Ciemat
Centro de
Investigaciones Energéticas,
Medioambientales
y Tecnológicas

Miner

TJ-II Library Manual

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

Informes Técnicos Ciemat

806
Agosto 1996

Informes Técnicos Ciemat

806
Agosto 1996

TJ-II Library Manual

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

Asociación EURATOM / CIEMAT para Fusión - 34

Instituto de Investigación Básica

Toda correspondencia 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 Estados Unidos.

Depósito Legal: M-14226-1995
NIPO: 238-96-001-0
ISSN: 1135-9420

Editorial CIEMAT

CLASIFICACIÓN DOE Y DESCRIPTORES

700320, 990200
TOKAMAK DEVICES, HELICAL CONFIGURATION, MAGNETIC FIELDS, COMPUTER CODES, STELLARATORS, HELICAL STELLARATORS, FORTRAN, PROGRAMMING, COMPUTER PROGRAM DOCUMENTATION

"TJ-II Library Manual"

Tribaldos, V.; van Milligen, B. Ph.; López-Fraguas, A.
27 pp. 0 figs. 2 refs.

Abstract

This report contains a detailed description of the TJ-II library and its routines. The library is written in FORTRAN 77 language and is available in the CRAY J916 and DEC Alpha 8400 computers at CIEMAT. This document also contains some examples of its use.

"TJ-II Library Manual"

Tribaldos, V.; van Milligen, B. Ph.; López-Fraguas, A.
27 pp. 0 figs. 2 refs.

Resumen

Este informe contiene una descripción detallada de la librería escrita para el Stellarator TJ-II, así como de las rutinas que incluye. La librería está escrita en el lenguaje FORTRAN 77 y se haya disponible en los ordenadores CRAY J916 y DEC Alpha 8400 del CIEMAT. Además en él se presentan una serie de ejemplos para facilitar su uso.

Table of Contents

Introduction

This Manual describes a numerical library called TJ-II. This library is intended to give computational tools for calculating several quantities for TJ-II Stellarator configurations. Those quantities include the magnetic field components and the normalized magnetic flux at every point. The routines we are going to present are programmed using the FORTRAN 77 language. The TJ-II library is a *shared* , usually called *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.

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*.

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 supposed to be constant on the magnetic surfaces they are functions of the normalized magnetic flux ψ, defined to be equal to zero on the axis and one at the edge. This is the reason why throughout this library there will be no mention to the density and temperature. If one wants to compute those quantities the easiest way is to choose a dependence of the type: $n(\mathbf{r}) = n(\mathbf{r}_{axis})\left(1-\psi(\mathbf{r})^{\alpha}\right)^{\beta}$ and $T(\mathbf{r}) = T(\mathbf{r}_{axis})\left(1-\psi(\mathbf{r})^{\alpha}\right)^{\beta}$. 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, $0 \le r_{eff} \le 1$, is defined by $r_{eff} = \sqrt{\psi}$. The use of the name radius is common but can be misleading.

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 with a neural network fit of the magnetic flux surfaces [2]. This method has shown to be accurate, giving RMS. errors less than 0.5%, and very quick for evaluation. Moreover the number of coefficients needed to describe the full three dimensional structure of TJ-II plasma is very low ($\approx$1000), and it is also possible to analytically obtain the spatial derivatives.

The name of the routines try to show the quantities they compute and the co-ordinate system used, Cartesian of cylindrical.

Structure of this Document

This manual document consist of the following parts:

- **Routines :** Summarizes the routines and gives a detailed explanation of their arguments and use.

- **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.

Conventions Used

This manual uses the conventions listed in Table 1.

Table 1 Conventions

Convention	Meaning
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.

Routines

The structure of this library is a source FORTRAN 77 code of approximately 3000 lines consisting in 50 internal routines (of which approximately 10 are accessible to the user). Throughout this library MKS units are used, i.e. the co-ordinates are given in meters and radians and the magnetic field is given in Tesla. 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 neural network coefficients of 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.

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

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

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.

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

Description: Short description of what the routine computes.

Table 2 : List of routines

Routine name	Arguments	Description
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 the normalized magnetic flux ψ at $\mathbf{r} = (R, \varphi, Z)$.
flux_car	4, R	Returns the normalized magnetic flux ψ at $\mathbf{r} = (X, Y, Z)$.
grad_flux_cyl	6, R	Returns $\partial\psi/\partial R$, $\partial\psi/\partial\varphi$ and $\partial\psi/\partial Z$ at $\mathbf{r} = (R, \varphi, Z)$.
grad_flux_car	6, R	Returns $\partial\psi/\partial X$, $\partial\psi/\partial Y$ and $\partial\psi/\partial Z$ at $\mathbf{r} = (X, Y, Z)$.
b_field_cyl	6, R	Returns the components of the magnetic field $\mathbf{B} = (B_R, B_\varphi, B_Z)$ at $\mathbf{r} = (R, \varphi, Z)$.
b_field_car	6, R	Returns the components of the magnetic field $\mathbf{B} = (B_X, B_Y, B_Z)$ at $\mathbf{r} = (X, Y, Z)$.
grad_b_cyl	6, R	Returns $\partial\lvert\mathbf{B}\rvert/\partial R$, $\partial\lvert\mathbf{B}\rvert/\partial\varphi$ and $\partial\lvert\mathbf{B}\rvert/\partial Z$ at $\mathbf{r} = (R, \varphi, Z)$.
grad_b_car	6, R	Returns $\partial\lvert\mathbf{B}\rvert/\partial X$, $\partial\lvert\mathbf{B}\rvert/\partial Y$ and $\partial\lvert\mathbf{B}\rvert/\partial Z$ at $\mathbf{r} = (X, Y, Z)$.
find_axis_cyl	4, R; 1, I	For a given toroidal plane φ returns the position (R, φ, Z) of the axis.
find_axis_car	5, R; 1, I	For a given toroidal plane φ returns the position (X, Y, Z) of the axis.

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 the neural network for 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 weights of the neural network and the coil currents. In the ALPHA system the equivalent file can be found in `/usr/users2/tribal/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, φ, Z, *flux*)

Description

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

Note

(1) 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, of the point.
Y	REAL	Input	Y co-ordinate, in meters, of the point.
Z	REAL	Input	Z co-ordinate, in meters, of the point.
flux	REAL	Output	Normalized magnetic flux ψ.

Note

(1) 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 $\partial\psi/\partial R$, $\partial\psi/\partial\varphi$, $\partial\psi/\partial Z$ for a point given in cylindrical co-ordinates (R, φ, Z)

Synopsis

call grad_flux_cyl (*R*, *φ*, *Z*, *dfluxdR*, *dfluxdfi*, *dfluxdZ*)

Description

Variable	Type	input/output	Description
R	REAL	Input	R co-ordinate, in meters, of the point.
φ	REAL	Input	φ co-ordinate, in radians, of the point.
Z	REAL	Input	Z co-ordinate, in meters, of the point.
dfluxdR	REAL	Output	R component of the normalized magnetic flux gradient $\partial\psi/\partial R$., in m^{-1}.
dfluxdfi	REAL	Output	φ component of the normalized magnetic flux gradient $\partial\psi/\partial\varphi$., in rad^{-1}.
dfluxdZ	REAL	Output	Z component of the normalized magnetic flux gradient $\partial\psi/\partial Z$., 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 $\nabla\psi = (\partial\psi/\partial R,\ 1/R\ \partial\psi/\partial\varphi,\ \partial\psi/\partial Z)$. 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, of the point.
Y	REAL	Input	*Y* co-ordinate, in meters, of the point.
Z	REAL	Input	*Z* co-ordinate, in meters, of the point.
dfluxdX	REAL	Output	*X* component of the normalized magnetic flux gradient $\partial\psi/\partial X$, in m^{-1}.
dfluxdY	REAL	Output	*Y* component of the normalized magnetic flux gradient $\partial\psi/\partial Y$, in m^{-1}.
dfluxdZ	REAL	Output	*Z* component of the normalized magnetic flux gradient $\partial\psi/\partial Z$., in m^{-1}.

Note

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

Name

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

Synopsis

call b_field_cyl (*R*, φ, *Z*, *BR*, *Bfi*, *BZ*)

Description

Variable	Type	input/output	Description
R	REAL	Input	*R* co-ordinate, in meters, of the point.
φ	REAL	Input	φ co-ordinate, in radians, of the point.
Z	REAL	Input	*Z* co-ordinate, in meters, of the point.
BR	REAL	Output	*R* component of the magnetic field B_R, in Tesla.
Bfi	REAL	Output	φ component of the magnetic field B_φ, in Tesla.
BZ	REAL	Output	*Z* component of the magnetic field B_Z, 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, of the point.
Y	REAL	Input	*Y* co-ordinate, in meters, of the point.
Z	REAL	Input	*Z* co-ordinate, in meters, of the point.
BX	REAL	Output	*X* component of the magnetic field B_X, in Tesla.
BY	REAL	Output	*Y* component of the magnetic field B_Y, in Tesla.
BZ	REAL	Output	*Z* component of the magnetic field B_Z 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 $\partial|\mathbf{B}|/\partial R$, $\partial|\mathbf{B}|/\partial\varphi$, $\partial|\mathbf{B}|/\partial Z$ for a point given in cylindrical co-ordinates (R, φ, Z).

Synopsis

call grad_b_cyl (*R*, *φ*, *Z*, *BR*, *Bfi*, *BZ*)

Description

Variable	Type	input/output	Description
R	REAL	Input	Radial co-ordinate, in meters, of the point.
φ	REAL	Input	Toroidal co-ordinate, in radians, of the point.
Z	REAL	Input	Z co-ordinate, in meters, of the point.
dBdR	REAL	Output	Radial component of the gradient of the magnetic field modulus $\partial|\mathbf{B}|/\partial R$, in Tesla m^{-1}.
dBdfi	REAL	Output	Toroidal component of the gradient of the magnetic field modulus $\partial|\mathbf{B}|/\partial\varphi$, in Tesla rad^{-1}.
dBdZ	REAL	Output	Z component of the gradient of the magnetic field modulus $\partial|\mathbf{B}|/\partial Z$, 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 $\nabla |\mathbf{B}| = (\partial|\mathbf{B}|/\partial R,\ 1/R\ \ \partial|\mathbf{B}|/\partial\varphi,\ \partial|\mathbf{B}|/\partial Z)$. 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 $\partial|\mathbf{B}|/\partial X$, $\partial|\mathbf{B}|/\partial Y$, $\partial|\mathbf{B}|/\partial Z$ 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, of the point.
Y	REAL	Input	Y co-ordinate, in meters, of the point.
Z	REAL	Input	Z co-ordinate, in meters, of the point.
dBdX	REAL	Output	X component of the gradient of the magnetic field modulus $\partial|\mathbf{B}|/\partial X$, in Tesla m^{-1}.
dBdY	REAL	Output	Y component of the gradient of the magnetic field modulus $\partial|\mathbf{B}|/\partial Y$, in Tesla m^{-1}.
dBdZ	REAL	Output	Z component of the gradient of the magnetic field modulus $\partial|\mathbf{B}|/\partial Z$, 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, φ, Z, *flux*, *iprint*)

Description

Variable	Type	input/output	Description
R	REAL	Output	R co-ordinate, in meters, of the point.
φ	REAL	Input	φ co-ordinate, in radians, of the point.
Z	REAL	Output	Z co-ordinate, in meters, of the point.
flux	REAL	Output	Normalized magnetic flux at the point (R, φ, Z).
iprint	INTEGER	Input	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 (φ, X, Y, Z, *flux*, *iprint*)

Description

Variable	Type	input/output	Description
φ	REAL	Input	φ co-ordinate, in radians, of the toroidal plane.
X	REAL	Output	X co-ordinate, in meters, of the axis.
Y	REAL	Output	Y co-ordinate, in meters, of the axis.
Z	REAL	Output	Z co-ordinate, in meters, of the axis.
flux	REAL	Output	Normalized magnetic flux at the point (R, φ, Z).
iprint	INTEGER	Input	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

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/tribal/tj2lib` directory on the ALPHA.

This program returns the magnetic axis position for the toroidal plane $\varphi = 0°$, 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 (`weight_TJII.std` and `field_TJII_std`) containing the namelist with the currents in the coils and the coefficients of the neural network 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 `weight_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 weights of the neural network and the coil currents. In the ALPHA system the equivalent file can be found in `/usr/users2/tribal/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 weight_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/tribal/tj2lib/00readme"
        rm -f fort.44 fort.45
        paz="/usr/users2/tribal/tj2lib"
        tj2=$paz"/libtj2.a"
        ln $paz/network/weight_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 weight_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/weight_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:

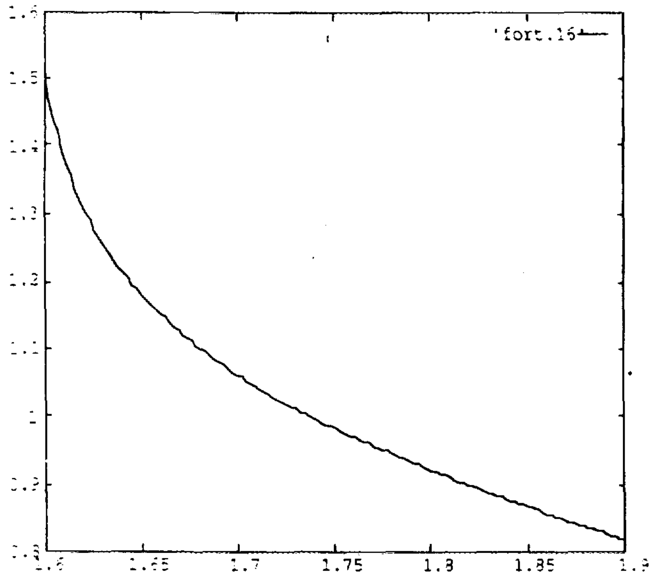
```
       program test
       implicit none
       real     Te0,ne0,Rmin,Rmax,fi,raxis,zaxis,flux,
     +          R,BR,Bfi,BZ,Bmod,Te, ne
       integer imax,i

       call init_tj2_lib('fort.44','fort.45')
       Te0 = 1.
       ne0 = 0.8
       Rmin = 1.6
       Rmax = 1.9
       imax = 128
       fi   = 0.
       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 network output is larger than one outside the plasma, due to extrapolation, since it is not defined outside the plasma, and 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`.

Acknowledgements

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, Nuclear Fusion **36**, 283, 1996.
V. Tribaldos and B. Ph. van Milligen, Rev. Sci. Inst. (1996) (in press).

Index