Informes Técnicos Ciemat 963
Mayo, 2001

TJ-II Library Manual

V. Tribaldos
B. Ph. van Milligen
A. López-Fraguas

Asociación EURATOM/CIEMAT para Fusión - 70

Departamento de Fusión y Física de Partículas Elementales

Toda correspondenica en relación con este trabajo debe dirigirse al Servicio de Información y Documentación, Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas, Ciudad Universitaria, 28040-MADRID, ESPAÑA.

Las solicitudes de ejemplares deben dirigirse a este mismo Servicio.

Los descriptores se han seleccionado del Thesauro del DOE para describir las materias que contiene este informe con vistas a su recuperación. La catalogación se ha hecho utilizando el documento DOE/TIC-4602 (Rev. 1) Descriptive Cataloguing On-Line, y la clasificación de acuerdo con el documento DOE/TIC.4584-R7 Subject Categories and Scope publicados por el Office of Scientific and Technical Information del Departamento de Energía de los Estdos Unidos.

Depósito Legal: M-14226-1995
ISSN: 1135-9420
NIPO: 402-01-008-3

Editorial CIEMAT

CLASIFICACIÓN DOE Y DESCRIPTORES

S70

COMPUTER PROGRAM DOCUMENTATION; COMPUTER CODES; FORTRAN; STELLARATORS; HELIAC STELLARATORS; TOKAMAK DEVICES; HELICAL CONFIGURTION; MAGNETIC FIELDS; PROGRAMMING.

TJ-II Library Manual. (Version 2)

Tribaldos, V.; Milligen, B. Ph. van; López-Fraguas, A.
45 pp. 0 fig. 3 refs.

Abstract:

This is a manual of use of the TJ2 Numerical Library that has been developed for making numerical computations of different TJ-II configurations.
This manual is a new version of the earlier manual CIEMAT report 806.

Manual de la Librería TJ-II. (Versión 2)

Tribaldos, V.; Milligen, B. Ph. van; López-Fraguas, A.
45 pp. 0 fig. 3 refs.

Resumen:

Este es un manual que describe el uso de la librería numérica TJ-II, que ha sido desarrollada para la realización de cálculos numéricos de distintas configuraciones magnéticas del dispositivo de Fusión TJ-II. Este documento reemplaza al de la versión anterior, publicada como Informe CIEMAT 806.

TJ-II Library Manual

Version 2.0, February 2000

V. Tribaldos, B. Ph. van Milligen and A. Lopez-Fraguas
Asociación EURATOM-CIEMAT para Fusión
Avenida Complutense 22, Madrid 28040, Spain

Abstract

This is a manual of use of the TJ2 Numerical Library that has been developed for making numerical computations of different TJ-II configurations.

1 Introduction

This Manual describes a numerical library called TJ-II. This library provides computational tools for calculating several quantities for TJ-II Stellarator configurations. Those quantities include the magnetic field components, the normalized magnetic flux at every point, the rotational transform, the volume enclosed by a certain magnetic flux, and the surface area of a given flux surface. The routines we are going to present are programmed using the FORTRAN 77 and FORTRAN 90 languages. The TJ-II library is a shared static library, this means that references found in the library are loaded into the executable image file at link time. Although this type of libraries makes the executable image file larger than non-shared, or dynamic, libraries it potentially makes codes to run faster since references to the library are not made at run-time. These routines have been compiled for both the CRAY J916 and the DEC Alpha 8400 computers, and must be linked to any program before calling them. Since the linking procedures for these computers are slightly different, some examples will be given in the last section. The test programs we will discuss are only available on those computers.

Another difference between the two computers is the way in which they treat the precision of data types. For the CRAY J916 system the single precision (REAL*4) is a 64 bit variable giving a range from $10^{-1200} < R < 10^{1200}$, whereas in the DEC Alpha 8400 system single precision is a 32 bit variable ranging from $10^{-30} < R < 10^{30}$. For this reason in the Alpha system the library is compiled in double precision, to achieve a range $10^{-300} < R < 10^{300}$ (IEEE standard). The calling programs in the DEC Alpha 8400 computer

must be in double precision, otherwise the results can be unpredictable. In the rest of this manual the term REAL should be understood as REAL*4 in the CRAY J916 system and REAL*8 in the DEC Alpha 8400.

1.1 Intended Audience

This manual is intended for programmers who already have a basic understanding of the FORTRAN language. Since it is developed for execution on UNIX like computers readers should also be familiar with operating system shell commands and text editor, such as *vi* , *edt* or *emacs*.

1.2 General Considerations

Generally the computation of most physical quantities in the plasmas found in thermonuclear fusion devices can be derived from two variables, namely: the magnetic field vector **B** and the normalized magnetic flux ψ . Since the density and temperature are assumed to be constant on the magnetic surfaces they are functions of the normalized magnetic flux ψ, which is equal to zero at the axis and one at the edge by definition. Therefore, throughout this library there will be no mention of the density and temperature. If one wants to compute those quantities the easiest way is to choose a dependence of the type: $n(\psi) = n(\psi_{axis})(1-\psi^{\alpha_1})^{\beta_1}$ and $T(\psi) = T(\psi_{axis})(1-\psi^{\alpha_2})^{\beta_2}$. Another quantity often used in plasmas is the effective radius. Although it is well defined for circular plasmas, in the case of the TJ-II Stellarator, due to its bean-shaped cross section, its definition is less intuitive. The normalized effective radius, r_{eff} , is defined by $r_{eff} \equiv \sqrt{\psi}$. The use of the name radius is common but can be misleading.

In any of the magnetic configurations of TJ-II there is a one-to-one correlation between the currents flowing through the coils and the magnetic surfaces, except for a scaling factor of the magnetic field, i.e. if all the currents are scaled by the same factor the magnetic surfaces remain the same, and only the strenght of the magnetic field changes by the same factor. Not all the possible configurations of TJ-II are included, mainly because of the work involved in the calculation of the equilibrium. However, in this new version of the Library a new routine has been included that allows the calculation of the magnetic field structure for any configuration by specifying the

coil currents. Please note that in this case the flux-related quantities (normalized flux, rotational transform, volume,...) are not necessarily consistent with the currents specified.

In this library the calculation of the magnetic field B has been done using the Biot-Savart law [1]. This means that in its present form it is only possible to compute the magnetic field for vacuum conditions ($\beta = 0$). In the near future it will also be possible to compute the magnetic field for non-zero beta conditions using a different method.

The computation of the normalized magnetic flux is performed in a different way with respect to the previous version of the Library. In the former version the flux calculations have been performed with a neural network fit of the magnetic flux surfaces [2]. Although this method has shown to be accurate, giving RMS. errors less than 0.5%, and very quick for evaluation, the preparatory work, involving a fitting procedure, is very time-consuming.

To better understand the difference between the calculations let us remember that the information about the magnetic flux is obtained from a magnetohydrodynamic code, the VMEC code. The information about the magnetic flux in this equilibrium in this code consist of two sets of coefficients, namely:

$$R(\psi) = \sum_{m}^{M} \sum_{n}^{N} R_{m\,n}(\psi) \cos(n\,\theta - m\varphi)$$

$$Z(\psi) = \sum_{m}^{M} \sum_{n}^{N} Z_{m\,n}(\psi) \sin(n\,\theta - m\varphi)$$

Thus the real problem of computing the magnetic flux for a given position is the inversion of the above equations. In the the first version of the Library this was accomplished by fitting the dependence, and in this second version by solving the above equations directly.

The latter method has shown to be more precise but at the same time slower (by approximately a factor of ten, depending on the number of moments). The benefit is twofold:

1) the precision is increased considerably (by more than a factor of ten)

2) there is no need for expensive and cumbersome neural network fits of the flux

The procedure for calculating the inversion of coordinates is as follows:

The flux dependence of the coefficients R_{mn} and Z_{mn} is approximated by a polynomial: $R_{mn} = R_{mn}(\psi)$ and $Z_{mn} = Z_{mn}(\psi)$. For those coefficients with $m \neq 0$ a multiplying factor of the form $\psi^{m/2}$ has to be added to have the correct dependence when ψ tends to zero. When a given point in space $r = (R, \phi, Z)$ is requested the equation

$$[R - R(\psi)]\,[Z - Z(\psi)] = 0$$

which is highly non-linear in ψ, is solved using a modified version of the Newton-Raphson method [3] in several dimensions. A large effort has been made to reduce the computing time necessary for solving the equation as much as possible. It has been verified that the mean number of times the complete series, giving R and Z, has to be summed is approximately 8 for an arbitrary point inside any configuration.

In this new version of the Library some other quantities have been added or modified, including:

1. A new method for computing the magnetic axis, consisting in computing the axis from the polynomial fit to the original VMEC coefficients of the axis.

2. The rotational transform ι at ψ

3. The volume enclosed by the magnetic flux ψ

4. The surface enclosed by the magnetic flux ψ

5. Routines that compute the location of the vacuum vessel

6. Routines to generate contour levels of constant magnetic flux, i.e. flux surfaces

7. A routine to generate the magnetic field corresponding to any combination of coil currents

As usual the names of the routines indicate the quantities they compute and the co-ordinate system that is used, either Cartesian or cylindrical.

1.3 Structure of this Document

This manual document consist of the following parts:
Sec. 2 Routines : Summarizes the routines and gives a detailed explanation of their arguments and use.
Sec. 3 Examples : Presents a sample FORTRAN program that makes use of some of the routines, and explains how to link the necessary files for running a user program with the Library.

1.4 Conventions Used

This manual uses the conventions listed in Table 1

init_tj2_lib	The bold format is used for the name of routines
UPPERCASE lowercase	The operating system shell differentiates between lowercase and uppercase characters. Literal strings that appear in text, examples, syntax descriptions, and function definitions must be typed exactly as shown
REAL	This term refers to the REAL*4 floating-point data types for the CRAY J916 system and REAL*8 floating-point data types for the Alpha 8400 computer.
INTEGER	This term refers to the INTEGER*4 data types for the CRAY J916 system and INTEGER*8 data types for the Alpha 8400 computer.
CRAY	This refers to the CRAY J916 system
ALPHA	This refers to the DEC Alpha 8400 system

Table 1: Convention Meaning

2 Routines

The structure of this library is a FORTRAN 77 and FORTRAN 90 source code of approximately 4000 lines consisting in 50 internal routines (of which approximately 20 are accessible to the user). **Throughout this library MKS units are used, i.e. the co-ordinates are given in meters and radians, the magnetic field is given in Tesla, coil currents in kA, and the volumes and surfaces are given in cubic and squared meters respectively.** To use the routines the library must be linked with the users code user_code.f . A detailed explanation of how to do this is shown in the examples section.

In order to give their results two input files are needed. One with the currents flowing trough the coils and another with the information about the configuration. To start the calculations an initialization routine called init_tj2_lib with the names of these files must be called. The library uses the FORTRAN units 44 and 45 for these files, make sure that your own code does not use these units.

The number of configuration files is limited since their creation requires

the solution of the equilibrium. It may nevertheless be of interest to know the value of the magnetic field in some cases for which no configuration file is available. Therefore, a subroutine is provided that allows the calculation of any possible magnetic field in TJ-II (subroutine create_fieldfile). However, the user of this routine should be aware that the normalization of the magnetic field for experimental conditions is not trivial and should therefore not use this routine if he is not familiar with this fact.

In the CRAY system the file /fusion/publica/tj2lib/00readme contains information about the available configurations and the name of the files with the parameter fits and the coil currents. In the ALPHA system the equivalent file can be found in /usr/users2/neural/tj2lib/00readme.

Please notice that in the ALPHA system the location of the Library has been changed from the old direction

In Table 2, a list of the available subroutines is given with a short description of their use. The table shows the following information:

1. Routine name: The name of the routine, an effort has been made to choose names that indicate, as clearly as possible, what the function of each routine is.

2. Arguments: Number of arguments and its data types (C for CHARACTER, I for INTEGER and R for REAL)

3. Description: Short description of what the routine computes.

Routine name	Arguments	Description
create_fieldfile	4, R; 1, C	Creates a new field initialization file
init_tj2_lib	2, C	Initializes the library. The arguments are the filenames of the input coil currents and the coefficients of the neural network
flux_cyl	4, R	Returns ψ at $r = (R, \phi, Z)$
flux_car	4, R	Returns ψ at $r = (X, Y, Z)$
grad_flux_cyl	6, R	Returns $\nabla_{cyl}\psi$ at $r = (R, \phi, Z)$
grad_flux_car	6, R	Returns $\nabla_{car}\psi$ at $r = (X, Y, Z)$
flux_surf_cyl	5, R	Returns $r = (R, \phi, Z)$ at magnetic coordinate point (ψ, θ, ϕ)
flux_surf_car	6, R	Returns $r = (X, Y, Z)$ at magnetic coordinate point (ψ, θ, ϕ)
b_field_cyl	6, R	Returns $B = (B_R, B_\phi, B_Z)$ at $r = (R, \phi, Z)$
b_field_car	6, R	Returns $B = (B_X, B_Y, B_Z)$ at $r = (X, Y, Z)$
grad_b_cyl	6, R	Returns $\nabla_{cyl}\|B\|$ at $r = (R, \phi, Z)$
grad_b_car	6, R	Returns $\nabla_{car}\|B\|$ at $r = (X, Y, Z)$
find_axis_cyl	4, R; 1, I	Returns $r_{axis} = (R, \phi, Z)$ for a given ϕ
find_axis_car	5, R; 1, I	Returns $r_{axis} = (X, Y, Z)$ for a given ϕ
iota	2, R	Returns the rotational transform for a given ψ
iota_cyl	4, R	Returns the rotational transform at $r = (R, \phi, Z)$
iota_car	4, R	Returns the rotational transform at $r = (X, Y, Z)$
volume	2, R	Returns the volume enclosed at given ψ
volume_cyl	4, R	Returns the volume enclosed at $r = (R, \phi, Z)$
volume_car	4, R	Returns the volume enclosed at $r = (X, Y, Z)$
surface	2, R	Returns the surface enclosed at given ψ
surface_cyl	4, R	Returns the surface enclosed at $r = (R, \phi, Z)$
surface_car	4, R	Returns the surface enclosed at $r = (X, Y, Z)$
tj2	3, R; 1, I	Returns a section of the vacuum vessel at given ϕ
tj2_cut	9, R; 1, I	Returns a section of the vacuum vessel with an arbitrary plane

Table 2: List of Routines

Name

create_fieldfile - Creates a new field initialization file

Synopsis

call create_fieldfile (Icc, Ihc, Ivf, Itf, "file2")

Description

Variable	Type	input/output	Description
Icc	REAL	Input	Current in the circular central coil, in kA.
Ihc	REAL	Input	Current in the helical central coil, in kA.
Ivf	REAL	Input	Current in the vertical field coil, in kA.
Itf	REAL	Input	Current in the toroidal field coil, in kA.
file2	CHARACTER	Input	name of the file containing the namelist with the currents flowing through the different coils of the device

Notes

(1) **Caution:** The use of this routine by users not familiar with the TJ-II magnetic coil system and field normalization **is not recommended.**

(2) The field generated is that which corresponds to the specified currents. The generated field is not normalized (e.g. to guarantee ECRH resonance).

(3) The user must have write permission for the directory specified through "file2" (when no path is included in this filename, the file "file2" is created in the current working directory).

(4) **After** creating the file "file2" using this subroutine, "init_tj2_lib" must be called with "file2" as its second argument.

(5) Note that, in general, no configuration file will be available that corresponds to the generated field.

(6) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8

Name

init_tj2_lib - Initializes the Library

Synopsis

call init_tj2_lib ("file1", "file2")

Description

Variable	Type	input/output	Description
file1	CHARACTER	Input	name of the file containing the coefficients of a given configuration
file2	CHARACTER	Input	name of the file containing the namelist with the currents flowing trough the different coils of the device

Notes

(1) Throughout the whole library the units used for these files are 44 for file1 and 45 for file2. Please check that your own code does not uses these units.

(2) In the CRAY system the file /fusion/publica/tj2lib/00readme contains information about the available configurations and the name of the files with the parameter fits and the coil currents. In the ALPHA system the equivalent file can be found in /usr/users2/neural/tj2lib/00readme.

Name

flux_cyl - Computes the normalized magnetic flux ψ for a point given in cylindrical co-ordinates (R, ϕ, Z)

Synopsis

call flux_cyl (R, Fi, Z, flux)

Description

Variable	Type	input/output	Description
R	REAL	Input	R co-ordinate, in meters.
Fi	REAL	Input	ϕ co-ordinate, in radians.
Z	REAL	Input	Z co-ordinate, in meters.
flux	REAL	Output	Normalized magnetic flux ψ

Notes
Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8

Name

flux_car - Computes the normalized magnetic flux ψ for a point given in Cartesian co-ordinates (X, Y, Z)

Synopsis

call flux_car (X, Y, Z, flux)

Description

Variable	Type	input/output	Description
X	REAL	Input	X co-ordinate, in meters.
Y	REAL	Input	Y co-ordinate, in meters.
Z	REAL	Input	Z co-ordinate, in meters.
flux	REAL	Output	Normalized magnetic flux ψ.

Notes

Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8

Name

grad_flux_cyl - Computes the derivatives of normalized magnetic flux $\partial\psi/\partial R$, $\partial\psi/\partial\phi$, $\partial\psi/\partial Z$ for a point given in cylindrical co-ordinates (R, ϕ, Z)

Synopsis

call grad_flux_cyl (R, Fi, Z, dfluxdR, dfluxdfi, dfluxdZ)

Description

Variable	Type	input/output	Description
R	REAL	Input	R co-ordinate, in meters.
Fi	REAL	Input	ϕ co-ordinate, in radians.
Z	REAL	Input	Z co-ordinate, in meters.
dfluxdR	REAL	Output	$\partial\psi/\partial R$, in m^{-1}.
dfluxdFi	REAL	Output	$\partial\psi/\partial\phi$, in rad^{-1}.
dfluxdZ	REAL	Output	$\partial\psi/\partial Z$, in m^{-1}.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.
(2) Remember that the gradient in cylindrical co-ordinates can be easily computed from the above quantities as $\nabla\psi = (\partial\psi/\partial R, 1/R\partial\psi/\partial\phi, \partial\psi/\partial Z)$. Please note the factor $1/R$ multiplying the derivative with respect to ϕ.

Name

grad_flux_car - Computes the derivatives of normalized magnetic flux $\partial\psi/\partial X$, $\partial\psi/\partial Y$, $\partial\psi/\partial Z$ for a point given in Cartesian co-ordinates (R, Y, Z)

Synopsis

call grad_flux_car (X, Y, Z, dfluxdX, dfluxdY dfluxdZ)

Description

Variable	Type	input/output	Description
X	REAL	Input	X co-ordinate, in meters.
Y	REAL	Input	Y co-ordinate, in meters.
Z	REAL	Input	Z co-ordinate, in meters.
dfluxdX	REAL	Output	$\partial\psi/\partial X$, in m^{-1}.
dfluxdY	REAL	Output	$\partial\psi/\partial Y$, in m^{-1}.
dfluxdZ	REAL	Output	$\partial\psi/\partial Z$, in m^{-1}.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

flux_surf_cyl - Computes the cylindrical co-ordinates (R, ϕ, Z) corresponding to the magnetic flux co-ordinates (ψ, θ, ϕ)

Synopsis

call flux_surf_cyl (flux, theta, Fi, R, Z)

Description

Variable	Type	input/output	Description
flux	REAL	Input	Normalized magnetic flux ψ.
theta	REAL	Input	θ co-ordinate, in radians.
Fi	REAL	Input	ϕ co-ordinate, in radians.
R	REAL	Output	R co-ordinate, in meters.
Z	REAL	Output	Z co-ordinate, in meters.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.
(2) Use this function to obtain a contour of a given flux surface ($\psi = cst.$), e.g. in a toroidal cross section $\phi = cst.$, by letting θ run from 0 to 2π.

Name

flux_surf_car - Computes the Cartesian co-ordinates (X, Y, Z) corresponding to the magnetic flux co-ordinates (ψ, θ, ϕ)

Synopsis

call flux_surf_car (flux, theta, Fi, X, Y, Z)

Description

Variable	Type	input/output	Description
flux	REAL	Input	Normalized magnetic flux ψ.
theta	REAL	Input	θ co-ordinate, in radians.
Fi	REAL	Input	ϕ co-ordinate, in radians.
X	REAL	Output	X co-ordinate, in meters.
Y	REAL	Output	Y co-ordinate, in meters.
Z	REAL	Output	Z co-ordinate, in meters.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.
(2) Use this function to obtain a contour of a given flux surface ($\psi = cst.$), e.g. in a toroidal cross section $\phi = cst.$, by letting θ run from 0 to 2π.

Name

b_field_cyl - Computes the components of the magnetic field in cylindrical co-ordinates $B = (B_R, B_\phi, B_Z)$ for a point given in cylindrical co-ordinates (R, ϕ, Z)

Synopsis

call b_field_cyl (R, Fi, Z, BR, Bfi, BZ)

Description

Variable	Type	input/output	Description
R	REAL	Input	R co-ordinate, in meters.
Fi	REAL	Input	ϕ co-ordinate, in radians.
Z	REAL	Input	Z co-ordinate, in meters.
BR	REAL	Output	R component of B, in Tesla.
Bfi	REAL	Output	ϕ component of B, in Tesla.
BZ	REAL	Output	Z component of B, in Tesla.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

b_field_car - Computes the components of the magnetic field in Cartesian co-ordinates $B = (B_X, B_Y, B_Z)$ for a point given in Cartesian co-ordinates (X, Y, Z)

Synopsis

call b_field_car (X, Y, Z, BX, BY, BZ)

Description

Variable	Type	input/output	Description
X	REAL	Input	X co-ordinate, in meters.
Y	REAL	Input	Y co-ordinate, in meters.
Z	REAL	Input	Z co-ordinate, in meters.
BX	REAL	Output	X component of B, in Tesla.
BY	REAL	Output	Y component of B, in Tesla.
BZ	REAL	Output	Z component of B, in Tesla.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

grad_b_cyl - Computes the derivatives of the magnetic field modulus in cylindrical co-ordinates $\partial|B|/\partial R$, $\partial|B|/\partial\phi$, $\partial|B|/\partial Z$ for a point given in cylindrical co-ordinates (R, ϕ, Z).

Synopsis

call grad_b_cyl (R, Fi, Z, dBdR, dBdfi, dBdZ)

Description

Variable	Type	input/output	Description
R	REAL	Input	R co-ordinate, in meters.
Fi	REAL	Input	ϕ co-ordinate, in radians.
Z	REAL	Input	Z co-ordinate, in meters.
dBdR	REAL	Output	$\partial\|B\|/\partial R$, in Tesla m^{-1}.
dBdfi	REAL	Output	$\partial\|B\|/\partial\phi$, in Tesla rad^{-1}.
dBdZ	REAL	Output	$\partial\|B\|/\partial Z$, in Tesla m^{-1}.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.
(2) Remember that the gradient in cylindrical co-ordinates can be easily computed from the above quantities as $\nabla|B| = (\partial|B|/\partial R, 1/R\partial|B|/\partial\phi, \partial|B|/\partial R)$. Please note the factor $1/R$ multiplying the derivative with respect to ϕ.

Name

grad_b_car - Computes the derivatives of the magnetic field modulus in Cartesian co-ordinates $\partial|B|/\partial X$, $\partial|B|/\partial Y$, $\partial|B|/\partial Z$ for a point given in Cartesian co-ordinates (X, Y, Z)

Synopsis

call grad_b_car (X, Y, Z, dBdX, dBdY, dBdZ)

Description

Variable	Type	input/output	Description
X	REAL	Input	X co-ordinate, in meters.
Y	REAL	Input	Y co-ordinate, in meters.
Z	REAL	Input	Z co-ordinate, in meters.
dBdX	REAL	Output	$\partial\|B\|/\partial X$, in Tesla m^{-1}.
dBdY	REAL	Output	$\partial\|B\|/\partial Y$, in Tesla m^{-1}.
dBdZ	REAL	Output	$\partial\|B\|/\partial Z$, in Tesla m^{-1}.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

find_axis_cyl - Calculates the position in cylindrical co-ordinates (R, ϕ, Z) of the axis at a given toroidal plane ϕ. Optionally, if the input variable iprint is equal to one, writes the evolution of the routine in finding the axis.

Synopsis

call find_axis_cyl (R, Fi, Z, flux, iprint)

Description

Variable	Type	input/output	Description
R	REAL	Output	R co-ordinate, in meters.
Fi	REAL	Input	ϕ co-ordinate, in radians.
Z	REAL	Output	Z co-ordinate, in meters.
Flux	REAL	Output	Normalized magnetic flux ψ at (R, ϕ, Z).
iprint	INTEGER	Output	If it is equal to zero not output is written. If equal to one writes the evolution of the finding process.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.
(2) The Output variable flux can be used as a measure of the error, since at the axis the magnetic flux should be zero.

Name

find_axis_car - Calculates the position in Cartesian co-ordinates (X, Y, Z) of the axis at a given toroidal plane ϕ. Optionally, if the input variable iprint is equal to one, writes the evolution of the routine in finding the axis.

Synopsis

call find_axis_car (Fi, X, Y, Z, flux, iprint)

Description

Variable	Type	input/output	Description
Fi	REAL	Input	ϕ co-ordinate, in radians.
X	REAL	Output	X co-ordinate, in meters.
Y	REAL	Output	Y co-ordinate, in meters.
Z	REAL	Output	Z co-ordinate, in meters.
Flux	REAL	Output	Normalized magnetic flux ψ at (X, Y, Z).
iprint	INTEGER	Output	If it is equal to zero not output is written. If equal to one writes the evolution of the finding process.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.
(2) The Output variable flux can be used as a measure of the error, since at the axis the magnetic flux should be zero.

Name

iota - Calculates the rotational transform at a given magnetic surface ψ.

Synopsis

call iota (Flux, aiota)

Description

Variable	Type	input/output	Description
Flux	REAL	Input	ψ.
aiota	REAL	Output	rotational transform ι.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

iota_cyl - Calculates the rotational transform at a given position in cylindircal co-ordinates (R, ϕ, Z) .

Synopsis

call iota_cyl (R, Fi, Z, aiota)

Description

Variable	Type	input/output	Description
R	REAL	Output	R co-ordinate, in meters.
Fi	REAL	Input	ϕ co-ordinate, in radians.
Z	REAL	Output	Z co-ordinate, in meters.
aiota	REAL	Output	rotational transform ι.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL meaus REAL*8.

Name

iota_car - Calculates the rotational transform at a given position in cartesian co-ordinates (X, Y, Z) .

Synopsis

call iota_car (X, Y, Z, aiota)

Description

Variable	Type	input/output	Description
X	REAL	Output	X co-ordinate, in meters.
Y	REAL	Input	Y co-ordinate, in meters.
Z	REAL	Output	Z co-ordinate, in meters.
aiota	REAL	Output	rotational transform ι.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

volume - Calculates the volume enclosed by a given magnetic surface ψ.

Synopsis

call volume (Flux, vol)

Description

Variable	Type	input/output	Description
Flux	REAL	Input	ψ.
vol	REAL	Output	volume enclosed by ψ in cubic meters.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

volume_cyl - Calculates the volume enclosed by a given position in cylindircal co-ordinates (R, ϕ, Z) .

Synopsis

call volume_cyl (R, Fi, Z, vol)

Description

Variable	Type	input/output	Description
R	REAL	Output	R co-ordinate, in meters.
Fi	REAL	Input	ϕ co-ordinate, in radians.
Z	REAL	Output	Z co-ordinate, in meters.
vol	REAL	Output	volume enclosed by (R, ϕ, Z) in cubic meters.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

volume_car - Calculates the volume enclosed by a given position in cartesian co-ordinates (X, Y, Z) .

Synopsis

call volume_car (X, Y, Z, vol)

Description

Variable	Type	input/output	Description
X	REAL	Output	X co-ordinate, in meters.
Y	REAL	Input	Y co-ordinate, in meters.
Z	REAL	Output	Z co-ordinate, in meters.
vol	REAL	Output	volume enclosed by (X, Y, Z) in cubic meters.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

surface - Calculates the surface enclosed by a given magnetic surface ψ.

Synopsis

call surface (Flux, surf)

Description

Variable	Type	input/output	Description
Flux	REAL	Input	ψ.
surf	REAL	Output	surface enclosed by ψ in squared meters.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

surface_cyl - Calculates the surface enclosed by a given position in cylindircal co-ordinates (R, ϕ, Z) .

Synopsis

call surface_cyl (R, Fi, Z, surf)

Description

Variable	Type	input/output	Description
R	REAL	Output	R co-ordinate, in meters.
Fi	REAL	Input	ϕ co-ordinate, in radians.
Z	REAL	Output	Z co-ordinate, in meters.
surf	REAL	Output	surface enclosed by (R, ϕ, Z) in squared meters.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

surface_car - Calculates the surface enclosed by a given position in cartesian co-ordinates (X, Y, Z) .

Synopsis

call surface_car (X, Y, Z, surf)

Description

Variable	Type	input/output	Description
X	REAL	Output	X co-ordinate, in meters.
Y	REAL	Input	Y co-ordinate, in meters.
Z	REAL	Output	Z co-ordinate, in meters.
surf	REAL	Output	surface enclosed by (X, Y, Z) in squared meters.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

Name

tj2 - Return a section of the TJ-II vacuum vessel at a given value of ϕ.

Synopsis

call tj2 (Fi, Npts, Rarr, Zarr)

Description

Variable	Type	input/output	Description
Fi	REAL	Input	ϕ co-ordinate, in radians.
Npts	REAL	Input/Output	On input: dimension of Rarr and Zarr arrays as declared in main program. On output, number of points returned in Rarr and Zarr arrays.
Rarr	REAL	Output	Array containing R co-ordinate of the vacuum vessel section, in meters.
Zarr	REAL	Output	Array containing Z co-ordinate of the vacuum vessel section, in meters.

Note
(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.
(2) On input, Npts should be about 400 or more.

Name

tj2_cut - Return a section of the TJ-II vacuum vessel with an arbitrary plane $A.X + B.Y + C.Z = D$.

Synopsis

call tj2_cut (A, B, C, D, Npts, Xcut, Ycut, Zcut, Xproject, Yproject)

Description

Variable	Type	input/output	Description
A	REAL	Input	A = X-component of normal vector to the plane, dimensionless.
B	REAL	Input	B = Y-component of normal vector to the plane, dimensionless.
C	REAL	Input	C = Z-component of normal vector to the plane, dimensionless.
D	REAL	Input	D = Offset of the plane from the origin, in meters.
Npts	REAL	Input/Output	On input: dimension of Xcut, Ycut, Zcut, Xproject, and Yproject arrays as declared in main program. On output, number of points returned in these arrays.

Continued on next page

Xarr	REAL	Output	Array containing X co-ordinate of the vacuum vessel section, in meters.
Yarr	REAL	Output	Array containing Y co-ordinate of the vacuum vessel section, in meters.
Zarr	REAL	Output	Array containing Z co-ordinate of the vacuum vessel section, in meters.
Xproject	REAL	Output	Array containing X co-ordinate of the projection of the vacuum vessel section on the given plane, in meters.
Yproject	REAL	Output	Array containing Y co-ordinate of the projection of the vacuum vessel section on the given plane, in meters.

Note

(1) Remember that in CRAY REAL means REAL*4 whereas in ALPHA REAL means REAL*8.

(2) On input, Npts should be about 6400 or more.

(3) The X- and Y-co-ordinates returned in Xproject and Yproject correspond to a "local" co-ordinate system on the plane. The relationship between this "local" co-ordinate system and the Cartesian co-ordinate system (X, Y, Z) will vary from case to case.

(4) The information returned in the arrays are pairs of points. A continuous line graphic representation of the vacuum vessel will only be obtained when connecting pairs of points (i.e. connect point 1 with point 2, point 3 with point 4, point 5 with point 6 etc., but DO NOT connect e.g. point 2 with point 3).

3 Examples

In this section, examples are provides for the practical use of the library. This example is available for both the CRAY and the ALPHA computers. The name of the test FORTRAN code is test.f and can be found in: /fusion/publica/tj2lib directory on the CRAY and in /usr/users2/neural/tj2lib directory on the ALPHA.

This program returns the magnetic axis position for the toroidal plane $\phi = 45°$, and writes a file, named fort.16, with the magnetic field modulus $|B|$, the density and the temperature profiles along the equatorial plane ($Z = 0$) for this toroidal section and the standard configuration of TJ-II Stellarator.

To run this code it is necessary to copy the files test.f and test to the user's directory and to execute the file test. The test file is a shell script that links the necessary input files (fit_TJII.std and field_TJII_std) containing the namelist with the currents in the coils and the fitted parameters describing the flux surfaces for the standard configuration of TJ-II, compiles and links the source code test.f with the library libtj2.a and runs the executable file test.i. Finally the files fort.44 and fort.45 are deleted. The variables paz and tj2 give the path to the desired files.

For the test case given here, the files fit_TJII_std and field_TJII_std correspond to the standard configuration of TJ-II Stellarator. In the CRAY system the file /fusion/publica/tj2lib/00readme contains information about the available configurations and the name of the files with the fit parameters and the coil currents. In the ALPHA system the equivalent file can be found in /usr/users2/neural/tj2lib/00readme.

The listing of the test script for ALPHA is written for the sh UNIX shell. Please note that we have used the compiler options -r8 -i8 to auto double the precision from REAL*4 to REAL*8 and from INTEGER*4 to INTEGER*8 because of the differences between both computers.

```
#       This script links the files fit_TJII.d and
#       field_TJII_std compile the test.f FORTRAN code and
#       links it with the library libtj2.a
#       Then it executes the resulting file test.i and deletes
#       the linked files fort.44 and fort.45

        echo "Running for TJ-II standard Configuration"
        echo "For other configurations see the file"
```

```
        echo "/usr/users2/neural/tj2lib/00readme"
        rm -f fort.44 fort.45
        paz="/usr/users2/neural/tj2lib"
        tj2=$paz"/libtj2.a"
        ln $paz/network/fit_TJII_std fort.44
        ln $paz/field/field_TJII_std    fort.45
        echo "Compiling and Linking test.f"
        f77 -o test.i -i8 -r8 test.f $tj2
        echo "Executing test.i"
        time test.i
        rm -f fort.44 fort.45
exit
```

The test script for CRAY system is the following, and is also written for the sh shell

```
#       This script links the files fit_TJII.d and
#       field_TJII_std compile the test.f FORTRAN code and
#       links it with the library libtj2.a
#       Then it executes the resulting file test.i and deletes
#       the linked files fort.44 and fort.45

        echo "Running for TJ-II standard Configuration"
        echo "For other configurations see the file"
        echo "/fusion/publica/tj2lib/00readme"
        rm -f fort.44 fort.45
        paz="/fusion/publica/tj2lib"
        tj2=$paz"/libtj2.a"
        ln $paz/network/fit_TJII_std fort.44
        ln $paz/field/field_TJII_std   fort.45
        echo "Compiling and Linking test.f"
        cf77 -Wl"-o test.i $tj2" -- test.f
        echo "Executing test.i"
        time test.i
        rm -f fort.44 fort.45
exit
```

To execute the test script the user must enter sh test.

The listing of the FORTRAN source code test.f is the same for both computers:

```
      program test
      implicit none
      real    Te0,ne0,Rmin,Rmax,fi,raxis,zaxis,flux,
     +        R,BR,Bfi,BZ,Bmod,Te, ne, Pi
      integer imax,i

      call init_tj2_lib('fort.44','fort.45')
      Pi = acos(-1.)
      Te0 = 1.
      ne0 = 0.8
      Rmin = 1.1
      Rmax = 1.4
      imax = 128
      fi   = Pi/4.
      call find_axis_cyl(raxis,fi,zaxis,flux)
      write(6,*) 'The plasma axis for fi = ',fi
      write(6,*) 'is at (R, Z) = (',raxis,',',zaxis,')'
      do i=1,imax
         R = Rmin+(Rmax-Rmin)*(i-1)/(imax-1.)
         call B_Field_Cyl(R,0.,0.,BR,Bfi,BZ)
         call Flux_Cyl   (R,0.,0.,flux)
         bmod = sqrt(BR*BR+BFi*BFi+BZ*BZ)
         flux = max(0.,min(1.,flux))
         Te = Te0*(1.-flux**2. )**2.
         ne = ne0*(1.-flux**1.5)**2.
         write(16,'(4(f12.8))') R,Bmod,Te,ne
      end do
      end
```

We have put flux = max(0.,min(1.,flux)) due to the fact that the fitting of the flux surfaces is not defined outside the plasma, and is not exactly equal zero near the axis, because it is an approximation to the real flux. In this way the direct output of the routine can be used to check whether a point is inside or outside the plasma.

Once the code has been run it is possible to plot the results with a shareware graphical application called GNUPLOT. Although this application is

shareware it is very powerful and allows any user in a UNIX system to easily plot the data contained in ASCII files. To plot the results from the test code enter gnuplot in the command line (in the following example the bold is used for the commands that the user must type):

```
> gnuplot

        G N U P L O T
        unix version 3.5
        patchlevel 3.50.1.17, 27 Aug 93
        last modified Fri Aug 27 05:21:33 GMT 1993

        Copyright(C) 1986 - 1993  Thomas Williams, Colin Kelley

        Send comments and requests for help to info-gnuplot@dartmouth.edu
        Send bugs, suggestions and mods to bug-gnuplot@dartmouth.edu

gnuplot>  set terminal tek410x
Terminal type set to 'tek410x'
gnuplot>  plot 'fort.16' using 1:2 with lines
```

After this you will get the profile of the magnetic field modulus

Since the density and temperature profiles are stored as the third and fourth columns in the file fort.16 if you would like to plot them all you have to do is to replace the command line from

```
plot 'fort.16' using 1:2 with lines
```

to

```
plot 'fort.16' using 1:3 with lines
```

For more information on the GNUPLOT program type man gnuplot.

Acknowledgments

The Authors are indebted to J. Guasp for his unrewarded work in the Biot-Savart routine and to J. A. Jimenez for his equilibrium calculations.

References

[1] J. Guasp and A. Lopez-Fraguas, Personal communication.

[2] V. Tribaldos and B. Ph. van Milligen, *Rev. Sci. Inst.* **68** 935 (1997).

[3] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes*, Cambridge University Press (1992).