LRP 800/05 February 2005

A software package to manipulate space dependencies and geometry in magnetic confinement fusion

J.-M. Moret

submitted for publication in
Rev. Sci. Instrum.

ISSN 0458-5895

A software package to manipulate space dependencies and geometry in magnetic confinement fusion

J.-M. Moret

CRPP - EPFL

Centre de Recherches en Physique des Plasmas

Association EURATOM - Confédération Suisse

École Polytechnique Fédérale de Lausanne

CH-1015, Lausanne, Switzerland

Improvement in the performances of magnetic confinement devices for nuclear fusion counts on the optimisation of the geometry of the plasma, either the 2-D cross-section shape in tokamaks with toroidal symmetry or the 3-D magnetic configuration in stellerators. The variation in time and space of the plasma parameters in these devices are measured using tomographic or imaging systems with a large number of detectors. To federate the geometrical manipulations implied in the analysis of experimental data, the description of the confining magnetic field configuration and the modelling and simulation of the physical processes inside the plasma, an object oriented software package has been developed. Classes in this package are used to describe in several coordinate systems, including magnetic flux coordinate, the geometry of the measurement systems, the configuration of the magnetic field and space and time dependant functions representing plasma parameters. Methods applying on these classes then easily implement coordinate system transformations, interpolation of and integro-differential calculus on space and time dependant functions, geometrical description and characteristics of the magnetic flux surfaces and intersection of measurement viewing lines with a coordinate mesh and with flux surfaces and calculation of the corresponding transfer matrix used in tomographic inversion. The selected numerical methods used in these manipulations and their performances are also presented.

1. Introduction

In a thermonuclear reactor, the minimum requirement that the energy loss from the confined plasma should be smaller than the energy produced by fusion reactions yields a condition on the confinement properties of the plasma. In the field of magnetic confinement

fusion research, extrapolations to future demonstration devices suggest that highly elongated and shaped plasmas are potentially advantageous.

The distinctive features of TCV (Tokamak à Configuration Variable) are primarily a vacuum vessel and poloidal magnetic field coil system permitting high elongations of up to 3 and an extreme flexibility in the plasma shape. The machine therefore offers the unique capability to improve understanding of the stability and confinement properties in highly shaped plasmas [1].

Investigation of these plasmas and specifically that of the spatial variation of the plasma parameters is based on measurements from multi-chord systems, tomography of the light emission from the plasma in various energy ranges as well as 2-D viewing systems. To capture all achievable shapes, all these instruments provide an increased coverage of the plasma volume compared to fixed shape devices with a large number of lines of sight for most of the systems. A non exhaustive list of measurements equipping TCV is given in Table 1 together with the size of the system

Category	System / Detectors	Number of cameras	Number of chords	Number of energy channels	Sampling rate
Multi-chord	Thomson scattering	*25*	*1*	*4*	*17Hz*
	Interferometer		*14*		*100kHz*
	ECE (Electron cyclotron emission) LFS (Low field side)		*1*	*32*	*200kHz*
	ECE HFS (High field side)		*1*	*32*	*200kHz*
	Hard X-ray CdTe diode arrays		*14*	*8*	*500Hz*
	X-ray multi wire gas detectors	*1*	*64*	*2*	*200kHz*

Table 1: List and size of the TCV measurement systems

Category	System / Detectors	Number of cameras	Number of chords	Number of energy channels	Sampling rate
Tomography	Soft X-ray Si diode arrays	*10*	*200*		*10kHz*
	Bolometers, metal foil detectors	*5*	*64*		*2kHz*
	Bolometers, AXUV diode arrays	*7*	*140*		*100kHz*
	Lyman-α, filtered AXUV diode arrays	*7*	*140*		*100kHz*
Imaging	Visible filtered CCD	*4*	*384 × 288*		*78Hz*
	IR thermographic cameras	*2*	*324 × 256*		*50Hz*

Table 1: List and size of the TCV measurement systems

The analysis of the data recorded by all these systems necessitates Abel or tomographic inversion, combination of the data from several systems, inputs to and comparisons with simulation codes, etc. It is then clear that this analysis must be based on a common way to manipulate their geometry and that of the confining magnetic field of the device on which the plasma parameters closely depend. Moreover the magnetic field configuration largely varies in time during a single experiment and this must be coupled to the time dependence of each measurement.

To federate these geometrical manipulations, a software package has been developed. This package uses object oriented programming so that its usage requires a limited knowledge from the users to get access to very powerful calculations and manipulations. Four basic concepts implemented in four object classes are defined:

- Description of the geometry of a measurement system (Section 2).
- Coordinate system in which the geometry of the equipment, of the magnetic field and of the plasma parameters is referred to (Section 4). This includes usual cartesian, cylindrical and toroidal coordinates plus the magnetic flux coordinates (Section 3).
- Space dependant functions (Section 5).
- Representation of the magnetic field with flux functions.

Then manipulations on the geometry and on space dependant functions are packed in methods applying on these objects. Here are some examples:

- Coordinate system conversion (Section 4).
- Interpolation of a space dependant function (Section 5).
- Conversion of a space dependant function on a new coordinate system (Section 5).
- Integro-differential calculus on space dependant functions, taking into account the metric of the coordinate system (Section 5).
- Intersection of measurement viewing lines with coordinate mesh and calculation of the corresponding transfer matrix (Section 6).
- Conversion of the representation of the magnetic field with flux functions to geometrical description of the flux surfaces (Section 6).
- Geometrical characteristics of the flux surfaces (Section 6).

Some details on the chosen implementation and the corresponding functionality, as well as the algorithms used and their performance are presented in Section 7. Section 8 illustrates the usage of the package with some typical applications. Although this package has up to now been applied only to axisymmetric geometries, it is ready to handle 3-D configurations.

2. Measurement system description

Implementation: class *dccl*

A large fraction of measurement systems for magnetic fusion devices [2] are based on the measurement of the emission by the plasma of light, waves or particles or on the propagation in the plasma of a light, wave or particle beam. The geometry of such a system is specified by viewing lines or chords crossing the plasma volume and the system measures the integral of the emissivity or of the propagation characteristics along these chords. In the package the chords are described by a reference point in cylindrical coordinates (Section 4) plus an elevation and an azimutal pointing angles $(R_d, Z_d, \varphi_d, \theta_d, \phi_d)$, as shown in Figure 1. These quintuplets can be represented as N-dimensional arrays of size $N_1 \times \ldots \times N_N$ to follow the organisation of the hardware (for example for a measurement instrument equipped with *5* systems of *10* detectors each, the size could be set to 10×5):

$$(R_d, Z_d, \varphi_d, \theta_d, \phi_d)_{l_1 \ldots l_N},\ l_l = 1 \ldots N_l,\ l = 1 \ldots N \qquad (1)$$

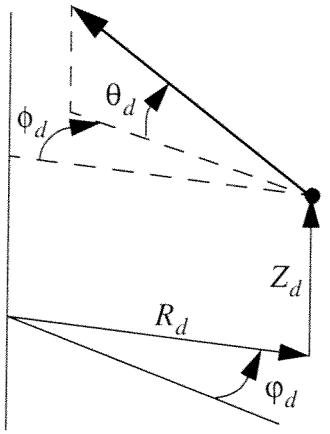

Fig. 1 *The parameters describing the geometry of a diagnostic chord.*

If the hardware moves in time, for example for scanning diagnostics, the dimensionality can be extended to include the discretised time:

$$(R_d, Z_d, \varphi_d, \theta_d, \phi_d)_{l_1 \ldots l_N l_t},\ l_l = 1 \ldots N_l,\ l = 1 \ldots N,\ l_t = 1 \ldots N_t \tag{2}$$

For measurement systems which do not enter this category, the measurement positions can be represented as discrete points in the most appropriate coordinate system.

3. Magnetic field description

Implementation: class *mfd*

Since in a magnetic fusion device, the plasma characteristics and the related physical processes are closely linked to the confining magnetic field, an adequate description of the magnetic field is necessary [3]. In an arbitrary coordinate system (ρ, θ, φ) the divergence-free magnetic field can be derived in its canonical form from two scalar functions χ and ψ:

$$\vec{B} = \nabla\chi \times \nabla\theta + \nabla\varphi \times \nabla\psi \tag{3}$$

where $\chi(\vec{r})$, $\psi(\vec{r})$, $\theta(\vec{r})$ and $\varphi(\vec{r})$ are continuous functions of space. If the configuration is such that the component of the magnetic field along the φ coordinate never vanishes in the volume under consideration, the functions $\chi(\vec{r})$, $\theta(\vec{r})$ and $\varphi(\vec{r})$ can be inverted, $\vec{r} = \vec{r}(\chi, \theta, \varphi)$. Then (χ, θ, φ) may be taken as a set of coordinates and the function ψ

expressed in terms of these new variables $\psi(\vec{r}) = \psi(\chi, \theta, \varphi)$. In the case of a toroidal geometry with a strong toroidal field, for which the previous positivity condition is satisfied, the system is periodic in the poloidal and toroidal directions thus θ and φ may be taken as the corresponding angle-like variables. Then the flux across a surface with constant φ is

$$\int \vec{B} \cdot \vec{ds}_{\varphi} = \int \frac{\vec{B} \cdot \nabla\varphi}{\nabla\chi \times \nabla\theta \cdot \nabla\varphi} d\chi d\theta = \int d\chi d\theta \tag{4}$$

where the Jacobian of the transformation $\vec{r} = \vec{r}(\chi, \theta, \varphi)$ and the expression (3) for the magnetic field have been used. Because of the 2π periodicity in θ, the toroidal flux enclosed by a surface with constant χ is $2\pi\chi$. This flux is generally an appropriate choice for the radial-like coordinate.

In the particular case of an axisymmetric tokamak configuration, the expression (3) for the magnetic field reduces to

$$\vec{B} = RB_{\varphi}\nabla\varphi + \nabla\varphi \times \nabla\psi \tag{5}$$

where R is the distance to the axis of symmetry and B_{φ} the toroidal magnetic strength. Due to the axisymmetry, $\vec{B} \cdot \nabla\psi = 0$, so that the magnetic field lines lie on surfaces with constant ψ. The poloidal flux between two neighboring flux surfaces ψ and $\psi + d\psi$ is

$$\oint \nabla\varphi \times \nabla\psi \cdot \vec{dl}_{\varphi} \times \vec{dl}_{\psi} = \oint \frac{|\nabla\varphi \times \nabla\psi|}{|\nabla\varphi||\nabla\psi|} d\varphi d\psi = 2\pi d\psi \tag{6}$$

Hence in a tokamak configuration $2\pi\psi$ is the poloidal flux function and it can also be used as the radial-like coordinate, providing that it is defined to be zero on the magnetic axis.

When the toroidal or poloidal flux is used to label the nested flux surfaces and serves as radial coordinate, it is appropriate for practical reasons to normalise it in such a way that it is zero on the inner most degenerated flux surface, the magnetic axis, and one on the last closed flux surface; this normalised flux will be symbolised by ψ^*.

4. Coordinate systems

Implementation: class *grid*

The geometry of the plasma, its confining magnetic field and its surrounding auxiliary systems will be referenced to a chosen coordinate system. To exploit any constraint or symmetry present in the problem and make it more readily soluble, several coordinate systems are supported by the package (Table 2 and Fig. 2), together with the methods to convert one system to the other. A further development of the package will also incorporate orthogonal flux coordinates (ρ, ζ, φ) in which the poloidal angle ζ is chosen such that magnetic field lines are straight lines in the (ζ, φ) planes with constant ρ.

Coordinate system	Description	Symbols
Cartesian	Orthogonal cartesian coordinate system	(X, Y, Z)
Cylindrical	Cylindrical coordinate system whose axis coincides with the main axis of the device	(R, φ, Z)
Toroidal	Toroidal coordinate system; the minor radius r is measured from the axis whose cylindrical coordinates are (R_0, Z_0)	(r, θ, φ)
Flux-coordinates	Flux coordinate system, ρ is the square-root of the normalised toroidal or poloidal flux ψ^*	(ρ, θ, φ)
Diagnostic-chord	One dimensional coordinate system to scan the chords of a diagnostic, s is the linear distance along the chord measured from the chord reference point (R_d, Z_d, φ_d)	(s)

Table 2: Coordinate systems

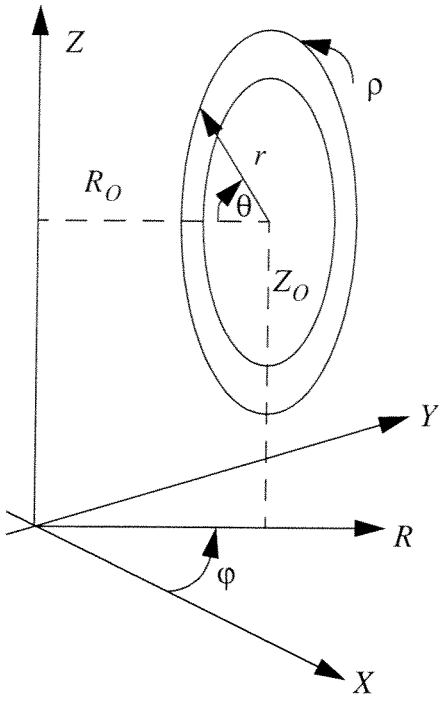

Fig. 2 Coordinate systems

4.1 Coordinate system discretisation

Location in space can be specified in different forms:

Discrete points. Coordinate triplets of a set of discrete points which can be casted in an N-dimensional arrays to follow the organisation of the hardware:

$$(q_1, q_2, q_3)_{l_1 \ldots l_N},\ l_l = 1 \ldots N_l,\ l = 1 \ldots N \tag{7}$$

Diagnostic-chord. In the one dimensional case of diagnostic-chord coordinate, a set of N_d discrete linear coordinates along each chord is specified

$$(s)_{l_d l_1 \ldots l_N},\ l_d = 1 \ldots N_d,\ l_l = 1 \ldots N_l,\ l = 1 \ldots N \tag{8}$$

Mesh points. The mesh points of fixed sorted grids along each coordinate:

$$(q_{1i}, q_{2j}, q_{3k}),\ i = 1 \ldots N_1, j = 1 \ldots N_2, k = 1 \ldots N_3 \tag{9}$$

4.2 Time dependence

To treat situations in which the flux surfaces change in time, the parameters defining the coordinate systems are also allowed to vary in time: this applies to the axis of a toroidal coordinate system $(R_0(\varphi, t), Z_0(\varphi, t))$, to the flux function itself $\psi^*(R, Z, \varphi, t)$ and to the minor radius of the flux surfaces $r^*(\rho, \theta, \varphi, t)$ which enters system coordinate transformations (Section 4.4) and is derived from ψ^* as explained in Section 6.4. This feature may also be used if the hardware moves in time, for example for scanning diagnostics, by extending the dimensionality of the coordinate triplets:

$$(q_1, q_2, q_3)_{l_1 \ldots l_N l_t},\ l_l = 1 \ldots N_l,\ l = 1 \ldots N,\ l_t = 1 \ldots N_t \tag{10}$$

or that of the linear coordinates along each chord

$$(s)_{l_d l_1 \ldots l_N l_t},\ l_d = 1 \ldots N_d,\ l_l = 1 \ldots N_l,\ l = 1 \ldots N,\ l_t = 1 \ldots N_t \tag{11}$$

4.3 Reduced and extended dimensionality

In some specific situations, the quantities to manipulate expressed in a proper coordinate system do not depend on all coordinates: for example in a tokamak, which is axisymmetric, most parameters are constant along the φ coordinate and thus depends only on (R, Z) or (r, θ) or (ρ, θ); sometimes the physical laws also impose such simplifications, for example in an axisymmetric equilibrium the plasma pressure is constant on a flux surface, thus expressed in a flux-coordinate system, it depends only on ρ but neither on θ nor on φ. The package is capable of handling reduced 0-D, 1-D or 2-D coordinate systems for such situations.

The quantities to manipulate may depend not only on the position, but also on some additional variables: for example an instrument may have the capability to measure at the same point in space the emissivity of the plasma at different photon energies. The package can then extend the basic space variable list and add additional variables, yielding in an multi-dimensional definition domain $(q_1, q_2, q_3, v_1, \ldots, v_M, t)$ either for discrete points

$$\begin{aligned} &(q_1, q_2, q_3, v_1, \ldots, v_M)_{l_1 \ldots l_N m_1 \ldots m_M l_t}, \\ &l_l = 1 \ldots N_l,\ l = 1 \ldots N, \\ &m_m = 1 \ldots M_m,\ m = 1 \ldots M, \\ &l_t = 1 \ldots N_t \end{aligned} \tag{12}$$

or for diagnostic-chords

$$
\begin{aligned}
&(s, v_1, \ldots, v_M)_{l_d l_1 \ldots l_N m_1 \ldots m_M l_t}, \\
&l_d = 1 \ldots N_d, \\
&l_l = 1 \ldots N_l, l = 1 \ldots N, \\
&m_m = 1 \ldots M_m, m = 1 \ldots M, \\
&l_t = 1 \ldots N_t
\end{aligned}
\tag{13}
$$

or for a mesh

$$
\begin{aligned}
&(q_{1i}, q_{2j}, q_{3k}, v_{1m_1}, \ldots, v_{Mm_M}), \\
&i = 1 \ldots N_1, j = 1 \ldots N_2, k = 1 \ldots N_3, \\
&m_m = 1 \ldots M_m, m = 1 \ldots M
\end{aligned}
\tag{14}
$$

4.4 Coordinate system conversion

Implementation: method *g2g* of class *grid*

Coordinate transformations from one system to the other are based on the formulas given in Table 3. Conversion to or from the flux-coordinate system requires the knowledge of the flux function in cylindrical coordinates $\psi^*(R, Z, \varphi)$ or the minor radius of the flux surface in flux coordinates $r^*(\rho, \theta, \varphi)$ respectively. Usually these functions are tabulated on a given grid and conversion is done using the interpolation method described in Section 5.1.

Transformation	Formulas
Cartesian to cylindrical	$R^2 = X^2 + Y^2$ $\tan\varphi = Y/X$ $Z = Z$ (15)
Cylindrical to cartesian	$X = R\cos\varphi$ $Y = R\sin\varphi$ $Z = Z$ (16)
Cylindrical to toroidal	$r^2 = (R_0 - R)^2 + (Z - Z_0)^2$ $\tan\theta = (Z - Z_0)/(R_0 - R)$ $\varphi = \varphi$ (17)

Table 3: Coordinate transformation formulas

Transformation	Formulas
Toroidal to cylindrical	$R = R_0 - r\cos\theta$ $Z = Z_0 + r\sin\theta$ (18) $\varphi = \varphi$
Cylindrical to flux coordinate	$\rho^2 = \psi^*(R, Z, \varphi)$ $\tan\theta = (Z - Z_{\psi^*=0})/(R_{\psi^*=0} - R)$ (19) $\varphi = \varphi$
Flux-coordinate to toroidal	$r = r^*(\rho, \theta, \varphi)$ $\theta = \theta$ (20) $\varphi = \varphi$
Diagnostic-chord to cylindrical	$R^2 = (R_d - s\cos\theta_d\cos\phi_d)^2 + (s\cos\theta_d\sin\phi_d)^2$ $Z = Z_d + s\sin\theta_d$ (21) $\tan(\varphi - \varphi_d) = (s\cos\theta_d\sin\phi_d)/(R_d - s\cos\theta_d\cos\phi_d)$

Table 3: Coordinate transformation formulas

Not all possible conversions are defined but a series of successive conversions can be used to carry out any conversion (Table 4). Conversion to diagnostic-chord coordinate is not possible since the chords are intrisicaly discrete and do not cover the whole space. It is replaced by intersection with surfaces of constant coordinate in a mesh (Section 6.1)

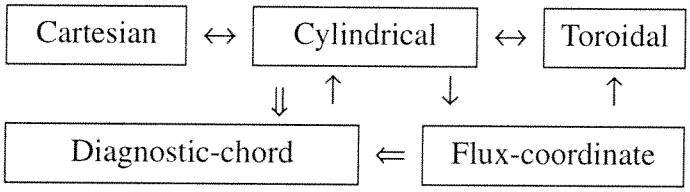

Table 4: Coordinate transformations (simple arrows) and intersections with surfaces of a mesh (double arrows)

5. Space dependant functions

Implementation: class *fct*

Functions of the space variables (q_1, q_2, q_3) and optionally of the additional variables $(v_1, \ldots, v_M)$ and/or the time are readily represented by their value on the discrete points

$$f((q_1, q_2, q_3, v_1, \ldots, v_M, t)_{l_1 \ldots l_N m_1 \ldots m_M l_t}) = f_{l_1 \ldots l_N m_1 \ldots m_M l_t} \quad (22)$$

or for a function defined along diagnostic chords

$$f((s, v_1, \ldots, v_M, t)_{l_d l_1 \ldots l_N m_1 \ldots m_M l_t}) = f_{l_d l_1 \ldots l_N m_1 \ldots m_M l_t} \tag{23}$$

or on the points of a mesh

$$f(q_{1i}, q_{2j}, q_{3k}, v_{1m_1}, \ldots, v_{Mm_M}, t_{l_t}) = f_{ijkm_1 \ldots m_M l_t} \tag{24}$$

Basic algebraic and mathematical operations and functions are applied to the discrete values $f_{l_1 \ldots l_N m_1 \ldots m_M l_t}$, $f_{l_d l_1 \ldots l_N m_1 \ldots m_M l_t}$ or $f_{ijkm_1 \ldots m_M l_t}$. Multi-evaluated functions can easily be represented by taking advantage of the capability to include an additional variable v to label each individual value.

5.1 Function interpolation

Implementation: method *f2f* of class *fct*

When mapping a function whose values are referenced to a given coordinate system to another coordinate system or to a different grid or set of points in the same coordinate system, the points in the target coordinate system may not coincide with those of the source system on which the function is specified. Thus it is necessary to interpolate the function between its definition points. Cubic spline interpolants (Appendix 9.1) are used to that purpose with an appropriate choice of the edge conditions depending on the coordinate: periodic for angular variables, symmetric for dependencies along a radial-like coordinate if applicable. This interpolation technique is only applicable to functions defined on a mesh, which can be decomposed as a linear combination of cubic spline base functions along each dimension

$$f(q_1, q_2, q_3, v_1, \ldots, v_M, t_{l_t}) = \tag{25}$$

$$\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \sum_{m_1=1}^{M_1} \cdots \sum_{m_M=1}^{M_M} c_{ijkm_1 \ldots m_M l_t} b_{1i}(q_1) b_{2j}(q_2) b_{3k}(q_3) b_{m_1}(v_1) \ldots b_{m_M}(v_M)$$

Here the base functions $(b_{1i}(q_1), b_{2j}(q_2), b_{3k}(q_3), b_{m_1}(v_1), \ldots, b_{m_M}(v_M))$ are specified by taking as knots the corresponding grid points $(q_{1i}, q_{2j}, q_{3k}, v_{1m_1}, \ldots, v_{Mm_M})$ and the coefficients $c_{ijkm_1 \ldots m_M l_t}$ satisfy the conditions

$$\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3}\sum_{m_1=1}^{M_1}\cdots\sum_{m_M=1}^{M_M} c_{ijkm_1\ldots m_M l_t} b_{1i}(q_{1i})b_{2j}(q_{2j})b_{3k}(q_{3k})b_{m_1}(v_{1m_1})\ldots b_{m_M}(v_{Mm_M}) = \quad (26)$$

$f_{i'j'k'm_1\ldots m_M l_t}$

5.2 Coordinate system metric

Implementation: method *metric* of class *grid*

For integro-differential calculus done in a specific coordinate system, it is necessary to know the metric of this coordinate system [4]. For the generic curvilinear to cartesian coordinate transformation written as

$$\begin{aligned} X &= X(q_1, q_2, q_3) \\ Y &= Y(q_1, q_2, q_3) \\ Z &= Z(q_1, q_2, q_3) \end{aligned} \quad (27)$$

the elements of the metric tensor g are defined by

$$g_{ij} = \frac{\partial X}{\partial q_i}\frac{\partial X}{\partial q_j} + \frac{\partial Y}{\partial q_i}\frac{\partial Y}{\partial q_j} + \frac{\partial Z}{\partial q_i}\frac{\partial Z}{\partial q_j} \quad (28)$$

from which can be derived the norm of the length elements

$$dl_i = \left|\vec{dl_i}\right| = \left|\frac{\partial X}{\partial q_i}\hat{e}_X + \frac{\partial Y}{\partial q_i}\hat{e}_Y + \frac{\partial Z}{\partial q_i}\hat{e}_Z\right| dq_i = \sqrt{g_{ii}}dq_i \quad (29)$$

that of the surface elements

$$ds_{ij} = \left|\vec{dl_i}\times\vec{dl_j}\right| = \left(dl_i^2 dl_j^2 - (\vec{dl_i}\cdot\vec{dl_j})^2\right)^{1/2} = \sqrt{g_{ii}g_{jj} - g_{ij}^2}dq_i dq_j \quad (30)$$

and the volume element

$$dV = [\vec{dl_1}, \vec{dl_2}, \vec{dl_3}] = \sqrt{det(g)}dq_1 dq_2 dq_3 \quad (31)$$

One can also perform differential vector operations on a function defined in a curvilinear coordinate system $f(q_1, q_2, q_3)$. For example the expression of the particle or energy flux associated with diffusive transport contains the gradient of the particle density or temperature, which if expressed in flux coordinates with arbitrary flux surface shapes has a non trivial expression whose geometrical component determines the confinement properties of the plasma [8]. In particular the expression for the norm of the coordinate gradients is

$$|\nabla q_i| = \sqrt{(g^{-1})_{ii}} \tag{32}$$

Cartesian, cylindrical and toroidal coordinate systems are orthogonal and their metric is diagonal and trivial. The flux-coordinate system is more complicated and the formulas are given in Table 5, all based on the radius of the flux surfaces $r^*(\rho, \theta, \varphi)$. These quantities are easily computed using the derivation capability of the package (Section 5.3) and represented as functions of space as introduced in Section 5

Metric	Formula
Metric tensor	$\begin{bmatrix} \left(\frac{\partial r^*}{\partial \rho}\right)^2 & \frac{\partial r^*}{\partial \rho}\frac{\partial r^*}{\partial \theta} & 0 \\ \frac{\partial r^*}{\partial \rho}\frac{\partial r^*}{\partial \theta} & r^{*2} + \left(\frac{\partial r^*}{\partial \theta}\right)^2 & 0 \\ 0 & 0 & (R_0 - r^* cos\theta)^2 \end{bmatrix}$ (33)
Length elements	$dl_\rho = \frac{\partial r^*}{\partial \rho} d\rho$ $dl_\theta = r^* \sqrt{1 + \left(\frac{\partial r^*}{r^* \partial \theta}\right)^2} d\theta$ (34) $dl_\varphi = (R_0 - r^* cos\theta) d\varphi$
Surface elements	$ds_\rho = ds_{\theta\varphi} = r^*(R_0 - r^* cos\theta)\sqrt{1 + \left(\frac{\partial r^*}{r^* \partial \theta}\right)^2} d\theta d\varphi$ $ds_\theta = ds_{\rho\varphi} = \frac{\partial r^*}{\partial \rho}(R_0 - r^* cos\theta) d\rho d\varphi$ (35) $ds_\varphi = ds_{\rho\theta} = r^* \frac{\partial r^*}{\partial \rho} d\rho d\theta$

Table 5: Flux-coordinate system metric

Metric	Formula
Volume element	$dV = r^* \frac{\partial r^*}{\partial \rho}(R_0 - r^* cos\theta) d\rho d\theta d\varphi$ (36)
Coordinate gradients	$\lvert\nabla\rho\rvert = \sqrt{1 + \left(\frac{\partial r^*}{r^* \partial\theta}\right)^2}\left(\frac{\partial r^*}{\partial \rho}\right)^{-1}$ $\lvert\nabla\theta\rvert = r^{*-1}$ $\lvert\nabla\varphi\rvert = (R_0 - r^* cos\theta)^{-1}$ (37)

Table 5: Flux-coordinate system metric

5.3 Function derivation

Implementation: method *der* of class *fct*

The same cubic spline decomposition can be easily used to calculate the partial derivatives of a function along its variables as presented in Appendix 9.1. The software package can also computes the components of the spatial gradient of a function using the metric of the coordinate system

$$\frac{\partial f}{\partial q_i} \nabla q_i \tag{38}$$

5.4 Function integral and average

Implementation: method *int* and *mean* of class *fct*

Simple definite and indefinite integrals of a function along its variables are easily computed based on its cubic spline decomposition as presented in Appendix 9.1. The package also offers the possibility to perform integrals on real integration length, surface or volume by taking into account the metric of the coordinate system, respectively

$$\int f dl_i, \int f ds_{ij}, \int f dV \tag{39}$$

. In the same manner it can computes line, surface of volume average of functions, respectively

$$\langle f \rangle_i = \frac{\int f dl_i}{\int dl_i}, \langle \langle f \rangle \rangle_{ij} = \frac{\int f ds_{ij}}{\int ds_{ij}}, \langle \langle \langle f \rangle \rangle \rangle = \frac{\int f dV}{\int dV} \tag{40}$$

6. Intersection with a mesh

6.1 Intersection of chords with a mesh

Implementation: method *inter* of class *dcd*

To compensate for the lack of conversion from any coordinate system to the coordinate along a diagnostic chord, the intersection of the chords with surfaces of constant coordinate in a mesh can be computed and used in many applications. For a given coordinate system (q_1, q_2, q_3) and a set of chord defined by eq. (2) the coordinate conversion of a point on the chord $l_1 \ldots l_N$ at time l_t whose linear coordinate is s is given by the functions

$$\begin{matrix} q_{1(l_1 \ldots l_N l_t)}(s) \\ q_{2(l_1 \ldots l_N l_t)}(s) \\ q_{3(l_1 \ldots l_N l_t)}(s) \end{matrix} \tag{41}$$

If the coordinate system is discretised on a mesh (eq. (7)), one can find if it exists the linear coordinate $(s_{1i}, s_{2j}, s_{3k})_{l_1 \ldots l_N l_t}$ of the points where

$$\begin{aligned} q_{1(l_1 \ldots l_N l_t)}(s_{1i(l_1 \ldots l_N l_t)}) &= q_{1i} \\ q_{2(l_1 \ldots l_N l_t)}(s_{2j(l_1 \ldots l_N l_t)}) &= q_{2j} \\ q_{3(l_1 \ldots l_N l_t)}(s_{3k(l_1 \ldots l_N l_t)}) &= q_{3k} \end{aligned} \tag{42}$$

If the inverse of the functions $q_{1(l_1 \ldots l_N l_t)}$, $q_{2(l_1 \ldots l_N l_t)}$ and $q_{3(l_1 \ldots l_N l_t)}$ exists and is analytical, the calculation is trivial, otherwise numerical solutions presented in Section 6.3 must be used. If the constant coordinate surfaces are not planes, there may be more than one intersection, see Figure 3. For the most common case of convex surfaces, as in the case of radial-like coordinates, two intersections must be considered determined by

$$\begin{aligned} q_{1(l_1 \ldots l_N l_t)}(s_{-1i(l_1 \ldots l_N l_t)}) &= q_{1(l_1 \ldots l_N l_t)}(s_{+1i(l_1 \ldots l_N l_t)}) = q_{1i} \\ q_{2(l_1 \ldots l_N l_t)}(s_{-2j(l_1 \ldots l_N l_t)}) &= q_{2(l_1 \ldots l_N l_t)}(s_{+2j(l_1 \ldots l_N l_t)}) = q_{2j} \\ q_{3(l_1 \ldots l_N l_t)}(s_{-3k(l_1 \ldots l_N l_t)}) &= q_{3(l_1 \ldots l_N l_t)}(s_{+3k(l_1 \ldots l_N l_t)}) = q_{3k} \end{aligned} \tag{43}$$

and ordered such that

$$
\begin{aligned}
s_{-1i(l_1 \ldots l_N l_t)} &\le s_{+1i(l_1 \ldots l_N l_t)} \\
s_{-2j(l_1 \ldots l_N l_t)} &\le s_{+2j(l_1 \ldots l_N l_t)} \\
s_{-3k(l_1 \ldots l_N l_t)} &\le s_{+3k(l_1 \ldots l_N l_t)}
\end{aligned}
\tag{44}
$$

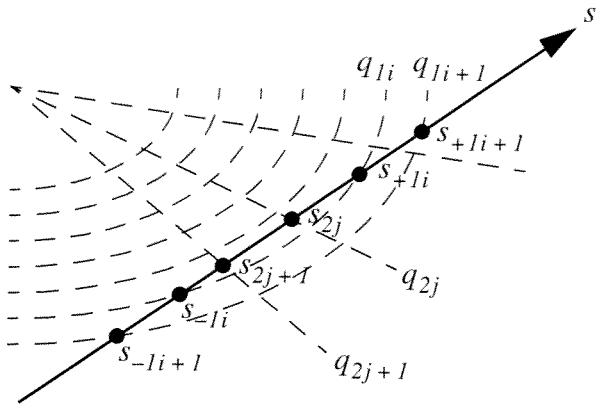

Fig. 3 *Intersection of a chord with constant coordinate surfaces.*

6.2 Transfer matrix in tomography

Implementation: method *tmat* of class *dcd*

A direct application of these intersections is the derivation of the transfer matrix T formed by the length of the part of the chord in one of the mesh cell, i.e. the volume $[q_{1i}, q_{1(i+1)}] \times [q_{2j}, q_{2(j+1)}] \times [q_{3k}, q_{3(k+1)}]$. An element of this matrix is computed using interval artithmetics as the total length of the combination of intervals

$$
\begin{aligned}
&T_{(ijk)(l_1 \ldots l_N l_t)} = \left\| I_{1i(l_1 \ldots l_N l_t)} \cap I_{2j(l_1 \ldots l_N l_t)} \cap I_{3k(l_1 \ldots l_N l_t)} \cap L_{(l_1 \ldots l_N l_t)} \right\|, \\
&i = 1, \ldots, N_1 - 1, j = 1, \ldots, N_2 - 1, k = 1, \ldots, N_3 - 1
\end{aligned}
\tag{45}
$$

For plane coordinate surfaces, individual intervals are given by

$$
\begin{aligned}
I_{1i(l_1 \ldots l_N l_t)} &= [s_{1i(l_1 \ldots l_N l_t)}, s_{1(i+1)(l_1 \ldots l_N l_t)}] \cup [s_{1(i+1)(l_1 \ldots l_N l_t)}, s_{1i(l_1 \ldots l_N l_t)}] \\
I_{2j(l_1 \ldots l_N l_t)} &= [s_{2j(l_1 \ldots l_N l_t)}, s_{2(j+1)(l_1 \ldots l_N l_t)}] \cup [s_{2(j+1)(l_1 \ldots l_N l_t)}, s_{2j(l_1 \ldots l_N l_t)}] \\
I_{3k(l_1 \ldots l_N l_t)} &= [s_{3k(l_1 \ldots l_N l_t)}, s_{3(k+1)(l_1 \ldots l_N l_t)}] \cup [s_{3(k+1)(l_1 \ldots l_N l_t)}, s_{3k(l_1 \ldots l_N l_t)}]
\end{aligned}
\tag{46}
$$

including the possibility that intersection coordinates are not sorted. For curved coordinate surfaces, the two intersections must be taken into account

$$
\begin{aligned}
I_{1i(l_1\ldots l_N l_t)} &= [s_{-1(i+1)(l_1\ldots l_N l_t)},\ s_{+1(i+1)(l_1\ldots l_N l_t)}] \,\Delta\, [s_{-1i(l_1\ldots l_N l_t)},\ s_{+1i(l_1\ldots l_N l_t)}] \\
I_{2j(l_1\ldots l_N l_t)} &= [s_{-2(j+1)(l_1\ldots l_N l_t)},\ s_{+2(j+1)(l_1\ldots l_N l_t)}] \,\Delta\, [s_{-2j(l_1\ldots l_N l_t)},\ s_{+2j(l_1\ldots l_N l_t)}] \\
I_{3k(l_1\ldots l_N l_t)} &= [s_{-3(k+1)(l_1\ldots l_N l_t)},\ s_{+3(k+1)(l_1\ldots l_N l_t)}] \,\Delta\, [s_{-3k(l_1\ldots l_N l_t)},\ s_{+3k(l_1\ldots l_N l_t)}]
\end{aligned}
\tag{47}
$$

where the operator Δ stands for the symmetric set difference. Provision is made to limit the view of the diagnostic in space by specifying a validity interval $L_{(l_1\ldots l_N l_t)}$ for each chord, for example if the chord view is restricted by some mechanical structure.

This transfer matrix is used when computing the integral along the chord of a function f_{ijkl_t} assumed to be constant in one mesh cell

$$
g_{(l_1\ldots l_N l_t)} = \sum_{i=1}^{N_1-1} \sum_{j=1}^{N_2-1} \sum_{k=1}^{N_3-1} T_{ijk(l_1\ldots l_N l_t)} f_{ijkl_t} \tag{48}
$$

and is a basic ingredient in tomographic inversion.

6.3 Intersection of chords with flux surfaces

Implementation: method *inter* of class *dcd*

The intersection of diagnostic chords with the coordinate surfaces of a flux coordinate system is of particular interest because the physics of plasma imposes that some plasma parameters are constant on a flux surface. To localise these intersections, the cylindrical coordinates of a point of linear coordinate s along the chord must be computed using eq. (21). Then according to eq. (19), the following equation must be solved with respect to s

$$
\psi^*(R(s), Z(s), \varphi(s)) = \psi^*(s) = \rho_i^2 \tag{49}
$$

for each flux surface, i.e. each value of ρ_i. The normalisation of the normalised flux function ψ^* (eq. (71)) is such that $\psi^*(s) > 1$ outside the plasma, $\psi^*(s) = 1$ on the last closed flux surface and $0 \le \psi^*(s) < 1$ inside the plasma. The exception of the divertor private region where also $\psi^* < 1$ can be treated by replacing ψ^* by $2 - \psi^*$ in this region. Thus the function $\psi^*(s)$ has a minimum value $\psi^*(s_0) = \psi_0^* = \rho_0^2$ between 0 and 1 inside the plasma for the chord coordinate s_0, referenced as the point of closest approach. This point can first be localised and then for each $\rho_i \ge \rho_0$ two solutions for eq. (49) can be found with $s_- \le s_0 \le s_+$.

The numerical method selected for these two steps is the Gauss-Newton iteration method, taking advantage of the fact that the derivatives of ψ^* can easily be computed as described in Section 5.3. At the point of closest approach, the following condition is fulfilled

$$\frac{d\psi^*}{ds}(s_0) = 0 \tag{50}$$

If the derivative of the function $\psi^*(s)$ is approximated by

$$\psi^{*'}(s + \Delta s) \cong \psi^{*'}(s) + \Delta s \psi^{*''}(s) \tag{51}$$

then the solution of eq. (50) can be found be iteratively replacing s_0 by

$$s_0 \leftarrow s_0 - \frac{\psi^{*'}(s_0)}{\psi^{*''}(s_0)} \tag{52}$$

To solve eq. (49), the function is approximated with its second order Taylor expansion

$$\psi^*(s + \Delta s) \cong \psi^*(s) + \Delta s \psi^{*'}(s) + \frac{\Delta s^2}{2} \psi^{*''}(s) \tag{53}$$

and the approximate solutions of $\psi^*(s + \Delta s) = \rho_i^2$ are

$$\Delta s_{\pm} = \frac{\pm [\psi^{*'2}(s) + 2\psi^{*''}(s)(\rho_i^2 - \psi^*(s))]^{1/2} - \psi^{*'}(s)}{\psi^{*''}(s)} \tag{54}$$

If the initial guess for $s_{\pm}$ is chosen as s_0 for which $\psi^{*'}(s_0) = 0$, then a compelling choice for the first iteration is

$$s_{\pm} = s_0 \pm \left(2 \frac{\rho_i^2 - \psi^*(s_0)}{\psi^{*''}(s_0)} \right)^{1/2} \tag{55}$$

and for the following iterations

$$s_{\pm} \leftarrow s_{\pm} + \frac{sgn\,\psi^{*'}(s_{\pm})[\psi^{*'2}(s_{\pm}) + 2\psi^{*''}(s_{\pm})(\rho_i^2 - \psi^*(s_{\pm}))]^{1/2} - \psi^{*'}(s_{\pm})}{\psi^{*''}(s_{\pm})} \tag{56}$$

The evaluation of the derivatives $\psi^{*'}(s)$ and $\psi^{*''}(s)$ uses the total derivative along the chord

$$\psi^{*'}(s) = \frac{d\psi^*}{ds} = \frac{\partial\psi^*}{\partial R}\frac{dR}{ds} + \frac{\partial\psi^*}{\partial Z}\frac{dZ}{ds} + \frac{\partial\psi^*}{\partial\varphi}\frac{d\varphi}{ds} \tag{57}$$

in which the partial derivatives of ψ^* are evaluated as described in Section 5.3 and the derivatives of R, Z and φ with respect to s are analytically derived from eq. (21). A similar development applies to $\psi^{*''}(s)$.

6.4 Flux surface geometry

Implementation: method *m2m* of class *mfd*

The particular case of an intersection of a flux surface with a straight line originating from the magnetic axis at a poloidal angle θ yields the description of the flux surface geometry in term of its radius $r^*(\rho, \theta, \varphi)$.

First the magnetic axis in a plane with constant φ must be localised by its coordinates $(R_{\psi^*=0}, Z_{\psi^*=0}) = (R_0, Z_0)$ using the conditions

$$\nabla\psi^*(R_0, Z_0) = 0 \tag{58}$$

where only the components in the R and Z directions are considered. Following the same idea as in Section 6.3, the gradient is approximated with

$$\nabla\psi^*(R + \Delta R, Z + \Delta Z) \cong \nabla\psi^*(R, Z) + \Delta\psi^*(R, Z)\begin{bmatrix}\Delta R \\ \Delta Z\end{bmatrix} \tag{59}$$

Here also only the elements in R and Z of the Laplacian $\Delta\psi^*$ are retained. Then the solution for eq. (58) can be found by iteratively replacing (R_0, Z_0) by

$$\begin{bmatrix}R_0 \\ Z_0\end{bmatrix} \leftarrow \begin{bmatrix}R_0 \\ Z_0\end{bmatrix} - \Delta\psi^{*-1}(R_0, Z_0)\nabla\psi^*(R_0, Z_0) \tag{60}$$

Along a radius at poloidal angle θ the first and second derivatives of ψ^* are easily computed as

$$\psi^{*\prime} = -\frac{\partial \psi^*}{\partial R}\cos\theta + \frac{\partial \psi^*}{\partial Z}\sin\theta$$
$$\psi^{*\prime\prime} = \frac{\partial^2 \psi^*}{\partial R^2}\cos^2\theta + 2\frac{\partial^2 \psi^*}{\partial R \partial Z}\cos\theta\sin\theta + \frac{\partial^2 \psi^*}{\partial Z^2}\sin^2\theta \tag{61}$$

Since on the magnetic axis $r^*_0 = \psi^*_0 = 0$, the initial guess for the intersection of this radius with the flux surface where $\psi^* = \rho_i^2$ is taken as

$$r^* = \left(\frac{2\rho_i^2}{\psi^{*\prime\prime}(R_0, Z_0)}\right)^{1/2} \tag{62}$$

and then refined with the iterations

$$r^* \leftarrow r^* + \frac{sgn\,\psi^{*\prime}(r^*)[\psi^{*\prime 2}(r^*) + 2\psi^{*\prime\prime}(r^*)(\rho_i^2 - \psi^*(r^*))]^{1/2} - \psi^{*\prime}(r^*)}{\psi^{*\prime\prime}(r^*)} \tag{63}$$

6.5 Flux surface geometrical parameters

Implementation: method *fsg* of class *mfd*

The geometrical parameters of flux surfaces are useful quantities in the analysis of the measured data and the subsequent simulations. Table 6 gives a list of these parameters and the corresponding formulas, which make an intensive use of the metric of a flux-coordinate system and the integro-differential calculation capabilities of the package.

Parameters	Formulas
Area	$S(\rho, t) = \int ds_\rho = \int_0^{2\pi}\int_{-\pi}^{\pi} r^*(R_0 - r^*\cos\theta)\left(1 + \left(\frac{\partial r^*}{r^*\partial\theta}\right)^2\right)^{1/2} d\theta d\varphi$ (64)
Cross-section	$A(\rho, \varphi, t) = \int ds_\varphi = \int_{-\pi}^{\pi}\int_0^{\rho} r^*\frac{\partial r^*}{\partial\rho} d\rho d\theta$ (65)
Volume	$V(\rho, t) = \int dV = \int_0^{2\pi}\int_{-\pi}^{\pi}\int_0^{\rho} r^*\frac{\partial r^*}{\partial\rho}(R_0 - r^*\cos\theta) d\rho d\theta d\varphi$ (66)

Table 6: Flux surface geometrical parameters

Parameters	Formulas
Differential cross-section	$\frac{\partial A(\rho, \varphi, t)}{\partial \rho} = \int_{-\pi}^{\pi} r^* \frac{\partial r^*}{\partial \rho} d\theta$ (67)
Differential volume	$\frac{\partial V(\rho, t)}{\partial \rho} = \int_{0}^{2\pi} \int_{-\pi}^{\pi} r^* \frac{\partial r^*}{\partial \rho} (R_0 - r^* \cos\theta) d\theta d\varphi$ (68)
Flux surface average of $\lvert\nabla\rho\rvert$	$\langle\langle \lvert\nabla\rho\rvert \rangle\rangle_\rho(\rho, t) = \frac{\int \lvert\nabla\rho\rvert ds_\rho}{\int ds_\rho}$ (69) $= S(\rho, t)^{-1} \int_{0}^{2\pi} \int_{-\pi}^{\pi} r^* \frac{\partial r^*}{\partial \rho} (R_0 - r^* \cos\theta) \left(1 + \left(\frac{\partial r^*}{r^* \partial \theta}\right)^2\right) d\theta d\varphi$

Table 6: Flux surface geometrical parameters

7. Implementation

The package uses object oriented programming and the concepts introduced above are implemented either as classes (Table 7) or methods on these classes (Table 8), as quoted in the previous sections. The description of the geometry of the chords of a diagnostic is stored in an object of class *dcd*, the points for a particular discretisation of a coordinate system are kept in an object of class *grid*; a particular space dependant function is held in an object of class *fct*. The class *psi* is a subclass of *fct* from which it inherits all the properties and methods plus specific ones to manipulate the description of the magnetic field using the corresponding flux function. It contains either the flux function in cylindrical coordinate normalised so that its value is zero on the last closed flux surface

$$\psi(R, Z, \varphi, t), \psi(\rho = 1) = 0 \tag{70}$$

or so that it is zero on the magnetic axis and one on the last close flux surface

$$\psi^*(R, Z, \varphi, t) = 1 - \frac{\psi(R, Z, \varphi, t)}{\psi(R_{\psi^*=0}, Z_{\psi^*=0}, \varphi, t)}, \psi^*(\rho = 0) = 0, \psi^*(\rho = 1) = 1 \tag{71}$$

or the radius of the flux surfaces in flux coordinates $r^*(\rho, \theta, \varphi, t)$ plus the position of the magnetic axis $(R_{\psi^*=0}(t), Z_{\psi^*=0}(t))$.

Class	Properties	Description
dcd		Diagnostic chord description
	rd,zd,pd,ad,td	$(R_d, Z_d, \varphi_d, \theta_d, \phi_d)_{l_1 \ldots l_N[l_t]}$ (eq. (1) or (2))
	t	Time base
grid		Points in a coordinate system
	type	Coordinate system (Table 2) and discretisation type (Section 4.1)
	q	$(q_1, q_2, q_3, [v_1, \ldots, v_M])_{l_1 \ldots l_N[m_1 \ldots m_M][l_t]}$ for discrete points (eq. (7), (10) or (12))
		$(s, [v_1, \ldots, v_M])_{l_d l_1 \ldots l_N[m_1 \ldots m_M][l_t]}$ for diagnostic chord coordinate (eq. (8), (11) or (13))
		$(q_{1i}, q_{2j}, q_{3k}, [v_{1m_1}, \ldots, v_{Mm_M}])$ for points on a mesh (eq. (9) or (14))
	t	Time base
	r0,z0	$(R_0(t), Z_0(t))$ for toroidal coordinates
		$(R_{\psi^*=0}(\varphi, t), Z_{\psi^*=0}(\varphi, t))$ for cylindrical to flux coordinate transformation (eq. (19))
	fsr	Flux surface radius $r^*(\rho, \theta, \varphi, t)$ stored in an object of class *mfd* for cylindrical to flux coordinate transformation (eq. (20))
	psi	Normalised flux function $\psi^*(R, Z, \varphi, t)$ stored in an object of class *mfd* for cylindrical to flux coordinate transformation (eq. (19))
	dcd	Object of class *dcd* for diagnostic-chord to cylindrical coordinate transformation (eq. (21))
fct		Function of the space variables
	grid	Object of class *grid* on which the function is defined
	f	$f_{l_1 \ldots l_N[m_1 \ldots m_M][l_t]}$ for discrete points (eq. (22))
		$f_{l_d l_1 \ldots l_N[m_1 \ldots m_M][l_t]}$ for diagnostic chords (eq. (23))
		$f_{ijk[m_1 \ldots m_M][l_t]}$ for mesh points (eq. (24))
	t	Time base

Table 7: Classes and their properties

Class		Description
	Properties	
mfd		Magnetic field description
	type	Type of flux surface representation
	fct	Object of class *fct* which contains the flux function $\psi(R, Z, \varphi, t)$ or the normalised flux function $\psi^*(R, Z, \varphi, t)$ or the flux surface radius $r^*(\rho, \theta, \varphi, t)$
	r0,z0	Position of the magnetic axis $(R_{\psi^*=0}(\varphi, t), Z_{\psi^*=0}(\varphi, t))$

Table 7: Classes and their properties

The methods applying on these objects are listed in Table 8.

Class		Description
	Methods	
dcd		
	inter	$(s_{[\pm]1il_1\ldots l_N l_t}, s_{[\pm]2jl_1\ldots l_N l_t}, s_{[\pm]3kl_1\ldots l_N l_t})$ diagnostic-chord linear coordinate of the intersections with the coordinate surfaces of a mesh (eq. (42) or (43), Section 6.1)
	tmat	$T_{ijkl_1\ldots l_N l_t}$ transfer matrix of the intersection lengths between diagnostic chords and coordinate surfaces of a mesh (eq. (45), Section 6.2)
grid		
	g2g	Coordinate system conversion (Table 3)
	metric	Coordinate system metric $dl_i, ds_i, dV, \lvert\nabla q_i\rvert$ (Section 5.2)
fct		
	f2f	Interpolation of a function on new points in a different coordinate system (Section 5.1)
	der	Partial derivatives of a function (Section 5.3)
	int	Definite or indefinite integral of a function (Section 5.4)
	mean	Average of a function (Section 5.4)
mfd		

Table 8: Methods

Class		Description
	Methods	
	m2m	$\psi(R, Z, \varphi, t) \leftrightarrow \psi^*(R, Z, \varphi, t)$, $(R_{\psi^*=0}(\varphi, t), Z_{\psi^*=0}(\varphi, t))$ normalisation, denormalisation of a flux function (eq. (70) and (71)) or $\psi^*(R, Z, \varphi, t) \rightarrow r^*(\rho, \theta, \varphi, t)$, $(R_{\psi^*=0}(t), Z_{\psi^*=0}(t))$ description of the flux surfaces with its radius (Section 6.4)
	fsg	$S(\rho, t)$, $A(\rho, \varphi, t)$, $V(\rho, t)$, $\partial A(\rho, \varphi, t)/\partial\rho$, $\partial V\rho, t/\partial\rho$, $\langle\langle\lvert\nabla\rho\rvert\rangle\rangle_\rho(\rho, t)$ flux surface geometrical parameters (Table 6)

Table 8: Methods

7.1 Performances

The most computer demanding operation is the function interpolation, especially when it is involved in the Gauss-Newton iterations to find the intersection of straight lines with flux surfaces. The evaluation of the cubic spline interpolants is based on the de Casteljau recurrent algorithm [7], modified to replace the recursion by local storage and to unroll the loops in order to take benefit from the processor registers, instruction pipe-lining and parallel instructions. All linear algebra is carried out using an optimised version of the BLAS library.

The computation of the coefficients of the cubic spline basis functions which enters the extrapolation expression in eq. (25) may be explicitly written as

$$c_{ijkm_1\ldots m_M l_t} = \sum_{i'=1}^{N_1}\sum_{j'=1}^{N_2}\sum_{k'=1}^{N_3}\sum_{m_1'=1}^{M_1}\cdots\sum_{m_M'=1}^{M_M} f_{i'j'k'm_1'\ldots m_M' l_t} B^{-1}_{1i'i} B^{-1}_{2j'j} B^{-1}_{3k'k} B^{-1}_{v_1 m_1' m_1}\ldots B^{-1}_{v_M m_M' m_M} \quad (72)$$

where the following symbols are used for the matrices

$$\begin{aligned} &B_{1ii'} = b_{1i}(q_{1i'}),\ B_{2jj'} = b_{2j}(q_{2j'}),\ B_{3kk'} = b_{3k}(q_{3k'}), \\ &B_{v_1 m_1 m_1'} = b_{v_1}(v_{1m_1'}),\ \ldots,\ B_{v_M m_M m_M'} = b_{v_M}(v_{Mm_M'}) \end{aligned} \quad (73)$$

Since the elements of the matrices $B_1, B_2, B_3, B_{v_1}, \ldots, B_{v_M}$ depend only on the mesh, their inverse can be pre-calculated when the coordinate system is specified and later used to perform fast interpolation of any function at any point.

8. Illustrative examples

8.1 Flux surface geometrical parameters

The evaluation of the flux surface geometrical parameters introduced in Section 6.5 is a good example of the way the objects defined in the package can be easily manipulated with their methods. This evaluation is based only on the poloidal flux produced on TCV by the inverse equilibrium code LIUQE [5] from the magnetic measurements [6]. Starting from this flux expressed on an axisymmetric cylindrical coordinate mesh for several times, Table 9 shows the steps performed to compute the plasma cross section area with eq. (65)

Comment	Code
Variables for the mesh in cylindrical coordinates (R_i, Z_k), $i = 1 \ldots 28$, $k = 1 \ldots 65$	`r,z`
Variable for the time array t_{l_t}	`t`
Variable for the time dependant flux function $\psi(R_i, Z_k, t_{l_t}) = \psi_{ikl_t}$, $i = 1 \ldots 28$, $k = 1 \ldots 65$, $l_t = 1 \ldots N_t$	`p`
Create the mesh in cylindrical coordinates	`grz = grid('cyl',(r,,z))`
Create magnetic field description based on $\psi(R, Z, t)$	`psi = mfd('flux',fct(p,grz,t))`
Compute the flux surface cross section area $A(\rho, t)$	`a = fsg('A',psi)`

Table 9: Computation of the flux surface cross-section

Comment	Code
Internally to the *fsg* method, convert magnetic field representation from $\psi(R, Z, t)$ to $r^*(\rho, \theta, t)$	`fsr = m2m('fsr',psi)`
Extract flux-coordinate mesh for the function $r^*(\rho, \theta, t)$	`g = fsr.grid`
Compute the surface element along the 3rd coordinate φ using the *metric* method: $ds_\varphi/(d\rho d\theta) = (\partial r^*/\partial\rho)r^*$	`m = metric('s3',g)`
Internally to the *metric* method, extract the function $r^*(\rho, \theta, t)$ from the coordinate system mesh grid	`r = g.fsr`
Derive r^* once along the 1st coordinate and computes $(\partial r^*/\partial\rho)r^*$	`m = der(r,(1,0,0))*r`
Compute the cross section area $\int_{-\pi}^{\pi}\int_{0}^{\rho} r^*\frac{\partial r^*}{\partial\rho}d\rho d\theta$ as the definite integral along 2nd coordinate θ and indefinite integral along 1st coordinate ρ	`a = int(m,def=2,ind=1)`

Table 9: Computation of the flux surface cross-section

8.2 Observation of a helical structure in toroidal symmetry

A helical perturbation follows the helicity of the magnetic lines along a magnetic surface. In a first step a poloidal angle coordinate ζ must be derived such that the magnetic lines in the (ζ, φ) plane appear as straight lines with a slope $\iota(\rho)$ different for each magnetic surface. This poloidal angle therefore satisfies

$$\frac{\vec{B}\cdot\nabla\zeta}{\vec{B}\cdot\nabla\varphi} = \iota(\rho) \tag{74}$$

Because of the toroidal symmetry, the form given by (5) can be used for the magnetic field in equation (74), leading to

$$\frac{\nabla\varphi\times\nabla\psi\cdot\nabla\theta\partial_\theta\zeta}{RB_\varphi R^{-2}} = \iota(\rho) \tag{75}$$

where the fact that $\nabla\psi$ is colinear to $\nabla\rho$ and $\nabla\varphi = R^{-1}$ have been used. The poloidal angle ζ is then derived from

$$\partial_\theta \zeta = \frac{\iota(\rho) R B_\varphi}{R^2 \nabla\varphi \times \nabla\psi \cdot \nabla\theta} \tag{76}$$

Assuming that the flux function ψ is given in cylindrical coordinates, the denominator in equation (76) is expressed as follows:

$$R^2 \nabla\varphi \times \nabla\psi \cdot \nabla\theta = R(\partial_Z \psi \partial_R \theta - \partial_R \psi \partial_Z \theta) = R \frac{\partial_Z \psi \Delta R + \partial_R \psi \Delta Z}{\Delta R^2 + \Delta Z^2} \tag{77}$$

in which the second equality is obtained by applying the transformations (17) and defining $\Delta R = R - R_0$ and $\Delta Z = Z - Z_0$. Then the value for ζ, for which the points $\zeta(\rho, \theta = 0) = 0$ and $\zeta(\rho, \theta = 2\pi) = 2\pi$ can be fixed, is computed with

$$\zeta(\rho, \theta) = 2\pi \frac{\int_0^\theta \frac{(\Delta R^2 + \Delta Z^2)/R}{\partial_Z \psi \Delta R + \partial_R \psi \Delta Z} d\theta}{\int_0^{2\pi} \frac{(\Delta R^2 + \Delta Z^2)/R}{\partial_Z \psi \Delta R + \partial_R \psi \Delta Z} d\theta} \tag{78}$$

The helical structure can be represented as a function of the flux coordinate; in this illustration a radialy localised function with a poloidal periodicity of one was chosen:

$$f(\rho, \theta) = e^{-\left(\frac{\rho - 0.7}{0.2}\right)^2} \cos\left(\zeta(\rho, \theta) - \frac{\pi}{4}\right) \tag{79}$$

This structure is observed by a 64 chords camera as the integral of f along each chord and is illustrated in Figure 4 and Figure 5.

The implementation of this computation using the package is summarised in Table 10.

Comment	Code
Variable for the mesh in flux coordinates $(\rho_{i'}, \theta_{j'})$, $i' = 1 \ldots 41, j' = 1 \ldots 129$	`rh,th`
Create the mesh in flux coordinates	`gf = grid('flux',(rh,th))`
Compute the position in cylindrical coordinates of the points of the flux coordinate mesh. Internally the *g2g* method computes the flux surface radius $r^*(\rho, \theta, t)$ from the flux function $\psi(R, Z, t)$, then do a flux to toroidal coordinate conversion $(r_{i'j'l_t}, \theta_{i'j'l_t}) = (r^*(\rho_{i'}, \theta_{j'}, t_{l_t}), \theta_{j'})$ followed by a toroidal to cylindrical conversion $(R'_{i'j'l_t}, Z'_{i'j'l_t}) = (R_{0l_t} - r_{i'j'l_t} cos\theta_{i'j'l_t}, Z_{0l_t} + r_{i'j'l_t} sin\theta_{i'j'l_t})$	`gc = g2g('cyl',gf,psi)`
Compute the intermediate quantities $\Delta R_{i'j'l_t} = R'_{i'j'l_t} - R_{0l_t}$ and $\Delta Z_{i'j'l_t} = Z'_{i'j'l_t} - Z_{0l_t}$	`dr = gc.q(1) - psi.r0` `dz = gc.q(3) - psi.z0`

Compute the partial derivatives of the flux function $\partial_R\psi$ and $\partial_Z\psi$ and map them on the flux coordinate mesh $\partial_R\psi(\rho_{i'}, \theta_{j'}, t_{l_t}) = \partial_R\psi_{i'j'l_t}$ and $\partial_Z\psi(\rho_{i'}, \theta_{j'}, t_{l_t}) = \partial_Z\psi_{i'j'l_t}$. Since the algorithm for evaluating the cubic splines in the interpolation estimates in that order the third, second, first derivatives and then the function itself, it is advantageous to combine the interpolation with the derivation operation and to stop the evaluation at the requested derivative; this is implemented in the *f2f* method.	`drpsi =`
	`f2f(psi,gf,der=(1,0,0))`
	`dzpsi =`
	`f2f(psi,gf,der=(0,1,0))`

Evaluation of $\partial_\theta\zeta(\rho, \theta, t)$ as $\partial_\theta\zeta'(\rho_{i'}, \theta_{j'}, t_{l_t}) = \frac{(\Delta R_{i'j'l_t}^{\,2} + \Delta Z_{i'j'l_t}^{\,2})/R'_{i'j'l_t}}{\partial_Z\psi_{i'j'l_t}\Delta R_{i'j'l_t} + \partial_R\psi_{i'j'l_t}\Delta Z_{i'j'l_t}} = \partial_\theta\zeta'_{i'j'l_t}$	`dzh1 = (dr*dr+dz*dz)/`
	`gc.q(1)/`
	`(dzpsi*dr+drpsi*dz)`

Integration of $\partial_\theta\zeta(\rho, \theta, t)$ to obtain $\zeta(\rho, \theta, t)$ $\zeta'(\rho_{i'}, \theta_{j'}, t_{l_t}) = \int_0^\theta \partial_\theta\zeta' d\theta = \zeta'_{i'j'l_t}$	`zh1 = int(dzh1,ind=2)`

Table 10: Observation of a helical structure

Comment	Code
Normalisation of $\zeta(\rho, \theta, t)$ to get $\zeta(\rho, 0, t) = 0$ and $\zeta(\rho, 2\pi, t) = 2\pi$: $\zeta(\rho_{i'}, \theta_{j'}, t_{l_t}) = 2\pi \frac{\zeta'(\rho_{i'}, \theta_{j'}, t_{l_t})}{\zeta'(\rho_{i'}, 2\pi, t_{l_t})} = \zeta_{i'j'l_t}$	`zh = 2*pi*zh1/zh1(,2*pi,)`
Construction of the helical structure function $f(\rho, \theta, t)$ as $f(\rho_{i'}, \theta_{j'}, t_{l_t}) = e^{-\left(\frac{\rho_{i'} - 0.7}{0.2}\right)^2} cos\left(\zeta_{i'j'l_t} - \frac{\pi}{4}\right) = f_{i'j'l_t}$	`f = exp(-((rh-0.7)/ 0.2)^2)*cos(zh-pi/4)`
Variables for geometrical parameters of the diagnostic chords $(R_d, Z_d, \varphi_d, \theta_d, \phi_d)_l$, $l = 1 \ldots 64$	`rd,zd,pd,ad,td`
Create the diagnostic chord description	`d = dcd(rd,yd,pd,ad,td)`
Create a discretised linear coordinate along the chords $s_{l_d l}$, $l_d = 1 \ldots N_d$, $l = 1 \ldots 64$	`sd = grid('diag',d)`
Interpolate the helical structure function $f(\rho, \theta, t)$ on the discrete points along the diagnostic chords to get $f_d(s, t)$ along each chord. Internally the *f2f* method will convert the linear coordinates $s_{l_d l}$ first in cylindrical coordinates $(R_{l_d l}, Z_{l_d l})$ and then in flux coordinate interpolating the $\psi(R, Z, t)$ function on these cylindrical positions to get $(\rho_{l_d l l_t}, \theta_{l_d l l_t})$; then the function $f(\rho, \theta, t)$ is interpolated for this coordinates, yielding $f_{l_d l l_t}$.	`fd = f2f(f,sd)`
Integrate the functions $f_d(s, t)$ along the first coordinate to obtain the line integral for each chord $g_d(t)$.	`g = int(fd,def=1)`

Table 10: Observation of a helical structure

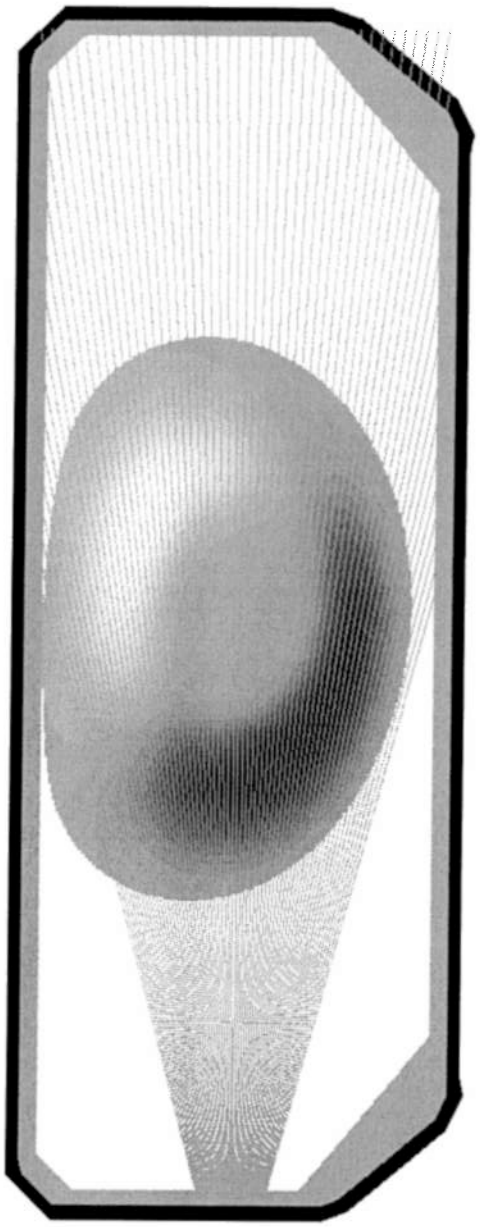

Fig. 4 *Observation of a helical structure by a multi chord camera. The structure in the plasma is represented in grey scale (black for negative values, grey for zero and white for positive value) superposed with the 64 chords of the camera.*

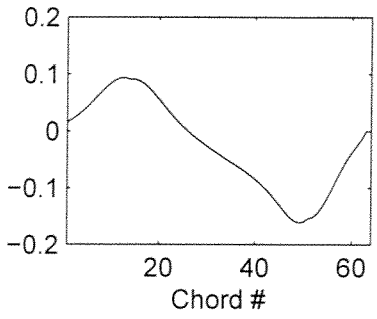

Fig. 5 Line integrated signal for each chord of the camera.

9. Appendix

9.1 Cubic spline interpolants

For a given set of N distinct knots x_i, $i = 1, \ldots, N$ a set of N base functions $b_i(x)$, $i = 1, \ldots, N$ can be derived [7], each of them being a piecewise cubic polynomial which satisfies the following conditions

- $b_i(x)$ are zero outside the interval $[x_{i-3}, x_{i+1}]$
- $b_i(x)$, $b'_i(x)$, $b''_i(x)$ are continuous

plus one of the following edge condition pair

- not a knot: $b'''_i(x)$ are continuous at x_2 and x_{N-1}
- symmetric around x_1 : $b'_i(x_1) = b'''_i(x_1) = 0$
- periodic: $b'_i(x_1) = b'_i(x_N)$ and $b''_i(x_1) = b''_i(x_N)$

Then if a function $f(x)$ is defined by its values at the grid points $f(x_i) = f_i$, it can be interpolated by the base function combination

$$f(x) = \sum_{j=1}^{N} c_j b_j(x) \tag{80}$$

where the coefficients are derived from the equations formed by the conditions

$$f(x_i) = \sum_{j=1}^{N} c_j b_j(x_i) = f_i,\ i = 1, \ldots, N \tag{81}$$

Functions of more than one variable can also be treated with extension

$$f(x_1, \ldots, x_N) = \sum_{i_1=1}^{N_1} \cdots \sum_{i_N=1}^{N_N} c_{i_1 \ldots i_N} b_{i_1}(x_1) \ldots b_{i_N}(x_N) \tag{82}$$

Note finally that the derivative of f with respect to its variables is easily written as

$$\frac{\partial^n f(x_1, \ldots, x_N)}{\partial x_j^{\,n}} = \sum_{i_1=1}^{N_1} \cdots \sum_{i_N=1}^{N_N} c_{i_1 \ldots i_N} b_{i_1}(x_1) \ldots \frac{\partial^n b_j(x_j)}{\partial x_j^{\,n}} \ldots b_{i_N}(x_N) \tag{83}$$

and easily computed since the base function are simple cubic polynomial. The same applies to definite or indefinite integrals

$$\int f(x_1, \ldots, x_N) dx_j = \sum_{i_1=1}^{N_1} \cdots \sum_{i_N=1}^{N_N} c_{i_1 \ldots i_N} b_{i_1}(x_1) \ldots \left(\int b_j(x_j) dx_j\right) \ldots b_{i_N}(x_N) \tag{84}$$

10. References

[1] F. Hofmann et al., Plasma Phys. and Contr. Fusion **36** (1994) B277.

[2] I.H. Hutchinson, "Principles of Plasma Diagnostics", 1992, Cambridge University Press, Cambridge

[3] A.H. Boozer, Phys. Fluids **26** (1983) 1288.

[4] G. Arfken, "Mathematical Methods for Physicists", 1970, Academic Press, New York.

[5] F. Hofmann, G. Tonetti, Nucl. Fusion **28** (1988) 1871.

[6] J.-M. Moret et al., Rev. Sci. Instrum. **69** (1998) 2333.

[7] C. de Boor, "A practical guide to splines", 1985, Springer Verlag, New York.

[8] J.-M. Moret et al., Phys. Rev. Letters **79** (1997) 2057.

Acknoledgments. This work was supported in part by the Swiss National Science Foundation.