How to use GICOSY

Version November, 2005

Contents

How to use GICOSY

GICOSY uses the method of transfer matrices, the elements of which are described by algebraic expressions. Each optical element, i.e. a drift distance, a fringing field, a magnetic or electrostatic quadrupole, a magnetic or electrostatic sector field, etc. is characterized by one transfer matrix. All transfer matrices are multiplied together to obtain the overall transfer matrix.

GICOSY uses parallelogram–like areas. It determines “normalized aberration coefficients” which assume an upright rectangle or an upright ellipse at the object in phase space and which assume an optical system amended by “virtual object lenses”. GICOSY provides a beam–envelope printer plot and at desired z–locations determines intensity–distribution plots. GICOSY also provides a number of graphic plots, depended on your environment (VAXstation, UIS window software).

The output of GICOSY is written onto a file called “GICOSYOUT.DAT”. The input of GICOSY must be on a file “GICOSYIN.DAT”. The first lines of this file can be used as comment. GICOSY ingnores them until the first line containing a (;) is found, this line is considered to be the first command for GICOSY. The last line in a new GICOSYIN.DAT must read: “END ;”. In between are the GICOSY–commands each of which must start on a new line. Each command can extend over 72 character and must end with (␣;), i.e. a semicolon preceded by a blank space. All characters behind the (;) symbol are regarded as comment. A line starting with a (;) symbol is a comment line. Each command must start with two or three characterizing words of each of which only the first letter is read. A blank space (␣) is the word limiter. Because GICOSY writes the entire input also to GICOSYOUT.DAT, this file, after beeing copied or renamed to GICOSYIN.DAT, can also work as a new input for GICOSY. The last line, however, does not have to be “END ;”.

0.1 General Input Commands

As soon as a GICOSY–command is identified by its characterizing words a string of default numbers is attached to it. The numbers contained in the command then change these default numbers one by one. In case there are fewer numbers given than contained in the default string, the leftover default values remain valid. For many commands however the use of default values is more or less senseless, especially for the description of the ionoptical elements.

0.1.1 The System Name

A name can be assigned to a calculation, which appears in the output and is also used internally to identify the system. The name is chosen using:

> **S**(YSTEM) **N**(AME) <titlestring> ;
> [example: S N 100.0 0.01 ;]

If the command does not appear, the system name defaults to TEST.

0.1.2 The Calculation Order

The highest order to which calculations will be performed can be chosen by the command:

> **C**(ALCULATION) **O**(RDER) <order> ;
> [example: C O 3 ;]
> [default: C O 1 ;]

In order to save computer time and to have an easy overview of the final output, it is recommended to keep the calculation order as small as possible.

0.1.3 The Calculation Mode

This command changes the way the transfer matrixes are calculated internally:

> **C**(ALCULATION) **M**(ODE) <mode> ;
> [example: C M 2 ;]
> [default: C M 1 ;]

In C M 1, transfer matrixes of the main optical elements are calculated by routines generated by the program Hamilton, in C M 2, transfer matrixes are calculated by using TPSA routines. In both calculation modes, the tranfer matrices are concatenated to get the system transfer matrix. In C M 3, no transfer matrixes for single optical elements are calculated, the system transfer matrix is derived from calculating the system as a whole. C M 2 and C M 3 both use an integrator. The accuracy of the integrator can be changed with

> **C**(ALCULATION) **A**(CCURACY) <accuracy> ;
> [example: C A 1.E-3 ;]
> [default: C A 1.E-4 ;]

0.2 Commands which Describe the Initial Particle Beam

Note that if any of the beam defining commands explained in sections 2 and 3 appear more than once, the latest command will override all previous ones.

0.2.1 Properties of the Reference Particle

The characteristic parameters of the reference particle are defined by the command:

> **R**(EFERENCE) **P**(ARTICLE)
> <energy K in MeV>
> <rest mass m_{00} in atomic mass units, i.e. 1/12 of a ^{12}C>
> <charge $\overline{q}e$ in charge units e, where $-e$ is the charge of one electron> ;
> [example: R P 100 4 1 ;]
> [default: R P 1 1 1 ;]

The charge of the particle is **always** assumed to be positive.

0.2.2 A Parallelogram–like Phase–Space Area

A parallelogram–like phase–space area is defined by the command:

> **P**(ARALLELOGRAM–LIKE) **X**(–direction)
> <source half width x_{00} in meter>
> <half aperture $\tan\alpha_{00}$ or a_{00} >
> <distance ℓ_x to the next angle–limiting aperture in meter> ;
> [example: P X 0.01 0.01 1.0 ;]
> [default: P X 0.01 0.01 0.0 ;]

Analogous definitions are necessary for the phase–space area in y–direction.

> [example: P Y 0.01 0.002 1.0 ;]
> [default: P Y 0.01 0.01 0.0 ;]

Note that usually $\ell_x \neq \ell_y$ and that $\ell_x = 0$ or $\ell_y = 0$ are interpreted as $\ell_x = \infty$ or $\ell_y = \infty$.

0.2.3 An Elliptical Phase–Space Area

An elliptical phase–space area is defined by the command:

T(WISS_PARAMETER) **X**(–direction)
< α_{Tx} >
< β_{Tx} in meter>
< $\varepsilon_x = x_{00}a_{00}$ in meter> ;
[example: T X 0.500 0.001 0.000001 ;]
[default: T X 0.001 0.001 0.000001 ;]

Analogous definitions are necessary for the phase–space area in y–direction.
Note that this definition is independent from one made by the P X command. This command is used to set the Twiss parameters, which can be viewed by the use of the P T command.
Note that the P E command also outputs the values of Twiss parameters. These parameters are derived from the phasespace defined by the P X and P Y command and are not influenced by the T X or T Y command.

[example: T Y 0.3 0.001 0.0000005 ;]

0.2.4 Maximal Energy– or Mass–Deviations from the Reference Particle

For some plot–commands and for the **P**(RINT) **N**(UMERICAL ...)–command one should define how large δ_K, δ_m or δ_p are assumed to be. The corresponding command is:

D(EVIATION) **P**(ARAMETER)
< δ_m in units, i.e. for 1 %: δ_m =0.01>
< δ_K in units, i.e. for 1 %: δ_K =0.01> ;
[example: D P 0.02 0.01 ;]
[default: D P 0. 0. ;]

If using mass–momentum coordinates for the matrix (see command **O**(UTPUT) **C**(OORDINATES)), the second parameter should be interpreted as the momentum deviation.

0.2.5 The floor coordinates

The floor coordinates may be set to arbitrary inital values with

F(LOOR) **C**(OORDINATES)
<U-coordinate in meter>
<V-coordinate in meter>
<W-coordinate in DEG> ;
[example: F C 3.5 20. 0 ;]
[default: F C 0 0 0 ;]

If these commands do not appear in the input, the default values are assumed nevertheless.

0.3 The Description of the Optical System

The Optical System consists of a sequence of ionoptical elements, like drift length, sectorfields and quadrupols. Real sectors and quadrupols have fringing fields, which extend into the nearby drift lenghts and cause distortions of the particle trajectories. GICOSY has two ways to take these effects into account. The first way is to calculate the actual shape of the fringing field by the use of fringing field coefficients, and than to obtain the transfer matrix of the fringing field by the use of an integrator. While this way leads to a very good calculation of the fringing field effects, it also uses a large amount of CPU time. The second way is the use of fringing field integrals, which offer a fast way to calculate the transfer matrix. Note also that the fringing–field effects are only small corrections so that usually approximate values of the integrals are satisfactory and that reasonably good results can be obtained if a set of standard fringing–field integrals are used, determined from a realistic fringing–field distribution, i.e. a field distribution in which Maxwell's equations are valid everywhere. However, before the system is finally built, one should perform calculations with fringing–field distributions that are as accurate as possible, obtained from measurements or explicit calculations. This will then modify the results slightly to first– and second–order where these modifications usually can be counterbalanced by varying the geometry or some quadrupole and hexapole parameters slightly. At last the accuracy of the calculation can be assured by the use of the coefficients and the integrator

The command to switch between both modes of calculation is:

> **F**(RINGING_FIELD) **T**(YPE) <type> <mode> ;
> [example: F F I ;]
> [default: F T S E ;]

<type> can be S(OFT) or I(NTEGRAL), where S means the use of the integrator and I denotes the calculation with integrals. The <mode> can be set to E(NABLE) or D(ISABLE). If the mode is DISABLE, the fringing fields are not calculated at all, although they appear in the input file. There is one exeption: if the fringing fields are used to describe the influence of an oblique edge of a sector, their calculation cannot be disabeled. Note, that a fringing field can only be defined between a sector or quadrupole or hexapole **and** a field–free region.

0.3.1 Start of the System

The sequence of optical elements should be started by the command:

S(TART) **S**(YSTEM) ;
[example: S S ;]

Before the Start System command, only the commands which define the reference particle and the phasespace should appear. The normal output of these commands in suppressed and appears in the printout of the header.

0.3.2 A Field–Free Region

The simplest optical element is a field–free region defined by the command:

D(RIFT) **L**(ENGTH) <length L in meter> ;
[example: D L 1.0 ;]

0.3.3 A Kicker

A simple device with 0.order–effects is a kicker defined by the command:

M(AGNETIC) **K**(ICKER) <deflection angle in DEG> ;
[example: M K 3.5 ;]

or by

E(LECTRIC) **K**(ICKER) <deflection angle in DEG> ;
[example: E K 4.8 ;]

0.3.4 A Thin Lens

A simple device with 1.order–effects is a thin lens defined by the command:

M(AGNETIC) **T**(HINLENS)
<refractive power f_x^{-1} in meter^{-1} >
<refractive power f_y^{-1} in meter^{-1} > ;
[example: M T 1.0 1.0 ;]

or by

E(LECTRIC) **T**(HINLENS)
<refractive power f_x^{-1} in meter^{-1} >
<refractive power f_y^{-1} in meter^{-1} > ;
[example: E T 1.0 1.0 ;]

0.3.5 Multipoles

A Magnetic Multipole

A general magnetic multipole, i.e. a superimposed quadrupole, hexapole and octopole, is defined by the command:

> **M**(AGNETIC) **M**(ULTIPOLE)
> <effective length ℓ of multipole in meter>
> < B_Q in T at pole tip>
> < B_H in T at pole tip>
> < B_O in T at pole tip>
> < B_D in T at pole tip>
> < B_Z in T at pole tip>
> <aperture radius G_0 in meter> ;
> [example: M M 0.2 1.0 0 0 0 0 0.2 ;]

B_Q, B_H, B_O, B_D and B_Z here are the quadrupole–, hexapole–, octopole–, decapole– and dodecapole–components of the magnetic flux density. For positive or negative values of B_Q, B_H, B_O, B_D and B_Z particles with positive x–values are driven towards or away from the optic axis, respectively.
This multipole–command is also applicable for a pure quadrupole, hexapole, octopole, decapole or dodecapol. However, to save computer time, it is advisable to use in such cases the following commands:

A Magnetic Quadrupole

A pure magnetic quadrupole is defined by the command:

> **M**(AGNETIC) **Q**(UADRUPOLE)
> <effective length ℓ of quadrupole in meter>
> < B_Q in T at pole tip>
> <aperture radius G_0 in meter> ;
> [example: M Q 0.2 1.0 0.02 ;]

A Magnetic Hexapole

A pure magnetic hexapole is defined by the command:

> **M**(AGNETIC) **H**(EXAPOLE)
> <effective length ℓ of hexapole in meter>
> < B_H in T at pole tip>
> <aperture radius G_0 in meter> ;
> [example: M H 0.2 0.5 0.02 ;]

A Magnetic Octopole

A pure magnetic octopole is defined by the command:

M(AGNETIC) **O**(CTOPOLE)
<effective length ℓ of octopole in meter>
$<B_O$ in T at pole tip>
<aperture radius G_0 in meter> ;
[example: M O 0.2 0.6 0.02 ;]

A Magnetic Decapole

A pure magnetic octopole is defined by the command:

M(AGNETIC) **D**(ECAPOLE)
<effective length ℓ of decapole in meter>
$<B_O$ in T at pole tip>
<aperture radius G_0 in meter> ;
[example: M D 0.2 0.6 0.02 ;]

A Magnetic Dodecapole

A pure magnetic octopole is defined by the command:

M(AGNETIC) **Z**(DODECAPOLE)
<effective length ℓ of dodecapole in meter>
$<B_O$ in T at pole tip>
<aperture radius G_0 in meter> ;
[example: M Z 0.2 0.6 0.02 ;]

An Electrostatic Multipole

An electrostatic multipole is defined by the command:

E(LECTROSTATIC) **M**(ULTIPOLE)
<effective length ℓ of multipole in meter>
$<V_Q$ in kV at electrode>
$<V_H$ in kV at electrode>
$<V_O$ in kV at electrode>
$<V_D$ in kV at electrode>
$<V_Z$ in kV at electrode>
<aperture radius G_0 in meter> ;
[example: E M 0.1 1.0 0 0 0 0 0.03 ;]

V_Q, V_H, V_O, V_D and V_Z here are the quadrupole–, hexapole–, octopole–, decapole– and dodecapole–components of the electrode potential. For positive or negative values of V_Q, V_H, V_O, V_D, V_Z particles with positive x–values are driven towards or away from the optic axis, respectively.

Analogously to the magnetic case the multipole–command is also applicable for a pure quadrupole, hexapole, octopole, decapole or dodecapole. However, to save computer time it is advisable to use in such cases the following commands:

An Electrostatic Quadrupole

A pure electrostatic quadrupole is defined by the command:

> **E**(LECTROSTATIC) **Q**(UADRUPOLE)
> <effective length ℓ of quadrupole in meter>
> < V_Q in kV at electrode>
> <aperture radius G_0 in meter> ;
> [example: E Q 0.1 1.0 0.03 ;]

An Electrostatic Hexapole

A pure electrostatic hexapole is defined by the command:

> **E**(LECTROSTATIC) **H**(EXAPOLE)
> <effective length ℓ of hexapole in meter>
> < V_H in kV at electrode>
> <aperture radius G_0 in meter> ;
> [example: E H 0.2 0.3 0.03 ;]

An Electrostatic Octopole

A pure electrostatic octopole is defined by the command:

> **E**(LECTROSTATIC) **O**(CTOPOLE)
> <effective length ℓ of octopole in meter>
> < V_O in kV at electrode>
> <aperture radius G_0 in meter> ;
> [example: E O 0.2 0.8 0.03 ;]

An Electrostatic Decapole

A pure electrostatic octopole is defined by the command:

E(LECTROSTATIC) **D**(ECAPOLE)
<effective length ℓ of decapole in meter>
$< V_O$ in kV at electrode>
<aperture radius G_0 in meter> ;
[example: E D 0.2 0.8 0.03 ;]

An Electrostatic Dodecapole

A pure electrostatic octopole is defined by the command:

E(LECTROSTATIC) **Z**(DODECAPOL)
<effective length ℓ of dodecapol in meter>
$< V_O$ in kV at electrode>
<aperture radius G_0 in meter> ;
[example: E Z 0.2 0.8 0.03 ;]

Fringing Fields for Magnetic or Electrostatic Multipoles

To third order, fringing fields for multipoles are effective only for their quadrupole– or hexapole–component. This fringing field is defined by the command:

F(RINGING) **F**(IELD) <table#> ;
[example: F F 2 ;]
[default: F F 1 ;]

The fringing–field integrals are taken from GIOSFF.DAT, if this mode is active; the coefficients are taken from GICOSYFF.DAT otherwise. Note that

- z is the coordinate along the optic axis and that
- G_0 is the aperture radius of the multipole.
- Note also that all integrals must be taken from z_a outside to z_b inside the main–field regions.

0.3.6 Sector Fields

A Magnetic Sector Field

A magnetic sector field is defined by the command:

M(AGNETIC) **S**(ECTOR_FIELD)
<radius of deflection ρ_{B0} in meter>
<angle of deflection ϕ_0 in degrees>

<half air gap G_0 in meter>
$< n_{B1} > < n_{B2} > < n_{B3} > < n_{B4} > < n_{B5} >$;
[example: M S 1.0 90 0.05 0.51 ;]

The magnetic flux density in this sector field equals

$$B(x, y = 0) = B_0[1 - n_{B1}\left(\frac{x}{\rho_{B0}}\right) - n_{B2}\left(\frac{x}{\rho_{B0}}\right)^2 - n_{B3}\left(\frac{x}{\rho_{B0}}\right)^3 - n_{B4}\left(\frac{x}{\rho_{B0}}\right)^4 - n_{B5}\left(\frac{x}{\rho_{B0}}\right)^5] \,.$$

In calculation mode 1 (C M 1) the matrix coefficients are only calculated up to second order for inhomogenous sector fields. For higher orders one has to use differential algebra (calculation mode 2 or 3). For first– and second–order calculations n_{B1} is defined for $-1 < n_{B1} < 1$. For higher–order calculations n_{B1} is defined only for $0 < n_{B1} < 1$.

Fringing Fields for Magnetic Sector Fields

Fringing fields for magnetic sector fields are defined by the command:

F(RINGING) **F**(IELD) <table#>
<entrance or exit angle ε in degrees>
<curvature ρ_{B0}/R of effective field boundary> ;
[example: F F 2 10.0 1.0 0.4853 ;]
[default: F F 1 0 0 ;] .

The fringing–field integrals are taken from GIOSFF.DAT, if this mode is active; the coefficients are taken from GICOSYFF.DAT otherwise. Note that

- ζ is a coordinate perpendicular to the effective field boundary,
- G_0 is the half air gap,
- the integral I_5 must be taken from ζ_a to the effective field boundary, i.e. to ζ =0,
- and the integrals I_1 and I_2 must be taken from ζ_a outside to ζ_b inside the main–field region.

Note that ε (see Fig. 4.8 in Ref. 7) is positive if the normal to the field boundary is further away from the center of curvature of the optic axis than the incoming or outgoing beam. Note further, that ρ_{B0}/R (see Fig. 8.6 in Ref. 7) is positive for a convex field boundary and negative for a concave one.

An Electrostatic Sector Field

An electrostatic sector field is defined by the command:

E(LECTROSTATIC) **S**(ECTOR-FIELD)
<radius of deflection ρ_{E0} in meter>
<angle of deflection ϕ_0 in degrees>
<half electrode gap G_0 in meter>
$< n_{E1} > < n_{E2} > < n_{E3} >$;
[example: E S 1.0 90 0.03 1 ;]
[default: E S 1.0 90 0.01 0 0 0 ;]

where the toroidicity of the mid–equipotential surface of the sector field is determined by R_E (see Fig. 4.17 in Ref. 7) with

$$\frac{\rho_{E0}}{R_E} = n_{E1} + n_{E2}\left(\frac{x}{\rho_{E0}}\right) + \frac{n_{E3}}{2}\left(\frac{x}{\rho_{E0}}\right)^2 + \ldots$$

In calculation mode 1 (C M 1;) the matrix coefficients are only calculated up to second order for inhomogenous sector fields. For higher orders one has to use differential algebra (calculation mode 2 or 3); For first– and second–order calculations n_{E1} is defined for $-2 < n_{E1} < 2$. For third–order calculations n_{E1} is defined only for $0 < n_{E1} < 2$. Special cases are:

- cylindrical electrodes: $n_{E1} = n_{E2} = n_{E3} = 0$
- spherical concentric electrodes: $n_{E1} = -n_{E2} = n_{E3} = 1$
- toroidal electrodes, i.e. electrodes which intersect the xz– as well as the yz–planes in circles: $n_{E1} \neq n_{E2} \neq n_{E3} \neq 0$

Fringing Fields for Electrostatic Sector Fields

Fringing fields for electrostatic sector fields are defined by the command:

F(RINGING) **F**(IELD) <table#>
<entrance or exit angle ε >
<curvatureρ_{E0}/R of effective field boundary>
[example: F F 3 0 0.09819 ;]
[default: F F 1 0 0 ;]

For the entrance angle ε **only** "0" is allowed up to now. The fringing–field integrals are taken from GIOSFF.DAT, if this mode is active; the coefficients are taken from GICOSYFF.DAT otherwise. Note that

- ζ is a coordinate perpendicular to the effective field boundary and that
- G_0 is the half electrode gap.
- Note further that all integrals must be taken from ζ_a outside to ζ_b inside the main–field region.

Note that ρ_{B0}/R is positive or negative for a convex or a concave field boundary, respectively.

0.3.7 Solenoid

A magnetic solenoid that rotates and focuses the beam

S(OLENOID) **L**(ENS)
<effective length in meter>
< B_0 in T inside the solenoid>
<aperture radius in meter> ;

The lens effect of a solenoid comes only in combination with its fringing fields.

F(RINGING) **F**(IELD) <table#>
[example: F F 0 ;
S L 1.0 4.0 0.05 ;
F F 0 ;]

The solenoid fringing fields are not implemented in the standard tables, yet.

0.3.8 Electrostatic Acceleration Column

Acceleration by an electrostatic longitudinal field defined by round apertures

A(CCELERATION) **C**(OLUMN)
<effective length in meter>
< E_0 of longitudinal field in MV/m>
<aperture radius in meter> ;

The lens effect comes only from the fringing fields.

F(RINGING) **F**(IELD) <table#>
[example: F F 0 ;

A C 1.0 0.25 0.05 ;
F F 0 ;]

The acceleration column fringing fields are not implemented in the standard tables, yet.

0.3.9 Changing the Direction of Deflection

In case of a positive radius GICOSY assumes that a sector field deflects the particle beam in the negative x–direction, i.e. to the right, looking downstream. A negative radius specifies the opposite direction. The old method with an additional matrix is still supported, it may only lead to wrong system plots. Thus the direction of deflection can be changed by:

C(HANGE) **B**(ENDING_DIRECTION) ;
[example: C B ;]

0.3.10 Rotation around the Optical Axis

R(OTATION) **M**(ATRIX)
<length in plot in meter>
<rotation angle in degree>
<aperture in plot in meter>˜;
[example: R M 0 90 0.10 ;]

The coordinate system is rotated around the optical axis. Thus beamlines in three dimensions can be calculated.

0.3.11 A displaced beam

A beam can be shifted from the symmetry axis of the elements by:

S(HIFT) **B**(EAM)
<x-displacement in meter>
<a-angle displacement in rad>
<y-displacement in meter>
<b-angle displacement in rad> ;
[example: S B 0.02 0.001 ;]

This command can be used to describe a displaced beam at the entrance of the system, it can also be used to describe a displaced optical element by displacing the beam before and after this element. WARNING: this looks a little bit silly in a graphical output.

0.4 The Selection of printable GICOSY Output

The GICOSY output always starts by listing the corresponding input file. The second portion of the output contains a compilation of the source– and particle–definitions and a detailed description of the optical system under consideration. In front and in back of each drift distance the floor coordinates U, V are listed together with the angle of the beam direction W. Note that the first sector field is always assumed to deflect to the right, i.e. towards negative V–values.

At any position z_i, corresponding to the position z_i in the input file, the following effects can be initiated:

0.4.1 The Printout of Transfer Matrices

The coordinates of the System Transfer Matrix

GICOSY uses symplectic coordinates with mass and energy as coordinates for transfer matrices. Sometimes it is desirable to see the Matrix based on mass and momentum instead, or to compare the Matrix to results obtained from another program like GIOS. The command that changes the coordinates is:

> **O**(UTPUT) **C**(OORDINATES)
> <coord1> <coord2>
> <coord3> <coord4> ;
> [example: O C N M L N ;]
> [default: O C S E G ;]

coord1 can be S(YMPLECTIC) or N(ONSYMPLECTIC), switching between the two different definitions of a and b. coord2 can be E(NERGY) or M(OMENTUM). coord3 can be G(ICO), which means time_difference×velocity, T(IME) or L(ENGTH). coord4 can be N(ONSCALED) or S(CALED), and has effect only, if coord3 is T(IME) or L(ENGTH). In this case, if coord4 is S(CALED), the time or length elements of the matrix are divided by the flight time or the travel distance of the reference particle. To compare the transfer matrix to GIOS, use the command O C N E T S. To compare the tranfer matrix to programs like TRANSPORT or MIRKO, use the command O C N M L N.
Note that when using coordinates different from the default, some commands (like Draw Beam etc.) may take more time than usual, because the matrix has to be converted to the new coordinates very often. Note also that in this case changing the coordinates in the middle of the system causes error.

The format of the System Transfer Matrix

There is also the possibility to change the format of the printout, making a comparison with GIOS more easy. The command is:

> **O**(UTPUT) **F**(ORMAT) <format> ;
> [example: O F G ;]
> [default: O F N ;]

format can be either N(ORMAL) or G(IOS).
When using a high calculation order, the printout of a single transfer matrix can extend over several pages of the output file. The order of the printout can be chosen to be smaller than the calculation order by:

> **O**(UTPUT) **O**(RDER) <order> ;
> [example: O O 2 ;]

Sometimes the resulting transfer matrix has many elements which are in the vicinity of 0., but are all different, thus making the output a little bit confusing to read. With:

> **O**(UTPUT) **T**(olerance) <value> ;
> [example: O T 1.E-5 ;]
> [default: O T 0. ;]

you can force GICOSY to output all elements of the transfer matrix, which absolute values are smaller than <value> to appear as 0.

The Printout of one System Transfer Matrix

A transfer matrix from z_0 to z_i is printed by the command:

> **P**(RINT) **M**(ATRIX) <order> ;
> [example: P M 1 ;]

where order is the output order of the matrix, overriding the values in den **CO** or **OO**–commands, if given. Note that one overall transfer matrix is automatically printed at the end of each GICOSY calculation; therefore, a **PM**–command is needed only when intermediate results are desired.

The Printout of all Transfer Matrices of an Optical System

A transfer matrix for each optical element as well as the full transfer matrix from $z = 0$ to $z = z_i$ are printed by one command at the beginning of the optical system:

P(RINT) **A**(LL) ;
[example: P A ;)

Note that this command will produce a very lengthy output and should therefore be used cautiously. The order of the output can be selected with the **OO**–command. This command will not work when using calculation mode 3.
The special command P A B E ; will produce ascii-files of all optical elements excluding the drift spaces. This is useful for Monte Carlo programs such as MOCADI.

The Printout of the Magnitudes of Aberrations

The matrix elements $(r_i|x^k a^\ell y^m b^n \delta_K^o \delta_m^p)$ multiplied by $x_{00}^k a_{00}^\ell y_{00}^m b_{00}^n \delta_{K0}^o \delta_{m0}^p$, are determined and listed in the same fashion as transfer matrices by the command:

P(RINT) **N**(UMERICAL_TRAJECTORY_DEVIATIONS) <order> ;
[example: P N 2 ;]

The **PN**–command provides a simple way to judge the relative importance of aberrations directly.

0.4.2 The Printout of the Position of the Next Images, Waists and Pupils

The distances from z_i to the next image, waist and pupil are calculated by the command:

P(RINT) **I**(MAGES_WAISTS_PUPILS) ;
[example: P I ;]

0.4.3 The Printout of the Beam Diameters, Twiss Parameters and Phase Shifts

The beam envelopes and Twiss parameters at $z = z_i$ as well as the phase shift ψ_i between z_0 and z_i are printed for parallelogram–like and for elliptical phase–space areas by the command:

P(RINT) **E**(NVELOPE_VALUES) ;
[example: P E ;] The Twiss parameters are valid only for the phasespace defined by the P X and P Y commands. To print out the Twiss parameters for the phasespace defined by the T X and T Y commands, use the command:

P(RINT) **T**(WISSPARAMETERS) ;
[example: P T ;]

0.4.4 Printer plot of the Beam Envelope

Each single printout of the system requires at least the same amount of time than the calculation of the system, therefor the commands should be used with caution. A printer plot is produced of the optical system under consideration as well as of the calculated first–order envelope of the real beam (together with a beam formed by trajectories that start from a point source under angles A and B) by the command:

P(RINT) **B**(EAM) <number of printer lines>
<shown maximal x–width in meter>
<shown maximal y–width in meter> ;
[example: P B 100 0.15 0.15 ;]
[default: P B 100 0.10 0.10 ;]

This command may occur only once.

0.4.5 Short Summary of the System

The GICOSY output file holds a complete description of the system, but it may become very large, if there are many elements in the system. Using the command:

W(RITE) **S**(UMMARY) ;
[example: W S ;]

a compressed summary of the system can be produced, which is written to the file GICOSYSUMM.DAT. Note that this summary does not contain full information about the elements.

0.4.6 Printer Plot of Phase–Space Distributions

A printer plot of a phase–space intensity distribution is initiated at z_i by the command:

P(RINT) **P**(HASE_SPACE) <identifier> (f,g)
<total number of particles>
<f–dimension of plot> <g–dimension of plot>
<total number of mass values>
<total number of energy values> ;
[example: P P E (X,A) 1000 0.025 0.050 1 1 ;]

The following choices are possible for either f or g in (f,g):

X = final X–position	XI = initial X–position
A = final A–angle	AI = initial A–angle

G = mass (quantized)
D = energy
Y = final Y–position YI = initial Y–position
B = final B–angle BI = initial B–angle
T = final time

The total number of mass values are distributed equally over the mass widths defined in the "D(EVIATION) P(ARAMETER)"–command. The same is valid for the number of energy values. If this number is 0, the distribution of energy values is continous.

There are three possible choices for the <identifier>:

- P: the initial distributions in f and g are chosen independently;
- E: only those (x, a) or (y, b) combinations are chosen which lie inside (x_{00},a_{00}) or (y_{00},b_{00}) ellipses;
- S: only those (x, a), (y, b) or (x, y) combinations are chosen which lie inside (x_{00},a_{00}), (y_{00},b_{00}) or (x_{00}, y_{00}) ellipses.

All parameters following (f,g) can be omitted. As default the <identifier> equals P and GICOSY assumes 1000 particles and one mass; the size of the plot is chosen automatically by the program according to a first–order calculation. If the identifier for the phasespace is followed by a T, the plot is made with particles already traced, provided those particles exist. The values for the number of particles, masses and energies have no meaning in this case.

0.4.7 The Printout of variables and Matrix elements

Variables, Strings and Matrix elements can be printed out either to the screen or to the GICOSY output file. Numerical constants are **not** valid arguments of this command. The command which prints to the screen is:

P(RINT) **S**(CREEN) <output list> ;
[example: P S '[X,A] = ' [X,A] ;]
[example: P S 'LEN1 = ' LEN1 ;]

The items in the output list are separated by blanks. The output to the file is done in a similar way:

P(RINT) **F**(ILE) <output list> ;

0.4.8 Disable the standard output

During the calculation, GICOSY writes the description of each optical element which is computed to the output file. For systems with many elements, this can result in very large files, which contain almost identical information for each calculation. To save disk space, use the command:

> **S**(ET) **O**(utput) <Enable or Disable> ;
> [example: S O D ;]
> [default: S O E ;]

to turn the output off and on again.

0.5 The Selection of Graphic Output

The commands which produce graphic plots write their output to the file META.DAT, where the graphic is stored device-independently. This metafile can then be viewed on a VAXstation running the UIS window software by using the program META.

0.5.1 Graphic Plot of Trajectories in x– and y–direction

A graphic output of single particles is obtained by a command very similar to the printer plot by:

> **D**(RAW) **B**(EAM)
> <shown maximal x–width in meter>
> <shown maximal y–width in meter>
> <number of steps in z direction>
> < x_0: number of trajectory starting points in the x–coordinate>
> < a_0: number of trajectory starting points in the a–coordinate>
> < g_0: number of trajectory starting points in the g–coordinate>
> < d_0: number of trajectory starting points in the d–coordinate>
> < y_0: number of trajectory starting points in the y–coordinate>
> < b_0: number of trajectory starting points in the b–coordinate>
> < no : type of multidimensional phase space> ;
> [example: D B 0.1 0.1 100 2 2 1 3 2 2 ;]
> [default: D B 0.1 0.1 1 1 1 1 1 1 1 ;]

$no = 1$, ions start only either in x or y plane, from a parallelogram like phase space volume
$no = 2$, ions start in all combinations of x and y, from a parallelogram like phase space volume
$no = 3$, ions start only either in x or y plane, from an elliptical phase space volume
$no = 4$, ions start in all combinations of x and y, from an elliptical phase space volume

The calculation is performed to the order given in the **CO**–command.

The total number of starting points of trajectories is distributed equally spaced within the initially defined phase–space areas x_0, a_0 or y_0, b_0 or within the range of masses (g_0) and energies (d_0) as defined in the **DP**–command. Note that the total number of trajectories increases rapidly according to the number of combinations of coordinates. This command may occur only once.

0.5.2 Graphic Plot of top–view of System

The command is similar to the **DB**–command, but it gives a more realistic impression how to actual system looks:

> **D**(RAW) **S**(YSTEM)
> <shown maximal x–width in meter>
> <scale factor x–direction/z–direction>
> <number of steps in z direction>
> < x_0: number of trajectory starting points in the x–coordinate>
> < a_0: number of trajectory starting points in the a–coordinate>
> < g_0: number of trajectory starting points in the g–coordinate>
> < d_0: number of trajectory starting points in the d–coordinate> ;
> [example: D S 0.1 3 100 3 2 1 1 ;]
> [default: D S 0.1 1 1 1 1 1 1 1 1 ;]

The calculation is performed to the order given in the **CO**–command.

The total number of starting points of trajectories is distributed equally spaced within the initially defined phase–space areas x_0, a_0 or within the range of masses (g_0) and energies (d_0) as defined in the **DP**–command. Note that the total number of trajectories increases rapidly according to

$$n_{tot} = x_0 * a_0 * g_0 * d_0.$$

This command may occur only once.

0.5.3 Graphic Plot of the Fields

A plot showing the strength of the fields of the optical elements is produced by

> **D**(RAW) **F**(IELDS)
> <shown maximal x–width in meter>
> <max. magnetic field strength in Tesla>
> <max. electric voltage in kV> ;
> [example: D F 0.2 1.5 2.2 ;]

This command may occur only once.

0.5.4 Graphic Plot of Expressions

A plot of expression containing matrix elements is produced by

> **D**(RAW) **E**(XPRESSION)
> <shown maximal x–width in meter>
> <max. value of the expression>
> <number of steps in z direction>
> <expression in paranthesis> ;
> [example: D E 0.2 1.5 20 ([X,X]*0.1+[X,A]*0.015) ;]

This command may occur up to 10 times.

0.5.5 Graphic Plot of Phase–Space Distributions

A graphic plot of a phase–space intensity distribution is initiated at z_i by the command:

> **D**(RAW) (P)**H**(ASE_SPACE) <identifier> (f,g)
> <total number of particles>
> <f–dimension of plot>
> <g–dimension of plot>
> <total number of mass values>
> <total number of energy values>
> <up to 10 values for contour lines (% of max. intensity)> ;
> [example: D H E (X,A) 1000 0.025 0.050 1 1 ;]

If no values for the contour lines are given, the values 0.1,0.3,0.5,0.7 and 0.9 are used. Look at the **PP**–command for more information.

For the commands **DB**,**DS**,**DE** a large number of steps in z direction gives better results, but will increase the calculation time. For draft pictures, set the number of steps to 1. Since a step cannot be larger than an optical element, this is often good compromise between quality of the picture and calculation time.

0.6 Arithmetic calculations

0.6.1 The Definition of Arithmetic Expressions

The definition of a name for an arithmetic expression is performed by the command:

VAR(six–letter_name) = … arithmetic expression … ;
[example: XVAL=[X,X]*0.1 + [A,A]*0.01 ;]
[example: A=2 *0.5 −SIN(B) ;]

The … arithmetic expression … may contain the following items:

- constants [numbers like 4, 0.5 or 1.E-6]
- elements of the transfer matrix at z_i, for instance, [X,A], [Y,B], [X,AA], [X,AAD], …
- operators [like (+), (^), (#EQ#), (#MAX#), …]
- functions [like SIN,COS,LOG,EXP,TAN, …]
- names of earlier defined variables
- system parameters like PI$, XM$ or TXA$

Variables, constants, operators and functions are used like in an ordinary programming language like FORTRAN. System parameters are like variables, but with an $ at the end. Unlike variables, their values cannot be changed by the user directly. System parameters are calculated by GICOSY during runtime. They have a threefold purpose:

1. to spare the user the work to enter basic constants like π or the speed of light directly
2. to supply the user with values defined by commands like P X or R P
3. to supply the user with certain numbers other than the matrix elements of the transfer matrix, which are calculated internally by GICOSY and are tedious or impossible to get by the use of arithmetic expressions alone

For a complete list of the available operators, functions and system parameters, see at the end of this manual.

0.6.2 The Use of an Arithmetic Expression

Use of a **previously defined** name of an arithmetic expression, i.e. an "algebraic variable" in a GICOSY–command, is achieved by replacing a constant in a command by a variable, or by writing the expression directly as a parameter of a command

[example: D L LEN1 ;]
[example: D L (LEN1 + LEN2) ;]

0.7 Fitting in GICOSY

0.7.1 Definition of a fit

It is possible tell GICOSY to change certain parameters of the system to minimize a variable. GICOSY needs to know the variables that are to be changed, and the locations in which the variables appear. The command which starts a fit is:

> **F**(IT) **B**(EGIN) <list of variables> ;
> [example: F B A B C ;]

The list can consist of up to 10 variables. The variables have to be defined prior to the F B command. There are a few restrctions for these variables:

- The variables may appear only in one fit. Use different variables for different fits.
- The variables must not be used as parameters of commands prior to the fit.
- Do not change the values of these variables with assignment statements after the fit.

Following the F B command are GICOSY commands, in which the fit variables have to appear. At the position where the variables that are to be fitted can be computed, the command:

> **F**(IT) **V**(ARIABLES) <list of fit variables> ;
> [example: F V F1 F2 ;]

must be inserted. Note that there is a difference between naming only one or several variables; in each case, different routines are used internally.
The command:

> **F**(IT) **E**(ND)<tolerance> <max. Iterations> <fit routine> ;
> [example: F E 1.E-5 150 1 ;]
> [default: F E 1.E-4 100 1 ;]

ends the fit. The value for the fit routine for a single fit variable may be between 1 and 6. If more than one variable is to be fitted, the value of the fit routine must be 1, which is the default. GICOSY tries to change the variables in a way that the fit variables become 0. The success of the fit depends on the chosen fit variable, the fit routine, the tolerance and the starting values for the variables. The results of the fit can be looked up in the GICOSY output. They are also written in a more compact form onto the file GICOSYFIT.DAT.
Sometimes it may be desirable to disable a fit for one calculation. This can be done by

commenting out the Fit Begin, Fit Variables and Fit End commands. A quicker way is to use the command:

> **F**(IT) **D**(ISABLE) ;
> [example: F D ;]

which may appear between a Fit Begin and Fit End and will prevent this fit from being executed.

0.7.2 Definition of variable limits

During a fit it may happen that the fit routine is trying to achieve its goal by assigning crazy values to the variables (like negative values to a drift length). It is therefor necessary to set certain limits for the variables. This can be done by:

> **VAR**(six–letter_name) $<$ <lower limit> <upper limit> $>$;
> [example: A $<$ 0.5 1.5 $>$;]
> [example: B $<$ (0.99∗B) (1.01∗B) $>$;]

0.8 Optical Systems consisting of Repetitive or of Mirror–Symmetric Cells

0.8.1 The Definition of a Block

A BLOCK of a system is characterized by a name which consists of several letters. Only the first letter of the name is used to identify the block. A block is defined by:

> **B**(LOCK) **B**(EGIN) <block name> ;
> . . .
> . . .
> **B**(LOCK) **E**(ND) ;

0.8.2 Using an earlier defined Block

The input command lines of a defined block are included in an optical system by one of the commands:

B(LOCK) **I**(NCLUDE) <block name> ;
B(LOCK) **I**(NCLUDE) <block name> R(EVERSE) ;

In the second case, the command lines of the block are included in reverse order. This is useful for the design of mirror symmetric systems. Note that the use of blocks does not save computation time.

0.9 Execution Control

GICOSY looks at the input file as a sequence of commands, which are executed one after another. The following commands change this behavior, allowing lines in the input file to be skipped or repeated several times.

0.9.1 Stopping the execution

To end a GICOSY calculation before reaching the **END ;**–command, use:

E(ND) **E**(XECUTION) ;
[example: E E ;]

0.9.2 Creating a Loop

You can repeat an arbitrary number of commands more than once by constructing a loop. A loop is started with

W(HILE) **B**(EGIN) <variable> ;
[example: W B A ;]

and ended with

W(HILE) **E**(ND) ;
[example: W E ;]

The commands in between are repeated until <variable> becomes smaller than 0. The value of <variable> has to be changed, or you risk to create an infinite loop.

0.9.3 Creating Labels

Every line in the GICOSY input file can have an optional label. This label consists of a sequence of up to 10 letters, concluded by a :.

[example: MAGNPART: D L 0.1 ;]

0.9.4 Using Labels

You can tell GICOSY to continue the calculation at a specific label with

> **G**(OTO) **L**(ABEL) <label> ;
> [example: G L MAGNPART ;]

This is similar to the GOTO command in a Fortran program.

If you want to return to your previous point in the input, use:

> **G**(OTO) **S**(UBROUTINE) <label> ;
> [example: G S MAGNPART ;]

GICOSY will execute the commands after the label and jump back when encountering:

> **R**(ETURN) **S**(UBROUTINE) ;
> [example: R S ;]

0.10 Other commands

0.10.1 Storing a Transfer Matrix on a file

The actual transfer matrix of the system is written to a file by the command

> **S**(AVE) **M**(ATRIX) <filename> <opt. identifier> ;
> [example: S M MAT1 A ;]

<filename> may consist of up to 6 characters. The file gets the extension .MAT.
Option A demands ASCII-Format. The file gets the extension .ASC.

0.10.2 Using the stored Transfer Matrix

The transfer matrix on a file is included in an optical system by the command

> **I**(NCLUDE) **M**(ATRIX) <filename> <opt. identifier> ;
> [example: I M MAT1 A ;]

It is checked that the system parameters used to calculate the matrix (particle energy) are the same as in the actual system. The file must have the extension .MAT or in case of option A for the ASCII-Format the extension .ASC.

0.10.3 Changing of filename

GICOSY uses default file names for the files containing the fit information or the grafic output. To change these names, use the command

> **S**(ET) **F**(ILENAME) <filename> <type> ;
> [example: S F FITOUT F ;]

<type> can be F, which means setting the filename for the fit output, , G to change the grafic filename or E to specify the names of the export files of all matrices. The filename must be given without extension.

0.10.4 Reading the coordinates of particles

The positions of particles can be read into GICOSY by the command:

> **I**(NCLUDE) **P**(ARTICLES)
> <filename>
> <mode of include, Trace oder Notrace> ;
> [example: I P RAYFIL T ;]

<filename> may consist of up to 6 characters. The file must have the extension .RAY. If this command is used, the particles will be used in the commands **PP** and **DH**. If the trace mode is chosen for these commands. If mode Trace is used, the particles are multiplied with the actual system transfer matrix. With this command, the trace mode is activated, the particles will be traced trough the following optical elements. The first line of RAYFIL must contain the word RAYFILE and the number of coordinates for the particles.

0.10.5 Saving the coordinates of particles

The positions of particles can be saved to a file with:

> **S**(AVE) **P**(ARTICLES)
> <filename>
> <number of coordinates, between 4 and 7> ;
> [example: S P RAYFIL 5 ;]

<filename> may consist of up to 6 characters. The file gets the extension .RAY. The actual particles positions are saved on the file with the specified number of coordinates.

0.10.6 Generating a distribution of particles

The positions of particles can be generated with the random generator with:

G(ENERATE) **P**(ARTICLES)
<type of phasespace:P,E or S>
<number of particles>
<number of coordinates>
<number of masses>
<number of number of energies> ;
[example: G P P 1000 6 3 0 ;]

The number of masses and energies have a meaning only when the number of coordinates is bigger than 5 or 6. A mass or energy number of 0 means a continous distribution. The particles will be traced through the following elements. The coordinates of the particles will be inside the region defined by the PX,PY and DP commands.

0.10.7 Beam Displacement

The generation of particles will yield a distribution "around" the optical axis. The shift the distribution away from this axis, use the command

B(EAM) **D**(ISPLACEMENT) <type of coordinate> <displacement> ;
[example: B D Y 0.01 ;]

This command also affects the implicit generation of particles for the PP and DH commands. type of coordinate may be X,A,Y,B,G or D.

0.10.8 Stopping the trace of particles

Sometimes, when particles are generated or read from a file

C(HANGE) **T**(RACE) <mode of change> ;
[example: C P N ;]

If the mode of change is N, the tracing of particles is stopped for the following elements. If the mode is E, the particles are also erased from memory.

0.10.9 Resetting the system transfer matrix

To set the system transfer matrix back to unity, use the command:

U(NIT) **M**(ATRIX) ;
[example: U M ;]

You don't have to use this command at the beginning of a calculation, because GICOSY resets the system transfer matrix automatically.

0.10.10 Changing the values of Variables interactively

During a GICOSY calculation, the values of variables can be changed interactively, using:

I(NTERACTIVE) **C**(ALCULATION) ;
[example: I C ;]

You can enter variable assignments using the keyboard. Leave the mode by entering ; in the first column.

0.10.11 Random generator

GICOSY uses its own random generator, which produces the same pseudo random numbers each time GICOSY is called. To switch to a different set of numbers, initialize the random generator using a different seed with:

I(NITIALIZE) **R**(ANDOMGENERATOR) <seed> ;
[example: I R 5 ;]

Seed must be an integer between 1 and 10 .

0.11 Appendix

List of operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Exponentiation
#DIM#	Distance
#DIST#	Distance
#EQ#	1. if expression is true
#GE#	
#LE#	
#LT#	
#MAX#	Maximum
#MIN#	Minimum
#MOD#	Modulo
#NE#	

List of functions

ABS	absolute value of argument
ACOS	arcus cosine, argument in rad
ACOSD	arcus cosine, argument in DEG
ASIN	arcus sinus, argument in rad
ASIND	arcus sinus, argument in DEG
ATAN	arcus tangens, argument in rad
ATAND	arcus tanges, argument in DEG
COS	cosine, argument in rad
COSD	cosine, argument in DEG
COSH	cosine hyperbolicus, argument in rad
COSHD	cosine hyperbolicus, argument in DEG
EXP	exponentiation, base e
FAC	
LOG	logarithm
LOG2	
LOG10	
NINT	next integer
SIGN	the sign of the argument
SIN	sinus, argument in rad
SIND	sinus, argument in DEG
SINH	sinus hyperbolicus, argument in rad
SINHD	sines hyperbolicus, argument in DEG
SQRT	square root
TAN	tangens, argument in rad
TAND	tangens, argument in DEG
TANH	tangens hyperbolicus, argument in rad
TANHD	tangens hyperbolicus, argument in DEG

List of system parameters

ADIST$
AM$
AMDIST$
AMU$
AMUMEV$
BDIST$
BMDIST$
BM$
CLIGHT$
DM$
EX$
EY$
EZERO$
GM$
LDIST$
LMDIST$
LX$
LY$
PI$
REFE$
REFM$
REFP$
REFZ$
TXA$
TXB$
TXG$
TXAI$
TXBI$
TXGI$
TYA$
TYB$
TYG$
TYAI$
TYBI$
TYGI$
UC$
VC$
VELOC$
WC$
XDIST$
XMDIST$
YDIST$
YMDIST$
ZPOS$