CTH-IEFT/PP-1992-15 ISSN

CASINO, A CODE FOR SIMULATION OF CHARGED PARTICLES IN AN AXISYMMETRIC TOKAMAK

By

Ö. Dillner

INSTITUTIONEN FÖR ELEKTROMAGNETISK FÄLTTEORI
CHALMERS TEKNISKA HÖGSKOLA

CASINO

A code for simulation of charged particles in an axisymmetric Tokamak

Örjan Dillner

Institute for Electromagnetic Field Theory and Plasma Physics
Chalmers University of Technology
S - 412 96 Göteborg, Sweden

Abstract

The present report comprises a documentation of CASINO, a simulation code developed as a means for the study of high energy charged particles in an axisymmetric Tokamak. The background of the need for such a numerical tool is presented. In the description of the numerical model used for the orbit integration, the method using constants of motion, the Lao-Hirsman geometry for the flux surfaces and a method for reducing the necessary number of particles is elucidated. A brief outline of the calculational sequence is given as a flow chart. The essential routines and functions as well as the common blocks are briefly described. The input and output routines are shown. Finally the documentation is completed by a short discussion of possible extensions of the code and a test case.

1. BACKGROUND

The guiding centres of charged particles moving along the curved field lines in a Tokamak magnetic field, tend to deviate from the flux surfaces on which they are produced, as a result of radial drifts caused by the curvature and the gradient of the magnetic field. Large deviations may occur for high energy trapped particles. The corresponding drift orbits may intersect the wall of the plasma container and result in loss of particles.

The confinement properties of such high energy charged particles have important implications for e.g. wall loading due to blistering from lost high energy charged particles, burn-up studies, high energy particles created by ICRF-heating or neutral beam. and injection problems concerning sawtooth and/or MHD induced particle losses.

Furthermore, in an ignited DT fusion plasma, the power losses are assumed to be compensated by the power deposition of the fusion generated alpha particles. Thus, the fraction of promptly lost high energy alphas is an important input parameter in the analysis of the plasma energy balance. Finally, finite orbit effects tend to broaden the power deposition profile of the fusion generated alphas.

In order to be able to study the orbit characteristics of high energy particles in Tokamaks we have developed a simulation code (CASINO). At the present stage the code assumes an axisymmetric magnetic field and neglects collisions but the code could be expanded to nonaxisymmetric situations and to allow for collisions. The code is especially useful as a tool to investigate the behaviour of high energy particles in Tokamaks with more realistic, i.e. more complicated, cross section geometries.

To get a reasonable accuracy in the results from particle simulations using Monte-Carlo methods we may have to follow thousands of particles. That implies that the code has to be fast. This problem is met in the CASINO code by using constants of motion (c.o.m.), instead of direct path integration. Furthermore, there is a special routine for reducing the necessary number of simulations.

CASINO utilizes analytical data input but is prepared for the use of experimental data. There is also the option of running the code for simulation of separate orbits including plotting possibility.

2. MODEL USED

2.1 Constants of motion - c.o.m.

The aim of the code is to predict the guiding centre orbits of single (high energy) particles. It is constructed around a core, a particle follower, which utilizes the method of constants of motion. This method can be used for an axisymmetric Tokamak if the collision times are longer than the bounce times. Under these conditions the canonical momentum

$$P_\varphi = mR^2\dot{\varphi} + qRA_\varphi$$

as well as the magnetic moment

$$\mu = \frac{v_\perp^2}{B}$$

and the particle energy

$$E = \frac{1}{2}mv^2$$

will be preserved during the orbit. Here R is the toroidal radius, $\dot{\varphi}$ is the toroidal angular velocity, m and q are the mass and the charge of the particle, A_φ is the toroidal component of the vector potential, and v and $v_\perp$ are the total velocity and the velocity component perpendicular to the magnetic field respectively.

For the toroidal velocity component, v_{TOR}, we have

$$v_{TOR} = R\dot{\varphi} = \pm v\sqrt{1 - \frac{\mu B}{v^2}\frac{B_{TOR}}{B_{TOT}}}$$

where B_{TOT} and B_{TOR} are the total magnetic field and the toroidal magnetic field component, respectively.

The vector potential A_φ is related to the magnetic flux $\tilde{\psi}(R)$ through a toroidal circle with radius R as

$$2\pi RA_\varphi = \tilde{\psi}(R)$$

Thus we can rewrite the constant of motion as

$$P_\varphi = \pm \frac{mR\,B_{TOR}}{Ze\,B_{TOT}} \sqrt{\frac{2E}{m} - \mu B_{TOT}} - \psi(R)$$

where $\psi(R) = [\tilde{\psi}(R) - \tilde{\psi}(R_0)]/2\pi$. Z is the charge number of the particle and R_0 is the toroidal radius of the centre of the cross-section. In the code, P_φ is represented as ANGMOM (the constant value) and P_ANMO, a function calculating the current value of P_φ.

The magnetic moment, MAGMOM, depends on the initial pitch-angle P_A, PANGEL, according to

$$\mu = \frac{2E}{mB_{TOT}}(1 - P_A^2)$$

where

$$P_A = \cos\varphi = \frac{v_{\parallel}}{v}$$

2.2 Geometry

The fact that the code is dealing with axisymmetric Tokamaks reduces the simulation geometry to the two-dimensional cross-section. Since the plasma parameters ion and electron temperature and particle density are supposed to be constant on the magnetic flux surfaces, it is advantageous to have a representation of the geometry where one of the two coordinates is constant on the flux surfaces. For that purpose the code uses the Lao-Hirsman representation of the flux surfaces.

In the Lao-Hirsman geometry the toroidal radius is represented as

$$R(\rho,\chi) = R_0(\rho) + a\rho\cos\chi + R_2(\rho)\cos 2\chi$$

and the z-coordinate as

$$Z(\rho,\chi) = E(\rho)\,[a\rho \sin\chi - R_2(\rho)\sin 2\chi]$$

where the dimensionless ρ, "the radius", and χ, "the poloidal angle", constitute an oblique coordinate system. A constant ρ (also designated "flux surface label") implies the same flux surface, i.e. the poloidal flux is constant. $R_0(\rho)$ is the shift function, that expresses the shift of the centre of the flux surface. $E(\rho)$ is the ellipticity and $R_2(\rho)$ is the triangularity of the plasma cross-section. These functions are represented as second degree polynomials in the code. The triangularity polynomial cannot, however, have a zero degree coefficient.

The contravariant base vectors are

$$\mathbf{a}_\rho = \frac{\partial \mathbf{r}}{\partial \rho} = \frac{\partial R}{\partial \rho}\hat{R} + \frac{\partial Z}{\partial \rho}\hat{Z}$$

$$\mathbf{a}_\chi = \frac{\partial \mathbf{r}}{\partial \chi} = \frac{\partial R}{\partial \chi}\hat{R} + \frac{\partial Z}{\partial \chi}\hat{Z}$$

$$\mathbf{a}_\varphi = \frac{\partial \mathbf{r}}{\partial \varphi} = R\hat{\varphi}$$

and the Jacobian

$$V = R\left(\frac{\partial R}{\partial \rho}\frac{\partial Z}{\partial \rho} - \frac{\partial R}{\partial \chi}\frac{\partial Z}{\partial \chi}\right)$$

The components of the metric tensor become

$$g_{11} = \mathbf{a}_\rho \mathbf{a}_\rho = \left(\frac{\partial R}{\partial \rho}\right)^2 + \left(\frac{\partial Z}{\partial \rho}\right)^2$$

$$g_{12} = g_{21} = \mathbf{a}_\rho \mathbf{a}_\chi = \frac{\partial R}{\partial \rho}\frac{\partial R}{\partial \chi} + \frac{\partial Z}{\partial \rho}\frac{\partial Z}{\partial \chi}$$

$$g_{22} = \mathbf{a}_\chi \mathbf{a}_\chi = \left(\frac{\partial R}{\partial \chi}\right)^2 + \left(\frac{\partial Z}{\partial \chi}\right)^2$$

$$g_{33} = \mathbf{a}_\varphi \mathbf{a}_\varphi = R^2$$

$$g_{13} = g_{31} = g_{23} = g_{32} = 0$$

2.3 The magnetic field

The toroidal magnetic field is represented as a paramagnetic function (PARMAF) divided by toroidal radius R (RMAJ)

$$B_{TOR} = \frac{PARMAF}{R} = \frac{B_0 R_0 (1 + \varepsilon_1 \rho + \varepsilon_2 \rho^2)}{R}$$

The poloidal magnetic field is given indirectly by the poloidal flux ψ (POFLUX). The flux growth rate perpendicular to the flux surface is

$$\partial\psi_{\perp} = B_{POL} R \partial r_{\perp} = |\nabla\psi| \partial r_{\perp}$$

where

$$|\nabla\psi| = \left|\frac{\partial\psi}{\partial\rho}\nabla\rho\right| = \left|\frac{\partial\psi}{\partial\rho}\right|\sqrt{g^{11}} = \left|\frac{\partial\psi}{\partial\rho}\right|\frac{1}{|V|}\sqrt{g_{22}g_{33}} = \left|\frac{\partial\psi}{\partial\rho}\frac{\sqrt{\left(\frac{\partial R}{\partial\chi}\right)^2 + \left(\frac{\partial Z}{\partial\chi}\right)^2}}{\frac{\partial R}{\partial\chi}\frac{\partial Z}{\partial\rho} - \frac{\partial R}{\partial\rho}\frac{\partial Z}{\partial\chi}}\right|$$

Thus the poloidal field, B_{POL}, is given by

$$B_{POL} = \left|\frac{\partial\psi}{\partial\rho}\frac{\sqrt{g_{22}}}{V}\right|$$

and the total magnetic field is then

$$B_{TOT} = \sqrt{B_{TOR}^2 + B_{POL}^2}$$

2.4 The current

The total current, I, is related to the poloidal field according to

$$I = \frac{1}{\mu_0}\int B_{POL} \cdot dr \approx \frac{1}{\mu_0}\sum B_{POL}\sqrt{g_{22}}\,\Delta\chi$$

where the summation is done over the outermost flux surface.

If the program is to run with a specified total current instead of poloidal flux, the best way, although a little longwinded, is to obtain the total current as follows.

1. Assign a value to the poloidal flux function coefficients corresponding to the desired current profile.

2. Run the program in mode MC=true and IPNUMB= IRNUMB=2.

3. Check the corresponding result for total current, CURRENT2, in output file for025.

4. Scale the poloidal flux function coefficients down in a degree corresponding to the ratio of actual current to desired current.

5. Run with the new coefficients and check the resulting current. If it is not correct, repeat from point 1, otherwise run in desired mode with these flux coefficients. Do not forget to set the right IPNUMB and IRNUMB

2.5 Advancing the guiding centre

For every time interval DEL_T, the guiding centre of the simulated particle is advanced as follows

1. Take a step

$$d\mathbf{r} = \Delta\rho\, \mathbf{a}_\rho + \Delta\chi\, \mathbf{a}_\chi$$

where $\Delta\rho$ and $\Delta\chi$ are determined by $v_{\|}$ and v_{drift} times DEL_T.

2. At the point $\mathbf{r}$ + $d\mathbf{r}$ take a perpendicular step $d\mathbf{r}_i$ whose size is determined as a fraction (CTEST) of $|d\mathbf{r}|$.

3. ($\mathbf{r}$ + $d\mathbf{r}$) and ($\mathbf{r}$ + $d\mathbf{r}$ + $d\mathbf{r}_i$) are then used for interpolation so that the angular momentum (P_ANMO) will remain constant.

The details of this procedure can be elucidated as follows:

1. The velocity along the magnetic field lines is

$$v_{\|} = v\sqrt{1 - \frac{\mu B_{TOT}}{v^2}}$$

and the time change of the parallel velocity can be written as

$$\frac{dv_{\parallel}}{dt} = \mathbf{v}_{gc}\cdot\nabla v_{\parallel} = \mathbf{v}_{\parallel}\cdot\left[v\sqrt{1 - \frac{\mu B_{TOT}}{v^2}} \right] = -\frac{1}{2}\mu \left| \nabla_{\parallel} B_{TOT} \right|$$

where

$$\nabla_{\parallel} B_{TOT} = \frac{\mathbf{B}}{B}\cdot\nabla B = \frac{B_{POL}}{B_{TOT}}\frac{\partial B}{\partial \chi}\frac{\mathbf{a}_{\chi}}{\sqrt{g_{22}}}\cdot\mathbf{a}_{\chi} = \frac{B_{POL}}{B_{TOT}}\frac{\partial B}{\partial \chi}\frac{1}{\sqrt{g_{22}}}$$

Thus

$$\frac{dv_{\parallel}}{dt} = -\frac{1}{2}\mu\frac{B_{POL}}{B_{TOT}}\frac{\partial B}{\partial \chi}\frac{1}{\sqrt{g_{22}}}$$

and the expected value of the parallel velocity after the time step Δt is

$$v_{\parallel exp} = v_{\parallel} + \frac{dv_{\parallel}}{dt}\Delta t$$

An estimate of the average parallel velocity during the time interval is therefore

$$v_{\parallel aver} = \frac{1}{2}(v_{\parallel exp} + v_{\parallel})$$

Thus, with no drift motion included, the charged particle would move along the flux surface while changing the toroidal angle according to

$$\Delta\chi_1 = v_{\parallel aver}\,\Delta t\frac{B_{POL}}{B_{TOT}}\frac{1}{\sqrt{g_{22}}}$$

The deviation of the particle from the flux surface is due to the drift velocity

$$\mathbf{v}_{drift} = \frac{m}{q}\left(v_{\parallel}^2 + \frac{v_{\perp}^2}{2}\right)\frac{B_{TOR}}{B_{TOT}^3}\,\mathbf{B}\times\nabla B$$

The resulting expression is

$$\mathbf{B}\times\nabla B = \frac{R}{V}\left[\frac{g_{12}}{\sqrt{g_{22}}}B_\chi\frac{\partial B}{\partial\chi} - \sqrt{g_{22}}B_\chi\frac{\partial B}{\partial\rho}\right]\hat{\varphi} +$$

$$+\frac{\sqrt{g_{11}}}{V}\left[\sqrt{g_{22}}B_\chi\frac{\partial B}{\partial\varphi} - RB_\varphi\frac{\partial B}{\partial\chi}\right]\hat{\rho} +$$

$$+\frac{\sqrt{g_{22}}}{V}\left[\sqrt{g_{33}}B_\varphi\frac{\partial B}{\partial\rho} - \frac{g_{12}}{\sqrt{g_{22}}}B_\chi\frac{\partial B}{\partial\varphi}\right]\hat{\chi}$$

If we restrict ourselves to the drift over the cross-section and assume axial symmetry, the drift during a time interval Δt causes the deviations:

$$\Delta\chi_2 = \left[v_\parallel^2 + \frac{\mu B}{2}\right]\frac{R}{V}\frac{B_{TOR}}{B_{TOT}^3}\frac{\partial B_{TOT}}{\partial\rho}\frac{m}{q}\Delta t$$

$$\Delta\rho = -\left[v_\parallel^2 + \frac{\mu B}{2}\right]\frac{R}{V}\frac{B_{TOR}}{B_{TOT}^3}\frac{\partial B_{TOT}}{\partial\chi}\frac{m}{q}\Delta t$$

The total first, uncorrected, prediction to advance the particle is therefore

$$\Delta\mathbf{r} = \Delta\rho\,\mathbf{a}_\rho + (\Delta\chi_1 + \Delta\chi_2)\,\mathbf{a}_\chi$$

2. The perpendicular interpolation step $d\mathbf{r}_i$ can be derived according to

$$0 = \Delta\mathbf{r}_i\cdot\Delta\mathbf{r} = g_{11}\Delta\rho\Delta\rho_i + g_{12}\Delta\rho\Delta\chi_i + g_{12}\Delta\chi\Delta\rho_i + g_{22}\Delta\chi\Delta\chi_i$$

Thus

$$\Delta\rho_i = -\frac{u_2}{u_1}\Delta\chi_i$$

where

$$u_1 = g_{11}\Delta\rho + g_{12}\Delta\chi$$

$$u_2 = g_{12}\Delta\rho + g_{22}\Delta\chi$$

The length of the interpolation step is set to be

$$|\Delta\mathbf{r}_i| = |\Delta\mathbf{r}|\,\mathrm{CTEST}$$

and therefore

$$|\Delta\chi_i| = \frac{|\Delta r|\,\mathrm{CTEST}}{\sqrt{g_{11}\left(\frac{u_2}{u_1}\right)^2 - 2\,g_{12}\left(\frac{u_2}{u_1}\right) + g_{22}}}$$

2.6 Reducing the necessary number of simulated particles

For the purpose of reducing the number of numerically simulated particles we use the following method:

1. Starting at the plasma limiter (ρ=1, χ=0) the radius is divided into segments. The closer to the limiter a particle is produced, the higher the possibility that the particle will be lost. Thus, in order to obtain the best statistical result for the loss ratio, the division into segments are made tighter close to the limiter. The centre of a segment ρ_2 is determined by the centre of the preceding segment ρ_1 according to

$$\rho_2 = \rho_1 - \frac{\mathrm{PTDENS}(\rho_1)}{\mathrm{IRNUMB}} = \rho_1 - \frac{\sqrt{1-\rho_1^2}+0.01}{1.01\ \mathrm{IRNUMB}}$$

If, in another case the characteristics of the plasma centre are more important, the weight function PTDENS has to be changed to emphasize the relative importance of the particles produced close to the toroidal axis.

2. In the same manner we divide the "pitch angle" space from P_A= -1.0 to P_A=1.0 into IPNUMB equally sized segments, DPANG, that is

$$\Delta P_A = \mathrm{DPANG} = \frac{2.0}{\mathrm{IPNUMB}}$$

3. From the centre (ρ_{beg}, $P_{A_{beg}}$, χ=0) of every segment in this "radius-pitch angle space", a particle orbit simulation is started. Every particle produced in this segment $[\Delta\rho, \Delta P_A]_{beg}$ is considered to follow the same orbit. The total number of simulated orbits will be around $[1.6\ \mathrm{IRNUMB} \cdot 2\ \mathrm{IPNUMB}]$

4. With the assumption of equal possibility for a particle to be produced with any pitch angle between -1 and 1, and

with a production rate of $S(\rho)$ particles per unit volume, the number of produced particles per unit time, Q, can be determined as

$$Q(\mathbf{r}) = S(\rho)\,\Delta^3 r \,\frac{\Delta P_A}{(\Delta P_A)_{TOT}}$$

where $(\Delta P_A)_{TOT}$ is the total pitch angle area, i.e. $(\Delta P_A)_{TOT}$ = 2.0 and the space volume element

$$\Delta^3 r = 2\pi\, V\, \Delta\rho\, \Delta\chi$$

where V is the Jacobian.

The different components $\Delta\rho, \Delta\chi, \Delta P_A$ of the "segment volume" will change during the simulation of an orbit but at every time step all particles produced inside are associated with the same orbit.

If we have zero time derivative for the distribution function, i.e.

$$\frac{df(\mathbf{r},\mathbf{v})}{dt} = 0$$

then

$\Delta^3 r\, \Delta P_A$ = constant (during each orbit simulation)

Thus

$$Q(\mathbf{r}) = Q_{beg} = S(\rho)\,[\Delta^3 r]_{beg}\,\frac{1}{\text{IPNUMB}}$$

5. In this way the total number of particles to be assigned to the same destiny as the simulated particle is

$$S_A = 2\pi\, V\, \frac{(\Delta\rho\, \Delta\chi)_{beg}}{\text{IPNUMB}} \sum S(\rho)$$

where $\Delta\rho_{beg}$ is the length of the radial segment and $\Delta\chi_{beg}$ is the change in angle coordinate χ at the first step.

NOTE! The method demands a constant time interval, Δt

3. NUMERICS

3.1 Philosophy

The code was originally thought as a numerical tool to be used against JET. This is the reason for the special geometry used, the Lao-Hirsman geometry. This representation of the flux surfaces is especially advantageous for modelling the conditions at JET.

This is also the background for using the magnetic flux as input. The current density is not directly experimentally measurable. Measuring coils are placed inside the Tokamak. The measured result used together with the Grad-Shafranov equation establishes the flux picture.

The basic philosophy in the development of the code have been that the code should be unsensitive to the origin of the input data. However, in the current version the routines to read off the experimental data are to be added.

3.2 Flow chart - a simple outline

In order to facilitate future reading of the program, a brief outline of the calculation sequence will be given below:

MAIN (the main program)

1. Opens inp.. files:

DATANA:	data to be used in the analytic setting of shift function, ellipticity, triangularity, paramagnetic function, temperatures and densities
INDAT1:	general input data such as grid density, running options, length of time interval etc.
PLODAT:	data for defining plot device, frame size and location
RUNDAT:	start data for separate orbit treatment
STATDA:	source exponent, subdivision of pitch-angle and ordinary space

2. Calls for

INREAD:	reads data from input file INDAT1
INIT:	initiates rho and chi values of grid points
ANREAD:	reads data from input file DATANA
ANALYT:	sets the analytical version of shift function, ellipticity, triangularity, paramagnetic function, temperatures and densities
LAOHIR:	sets the Lao-Hirsman representation of the magnetic surfaces
METRIC:	calculates length, area and volume elements of the Lao-Hirsman representation. Inactive at current stage.
B_FIEL:	sets the magnetic fields and field derivatives at grid points
SOURCE:	sets the reaction rates

3. Chooses physical mode, six alternatives of which three are dummies at current stage

alt1: RUN1MC:	running the constants of motion (c.o.m.) mode with generated start data for statistical treatment
alt2: RUN1GI:	running c.o.m. mode with given start data for separate orbit treatment (including plot option)
alt4: RUN2GI:	running direct integration mode with given start data for separate orbit treatment (including plot option)

3.3 Calculational sequence for some essential routines

RUN1GI

Produces c.o.m. trajectories for chosen number of particles

1. Reads data from RUNDAT: number of particles and laps

2. If orbits are to be plotted (DOPLOT=true)

```
For every particle
    Calls for
    STRTGI:   Start routine. Reads data from RUNDAT
              for initial values of pitch-angles, par-
              ticle energies and positions. Calculates
              magnetic moment and canonical mo-
              mentum. Initiates some variables to be
              used in ADVCOM

    PENMOV:   Initiates the plot tool. Reads data from
              PLODAT for frame size, scale of axis and
              plot device. Plots the outer plasma con-
              tour. Moves pen to initial positions.

        As long as the particle is not lost or number of
        laps fulfilled
            Calls for
            LAPNUM: Increases number of laps
            by one when the particle passes equa-
            torial line

            ADVCOM:   Advances the particle one
                      step to next position where
                      c.o.m. are unchanged

            PLOT:     Draws a line to the position
                      calculated above

            PERTUR:   Routine for disturbance of
                      the particle, i.e. a collision
                      simulator. Dummy at cur-
                      rent stage
```

3. If orbits are not to be plotted (DOPLOT=false)

```
For every particle
    Calls for
    STRTGI
        As long as the particle is not lost or the num-
        ber of laps fulfilled
            Calls for
            LAPNUM

            ADVCOM

            PERTUR
```

4. Return

RUN1MC

The midplane is divided into segments whose size depends on the chosen IRNUMB and on the distance from the centre. In each segment IPNUMB particles are generated based on a subdivision of pitch-angle space in equally sized segments. Each of these particles generate orbits that corresponds to a number of particles because of the inherent routine for particle reduction. The corresponding number of lost particles are calculated.

```
1. Reads data from STATDA, IRNUMB, IPNUMB and BORDER

2. The radial distance to the cross-section centre for the first
   point of particle production is calculated

3.If the radial distance to the cross-section centre for the point of
  particle production is greater than BORDER

        For every segment in pitch-angle space and initial position
        to the right of the cross-section centre

                Calculates pitch-angle

                Calls for
                STRTMC:    Start routine. Gives the particle energy.
                           Calculates magnetic moment and cano-
                           nical momentum. Initiates some va-
                           riables to be used in ADVCOM

                As long as the simulated particle has not passed the
                midplane for a second time and is not lost

                        Increases the total sum of particles in accor-
                        dance with the reduction routine

                        Calls for
                        ADVCOM:    Advances the particle one step to
                                   next position where c.o.m. are
                                   unchanged

                If the particles corresponding to the current trajec-
                tory are lost

                        Increase the total sum of lost particles with
                        the current sum

        For every segment in pitch-angle space and initial position
        to the left of the cross-section centre

                As above...

        Decreases the radial distance to the cross-section centre for
        the point of particle production by one segment whose size
        is determined by the function PTDENS and IRNUMB

4. Output

5. Return
```

ADVCOM

1.Current values of poloidal, toroidal and total magnetic field are calculated, as are the Jacobian, metric tensor, position and the chi derivative of total magnetic field.

2. The advancement step is calculated as

   2 a. Step parallel to the flux surface

   2 b. Step corresponding to drift motion

   2 c. These two steps are added to a total step

   2 d. A new step which is perpendicular and proportional to the total step is calculated

   2 e. Interpolation between total and perpendicular step gives the point where canonical angular momentum is conserved

3. Output of the new position

3.4 Description of essential routines

ADVCOM

See flowchart

ANALYT

Called in: MAIN

Calculates and sets into arrays the analytical values of shift-function, ellipticity, triangularity, paramagnetic function, poloidal flux function, temperatures and particle densities at the grid points.

ANREAD

Called in: MAIN

Reads data from file DATANA to be used in the analytical versions of shift function, ellipticity, triangularity, paramagnetic function, poloidal flux function, temperatures and particle densities.

B_FIEL

Called in: MAIN

Calculates and sets into matrixes the poloidal, toroidal and total magnetic field as well as their space derivatives, based on the paramagnetic function and poloidal flux function.

INIT

Called in: MAIN

Initiates the coordinate values of the grid points and the grid point density.

INREAD

Called in: MAIN

Reads data from input file INDAT1

LAOHIR

Called in: MAIN

Sets the Lao-Hirsman representation of magnetic flux surfaces and the grid matrixes for the Jacobian and the components of the metric tensor.

$$R(\rho,\chi) = R_o(\rho) + a\rho \cos\chi + R_2(\rho) \cos 2\chi$$

$$Z(\rho,\chi) = E(\rho) [a\rho \sin\chi - R_2(\rho) \sin 2\chi]$$

$R_o(\rho)$ = shift function, $E(\rho)$ = ellipticity, $R_2(\rho)$ = triangularity

ρ = flux surface label $0 \leq \rho \leq 1$

LAPNUM

Called in: RUN1GI

Increases the number of laps by one when the particle passes the equatorial line

PENMOV

Called in: RUN1GI

Initiates the plot tool. Reads data from PLODAT for frame size, scale of axes and plot device. Plots the outer plasma contour. Moves pen to initial positions.

PLOT

Called in: RUN1GI

Draws a line corresponding to the step calculated in ADVCOM

RUN1GI

See flowchart

RUN1MC

See flowchart

SOURCE

Called in: MAIN

Sets the reaction rate at the grid points either as a function of ion temperature and ion densities or as an exponential expression with the source exponent SOUR_Q (STATDA).

STRTGI

Called in: RUN1GI

Start routine. Reads data from RUNDAT for initiating values of pitch-angles, particle energies and positions. Calculates magnetic and canonical momentums. Initiates some variables to be used in ADVCOM

STRTMC

Called in: RUN1MC

Start routine. Gives the particle energy. Calculates magnetic and canonical momentums. Initiates some variables to be used in ADVCOM

3.5 Description of essential functions

INPOSU(..see ex. in ADVCOM..)

Used to calculate the magnetic field, Jacobian, metric tensor components etc. Interpolates the four values at the grid points surrounding the current position.

P_ANMO(ρ,χ)

Calculates current canonical angular momentum

P_FLUX(rho)

Interpolates the values of poloidal magnetic flux at the nearest grid points to a value at the current position

V_PAR(B_{tot})

Calculates the particle velocity parallel to the magnetic field lines

3.6 Common blocks

Variable entities to be used globally in the code are declared in six common blocks. The entities, both scalar variables and arrays, are declared LOGICAL, INTEGER or DOUBLE PRECISION in the Fortran sense.

COMANA

Included in: MAIN, ANREAD, ANALYT, LAOHIR, RUN1MC, ADVCOM

Common block for analytical parameters. The parameter values are stored in input data file DATANA and read in routine ANREAD.

COMGEO

Included in: All essential routines and functions except LAPNUM

Common block for parameter entities used in the representation of the flux surface geometry.

COMIN

Included in: MAIN, SOURCE, INREAD, RUN1GI, RUN1MC, ADVCOM, STRTGI, STRTMC, PENMOV, P_ANMO, V_PAR

Common block for parameters used as option determinants or in the numerical simulation

COMPLAS

Included in: MAIN, ANREAD, ANALYT, B_FIEL, SOURCE, RUN1MC, ADVCOM, STRTGI, STRTMC, P_ANMO, V_PAR, P_FLUX

Common block for parameters used in the representation of magnetic fields and associated functions as for temperatures, particle densities and reaction rates

COMRUN

Included in: MAIN, SOURCE, INREAD, RUN1GI, RUN1MC, ADVCOM, STRTGI, STRTMC, PENMOV, P_ANMO, V_PAR

Common block for <u>variables</u> whose values change or are set during the simulation. The exceptions are EPS and EPS_I, parameters used to avoid numerical singularities.

PARAMET

Included in: All essential routines and functions

Parameter settings that are unchanged from one simulation to another.

3.7 Included routines

The plotting routines in the CASINO code are included HGG-routines:

ENVBI(' ')

Initiates the HGG plotting system

AFRAM(AXMIN,AYMIN,AXMAX,AYMAX)

Defines the device related scale

RFRAM(XMIN,YMIN,XMAX,YMAX)

Defines the user related scale

PLDEV(IDEV)

Chooses plot device (IDEV = device number)

RPLOT(R,Z,IPEN)

The plotting routine. Coordinates in argument are in user related scale

PLHOME

Resets plotter. Necessary for certain older plot devices

3.8 Computer requirements

Current code version is adapted for computer VAX VMS V4.7 and fortran version VAX FORTRAN V4.8

3.9 Execution times

The time for execution of the code can be divided into two separate parts. The first part, the time necessary to initiate the field matrixes etc., is constant for a constant number of grid points. As an example, if the number of grid points is 96×96 this part will be of the order of 50 cpu-sec.

The second part, the time necessary to calculate the orbits, is close to proportional to IRNUMB and IPNUMB. For example, when IRNUMB=IPNUMB=60 and $\Delta t = 10^{-7}$s the total execution time are of the order of 1100 cpu-sec.

4. INPUT AND OUTPUT DESCRIPTIONS

4.1 Input files:

DATANA

Input file for data to be used in the analytic setting of shift function, ellipticity, triangularity, paramagnetic function, temperatures and densities

RMIN

The minor radius of the Tokamak

SHIFT0 SHIFT1 SHIFT2

The shift function is given as a polynomial at the grid points

$$SHIFT = SHIFT0 + \rho \, SHIFT1 + \rho^2 \, SHIFT2$$

SHIFT0 must have the same value as R0 (see below) since in practice they are the same entity

ELLIP0 ELLIP1 ELLIP2

The ellipticity is given as a polynomial at the grid points

$$ELLIPT = ELLIP0 + \rho \, ELLIP1 + \rho^2 \, ELLIP2$$

TRIAN0 TRIAN1 TRIAN2

The triangularity is given as a polynomial at the grid points. NOTE! TRIAN0 must always equal zero.

$$TRIANG = TRIAN0 + \rho \, TRIAN1 + \rho^2 \, TRIAN2$$

B_TOVA R0

The paramagnetic function is defined at the grid points as

$$\mathrm{PARMAF} = \mathrm{B\_TOVA\,R0}\,(1 + \rho\,\mathrm{BTOPL1} + \rho^2\,\mathrm{BTOPL2})$$

BTOPL1 BTOPL2

See above

POFLU2 POFLU3 POFLU4

The poloidal flux function is defined at the grid points as

$$\mathrm{POFLUX} = \rho^2\,(\mathrm{POFLU2} + \rho\,\mathrm{POFLU3} + \rho^2\,\mathrm{POFLU4})$$

TI0 TE0 ND0 NT0 NE0

See below.

TI_EXP TE_EXP ND_EXP NT_EXP NE_EXP

TEMP_I and TEMP_E are the ion and electron temperatures. They are given at the grid points as
$\mathrm{TEMP\_I} = \mathrm{TI0}\,(1 - \rho^2)^{\mathrm{TI\_EXP}}$
$\mathrm{TEMP\_E} = \mathrm{TE0}\,(1 - \rho^2)^{\mathrm{TE\_EXP}}$

ND, NT and NE are the deuterium, tritium and electron densities. They are defined at grid points as
$\mathrm{ND} = \mathrm{ND0}\,(1 - \rho^2)^{\mathrm{ND\_EXP}}$
$\mathrm{NT} = \mathrm{NT0}\,(1 - \rho^2)^{\mathrm{NT\_EXP}}$
$\mathrm{NE} = \mathrm{NE0}\,(1 - \rho^2)^{\mathrm{NE\_EXP}}$

INDAT1

Input file for general input data such as grid density, running options, length of time interval etc.

NRHO	**NCHI**
	Number of grid points in rho and chi directions. Ought to be at least 50x50
PHYMOD	(1, 2 or 3)
	Choice of simulation option, 1= c.o.m., 2= direct integration of guiding centre, 3= full orbit integration (inactive at current stage). Option in main program MAIN.
DOPLOT	(true or false)
	Orbits simulated in the "NOT MC" mode are plotted if DOPLOT = true. Option in routine RUN1GI.
ANARUN	(true or false)
	Geometry and fields are given analytically when ANARUN = true. True is the only option since EXPVER is a dummy routine at current stage. Option in main program MAIN.
MC	(true or false)
	MC=true, mode with generated start data for statistical treatment. MC=false, mode with given start data for separate orbit treatment.

DEL_T

The time interval in seconds between numeric steps in the orbit simulation.Typical size: 10^{-7} - 10^{-8} s

Z_F MASS_F

Charge number and mass [kg] for the particle being followed.

EPS

The minimum distance to the cross-section centre allowed for a simulated particle. It is also used as a limit in the division of the radius used in the particle generation in routine RUN1MC. <u>Must be greater than EPS I</u>. Typical size: 0.001

EPS_I

Used to prevent a singularity in the centre of the grid system (RHO(0)=EPS_I*DRHO). Typical size: 10^{-10}

CTEST

The ratio of the interpolation step to the total advance step as calculated in routine ADVCOM If this ratio is too large the interpolation process may fail. Typical size: 0.2

PLODAT

Input file for data initiating the plot system.

AXMIN AYMIN AXMAX AYMAX

Defines the physical frame. The terminal screen is defined AXMIN=0, AXMAX=4095, AYMIN=0, AYMAX=3119. Thus

$0 \leq \text{AXMIN} \leq \text{AXMAX} \leq 4095$

$0 \leq \text{AYMIN} \leq \text{AYMAX} \leq 3119$

If a frame (LFRAME) is desired the limits must be set to

$95 \leq \text{AXMIN}$

$95 \leq \text{AYMIN}$

XMIN YMIN XMAX YMAX

The limits of the user defined frame. To get the right proportions we must have (XMAX-XMIN)/(AXMAX-AXMIN) = (YMAX-YMIN)/(AYMAX-AYMIN)

IDEV

(1..9)

Used to identify the plot device: 2=VT 100 screen, 6=HP-plotter, etc.

RUNDAT

Input file for initial data in separate orbit mode, that is MC=false.

PANUMB

The number of particles to simulate.

LAPS

The number of passings of the midplane for each simulated particle

The data below are repeated for every simulated particle, i.e. the sequence is repeated PANUMB times.

PANGEL

The "pitch-angle" of the particle. NOTE! in CASINO pitch-angle is defined as

$$P_A = \frac{v_{||}}{v} = \cos\varphi$$

ENERGY

The particle energy.

P_RHO **P_CHI**

The initial position.

STATDA

Input file for data used in establishing an exponential expression of the production rate and for data used when option MC=true is chosen.

SOUR_Q

The source exponent used in the expression of the production rate at grid points as defined in routine SOURCE.

IRNUMB

The midplane is divided into segments whose size depends on the chosen IRNUMB and on the distance from the centre. In each segment IPNUMB particles are generated based on a subdivision of pitch-angle space in equally sized segments.Each of these particles generate orbits that correspond to a number of particles.because of the inherent routine for particle reduction. The corresponding number of lost particles is calculated.

IPNUMB

See above.

BORDER

Denotes the limit in the division of the radius used in the particle generation in routine RUN1MC.

4.2 Output files:

Output of simulation result data and running diagnostics are done by Fortran data files named for024, for025 and for031. If option DOPLOT = true is used, the results are plotted on chosen device and indications for an orbit crossing the vessel wall, "LOST PARTICLE", and when an orbit simulation is stopped because of too many step calculations, " I=10000", is written on the terminal.

for024

Used for diagnostics. Non-active for the moment (c-marked).

for025

Used as output file if MC=true, i.e. in routine RUN1MC.

NRHO NCHI

Control echo of the number of grid points in rho and chi direction. Ought to be at least 50x50

DEL_T

Control echo of the time interval in seconds between numeric steps in the orbit simulation. Typical size: 10^{-7} - 10^{-8} s

BORDER

Used as limit in the division of the radius used in the particle generation in routine RUN1MC.

IRNUMB IPNUMB

The midplane is divided into segments whose size depends on the chosen IRNUMB and on the distance from the centre. In each segment IPNUMB particles are generated based on a subdivision of pitch-angle space in equally sized segments.Each of these particles generate orbits that corresponds to a number of particles because of the inherent routine for particle reduction. The corresponding number of lost particles is calculated.

ANTAL

The total number of segments in the division of the radius. An approximate value of ANTAL is

ANTAL = 1.6 IRNUMB

NOTE! Total number of simulated orbits =

2 (IPNUMB · ANTAL)

NUMBER OF STEP-CALCULATIONS OVER 2000

Diagnostics. A few percent of ANTAL is acceptable but a greater number may be an indication of malfunction.

SNUMAX SNUSUM SNUAVE

Diagnostics. SNUMAX=maximum number of step calculations at one orbit simulation, SNUSUM = total number of step calculation for all orbit simulations, SNUAVE=SNUSUM/total number of simulated orbits.

TOTSUM/TEST1

Diagnostics. The ratio ought to be 1.00 otherwise the statistics are too uncertain and IRNUMB/IPNUMB and/or the number of grid points should be increased or DEL_T decreased. The method of particle reduction inherent in RUN1MC reduces the necessary amount of particle simulations TOTSUM is the total number of particles that corresponds to the simulations. TEST1 is the production rate integrated over the cross-section.

TOTLOS

The total number of lost particles.

SOUR_Q

The source exponent used in the expression of the production rate at grid points as defined in routine SOURCE.

ASPECT

The aspect ratio.

TOTLOS/TEST1

The total fraction of lost particles. Currently the most important output.

CURRENT2

The total current calculated by taking the line integral of the poloidal magnetic field along the outermost flux surface. The current is not given explicitly in the input but implicitly by the established poloidal flux.

for031

Used as output file if MC=false, i.e. in routine RUN1GI.

LOSSMX(J)

Loss matrix. The number of lost particles in "chi-area" J, where J goes between 1 and NCHIMA.

5. POSSIBLE FUTURE EXTENSIONS

In its present form the code is applicable to collisionless particles moving in axisymmetric Tokamak magnetic fields. A number of extensions of the code are possible which would further increase its usefulness. In particular, including the effects of (i) Coulomb collisions and (ii) toroidal magnetic field ripple should prove very valuable.

(i) Coulomb collisions

Even though the bounce time is much smaller than the collision time, collisional effects may still play an important role e.g. for the confinement of high energy charged particles. An example of this is the collisional pitch angle scattering, which tend to scatter particles into the loss regions in velocity space corresponding to the prompt losses occuring on the bounce time scale.

(ii) Toroidal magnetic field ripple

When the discrete character of the toroidal magnetic field coils in a Tokamak is taken into account, the magnetic field strength will contain an additional contribution

$$\frac{\Delta B}{B} \approx \delta \cos N\varphi$$

where N is the number of field coils and δ denotes the strength of the toroidal magnetic field ripple. Since the total magnetic field is no longer axisymmetric the longitudinal constant of motion is no longer conserved. Even though the strength of the ripple in general is only of the order of a few percent, it may never the less give rise to important effects, e.g. to strongly enhanced diffusion of banana trapped high energy particles.

Including the effects (i) and (ii) should be a rather straight forward procedure. Coulomb collisions could be included by adding to the code a prescription for small random perturbations of the pitch angle and the energy of the particle. The effect of magnetic field ripple can also be modelled as a collision process where the particle slightly changes its properties during the "collision" with the small localized field ripple.

ACKNOWLEDGEMENTS

My warmest thanks to my supervisor, Håkan Hamnén, for his stimulating guidance, encouraging support and generous help in developing the CASINO code.

I also want to thank Hans-Georg Gustavsson for his support in handling the computers.

TEST CASE

INPUT FILES:

INDAT1.DAT

```
96  96        [NRHO, NCHI]
1             [PHYMOD, 1=const. of motion, 2=guiding centre, 3=full orbit]
TRUE          [DOPLOT (plot the trajec. inst. of statist. treatm.]
TRUE          [ANARUN (run with analyt. versions of geom. and phys.)]
TRUE          [MC (monte-carlo, machinprod. start-data for particle)]
1.0D-7        [DEL_T (timeint. betw. perturb. as numb. of whole gyr.)]
2  6.65D-27    [Z_F, MASS_F (chargenumb. and mass in u for foll. part.)]
1.D-3         [EPS]
1.D-10        [EPS_I]
2.D-1         [CTEST]
1             [VERS (version)]
```

DATANA.DAT

```
1.0D0                                 [RMIN]
3.0D0   0.0D0   0.00D0                [SHIFT0, SHIFT1, SHIFT2]
1.D0    0.0D0    0.0D0                [ELLIP0, ELLIP1, ELLIP2]
0.0D0   0.0D0   0.0D0                [TRIAN0, TRIAN1, TRIAN2]
3.4D0   3.00                         [B_TOVA, R0]
4.0D-4 5.0D-6                         [BTOPL1, BTOPL2]
1.4153D0   0.D0 -0.3538D0                 [POFLU2, POFLU3, POFLU4]
15.D0  15.D0   1.D20 1.D20  2.D20 [TI0, TE0( temp in keV), ND0, NT0, NE0]
1.D0   1.D0    1.D0  1.D0   1.D0  [TI_EXP, TE_EXP, ND_EXP, NT_EXP, NE_EXP]
```

STATDA.DAT

```
2.0         SOUR_Q
60          IRNUMB
60          IPNUMB
0.D0        BORDER
```

OUTPUT FILE:

FOR025.DAT

```
 NRHO=             96   NCHI=           96
 DEL_T=   1.0000000000000000E-07
 BORDER=   9.9999999999999999E-04
 BRHO= -6.6597842147084218E-03
 IRNUMB=            60 IPNUMB=           60
 ANTAL=            94
 ANTAL OVER 2000=            0
 SNUMAX=           126 SNUSUM=       245271SNUAVE=           21

 TOTSUM/TEST1=    1.003495201853494
 TOTLOS=    3.284753205530915

 SOUR_Q=    2.000000000000000
 ASPECT=    3.000000000000000

 ______________________________________________
 TOTLOS/TEST1=    0.1665083406272450
 CURRENT2=     2.500040099902547      MA
 ______________________________________________
```