853269E-01</td>
</tr>
<tr>
<td>3.863340E-01</td>
<td>2.435565E+00</td>
<td>3.868375E-01</td>
</tr>
<tr>
<td>3.878236E-01</td>
<td>2.479796E+00</td>
<td>3.883272E-01</td>
</tr>
<tr>
<td>3.893342E-01</td>
<td>2.522257E+00</td>
<td>3.898378E-01</td>
</tr>
<tr>
<td>3.908348E-01</td>
<td>2.567777E+00</td>
<td>4.003282E-01</td>
</tr>
<tr>
<td>4.013356E-01</td>
<td>2.986864E+00</td>
<td>4.153295E-01</td>
</tr>
<tr>
<td>4.253373E-01</td>
<td>3.24913E+00</td>
<td>4.303308E-01</td>
</tr>
<tr>
<td>4.403386E-01</td>
<td>3.467440E+00</td>
<td>4.453321E-01</td>
</tr>
<tr>
<td>4.553399E-01</td>
<td>3.668189E+00</td>
<td>4.603334E-01</td>
</tr>
<tr>
<td>4.705510E-01</td>
<td>3.800847E+00</td>
<td>4.755445E-01</td>
</tr>
<tr>
<td>4.855523E-01</td>
<td>3.811178E+00</td>
<td>4.905458E-01</td>
</tr>
</tbody>
</table>

Q

53
<table>
<thead>
<tr>
<th>Voltage 1</th>
<th>Voltage 2</th>
<th>Voltage 3</th>
<th>Voltage 4</th>
<th>Voltage 5</th>
<th>Voltage 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.005536E-01</td>
<td>3.812824E+00</td>
<td>5.055470E-01</td>
<td>3.812006E+00</td>
<td>5.105405E-01</td>
<td>3.810700E+00</td>
</tr>
<tr>
<td>5.155549E-01</td>
<td>3.808847E+00</td>
<td>5.205483E-01</td>
<td>3.806463E+00</td>
<td>5.255418E-01</td>
<td>3.803827E+00</td>
</tr>
<tr>
<td>5.305562E-01</td>
<td>3.800888E+00</td>
<td>5.355496E-01</td>
<td>3.797649E+00</td>
<td>5.405431E-01</td>
<td>3.794283E+00</td>
</tr>
<tr>
<td>5.455575E-01</td>
<td>3.790346E+00</td>
<td>5.505509E-01</td>
<td>3.786541E+00</td>
<td>5.555444E-01</td>
<td>3.782531E+00</td>
</tr>
<tr>
<td>5.605378E-01</td>
<td>3.778366E+00</td>
<td>5.655522E-01</td>
<td>3.774137E+00</td>
<td>5.705457E-01</td>
<td>3.769613E+00</td>
</tr>
<tr>
<td>5.755391E-01</td>
<td>3.765215E+00</td>
<td>5.805535E-01</td>
<td>3.760652E+00</td>
<td>5.855470E-01</td>
<td>3.756090E+00</td>
</tr>
<tr>
<td>5.905404E-01</td>
<td>3.751374E+00</td>
<td>5.955548E-01</td>
<td>3.746831E+00</td>
<td>6.005483E-01</td>
<td>3.742003E+00</td>
</tr>
<tr>
<td>6.055417E-01</td>
<td>3.737262E+00</td>
<td>6.105561E-01</td>
<td>3.732544E+00</td>
<td>6.155496E-01</td>
<td>3.727885E+00</td>
</tr>
<tr>
<td>6.205430E-01</td>
<td>3.723113E+00</td>
<td>6.255574E-01</td>
<td>3.718336E+00</td>
<td>6.305509E-01</td>
<td>3.713484E+00</td>
</tr>
<tr>
<td>6.355443E-01</td>
<td>3.708887E+00</td>
<td>6.405377E-01</td>
<td>3.704164E+00</td>
<td>6.455522E-01</td>
<td>3.699412E+00</td>
</tr>
</tbody>
</table>

(Note: If you want to scan the above voltage table to try this example, you will need to replace the \’s with carriage returns to get the table format of one time and one voltage per record.)
Chapter 4

Customizing the Code

In this chapter some of the basic structure of the program is described, and the use of pre-compilation directives in adapting the code is discussed. The material in this section is not ordinarily needed to use the program. This section alone is not likely to make the reader an ESME expert; however, it will be useful and perhaps even sufficient for someone wanting to add a custom feature via the SHAZAM entries. Some of this information should also be useful to those needing to install the code in a new environment.

4.1 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 contains an edited listing of the modules containing the global variables. The fourth section provides an example of a SHAZAM routine which is more elaborate than any other written by the authors. It is interesting because it uses the the same routine called explicitly with the 2 command and called with ITRAP flags at both the beginning and end of an iteration.

4.1.1 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 modules each of which contain a group of closely related quantities. Higher level subroutines communicate through these modules. 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 the modules; only loop counters and a few intermediate results are purely local variables. Thus, the program is divided into numerous subroutines but important variables are available by USEing the needed modules. The large number of parameters often required to specify a distribution and the rf systems which act upon it encourage this structure. The structure has also proved amenable to extension without major interference with existing features.

4.1.2 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 time to track etc. and then applys the difference equations to each particle on each iteration. 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. Optional calls to one of the 10 SHAZAM entries can be made either preceding or following an iteration. For most purposes the alternatives are equally satisfactory, but some quantities are re-initialized at the beginning of turns and can not be changed by a SHAZAM call at the end.
4.1.3 Important variables

The most useful modules are listed below. The function of important variables is noted. Many items are the input data or the default values used in their absence; these are explained in the command descriptions. The command that stores problem parameters into a particular module is noted. When a variable is not explained either here or in a command description, it is an intermediate working quantity that is likely to be unuseful or risky to change.

C
C bktsup.f (edited)
  MODULE BKTSUP
C  Block parameterizing isolated or barrier bucket generation.
  USE PARAMETERS
C
D  DOUBLE PRECISION THL(NSRC),THU(NSRC)
! Lower and upper phase bounds for truncated rf waveform (suppressed buckets)
C
END MODULE BKTSUP
C
C blankcom.f (edited)
  MODULE BLANKCOM
C
  USE PARAMETERS
C
  THE MAIN STORAGE BLOCK IN DYNMEM CONTAINS KNPHASE PHASE SPACE COORDINATES
C
  INTEGER KOUNT,KNTSC,KLASSES,KLIMIT(0:NKLIM),KWEERKNT=0
  LOGICAL PARTION
! KOUNT number of macroparticles, KNTSC number which contribute to beam charge,
! KLASSES number of partitions of macroparticle distribution, KLIMIT index of
! partition boundaries, i.e., first particles in each class. Partitioning enabled
! by switch PARTION
! KWEERKNT number of lost macroparticles
C
END MODULE BLANKCOM
C
C bucket.f (edited)
  MODULE BUCKET
C  BUCKET PARAMETERS TURN-BY-TURN
  DOUBLE PRECISION :: GNUS,SBCKT,HBCKT
! GNUS synchrotron tune, SBCKT bucket area, HBCKT bucket height
  INTEGER :: NFIRST=1,NLAST
C
END MODULE BUCKET
C
C bunch.f (edited)
  MODULE BUNCH
C  BUNCH PARAMETERS TURN-BY-TURN
    DOUBLE PRECISION :: THBAR,EBAR,THRMS,ERMS,EPSILON,ANORM=1.D6

! THBAR average phase, EBAR average energy, THRMS rms phase spread,
! ERMS rms energy spread, EPSILON rms emittance (unless modified by
! ANORM .NE. TENTO6), ANORM renormalizing factor

C END MODULE BUNCH
C ---------------------------------------------------------------
C const.f (edited)
MODULE CONST
C CONTAINS MATHEMATICAL AND PHYSICAL CONSTANTS, PROGRAM CONSTANTS, AND
C CONVERSION FACTORS FROM EXTERNAL TO INTERNAL UNITS

DOUBLE PRECISION :: HALFPIE=1.57079632674879D+0,
1 PIE=3.141592653589793D+0,TWOPIE=6.283185307179586D+0,
2 FOURPIE=12.56637061435917D0,RADDEG=57.29577951308232D0,
3 DEGRAD=1.7453292519943296D-02,C=2.99792458D+8,
4 EMPCSQ=9.3827231D+2,EMECSQ=5.1099906D-1,
5 RE=2.81794092D-15,RP=1.534698D-18,QE=1.60217733D-19,
6 ZNAUGHT=3.7673031D+2
! Multiples of pi, angle conversions, velocity of light, proton and
! electron rest energies, electron and proton classical radii, free space
! impedance
C END MODULE CONST
C ---------------------------------------------------------------
C current.f (edited)
MODULE CURRENT
C MISCELANEOUS PARAMETERS AND INTERMEDIATE RESULTS TURN-BY-TURN
C
USE PARAMETERS

DOUBLE PRECISION :: ES,PS,RS,BETAS,BETASQ,ETA,E0,P0,BETA0,THETAS,
1 THREF,EREF,TAU,TIME=0.D0,DTIME,DEBFLD,PDOT,DP0DT, ELO, EHI,
2 TGCURR, GMSQINV, GAMMAT, DELR, DELES, SBTGT, HBTGT, RSCALE=ONE,
3 RSCAL=4.D-1,TAUINIT,DELEIMP=ZERO,ALFA0,ALFA1,ALFA2,ALFA3,
4 EV(NSRC),PSI0(NSRC)=NSRC*ZERO,PSI(NSRC),FREQ(NSRC),
5 DELTRF(NSRC)
! ES synchronous energy, PS synchronous momentum, RS synchronous radius,
! BETAS synchronous velocity, BETASQ beta_s^2, ETA time slip factor,
! E0 central orbit energy, P0 central momentum, BETA0 central velocity,
! THETAS shift to total rf waveform to put THETA=0 at synchronous phase
! THREF,EREF theta-E coordinates of reference particle, TAU beam circulation
! period, TIME calculated time, DTIME iteration interval
! DEBFLD energy change per turn of central orbit,
! PDOT rate of synchronous momentum change, DP0DT rate of change of central
! momentum, ELO,EHI limits of energy corresponding to PIPRAD input parameter,
! TGCURR time of starting gamma-t jump, GMSQINV gamma^-2, GAMMAT transition
! gamma on reference orbit, DELR=RS-REQ, DELES synchronous energy change per turn,
! SBTGT & HBTGT target values for constant area and constant height buckets
! impedance, ALFA 0-3 coefficients in the momentum expansion of path length

57
! RSCALE time scaling factor for the main tracking, RSCAL time scaling for contouring
! TAUINIT initial value of beam circulation period, DELEIMP energy loss to real
! EV the rf voltages, PSI0 the accumulated phase slips of asynchronous rf systems
! PSI phase increment each iteration, FREQ rf frequencies, DELTRF energy offset of
! asynchronous systems
! phase, DELEIMP energy loss per turn to real impedance
INTEGER M,MM,NITS,MHIST,MOST
! M turn number in current T command, MM turn number, NTurns turns counter
! for writing history record, MHIST calculated turns interval for writing
! history records, MOST the index for the principal rf system (greatest
! bucket height)
LOGICAL TRPSW0,TRPSW1
! flags for traps in CYCPROG active at beginning of turn and at end of turn
C
END MODULE CURRENT
C -------------------------------------------------- ------------------
C curves.f (edited)
MODULE CURVES
C PHASE SPACE COORDINATES OF BUCKET OR OTHER CONTOUR OF INTEREST
C
USE PARAMETERS
C
DOUBLE PRECISION CURVE(ICURVE,2),TURNPT(4,3),
1 TOP(ICURVE/2,2),BOTTOM(ICURVE/2,2)
INTEGER NC
! CURVE theta,E coordinates of a curve to be plotted on phase plane,
! TURNPT the turning points of the curve (third coordinate is number of the
! point along the curve), TOP, BOTTOM coordinates for top and bottom halves
! of closed curve, NC number of points on curve
C
END MODULE CURVES
C -------------------------------------------------- ------------------
C cyclp.f (edited) Parameters not described are from the T command.
MODULE CYCLP
C PARAMETERS GOVERNING DURATION & OPTIONAL FEATURES OF THE TRACKING CALCULATION
C
USE PARAMETERS
C
DOUBLE PRECISION :: TBEGIN,TEND,TSTART=0.D0,TSTOP=0.D0,
1 TTRACK=0.D0,THREF0=0.D0,EREF0=0.D0,
2 ETATRP=1.D-3,PHISTRP=9.5D-1,HISTSIZ=1.D4,
3 RSCALE0=1.D+0,RAILU=TWO,RAILL=HALF,GNUSCAL=1.D-2
INTEGER :: LGRTHM=1,ITRAP(NTRP),MGRACE(NTRP),NTRAP
! NTRP is a compiled parameter (parameters.inc)
LOGICAL :: HISTORY=.FALSE.,BBDRY=.FALSE.,AUTOSCL=.FALSE.
C
END MODULE CYCLP
C
C dynmem.f

MODULE DYNMEM
    DOUBLE PRECISION, ALLOCATABLE :: PHASETH(:,), PHASEE(:,)
! The phase space coordinates
    DOUBLE PRECISION, ALLOCATABLE :: EVRES(:,), ZRES(:,)
! Voltage produced by resonator in Green’s function solution and work array
    DOUBLE PRECISION, ALLOCATABLE :: DQ(:,), THTA(:,), DQ2(:,)
! Intermediate arrays in collective voltage calculations (FD & TD)
    DOUBLE PRECISION, ALLOCATABLE :: DQDT(:,)
! Intermediate charge histogram for Green’s function solution
    DOUBLE PRECISION, ALLOCATABLE :: USC(:,)
! Intermediate array in calculation of spline fit to charge histogram
    DOUBLE PRECISION, ALLOCATABLE :: TDTBL(:,)
! Input array of elementary solution for time domain
    DOUBLE PRECISION, ALLOCATABLE :: EVTD(:,), FIT(:,)
! Time domain result for collective voltage and a work array
    DOUBLE PRECISION, ALLOCATABLE :: WSAVE(:,), F(:,), G(:,), EVFD(:)
! Work array for FFT and transform and original function
    DOUBLE PRECISION, ALLOCATABLE :: RTABL1(:,), RTABL2(:,), RTABL3(:,)
! Magnet ramp table for KURVB=4
    DOUBLE PRECISION, ALLOCATABLE :: BLTABL1(:,), BLTABL2(:,), BLTABL3(:,)
! Optional bunch length table for voltage feedback
    DOUBLE PRECISION, ALLOCATABLE :: VTABL(:)
! Voltage from external file
    DOUBLE PRECISION, ALLOCATABLE :: PTABL(:)
! Phase from external file
    DOUBLE PRECISION, ALLOCATABLE :: FTABL(:,)
! Frequency from external file
    DOUBLE PRECISION, ALLOCATABLE :: DTABL(:,)
! Energy offset (i.e. radial offset) from external file
    DOUBLE PRECISION, ALLOCATABLE :: VTABL1(:,), VTABL2(:,), VTABL3(:,)
! Voltage vs. time table with cubic spline coefficients
    DOUBLE PRECISION, ALLOCATABLE :: PTABL1(:,), PTABL2(:,), PTABL3(:,)
! Phase vs. time table with cubic spline coefficients
    DOUBLE PRECISION, ALLOCATABLE :: FTABL1(:,), FTABL2(:,), FTABL3(:,)
! Frequency vs. time table with cubic spline coefficients
    DOUBLE PRECISION, ALLOCATABLE :: DTABL1(:,), DTABL2(:,), DTABL3(:,)
! Offset vs. time table with cubic spline coefficients
    DOUBLE PRECISION, ALLOCATABLE :: VBCOEFT(:,,:), VBXI(:,,:), VBYI(:,,:)
! Bernstein polynomial fit tables for Voltage
    DOUBLE PRECISION, ALLOCATABLE :: PBCOEFT(:,,:), PBXI(:,,:), PBYI(:,,:)
! Bernstein polynomial fit tables for Phase
    DOUBLE PRECISION, ALLOCATABLE :: FBCOEFT(:,,:), FBXI(:,,:), FBYI(:,,:)
! Bernstein polynomial fit tables for Frequency
    DOUBLE PRECISION, ALLOCATABLE :: DBCOEFT(:,,:), DBXI(:,,:), DBYI(:,,:)
! Bernstein polynomial fit tables for Offset
    DOUBLE PRECISION, ALLOCATABLE :: VBXITMP(:,), VBYITMP(:,), VBCTMP(:,)
! Temporary tables for calculating intermediate quantities in Bernstein polynomial fits
    DOUBLE PRECISION, ALLOCATABLE :: FBXITMP(:,), FBYITMP(:,), FBCTMP(:,)

59
DOUBLE PRECISION :: TSTART=ZERO,TSTOP=ZERO,TTRACK=ZERO,
      1 RSCALE0=ONE
! Same as cyclp.f tracking parameters of same name
INTEGER :: KNTL(NKLIM)
! The final indices of flowlines
LOGICAL :: STATIC=.TRUE.,ACCUMUL8=.FALSE.,AUTOSCAL=.FALSE.
! STATIC is switch to hold the time constant during generation of a contour
! ACCUMUL8 is switch to retain contours from all generation times
! AUTOSCAL is a switch enabling time autoscaling
END MODULE FLOWP

C -------------------------------------------------- ------------------
C fourir.f (edited) Parameters not described are from the F or B commands.
MODULE FOURIR
C PARAMETERS DEFINING AND RESULTS OF FOURIER TRANSFORM OF CHARGE DISTRIBUTION
C
USE PARAMETERS
C
DOUBLE PRECISION :: BW=0.D0,OBWT=3.D0,FAMPL(NFMAX),FAZE(NFMAX)
! BW width of bin in charge histogram for fft, FAMPL phasor amplitude,
! FAZE phase of phasor, W work space for fft, F Fourier amplitudes,
! G function being analyzed
INTEGER :: NBINFFT=256,MFFT=1,NNF=0,NBSAV=0,NIXNOIS=0,ITKNT=1,
      1 NF(NFMAX),MLAST
! NBSAV last bin count, MLAST last turn number that FFT was called
LOGICAL :: FFTON=.FALSE.,FFTOUT=.FALSE.,FTFWRT=.FALSE.
C
END MODULE FOURIR

C -------------------------------------------------- ------------------
C grafix.f (edited) Parameters not described are from the O command.
MODULE GRAFIX
C PARAMETERS DEFINING DESIRED GRAPHICAL OUTPUT
DOUBLE PRECISION :: THPMIN,THPMAX,DEPMIN,DEPMAX,RFVMIN,RFVMAX,
      1 DTHCURV,DECURV,THBMIN,THBMAX,EBMIN,EBMAX,SREF,REFAREA,
      1 CVBMIN,CVBMAX,DELCON=1.D-2,THEXCMI,THEXCPL,DEEXCMI,DEEXCPL,
      2 SEXCL,XCRNR=1.35D2,YCRNR=6.9D1,XAXISL=7.5D2,YAXISL=4.75D2

INTEGER :: MPLOT=1000,IRF=1,ICONTUR=1,IEREF=1,NBINTH=50,NBINE=50,
      1 IFBMIN=1,IFBMAX=0,NPJMP=1,KLPLOT=0
LOGICAL :: PLTSW(40),DRWREF,NODRAW=.TRUE.,POSTP,LINTER,ZTABFLG,
      1 TDBFFLG
! DRWREF flag signals a reference contour for phase plane plot, NODRAW
! flag indicating no plotting, ZTABFLG flag to plot impedance table,
! TDBFFLG flag to plot time domain basis function, LINTER flag to pause
! between plots
CHARACTER :: PGINFIL*256,PGOUTFIL*256,PGDEVICE*32
! PGINFIL name of main data file, PGOUTFIL name of postscript output file
! PGDEVICE driver name for PGPLOT output
C DESCRIPTIVE HEADING FOR GRAPHICAL OUTPUT
C
CHARACTER (LEN=55) ::
  1 TITL='ESME (esmf95) macroparticle rf simulation '
INTEGER :: TITLEN=50
! TITLEN number of characters in plot heading
C
END MODULE GRAFIX
C
-------------------------------------------------------------------------------
C io.f (edited)
MODULE IO
C IO UNITS AND ARRAY SIZES
C
CHARACTER*512 PGMARG, LOGFIL
! PGMARG the command line arguments, LOGFIL the ascii output file
CHARACTER*512 TEMPFIL
! TEMPFIL prefix identifying temporary files (e.g., FORT.9 --> TEMPFIL.9)
C
INTEGER KINUNIT,KOUTUNIT,KVERBOSE
! KINUNIT,KOUTUNIT unit numbers for principal input and ascii output files
! KVERBOSE integer to select quantity of log file output
INTEGER KMAXCVB,KNPHASSE,KMAXPTS,KIFFT,KLTABL,KNSRC
! KMAXCVB Length of arrays for collective potential calculation and plots
! KNPHASSE Length of arrays for macroparticle coordinates and attributes
! KMAXPTS Maximum number of points in HISTORY plots
! KIFFT Maximum number of points for FFT
! KLTABL Length for many auxiliary tables
! KNSRC maximum number of rf sources; allocated dimension in many rf tables
C
LOGICAL :: LCOLOR,ASCII
! LCOLOR switch for color plots
END MODULE IO
C
-------------------------------------------------------------------------------
C mrange.f (edited) The parameters are from the M command.
MODULE MTNRANGE
C Module containing mountain range parameters for saving data;
C plotting parameters are local to MRPLT.
DOUBLE PRECISION MRMINB,MRMAXB,TMBED,TMEND
INTEGER MRMPLOT,NUMBIN
LOGICAL :: IPU=.TRUE.,FASPEC=.FALSE.,FPSPEC=.FALSE.,
          1 EVFDSPEC=.FALSE.,EVTDSPEC=.FALSE.,EVRESPEC=.FALSE.
C
END MODULE MTNRANGE
C
-------------------------------------------------------------------------------
C parameters.f
MODULE PARAMETERS
C
C GLOBAL PARAMETER DEFINITIONS
C ----------------------------
C
    LOGICAL, PARAMETER :: LDMEM=.TRUE.
    PARAMETER (LDMEM=.TRUE.)
C BCKRAT = bucket ratio used in MATCH subroutine
    DOUBLE PRECISION, PARAMETER :: BCKRAT=1.26D0
    PARAMETER (BCKRAT=1.26D0)
C ICURVE = max no of points in a phase contour
    INTEGER, PARAMETER :: ICURVE = 50001
    PARAMETER (ICURVE = 50001)
C KNTLIM = maximum number of integration steps in constructing flowline or contour
    INTEGER, PARAMETER :: KNTLIM = 250000
    PARAMETER (KNTLIM = 250000)
C IFBPRS = length of circular buffer for averaging successive turns in phase feedback
    INTEGER, PARAMETER :: IFBPRS = 1001
    PARAMETER (IFBPRS = 1001)
C IFFT = maximum array size for FFT
    INTEGER, PARAMETER :: IFFT = 1024
    PARAMETER (IFFT = 1024)
C IFLSIZ = max no of bins for Bersntein polynomial smoothing
    INTEGER, PARAMETER :: IFLSIZ=500
    PARAMETER(IFLSIZ=100)
C IINUNIT = 5 INPUT UNIT
    INTEGER, PARAMETER :: IINUNIT=5
    PARAMETER (IINUNIT=5)
C IFINUNIT = 32 UNIT FOR OPTIONAL INPUT FILE
    INTEGER, PARAMETER :: IFINUNIT=32
    PARAMETER (IFINUNIT=32)
C IFLOGUNIT = 33 UNIT FOR OPTIONAL LOG FILE
    INTEGER, PARAMETER :: IFLOGUNIT=33
    PARAMETER (IFLOGUNIT=33)
C IOUTUNIT = 6 OUTPUT UNIT
    INTEGER, PARAMETER :: IOUTUNIT=6
    PARAMETER (IOUTUNIT=6)
C IRLLEN = max no of resonators in time-domain Green’s func. soln.
    INTEGER, PARAMETER :: IRLLEN=10
    PARAMETER (IRLLEN=5)
C IVERBOSE = 0 NAMELISTS ARE NOT PRINTED
    INTEGER, PARAMETER :: IVERBOSE=1
    PARAMETER (IVERBOSE=1)
C LTABL = 750 MAXIMUM LENGTH FOR SOME DATA TABLES
    INTEGER, PARAMETER :: LTABL=750
    PARAMETER (LTABL=750) -- Very large value
C MAXBINS = number of bins in rf voltage plot (PLTWAV)
    INTEGER, PARAMETER :: MAXBINS=400
    PARAMETER (MAXBINS=400)
C MAXCNT = max no of attempts to match a bucket to a bunch
INTEGER, PARAMETER :: MAXCNT=12
C PARAMETER (MAXCNT=12)
C MAXCVB = max no of bins in longitudinal charge distribution
C (used for all collective voltage calculations in time domain)
 INTEGER, PARAMETER :: MAXCVB=10000
C PARAMETER (MAXCVB=10000)
C MAXPLT = max no of history plots
 INTEGER, PARAMETER :: MAXPLT=100
C PARAMETER (MAXPLT=50)
C MAXPTS = max no of points in a history plot
 INTEGER, PARAMETER :: MAXPTS=1000
C PARAMETER (MAXPTS=1000)
C MAXTRC = max no of points on a trace in a mountain range plot
 INTEGER, PARAMETER :: MAXTRC=4704
C PARAMETER (MAXTRC=2052) ? How big is big enough?
C MAXTTL = maximum length of plot title
 INTEGER, PARAMETER :: MAXTTL=50
C PARAMETER (MAXTTL=50)
C NDBLSIZ = size of a DOUBLE PRECISION, in bytes
 INTEGER, PARAMETER :: NDBLSIZ = 8
C PARAMETER (NDBLSIZ = 8)
C NFMAX = max no of Fourier components saved for history
 INTEGER, PARAMETER :: NFMAX = 50
C PARAMETER (NFMAX = 50)
C NIT = no of iterations for binary search (implicitly def fnct)
 INTEGER, PARAMETER :: NIT=25
C PARAMETER (NIT=25)
C NKLIM = max no of distinct classes of phase space point
 INTEGER, PARAMETER :: NKLIM=200
C PARAMETER (NKLIM=600) ! large value for MI multibatch injection
C NPHASE = max no of phase points
 INTEGER, PARAMETER :: NPHASE = 25000
C PARAMETER (NPHASE = 25000)
C NPMAX = max no iterations (SYNCH) of synchronous phase for multiple RF systems
 INTEGER, PARAMETER :: MAXIT =50
C PARAMETER(MAXIT =50)
C NRD = size of the random no array for Knuth’s Random no generator
C **must be > 100**
 INTEGER, PARAMETER :: NRD=200
C PARAMETER (NRD=200)
C NREALSZ = size of a REAL, in bytes
 INTEGER, PARAMETER :: NREALSZ = 4
C PARAMETER (NREALSZ = 4)
C NSPARE = no of slots for user-defined histories
 INTEGER, PARAMETER :: NSPARE = 50
C PARAMETER (NSPARE = 50)
C NSRC = max no of RF systems
 INTEGER, PARAMETER :: NSRC = 10
C PARAMETER (NSRC = 10)
C NTRIALS = no of attempts to complete a contour
 INTEGER, PARAMETER :: NTRIALS = 64
INTEGER, PARAMETER :: NTRIALS=6
C PARAMETER (NTRIALS=6)
C NTRP = max no of simultaneously active trapping conditions
INTEGER, PARAMETER :: NTRP = 4
C PARAMETER (NTRP = 4)
C PVER = Post-processor version no
DOUBLE PRECISION, PARAMETER :: PVER=2.11
C PARAMETER (PVER=2.11)
C SSIZ = ? (MATCH subroutine)
DOUBLE PRECISION, PARAMETER :: SSIZ=5.D-1
C PARAMETER (SSIZ=5.D-1)
C TITLL = axis label length for user-defined history plots
DOUBLE PRECISION, PARAMETER :: TITLL=14
C PARAMETER (TITLL=14)
C TOL = tolerance for moving bucket factor interpolation
DOUBLE PRECISION, PARAMETER :: TOL=1.D-6
C PARAMETER (TOL=1.D-6)
C TOL2 = tolerance for MATCH subroutine
DOUBLE PRECISION, PARAMETER :: TOL2=1.D-1
C PARAMETER (TOL2=1.D-1)
C UNITL = length of unit field in user-defined history plots
DOUBLE PRECISION, PARAMETER :: UNITL=7
C PARAMETER (UNITL=7)
C VERNUM = ESME version number
DOUBLE PRECISION, PARAMETER :: VERNUM=2006.5
C PARAMETER (VERNUM=2005.5)
C
C ==============================================================
C--PARAMETERS USED IN PLOTTING ROUTINES (PGPLOT VERSION ONLY)
C
C COLOR = .FALSE.
LOGICAL, PARAMETER :: COLOR=.TRUE.
C PARAMETER (COLOR=.TRUE.)
C
C Viewport margins in normalized coordinates
C
REAL, PARAMETER :: VPLFT=0.10, VPRGT=0.90, VPBTM=0.10,
1 VPTOP=0.75
C PARAMETER (VPLFT=0.10, VPRGT=0.90, VPBTM=0.10, VPTOP=0.75)
C Plot title character size, sub-title character size,
C Notes character size
REAL, PARAMETER :: CTSIZ=1.5, CSSIZ=1.0, CNSIZ=0.70
C PARAMETER (CTSZ=1.5, CSSIZ=1.0, CNSIZ=0.70)
C
C ==============================================================
C
C Finally, some useful and not so useful definitions ...
C
DOUBLE PRECISION, PARAMETER :: BIG=9.D37, SMALL=0.7D-38
PARAMETER (BIG=9.D37, SMALL=0.7D-38)
DOUBLE PRECISION, PARAMETER :: VYBIG=1.7D38,
1   VYSMALL=0.3D-38
PARAMETER (VYBIG=1.7D38, VYSMALL=0.3D-38)
DOUBLE PRECISION, PARAMETER :: ZERO=0.D0, ONE=1.D0,
1   TWO=2.D0, THREE=3.0D0, FOUR=4.0D0, FIVE=5.D0, SIX=6.D0,
2   TEN=1.D1
PARAMETER (ZERO=0.D0, ONE=1.D0, TWO=2.D0, THREE=3.0D0, FOUR=4.0D0,
1   FIVE=5.D0, SIX=6.D0, TEN=1.D1)
PARAMETER (ONEPT6=1.6D0, PT52=0.52D0)
PARAMETER (ONEPT6=1.6D0, PT52=0.52D0)
PARAMETER (HALF=5.D-1)
PARAMETER (HALF=5.D-1)
PARAMETER (EIGHTEEN=1.8D+1)
PARAMETER (EIGHTEEN=1.8D+1)
PARAMETER (TENTO6=1.D+6, HALFMI6=5.D+5)
PARAMETER (TENTO6=1.D+6, HALFMI6=5.D+5)

END MODULE PARAMETERS

C pgpltcom.f (edited)
MODULE PGPLTCOM
C-------------------------------------------------- ---------------------
C PGPLOT: module for common variables and parameters for plot routines
C-------------------------------------------------- ---------------------
C Maximum number of concurrent devices (should match GRIMAX).
C-------------------------------------------------- ---------------------
INTEGER PGMAXD
PARAMETER (PGMAXD=8)
C-------------------------------------------------- ---------------------
C Identifier of currently selected device.
C-------------------------------------------------- ---------------------
INTEGER PGID
C-------------------------------------------------- ---------------------
C Device status (indexed by device identifier).
C-------------------------------------------------- ---------------------
INTEGER PGDEVS
C =0 if device is not open; 1 if device is open.
C PGADVS  Set to 0 by PGBEGIN, set to 1 by PGPAGE; used to suppress
C the prompt for the first page.
C PROMPT  If .TRUE., ask user before clearing page; set by PGASK
C and (indirectly) by PGBEGIN, used in PGENV.
C PGBLEV  Buffering level: incremented by PGBBUF, decremented by
C PGBUF.
C PGPFIX  TRUE if PGPAP has been called, FALSE otherwise.
C
INTEGER PGDEVS(PGMAXD), PGADVS(PGMAXD), PGBLEV(PGMAXD)
LOGICAL PGPRMP(PGMAXD), PGPFIX(PGMAXD)

C-------------------------------------------------- ---------------------
C Panel parameters (indexed by device identification).
C-------------------------------------------------- ---------------------
C NX Number of panels in x direction
C NY Number of panels in y direction
C NXC Ordinal number of current X panel
C NYC Ordinal number of current Y panel
C XSZ X dimension of panel (device units)
C YSZ Y dimension of panel (device units)
C PGROWS TRUE if panels are used in row order, FALSE for column order.
C
INTEGER PGNX (PGMAXD), PGNY (PGMAXD)
INTEGER PGNXC (PGMAXD), PGNYC (PGMAXD)
REAL PGXSZ (PGMAXD), PGYSZ (PGMAXD)
LOGICAL PGROWS(PGMAXD)

C-------------------------------------------------- ---------------------
C Attributes (indexed by device identification).
C-------------------------------------------------- ---------------------
C PGFAS fill-area style
C PGCHSZ character height
C PGAHS arrow-head fill style
C PGAHA arrow-head angle
C PGAHV arrow-head vent
C PGTBCI text background color index
C PGMNCI lower range of color indices available to PGGRAY/PGIMAG
C PGMXCI upper range of color indices available to PGGRAY/PGIMAG
C PGITF type of transfer function used by PGGRAY/PGIMAG
C PGHSA hatching line angle
C PGHSS hatching line separation
C PGHSP hatching line phase
C
INTEGER PGFAS (PGMAXD)
REAL PGCHSZ(PGMAXD)
INTEGER PGAHS (PGMAXD)
REAL PGAHA (PGMAXD)
REAL PGAHV (PGMAXD)
INTEGER PGTBCI(PGMAXD)
INTEGER PGMNCI(PGMAXD)
INTEGER PGMXCI(PGMAXD)
INTEGER PGITF (PGMAXD)
REAL PGHSA (PGMAXD)
REAL PGHSS (PGMAXD)
REAL PGHSP (PGMAXD)

C-------------------------------------------------- ---------------------
C Viewport parameters (indexed by device identification); all are device coordinates:
C-------------------------------------------------- ---------------------
C PGXOFF X coordinate of blc of viewport.
C PGYOFF  Y coordinate of blc of viewport.
C PGXVP   X coordinate of blc of viewport, relative to blc of subpage.
C PGYVP   Y coordinate of blc of viewport, relative to blc of subpage.
C PGXLEN  Width of viewport.
C PGYLEN  Height of viewport.
C
REAL   PGXOFF(PGMAXD), PGYOFF(PGMAXD)
REAL   PGXVP  (PGMAXD), PGYVP  (PGMAXD)
REAL   PGXLEN(PGMAXD), PGYLEN(PGMAXD)
C-------------------------------------------------- ---------------------
C Scaling parameters (indexed by device identification):
C-------------------------------------------------- ---------------------
C PGXORG  device coordinate value corresponding to world X=0
C PGYORG  device coordinate value corresponding to world Y=0
C PGXSCL  scale in x (device units per world coordinate unit)
C PGYSCL  scale in y (device units per world coordinate unit)
C PGXPIN  device x scale in device units/inch
C PGYPIN  device y scale in device units/inch
C PGXSP   Character X spacing (device units)
C PGYSP   Character Y spacing (device units)
C
REAL   PGXORG(PGMAXD), PGYORG(PGMAXD)
REAL   PGXSCL(PGMAXD), PGYSCL(PGMAXD)
REAL   PGXPIN(PGMAXD), PGYPIN(PGMAXD)
REAL   PGXSP  (PGMAXD), PGYSP  (PGMAXD)
C-------------------------------------------------- ---------------------
C Window parameters (indexed by device identification); all are world
C coordinate values:
C-------------------------------------------------- ---------------------
C PGXBLC  world X at bottom left corner of window
C PGXTRC  world X at top right corner of window
C PGYBLC  world Y at bottom left corner of window
C PGYTRC  world Y at top right corner of window
C
REAL   PGXBLC(PGMAXD), PGXTRC(PGMAXD)
REAL   PGYBLC(PGMAXD), PGYTRC(PGMAXD)
C-------------------------------------------------- ---------------------
C The following parameters are used in the contouring routines to pass
C information to the action routine. They do not need to be indexed.
C-------------------------------------------------- ---------------------
C TRANS   Transformation matrix for contour plots; copied
C from argument list by PGCONT and used by PGCP.
C
INTEGER PGCINT, PGCMIN
REAL    TRANS(6)
CHARACTER*32 PGCLAB
C-------------------------------------------------- ---------------------
COMMON /PGPLT1/ PGID,PGDEVS,PGADVSS,PGNX, PGNY, PGNXC, PNGYC ,
1   PGXPIN,PGYPIN,PGXSP, PGYSP, PGXSZ, PGYSZ,
2   PGXOFF,PGYOFF,PGXVP, PGYVP, PGXLEN,PGYLEN,PGXORG,PGYORG,
3   PGXSCL, PGSCL, PGXBL, PGTRC, PGBLC, PGTBC, PGYTRC, TRANS,
4   PGPRMP, PGFAS, PGCHSZ, PGBLEV, PGRROWS,
5   PGAHS, PGHA, PGAHV, PGTBCI, PGMNCI, PGMXCI, PGCINT, PGCMIN,
6   PGFFIX, PGITF, PGHSA, PGHSS, PGBSP
COMMON /PGPLT2/ PGCLAB
SAVE /PGPLT1/
SAVE /PGPLT2/
C-------------------------------------------------- ---------------------
END MODULE PGPLTCOM
C-------------------------------------------------- ---------------------
C poplate.f (edited) The parameters are from the P command.
MODULE POPLATE
USE PARAMETERS
C THE PARAMETERS DEFINING THE INITIAL PHASESPACE DISTRIBUTION(S)
DOUBLE PRECISION :: THMIN=-9.D1, THMAX=9.D1, REMIN=-1.D0,
1  REMAX=1.D0, HREPT=1.D0, DITHTH=0.D0, DITHE=0.D0, SBNCH=1.D-1,
2  THOFF=0.D0, EOFF=0.D0, THTRAN=0.D0, ETRAN=0.D0, WINJ=0.D0,
3  PINJ=0.D0, D0TH(LTABL), D0EN(LTABL), TH0(LTABL+1)
C Caution: The bunch duplication arrays are *not* dynamically allocated according to
INTEGER :: KIND=1, IPOP=1, NTH=1, NE=2, NPOINT=2, NREPT=0, IBCKT(LTABL)
C
END MODULE POPLATE
C-------------------------------------------------- ---------------------
C random.f
MODULE RANDOM
USE PARAMETERS
EXTERNAL DLARAN
DOUBLE PRECISION :: DLARAN
! DLARAN is the random number generator, XRAND a quasi-random number
DOUBLE PRECISION XRAND
INTEGER ISEED(4)
! ISEED the starting value for random number generator initially set in
! esinit
C
END MODULE RANDOM
C-------------------------------------------------- ---------------------
C rfp.f (edited) Parameters not described are from the A command.
MODULE RFP
USE PARAMETERS
C PARAMETERS DEFINING THE RF SYSTEMS
DOUBLE PRECISION :: HDECR=1.D0, SDECR=1.D0, PHISLIM=9.5D-1,
1  BHHOLD=0.0, BAHOLD=0.0, HBINIT, SBINIT,
C1, C2 constants in iso-adiabatic voltage curves

INTEGER :: NRF=1, HGCD, HMAX, ISYNC=0,
1 H(NSRC), HW(NSRC), KURVE(NSRC), NTV(NSRC), NTABV(NSRC),
2 KURVF(NSRC), NTF(NSRC), NTABF(NSRC),
3 KURVP(NSRC), NTP(NSRC), NTABP(NSRC),
4 KURVD(NSRC), NTD(NSRC), NTABD(NSRC)

HGCD the greatest common divisor of the rf system harmonic numbers,
HMAX the highest harmonic number, NTV, NTF, NTP number of entries in voltage
table, frequency table, and phase tables respectively
NTABV, NTABF, NTABP, NTABD index of the last used entry in the voltage, frequency,
phase, and offset tables respectively
LOGICAL :: VKON=.TRUE., FRKON=.FALSE., PHKON=.FALSE.,
1 PHSLIP=.FALSE., DKON=.FALSE., VMATCH(NSRC)
LOGICAL :: HOLDBH=.FALSE., HOLDBA=.FALSE., CNTINU(NSRC)

END MODULE RFP

C THE LATTICE PARAMETERS INCLUDING TIME DEPENDENCES
DOUBLE PRECISION :: REQ, GAMTSQ, ALPHA0, ALPHA1=0.D0, ALPHA2=0.D0,
1 ALPHA3=0.D0, TAUINF, W0I, W0F=0.D0, EK0I, EK0F,
2 TI=0.D0, T0=0.D0, TM=0.D0, TF=0.D0,
3 P0I, P0F=0.D0, PI, PF, FRAC=1.D0, DES=0.D0, PIRAD=1.D0,
4 W0IDOT=0.D0, W0FDOT=0.D0, EKIDOT, EKFDOT, P0IDOT=0.D0,
5 P0FDOT=0.D0, PIDOT, PFDOT, THLO, THHI, THRNG, CRA=0.D0, CRB=0.D0,
6 CRC=0.D0, TR3=0.D0, RRANGE,
7 CHGNO=1.D0, EM0CSQ=9.38272029D2, EMCSQ
THLO bottom of calculation range -pi/FRAC, THHI top of calculation range
pi/FRAC, THRNG calculation range 2 pi/FRAC, EMCSQ rest energy of beam particles
INTEGER :: KURVEB=1, NTR=0
LOGICAL :: JNRAMP=.FALSE., EBDRY=.FALSE.
The variables in the former gamjmp.inc
DOUBLE PRECISION GAMPAR(3), ETAJMP
INTEGER :: KINDG=0
LOGICAL :: GMAJMP=.FALSE.

END MODULE RINGP

C ---------------------------------------------------
C
C ringp.f (edited) Parameters not described are from the R command.
MODULE RINGP

USE PARAMETERS

THE LATTICE PARAMETERS INCLUDING TIME DEPENDENCES
DOUBLE PRECISION :: REQ, GAMTSQ, ALPHA0, ALPHA1=0.D0, ALPHA2=0.D0,
1 ALPHA3=0.D0, TAUINF, W0I, W0F=0.D0, EK0I, EK0F,
2 TI=0.D0, T0=0.D0, TM=0.D0, TF=0.D0,
3 P0I, P0F=0.D0, PI, PF, FRAC=1.D0, DES=0.D0, PIRAD=1.D0,
4 W0IDOT=0.D0, W0FDOT=0.D0, EKIDOT, EKFDOT, P0IDOT=0.D0,
5 P0FDOT=0.D0, PIDOT, PFDOT, THLO, THHI, THRNG, CRA=0.D0, CRB=0.D0,
6 CRC=0.D0, TR3=0.D0, RRANGE,
7 CHGNO=1.D0, EM0CSQ=9.38272029D2, EMCSQ
THLO bottom of calculation range -pi/FRAC, THHI top of calculation range
pi/FRAC, THRNG calculation range 2 pi/FRAC, EMCSQ rest energy of beam particles
INTEGER :: KURVEB=1, NTR=0
LOGICAL :: JNRAMP=.FALSE., EBDRY=.FALSE.
The variables in the former gamjmp.inc
DOUBLE PRECISION GAMPAR(3), ETAJMP
INTEGER :: KINDG=0
LOGICAL :: GMAJMP=.FALSE.

END MODULE RINGP

C ---------------------------------------------------
C
CREATED SPARES.F

MODULE SPARES

Contains "spare" variables which are written to history along with everything else. These can be whatever the user wishes; these commons are available to CYCPROG and SHAZAM. Also contains labels for spares used in history plots.

USE PARAMETERS

DOUBLE PRECISION SPARE(NSPARE)
CHARACTER*14 SPLABL(NSPARE)
CHARACTER*7 SPUNIT(NSPARE)

END MODULE SPARES

CREATED SPCHARGE.F (edited) Parameters not described are from the B command.

MODULE SPCHARGE

USE PARAMETERS

PARAMETERS DEFINING THE SPACE CHARGE & WALL IMPEDANCE ENERGY/TURN

DOUBLE PRECISION :: A=5.D-3,B=5.D-2,ENQ=2.D10,TCHGON=0.D0,
1 BWSC,TDELTA,ARTP,VPKFD,VPKTD,VPKHD,BNCHFCT,
2 BETAD,ALPHAD,AOV,B,OMEGA,CAP,EYE0,
3 ZTABL(LTABL,5),RESTBL(IRLLEN,4),RV0(IRLLEN),RV0DOT(IRLLEN)
! BWSC width of bins in charge histogram, TDELTA time step from time domain
! basis table file, ZTABL impedance vs frequency spline table, EVFD collective
! voltage calculated in frequency domain, RESTBL table of simple resonances,
! EVRES collective voltage from special time domain resonance calculation,
! TDTBL time domain basis function, EVTVD collective voltage from time domain
! calculation.
!! Variables not identified here or in command parameter table are intermediate
!! result tables --- no touchee!

INTEGER :: NZ=0,NR=0,NT=0,MSC=1,NBIND=IFFT,NBRES=MAXCVB,
1 NBINTD=MAXCVB
! NZ no of entries in impedance table, NR number of entries in resonance table,
! NT no of entries in time domain basis table
LOGICAL :: SCON=.FALSE.,FDON=.FALSE.,FDSON=.FALSE.,
1 QREZON=.FALSE.,TDON=.FALSE.

END MODULE SPCHARGE

CREATED TIMES.F

MODULE TIMES

A BLOCK TO CONTAIN CPU TIME SINCE START
DOUBLE PRECISION CPUBEG, CPUNOW
INTEGER ITIME, IRCOD1, IRCOD2

C
END MODULE TIMES
C

4.2 Makefile and Compilation Switches

For Unix systems there exists a Makefile which allows the code to be compiled in optional forms. It uses source files that have extensions of .F for subprograms and .f for modules containing global variables. The existing source has preprocessor directives for compiling an optional map where long-term numerical stability is an issue and for Sun and AIX system dependencies. The Makefile is easily modified for debugging or optimized compilation and linking. A representative version follows:

SHELL = /bin/sh
VPATH = ../src : ../modules
# -------------------------------------------------- ------------------
# GNUmakefile for ESME with PGPLOT in Fortran 90 or 95
# GNU make is *required*
# -------------------------------------------------- ------------------
# July 2005
#
# In case of difficulty please visit
#
# http://www-ap.fnal.gov/ESME
#
# or contact
#
# James MacLaclan  AD/MI Fermilab maclachlan@fnal.gov
# or
# Francois Ostiguy AD/AP Fermilab ostiguy@fnal.gov
#
# -------------------------------------------------- ------------------
# The X11 libraries
#
#X11LIBS = -L/usr/X11R6/lib -lx11 -ldl
X11LIBS = -R/usr/openwin/lib -L/usr/openwin/lib -lx11 -lxext -lsocket -lns1 -Bdynamic
# The PGPLOT library
#
PGPLOTLIB = /usr/local/ap/lib/libpgplot.a
#PGPLOTLIB = /usr/local/lib/libpgplot.a
#
# Extra libraries. libg2c.a is needed only if pgplot has been compiled with GNU g77.

# For Solaris F90/95

LOCALIBS =
RUNTIMELIBS = -lf77compat
# The run time library is needed because PGPLOT has been compiled with f77

# For Absoft under linux

#LOCALIBS = /usr/local/libg2c.a /usr/lib/libm.a
#RUNTIMELIBS = /usr/absoft/lib/libU77.a

# For PGI under linux (on heimdall)

#LOCALIBS =
#RUNTIMELIBS = -lpthread

# For gfortran under Solaris also g95 (?)  

#LOCALIBS =
#RUNTIMELIBS = -l/opt/SUNWspro/lib -lf77compat -lfui -lfai -lfai2 -lfsumai -lfprodai -lfsumai2 -lfprodai2

#-------------------------------------------------- -------
# DEFINES
#-------------------------------------------------- -------
# SUN define for SUNOS or generic UNIX
# AIX define for IBM AIX
# DEBUG Debugging. Turns off buffering for plots.
# CVCALC enable calculation of collective potential
# (Time Domain or Freq Domain)
# BNDRY enable boundary tests in inner loop
# MODDEQ use modified map instead of phase angle change at transition
# (see documentation)
#-------------------------------------------------- -------
# Solaris with Sun f95

#-------------------------------------------------- -------
# F95 = f95
#DEFS = -DSUN -DDEBUG -DCVCALC -DBNDRY
DEFS = -DSUN -DCVCALC -DBNDRY
#DEFS = -DSUN -DCVCALC
#DEFS = -DSUN -DBNDRY
#DEFS = -DSUN -DCVCALC -DBNDRY -DMODDEQ
#FFLAGS = -g -C
#FFLAGS = -O5 -fround=nearest -pad=local -dalign -fns=yes -depend=yes -xlibmopt -libmil
FFLAGS = -fast -ftrap=%none
#LDFLAGS = -g
#LDFLAGS = -Bstatic -xildoff -cache64/32/4:8192/512/2
#-------------------------------------------------- -------
# LINUX with ABSOFT COMPILER
#-------------------------------------------------- -------
#F90  = /usr/absoft/bin/f90
#DEFS = -DSUN -DCVCALC -DBNDRY
#DEFS = -DSUN -DCVCALC -DBNDRY
#DEFS = -DSUN -DCVCALC -DBNDRY -DMODDEQ
#FFLAGS  = -f -B108 -O
#LDFLAGS_STATIC = -X-Bstatic
#LDFLAGS =
# -----------------------------------------------
# LINUX with PGI COMPILER (on heimdall)
# -----------------------------------------------
#F95  = /usr/local/pgi/linux86/6.1/bin/pgf95
#DEFS = -DSUN -DCVCALC -DBNDRY
#DEFS = -DSUN -DCVCALC -DBNDRY
#DEFS = -DSUN -DCVCALC -DBNDRY -DMODDEQ
#FFLAGS = -fast -Mbackslash # You should not need this
#FFLAGS = -fast
#FFLAGS = -g -C
#LDFLAGS = -g
#LDFLAGS = -Bstatic -lpthread
# -----------------------------------------------
# Sun (Solaris) UNIX with GNU gfortran COMPILER
# -----------------------------------------------
#F95  = /usr/local/ap/gcc4/bin/gfortran
#DEFS = -DSUN -DCVCALC -DBNDRY
#DEFS = -DSUN -DCVCALC -DBNDRY
#DEFS = -DSUN -DCVCALC -DBNDRY -DMODDEQ
#FFLAGS = -g
#LDFLAGS = -Wl,-rpath,/usr/local/ap/gcc4/lib
# -----------------------------------------------
# UNIX with GNU g95 COMPILER
# -----------------------------------------------
#F95  = /usr/local/ap/gcc4/bin/gcc  # path?
#DEFS = -DSUN -DCVCALC -DBNDRY
#DEFS = -DSUN -DCVCALC -DBNDRY
#DEFS = -DSUN -DCVCALC -DBNDRY -DMODDEQ
#FFLAGS = -O2
#LDFLAGS =
# -----------------------------------------------
# YOU SHOULD NOT HAVE TO CHANGE ANYTHING BELOW THIS LINE !
# -----------------------------------------------

.SUFFIXES:
.SUFFIXES: .c .f .F .o .mod

OBJDIR = ./
FCOMPILE = $(F95) -c $(DEFS) $(FFLAGS)
LINK = $(F95) $(FFLAGS) $(LDFLAGS)

bin_PROGRAMS = esme

# Source files for new PGPLOT graphics
# GRAPHSRC = drawinit.F esinit.F esmain.F esquit.F  \

74
exclbnd.F grafset.F history.F mrplt.F \ 
phplt.F pltcntur.F pltenrgy.F pltfdvc.F \ 
pltfou.F pltphase.F pltresn.F pltspchg.F \ 
plttdbf.F plttdvc.F pltthta.F pltwav.F \ 
pltttab.F select.F

SHAZSRC = gmoments.F pltfrqnc.F pltshaz5.F shazam.F \ 
shazam1.F shazam2.F shazam3.F shazam4.F \ 
shazam9.F

esme_srcs = beamsc.F beedot.F bernst.F bfunct.F \ 
boost.F bncb.F buckit.F chgdist.F \ 
contour.F cycprog.F dbfun.F dbvolt.F \ 
delesc.F display.F dlaran.F dtabout.F \ 
dtblout.F drfv.F eloss.F engfmt.F \ 
evhbfux.F evsbfix.F fampout.F famphst.F \ 
fampwrt.F fdvcoll.F fft.F ffit.F \ 
fftset.F fillbe.F fillbfg.F fillbg.F \ 
fillbp.F fillbpr.F fillbr.F fillbu.F \ 
fillfix.F floline.F flowprg.F fourr.F \ 
freqnc.F ftblout.F gammas.F gcd.F \ 
get.F getascii.F getbin.F getdat.F \ 
glalb.F hpsort.F hiqres.F histwrt.F \ 
intpgam.F inverf.F karve.F lentrue.F \ 
linbkt.F loop.F lowlvl.F match.F \ 
mmbfcr.F memalloc.F memfree.F moments.F \ 
mrinit.F mrsave.F normabsc.F nbis.F \ 
outlb.F outside.F phases.F phasflo.F \ 
phfeed.F point.F polcoeft.F popul8.F \ 
prefix.F prntout.F ptblout.F refcont.F \ 
rprog.F ringpar.F rmoments.F root.F \ 
rtnl.F rffv.F savasci.F save.F \ 
savbin.F septrix.F slipph.F spline.F \ 
strays.F synch.F tablout.F tblscale.F \ 
tdvcoll.F topbtm.F trap.F volts.F \ 
vfeed.F vtblout.F wght.F zpcw.F \ 
ztabout.F ztabsp.F \ 
${(GRAPHSRC)} ${(SHAZSRC)}

esme_modules = \
parameters.f \
bucket.f dynmem.f bktsup.f blankcom.f bunch.f \
const.f current.f curves.f cyclp.f feeds.f \
flowp.f fourir.f grafix.f histcom.f io.f \
mtnrange.f pgpltcom.f plt.f poplate.f random.f \
rpf.f ringp.f shaz2.f shaz5.f shaz7.f \
spar.f spcharge.f

esme_mods = ${esme_modules:.f=.mod}
esme_mods_objs = ${esme_modules:.f=.o}
esme_objs = ${esme_srcs:.F=.o}

.F.o:
$(FCOMPILE) $<

.f.mod:
$(FCOMPILE) $<

all: modules esmF95

esmF95: $(esme_mods_objs) $(esme_objs)
   $(LINK) \n   -o esmF95 $(esme_objs) $(esme_mods_objs) $(PGPLOTLIB) $(RUNTIMELIBS) $(X11LIBS)

modules: $(esme_mods)

clean:
/bin/rm *.o *.mod esmF95

.NOEXPORT:

This example will probably fit most needs but is not a supported item. Individual circumstances are likely to require some modifications. Notice the optional compilation for certain features of the code that involve IF tests in the innermost tracking loop. When running on a pipelined machine, there can be some gain in efficiency in removing these tests if the data is such that they are always negative, i.e., if the corresponding feature is not being used. The use of the alternative single particle map is discussed in Sec. 1.2.

4.3 Sample SHAZAM Code

The following code is interesting because it utilizes nearly every basic feature of the SHAZAM facility, viz., explicit initializing call, call at beginning of iteration, call at end of iteration, use of globals from several modules, and use of a private common for lower level routines. It is the upper level of routines for beam loading correction in which the initialization establishes the cavity impedance, the call at the start of a turn calculates the correction, and the call at the end of the turn applies that correction with a one-turn delay. One feature not exploited in this example is the possibility of returning to the calling routine (TRAP) with the logical TOTRAP set true to request interruption of the tracking. It is in F77, however. It has not been resurrected for the F95 version so far.

SUBROUTINE SHAZAM2(FROMTRAP,TOTRAP)
C
   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
#include "parameters.inc"
#include "blankcom.inc"
#include "ringp.inc"
#include "const.inc"
#include "fourir.inc"
#include "spchg.inc"
#include "rfp.inc"
#include "cyclp.inc"
#include "current.inc"
#include "bunch.inc"
LOGICAL FROMTRAP,TOTRAP,PREMAP
SAVE PREMAP

NAMELIST /BMLD/
  1 GAIN,RSHUNT,QCAV,DTUNE,ICAV,NHALF,A10U8

TOTRAP=.FALSE.

C Initialize stuff fundamental for beam loading
IF(.NOT.FROMTRAP) THEN
  PREMAP=.TRUE.
C Set default parameters
  CALL INITDAT
  SPUNIT(10)=’MeV’
  READ(KINUNIT,BMLD)
  DO 10 I=1,10
    IF(ICAV(I).EQ.0) GO TO 11
  10 NCAVIT=I
  11 NBBL=NBINFFT
     BINBL=THRNG/DBLE(NBBL)
     CALL RFFTI(NBINFFT,WSAVE)
     RETURN
ENDIF
IF(PREMAP) THEN
C Compute voltage; store in EVBL().
  CALL BLVOLT
ELSE
C Apply beam loading voltage to particles
  CALL BLKICK
ENDIF
PREMAP=.NOT.PREMAP
RETURN
END
SUBROUTINE INITDAT
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C Set default values for NAMELIST variables
C
#include "parameters.inc"
#include "shaz2.inc"

C
   NHALF=2
   GAIN=ZERO
   DO 5 I=2,9
   5 ICAV(I)=0
   ICAV(1)=1
   DO 10 I=1,LTABLE
   10 A10U8(I)=ONE
END

Acknowledgement (J. MacLachlan)

ESME has had important fundamental and technical contributions from several people over the years, most notably Peter Lucas (1986 – 87), Steve Stahl (1987 – 92), and Jean-François Ostiguy (1996 – ). Without these collaborators the program would be less versatile and less reliable. To them and several others who have patiently explained their difficulties and criticism I am grateful. It is unlikely that development would have continued without their help and encouragement.
Appendix A

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 global variables (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 embedded in ESME. This graphics driver is modelled after 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. 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. The subroutines called are the standard ESME routines with the exception of mrplot, which has been modified by the inclusion of the file opening code from mrinit. The latter subroutine is not needed for any other function and is therefore omitted. The modified mrplt is also included below following the main post processor ESPOSTP. The following code is several months behind the esmF95 described in the front of this note; the coding is in F77. There have been changes in the program data so that this post-processor needs updates.

Since f77 is a subset of f95, a functional post processor should be obtained by changing \# include “somefile.inc” to USE somefile.f and compiling the result with f95.

```fortran
PROGRAM ESPOSTP
C Postprocessor for mountain range files (UNIT=20), history files (UNIT=9),
C and save files (UNIT=18 or 7) to produce different graphical displays from
C a single run.
C
C (C) Universities Research Association (Fermilab)
C
C See the file COPYRIGHT included in the ESME distribution.
C
C Main Author:    James MacLachlan
C
C files:
```

\(^1\)POSTP=T, O Command, Section 2.2.4.
input - command input
output - informative output, command confirmations, & diagnostics

unit 6 = output
unit 7 - input of previously saved parameters & coordinates
unit 9 - history of parameters for history plotting (optional)
unit 18 - save of data in COMMON for post-processing (optional).
unit 19 - history pairs on file instead of plotted (optional)
unit 20 - histograms of particle # vs theta (for mountain ranges)

USAGE: postp [args [parameters]]
-----

where [args] is one or more of the following

-i : interactive output
-a4 : output on a4 paper
-bw : turn color output off
-D int : debugging message level
-f filename : use filename for input instead of standard input
-h : help
-d driver : PGPlot name for desired graphics driver (default CPS)
-o filename : use filename for input instead of standard output
-p filename : use filename for postscript output instead of default
-V : version number and Copyright info
-A : write intermediate files, e.g. history, in ASCII

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
LOGICAL DONE,ACTION,EXFG
CHARACTER*1 CMND
CHARACTER*74 WORDS

#include "parameters.inc"
#include "version.inc"
#include "grafix.inc"
#include "blankcom.inc"
#include "curves.inc"
#include "io.inc"
#include "plt.inc"

CHARACTER*132 CHARGM,CHDATAF
CHARACTER*128 GRAFIL,HFIL,MRFIL,OHFIL,OGRafil,OMRFIL
DATA IDUM /0/
DATA GRAFIL,HFIL,MRFIL /3*** /
NAMELIST /USEFIL/ GRAFIL, HFIL, MRFIL

Set the backslash character to avoid portability pbs.
Some UNIX compilers interpret backslash as the C
C escape character. PGPLOT interprets it differently.

BS = CHAR(92)

C Call the memory allocation. The argument indicates that the
C array dimensions should be set to the same as the defaults
C for static arrays.

CALL MEMALLOC(.TRUE.)
CALL ESMEINIT

C-- ************************************************ *********
C Return point after execution of a command.
C

10 READ(KINUNIT,2000) CMND,WORDS
2000 FORMAT(A1,A74)

C-- ************************************************ *********

WRITE(KOUTUNIT,2010)
2010 FORMAT(1X,80('+'))

WRITE(KOUTUNIT,2020) CMND,WORDS
2020 FORMAT(' CMND IS ',A1,A74)

GO TO (100,200,500,1200,1300,1800,1900,1950),
     1 INDEX('ZXGODQWHNKY',CMND)

WRITE(KOUTUNIT,
     + FMT='(''THE IMPLEMENTED COMMANDS ARE'')')
WRITE(KOUTUNIT,
     + FMT='('' D: DISPLAY PHASE SPACE WITH CURRENT'',
     + '' PLOT PARAMETERS'')')
WRITE(KOUTUNIT,
     + FMT='('' G: GET A PREVIOUS RESULT FROM TAPE7'')')
WRITE(KOUTUNIT,
     + FMT='('' H: HISTORY OF IMPORTANT PARAMETERS'',
     + '' TURN-BY-TURN'')')
WRITE(KOUTUNIT,
     + FMT='('' K: "KUT" A PIECE OUT OF THE DISTRIBUTION'')')
WRITE(KOUTUNIT,
     + FMT='('' N: PLOT MOUNTAIN RANGES'')')
WRITE(KOUTUNIT,
     + FMT='('' O: OUTPUT OPTIONS'')')
WRITE(KOUTUNIT,
     + FMT='('' Q: QUIT'')')
WRITE(KOUTUNIT,
     + FMT='('' W: WRITE WISE WORDS INTO PRINTED OUTPUT'')')
WRITE(KOUTUNIT,
     + FMT='('' X: PLOT REMAINDER OF TAPE7'')')
C-- Z: REWIND GRAPHICAL OUTPUT TAPE TO BEGINNING (NOW THE
C ONLY WAY TO GET AT A RECORD BEFORE THE CURRENT ONE)

100 REWIND(7)
C Save logical telling whether the graphics package is initialized or not
C and reset it after restoring COMMONs.
SAVND=NODRAW
CALL GET(.TRUE.,ITURN,DONE)
NODRAW=SAVND
POSTP=.FALSE.
WRITE(KOUTUNIT,*) 'DATA FOR TURN ',ITURN,' READ'
GO TO 10

C-- X: JUST PLOT EVERYTHING FROM HERE ON OUT

200 CONTINUE
C Save logical telling whether graphics package is initialized or not and
C 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(ICONTUR.GT.0) CALL REFCONT
IF(DTHCURV.EQ.0 .AND. DECURV.EQ.0) GO TO 220
DO 210 I=1,NC
   CURVE(I,1)=CURVE(I,1)+DTHCURV
   CURVE(I,2)=CURVE(I,2)+DECURV
210 CONTINUE
220 CALL PHPLT
   GO TO 200

C-- G: GET COORDINATES AND MACHINE PARAMETERS

500 WRITE(KOUTUNIT,
   + '(''READING COORDINATES AND MACHINE PARAMETERS'')'
   + CALL GET(.FALSE.,IDUM,DONE)
   POSTP=.FALSE.
   GO TO 10

C-- O: OPTIONS FOR OUTPUT GRAPHICS
700 CALL GRAFSET
    GO TO 10

C-- D: DISPLAY GRAPHICALLY PHASE POINTS AND (OPTIONALLY) BUCKET
C
900 CALL DISPLAY
    GO TO 10

C-- Q: QUIT PROGRAM ENTIRELY; NOTHING FURTHER TO DO
C
1100 CALL ESMEQUIT
C
C-- W: WRITE WISE WORDS OF EXPLANATION INTO PRINTED OUTPUT (come here for blank, also)
C
1200 GO TO 10
C
C-- H: HISTORY OF IMPORTANT PARAMETERS COLLECTED ON TAPE9 EACH STEP
C
1300 CALL HISTORY(.FALSE.)
    GO TO 10
C
C-- N: PLOT "MOUNTAIN RANGE"
C
1800 CALL MRPLT
    GO TO 10
C
C-- K: KUT A PIECE OUT OF DISTRIBUTION ACCORDING TO SEVERAL OPTIONS
C
1900 CALL KARVE
    GO TO 10
C
C-- Y: Memory allocation
C
1950 CONTINUE
    #ifdef DYNMEM
        IF (VALIDP) CALL MEMFREE
    #endif
        CALL MEMALLOC(.FALSE.)
    GO TO 10
C
C-- EOF UNIT 7
C
9999 CONTINUE
    END

SUBROUTINE MRPLT
C
C This subroutine reads data from LUN MRUNIT and plots the results
in a mountain-range (i.e. stacked histogram) format according to the
parameters in NAMELIST MRPLOT.

5-Mar-93 - J. MacLachlan - Add Bernstein polynomial smoothing option

24-Sep-97 - J. A. MacLachlan, Fermilab BD/HQ
(Re-)Installed provisions for Fourier spectrum plots

03-Oct-03 - J. A. MacLachlan, Fermilab BD/AI
Version of standard es2003 code modified for postprocessor

INPUT:

NAMELIST /MRPLOT/

1  MRPMIN,MRPMAX,SCALE,TOPTOB,NTRACE,MSTART,MSTOP,NSKIP,
2  TMSTART,TMSTOP,TBASE,SMOOTH,ITNO,LIM,NRNBIN,OBJWGT,
3  XCRNR,YCRNR,XAXISL,YAXISL

MRPMIN(0.0)-lower ordinate limit of plotting range
MRPMAX(0.0)-upper ordinate limit of plotting range
Defaults of 0.0,0.0 imply that data is to be plotted over the entire range.
SCALE(0.3)-height of first trace (minimum to maximum), where 1.0
is the height of the left axis.
TOPTOB(0.7)-proportion of vertical scale that is to
be occupied by NTRACE traces (approximate if TBASE=T).
NTRACE(100)-number of traces on a page.
MSTART(0)-turn number at which to start plots.
MSTOP(0)-turn number at which to stop plots.
NSKIP(0)-number of recorded traces to skip between plotted traces
TMSTART(0.0)-time at which to start plots.
TMSTOP(0.0)-time at which to stop plots.
The defaults of 0,0,0,0,0.0 imply that all of the data is
to be plotted.
TBASE(F)-if T, trace separation is proportional to the time.
SMOOTH(1)- 0 => no smooth; -1 => 1-2-1 averaging; 1 => Bernstein smoothing
ITNO(0)- number of smoothing/fitting iterations in Bernstein smoothing
LIM(F)-dotted lines are plotted connecting consecutive traces’
leftmost and rightmost non-zero bins; works only for SMOOTH = 1
NRNBIN(400)-The number of points on a smoothed curve for SMOOTH = 1
OBJWGT-relative weight of fitting to smoothing in object function for
Bernstein polynomial smoothing

The mountain ranges can be stacked according to time or turn number.
If TBASE=T, then consecutive traces are spaced proportional to their time.
If TBASE=F, then consecutive traces are equispaced. The time-stacking method
better simulates the mountain ranges normally recorded by an oscilloscope.
However, then the traces may not be equally spaced (due to a change in
circulation frequency, such as would be encountered in a non-relativistic,
accelerating mode of operation). In this case TOPTOB determines the amount
of a page occupied by NTRACE traces only approximately.

Note: in the comments below, "record" refers to a single trace, i.e, time,
turn number, and NBIN bins of data.
This particular version contains the file open code from MRINIT because that subroutine is not otherwise needed for the postprocessor.

```
IMPLICIT NONE
#include "parameters.inc"
#include "grafix.inc"
#include "heading.inc"
#include "mrange.inc"
#include "const.inc"
#include "io.inc"
#include "plt.inc"

REAL TRACE(-1:MAXTRC),BINTH(-1:MAXTRC),EDGE(50,2),EDGHGT(50)
REAL MRMINP,MRMAXP,MRPMIN,MRMAX

DOUBLE PRECISION TRACEO(MAXTRC),TRACE2(MAXTRC),BINTH0(MAXTRC)
DOUBLE PRECISION TRMINUS,TRZERO,TRPLUS,TREC,TRECL,THETA,THMIN,
1         THMAX,FF,OBJWGT,UTBL(MAXTRC)

REAL SCALE,TOPTOB,TMSTART,TMSTOP,XPC,YPC,XAXL,YAXL,
1         BINW,BWDTH,DELMIN,DELMAX,TRMAX,TRMIN,DT1,TSEP1,
2         Y1,Y2,DELTAY,OFFSET,TRASEP
INTEGER SMOOTH,MSTART,MSTOP,NSKIP,ITNO,NRNBIN,MRUNIT,NB1,I,
1         IFIRST,IFST,ILAST,ILST,MRECL,MREC,J,N0,N2,
2         NTRACE,ILEN,ICHTI,NTPLT,KLO,NPTP,NTP,NNO,NN2
INTEGER LNBLNK
LOGICAL PLTALL,PTURN,PTIM,NUPLOT,TBASE,UXTEND,LXTEND
LOGICAL LIM
CHARACTER*9 CHRBEG
CHARACTER*5 CHINT
NAMELIST /MRPLOT/
1         MRPMIN,MRPMAX,SCALE,TOPTOB,NTRACE,MSTART,MSTOP,NSKIP,
2         TMSTART,TMSTOP,TBASE,SMOOTH,ITNO,LIM,NRNBIN,OBJWGT,
3         XCRNR,YCRNR,XAXISL,YAXISL
DATA MRPMIN,MRPMAX /0.0,0.0/
DATA LIM,TBASE / .FALSE., .FALSE. /
DATA SCALE,TOPTOB / .30,.70 /
DATA NTRACE,NSKIP,NRNBIN /100,0,0/
DATA MSTART,MSTOP,TMSTART,TMSTOP,OBJWGT /0,0,0,0,0.0,3.D0/
DATA SMOOTH,ITNO /0,1/

READ(KINUNIT,MRPLOT)

CALL DRAWINIT

C Set corner position and axis lengths.
XPC=XCRNR
YPC=YCRNR
XAXL=XAXISL
YAXL=YAXISL
```
C Open output file.
   IF(ASCII) THEN
      IF(LNBLNK(TEMPFIL).GT.0) THEN
         OPEN(UNIT=20,FILE=TEMPFIL(1:LNBLNK(TEMPFIL))//'.20',
         1 FORM='FORMATTED',ACCESS='SEQUENTIAL',STATUS='UNKNOWN')
      ELSE
         OPEN(UNIT=20,FILE='UNDEFINED'//'.20',
         1 FORM='FORMATTED',ACCESS='SEQUENTIAL',STATUS='UNKNOWN')
      ENDIF
   ELSE
      IF(LNBLNK(TEMPFIL).GT.0) THEN
         OPEN(UNIT=20,FILE=TEMPFIL(1:LNBLNK(TEMPFIL))//'.20',
         1 ACCESS='SEQUENTIAL',STATUS='UNKNOWN',FORM='UNFORMATTED')
      ELSE
         OPEN(UNIT=20,FILE='UNDEFINED'//'.20',
         1 ACCESS='SEQUENTIAL',STATUS='UNKNOWN',FORM='UNFORMATTED')
      ENDIF
   ENDIF
   MRUNIT=20
   REWIND 20
C Read number of bins and ordinate range
   IF(ASCII) THEN
      READ(MRUNIT,*,ERR=900,END=910) NBN,THMIN,THMAX
   ELSE
      READ(MRUNIT,ERR=900,END=910) NBN,THMIN,THMAX
   ENDIF
   BINW=(THMAX-THMIN)/NBN
   IF(NRNBIN.EQ.0) NRNBIN=NBN
   BWDTH=(THMAX-THMIN)/NRNBIN
C Determine theta positions of bins.
   DO 5 I=1,NBN
      BINTH0(I)=THMIN+(I-0.5)*BINW
   5 CONTINUE
   WRITE(KOUTUNIT,800) MRPMIN,MRPMAX
     800 FORMAT(' Plotting limits requested are',1PE11.3,' to',E10.3)
   WRITE(KOUTUNIT,810) THMIN,THMAX
     810 FORMAT(' Data limits are',1PE11.3,' to',E10.3)
C Set theta limits to be used in plotting mountain range.
   IF(MRPMIN.EQ.0.0 .AND. MRPMAX.EQ.0.0) THEN
      MRMINP=THMIN+BWDTH/2.
      MRMAXP=THMAX-BWDTH/2.
      WRITE(KOUTUNIT,*) ' Ordinate limits in mountain range are ',
      1 ' those of data'
   ELSE
      MRMINP=MRPMIN
      MRMAXP=MRPMAX
   ENDIF
C Determine whether the ordinate range is "extended" (i.e, one or c both limits are greater than the limits in data).
   LXTEND=.FALSE.
UXTEND=.FALSE.
IF (MRMINP.LT.THMIN) THEN
  WRITE(KOUTUNIT,*) ' Warning: lower limit specified is out of ', +
  ' data range. Traces will be zero beyond data range.'
LXTEND=.TRUE.
IFIRST=-1
C These are the endpoints of the extension of the trace to left edge of plot.
  BINTH(-1)=MRMINP
  TRACE(-1)=0.0
  BINTH(0)=THMIN-BWDTH/2.
  TRACE(0)=0.0
ELSE
C Find first bin within plot range; use previous bin to plot to limit.
  IFIRST=INT((MRMINP-THMIN)/BWDTH)
  IFST=INT((MRMINP-THMIN)/BINW) + 1
  DELMIN=MRMINP-BINTH0(IFST)
  BINTH(IFIRST)=MRMINP
ENDIF
IF (MRMAXP.GT.THMAX) THEN
  WRITE(KOUTUNIT,*) ' Warning: upper limit specified is out of ', +
  ' data range. Traces will be zero beyond data range.'
UXTEND=.TRUE.
ILAST=NRNBIN+2
C These are the endpoints of the "extension" to the trace
C to the right edge of the plot.
  TRACE(NRNBIN+1)=0.0
  BINTH(NRNBIN+1)=BINTH0(NBN)
  TRACE(NRNBIN+2)=0.0
  BINTH(NRNBIN+2)=MRMAXP
ELSE
C Find last bin within plot range; use next bin to plot to limit.
  ILAST=INT(NRNBIN-(THMAX-MRMAXP)/BWDTH+2)
  ILST=INT(NBN-(THMAX-MRMAXP)/BINW+2)
  DELMAX=BINTH0(ILST)-MRMAXP
  BINTH(ILAST)=MRMAXP
ENDIF
C Determine whether to plot all records (PLTALL=T) or only records from
C MSTART to MSTOP (PTURN=T) or records from TMSTART to TMSTOP (PTIM=T).
IF (MSTART.EQ.0 .AND. MSTOP.EQ.0 .AND. TMSTART.EQ.0.0 .AND. +
  TMSTOP.EQ.0.0) THEN
  WRITE(KOUTUNIT,*) 'Entire mountain range record will be ', 1
  ' plotted'
  PLTALL=.TRUE.
  PTIM=.FALSE.
  PTURN=.FALSE.
ELSE IF (MSTART.NE.0 .OR. MSTOP.NE.0) THEN
  WRITE(KOUTUNIT,*) 'Records will be plotted from turn ',MSTART, +
  ' through turn ',MSTOP
  PTURN=.TRUE.
  PLTALL=.FALSE.
PTIM=.FALSE.
ELSE
    WRITE(KOUTUNIT,*) 'Records will be plotted from time ',
            + 'TMSTART,' through ',TMSTOP
    PTIM=.TRUE.
PLTALL=.FALSE.
PTURN=.FALSE.
ENDIF
IF (KVERBOSE.GT.0) THEN
    WRITE(KOUTUNIT,MRPLOT)
ENDIF
C Advance to "header" of first record to be plotted.
10 CONTINUE
IF(ASCII) THEN
    READ(MRUNIT,*,ERR=900,END=910) TRECL,MRECL
ELSE
    READ(MRUNIT,ERR=900,END=910) TRECL,MRECL
ENDIF
IF(PLTALL .OR. (PTURN .AND. (MRECL.GE.MSTART)) .OR.
+ (PTIM .AND. (TRECL.GE.TMSTART))) THEN
    GO TO 30
ELSE
    C Read though the NBN numbers for this record to get
    C to the next one.
    IF(ASCII) THEN
        DO 20 I=1,NBN
20       READ(MRUNIT,*,ERR=900,END=910) TRACE0(I)
    ELSE
        READ(MRUNIT,ERR=900,END=910) (TRACE0(I), I=1,NBN)
    ENDIF
    GO TO 10
ENDIF
30 CONTINUE
C Now at first record desired. The first two records determine
C the scale for this and subsequent plots.
C Indicate that this is the start of a new plot.
    NUPLOT=.TRUE.
C Read first record into our working array.
    IF(ASCII) THEN
        DO 40 I=1,NBN
40       READ(MRUNIT,*,ERR=900,END=910) TRACE0(I)
    ELSE
        READ(MRUNIT,ERR=900,END=910) (TRACE0(I), I=1,NBN)
    ENDIF
C Smooth data if requested.
    IF(SMOOTH) 42,46,44
    TRMINUS=TRACE0(1)
    DO 43 J=1,ITNO
78     DO 43 I=1,NBN
78
IF(I.LT.NBN) TRPLUS=TRACE0(I+1)
TRZERO=TRACE0(I)
TRACE0(I)=(TRMINUS+2.*TRZERO+TRPLUS)/4.

43 TRMINUS=TRZERO
GO TO 46

44 CALL BERNST(TRACE0,NBN,ITNO,OBJWGT,N0,N2)
C Determine maximum and minimum bin contents.
46 IF(IPU) THEN
   TRMAX=TRACE0(1)
   TRMIN=TRMAX
   DO 50 I=1,NBN
      IF(TRACE0(I).GT.TRMAX) TRMAX=TRACE0(I)
      IF(TRACE0(I).LT.TRMIN) TRMIN=TRACE0(I)
   50 CONTINUE
ELSE
   TRMIN=0.
   IF(FASPEC) TRMAX=1.
   IF(FPSPEC) TRMAX=.5
ENDIF
C Skip NSKIP records.
DO 60 I=1,NSKIP
IF(ASCII) THEN
   READ(MRUNIT,*,ERR=900,END=910) TREC,MREC
   DO 55 J=1,NBN
      READ(MRUNIT,*,ERR=900,END=910) TRACE0(J)
   55 CONTINUE
ELSE
   READ(MRUNIT,ERR=900,END=910) TREC,MREC
   READ(MRUNIT,ERR=900,END=910) (TRACE0(J), J=1,NBN)
ENDIF
60 CONTINUE
C Read second record header into our working array.
IF(ASCII) THEN
   READ(MRUNIT,*,ERR=900,END=915) TREC,MREC
ELSE
   READ(MRUNIT,ERR=900,END=915) TREC,MREC
ENDIF
C DT1 determines subsequent trace separarations for TBASE=T.
DT1=TREC-TRECL
C Compute separation between first two traces. (If traces are equispaced,
C i.e. TBASE=T, then this is the separation between all traces.)
TSEP1=(TRMAX-TRMIN)*TOPTOB/(SCALE*(NTRACE-1))
C Determine vertical axis limits.
IF(TBASE) THEN
   C Subtract off initial time and stamp it as subtitle on every plot.
   C Make subtitle read initial time.
      Y1=0.0
      Y2=(TREC-TRECL)*(NTRACE-1)/TOPTOB
      WRITE(CHRBEG,’(1PE9.2)’) TRECL
   ELSE
      Y1=0.0
   END
Y2 = (MREC - MRECL) * (NTRACE - 1) / TOPTOB
WRITE (CHRBE, '(I9)') MRECL
IF (MRECL.EQ.0) THEN
   ILEN = 1
ELSE
   ILEN = INT (LOG10 (REAL (ABS (MRECL)))) + 1
   IF (ILEN.LT.0) ILEN = ILEN + 1
ENDIF
ENDIF

C Write interval between displayed traces (in turns), into character variable.
WRITE (CHINT, '(I5)') (MREC - MRECL) / (NSKIP + 1)
ICHTI = INT (LOG10 (REAL (ABS ((MREC - MRECL) / (NSKIP + 1))) + 1
DELTAY = Y2 - Y1

C This is the top of the loop over pages (NTRACEs each).
100 CONTINUE

C Clear device (i.e., erase CRT or advance paper).
CALL PGPAGE

C Draw a box around the plot.
   CALL PGSVP (VPLFT, VPRGT, VPBTM, VPTOP)
   CALL PGSWIN (MRMINP, MRMAXP, TRMIN, (TRMAX - TRMIN) / SCALE + TRMIN)
   CALL PGBOX ('ABCGNPTS’, 0.0, 0.0, ’ABCG’, 0.0, 0.0)
   IF (.NOT. PLTSW (7)) THEN
      CALL PGIDEN
      CALL PGSCH (0.50)
      CALL PGMTXT ('B’, 3.5 / 0.5, 0.0, 0.0, ’ESME2003’)
   C
   CALL PGSCH (CTSIZ)
   CALL PGMTXT ('T’, 9.0 / CTSIZ, 0.5, 0.5, TITL (1: TITLEN))
   CALL PGSCH (CSSIZ)
   IF (TBASE) THEN
      CALL PGMTXT ('T’, 7.5 / CSSIZ, 0.5, 0.5, 'every' / CHINT (6-ICHTI: 5) //
      'turns, from' // CHRBE (1: 9) // 'sec')
   ELSE
      CALL PGMTXT ('T’, 7.5, 0.5, 0.5,
      'every' / CHINT (6-ICHTI: 5) //
      'turns, from turn' // CHRBE (10-ILEN: 9))
   ENDIF
   C
   C Label horizontal axes
   IF (IPU) CALL PGLAB (BS // 'gh [degree]',
   'Beam Current Profiles')
   IF (FASPEC)
   + CALL PGLAB ('Harmonic Number', 'Fourier Spectra')
   IF (FPSPEC)
CALL PGLAB('Harmonic Number', ' ', 'Power Spectra')
ENDIF

C
C Use solid line.
CALL PGSLSL(1)
C For a new page, set OFFSET to -TSEP1.
C TSEP1 will be added to this for first trace.
OFFSET=-TSEP1
NTPLT=0
200 CONTINUE
C Convert TRACE0 at tabular interval BINW to TRACE at interval BWDTH
IF(NBN.EQ.NRNBIN) THEN
DO 210 I=1,NBN
BINTH(I)=BINTH0(I)
210 TRACE(I)=TRACE0(I)
ELSE
CALL SPLINE (BINTH0, TRACE0, NBN, VYBIG, VYBIG, TRACE2, UTLB)
THETA=BINTH0(1)
KLO=1
DO 220 I=1,NRNBIN
CALL TABLOUT (BINTH0, TRACE0, TRACE2, NBN, THETA, FF, KLO)
THETA=THETA+BWDTH
BINTH(I)=THETA
220 TRACE(I)=FF
ENDIF
C Plot traces
IF(.NOT.LXTEND) THEN
IF(IFIRST.EQ.0) THEN
C Just extend first value to edge of plot if in zeroth bin.
TRACE(0)=TRACE(1)
BINTH(0)=MRMINP
ELSE
C Interpolate between trace values at IFIRST and IFIRST+1 to set ordinate
C at edge of plot.
TRACE(IFIRST)=TRACE(IFIRST)+DELMIN*
+(TRACE(IFIRST+1)-TRACE(IFIRST))/BWDTH
ENDIF
ENDIF
IF(.NOT.UXTEND) THEN
IF(ILAST.EQ.(NRNBIN+1)) THEN
TRACE(NRNBIN+1)=TRACE(NRNBIN)
ELSE
C Interpolate between trace values at ILAST and ILAST-1 to set ordinate
C at edge of plot.
TRACE(ILAST)=TRACE(ILAST)-DELMAX*
+(TRACE(ILAST)-TRACE(ILAST-1))/BWDTH
ENDIF
ENDIF
IF(TBASE.AND.NTPLT.NE.0) THEN
C Trace separation based on time if plotting vertically according to time.
TRASEP=DT1/(TREC-TRECL)*TSEP1
ELSE
   TRASEP=TSEP1
ENDIF
OFFSET=OFFSET+TRASEP
NPTP=ILAST-IFIRST+1
C
DO 150 I=IFIRST,ILAST
   TRACE(I)=TRACE(I)+OFFSET
150 CONTINUE
CALL PGLINE(NPTP-2,BINTH(IFIRST),TRACE(IFIRST))
NTPLT=NTPLT+1
IF(LIM .AND. SMOOTH.EQ.1) THEN
   IF(NTPLT.LT.50) THEN
      NN0=FLOAT(NRNBIN*(N0-2))/FLOAT(NBN)
      NN2=FLOAT(NRNBIN*(N2+1))/FLOAT(NBN)
      EDGE(NTPLT,1)=BINTH(NN0)
      EDGE(NTPLT,2)=BINTH(NN2)
      EDGHGT(NTPLT)=OFFSET
   ENDIF
ENDIF
ENDIF
IF(.NOT.NUPLT) THEN
C If starting a plot, then TREC and MREC already read. Otherwise, need to
read them (and store the last record’s numbers in TRECL and MRECL).
   TRECL=TREC
   MRECL=MREC
C Skip NSKIP records.
   DO 260 I=1,NSKIP
      IF(ASCII) THEN
         READ(MRUNIT,*,ERR=900,END=910) TREC,MREC
         DO 255 J=1,NBN
            READ(MRUNIT,*,ERR=900,END=910) TRACE0(J)
         ELSE
            READ(MRUNIT,ERR=900,END=910) TREC,MREC
            READ(MRUNIT,ERR=900,END=910) (TRACE0(J), J=1,NBN)
         ENDIF
255 CONTINUE
   ELSE
      READ(MRUNIT,*,ERR=900,END=910) TREC,MREC
      READ(MRUNIT,ERR=900,END=910) (TRACE0(J), J=1,NBN)
   ENDIF
260 CONTINUE
   IF(ASCII) THEN
      READ(MRUNIT,*,ERR=900,END=920) TREC,MREC
   ELSE
      READ(MRUNIT,ERR=900,END=920) TREC,MREC
   ENDIF
ELSE
   NUPLOT=.FALSE.
ENDIF
C Set NUPLT to FALSE once its been read as TRUE.
C If this is desired record, read it.
   IF(PLTALL .OR. (PTURN .AND. (MREC.LE.MSTOP)) .OR.
   + (PTIM .AND. (TREC.LE.TMSTOP))) THEN
      IF(ASCII) THEN
         READ(MRUNIT,*,ERR=900,END=930) TREC,MREC
      ELSE
         READ(MRUNIT,ERR=900,END=930) TREC,MREC
      ENDIF
   ELSE
      READ(MRUNIT,ERR=900,END=930) TREC,MREC
   ENDIF
ELSE
C
DO 270 I=1,NBN
270 READ(MRUNIT,*,ERR=900,END=915) TRACE0(I)
ELSE
   READ(MRUNIT,ERR=900,END=915) (TRACE0(I), I=1,NBN)
ENDIF
C Smooth data if requested.
IF(SMOOTH) 282,286,284
282 TRMINUS=TRACE0(1)
DO 283 J=1,ITNO
   DO 283 I=1,NBN
      IF(I.LT.NBN) TRPLUS=TRACE0(I+1)
      TRZERO=TRACE0(I)
      TRACE0(I)=(TRMINUS+2.*TRZERO+TRPLUS)/4.
   TRMINUS=TRZERO
   GO TO 286
284 CALL BERNST(TRACE0,NBN,ITNO,OBJWGT,N0,N2)
ELSE
C Done plotting.
RETURN
ENDIF
286 IF(NTPLT.LT.NTRACE) THEN
C If fewer than NTRACE traces plotted, plot this.
   GO TO 200
ELSE
C Set vertical axis limits for next page.
   IF(TBASE) THEN
      Y1=TREC
   ELSE
      Y1=MREC
   ENDIF
   Y2=Y1+DELTAY
C Plot the boundary curves if requested
   IF(LIM .AND. SMOOTH.EQ.1) THEN
      IF(NTPLT.GT.50) THEN
         NTP=50
      ELSE
         NTP=NTPLT
      ENDIF
      CALL PGSLS(2)
      CALL PGLINE(NTP,EDGE(1,1),EDGHGHT)
      CALL PGLINE(NTP,EDGE(1,2),EDGHGHT)
      CALL PGSLS(1)
   ENDIF
C
C Start a new page.
   GO TO 100
ENDIF
C Error or EOF condition:
900 CONTINUE
WRITE(KOUTUNIT,*) ’***** MRPLT: Error reading input file’, 
1          MREC,NSKIP,NBN  
RETURN
910 CONTINUE
WRITE(KOUTUNIT,*) ’***** MRPLT: End of file reached; at least ’, 
1           ’2+NSKIP records must be present for a plot.’, 
2          MREC,NSKIP,NBN
GO TO 930
915 CONTINUE
WRITE(KOUTUNIT,*) ’***** MRPLT: End of file reached during ’, 
1      ’record read’,MREC,NSKIP,NBN
GO TO 930
920 CONTINUE
WRITE(KOUTUNIT,*) ’End of file reached (MREC,NSKIP,NBN):’, 
1          MREC,’,’NSKIP,’,’NBN
C  Plot the boundary curves if requested
930 IF(LIM .AND. SMOOTH.EQ.1) THEN
  IF(NTPLT.GT.50) THEN
    NTP=50
  ELSE
    NTP=NTPLT
  ENDIF
  CALL PGSLS(2)
  CALL PGLINE(NTP,EDGE(1,1),EDGHGHT)
  CALL PGLINE(NTP,EDGE(1,2),EDGHGHT)
  CALL PGSLS(1)
ENDIF
RETURN
END
## Appendix B

### Command Summary

#### A Command, Namelist /RF/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRF</td>
<td>1</td>
<td>-</td>
<td>Number of active RF sources&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>H(1:10)</td>
<td>1</td>
<td>-</td>
<td>Harmonic numbers of sources (integers); a negative value forces a choice of the principal system.</td>
</tr>
<tr>
<td>HW(1:10)</td>
<td>1</td>
<td>-</td>
<td>Voltage sources will be expressed over the $\theta$ range $-180^\circ$/HW $\leq \theta \leq 180^\circ$/HW to produce isolated buckets or barriers.</td>
</tr>
<tr>
<td>ISYNC</td>
<td>0</td>
<td>-</td>
<td>Indicates synchronism condition to be imposed on RF:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 – None, voltages and phases remain as programmed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 – Phase of RF waveform shifted to synchronous, stable point</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 – Magnitude of RF waveform scaled to give correct synchronous energy gain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 – Source 2 Landau cavity to source 1, synchronism assured only for sources 1 and 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 – Determines synchronous phase for source 1 only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 – Determines synchronous phase for the principal source&lt;sup&gt;b&lt;/sup&gt; only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6 – Determines synchronous phase for source 1 wrt B-dot only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7 – Determines synchronous phase for principal source wrt B-dot only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 – Like ISYNC=1 except energy loss to real impedance accounted for</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9 – Like ISYNC=1 except voltages scaled to account for loss to real Z&lt;sub&gt;f&lt;/sub&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10 – Like ISYNC=2 except energy loss to real impedance accounted for</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11 – Like ISYNC=3 except energy loss to real impedance accounted for</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12 – Like ISYNC=4 except energy loss to real impedance accounted for</td>
</tr>
<tr>
<td>VI(1:10)</td>
<td>0.0</td>
<td>MV</td>
<td>Voltage ($&gt;0$) of source I at time TVBEG(I)</td>
</tr>
<tr>
<td>VF(1:10)</td>
<td>0.0</td>
<td>MV</td>
<td>Voltage ($&gt;0$) of source I at time TVEND(I)</td>
</tr>
<tr>
<td>TVBEG(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to beginning of RF voltage change</td>
</tr>
<tr>
<td>TVEND(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to end of RF voltage change</td>
</tr>
<tr>
<td>TV0(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to VI if different from TVBEG</td>
</tr>
<tr>
<td>TVM(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to VF if different from TVEND</td>
</tr>
<tr>
<td>KURVE(1:10)</td>
<td>0</td>
<td>-</td>
<td>Specifies type of RF voltage variation between times TVBEG and TVEND:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 – None, voltage maintained at VI(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 – Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 – Iso-adiabatic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 – Sigmoid</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 – Fit and cubic spline interpolation of voltage table&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 – Simple parabola, initial slope zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6 – Fit and Bernstein polynomial interpolation of voltage table&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
<tr>
<td>NTV(1:10)</td>
<td>0</td>
<td></td>
<td>Number of points to be taken from voltage table (KURVE=4,6)</td>
</tr>
<tr>
<td>VKON</td>
<td>T</td>
<td>-</td>
<td>Indicates whether programmed voltage curves are to be active</td>
</tr>
<tr>
<td>PSII(1:10)</td>
<td>0</td>
<td>deg</td>
<td>Phase of source I at time TPBEG(I)</td>
</tr>
<tr>
<td>PSIF(1:10)</td>
<td>0</td>
<td>deg</td>
<td>Phase of source I at time TPEND(I)</td>
</tr>
<tr>
<td>TPBEG(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to beginning of RF phase change</td>
</tr>
<tr>
<td>TPEND(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to end of RF phase change</td>
</tr>
<tr>
<td>TPO(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to PSII if different from TPBEG</td>
</tr>
<tr>
<td>TPM(1:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to PSIF if different from TPEND</td>
</tr>
<tr>
<td>KURVP(1:10)</td>
<td>0</td>
<td>-</td>
<td>Specifies type of RF phase variation between times TPBEG and TPEND:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 – None, phase maintained at PSII(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 – Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 – Quadratic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 – Fit and cubic spline interpolation of phase table&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 – Fit and Bernstein polynomial interpolation of phase table&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
<tr>
<td>NTP(1:10)</td>
<td>0</td>
<td></td>
<td>Number of points to be taken from phase table (KURVP=4,5)</td>
</tr>
<tr>
<td>PHKON</td>
<td>F</td>
<td>-</td>
<td>Indicates whether or not phase curves are to be active</td>
</tr>
</tbody>
</table>

<sup>a</sup>The permissible number of systems is set by the parameter NSRC in the parameters.f module.

<sup>b</sup>that is, the one taken alone which produces the greatest bucket height

<sup>c</sup>Fit to values read from file. See Section 3.2.2.
### A Command (cont)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Default Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRI(I:10)</td>
<td>0.0</td>
<td>MHz</td>
<td>Frequency of source I at time TFBEG(I)</td>
</tr>
<tr>
<td>FRF(I:10)</td>
<td>0.0</td>
<td>MHz</td>
<td>Frequency of source I at time TFEND(I)</td>
</tr>
<tr>
<td>TFBEG(I:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to beginning of frequency change</td>
</tr>
<tr>
<td>TFEND(I:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to end of frequency change</td>
</tr>
<tr>
<td>TFR(I:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to FRF if different from TFBEG</td>
</tr>
<tr>
<td>TFM(I:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to FRI if different from TFEND</td>
</tr>
<tr>
<td>KURVF(I:10)</td>
<td>0 -</td>
<td></td>
<td>Specifies type of frequency variation between times TFBEG and TFEND:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - None, frequency maintained at FRI(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Quadratic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 - Sigmoid</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 - Fit and cubic spline interpolation of frequency table(^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 - Fit and Bernstein polynomial interpolation of frequency table(^a)</td>
</tr>
<tr>
<td>NTF(I:10)</td>
<td>0</td>
<td></td>
<td>Number of points to be taken from frequency table (KURVF=4,5)</td>
</tr>
<tr>
<td>FRKON</td>
<td>F</td>
<td></td>
<td>Indicates whether frequency curves are to be active</td>
</tr>
<tr>
<td>DELTRFI(I:10)</td>
<td>0.0</td>
<td>MeV/c</td>
<td>Energy offset of source I at time TDBEG(I)</td>
</tr>
<tr>
<td>DELTRFF(I:10)</td>
<td>0.0</td>
<td>MeV/c</td>
<td>Energy offset of source I at time TDEND(I)</td>
</tr>
<tr>
<td>TDBEG(I:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to beginning of momentum offset change</td>
</tr>
<tr>
<td>TDEND(I:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to end of momentum offset change</td>
</tr>
<tr>
<td>TD0(I:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to DELTRFI if different from TDBEG</td>
</tr>
<tr>
<td>TD0(I:10)</td>
<td>0.0</td>
<td>s</td>
<td>Time corresponding to DELTRFF if different from TDEND</td>
</tr>
<tr>
<td>KURVD(I:10)</td>
<td>0 -</td>
<td></td>
<td>Specifies type of momentum offset variation between times TDBEG and TDEND:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - None, frequency maintained at DELTRFI(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Quadratic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 - Sigmoid</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 - Fit and cubic spline interpolation of momentum offset table(^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 - Fit and Bernstein polynomial interpolation of momentum offset table(^a)</td>
</tr>
<tr>
<td>NTD(I:10)</td>
<td>0</td>
<td></td>
<td>Number of points to be taken from momentum offset table (KURVD=4,5)</td>
</tr>
<tr>
<td>DKON</td>
<td>F</td>
<td></td>
<td>Indicates whether momentum offset curves are to be active</td>
</tr>
<tr>
<td>PILCRV</td>
<td>'DUMMY'</td>
<td></td>
<td>Full path for rf curves file</td>
</tr>
<tr>
<td>CONTINU(I:10)</td>
<td>F</td>
<td></td>
<td>Sets the starting voltage, phase and/or frequency for the corresponding sources to the current values — for smoothly piecing curve segments together</td>
</tr>
<tr>
<td>VMATCH(I:10)</td>
<td>F</td>
<td></td>
<td>(V_{\text{MATCH}}(I) = T ) sets (V(I)) so that source I matches the emittance of the current distribution(^b)</td>
</tr>
<tr>
<td>HOLDBH</td>
<td>F</td>
<td></td>
<td>Switch to calculate the voltage of principal source(^d) to hold the current bucket height. (see also HDECRI)</td>
</tr>
<tr>
<td>HDECRI</td>
<td>1.0</td>
<td></td>
<td>Factor by which bucket height for principal source(^e) is to be adjusted on successive turns if HOLDBH = T</td>
</tr>
<tr>
<td>BHHOLD</td>
<td>0.0</td>
<td>MeV</td>
<td>If not zero, calculation triggered by HOLDBH will aim for BHHOLD</td>
</tr>
<tr>
<td>HOLDBA</td>
<td>F</td>
<td></td>
<td>Switch to calculate the voltage of principal source(^d) to hold the current bucket area. (see also SDECRI)</td>
</tr>
<tr>
<td>SDECRI</td>
<td>1.0</td>
<td></td>
<td>Factor by which bucket area for principal source(^e) is to be adjusted on successive turns if HOLDBA = T</td>
</tr>
<tr>
<td>BAHOLD</td>
<td>0.0</td>
<td>eVs</td>
<td>If not zero, calculation triggered by HOLDBA will aim for BAHOLD</td>
</tr>
<tr>
<td>PHISLIM</td>
<td>.95</td>
<td></td>
<td>Voltage may not be reduced such that (\sin \phi_s &gt; \text{PHISLIM}) using options HOLDBH and HOLDBA</td>
</tr>
<tr>
<td>PHSLIP</td>
<td>F</td>
<td></td>
<td>Switch indicating that the phase of at least one source is to be varied to correspond to an energy offset from the synchronous value (see DELTRFI/F)</td>
</tr>
</tbody>
</table>

\(^a\)Fit to values read from a file. See Section 3.2.2.

\(^b\)Which means, in this instance, that a P command, or its equivalent, should precede the A command.

\(^c\)The constant bucket area or bucket height is calculated from the principal rf source.
### B Command, Namelist /SCHG/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.005 m</td>
<td>Effective beam radius</td>
</tr>
<tr>
<td>B</td>
<td>0.05 m</td>
<td>Effective beampipe radius</td>
</tr>
<tr>
<td>ENQ</td>
<td>$2 \cdot 10^{14}$</td>
<td>Number of ions (of Z=CHGNO, see R command) to be represented by the distribution</td>
</tr>
<tr>
<td>NBINS</td>
<td>100</td>
<td>Number of bins for histogram of charge distribution</td>
</tr>
<tr>
<td>MSC</td>
<td>1</td>
<td>Collective effects are to be calculated $\text{MSC \times}$ times between rf cavities</td>
</tr>
<tr>
<td>TCHGON</td>
<td>0 s</td>
<td>&gt; 0: Time starting from 0 in which charge is ramped from 0 to ENQ &lt; 0: Time starting from 0 in which charge is ramped from ENQ to 0</td>
</tr>
<tr>
<td>SCON</td>
<td>F</td>
<td>Activate time domain perf. cond. wall calculation</td>
</tr>
<tr>
<td>TDON</td>
<td>F</td>
<td>Activate time domain wake field calculation with supplied response function</td>
</tr>
<tr>
<td>FDON</td>
<td>F</td>
<td>Activate frequency domain wall impedance calculation</td>
</tr>
<tr>
<td>FDSCON</td>
<td>F</td>
<td>Activate frequency domain calculation of perf. cond. wall voltage. Requires FDON = T also.</td>
</tr>
<tr>
<td>QREZON</td>
<td>F</td>
<td>Activate time domain calculation for high-Q resonance</td>
</tr>
<tr>
<td>NBINFFT</td>
<td>256</td>
<td>Number of bins to be used in Fourier transform</td>
</tr>
<tr>
<td>MFFT</td>
<td>1</td>
<td>Interval (in turns) between Fourier transforms</td>
</tr>
<tr>
<td>NNF</td>
<td>0</td>
<td>Number of Fourier harmonics to be stored in history</td>
</tr>
<tr>
<td>NFO(1:NNF)</td>
<td>0</td>
<td>Harmonic numbers of Fourier spectrum components to be stored</td>
</tr>
<tr>
<td>NBRES</td>
<td>1000</td>
<td>Number of time slices for time domain solution of high-Q resonator</td>
</tr>
<tr>
<td>NIXNOIS</td>
<td>0</td>
<td>Three-way switch to control smoothing of charge distribution $1 \Rightarrow 1$ no smoothing measures $1 \Rightarrow$ Bernstein polynomial smoothing</td>
</tr>
<tr>
<td>TITN1</td>
<td>1</td>
<td>Number of iterations for either 1-2-1 or Bernstein smoothing</td>
</tr>
<tr>
<td>OBWT</td>
<td>3</td>
<td>Weight of fitting vs. smoothing in object function for Bernstein smoothing</td>
</tr>
<tr>
<td>FILIMP</td>
<td>'DUMMY'</td>
<td>Full path for file containing the impedance table</td>
</tr>
<tr>
<td>PILRES</td>
<td>'DUMMY'</td>
<td>Full path for file containing the resonance list</td>
</tr>
<tr>
<td>PLITDB</td>
<td>'DUMMY'</td>
<td>Full path for file containing time domain response basis</td>
</tr>
</tbody>
</table>

*The number of such calculations per turn will be $\text{MSC \times RSCALE}$; RSCALE is set by the T command.

*b* see Appendix 3.2.6

*c* For instruction in the use of NF(1:NNF), see the description following the F command.

### C Command, Namelist /FLOW/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES</td>
<td>2</td>
<td>The number of flow lines to be drawn</td>
</tr>
<tr>
<td>STVAL</td>
<td>0.3 $^\circ$</td>
<td>$\theta$, $E$ starting values for contours</td>
</tr>
<tr>
<td>STATIC</td>
<td>.TRUE.</td>
<td>Contours drawn with parameters fixed at initial values for STATIC = .TRUE.; parameters vary according to their programs for STATIC = .FALSE.</td>
</tr>
<tr>
<td>PLTKBT</td>
<td>.TRUE.</td>
<td>Option switch to include bucket contour on flow line plot</td>
</tr>
<tr>
<td>TSTOP</td>
<td>0. s</td>
<td>End of interval over which maps are generated</td>
</tr>
<tr>
<td>TTRACK</td>
<td>0. s</td>
<td>Time over which periodic flow maps will be generated</td>
</tr>
<tr>
<td>ACCEL0</td>
<td>1.0</td>
<td>Number of beam turns per step between mappings</td>
</tr>
<tr>
<td>PARTION</td>
<td>T</td>
<td>Switch indicates if points on separate flow lines belong to different partitions of the phase points</td>
</tr>
<tr>
<td>ITRAP(1:4)</td>
<td>1</td>
<td>Flags a condition for which the C command should be interrupted; see T command for values</td>
</tr>
<tr>
<td>ETAI'</td>
<td>.001</td>
<td>Trapping parameter; see T command</td>
</tr>
<tr>
<td>PHISTRP</td>
<td>.95</td>
<td>Trapping parameter; see T command</td>
</tr>
<tr>
<td>MOCRACE</td>
<td>0</td>
<td>Trapping parameter; see T command</td>
</tr>
</tbody>
</table>

*a* Starting values for contours have units $\text{degrees,MeV}$. 

*b* TSTOP set to 0. when calculation complete or interrupted by an ITRAP option. 

*c* TSTOP takes precedence; if TSTOP=0., TTRACK determines duration.
## F Command, Namelist /FFT/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFTON</td>
<td>F</td>
<td>Activate Fourier transform calculation</td>
</tr>
<tr>
<td>FFTOUT</td>
<td>F</td>
<td>If TRUE, Fourier transform is printed(^a)</td>
</tr>
<tr>
<td>FFTWRT</td>
<td>F</td>
<td>If TRUE, Fourier transform is written to Fortran UNIT=21(^a)</td>
</tr>
<tr>
<td>NBINFFT</td>
<td>256</td>
<td>Number of bins to be used in FFT</td>
</tr>
<tr>
<td>NNF</td>
<td>0</td>
<td>Number of Fourier harmonics to be stored in history</td>
</tr>
<tr>
<td>NFF(NNF)</td>
<td>0</td>
<td>Harmonic numbers of Fourier spectrum components to be stored</td>
</tr>
<tr>
<td>MFFT</td>
<td>1</td>
<td>Frequency of Fourier transform calculation</td>
</tr>
<tr>
<td>NIXNOIS</td>
<td>0</td>
<td>Three-way switch to control smoothing of azimuthal histogram</td>
</tr>
<tr>
<td></td>
<td>-1 =&gt; 1-2-1 averaging of adjacent bins</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 =&gt; no smoothing measures</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 =&gt; Bernstein polynomial smoothing</td>
<td></td>
</tr>
<tr>
<td>ITKNT</td>
<td>1</td>
<td>Number of iterations for either 1-2-1 or Bernstein smoothing</td>
</tr>
<tr>
<td>OBWT</td>
<td>3.0</td>
<td>Weight of fitting vs. smoothing in object function for Bernstein smoothing</td>
</tr>
</tbody>
</table>

\(^a\)Careful! A large volume of output can be produced if MFFT is too small.  
\(^b\)A large output volume can result when MFFT is too small.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPLT(1:2,1:50)</td>
<td>0</td>
<td>-</td>
<td></td>
<td>Index of element in history records; NPLT(1,J) is independent variable, NPLT(2,J) is dependent variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Real records:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 – Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 – PD0DT, the rate of momentum change proportional to B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 – PDOT, dp/dt</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 – THBAR, the mean value of ( \theta ) for the distribution</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5 – EBAR, the average energy of the distribution</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6 – THRMS, the rms spread in ( \theta ) of the particles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7 – ERMS, the rms energy spread</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8 – ES, the synchronous energy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9 – E0, the energy on the reference orbit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10 – ES-E0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11 – THREF, the azimuth of a particle tracked from (0,ES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12 – EREF, the energy of a particle tracked from (0,ES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13 – EPSILON, the emittance(^b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14 – NUS, the synchrotron frequency</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15 – SBCKT, the RF bucket area for the principal system(^c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16 – HBCKT, the RF “bucket” height for the principal system</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17 – ETA, ( \gamma^{-2} - \gamma^{-2} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18 – RSCALE; see T command</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19 – TAU, synchronous revolution period</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20 – PSIADD, phase feedback; see L command</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21 – DAMPL, voltage feedback factor; see L command</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22 – DELR, synchronous orbit radius - reference orbit radius</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>23 – VPKFD, peak voltage from collective potential in frequency domain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24 – VPKTD, peak voltage from collective potential in time domain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25 – VPKHQ, peak voltage from resonances in time domain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26 – BNCHFCT, bunching factor ( \langle I_b \rangle/\bar{I_b} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>27 – DELEIMP, energy loss per turn to real part of longitudinal impedance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>28 – PDOT-DP0DT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>29 – DPOP, relative momentum spread of distribution</td>
</tr>
<tr>
<td>NWRT(1:2,1:50)</td>
<td>0</td>
<td>-</td>
<td></td>
<td>Indices of elements in history records; works like NPLT but writes records to FORTRAN UNIT 19</td>
</tr>
<tr>
<td>XCRNR</td>
<td>135.0</td>
<td>0.001</td>
<td></td>
<td>Fraction of full width of plot frame between lefthand edge and left side of plot</td>
</tr>
<tr>
<td>YCRNR</td>
<td>69.0</td>
<td>0.001</td>
<td></td>
<td>Fraction of full height of plot frame between bottom edge and bottom of plot</td>
</tr>
<tr>
<td>XAXISL</td>
<td>750.0</td>
<td>0.001</td>
<td></td>
<td>Plot width as a fraction of frame width</td>
</tr>
<tr>
<td>YAXISL</td>
<td>475.0</td>
<td>0.001</td>
<td></td>
<td>Plot height as a fraction frame height</td>
</tr>
</tbody>
</table>

\(^a\)The default value of 0 indicates to SUBROUTINE HISTORY that all of the desired history plots have been generated, so only the first set of consecutive nonzero entries to array NPLT or NWRT will select data.

\(^b\)EPSILON = ANORM \( \tau \left( \sqrt{\sum (\theta_i - \bar{\theta})^2 \sum (E_i - \bar{E})^2 - \sum (\theta_i - \bar{\theta})(E_i - \bar{E})} \right)/N \) eVs; see text for ANORM.

\(^c\)The principal system is the one giving the greatest bucket height.
### K Command, Namelist /KUTS/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KUT</td>
<td>0</td>
<td>-</td>
<td>Cut the last KUT particles from the distribution</td>
</tr>
<tr>
<td>KNTSET</td>
<td>0</td>
<td>-</td>
<td>Reset particle count</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(&lt; 0 \implies \text{KNTSC} = 0); removes macroparticles generating collective potential</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(&gt; 0 \implies \text{KWEERKNT} \implies \text{KOUNT}; ) removes entire distribution</td>
</tr>
<tr>
<td>K1</td>
<td>0</td>
<td>-</td>
<td>Starting point for a partial removal of the distribution</td>
</tr>
<tr>
<td>K2</td>
<td></td>
<td></td>
<td>End point for partial removal of distribution</td>
</tr>
<tr>
<td>KLASS</td>
<td>0</td>
<td>-</td>
<td>Selects a partition of the distribution to be removed</td>
</tr>
<tr>
<td>KICKER</td>
<td>FALSE.</td>
<td>-</td>
<td>Kicker is to remove beam between TH1 and TH2</td>
</tr>
<tr>
<td>NOTCHER</td>
<td>FALSE.</td>
<td>-</td>
<td>Notcher is to remove beam below TH1 and above TH2</td>
</tr>
<tr>
<td>TH1</td>
<td>0. degree</td>
<td></td>
<td>Lower limit for KICKER or NOTCHER</td>
</tr>
<tr>
<td>TH2</td>
<td>0. degree</td>
<td></td>
<td>Upper limit for KICKER or NOTCHER</td>
</tr>
<tr>
<td>FRACTION</td>
<td>1.0</td>
<td>-</td>
<td>retain a randomly selected FRACTION of the macroparticles</td>
</tr>
</tbody>
</table>

*a*KWEERKNT is the number of macroparticles lost  
b*Defaults to KOUNT - KWEERKNT  
c*See PARTITION in P command

### L Command, Namelist /LLRF/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHFBON</td>
<td>F</td>
<td>-</td>
<td>Activates phase feedback</td>
</tr>
<tr>
<td>NTUAVG</td>
<td>1</td>
<td>-</td>
<td>The number of past turns to average in computing the feedback; the default NTUAVG = 1 represents infinite-bandwidth feedback</td>
</tr>
<tr>
<td>NTURES</td>
<td>1</td>
<td>-</td>
<td>The number of turns for the feedback to respond; the present signal is compared to the signal of NTURES turns ago.</td>
</tr>
</tbody>
</table>
| ITFB     | 0       | -    | The form of phase feedback:  
0 – Critical damping  
1 – Critical damping; gain reduced proportionally to \(\nu^2\) within non-adiabatic interval  
2 – Fixed  
3 – Fixed; gain reduced proportionally to \(\nu^2\) within non-adiabatic interval  
4 – Fixed gain; turns to average \(5/\nu_s\) |
| FBFAC    | 1.0     | -    | The gain applied to the phase feedback |
| USEWT    | F       | -    | Applies weight function W to phase signal over NTUAVG turns |
| W(NTUAVG)| 0.0     | -    | Weight function multiplying phase signal |
| DLIMIT   | 5.7296  | deg  | The upper limit on the magnitude of the phase feedback on a given turn |
| VFON     | F       | -    | Activates voltage feedback |
| VFBFCTR  | 1.0     | -    | The gain applied to the voltage feedback |
| VLIMIT   | .1      | -    | Limit on the fraction of voltage feedback (DAMPL) |
| FILBWD   | "DUMMY" | -    | Name of time/bunch length curve for voltage feedback (see text) |
| ETAJMP   | 0.0     | -    | The value of \(\eta^a\) at which to jump the phase of the RF |

*a*\(\eta = \gamma_T^{-2} - \gamma^{-2}\)

### M Command, Namelist /MRANGE/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPU</td>
<td>TRUE.</td>
<td>-</td>
<td>Set display to beam current</td>
</tr>
<tr>
<td>FASPEC</td>
<td>FALSE.</td>
<td>-</td>
<td>Set display to Fourier amplitude spectrum of beam current</td>
</tr>
<tr>
<td>FPSPC</td>
<td>FALSE.</td>
<td>-</td>
<td>Set display to Fourier power spectrum of beam current</td>
</tr>
<tr>
<td>EVTDSPEC</td>
<td>FALSE.</td>
<td>-</td>
<td>Set display to Fourier spectrum of TD collective voltage</td>
</tr>
<tr>
<td>EVRESPEC</td>
<td>FALSE.</td>
<td>-</td>
<td>Set display to Fourier spectrum of wakefield voltage from resonances</td>
</tr>
<tr>
<td>EVFDSPEC</td>
<td>FALSE.</td>
<td>-</td>
<td>Set display to Fourier spectrum of FD collective voltage</td>
</tr>
<tr>
<td>TMBEGIN</td>
<td>0.0</td>
<td>s</td>
<td>Time at which to start saving mountain range data</td>
</tr>
<tr>
<td>TMEND</td>
<td>0.0</td>
<td>s</td>
<td>Time after which to stop saving mountain range data</td>
</tr>
<tr>
<td>MRMPLOT</td>
<td>1</td>
<td>turn</td>
<td>Turn interval at which to record mountain range data</td>
</tr>
<tr>
<td>MRNBIN</td>
<td>100</td>
<td>-</td>
<td>Number of bins in plot range MRBMAX - MRBMIN</td>
</tr>
<tr>
<td>MRBMIN</td>
<td></td>
<td></td>
<td>Minimum abscissa for mountain range</td>
</tr>
<tr>
<td>MRBMAX</td>
<td></td>
<td></td>
<td>Maximum abscissa for mountain range</td>
</tr>
</tbody>
</table>

*a*Defaults to \(-180^\circ/\text{FRAC}\) for current pickup data and to one for Fourier amplitude or power spectrum.  
b*Defaults to \(180^\circ/\text{FRAC}\) for current pickup data and NBINFFT/2 for Fourier amplitude or power spectrum.
### N Command, Namelist /MRPLOT/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRPMIN</td>
<td>0.0°</td>
<td></td>
<td></td>
<td>Minimum abscissa value for mountain range plot</td>
</tr>
<tr>
<td>MRPMAX</td>
<td>0.0°</td>
<td></td>
<td></td>
<td>Maximum abscissa value for mountain range plot</td>
</tr>
<tr>
<td>NTRACE</td>
<td>100</td>
<td></td>
<td></td>
<td>Number of traces on page</td>
</tr>
<tr>
<td>NSKIP</td>
<td>0</td>
<td></td>
<td></td>
<td>Number of records to be skipped between each trace</td>
</tr>
<tr>
<td>TOPTOB</td>
<td>0.7</td>
<td></td>
<td></td>
<td>The fraction of the vertical range over which NTRACE traces are to be plotted (approximate if TBASE=T)</td>
</tr>
<tr>
<td>SCALE</td>
<td>0.3</td>
<td></td>
<td></td>
<td>The height of the first trace, in units in which the entire vertical range of the plot is 1.0</td>
</tr>
<tr>
<td>MSTART</td>
<td>0°</td>
<td></td>
<td></td>
<td>Turn number at which to start plots</td>
</tr>
<tr>
<td>MSTOP</td>
<td>0°</td>
<td></td>
<td></td>
<td>Turn number at which to stop plots</td>
</tr>
<tr>
<td>TMSTART</td>
<td>0.0°</td>
<td>s</td>
<td></td>
<td>Time at which to start plots</td>
</tr>
<tr>
<td>TMSTOP</td>
<td>0.001s</td>
<td>s</td>
<td></td>
<td>Time at which to stop plots</td>
</tr>
<tr>
<td>TBASE</td>
<td>F</td>
<td></td>
<td></td>
<td>Switch causing plot trace separation to be proportional to time</td>
</tr>
<tr>
<td>NRNBIN</td>
<td>100</td>
<td></td>
<td></td>
<td>Number of points to plot on a trace</td>
</tr>
<tr>
<td>SMOOTH</td>
<td>0</td>
<td></td>
<td></td>
<td>Smoothing option</td>
</tr>
<tr>
<td>OBJWGT</td>
<td>3.</td>
<td></td>
<td></td>
<td>Weight of fitting term of object function w.r.t smoothing term for polynomial smoothing</td>
</tr>
<tr>
<td>ITNO</td>
<td>1</td>
<td></td>
<td></td>
<td>Number of iterations for either 1-2-1 or Bernstein smoothing</td>
</tr>
<tr>
<td>LIM</td>
<td>F</td>
<td></td>
<td></td>
<td>Switch for plotting dotted lines connecting left-most and rightmost non-zero points of consecutive traces</td>
</tr>
<tr>
<td>XCRNR</td>
<td>135.001</td>
<td></td>
<td></td>
<td>Fraction of full width of plot frame between left-hand edge and left side of plot</td>
</tr>
<tr>
<td>YCRNR</td>
<td>69.001</td>
<td></td>
<td></td>
<td>Fraction of full height of plot frame between bottom edge and bottom of plot</td>
</tr>
<tr>
<td>XAXISL</td>
<td>750.001</td>
<td></td>
<td></td>
<td>Plot width as a fraction of frame width</td>
</tr>
<tr>
<td>YAXISL</td>
<td>475.001</td>
<td></td>
<td></td>
<td>Plot height as a fraction of frame height</td>
</tr>
</tbody>
</table>

*a* Defaults of 0.0 for MRPMIN and MRPMAX imply data is to be plotted over its entire range; abscissa units are degrees for IPU=T and harmonic number for FAxSPEC=T.

*b* The defaults of 0 for MSTART and MSTOP, or 0.0 for TMSTART and TMSTOP, imply that all mountain range records are to be plotted.

*c* SMOOTH = 1 also required.
**O Command, Namelist /GRAPH/**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLOT</td>
<td>1000</td>
<td>turn</td>
<td>Output every MPLOT turns</td>
</tr>
<tr>
<td>POSTP</td>
<td>F</td>
<td></td>
<td>Write all data in COMMON blocks to unit 18; do not call plotting routine.</td>
</tr>
<tr>
<td>TITL</td>
<td>exists</td>
<td></td>
<td>String of up to 50 characters for plot headings</td>
</tr>
<tr>
<td>PLTSW</td>
<td>T</td>
<td></td>
<td>Select plot options:</td>
</tr>
<tr>
<td>(1)</td>
<td>T</td>
<td></td>
<td>Draw phase space plot</td>
</tr>
<tr>
<td>(2)</td>
<td>T</td>
<td></td>
<td>Plot phase space points (different symbol for each class)</td>
</tr>
<tr>
<td>(3)</td>
<td>F</td>
<td></td>
<td>Interconnect points within each class</td>
</tr>
<tr>
<td>(4)</td>
<td>F</td>
<td></td>
<td>Draw lines at centroid and ±σ</td>
</tr>
<tr>
<td>(5)</td>
<td>F</td>
<td></td>
<td>Draw voltage waveform</td>
</tr>
<tr>
<td>(6)</td>
<td>F</td>
<td></td>
<td>Set plot boundaries to turning points of contour</td>
</tr>
<tr>
<td>(7)</td>
<td>F</td>
<td></td>
<td>Suppress captions, axis labels, etc.</td>
</tr>
<tr>
<td>(8)</td>
<td>T</td>
<td></td>
<td>Plot θ histogram</td>
</tr>
<tr>
<td>(9)</td>
<td>F</td>
<td></td>
<td>Set θ histogram limits to turning points of contour</td>
</tr>
<tr>
<td>(10)</td>
<td>T</td>
<td></td>
<td>Plot E histogram</td>
</tr>
<tr>
<td>(11)</td>
<td>F</td>
<td></td>
<td>Set E histogram limits to turning points of contour</td>
</tr>
<tr>
<td>(12)</td>
<td>F</td>
<td></td>
<td>Plot Fourier amplitudes</td>
</tr>
<tr>
<td>(13)</td>
<td>F</td>
<td></td>
<td>Include phases in plot of Fourier spectrum</td>
</tr>
<tr>
<td>(14)</td>
<td>F</td>
<td></td>
<td>Plot a curve produced by a shazam routine in place of rf waveform</td>
</tr>
<tr>
<td>(15)</td>
<td>F</td>
<td></td>
<td>Plot the real and imaginary impedance from the input</td>
</tr>
<tr>
<td>(17)</td>
<td>T</td>
<td></td>
<td>Start bucket contour at unstable fixed point</td>
</tr>
<tr>
<td>(18)</td>
<td>T</td>
<td></td>
<td>Start bucket contour above stable fixed point at E = E₀ + HBCKT</td>
</tr>
<tr>
<td>(19)</td>
<td>T</td>
<td></td>
<td>Plot flow line points (different symbol for each class)</td>
</tr>
<tr>
<td>(20)</td>
<td>T</td>
<td></td>
<td>Interconnect flow line points within each class</td>
</tr>
<tr>
<td>(21)</td>
<td>F</td>
<td></td>
<td>Bunch perf. cond. wall voltage vs. θ</td>
</tr>
<tr>
<td>(22)</td>
<td>F</td>
<td></td>
<td>Include distribution in previous plot</td>
</tr>
<tr>
<td>(23)</td>
<td>F</td>
<td></td>
<td>Wake field (time domain) voltage vs. θ</td>
</tr>
<tr>
<td>(24)</td>
<td>F</td>
<td></td>
<td>Include distribution in previous plot</td>
</tr>
<tr>
<td>(25)</td>
<td>F</td>
<td></td>
<td>Frequency domain voltage vs. θ (inc. perf. cond. wall if FDSCON = T)</td>
</tr>
<tr>
<td>(26)</td>
<td>F</td>
<td></td>
<td>Include distribution in previous plot</td>
</tr>
<tr>
<td>(27)</td>
<td>F</td>
<td></td>
<td>Voltage from special time domain calculation for resonators vs. θ</td>
</tr>
<tr>
<td>(28)</td>
<td>F</td>
<td></td>
<td>Include distribution in previous plot</td>
</tr>
<tr>
<td>(29)</td>
<td>F</td>
<td></td>
<td>Create don’t-plot area in phase plane defined by four points</td>
</tr>
<tr>
<td>(30)</td>
<td>F</td>
<td></td>
<td>Create don’t-plot area in phase plane defined by matched contour</td>
</tr>
<tr>
<td>(31)</td>
<td>F</td>
<td></td>
<td>PLTSW(31)=T defines don’t-plot area outside closed figure</td>
</tr>
<tr>
<td>(32)</td>
<td>F</td>
<td></td>
<td>PLTSW(32)=T adds collective voltage to voltage waveform (PLTSW(5)=T)</td>
</tr>
<tr>
<td>(34)</td>
<td>F</td>
<td></td>
<td>Replace Harmonic Number abscissa with Harmonic Frequency in Fourier amplitude plot (PLTSW(12)=T)</td>
</tr>
</tbody>
</table>

- **NPJMP** | 1 | - | In phase space plot, plot only every NPJMPth point |
- **KLPLOT** | 0 | - | Select classes in phase space plot and projections (see Sec. 2.2.3) |
  - 0 — All classes plotted |
  - 1 ≤ KLPLOT ≤ KLASSES — Plot class KLPLOT only |
- **IRF** | 1 | - | Selects voltage source for contour plotting: |
  - < 0 — No contour plotted |
  - 0 — All active (NRF) sources |
  - 1-10 — Source IRF (1 ≤ IRF ≤ NRF) |
- **ICONTUR** | 1 | - | Select the type of reference contour to plot on phase space plot |
  - 0 — No contour |
  - 1 — Bucket contour |
  - 2 — Contour of initial bunch area SBNCH |
  - 3 — Contour of the specified area REFAREA |
  - 4 — Contour containing 95% of the particles |
  - 5 — Flow lines chosen by LINES, ELMIN, and ELMAX |
- **REFAREA** | 0.1 | eVs | Area of reference contour for ICONTUR = 3 |
- **LINES** | 1 | - | Number of flow lines for ICONTUR = 5 |
- **ELMIN** | 0.0 | MeV | Energy above E₀ for first flow line |
- **ELMAX** | 1.0 | MeV | Energy of top flow line |
- **THPMIN** | 0.0 | deg | Lower θ limit for phase space plot |
- **THPMAX** | 0.0 | deg | Upper θ limit for phase space plot |
- **DEPMIN** | 0.0 | MeV | Lower E limit for phase space plot |
- **DEPMAX** | 0.0 | MeV | Upper E limit for phase space plot |

---

*aAt least one of PLTSW(17) or PLTSW(18) must be true for ICONTUR=1 else program sets both .TRUE.*

*bsee definitions of THEXCMI, THEXCP, DEEXCMI, DEEXCP*

*csee definition of SEXCL*

*dFor ICONTUR = 0 or 5, the SBCKT and HBCKT values are calculated from the principal rf system only, the system producing the greatest bucket height.*

*eIf ELMIN < 0., the magnitude only will be used.*

*fTHPMIN and THPMAX both 0.0 results in a plotting range −180°/FRAC ≤ θ ≤ 180°/FRAC.*

*gDEPMIN and DEPMAX both 0.0 results in a plotting range approximately the range of particle energies.*
<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFVMIN</td>
<td>0.0(^\circ)</td>
<td>MV</td>
<td>Lower limit for optional rf waveform plot (PLTSW(5))</td>
</tr>
<tr>
<td>RFVMAX</td>
<td>0.0(^\circ)</td>
<td>MV</td>
<td>Upper limit for optional rf waveform plot (PLTSW(5))</td>
</tr>
<tr>
<td>IEREF</td>
<td>1</td>
<td>-</td>
<td>Determines energy origin for phase space:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 — E0, the reference energy (often = ES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 — ES, the synchronous energy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 — EBAR, the average particle energy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 — EREF, the reference particle energy</td>
</tr>
<tr>
<td>NBINTH</td>
<td>50</td>
<td>-</td>
<td>The number of bins for the (\theta) histogram</td>
</tr>
<tr>
<td>THBMIN</td>
<td>0.0(^\circ)</td>
<td>deg</td>
<td>Lower limit for (\theta) histogram</td>
</tr>
<tr>
<td>THBMAX</td>
<td>0.0(^\circ)</td>
<td>deg</td>
<td>Upper limit for (\theta) histogram</td>
</tr>
<tr>
<td>NBINE</td>
<td>50</td>
<td>-</td>
<td>The number of bins for the E histogram</td>
</tr>
<tr>
<td>EBMIN</td>
<td>0.0(^\circ)</td>
<td>MeV</td>
<td>Lower limit for E histogram</td>
</tr>
<tr>
<td>EBMAX</td>
<td>0.0(^\circ)</td>
<td>MeV</td>
<td>Upper limit for E histogram</td>
</tr>
<tr>
<td>IFBMIN</td>
<td>1</td>
<td>-</td>
<td>Lower limit for FFT plot - If specified, specify IFBMAX also</td>
</tr>
<tr>
<td>IFBMAX</td>
<td>0</td>
<td>-</td>
<td>Upper limit for FFT plot</td>
</tr>
<tr>
<td>CVBMIN</td>
<td>0.0(^\circ)</td>
<td>deg</td>
<td>Lower (\theta) limit for all collective voltage plots</td>
</tr>
<tr>
<td>CVBMAX</td>
<td>0.0(^\circ)</td>
<td>deg</td>
<td>Upper limit for all collective voltage plots</td>
</tr>
<tr>
<td>THEXCMI</td>
<td>0</td>
<td>deg</td>
<td>Lower (\theta) value defining rectangular don’t plot area</td>
</tr>
<tr>
<td>THEXCL</td>
<td>0</td>
<td>deg</td>
<td>Upper (\theta) value defining rectangular don’t plot area</td>
</tr>
<tr>
<td>DEEXCMI</td>
<td>0</td>
<td>MeV</td>
<td>Lower energy value defining rectangular don’t plot area</td>
</tr>
<tr>
<td>DEEXCL</td>
<td>0</td>
<td>MeV</td>
<td>Upper energy value defining rectangular don’t plot area</td>
</tr>
<tr>
<td>SEXCL</td>
<td>0</td>
<td>eVs</td>
<td>Area of matched contour defining a don’t-plot area</td>
</tr>
<tr>
<td>DTHCURV</td>
<td>0</td>
<td>deg</td>
<td>Amount by which contour will be moved in (\theta) direction</td>
</tr>
<tr>
<td>DECURV</td>
<td>0</td>
<td>MeV</td>
<td>Amount by which contour will be moved in E direction</td>
</tr>
<tr>
<td>DELCON</td>
<td>0.01</td>
<td>-</td>
<td>Determine bucket to precision DELCON(^2) 360(^\circ) w/ RF</td>
</tr>
<tr>
<td>KNTLIM</td>
<td>500000</td>
<td>-</td>
<td>Number of iterations of difference equation which will be attempted to close contour</td>
</tr>
<tr>
<td>XCRNR</td>
<td>135.001</td>
<td>0.001</td>
<td>Fraction of full width of plot frame between lefthand edge and left side of plot</td>
</tr>
<tr>
<td>YCRNR</td>
<td>69.001</td>
<td>0.001</td>
<td>Fraction of full height of plot frame between bottom edge and bottom of plot</td>
</tr>
<tr>
<td>XAXISL</td>
<td>750.001</td>
<td>0.001</td>
<td>Plot width as a fraction of frame width</td>
</tr>
<tr>
<td>YAXISL</td>
<td>475.001</td>
<td>0.001</td>
<td>Plot height as a fraction frame height</td>
</tr>
</tbody>
</table>

\(^a\)Default 0.0 results in autoscaling of minimum of voltage axis

\(^b\)Default 0.0 results in autoscaling of maximum of voltage axis

\(^c\)A particle which ESME tracks as a reference. See the T command for optional initial values

\(^d\)Limits of 0.0 for both THBMIN and THBMAX result in the plot range being the same as for the phase space plot.

\(^e\)Limits of 0.0 for both EBMIN and EBMAX result in the plot range being the same as for the phase space plot.

\(^f\)IFBMAX = 0 results in the upper limit being the greatest Fourier harmonic computed.

\(^g\)Limits of 0.0 for both CVBMIN and CVBMAX result in the range for the plot being \(\pm 180\(^\circ\)\)/FRAC.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KIND</td>
<td>1</td>
<td>-</td>
<td></td>
<td>Chooses the type of distribution to be generated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1–Rectangular outline, NTH by NE points, limited by THMIN, THMAX, REMIN, REMAX (see text)(^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2–Uniform rectangular grid NTH by NE, limits as in KIND = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3–Random uniform distribution of NPOINT points within rectangular limits as in KIND = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4–Random uniform in (\theta), limits THMIN, THMAX; Gaussian in E, REMIN, REMAX = (\pm 2\sigma), NPOINT points</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5–Gaussian in (\theta), THMIN,THMAX = (\pm 2\sigma); random uniform in E, limits REMIN, REMAX, NPOINT points</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6–Rectangular grid, regular in (\theta), Gaussian in E, NTH by NE points</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7–Bunch outline of NPOINT particles (see text)(^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8–Regular grid of approximately NTH by NE particles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9–Random uniform bunch of NPOINT particles within contour</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10–Bi-Gaussian distribution of NPOINT particles, 95% within contour</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11–NPOINT uniformly spaced particles on flow lines just above and below bucket boundary (see text)(^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12–NPOINT particles, random uniform in E, limits REMIN,REMEX, parabolic in (\theta), limits THMIN,THMAX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13–Parabolic bunch of NPOINT particles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14–Elliptical (aka Hoffman-Pedersen) bunch of NPOINT particles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15–NPOINT particles, random uniform in (\theta), limits THMIN,THMAX; parabolic in E, limits at REMIN,REMEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16–NPOINT particles within matched contour, low-noise uniform quasi-random</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17–NPOINT particles within matched contour, low-noise Gaussian quasi-random</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18–NPOINT particles in matched contour, low-noise uniform pseudo-random (Sobel sequence)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19–NPOINT particles in matched contour, low-noise Gaussian pseudo-random (Sobel sequence)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20–NPOINT particles, pseudo-random within limits REMIN,REMEX, THMIN,THMAX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21–Read particle coordinates from file FILDST into a single partition (^b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22–Place a pointer particle at THMIN,REMINS and one at THMAX,REMEX in separate partitions (see text)(^a)</td>
</tr>
<tr>
<td>NPOINT</td>
<td>1</td>
<td>-</td>
<td></td>
<td>Number of particles generated for all distributions except KIND = 1, 2, 6, 8, in which NTH and NE are used</td>
</tr>
<tr>
<td>NTH</td>
<td>2</td>
<td>-</td>
<td></td>
<td>Number of grid points in (\theta) direction</td>
</tr>
<tr>
<td>NE</td>
<td>2</td>
<td>-</td>
<td></td>
<td>Number of grid points in E direction</td>
</tr>
<tr>
<td>THMIN</td>
<td>-90.0</td>
<td>deg</td>
<td></td>
<td>Lower (\theta) limit on rectangular distributions</td>
</tr>
<tr>
<td>THMAX</td>
<td>90.0</td>
<td>deg</td>
<td></td>
<td>Upper (\theta) limit on rectangular distributions</td>
</tr>
<tr>
<td>REMIN</td>
<td>None</td>
<td>MeV</td>
<td></td>
<td>Lower energy limit on rectangular distributions; relative to the synchronous energy, ES</td>
</tr>
<tr>
<td>REMAX</td>
<td>None</td>
<td>MeV</td>
<td></td>
<td>Upper energy limit on rectangular distributions</td>
</tr>
<tr>
<td>SBNCN</td>
<td>0.1</td>
<td>eVs</td>
<td></td>
<td>Area within limiting contour</td>
</tr>
<tr>
<td>IPOPO</td>
<td>1</td>
<td>-</td>
<td></td>
<td>Specifies RF source for generating bunch limiting contour:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0–All active (NRF) sources</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1–Source (1 \leq I \leq N)RF</td>
</tr>
<tr>
<td>THOFF</td>
<td>0.0</td>
<td>deg</td>
<td></td>
<td>Amount to displace distribution generated in current call to POPUL8 in (\theta) direction</td>
</tr>
<tr>
<td>EOFF</td>
<td>0.0</td>
<td>MeV</td>
<td></td>
<td>Amount to displace currently generated distribution in E direction</td>
</tr>
<tr>
<td>THTRAN</td>
<td>0.0</td>
<td>deg</td>
<td></td>
<td>Amount to displace all particles (generated in this and previous calls to POPUL8) in (\theta) direction</td>
</tr>
<tr>
<td>ETRAN</td>
<td>0.0</td>
<td>MeV</td>
<td></td>
<td>Amount to displace all particles in E direction</td>
</tr>
<tr>
<td>WINJ</td>
<td>0.0</td>
<td>MeV</td>
<td></td>
<td>Injection energy; when WINJ and PINJ both have default value of 0.0 injection is at energy (E_0) of central orbit</td>
</tr>
<tr>
<td>PINJ</td>
<td>0.0</td>
<td>MeV/c</td>
<td></td>
<td>Injection momentum; alternate for WINJ above (q. v.)</td>
</tr>
<tr>
<td>NREPT</td>
<td>0</td>
<td>-</td>
<td></td>
<td>Number of times a matched bunch distribution should be replicated (^c)</td>
</tr>
<tr>
<td>HREPT</td>
<td>1</td>
<td>-</td>
<td></td>
<td>The harmonic number at which replications are to be made</td>
</tr>
<tr>
<td>DITHTH</td>
<td>0.0</td>
<td>degree</td>
<td></td>
<td>An optional rms azimuth for Gaussian centroid scatter</td>
</tr>
<tr>
<td>DITHE</td>
<td>0.0</td>
<td>MeV</td>
<td></td>
<td>An optional rms energy for Gaussian centroid scatter</td>
</tr>
<tr>
<td>IBCKTF(1:LTABL)</td>
<td>0</td>
<td>-</td>
<td></td>
<td>The number(s) of buckets relative to original bunch (\pm)</td>
</tr>
<tr>
<td>D0HTH(1:LTABL)</td>
<td>0</td>
<td>degree</td>
<td></td>
<td>Displacement of bunch in azimuth relative to its own bucket center</td>
</tr>
<tr>
<td>DOEN(1:LTABL)</td>
<td>0</td>
<td>MeV</td>
<td></td>
<td>Displacement of bunch in energy relative to its own bucket center</td>
</tr>
<tr>
<td>PARTITION</td>
<td>T</td>
<td>-</td>
<td></td>
<td>Partition distribution into separate classes;(^d) each separate use of the P command with PARTITION = T introduces at least one new partition</td>
</tr>
<tr>
<td>RENORM</td>
<td>F</td>
<td>-</td>
<td></td>
<td>Calculate ANORM for matched bunch so that EPSILON (\equiv) SBNCN. RENORM is reset to .FALSE. at each use.</td>
</tr>
<tr>
<td>FILDST</td>
<td>‘DUMMY’</td>
<td>-</td>
<td></td>
<td>Name of a file containing a coordinate pair count and coordinate pairs in succeeding records for KIND=21</td>
</tr>
</tbody>
</table>

\(^a\) distribution will not contribute to collective potential
\(^b\) distribution will contribute to collective potential
\(^c\) The bunch replication feature requires PARTITION = T.
\(^d\) Different classes of particles may be plotted with distinct symbols.
### R Command, Namelist /RING/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQ</td>
<td>None</td>
<td></td>
<td>m</td>
<td>The reference radius for the central orbit</td>
</tr>
<tr>
<td>GAMTSQ</td>
<td>None</td>
<td></td>
<td>-</td>
<td>Square of transition ( \gamma ) (&lt; 0 acceptable)</td>
</tr>
<tr>
<td>ALPHA1</td>
<td>0.0</td>
<td></td>
<td>-</td>
<td>Coefficient of ((\Delta p/p)^{\alpha}) in series expansion for length difference between particle trajectory and reference orbit</td>
</tr>
<tr>
<td>ALPHA2</td>
<td>0.0</td>
<td></td>
<td>-</td>
<td>Coefficient of ((\Delta p/p)^{\beta}) in series for path length difference</td>
</tr>
<tr>
<td>W0I</td>
<td>None</td>
<td></td>
<td>MeV</td>
<td>Kinetic energy on the central orbit at ( T = T_I )</td>
</tr>
<tr>
<td>W0F</td>
<td>0.0</td>
<td></td>
<td>MeV</td>
<td>Kinetic energy on the central orbit at ( T = T_F ) (see text)</td>
</tr>
<tr>
<td>P0I</td>
<td>None</td>
<td></td>
<td>MeV/c</td>
<td>Momentum on the central orbit at ( T = T_I )</td>
</tr>
<tr>
<td>P0F</td>
<td>0.0</td>
<td></td>
<td>MeV/c</td>
<td>Momentum on the central orbit at ( T = T_F ) (see text)</td>
</tr>
<tr>
<td>T0</td>
<td>0.0</td>
<td></td>
<td>s</td>
<td>Start time of magnetic field change</td>
</tr>
<tr>
<td>TF</td>
<td>0.0</td>
<td></td>
<td>s</td>
<td>End time of magnetic field change</td>
</tr>
<tr>
<td>TM</td>
<td>0.0</td>
<td></td>
<td>s</td>
<td>Time at which to set curve value to W0I (P0I) if different from T0</td>
</tr>
<tr>
<td>TSTART</td>
<td>0.0</td>
<td></td>
<td>s</td>
<td>Time at which tracking begins</td>
</tr>
<tr>
<td>FRAC</td>
<td>1.</td>
<td></td>
<td>-</td>
<td>Determines azimuthal periodicity, calculation restricted to (-180^\circ/FRAC \leq \vartheta \leq 180^\circ/FRAC)</td>
</tr>
<tr>
<td>PIPRAD</td>
<td>1.0</td>
<td></td>
<td>m</td>
<td>Radius of beam pipe for particle loss</td>
</tr>
<tr>
<td>EBDRY</td>
<td>F</td>
<td></td>
<td>-</td>
<td>Switch to set absorbing beam pipe walls at REQ ( \pm ) PIPRAD</td>
</tr>
<tr>
<td>DES</td>
<td>0.0</td>
<td></td>
<td>MeV</td>
<td>Constant energy offset of synchronous orbit relative to reference orbit</td>
</tr>
<tr>
<td>KURVEB</td>
<td>1.</td>
<td></td>
<td>-</td>
<td>Magnetic field ramp from W0I to W0F or P0I to P0F:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 – Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 – Increasing parabolic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 – Biased sinusoidal (See text for definitions CRA, CRB, CRC, TR3.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 – Ramp table to be read from file given by value of FILRMP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5 – Parabolic with final slope W0FDOT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6 – Parabolic with final slope P0FDOT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7 – Cubic with zero initial slope and curvature</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8 – Decreasing parabolic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9 – Ramp table from file given by FILRMP - polynomial representation</td>
</tr>
<tr>
<td>CRA</td>
<td>0.0</td>
<td></td>
<td>-</td>
<td>Time shift of first harmonic of KURVEB=3 ramp</td>
</tr>
<tr>
<td>CRB</td>
<td>0.0</td>
<td></td>
<td>-</td>
<td>Relative amplitude of second harmonic in KURVEB=3</td>
</tr>
<tr>
<td>CRC</td>
<td>0.0</td>
<td></td>
<td>-</td>
<td>Time shift of second harmonic for KURVEB=3</td>
</tr>
<tr>
<td>TR3</td>
<td>0.0</td>
<td></td>
<td>s</td>
<td>Half-period of fundamental for KURVEB=3; TR3=0. ( \Rightarrow ) half-period = TF-TI</td>
</tr>
<tr>
<td>W0IDOT</td>
<td>0.0</td>
<td></td>
<td>MeV/s</td>
<td>Slope of parabolic ramp at TI (KURVEB=5)</td>
</tr>
<tr>
<td>W0FDOT</td>
<td>0.0</td>
<td></td>
<td>MeV/s</td>
<td>Slope of parabolic ramp at TF</td>
</tr>
<tr>
<td>P0IDOT</td>
<td>0.0</td>
<td></td>
<td>MeV/s</td>
<td>Slope of parabolic ramp at TI (KURVEB=6)</td>
</tr>
<tr>
<td>P0FDOT</td>
<td>0.0</td>
<td></td>
<td>MeV/s</td>
<td>Slope of parabolic ramp at TF</td>
</tr>
<tr>
<td>JNRAMP</td>
<td>F</td>
<td></td>
<td>-</td>
<td>Establishes starting point of ramp as point at which program finds itself–for smoothly piecing ramp segments together</td>
</tr>
<tr>
<td>FILRMP</td>
<td>‘DUMMY’</td>
<td></td>
<td>-</td>
<td>The file name (full path) of the ramp table file (see 3.2.3)</td>
</tr>
<tr>
<td>GMAJMP</td>
<td>F</td>
<td></td>
<td>-</td>
<td>Set ( \gamma )-jump on (See text for compatibility.)</td>
</tr>
<tr>
<td>KINDG</td>
<td>1</td>
<td></td>
<td>-</td>
<td>Type of ( \gamma )-variation: ( ^{\alpha} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 – Linear ( \gamma_T = GAMPAR(1) + GAMPAR(2) \times T )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 – Decreasing exponential ( \gamma_T = GAMPAR(1) + GAMPAR(3) \times (1 - e^{-T/GAMPAR(2)}) )</td>
</tr>
<tr>
<td>GAMPAR(1:3)</td>
<td>0.0</td>
<td></td>
<td>-</td>
<td>Coefficients for ( \gamma )-variation (real ( \gamma )-only)</td>
</tr>
<tr>
<td>CHGNO</td>
<td>1.0</td>
<td></td>
<td>-</td>
<td>Charge of beam particle in units (</td>
</tr>
<tr>
<td>EM0CSQ</td>
<td>938.27231</td>
<td></td>
<td>MeV</td>
<td>Rest energy of beam particle</td>
</tr>
</tbody>
</table>

\( ^{\alpha} \) \( T = 0 \) corresponding to time at which R command is invoked with GMAJMP = .TRUE.
### T Command, Namelist /CYCLE/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSTOPb</td>
<td>0.0</td>
<td>s</td>
<td>Time at which to stop tracking</td>
</tr>
<tr>
<td>TTRACKb</td>
<td>0.0</td>
<td>s</td>
<td>Duration of time to track</td>
</tr>
<tr>
<td>RSCALE0</td>
<td>1.0</td>
<td></td>
<td>Sets time scaling if AUTOSCL=F; multiplies range of scaling for AUTOSCL=T</td>
</tr>
<tr>
<td>AUTOSCL</td>
<td>.FALSE.</td>
<td></td>
<td>Switch to activate auto scaling of time step</td>
</tr>
<tr>
<td>RAIL1</td>
<td>2.0</td>
<td></td>
<td>Upper limit for RSCALE</td>
</tr>
<tr>
<td>RL0</td>
<td>0.5</td>
<td></td>
<td>Lower limit for RSCALE</td>
</tr>
<tr>
<td>GNUSCAL</td>
<td>0.01</td>
<td></td>
<td>Proportionality factor between RSCALE and GNUS(^{-1}) in auto-scaling</td>
</tr>
<tr>
<td>LGRTHM</td>
<td>1</td>
<td></td>
<td>Select difference equations used in tracking</td>
</tr>
<tr>
<td>ITRAP(1:4)</td>
<td>0</td>
<td></td>
<td>Indicates a condition for which tracking should be interrupted before time indicated by TTRACK or TSTOP:</td>
</tr>
<tr>
<td>ETA TRP</td>
<td>.001</td>
<td></td>
<td>For ITRAP = 3; tracking stopped when (\eta = \text{ETATRP})</td>
</tr>
<tr>
<td>PHISTRP</td>
<td>.95</td>
<td></td>
<td>For ITRAP = 4; tracking stopped when (</td>
</tr>
<tr>
<td>MGRACE</td>
<td>0</td>
<td></td>
<td>Allow a ”grace period” of MGRACE turns before trapping conditions are checked</td>
</tr>
<tr>
<td>HISTSIZ</td>
<td>10</td>
<td></td>
<td>Approximate number of history records for one execution of T</td>
</tr>
<tr>
<td>BBDRY</td>
<td>F</td>
<td></td>
<td>Remove particles tracked outside of region (-180^\circ / \text{FRAC} \leq \theta \leq 180^\circ / \text{FRAC})</td>
</tr>
<tr>
<td>THREF0</td>
<td>0.0 degree</td>
<td></td>
<td>Choose special azimuth for reference particle</td>
</tr>
<tr>
<td>EREF0</td>
<td>0.0 MeV</td>
<td></td>
<td>Choose special energy difference from (E_s) for reference particle</td>
</tr>
</tbody>
</table>

---

### Y Command, Namelist /MEMORY/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNPHASE</td>
<td>25000</td>
<td></td>
<td>The storage allocation for each phase coordinate — max. value for KOUNT</td>
</tr>
<tr>
<td>KMAXCVB</td>
<td>10000</td>
<td></td>
<td>The storage allocation for large tables of data or intermediate results in time domain collective voltage calculations</td>
</tr>
<tr>
<td>KIFFT</td>
<td>IFFT=1024</td>
<td></td>
<td>Allocates memory for FFT’s of length KIFFT</td>
</tr>
<tr>
<td>KLIABL</td>
<td>LTABLE</td>
<td></td>
<td>Sets the maximum length of several parameter and auxiliary variable tables; others fixed at LTABLE</td>
</tr>
<tr>
<td>KNSRC</td>
<td>NSRC=10</td>
<td></td>
<td>Limits the number of auxiliary tables related to rf curves</td>
</tr>
</tbody>
</table>

---

\(^a\)TSTOP is set to 0.0 when tracking is completed, or interrupted by an ITRAP option.
\(^b\)TSTOP takes precedence; if TSTOP=0.0, then TTRACK determines duration of tracking.
\(^c\)See R command
\(^d\)See Section 3.2
\(^e\)Default value for THREF0 or EREF0 give synchronous values; THREF0 = -EREF0 gives THBAR and EBAR at each turn.

\(^a\)KMAXCVB sets the maximum allowable value for input parameters NBINSC and NBRES.
Bibliography


[6] J. MacLachlan, ‘‘Particle Tracking in E-φ Space for Synchrotron Design & Diagnosis’’, Fermilab-CONF-92/333 (Nov. 92), presented at 12th Int’l Conf. on Appl. of Acc. in Res. and Ind., Denton TX, 4 Nov. 1992


[17] The copyright for the pgplot plotting package is held by California Inst. of Technology. It is written in standard f77 and is freely available for non-commercial use.