

January 1996



**FOM**-Instituut voor Plasmafysica Rijnhuizen    **Associatie Euratom-FOM**

Postbus 1207  
3430 BE Nieuwegein  
Nederland  
Edisonbaan 14  
3439 MN Nieuwegein  
Tel. + 31(0)30-6096999  
Fax. + 31(0)30-6031204

# **HELENA**

## **Installation and User Guide**

S. POEDTS, H.A. HOLTIES, J.P. GOEDBLOED,  
G.T.A. HUYSMANS\*, and W. KERNER\*

\* JET Joint Undertaking, Culham, United Kingdom.

**RIJNHUIZEN  
REPORT  
96-228**



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation guide</b>	<b>5</b>
2.1	HELENA and satellite programs . . . . .	5
2.2	Installation of REVISE . . . . .	5
2.3	Installation of PPPLIB and HGOLIB . . . . .	5
2.4	Installation of HELENA . . . . .	5
<b>3</b>	<b>User guide</b>	<b>7</b>
3.1	Setting the dimensions . . . . .	7
3.2	Pre-compiling, compiling, and linking . . . . .	7
3.3	Input file . . . . .	7
3.3.1	Namelist SHAPE . . . . .	7
3.3.2	Namelist PROFILE . . . . .	9
3.3.3	Namelist PHYS . . . . .	10
3.3.4	Namelist NUM . . . . .	10
3.4	Submitting a job . . . . .	11
3.5	Output files . . . . .	12
3.5.1	Listing . . . . .	12
3.5.2	Mapping file . . . . .	12
3.5.3	Vector plot file . . . . .	13
3.5.4	Plot file . . . . .	13
3.6	Connection with HBT . . . . .	13
3.6.1	HBT $\Rightarrow$ HELENA . . . . .	14
3.6.2	HELENA $\Rightarrow$ HBT . . . . .	15
3.7	Handy hints . . . . .	15
<b>A</b>	<b>REVISE</b>	<b>17</b>
A.1	Introduction . . . . .	17
A.2	Contents and use . . . . .	17
A.3	Installation of REVISE . . . . .	18
A.4	IBM-VM version of REVISE . . . . .	18
A.5	List of files . . . . .	19
<b>B</b>	<b>Installation script for PPPLIB</b>	<b>20</b>
<b>C</b>	<b>Installation script for HGOLIB</b>	<b>21</b>
	<b>References</b>	<b>22</b>

# 1 Introduction

The aim of this 'Installation and user guide' for the program HELENA is to provide *practical information* for the installation of the program and for using it to calculate toroidal plasma equilibria and corresponding mapping files which are required by CASTOR to analyze the MHD stability of these equilibria. For a more technical and detailed description of the applied numerical methods the reader is referred to the literature cited below.

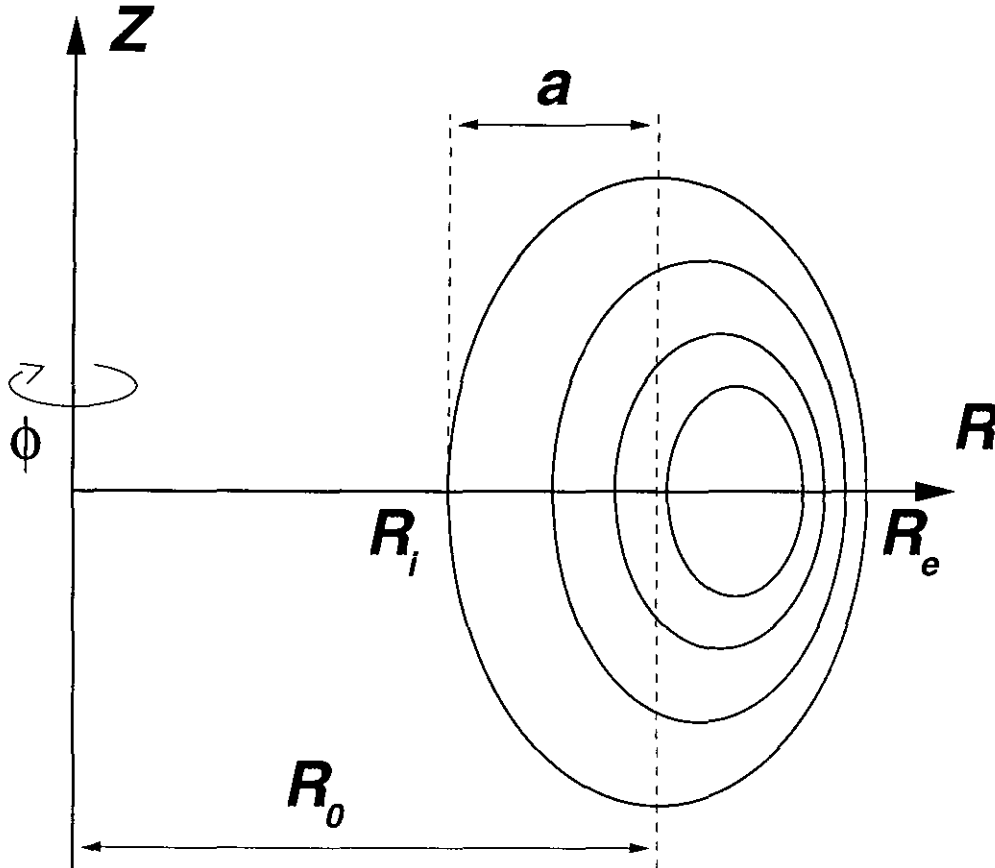


Figure 1: Schematic representation of the tokamak geometry with cylindrical coordinates  $(R, \phi, Z)$ .  $R_e$  and  $R_i$  are the two values of the major radius where the last closed magnetic surface crosses the equatorial plane, where  $R_i < R_e$ ;  $a = (R_e - R_i)/2$  and  $R_0 = (R_e + R_i)/2$ .

HELENA is a Fortran 77 program that solves the MHD equilibrium equation of a toroidal axisymmetric plasma with the use of isoparametric bicubic Hermite finite elements. This numerical technique yields very accurate solutions and has good convergence properties.

In toroidal geometry, the MHD equilibrium force balance  $\nabla P = \vec{J} \times \vec{B}$  is described by the Grad-Shafranov equation for the poloidal flux  $\psi(R, Z)$

$$\Delta^* \psi \equiv R \frac{\partial}{\partial R} \left( \frac{1}{R} \frac{\partial \psi}{\partial R} \right) + \frac{\partial^2 \psi}{\partial Z^2} = \mu_0 R J_\phi, \quad (1)$$

where  $J_\phi$  is the toroidal current density,  $R$  is the distance from the axis of symmetry, and  $Z$  is the vertical coordinate, as indicated in Fig. 1. Due to the toroidal symmetry, the right-hand side of Eq. (1) can be written in terms of two flux functions, viz. the pressure  $P(\psi)$  and the diamagnetic profile  $F(\psi) = RB_\phi$ :

$$\mu_0 R J_\phi = -\mu_0 R^2 \frac{dP(\psi)}{d\psi} - \frac{1}{2} \frac{dF^2(\psi)}{d\psi}. \quad (2)$$

In terms of  $\psi(R, Z)$ , the magnetic field is represented by

$$\vec{B} = \nabla\psi \times \nabla\phi + F\nabla\phi, \quad (3)$$

so that the poloidal magnetic field is obtained from  $B_p = \nabla\psi/R$ .

The formulation of the equilibrium problem of finding  $\psi(R, Z)$  in terms of physical quantities contains two trivial scaling parameters, viz. the vacuum magnetic field  $B_0$  at the geometric center of the plasma and the scalelength of the problem:  $a$ , the minor radius of the plasma boundary. These two parameters only provide the dimensions of the physical quantities and do not enter the problem otherwise. Similarly, the total poloidal flux,  $2\pi\psi_1$ , only enters as a scaling parameter normalizing the  $J_\phi$  and  $B_\phi$  profiles. This scaling is accounted for by the parameter  $\alpha = a^2 B_0 / \psi_1$ . The two profiles  $P$  and  $F$  are then normalized as follows:

$$\bar{P}(\bar{\psi}) = \left( \frac{\mu_0 \alpha^2}{\epsilon B_0^2} \right) P(\psi), \quad \bar{F}^2(\bar{\psi}) = \left( \frac{\epsilon \alpha^2}{a^2 B_0^2} \right) (F^2(\psi) - R_0^2 B_0^2), \quad (4)$$

where  $\bar{\psi} \equiv \psi/\psi_1$  is the normalized flux. The flux  $\bar{\psi}$ , will play the role of a radial coordinate. The physical parameters can then be scaled as:

$$\begin{aligned} B_\phi &= B_0 \cdot \frac{1}{1 + \epsilon x} \sqrt{1 + \frac{\epsilon}{\alpha^2} \bar{F}^2(\bar{\psi})}, \\ \vec{B}_p &= \frac{\epsilon B_0}{\alpha} \cdot \frac{-1}{1 + \epsilon x} \nabla\bar{\psi} \times \vec{e}_\phi, \\ J_\phi &= \frac{\epsilon B_0}{\mu_0 a \alpha} \cdot \frac{-1}{\epsilon} \left[ (1 + \epsilon x) \bar{P}'(\bar{\psi}) + \frac{\bar{F} \bar{F}'(\bar{\psi})}{1 + \epsilon x} \right], \\ \vec{J}_p &= \frac{\epsilon B_0}{\mu_0 a \alpha^2} \cdot \frac{\bar{F} \bar{F}'(\bar{\psi})}{1 + \epsilon x} \left[ 1 + \frac{\epsilon}{\alpha^2} \bar{F}^2(\bar{\psi}) \right]^{-1/2} \nabla\bar{\psi} \times \vec{e}_\phi, \end{aligned} \quad (5)$$

where  $\epsilon = a/R_0$  is the inverse aspect ratio and the normalized coordinates  $x = (R - R_0)/a$  and  $y = Z/a$  are used in the poloidal plane. In terms of these dimensionless quantities, the Grad-Shafranov equation for  $\bar{\psi}(x, y)$  becomes:

$$\Delta^* \bar{\psi} = \frac{\partial^2 \bar{\psi}}{\partial x^2} + \frac{\partial^2 \bar{\psi}}{\partial y^2} - \frac{\epsilon}{1 + \epsilon x} \frac{\partial \bar{\psi}}{\partial x} = -\frac{(1 + \epsilon x)^2}{\epsilon} \bar{P}'(\bar{\psi}) - \frac{1}{\epsilon} \bar{F}(\bar{\psi}) \bar{F}'(\bar{\psi}), \quad (6)$$

The solution  $\bar{\psi}(x, y)$  of the above equation is a smooth continuous function of  $x$  and  $y$  so that the solution consists of a set of nested closed flux surfaces around the magnetic axis where  $B_p = 0$ . Note also that in terms of these dimensionless quantities, the plasma beta and the poloidal plasma beta are defined as

$$\beta \equiv \frac{2\epsilon}{\alpha^2} \langle \bar{P} \rangle, \quad \text{and} \quad \beta_p \equiv \frac{8\pi \epsilon \bar{S} \langle \bar{P} \rangle}{\alpha^2 \bar{I}^2}, \quad (7)$$

respectively, where  $\bar{S}$  is the dimensionless area of the poloidal cross section of the plasma ( $S/a^2$ ).

In order to separate the *shape* of the two input profiles  $\bar{P}(\bar{\psi})$  and  $\bar{F}\bar{F}'(\bar{\psi})$  from the *amplitude*, two unit profiles are defined which are normalized to one on the magnetic axis ( $\bar{\psi} = 0$ ):

$$\begin{aligned}\frac{1}{2}AB\Pi(\bar{\psi}) &= -\bar{P}'(\bar{\psi}), \\ A\Gamma(\bar{\psi}) &= -\frac{1}{\epsilon}(\bar{P}'(\bar{\psi}) + \bar{F}\bar{F}'(\bar{\psi})),\end{aligned}\quad (8)$$

with  $\Pi(0) = \Gamma(0) = 1$ . The quantities  $A$  and  $B$  determine the amplitudes of the two input profiles. However, only  $B$  is an input quantity for HELENA. The value of  $A$  is determined by the condition that  $\bar{\psi} = 1$  on the plasma boundary. The definition of the  $\Pi(\bar{\psi})$  and  $\Gamma(\bar{\psi})$  profiles is the same as used in the HBT equilibrium and stability code [3, 2]. In terms of  $\Pi(\bar{\psi})$  and  $\Gamma(\bar{\psi})$  the Grad-Shafranov equation is written as follows:

$$\Delta^*\bar{\psi} = \frac{\partial^2\bar{\psi}}{\partial x^2} + \frac{\partial^2\bar{\psi}}{\partial y^2} - \frac{\epsilon}{1+\epsilon x}\frac{\partial\bar{\psi}}{\partial x} = A[\Gamma(\bar{\psi}) + Bx(1 + \frac{\epsilon x}{2})\Pi(\bar{\psi})].\quad (9)$$

As an analytic test case, the so-called Soloviev equilibrium [6] has been programmed in HELENA. This equilibrium is an analytic solution of the Grad-Shafranov equation where the two equilibrium profiles  $\bar{P}(\bar{\psi})$  and  $\bar{F}\bar{F}'(\bar{\psi})$  are independent of  $\bar{\psi}$ . In terms of the ellipticity  $E = b/a$  and the triangularity  $\tau$  of the plasma boundary, the general up-down symmetric solution is given by:

$$\bar{\psi} = (x - \frac{\epsilon}{2}(1 - x^2))^2 + ((1 - \frac{\epsilon^2}{4})(1 + \epsilon x)^2 + \tau x(1 + \frac{\epsilon x}{2}))\frac{y^2}{E^2},\quad (10)$$

where  $x$  and  $y$  are the normalized coordinates.

The Grad-Shafranov equation is solved inside a region bounded by the arbitrary but given shape of the outermost flux surface  $\bar{\psi} = 1$ , i.e. the plasma boundary. In other words, the boundary condition of Eq. (6) is  $\bar{\psi} = 1$ . Furthermore, HELENA assumes the plasma to be up-down symmetric <sup>1</sup> and solves Eq. (6) in the upper half of the poloidal plane only with the additional boundary condition  $\partial\bar{\psi}/\partial x(y = 0) = 0$ . See [4] and [1] for more information on the isoparametric bicubic Hermite finite elements and the way they are combined with a Galerkin method to solve the nonlinear Grad-Shafranov equation by an iteration procedure.

HELENA is used by many people and different versions of the program are available. The REVISE package (see Appendix A) is used too keep track of the different versions of the program. This installation and user guide applies to version 8.

---

<sup>1</sup>There also exists an extension of HELENA to general non up-down symmetric plasma shapes [5].

## 2 Installation guide

This installation guide contains information to install HELENA from scratch, i.e. assuming none of the help programs (see below) has been installed yet. First, it is explained what subroutine libraries are used by HELENA. Next, it is explained how to install these subroutine libraries and the program HELENA.

### 2.1 HELENA and satellite programs

HELENA is not a stand-alone program. It makes use of PPPLIB (Plasma Physics Plot LIBrary) subroutines for generating graphics and of HGOLIB (Hans GOedbloed LIBrary) subroutines for the numerics. However, both PPPLIB and HGOLIB are written in standard Fortran 77 so these subroutine libraries are portable. They need to be installed first, before running HELENA. In addition, some NAG routines are used. These have to be replaced by other similar routines if NAG is not available.

Moreover, HELENA and PPPLIB are provided in REVISE format. This means REVISE needs to be installed first. More information on the use of REVISE and on how to install this package is given in Appendix A.

Hence, the following files are required for a successful installation of HELENA version 8, in addition to REVISE and NAG: hel8.source, ppp15e.source, hgolib.f, ppplib.mk, hgolib.mk. The latter two files are Unix scripts which create the libraries PPPLIB and HGOLIB as discussed below.

### 2.2 Installation of REVISE

First of all REVISE needs to be installed. The way this is done is described in Appendix A.

### 2.3 Installation of PPPLIB and HGOLIB

HELENA 8 works with version 15e of PPPLIB. To install this version of the Plasma Physics Plot library the file ppp15e.source first needs to be pre-compiled. The resulting fortran file has to be split in different files, one for each subroutine. This can be done with the 'fsplit' command. Next, each of these fortran files needs to be compiled and the compiled files have to be put in a library (with the 'ar' command). For Unix machines, a script is available to do all this with just one command. The script is called ppplib.mk. Its contents is printed out in Appendix B.

The installation of HGOLIB is very similar to the installation of PPPLIB except for the fact that HGOLIB is not provided in REVISE format so that the pre-compilation step is not required here. Again, a script is available to install HGOLIB on Unix machines. This script is called hgolib.mk. Its contents is printed out in Appendix C.

### 2.4 Installation of HELENA

The installation of HELENA is very simple. It suffices to make a directory, for instance \$HOME/helena, and to copy the file hel8.source in it.

## 3 User guide

This user guide describes the different steps required to run HELENA successfully. All input parameters are described as well as all the output produced by a successful run of the code.

### 3.1 Setting the dimensions

Normally, this step can be skipped. However, maximal dimensions are specified in COMDECK PARAM and whenever a larger resolution is required, e.g. when reconstructing equilibria close to an X-point, these parameters need to be adjusted.

### 3.2 Pre-compiling, compiling, and linking

First, hel8.source needs to be pre-compiled. The resulting fortran file hel8.f can then be compiled and linked to NAG, PPPLIB, and HGOLIB. On a Cray computer the following two commands do the job:

1) pre hel8

2) cf77 -o hel8 -lnag -Wf'-em -l hel8.lst" hel8.f \$HOME/libs/ppplib.a \$HOME/libs/hgolib.a

and putting them in one script makes it even easier.

### 3.3 Input file

HELENA uses NAMELIST formatting to read in data from input unit 10. Four namelists are defined, viz. SHAPE, PROFILE, PHYS, and NUM. The parameters in namelist SHAPE determine the shape of a poloidal cross-section of the outer flux surface. The parameters in namelist PROFILE determine the two required input profiles. Namelist PHYS determines some physical parameters. Namelist NUM determines some numerical parameters. The meaning of the constants in each namelist is described below.

#### 3.3.1 Namelist SHAPE

HELENA uses cartesian coordinates  $(x, y)$  and polar coordinates  $(r, \theta)$  with the center in the geometric center of the poloidal cross-section, i.e.  $(R_e + R_i)/2$  (see Fig. 1). The shape of a poloidal cross-section of the outer flux surface, i.e.  $r = \sqrt{x^2 + y^2} = f(\theta)$ , can be determined in three different ways.

The first possibility is to specify it through the coefficients of the parametric form:

$$\begin{aligned} x &= a \cos(\gamma + \tau \sin(\gamma) + \kappa \sin(2\gamma)), \\ y &= b \sin(\gamma), \end{aligned} \tag{11}$$

where  $\theta$  is found from  $\gamma$  by inversion, i.e. the function

$$\theta(\gamma) \equiv \arctan\left(\frac{y}{x}\right) = \arctan\left(\frac{b/a \sin(\gamma)}{\cos(\gamma + \tau \sin(\gamma) + \kappa \sin(2\gamma))}\right), \tag{12}$$

is inverted to  $\gamma(\theta)$  so that

$$r(\theta) = r(\gamma(\theta)) \equiv \sqrt{x^2(\gamma(\theta)) + y^2(\gamma(\theta))}. \quad (13)$$

The Fourier coefficients  $f_m$  of

$$r = f(\theta) = \sum_{m=0}^{M_1} f_m \cos(m\theta), \quad (14)$$

are also determined in this case.

The second possibility is to give the shape as a truncated Fourier series, i.e.

$$r = f(\theta) = \sum_{m=0}^{M_2} f_m \cos(m\theta), \quad (15)$$

where both the number of Fourier terms,  $M_2$ , and the coefficients,  $f_m$ , have to be specified.

The third possibility defines  $r = f(\theta)$  as the perturbation of a circle:

$$r = f(\theta) = 1 + P_1 \cos\left(\frac{\theta - P_2}{P_3}\right)^{P_4}, \quad (16)$$

where  $P_1, \dots, P_4$  are input parameters. As for possibility 1, the Fourier coefficients  $f_m$  (see Eq. (14)) are also determined.

The namelist SHAPE then contains the following parameters:

ISHAPE = switch parameter, determines the way in which the shape of the plasma boundary is given:

- 1: by the formula (11);
- 2: by the Fourier coefficients of  $r(\theta)$ , i.e. formula (15);
- 3: by a perturbation of a circle determined by  $P_1, \dots, P_4$  in formula (16);

ELLIP = ellipticity of the plasma boundary, i.e.  $E = b/a$ ; (for ISHAPE = 1)

TRIA = triangularity  $\tau$  of the plasma boundary; (for ISHAPE = 1)

QUAD = quadraticity  $\kappa$  of the plasma boundary; (for ISHAPE = 1)

MF $M$  = number of Fourier coefficients ( $M_2$  in Eq. (15)  $M_2 \geq M_1$ ); (for ISHAPE = 2)

FM( $m$ ) = Fourier coefficients  $f_m$  ( $m = 0, \dots, M_2$ ), (Eq. (15)); (for ISHAPE = 2)

PAR1-4 =  $P_1, \dots, P_4$ , see formula (16); (for ISHAPE = 3)

MHARM = number of harmonics used to describe boundary ( $M_1$  in Eq. (14));

ISOL = switch parameter, determines whether Soloviev equilibrium is used or not:

- 0: ISHAPE is used to determine shape of plasma boundary;
- 1: Soloviev equilibrium is used;

IMESH = switch parameter, determines whether mesh accumulation is used or not, for

IMESH.NE.0 a non-equidistant mesh in  $\sqrt{\bar{\psi}}$  is generated:

- 0: no mesh accumulation;
- 1: mesh accumulation with the formula
  - $\bar{\psi} = i^2 + A_m i^3 + B_m i^4 - (A_m + B_m) i^5$ , with  $0 \leq i \leq 1$
  - (see subroutine RADMESH);



2: different mesh accumulation formula, viz.

$$\bar{\psi} = i^2 + A_m i^{C_m} (i - B_m)^{C_m}, \text{ with } 0 \leq i \leq 1$$

(see subroutine RADMESH);

AMESH = parameter  $A_m$  for mesh accumulation; (for IMESH = 1 or 2)

BMESH = parameter  $B_m$  for mesh accumulation; (for IMESH = 1 or 2)

CMESH = parameter  $C_m$  for mesh accumulation; (for IMESH = 2)

### 3.3.2 Namelist PROFILE

IGAM = switch parameter, determines whether the profile  $\Gamma(\bar{\psi})$  (see Eq. (8)) is specified or  $\bar{F}(\bar{\psi})\bar{F}'(\bar{\psi})$  and *how* this input profile is specified:

- 1:  $\Gamma(\bar{\psi})$  is given as 8th order polynomial with coefficients AGA, BGA, ..., HGA, i.e.  $\Gamma(\bar{\psi}) = 1 + a_\gamma \bar{\psi} + \dots + h_\gamma \bar{\psi}^8$ ;
- 2:  $\Gamma(\bar{\psi})$  is given as 8th order polynomial (as for IGAM = 1) in the inner region  $\bar{\psi} < 2 \text{ CB2} - 1$  and by  $\Gamma(\bar{\psi}) = 1 - \bar{\psi} + \text{CB1} \left[ 1 - \left( (\bar{\psi} - \text{CB2}) / (1 - \text{CB2}) \right) ** (2 \text{ CB3}) \right]$  in the outer region  $\bar{\psi} > 2 \text{ CB2} - 1$ ;
- 3:  $\Gamma(\bar{\psi})$  is given as 8th order polynomial (as for IGAM = 1) in the inner region  $\bar{\psi} < 2 \text{ CB2} - 1$  and *modified in the outer region  $\bar{\psi} > 2 \text{ CB2} - 1$  by adding the profile  $\text{CB1} \left[ 1 - \left( (\bar{\psi} - \text{CB2}) / (1 - \text{CB2}) \right) ** (2 \text{ CB3}) \right]$* ;
- 4:  $\Gamma(\bar{\psi})$  is read in via the namelist input DF2 on an equidistant  $\bar{\psi}$ -mesh (not necessarily normalised to 1);
- 5:  $\bar{F}(\bar{\psi})\bar{F}'(\bar{\psi})$  is given as 8th order polynomial with coefficients AGA, BGA, ..., HGA, i.e.  $\bar{F}(\bar{\psi})\bar{F}'(\bar{\psi}) = 1 + a_\gamma \bar{\psi} + \dots + h_\gamma \bar{\psi}^8$ ;
- 6: input profile is  $\bar{F}(\bar{\psi})\bar{F}'(\bar{\psi})$ , given as 8th order polynomial in  $\bar{\psi}$  with coefficients fixed coefficients (special case), specified in FUNCTION DGDPSI (used as test/comparison);
- 7:  $\bar{F}(\bar{\psi})\bar{F}'(\bar{\psi})$  is read in via the namelist input DF2 on an equidistant  $\bar{\psi}$ -mesh (not necessarily normalised to 1);
- 8:  $\bar{F}(\bar{\psi})\bar{F}'(\bar{\psi})$  is read in via the namelist input DF2 on an equidistant  $\bar{\psi}$ -mesh (not necessarily normalised to 1) and *modified in the outer region  $\bar{\psi} > 2 \text{ CB2} - 1$  by adding the profile  $\text{CB1} \left[ 1 - \left( (\bar{\psi} - \text{CB2}) / (1 - \text{CB2}) \right) ** (2 \text{ CB3}) \right]$* ;

AGA-HGA = coefficients  $a_\gamma, \dots, h_\gamma$  of the 8th order polynomial for  $\Gamma(\bar{\psi})$  for the cases IGAM = 1 - 4, and for  $\bar{F}(\bar{\psi})\bar{F}'(\bar{\psi})$  for the cases IGAM = 5;

CB1-CB3 = coefficients used to change the profile  $\Gamma(\bar{\psi})$  in the cases IGAM = 2 and 3 and to change the profile  $\bar{F}(\bar{\psi})\bar{F}'(\bar{\psi})$  in the case IGAM = 8;

DF2 = input profile for  $\Gamma(\bar{\psi})$  in the case IGAM = 4 and for  $\bar{F}(\bar{\psi})\bar{F}'(\bar{\psi})$  in the cases IGAM = 7 and 8;

IPAI = switch parameter, determines how  $\Pi(\bar{\psi})$  ( $\sim \bar{F}'(\bar{\psi})$ ) (see Eq. (8)) is specified, analogous to IGAM:

- 1:  $\Pi(\bar{\psi})$  is given as 8th order polynomial with coefficients API, BPI, ..., HPI, i.e.  $\Pi(\bar{\psi}) = 1 + a_\pi \bar{\psi} + \dots + h_\pi \bar{\psi}^8$ ;
- 2: as IPAI = 1;

- 3:  $\Pi(\bar{\psi})$  is given as 8th order polynomial (as for  $\text{IPAI} = 1$ ) in the inner region  $\bar{\psi} < 2 \text{CP2} - 1$  and *modified in the outer region*  $\bar{\psi} > 2 \text{CP2} - 1$  *by adding the profile*  $\text{CP1} \left[ 1 - \left( (\bar{\psi} - \text{CP2}) / (1 - \text{CP2}) \right) ** (2 \text{CP3}) \right]$ ;
- 4: as  $\text{IPAI} = 1$ ;
- 5: as  $\text{IPAI} = 1$ ;
- 6: input profile is  $\bar{P}'(\bar{\psi})$ , given as 8th order polynomial in  $\bar{\psi}$  with coefficients fixed coefficients (special case), specified in FUNCTION  $\text{DPDPSI}$  (used as test/comparison);
- 7:  $\bar{P}'(\bar{\psi})$  is read in via the namelist input  $\text{DPR}$  on an equidistant  $\bar{\psi}$ -mesh (not necessarily normalised to 1);
- 8:  $\bar{P}'(\bar{\psi})$  is read in via the namelist input  $\text{DPR}$  on an equidistant  $\bar{\psi}$ -mesh (not necessarily normalised to 1) and *modified in the outer region*  $\bar{\psi} > 2 \text{CP2} - 1$  *by adding the profile*  $\text{CP1} \left[ 1 - \left( (\bar{\psi} - \text{CP2}) / (1 - \text{CP2}) \right) ** (2 \text{CP3}) \right]$ ;

$\text{API-HPI}$  = coefficients  $a_\pi, \dots, h_\pi$  of the 8th order polynomial for  $\Pi(\bar{\psi})$  for the cases  $\text{IPAI} = 1 - 4$ , and for  $\bar{P}'(\bar{\psi})$  for the cases  $\text{IPAI} = 5$ ;

$\text{CP1-CP3}$  = coefficients used to change the profile  $\Pi(\bar{\psi})$  in the cases  $\text{IPAI} = 2$  and  $3$  and to change the profile  $\bar{P}'(\bar{\psi})$  in the case  $\text{IPAI} = 8$ ;

$\text{DPR}$  = input profile for  $\Pi(\bar{\psi})$  in the case  $\text{IPAI} = 4$  and for  $\bar{P}'(\bar{\psi})$  in the cases  $\text{IPAI} = 7$  and  $8$ ;

$\text{NPTS}$  = number of points on which input profiles  $\text{DF2}$  ( $\text{IGAM} = 4, 7, \text{ and } 8$ ) and  $\text{DPR}$  ( $\text{IPAI} = 4, 7, \text{ and } 8$ ) are given.

### 3.3.3 Namelist PHYS

$\text{EPS}$  = inverse aspect ratio, i.e.  $a/R_0$ ;

$\text{ALFA}$  = parameter  $\alpha$ , defined in the introduction; can be used to scale the total current,  $q_0 \sim \alpha$  (approximately),  $\beta \sim 1/\alpha^2$ ,  $\bar{I} \sim 1/\alpha$  (so that  $g \sim 1/\alpha$  and  $g_N \sim 1/\alpha$ , for definition of  $g$  and  $g_N$  see Sect. 3.5.1) while  $\beta_p$  does not depend on  $\alpha$ ;

$\text{B}$  = parameter  $B$  defined in the introduction; determines the input amplitude of the pressure profile;

$\text{C}$  = parameter introduced once to *artificially* scale the  $\bar{F}\bar{F}'$  profile with respect to the  $\bar{P}'$  profile; for normal use:  $\text{C} = 1!$  (= initialisation).

### 3.3.4 Namelist NUM

$\text{NR}$  = number of radial grid points;

$\text{NP}$  = number of poloidal grid points (power of  $2 + 1$ );

$\text{NRMAP}$  = number of radial grid points for the mapping;

$\text{NPMAP}$  = number of poloidal grid points for the mapping (power of  $2 + 1$ );

$\text{NITER}$  = maximum number of iterations with fixed grid, after  $\text{NITER}$  iterations the resolution is doubled in both spatial directions;

$\text{NMESH}$  = maximum number of times grid is adapted (resolution doubled);

$\text{AMIX}$  = mixing of results for faster convergence:  $\bar{\psi}_{i+1} = (1 - \text{AMIX}) \bar{\psi}_{i+1} + \text{AMIX} \bar{\psi}_i$ .

### 3.4 Submitting a job

With the compilation command in Sect. 3.2, the executable file has been named `hel8`. When the namelist input file (see above) is called allocated to unit 10, i.e. when it is called `fort.10`, HELENA could be run interactively by simply giving the command ‘`hel8`’. However, HELENA requires a lot of core memory and usually the amount of core memory is limited for interactive jobs. Therefore, HELENA is usually run in batch.

When submitting HELENA in batch, memory and maximum run time have to be specified. These parameters determine the jobclass in which the job will run. HELENA 8 uses typically less than 20 Mwords and less than 60 seconds CPU time (on a Cray C98, but depends on mesh size and number of iterations). Also, the namelist input file needs to be allocated to unit 10, i.e. the input file must be called `fort.10`. The following example script makes a new directory, copies the file data to `fort.10`, runs `hel8`, and renames the output files (see below) on a Cray computer.

---

```
#!/bin/sh
# This shell script, HELENA, runs hel8 in batch
# Usage: qsub HELENA

# Job parameters
# =====
#QSUB -lM 20.0mw
#QSUB -lT 60
#QSUB

# File definitions
# =====
mkdir $HOME/helena/KTE004
cd $HOME/helena/KTE004
cp $HOME/helena/data fort.10

# Run
# ===
hpm $HOME/helena/hel8 < fort.10

# Output
# =====
mv fort.12 KTE004.mapping
mv fort.20 KTE004.listing
mv fort.21 KTE004.vector
mv PCUBEQ KTE004.plot
rm fort.10
rm fort.52
# End of shell script
```

---

## 3.5 Output files

HELENA output consists of four files, viz. a ‘listing’ with information on the convergence procedure and the equilibrium that has been found, a ‘mapping file’ with the information on the equilibrium that is required for CASTOR (this file is an input file for CASTOR), a ‘vector plot file’ which is required as input to make flow plots of CASTOR results, and a ‘plot file’ which is a postscript file with a plot of the mesh and of the equilibrium profiles. More details on each of these output files is given below in the following subsections.

### 3.5.1 Listing

The listing is printed out on unit 20. The listing contains:

- 1) a print out of the input parameters;
- 2) information on the iteration procedure, such as
  - a)  $x$ -position of magnetic axis;
  - b) value of  $\bar{\psi}$  at magnetic axis;
  - c) relative error;
- 3) information on CPU-time consumption of several subroutines;
- 4) information on the shape of the boundary ( $\theta, r = f(\theta)$ );
- 5) the radial profiles  $RB_\phi(\bar{\psi}), \bar{\psi}, q(\bar{\psi}), \bar{P}(\bar{\psi})$ ;
- 6) Dimensionless plasma parameters:

- $\text{BETA} \equiv \beta \equiv \frac{2\epsilon}{\alpha^2} \langle \bar{P} \rangle$       and       $\text{BETAPL} \equiv \beta_p \equiv \frac{8\pi \bar{S} \langle \bar{P} \rangle}{\bar{I}^2}$ ;

- $\text{TOTAL CURRENT} \equiv \bar{I} \equiv \frac{\mu_0}{a[m]B_0[T]} I_p[A]$ ;

- Troyon factor [7]:

$$g \equiv \frac{a[m]B_0[T]}{I_p[MA]} \beta(\%) \equiv \frac{\mu_0 10^8}{\bar{I}} \beta \quad \text{and} \quad g_N \equiv 100 \frac{\beta}{\bar{I}};$$

- 7) the radial profile  $\langle \bar{J}_\phi(\bar{\psi}) \rangle$  and the half-length of the poloidal circumference of the flux surfaces;
- 8) error messages;

### 3.5.2 Mapping file

The mapping file is printed out on unit 12. The profiles are scaled to  $q(0) = 1$ . The mapping file contains the following information (a prime denotes a derivation with respect to  $s$ ):

- 1) JS0: gives number of radial grid points, viz. JS0+1;
- 2) (CS(JS), JS=1, JS0+1): the JS0+1 values of  $s = \sqrt{\bar{\psi}}$  ( $0 \leq s \leq 1$ );
- 3) (QS(JS), JS=1, JS0+1): the JS0+1 values of  $q(s)$ ;
- 4) DQS(1), DQEC:  $q'(0)$  and  $q'(1)$ ;
- 5) (DQS(JS), JS=2, JS0+1): the JS0 values of  $q'(s)$  (for  $s \neq 0$ );

- 6) (CURJ(JS), JS=1, JS0+1): the JS0+1 values of  $\bar{J}(s)$  (NOT USED!);
- 7) DJ0, DJE:  $\bar{J}'(0)$  and  $\bar{J}'(1)$  (NOT USED!);
- 8) NCHI: the number of poloidal grid points;
- 9) (CHI(JS), JS=1, NCHI): the NCHI values of  $\chi$  ( $0 \leq \chi \leq \pi$ );
- 10) (GEM11(JS), JS=NCHI+1, (JS0+1)\*NCHI): the values of  $\nabla\bar{\psi} \cdot \nabla\bar{\psi}$  in every grid point ( $s \neq 0, \chi$ );
- 11) (GEM12(JS), JS=NCHI+1, (JS0+1)\*NCHI): the values of  $\nabla\bar{\psi} \cdot \nabla\chi$  in every grid point ( $s \neq 0, \chi$ );
- 12) CPSURF, RADIUS: total (dimensionless) poloidal flux and  $a/R_{\text{mag}}$ , where  $a$  is the small radius of the torus and  $R_{\text{mag}}$  is the large radius of the magnetic axis;
- 13) (GEM22(JS), JS=NCHI+1, (JS0+1)\*NCHI): the values of  $\nabla\varphi \cdot \nabla\varphi \equiv 1/R^2$  in every grid point ( $s \neq 0, \chi$ );
- 14) RAXIS/(1.+EPS\*XAXIS): large radius (normalized to unity);
- 15) (P0(JS), JS=1, JS0+1): the JS0+1 values of  $\bar{P}(s)$  (normalized to  $q(0) = 1$ );
- 16) DP0, DPE:  $\bar{P}'(0)$  and  $\bar{P}'(1)$ ;
- 17) (RBPHI(JS), JS=1, JS0+1): the JS0+1 values of  $RB_\phi(s)$  (normalized to  $q(0) = 1$ );
- 18) DRBPHI0, DRBPHIE:  $(RB_\phi)'(0)$  and  $(RB_\phi)'(1)$ ;
- 19) (VX(JS), JS=1, NCHI):  $x$ -values of grid points on outer flux surface;
- 20) (VY(JS), JS=1, NCHI):  $y$ -values of grid points on outer flux surface;
- 21) EPS: inverse aspect ratio, i.e.  $a/R_0$ .

### 3.5.3 Vector plot file

On unit 21, HELENA writes information on the mapping (grid). With this information and information on an eigenvector found by CASTOR, the program `flowplot.f` can make a flow plot of that eigenvector. Hence, the output file `fort.21` of HELENA is an input file for `flowplot.f` in case a flow plot is wanted of a CASTOR result.

### 3.5.4 Plot file

In the PPPLIB postscript output file 'PCUBEQ', HELENA plots the grid that has been found and the radial profiles of the normalized profiles of pressure and current density, the  $q$ -profile, and the radial profile of the flux  $\bar{\psi}$ .

## 3.6 Connection with HBT

It may occur that one wants to check equilibria constructed with HELENA on ballooning stability. This cannot be done by CASTOR, but the program HBT [3] has a module BALON that checks on ballooning stability. Vice versa, one may want to study the stable part of the spectrum of equilibria constructed with HBT, e.g. to study 'gap modes' for these equilibria. In this section, it is described how HBT equilibria can be reconstructed with HELENA and vice versa.

HBT	HELENA	this guide	used in or defined by
PAR1	ELLIP	$E$ , ellipticity	Eq. (11)
PAR2	TRIA	$\tau$ , triangularity	Eq. (11)
PAR3	QUAD	$\kappa$ , quadraticity	Eq. (11)
MFM	MFM	$M_2$	Eq. (15)
-	PAR1 - PAR4	$P_1, \dots, P_4$	Eq. (16)
-	AMESH, BMESH, CMESH	$A_m, B_m, C_m$	Namelist SHAPE
MHARM	MHARM	$M_1$	Eq. (14)
<sup>-1</sup>	AGA, BGA, ..., HGA	$a_\gamma, b_\gamma, \dots, h_\gamma$	Namelist PROFILE
<sup>-1</sup>	API, BPI, ..., HPI	$a_\pi, b_\pi, \dots, h_\pi$	Namelist PROFILE
EPS	EPS	$\epsilon$	Eq. (5)
EBETPL	EPS*BETAPL	-	$\epsilon\beta_p$
BETAA	BETA	$\beta$	Sec. 3.5.1
ELONG	-	-	$1/\pi \times$ half length of the $\bar{\psi} = 1$ surface
DEL	-	-	$x$ -position of the magnetic axis
ALFA <sup>2</sup>	ALFA <sup>2</sup>	$\alpha$	Eq. (5)
B <sup>2</sup>	B <sup>2</sup>	B	Eq. (8)
QS	-	-	Eq. (17)
Q0	QS(1)	$q_0 = q(0)$	Sec. 3.5.2
Q1	QS(JS0+1)	$q_1 = q(1)$	Sec. 3.5.2
Q0W	-	-	$q_0/\alpha$

<sup>1</sup> Notice that the same input parameters in HBT *can have a different meaning!*

<sup>2</sup> *Output* parameter of HBT, *input* parameter for HELENA

**Table 1:** The meaning of HBT in- and output in HELENA terminology

### 3.6.1 HBT $\Rightarrow$ HELENA

This way is simple as long as one makes use of the small overlap in the possibilities to specify the two input profiles in both codes. The definition of  $\Gamma(\bar{\psi})$  and  $\Pi(\bar{\psi})$  is the same in both codes and to some extent they can be specified in the same way in both HELENA and HBT (see [3] for more information on the specification of the two input profiles). The shape of the poloidal cross-section can also be specified in the same way in both codes. Moreover, HBT prints out the values of the parameters B and ALFA. Hence, as long as the shape of the cross-section and the two input profiles are specified in the same way in both codes, it is easy to reconstruct an HBT equilibrium of HBT with HELENA: it suffices to specify the same values for B and ALFA and HELENA will converge to the same equilibrium with the same  $\beta$  and  $\beta_p$  (check!).

### 3.6.2 HELENA $\Rightarrow$ HBT

The reconstruction of HELENA equilibria with HBT is also trivial as long as the two input profiles and the shape of the cross-section are specified in the same manner because HELENA prints out the parameter  $\delta$  which is needed as input for HBT:  $\delta$  is the  $x$ -position of the magnetic axis (see HELENA output, Sect. 3.5.1, item 2a). Another difference of the input for the two codes is the way the current profile is scaled. HBT requires a parameter  $q^*$  (QS), which scales inversely to the total current (just as  $\alpha$ ), as input. This parameter is defined by

$$q^* \equiv \frac{aB_0L}{\mu_0 R_0 I_p} = \frac{\epsilon L}{\bar{I}}, \quad (17)$$

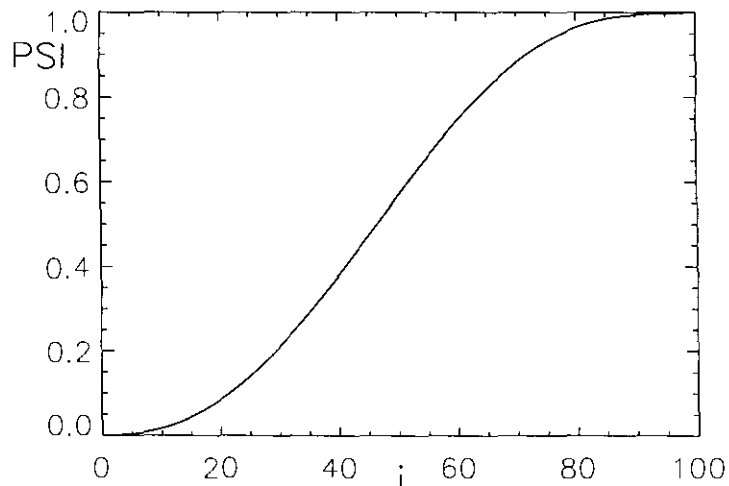
where  $L$  ( $= 2\pi$  ELONG) is the poloidal circumference of the plasma boundary. This input parameter can be derived from the HELENA output file:  $\epsilon$  and  $\bar{I}$  are printed out and  $L$  corresponds to twice the (half) length of the outer flux surface, as printed out at the end of the HELENA listing (see Sect. 3.5.1, item 7). Hence, it is recommended to specify the profiles and the shape of the cross-section in the same way in both codes and to specify  $\delta$  and  $q^*$  as obtained from the HELENA output. HBT will then converge to the same equilibrium with the same  $\beta$  and  $\beta_p$  (check!).

Finally, in Table 1 some output parameters of HBT are described in HELENA terminology in order to enable a closer comparison of the output of both codes.

### 3.7 Handy hints

Last but not least, below some hints are listed that may be handy when constructing equilibria with HELENA, e.g. for MHD stability analysis:

- The output file on unit!20 should be checked for the message ‘node missed’. If this occurs, the mapping file is corrupted. This can be cured by changing the number of radial or poloidal grid points (NR, NP, NRMAP, NPMP).
- For equilibria with a large shear near the plasma boundary it is advisable to accumulate the mesh near the plasma boundary; the parameter values IMESH = 1, AMESH = 8, and BMESH = -14.1 usually yield a satisfactory result. The resulting  $\bar{\psi}$ , defined in Sect. 3.3.1 (in Namelist SHAPE) for this parameter choice is plotted below.



- To make an equilibrium with given shape of the cross-section,  $\epsilon$ ,  $\beta$ , input profiles, and total current, proceed as follows:
  1. run HELENA with the given profiles,  $\epsilon$ , cross-section shape and with  $\alpha=1$  and  $B = 1$ ,
  2. adjust  $\alpha$  such that the total current is correct:  $\alpha_{\text{new}} = \bar{I}_{\text{old}}\alpha_{\text{old}}/\bar{I}_{\text{given}}$ ,
  3. keep  $\epsilon$  and  $\alpha$  fixed such that  $\bar{I}$  remains correct and adjust input parameter B to get the right  $\beta$ .



# A REVISE

## A.1 Introduction

The REVISE package provides a portable facility for the systematic maintenance of large computer programs which combines the positive features of updating and screen editing (rigor and speed), while avoiding the negative ones (slowness and generation of errors). The REVISE package was developed to control and exchange the progression of changes, avoiding the introduction of mistakes, informing fellow collaborators in an international group of scientists of new versions of the numerical tools they are all using for different purposes on different computers.

The basic information of a current version of a changing program (deck names, numbered lines, index, etc.) is stored into a LIST file which serves as a starting point for comparison with later versions of the same program. A REVISE LIST file has all COMMON blocks in the beginning and contains switches for parts of the program (a single Fortran statement or a block of statements) that need to be different at different sites and/or on different computers.

## A.2 Contents and use

REVISE consists of five programs, written in Fortran 77:

- 1) `new.f` makes a listing `fn.list` out of a source file `fn.source`. This listing contains information e.g. date and time it was made, absolute line numbers, line numbers per subroutine, and an index at the end. The optional parameter `1` in `new` (and also in `mod`) controls the output format (it suppresses carriage control characters, see the comment in the Unix script files `new` and `mod`).
- 2) `pre.f` is a pre-compiler that converts a REVISE source file `fn.source` into a Fortran file `fn.f` which then has to be compiled with a standard Fortran compiler in order to get an executable file. The pre-compiler places the COMMON blocks where necessary and selects Fortran statements applicable on the specified computer at the specified site.
- 3) `com.f` makes a list of modifications from an old listing (previous version of a code) and a new source file (new version of a code).
- 4) `mod.f` makes a new listing out of an old listing and a list of modifications.
- 5) `ext.f` makes a source file out of a listing (opposite of `new.f`).

In `new.f` and in `mod.f` the parameter `ILMAX` (maximum number of lines per page) needs to be set (depending on local printer) and calls to `DATE` and `TIME` functions may need to be adjusted (depending on the computer that is used, e.g. `DATE` and `CLOCK` on Cray but `DATIM` on IBM(VM)). In `pre.f` the switches for single and block line activation need to be set (depending on site and computer).

After compilation one gets executable files, viz. `new.o`, `pre.o`, `com.o`, `mod.o`, and `ext.o` (when these names are specified with the `-o` option). However, one would then have to allocate all necessary files explicitly. Therefore, scripts have been written which take care of the proper file allocations. The call, the parameters, the input files, and the generated output files are specified in Table A.1. For the parameters `fn`, `nfn`, `mfn` the user has to specify the file names. If `mfn` is omitted in `com`, a file name "m+nfn.modif"

Call	Input files	Output files
<b>new</b> fn [l]	fn.source	fn.list
<b>pre</b> fn [l]	fn.source	fn.f
<b>com</b> fn nfn [mfn]	fn.list nfn.source	mfn.modif
<b>mod</b> fn mfn [nfn] [l]	fn.list mfn.modif	mfn.listm nfn.listr
<b>ext</b> fn	fn.list(r)	fn.source

**Table A.1:** The REVISE Unix script files

will be constructed. If nfn is omitted in mod, creation of “nfn.listr” will be suppressed. The optional parameter l in new and mod controls output format. When the scripts are called without any input parameters or with wrong or insufficient input, information on the required input parameters is ‘echoed’ on the standard output device.

### A.3 Installation of REVISE

On Unix systems, REVISE can be installed in the following way:

1. make a directory ‘revise’;
2. copy the files new.f, pre.f, com.f, mod.f, ext.f, and revise.install in this directory;
3. edit the files new.f, com.f, and pre.f to adjust the parameters to your site or computer as described in the previous section;
4. replace the fortran compiler call cf77 in the file revise.install if your fortran compiler is different, e.g. cf77 needs to be replaced by xlf on IBM RISC workstations;
5. execute revise.install, this creates the files new.o, pre.o, com.o, mod.o, and ext.o;
6. copy the script files pre, new, com, mod, and ext to the directory ‘revise’ and put this directory in your search PATH.

### A.4 IBM-VM version of REVISE

The above descriptions of the contents, use, and installation of revise apply to Unix systems. There is also a version of REVISE available for IBM with the VM operating system.

Call	Input files	LRECL	Output files	LRECL
NEW fn [l]	fn SOURCE A	80	fn LIST A	93
PRE fn [l]	fn SOURCE A	80	fn FORTRAN A	80
COM fn nfn [mfn]	fn LIST A	93	mfn MODIF A	80
	nfn SOURCE A	80		
MOD fn mfn [nfn] [l]	fn LIST A	93	mfn LISTM A	93
	mfn MODIF A	80	nfn LISTR A	93
EXT fn	fn LIST(R) A	93	fn SOURCE A	80

**Table A.2:** The REVISE VM EXEC files

In this case, the fortran files are called NEW FORTRAN A, PRE FORTRAN A, COM FORTRAN A, MOD FORTRAN A, and EXT FORTRAN A, respectively. The five programs are compiled with the FORTVS2 compiler, without optimisation or vectorisation. This produces five files NEW TEXT A, PRE TEXT A, COM TEXT A, MOD TEXT A, and EXT TEXT A.

The script files are replaced by EXEC files written in REXX but the call and the parameters are essentially the same as for the Unix scripts and are specified in Table A.2. Again, for the parameters fn, nfn, mfn the user has to specify the file names. If mfn is omitted in COM, a file name "Mfn MODIF" will be constructed. If nfn is omitted in MOD, creation of "nfn LISTR" will be suppressed. All files have RECFM F and LRECL as indicated in the table. The optional parameter l in NEW and MOD controls output format.

## A.5 List of files

The complete list of files required to install REVISE consists of:

- the 5 fortran files new.f, pre.f, com.f, mod.f, and ext.f (or NEW FORTRAN A, PRE FORTRAN A, COM FORTRAN A, MOD FORTRAN A, and EXT FORTRAN A, respectively, on IBM VM),
- the 5 script files new, pre, com, mod, and ext (or the 5 EXEC files NEW, PRE, COM, MOD, and EXT on IBM VM),
- the installation file revise.install,
- the information file revise.note (contains a short description of REVISE and on how to use it on VM),

i.e. 12 files in total.

## B Installation script for PPPLIB

The following Unix script is called 'ppplib.mk'. It installs PPPLIB version 15e from input file \$HOME/ppplib/ppp15e.source and puts it in the directory \$HOME/libs.

---

```
#This shell script ppplib.mk constructs the ppplib.a library
echo =====
echo Making working directory
echo =====
mkdir $HOME/wd
echo =====
echo Copying ppp15e.source
echo =====
cp $HOME/ppplib/ppp15e.source $HOME/wd/ppplib.source
cd $HOME/wd
echo =====
echo Precompiling ppplib.source
echo =====
pre ppplib
echo =====
echo Splitting ppplib.f
echo =====
fsplit ppplib.f
echo =====
echo Cat BLOCK DATA subpograms
echo =====
cat begplt.f lhead.f pos.f calpos.f ebdasc.f > cat.f
mv cat.f begplt.f
echo =====
echo Compiling all ppplib routines
echo =====
\rm ppplib.f
cf77 *.f -c
\rm *.f
echo =====
echo Creating library ppplib.a
echo in directory $HOME/libs
echo =====
ar vq ppplib.a *.o
ranlib ppplib.a
mkdir $HOME/libs
mv ppplib.a $HOME/libs/ppplib.a
\rm *
cd $HOME
rmdir wd
#end of shell script
```

---

## C Installation script for HGOLIB

The following script, hgolib.mk, creates a library from input file \$HOME/libs/hgolib.f and puts it in the directory \$HOME/libs (which is constructed if it does not yet exist).

---

```
#This shell script hgolib.mk constructs the hgolib.a library
echo =====
echo Making working directory
echo =====
mkdir $HOME/wd
echo =====
echo Copying hgolib.f
echo =====
cp $HOME/libs/hgolib.f $HOME/wd/hgolib.f
cd $HOME/wd
echo =====
echo Splitting hgolib.f
echo =====
fsplit hgolib.f
echo =====
echo Compiling all hgolib routines
echo =====
\rm hgolib.f
cf77 *.f -c
\rm *.f
echo =====
echo Creating library hgolib.a
echo in directory $HOME/libs
echo =====
ar vq hgolib.a *.o
ranlib hgolib.a
mkdir $HOME/libs
mv hgolib.a $HOME/libs/hgolib.a
\rm *
cd $HOME
rmdir wd
#end of shell script
```

---

## References

- [1] G. Huijsmans, Ph.D. thesis, Vrije Universiteit Amsterdam, 1991, "External resistive modes in tokamaks"
- [2] J.P. Goedbloed, *Comp. Phys. Comm.*, **24**, 311-321 (1981), "Conformal mapping methods in two-dimensional magnetohydrodynamics"
- [3] G.T.A. Huysmans, R.M.O. Galvão, and J.P. Goedbloed, *Rijnhuizen Report 90-194*, 1990, "Documentation of the high beta stability codes HBT and HBTAS at JET"
- [4] G.T.A. Huysmans, J.P. Goedbloed, and W. Kerner, *Proc. CP90 Conf. on Comp. Phys. Proc.*, World Scientific Publ. Co., 1991, p. 371, "Isoparametric Bicubic Hermite Elements for Solution of the Grad-Shafranov Equation"
- [5] G.T.A. Huysmans and J.P. Goedbloed, *Rijnhuizen Report 93-218*, 1993, "Low-n ideal and resistive MHD stability of JET discharges"
- [6] L.S. Soloviev, *Reviews of Plasma Physics*, Vol. 6, ed. M.A. Leontovich (Consultants Bureau, New York), 1975, p. 257.
- [7] F. Troyon, A. Roy, W.A. Cooper, F. Yasseen, A. Turnbull, *Plasma Phys. and Contr. Fusion*, **30**, 1597-1609 (1988), "Beta limit in tokamaks: experimental and computational status."