PROGRAM DYNAC

Version 8, Revision 0, 16-Dec-2025

USER GUIDE

THIS SOFTWARE IS FREE[1] AND WAS ORIGINALLY PRODUCED AT CERN/PS AND CEN/SACLAY

AUTHORS :

P. LAPOSTOLLE*	CONSULTANT	(Paris, France)
S. VALERO	CONSULTANT	(Strassbourg, France)
E. TANKE	SCKCEN	(Mol, Belgium)

* We feel very indebted to the late P. LAPOSTOLLE, who was not only a co-author of DYNAC, but also a major contributor to the field of linear accelerator physics.

Modified and maintained by:
E. Tanke

Contact:
E. Tanke, e-mail: dynacatgithub@gmail.com

DYNAC WEBSITE
https://github.com/dynac-source

With contributions from:
D.Alt, previously at NSCL/MSU, East Lansing, MI, USA
Ji-Ho Jang, RAON/IBS, Daejeon, South Korea
François Méot, previously at BNL, Upton, NY, USA

[1] See Software License Agreement on next page

Software License Agreement

NOTICE TO USERS:
This End User License Agreement (the "Agreement") is a legally binding agreement between you (either an individual or an entity, the "User"), and the DYNAC main developer (currently Eugene Tanke) regarding a) the DYNAC software (the "Software" or the "Program", which includes scripts and the Graphical User Interface (GUI) software), and b) all successor upgrades, revisions, patches, fixes, modifications, copies, additions or maintenance releases of the Software, if any, licensed to you by the DYNAC main developer (collectively, the "Updates"), and c) related user documentation and explanatory materials or files provided in written, "online" or electronic form (the "Documentation" and together with the Software and Updates, the "PRODUCT" or the "Distribution Package").

CAREFULLY READ THE FOLLOWING LEGAL AGREEMENT. USE OF THE PRODUCT PROVIDED WITH THIS AGREEMENT CONSTITUTES YOUR ACCEPTANCE OF THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT INSTALL AND/OR USE THE PRODUCT. YOUR USE OF THE PRODUCT IS CONDITIONED UPON COMPLIANCE WITH THE TERMS OF THIS AGREEMENT.

1. Intellectual property rights

The DYNAC Program and associated software ("PRODUCT") is owned by the DYNAC main developer. The DYNAC Graphical User Interface (DGUI) is an example of associated software. Your possession, installation and/or use of the PRODUCT does not transfer to you any title to the intellectual property in the PRODUCT, and you will not acquire any rights in the PRODUCT except as expressly set forth in this Agreement.

The DYNAC Program and associated software is freeware and as such the authors cannot be held accountable for any problems and or costs arising from its usage. As freeware, DYNAC and associated software may be used for commercial goals, with the exception of commercializing DYNAC PRODUCT, parts of the DYNAC Program and/or associated software itself.

2. Scope of the License

You are granted a non-exclusive license to use the PRODUCT as set forth herein. You can use the PRODUCT as set forth in the Agreement for non-commercial purposes in non-business, non-commercial environment. As freeware, DYNAC and associated software may be used for commercial goals, with the exception of commercializing DYNAC, parts of DYNAC and/or associated software itself.

The Software may be freely distributed, but no person or company may charge a fee for the distribution of the PRODUCT without written permission from the DYNAC main developer.

3. Limited warranties

The DYNAC main developer DOES NOT WARRANT THAT THE SOFTWARE IS FIT FOR ANY PARTICULAR PURPOSE. The DYNAC main developer DISCLAIMS ALL OTHER WARRANTIES WITH RESPECT TO THE SOFTWARE, EITHER EXPRESS OR IMPLIED. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES OR LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY MAY LAST, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

4. Legality statement

The program that is licensed to you is absolutely legal and you can use it provided that you are the legal owner of all files or data you are going to use with our software or have permission from the legitimate owner to perform these acts. Any illegal use of our software will be solely your responsibility. Accordingly, you affirm that you have the legal right to access all data, information and files.

You further attest that the data and/or files will not be used for any illegal purpose.

5. Final provisions

All rights not expressly granted here are reserved by the DYNAC main developer.

INDEX

1 INTRODUCTION

Two different approaches in accelerating elements (i.e. cavities and gaps) exist in the code. First, a very fast analytical method based on the concept of an ‘equivalent field’ giving a full set of quasi-Liouvillian equations (see ref.[1]), only capable however of handle one single charge state. The second approach, which can handle several charge states in a bunch, is a relatively fast and very accurate step by step numerical method based on Bode’s rule applied to the ‘reduced coordinates’ resulting from a Picht transformation (see ref.[1]).

The analytical approach in accelerating elements (i.e. cavities and gaps) including new concepts has been introduced in DYNAC in 1994 giving a full set of quasi-Liouvillian equations and resulting in a convenient matrix formalism (see ref. [1]). Each of the 6D coordinates of the macro-particles is known in any position in the accelerating elements, thus allowing space charge computations in arbitrary positions in the cavities.
These equations are available both for non-relativistic electrons with significant acceleration as well as for heavy ions undergoing large velocity variations, albeit only for single charge state beams. A numerical method has been added to DYNAC in order to be able to simulate multi-charge state beams in accelerating elements.

Both methods are well suited to simulate particles, including non-relativistic electrons, accelerated through long accelerating elements with complex fields (e.g. complex helix, multi-gap cavities including superconducting ones) where their transit time may be of the order of 10π or more and where their velocities vary by 10% or more, or where their relativistic gamma varies by a factor 3 or 4. IH structures (where the design particle typically lies outside the beam) may be simulated.
The field of accelerating gaps and/or cavities can be read in the form of coordinates (z, E (z)), or it can be described by Transit Time Factors or by Fourier series expansion.
Apart from the above-mentioned RF structures, also cavities of the buncher (in thin lens approximation) and RFQ (available for protons and heavy ions, see ref.[6]) types are included in the code.
Furthermore, a DC electron gun (see ref.[7]) as well as a Stripper, describing plural and multiple scattering of heavy ions in solids are included in the code (ref.[8], [9]). Computations of synchrotron radiation (for electrons) in bending magnets are possible (ref.[10]).

Three very different space charge routines (see ref. [2] to [5]) are available in the code, which allows one to check the validity of space charge computations. These routines are:
a) SCHEFF (a relativistic version developed at LANL, but further developed to handle multi-charge state beams)
b) SCHERM (developed at CERN, CEA/SACLAY and LANL by P. Lapostolle et al; handles single charge state beams)
c) HERSC (developed at CERN and CEA/SACLAY by P. Lapostolle et al; handles single charge state beams)

Apart from the usual optical lenses, special lenses such as a quadrupole associated with a sextupole field, a solenoid associated with a quadrupole field, steering magnets and solenoids with an arbitrary field as well as electrostatic quadrupoles and dipoles are included.

Second order transfer formalisms for most of the optical lenses are incorporated in the current version of the code; these elements can also handle multi-charge state beams.

As mentioned, heavy ion beams can be simulated, including multi-charge state ones. When using the plotting post-processor, multi-charge state beams will be color coded.

Misalignments and/or systematic or random defects in the matching parameters of cavities are possible. A physical acceptance of the machine can be defined.

The input beam can be generated from hit-or-miss Monte Carlo for different types of distributions or can be read in a file on the disk.

DYNAC can be compiled and run on Linux, MAC and MSWindows platforms. A GNU-plot based post-processor is also available, making graphics output possible on all three of the aforementioned platforms. In addition, a Python based graphical user interface, DGUI (available on the DYNAC web site), can be used to run DYNAC and to plot graphical output.

The readme.txt file on the dynac download web page gives instructions on how to install DYNAC.

With DYNAC V8, another step has been made to conform with modern Fortran standards (e.g. there are no more COMMON statements in the source code and MODULEs have been introduced). The source file of DYNAC V8 has extension .f08, whereas DYNAC V7 has .f90; V6 and older had .f as an extension.

REFERENCES:

[1] P.Lapostolle, E.Tanke and S.Valero: A New Method in Beam Dynamics Computations for Electrons and Ions in Complex Accelerating Elements, Particle Accelerators, 1994, Vol.44, pp. 215-255.

[2] F.Guy, Los Alamos Group AT-1 Memorandum AT-1:85-90, March 6, 1985

[3] P.Lapostolle and al.: A Modified Space Charge Routine for High Intensity Bunched Beams, NIM 379 (1996) pp. 21-40.

[4] P.Lapostolle and al.: HERSC, A New 3D Space Charge Routine for High Intensity Bunched Beams, LINAC2002, Gyeongju, South Korea.

[5] P.Lapostolle, E.Tanke and S.Valero: A 3-d space charge routine, HERSC, based on a sequence of 3-d Hermite orthogonal functions (obtainable from the DYNAC website).

[6] C.Biscari: Computer programs and methods for the design of high intensity RFQ, CERN/PS 85-67 (LI)
See also, E.Tanke and S.Valero: Motion of particles in a RFQ (obtainable from the DYNAC website).

[7] E. Tanke and S.Valero: Motion of electrons in a DC gun (obtainable from the DYNAC website).

[8] D.A. Eastham: Plural and multiple scattering of heavy ions in solids, DL/NSF/P11.

[9] D. Groom: Energy loss in matter by heavy particles, PDG-93-06.

[10] M.Sands: The Physics of Electron Storage Rings, SLAC-R121
See also, E.Tanke and S.Valero: Computation of synchronous radiation in bending magnets (obtainable from the DYNAC website).

2 INPUT FILES

One can start execution of the code from DGUI, a python3 based graphical user interface to DYNAC. The source code for DGUI is available from the DYNAC web site.

Alternatively, one can also start execution of the code from the command line. The format is the following:

```
dynacv8 [-h] [-v] [-mingw] [-p] [--pipe] [-mcNN] [filen]
```

where
1) filen is the file name of the DYNAC input file containing input data typifying the accelerator and/or transport line; this file is mandatory (Unit 7). The full path to the input file can be included in filen; by default, if just the input file name is given, the current directory is the one that is assumed to contain the input file.
If no input file name is given upon invoking the dynac executable, the executable will by default check if a file called dynac.in exists. If it does, it will execute that file; if not, the program will stop.
2) –h causes some ‘help’ information to be printed on the screen
3) -v causes the DYNAC version number to pe printed on the screen
4) –mingw can be used as an option when using certain MINGW versions on MSWINDOWS. This may be required as certain gfortran intrinsics behave differently (e.g. "ctime") depending on the gfortran version.
5) -p is used by DGUI (the DYNAC graphical user interface) to specify the datafile path. There should be no space between -p and the path. This option is not needed when running DYNAC from the command line.
6) --pipe can be used to pipe the DYNAC keywords and data (as opposed to reading the input from file)
7) --mcNN can be used when running multiple instances of DYNAC in parallel (multi-core). NN is then to be used to specify the core or thread number.

The first line inside the DYNAC input file ‘filen’ is a mandatory comment line; this can be used to give a brief description of what the input file represents.

All other input files (e.g. optional files describing particle distribution, solenoid field, electromagnetic field etc) if any are defined within the DYNAC input file. The formats of these are described in the corresponding type code entries in this user guide.

One can also simply double click the dynac executable in the dynac/bin folder after which an "open file dialog box" will be opened ("Finder" on MAC, "zenity" on Linux or "Explorer" on Windows) in order for the user to select an input file.

3 OUPUT FILES

Note that Units as described below refer to I/O units as defined in FORTRAN.

1) File 'dynac.dmp' (Unit 50):
Print of cavity cell or gap related data at the output. The first lines of dynac.dmp indicate the column titles: cavity or cell number, distance (m), transmission (%), synchronous phase (deg), COG (Center Of Gravity) time of flight (deg), COG relativistic beta, COG energy (MeV), Reference time of flight (deg), Reference relativistic beta, Reference energy (MeV), horizontal emittance (mm.mrad, RMS normalized), vertical emittance (mm.mrad, RMS normalized), longitudinal emittance (RMS, ns.keV), Reference energy gain (MeV), Effective voltage (MV), user defined ElementName and field amplitude setting (this is the one as defined in the input file e.g., FFIELD for CAVNUM). Refer to the dynac.print output file to also see beam related data at other locations than cavities (e.g. at optical lenses).

2) File 'dynac.short' (Unit 12):
Print of some essential beam dynamics related information; emittances are 4RMS, beam sizes are 2RMS.

3) File 'dynac.long' (Unit 16):
Print of extensive information concerning the beam dynamics computations.

4) User defined file, e.g. 'mybeam.dst' (Unit 58):
Print of the coordinates of the particles invoked by the type code entry WRBEAM in the command list. The format of 'mybeam.dst' is described under RDBEAM, remark 3).

5) Files 'beam_core_H.dst' (Unit 63) and 'beam_remove_H.dst' (Unit 62):
As file 'mybeam.dst' but with CHASE card; print the coordinates of the particles kept and removed in the horizontal plane by type code CHASE.

6) Files 'beam_core_V.dst' (Unit 65) and 'beam_remove_V.dst' (Unit 64):
As file 'mybeam.dst' but with CHASE card; print the coordinates of the particles kept and removed in the vertical plane by type code CHASE.

7) File 'beam_core_L.dst' (Unit 61) and 'beam_remove_L.dst' (Unit 60):
As file 'mybeam.dst' but with CHASE card; print the coordinates of the particles kept and removed in the longitudinal plane by type code CHASE.

8) File 'emit.plot' (Unit 66):
Data file used for the plots

9) File 'dynac.print' (Unit 71):
Print of the 1 x RMS profiles and the normalized RMS emittances of the beam at the output positions of the optical lenses and the accelerating elements in the directions x, y and z as well the beam energy and the number of retained particles. In addition, the maximum extensions (both positive and negative w.r.t. the beam axis) in the x and y directions are printed, as well as the same for the bunch length w.r.t. the bunch center (both in units of time and degrees) and the energy extent w.r.t. the COG energy. Furthermore, the dispersion *Dx* (horizontal) and *Dy* (vertical) are printed; these are based on the definitions in [1]:

$$D_x(s)=\frac{\langle x(s)\,p_t(s)\rangle}{\langle p_t^2(s)\rangle}\ ,\ D_y(s)=\frac{\langle y(s)\,p_t(s)\rangle}{\langle p_t^2(s)\rangle}$$

where p_t is the fractional deviation of the particle’s velocity from the one of the COG. Note that currently no Dx and Dy values are calculated for individual charge states, i.e. a single charge state beam is assumed.
The following 6 columns contain values for the 1 x RMS energy spread, the energy of the reference, TOF of the reference and of the COG and the average position in X and Y. The next 6 columns contain values for the α and β Twiss parameters for the horizontal, vertical and longitudinal planes respectively. The next column lists the user defined name of the optical element (if defined by the user in the input file, otherwise NoUserDefinedElementName will be listed). The following 3 columns list the available half aperture in X, Y and radially, as defined by the REJECT card. Note that if there is no REJECT card present in the DYNAC input file, 100 cm will be used as default half aperture. The last 4 columns list the halo parameters [2] in X, Y and Z as well as the 4d transverse emittance. The first line of dynac.print indicates the meaning of each column.

10) File 'rfq_list.data' (Unit 70):
Print of the characteristics of each cell in the RFQ. Starting from DYNACV6R16 this file also includes the synchronous phase at the middle of each cell, thus making the previously used 'rfq_listmid.data' file obsolete. The first line of rfq_list.data indicates the meaning of each column.

11) File 'lost_particles.data' (Unit 49):
Print of the locations of where particles were lost, their coordinates, as well as the reason why they were lost and the user defined elements where they were lost. The reason for loss is given in the following way:
Loss reason = -2 --> energy
Loss reason = -1 --> phase
Loss reason = 1 --> radius
Loss reason = 2 --> x position
Loss reason = 3 --> y position
Also the user defined name of the element where the particle was lost is listed.

12) File 'rfq_coef.data' (Unit 75):
Print of the RFQ relevant coefficients (e.g. A10) of each cell in the RFQ. The first line of rfq_coef.data indicates the meaning of each column.

13) File 'for_trace3d.t3d' (Unit 48):
A TRACE3D input file generated by DYNAC if a T3D card is present in the DYNAC input file.

14) File 'cavdat.out' (Unit 13):
Print of cavity related data when CAVNUM or CAVMC is used. At the top of the file, the column titles are given. The meaning of each title is given below.
ncav ncell: Cavity number and cell number within the cavity
Zin,acc: Starting position (accumulated)
Zin Zmid Zavg Zout, L cell: Starting, middle, average and ending position of a cell within the cavity, cell length. The average position is the location near or at the middle of the cavity, whereby the field integrals on either side of this position are equal
Wref,out dWref: Output energy and energy gain of the reference for the cell within the cavity
Ph RF TOFref: RF phase and time of flight of the reference over the cell within the cavity
TOF Ph RF: Time of flight of the reference over the cell within the cavity and the RF phase
E0TL: Maximum energy gain for the cell within the cavity
Tk Sk: Transit time factors for the reference
PHASE Ph crest: Phase of the reference and phase giving maximum energy gain for the cell within the cavity

15) Files 'chase_emith.out' (Unit 44), 'chase_emitv.out' (Unit 45), 'chase_emitl.out' (Unit 46):
Print of reduced emittances based on beam fractions set with (optional) CHASE card for the horizontal plane, the vertical plane and the longitudinal plane respectively. In addition, the maximum extensions (both positive and negative w.r.t. the beam axis) in the x and y directions are printed, as well as the same for the bunch length w.r.t. the bunch center (in units of time) and the energy extent w.r.t. the COG energy.

16) File 'rfq_profile.dat' (Unit 98):
Print of the RFQ electrodes profile (only if RFQPTQ was used). The first line of rfq_profile.dat indicates the meaning of each column. This file will not be printed if DYNAC is executed in parallel mode.

[1] Beam Size, Emittance and Optical Functions in LEP, John M. Jowett, CERN SL Note 96–38 (AP)
[2] Beam halo definitions based on moments of the particle distribution, Allen and Wangler: Physical Review ST AB, 5(124202), Dec.2002

4 PARTICLE COORDINATES AND EMITTANCES

4.1 PARTICLE COORDINATES

Each particle is defined by 10 coordinates/properties, in the code arranged in the F (J, I) array, where I represents the particle number and J one of the 10 coordinates or properties. One reshuffles F (J, I) such that the "good" particles are on top of the stack; the number of good particles is NGOOD. Particles are considered "bad" when they are outside the window defined by the type code REJECT or by the physical limits of the elements (e.g. the aperture of a quadrupole). Particles will also be considered lost when their relativistic gamma is computed to be smaller than one. The 10 coordinates/properties are:

F (1,I): initial particle number of the particle I

F (2,I): x (cm)

F (3,I): x' (mrad)

F (4,I): y (cm)

F (5,I): y' (mrad)

F (6,I): time of flight (sec)

F (7,I): total energy (MeV), i.e. sum of total rest mass and total kinetic energy

F (8,I): =1 the particle is kept, =0 the particle is lost

F (9,I) : electrical charge state (e.g. 27 for Pb^{27+})

F (10,I) : zone limit as defined with the optional type code ZONES

Furthermore, NPOINT equals the total number of particles and NGOOD the total number of good particles.

4.2 EMITTANCES

Macro particle related definitions:

x_i: Horizontal displacement of the arbitrary macro particle i with respect to the center of gravity (mm)
θ_i: Horizontal angle of the macro particle with respect to the center of gravity (mrad)
y_i: Vertical displacement of the arbitrary macro particle i with respect to the center of gravity (mm)
ϕ_i: Vertical angle of the macro particle with respect to the center of gravity (mrad)
dE_i: Energy offset of the macro particle with respect to the center of gravity (MeV total)
$d\varphi_i$: Phase offset of the macro particle with respect to the center of gravity (rad)

Emittance related definitions:

β, γ: Relativistic beta and gamma of the center of gravity

1) Horizontal emittance ε_x (RMS, phase space(x, θ))
The beam phase ellipse has the following construction:

$$A_{11}x^2 - 2\sqrt{A_{12}}x\theta + A_{22}\theta^2 = 4\Delta_x$$

$$A_{11} = \sum_i x_i^2,\ A_{22} = \sum_i \theta_i^2,\ A_{12} = \left(\sum_i x_i\theta_i\right)^2,\ \Delta_x = A_{11}A_{22} - A_{12}$$

The beam phase ellipse area is$\xi_x = 4\pi\sqrt{\Delta_x}$. The RMS (normalized and not normalized) emittances are:

$$\varepsilon_x(\mathrm{normalized}) = \beta\gamma\xi_x/\pi\ ,\ \varepsilon_x(\mathrm{notnormalized}) = \xi_x/\pi \quad \text{(mm-mrad)}$$

2) Vertical emittance ε_y (RMS, phase space(y, ϕ))
The area $\xi_y$of the beam phase ellipse is given by: $\xi_y = 4\pi\sqrt{\Delta_y}$

$$B_{11} = \sum_i y_i^2 \,, B_{22} = \sum_i \phi_i^2 \,, B_{12} = \left(\sum_i y_i \phi_i\right)^2, \Delta_y = B_{11} B_{22} - B_{12}$$

The RMS (normalized and not normalized) emittances are, respectively:

$$\varepsilon_y(\text{normalized}) = \beta\gamma\xi_y/\pi \;,\; \varepsilon_y(\text{not normalized}) = \xi_y/\pi \quad \text{(mm-mrad)}$$

3) Longitudinal emittance ε_z (RMS, phase space$(\mathrm{dE}, d\varphi)$).
The area $\xi_z$of the beam phase ellipse is given by: $\xi_z = 4\pi\sqrt{\Delta_z}$

$$C_{11} = \sum_i \mathrm{dE}_i^2 \,, C_{22} = \sum_i d\varphi_i^2 \,,\; C_{12} = \left(\sum_i \mathrm{dE}_i d\varphi_i\right)^2, \Delta_z = C_{11} C_{22} - C_{12}$$

The longitudinal RMS emittance is: $\varepsilon_z = \xi_z/\pi$ (MeV-rad)

Beam emittance definitions used in DYNAC input:

Input emittances definitions are given in section 6.1.2. (See Type Code GEBEAM).

Note: When the Twiss parameters are used in the GEBEAM card, the transverse and longitudinal emittances must not include π.

Beam emittance definitions used in DYNAC output:

A) File ‘dynac.short’:
Transverse emittances, both normalized and not normalized, are 4 RMS.
Longitudinal emittance, both in keV-deg and ns-keV, is 4 RMS.

B) File ‘dynac.dmp’:
Transverse emittances: (mm.mrad, RMS, normalized)
Longitudinal emittance: (RMS, ns.kev)

C) File ‘dynac.long’:
Definitions of transverse and longitudinal emittances are detailed in the file.

D) File ‘dynac.print’:
Transverse emittances: (mm.mrad, RMS, normalized)
Longitudinal emittance: (RMS, ns.kev)
4d Transverse emittance: (mm^2.mrad^2, RMS, normalized) , based on the determinant of the 4x4 beam matrix.

5 SUMMARIES OF AVAILABLE DYNAC TYPE CODE ENTRIES

Note that details regarding code entries can be found in chapter 6.

5.1 TYPE CODES FOR THE INPUT BEAM

INPUT	define energy and phase of the reference particle
GEBEAM	generate randomly the 6D coordinates of particles
RDBEAM	read the 6D coordinates of particles from file
ETAC	generate different charge states for particles

5.2 TYPE CODES RELATED TO OPTICAL LENSES*

BMAGNET	bending magnet
DRIFT	free space
QUADRUPO	quadrupole (magnetic field)
SOLENO	solenoid
STEER	steering magnet
SEXTUPO	sextupole
QUASEX	quadrupole associated with sextupole
SOQUAD	solenoid associated with quadrupole
FDRIFT	drift with multiple space charge computations
FSOLE	solenoid with an arbitrary field (read on the disk)
SECORD	second order matrix formalism for optical lenses
RASYN	synchrotron radiation for electrons in bending magnets
QUAELEC	quadrupole (electric field)
QUAFK	quadrupole (magnetic or electric field)
EDFLEC	electrostatic dipole (electric field)
FIRORD	first order matrix formalism for optical lenses
WIENANA	Wien filter (can be used to simulate a chopper), based on an analytical simulation, ***beta version***
WIENNUM	Wien filter, based on a numerical simulation, ***beta version***

5.3 TYPE CODE FOR ACCELERATING GAPS, CAVITIES, BUNCHER, DC GUN, STRIPPER*

CAVMC	multi cell accelerating element, single charge state beam
CAVNUM	complex accelerating element, single or multi charge state beam
CAVSC	motion of the particles in single cell symmetrical cavity
BUNCHER	buncher in thin lens approximation
MHB	multi-harmonic buncher in thick lens approximation
FIELD	read electric field of cavities from file
RWFIELD	rewinds the file containing the electric field data
HARM	electric field in the form of a Fourier series expansion
EGUN	motion of electrons in a DC gun
RFQPTQ	motion of the particles in an RFQ
STRIPPER	plural and multiple scattering of heavy ions in solids

5.4 TYPE CODES CONCERNING FUNCTIONING MODES

NREF	define a new reference particle
TOF	time of flight in connection with the RF phase
NEWF	define a new frequency
REFCOG	detach C.O.G. from synchronous particle (e.g. for IH)

5.5 TYPE CODES USED IN ORDER TO REDEFINE THE BEAM

TILT	rotation and shift of the bunch
TILZ	place at an angle the ellipse (x, z)
REJECT	window for the beam acceptance
CHASE	analysis of emittance by elimination of remote particles
COMPRES	compress the phase extension of particles in between ±π
DCBEAM	select DC or bunched beam

** For the type codes of beam line elements listed under 5.2 and 5.3, one can (optionally) add a user defined name for the beam line element. This user defined name should be listed right behind and on the same line as the type code in the input file. The user defined beam line element names will be listed in the dynac.print and dynac.dmp output files, as well as in the lost_particles.data file. Example:*

CAVNUM MB-CM1-CAV2

Here, MB-CM1-CAV2 could mean something like the second cavity in the first cryomodule in the medium beta beam line. Excluded are the type codes FIRORD, FIELD, RWFIELD and HARM, as these are not beam line elements.

5.6 TYPE CODES USED FOR MAGNET OR CAVITIES TOLERANCES AND ERRORS

ALINER alignment tolerances in x, x', y, y'
CHANGREF change of reference frame
ZROT turn plane (X, Y) around the direction of beam travel
RANDALI generate random misalignments in X, X', Y and Y'
TWQA rotate quadrupoles about the beam travel direction
MMODE errors on RF phase and amplitude

5.7 TYPE CODES USED FOR SPACE CHARGE COMPUTATIONS (S.C.C)

SCDYNAC cause space charge in the elements (except in bending magnets, see SCDYNEL)
~~SCDYNEL together with SCDYNAC, space charge calculations in bending magnets~~ (SCDYNEL deprecated since V7R2, no longer needed as a special card for space charge calculations in bending magnets)
SCPOS selects the location for space charge computations in cavities

5.8 TYPE CODES USED FOR OUTPUT PRINT

WRBEAM print particle coordinates to file
EMIT print beam characteristics to file
EMITL like EMIT, but can also read a comment from the input file and print it in dynac.short
EMIPRT select location of print beam characteristics
DFLOCS print particle coordinates to file for named beam line elements

5.9 TYPE CODES USED FOR OUTPUT PLOT

EMITGR X-X', Y-Y', Y-X and Z-Z' scatter plots
ENVEL envelopes of the beam
PROFGR bunch profiles and X-Z, Y-Z scatter plots
EMITGRD X-X', Y-Y', Y-X and Z-Z' intensity plots
PROFGRD bunch profiles and X-Z, Y-Z intensity plots

5.10 TYPE CODES FOR OTHER USES

STOP ends simulation; this card is mandatory
ACCEPT find an acceptance for the beam at input
COMMENT add a comment to the command list (alternatively, one can use a semi-colon as first character)
ZONES define color-coded zones in X-Y-Z space
T3D generate a TRACE3D input file
BINDIST enable to generate or read particle distributions using binary representation

6 DESCRIPTIONS OF AVAILABLE TYPE CODE ENTRIES

Note that each type code may have several lines of input data (marked as 1), 2), etc) and that each line of input data may contain more than one parameter.

6.1 INPUT BEAM

6.1.1 TYPE CODE: INPUT

Define energy and phase of the reference particle (i.e. the C.O.G. or the synchronous particle) at the entrance of the machine.

ENTRY:

1) UEM ATM Q

UEM: rest mass (MeV)

Examples of rest mass:

Proton:	938.27231	MeV
H^- :	939.301404	MeV
Mesons:	33.9093	MeV
Pions:	139.5685	MeV
Kaons:	493.667	MeV
Heavy ions:	931.49432	MeV
Electrons:	0.511	MeV

ATM: Atomic number

Q: electrical charge state (in unit of electric charge)

2) ENEDEP TOF

ENEDEP: total kinetic energy of the particle reference at the input (MeV)

TOF: time of flight, i.e. adjustment of the RF phase (deg) to be applied to particles

REMARKS:
(1) INPUT card must be preceded by GEBEAM card. It is not used in combination with RDBEAM card.

(2) At the start (after INPUT card), energy and time of flight of the COG and of the reference particle are coinciding and are considered disconnected (i.e. they will evolve separately, see type code REFCOG).

EXAMPLE: (entries for a 2.5 MeV H^- beam)

```
INPUT
939.301404 (UEM in MeV)  1.  (ATM)    -1. (Q)
2.50 (ENEDEP in MeV)        0.  (TOF in deg)
```

EXAMPLE: (entries for a 0.25 MeV/u Pb^{27+} beam)

```
INPUT
931.49432(UEM in MeV)    208. (ATM)    27. (Q)
52. (ENEDEP in MeV)           0.   (TOF in deg)
```

6.1.2 TYPE CODE: GEBEAM

Generate the 6-D coordinates for a cloud of particles by using a Monte Carlo method.

ENTRY:

1) LAW ITWISS

LAW defines the type of distribution

LAW=1: the particles are generated in real space (X, Y, Z) using a hit-or-miss Monte Carlo method within a uniform distribution. Then X', Y', Z' are chosen from within each phase-plane ellipse and Z, Z' are converted to phase and energy.

LAW=2: the particles are generated in a six-dimensional ellipsoid from a hit-or-miss Monte Carlo method within a uniform distribution and Z, Z' are converted to phase and energy.

LAW=3: the particles are generated like with LAW=1, but within the following distribution:

$$\rho(r) = (1 + r^2/2 + r^4/8)\exp(-r^2/2)\,,\, r^2 = X^2 + Y^2 + Z^2$$

NOTE: Such a quasi-uniform distribution avoids discontinuities at the frontiers of the ellipses.

LAW=4: as in LAW=2, but within the Gaussian distribution:

$$\rho(r) = \exp(-r^2/2)$$

LAW=5: The particles are generated randomly in a cylinder having its axis in the Z-direction (e.g. continuous beam). In the transverse directions, the distributions are uniform.

LAW=6: Like for LAW=5, the particles are generated randomly in a cylinder having its axis in the Z-direction (e.g. continuous beam), but in the transverse directions the distributions are Gaussian.

LAW=7: the particles are generated in a three-dimensional ellipsoid (X, Y, Z) from a hit-or-miss Monte Carlo method, whereby X', Y' and Z' are chosen in each phase plane. Z, Z' are converted to phase and energy.

ITWISS = 1: read Twiss parameters for emittance definitions.

ITWISS = 0: read emittance boundaries for an upright ellipsoid.

2) FH IMAX

FH: frequency (Hz)

IMAX: total number of particles (see remark 3)

3) CENTRE (I) (I = 1 to 6)

Center of the beam ellipsoid:

CENTRE (1): horizontal position (cm)

CENTRE (2): angle offset X' (mrad)

CENTRE (3): vertical position (cm)

CENTRE (4): angle offset Y' (mrad)

CENTRE (5): energy offset (MeV) (total energy)

CENTRE (6): phase offset Z (sec)

The following entries are depending on the parameter ITWISS.

If ITWISS = 0, read the following line 4) giving the maximum values for the limits of the random distribution, these limits are in (+ -). In this case, no other entry is required. However, when ITWISS = 0 the beam ellipsoid generated is upright. For a rotated (non-erect) beam ellipsoid use the type code TILT behind the GEBEAM one.

4) XMAX XPMAX YMAX YPMAX DMAX TTMAX

XMAX: horizontal beam extent (X in cm)

XPMAX: horizontal beam angle (X' in mrad)

YMAX: vertical beam extent (Y in cm)

YMAX: vertical beam angle (Y' in mrad)

DMAX: (total) energy spread (dW in MeV)

TTMAX: phase spread (dT in sec)

If ITWISS = 1, read lines 4, 5 and 6 (Twiss parameters for the limits of the random distribution):

4) ALPHAX BETAX EMITX

5) ALPHAY BETAY EMITY

6) ALPHAZ* BETAZ* EMITZ

BETAX and BETAY are in mm/mrad, BETAZ is in deg/keV

Emittances EMITX and EMITY are in mm.mrad, EMITZ is in keV.deg

*NOTE: EMITX, EMITY and EMITZ are corresponding to 4RMS. In the case of LAW=5 and LAW=6, a phase spread of +/-180 deg (corresponding to a continuous beam) will automatically be generated.
In the case of LAW=5 ALPHAZ corresponds to the half energy spread in MeV (total); BETAZ and EMITZ should be set to 0. (but will not be used by DYNAC).
In the case of LAW=6 ALPHAZ corresponds to the half energy spread in MeV (total) and BETAZ to the number of standard deviations for the transverse Gaussian distributions. EMITZ should be set to 0. (but will not be used by DYNAC).

REMARKS:

(1) GEBEAM card must imperatively be followed by INPUT card.

(2) The maximum number of macro particles tested so far with DYNAC (i.e. corresponding to IMAX) is 10000000.

EXAMPLE (the Twiss parameters are read):

```
GEBEAM
3 (LAW) 1 (ITWISS)
0.805E09 (FH in Hz) 9000 (IMAX, in this case 9000 particles)
0. 0. 0. 0. 0. 0. (CENTRE(I), I=1,6)
0.8934 (ALPHAX) 3.6784 (BETAX in mm/mrad) 1.581 (EMITX in mm.mrad)
5.2992 (ALPHAY) 17.424 (BETAY in mm/mrad) 1.562 (EMITY in mm.mrad)
1.9115 (ALPHAZ) 0.0227 (BETAZ in deg/keV) 746.6 (EMITZ in keV.deg)
```

6.1.3 TYPE CODE: RDBEAM

Read the 6D coordinates of particles from file (e.g. 'dynac_in.dst') in ASCII format. Note that the RDBEAM card cannot be used in combination with the INPUT card.

ENTRY:
1) filen

Name of the file containing the particle distribution. Include the full path to this file to ensure that when DYNAC is run from DGUI, it can access the file.

2) IFLAG

IFLAG indicates the structure of the file 'filen'. In addition, the unit used for phase can be chosen. In legacy input files IFLAG may have a value of 100 smaller than the values listed below; it is strongly suggested to update these files by adding 100 to the value of IFLAG. The current numbering was introduced in order to be compatible with WRBEAM (see WRBEAM).

IFLAG=100: File using 6 coordinates for each macro-particle (see remark 3), with phase in radians

IFLAG=101: Like IFLAG=100, but also includes particle number (see remark 3), with phase in radians

IFLAG=102: Like IFLAG=100, but also includes charge state(s) (see remark 3), with phase in radians

IFLAG=103: Like IFLAG=102, but also includes particle number (see remark 3), with phase in radians

IFLAG=109: Like IFLAG=100, but also includes charge state(s) and rest mass (see remark 3), with phase in radians

IFLAG=110: File using 6 coordinates for each macro-particle (see remark 3), with phase in ns

IFLAG=111: Like IFLAG=110, but also includes particle number (see remark 3), with phase in ns

IFLAG=112: Like IFLAG=110, but also includes charge state(s) (see remark 3), with phase in ns

IFLAG=113: Like IFLAG=112, but also includes particle number (see remark 3), with phase in ns

IFLAG=119: Like IFLAG=110, but also includes charge state(s) and rest mass (see remark 3), with phase in ns

3) FREQ TOF

FREQ: frequency (MHz)

TOF: adjustment to be applied on the RF phase (deg)

4) UEM ATM

UEM, ATM: as in INPUT card

5) WINREF Q

WINREF and Q are the input energy and the charge state of the reference

EXAMPLE: (entries for a 2.5 MeV H$^-$ beam)

```
RDBEAM
c:\mycodes\dynac\datafiles\dynac_in.dst (name of file containing the particle distribution)
100 (IFLAG; here for a file with 6 coordinates per macro particle)
402.5 (FREQ in MHz) 0. (TOF in deg)
939.301404 (UEM in MeV)   1 (ATM)
2.5 (WINREF in MeV)  -1 (Q)
```

REMARKS:

(1) After the RDBEAM card, the energy and the time of flight of the COG (center of gravity) are independent from the ones of the reference particle.

(2) The coordinates of the first particle are overwritten by the following on-axis ones (example for IFLAG=100):

X=0., Y=0., X'=0., Y'=0., TOF (of reference), energy (of reference)

(3) Structure of the particle distribution files (e.g. 'dynac_in.dst'):

A) Standard file (IFLAG = 100 or 110):

First line: NPOINT (number of particles), IFLAG, FREQ (frequency in MHz), (Wref(MeV),Wcog(MeV) if abs(IFLAG) ≥ 100) whereby any data on the first line after IFLAG are ignored.

Following lines: For each of the NPOINT particles, the 6 particle coordinates are to be on one line:

X (cm) X'(rad) Y (cm) Y'(rad) phase (rad or ns) energy (MeV)

B) File with particle number (IFLAG = 101 or 111):

First line: NPOINT (number of particles), IFLAG, FREQ (frequency in MHz), (Wref(MeV),Wcog(MeV) if abs(IFLAG) ≥ 100) whereby any data on the first line after IFLAG are ignored.

Following lines: For each of the NPOINT particles, the 8 particle coordinates are to be on one line:

X(cm) X'(rad) Y(cm) Y'(rad) phase(rad or ns) energy(MeV) pn

pn is the particle number, but will be ignored by DYNAC.

C) File with charge state(s) (IFLAG = 102 or 112):

First line: NPOINT (number of particles), IFLAG, FREQ (frequency in MHz), (Wref(MeV),Wcog(MeV) if abs(IFLAG) ≥ 100) whereby any data on the first line after IFLAG are ignored.

Following lines: For each of the NPOINT particles, the 6 particle coordinates are followed by the electric charge state on one line:

X(cm) X'(rad) Y(cm) Y'(rad) phase(rad or ns) energy(MeV) Q

Q is the charge state.

D) File with charge state(s) (IFLAG = 103 or 113):

First line: NPOINT (number of particles), IFLAG, FREQ (frequency in MHz), (Wref(MeV),Wcog(MeV) if abs(IFLAG) ≥ 100) whereby any data on the first line after IFLAG are ignored.

Following lines: For each of the NPOINT particles, the 6 particle coordinates are followed by the electric charge state on one line:

X(cm) X'(rad) Y(cm) Y'(rad) phase(rad or ns) energy(MeV) Q pn

Q is the charge state and pn is the particle number. The particle number will be ignored by DYNAC.

E) File with charge state(s)and rest mass (used in order to be compatible with some other beam dynamics programs) (IFLAG = 109 or 119):

First line: NPOINT (number of particles), IFLAG, FREQ (frequency in MHz), (Wref(MeV),Wcog(MeV) if abs(IFLAG) ≥ 100) whereby any data on the first line after IFLAG are ignored.

Following lines: For each of the NPOINT particles, the 8 particle coordinates are to be on one line:

X(cm) X'(rad) Y(cm) Y'(rad) phase(rad or ns) energy(MeV) Q Mo

Q is the electric charge state and Mo is the rest mass, but both of these will be ignored by DYNAC.

(4) The maximum number of macro particles tested so far with DYNAC (i.e. corresponding to IMAX) is 10000000.

(5) One can read the particle distribution file in binary format instead; see type code BINDIST.

6.1.4 TYPE CODE: ETAC

Generate different charge states for a multi-charge state beam.

NOTE: if used, this card is to be placed after the INPUT or RDBEAM card.

ENTRY:

1) N

N: number of charge states (maximum 20 different charge states)

If N = 0, read the charge state for each particle from file:

2) filen

Name of an ASCII file containing the particle charge state distribution. Include the full path to this file to ensure that when DYNAC is run from DGUI, it can access the file. This file should contain on the first line the total number of particles (NTOT), followed by NTOT lines with each line containing the charge state of each of these particles. (Fortran Unit 56 is used for this file)

If N > 0, DYNAC will generate the charge states; the charge states are distributed among the particles using a hit-or-miss Monte Carlo method.
Read the N following lines:

2) CHARGE (1) PCENT (1)   EOFF (1)
. . . .
N) CHARGE (N) PCENT (N)   EOFF (N)

CHARGE (I): charge state for charge state number I

PCENT (I): percentage for charge state number I

EOFF(I): absolute energy offset of charge state with respect to the
COG (MeV)

EXAMPLE: 3 charge states (i.e. 24, 25 and 26), each representing ~33% of all particles.

```
ETAC
3 (number of charge states N)
24 (CHARGE state of 24) 33 (PCENT, i.e. 33% of 24+) -0.01 (EOFF in MeV)
25 (CHARGE state of 25) 34 (PCENT, i.e. 34% of 25+)  0.   (EOFF in MeV)
26 (CHARGE state of 26) 33 (PCENT, i.e. 33% of 26+)  0.01 (EOFF in MeV)
```

EXAMPLE: Read charge states from file

```
ETAC
0 (N; i.e. read from file)
c:\mycodes\dynac\datafiles\my_charge_state_file.txt
```

REMARKS:
(1) The reference charge state is defined by INPUT
(2) One can also use the ETAC card to simulate different masses. In this case, the reference mass is the one defined by INPUT. The different masses then are defined as having a mass identical to the reference mass and by in addition listing charge states for them in ETAC such, that the charge to mass ratio corresponds to that of the mass of interest. Example: consider the case of masses 90, 91 and 92 (reference mass) with all three masses having charge state 1. The CHARGE value for these would be 1.02222, 1.01099 and 1. respectively. You may then need to adjust the energies as well; this can be done with EOFF.

6.2 OPTICAL LENSES

6.2.1 TYPE CODE: BMAGNET

Transport through a bending magnet (second-order matrix formalism if SECORD card is preceding this one in the command list).

ENTRY:

1) NSEC

NSEC: number of sectors in which the dipole is to be divided. In the case of a multi-charge state beam in the magnet and/or with the occurrence of space charge computations in the magnet, this parameter is required to be larger than 1 (see remarks).

2) ANGL, RMO, BAIM, XN, XB

ANGL: bending angle of the central trajectory (deg)

RMO: bending radius of the central trajectory (cm)

BAIM: magnetic field (kG)
If BAIM = 0.0, the rigidity and field will be calculated based on the
reference particle.
If BAIM > 0.0, the rigidity will be based on the field entered here.

XN: field gradient n (dimensionless parameter corresponding to the parameter n in the code TRANSPORT, see remark (1))

XB: second order field gradient beta (dimensionless parameter corresponding to the parameter beta in the code TRANSPORT, see remark (1))

3) PENT1 RAB1 EK1 EK2 APB(1)

PENT1: angle of pole-face rotation at the entrance (deg), see remark (2) for conventions.

RAB1: curvature of the entrance pole-face (cm), see remark (2) for conventions

EK1: An integral related to the extent of the fringe field (dimensionless parameter corresponding to the parameter K1 in the code TRANSPORT, see remark 3).

EK2: A second integral related to the extent of the fringe field (dimensionless parameter corresponding to the parameter K2 in the code TRANSPORT, see remark 3).

APB(1): vertical half aperture at the entrance (cm)

4) PENT2 RAB2 SK1 SK2 APB(2)

PENT2: angle of pole-face rotation at the exit (deg), see remark (2) for conventions.

RAB2: curvature of the exit pole-face (cm)

SK1: An integral related to the extent of the fringe field (dimensionless parameter corresponding to the parameter K1 in the code TRANSPORT, see remark 3).

SK2: A second integral related to the extent of the fringe field (dimensionless parameter corresponding to the parameter K2 in the code TRANSPORT, see remark 3).

APB(2): vertical half aperture at the exit (cm)
REMARKS:

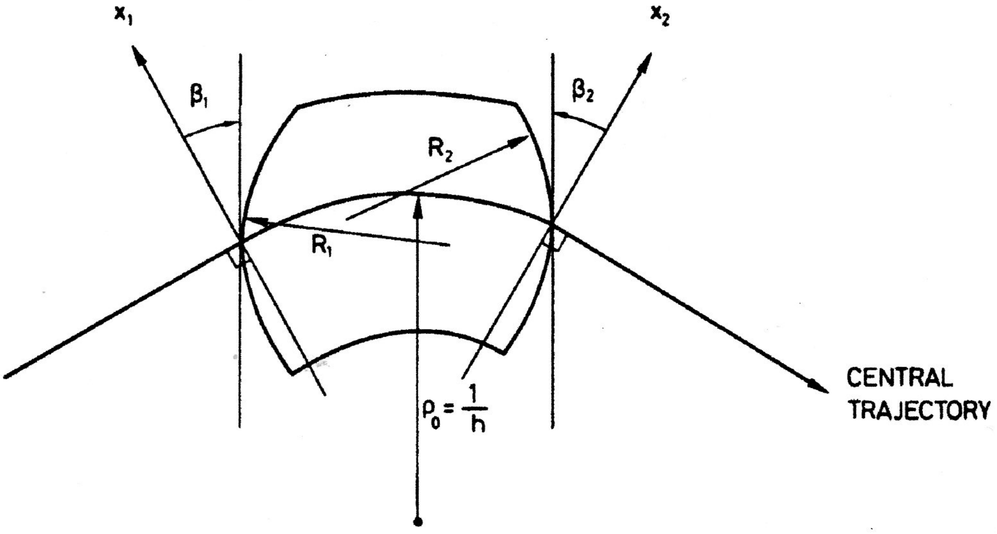
(1) The mid-plane field $B_y(x, y=0, t)$ is expressed in terms of the dimensionless quantities n and β as:

$$B_y(x,y=0,t)=B_y(0,0,t)\left[1-\mathrm{nhx}+\beta h^2x^2\right] \quad , \quad h=1/\rho_0$$

$$n=-\left[\frac{1}{\mathrm{hB}_y}\left(\frac{\partial B_y}{\partial x}\right)\right]_{x=y=0} \quad , \quad \beta=\left[\frac{1}{2h^2B_y}\left(\frac{\partial^2 B_y}{\partial x^2}\right)\right]_{x=y=0}$$

(2) The sign conventions are the ones in use in the code TRANSPORT (see SLAC Report-75):
- A positive bend is to the right looking in the direction of particle travel, a negative one to the left. The ZROT card (see this type code for examples) may be used to represent bends in other directions.
- In terms of sign conventions, to change the input parameters of a bending magnet set up for a proton beam to an equivalent H$^-$ beam and equivalent bending, one needs to change the sign on both ANGL and RMO
- A positive sign of the angle of rotation on either entrance (PENT1) or exit (PENT2) pole-faces corresponds to a non-bend plane focusing action and bend plane defocusing action.
- A positive sign of radius RAB1, RAB2 implies a convex curvature (it represents a negative sextupole component, see SLAC Report-75, page 71).

The sign conventions are displayed in the following figure.



FIELD BOUNDARIES FOR BENDING MAGNETS

The TRANSPORT sign conventions for x, β, R and h are all positive as shown in the figure. The positive y direction is out of the paper. Positive β's imply transverse focusing. Positive R's (convex curvatures) represent negative sextupole components of strength $S = (-h/2R)\sec^3\beta$. (See SLAC-75, page 71.)

(3) If APB(1) or/and APB(2) is null, the program inserts a default value zero for EK1 and EK2, as well as for SK1 and SK2.
If APB(1) or/and APB(2) is not zero and EK1 or/and SK1 is negative, the program inserts a default value of EK1 = 0.5 or/and of SK1 = 0.5.

Typical values of EK1/SK1 and EK2/SK2 are given below for four types of fringing boundaries:

a) A linear drop-off of the field:
K1 = b/6g   K2 = 3.8
where b is the extent of the linear fringe field and g is the vertical half aperture.

b) A clamped “ROGOWSKI” fringe field:
K1 = 0.4 K2 = 4.4

c) An unclamped “ROGOWSKI” fringe field:
K1 = 0.7 K2 = 4.4

d) A square-edge non-saturating magnet:
K1 = 0.45 K2 = 2.8

(4) If in the input file BMAGNET is preceded by the combination of GEBEAM, INPUT and ETAC, for space charge calculations one has to place the SCDYNAC card after ETAC.

(5) In the case of a multi-charge state beam, NSEC should be set to a value larger than 1 (at least 5 is advised).

(6) In the case of space charge calculations, NSEC should be set to an **even value** larger than 1. **With a multi-charge state beam, only SCHEFF should be used for space charge calculations.**

EXAMPLES:

a) The case of a single-state charge beam and without space charge computations in a typical first-order transport for a wedge magnet whose total bend is 10 deg and bending radius of central trajectory is 40 cm:

```
BMAGNET
1  (NSEC)
10. (ANGL in deg) 40. (RMO in cm) 0. (BAIM in kG) 0. (XN) 0. (XB)
0. (PENT1 in deg) 0. (RAB1 in cm) 0.46(EK1) 2.75(EK2) 0. (APB(1))
0. (PENT2 in deg) 0. (RAB2 in cm) 0.46(SK1) 2.75(SK2) 0. (APB(2))
```

b) The case of a multi-charge state beam and space charge computations in a bending magnet having the following characteristics:

- Bend angle of central trajectory: 90 deg
- Bending radius of the central trajectory: 100 cm
- Field gradient: XN = 0.25
- Second field gradient: XB = 0.25
- Rotation angle of entrance and of exit face-poles: 27.71 deg
- First fringe field coefficients at entrance and exit: EK1, SK1 = 0.46
- Second fringe-field coefficient: EK2, SK2 = 2.75
- Curvature of the entrance and exit pole-faces: RAB1, RAB2 = 27937 cm
- Vertical half aperture at entrance and exit: APB(1), APB(2) = 3.3 cm:

```
SECORD (initiate second-order matrix formalism)
. . . . . .
SCDYNAC (with 3, initiate space charge computations, see this Type Code)

BMAGNET
6  (NSEC)
90. (ANGL in deg) 100. (RMO in cm) 0. (BAIM in kG) 0.25 (XN) 0.25 (XB)
27.71(PENT1 in deg) 27937(RAB1 in cm) 0.46(EK1) 2.75(EK2) 3.3(APB (1))
27.71(PENT2 in deg) 27937(RAB2 in cm) 0.46(SK1) 2.75(SK2) 3.3(APB (2))
```

6.2.2 TYPE CODE: DRIFT

Transport through a free drift space

ENTRY:

1) LD

LD: length of the drift space (cm)

REMARKS:

(1) Negative values can be used. In this case, no space charge computation is possible.

(2) Space charge computations are automatically made with respect to the middle of the drift (see type code SCDYNAC).

(3) If the drift length is less than 0.00001 cm (i.e. 10E-7 m), space charge computation is automatically disabled in the drift.

6.2.3 TYPE CODE: QUADRUPO

Transport through a magnetic quadrupole (second-order matrix formalism if SECORD card precedes this one in the command list).

ENTRY:

1) XL BQ RG

XL: effective field length (cm)

BQ: field at pole tip (kG) (see remark (1) for sign convention)

RG: radius of the circle tangent to the pole tips (cm)

REMARKS:
(1) If the magnetic momentum $(B\rho)=3.356\,E_r\sqrt{\gamma^2-1}/q$ (kG-cm) (where: E_r(Mev) is the rest mass, γ is the relativistic gamma and q is the charge state) is positive, a positive field B (kG) implies a horizontal focusing and a negative field a vertical focusing. Otherwise, if $(B\rho)$ is negative the sign conventions for the field B must be inverted.

2) Space charge computations (if enabled) are automatically made with respect to the middle of the quadrupole (see type code SCDYNAC).

6.2.4 TYPE CODE: SOLENO

Transport in a solenoid (second-order matrix formalism if SECORD card is preceding this one in the command list). The fringe-field necessary to produce the focusing is included in the transport matrix.
Either the strength $k=(B)(1/2B\rho)$ or the field B is possible as input parameter (see the parameter ARG).

ENTRY:

1) IMKS XS ARG

IMKS: Integer flag defining the type of entry ARG (see ARG)

XS: effective length L (cm)

ARG: strength k or field B

If IMKS = 0, ARG is the strength $k=(B)(1/2B\rho)$ (cm-1), otherwise ARG is the field B (kG) inside the solenoid (see QUADRUPO for the definition of the magnetic momentum $(B\rho)$).

REMARKS:

(1) A positive sign of k causes a clockwise rotation about the z-axis by an angle $\alpha=\mathrm{L}\mathrm{k}$.
(2) Space charge computations (if enabled) are automatically made with respect to the middle of the solenoid (see type code SCDYNAC).

6.2.5 TYPE CODE: STEER

Transport in a thin steering element. The steering element can be magnetic or electrostatic, depending on the NVF entry.

ENTRY:

1) FLD NVF

FLD: Integral of magnetic field $\int \mathrm{Bdz}$ (T*m) or of electric field (m*kV/m), whereby the latter corresponds to plate voltage * plate length/plate separation.

NVF = 0: bend plane focusing action (horizontal, magnetic)

NVF = 1: no-bend plane focusing action (vertical, magnetic)

NVF = 2: bend plane focusing action (horizontal, electrostatic)

NVF = 3: no-bend plane focusing action (vertical, electrostatic)

REMARK: space charge computations and second-order matrix formalism are not possible in this steering magnet.

6.2.6 TYPE CODE: SEXTUPO

Sextupole magnet, typically used (in combination with SECORD card) to modify second-order aberrations in a beam transport system.
Either the strength k_s^2 or the field B is possible as input parameter (see the entry ARG).

ENTRY:

1) IMKS2 ARG XSX RSX

IMKS2: flag defining the entry ARG (see ARG)

ARG: strength k_s^2 or field B
If IMKS2 = 0 then ARG is the strength $k_s^2=(B/R^2)(1/B\rho)$ (cm-3), otherwise
ARG is the field B (kG) at pole tip (see QUADRUPO for the definition of the magnetic momentum$(B\rho)$).

XSX: effective field length (cm)

RSX: radius R of the circle tangent to the pole tips (cm)

REMARKS:

(1) If the SECORD card is not included in the command list, the sextupole acts as a drift.
(2) Space charge computations (if enabled) are automatically made with respect to the middle of the sextupole (type code SCDYNAC)
(3) A negative strength k_s^2 is equivalent to a convex curvature in bending magnet pole-faces (see SLAC Report-75, page 71).

6.2.7 TYPE CODE: QUADSXT

The quadrupole field is associated with a sextupole field (second-order matrix formalism if SECORD card is included in the command list).
Either strengths or fields are possible as input parameters (see entries ARGS and ARGQ).

ENTRY:
1) IKSQ ARGS ARGQ XL RG

IKSQ: flag defining the entries ARGS and ARGQ

ARGS: strength $k_s^2=(B/R^2)(1/B\rho)$ (cm-3) or field B (kG) of sextupole:
IKSQ = 0 then ARGS =k_s^2 (cm-3), otherwise ARGS =B (kG) (see QUADRUPO for the definition of the magnetic momentum$(B\rho)$).

ARGQ: strength $k_q^2=(B/R)(1/B\rho)$ (cm-2) or field B (kG) of quadrupole:
IKSQ = 0 then ARGQ =k_q^2 (cm-3), otherwise ARGQ = B (kG) at pole tip.

XL: effective length of the lens (cm)

RG: radius of the circle tangent to the pole tips (cm)

REMARKS:

(1) If ARGS = 0, QUADSXT acts like a magnetic quadrupole.

(2) If SECORD card is included in the list and ARGQ = 0, QUADSXT performs like a pure sextupole.

(3) Space charge computations are automatically made with respect to the middle of the effective field length (see type code SCDYNAC).

6.2.8 TYPE CODE: SOQUAD

Solenoid associated with a quadrupole (second-order matrix formalism if SECORD card is in the command list). Either strengths or fields are possible as input parameters (see entries ARGS and ARGQ).

ENTRY:

1) IKSQ ARGS ARGQ XL RG

IKSQ: flag defining the entries ARGS and ARGQ

ARGS: strength or field of solenoid.
IKSQ = 0, then ARGS is the strength $k=(B)(1/2B\rho)$ (cm-1), otherwise ARG is the field *B* (kG) inside the solenoid.

ARGQ: strength or field of quadrupole.
IKSQ = 0, then ARGQ is the strength $k_q^2=(B/R)(1/B\rho)$ (cm-2), otherwise ARGQ is the field *B* (kG) at the pole tips.

XL: effective length L of the lens (cm)

RG: radius of the circle tangent to the pole tips (cm)

REMARKS:

1) If ARGQ = 0, SOQUAD acts like a solenoid, if ARGS = 0 then SOQUAD performs like a magnetic quadrupole.

2) Space charge computations (if enabled) are automatically made with respect to the middle of the effective field length (see type code SCDYNAC).

(3) A positive sign of k causes a clockwise rotation about the z-axis by an angle $\alpha=\mathrm{L}k$. A positive sign of k_q^2 implies horizontal focusing.

6.2.9 TYPE CODE: FDRIFT

Transport in a drift length. FDRIFT card is appropriate for multiple space charge computations in long drifts.

ENTRY:

1) XL NPART IMIT

XL: total drift length (cm)

NPART: Number of elementary drifts considered in the total drift length.

IMIT = 0: the characteristics of the beam are not printed after each elementary drift in the file ‘dynac.short’.

IMIT = 1: the characteristics of the beam are systematically printed in the file ‘dynac.short’ after each elementary drift.

REMARK: Space charge computations (if enabled) are automatically made with respect to the middle of each elementary drift (see type code SCDYNAC).

EXAMPLE: A drift of 200 cm length is divided in 5 elementary drifts, the characteristics of the beam are not to be printed after each of these elementary drifts:

```
FDRIFT
200. (XL in cm)   5 (NPART)   0 (IMIT)
```

6.2.10 TYPE CODE: FSOLE

Transport in a solenoid with an arbitrary field read from file (second-order matrix formalism with SECORD card is in the command list).

ENTRY:
1) filen

Name of an ASCII file containing the arbitrary solenoid field (see remark 3). Several fields may be contained in one file and/or several files on consecutive FSOLE cards may be used. Include the full path to this file to ensure that when DYNAC is run from DGUI, it can access the file.

2) BFACT NPART

BFACT: dimensionless factor adjusting the magnitude of the field read.
The effective field is given by:

(Effective field (kG)) = field (kG)* BFACT

NPART: number of elementary field lengths considered in the total field length (see remark (1)).

REMARKS:

(1) The total field length is divided in NPART elementary field lengths. Each of these elementary lengths is considered as a solenoid having a uniform field equal to the one at the middle of the elementary field length.

(2) If space charge computations are required in the solenoid (see type code SCDYNAC), NPART **must** be an **EVEN INTEGER NUMBER**. In this case, NPART/2 positions of space charge computations (if enabled) are automatically defined in the solenoid.

(3) The field is read from file in the form Z (in m) and B(Z) (in kG) as follows:

First field:

n1 ← number of points (Z, B(Z))

Z (1)   B (Z (1))
. . . .

Z (n1)   B (Z (n1))

Second field:

n2

Z (1)   B (Z (1))
. . . .

Z (n2)   B (Z (n2))

Third field:

. . . .

If n2 = 0, the file is automatically rewound, and all following solenoids make use of the same field.

EXAMPLE:

FSOLE ← read first field from file
BFACT (= 4.247) NPART (= 5)
. . . .
FSOLE ← read second field
BFACT (= 4.247) NPART (= 5)
. . . .

6.2.11 TYPE CODE: SECORD

Following the SECORD card, second order transport matrix formalism in optical lenses is enabled.

ENTRY: none

REMARK:
(1) The default in DYNAC is first order

6.2.12 TYPE CODE: RASYN

Enable synchrotron radiation (for electrons) in bending magnets following the RASYN card.

ENTRY: none

REMARK: When the central trajectory of the bending magnet is large, it is recommended to divide this bending magnet in a succession of smaller elementary bending magnets. This will allow for a better accuracy of the synchronous radiation computations.

EXAMPLE:

Consider a bending magnet having the following characteristics:

- Bend angle of the central trajectory: 40 deg

- Radius of central trajectory: 5000 cm

- $n=0.5, \beta=0.$

- Angle of pole-face rotation at entrance and exit: 30 deg

- Curvature of pole-faces: 1000 cm

- Extents of the fringe-field: $K_1=0.7, K_2=0$

- Vertical half aperture at entrance and exit: 10 cm

The entry for this bending magnet (see type code BMAGNET) would look like:

```
BMAGNET
  1  (NSEC)
  40.(ANGL in deg)  5000.(RMO in cm)  0. (BAIM in kG) 0.5(XN) 0.(XB)
  30.(PENT1 in deg) 1000.(RAB1 in cm) 0.7(EK1) 0.(EK2) 10.(APB (1))
  30.(PENT2 in deg) 1000.(RAB2 in cm) 0.7(SK1) 0.(SK2) 10.(APB (2))
```

References:

[1] M.Sands: The Physics of Electron Storage Rings, SLAC-R121
[2] E.Tanke and S.Valero: Synchrotron radiation effects for electrons in the code DYNAC, (accessible on the website).

6.2.13 TYPE CODE: QUAELEC

Transport in an electrostatic quadrupole (second-order matrix formalism if SECORD card precedes this one in the command list).

ENTRY:

1) XLQUA VOLT RS

XLQUA: effective field length (cm)

VOLT: electric voltage at pole tip (kV) (see remark (1) for sign convention)

RS: radial distance of pole tip from central axis (cm)

REMARKS:

(1) If the charge state is positive, a positive electric voltage implies a horizontal focusing and a negative electric voltage implies a vertical focusing. Otherwise, if the charge state is negative the sign convention for the electric voltage V is inverted.

2) Space charge computations (if enabled) are automatically made with respect to the middle of the quadrupole (see type code SCDYNAC).

6.2.14 TYPE CODE: QUAFK

Transport in an electrostatic or a magnetic quadrupole when the strength k_q^2 is given as input parameter (second-order matrix formalism if SECORD card precedes this one in the command list).

ENTRY:

1) ITYQU ARG XL RS

ITYQU: flag defining the type of quadrupole (magnetic or electrostatic).
ITYQU = 0, then it is an electrostatic quadrupole, otherwise it is a magnetic quadrupole

ARG: strength k_q^2 (cm-2), see remark 1.

XL: effective field length (cm)

RS: radial distance of pole tip from central axis (cm)

REMARKS:

(1) The strength of an electrostatic quadrupole is given by: $k_q^2 = 2\left(V/r^2\right)\left(1/\left(E\rho\right)\right)$ *(cm-2),* where V *(kV)* is the electric voltage at pole tip and $\left(E\rho\right) = 1000\, E_{\text{rest}}\left(\gamma^2 - 1\right)/q$ (kV) is the electric momentum with E_{rest} the rest mass, γ the relativistic gamma and q the charge state.

(2) A positive strength K implies horizontal focusing and a negative K vertical focusing.

(3) Space charge computations (if enabled) are automatically made with respect to the middle of the quadrupole (see type code SCDYNAC).

6.2.15 TYPE CODE: EDFLEC

Transport in a spherical electrostatic dipole (only first order matrix; currently no second-order matrix formalism available).

ENTRY:

1) NSEC

NSEC: number of sectors in which the dipole is to be divided. In the case of a multi-charge state beam in the dipole and/or with the occurrence of space charge computations in the dipole, this parameter is required to be larger than 1.

2) RMO ANGL RADII ELECF

RMO: bending radius (cm)

ANGL: bend angle (deg)

RADII: vertical radii of curvature (cm)

ELECF: Electric field to be used in the calculations (kV/cm)

REMARKS:
(1) By setting RADII to a large number, one can approximate the case of a cylindrical electrostatic dipole
(2) Contrary to the magnetic dipole in DYNAC, the electrostatic dipole in DYNAC bends by default to the left.
(3) The nominal field En is calculated by DYNAC based on RMO, ANGL and RADII. If ELECF<0 the field En will be used, otherwise ELECF will be used.

6.2.16 TYPE CODE: FIRORD

Following the FIRORD card, first order transport matrix formalism in optical lenses is enabled.

ENTRY: none

REMARK:
(1) The default in DYNAC is first order

*6.2.17 TYPE CODE: WIENANA, **beta version***

Transport through a Wien filter, based on an analytical method. Use the type code WIENNUM if you wish it to be based on a numerical method. The subroutines related to WIENANA are based on routines in the program ZGOUBI and in particular those associated with the SEPARA card in ZGOUBI. These routines have been copied and adapted with kind permission from François Méot, formerly at BNL, Upton, NY, USA. Some more testing and bench-marking is required of this adaptation as implemented in DYNAC, hence this is marked as ***beta version***. It can be used to simulate a chopper (see remark 2).

ENTRY:

1) HV XL E B

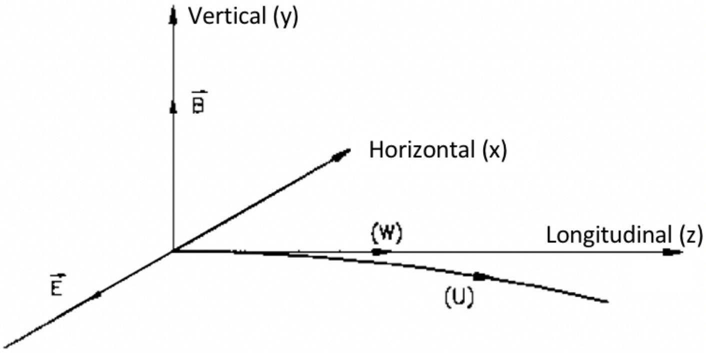
HV: mode of operation:
HV = 0 Element inactive (i.e. will function as a drift)
HV = 1 Horizontal separation
HV = 2 Vertical separation

XL: length of the separator (cm)

E: electric field (V/m)

B: magnetic field (T)

REMARKS:
(1) A Wien filter is an electrostatic separator in which an electrical field and a magnetic field are applied, whereby these two fields are perpendicular to each other (see figure below).



Horizontal separation between a wanted particle, (W), and an unwanted particle, (U). (W) undergoes a linear motion while (U) undergoes a cycloidal motion.

The electric and magnetic field intensities E_0 and B_0 in the central, uniform field region, normally satisfy the relation

$$B_0 = -E_0 / \beta_w c$$

for the selection of "wanted" particles of velocity $\beta_w c$.

(2) One can use the WIENANA card to simulate a chopper by setting the B field to zero.
EXAMPLE: A horizontally deflecting chopper with (optional) user defined name LEBT-CHOP-1, length of 15 cm and applied voltage of 40 kV

```
WIENANA LEBT-CHOP-1
1 (HV) 15. (XL) 40000. (E) 0. (B)
```

*6.2.18 TYPE CODE: WIENNUM, **beta version***

Transport through a Wien filter; based on a numerical method. Use the type code WIENANA if you wish it to be based on an analytical method. The subroutines related to WIENNUM are based on routines in the program ZGOUBI and in particular those associated with the WIENFILT card in ZGOUBI. These routines have been copied and adapted with kind permission from François Méot, formerly at BNL, Upton, NY, USA. Some more testing and bench-marking is required of this adaptation as implemented in DYNAC, hence this is marked as ***beta version***.

ENTRY:

1) HV XL E B

HV: mode of operation:
HV = 0 Element inactive (i.e. will function as a drift)
HV = 1 Horizontal separation
HV = 2 Vertical separation

XL: length of the separator (cm), i.e. the distance between the effective field boundaries

E: electric field (V/m)

B: magnetic field (T)

2) XIE FFIE FFIB

XIE: integration zone extent at the input (cm)

FFIE: fringe field extent of the electrical field at the input (cm), (≃ gap height)

FFIB: fringe field extent of the magnetic field at the input (cm), (≃ gap height)

3) CIE0 CIE1 CIE2 CIE3 CIE4 CIE5

CIE0–CIE5 Entrance face fringe field coefficients for E (dimensionless)

4) CIB0 CIB1 CIB2 CIB3 CIB4 CIB5

CIB0–CIB5 Entrance face fringe field coefficients for B (dimensionless)

5) XOE FFOE FFOB

As for XIE FFIE FFIB under 2), but for the exit face

6) COE0 COE1 COE2 COE3 COE4 COE5

COE0–COE5 Exit face fringe field coefficients for E (dimensionless)

7) COB0 COB1 COB2 COB3 COB4 COB5

COB0–COB5 Exit face fringe field coefficients for B (dimensionless)

8) XPAS

XPAS: integration step (cm)

REMARKS:
(1) This model is defined from the expression of the 2-D field and derivatives in the median plane. 3-D extrapolation of these off the median plane is drawn from Taylor expansions and Maxwell's equations.

(2) The length XL of the element is the distance between its entrance and exit Effective Field Boundaries (EFB). The electric and magnetic field intensities E_0 and B_0 in the central, uniform field region, normally satisfy the relation

$B_0 = -E_0 / \beta_w c$

for the selection of “wanted” particles of velocity $\beta_w c$. Ray-tracing in field fall-off regions extends over a distance XIE (XOE) beyond the entrance (exit) EFB by means of prior and further automatic change of frame.

(3) There are four fringe field descriptions that allow for the description of the entrance and exit fringe fields outside the uniform field region: one pair (B and E) for the entrance, and one pair (B and E) for the exit. Each description consists of a coefficient λ (fringe field extent: FFIE, FFIB for the entrance/input and FFOE, FFOB for the exit/output) and the coefficients $C_0 - C_5$, following the model [1]:

$$F = 1 / (1 + \exp (P(s)))$$

with P(s) defined as:

$$P(s) = C_0 + C_1 (s/\lambda) + C_2 (s/\lambda)^2 + C_3 (s/\lambda)^3 + C_4 (s/\lambda)^4 + C_5 (s/\lambda)^5$$

and s as the distance to the EFB. When fringe fields overlap inside the element (i.e., XL ≤ XIE + XIO), the field fall-off is expressed as

$$F = FE + FS - 1$$

where FE (FS) is the value of the coefficient with respect to the entrance (exit) EFB.

(4) If $\lambda_E = 0$ ($\lambda_S = 0$) for either the electric or magnetic component, then both are considered as sharp edge fields and XE(XS) is forced to zero (for the purpose of saving computing time). In this case, the magnetic wedge angle vertical first order focusing effect is simulated at entrance and exit by a kick $P2 = P1 - Z1 \tan(\varrho/\rho)$ applied to each particle. Here P1, P2 are the vertical angles upstream and downstream the EFB, Z1 the vertical particle position at the EFB, ρ the local horizontal bending radius and ϱ the wedge angle experienced by the particle. ϱ depends on the horizontal angle T. This is not done for the electric field however, it is advised not to use a sharp edge electric dipole model since this entails non symplectic mapping, and in particular precludes accounting for momentum effects of the non-zero longitudinal electric field component.

[1] H.A. Enge, Deflecting magnets in Focusing of Charged Particles, ed. A. Septier, Vol. II, pp 203-264, Academic Press Inc.,1967.

6.3 ACCELERATING GAPS, CAVITIES, BUNCHERS, ELECTRON GUN, RFQ, STRIPPER

6.3.1 TYPE CODE: CAVMC

Describe the motion of particles in complex and/or long accelerating elements (e.g. helix, multi-cell cavity etc). The description is based on an analytical method, which is fast, **but is not to be used for multi-charge state beams**. For multi-charge state beams, refer to type code CAVNUM.

Due to the use of the equivalent field, the RF phase in CAVMC will require a different setting than the RF phase PHREF say in CAVNUM. One method to obtain the phase required in CAVMC is to scan the RF phase to find the maximum energy gain. Again, note that this value (call it DPH) will not be equal to 0 deg due to the use of the equivalent field. The value for the RF phase in CAVMC can then be set to PHREF+DPH. Another method is to shift the RF phase until the same energy gain as with CAVNUM is obtained.

NOTE: The axial field of the cavity can be read from file (e.g. 'field.txt'; see type code FIELD) or can be read in the command list in the form of a series Fourier expansion (see type code HARM).

ENTRY:

1) IDUM

IDUM: cavity number (dummy variable in the command list for convenience; the code automatically counts the cavities number).

2) XESLN DPHASE FFIELD ISEC IDUMMY

XESLN (cm): difference between the length of the axial field read (see type code FIELD or type code HARM) and the effective physical length of the cavity. Space charge computations are acting on the length:

(Space charge length) = (length of the axial field) – XESLN

Note: When the axial field length is greater than the effective physical length of the cavity, in order to respect the total length of the machine the cavity must be preceded by a negative drift of length: XESLN, see example 2 below.

DPHASE (deg): phase offset (deg) relative to the RF phase at the crest (i.e. phase giving the maximum energy gain).

FFIELD (in percent): relative level of the electric field:

(Effective electric field) = (electric field)*(1+FFIELD/100)

ISEC = 0: the RF phase is taken at the entrance of the accelerating element.

ISEC ≠ 0: The RF phase is adjusted (by the code) at the middle of the accelerating element.

IDUMMY is a dummy parameter; it allows for easy switching between CAVMC and CAVNUM routines.

REMARKS:

(1) The RF phase may be in connection (or not) with the time of flight. In this case, the code can automatically adjust the phase offset with respect to the T.O.F. (see type code TOF)

(2) The reference particle can be disconnected (or not) from the COG of the bunch (see type code REFCOG)

(3) Space charge computations can be made with respect to an arbitrary position in the cavity (see type code SCPOS). The default position is at the middle of the cavity.

EXAMPLE 1:

```
CAVMC
12 (IDUM, dummy variable)
0. (XESLN in cm)  -30. (DPHASE in deg)  0 (FFIELD)  0 (ISEC) 1 (IDUMMY)
```

In this example, if the TOF is passive the RF phase is given by:

(RF Phase) = (crest phase) + (-30 deg)

When the TOF is activated and adjustments are required (see type code TOF), one will have:

(RF Phase) = (crest phase) + (-30 deg + adjustments) + TOF

When adjustments are not required and the TOF is activated:

(RF Phase) = (crest phase) + (-30 deg) + TOF

EXAMPLE 2: the axial field is read in the form of a series Fourier expansion (see type code HARM). The axial field length is 7.605 cm and the physical length of the accelerating gap is 3.435 cm:

```
HARM
7.605 101.28E06   0.0586330 2
18
(Follow the 18 Fourier coefficients)
CAVMC
1
4.175(XESLN in cm)  0.   0.  0   1
DRIFT
-4.175
. . . . . . . . . . . . . . . . .
```

6.3.2 TYPE CODE: CAVNUM

Describes the motion of particles in complex and/or long accelerating elements (e.g. helix, multi-cell cavity etc). The description is based on an accurate and relatively fast specific numerical method (see remarks 3 and 4), which is valid both for single- and multi-charge state beams. For single-charge state beams, type code CAVMC offers a faster (though somewhat less accurate) alternative.

NOTE: The axial field of the cavity can be read from file (e.g. 'field.txt'; see type code FIELD) or can be read in the command list in the form of a series Fourier expansion (see type code HARM).

ENTRY:
1) IDUM

IDUM: cavity number (dummy variable in the command list for convenience; the code automatically counts the cavities number).

2) DUMMY DPHASE FFIELD INTRVL IELEC

DUMMY (float) is a dummy variable

DPHASE (deg): phase offset (deg) relative to the RF phase at the crest (i.e. phase giving the maximum energy gain).

FFIELD (in percent): relative level of the electric field:

(Effective electric field) = (electric field)*(1+FFIELD/100)

INTRVL (integer): number of integration intervals per cell in the numerical routine (see remark 6)

IELEC (integer), see remark 4:

IELEC = 0 acceleration for non-relativistic particles with Erest < 1 MeV
Otherwise IELEC ≠ 0

REMARKS:
(1) One can choose to connect the RF phase with the time of flight. In this case, the code can automatically adjust the phase offset with respect to the T.O.F. (see type code TOF)

(2) One can choose to connect or disconnect the reference particle from the COG of the bunch (see type code REFCOG)

(3) One focuses on a numerical "step-by-step" method based on the 5 points Bode's rule. The interval of size *h* in the azimuthal direction *z* is divided in 4 parts of equivalent lengths:

$$z_1 - z_0 = h/4, z_2 - z_0 = h/2, z_3 - z_0 = 3h/4, z_4 - z_0 = h$$

The 5 points Bode's rule is as follows:

$$\int_{z_0}^{z_4 = z_0 + h} f(z)\,\mathrm{dz} = \frac{h}{90}\left[7f(z_0) + 32f(z_1) + 12f(z_2) + 32f(z_3) + 7f(z_4)\right]$$

Such a process is very convenient when the shape of the electric field *E(z)* becomes complex, since one has 4 positions of *E(z)* for each interval of size *h*.
(Note: an improvement has been made in the code in that it uses the 6 points Bode's rule, but the principle is the same as the one explained above)

The transverse motion can be derived from an integration of the equation of the type:

$$\frac{d(\mathrm{mv}_r)}{\mathrm{dt}} = q(E_r - v_z B_\theta)$$

After integration over the interval of size *h*, the transverse momentum is changed by the amount $\Delta(mv_r)$ and the variation in slope *r'* becomes:

$$\Delta r' = \frac{\Delta(\mathrm{mv}_r)}{\mathrm{mv}_z} - \frac{\mathrm{mv}_r}{\mathrm{mv}_z}\Delta(\mathrm{mv}_z) + ...$$

The extra-terms are due to the fact that *r* and *r'* are not canonically conjugate, the conjugate of *r* is *mv*. As a consequence, computations are complicated and developing second order corrections to improve the accuracy of the transverse motion is hardly possible.

This problem can be resolved by using the Picht transformation:

$$R = r\sqrt{\beta\gamma},\ R' = \mathrm{dR}/\mathrm{dz}$$

where *R, R'* are the so called 'reduced coordinates'.

The advantages of the Picht transformation result from the fact that *R, R'* are canonically conjugate and that the profile of *R* with respect to *z* becomes much simpler than the one of the real coordinate *r* . This allows reducing the number of intervals. It is recommended to take for INTRVL a number within 8 to 12 (see also remark 6)

(4) By using the 'reduced coordinates' the overall equation in the transverse direction is given by:

$$\frac{d^2R}{\mathrm{dz}^2} - R\frac{q}{2m_0c^3}\frac{1}{\beta^3\gamma^3}\frac{\partial E_z}{\partial t} + R\left(\frac{q}{2m_0c^2}\right)^2\frac{\gamma^2+2}{\beta^4\gamma^4}E_z^2 = 0$$

For electrons where $m_0c^2 < 1$, the third term of this equation becomes preeminent and with IELEC = 0 this term is computed. With $m_0c^2 > 1$ this third term becomes negligible; it is ignored with IELEC ≠ 0, this permitting to reduce the computing time.

(5) Space charge computations can be made with respect to an arbitrary position in the cavity (see type code SCPOS). The default position is at the middle of the cavity.

(6) In the case of space charge calculations, INTRVL should be set to an **even value**. As an example, if INTRVL = 8 the space charge is automatically computed after the first interval and is extended to the first and the second interval. The second space charge computation is then made after the third interval and concerns the intervals 3 and 4. This process is automatically applied in the same way for the following intervals. If the field consists of multiple cells (see comment (2) in FIELD card), the number of intervals for each cell is INTRVL.

(7) The frequency used by CAVNUM is the frequency read by the FIELD card

EXAMPLE 1:

CAVNUM
12 (IDUM, dummy variable)
0. (DUMMY) -30. (DPHASE in deg) 0 (FFIELD) 8 (INTRVL) 1 (IELEC)

In this example, if the TOF is passive the RF phase is given by:

(RF Phase) = (crest phase) + (-30 deg)

When the TOF is activated and adjustments are required (see type code TOF), one will have:

(RF Phase) = (crest phase) + (-30 deg + adjustments) + TOF

When adjustments are not required and the TOF is activated:

(RF Phase) = (crest phase) + (-30 deg) + TOF

(8) Detailed cell by cell data are printed to file cavdat.out for each cavity. See Chapter 3, line item 15) for details on the data written to cavdat.out.

6.3.3 TYPE CODE: CAVSC

Describe the motion of particles in a single symmetrical accelerating element (e.g. accelerating gaps of a DTL).

NOTE: A code like SUPERFISH generates transit time factors (TTF) describing the axial field and the RF phase at the middle of the accelerating gap. From these, the DYNAC code computes new TTF (i.e. T, TP, TPP, S, SP, and SPP) and a new RF phase, corresponding to the entrance of the gap.

ENTRY:

1) ETCELL (I) (I = 1, 16)

ETCELL (1): cell or gap number (dummy variable, not used in DYNAC)

ETCELL (2): energy (MeV) (dummy variable, not used in DYNAC)

ETCELL (3): beta (dummy variable, not used in DYNAC)

ETCELL (4): cell length (cm)

ETCELL (5): T (TTF as in codes SUPERFISH or PARMILA)

ETCELL (6): TP (derivative of T, as in codes SUPERFISH or PARMILA)

ETCELL (7): S (dummy variable, not used in DYNAC)

ETCELL (8): SP (dummy variable, not used in DYNAC)

ETCELL (9): quad length (cm) (dummy variable, not used in DYNAC)

ETCELL (10): quad strength (kG/cm) (dummy variable, not used in DYNAC)

ETCELL (11): electric field (MV/m)

ETCELL (12): RF phase in the middle of the gap (deg)

ETCELL (13): accumulated length (cm) (dummy variable, not used in DYNAC)

ETCELL (14): TPP (second derivative of T)

ETCELL (15): frequency (MHz)

ETCELL (16): attenuation factor of the electric field

REMARKS:

(1) The reference particle and the COG of the bunch are allowed to be connected or disconnected (see type code REFCOG).

(2) Space charge computations can be made relative to any position in the accelerating gap (see type code SCPOS). The default position is at the middle of the accelerating gap (see type code SCDYNAC).

EXAMPLE: A 402.5 MHz DTL cell, operating at -45 deg RF phase:

```
CAVSC
 1  2.5253  0.07318  5.4326  0.5835  0.0961  0.5652  0.0259  3.5 0.  1.13  -45.  371.18257  0.0021465  402.5  1.
```

6.3.4 TYPE CODE: BUNCHER

Describe the motion of particles crossing a buncher (thin lens approximation).

ENTRY:

1) PV PDP PHARM PRLIM

PV: effective voltage (MV)

PDP: phase of RF (deg) (see remark (1))

PHARM: harmonic factor relative to the linac frequency (dimensionless)

PRLIM: radius of aperture (cm)

REMARKS:

(1) The phase of the RF at the entrance of the buncher can be the one of the reference particle or of the COG (see type code REFCOG).

(2) The RF phase can be connected (or not) with the time of flight. The code may automatically adjust the phase offset with respect to the TOF (see type code TOF).

(3) No space charge computation is possible in the buncher.

(4) PHARM is defined with respect to the most recently set frequency by either the GEBEAM, RDBEAM, or NEWF type codes.

EXAMPLE:

BUNCHER
0.0454 (PV in MV) -90. (PDP in deg) 1 (PHARM) 1.5 (PRLIM in cm)

6.3.5 TYPE CODE: MHB

Describe the motion of particles crossing a multi-harmonic buncher (thick lens approximation).

NOTE: The axial field of the cavity should be read from file (e.g. 'field.txt'; see type code FIELD).

ENTRY:

1) NF
2) AMP(1), AMP(2),….AMP(NF)
3) DPH(1), DPH(2),….DPH(NF)
4) FFIELD, INTRVL

NF: number of RF frequencies (NF < 6). The base frequency will be the one that was already defined prior to the MHB card, e.g. by GEBEAM or RDBEAM.

AMP: RF amplitude for each of the NF frequencies

DPH: phase **offset** w.r.t. the -90 deg zero crossing for each of the NF frequencies

FFIELD (in percent): relative level of the electric field:

(Effective electric field) = (electric field)*(1+FFIELD/100)

INTRVL (integer): number of integration intervals per cell in the numerical routine (see remark 2)
REMARKS:

(1) Space charge computation in the MHB should only be made with SCHEFF.
(2) In the case of space charge calculations, INTRVL should be set to an **even value**. As an example, if INTRVL = 8 the space charge is automatically computed after the first interval and is extended to the first and the second interval. The second space charge computation is then made after the third interval and concerns the intervals 3 and 4. This process is automatically applied in the same way for the following intervals. If the field consists of multiple cells (see comment (2) in FIELD card), the number of intervals for each cell is INTRVL.

EXAMPLE: A multi-harmonic buncher, operating at three frequencies:

```
MHB
3 (NF)
1.  (AMP(1) )      0.5  (AMP(2) )      0.33333  (AMP(3) )
0.  (DPH(1))    -180.  (DPH(2))       0.          (DPH(3))
0.  (FFIELD)        8   (INTRVL)
```

6.3.6 TYPE CODE: FIELD

The axial field of the cavity is read from file (e.g. 'field.txt') in the same format as the code SUPERFISH (see remark (1)), but should cover the full field (not the half field).

ENTRY:

1) filen

Name of an ASCII file containing one or several electromagnetic fields. Include the full path to this file to ensure that when DYNAC is run from DGUI, it can access the file. If in a series of FIELD cards the name is the same, DYNAC will continue to read from that same file. If the name has changed, the old file will be closed and the new one will be opened. The format of the file is described under remark (1).

2) ATT

ATT: factor, allow an adjustment of the level of the electric field:

(Effective field) = (field read) * ATT

REMARKS:

(1) The axial field is represented by a series of data of the form:

Z (m), E(Z) (V/m)

Note that it should cover the full field (not the half field). It is converted by the code DYNAC to the following units:

Z (cm), E (Z) (MV/cm)

The field file (e.g. 'field.txt') can incorporate different fields as follows:

First field:

FH (frequency in Hertz)

$z_1 \quad E(z_1)$

$z_2 \quad E(z_2)$

. . . .

$z_m \quad E(z_m)$

0. 0. ← End of the first field (mandatory)

Second field:

FH

$z_1 \quad E(z_1)$

$z_2 \quad E(z_2)$

. . . .

$z_m \quad E(z_m)$

0. 0. ← End of the second field (mandatory)

. . . .

(2) When each cavity necessitates a new field, the FIELD card must precede the card CAVMC or CAVNUM (or

alternatively, one can use a new filen each time).
EXAMPLE:

```
FIELD    ← first field in the field file (e.g. 'field.txt')
. . . .
CAVMC
. . . .
FIELD    ← second field
. . . .
CAVMC
. . . .
```

(3) When a number of cavities are using the same field, one will have:

```
FIELD    ← field
. . . .
CAVMC
. . . . .
CAVMC
. . . . .
CAVMC
. . . . .
```

(4) The frequency read from the file containing the field description will not change the master frequency set by the RDBEAM, GEBEAM or NEWF cards.

(5) DYNAC looks for zero crossings in the field data to determine how many cells there may be in a single field. So if in either tail of the field distribution there are small field amplitudes fluctuating around zero, DYNAC will (wrongly) assess these as cells. The recommendation in preparing the field data then is to essentially eliminate those fluctuations around zero in the parts of the field where they occur.

(6) The number of field data points in the file should not exceed 4000.

6.3.7 TYPE CODE: RWFIELD

Cause the field file (e.g. 'field.txt') to be rewound to the start.

NO ENTRY

6.3.8 TYPE CODE: HARM

The field of the cavity is given in the command list in the form of a Fourier series expansion.

ENTRY:

1) ZLG FH ATTE NCEL

ZLG: effective length of the field (cm)

FH: frequency (Hz)

ATTE: dimensionless factor, allows adjusting the field amplitude:

(Effective field) = (field) * ATTE

NCEL: number of cells in the cavity (e.g. for a multi-cell cavity).

2) NHARM

NHARM: number of the terms in the Fourier series expansion describing the field (see remark (1)).

3) A(I)   (I=1, NHARM)

A(I): Fourier coefficients (MV/cm) (see remark (1))
REMARKS:

(1) The field is in the form of a Fourier series expansion as:

$$E(z)=\sum_{j=0}^{\text{nharm}} a(j)\cos(\pi\,\mathrm{jz}/l)\text{ , } l=\text{length of the field}$$

The coefficients a(j), with j=1,NHARM (corresponding to *j=0* to *j=nharm*, where NHARM=*nharm*-1 and NHARM < 199) can be computed (from the shape of the axial field) by the code HARGEN (this code and the user guide can be obtained from the DYNAC authors). When the coefficients a(j) are provided by HARGEN, they are in units of (V/m). One can converted these to (MV/cm) by making use of the parameter ATTE (with ATTE = 1.0000E-08, see the example).

(2) When several cavities make use of an identical field, one has:

```
HARM   ← field for cavities 1 to N
......
CAVMC
1
......
CAVMC
N
......
HARM   ← new field
......
CAVMC
N+1
......
```

EXAMPLE: The coefficients A(I) are provided by the code HARGEN in units (V/m), the field is the one of a six-cell cavity.

```
 HARM
 100.15 (ZLG in cm)  0.805E09 (FH in Hz)  1.E-08 (ATTE)  6 (NCELL)
 36 (NHARM)
  0.19233E+06   -0.44891E+07    0.29043E+06 ← 36 Fourier coefficients
 -0.54317E+06    0.71625E+05    0.48210E+07
 -0.15519E+06    0.92716E+07   -0.41009E+06
 -0.13287E+08   -0.25629E+06    0.31508E+07
  0.84939E+05    0.12798E+07    0.20728E+06
  0.10077E+05    0.13440E+06   -0.23467E+06
 -0.17716E+05   -0.51720E+04   -0.99125E+05
  0.11994E+06   -0.70677E+05   -0.38300E+05
 -0.34711E+04   -0.43198E+06    0.40123E+05
  0.49988E+06    0.29710E+05   -0.90207E+05
 -0.17489E+04   -0.62260E+05   -0.17462E+05
 -0.42485E+04   -0.13850E+05    0.15663E+05
  CAVMC
  ......
```

6.3.9 TYPE CODE: EGUN

Motion of electrons through a DC electron gun; the axial electric field is read from file (e.g. 'egun_field.txt'; see remark (2)).

ENTRY:

1) filen

Name of the file containing the axial electric field in the form (z, E(z)), where z is in m and E(z) is in MV/m (see remark (2)). Include the full path to this file to ensure that when DYNAC is run from DGUI, it can access the file.

2) FMULT INDP

FMULT: dimensionless factor acting on the level of the axial electric field read:

(Effective field (MV/m)) = (field read (MV/m)) * FMULT

INDP: define the number of space charge computations required in the electron gun if the SCDYNAC card is preceding this one (i.e. space charge computations enabled), otherwise INDP is a dummy parameter.

INDP = 1: 8 space charge computations

INDP = 2: 16 space charge computations

INDP = 3: 32 space charge computations

CAUTION: At the entrance of the electron gun (i.e. at the cathode) the energy must be at least 20 eV (with an energy less than 20 eV problems may occur in the space charge computations).

REMARKS:

(1) Only the SCHEFF space charge routine should be used in conjunction with EGUN. One may change the space charge method after the DC gun (see type code SCDYNAC).

(2) The axial electric field is read from file (e.g. 'egun_field.txt') in the form (z, E(z)), where z is in m and E(z) is in MV/m:

```
N     (Number of points (Z, E(Z))
Z1      E(Z1)
. . . . . . . . .  .
ZN      E(ZN)
```

EXAMPLE:

8 different space charge computations are required in the DC gun:

```
SCDYNAC (with SCHEFF)
. . . . . . . . . .
EGUN
c:\mycodes\dynac\datafiles\egun_field.txt (file name of file containing the electric field)
19.16402 (ATTE)   1 (INDP)
```

Reference: E. Tanke and S.Valero: Motion of electrons in an electron gun (obtainable from the DYNAC website).

6.3.10 TYPE CODE: RFQPTQ

Describes the motion of protons (or ions) in an RFQ. Note that as such, it cannot generate the layout of the RFQ. The RFQ description to be used as input for DYNAC can be based on output files of codes like PARMTEQ (see remark (1)).

ENTRY

1) filen

Name of the file containing the cell by cell parameters of the RFQ (e.g. 'myrfq_cells.txt'); see remark (1). Include the full path to this file to ensure that when DYNAC is run from DGUI, it can access the file.

2) NCELTOT

NCELTOT: number of cells in the RFQ to be used in the simulation (this number can be chosen to be less than the effective total number of cells residing in the file 'filen'). The default value for the maximum number of cells in the RFQ is 500.

3) VS VB FPH WINDOW

VS: factor applied to the intervane voltage VV of the synchronous particle (%)
The actual voltage V0 acting on the synchronous particle will be given by:
V0 = VV (1 + VS/100)
The voltage V0 will be applied to the synchronous particle in all cells.

VB: factor applied to the intervane voltage VV for particles (%)
The actual voltage V1 acting on particles is set to:
V1 = VV (1 + VB/100)
V1 will be applied to the particles in all cells.

The synchronous particle and particles in the beam evolve separately if VS#VB (see remark (4)).

FPH: factor applied to the RF phase at the entrance of all accelerating cells (Type 0 and Type 2) in %
The actual RF phase 'PH1' with respect to the RF phase PH0 in 'filen' will be given by:
PH1 = PH0 (1 + FPH/100)
The factor FPH may be required to adjust the energy of the DYNAC synchronous particle to the one listed in the output file of PARMTEQ (see remark 5).
NOTE: FPH is not applied to cells of Type 1, 3, 4 and 5

WINDOW: parameter
The set of RFQ equations handle bunches having a half phase extension $\Delta\varphi$ such that $|\Delta\varphi| \leq \pi$ in the case of continuous beams the phase extension of the bunch must be compressed between $\pm\,\pi$ at the entrance of the RFQ.

Furthermore, by default, the transverse window in the RFQ used for defining the available aperture is formed by a rectangle centred on the beam axis, whereby the sides of the rectangle are defined by the available horizontal and vertical apertures between the electrodes. The sides of the rectangle change along with the modulation of the electrodes. For each step within each cell a particle is lost if abs(xm) > vanx or abs(ym) > vany, whereby xm and ym are the H and V positions of the particle and vanx and vany are the distances from the beam axis to the vane in H and V respectively. In locations of symmetry (horizontal aperture equals vertical aperture equals Asym), particles are by default eliminated if they have a radial position outside this aperture Asym.

With the WINDOW parameter, one can select approaches different from what is described above.
WINDOW > 0 causes the phase extension of the bunch to be compressed between $\pm\,\pi$ at the entrance of the RFQ
In this case, the RF phase of the synchronous particle will automatically be set to the value of the phase of the COG of the compressed beam.
WINDOW < 0 Same effect as with WINDOW > 0, but in addition, the aperture Asym gets multiplied by the absolute value of WINDOW. The idea here is to be less sensitive to beam loss, in particular in the case where the beam size is close to the available aperture. The rationale is that particles at these larger radii, rather than hitting an electrode surface, may travel between the electrodes and therefor would not necessarily be lost.
WINDOW = 0 causes disabling of this function.

REMARKS:
(1) The file ‘filen’ should contain one line per cell for each cell, containing the following 12 parameters:
NCELL ITYP VV(kV) CL(cm) A10 a(cm) m R0(cm) RHO(cm) PHASE(deg) FVOLT NCN
In the case of **ITYP=6**, only VV, PHASE and NCN are used (the others are dummy data). Furthermore, this line with these 12 parameters should be followed by a new line containing the name of the file, that contains the profile of the vanes (see also under ITYP=6 below). The full path to this file should be specified. The file containing the profile of the RMS should contain one set of coordinate pairs (z,R(z)) per line. Both z (longitudinal coordinate) and R (transverse radius) units are in meter. Example:

```
1  6  90.00 3.2211  0.00000  5.  1.  0.5223  0.5223  -90.  0.  0
c:\mycodes\dynac\datafiles\my_rfq_rms_profile.dat
```

Also in the case of **ITYP=7**, only VV, PHASE and NCN are used (the others are dummy data). Furthermore, this line with these 12 parameters should be followed by a new line containing the name of the file containing the profile of the vanes (see also under ITYP=7 below). Again, the full path to this file should be specified. The file containing the profile of the fringe field should contain one set of coordinate pairs (z,R(z)) per line. Both z (longitudinal coordinate) and R (transverse radius) units are in meter. The z coordinate can start from a non-zero value: in that case, DYNAC will subtract the first z coordinate value from all z coordinates. If the profile length for ITYP=7 exceeds that of ITYP=4, it should be truncated to that of ITYP=4. If the profile length for ITYP=7 is less than that of ITYP=4, a DRIFT should be added such that the total length of the ITYP=7 FF length and the DRIFT equals that of ITYP=4. Example of the line with the 12 parameters and following fringe field file name:

```
416  7  109.75  5.  0.  0.  1.  0.4877  0.3  .152.34  0.  404
c:\mycodes\dynac\datafiles\my_rfq_ff_profile.dat
```

The 12 parameters for each line are defined as follows:

- NCELL: cell number (the first number **must be** NCELL = 1)

- ITYP: Type of RFQ cell:
    ITYP = 0: standard accelerating cell
    ITYP = 1: transition cell of type T
    ITYP = 2: transition cell of type E
    ITYP = 3: transition cell of type M
    ITYP = 4: fringe-field region, type F (after type T, M or accelerating cell)
    ITYP = 5: RMS cell (single cell of type R)
    ITYP = 6: RMS cell (single cell of type R, based on vane profile)
    ITYP = 7: fringe-field region, type F (based on vane profile)

- VV: inter-vane voltage (kV); note that some RFQs have a ramped voltage.

- CL: cell length (cm)

- A10: main coefficient of the acceleration (no dimension); for some cell types, this will be dummy data

- a: smallest aperture of the vane (cm)

- m: modulation factor (no dimension)

- R0: mean aperture at the vane tip (at the middle of the cell)   (cm)

- RHO: transverse radius of curvature of the surface of electrodes at the vane tip (cm)

- PHASE: RF phase at the entrance of the cell (deg). Note that PARMTEQ uses the phase at the middle of the cell with sin(PHASE) whereas DYNAC uses the entrance of the cell with cos(PHASE).

- FVOLT: factor with respect to 1 applied to the inter-vane voltage only acting on particles (no action on the reference particle, see remark 4); one has:

(Actual voltage on particles) = (1 + FVOLT)*(voltage VV)

Note that the factor FVOLT must be set cell by cell by the user in the file ‘filen’

- NCN: Additional cell numbering, see remark (2)

The last line in the file **must** contain zeroes, i.e.:

0 0. 0. 0. 0. 0. 0. 0. 0

Apart from the parameters FVOLT and NCN, all these parameters are generated by the code PARMTEQ. Other parameters, like the coefficients A01, A03 and A12, are computed by the code DYNAC (see the references below).

Starting from the output file of PARMTEQ describing the machine, a specific program (PTQ2DYN), available on the DYNAC website, can aid in generating the file 'filen' for DYNAC based on data from PARMTEQ output. As there are different versions of PARMTEQ, one may need to adapt this code to the PARMTEQ version in use.
It is important to note here that PTQ2DYN expects a PARMTEQ file which, for the cells other than the standard accelerating cells, have a letter next to the cell number describing the cell type (T, F, M, R etc).
(2) Note that for simulating PARMTEQ cells, odd PARMTEQ cells have $A_{01}V$ positive and even cells have $A_{01}V$ negative (see references below for definition of coefficients A_{01}, V is the inter-vane voltage). This is handled through the NCN parameter:

For the RMS (cell type 5):	NCN = 0
For cell E (cell type 2):	NCN = 1
First standard cell (cell type 0)	NCN = 0
Second standard cell (cell type 1)	NCN = 1
For cell F (cell type 4) :	NCN as in the preceding transition cell

Setting the values for NCN can be automatically done by the program PTQ2DYN.

A positive value of $A_{01}V$ indicates that the horizontal vanes are closer to the axis at the beginning of the vane, a negative value of $A_{01}V$ indicates that the vertical vanes are closer.

Although PARMTEQ typically lists the Radial Matching Section (RMS) of the RFQ as consisting of a grand total of 4, 6 or 8 RMS cells (usually an even number of cells), the RMS is usually calculated as one complete section.
In DYNAC, the value of CL for the RMS should be the one corresponding to the total length of the RMS section. The value of a (cm) for the RMS should be the one at the entrance of the RMS, the value of RHO should be identical to the mean aperture at the vane tip R0 of the first cell following the RMS section and NCN should be set to 0. The RMS does not require R0; its value is ignored. Setting the values for the single RMS cell from the series of RMS cells can be automatically done by the program PTQ2DYN.

(3) When one has a sequence of RFQ cells, the phase shift between adjacent cells is 180 deg. It is then more practical to have the parameter PHASE indicating a phase shift at the entrance of the cell rather than an absolute phase.

(4) The reference particle and the particles in the beam may evolve separately. The reference particle energy is depending on the inter-vane voltage Vref. The energy of the other particles depends on the inter-vane voltage Vpart*(1+FVOLT), where FVOLT can be set cell by cell. Depending on the option chosen in the type code REFCOG, the time of flight of the particles can be relative to the synchronous particle or to the COG of the bunch.

(5) Note that for accelerating cells (ITYP=0), the RF phase parameters in the PARMTEQ output file are not entirely consistent with the energy gain of the synchronous particle in the same output file. This can cause a slight difference in energy of the synchronous particle between PARMTEQ and DYNAC. This slight difference can be totally corrected by using the following method. Note that the method does not act on cells of type 1, 3, 4 and 5 which will require another approach, see remark (9).

Looking at accelerating cells of type 0 and 2, one can show (provided that FPH is not too large, i.e. |FPH|<15 %) that the energy of the synchronous particle with respect to FPH is respecting a linear law such that:

$$W\ (FPH) = A.FPH + B$$

To calculate the coefficients A and B, one considers two values for FPH, for instance FPH = 0 and FPH = 6, and one computes with the code DYNAC the two energies W (FPH = 0) and W (FPH = 6) of the synchronous particle. Since, only the synchronous particle is concerned in these computations, a small number of particles are sufficient in the beam for this operation.

The two values W (FPH = 0) and W (FPH = 6) can be found in the DYNAC output file 'rfq_list.data', see remark (8). They will allow to obtain the two coefficients A and B from the above linear equation, from which it is a simple matter to adjust the factor FPH (and thus the phase RF at the entrance of the accelerating cells) such that the DYNAC synchronous particle will have the same energy as the one in the PARMTEQ output file.

(6) Of the three space charge methods available in DYNAC, only SCHEFF should be used for the RFQ, since it is the only one which can handle the continuous and mono-kinetic beam one generally has at the entrance of the RFQ. One may change the space charge method after the RFQ (see type code SCDYNAC). Space charge computations (if enabled) are automatically made with respect to the middle of each cell.

(7) The coefficients A10 acting on accelerating cells (standard cells of type 0) are the ones found in the PARMTEQ output file. Other coefficients, such as A01, A03 and A12 are computed by DYNAC. All coefficients for the other cell types (i.e. cells of type T, E, M and F) are computed by DYNAC. The dynamics is computed from a multi-polar expansion of the potential function and fields, see references.

(8) Three output files are provided by DYNAC, summarizing the results of the simulation (‘rfq_list.data’ and ‘rfq_coef.data’) and one listing the RFQ electrodes profile (‘rfq_profile.dat’).

(a) ‘rfq_list.data’ which is the principal output file lists for each cell on one line the 9 following numbers:

ncell Lcell(m) Ztot(m) Phi(deg) Phm(deg) Pho(deg) Wsyn Wcog ngood

with:

- ncell : cell number
- Lcell(m) : cell length
- Ztot(m) : accumulated length
- Phi(deg) : RF phase at the entrance of the cell
- Phm(deg) : RF phase at the middle of the cell
- Pho(deg) : RF phase at the exit of the cell
- Wsyn (MeV) : total energy of the reference particle
- Wcog (MeV) : total energy of the cog of the beam
- ngood : number of macro particles left

(b) ‘rfq_coef.data’ lists for each cell on one line the 6 following numbers:

ncell A01(m-2) A10 A12 r0(m) A03(m-6)

with:

- ncell : cell number
- A01 (m^{-2}) : main transverse coefficient
- A10 : main accelerating coefficient
- A12 : multi-polar coefficient
- r0 (m) : mean aperture radius of the electrodes
- A03 (m^{-6}) : multi-polar coefficient

(c) ‘rfq_profile.dat’ lists for each step in each cell on one line the 6 following numbers:

cell# type step z(m) rx(m) r0(m) ry(m)

with:

- cell# : cell number
- type : cell type (as ITYP above)
- step : step number in cell
- rx(m) : aperture radius of the horizontal electrodes
- r0 (m) : mean aperture radius of the electrodes
- ry(m) : aperture radius of vertical electrodes

Note: apart from the accelerating coefficients A10 of standard cells (type 0) which are the ones in the output files of PARMTEQ, all the coefficients are computed by DYNAC, also see remark (9).

(9) This remark relates to the transitions cells T, E, M and to the fringe-field F-cell.

-Cell T (of Type 1): This transition cell must follow an accelerating cell of Type 0 and should be followed by an M-cell or an F-cell.
The smallest aperture of the vane ‘a’ and the modulation factor ‘m’ in the output PARMTEQ file are the ones at the end of this cell (i.e. m = 1). In DYNAC these two parameters are required at the entrance of the T-cell, i.e. the same as the ones at the end of the previous accelerating cell. This correction is automatically made by the program PTQ2DYN.

The factor FPH, which is acting only on the RF phase of accelerating cells of Type 0 and of Type 2, does not operate on the RF phase of the T-cell. Therefore, looking at the RF phase in the PARMTEQ output file, a slight change of this RF phase could be needed in order to adjust the energy gain of the synchronous particle in the T-cell in DYNAC to the one in the output file of PARMTEQ. This RF phase adjustment could be easily made by the user, based on the technique in

remark 5 (once it's completed).
For this adjustment, one takes into account that with a relative reduction in the RF phase, the energy gain of the T-cell will decrease, and vice versa. The change of the energy gain is directly proportional to the change of the RF phase.

-CELL E (of Type 2): Entry transition cell before the first accelerating cell but after the Radial Matching Section.

-CELL M (of type 3): If used, this cell must follow a T-cell. As in this cell the modulation factor m = 1, the energy gain is zero. The RF phase of this cell must be the one at the output of the preceding T-cell (minus 180 deg).

-CELL F (fringe-field region of Type 4 or 7): Usually the fringe-field region is following a T-cell or an M-cell. However, sometimes it is placed after an accelerating cell of Type 0. In all cases, the RF phase must be the one at the exit of the previous cell and the sign of the previous inter-vane potential must be kept for the fringe-field region. This is automatically done by the code PTQ2DYN.

Special attention should be paid to the 'average radius R0' taken for the fringe-field region. Although PARMTEQ typically lists this average radius as the one of the previous cell, this radius is not compatible with the set of equations describing the dynamics in the fringe-field region (see references). This 'average radius' must be the one at the middle of the field region. It also can depend on the possible occurrence of a clamp-field at the end of the RFQ. Thus, it is recommended (if no information exist on the shape of the fringe field) to increase 2 or 3 times the average radius of the previous cell. Although this is done by the code PTQ2DYN (by default it increases the average radius by a factor 3), the user may need to further adjust this parameter.

EXAMPLE:

```
RFQPTQ
c:\mycodes\dynac\datafiles\myrfq_cells.txt ('filen' list)
219 (NCELL)
0. 0. 0. 1.
```

EXAMPLE: In order to simulate the transport of a beam which has an energy other than the design input beam energy of the RFQ, one can use a sequence like:

```
REFCOG
1
NREF
0.   DEW  0  1
RFQPTQ
[followed by usual entries]
REFCOG
0
```

Here DEW would need to be set to DEW=[design input beam energy of RFQ] – [energy of beam at the input]

(10) The value of Effective Voltage in the rfqparm.dmp section inside the dynac.dmp file corresponds to pi/4 * AV.

(11) In order to be able to print the particle distribution to file behind one, some or all cells in an RFQ, a special case of WRBEAM can be used. ***The WRBEAM is to be placed before the RFQPTQ card.*** See the WRBEAM card for more details.

References:

- Computer programs and methods for the design of high intensity RFQs, C.Biscari, CERN/PS 85-67 (LI)
- Dynamics through an RFQ cell in the code DYNAC, E.Tanke and S.Valero (obtainable from the DYNAC website).

6.3.11 TYPE CODE: STRIPPER, ***beta version***

Describe plural and multiple scattering of heavy ions in (solid) stripper foils. This model needs more work, hence this is marked as ***beta version***.

ENTRY

1) QS ATMS THS ANP

QS : atomic number of stripper ions

ATMS: atomic mass of stripper ions (AMU)

THS : thickness of the stripper (g/cm**2)

ANP : atomic number of the projectile

REMARKS:

(1) The computations are valid for slow heavy ions for which the α-Bohr parameter is $\alpha \geq 1$(see references 1,2,3).

(2) The average value of the scattering angle of incident particles is depending on the atomic number of the stripper ions and of the incident particles, on the thickness of the stripper and on the energy of incident particles (see references below). The scattering angles are distributed (separately) in angles X' and Y' of incident particles based on a hit-or-miss Monte Carlo method within a Gaussian distribution having the average value of the scattering angle as squared variance.

(3) The energy loss of the incident ions is depending on the atomic number and the atomic mass of stripper ions and on the energy of the incident ions and is based on Eastham (ref.1). For the carbon charge stripper case, this calculation is based on Bethe-Bloch's formula (ref. 4) and has been implemented as function dkeBethe in DYNAC by Ji-Ho Jang, RAON/IBS, Daejeon, South Korea.

(4) In the case of a carbon foil stripper (QS=6, ATMS=12), the charge state distribution after the stripper will be calculated following 3).

EXAMPLE:
Lead ions on a carbon foil of 0.1 mg/cm^2 :

```
STRIPPER
6 (QS)  12. (ATMS)  1.e-04 (THS in g/cm**2)  82. (ANP for lead ions)
```

References:
1) D.A. Eastham: Plural and multiple scattering of heavy ions in solids, DL/NSF/P11
2) Planned document: E.Tanke and S.Valero: Solid stripper foils in the code DYNAC (will be made available on the DYNAC website)
3) E.Baron et al, NIM A328 (1993) p.177-182
4) D. Groom: Energy loss in matter by heavy particles, PDG-93-06
5) J.F. Ziegler: The Stopping of Energetic light ions in elemental matter, J. Appl. Phys / Rev. Appl. Phys., 85, 1249-1279 (1999)

6.4 FUNCTIONING MODES

6.4.1 TYPE CODE: NREF

Define a new reference particle in phase space (W, PHASE) (needed when the accelerator makes use of a set of subsequent reference particles as in IH-structures).

ENTRY:

1) DEPHAS DEW IREF IREWF

DEPHAS: New phase relative to old one (deg)

DEW: New energy (MeV) relative to old one (see flag IREWF)

IREF=0: DEPHAS and DEW are relative to the synchronous particle

IREF=1: DEPHAS and DEW are relative to the COG

IREFW = 0: DEW is in % (in dW/W)

IREFW = 1: DEW is in MeV (in dW)

IREFW = 2: DEPHAS and DEW are new reference phase and energy in absolute units.

REMARKS:
(1) Depending on the option chosen in the REFCOG card, the new reference particle may be the synchronous particle or the COG (see type code REFCOG)

(2) When using IREFW=2, setting the phase in "absolute units" means that the phase of the reference particle is explicitly reset relative to the current TOF of the reference particle

EXAMPLE: the phase of the new reference particle is at (– 8 deg) relative to the old synchronous particle and its energy is at (-0.2496 MeV) from the one of the old synchronous particle:

```
NREF
-8. (DEPHAS in deg)  -0.2496 (DEW in MeV)  0 (IREF)  1 (IREFW)
```

6.4.2 TYPE CODE: TOF

Activate the time of flight in connection with the RF phase of RF elements (i.e. cavities, accelerating gaps and bunchers). This mode of operation is required for proper functioning when using MMODE to simulate RF phase and/or amplitude errors.

ENTRY:

1) INDIC ICOR

INDIC = 0: the TOF is active, otherwise it is passive (see remark (2) for detailed instructions on what else is required to activate the time of flight).

ICOR = 0: no adjustments on the phase of the RF elements.

ICOR = 1: adjustments on the phase of the RF elements with respect to the TOF (see remark (2)).

REMARKS:

(1) The TOF can be the one of the COG or of the reference particle (see type code REFCOG).

When the TOF is passive (INDIC = 1 or no TOF card exists in the command list) the RF phase at the entrance of RF elements is given by (see type code CAVMC for the definition of the crest phase and the phase offset):

(RF Phase) = (crest phase) + (phase offset)

Phase offset corresponds to the DPHASE card in the CAVMC and CAVNUM type codes. In this case, ICOR is a dummy parameter.
If the TOF is active (INDIC = 0) without adjustments (ICOR = 0), the RF phase is given by:
(RF Phase) = (crest phase) + (phase offset) + TOF

(2) In order to change an input file from one that does not have TOF enabled to one that works with time of flight, one introduces the TOF type code with INDIC=0 and ICOR=1 in the input file (e.g. after the location where the input beam and reference have been defined). If ICOR = 1, adjustments are automatically made on the phase offset of RF elements in such a way that the TOF is compensated. In this case, one has:

(RF Phase) = (crest phase) + (phase offset + adjustment) + TOF

This new phase, which includes the adjustment (i.e. phase offset + adjustment), is printed (with respect to 2kπ) in files 'dynac.short' and ‘dynac.long’. The next and final step is to replace the DPHASE settings in the input file with the new phases printed in dynac.short and to set ICOR to zero (ICOR=0). Now the input file will work with time of flight and so will work as intended when the MMODE card is used.
This adjustment has not been implemented in CAVSC.

(3) The card TOF can be introduced in any position in the command list. It affects the accelerating elements following the type code.

(4) The MAXMIN card used in older versions of DYNAC is now obsolete and has been deleted.

EXAMPLES:
(1) The TOF is active and no adjustment on the phase offset is required:

```
TOF
0 (INDIC)  0 (ICOR)
```

(2) The TOF is active and adjustment on the phase offset is needed:

```
TOF
0 (INDIC)  1 (ICOR)
```

(3) The T.O.F. becomes passive:

```
TOF
1 (INDIC)  2 (ICOR, which in this case is a dummy variable)
```

6.4.3 TYPE CODE: NEWF

Define a new frequency (needed when using different RF frequencies for different parts of the machine).

ENTRY:

1) FH

FH: New RF frequency (Hz)

6.4.4 TYPE CODE: REFCOG

Either detach or link the COG and the synchronous particle in accelerating elements (accelerating gaps, cavities and bunchers)

ENTRY:

1) ISHIFT

ISHIFT = 0: the reference particle is the COG of the bunch

ISHIFT = 1: the reference particle and the COG progress independently

ISHIFT = 2: at the start after REFCOG card, COG and synchronous particle are coinciding, afterwards they evolve independently

REMARK: The REFCOG card can be introduced in any position in the command list. It is effective on the following accelerating elements. By default (i.e., if no REFCOG type code is used), ISHIFT=0.

6.5 REDEFINING THE BEAM

6.5.1 TYPE CODE: TILT

Define rotation and shift of the beam ellipsoid with respect to the COG.

ENTRY:

1) ICG

ICG = 1: the new reference particle is set to the current COG of the bunch.

ICG = 0: the reference particle remains unchanged.

2) TIPHA TIX TIY SHIFW SHIFP

TIPHA: Shift on the phase axis (deg)

TIX: Shift on the horizontal axis (cm)

TIY: Shift on the vertical axis (cm)

SHIFW: change the position in energy of the COG (MeV)

SHIFP: change the position in phase of the COG (deg)

REMARKS:

(1) The beam ellipsoid generated by the GEBEAM card (with ITWISS = 0) is upright (see type code GEBEAM). In this case, if a rotated (non-erect) or shifted initial beam ellipsoid is needed, the card TILT should directly go after the card INPUT:

```
GEBEAM
. . . . .
INPUT
. . . . .
TILT
. . . . .
```

(2) The card TILT can be introduced in any place in the input file. It will affect the beam ellipsoid for all following elements.

EXAMPLE: one changes the position of the COG of the beam ellipsoid in energy and in phase, the synchronous particle remains unchanged:

```
TILT
0 (ICG)
0.(TIPHA in deg) 0.(TIX in cm) 0.(TIY in cm) 1.5(SHIFW in MeV) 12.5 (SHIFP in deg)
```

6.5.2 TYPE CODE: TILZ

Rotate the beam around COG in the plane (X, Z) after an upright ellipse has been generated by GEBEAM card (with ITWISS = 0, see type code GEBEAM).

ENTRY

1) TILTA

TILTA: slope (deg)

REMARK: The card TILZ should directly follow the INPUT card:

```
GEBEAM
. . . . .
INPUT
. . . . .
TILZ
. . . . .
```

6.5.3 TYPE CODE: REJECT

Cause the identification, counting and elimination of particles that reach the longitudinal and transverse limits as defined in this entry.

ENTRY:

1) IFW WDISP WPHAS WX WY RLIM

IFW = 0: the limit of WDISP in one half energy spread dW/W, where this is w.r.t. the COG

IFW = 1: the limit of WDISP in one half energy spread dW (MeV), where this is w.r.t. the COG

IFW = 10: the limit of WDISP in one half energy spread dW/W, where this is w.r.t. the reference

IFW = 11: the limit of WDISP in one half energy spread dW (MeV), where this is w.r.t. the reference

WPHAS: limit in one half phase extent (deg)

WX: limit in one half horizontal beam extent (cm)

WY: Limit in one half vertical beam extent (cm)

RLIM: Limit in one half transverse aperture (cm)

REMARKS:

(1) DGUI now also has an option to plot the available aperture as defined by REJECT. If no aperture was defined in the DYNAC input file (i.e. no REJECT card present), a default value of 100 cm will be taken. It is recommended to place a first REJECT card either directly following the RDBEAM card and its entries, or following the GEBEAM card and its entries. Alternatively, one can place a first REJECT card before the RDBEAM or GEBEAM card.

(2) The limits (in (+ -)) are relative to the COG

(3) If no REJECT card is used, the default values are:

IFW = 1 WDISP = 1000 MeV, WPHAS = 4000 deg, WX = 100 cm, WY = 100 cm,
RLIM = 400 cm

(4) With REJECT one only ***sets*** the limits; the beam will need to go through a beam line element in order for particles to be eliminated when outside these limits.

(5) The position where the particles are lost, as well as their coordinates and the reason for loss are printed in the lost_particles.data file. The reason for loss is given in the following way:
Loss reason = -2 --> energy
Loss reason = -1 --> phase
Loss reason = 1 --> radius
Loss reason = 2 --> x position
Loss reason = 3 --> y position

EXAMPLE:

REJECT
1 (IFW) 1.5 (WDISP in MeV) 60. (WPHAS in deg) 2.5 (WX in cm) 2.5 (WY in cm) 3. (RLIM in cm)

6.5.4 TYPE CODE: CHASE

Cause the identification and counting of particles far from the COG of the bunch in the three phase planes, starting with the particles farthest away from to the COG. The remaining bunch core is then analyzed in terms of beam and emittance size. The reduced emittance one obtains is based on the beam fractions set with the CHASE card for the horizontal plane, the vertical plane and the longitudinal plane respectively. Data are written to the files 'chase_emith.out', 'chase_emitv.out' and 'chase_emitl.out' as a function of position. Additional data points can be obtained at any location where an EMIT or EMITL card has been inserted in the input file.

ENTRY:

1) FRACTX FRACTY FRACTL

FRACTX: Fraction of particles to be retained in the (X, X') plane

FRACTY: Fraction of particles to be retained in the (Y, Y') plane

FRACTL: Fraction of particles to be retained in the (Energy, Phase) plane

EXAMPLE:

If one wishes to have:
0 % of the particles to be rejected in the (X, X') plane
0% of the particles to be rejected in the (Y, Y') plane
15% of the particles to be rejected in the (Energy, Phase) plane
one would enter:

```
CHASE
1. (FRACTX) 1. (FRACTY) 0.85 (FRACTL)
```

REMARKS:

(1) The routine CHASE is computing time intensive

(2) The particles rejected by CHASE are not eliminated from the beam in the sense of the REJECT card; they are only temporarily eliminated in order to calculate relevant parameters on the beam core.

(3) The action of the CHASE card can be disabled by setting the fractions to 1 in another CHASE card:

```
CHASE
1.  1. 1.
```

(4) The 'chase_emith.out' file shows data after eliminating particles in the horizontal plane. The emittances shown for the other two planes take these rejected particles into account. The same approach holds for the other two files.

(5) When a CHASE card is inserted before the WRBEAM card in the input file, the coordinates of the particles removed by CHASE are printed in the file ‘beam_remove_H.dst’ as well as the ones kept by CHASE in the file 'beam_core_H.dst' if FRACTX<1. Likewise for ‘beam_remove_V.dst’, ‘beam_core_V.dst’ (FRACTY<1) and ‘beam_remove_L.dst’ , ‘beam_core_L.dst’ (FRACTY<1).

6.5.5 TYPE CODE: COMPRES

Cause the phase extension of the particles (with respect to COG) to be compressed within the specified window. The COMPRES card can for instance be used at the entrance of the RFQ (see type code RFQPTQ) when the phase extent is larger than 2*pi. Particles outside the specified boundary will be shifted within.

ENTRY:

1) WINDOW

WINDOW: Represents the total size of the window (in degrees)

REMARKS:
(1) The RFQPTQ card also has the COMPRES functionality built in (see WINDOW entry under RFQPTQ).

6.5.6 TYPE CODE: DCBEAM

Can be used to define if the beam is to be considered bunched or not; this is in view of space charge calculations. In the case of calculations with space charge, this card causes the beam to remain within 360 deg total length. This card is not needed for calculations without space charge.

ENTRY:

1) IFCONT

IFCONT = 0 : Select this value for bunched beams

IFCONT = 1 : Select this value for DC beams

REMARKS:

(1) The default value for IFCONT is 0 (bunched beam).
(2) If GEBEAM is used with LAW=5 or LAW=6, IFCONT is automatically set to 1.
(3) If RFQPTQ is used, IFCONT is automatically set to 0 at the end of the RFQ
(4) If BUNCHER is used, IFCONT is automatically set to 0 at the end of the BUNCHER

6.6 MAGNETS OR CAVITIES TOLERANCES AND ERRORS

6.6.1 TYPE CODE: ALINER

Alignment errors in X, Y, X', Y'

ENTRY:

1) XL YL XPL YPL

XL: displacement in X (cm)

YL: displacement in Y (cm)

XPL: rotation on X' (mrad)

YPL: rotation on Y' (mrad)

EXAMPLE:

```
ALINER
-0.1 (XL in cm) 0.2 (YL in cm) 0.1 (XPL in mrad) -0.2 (YPL in mrd)
QUADRUPO
. . . . . .
```

6.6.2 TYPE CODE: CHANGREF

Change of reference frame

ENTRY:

1) XC YC A

XC: displacement in X-direction (cm)

YC: displacement in Y-direction (cm)

A: rotation relative to the positive beam direction (deg)

REMARK: A positive sign of rotation signifies clockwise rotation about the positive direction of beam travel.

6.6.3 TYPE CODE: ZROT

Turn the plane (X, Y) around the direction of beam travel. The ZROT card may be used to specify a rotated magnet (see examples).

ENTRY:

1) ZROTA

ZROTA: angular rotation (deg) (-180 deg ≤ ZROTA ≤ 180 deg)

REMARK: A positive sign of angular rotation signifies clockwise rotation about the positive direction of beam travel. Note that in effect the particle coordinates are changed, not the element itself.

EXAMPLES:

(1) For a bending magnet, a bend down is represented by rotating the X, Y coordinates by (+ 90 deg) as follows:

```
ZROT
90. (ZROTA of +90. deg)
BMAGNET   ← the bending magnet is now down by 90 deg
. . . . .
ZROT      ← return coordinates to the initial orientation
-90. (ZROTA of -90. deg)
```

(2) A bend up is accomplished via -90 deg rotation as follows:

```
ZROT
-90.
BMAGNET   ← the bending magnet is now up by 90 deg
. . . . .
ZROT      ← return coordinates to the initial orientation
90.
```

(3) A bend to the left (looking in the positive direction of beam travel) is accomplished by rotating the x, y coordinates by + 180 deg:

```
ZROT
180.
BMAGNET
. . . . .
ZROT      ← return coordinates to the initial orientation
-180.
```

6.6.4 TYPE CODE: RANDALI

Generate misalignments in X, X', Y and Y' using a hit-or-miss Monte Carlo method.

ENTRY

1) ILIER

ILIER = 0: disable misalignments; in this case, no other entry is required.

ILIER = 1: enable misalignments; in this the next entry has to be:

2) XL YL XPL YPL

XL: displacement in X (cm)

YL: displacement in Y (cm)

XPL: rotation on X’ (mrad)

YPL: rotation on Y’ (mrad)

REMARKS:

(1) The errors are randomly distributed between (-X, X), (-Y, Y), (-X’, X’) and (Y, Y’). They will affect all the elements following the command card RANDALI.

(2) The RANDALI card can be introduced in any position in the command list.
EXAMPLE: Misalignments for two quadrupoles only

```
RANDALI
1 (ILIER)
0.005 (XL in cm) 0.005 (YL in cm) 0.1 (XPL in mrad) 0.1 (YPL in mrad)
QUADRUPO
. . . . . . .
DRIFT
. . . . . . .
QUADRUPO
. . . . . . .
RANDALI    ← disable the misalignments
0 (ILIER)
. . . . . . .
QUADRUPO
 . . . . . .
```

6.6.5 TYPE CODE: TWQA

Rotate quadrupoles about the beam direction.

ENTRY:

1) IRAND QTWIST

IRAND = 0: give indication of systematic rotation

IRAND = 1: rotations are generated from a hit-or-miss Monte Carlo method

QTWIST: angle of rotation (deg) (see remarks)

REMARKS:
(1) If IRAND = 0, a positive sign of QTWIST causes a clockwise rotation with respect to the positive beam travel direction.

(2) If IRAND = 1, rotations are randomly generated between (-QTWIST, QTWIST)

(3) If QTWIST = 0., the rotation is disabled for all following quadrupoles.

(4) TWQA card can be introduced in any position in the command list. It will affect the quadrupoles until QTWIST = 0.

(5) The ERPA card, used in older versions of DYNAC is now obsolete and has been deleted.

EXAMPLE: Rotation for two quadrupoles only

```
TWQA
0 (IRAND) 1.3 (QTWIST)
QUADRUPO
. . . . . . .
DRIFT
. . . . . . .
QUADRUPO
. . . . . . .
RANDALI     ← disable rotations
0 (IRAND) 0. (QTWIST)
. . . . . . .
QUADRUPO
 . . . . . .
```

6.6.6 TYPE CODE: MMODE

Introduce systematic or random errors on RF phases and amplitudes in accelerating elements (i.e. cavities, accelerating gaps and buncher; see remark (3)).

ENTRY:

1) IERPF VPHASE VFIELD

IERPF = 0: disable systematic and random errors
IERPF = 1: systematic error on RF phases and amplitudes

IERPF = 2: random errors on RF phases and amplitudes

VPHASE: phase change to be added to the nominal RF phase (deg)
VFIELD: change to the RF amplitude (in %):

(Level of field) = (initial field level) * (1+ VFIELD/100)

REMARKS:

(1) The card MMODE can be put in any position in the command list. It will affect the following accelerating elements.

(2) MMODE is only acting on particles (no change to the reference particle).

(3) The random errors are distributed using a hit-or-miss Monte Carlo method between (- VPHASE, VPHASE) and (- VFIELD, VFIELD).

(4) For a beamline with multiple cavities, changing the amplitude or phase in the first cavity (as would happen if the MMODE type code is used there) will cause a different time of arrival (time of flight) of the beam in the second cavity, including w.r.t. the RF phase of that cavity. To study such effects with DYNAC requires the use of the TOF type code. With the TOF type code, the time of flight of the particles will be taken in to account.

(5) The BFLDLVL card used in older versions of DYNAC is now obsolete and has been deleted

EXAMPLE: A systematic change of 2 deg to the nominal RF phase and of 1 % to the cavity field level is required to one cavity. These changes are to be disabled for subsequent cavities.

```
MMODE
1 (IERPF)  2. (VPHASE in deg)  1. (VFIELD in %)
CAVMC
. . . . . .
MMODE
0 (IERPF)  1. (VPHASE, dummy variable) 2. (VFIELD, dummy variable)
```

6.7 SPACE CHARGE COMPUTATION

Three different space charge routines are incorporated in DYNAC:

1) HERSC: the beam self-field equations are reduced to the 3-D Poisson's equation within boundary conditions (see ref.1 and 2 below). In conjunction with EGUN or with RFQPTQ, SCHEFF should be used instead.

2) SCHERM: the bunch is constituted of several ellipsoids in the longitudinal z-direction, whereas in the transverse directions it keeps a simpler symmetrical shape respecting RMS sizes (see ref.3 below). In conjunction with EGUN or with RFQPTQ, SCHEFF should be used instead.

3) SCHEFF: a modified version of the routine developed at LANL. This modified version includes relativistic corrections and can deal with a multi-charge state beam (even when the charge states in the beam are separated, such as in a bending magnet). The distribution is developed in rings of elementary charges (see ref.3 and 4 below) and each ring is assigned a potential.

NOTE: HERSC and SCHERM cannot and should not be used with a multi-charge state beam.

REFERENCES:

[1] P. Lapostolle and al., "HERSC: A New 3 D Space Charge Routine for High Intensity Bunched Beams", linac2002, Gyeongju, South Korea.
[2] P. Lapostolle, E.Tanke and S. Valero: A 3-d space charge routine, HERSC, based on a sequence of 3-d Hermite orthogonal functions (obtainable in the DYNAC website).
[3] P.Lapostolle and al. "A Modified Space Charge Routine for High Intensity Bunched Beams", NIM A 379 (1996) pp. 21-40.
[4] F.Guy, Los Alamos Group AT-1 Memorandum AT-1:85-90, March 6, 1985

6.7.1 TYPE CODE: SCDYNAC

Select the space charge method and its parameters.

ENTRY:

1) ISCSP

ISCSP allows selecting the space charge method

ISCSP = 1 or ISCSP = -1: HERSC (only to be used with bunched non-relativistic beams)

ISCSP = 2: SCHERM (only to be used with bunched non-relativistic beams)

ISCSP = 3: SCHEFF (can be used both with relativistic and non-relativistic beams, as well as with DC beams)

2) BEAMC SCE10

BEAMC: electrical beam current (mA)

SCE10 is a flag permitting to select elements in which space charge computations are made (see remark (3)). This can be of particular interest for the investigation of DTL type structures (i.e. Alvarez structures)

SCE10=1: Space charge calculated for all relevant elements, but not at drifts.

SCE10=2: Space charge calculated for accelerating elements only

SCE10=3: Space charge calculated for all relevant elements

The following lines depend on the space charge method selected:

A) Space charge routine HERSC (with ISCSP = 1 or ISCSP = -1):

- If ISCSP = 1, the following entry will be read:
  3) RDCF

RDCF is a fraction less or equal to one. It permits, when the number N of particles is large (i.e. N > 15000), the selection of a reduced number N* = RDCF*N of particles for the computation of Hermite coefficients A_{lmn} (this allows saving computing time; see remark (1)).

If RDCF = 0, the code automatically defines this fraction.

No supplementary entry is required with ISCSP = 1, the default parameters in HERSC routine are in use (see below for the default parameters).

EXAMPLE:

```
SCDYNAC
1 (ISCSP)
100. (BEAMC in mA) 3 (SCE10)
0. (RDCF)
```

- If ISCSP = -1, the following entries 3), 4), 5) and 6) will be read:

3) LMAXI MMAXI NMAXI

LMAXI, MMAXI and NMAXI are the upper limits of the Hermite series expansion representing the distribution of particles (see notes and remark (1)).

The default parameters in the code are: LMAXI = MMAXI = NMAXI = 11

4) FXRMS FYRMS FZRMS

Some few statistically isolated and very distant particles could affect the accuracy of the Hermite coefficients A_{lmn}. It is then recommended to remove these few "misplaced" particles. Therefore, HERMITE coefficients are computed from the particles included in a cube defined in RMS multiples (FXRMS, FYRMS and FZRMS) as follows in the x, y and z-directions:

Size of the cube in x-direction: (+ -)RMS(x) * FXRMS

Size of the cube in y-direction: (+ -)RMS(y) * FYRMS

Size of the cube in z-direction: (+ -)RMS(z) * FZRMS

where RMS(x), RMS(y) and RMS(z) are the horizontal, vertical and longitudinal RMS half beam sizes respectively.

The default parameters in the code are: FXRMS = FYRMS = FZRMS = 2.5

5) EPS

Among the terms $A_{lmn}\,\delta_{lmn}$ emerging in the Hermite series expansion from the upper limits LMAXI, MMAXI and NMAXI, only a few tens of these are significant, the other terms can be removed. The parameter EPS allows removing these non-significant terms.

The default parameter in the code is: EPS = 8.E-03

6) RDCF

Optional selection of a reduced number of particles as described above.

EXAMPLE:

```
SCDYNAC
-1 (ISCSP)
150. (BEAMC in mA) 3 (SCE10)
13 (LMAXI) 13 (MMAXI) 15 (NMAXI)
2.5 (FXRMS) 2.5 (FYRMS) 2.5 (FZRMS)
8.E-03 (EPS)
0. (RDCF)
```

NOTES:

(1) The default parameters for HERSC (with ISCSP = 1) have been found appropriate for the majority of types of particle distributions encountered in cases of real machines. Therefore, apart from very specific and complex particle distributions it is recommended to use the option ISCSP = 1.

(2) The routine HERSC is computing time intensive (at least, compared to SCHEFF).

(3) The upper limits for LMAXI, MMAXI and NMAXI are 21.

B) Space charge routine SCHERM (with ISCSP = 2, see ref.3)

3) IDUMMY (dummy entry)

EXAMPLE:

```
SCDYNAC
2 (ISCSP)
120. (BEAMC in mA) 3 (SCE10)
0. (IDUMMY, i.e. dummy variable, not used in the code)
```

C) Space charge routine SCHEFF (with ISCSP=3, see ref.3 and 4)

3) IREAD

- If IREAD = 0, the default parameters in the code are in use. In this case, no other parameters are required.

EXAMPLE:

```
SCDYNAC
3 (ISCSP)
130. (BEAMC in mA) 3 (SCE10)
0. (IREAD)
```

- If IREAD = 1, the following entry will be read:

4) SCE(2) SCE(3) SCE(4) SCE(5) SCE(6) SCE(7) SCE(9)

SCE(2): radial mesh interval in RMS multiples

SCE(3): longitudinal half mesh interval in RMS multiples

SCE(4): no. of radial mesh intervals (maximum 20)

SCE(5): no. of longitudinal mesh intervals (maximum 40)

SCE(6): no. of adjacent bunches, applicable for buncher studies, should be 0 for DTL
SCE(7): pulse length; if not beta*lambda (transport studies) distance between beam pulses. If set to zero, one will get the default "beta*lambda" (or "beta*lambda/2" for an RFQ); units are cm.
SCE(9): option to integrate space charge forces over boxes. If SCE (9) = 0, no integrations are made.

NOTE: the default parameters for SCHEFF are:
SCE (2) = 4 (in RMS multiples), SCE (3) = 4 (in RMS multiples), SCE (4) = 20, SCE (5) = 40, SCE (6) = 0, SCE (7) =0, SCE (9) = 1.
Although these default parameters have been found appropriate for various types of finite particle distributions encountered in practical cases of real machines, care needs to be taken to assure that these default parameters are optimal for the particular application in question.

In the case of a DC (continuous) beam (i.e. not bunched), the number of adjacent “bunches” SCE(6) needs to be set to larger value than the default one (a value in the range 20 to 40 is suggested) in order to further define the longitudinal size of the mesh for this case.

EXAMPLE (continuous beam, 40 adjacent bunches):

SCDYNAC
3 (ISCSP)
120. (BEAMC in mA) 3 (SCE10)
1. (IREAD)
4 (SCE(2)) 4 (SCE(3)) 20 (SCE(4)) 40 (SCE(5)) 40 (SCE(6)) 0 (SCE(7)) 1 (SCE(9))

REMARK 1: In the HERSC method (ref.1 and 2 in this section), one represents the density of electric charge $\rho(x,y,z)$ by a series expansion of the Hermite functions $\hbar_l(x), \hbar_m(y), \hbar_n(z)$ in the form:

$$\rho(x,y,z) = \sum_{l=0}^{l'} \sum_{m=0}^{m'} \sum_{n=0}^{n'} A_{\mathrm{lmn}} \delta_{\mathrm{lmn}}(x,y,z)$$

with:

$$\delta_{\mathrm{lmn}}(x,y,z) = \hbar_l(x)\hbar_m(y)\hbar_n(z)$$

The Hermite coefficients A_{lmn} are given by:

$$A_{\mathrm{lmn}} \approx \frac{q}{\varepsilon_0 \|\delta_{\mathrm{lmn}}\|} \sum_{i=1}^{N} H_l(x) H_m(y) H_n(z)$$

where$H_l(x)$, $H_m(y)$ and $H_n(z)$ represent the Hermite polynomial of degrees l, m, n respectively and $\|\delta_{\mathrm{lmn}}\| = (2\pi)^{3/2} l!m!n!$.

The set of beam self-fields E_x, E_y, E_z is solution of the Poisson's equation:

$$\nabla U \simeq -\sum_{l=0}^{l^{\square}} \sum_{m=0}^{m^{\square}} \sum_{n=0}^{n^{\square}} A_{\mathrm{lmn}} \hbar_l(x)\hbar_m(y)\hbar_n(z)$$

The method HERSC is not directly applicable when discontinuities appear in the distribution (e.g. the distribution of a homogeneously charged sphere). Despite the fact that such discontinuities are unusual in real situations, a process solving this problem has been introduced in the routine HERSC (ref.1 and 2 in this section). Discontinuities in distributions are avoided by using LAW= 2, 3 or 4 in the type code GEBEAM.

REMARK 2: The SCDYNAC card can be introduced anywhere in the command list (this allows changing the space charge method and/or its parameters).

REMARK 3: The default position of space charge computations is at the middle of the elements. It can be changed in the cavities (only for CAVMC, see type code SCPOS).

6.7.2 TYPE CODE: SCPOS

Define an arbitrary position for space charge computations in cavities.

ENTRY:

1) XPSC

XPSC: fraction (less or equal to one); the position of space charge computation is defined by:

Position = [(XPSC) * (length of the cavity)]

EXAMPLE: The position of space charge computation is at a quarter from the front of the cavity:

SCPOS
0.25 (XPSC)
CAVMC
. . . . .

REMARK: The default position is at the middle of the accelerating element.

6.8 OUTPUT PRINT

6.8.1 TYPE CODE: WRBEAM

Print particle coordinates to a file (e.g. beam.dst).

ENTRY:

1) filename *LOCINDIC*
   where *LOCINDIC* is an optional entry, which can be used together with the RFQPTQ type code (see below)
2a) for IFLAG greater or equal zero:
   IREC IFLAG
2b) for IFLAG less than zero and NRFQC greater than zero:
   IREC IFLAG NRFQC
2c) for IFLAG less than zero and NRFQC less than zero:
   IREC IFLAG NRFQC NC1 NC2 …..NCi

IREC = 0: the coordinates of the particles are printed relative to the COG

IREC = 1: the absolute energy of particles is printed, the phase is relative to the COG

IREC = 2: the coordinates of the particles are printed relative to the reference

IREC = 3: the absolute energy and phase of the particles is printed

IFLAG=100: File using 6 coordinates for each macro-particle (see remark 3), with phase in radians

IFLAG=101: Like IFLAG=100, but also includes particle number (see remark 3), with phase in radians

IFLAG=102: Like IFLAG=100, but also includes charge state(s) (see remark 3), with phase in radians

IFLAG=103: Like IFLAG=102, but also includes particle number (see remark 3), with phase in radians

IFLAG=110: File using 6 coordinates for each macro-particle (see remark 3), with phase in ns

IFLAG=111: Like IFLAG=110, but also includes particle number (see remark 3), with phase in ns

IFLAG=112: Like IFLAG=110, but also includes charge state(s) (see remark 3), with phase in ns

IFLAG=113: Like IFLAG=112, but also includes particle number (see remark 3), with phase in ns

The energy of the reference and of the COG (MeV) will be printed as last value on the first line. In legacy input files, the value of IFLAG was 100 smaller. Using abs(IFLAG) > 99 allows for better integration with the DYNAC Graphical User Interface (DGUI), which is why the use of abs(IFLAG) < 100 is discouraged.

In order to be able to print the particle distribution to file behind one, some or all cells in an RFQ, a special case of WRBEAM can be used. ***For this to be effective, the WRBEAM is to be placed before the RFQPTQ card.*** For this purpose, IFLAG will need to be negative, but otherwise has the same meaning as listed above.

If one wishes to print the particle distribution at the end of each cell, NRFQC will need to be an integer larger than the total number of RFQ cells. An example for this case: an RFQ with 100 cells for which one wishes to have the distributions printed to file myfile.dst at the *end of each cell* could look like this:
WRBEAM
myfile.dst
0 (IREC) -100 (IFLAG) 111 (NRFQC)
Note that the optional *LOCINDIC* entry is omitted here.
Should one wish to print the particle distribution at the *middle of each cell*, the *LOCINDIC* entry would need to be set to **rfqmidcell** , so for example:
WRBEAM
myfile.dst rfqmidcell
0 (IREC) -100 (IFLAG) 111 (NRFQC)

Should one wish to print the particle distribution at the *middle and the end of each cell*, the *LOCINDIC* entry would need to be set to **rfqmaecell** , so for example:
WRBEAM
myfile.dst rfqmaecell
0 (IREC) -100 (IFLAG) 111 (NRFQC)

If one wishes to print the particle distribution behind one particular cell, NRFQC will need to be an integer equal to that RFQ cell number. An example for this case: an RFQ with 100 cells for which one wishes to have the distributions printed to file myfile.dst behind cell 77 could look like this:
WRBEAM
myfile.dst
0 (IREC) -100 (IFLAG) 77 (NRFQC)

If one wishes to print the particle distribution behind certain cells, NRFQC will need to be an integer equal to the number of RFQ cells for which one would like to print out the distribution, but with a sign change. One can print distributions corresponding to up to 20 cells this way. NC1, NC2 etc. are then the cell numbers for which one wishes to print the distribution. An example for this case: an RFQ for which one wishes to have the distributions printed to file myfile.dst for cells 33, 55 and 98 could look like this:
WRBEAM
myfile.dst
0 (IREC) -100 (IFLAG) -3 (NRFQC) 33 (NC1) 55 (NC2) 98 (NC3)

REMARKS:

(1) When a CHASE card is inserted before the WRBEAM card in the input file, the coordinates of the particles removed by CHASE are printed in the file ‘beam_remove_H.dst’ as well as the ones kept by CHASE in the file 'beam_core_H.dst' if following the CHASE card FRACTX<1. Likewise for ‘beam_remove_V.dst’, ‘beam_core_V.dst’ (FRACTY<1) and ‘beam_remove_L.dst’ , ‘beam_core_L.dst’ (FRACTY<1).

(2) When the ETAC card is inserted before the WRBEAM card in the input file, the charge state of each particle is printed at the end of each line in the files concerned.

(3) Structure of the particle distribution file:

First line: NPOINT (number of particles), IFLAG, FREQ (frequency in MHz), (Wref(MeV),Wcog(MeV) if abs(IFLAG) ≥ 100)
Following lines: For each of the NPOINT particles:
A) Standard file (IFLAG=100 or 110):
X (cm) X'(rad) Y (cm) Y'(rad) phase (rad or ns) energy (MeV)
B) Standard file with particle number(IFLAG=101 or 111):
X (cm) X'(rad) Y (cm) Y'(rad) phase (rad or ns) energy (MeV) particle#
C) File with charge state(s) (IFLAG=102 or 112):
X(cm) X'(rad) Y(cm) Y’(rad) phase(rad or ns) energy(MeV) Q
where Q is the charge state.
D) File with charge state(s) and particle number (IFLAG=103 or 113):
X(cm) X'(rad) Y(cm) Y’(rad) phase(rad or ns) energy(MeV) Q particle#
where Q is the charge state.

The particle number printed corresponds to the number initially assigned to the particle (i.e. at the input).

(4) One can write the particle distribution file in binary format, see type code BINDIST. The default is ASCII.

(5) Files created with IFLAG=103 or 113 are currently not compatible with RDBEAM.

6.8.2 TYPE CODE: EMIT

Print the beam characteristics in the file 'dynac.short' at the place where EMIT card is introduced in the command list.

ENTRY: none

REMARKS:
The following explains the output in 'dynac.short' for an EMIT card in the input file:
First line
Particle reference: beta, kinetic energy (MeV), TOF (deg)
COG: kinetic energy (MeV), TOF (deg)
Offset between COG and particle reference: energy offset (MeV),
TOF offset (MeV-deg)

Second line
COG: coordinates for x, xp, y, yp (mm and mrad)

Third line (Courant-Snyder parameters):
Alpha-x, beta-x(mm/mrad), alpha-y, beta-y(mm/mrad), alpha-z,
beta-z (ns/keV)

Fourth line (longitudinal Courant-Snyder parameters and emit-z)
Alpha-z, beta-z(deg/keV), emit-z(non norm.,keV.deg), freq.(MHz)

Fifth line (bunch extensions (+-) in phase space [dPHI, dW])
dPHI(deg), dW(keV), r(12), long. emittance (keV.ns), particles left
Sixth line (beam extension and emittance in phase space [x, xp])
x(mm), xp(mrad), r(12), hor. Emittance (norm & non norm, mm.mrad)

Seventh line:
y(mm), yp(mrad), r(12), vert. emittance (norm & non norm, mm.mrad)

NOTE: The coefficient r(12) represents the correlation factor of the phase ellipse (i.e., the slope of the phase ellipse). In the TRANSPORT notation (see the ref. below), it is given by:

$$r(12)=\sigma_{12}/\sqrt{\sigma_{11}\sigma_{22}}=-\alpha/\sqrt{1+\alpha^2}$$

EXAMPLE:

```
*********      beam (emit card)      *********
0.02934  0.4850E+01 -0.6684E+02  0.4857E+01 -0.7059E+02   0.70399E-02  -0.37457E+01 MeV-deg
0.028   -0.048   -0.098   -0.817 mm and mrad
0.17448E+01   0.35760E+00  -0.16647E+01   0.23880E+00   0.38290E+00   0.35480E-02
0.3829   0.276931E+00   0.762572E+03 keV.deg   216.816 MHz
0.14532E+02    56.19   -0.3576    9.770 ns.keV     8296. particles left
2.912   16.375   -0.8676   0.69650E+00 mm.mrad (norm)   23.709 (non norm)
2.409   19.590    0.8572   0.71387E+00 mm.mrad (norm)   24.301 (non norm)
```

Ref.: CERN Document server Record#133647 Transport

6.8.3 TYPE CODE: EMITL

Like EMIT, print the beam characteristics in the file 'dynac.short' at the place where this card is introduced in the command list. In addition, read a label (LABEL) from the input file; this label will be printed to the dynac.short file. This label can be used for clarity or when running DYNAC from a script (e.g. for the purpose of fitting).

ENTRY:
1) LABEL

LABEL: Label (80 characters maximum) to be given to the data block in the dynac.short file.

6.8.4 TYPE CODE: EMIPRT

Print the characteristics of the beam in the file 'dynac.short' after each of the element types selected from the entry IEMQESG (see Type code EMIT).

NOTE: The beam characteristics are by default systematically printed after each cavity, single DTL gap, buncher, RFQ, electron gun and stripper.

ENTRY:

1) IEMQESG

IEMQESG = 0: disable the prints after optical lenses.
IEMQESG = 1: print after optical lenses (not after drifts).
IEMQESG = 2: print after optical lenses and positive drifts.

REMARK: The EMIPRT card can be placed in any arbitrary position in the command list.

6.8.5 TYPE CODE: DFLOCS

Print particle coordinates to file for named beam line elements (see remark 1). The list of named beam line elements for which particle distribution files need to be printed should be placed in a file. The format of the particle distribution files is the same as obtained with the WRBEAM card.

ENTRY:

1) filename
   where filename is the name (including path) of the list of named beam line elements for which particle distribution files need to be printed (see remark 2 for the file format)
2) IREC IFLAG
   whereby all distribution files created will be based on the IREC and IFLAG values set here

IREC = 0: the coordinates of the particles are printed relative to the COG
IREC = 1: the absolute energy of particles is printed, the phase is relative to the COG
IREC = 2: the coordinates of the particles are printed relative to the reference
IREC = 3: the absolute energy and phase of the particles is printed

IFLAG=100: File using 6 coordinates for each macro-particle (see remark 3), with phase in radians
IFLAG=101: Like IFLAG=100, but also includes particle number (see remark 3), with phase in radians
IFLAG=102: Like IFLAG=100, but also includes charge state(s) (see remark 3), with phase in radians
IFLAG=103: Like IFLAG=102, but also includes particle number (see remark 3), with phase in radians

IFLAG=110: File using 6 coordinates for each macro-particle (see remark 3), with phase in ns
IFLAG=111: Like IFLAG=110, but also includes particle number (see remark 3), with phase in ns
IFLAG=112: Like IFLAG=110, but also includes charge state(s) (see remark 3), with phase in ns
IFLAG=113: Like IFLAG=112, but also includes particle number (see remark 3), with phase in ns

EXAMPLE:

```
DFLOCS
C:\john\dynac\datafiles\my_dist_locations.txt
1 (IREC) 100 (IFLAG)
```

REMARKS:
(1) For the type codes of beam line elements listed under 5.2 and 5.3, one can (optionally) add a user defined name for the beam line element. This user defined name should be listed right behind and on the same line as the type code in the input file (see chapter 5), e.g.
CAVNUM MB-CM1-CAV2

Currently, the DFLOCS card can be used in conjunction with
  BMAGNET
  CAVMC
  CAVNUM
  DRIFT
  FSOLE
  MHB
  QUADRUPO
  SOLENO
Upon request, more beam line element types may be added to this list in following revisions of DYNAC.

(2) The format of the file that lists the named beam line elements (my_dist_locations.txt in the example above) for which particle distribution files need to be printed is

user_defined_element_name_1 indic_1
user_defined_element_name_2 indic_2
user_defined_element_name_3 indic_3
….
….
user_defined_element_name_n indic_n

Here, indic_n can be set to 0, 1 , 2 or 3. The meaning of these is:
0 (do not write a particle distribution file)
1 (write a particle distribution file at the input of the element)
2 (write a particle distribution file at the output of the element)
3 (write a particle distribution file at the input and output of the element)

The name of the particle distribution file will be the user defined element name followed by _in.dst or _out.dst (depending on the value of indic_n)

EXAMPLE:

Assume the DYNAC input file contains the following type codes and user defined names

CAVNUM MB-CM1-CAV1
..
CAVNUM MB-CM1-CAV2
..
CAVNUM MB-CM1-CAV3
..
CAVNUM MB-CM1-CAV4
..
CAVNUM MB-CM2-CAV1

Then the my_dist_locations.txt file (as in the example above) could look for instance like this
CAVNUM MB-CM1-CAV1 1
CAVNUM MB-CM1-CAV2 2
CAVNUM MB-CM1-CAV4 0
CAVNUM MB-CM2-CAV1 3

In this example (and with the DFLOCS card in the DYNAC input file), the following distribution files will be written:
CAVNUM MB-CM1-CAV1_in.dst
CAVNUM MB-CM1-CAV2_out.dst
CAVNUM MB-CM2-CAV1_in.dst
CAVNUM MB-CM2-CAV1_out.dst

No distribution file will be written for CAVNUM MB-CM1-CAV3 (because not listed in my_dist_locations.txt) and also not for CAVNUM MB-CM1-CAV4 (because indic_n = 0)

6.9 OUTPUT PLOT

6.9.1 TYPE CODE: EMITGR

X-X', Y-Y', Y-X and Z-Z' scatter plots

ENTRY:
1) TITLE

TITLE: Title of the graph (80 characters maximum)

2) IDWDP RMSMTP

IDWDP = 0: COG and synchronous particle coincide in Z-Z' plot (for instance in case of DTL Alvarez structure)

IDWDP = 1: COG and synchronous particle are distinct in Z-Z' plot (for instance in the case of IH structure)

RMSMTP: if RMSMTP is not zero, ellipses will be drawn on X-X', Y-Y' and Z-Z' plots. The ellipse emittance is corresponding to:

(Emittance of ellipse) = RMSMTP * (RMS emittance of X-X', Y-Y' or Z-Z')

3) XLIM1 YLIM1 XLIM2 YLIM2 XLIM3 YLIM3 XLIM4 YLIM4

XLIMn: Limits in (+ -) of horizontal coordinate (cm or deg)

YLIMn: Limits in (+ -) of vertical coordinate (mrad, cm or MeV)

REMARKS:

(1) Chapter 7.1 contains a sample output created by EMITGR.

(2) In the case of multi-charge state beams, the macro particles will automatically be color coded according to their charge state, see chapter 7 for more details.

(3) In the case of space charge dominated beams, color coding macro-particles according to their relative radius may be of interest, see type code ZONES and chapter 7 for more details.

(4) There is currently no auto-scaling on the XLIMn, YLIMn limits (i.e. do not set both of the (+ -) limits to zero)

EXAMPLE (for an IH structure):

EMITGR
BEAM AT IH INPUT (TITLE)
1 (IDWDP) 2.5 (RMSMTP)
2. (XLIM1 in cm) 20. (YLIM1 in mrad) 2. (XLIM2 in cm) 20. (YLIM2 in mrad) 2. (XLIM3 in cm) 2. (YLIM3 in cm) 40. (XLIM4 in deg) 3.5 (YLIM4 in MeV)

6.9.2 TYPE CODE: ENVEL

Envelope plots for X, Y, PHASE and ENERGY

ENTRY:

1) TITLE

TITLE: Title of the graph (60 characters maximum)

2) RMSMTP

RMSMTP: The size of the envelopes in the plots will correspond to the rms size of the envelope * RMSMTP.

3) ZDEB ZFIN

ZDEB: Starting position of the plot (m)

ZFIN: End of the plot (m)

4) XXMAX, XYMAX, YWMAX, YPMAX

XXMAX: Maximum for the scale on the horizontal beam envelope plot (cm)
XYMAX: Maximum for the scale on the vertical beam envelope plot (cm)
YWMAX: Maximum for the scale on the energy dispersion envelope plot (dW/W in per mil)
YPMAX: Maximum for the scale on the phase envelope plot (deg)

REMARKS:
If ZFIN is greater than the effective length of the lattice, the graphs are automatically stopped at this effective length.
If XXMAX is less or equal zero, auto scaling for this axis will be used. Idem for XYMAX, YWMAX and YPMAX.

6.9.3 TYPE CODE: PROFGR

Particle plots in X-Z, Y-Z and bunch profiles in X, Y, Z, X', Y' and Z'

ENTRY:

1) TITLE

TITLE: Title of the graph (80 characters maximum)

2) IDWDP ISCALE

IDWDP = 0: COG and synchronous particle coincide in Z-Z' plot (for instance in case of DTL Alvarez structure)

IDWDP = 1: COG and synchronous particle are distinct in Z-Z' plot (for instance in the case of IH structure)

ISCALE = 0: vertical scales of bunch profiles are not logarithmic

ISCALE = 1: vertical scales of bunch profiles are logarithmic

3) XLIM YLIM ZLIM DISTMIN

XLIM: Limits in (+ -) of vertical (X) scale in X-Z plot (cm)

YLIM: Limits in (+ -) of vertical (Y) scale in Y-Z plot (cm)

ZLIM: Limits in (+ -) of horizontal (Z) scale in X-Z, Y-Z plots (deg)

DISTMIN: the lower vertical limit on profile plots if ISCALE=1; otherwise, if ISCALE=0, it is a dummy value (DISTMIN will default to zero)

REMARKS:

(1) Chapter 7.1 contains a sample output created by PROFGR.

(2) Scales for the bunch profiles default to the following values:

- Horizontal scales: + - 5*RMS
- Vertical scales: from 0 to 1 if non-LOG scale and from DISTMIN to 1 if LOG scale.

(3) In the case of multi-charge state beams, the macro-particles will automatically be color coded according to their charge state, see chapter 7 for more details.

(4) In the case of space charge dominated beams, color coding the macro-particles according to their relative radius may be of interest, see type code ZONES and chapter 7 for more details.

EXAMPLE:

```
PROFGR
BEAM AT IH INPUT (TITLE)
0 (IDWDP) 0 (ISCALE)
1. (XLIM in cm) 1. (YLIM in 1cm) 40. (ZLIM in deg) 0 (DISTMIN, dummy here)
```

6.9.4 TYPE CODE: EMITGRD

Like EMITGR, but produces X-X', Y-Y', Y-X and Z-Z' intensity plots

ENTRY:

1) TITLE

TITLE: Title of the graph (80 characters maximum)

2) IDWDP RMSMTP

IDWDP = 0: COG and synchronous particle coincide in Z-Z' plot (for instance in case of DTL Alvarez structure)

IDWDP = 1: COG and synchronous particle are distinct in Z-Z' plot (for instance in the case of IH structure)

RMSMTP: if RMSMTP is not zero, ellipses will be drawn on X-X', Y-Y' and Z-Z' plots. The ellipse emittance is corresponding to:

(Emittance of ellipse) = RMSMTP * (RMS emittance of X-X', Y-Y' or Z-Z')

3) XLIM1 YLIM1 XLIM2 YLIM2 XLIM3 YLIM3 XLIM4 YLIM4

XLIMn: Limits in (+ -) of horizontal coordinate (cm or deg)

YLIMn: Limits in (+ -) of vertical coordinate (mrd, cm, MeV)

REMARKS:

(1) The different intensities differ in color; a color bar is shown next to each plot. Multi-charge state beams are colored by intensity, not by charge state. Coloring associated with the ZONES card is ignored.

EXAMPLE (for an IH structure):

```
EMITGRD
BEAM AT IH INPUT  (TITLE)
1 (IDWDP) 2.5 (RMSMTP)
2. (XLIM1 in cm) 20. (YLIM1 in mrad) 2. (XLIM2 in cm) 20. (YLIM2 in mrad) 2. (XLIM3 in cm) 2. (YLIM3 in cm) 40.
(XLIM4 in deg) 3.5 (YLIM4 in MeV)
```

6.9.5 TYPE CODE: PROFGRD

Like PROFGR, but produces intensity plots in X-Z, Y-Z and bunch profiles in X, Y, Z, X’, Y’ and Z'

ENTRY:

1) TITLE

TITLE: Title of the graph (80 characters maximum)

2) IDWDP ISCALE

IDWDP = 0: COG and synchronous particle coincide in Z-Z' plot (for instance in case of DTL Alvarez structure)

IDWDP = 1: COG and synchronous particle are distinct in Z-Z' plot (for instance in the case of IH structure)

ISCALE = 0: vertical scales of bunch profiles are not logarithmic

ISCALE = 1: vertical scales of bunch profiles are logarithmic

3) XLIM YLIM ZLIM DISTMIN

XLIM: Limits in (+ -) of vertical (X) scale in X-Z plot (cm)

YLIM: Limits in (+ -) of vertical (Y) scale in Y-Z plot (cm)

ZLIM: Limits in (+ -) of horizontal (Z) scale in X-Z, Y-Z plots (deg)

DISTMIN: the lower vertical limit on profile plots if ISCALE=1; otherwise, if ISCALE=0, it is a dummy value (DISTMIN will default to zero)

REMARKS:

(1) The different intensities differ in color; a color bar is shown next to each plot. Multi-charge state beams are colored by intensity, not by charge state. Coloring associated with the ZONES card is ignored.

(2) Scales for the bunch profiles default to the following values:

- Horizontal scales: + - 5*RMS

- Vertical scales: from 0 to 1 if non-LOG scale and from DISTMIN to 1 if LOG scale.

EXAMPLE:

```
PROFGRD
BEAM AT IH INPUT (TITLE)
0 (IDWDP) 0 (ISCALE)
1. (XLIM in cm) 1. (YLIM in 1cm) 40. (ZLIM in deg) 0 (DISTMIN, dummy here)
```

6.10 OTHER TYPE CODES

6.10.1 TYPE CODE: STOP

Ends simulation, this card is mandatory

ENTRY: none

6.10.2 TYPE CODE: ACCEPT

Find an acceptance for the beam at the input (based on particles remaining at the output of the command list). Plots are provided by the ACCEPT card (see remark).

ENTRY:

1) TITLE (for particles kept)

2) IDWDP RMSMTP

3) XLIM1 YLIM1 XLIM2 YLIM2 XLIM3 YLIM3 XLIM4 YLIM4

4) TITLE (for particles lost)

5) IDWDP RMSMTP

6) XLIM1 YLIM1 XLIM2 YLIM2 XLIM3 YLIM3 XLIM4 YLIM4

These parameters are the same as the ones in use in type code EMITGR (see this type code for definitions)

REMARK: To find an acceptance for the machine with the ACCEPT card, the code starts from emittances at the input of the command list. These emittances tend to grow through the accelerator and or may be modified by the REJECT card(s) at appropriate places in the command list.
A limited number of particles will make it to the end of the command list. The ACCEPT card will plot the input coordinates of the particles that made it to the end and will print the corresponding emittances in the file ’dynac.short’. Subsequently it will plot the input coordinates of the particles that did not make it to the end and will print the corresponding emittances in the file ’dynac.short’.
Furthermore, 2 particle distribution files will be generated: input_kept.dst (corresponding to the particles that made it to the end) and input_lost.dst (corresponding to the particles that did not make it to the end).

6.10.3 TYPE CODE: COMMENT

Add a comment to the list.

ENTRY:

1) FREETXT

FREETXT: (string) your comment (80 characters maximum)

REMARK: Comment line(s) may also be created directly **above** any type code entry by having as first character of such lines the semi-colon ‘;’.

6.10.4 TYPE CODE: ZONES

Define color coded zones in X-Y-Z space for plotting purposes. This can be of particular interest for the investigation of the behavior of space charge dominated beams. The 3D ellipsoid in X-Y-Z space is transformed into a sphere within which one can define zones in terms of normalized radii (multiples of the RMS radius a, b and c for x, y and z respectively i.e. x/a, y/b, z/c)).

ENTRY:

1) ITYP NZONES

ITYP: defines the type of zones.

ITYP=0: the zones are defined in the 3-d space (x/a, y/b, z/c)

ITYP=1: the zones are defined in the plane (x/a, y/b)

NZONES: number of zones (minimum 2, maximum 5, but set to 0 to turn off the effect of the ZONES card)

2) ZONES(1) . . . ZONES(I) ... ZONES(NZONES-1)

ZONES(I): zone limit in terms of normalized radius in RMS multiples. The number of zone limits to be specified is NZONES-1. If NZONES=0, one ZONES(I) value should be set (i.e. ZONES(1)), but its value is ignored.

EXAMPLE:

```
ZONES
0 (ITYP) 5 (NZONES)
0.5 (ZONES(1) 1.5 (ZONES(2)) 2. (ZONES(3)) 3. (ZONES(4))
```

This will set the limits of the zones for color coding macro-particles with normalized radius R in the following way:

Zone 1: 0 RMS less or equal R less than 0.5 RMS
Zone 2: 0.5 RMS less or equal R less than 1.5 RMS
Zone 3: 1.5 RMS less or equal R less than 2.0 RMS
Zone 4: 2.0 RMS less or equal R less than 3.0 RMS
Zone 5: R greater or equal to 3.0 RMS

REMARKS:

(1) The ZONES card can, for instance, be introduced after RDBEAM or GEBEAM cards. It will then apply color coding to the macro particles for all elements for subsequent EMITGR and PROFGR type codes.

(2) At the position where EMITGR, PROFGR type codes are introduced, plots will be produced showing the colors of particles as they were defined at the position of the ZONES card. This can be of particular interest when trying to understand halo formation in space charge dominated beams.

(3) The zones may be redefined anywhere along the accelerator/transport line simply by placing another ZONES card at the appropriate point. Again, following EMITGR and PROFGR type codes will then be using these newly defined zones. Setting NZONES to zero will turn off the effect of the ZONES card.

(4) In the case of a multi-charge state beam, the zones card will be ignored by the plotting routine.

6.10.5 TYPE CODE: T3D

This type code will cause DYNAC to write a TRACE3D input file. TRACE3D is a beam envelope code originally developed at LANL, Los Alamos, NM, USA. It is good practice to place this type code right after the first line in the input file.

ENTRY: none

EXAMPLE:

```
T3D
```

REMARKS:

(1) For long accelerating elements consisting of multiple cells, "gaps" (thin lens equivalent) will be created (TRACE3D element type 10), but only if the accelerating elements are represented in the DYNAC input file by the CAVNUM card (so not for the CAVMC card). The FIELD card will indirectly contribute to the input required by TRACE3D.

(2) Currently the following DYNAC elements are "translated" to the TRACE3D input file: DRIFT, QUADRUPO, SOLENO, CAVSC, CAVNUM, BUNCHER, INPUT, GEBEAM, RDBEAM, NEWF, SCDYNAC and BMAGNET.

(3) The T3D card will not work for a multi-charge state beam; an error message will be printed in the TRACE3D input file if an ETAC card is encountered.

(4) The following cards will be ignored: ZONES, STOP, DCBEAM, FIRORD, COMMENT, EMIT, EMITGR, WRBEAM, ENVEL, REJECT, CHASE, ACCEPT, EMITL, EGUN, COMPRES, RASYN, SECORD STRIPPER, EDFLEC, PROFGR, QUAELEC, CAVMC and BINDIST. The remaining cards have yet to be considered.

(5) The emittance sizes generated for the T3D input file are 4 times RMS (non-normalized); **this may need to be a different factor depending on the type of distribution, in particular when RDBEAM is used**.

6.10.6 TYPE CODE: BINDIST

BINDIST enables to print or read particle distributions in binary format. This can be achieved by placing a BINDIST card prior to the WRBEAM of interest in the DYNAC input file. Following the BINDIST card, any WRBEAM will write the distribution in binary format. If BINDIST is placed before RDBEAM, RDBEAM will read the distribution assuming it is in binary format. If no BINDIST card is present, the default is ASCII.

ENTRY: none

EXAMPLE:

```
BINDIST
```

REMARKS:
(1) The data in the particle distribution files (both binary and ASCII) written by DYNAC are not necessarily directly compatible with other beam dynamics programs.
(2) It is very likely that the binary format between different computers is different (e.g. 32 bit vs. 64 bit machines, LINUX vs. MAC vs. Windows).

7 DESCRIPTION OF THE GRAPHICS POST PROCESSOR PLOTIT
Based on dynplt V4R0 (16-Dec-2025)

7.1 INTRODUCTION TO PLOTIT

PLOTIT is a script that calls the dynplt FORTRAN post processor to produce graphics output for DYNAC, based on gnuplot. It has been tested on LINUX, MAC and MSWINDOWS systems and it is suggested to place it in the dynac/bin directory. When invoking PLOTIT (type PLOTIT on the command line), it will ask you if you are using MSWINDOWS or LINUX. In order to avoid having to answer this question every time you use PLOTIT, you can edit the plotit (LINUX, MAC) or plotit.bat (MSWINDOWS) file, which should be in the "bin" directory. Inside this file simply add the letter:
L or l (for the LINUX case) i.e. (../plot/dynplt l)
OR
M or m (for the MAC case) i.e. (../plot/dynplt m)
OR
W or w (for the MSWINDOWS case) i.e. (..\plot\dynplt w)

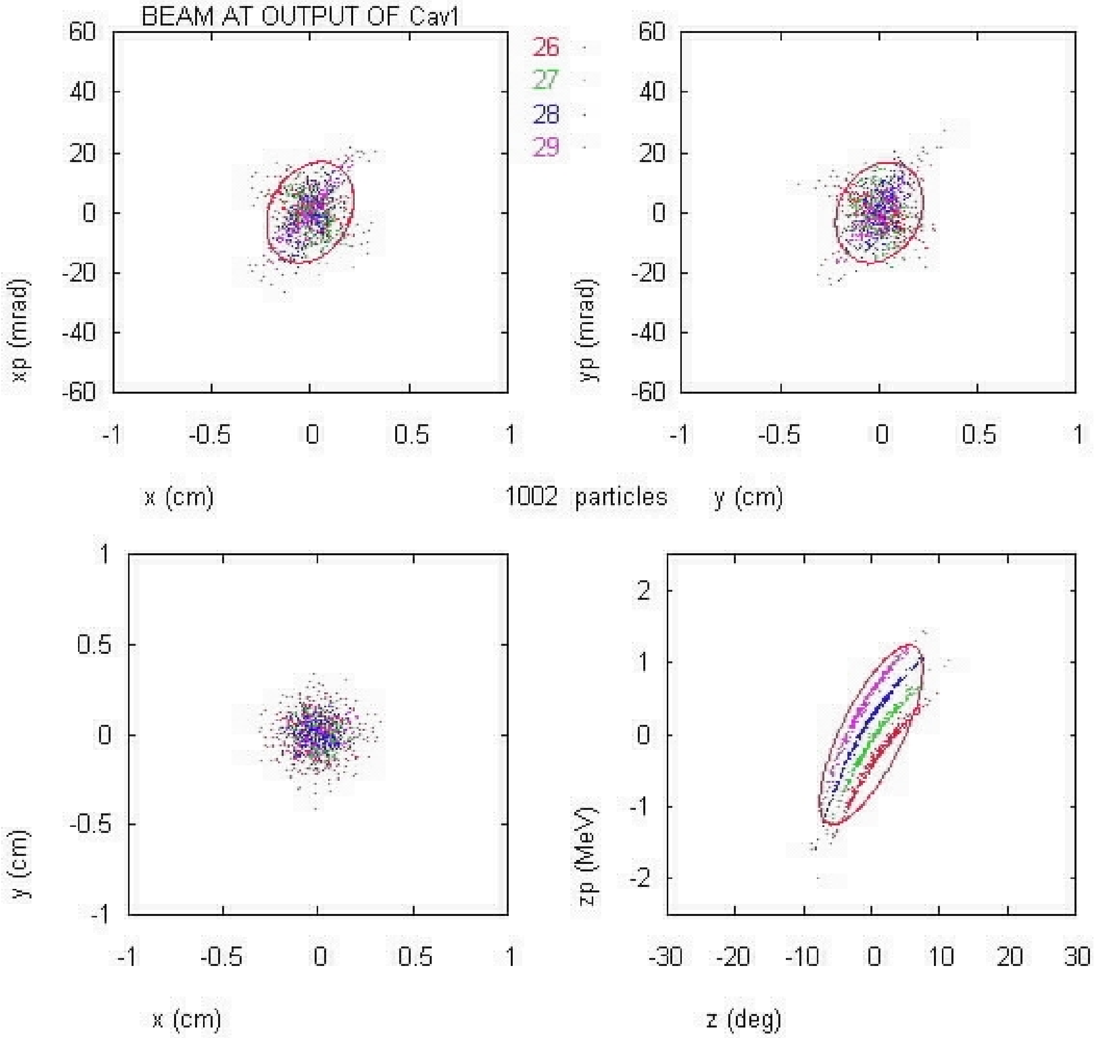
As mentioned, the PLOTIT script calls the dynplt program, which is located in the *plot* directory. Compiling dynplt.f08 with the complt script yields the dynplt program. To see the list of argument options for the dynplt program, type
path-to-dynplt/dynplt -h
The result of this should be:

```
PLOTIT V4R0 16-Dec-2025
 Command format:
 dynplt [-h] X [-tt] [-p] [-dg]
 where X is the operating system, which needs to be one of the following 3:
   L or l for LINUX
   M or m for MAC
   W or w for WINDOWS
 Optional arguments:
 -h will list the argument options (this list)
 -p can be used to specify the datafile path. There should be no
    space between -p and the path. This option is used by the
    DYNAC GUI.
 -tt can be used to specify the GNUPLOT terminal type, which defaults to:
    wxt for LINUX
    wxt for MAC
    wxt for WINDOWS
```

The default gnuplot terminal can be modified by using the -tt option. Here is an example when using a MAC:
../plot/dynplt m -ttxxxx
where xxxx is the gnuplot terminal name. Note that the aqua terminal on MAC yields a very slow response in gnuplot for scatter plots, whereas wxt is much faster.
If one wants to save the plots to picture files, this can be done by using one of the following 3 terminal types with the -tt option:
-ttjpeg
-ttpng
-ttgif

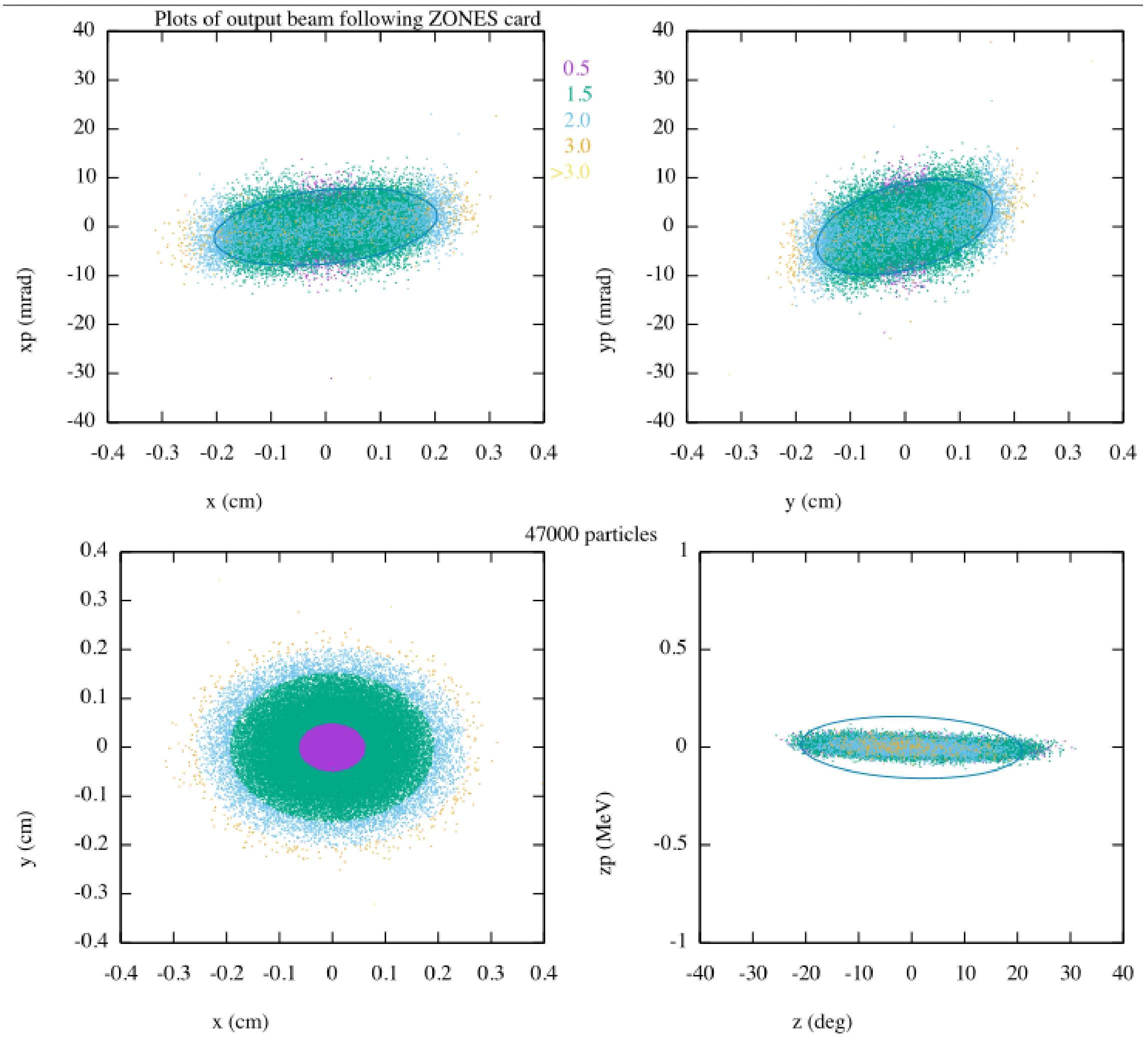
When using MINGW on MSWINDOWS, you may need to use WMG (or wmg) instead of W (or w). This is required for some versions of MINGW as certain gfortran intrinsics may behave differently (in particular "ctime", which would otherwise yield date and time reporting in the wrong format).

In the case of multi-charge state beams, macro particles of different charge state will be plotted with a different color. An index of the different colors is added to the plots. At present, for multi-charge state beams, the ZONES card will be ignored if entered. An example of such a plot is shown below, depicting a beam accelerated by a cavity and containing 4 different charge states (27, 28, 29 and 30).

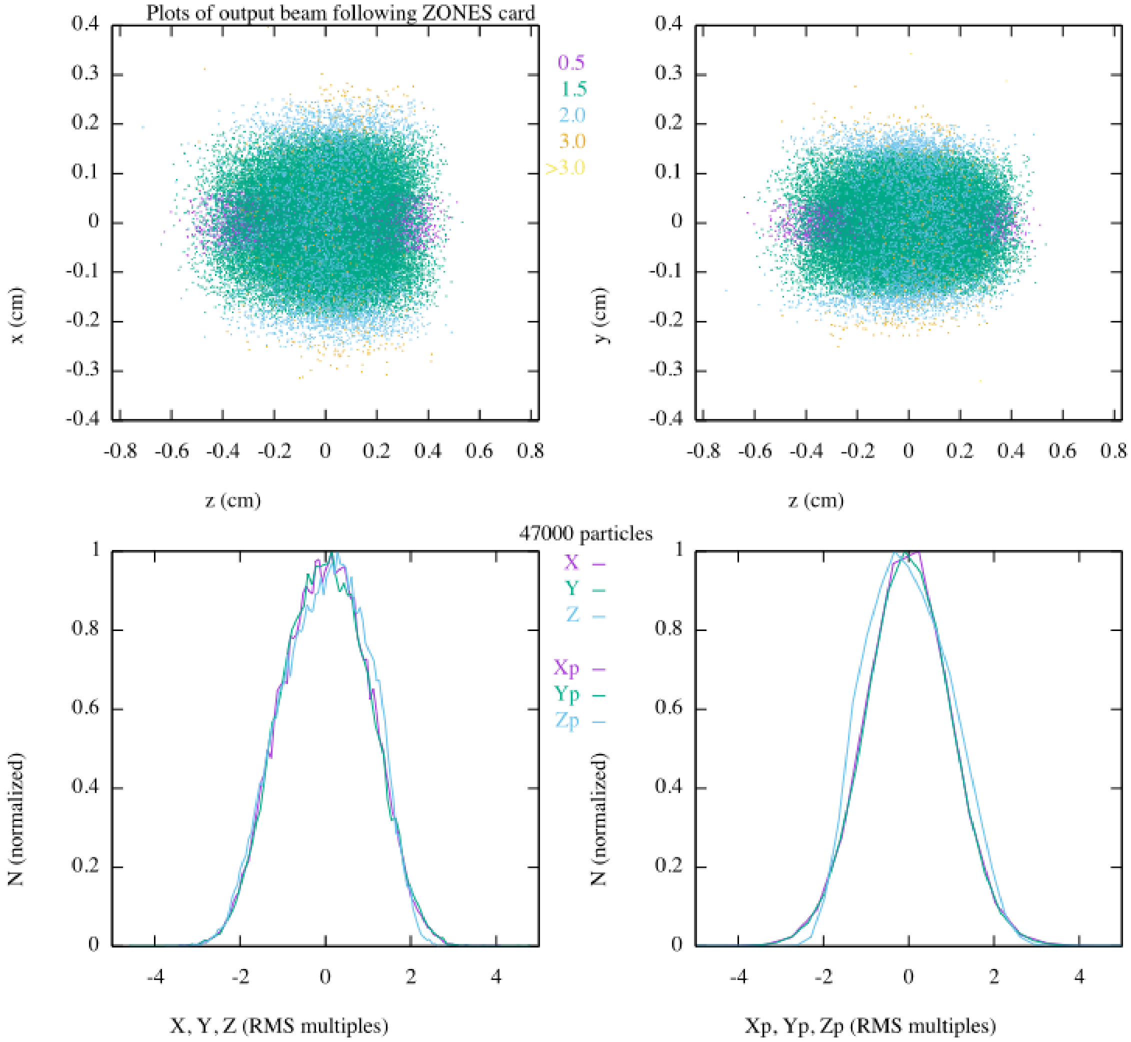


Plots resulting from an EMITGR card for a multi-charge state beam accelerated through a cavity.

In the case of single charge state beams, macro particles in different zones as (optionally) defined by the type code ZONES, will be plotted with a different color. An index of the different zones and colors is added to the plots.



Plots resulting from an EMITGR card with zones defined by the ZONES card.



Plots resulting from a PROFGR card with zones defined by the ZONES card.

7.2 HOW CAN PLOT FILES BE SAVED?

If one wants to save the plots to picture files, this can be done by using one of the following 3 terminal types with the -tt option:

```
-ttjpeg
-ttpng
-ttgif
```

Example:

```
../plot/dynplt m -ttpng
```

7.3 TIPS

MSWINDOWS: The size of the plotting window and the location where it will appear on the screen can be changed by setting the appropriate parameters in the gnuplot ini file.

8 APPENDIX

8.1 Glossary

CERN	**European Organization for Particle Physics**
COG	**Center of Gravity**
DTL	**Drift Tube Linac**
IH	**Inter-digital H (type of DTL structure)**
LANL	**Los Alamos National Laboratory**
RMS	**Root Mean Squared**
RFQ	**Radio Frequency Quadrupole**
TOF	**Time of Flight**
TTF	**Transit Time Factor**