Creating Tokamak Equilibria with Corsica

R. H. Bulmer

Lawrence Livermore National Laboratory

February 27, 2004

1 Introduction

This user's guide describes the creation of tokamak equilibria with a so-called "dead-start" procedure that takes a small set of parameters from a text file to create an initial tokamak equilibrium instead of starting up Corsica with a binary save-file. The initial equilibrium created by the dead-start process may then be modified by the user to refine the plasma shape and/or profiles to achieve the desired end state.

One or more text files are created by the user, as described in the following sections, to specify the tokamak plasma and associated structures (poloidal field coils, first-wall or limiter surface, passive structure, etc.). The Corsica executable, `caltrans`, is launched with the script file `tokamak.bas` named on the command-line:

```
caltrans [-probname pname] tokamak.bas
```

Optional argument "`-probname` *pname*" specifies that string *pname* will be used to name the session log file and graphics file. Type "`tokamak`" at the Corsica prompt to get a list of user-callable routines defined in the script file.

The dead-start procedure, `tokamak_ds`, or its synonym, `ds`, is executed identifying at least one file containing tokamak parameters and optionally a 2nd file name containing poloidal field coil specifications:

```
ds("tokamak.inp"[,"pfcoil.inp"])
```

On completion of the dead-start procedure, the equilibrium will be saved in a binary save-file for future use.

2 Tokamak parameters

The essential tokamak parameters used by the dead-start procedure are summarized in Table 1. They are used to define either wall-limited or X-point limited (either double-null, DN, or single-null, SN) configurations. The elevation of the magnetic axis, Z_{axis}, is usually zero, but for single-null (asymmetric) configurations it may be non-zero to offset the plasma axis vertically. The separatrix separation, ΔR_{sep}, is the distance between active and inactive separatrix surfaces, measured at the outboard midplane. It is zero for limited plasmas, a very small value (e.g., 1×10^{-6} m) for double-null plasmas, and $|\Delta R\_sep| > 1 \times 10^{-3}$ m for single-null plasmas. If ΔR_{sep} is positive, the active X-point will be at the top (SNT) and if it is negative, the active X-point will be at the bottom (SNB). The flux of the poloidal field coils linking the plasma, Ψ_{ext}, may be specified, or set to zero if it is not to be constrained.

Table 1: Tokamak parameters

name	*units*	*description*
I_φ	MA	toroidal current
R_0	m	major radius
a	m	minor radius
Z_{axis}	m	elevation of magnetic axis
κ	—	elongation
δ	—	triangularity
ΔR_{sep}	m	separatrix separation
β_p	—	poloidal beta
ℓ_i	—	internal inductance
Ψ_{ext}	Wb	external flux linkage
B_φ	T	toroidal field at R_{tor}
R_{tor}	m	radius for B_φ

Auxiliary parameters, listed in Table 2, define the computational domain and some graphics parameters.

Table 2: Auxiliary parameters

name	*units*	*description*
R_{min}	m	inner radial position of grid
R_{max}	m	outer radius of grid
Z_{min}	m	lower extent of grid
Z_{max}	m	upper extend of grid
N_R	—	number of radial grid points
N_Z	—	number of vertical grid points
r_{min}	m	inner radius of plot scale
r_{max}	m	outer radius of plot scale
z_{min}	m	lower range of plot scale
z_{max}	m	upper range of plot scale
J_{coil}	MA/m^2	scale factor for coil plots

There are some restrictions on the domain of the R-Z grid: $R_{min} > 0$ and $Z_{min} = -Z_{max}$. The number of grid points, N_R and N_Z, should each be restricted to $2^n + 1$ for efficiency in the Buneman solver, with the smallest practical value being $2^5 + 1 = 33$. Generally the number of grid points is chosen to yield similar resolution in the radial and vertical directions, so for elongated (∼2:1) grids a typical specification is $N_R \times N_Z = 33 \times 65$. There are no restrictions on the plot scale parameters except that $z_{min} = -z_{max}$ and $J_{coil} > 0$. This current density is used only in certain plot routines where the cross-sections of poloidal field coils are drawn in proportion to their current, with scale factor J_{coil}.

The tokamak parameters are placed in a text file of any name, but the dead-start procedure looks for the default name "`tokamak.inp`" if another name is not provided. The file will be read by the dead-start procedure using the Basis stream I/O facility, which, in this application, looks for one (or possibly two) input items on each line and when found, ignores the rest of the line. Blank lines and whitespace are ignored. The items must appear in the order shown below.

Format of `tokamak.inp` file

"identification"	
I_φ	
R_0	
a	
Z_{axis}	
κ	
δ	
ΔR_{sep}	
β_p	
ℓ_i	
Ψ_{ext}	
B_φ	R_{tor}
R_{min}	
R_{max}	
Z_{min}	
Z_{max}	
N_R	N_Z
r_{min}	
r_{max}	
z_{min}	
z_{max}	
J_{coil}	

The 1st non-blank line must contain a *quoted* problem identification string, which is used to construct a binary save-file name[1]. The identification string is also used in the Corsica problem identification contained in variable `probid`. Two lines in this file are assumed to contain two quantities: (1) the toroidal field specification has a mandatory radial position, R_{tor} [m], and (2) the line

[1] The save-file name is constructed from the *identification* string by converting all characters to lower-case, replacing space and "`/`" characters with "_", and appending "`.sav`".

containing the number of radial and vertical grid points.

It is recommended that the plasma major radius, R_0, the radius R_{tor} for B_φ and the center of the R-Z grid (code variable `ro`) have the *same* value, i.e.,

$$R_0 = R_{tor} = (R_{min} + R_{max})/2$$

to avoid confusion. Once an equilibrium is created by the dead-start procedure, it can be modified for arbitrary R_0 within the confines of the R-Z grid.

User comments and notes may appear after the values in the dead-start input file, as shown in the example input files in Appendix A.

The grid and graphics scale parameters may be omitted from the input file as shown in App. A.1, or just the graphics parameters omitted as shown in App. A.2, as default values will ge generated from the tokamak dimensions. A complete input file, containing: (1) tokamak parameters, (2) grid specifications, and (3) plot parameters, is listed in app. A.3.

3 Poloidal field coil specifications

Axisymmetric poloidal field (PF) coils in Corsica are characterized by the mean radius, R_c, and vertical position, Z_c, of the current centroid. The current is uniformly distributed in filamentary current loops arrayed over a rectangular or parallelogram cross-section of size $\Delta R_c \times \Delta Z_c$. The number of filaments in each coil element is $n_c = n_{\Delta R_c} \times n_{\Delta Z_c}$. The Corsica model for parallelogram cross-sections follows the EFIT convention[2]. Two types of parallelogram cross-section models are available, as shown in Figure 1. Type-1 parallelogram coils have angle $\alpha_c \neq 0$, and Type-2 parallelogram coils have angle $\alpha_{c2} \neq 0$. The mapping of the PF coil parameters to Corsica variable names is given in Table 3. Except for `nc`, all of these quantities are 1D arrays of length `nc`.

Table 3: PF coil parameters (see Fig. 1)

	Corsica		
quantity	*name*	*units*	*description*
N_c	`nc`	—	number of PF coils
R_c	`rc`	m	mean radius
Z_c	`zc`	m	vertical position
ΔR_c	`drc`	m	radial build
ΔZ_c	`dzc`	m	vertical build
α_c	`ac`	rad.	Type-1 inclination
α_{c2}	`ac2`	rad.	Type-2 inclination
n_{R_c}	`nrc`	—	filaments across ΔR_c
n_{Z_c}	`nzc`	—	filaments across ΔZ_c
—	`pfid`	—	coil name

Explicit PF coil specifications may be provided by the user for the dead-start

[2] See `http://web.gat.com/efit/`.

procedure, but if they are not provided a coil set will be automatically generated, as described in the following subsections.

3.1 Automatic PF coils

In some cases one may not care about the PF coil configuration, but they are necessary to construct a free-boundary equilibrium. If a coil specification file is not provided, the dead-start procedure will generate a set of up/down symmetric PF coil positions using:

$$R_c(i) = R_{0_c} + a_c \cos\Big(\theta(i) - \delta_c \sin\theta(i)\Big)$$

$$Z_c(i) = a_c \kappa_c \sin\theta(i)$$

$$\theta(i) = \frac{2\pi}{N_c}(i - \frac{1}{2})$$

for $i = 1, 2, \ldots N_c$ with N_c an *even* number. Note that a symmetric coil set can be used to generate an asymmetric plasma configuration (one where $\Delta R_{sep} \neq 0$) as long as N_c is sufficiently large. The Corsica variable names corresponding to these quantities are listed in Table 4. The user may specify any of the quantities listed in Table 4, or use the default values derived from the plasma shape. The generated coil specifications will be placed in the code variables listed in Table 3.

Table 4: Analytic PF coil parameters

coil		*variable*	*default*
parameter	*units*	*name*	*value*
N_c	—	`ncoil`	64
R_{0_c}	cm	`rocoil`	$(5R_0 - a)/4$
a_c	cm	`rbcoil`	$(3R_0 + a)/4$
κ_c	—	`elcoil`	$\max(1, \kappa)$
δ_c	—	`dcoil`	δ

3.2 User-provided PF coils

If PF coil specifications are known, they may be placed in a text file, with default file name "`pfcoil.inp`", in the following format.

Format of `pcoil.inp` file

"coil-id" descriptive information
N_c `coils`

`name`	`Rc [m]`	`Zc [m]`	`DRc [m]`	`Dzc [m]`	`n_turn`	`NI_cap`	`B_cap`
"name1"	$R_c(1)$	$Z_c(1)$	$\Delta R_c(1)$	$\Delta Z_c(1)$	$n_{turn}(1)$	$NI_{cap}(1)$	$B_{cap}(1)$
"name2"	$R_c(2)$	$Z_c(2)$	$\Delta R_c(2)$	$\Delta Z_c(2)$	$n_{turn}(2)$	$NI_{cap}(2)$	$B_{cap}(2)$
"name3"	$R_c(3)$	$Z_c(3)$	$\Delta R_c(3)$	$\Delta Z_c(3)$	$n_{turn}(3)$	$NI_{cap}(3)$	$B_{cap}(3)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
"nameN"	$R_c(N_c)$	$Z_c(N_c)$	$\Delta R_c(N_c)$	$\Delta Z_c(N_c)$	$n_{turn}(N_c)$	$NI_{cap}(N_c)$	$B_{cap}(N_c)$

The first line contains the coil set name (quoted) and some descriptive information which must be present but is not used, the 2nd line specifies the number of coils followed by the word "`coils`" and the 3rd line is a header for the coil specifications which follow. Each entry describes one coil and consists of the quoted coil name followed by its parameters.

These coil specifications may be up/down asymmetric.

The columns labeled "`n_turn`", "`NI_cap`" and "`B_cap`" contain, respectively, the number of turns, current-carrying capability and field capability for each coil. These three quantities are not used by Corsica, but are provided in the input file for use by the user (if the values are not known set them to unity). They will be available during a session under variable names: `ntc`, `cccap` and `bccap`. They must be present as each line must contain 8 items.

There is no provision to specify parallelogram coil cross-sections through the dead-start procedure, but once an equilibrium is created with rectangular coils, the inclination angles α_c and α_{c2} for coils with parallelogram cross-sections can be set by the user, followed by executing the equilibrium with the `run` command.

4 Dead-start procedure

Once the tokamak parameter file is ready (and perhaps a file describing PF coils), execute the dead-start procedure by invoking the function `tokamak_ds` or its synonym `ds`. The procedure accepts up to two file names, identifying the tokamak parameter file and, optionally, the PF coil specification file.

Dead-start procedure (`tokamak_ds` or synonym `ds`)...

`ds(`*`tokamak_parameter_file`*`[,`*`coil_specification_file`*`])`

The file name defaults are "`tokamak.inp`" and "`pfcoil.inp`", respectively. The dead-start-procedure contains a help message for use during a session:

```
ds("help")
```

There are three ways in which the dead-start procedure can be executed: (1) let the procedure lay out the PF coil configuration automatically, (2) auto-generate the coil set but with specified parameters for the analytic coil model, or (3) use explicit coil specifications from a file. The procedure executes Corsica with both the plasma and coil symmetry flags (`nsym` and `nsymc`) set to 2, so the equilibrium is computed assuming up/down *asymmetry*, even though the desired configuration may indeed be symmetric. If the coils are symmetric and it is intended that the plasma be symmetric, execute the `set_symmetric` routine *after* the dead-start procedure has completed, to *enforce* up/down symmetry (see §4.3). This has the benefit of significantly reducing the computational cost of future executions.

The three options described above are demonstrated in the following subsections, followed by a subsection describing how coil circuits are specified.

4.1 Automatic coil generation

To create a tokamak equilibrium using the automatic coil generation feature, execute the dead-start procedure with only the tokamak parameter file (see the example file in App. A.1) as input[3].

```
ds("circular1.inp")
```

The dead-start procedure goes through a series of four steps to produce its final equilibrium:

1. creates an initial equilibrium with simple profile models and crude shape constraints;
2. installs a set of "fuzzy-marker" points which are derived from the shape parameters in the input file;
3. installs an "ohmic" profile model; and
4. executes a "constrained equilibrium" problem, using the Corsica `ceq` package, to adjust ohmic profile parameters to achieve the desired values of ℓ_i and β_p as specified in the input file.

When the last step has been completed, the equilibrium is written to disk in a save-file named, in this case, "`circular1.sav`". At this point the user may modify the equilibrium and create additional save-files by executing

```
saveq(file_name)
```

By convention, the save-file name must end in "`.sav`".

A Corsica session is terminated by typing `quit` or `^D` (CTRL-D). In addition to the save-file(s), Corsica will write a session log file named *`pname`*`.log`, a graphics file *`pname`*`.`*`nnn`*`.ncgm`, and a graphics log file *`pname`*`.`*`nnn`*`.cgmlog`, where *`nnn`* is a sequence number.

To use the saved equilibria in a future session, launch the code with the save-file name on the command-line to avoid the time-consuming dead-start procedure:

```
caltrans [-probname pname] circular1.sav
```

or during a session, restore another equilibrium with:

```
restore "name.sav"
run
```

[3] This assumes a file named `pfcoil.inp` *does not* exist in the current working directory or Basis search path.

4.2 Automatic coil generation, with modifications

To modify the default analytic coil model, specify one or more of the parameters in Table 4 prior to executing the dead-start-procedure, for example:

```
ncoil=32
elcoil=1.5
dcoil=0.2
ds("circular1.inp")
```

4.3 PF coil specifications from a file

When the PF coil specifications are contained in a file, as in App. A.4, execute the dead-start procedure with that file name as the 2nd argument, for example:

```
ds("kstar-tokamak.inp","kstar-pfcoil.inp")
```

In this case the plasma is up/down symmetric as are the PF coils, so to *impose* symmetry in the equilibrium after the procedure has finished, execute:

```
set_symmetric
```

The `set_symmetric` routine will write its own save-file using "`-sym.sav`" as the suffix.

4.4 Coil circuits

The PF coils, whether generated automatically or via an input file, are initially configured in individual circuits by the dead-start procedure. The coil currents are contained in variable `cc`, an array of length `nc`. Array `ic`, also of length `nc`, contains the circuit indices for the coils, and is initialized to `iota(nc)` by the dead-start procedure. This setting allows each coil current to be individually varied to satisfy the plasma shape and any other constraints on the Grad-Shafranov problem.

Once an equilibrium is established, the circuits can be rearranged by the user by modifying `ic` and `cc` entries. To fix a particular coil current, set its `ic` entry to zero and set the corresponding `cc` entry to the desired NI value, in MA-turns. Remember that the `ic` list must be a *continuous sequence* of integers from one to the number of free coils, so if an element of `ic` is zero, subsequent elements must be decremented. To turn a coil off, use a *small* value for its current (the code uses `cc` in the denominator of certain expressions so a value of exactly zero will trigger a divide-by-zero error.)

To connect coils in series, make their `ic` entries the same, and initialize the corresponding `cc` entries to have the desired relationship. For example, if coils *i* and *j* (with $j > i$) are to be arranged in an anti-series configuration, then

```
ic(j)=ic(i)
cc(j)=-cc(i)
ic(j+1:nc)=ic(j+1:nc)-1
run
```

5 Additional input specifications

When a Corsica equilibrium exists (either from the dead-start procedure or from any other source), additional geometric structures may be added to the model. These include the definition of the first-wall/limiter/divertor configuration, passive structure which is used for vertical stability calculations, and the toroidal field coil geometry. If a specific plasma shape is desired, it may also be imported from a file. There are four auxiliary routines: `read_fwall`, `read_passive`, `read_tfcoil` and `read_shape` to import these elements, as defined in the following subsections. Sample input files are contained in Appendix B.

5.1 First-wall, limiter and divertor geometry

The geometry of the first-wall, limiter and/or divertor structures is contained in the Corsica `rplate(nplates,2)`, `zplate(nplates,2)` arrays, in centimeters, where the two pairs of coordinates

```
    rplate(i,1), zplate(i,1)
    rplate(i,2), zplate(i,2)
```

define the end-points of a straight-line segment for the `i`th element. These may be set explicitly by the user by first setting the number of plate elements, `nplates`, then filling the arrays.

The `read_fwall` routine is optionally available to read the wall data from a formatted text file, with default name "`fwall.inp`", containing the *R*-*Z* coordinates in *meters* of N_w=`nplates`+1 points defining a *continuous*, closed contour surrounding the plasma region.

Read first-wall/limiter/divertor configuration with...

`read_fwall(`*`first_wall_file`*`)`

Format of `fwall.inp` file

"fwall-id" *descriptive information*

N_w `points`

R [m]	Z [m]	Legend
$R(1)$	$Z(1)$	$k(1)$
$R(2)$	$Z(2)$	$k(2)$
$\vdots$	$\vdots$	$\vdots$
$R(N_w)$	$Z(N_w)$	$k(N_w)$

The first line contains a quoted name for the information followed by descriptive information, the 2nd line holds the number of points, the 3rd line is a header for the data, which is followed by N_w lines of *R*-*Z* coordinates and an integer legend code, k. The legend code is `0` to indicate that the point is on the first-wall, `1` to indicate the point is on the limiter surface, or `2` to indicate the point lies on the divertor structure. A *nominal* limiter-point may be identified with k =`-1`, which will be used to set the Corsica limiter-point coordinates `(rlim,zlim)`. The legend code is used by some of the plotting routines defined in Corsica script `graphics.bas` (see §7). A sample `fwall.inp` file is shown in App. B.1.

To read this file and map its contents to the 2D `rplate`, `zplate` arrays, execute the routine with a file like that in App. B.1.

```
read_fwall("fwall.inp")
```

The first-wall geometry variables, `nplates`, `rplate` and `zplate`, are contained in save-files, so once they have been defined they will persist. The legend code, k, however is not preserved, so one must re-read the `fwall.inp` file to take advantage of the (optional) legend code in graphical output.

5.2 Passive structure specifications

Passive structure models in Corsica are contained in so-called "wire" elements which use the same rectangular/parallelogram model as is used for the poloidal field coils (Figure 1). The Corsica passive structure variables are listed in Table 5. Except for `nwires`, all of these quantities are 1D arrays of length `nwires`.

Table 5: Passive structure definition

Corsica *name*	*units*	*description*
nwires	—	number of wire elements
rwires	m	mean radius
zwires	m	vertical position
drwires	m	radial dimension
dzwires	m	vertical dimension
awires	rad.	Type-1 inclination
awires2	rad.	Type-2 inclination
nrwires	—	filaments across drwires
nzwires	—	filaments across dzwires
rhwires	Ohm m	resistivity
idwires	—	element name

The user may set these quantities by first specifying `nwires` then filling all the arrays, or use the `read_passive` routine to import the passive structure definition from a text file, as described below.

`read_passive(`*`passive_structure_file`*`)`

Since the passive structure need not be poloidally continuous, the elements are represented by "plates" of rectangular or parallelogram section where the end-points: (R_1, Z_1) and (R_2, Z_2) of the element center-line are specified, along with its thickness, t, and resistivity, ρ.

Format of `passive.inp` file

```
"passive-id"   descriptive information
N segments
R1 [m]   Z1 [m]   R2 [m]   Z2 [m]   thk [m]   rho [Ohm m]
 R1(1)    Z1(1)    R2(1)    Z2(1)     t(1)       ρ(1)
 R1(2)    Z1(2)    R2(2)    Z2(2)     t(2)       ρ(2)
   ⋮        ⋮        ⋮        ⋮         ⋮          ⋮
 R1(N)    Z1(N)    R2(N)    Z2(N)     t(N)       ρ(N)
```

The 1st line contains the quoted name for the passive structure, followed by arbitrary (but mandatory) descriptive information. The 2nd line contains the number of segments, `nwires`, the 3rd line is a header for the wire parameters, which follow, one line per wire element.

A sample file is given in App. B.2, which is imported into the session with

```
read_passive("passive.inp")
```

Prepare a passive structure model for use by the Corsica vertical stability package by executing the `psm` (“passive-structure-model”) routine. This routine makes use of the information in `idwires` if it is available, but must be explicitly set by the user for each element. To see how `idwires` is used, look at the ITER passive structure model with:

```
caltrans iter.sav
idwires # look at the contents of idwires
psm # prepare the passive structure model
```

The passive structure specifications are preserved in save-files (including the `idwires` array) so once the equilibrium has been saved, there is no need to re-read the `passive.inp` file.

5.3 Toroidal field coil configuration

The vacuum toroidal field in an axisymmetric tokamak equilibrium is completely specified by the toroidal flux, $R_{tor}B_{\varphi}$. It is useful, however, to also have the geometric configuration of the toroidal field (TF) coils available to include in graphical output, and the centerline of the TF coil current may be used to evaluate $\mathbf{J}_{TF} \times \mathbf{B}_{pol}$ out-of-plane forces on the coils. The `read_tfcoil` routine can be used to import the R-Z coordinates of the inner periphery, (R_{TF_i}, Z_{TF_i}), outer periphery, (R_{TF_o}, Z_{TF_o}), and centerline, (R_{TF}, Z_{TF}).

```
read_tfcoil(tfcoil_file)
```

The format of the input file, with default name “`tfcoil.inp`”, consists of a file header with quoted name and arbitrary information, followed by three sets of coordinates as shown below. Note that each set consists of (1) descriptive header, (2) the number of points in the set, (3) a column header, and (4) the coordinates.

Format of `tfcoil.inp` file

"tfcoil-id"   descriptive information
TF Coil Inner Periphery
N_i points

R [m]	Z [m]
$R_{TF_i}(1)$	$Z_{TF_i}(1)$
$R_{TF_i}(2)$	$Z_{TF_i}(2)$
$\vdots$	$\vdots$
$R_{TF_i}(N_i)$	$Z_{TF_i}(N_i)$

TF Coil Outer Periphery
N_o points

R [m]	Z [m]
$R_{TF_o}(1)$	$Z_{TF_o}(1)$
$R_{TF_o}(2)$	$Z_{TF_o}(2)$
$\vdots$	$\vdots$
$R_{TF_o}(N_o)$	$Z_{TF_o}(N_o)$

TF Coil Centerline
N points

R [m]	Z [m]
$R_{TF}(1)$	$Z_{TF}(1)$
$R_{TF}(2)$	$Z_{TF}(2)$
$\vdots$	$\vdots$
$R_{TF}(N)$	$Z_{TF}(N)$

A sample input file is given in App. B.3, which is imported with

```
read_tfcoil("tfcoil.inp")
```

TF coil specifications are held in script variables and are not preserved in save-files; they must always be re-read if the information is needed.

5.4 Plasma shape

The fuzzy-marker points in Corsica are contained in 1D arrays `rfbd`, `zfbd` [cm] of length `nfbd`. They define a plasma boundary (and perhaps include separatrix strike-lines) for which ψ_{edge} is to be held approximately fixed with a weighting factor, in array `alfbd`. As with first-wall coordinates, they may be set directly by the user, or imported with the `read_shape` routine.

`read_shape(`*`shape_file`*`)`

The format of the input file is a list of R-Z coordinates with three header lines, as shown below.

Format of `shape.inp` file

"shape-id"   descriptive information
N points

R [m]	Z [m]
$R(1)$	$Z(1)$
$R(2)$	$Z(2)$
$\vdots$	$\vdots$
$R(N)$	$Z(N)$

A sample file is given in App. B.4, which is imported with

```
read_shape("shape.inp")
```

The user must explicitly assign values to the weight vector, `alfbd`. The plasma shape is preserved in save-files so there is no need to re-read the data.

6 Refining an equilibrium

Once a tokamak equilibrium has been created with the dead-start procedure, it can be modified in a variety of ways. As an example, we start with the equilibrium created by one of the circular samples, App. A.1, and modify the pressure and q-profile to create an equilibrium with $\beta_N = 1$, $q(0) = 1.1$ and $q(a) = 2.9$. Normalized beta is an equilibrium output quantity contained in Corsica variable `ctroy`. The q-profile is an output quantity contained in `qsrf(1:msrf)`, where `msrf` is the number of flux surfaces, so

$\beta_N \mapsto$ `ctroy`
$q(0) \mapsto$ `qsrf(1)`
$q(a) \mapsto$ `qsrf(msrf)`

We will use the constrained-equilibrium package, `ceq`, to solve three nonlinear equations satisfying the desired constraints, as follows:

```
caltrans circular1.sav
package ceq
nctot=3
vo=["ctroy","qsrf(1)","qsrf(msrf)"]
vo0=[1,1.1,2.9]
vi=["betaj","alfa(0)","plcm"]
x0=[betaj,alfa(0),plcm]
ihy=20; run
saveq("circular1-mod.sav")
```

Here, we launch the code using the previously saved equilibrium contained in `circular1.sav`. Next, make sure the code is using the `ceq` package, indicated by the prompt string “`ceq> `”—if not, issue the “`package ceq`” command. Specify the number of constraint equations in variable `nctot` (3 in this case), then specify the *names* of the three quantities to be constrained in array `vo` and their desired values in array `vo0`.

The names of the independent variables are prescribed in array `vi` and their initial values in `x0`. We choose to use the variable `betaj` to scale the pressure and hence β_N, and one of the profile parameters, `alfa(0)`, to affect $q(0)$. The constraint on $q(a)$ will be satisfied by varying the toroidal current, `plcm`. In specifying the initial guess, `x0`, it is recommended that their *present values* be used.

To execute the constrained equilibrium solver, specify an upper limit for the number of iterations, say `ihy=20`, and execute the solver with the `run` command. If it is successful, the equilibrium can be saved to disk for future use with the `saveq` routine.

7 Making plots

Corsica uses the Basis EZN package which provides flexible plotting capabilities, with graphical output going to NCGM files and X-windows for viewing during a session. Several plotting routines are defined by the `ploteq.ezn` and `graphics.bas` scripts, which are part of the Corsica distribution. The `ploteq.ezn` script is automatically read into each session, but the alternate `graphics.bas` must be explicitly read with a

```
read graphics.bas
```

statement. It provides some enhancement and flexibility to the default plotting routines. To get a list of the user-callable routines defined in `graphics.bas`, execute:

```
graphics
```

Some of the more useful `graphics.bas` routines are:

`coils`	display coil parameters
`colors`	display available colors and color-map table (for `cmap`)
`fonts`	display font information
`layout`	plot configuration
`pb`	plot plasma boundary and a few other things
`pbx`	plot all X-point surfaces
`pfbd`	plot annotated fuzzy-marker points
`profiles`	plot plasma profiles
`pls`	plot annotated plasma boundary

All of these routines will display a help message if invoked with argument: `"help"`. The `layout` and `profiles` routines are described below.

7.1 Configuration plots with `layout`

The `layout` routine makes a plot of the overall equilibrium configuration, including poloidal flux contours, PF coils, TF coils, first-wall/limiter/divertor geometry and passive structure.

```
layout(coil_style,show_legend)
```

Argument `coil_style` is an integer in {0,1,2,3,4} and `show_legend` an integer, either 0 or 1. The `coil_style` argument specifies how the PF coils are to be displayed:

`coil_style`	*shows PF coils with...*
0	actual cross-sections and coil names
1	cross-sections drawn in proportion to their current [default]
2	actual cross-sections and with filament distribution
3	actual cross-sections with coil numbers
4	like 2 but also shows R-Z grid lines

The `show_legend` argument specifies whether a list of parameters are to be included:

`show_legend`	*effect...*
`0`	do not show list of parameters
`1`	show parameters [default]

The `layout` routine has several other options, specified through global variables:

variable name	*description...*
`kextflux`	if `-1`, show ψ contours outside the plasma [default: `0`]
`ncplot`	show only the first `ncplot` coils [default: `nc`]
`nlevels`	show `nlevels` flux contours [default: `11`]
`solidCoils`	`true|false` to draw coils with colored cross-sections [default: `true`]
`solidPlasma`	`true|false` to show plasma flux with color contours [default: `false`]
`rclmin`	plot scale R_{min}, cm
`rclmax`	plot scale R_{max}, cm
`zclmin`	plot scale Z_{min}, cm
`zclmax`	plot scale Z_{max}, cm

The variable `ncplot` and the plot scale parameters (`rclmin`, etc.) are preserved in save-files. The other variables are defined in each session, when the script file `graphics.bas` is read into the session.

Sometimes it is desired to plot flux surfaces in the scrape-off layer (SOL) as opposed to setting `kextflux=-1`, which shows flux surfaces outside the confined plasma to the edge of the R-Z grid. The SOL flux surfaces are specified with `nsol`, the number of surfaces, and `solo(1:nsol)` values of ΔR_{sep}, in cm, the radial distance from the outboard edge of the plasma to each desired SOL surface.

Sample `layout` output is shown in Figure 2—it was created with

```
caltrans iter.sav
nsol=7; solo=iota(0,nsol-1)
read_fwall("fwall.inp")
read_tfcoil("tfcoil.inp")
layout(0,0)
```

using the input files given in App. B.1 and B.3.

The `layout` routine defined in `graphics.bas` specifies colors for certain components using symbolic names. To get a list of components and their current color setting, issue the `colors` command. This will also display a graphic of the available colors and their names (which are contained in Basis variable `color`). The user may change a component color using the color-mapping routine, `cmap`, which takes two arguments.

Change component colors for `layout` with...

`cmap(`*`component_name,color`*`)`

where `component_name` is the component name from the `colors` command and `color` is the color name from the Basis `color` list. Since the component names are sometimes difficult to associate with a particular element in the plot, one may need to refer to the `graphics.bas` script itself to determine the association.

7.2 Profile plots with `profiles`

The `profiles` routine plots various profile quantities, with the version defined in `graphics.bas` utilizing symbolic component colors accessible with `cmap`.

Plot plasma profiles with…

```
    profiles
```

Sample output is shown in Figure 3.

8 EQDSK files

Equilibria are often transported between various codes using EQDSK[4] files. Corsica writes such files, as well as a few other file formats, with the `weqdsk` routine.

Write EQDSK (and other) files with…

```
    weqdsk(eqdsk_type,eqdsk_suffix,time_units,time_fw)
```

argument	*description…*
eqdsk_type	character code(s) for file type(s) [default: "ag"]
eqdsk_suffix	file name suffix [default: "_teq" or "_inv_teq"]
time_units	units for time field, one of {"s", "ms", "us"} [default: *"ms"*].
time_fw	integer field width for the time entry [default: 5]

The *eqdsk_type* argument specifies the format of the file, the choice of which depends on the type of equilibrium (direct-solve or inverse) and the intended use of the file. The following file types are available.

Type code	*Equilibrium type*	*File description*
a	direct or inv.	auxiliary EQDSK file (EFIT a-file)
d	direct	for input to Alan Glasser's DCON code
g	direct or inv.	primary EQDSK file (EFIT g-file)
i	inverse	for input to Glasser's DCON
t	inverse	for input to Alan Turnbull's GATO code

The *eqdsk_type* can be a concatenation, e.g., "ag", of any relevant file type codes from the above list.

[4] The format of EQDSK files (often called a- and g-files) is described in `http://web.gat.com/efit/efit_outputs.html`.

The `eqdsk_suffix` is used to distinguish files written by Corsica from those written by EFIT, but can be any string (including `" "`, to omit the suffix).

The output files will have names constructed from the contents of the Corsica `shotName` and `shotTime` variables, with the format:

```
<eqdsk_type>shotName.<time_string><eqdsk_suffix>
```

The `shotName` variable is a character variable which may contain any string, usually the string representation of a shot number.

The `shotTime` variable contains a time point, in seconds. If the equilibrium pertains to an experimental shot, then this would be the time point in the scenario represented by the equilibrium. More generally, `shotTime` can be set to any number to affect the file name of the `weqdsk` output: it is used to construct the `time_string` using the units specified by `time_units` with a field width specified by `time_fw`.

An in-line help message may be displayed with

```
weqdsk("help")
```

Although EQDSK files are applicable for only free-boundary (direct-solve) equilibria, Corsica will compute a self-consistent flux a 2D grid, $\psi(R, Z)$, from an inverse equilibrium where $R(\psi, \theta), Z(\psi, \theta)$ are known, with the following procedure:

```
inv_k=0
teq_inv
get_vacflux
weqdsk
```

which will write a- and g-files.

9 Equilibria for DCON

The routines described in the previous sections are frequently used to prepare equilibria for use by Alan Glasser in developing and testing the MHD stability code, DCON. The equilibria are transmitted to Glasser[5] in the form of a free-boundary EQDSK files (a-file, g-file), Glasser's binary format for free-boundary equilibria (d-file) and a binary file containing the inverse equilibrium (i-file). A typical equilibrium specification from Glasser is:

$$\begin{array}{rcl}
a & = & 1 \\
R & = & 3 \\
\kappa & = & 1 \\
\delta & = & 0 \\
\beta_N & = & 1 \\
q(0) & = & 1.1 \\
q(a) & = & 2.9
\end{array}$$

[5] `<ahg@lanl.gov>`

An input file for the dead-start procedure is created, then a Corsica script as shown in Appendix C. It first executes the dead-start procedure then modifies the equilibrium to achieve the specified plasma parameters, which is saved in a free-boundary save-file, `test10.sav`, in this example. Some plots are made, then the a-, g- and d-files are written with `weqdsk`. An inverse solution is obtained using the `start_inv` routine and it is displayed with the `contour` graphics command then written to an i-file.

The equilibria are created by executing

```
caltrans test10.bas
```

In this case, Corsica will take the problem name string from the *basename* of the file named on the command line, i.e., *pname*="test10" in this case. This string will be used to set the `shotName` variable—used to name the output files from the `weqdsk` routine. The output files from this session are:

```
atest10.01000_teq
dtest10.01000_teq
gtest10.01000_teq
itest10.01000_inv_teq
test10.001.cgmlog
test10.001.ncgm
test10.log
test10.sav
tokamak_lim.sav
```

It is recommended that the graphics file be converted to PDF format. The `~bulmer/bin/ncgm2pdf` command[6] can be used for this purpose, as follows:

```
ncgm2pdf -o test10.pdf test10.001.ncgm
```

The relevant files (`*_teq` and `*.pdf`), along with a `README` file describing the equilibria, are then placed in a subdirectory with the same problem named and a tar-ball is created and e-mailed to Glasser.

[6] The `ncgm2pdf` command has several command-line options, execute "`ncgm2pdf -h`" for details.

Appendix

A Dead-start examples

The following subsections contain examples of using the dead-start procedure defined in script file `tokamak.bas`. The first three examples (App. A.1–A.3) are for generating tokamaks with circular cross-sections, and demonstrate the use of optional input quantities in the `tokamak.inp` file, but with the PF coil set generated automatically.

The next two cases (App. A.4–A.5) demonstrate placing PF coil specifications in a file and generating an up/down symmetric double-null configuration then an asymmetric single-null configuration.

A.1 Circular tokamak #1

This example represents the simplest kind of input for the dead-start procedure. Only the essential tokamak parameters are defined—the dead-start procedure will select an appropriate grid and graphics parameters, and the PF coil set will be generated automatically (with coil specifications derived from the plasma parameters) as described in §3.1.

File `circular1.inp`

```
"Circular1"

Plasma...
      0.50 MA       plasma current
      3.00 m        major radius
      1.00 m        minor radius
      0.00 m        Zaxis
      1.00          95% elongation
      0.00          95% triangularity
      0.00 m        Dsep (DN)
      0.25          poloidal beta
      1.00          li
      0.00 Wb       External flux linkage

Toroidal field...
      1.00 T @ R = 3.0 m
```

End file `circular1.inp`

To create an equilibrium with the above input file, execute:

```
caltrans [-probname pname] tokamak.bas
ds("circular1.inp")
<<< modify equilibrium as desired >>>
quit
```

A.2 Circular tokamak #2

This example has the same tokamak parameters as for the 1st circular tokamak case, but here the grid values have been specified and only the graphics parameters will be chosen automatically. Process it in the same way:

```
ds("circular2.inp")
```

File `circular2.inp`

```
"Circular2"

Plasma...
      0.50 MA       plasma current
      3.00 m        major radius
      1.00 m        minor radius
      0.00 m        Zaxis
      1.00          95% elongation
      0.00          95% triangularity
      0.00 m        Dsep (DN)
      0.25          poloidal beta
      1.00          li
      0.00 Wb       External flux linkage

Toroidal field...
      1.00 T @ R = 3.0 m

Computational grid (optional)...
      1.50 m        Rmin
      4.50 m        Rmax
     -1.50 m        Zmin
      1.50 m        Zmax
   65 x 65          No. grid points (Nr x Nz)
```

End file `circular2.inp`

A.3 Circular tokamak #3

This case has the same tokamak and grid parameters as the 2nd example, but here graphics parameters have been provided. Process it in the same way as above:

```
ds("circular3.inp")
```

File `circular3.inp`

```
"Circular3"

Plasma...
      0.50 MA       plasma current
      3.00 m        major radius
      1.00 m        minor radius
      0.00 m        Zaxis
      1.00          95% elongation
      0.00          95% triangularity
      0.00 m        Dsep (DN)
      0.25          poloidal beta
      1.00          li
      0.00 Wb       External flux linkage

Toroidal field...
      1.00 T @ R = 3.0 m

Computational grid (optional)...
      1.50 m        Rmin
      4.50 m        Rmax
     -1.50 m        Zmin
      1.50 m        Zmax
   65 x 65          No. grid points (Nr x Nz)

Plot Scales (optional)...
      0.00 m        Rmin
      5.00 m        Rmax
     -2.00 m        Zmin
      2.00 m        Zmax
      0.05 MA/m^2   Current density for drawing coil cross-sections
```

End file `circular3.inp`

A.4 Shaped double-null tokamak with coil specifications

This example demonstrates using PF coil specifications from a file to create a shaped double-null configuration. The equilibrium is generated with the dead-start procedure, then up/down symmetry is enforced with the `set_symmetric` routine, which creates a 2nd save-file.

```
ds("kstar-tokamak.inp","kstar-pfcoil.inp")
set_symmetric
```

Note the value of ΔR_{sep} (1×10^{-6}) to signal that the equilibrium is to be limited by an X-point. The elongation must be sufficiently large to create an X-point within the domain of the R-Z grid.

File `kstar-tokamak.inp`

```
"KSTAR/DN"

Plasma...
      2.00 MA       plasma current
      1.80 m        major radius
      0.50 m        minor radius
      0.00 m        Zaxis
      1.90          95% elongation
      0.30          95% triangularity
      1e-6 m        Dsep (DN)
      1.00          poloidal beta
      0.73          li
     -4.00 Wb       External flux linkage

Toroidal field...
3.50 T @ R = 1.80 m

Computational grid...
      1.00 m        Rmin
      2.60 m        Rmax
     -1.50 m        Zmin
      1.50 m        Zmax
 33 x 65            No. grid points (Nr x Nz)

Plot Scales...
      0.00 m        Rmin
      4.00 m        Rmax
     -2.50 m        Zmin
      2.50 m        Zmax
      2.00 MA/m^2   Current density for drawing coil cross-sections
```

End file `kstar-tokamak.inp`

The `n_turn`, `NI_cap` and `B_cap` entries in the following file are place-holders—these quantities are not used by Corsica to create the equilibrium.

File `kstar-pfcoil.inp`

```
"KSTAR" PF coil set of 05/01/99 from Kim
 14 coils
name    Rc [m]   Zc [m]  DRc [m]  DZc [m]   n_turn    NI_cap  B_cap
"PF1U"  0.5610   0.2470   0.2135   0.4764        1      1       1
"PF2U"  0.5610   0.6932   0.2135   0.3808        1      1       1
"PF3U"  0.5610   0.9960   0.2135   0.1896        1      1       1
"PF4U"  0.5610   1.2510   0.2135   0.2852        1      1       1
"PF5U"  1.0850   2.2960   0.3330   0.3808        1      1       1
"PF6U"  3.0900   1.9200   0.1896   0.3808        1      1       1
"PF7U"  3.7300   0.9600   0.1418   0.2852        1      1       1
"PF1L"  0.5610  -0.2470   0.2135   0.4764        1      1       1
```

```
"PF2L"  0.5610  -0.6932   0.2135   0.3808        1      1       1
"PF3L"  0.5610  -0.9960   0.2135   0.1896        1      1       1
"PF4L"  0.5610  -1.2510   0.2135   0.2852        1      1       1
"PF5L"  1.0850  -2.2960   0.3330   0.3808        1      1       1
"PF6L"  3.0900  -1.9200   0.1896   0.3808        1      1       1
"PF7L"  3.7300  -0.9600   0.1418   0.2852        1      1       1
```

End file `kstar-pfcoil.inp`

A.5 Shaped single-null tokamak

This example has the same plasma parameters as in App. A.4 except the magnitude of the ΔR_{sep} entry has been increased to -2 cm to create a single-null-bottom (SNB) configuration. The PF coil specifications are the same as defined in App. A.4. The equilibrium is generated with:

```
ds("kstar-tokamak-snb.inp","kstar-pfcoil.inp")
```

To obtain an asymmetric SNB solution, Corsica must remain in up/down asymmetric mode.

File `kstar-tokamak-snb.inp`

```
"KSTAR/SNB"

Plasma...
      2.00 MA       plasma current
      1.80 m        major radius
      0.50 m        minor radius
      0.00 m        Zaxis
      1.90          95% elongation
      0.30          95% triangularity
     -0.02 m        Dsep (DN)
      1.00          poloidal beta
      0.73          li
     -4.00 Wb       External flux linkage

Toroidal field...
3.50 T @ R = 1.80 m

Computational grid...
      1.00 m        Rmin
      2.60 m        Rmax
     -1.50 m        Zmin
      1.50 m        Zmax
 33 x 65            No. grid points (Nr x Nz)

Plot Scales...
      0.00 m        Rmin
      4.00 m        Rmax
     -2.50 m        Zmin
      2.50 m        Zmax
      2.00 MA/m^2   Current density for drawing coil cross-sections
```

End file `kstar-tokamak-snb.inp`

B Auxiliary input

The tokamak parameters and (optionally) PF coil specifications, represent the minimum input required to model a tokamak in Corsica. In most applications, one will also want to define geometric structures for a limiter surface, divertor and first-wall, and passive structure elements which are used when evaluating vertical stability. Additionally, one may specify the desired shape of the plasma boundary with a set of coordinates. The routines defined in the following subsections facilitate loading such information into a Corsica session.

B.1 First-wall, limiter and divertor structures

The file format for first-wall/limiter/divertor structures is described in §5.1. The file below specifies these components for ITER, and is loaded into a session with

```
caltrans [-probname pname] iter.sav graphics.bas
read_fwall("fwall.inp")
```

The geometry will be preserved in save-files in the `rplate`, `zplate` arrays, but the legend code will not persist. To use the legend code in graphical output (see §7.1), the file must be explicitly read into each session.

File `fwall.inp`

```
"ITER-FEAT/FWall 29.10.99" from Y. Gribov
        61 points
    R [m]      Z [m]    legend (0=FW, 1=LIM, 2=DIV)
    4.0486    -1.1873     0
    4.0649    -1.3139     0
    4.3598    -2.4412     0
    4.3368    -2.5342     0
    4.2462    -2.5653     0
    3.9650    -2.5533     2
    4.0636    -2.5700     2
    4.2914    -2.6700     2
    4.4448    -2.8659     2
    4.4873    -3.1111     2
    4.4086    -3.3472     2
    4.2398    -3.6118     2
    4.0709    -3.8775     2
    4.3419    -4.1218     2
    4.4240    -4.0967     2
    4.4766    -3.8857     2
    4.5235    -3.6971     2
    4.5557    -3.6596     2
    4.7613    -3.6169     2
    4.9679    -3.6541     2
    5.1456    -3.7658     2
    5.2687    -3.9360     2
    5.2671    -3.9854     2
    5.1584    -4.1866     2
    5.0422    -4.4017     2
    5.0730    -4.4738     2
    5.5649    -4.6087     2
    5.5645    -4.2543     2
    5.5635    -3.9009     2
    5.6114    -3.6484     2
    5.7495    -3.4317     2
    5.9582    -3.2817     2
    6.2077    -3.2199     2
    6.4623    -3.2551     2
    6.5489    -3.2845     2
    6.4445    -3.2121     0
    6.3792    -3.1402     0
    6.4056    -3.0467     0
    7.6829    -1.7572     0
    7.7866    -1.6037     0
    8.1525    -0.7572     0
    8.3242    -0.2491     0
    8.3242    -0.2491     1
    8.4107     0.2801     1
    8.4110     0.5000    -1 # Nominal limiter point
    8.4097     0.8164     1
    8.3211     1.3453     1
    8.3211     1.3453     0
    8.1344     1.9174     0
    7.8724     2.4592     0
    7.5399     2.9608     0
    7.1429     3.4131     0
    6.6887     3.8079     0
    5.7652     4.5071     0
    5.4587     4.6084     0
    4.9289     4.6034     0
    4.6727     4.5299     0
    4.4907     4.3353     0
    4.1057     3.6000     0
    4.0486     3.3681     0
    4.0486    -1.1873     0
```

End file `fwall.inp`

B.2 Passive structure

The file format for passive structure elements is described in §5.2. The file below specifies the passive structure (double-walled vacuum vessel) for ITER, and is loaded into a session with

```
read_passive("passive.inp")
```

Note that the user must explicitly set the `idwires` array, for (optional) use by the `psm` routine. The passive structure specifications are preserved in savefiles.

File `passive.inp`

```
"ITER-FEAT/PSM 01.11.99" from Y. Gribov
        115 segments
    R1 [m]     Z1 [m]     R2 [m]     Z2 [m]    thk [m] rho [Ohm m]
    3.5400     1.1300     3.5400     2.3300     0.0600    8.50E-07 Inner VV
    3.5400     2.3300     3.5600     3.5300     0.0600    8.50E-07 Inner VV
    3.5600     3.5300     3.5900     3.8600     0.0600    8.50E-07 Inner VV
    3.5900     3.8600     3.6900     4.1700     0.0600    8.50E-07 Inner VV
    3.6900     4.1700     3.8500     4.4600     0.0600    8.50E-07 Inner VV
    3.8500     4.4600     4.0600     4.7100     0.0600    8.50E-07 Inner VV
    4.0600     4.7100     4.3200     4.9100     0.0600    8.50E-07 Inner VV
    4.3200     4.9100     4.6100     5.0600     0.0600    8.50E-07 Inner VV
    4.6100     5.0600     4.9300     5.1400     0.0600    8.50E-07 Inner VV
    4.9300     5.1400     5.2600     5.1600     0.0600    8.50E-07 Inner VV
    5.2600     5.1600     5.5800     5.1200     0.0600    8.50E-07 Inner VV
    5.5800     5.1200     5.8900     5.0000     0.0600    8.50E-07 Inner VV
    5.8900     5.0000     6.1600     4.8400     0.0600    8.50E-07 Inner VV
    6.1600     4.8400     6.4600     4.6200     0.0600    8.50E-07 Inner VV
    6.4600     4.6200     6.7600     4.3900     0.0600    8.50E-07 Inner VV
    6.7600     4.3900     6.9900     4.2100     0.0600    8.50E-07 Inner VV
    6.9900     4.2100     7.3400     3.9200     0.0600    8.50E-07 Inner VV
    7.3400     3.9200     7.6200     3.6500     0.0600    8.50E-07 Inner VV
    7.6200     3.6500     7.8900     3.3300     0.0600    8.50E-07 Inner VV
    7.8900     3.3300     8.1400     3.0000     0.0600    8.50E-07 Inner VV
    8.1400     3.0000     8.3500     2.6400     0.0600    8.50E-07 Inner VV
    8.3500     2.6400     8.5400     2.2600     0.0600    8.50E-07 Inner VV
    8.5400     2.2600     8.6900     1.8700     0.0600    8.50E-07 Inner VV
    8.6900     1.8700     8.8200     1.4300     0.0600    8.50E-07 Inner VV
    8.8200     1.4300     8.9100     0.9060     0.0600    8.50E-07 Inner VV
```

```
8.9100     0.9060     8.9200     0.3320     0.0600    8.50E-07 Inner VV
8.9200     0.3320     8.8600    -0.1700     0.0600    8.50E-07 Inner VV
8.8600    -0.1700     8.6900    -0.7700     0.0600    8.50E-07 Inner VV
8.6900    -0.7700     8.6200    -0.9580     0.0600    8.50E-07 Inner VV
8.6200    -0.9580     8.4300    -1.4000     0.0600    8.50E-07 Inner VV
8.4300    -1.4000     8.2300    -1.8500     0.0600    8.50E-07 Inner VV
8.2300    -1.8500     7.9200    -2.5700     0.0600    8.50E-07 Inner VV
7.9200    -2.5700     7.6000    -3.3100     0.0600    8.50E-07 Inner VV
7.6000    -3.3100     7.4600    -3.6400     0.0600    8.50E-07 Inner VV
7.4600    -3.6400     7.2600    -4.0000     0.0600    8.50E-07 Inner VV
7.2600    -4.0000     6.9900    -4.3500     0.0600    8.50E-07 Inner VV
6.9900    -4.3500     6.7100    -4.6100     0.0600    8.50E-07 Inner VV
6.7100    -4.6100     6.3700    -4.8400     0.0600    8.50E-07 Inner VV
6.3700    -4.8400     6.0000    -5.0100     0.0600    8.50E-07 Inner VV
6.0000    -5.0100     5.6000    -5.1200     0.0600    8.50E-07 Inner VV
5.6000    -5.1200     5.1900    -5.1600     0.0600    8.50E-07 Inner VV
5.1900    -5.1600     4.9500    -5.1500     0.0600    8.50E-07 Inner VV
4.9500    -5.1500     4.8300    -5.1300     0.0600    8.50E-07 Inner VV
4.8300    -5.1300     4.4800    -5.0200     0.0600    8.50E-07 Inner VV
4.4800    -5.0200     4.1700    -4.8300     0.0600    8.50E-07 Inner VV
4.1700    -4.8300     3.9200    -4.5800     0.0600    8.50E-07 Inner VV
3.9200    -4.5800     3.7200    -4.2700     0.0600    8.50E-07 Inner VV
3.7200    -4.2700     3.6000    -3.9300     0.0600    8.50E-07 Inner VV
3.6000    -3.9300     3.5400    -3.5700     0.0600    8.50E-07 Inner VV
3.5400    -3.5700     3.5400    -2.5800     0.0600    8.50E-07 Inner VV
3.5400    -2.5800     3.5400    -1.3200     0.0600    8.50E-07 Inner VV
3.5400    -1.3200     3.5400    -0.0658     0.0600    8.50E-07 Inner VV
3.5400    -0.0658     3.5400     1.1300     0.0600    8.50E-07 Inner VV
3.2700     1.1500     3.2700     2.3500     0.0600    8.50E-07 Outer VV
3.2700     2.3500     3.2800     3.5500     0.0600    8.50E-07 Outer VV
3.2800     3.5500     3.3100     3.9200     0.0600    8.50E-07 Outer VV
3.3100     3.9200     3.4100     4.2800     0.0600    8.50E-07 Outer VV
3.4100     4.2800     3.5700     4.6100     0.0600    8.50E-07 Outer VV
3.5700     4.6100     3.7900     4.9200     0.0600    8.50E-07 Outer VV
3.7900     4.9200     4.0600     5.1700     0.0600    8.50E-07 Outer VV
4.0600     5.1700     4.3700     5.3800     0.0600    8.50E-07 Outer VV
4.3700     5.3800     4.7100     5.5300     0.0600    8.50E-07 Outer VV
4.7100     5.5300     5.0700     5.6100     0.0600    8.50E-07 Outer VV
5.0700     5.6100     5.4400     5.6300     0.0600    8.50E-07 Outer VV
5.4400     5.6300     5.8100     5.5900     0.0600    8.50E-07 Outer VV
5.8100     5.5900     6.1600     5.4800     0.0600    8.50E-07 Outer VV
6.1600     5.4800     6.5900     5.2800     0.0600    8.50E-07 Outer VV
6.5900     5.2800     7.0100     5.0400     0.0600    8.50E-07 Outer VV
7.0100     5.0400     7.3300     4.8200     0.0600    8.50E-07 Outer VV
7.3300     4.8200     7.6300     4.5900     0.0600    8.50E-07 Outer VV
7.6300     4.5900     7.8600     4.3900     0.0600    8.50E-07 Outer VV
7.8600     4.3900     8.1300     4.1100     0.0600    8.50E-07 Outer VV
8.1300     4.1100     8.4100     3.7900     0.0600    8.50E-07 Outer VV
8.4100     3.7900     8.6500     3.4700     0.0600    8.50E-07 Outer VV
8.6500     3.4700     8.8700     3.1300     0.0600    8.50E-07 Outer VV
8.8700     3.1300     9.0600     2.7700     0.0600    8.50E-07 Outer VV
9.0600     2.7700     9.2300     2.4100     0.0600    8.50E-07 Outer VV
9.2300     2.4100     9.3700     2.0300     0.0600    8.50E-07 Outer VV
9.3700     2.0300     9.5300     1.4300     0.0600    8.50E-07 Outer VV
9.5300     1.4300     9.6000     0.9120     0.0600    8.50E-07 Outer VV
9.6000     0.9120     9.6100     0.3320     0.0600    8.50E-07 Outer VV
9.6100     0.3320     9.5400    -0.2990     0.0600    8.50E-07 Outer VV
9.5400    -0.2990     9.4200    -0.7700     0.0600    8.50E-07 Outer VV
9.4200    -0.7700     9.2500    -1.2300     0.0600    8.50E-07 Outer VV
9.2500    -1.2300     9.0300    -1.7400     0.0600    8.50E-07 Outer VV
9.0300    -1.7400     8.8100    -2.2400     0.0600    8.50E-07 Outer VV
8.8100    -2.2400     8.6000    -2.7400     0.0600    8.50E-07 Outer VV
8.6000    -2.7400     8.3200    -3.3900     0.0600    8.50E-07 Outer VV
8.3200    -3.3900     8.1600    -3.7500     0.0600    8.50E-07 Outer VV
8.1600    -3.7500     7.9600    -4.1400     0.0600    8.50E-07 Outer VV
7.9600    -4.1400     7.7500    -4.4400     0.0600    8.50E-07 Outer VV
7.7500    -4.4400     7.4100    -4.8200     0.0600    8.50E-07 Outer VV
7.4100    -4.8200     7.0700    -5.0900     0.0600    8.50E-07 Outer VV
```

```
    7.0700    -5.0900     6.5900    -5.3700     0.0600    8.50E-07 Outer VV
    6.5900    -5.3700     6.2900    -5.4900     0.0600    8.50E-07 Outer VV
    6.2900    -5.4900     5.8700    -5.6000     0.0600    8.50E-07 Outer VV
    5.8700    -5.6000     5.4400    -5.6600     0.0600    8.50E-07 Outer VV
    5.4400    -5.6600     5.0000    -5.6500     0.0600    8.50E-07 Outer VV
    5.0000    -5.6500     4.6600    -5.5900     0.0600    8.50E-07 Outer VV
    4.6600    -5.5900     4.3300    -5.4700     0.0600    8.50E-07 Outer VV
    4.3300    -5.4700     4.0300    -5.2900     0.0600    8.50E-07 Outer VV
    4.0300    -5.2900     3.7800    -5.0600     0.0600    8.50E-07 Outer VV
    3.7800    -5.0600     3.5600    -4.7800     0.0600    8.50E-07 Outer VV
    3.5600    -4.7800     3.4100    -4.4700     0.0600    8.50E-07 Outer VV
    3.4100    -4.4700     3.3100    -4.1400     0.0600    8.50E-07 Outer VV
    3.3100    -4.1400     3.2700    -3.7900     0.0600    8.50E-07 Outer VV
    3.2700    -3.7900     3.2700    -2.4500     0.0600    8.50E-07 Outer VV
    3.2700    -2.4500     3.2700    -1.2500     0.0600    8.50E-07 Outer VV
    3.2700    -1.2500     3.2700    -0.0508     0.0600    8.50E-07 Outer VV
    3.2700    -0.0508     3.2700     1.1500     0.0600    8.50E-07 Outer VV
    8.2300    -1.8500     7.5200    -2.5700     0.0600    8.50E-07 Outer Wing
    7.5200    -2.5700     6.8100    -3.2800     0.0600    8.50E-07 Outer Wing
    6.8100    -3.2800     7.6000    -3.3100     0.0600    8.50E-07 Outer Wing
    3.5400    -2.5800     3.8700    -2.5800     0.0600    8.50E-07 Inner Wing
    3.8700    -2.5800     3.5400    -1.3200     0.0600    8.50E-07 Inner Wing
```

End file `passive.inp`

B.3 Toroidal field coils

The file format for the toroidal field coil geometry is described in §5.3. The file below specifies the TF coils for ITER, and is loaded into a session with

```
read_tfcoil("tfcoil.inp")
```

This information is used primarily by the `layout` routine and is not preserved in save-files—it must be read into each session if the TF coils are to be included in graphical output.

File `tfcoil.inp`

```
"ITER-FEAT/TFCoil 01.11.99" from Y. Gribov
TF Coil Inner Periphery
       102 points
    R [m]      Z [m]
    3.0750    -0.0192
    3.0750     0.2888
    3.0750     0.5967
    3.0750     0.9047
    3.0750     1.2126
    3.0750     1.5206
    3.0750     1.8285
    3.0750     2.1365
    3.0750     2.4445
    3.0750     2.7524
    3.0750     3.0604
    3.0750     3.3683
    3.0750     3.6763
    3.0843     3.9839
    3.1368     4.2871
    3.2345     4.5788
    3.3752     4.8524
    3.5556     5.1016
    3.7716     5.3207
    4.0183     5.5046
    4.2899     5.6492
    4.5802     5.7510
    4.8817     5.8131
```

5.1880	5.8427
5.4958	5.8398
5.8016	5.8042
6.1018	5.7364
6.3933	5.6374
6.6773	5.5184
6.9549	5.3851
7.2253	5.2378
7.4879	5.0770
7.7419	4.9030
7.9868	4.7163
8.2218	4.5174
8.4465	4.3068
8.6601	4.0850
8.8622	3.8527
9.0521	3.6103
9.2296	3.3586
9.3939	3.0983
9.5448	2.8299
9.6819	2.5541
9.8047	2.2718
9.9131	1.9835
10.0066	1.6901
10.0850	1.3924
10.1483	1.0910
10.1961	0.7868
10.2283	0.4806
10.2450	0.1731
10.2460	-0.1348
10.2313	-0.4424
10.2009	-0.7488
10.1551	-1.0533
10.0938	-1.3550
10.0172	-1.6533
9.9255	-1.9472
9.8190	-2.2362
9.6980	-2.5193
9.5627	-2.7959
9.4135	-3.0652
9.2507	-3.3267
9.0749	-3.5795
8.8865	-3.8230
8.6859	-4.0566
8.4737	-4.2797
8.2504	-4.4918
8.0166	-4.6922
7.7729	-4.8804
7.5200	-5.0560
7.2584	-5.2185
6.9890	-5.3675
6.7123	-5.5026
6.4290	-5.6234
6.1387	-5.7257
5.8394	-5.7975
5.5341	-5.8371
5.2264	-5.8442
4.9196	-5.8185
4.6173	-5.7606
4.3252	-5.6642
4.0509	-5.5249
3.8008	-5.3456
3.5807	-5.1307
3.3956	-4.8849
3.2498	-4.6140
3.1465	-4.3242
3.0883	-4.0221
3.0750	-3.7146
3.0750	-3.4067

```
    3.0750    -3.0987
    3.0750    -2.7908
    3.0750    -2.4828
    3.0750    -2.1749
    3.0750    -1.8669
    3.0750    -1.5590
    3.0750    -1.2510
    3.0750    -0.9430
    3.0750    -0.6351
    3.0750    -0.3271
    3.0750    -0.0192
TF Coil Outer Periphery
       101 points
    R [m]      Z [m]
    2.1690     0.0111
    2.1690     0.3809
    2.1690     0.7507
    2.1690     1.1204
    2.1690     1.4902
    2.1690     1.8600
    2.1690     2.2297
    2.1690     2.5995
    2.1690     2.9693
    2.1690     3.3390
    2.1690     3.7088
    2.1690     4.0786
    2.1690     4.4483
    2.2669     4.8046
    2.4078     5.1463
    2.5892     5.4682
    2.8084     5.7657
    3.0623     6.0342
    3.3470     6.2698
    3.6583     6.4689
    3.9915     6.6287
    4.3418     6.7466
    4.7069     6.7855
    5.0767     6.7855
    5.4464     6.7832
    5.8148     6.7527
    6.1785     6.6869
    6.5342     6.5865
    6.8797     6.4549
    7.2172     6.3040
    7.5463     6.1356
    7.8661     5.9499
    8.1755     5.7476
    8.4739     5.5292
    8.7603     5.2954
    9.0339     5.0467
    9.2939     4.7839
    9.5398     4.5078
    9.7706     4.2190
    9.9859     3.9184
   10.1850     3.6068
   10.3673     3.2852
   10.5324     2.9544
   10.6798     2.6153
   10.8090     2.2689
   10.9197     1.9161
   11.0116     1.5580
   11.0845     1.1955
   11.1380     0.8297
   11.1722     0.4616
   11.1868     0.0921
   11.1819    -0.2776
   11.1575    -0.6465
   11.1136    -1.0136
   11.0503    -1.3779
   10.9679    -1.7383
   10.8665    -2.0938
   10.7465    -2.4435
   10.6081    -2.7864
   10.4518    -3.1214
   10.2780    -3.4478
   10.0872    -3.7645
    9.8799    -4.0706
    9.6567    -4.3654
    9.4183    -4.6479
    9.1652    -4.9175
    8.8982    -5.1733
    8.6181    -5.4146
    8.3257    -5.6407
    8.0216    -5.8511
    7.7069    -6.0452
    7.3824    -6.2223
    7.0489    -6.3820
    6.7075    -6.5238
    6.3590    -6.6473
    6.0019    -6.7429
    5.6373    -6.8036
    5.2686    -6.8288
    4.8990    -6.8214
    4.5294    -6.8152
    4.1635    -6.7636
    3.8087    -6.6608
    3.4719    -6.5089
    3.1600    -6.3108
    2.8793    -6.0706
    2.6353    -5.7931
    2.4331    -5.4840
    2.2766    -5.1493
    2.1690    -4.7958
    2.1690    -4.4261
    2.1690    -4.0563
    2.1690    -3.6865
    2.1690    -3.3168
    2.1690    -2.9470
    2.1690    -2.5772
    2.1690    -2.2075
    2.1690    -1.8377
    2.1690    -1.4679
    2.1690    -1.0982
    2.1690    -0.7284
    2.1690     0.0111
TF Coil Centerline
        18 TF Coils
       202 points
    R [m]      Z [m]
    2.6543     0.0359
    2.6543     0.2059
    2.6543     0.3758
    2.6543     0.5458
    2.6543     0.7157
    2.6543     0.8857
    2.6543     1.0556
    2.6543     1.2255
    2.6543     1.3955
    2.6543     1.5654
    2.6543     1.7354
    2.6543     1.9053
    2.6543     2.0753
    2.6543     2.2452
    2.6543     2.4152
    2.6543     2.5851
    2.6543     2.7551
```

```
  2.6543     2.9250
  2.6543     3.0950
  2.6543     3.2649
  2.6543     3.4349
  2.6543     3.6048
  2.6543     3.7748
  2.6574     3.9447
  2.6717     4.1140
  2.6976     4.2819
  2.7349     4.4477
  2.7835     4.6105
  2.8432     4.7696
  2.9137     4.9242
  2.9945     5.0736
  3.0855     5.2171
  3.1861     5.3540
  3.2958     5.4838
  3.4142     5.6056
  3.5407     5.7191
  3.6747     5.8236
  3.8155     5.9187
  3.9625     6.0039
  4.1150     6.0788
  4.2723     6.1431
  4.4336     6.1964
  4.5982     6.2386
  4.7651     6.2705
  4.9334     6.2939
  5.1027     6.3088
  5.2725     6.3152
  5.4424     6.3130
  5.6120     6.3023
  5.7809     6.2830
  5.9485     6.2552
  6.1145     6.2190
  6.2785     6.1745
  6.4401     6.1218
  6.5990     6.0616
  6.7562     5.9970
  6.9116     5.9282
  7.0652     5.8555
  7.2168     5.7787
  7.3664     5.6981
  7.5138     5.6136
  7.6590     5.5252
  7.8018     5.4331
  7.9422     5.3373
  8.0800     5.2379
  8.2152     5.1350
  8.3477     5.0285
  8.4773     4.9186
  8.6041     4.8054
  8.7278     4.6889
  8.8485     4.5693
  8.9660     4.4465
  9.0803     4.3207
  9.1912     4.1920
  9.2988     4.0604
  9.4029     3.9261
  9.5034     3.7891
  9.6004     3.6495
  9.6936     3.5075
  9.7832     3.3630
  9.8689     3.2163
  9.9508     3.0674
 10.0288     2.9164
 10.1029     2.7634
 10.1729     2.6086
```

10.2388	2.4520
10.3007	2.2937
10.3584	2.1338
10.4119	1.9725
10.4612	1.8099
10.5063	1.6460
10.5470	1.4811
10.5835	1.3151
10.6156	1.1482
10.6433	0.9805
10.6667	0.8122
10.6856	0.6433
10.7002	0.4740
10.7103	0.3044
10.7161	0.1345
10.7174	-0.0354
10.7142	-0.2053
10.7067	-0.3751
10.6947	-0.5446
10.6783	-0.7138
10.6575	-0.8825
10.6323	-1.0505
10.6027	-1.2179
10.5688	-1.3844
10.5305	-1.5500
10.4880	-1.7145
10.4412	-1.8779
10.3901	-2.0400
10.3348	-2.2007
10.2754	-2.3599
10.2118	-2.5175
10.1442	-2.6734
10.0725	-2.8274
9.9968	-2.9796
9.9172	-3.1297
9.8337	-3.2777
9.7463	-3.4235
9.6552	-3.5670
9.5604	-3.7080
9.4619	-3.8465
9.3599	-3.9824
9.2543	-4.1156
9.1454	-4.2460
9.0330	-4.3735
8.9174	-4.4981
8.7986	-4.6196
8.6766	-4.7379
8.5516	-4.8530
8.4236	-4.9649
8.2928	-5.0733
8.1592	-5.1783
8.0229	-5.2798
7.8840	-5.3777
7.7426	-5.4720
7.5987	-5.5625
7.4526	-5.6493
7.3043	-5.7322
7.1538	-5.8112
7.0014	-5.8863
6.8470	-5.9574
6.6908	-6.0244
6.5330	-6.0873
6.3735	-6.1461
6.2126	-6.2007
6.0494	-6.2482
5.8841	-6.2875
5.7170	-6.3183
5.5485	-6.3406

```
    5.3792    -6.3543
    5.2093    -6.3595
    5.0394    -6.3560
    4.8699    -6.3440
    4.7012    -6.3234
    4.5338    -6.2942
    4.3681    -6.2567
    4.2045    -6.2108
    4.0434    -6.1566
    3.8854    -6.0941
    3.7327    -6.0197
    3.5866    -5.9329
    3.4482    -5.8343
    3.3184    -5.7247
    3.1982    -5.6047
    3.0883    -5.4751
    2.9894    -5.3369
    2.9024    -5.1910
    2.8277    -5.0384
    2.7658    -4.8802
    2.7173    -4.7174
    2.6824    -4.5511
    2.6613    -4.3825
    2.6543    -4.2128
    2.6543    -4.0428
    2.6543    -3.8729
    2.6543    -3.7029
    2.6543    -3.5330
    2.6543    -3.3630
    2.6543    -3.1931
    2.6543    -3.0231
    2.6543    -2.8532
    2.6543    -2.6832
    2.6543    -2.5133
    2.6543    -2.3433
    2.6543    -2.1734
    2.6543    -2.0034
    2.6543    -1.8335
    2.6543    -1.6636
    2.6543    -1.4936
    2.6543    -1.3237
    2.6543    -1.1537
    2.6543    -0.9838
    2.6543    -0.8138
    2.6543    -0.6439
    2.6543    -0.4739
    2.6543    -0.3040
    2.6543    -0.1340
    2.6543     0.0359
```

End file tfcoil.inp

B.4 Plasma shape

The file format for importing plasma shape coordinates is described in §5.4. The file below specifies the nominal plasma shape for ITER, and is loaded into a session with

```
read_shape("shape.inp")
```

The input coordinates are in units of meters, which are copied to Corsica's rfbd, zfbd arrays in centimeters, and are preserved in save-files. The user must set the appropriate weight vector in alfbd to use these shape coordinates.

File `shape.inp`

```
"ITER-FEAT/Shape (SNB) 12.04.00" from Y. Gribov
        80 points
    R [m]     Z [m]
    6.0094   -5.4566
    5.9133   -5.2071
    5.8119   -4.9570
    5.7104   -4.7219
    5.5798   -4.4308
    5.4460   -4.1396
    5.3113   -3.8486
    5.0811   -3.3421 X point
    4.9261   -2.9753
    4.8127   -2.6841
    4.7097   -2.3930
    4.6179   -2.1018
    4.5424   -1.8314
    4.4669   -1.5196
    4.3879   -1.1315
    4.3245   -0.7434
    4.2750   -0.3552
    4.2386    0.0329
    4.2191    0.3241
    4.2066    0.6152
    4.2005    1.0033 min. radius
    4.2044    1.2944
    4.2160    1.5855
    4.2365    1.8767
    4.2664    2.1677
    4.3080    2.4589
    4.3639    2.7500
    4.4448    3.0646
    4.5380    3.3322
    4.6829    3.6233
    4.8354    3.8218
    4.9330    3.9069
    5.0306    3.9681
    5.1283    4.0078
    5.2259    4.0335
    5.3236    4.0460
    5.4213    4.0476 max. height
    5.5188    4.0403
    5.6165    4.0252
    5.7142    4.0035
    5.9095    3.9441
    6.1976    3.8174
    6.3726    3.7204
    6.5930    3.5782
    6.7864    3.4293
    6.9836    3.2548
    7.1789    3.0534
    7.3511    2.8470
    7.4718    2.6844
    7.6181    2.4589
    7.7783    2.1677
    7.9098    1.8767
    8.0155    1.5855
    8.0969    1.2944
    8.1554    0.9973
    8.1801    0.8092
    8.1952    0.6152
    8.1995    0.4211 max. radius
    8.1930    0.2270
    8.1753    0.0329
    8.1464   -0.1611
    8.1059   -0.3552
    8.0222   -0.6463
    7.9094   -0.9374
    7.7648   -1.2294
    7.5855   -1.5196
    7.3647   -1.8108
    7.1789   -2.0175
    6.9836   -2.2066
    6.7629   -2.3930
    6.5015   -2.5870
    6.3001   -2.7209
    6.0316   -2.8782
    5.7142   -3.0486
    5.4569   -3.1693
    5.0811   -3.3421 X point
    4.5777   -3.5574
    4.3444   -3.6544
    4.0541   -3.7763
    3.6636   -3.9426
```

End file `shape.inp`

C Input files for DCON equilibria

The following file is typical for creating equilibria with the dead-start procedure for testing DCON as described in §9. It contains the same specifications as in App. A.3 except for the large number of grid-points (128×128).

File `test10.inp`

```
"Tokamak/LIM"

Plasma...
      0.50 MA       plasma current
      3.00 m        major radius
      1.00 m        minor radius
      0.00 m        Zaxis
      1.00          95% elongation
      0.00          95% triangularity
      0.00 m        Dsep (DN)
      0.25          poloidal beta
      1.00          li
      0.00 Wb       External flux linkage

Toroidal field...
      1.00 T @ R = 3.0 m

Computational grid (optional)...
      1.50 m        Rmin
      4.50 m        Rmax
     -1.50 m        Zmin
      1.50 m        Zmax
 128 x 128          No. grid points (Nr x Nz)

Plot Scales (optional)...
      0.00 m        Rmin
      5.00 m        Rmax
     -2.00 m        Zmin
      2.00 m        Zmax
      0.05 MA/m^2   Current density for drawing coil cross-sections
```

End file `test10.inp`

The free-boundary and inverse equilibria for DCON are refined with the following script. It is designed to be executed with:

```
caltrans test10.bas
```

as it reads the `tokamak.bas` script, then uses the basename from the file named on the command-line as the problem name string to set `shotName`, which is used in the file names created by `weqdsk`.

File `test10.bas`

```
# caltrans test10.bas

read tokamak.bas

# Variables used by weqdsk...
shotName=probname
shotTime=1

# Create a nominal direct-solve eq from scratch...
tokamak_ds(trim(probname)//".inp")

# Converge to q(0)=1.1, q(a)=2.9, betaN=1...
probid="DCON "//trim(probname)//" direct-solve"
nctot=3
vo=["ctroy","qsrf(1)","qsrf(msrf)"]
vo0=[1,1.1,2.9]
vi=["betaj","alfa(0)","plcm"]
x0=[betaj,alfa(0),plcm]
ihy=20; run

# Save the direct-solve eq...
saveq(trim(probname)//".sav")

# Make some plots...
kextflux=-1
layout(1,0)
zoom
layout
profiles

# Write direct-solve EQDSK files...
weqdsk("ag")  # Standard a- and g-files
weqdsk("d")   # Glasser binary format

# Make an inverse equilibrium...
probid="DCON "//trim(probname)//" inverse"
nht=500; epsrk=1.0e-08
start_inv
contour

# Write inverse-solve EQDSK...
weqdsk("i") # Glasser binary format

quit
```

End file `test10.bas`

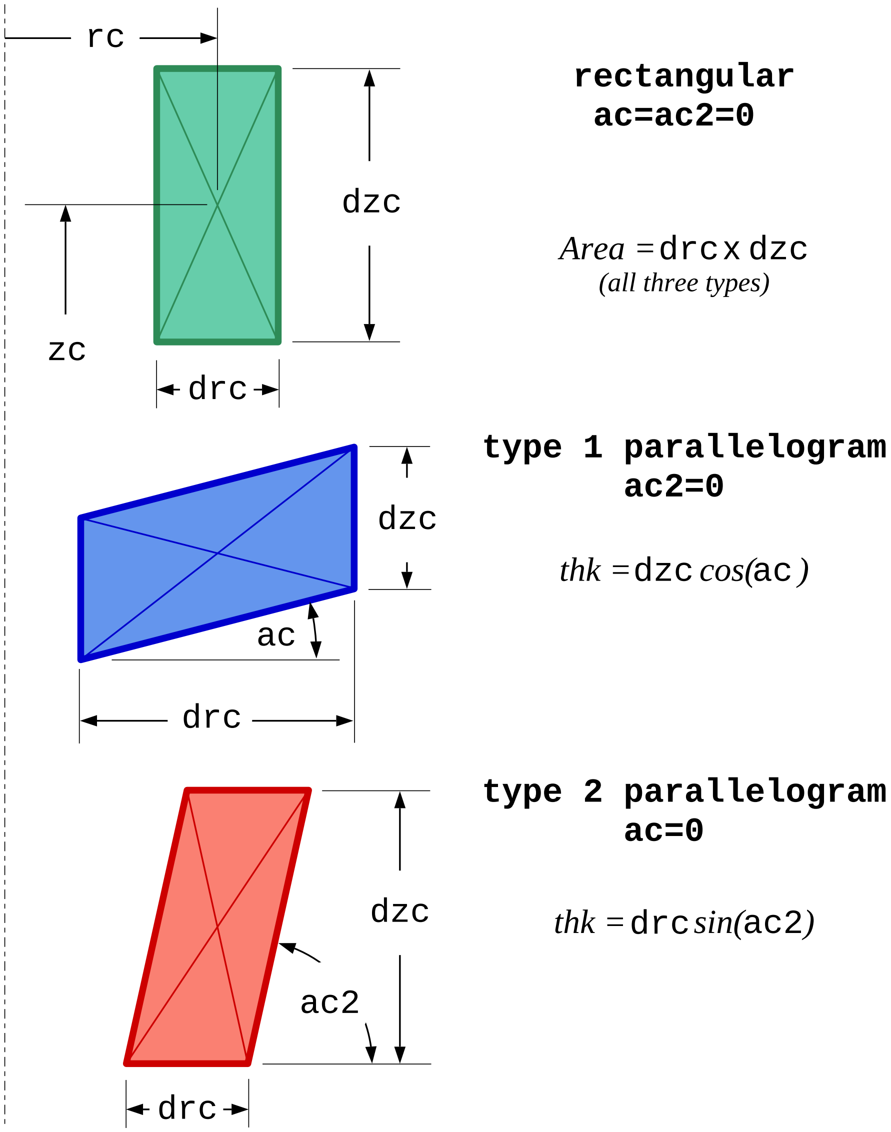

Figure 1: **Corsica PF coil geometric specifications**— conventional rectangular cross-sectional coils have $\alpha_c = \alpha_{c2} = 0$, Type-1 parallelogram coils have $\alpha_c \neq 0$, and Type-2 parallelogram coils have $\alpha_{c2} \neq 0$.

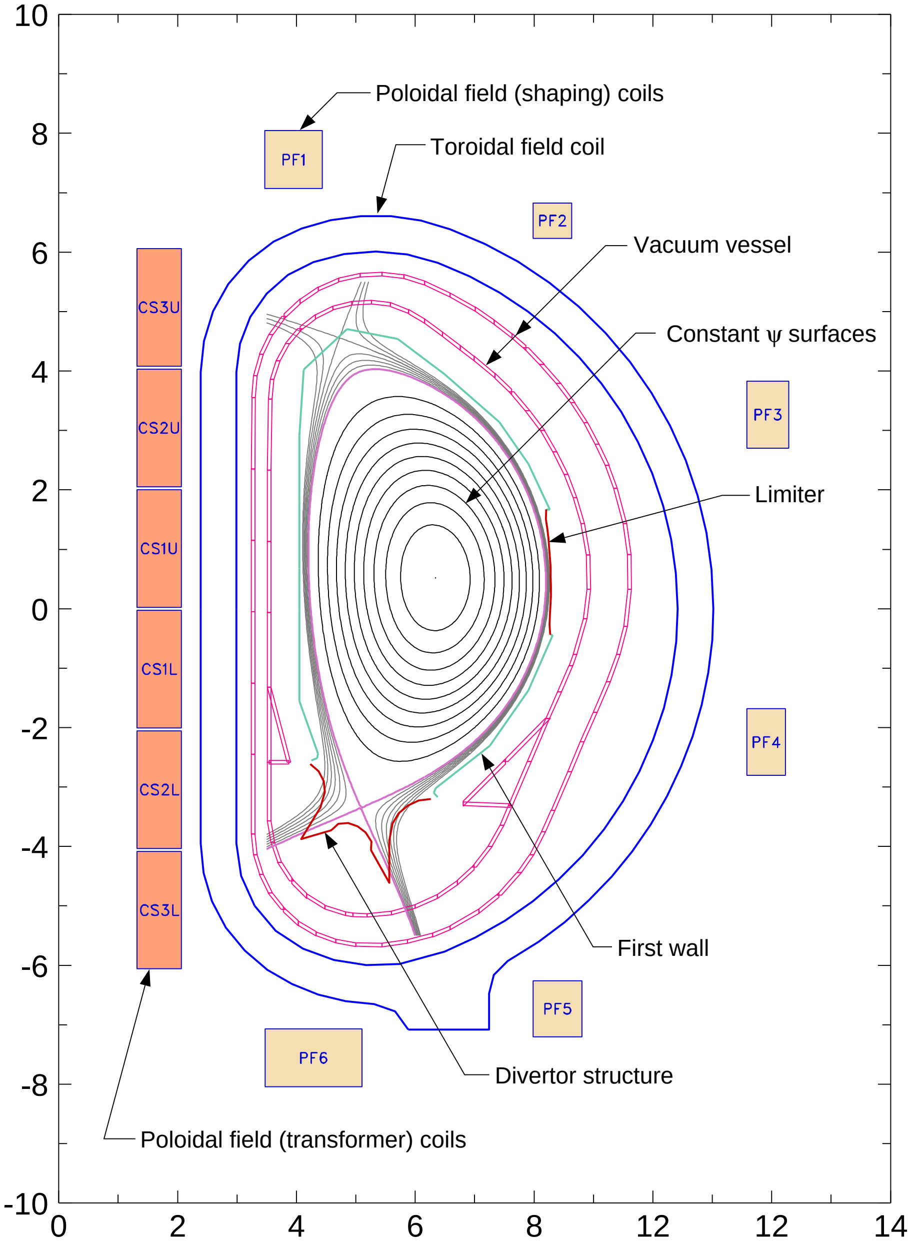

Figure 2: **Sample `graphics.bas:layout` output** (with added annotations and color changes for clarity). The first-wall/limiter/divertor, passive structure and TF coil specifications were imported into the Corsica model with the routines described in §5 using the data files listed in Appendix B.

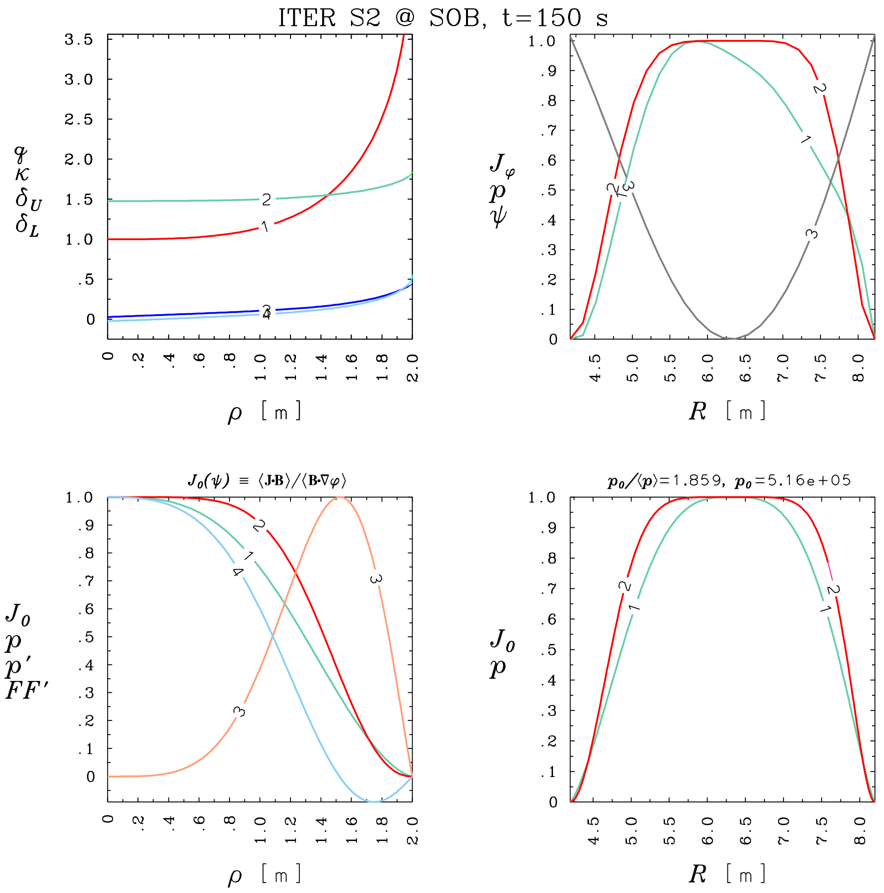

Figure 3: **Sample `graphics.bas:profiles` plot**. The two figures on the left show various quantities as a function of $\rho = a\sqrt{\tilde{\phi}}$, where $\tilde{\phi} = \frac{\Phi - \Phi_{axis}}{\Phi_{edge} - \Phi axis}$ and Φ is toroidal flux. The figures on the right show profile quantities as a function of major radius.