Notes on coupling an equilibrium code to the RAPTOR suite of codes

F. Felici, federico.felici@epfl.ch

May 10, 2019

Version 2.2

Contents

1 Introduction

This document describes the quantities that depend on the plasma equilibrium, that are required as input for the real-time suite of codes linked to RAPTOR.

This suite is broadly defined as a set of algorithms to perform real-time physics based estimation and prediction of plasma profiles. It may include RAPTOR and other observers such as neObs, and source calculation codes such as RABBIT and TORBEAM.

It is assumed that a real-time equlibrium code returns $\psi(R,Z)$ on a known (R,Z) grid. We describe here the post-processing calculations that should be done based on this $\psi(R,Z)$ map. These can be done either as post-processing steps within the equilibrium code itself, or in separate codes/modules.

The outputs should all be given in the code's own COCOS convention [1][1], and should be consistent with the experimental signs (e.g. the code's output $F = RB_\phi$ must flip sign with the experimental B_0, and q must flip sign with both B_0 and I_p).

2 Flux surface integrals

We propose a specific set of geometry profiles to be calculated in the equilibrium code, for specific values of normalized ψ. These are chosen to avoid infinite quantities at the edge or badly defined quantities at the axis.

One has to choose on which values of $\psi_N = (\psi - \psi_a)/(\psi_b - \psi_a)$, where ψ_a and ψ_b are the flux at the axis and boundary respectively, to compute the contour integrals and shape parameters. This set of psiN points on which the geometry quantities are evaluated should ideally be given as an input parameter vector to the code. A default choice is described below:

2.1 Choice of default ψ_N grid

Near the axis, one is limited by numerical errors due to grid resolution in computing the contour integrals. Close to the boundary for diverted plasmas, however, one would like to have more contours since the value of some contour integrals rapidly diverges to zero. It is proposed to quadratically distribute the points as follows:

Starting from an equidistant grid $x \in [0,1]$ with number of points N_p, the values of ψ_N for which the contours are calculated $\psi_N = 1 - (1-x)^2$. For 16 points, this gives

$$\psi_N = \left[\begin{array}{c|c|c|c|c|c|c|c} 0 & 0.1289 & 0.2489 & 0.3600 & 0.4622 & 0.5556 & 0.6400 & 0.7156 \\ \hline 0.7822 & 0.8400 & 0.8889 & 0.9289 & 0.9500 & 0.9822 & 0.9956 & 1.0000 \end{array}\right] \tag{1}$$

Note that the point at 0.95 should be at 0.96 according to the formula, but has been moved in the interest of calculating q_{95} at the same time as the other contour integrals.

2.2 Flux-surface quantities required as output from the equilibrium code

We define in Table 1 the profiles to be calculated as outputs of the equilibrium code. We also specify their (analytical) values at the magnetic axis and for a diverted last closed flux surface.

[1]Known COCOS values:, LIUQE (COCOS=17), JANET (COCOS=17), CHEASE (COCOS=2), EQUINOX (COCOS=7). E-mail the author with your COCOS number to add it to the list.

Quantity	Suggested name	Value on axis	Value on flux surfaces without $\times$-point	Value on separatrix with $\times$-point	Relation to RAPTOR internal variable (COCOS=11)
$\langle 1/R \rangle$	`iR`	$1/R_a$	C_0/C_1	$1/R_\times$	–
$\langle 1/R^2 \rangle$	`iR2`	$1/R_a^2$	C_2/C_1	$1/R_\times^2$	`iR2`
$\langle B_p^2 \rangle$	`Bp2`	0	C_3/C_1	0	`dpsi2oR2` $= (4\pi^2)(\cdot)$
$\langle (RB_p)^2 \rangle$	`R2Bp2`	0	C_4/C_1	0	`dpsi2` $= (4\pi^2)(\cdot)$
$\langle RB_p \rangle$	`RBp`	0	C_5/C_1	0	`dpsi` $= (2\pi)(\cdot)$
$1/(\oint \frac{\mathbf{B_p}\cdot \mathbf{d}\ell}{B_p^2})$	`H`	$\frac{F(0)}{2\pi R_{ax}^2 q_0}$	$1/C_1$	0	`dpsidV` $= H$
$(1/q)$	`iota`	$1/q_0^\dagger$	$2\pi/(FC_2)$	0	`iota`
$F = RB_\phi$	`F`	$F(0)$	$F(\psi)$	$F(\psi_b)$	`F`
$\langle \frac{j_\phi}{R} \rangle$	`jphioR`	$p' + \frac{1}{\mu_0}\langle \frac{1}{R^2} \rangle FF'$	$p' + \frac{1}{\mu_0}\langle \frac{1}{R^2} \rangle FF'$	$p' + \frac{1}{\mu_0}\langle \frac{1}{R^2} \rangle FF'$	-

Table 1: Geometric profiles to be returned by equilibrium code and expressions for correct evaluation including axis and LCFS.
$*$Note the relation $2\pi/\oint \frac{d\ell}{B_p} = 2\pi H = F\left\langle \frac{1}{R^2} \right\rangle \frac{1}{q}$. Hence only three out of the four quantities H, F, $1/q$, $\langle 1/R^2 \rangle$ are required, and the third can always be recovered. $*$Note also that the surface area of the 3D flux surface (not to be confused with the poloidal cross-sectional area) is equal to $S_\psi = 2\pi C_5$.
†Note that q_0 can be computed accurately from a local expansion of the equilibrium $\psi(R, Z)$ around the magnetic axis using equation (88) in [2].

2.2.1 Contour integral definition with cosistent signs

Quantities in Table 1 are defined as ratios of contour integrals on the surfaces of constant ψ on the poloidal cross-section. Following CHEASE [3], one defines the contour integrals $C_0 \dots C_5$ of the form:

$$C_i = \oint |\nabla\psi|^\mu R^\nu \mathrm{d}\ell \tag{2}$$

where $\mathrm{d}\ell$ is the element of the poloidal flux surface contour. To be independent of the definition of ψ in each code, they are written here in terms of the poloidal field B_p, and rewritten in generic that works for any COCOS choice of a particular code.

To be fully consistent, the contour integrals should be evaluated with the proper sign of the contour integration, with the line integration element $\mathbf{dl_p}$ in the direction of positive θ for the code's coordinate system. Alternatively, absolute values can be used in the calculations, but the results must be corrected depending on the sign of plasma current and the sign convention of the code (see (4)–(9)).

Using the COCOS convention we may write

$$|B_p| = \frac{1}{(2\pi)^{e_{B_p}}} \frac{|\nabla\psi|}{R} \quad \text{and} \quad \oint \mathbf{B_p}\cdot\mathbf{dl_p} = \sigma_{I_p}\sigma_{\rho\theta\phi} \oint |B_p| dl_p \tag{3}$$

This gives the following definitions of the contour integrals

$$C_0 = \oint \frac{\mathbf{B_p}\cdot\mathbf{dl_p}}{B_p^2 R} = \sigma_{I_p}\sigma_{\rho\theta\phi} \oint \frac{1}{R}\frac{\mathrm{d}l_p}{|B_p|} = \sigma_{I_p}\sigma_{\rho\theta\phi}(2\pi)^{e_{Bp}} \oint \frac{\mathrm{d}l_p}{|\nabla\psi|} \tag{4}$$

$$C_1 = \oint \frac{\mathbf{B_p} \cdot \mathbf{dl_p}}{B_p^2} = \sigma_{I_p}\sigma_{\rho\theta\phi} \oint \frac{\mathrm{d}l_p}{|B_p|} = \sigma_{I_p}\sigma_{\rho\theta\phi}(2\pi)^{e_{Bp}} \oint \frac{R\mathrm{d}l_p}{|\nabla\psi|} \tag{5}$$

$$C_2 = \oint \frac{\mathbf{B_p} \cdot \mathbf{dl_p}}{R^2 B_p^2} = \sigma_{I_p}\sigma_{\rho\theta\phi} \oint \frac{1}{R^2}\frac{\mathrm{d}l_p}{|B_p|} = \sigma_{I_p}\sigma_{\rho\theta\phi}(2\pi)^{e_{Bp}} \oint \frac{\mathrm{d}l_p}{R|\nabla\psi|} \tag{6}$$

$$C_3 = \oint \mathbf{B_p} \cdot \mathbf{dl_p} = \sigma_{\rho\theta\phi}\mu_0 I_p(\psi) = \frac{\sigma_{I_p}\sigma_{\rho\theta\phi}}{(2\pi)^{e_{Bp}}} \oint \frac{|\nabla\psi|}{R}\,\mathrm{d}l_p \tag{7}$$

$$C_4 = \oint R^2 \mathbf{B_p} \cdot \mathbf{dl_p} = \sigma_{I_p}\sigma_{\rho\theta\phi} \oint R^2 B_p^2 \frac{\mathrm{d}l_p}{|B_p|} = \frac{\sigma_{I_p}\sigma_{\rho\theta\phi}}{(2\pi)^{e_{Bp}}} \oint R|\nabla\psi|\,\mathrm{d}l_p \tag{8}$$

$$C_5 = \oint R\,\mathrm{d}l_p \tag{9}$$

Here, σ_{Ip} is the sign of the plasma current in the code's COCOS convention. $\sigma_{\rho\theta\phi}$, σ_{Bp} and e_{Bp} are the values corresponding to the code's COCOS choice [1].

2.2.2 Note on values at the axis or LCFS

For closed flux surfaces excluding the axis or diverted last closed flux surface, the contour integrals can be computed by numerical integration around the contour. The accuracy of the contour integration is determined by the spatial grid resolution and numerical contour integration method, and may be compromised close to the axis (due to very few grid points) and close to a diverted LCFS (due to strong curvature of the flux surfaces close to the $\times$-point.

Consider cases with $\mu < 0$. In these cases both $\mathrm{d}\ell$ and $|\nabla\psi|^\mu$ go to infinity at the axis, and the result is ill-defined. At a LCFS with $\times$-point, $|\nabla\psi|^\mu$ goes to infinity and the integral goes to infinity. This is why many flux surface quantities need to be evaluated *only* at closed, or limited flux surfaces to compute the expressions of Table 1. Values at the axis and diverted LCFS should never be computed by contour integration, nor should they be extrapolated. Instead, their exact values are prescribed in Table 1.

2.2.3 Outline of numerical algorithm to compute C_i

INITIALIZATION: load $(\psi_{N,grid}, R_{grid}, Z_{grid})$;
STEP: Get data for time k: $(\Psi_N), R_{ax}, Z_{ax}$;
for *Value* i *on* $\psi_{N,grid}$ **do**
  $\psi_{N,i} = \psi_{N,grid}[i]$;
  if $\psi_{N,i} = 0$ *OR* ($\psi_{N,i} = 1$ *AND PlasmaIsDiverted)* **then**
    Do not compute contour integrals, treat later using special rules.
  else Closed flux surface
    (R_c, Z_c) = ContourFinder$(\Psi_N, R_{grid}, Z_{grid}, \psi_{N,grid}, R_{ax}, Z_{ax})$ [a];
    L_c = ComputeSegmentLength(R_c, Z_c);
    $|\nabla\psi|_c$ = InterpolatingDerivative$(\Psi_N, R_{grid}, Z_{grid}, R_c, Z_c)$;
    for $j = 0 : 5$ **do**
      $G_c = |\nabla\psi|_c^\mu R_c^\nu$ (Integrand from eq(2));
      C_j = TrapezoidalInterpolation(L_c, G_c)
    end
end

Algorithm 1: Outline of algorithm for computing contour integrals $C_0, \ldots, C_5$

[a] CountourFinder() should use known axis position to find good starting points for each contour

3 Shape descriptors

Some codes require, e.g. for neoclassical calculations and for some advanced transport models, profiles of the shaping information for each flux surface. Expressions for these quantities are given in Table 2. The values should be returned for each flux surface on the same ψ_N grid as used in Section 2.2.

Quantity	Value on axis	Flux surfaces with or without ×-point	suggested variable name
R_{geom}	R_{ax}	$(R_{max} + R_{min})/2$	`Rgeom`
ϵ	0	$a/R_{geom} = (R_{max} - R_{min})/(R_{max} + R_{min})$	`epsilon`
κ	0	$(Z_{upper} - Z_{lower})/(R_{max} - R_{min})$	`kappa`
δ	0	$(2R_{geom} - R_{upper} - R_{lower})/(R_{max} - R_{min})$	`delta`

Table 2: Flux surface shape quantities

4 Surface/Volume integrated quantities

In addition to the flux quantities of Section 2.2, some integral quantities are needed. These should be returned on the same ψ_N grid as the flux quantities of Section 2.2.

Depending on the equilibrium reconstruction code, it may or may not be easily possible to perform 2D surface integration over flux surfaces. If accurate integration is possible, then this is preferable, otherwise, contour integrals can be integrated over ψ. Both options are listed in Table 3.

Quantity	unit	Suggested name	Value on axis	Value as surface integrals in (R,Z) plane	Value as integrals over ψ
$V(\psi)$ (Volume)	[m^3]	`Vol`	0	$2\pi \int^{\Omega_p} R dS_\phi$	$2\pi c_* \int_{\psi_a}^{\psi} C_1 d\psi$ $(\star)$
$S(\psi)$ (Pol. area)	[m^2]	`Area`	0	$\int^{\Omega_p} dS_\phi$	$c_* \int_{\psi_a}^{\psi} C_0 d\psi$ $(\star)$
$\Phi(\psi)$ (Tor. flux)	[Wb]	`Phi`	0	$\int^{\Omega_p} \frac{F}{R} dS_\phi$	$c_* \int_{\psi_a}^{\psi} q d\psi$ $(\star)$
$\rho_{tor_N}(\psi)$	[-]	`rhotor`	0	-	$\sqrt{\Phi(\psi)/\Phi(\psi_b)}$
$\mu_0 I_p(\psi)$ (current)	[A]	`mu0Ip`	0	$\mu_0 \int^{\psi_N<1} j_\phi dS_\phi$	$\sigma_{\rho\theta\phi} C_3$

Table 3: Quantities related to surface integrals to be computed by equilibrium code. Beware that integrands marked with $(\star)$ diverge (are infinity at the LCFS), so trapezoidal integration is discouraged (see below). $c_* = 2\pi^{-e_{B_p}} \sigma_{\rho\theta\phi} \sigma_{B_p}$

Alternative integration scheme for diverging LCFS quantities In case one is integrating over ψ, some profiles take infinite value on the LCFS in case of an $\times$-point (as discussed previously). Integrating these quantities numerically up to the edge can then give inaccurate results. A typical example is the integration of q to obtain the toroidal flux. Using COCOS=11 sign conventions for brevity:

$$\Phi(\psi) = \int_{\psi_a}^{\psi} q(\psi) \mathrm{d}\psi \tag{10}$$

If q is equal to infinity for diverted LCFS, and grows to large values at ψ_N close to 1, it is easy to make an error in the value of $\Phi_b = \Phi(\psi_b)$, which is needed for the definition of ρ_{tor}, for example. A strategy to compute $\Phi(\psi)$ more accurately is as follows:

- Given $1/q$ on some grid of known ψ values (with grid index i), and recalling $\frac{1}{q} = \frac{\partial \psi}{\partial \Phi}$
- Write $\int_{\Phi_i}^{\Phi_{i+1}} 1/q \, \mathrm{d}\Phi = \psi_{i+1} - \psi_i$
- Formulate a numerical approximation to this integral. For trapeze integration, we get:

$$\frac{1}{2}\left(\frac{1}{q_i} + \frac{1}{q_{i+1}}\right)(\Phi_{i+1} - \Phi_i) = (\psi_{i+1} - \psi_i) \tag{11}$$

yielding a recursive equation for Φ_{i+1}

$$\Phi_{i+1} = \Phi_i + 2\left(\frac{1}{q_i} + \frac{1}{q_{i+1}}\right)^{-1}(\psi_{i+1} - \psi_i) \tag{12}$$

Note that the great advantage of this scheme is that $1/q_N = 0$ for a diverted LCFS, thus entirely avoiding the singularity.

5 Scalar quantities

Finally, the following scalar quantities are desired outputs of the code. See Table 4

Quantity	unit	Sugg. name	Value as surface integrals in (R,Z) plane	Value as integrals over ψ
W_k (Kin. energy)	[J]	`Wk`	$2\pi\frac{3}{2}\int^{\Omega_p} p(\psi)RdS_\phi$	$2\pi c_* \int_{\psi_a}^{\psi_b} pC_1 d\psi$ $(\star)$
W_{pol} (Pol. magnetic energy)	[J]	`Wpol`	$2\pi\int^{\Omega_p} \frac{B_p^2}{2\mu_0} RdS_\phi$	$\pi c_* \sigma_{I_p} \int_{\psi_a}^{\psi_b} I_p(\psi) d\psi$
W_{tor} (Plasma contribution to toroidal magnetic energy)	[J]	`Wtor`	$2\pi\int^{\Omega_p} \frac{B_\phi^2 - B_{\phi,vac}^2}{2\mu_0} RdS_\phi$	$\frac{2\pi c_*}{2\mu_0}\int_{\psi_a}^{\psi_b}(F^2 - R_0^2 B_0^2) C_2 d\psi$ $(\star)$
Φ_p (Plasma contribution to toroidal flux)	[Wb]	`Phip`	$\int^{\Omega_p}(B_\phi - B_{\phi,vac})dS_\phi$	$\Phi_b - c_* R_0 B_0 \int_{\psi_a}^{\psi_b} C_2 d\psi$ $(\star)$
ψ_a	COCOS	`psia`		
ψ_b	COCOS	`psib`		
R_{ax}	[m]	`Rax`		
Z_{ax}	[m]	`Zax`		
I_p	[A]	`Iptot`		
B_0	[T]	`B0`		

Table 4: Scalar quantities to be computed by equilibrium code. 'COCOS' units indicates that the unit depends on the chosen COCOS convention. $(\star)$ indicates again that the integrand diverges at a diverted LCFS.

6 Calculations for diagnostics interpretation

These only apply when the equilibrium code is to be interfaced to RAPTOR-observer or another code that uses real-time kinetic diagnostic data. The equilibrium code should then return some extra information on the mapping between equilibrium and a diagnostic. Depending on the preference of the equilibrium code developer, this is done either as part of the equilibrium code itself, or as a post-processing step in a separate module that receives the equilibrium map information from the equilibrium code.

6.1 Point-wise diagnostics

Inputs:

- Two vectors of (R, Z) [m] coordinates of point-wise diagnostic measurements (e.g. Thomson Scattering)

Outputs:

- ψ_N and ρ_{tor} values inside the plasma corresponding to these (R,Z) points, with a -1 value indicating the the point is outside the plasma region.

6.1.1 Outline of fast algorithm for evaluation of Ψ_N and ρ_{tor} on fixed R, Z points

INITIALIZATION: Given a fixed 2D grid (R_{grid}, Z_{grid}) and fixed evaluation points (R_c, Z_c):
$W_c = \text{InterpolationWeights}(R_{grid}, Z_{grid}, R_c, Z_c)$[a]
STEP: Get data for time k: $\bar{\Psi}_N$, $\psi_{N,grid}$, $\rho_{tor,N}$ on $\psi_{N,grid}$ (from Table 3)
$\bar{\Psi}_{N,c} = W_j \bar{\Psi}_N$ (Compute Ψ_N values on points along chord j) ;
$\rho_{tor,N,c} = \text{Interpolate1D}(\psi_{N,grid}, \rho_{tor,N}, \bar{\Psi}_{N,c})$

Algorithm 2: Outline of algorithm for computing integrals of basis functions on diagnostic chords.

[a] Compute weight matrix W_c that interpolates quantities from (R, Z) grid points onto the points R_c, Z_c, such that $\bar{\Psi}_{N,c} = W_c \bar{\Psi}_N$ where $\bar{\Psi}_{N,c}$ is the vector of Ψ values on R_c, Z_c and $\bar{\Psi}_N$ is the vector of Ψ_N values on the (R_{grid}, Z_{grid}) grid.

6.2 Line-integrated diagnostics

Inputs:

- Vector of n_l launch points (R, Z) (in [m]) and orientation vector $[k_r, k_z, k_\phi]$ (such that $\mathbf{k} = k_r \mathbf{e_r} + k_z \mathbf{e_z} + k_\phi \mathbf{e}_\phi$ is the direction of the chord towards the plasma, of any line integrating diagnostics.

- $m \times n_b$ matrix containing: Values of m basis functions $b_i(\bar{\psi}_N)$ for $i = \{1, \ldots, m\}$, evaluated on $\bar{\psi}_N$ grid (vector of n points). This may be a different grid than the input grid ψ_N defined in (1).

- n_b-sized vector containing $\bar{\psi}_N$ grid on which basis functions (above) are evaluated. above.

- Alternatively to the two points above, analytical expressions for the m basis functions $b_i(\psi_N)$ can be prescribed, then only m should be an input parameter to the code. For this option, B-splines are recommended due to the fact that they have a local support.

Outputs:

- Matrix $(m \times n_l)$ of line integrals over the chord for each basis function, each entry i, j being

$$M_{ij} = \int b_i(\psi_N(R, Z))ds_j \tag{13}$$

  With ds_j the line element along the jth diagnostic chord. In case of double-pass chords, the full path length should be considered.

6.2.1 Outline of algorithm for line integration of basis functions a on a fixed chord

INITIALIZATION:;
Given a fixed 2D grid (R_{grid}, Z_{grid}) grid and fixed launch point and launch direction:
for *Each chord j* **do**
  Compute a set of points (R_j, Z_j) that lies along the chord.
  Compute the path lengths L_j for each line segment.
  W_j = InterpolationWeights$(R_{grid}, Z_{grid}, R_j, Z_j)$ (Similar to Algorithm 2)
end
STEP:
Get data for time k: $\bar{\Psi}_N$ (2D psi map)
for *Each chord j* **do**
  $\bar{\Psi}_{N,j} = W_j \bar{\Psi}_N$ (Compute Ψ_N values on points along chord j) ;
  for *Each basis function i* **do**
    $V_{i,j} = b_i(\bar{\Psi}_{N,j})$, (Value of basis functions on each chord point)
    M_{ij} = TrapezoidalIntegration$(L_j, b_{i,j})$ (Line integral of basis function on chord)
  end
end

Algorithm 3: Outline of algorithm for computing integrals of basis functions on diagnostic chords.

6.3 ECE (cold approximation, X mode)

Inputs:

- Vector $(n_{ECE} \times 1)$ with frequency of each ECE channel [GHz].
- A vector of integers $(n_{ECE} \times 1)$ indicating the harmonic used (1,2,3..)
- Two matrices $(n_{ECE} \times 3)$ of launch points and receiver vectors of the antenna (pointing into the vessel) (see definition above).
- Plasma electron density values on a ψ_N grid [m^{-3}]

- *psiN* grid used to specify electron density profile.

Outputs:

- Vector of n_{ECE} with values of ψ_N where $B_{tot}(R,Z) = \sqrt{B_\phi^2 + B_p^2}$ matches for the first time the ECE cold resonance condition $f[\mathrm{Hz}] = q_e B/2\pi m_e$, along the antenna path.
- Vector of booleans indicating whether an ECE channel has a resonance inside the plasma (1=yes, 0=no).
- Vector of n_{ECE} values $f_{cutoff,max}$ indicating the maximum fraction of the ECE cutoff density encountered during the path between the launcher and the cold absorption point.

  For X-mode, there is only an upper density limit: $n_{cutoff,X} = l(l-1)B^2 \times 9.7197 \times 10^{18}$ hence

  $$f_{cutoff,max} = \max_s(f_{cutoff}(s)) = \max_s(n_e(s)/n_{cutoff,X}(s)) \tag{14}$$

  where s is the distance along the antenna launch vector, evaluated between the launch point and the cold resonance point, and

  $$n_{cutoff,X} = l(l-1)B^2 \times 9.7197 \times 10^{18} \tag{15}$$

  Here, l is the harmonic number and B is the total field in [T].

Changelist

- v2.2: Remove 2π from H definition
- v2.1: Add algorithm outlines
- v2.0: Complete rewrite with details of diagnostics interfaces, integral quantities etc (for JET, July 2018)

References

[1] O Sauter and S.Yu. Medvedev. Tokamak coordinate conventions. *Computer Physics Communications*, 184(2):293–302, feb 2013.

[2] J.-M. Moret, B.P. Duval, H.B. Le, S Coda, F Felici, and H Reimerdes. Tokamak equilibrium reconstruction code LIUQE and its real time implementation. *Fusion Engineering and Design*, 91(0):1–15, feb 2015.

[3] H Lutjens, A Bondeson, and O Sauter. The CHEASE code for toroidal MHD equilibria. *Computer Physics Communications*, 97(3):219–260, 1996.

[4] W. Suttrop and A.G. Peeters. Practical Limitations to Plasma Edge Electron Temperature Measurements by Radiomatry of Electron Cyclotron Emission. Technical report, IPP Garching, 1996.