ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Genova

INFN/TC-90/04
27 Febbraio 1990

P. Fernandes and R. Parodi:

OSCAR2D USER'S GUIDE

Servizio Documentazione
dei Laboratori Nazionali di Frascati

Popine

2

France

INFN - Istituto Nazionale di Fisica Nucleare
Sezione di Genova

INFN/TC-90/04
27 Febbraio 1990

OSCAR2D USER'S GUIDE

P. Fernandes
IMA - CNR, Genoa (Italy)

R. Parodi
INFN - Genoa (Italy)

Program version: V2.0
Release date: November 3, 1989
Last modification after release: Febbruary 13, 1990

CONTENTS

GENERAL INFORMATION

Our code OSCAR2D has been implemented starting from the old LALA [1,2,3] program (by Hoyt of Los Alamos National Laboratories) so it keeps the general style of it. However OSCAR2D has several new features and capabilities which make it much more powerful that its "ancestor". The most important of them are listed below.

1. The only bound to the number of mesh points you can use in the discretization of your problem geometry is the available memory (and the time you can wait for the output, of course) since OSCAR2D makes use of dynamic allocation of memory.

2. A wider class of problem geometries is allowed since the mesh generator has been extended. Particularly, OSCAR2D can deal with structures whose cross-section is multiply connected.

3. Structures made of two or more conducting materials are allowed (but not dielectric or magnetic insertions).

4. OSCAR2D can compute azimuthally symmetric TM and TE modes of both single cavities and multicell structures having cylindrical symmetry [4,5,7].

5. OSCAR2D can compute TM and TE modes of finite length constant cross-section resonant structures [7].

6. OSCAR2D can compute TM and TE modes of constant cross-section waveguides [7].

7. OSCAR2D can compute the n lowest resonant modes in a single run as follows:
   (i) first, the Point Successive Over-Relaxation iterates are kept orthogonal to all the previously computed modes to produce an approximation to the next one;
   (ii) then, starting from this approximation, the next mode is computed by a Rayleigh Quotient Iteration where the linear system is solved by the Bi-Conjugate Gradient method [9].

8. Given an approximation of the frequency of the n-th mode and some qualitative information about its field, OSCAR2D can compute it without computing the n-1 lowest modes by a Rayleigh Quotient. Iteration where the linear system is solved by the Bi-Conjugate Gradient method [9].

9. The fields and the frequencies computed by OSCAR2D are more accurate than those computed by LALA because of:
   (i) a better treatement of the boundary conditions [5,6],
   (ii) the use in the iteration of the Rayleigh quotient of the discretized problem, instead of the one of the continuous problem which is now used only at the end [5].
   (iii) the use of the Rayleigh Quotient Iteration plus the Bi-Conjugate Gradient method to solve the matrix eigenvalue problem [9].

FORMAT OF THE INPUT DATA

The first line of the input is a character string (up to 80 characters) identifying the run. It will be printed as the first line of the output.
The remaining lines of the input are read by the following statements:

NAMELIST/NINPUT/ <list of variables and arrays>
..
READ(5,NINPUT)

Thus the input file must have the following format:

<character string identifying the run>
<SP>$NINPUT
<SP><variable>=<value>
or
<SP><array>=<list of values>
..
<SP>$END

where <SP> indicates a space.
The variables and the arrays in the NAMELIST/NINPUT/ are described in that follows.
For details about the sintax of the input data for a READ by NAMELIST see the FORTRAN manual.
Many of the input data have been preset so that they need not be entered if the default value is to be used.

INPUT DATA DEFINING THE PROBLEM

1. CART= Coordinates definition.

   i 0 Axially symmetric problem in (R,PHI,Z) cylindrical coordinates.

   ii 1 Constant cross-section problem (waveguides and cavities) in (X,Y,Z) rectangular coordinates.
   Full problems only are allowed in this case: do not exploit symmetries to reduce the number of mesh points.

   (Preset to 0)

2. ZLEN= Length in the Z direction of a constant cross-section cavity in rectangular coordinates (to be used only in connection with CART=1). If set to 0 the program computes cut off frequency and fields of a infinitely long waveguide of constant cross-section.
   If ZLEN.ne.0 then ZORD is required.
   Even though from a logical point of view this input contributes to define the problem, it is actually used by the program only to generate the final output. So, if you want to rerun a problem by just changing it, you can set NOCOMP=1 and LOAD=0 to save CPU time.
   (Preset to 0)

3. ZORD= Mode orders in the Z direction (up to 100) for a finite length cavity (to be used in connection with CART=1 and nonzero ZLEN)
   Even though from a logical point of view this input contributes to define the problem, it is actually used by the program only to generate the final output. So, if you want to rerun a problem by just changing it, you can set NOCOMP=1 and LOAD=0 to save CPU time.
   (Preset to 0)

4. TE= Mode type.

   i 1 TE modes

   ii 0 TM modes

   Notice that changing the mode type is not enough to switch from a TE mode computation to a TM mode computation since TYPE must be changed consistently.
   (Preset to 0)

5. NOL= number of line segments and arcs that are used to describe the geometry of the problem, including the 9999 separators (see below). Thus it is the number of values you will then give for each of RB, ZB, RC, ZC, R, R2, TYPE, METAL and SIGM.
   (1.le.NOL.le.40)

Since multiply connected regions are allowed, the boundary may consist of several closed curves: separate them from one another by a 9999 value in RB, ZB, RC, ZC, R, R2. The corresponding value of METAL and SIGM should be 0 while any value could be given to TYPE. A 9999 separator is not needed after the last curve.
While the external boundary must be given clockwise, the boundaries of possible internal "holes" must be given counterclockwise.

6. RB= R or Y coordinates of the starting points of each of the NOL segments. A 9999 value must be used to separate different closed curves.

7. ZB= Z or X coordinates of the starting points of each of the NOL segments. A 9999 value must be used to separate different closed curves.

8. RC= R or Y coordinates of the centers of the arcs. Zero for linear segments. Corresponding to a 9999 separator in ZB and RB, a 9999 value must be given to RC, too.
Can be omitted if just linear segments are used.

9. ZC= Z or X coordinates of the centers of the arcs. Zero for linear segments. Corresponding to a 9999 separator in ZB and RB, a 9999 value must be given to ZC, too.
Can be omitted if just linear segments are used.

10. R= radii of the circular arcs and horizontal semiaxis of the elliptical arcs. Positive if counterclockwise, negative if clockwise. Zero for linear segments. Corresponding to a 9999 separator in ZB and RB, a 9999 value must be given to R, too. Can be omitted if just linear segments are used.

11. R2= vertical semiaxis of the elliptical arcs. Positive if counterclockwise, negative if clockwise. Zero for linear segments, equal to R for circular arcs. Corresponding to a 9999 separator in ZB and RB, a 9999 value must be given to R2, too.
Can be omitted if just linear segments and circular arcs are used.

12. ZSHIFT= value to be added to any Z or X coordinate
(Preset to 0.)

13. RSHIFT= value to be added to any R or Y coordinate
(Preset to 0.)

14. LREFL= n the N-th segment is the left symmetry plane to be dealt with by reflection type boundary conditions.

15. RREFL= n the N-th segment is the right symmetry plane to be dealt with by reflection type boundary conditions.

16. METAL= to be set to 1 for all metallic surfaces.
Zero for ideal boundaries (i.e. symmetry planes and axes).
Corresponding to a 9999 separator in ZB and RB, a zero value should be given, too.
(Preset to 0).

17. TYPE= type of boundary condition to use.
Neumann = 0; Dirichlet = 1. Corresponding to a 9999 separator in ZB and RB, any value could be given to TYPE.
The values assigned to TYPE must be consistent with those assigned to CART, TE and METAL according to the following scheme.

    i Axially symmetric problem, TM modes (CART=0,TE=0):
    Neumann condition means electric field normal to the boundary. Dirichlet condition means electric field parallel to the boundary.
    On metallic boundaries use Neumann condition.

    ii Axially symmetric problem, TE modes (CART=0,TE=1):
    Neumann condition means magnetic field normal to the boundary. Dirichlet condition means magnetic field parallel to the boundary.
    On metallic boundaries use Dirichlet condition.

    iii Constant cross-section problem, TM modes (CART=1,TE=0):
    Dirichlet condition means magnetic field parallel to the boundary.
    On metallic boundaries use Dirichlet condition. Do not use Neumann condition: fields on boundary will be wrong.

    iv Constant cross-section problem, TE modes (CART=1,TE=1):
    Neumann condition means electric field normal to the boundary.
    On metallic boundaries use Neumann condition. Do not use Dirichlet condition: fields on boundary will be wrong.

18. H= mesh size. It is slightly adjusted by the program so that the structure length is an integral number of times the mesh size.

19. NBND= number of boundary points of the problem. To be used for the geometries for which the automatic computation of NBND in the code fails.

INPUT DATA DEFINING THE INITIAL APPROXIMATION.

1. FREQ= Guessed frequency to start the iteration loop (in Mhz).
For cylindrically symmetrical cavities it is a guess of the resonant frequency, for constant cross-section cavities and waveguides it is a guess of the cutoff frequency of the waveguide.
(Preset to 1.15E4/RMAX)

2. LOAD= Specify the initial guessed solution of the problem.
Several useful cases are covered but for particular problems modification by the user of the FLOAD function may be required.

   i 0 Previously computed field.
   A field and a frequency coming from a previous run on the same geometry are read from a disk file and used to start the iteration. The disk file must have the extension BIN; the default name is OSCAR and it can be changed by assigning to FILNAM the name you want.

   ii 1 Bessel function.
   The suitable Bessel function is selected for both TM and TE modes both in cylindrical and rectangular coordinates.
   In cylindrical coordinates use NXZ, XZER, NYZ to set the number of axial nodes and their positions and the number of radial nodes.
   In rectangular coordinates use CX, CY, NPHZ, PHZER, NRZ to set the center, the number of angular nodes and their positions and the number of radial nodes.

   iii 2 Rectangular step function.
   Use NXZ, XZER, NYZ, YZER to set vertical and horizontal nodes.

   iv 3 Polar step function.
   Use CX, CY, NPHZ, PHZER, NRZ, RZER to set center, angular nodes and radial nodes.

   v 4 Elliptical step function.
   Use CX, CY, A, B to set center and eccentricity of confocal ellipses and hyperbolas. Use NPHZ, PHZER to set nodes on branches of hyperbola and NRZ, RZER to set nodes on ellipses. In this case PHZER is the angle between the horizontal axis and the asymptote of the branch of hyperbola on which you want to set a node. PHZER must be in the range [0,PI] if A > B and in the range [-PI/2,+PI/2] if A < B. RZER is the minor semiaxis of the ellipse on which you want to set a node either if A > B or A < B. If A = B PHZER must be used as with LOAD=3 and in the range [0,2*PI].

   vi 5 Polar/elliptical general load.
   Use NOC to set the number of different polar or elliptical step functions you want. For each of them use CX, CY, A, B, NPHZ, PHZER, NRZ, RZER as with LOAD=4 and CS to set a sign.
   For a given point the polar or elliptical step function having the nearest center is always used.

   vii 6 Superposition of SIN(k1*X)*SIN(k2*Y) functions.
   Use NOSC to set the number of oscillations of the highest component in both X and Y.

viii 7 Superposition of SIN(k1*R)*SIN(k2*PHI) functions.
Use NOSC to set the number of oscillations of the highest component in both R and PHI.
(Preset to 1)

3. LOADBG= Control flag used to debug user written initial approximations.
If it is set to 1 the program plots the initial approximation and then stops.
It can also be useful to check whether the input data really define the initial approximation you want.
(Preset to 0)

4. NXZ= Number (<25) of nodes of the guessed solution along X or Z axis.
Used if LOAD=1 and CART=0 or if LOAD=2.
(Preset to 0)

5. XZER= X or Z coordinates of the NXZ nodes of the guessed solution (in increasing order).
Used if LOAD=1 and CART=0 or if LOAD=2.

6. NYZ= Number (<10) of nodes of the guessed solution along Y or R axis.
Used if LOAD=1 and CART=0 or if LOAD=2.
(Preset to 0)

7. YZER= Y or R coordinates of the NYZ nodes of the guessed solution (in increasing order).
Used if LOAD=2.

8. NOC= Number of centers (<10)of the polar and elliptical step functions in a polar/elliptical general load.
Used if LOAD=5.
(Preset to 1)

9. CX= X or Z coordinate of center of Bessel functions, polar step functions or elliptical step functions.
Used if LOAD=1 and CART=1 or LOAD=3,4 or 5.
When LOAD=5 NOC values of CX are needed, if only the first is given the others are set equal to that one (even though they are shown in the output as -1.).
(Preset to (XMAX+XMIN)/2.)

10. CY= Y or R coordinate of center of Bessel functions, polar step functions or elliptical step functions.
Used if LOAD=1 and CART=1 or LOAD=3,4 or 5.
When LOAD=5 NOC values of CY are needed, if only the first is given the others are set equal to that one (even though they are shown in the output as -1.).
(Preset to (YMAX+YMIN)/2. if CART=1, preset to 0. if CART=0)

11. CS= Sign of polar or elliptical step functions.
Used, if LOAD=5, to give a sign to each of the NOC step functions.
NOC values of CS are needed, if only the first is given the others are set equal to that one.
(Preset to 1.)

12. A= Horizontal semiaxis of the largest ellipse of an elliptical step function.
Together with B it defines the eccentricity of an elliptical step function.
Used if LOAD=4 or 5.
When LOAD=5 NOC values of A are needed, if only the firstis given the others are set equal to that one.

13. B= Vertical semiaxis of the largest ellipse of an elliptical step function.
Together with A it defines the eccentricity of an elliptical step function.
Used if LOAD=4 or 5.
When LOAD=5 NOC values of B are needed, if only the first is given the others are set equal to that one.

14. NPHZ= Number (<10) of angular nodes of a Bessel or polar step function or number of hyperbolical nodes of an elliptical step function.
Used if LOAD=1 and CART=1 or if LOAD=3,4 or 5.
When LOAD=5 NOC values of NPHZ are needed, if only the first is given the others are set equal to that one.
(Preset to 0)

15. PHZER= Angles between the horizontal axis and the NPHZ angular nodes of a Bessel or polar step function or the asymptotes of the NPHZ hyperbolical nodes of an elliptical step function (in increasing order).
Used if LOAD=1 and CART=1 or if LOAD=3,4 or 5.
If LOAD=1 or 3 or A=B then PHZER must be in the range [0,2*PI] else if A>B then PHZER must be in the range [0,PI] else if A<B then PHZER must be in the range [-PI/2,PI/2].
When LOAD=5 NOC sets of PHZER angles are needed (one for each step function), if only the first is given the others are set equal to that one. If different angular or hyperbolical nodes are desired for each step function, set the values for the n-th step function starting at the [10*(n-1)+1]-th element of the PHZER vector.

16. NRZ= Number (<10) of radial nodes of a Bessel or polar step function or number of elliptical nodes of an elliptical step function.
Used if LOAD=1 and CART=1 or if LOAD=3,4 or 5.
When LOAD=5 NOC values of NRZ are needed, if only the first is given the others are set equal to that one.
(Preset to 0)

17. RZER= Radii of the NRZ polar nodes of a polar step function or minor semiaxes of the NRZ elliptical nodes of an elliptical step function (in increasing order).
Used if LOAD=3,4 or 5.
When LOAD=5 RZER must be less than MIN(A,B).
When LOAD=5 NOC sets of RZER radii are needed (one for each step function), if only the first is given the others are set equal to that one. If different radial or elliptical nodes are desired for each step function, set the values for the n-th step function starting at the [10*(n-1)+1]-th element of the RZER vector.

18. NOSC= Number of oscillations of the highest components of a superposition of SIN functions.
Used if LOAD=6 or 7.

INPUT DATA DEFINING THE ITERATION.

From the viewpoint of the iteration OSCAR2D can be run in essentially three different ways controlled by the NOHM, NOLM and BCG values:

i With NOHM=1, NOLM=1 and BCG=0 to compute the lowest mode by the Point Succesive Over-Relaxation method (or the n-th mode, if you have already computed the first n-1 modes and you use ORTH).
However, the PSOR method is not recommended, in general (see HINT 4).

ii With NOHM=1, NOLM=1 and BCG=1 to compute by the Rayleigh Quotient Iteration the mode "nearest" to the initial approximation you give (see HINT 5).

iii With NOHM=n and NOLM=m (n.gt.1 .and. 1.le.m.le.n) to compute the next n-(m-1) modes, having already computed the first m-1 modes.
For each mode in turn a first approximation is obtained by the PSOR iteration with orthogonalization against the already computed modes ("first step") which isthen refined by the Rayleigh Quotient Iteration ("second step") (see HINT 1).

For other information about the above algorithms, their features and practical use see HINTS and reference [9].

1. NOCOMP= Computation control flag.
If NOCOMP is set to 1 no iteration is carried out and the output is produced taking the initial approximation as it were the final solution.
NOCOMP=1 makes sense only when LOAD=0.
When you want to rerun a problem which has been already solved by just changing ZLEN, ZORD or any input data defining the final output, you will get a great saving of CPU time by setting NOCOMP=1 and LOAD=0.
(Preset to 0)

2. NOHM= Number of the highest mode you want to compute (1.le.NOHM.le.99)
(Preset to 1)
(See the beginning of this section, too)

3. NOLM= Number of the lowest mode you want to compute
(1.le.NOLM.le.NOHM)
If NOLM.gt.1 you must have computed the NOLM-1 lower modes in a previous run and the corresponding binary files must be still present on the disk.
(Preset to 1)
(See the beginning of this section, too)

4. BCG= Method used to solve the eigenvalue problem.

   i 0 Point Successive Overrelaxation + Rayleigh Quotient

   ii 1 Rayleigh Quotient Iteration + Bi-Conjugate Gradient

   Do not use with NOHM.gt.1.
   (Preset to 0)
   (See the beginning of this section, too)

5. ORTH= Names of files (max 100) where the fields of previously computed lower modes have been stored (enclosed by apostrophes).
Periodically during the PSOR iteration (NOHM=1 and BCG=0) the approximate solution is made orthogonal to the indicated lower modes to obtain convergence to the next lowest one.
Do not use with BCG=1 or NOHM.gt.1.

6. SYM= Symmetry type.
Periodically during the PSOR iteration (NOHM=1 and BCG=0) the approximate fiel is made symmetrical or skew-symmetrical as required. Allowed only if the geometry of the problem is symmetrical.
This can be thought as an orthogonalization with respect to all the modes having the other symmetry and thus used to reduce the number of the previously computed modes that must be explicitly given in ORTH.
Do not use with BCG=1 or NOHM.gt.1.

   i 0 no symmetry

   ii 1 even symmetry

   iii -1 odd symmetry

   (Preset to 0)

7. RELAX= overrelaxation parameter used in the PSOR iteration (NOHM=1 and BCG=0 or "first steps" of a NOHM.gt.1 run). Greater than 0 and less than 2.
(Preset to 1.9 if NOHM=1, to 1.6 if NOHM.gt.1)
(Forced to min(RELAX,1.6) if TE=1 and CART.ne.1)

8. RELAX1= new overrelaxation parameter used in the PSOR iteration (NOHM=1 and BCG=0 or "first steps" of a NOHM.gt.1 run) after convergence criterion on eigenvalue is satisfied or if convergence is oscillatory.
(Preset to 1.)
(Forced to RELAX if NOHM.gt.1; forced to min(RELAX,1.6) if TE=1 and CART.ne.1)

9. DIRFAC= part of the underrelaxation factor not depending on distance used in the PSOR iteration (NOHM=1 and BCG=0 or "first steps" of a NOHM.gt.1 run) for DIRICHLET boundary points [7].
Greater than 0 and less than 1.
(Preset to 0.5)

10. CONVL= Eigenvalue convergence criterion.
To be compared with the relative difference in the eigenvalue between the previous iteration and the current one if BCG=0 or in the "first steps" of a NOHM.gt.1 run.
To be compared with the norm (greatest component) of the relative residual of the approximate eigensolution if BCG=1 (see RESIDL in section 7).
(Preset to 1.E-5 if NOHM=1; preset to 1.E-3 if NOHM.gt.1)

11. CONVL1= Eigenvalue convergence criterion for the "second steps".
To be compared with the norm of the relative residual of the approximate eigensolution(see RESIDL in section 7). Used only if NOHM.gt.1.
(Preset to 1.E-5)

12. CONVF= Eingenvector convergence criterion. To be compared with the greatest difference in normalized eigenvector components between the previous iteration and the current one if BCG=0 or in the "first steps" of a NOHM.gt.1 run.
To be compared with the relative residual (residual divided by the norm of the approximate solution) of the Bi-Conjugate

Gradient iteration if BCG=1.
(Preset to 1.E-3 if NOHM=1; preset to 1. if NOHM.gt.1)

13. CONVF1= Eigenvector convergence criterion for the "second steps".
To be compared with the relative residual (residual divided by the norm of the approximate solution) of the Bi-Conjugate
Gradient iteration. Used only if NOHM.gt.1.
(Preset to 1.E-3)
(Forced to CONVL1 if CONVF1.gt.CONVL1)

14. RUNTIM= time in seconds for the problem to iterate before it computes the output and gets off. If the problem converges before RUNTIM is up, RUNTIM is ignored.
If NOHM.gt.1 RUNTIM is used for each "first step".
(Preset to 600 if BCG=0; preset to 1800 if BCG=1)

15. RUNTIM1= the same as RUNTIM but for the "second step".
Used only if NOHM.gt.1.
(Preset to 1800)

16. FILNAM= Name you want to give to the files used by the program (enclosed by apostrophes).
The files involved are

name.OUT (always): Printed output (formatted ASCII)

name.BIN (if NOHM=1): It contains the frequency and the field (unformatted binary).
If LOAD=0 the program reads it to use both frequency and field obtained in a previous run as an initial approximation.
When the problem has been solved the program writes it to be used later on, either as an initial approximation (LOAD=0) or as a previously computed lower mode when orthogonalization is used
(ORTH='name').

<first six characters of name>M01.BIN
<first six characters of name>M02.BIN
. .
<first six characters of name>M<n>.BIN
(if NOHM=n)
Each of them contains the frequency and the field of a mode (unformatted binary).
Whenever a new mode has been computed the corresponding file is written to be used later on. When the computation of a new mode begins the previously written files are read again to carry out the orthogonalization in the "first step".
The first m-1 of these files must be present on the disk if you want to set NOLM=m.

name.CHN (if NOHM=1 and GRAPH=1,3,5 or 7): It contains the field inside the cavity and the data structures describing the boundary of the problem (till now formatted ASCII).
When the problem has been solved the program writes it to be used later on for graphic postprocessing or to feed the multipacting simulation program.

name.AXI (if NOHM=1 and GRAPH=2,3,6, or 7): It contains the field on the cavity axis (formatted ASCII).
When the problem has been solved the program writes it to be used later on for graphic postprocessing.

name.BOU (if NOHM=1 and GRAPH=4,5,6 or 7): It contains the field on the cavity boundary (formatted ASCII).

When the problem has been solved the program writes it to be used later on for graphic postprocessing.

If NOHM=n and still under the control of the GRAPH value, one file with .CHN, .AXI and .BOU extension and the same contents as above is created with the following names for each of the computed modes:
<first six characters of name>M01.extension
<first six characters of name>M02.extension
..
<first six characters of name>M<n>.extension

(Preset to OSCAR)

INPUT DATA DEFINING THE FINAL OUTPUT.

1. NACC= Normalization type.

   i 0 Normalize the fields in such a way that the stored energy (or the stored energy per unit length in the CART=1 and ZLEN=0 case) is fixed.

   ii 1 Normalize the fields in such a way that the effective accelerating field seen by a beam particle traveling at the relative velocity BETA along the cavity axis (i.e. VT/L where V=axial voltage, T=transit time factor and L=cavity length) is fixed.
   NACC=1 can be used only in TM axially symmetric accelerating cavities.

   (Preset to 0)

2. W= Normalization value.
   If NACC=0.and.(CART.ne.1.or.ZLEN.ne.0) the electromagnetic field is normalized in such a way that the stored energy in the cavity in mJ is equal to the normalization value W.
   If NACC=0.and.CART=1.and.ZLEN=0 the electromagnetic field is normalized in such a way that the stored energy per unit length in the waveguide in mJ/meter is equal to the normalization value W.
   If NACC=1 the electromagnetic field is normalized in such a way that the effective accelerating field in MV/meter seen by a beam particle traveling at the relative velocity BETA along the cavity axis (i.e. VT/L where V=axial voltage, T=transit time factor and L=cavity length) is given by the W value.
   (Preset to 1.)

3. SIG= Conductivity of all the metallic surfaces in MHOS/meter.
   The surface resistance is computed from conductivity (and frequency) by assuming a normal conducting material.
   (Preset to 5.91e+7 (copper conductivity))
   If the structure is made of two or more materials use SIGM (and, possibly, SIGF and SIGB) to assign the right conductivity to each part of the boundary.

4. SIGM= Conductivities to be assigned to each of the NOL line segments composing the boundary. Corresponding to a 9999 separator in ZB and RB, a zero value should be given to SIGM.
   If the structure is made of just one material use SIG to give its conductivity.
   If you use SIGM when CART=1 and ZLEN.ne.0, give SIGF and SIGB, too.

5. SIGF= Conductivity of the front plate of a finite length constant cross-section resonant structure (CART=1, ZLEN.ne.0).
   If you use SIGF, give SIGM and SIGB, too.

6. SIGB= Conductivity of the back plate of a finite length constant cross-section resonant structure (CART=1, ZLEN.ne.0).
   If you use SIGB, give SIGM and SIGF, too.

7. HALF= Set to 1 to correctly compute output quantities when only a half of a symmetric structure is given to the program. It works only with cylindrically symmetrical structures (CART=0).
   Otherwise leave unchanged.
   (Preset to 2)

8. BETA= Relative velocity (v/c) of the particle traveling through the structure. It is used to compute the Transit Time Factor (TTF) in accelerating structures (see also DRIL).
   (Preset to 1.)

9. DRIL= Drift length. It specifies the phase between the particle traveling through the structure and the accelerating field by giving the z coordinate of any point where the particle is when the accelerating field goes through zero because of its sinusoidal time dependence.
It is used to compute the Transit Time Factor (TTF) in accelerating structures as follows
TTF=ABS(INT(0,L,Ez*sin(((2*pi)/(BETA*lambda))*(z-DRIL))*dz))
/INT(0,L,ABS(Ez)*dz)
where L is the structure length and lambda is the RF wavelength.
For an accelerating mode often a right value for DRIL is the length of the drift tube, so the name "drift length".
(Preset to 0.)

10. AUTDRIL= Automatic computation of DRIL.
If set to 1 the following values are given to DRIL:
(L/2.)-(BETA*lambda)/4. for even modes,
(L/2.)-(BETA*lambda)/2. for odd modes
Do not use for unsymmetric structures.
When only a half of a symmetric structure is given to the program (HALF=1) the symmetry plane is supposed to be on the right and you must use ODD to specify the mode symmetry (even or odd).
(Preset to 0)

11. ODD= Mode symmetry.

    i 0 Even symmetry

    ii 1 Odd symmetry

    It is needed to specify the mode symmetry for the automatic computation of DRIL when only a half of a symmetric structure is given to the program.
    Used only if AUTDRIL=1 and HALF=1.

12. RSNT= between 1 and 20 values of the R coordinate along which the shunt impedance is to be calculated off the axis. It makes sense only for axially symmetric TM accelerating cavities (TE=0, CART=0).
(Preset to 0)

13. REH= between 1 and 20 values of the R (or Y) coordinate along which the electric and magnetic fields are to be calculated.
(Preset to 0)

14. PCn= Print control flags.

    i PC1;PC2;PC3 if .T. information on mesh generation is printed.
    (Preset to .F.)

    ii PC4 if .T. the eigenvector is printed for each iteration.
    (Preset to .F.)

    iii PC5 If .T. the eigenvalue is printed for each iteration.
    (Preset to .T.)

    iv PC6 if .T. E and H values on the boundary are printed.
    (Preset to .F.)

    v PC7 if .T. the eigenvector is printed.
    (Preset to .F.)

vi PC8 If .T. the cavity geometry and a contour plot of the eigenvector at the end of the iteration are printed.
(Preset to .T.)

vii PC9;PC10;PC11 If. T. information on finite difference equations on boundary is printed.
WARNING: PC11=.T. stops the execution.
(Preset to .F.)

viii PC13 if.T. a tracing of the execution is performed.
(Preset to .F.)

15. GRAPH= Graphic control flag.

i 0 Neither .CHN, nor. AXI, nor .BOU files (see FILNAM) are generated.

ii 1 .CHN files are generated for graphic postprocessing or multipacting simulation.

iii 2 .AXI files are generated for graphic postprocessing.

iv 4 .BOU files are generated for graphic postprocessing.

To produce more than just one type of the files above set GRAPH to the sum of the corresponding values (3 for CHN and .AXI, 7 for all and so on).
Some information about graphic postprocessors and a wide set of graphic output samples can be found in Appendix a
The NEWTRAJ postprocessor for multipacting simulation is sent on request with the OSCAR2D distribution kit but no documentation exists about it and it has to be adapted to local graphic libraries and peripherals.
(Preset to 0)

OUTPUT

Since most of the standard output is self explaining (see Appendix b) we will not describe the output in full details. We will clarify, instead, just a few less obvious points.

1. REQU. MESH SIZE and USED MESH SIZE
   They are respectively the mesh size given by the user as input data and the mesh size actually used by the program (see H in section 3).

2. <printer plot of the mesh>
   The mesh is printed as a matrix of characters. Each mesh point is represented by a character with the following meaning.

   i 1 = internal point

   ii 0 = external pont

   iii <any letter from A to T> = boundary point
   Which letter is used to represent a boundary point depends on the boundary slope at that point.

3. <information about the iterations>

   i LAMBDA
   Current approximation of the eigenvalue of the matrix eigenproblem.

   ii RESIDL
   Norm of the relative residual of the current approximation of the eigensolution of the matrix eigenproblem. The norm used is the greatest component of the vector and the residual is made relative by dividing it by the norm of the current approximation of the eigenvector.

   iii DL/L
   Relative difference between the current approximation of the eigenvalue of the matrix eigenproblem and the previous one.

   iv MEAN(DF/FMAX)
   Average difference in normalized eigenvector components between the previous iteration and the current one.

   v MAX(DF/FMAX)
   Greatest difference in normalized eigenvector components between the previous iteration and the current one.

   vi DEPARTURE FROM RIGHT PARITY: MEAN= ... MAX= ...
   Average and greatest value of ||F+|-|F-|| where F+ and F- are the eigenvector components corresponding to any couple of geometrically symmetrical mesh points.

   vii I and J
   Z and R (or X and Y) coordinates, in mesh size units, of the mesh point where MAX(DF/F) or MAX DEPARTURE FROM RIGHT PARITY occur.

   viii COMPONENTS OF PREVIOUSLY COMPUTED MODES
   Numerical approximation of (Hc,Hp)/(Hp,Hp) where Hc and Hp are eigenfunctions of the differential eigenproblem (the current mode and any of the previously computed ones, respectively).

ix FINAL LAMBDA
Final approximation of the eigenvalue of the differential eigenproblem that is to say k**2=(omega/c)**2 (in cm**-2).

4. NORMALIZATION FACTOR
Normalization factor for the electromagnetic field used to make the stored energy, the stored energy per unit length or the effective accelerating field (depending on the case) equal to the W input.

5. PERIODIC LENGTH
Equal to BETA*lambda/2 where lambda is the RF wavelength.

6. MAGNETIC GEOMETRIC FACTOR
Equal to omega*mu*INT(H**2*dV)/INT(H**2*dS).

7. ELECTRIC GEOMETRIC FACTOR
Equal to (omega*mu**2/epsilon)*INT(H**2*dV)/INT(E**2*dS).

8. <printer plot of the field>
The field and the cavity shape are printed as a matrix of characters.
External mesh points and boundary points are represented by a space and a dollar, respectively, while a character is assigned to each internal mesh point depending on the value of F at that point in the following way. The range of the F values is divided in intervals of equal width and the same character is assigned to mesh points having F values in the same interval. Digits from 1 to 9 indicate increasing positive values, letters from A to I indicate negative values increasing in magnitude, 0 indicates values (both positive and negative) belonging to any of the two intervals having zero as an endpoint. For a better readability of the printer plot, intervals of F values represented by contiguous letters or digits are separated by an interval of F values represented by a space.
The overall effect is like a rough contour plot of F and contour lines of F are field lines. The different meaning of F and their contour lines in the various cases is as follows.

   i Axially symmetric problem, TM modes (CART=0,TE=0):
   F=R*HPHI where HPHI is the azimuthal component of the magnetic field, contour lines of F are electric field lines.

   ii Axially symmetric problem, TE modes (CART=0,TE=1):
   F=R*EPHI where EPHI is the azimuthal component of the electric field, contour lines of F are magnetic field lines.

   iii Constant cross-section problem, TM modes (CART=1,TE=0):
   F=EZ where EZ is the Z component of the electric field, contour lines of F are magnetic field lines.

   iv Constant cross-section problem, TE modes (CART=1,TE=1):
   F=HZ where HZ is th Z component of the magneti field, contour lines of F are electric field lines.

9. <boundary field table>
If PC6=.T. a table is printed showing the fields at the boundary points. The symbols found in that table have the following meaning.

   i I and J
   Z and R (or X and Y) coordinates, in mesh size units, of the bondary point where the fields are computed.

ii S1
Order number of the boundary part to which the segment associated with the boundary point belongs (or zero if that boundary part is not a metallic wall).

iii S2
Order number of the boundary part to which a possible second segment associated with the boundary point belongs. Zero if that boundary part is not a metallic wall or no second segment exists. A second segment may exist only at endpoints of boundary parts.

iv LEN Length of the boundary segment associated with the boundary point (or sum of the lengths of the boundary segments when there are two segments associated with the boundary point).

v EPHI
Azimuthal component of electric field.

vi EZ
Electric field component along the Z axis.

vii EN
Electric field component normal to the boundary.

viii HPHI
Azimuthal component of magnetic field.

ix HZ
Magnetic field component along the Z axis.

x HT
Magnetic field component tangential to the boundary.

WARNING MESSAGES

(Self explaining messages are not reported)

LINE SEGMENT NO. n: STARTING/ENDING POINT MAY BE BAD
The starting (or ending) point of the n-th line segment is not consistent with center and radius (or axes) to the accuracy of 1.E-5*H.
The problem may equally run but if the fatal error message "TOO MANY BOUNDARY POINTS" was issued too you may have to correct the input data defining the geometry.

TIME ALLOWED FOR STEP 1 EXCEEDED
During the computation of a mode in a run with NOHM.gt.1 the time allowed for step 1 (RUNTIM) expired before the PSOR iteration converged. The program stops and the remaining modes are not computed but all the computed ones are correct.

TIME ALLOWED FOR STEP 2 EXCEEDED
During the computation of a mode in a run with NOHM.gt.1 the time allowed for step 2 (RUNTIM1) expired before the Rayleigh Quotient Iteration converged. The program stops and the remaining modes are not computed. All the computed modes but the last are correct.

FIELDS ON NEUMANN BOUNDARIES WILL BE WRONG
MAXIMUM SURFACE FIELDS MAY BE WRONG
You have used Neumann conditions in a constant cross-section TM problem (see TYPE). If PC6=.T. the fields on the boundary segments where Neumann conditions have been used will be wrong. The maximum surface fields will be wrong when they occurs on boundary segments where Neumann conditions have been used. All the other quantities will be correct, however.

FIELDS ON DIRICHLET BOUNDARIES WILL BE WRONG
MAXIMUM SURFACE FIELDS MAY BE WRONG
You have used Dirichlet conditions in a constant cross-section TE problem (see TYPE). If PC6=.T. the fields on the boundary segments where Dirichlet conditions have been used will be wrong. The maximum surface fields will be wrong when they occurs on boundary segments where Dirichlet conditions have been used. All the other quantities will be correct, however.

ABNORMAL RETURN FROM ROUTINE CORNOD:
THE CURVE F=0 IS PROBABLY CLOSED
A bad F field has been obtained. Perhaps the problem has been prematurely stopped by RUNTIM expiration or it does not converge.
CORNOD is used only if NACC=1.

FATAL ERROR MESSAGES

(Self explaining messages are not reported)

BAD INPUT DATA
You have left out some mandatory data or the format of the input is incorrect. Have you given values to NOL,RB,ZB and H? You cannot describe any problem without giving them, at least. Is the identification line present? Is the NAMELIST input starting in column two?

TOO MANY LINE SEGMENTS
The number of line segments defining the problem geometry must be no greater than 40.

TOO MANY CURVES
The number of closed curves defining the problem geometry must be no greater than 10

TOO MANY MESH POINTS
The program made a request to the operating system to get the memory it needs to run the problem but that request was not satisfied. Retry with a larger value of H or ask your system manager to allow you more memory, if possible.

TOO MANY BOUNDARY POINTS
The amount of memory allocate for boundary points has been exausted. An automatic calculation of NBND (number of boundary points) is provided but it may fail on particular cases. Roughly estimate in excess the required number of boundary points and give NBND as input unless the warning message "LINE SEGMENT NO. n: STARTING/ENDING POINT MAY BE BAD" was issued. In this case you may have rather to correct the input data defining the geometry.

TOO MANY MODES
NOHM must be no greater than 99

PC11:BOUND. PTS. GT 500
The PC11 option is not allowed if the number of boundary points exceeds 500.

WRONG SIZE OF MATRICES
Internal data generated by the program are not consistent. Please contact the authors. (Sometimes the problem may be overcome by changing H.)

ERROR (SUM)
Internal data generated by the program are not consistent. Please contact the authors.

MATBLD: NOT CONSISTENT
Internal data generated by the program are not consistent. Please contact the authors.

ERROR IN FUNCTION IDN
Internal data generated by the program are not consistent. Please contact the authors.

CONTROL VARIABLE OUT OF RANGE IN COMPUTED GOTO
Internal data generated by the program are not consistent. Please contact the authors.

SDVR CALL WITHOUT AXIS
Internal data generated by the program are not consistent. Please contact the authors.

FAILURE IN COMPUTING BESSEL FUNCTION
BESJ routine returned an error code different from 0 in computing initial approximation. See BESJ list. BESJ is used only if LOAD=1.

ONLY FIRST SIX ZEROS OF BESSEL FUNCTIONS AVAILABLE
The program knows only the first six zeros of Bessel functions, so NRZ cannot be set to a value greater than 5 when LOAD=1

NUMBER OF NODES .GT. 10 NOT ALLOWED
The greatest number of nodes of the initial approximation is 10 for each kind of them but the axial ones. So each of NYZ, NPHZ and NRZ cannot be set to a value greater than 10.

NUMBER OF AXIAL NODES .GT. 25 NOT ALLOWED
The greatest number of axial nodes of the initial approximation is 25. So NXZ cannot be set to a value greater than 25.

NUMBER OF CENTERS .GT. 10 NOT ALLOWED
A general polar/elliptical load (LOAD=5) can have 10 centers at most. So NOC cannot be set to a value greater than 10.

%SYSTEM-F-FLTOVFF (from VAX/VMS)
A floating overflow in the subroutine RITER means that the PSOR iteration diverges. Reduce RELAX (see HINT 6).

HINTS

1. COMPUTATION OF THE FIRST N MODES.
If you run the program with NOHM=n and with the default convergence values the modes are not guaranteed to be computed strictly in the order of increasing frequencies as some exchanges among near modes are possible. Thus you cannot be sure that the n computed modes are just the lowest n modes. A tighter convergence criterion for the "first steps" can force the program to compute the modes in the increasing order but may waste some CPU time as one or two modes, at most, are usually missed among the very last ones. So it is usually better to do a first run with NOHM=n-2 (or n-3) and the default convergence values, followed by a second one with NOLM=n-1 (or n-2), NOHM=n and a tighter convergence criterion for the "first steps". In our experience a value of 1.e-7 for CONVL is sufficiently tight in most cases.

2. COMPUTATION OF A SINGLE HIGH MODE.
If you are just interested in one high mode (or few) and you are not able to set directly a suitable initial approximation to pick it out, you can first compute it (together with all the lower modes and possibly a few higher ones) by a run with NOHM=n and a coarser than usual mesh (to save CPU time). Then you will be able to compute it with a full mesh by running the program with NOHM=1, BCG=1 and using the information from the previous run to set up the initial approximation (FREQ and LOAD). When choosing the mesh size for the first run you should not forget that a mesh still adequate for the lowest mode may be too coarse for a much higher one.
Notice that you cannot use LOAD=0 in the second run because the mesh size is different from the first one.

3. THE SAME HIGH MODE IN SLIGHTLY DIFFERENT GEOMETRIES.
If you have to run the same high mode by slightly varying the cavity shape you need not to carry out the process above (HINT 2) for each different geometry.
If the number of mesh points along Z and R are both unchanged you can set LOAD=0 to use the solution obtained for a shape as the initial approximation for the next one. If this is not the case you can at least use the knowledge of the solution for a shape to set up the initial approximation for the next one.

4. PSOR METHOD.
As a general rule the PSOR method (NOHM=1 and BCG=0) is not recommended but to get an initial approximation to start the Rayleigh Quotient Iteration since it is reliable just for the lowest mode of single cell cavities. Indeed, in all the other cases the PSOR method may give poor fields and, for instance, it actually happens for higher modes or multicell structures.
Thus, if you have any doubt about a field you have got by the PSOR method, refine it (or rerun the problem) by using the
Rayleigh Quotient Iteration (NOHM=1 and BCG=1).

5. INITIAL APPROXIMATION SETTING AND CONVERGENCE PROPERTIES.
When NOHM.gt.1 (PSOR followed by RQI) only computation order of modes and iteration time are affected by the initial approximation of the field (LOAD), whereas if NOHM=1 and BCG=0 (PSOR only) it affects not just iteration time, but field accuracy,too. The initial approximation of the frequency is almost unimportant in both cases. On the contrary when NOHM=1 and BCG=1 (RQI only) a reasonably good approximation of the frequency is needed, together with a rough knowledge of the node layout, to get just a particular mode.
In principle, if BCG=1 the program is expected to single out just the wanted mode without any trouble when: (i) the initial approximation of the field has the same number of nodes roughly at the same place as the wanted mode; (ii) the initial approximation of the frequency is closer to that of the wanted mode than to that of any other mode.

In practice, if the first condition is fulfilled, the initial frequency often can lie in a somewhat wider range than stated by the second one without causing any trouble. If the initial frequency is too far from that of the mode having the node layout of the initial field, it is unpredictable to which particular mode the iteration will converge. That unpredictable mode will be correctly computed, however.

6. RELAX VALUE SETTING.
Even though the default value of RELAX is adequate in most cases, on occasion you might have to reduce it as a too high value of the overrelaxation factor causes oscillations or even divergence in the PSOR iteration. Divergence shows itself as a floating overflow in the subroutine RITER.
If oscillations take place in the "first steps" of a NOHM.gt.1 run the order of computed modes is unpredictable (even though each of them is correct) and there may be several of both missed and repeatedly computed modes.
The user should know that a value of RELAX adequate for a particular cavity shape and mesh size may be too high for a different shape or a coarser mesh; moreover, even if the cavity shape and the mesh size are left unchanged it may become progressively too high as the mode order increases.

7. CONSTANT CROSS-SECTION PROBLEMS, TE MODES.
Notice that when CART=1 and TE=1 the first solution you get is a constant static (null frequency) magnetic field.
This obvious solution is of no interest. Besides, several output quantities are wrong since formulas used in computing them no longer hold when frequency vanishes. Definitely disregard this solution, it is just needed to compute the others.
Not to waste too much CPU time and to obtain a better approximation of the constant solution it is convenient to compute it separately by the PSOR method (NOHM=1 and BCG=0) and with the following input data LOAD=2,FREQ=1.E-38,RUNTIM=0,

8. MULTICELL STRUCTURES.
When computing multicell structures you should know that a mesh size which works satisfactorily for a n-cell structure may be too large for the same structure with just one more cell added. Indeed awfully bad fields may be obtained for the longer structure by using the same mesh size giving quite good results for the shorter one.
If this happens reduce the mesh size or give to the program just a half of the structure (if symmetric, of course).

9. DEGENERATE AND CLUSTERED MODES.
When a multiple eigenvalue (degenerate mode) occurs, a NOHM.gt.1 run converges to any basis of the corresponding invariant subspace, not just the one you expect. For sake of preciseness the particular basis you will find may depend on the initial approximation you have given. This behavior of the program is quite correct, indeed. Anyway, if you want just a particular eigenvector belonging to the invariant subspace, you can usually pick it out by a BCG=1 run starting from a suitable initial approximation and the frequency found in the previous run.
Just the same happens when eigenvalues take place in very tight clusters since there is a continuous transition in the program behavior from the case of a multiple eigenvalue to that of well separated simple eigenvalues through the case of clustered ones. In this latter case, however, even though also in experimental work it may be difficult to excite a clear mode, in principle there is no single invariant subspace corresponding to the cluster but a single eigenvector corresponding to each eigenvalue. So the solutions you may find are not all equally good. Skilfully select one of them where possible.
Structures giving rise to eigenvalue clusters often consists of several cavities connected by small holes. In those cases be careful that coupling holes are discretized by a sufficient number of mesh points. Otherwise very bad results may be obtained. When multiples or tightly clustered eigenvalues occur, the order in which the modes are found may be quite different from that of increasing frequency unless CONVL is suitably reduced. In these

cases instead of following HINT 1 you may have rather to set CONVL=1.e-7 from the beginning and then to further reduce it in the second run.

EXAMPLES

The following examples illustrate the major features of the OSCAR2D program. They are supposed to be run in the given order.
For each of them the input data file and the corresponding output file are provided in the distribution kit.

1. EXAMPLE01-EXAMPLE02-EXAMPLE03

```
***** LOWEST TM MODE BY PSOR *****
$NINPUT
NOL=11,FREQ=4700.,H=0.075,
RB=0.,0.65,0.65,1.35,1.8,2.8,1.8,1.35,0.65,0.65,0.,
ZB=0.,0.,0.93,1.33,1.33,2.33,3.33,3.33,3.73,4.66,4.66,
RC=0.,0.,1.35,0.,1.8,1.8,0.,1.35,0.,0.,0.,
ZC=0.,0.,0.93,0.,2.33,2.33,0.,3.73,0.,0.,0.,
R=0.,0.,0.4,0.,-1.,-1.,0.,0.4,0.,0.,0.,
R2=0.,0.,0.7,0.,-1.,-1.,0.,0.7,0.,0.,0.,
TYPE=1,8*0,2*1,
METAL=0,8*1,2*0,
BETA=0.855,
DRIL=0.93,
NACC=1,
PC6=.T.,
FILNAM='TM010',
$END

***** NEXT TM MODE BY PSOR *****
$NINPUT
NOL=11,FREQ=9500.,H=0.075,
RB=0.,0.65,0.65,1.35,1.8,2.8,1.8,1.35,0.65,0.65,0.,
ZB=0.,0.,0.93,1.33,1.33,2.33,3.33,3.33,3.73,4.66,4.66,
RC=0.,0.,1.35,0.,1.8,1.8,0.,1.35,0.,0.,0.,
ZC=0.,0.,0.93,0.,2.33,2.33,0.,3.73,0.,0.,0.,
R=0.,0.,0.4,0.,-1.,-1.,0.,0.4,0.,0.,0.,
R2=0.,0.,0.7,0.,-1.,-1.,0.,0.7,0.,0.,0.,
TYPE=1,8*0,2*1,
METAL=0,8*1,2*0,
BETA=0.855,
AUTDRIL=1,
NXZ=1,XZER=2.33,
SYM=-1,
FILNAM='TM011',
$END

***** 3-RD TM MODE BY PSOR *****
$NINPUT
NOL=11,FREQ=9500.,H=0.075,
RB=0.,0.65,0.65,1.35,1.8,2.8,1.8,1.35,0.65,0.65,0.,
ZB=0.,0.,0.93,1.33,1.33,2.33,3.33,3.33,3.73,4.66,4.66,
RC=0.,0.,1.35,0.,1.8,1.8,0.,1.35,0.,0.,0.,
ZC=0.,0.,0.93,0.,2.33,2.33,0.,3.73,0.,0.,0.,
R=0.,0.,0.4,0.,-1.,-1.,0.,0.4,0.,0.,0.,
R2=0.,0.,0.7,0.,-1.,-1.,0.,0.7,0.,0.,0.,
TYPE=1,8*0,2*1,
METAL=0,8*1,2*0,
BETA=0.855,
```

```
AUTDRIL=1,
NYZ=1,
SYM=1,
ORTH='TM010',
FILNAM='TM020',
$END
```

In these examples the three lowest TM modes of a single cell axisymmetric cavity are computed by the PSOR method.
This method is not reliable in general (see HINT 4); it might be sometimes useful to obtain an initial approximation for the Rayleigh Quotient Iteration.
We stress that to proceed as in these examples you need to know the mode order in advance. Otherwise you cannot correctly set symmetry and node layout and, as a consequence, you need more CPU time to get possibly worse fields. On the contrary good initial approximations of the mode frequencies are unnecessary.
Notice that in EXAMPLE01 a different normalization (NACC=1) than in the other two examples is used, DRIL is explicitly given and the program is requested to output the fields on the boundary, too. In EXAMPLE02 and EXAMPLE03 SYM is used to reduce the number of lower modes explicitly given in ORTH.

2. EXAMPLE04-EXAMPLE05

```
***** 10 LOWEST TM MODES (1-ST RUN OF 2) *****
$NINPUT
NOL=11,FREQ=4700.,H=0.075,
RB=0.,0.65,0.65,1.35,1.8,2.8,1.8,1.35,0.65,0.65,0.,
ZB=0.,0.,0.93,1.33,1.33,2.33,3.33,3.33,3.73,4.66,4.66,
RC=0.,0.,1.35,0.,1.8,1.8,0.,1.35,0.,0.,0.,
ZC=0.,0.,0.93,0.,2.33,2.33,0.,3.73,0.,0.,0.,
R=0.,0.,0.4,0.,-1.,-1.,0.,0.4,0.,0.,0.,
R2=0.,0.,0.7,0.,-1.,-1.,0.,0.7,0.,0.,0.,
TYPE=1,8*0,2*1,
METAL=0,8*1,2*0,
NACC=1,
BETA=0.855,
AUTDRIL=1,
NOHM=7,
FILNAM='R7P3',
$END

***** 10 LOWEST TM MODES (2-ND RUN OF 2) *****
$NINPUT
NOL=11,FREQ=4700.,H=0.075,
RB=0.,0.65,0.65,1.35,1.8,2.8,1.8,1.35,0.65,0.65,0.,
ZB=0.,0.,0.93,1.33,1.33,2.33,3.33,3.33,3.73,4.66,4.66,
RC=0.,0.,1.35,0.,1.8,1.8,0.,1.35,0.,0.,0.,
ZC=0.,0.,0.93,0.,2.33,2.33,0.,3.73,0.,0.,0.,
R=0.,0.,0.4,0.,-1.,-1.,0.,0.4,0.,0.,0.,
R2=0.,0.,0.7,0.,-1.,-1.,0.,0.7,0.,0.,0.,
TYPE=1,8*0,2*1,
METAL=0,8*1,2*0,
NACC=1,
BETA=0.855,
AUTDRIL=1,
NOLM=8,NOHM=10,
CONVL=1.E-7,
FILNAM='R7P3',
$END
```

In each of these examples the program computes more than one mode in a single run (NOHM.gt.1). For each mode the program automatically makes use of PSOR to get an initial approximation for RQI. In our experience this method may fail only in the situations described in HINT 6, HINT 8 and HINT 9 where a cure is suggested, too.
This is the recommended way to use OSCAR2D for general purposes. Altogether these two examples computes the ten lowest TM modes of the same cavity of previous examples. This computation is split in two runs following HINT 1.
Notice that, since each mode needs a different DRIL value, the automatic computation of DRIL is used.
Some graphic outputs of EXAMPLE05-EXAMPLE06 are shown in Appendix a.

3. EXAMPLE06

```
***** SINGLE HIGHER TM MODE BY RQI+BCG *****
$NINPUT
NOL=11,FREQ=18000.,H=0.075,
RB=0.,0.65,0.65,1.35,1.8,2.8,1.8,1.35,0.65,0.65,0.,
ZB=0.,0.,0.93,1.33,1.33,2.33,3.33,3.33,3.73,4.66,4.66,
RC=0.,0.,1.35,0.,1.8,1.8,0.,1.35,0.,0.,0.,
ZC=0.,0.,0.93,0.,2.33,2.33,0.,3.73,0.,0.,0.,
R=0.,0.,0.4,0.,-1.,-1.,0.,0.4,0.,0.,0.,
R2=0.,0.,0.7,0.,-1.,-1.,0.,0.7,0.,0.,0.,
TYPE=1,8*0,2*1,
METAL=0,8*1,2*0,
BETA=0.855,
AUTDRIL=1,
NACC=1,
BCG=1,
NXZ=1,XZER=2.33,NYZ=2,
FILNAM='TM031',
$END
```

The last mode obtained in EXAMPLE04 is computed alone. Reasonably good approximations of frequency and node layout are given to single out just that mode. Following HINT 5 an initial frequency closer to the true one than to the frequency of any other mode. Anyway convergence to the same mode has been observed also for initial frequencies less than that of the immediately lower mode (6-th mode of EXAMPLE04) or greater than that of the immediately higher one (1-st mode of EXAMPLE05).
If EXAMPLE06 is run again with the frequency obtained in EXAMPLE04 (last mode) and a finer mesh, EXAMPLE04 and EXAMPLE06 altogether illustrate HINT 2, too.
Both printed output and graphic output of EXAMPLE06 are reported in Appendix b.

4. EXAMPLE07

```
***** PREVIOUS MODE IN A SLIGHTLY MODIFIED GEOMETRY *****
$NINPUT
NOL=11,FREQ=18000.,H=0.075,
RB=0.,0.65,0.65,1.65,1.8,2.8,1.8,1.65,0.65,0.65,0.,
ZB=0.,0.,0.33,1.33,1.33,2.33,3.33,3.33,4.33,4.66,4.66,
RC=0.,0.,1.65,0.,1.8,1.8,0.,1.65,0.,0.,0.,
ZC=0.,0.,0.33,0.,2.33,2.33,0.,4.33,0.,0.,0.,
R=0.,0.,1.,0.,-1.,-1.,0.,1.,0.,0.,0.,
R2=0.,0.,1.,0.,-1.,-1.,0.,1.,0.,0.,0.,
TYPE=1,8*0,2*1,
METAL=0,8*1,2*0,
BETA=0.855,
AUTDRIL=1,
```

```
NACC=1,
BCG=1,
LOAD=0,
FILNAM='TM031',
$END
```

The same mode obtained in the previous example is computed in a cavity with a slightly changed geometry.
Following HINT 3 the solution obtained by running EXAMPLE06 is used as initial approximation for EXAMPLE07. Notice that to do so the filename must be the same in both examples.

5. EXAMPLE08-EXAMPLE09-EXAMPLE10

```
***** 4-CELL STRUCTURE (4 LOWEST TM MODES) *****
$NINPUT
NOL=29,FREQ=4474.,H=0.1,
RB=0.,2*0.93,1.58,1.65,2.92,1.65,1.58,0.93,1.58,1.65,2.92,
     1.65,1.58,0.93,1.58,1.65,2.92,1.65,1.58,0.93,1.58,1.65,2.92,
     1.65,1.58,2*0.93,0.,
ZB=2*0.,3.5,2*3.9,5.17,2*6.44,6.84,2*7.24,8.51,2*9.78,10.18,
     2*10.58,11.85,2*13.12,13.52,2*13.92,15.19,2*16.46,16.86,
     2*20.36,
RC=2*0.,1.58,0.,2*1.65,0.,2*1.58,0.,2*1.65,0.,2*1.58,0.,
     2*1.65,0.,2*1.58,0.,2*1.65,0.,1.58,3*0.,
ZC=2*0.,3.5,0.,2*5.17,0.,2*6.84,0.,2*8.51,0.,2*10.18,0.,2*11.85,
     0.,2*13.52,0.,2*15.19,0.,16.86,3*0.,
R=2*0.,0.4,0.,2*-1.27,0.,2*0.4,0.,2*-1.27,0.,2*0.4,0.,2*-1.27,
   0.,2*0.4,0.,2*-1.27,0.,0.4,3*0.,
R2=2*0.,0.65,0.,2*-1.27,0.,2*0.65,0.,2*-1.27,0.,2*0.65,0.,
     2*-1.27,0.,2*0.65,0.,2*-1.27,0.,0.65,3*0.,
TYPE=0.,26*0.,0.,1.,
METAL=0.,26*1.,2*0.,
AUTDRIL=1,
BETA=1.,
NOHM=4,
FILNAM='MCS4',
$END

***** 5-CELL STRUCTURE (PI MODE) *****
$NINPUT
NOL=35,FREQ=4534.,H=0.07,
RB=0.,0.93,0.93,1.58,1.65,2.92,1.65,1.58,0.93,1.58,1.65,2.92,
     1.65,1.58,0.93,1.58,1.65,2.92,1.65,1.58,0.93,1.58,1.65,2.92,
     1.65,1.58,0.93,1.58,1.65,2.92,1.65,1.58,0.93,0.93,0.,
ZB=2*0.,3.5,2*3.9,5.17,2*6.44,6.84,2*7.24,8.51,2*9.78,10.18,
     2*10.58,11.85,2*13.12,13.52,2*13.92,15.19,2*16.46,16.86,
     2*17.26,18.53,2*19.80,20.20,2*23.7,
RC=2*0.,1.58,0.,2*1.65,0.,2*1.58,0.,2*1.65,0.,2*1.58,0.,
     2*1.65,0.,2*1.58,0.,2*1.65,0.,2*1.58,0.,2*1.65,0.,1.58,3*0.,
ZC=2*0.,3.5,0.,2*5.17,0.,2*6.84,0.,2*8.51,0.,2*10.18,0.,2*11.85,
     0.,2*13.52,0.,2*15.19,0.,2*16.86,0.,2*18.53,0.,20.2,3*0.,
R=2*0.,0.4,0.,2*-1.27,0.,2*0.4,0.,2*-1.27,0.,2*0.4,0.,2*-1.27,
   0.,2*0.4,0.,2*-1.27,0.,2*0.4,0.,2*-1.27,0.,0.4,3*0.,
R2=2*0.,0.65,0.,2*-1.27,0.,2*0.65,0.,2*-1.27,0.,2*0.65,0.,
     2*-1.27,0.,2*0.65,0.,2*-1.27,0.,2*0.65,0.,2*-1.27,0.,
     0.65,3*0.,
TYPE=1.,32*0.,2*1.,
```

```
METAL=0.,32*1.,2*0.,
AUTDRIL=1,
BETA=1.,
BCG=1,
NXZ=4,
XZER=6.84,10.18,13.52,16.86,
FILNAM='MCS5',
RUNTIM=7200,
$END

***** 5-CELL STRUCTURE (HALF GEOMETRY) *****
$NINPUT
NOL=19,FREQ=4500.,H=0.1,
RB=0.,0.93,0.93,1.58,1.65,2.92,1.65,1.58,0.93,1.58,1.65,2.92,
     1.65,1.58,0.93,1.58,1.65,2.92,0.,
ZB=2*0.,3.5,2*3.9,5.17,2*6.44,6.84,2*7.24,8.51,2*9.78,10.18,
     2*10.58,2*11.85,
RC=2*0.,1.58,0.,2*1.65,0.,2*1.58,0.,2*1.65,0.,2*1.58,0.,
     1.65,2*0.,
ZC=2*0.,3.5,0.,2*5.17,0.,2*6.84,0.,2*8.51,0.,2*10.18,0.,11.85,
     2*0.,
R=2*0.,0.4,0.,2*-1.27,0.,2*0.4,0.,2*-1.27,0.,2*0.4,0.,-1.27,
   2*0.,
R2=2*0.,0.65,0.,2*-1.27,0.,2*0.65,0.,2*-1.27,0.,2*0.65,0.,
     -1.27,2*0.,
TYPE=1,16*0,0,1,
LREFL=1,RREFL=18,
METAL=0,16*1,2*0,
HALF=1,
BETA=1.,AUTDRIL=1,ODD=0,
NOHM=3,
FILNAM='MCS5HN',
$END
```

In these examples some TM modes of a 4-cell and a 5-cell structures with the same cell shape are computed.
Altogether they illustrate HINT 8 since if you run EXAMPLE09 with the same mesh size as EXAMPLE08 very bad fields are obtained.
If you want to compute many modes the best way is probably to run EXAMPLE10 with a finer mesh. EXAMPLE10 computes just modes with even symmetry; to compute the others the Neumann condition on the symmetry plane must be changed to a Dirichlet one.
ome graphic outputs of EXAMPLE08 are shown in Appendix a.

6. EXAMPLE06

7. EXAMPLE11

```
*** DAW STRUCTURE (MULTIPLY CONNECTED) ***
 $NINPUT
 NOL=26
 FREQ=1000.,H=0.5,
 RB=0.,2.,2.,16.,16.,20.,20.,16.,16.,20.,20.,16.,16.,2.,2.,0.,
    9999.,2.,13.85,13.85,2.,9999.,2.,13.85,13.85,2.,

ZB=0.,0.,3.5,3.5,6.4,6.4,15.4,15.4,19.2,19.2,28.2,28.2,31.1,31.1,
 34.6,34.6,9999.,10.9,10.9,9.9,9.9,9999.,24.7,24.7,23.7,23.7,
 TYPE=0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,9999,0,0,0,0,9999,0,0,0,0,
 METAL=0,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,0,1,1,1,1,
```

```
NBND=500,
NXZ=2,XZER=10.4,24.2,
BCG=1,
BETA=0.95,AUTDRIL=1,
NACC=1,
FILNAM='DAW',
$END
```

This is an example of a structure (disk-and-washer) with a multiply connected section (notice the use of the 9999 separator).
Notice also that NBND is explicitly given because in this complicate geometry its automatic computation fails owing to a high line to surface ratio.

8. EXAMPLE12

```
***** TE MODE *****
$NINPUT
TE=1,
NOL=11,FREQ=10000.,H=0.075,
RB=0.,0.65,0.65,1.35,1.8,2.8,1.8,1.35,0.65,0.65,0.,
ZB=0.,0.,0.93,1.33,1.33,2.33,3.33,3.33,3.73,4.66,4.66,
RC=0.,0.,1.35,0.,1.8,1.8,0.,1.35,0.,0.,0.,
ZC=0.,0.,0.93,0.,2.33,2.33,0.,3.73,0.,0.,0.,
R=0.,0.,0.4,0.,-1.,-1.,0.,0.4,0.,0.,0.,
R2=0.,0.,0.7,0.,-1.,-1.,0.,0.7,0.,0.,0.,
TYPE=11*1
METAL=0,8*1,2*0,
FILNAM='TE010',
$END
```

This is an example of a TE mode computation in an axisymmetric cavity. For sake of preciseness the cavity is the same of the first six examples where TM modes are computed. Notice that the TYPE values, too, have been changed, not just the TE one.

9. EXAMPLE13

```
***** ELLIPTICAL WAVEGUIDE TM MODE *****
$NINPUT
NOL=1,FREQ=1800.,H=0.6,
CART=1,
RB=10.,
ZB=1.,
RC=10.,
ZC=16.,
R=-15.,
R2=-10.,
METAL=1,
TYPE=1,
LOAD=4,
A=15.,B=10.,
NRZ=0,
NPHZ=2,
PHZER=1.05,2.09,
FILNAM='EWTM',
BCG=1,
$END
```

This is an example of a TM mode computation in a waveguide.

10. EXAMPLE14-EXAMPLE15

```
***** RECTANGULAR WAVEGUIDE TE MODE (1-ST RUN OF TWO) *****
$NINPUT NOL=4,H=0.25,
CART=1,
TE=1,
RB=0.,2*10.,0.,
ZB=2*0.,2*11.,
TYPE=4*0,
METAL=4*1,
LOAD=2,FREQ=1.E-38,RUNTIM=0,
FILNAM='RWTEM01',
$END

***** RECTANGULAR WAVEGUIDE TE MODE (2-ND RUN OF TWO) *****
$NINPUT NOL=4,FREQ=2000.,H=0.25,
CART=1,
TE=1,
RB=0.,2*10.,0.,
ZB=2*0.,2*11.,
TYPE=4*0,
METAL=4*1,
LOAD=6,NOSC=3,
NOLM=2,NOHM=5,
FILNAM='RWTE',
$END
```

In these examples the five lowest TE modes of a waveguide are computed.
The null frequency constant field solution is computed in a separate run following HINT 7. Notice the relation between the two filenames.
In EXAMPLE15 a superpositin of sinusoidal functions (LOAD=6) is used as initial approximation. In so doing the modes are computed in a better order (i.e. less different from the strictly increasing one) than by using LOAD=1 or 2, but at the cost of a somewhat longer execution time.

11. EXAMPLE16

```
***** ELLIPTICAL CROSS SECTION CAVITY, TM MODE *****
$NINPUT
NOL=1,FREQ=1150.,H=0.6,
CART=1,
RB=10.,
ZB=1.,
RC=10.,
ZC=16.,
R=-15.,
R2=-10.,
METAL=1,
TYPE=1,
ZLEN=20.,
SIG=3.54E+7,
FILNAM='ECSCTM',
NOHM=3,ZORD=0,1,2,
$END
```

This is an example of a TM mode computation in a constant cross-section cavity.
Notice that this is not a copper cavity.

12. EXAMPLE17

```
***** RECTANGULAR CROSS SECTION CAVITY, TE MODE *****
$NINPUT NOL=4,H=0.25,
CART=1,
TE=1,
RB=0.,2*10.,0.,
ZB=2*0.,2*11.,
TYPE=4*0.,
METAL=4*1.,
ZLEN=15.,ZORD=2,
SIGM=4*5.91E+7,SIGF=3.54E+7,SIGB=3.54E+7,
LOAD=2,NYZ=1,YZER=5.,
FREQ=1500.,
BCG=1,
FILNAM='RCSCTE',
$END
```

This is an example of a TE mode computation in a constant cross-section cavity. Notice that this cavity is made of two different materials.

INSTALLATION NOTES

1. DISTRIBUTION KIT
The distribution tape for a VAX/VMS environment is usually a VAX/VMS standard tape recorded at 1600 bpi with label OSCAR and it contains the following files:

   1. OSCAR2D.USG
   User's Guide.

   2. OSCAR2D20.FOR
   OSCAR2D program, version 2.0 (FORTRAN source).

   3. TIMEX.MAR
   TIMEX subroutine (MACRO source).

   4. COMPILE.COM
   Command file to compile and link OSCAR2D.

   5. EXECUTE.COM
   Command file to execute OSCAR2D.

   6. EXAMPLE01.INP-EXAMPLE17.INP
   Input data files of the examples reported in the User's Guide.

   7. EXAMPLE01.OUT-EXAMPLE17.OUT
   Output files produced by running the examples reported in the User's Guide.

   Both inputs and outputs of the examples reported in the User's Guide have been included in the distribution kit essentially for two aims. First, you may start to learn to use the program by studying them. Second, if you have to convert OSCAR2D for an environment different from the VAX/VMS one, the examples may be useful also to test the program after conversion. In this case remember that the examples with the same geometry must be run in the given order.

   For an environment different from the VAX/VMS one the distribution tape contains the same files in the same order but is usually written in the following format: 1600 bpi, ASCII, NOLABEL, 133 characters per record, 7980 characters per block.

   Whenever possible the distribution kit is sent by network, each file as a separate message. Since the messages may be received in a random order, the file names are used as message subjects.

2. INSTALLATION UNDER VAX/VMS
First install the program by the following VAX/VMS DCL commands

   $MOUNT MSA0: OSCAR
   $COPY MSA0:*.* *.*
   $SUBMIT/NOTIFY COMPILE

   and then you can execute it by

   $SUBMIT/NOTIFY EXECUTE
   or
   $SUBMIT/NOTIFY/PAR=infile EXECUTE

In the first case the input data will be read from the file OSCAR.INP, in the second case the input data will be read from the file infile.INP.
The output files will be outfile.OUT and outfile.BIN if FILNAM='outfile' is present in the input data or OSCAR.OUT and OSCAR.BIN otherwise.

3. CONVERSION FOR AN ENVIRONMENT DIFFERENT FROM THE VAX/VMS ONE
In the conversion you will have to take care of the following things, at least.

1. OSCAR2D makes use of dynamic allocation of memory. This is not a standard feature of the FORTRAN language, so its implementation is machine dependent. Anyway, if you want, you can statically allocate a large amount of memory without any change in the program structure.

2. All INTEGER variables are supposed to be 4 byte long except an explicitly declared INTEGER*2 array; all REAL variables are declared as REAL*8. Different lengths may upset the data structures.

3. On some machine, alignment problems might arise with some COMMONs. Should it unfortunately happen, you will have to reorder those COMMONs (a tedious task).

4. The PRPLT routine generates ASCII codes to "plot" cavity shape and field lines on the line printer. You may have to modify it for different codes (e.g. EBCDIIC).

5. OSCAR2D makes use of an assembler subroutine to get the CPU time.

Should you find any other points to take care of in the conversion, inform the authors, please, so that they can update these notes.

Good luck. And should you have any problems, contact the authors, please. Any comments about OSCAR2D and its documentation will be greatly appreciated, too.

REFERENCES

[1] H.C.Hoyt, IEEE Trans. Nuc. Sci., vol.NS-12, p.153 (1965).

[2] H.C.Hoyt et al., Rev. Sci. Instr., vol.37, p.755 (1966).

[3] W.F.Rich et al., Report LA-4219, LASL(Los Alamos 1969).

[4] P.Fernandes et al.,Particle Accelerators, vol.12, p.131 (1982)

[5] P.Fernandes et al.,IEEE Trans. Mag., vol.MAG-19, p.2421 (1983)

[6] M.Cena et al.,IMA-CNR Internal Report 151/84 (Genoa 1984)

[7] P.Fernandes et al., IEEE Trans. Mag., vol.MAG-21, p.2246 (1985)

[8] P.Fernandes et al., Proceedings of the Linear Accelerator Conference 1986, Stanford, p.330

[9] P.Fernandes et al., IEEE Trans. Mag., vol.24, p.154 (1988)

AUTHOR'S ADDRESSES

PAOLO FERNANDES
ISTITUTO per la MATEMATICA APPLICATA del C.N.R.
Via Leon Battista Alberti 4, 16132 GENOVA (ITALY)
Phone: ++39 10 515510
Fax: ++39 10 517801
E-mail: FERNANDES@IMAGE.GE.CNR.IT
HEPNET/DECNET address: IMAGE::FERNANDES (or 38.467::FERNANDES)

RENZO PARODI
I.N.F.N. Sezione di Genova
Via Dodecaneso 33, 16146 GENOVA (ITALY)
Phone: ++39 10 5993326
Fax: ++39 10 308534
Telex: 281154 INFNGE
E-mail: PARODI@VAXGE.INFN.IT
HEPNET/DECNET address: VAXGE::PARODI (or 38.254::PARODI)

APPENDIX A:

GRAPHIC POSTPROCESSORS AND GRAPHIC OUTPUT SAMPLES.

The graphic postprocessors FLF, FLT, FLDAXI and FLDBOU are distributed with OSCAR2D but no documentation exists about them and they have to be adapted to local graphic libraries and peripherals.

1. FLF (Field Line Finder) finds the field lines from the information contained in a .CHN file and FLT (Field Line Tracer) plots them.
   The field lines are found as contour lines of F. The different meaning of F and their contour lines in the various cases is as follows.

   i Axially symmetric problem, TM modes (CART=0,TE=0):
   F=R*HPHI where HPHI is the azimuthal component of the magnetic field, contour lines of F are electric field lines.

   ii Axially symmetric problem, TE modes (CART=0,TE=1):
   F=R*EPHI where EPHI is the azimuthal component of the electric field, contour lines of F are magnetic field lines.

   iii Constant cross-section problem, TM modes (CART=1,TE=0):
   F=EZ where EZ is the Z component of the electric field, contour lines of F are magnetic field lines.

   iv Constant cross-section problem, TE modes (CART=1,TE=1):
   F=HZ where HZ is the Z component of the magnetic field, contour lines of F are electric field lines.

   A field line is plotted as a solid line when F is positive along it; as a dashed one otherwise. Notice that the sign of F cannot change along a field line since field lines are contour lines and
   F is constant along them.
   The direction of field lines is not shown in the plot but can be easily found by the following rule: the direction of a field line is such that F is growing from right to left.

2. FLDAXI plots the graphs of the electric field and electric field squared along the axis from the information contained in a .AXI file.
   It can handle just axial symmetric TM modes.

3. FLDBOU plots the graphs of the electric field, magnetic field and magnetic field squared on the boundary from the information contained in a .BOU file. The graph of magnetic field squared is labelled "power density" as these two quantities are proportional.
   FLDBOU can handle just axial symmetric TM modes.

A wide set of graphic output samples is shown in the following pages.
It consists of:

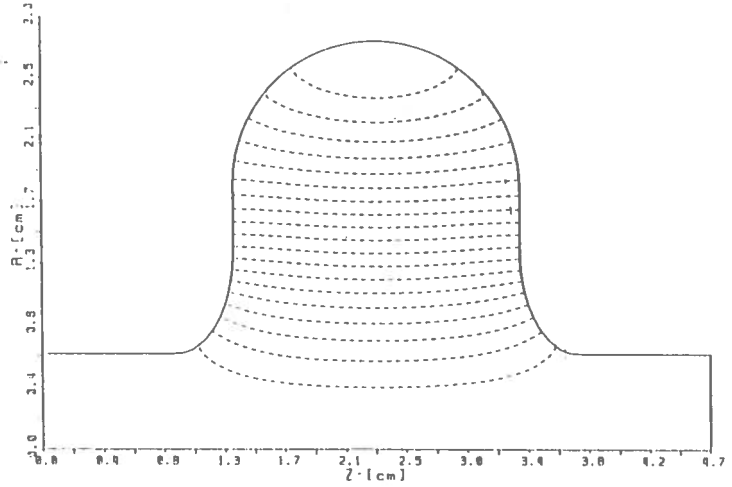
1. Field line plots of the ten modes of EXAMPLE04-EXAMPLE05
   For the two highest modes some graphs of the fields along the axis and on the boundary are shown, too.

2. Field line plots of the four modes of EXAMPLE08
   For the highest mode all graphs of the fields along the axis and on the boundary are shown, too.

3. Field line plots in more complicate geometries.

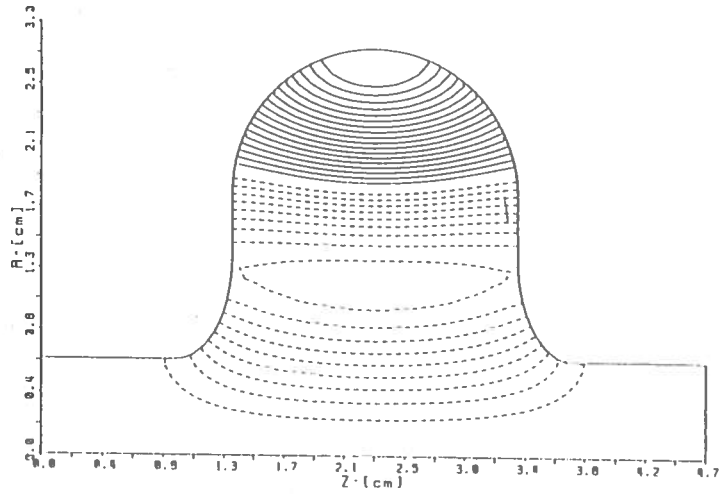
***** 10 LOWEST TM MODES *****

file EXA05M01 frequency = 4579.70 Mhz computed by the OSCAR2D code written by P.FERNANDES and A.PARODI INFN-GENOA



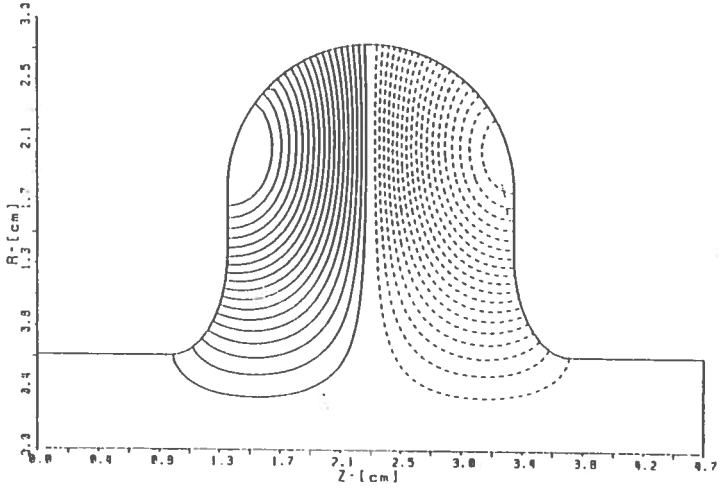
***** 10 LOWEST TM MODES *****

file EXA05M02 frequency = 10145.09 Mhz computed by the OSCAR2D code written by P.FERNANDES and A.PARODI INFN GENOA



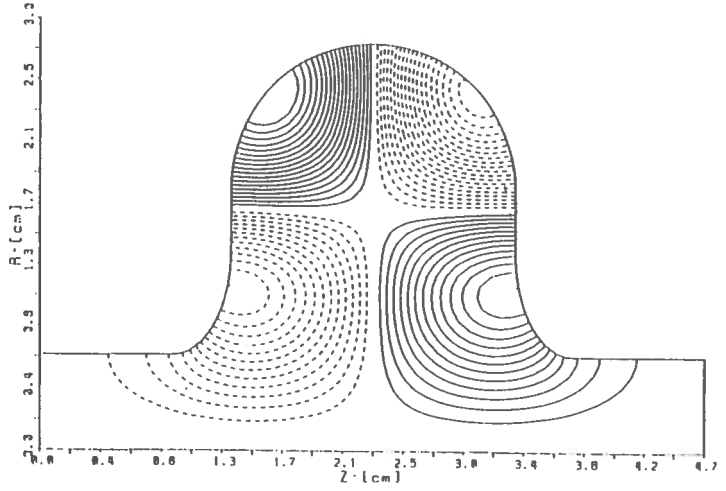
***** 10 LOWEST TM MODES *****

file EXA05M03 frequency = 9572.16 Mhz computed by the OSCAR2D code written by P.FERNANDES and A.PARODI INFN-GENOA



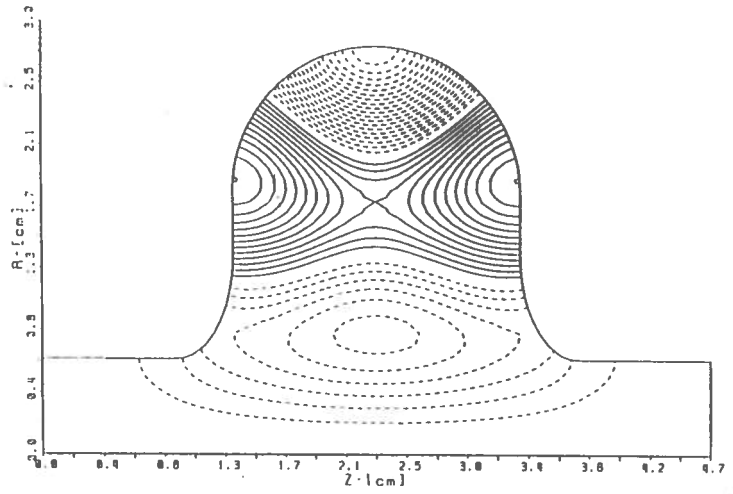
***** 10 LOWEST TM MODES *****

file EXA05M04 frequency = 13655.40 Mhz computed by the OSCAR2D code written by P.FERNANDES and A.PARODI INFN GENOA



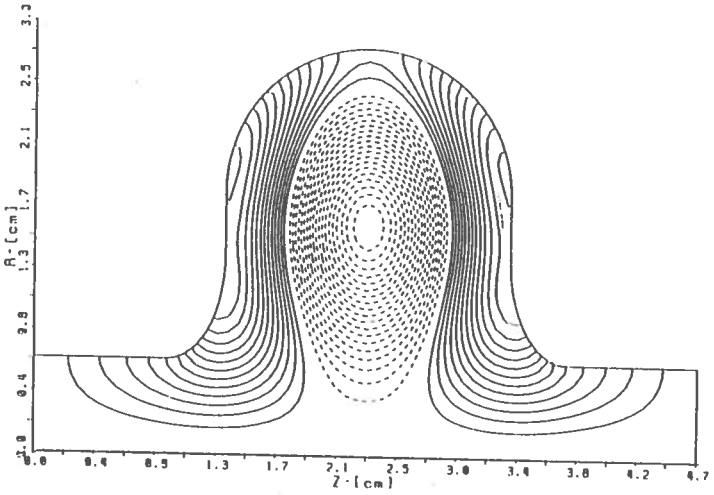
***** 10 LOWEST TM MODES *****

file EXA05M05 frequency = 14870.71 Mhz computed by the OSCAR2D code written by P.FERNANDES and A.PARODI INFN-GENOA



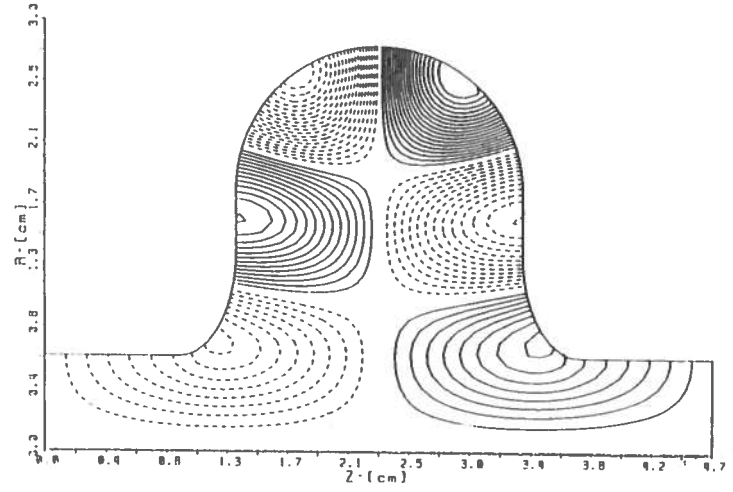
***** 10 LOWEST TM MODES *****

file EXA05M06 frequency = 16351.10 Mhz computed by the OSCAR2D code written by P.FERNANDES and A.PARODI INFN-GENOA



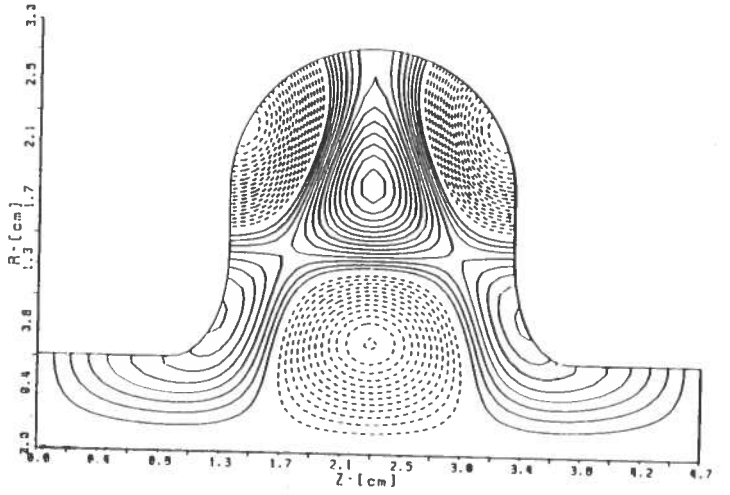
***** 10 LOWEST TM MODES *****

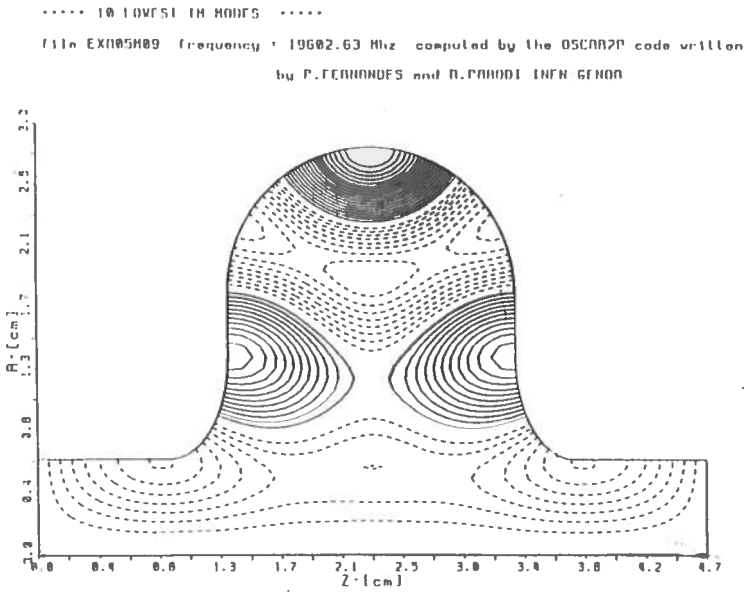
file EXA05M07 frequency = 17761.85 Mhz computed by the OSCAR2D code written by P.FERNANDES and A.PARODI INFN-GENOA



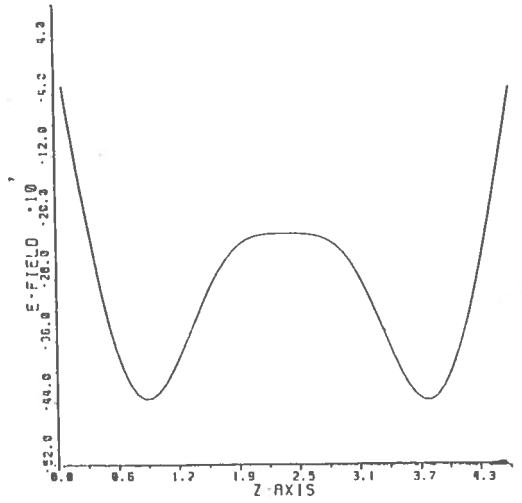
***** 10 LOWEST TM MODES *****

file EXA05M08 frequency = 18732.20 Mhz computed by the OSCAR2D code written by P.FERNANDES and A.PARODI INFN-GENOA

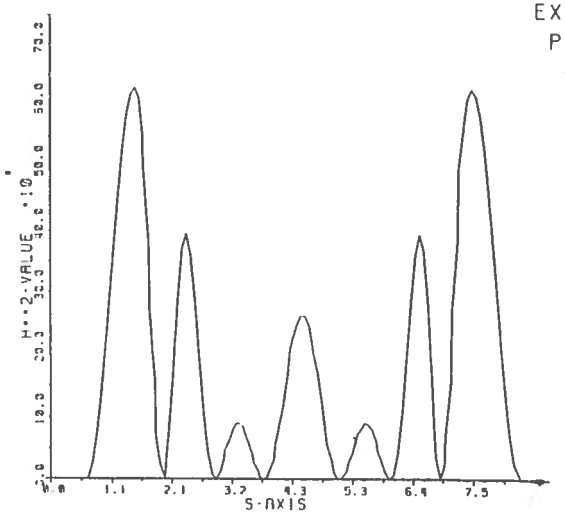




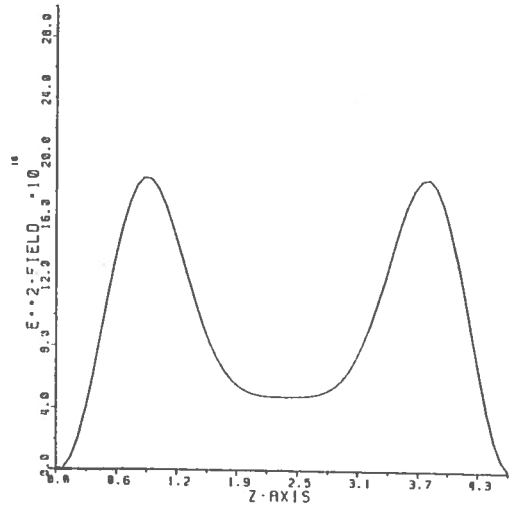
EXA05M09
AXIAL FIELD

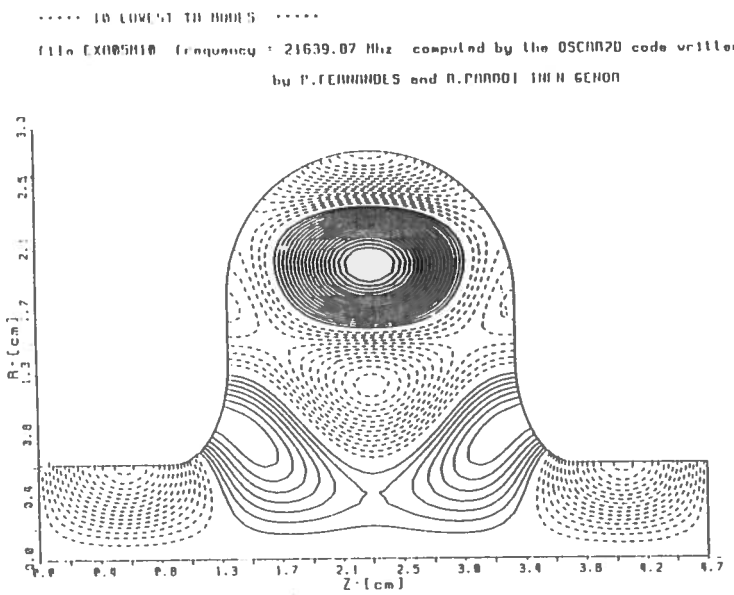


EXA05M09
POWER DENSITY
ON BOUNDARY

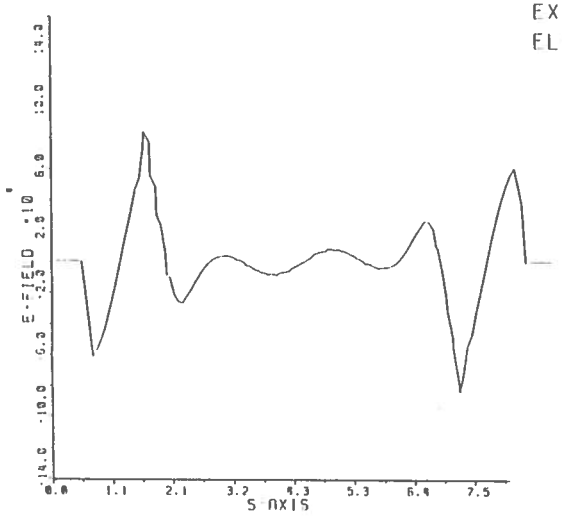


EXA05M09
AXIAL FIELD

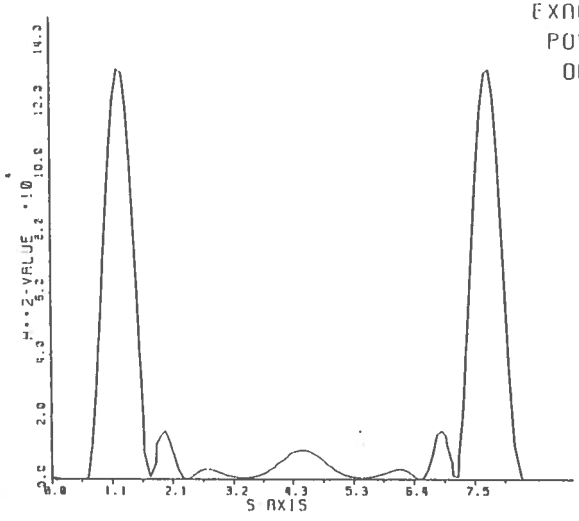




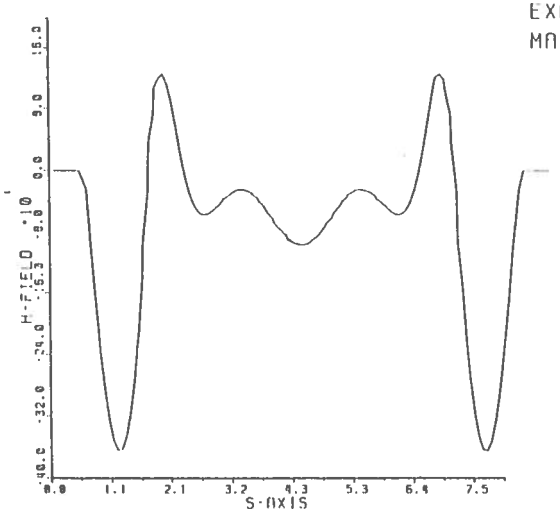
EXA05M10
ELECTRIC FIELD
ON BOUNDARY



EXA05M10
POWER DENSITY
ON BOUNDARY

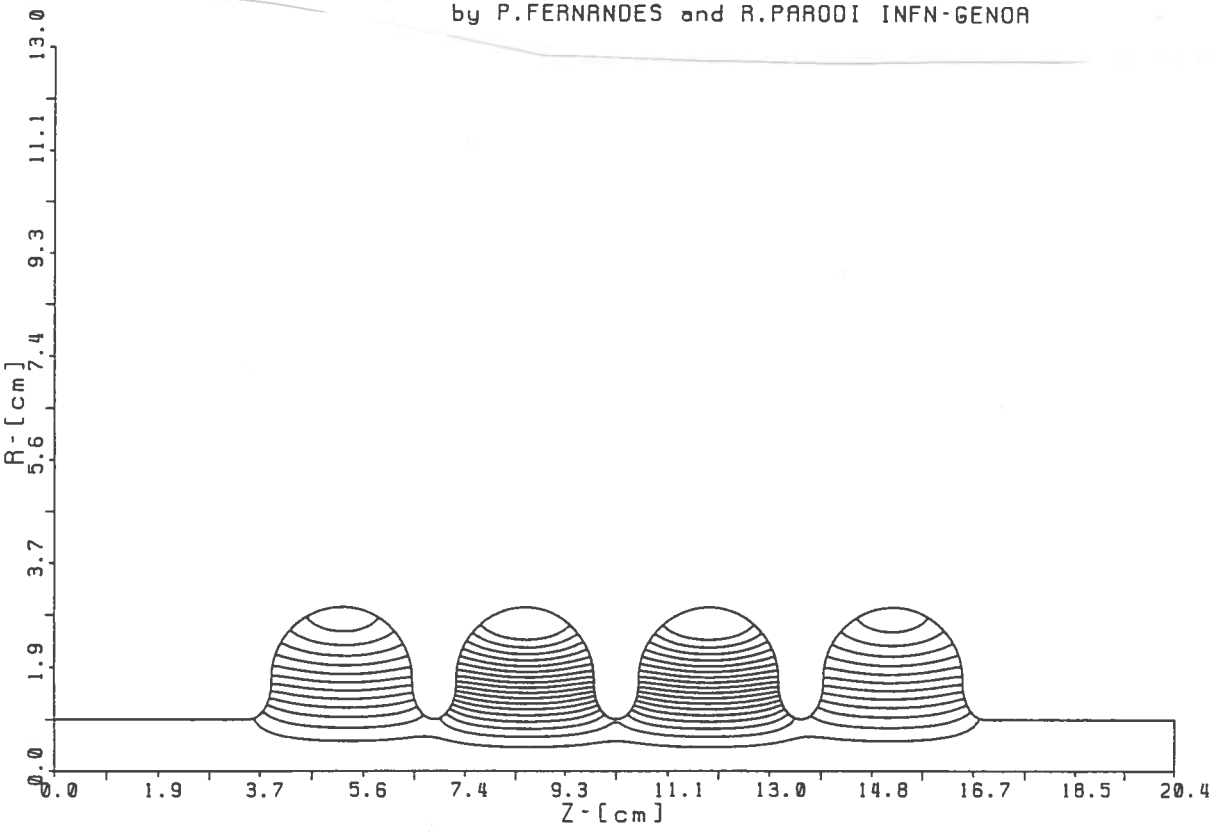


EXA05M10
MAGNETIC FIELD
ON BOUNDARY



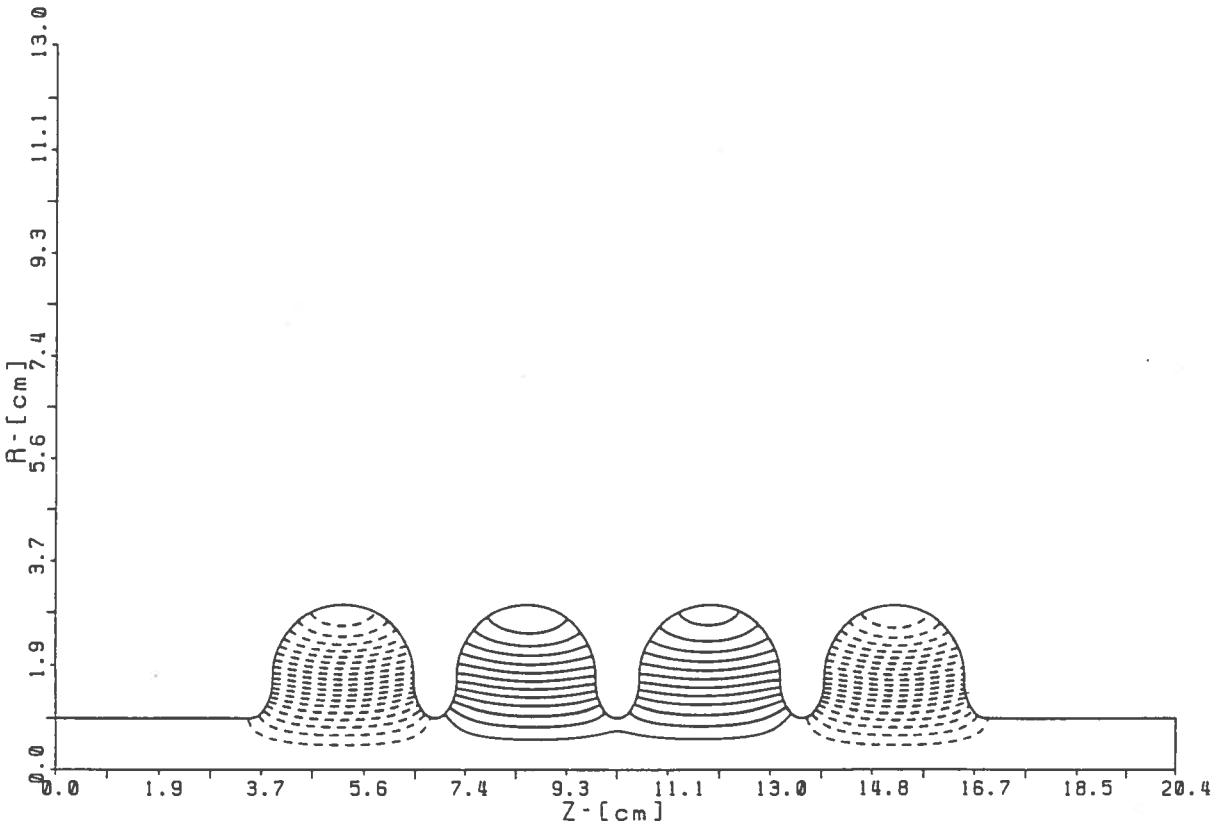
••••• 4-CELL STRUCTURE (8 LOWEST TM MODES) •••••

file EXAMPLM01 frequency = 4479.36 Mhz computed by the OSCAR2D code written by P.FERNANDES and R.PARODI INFN-GENOA



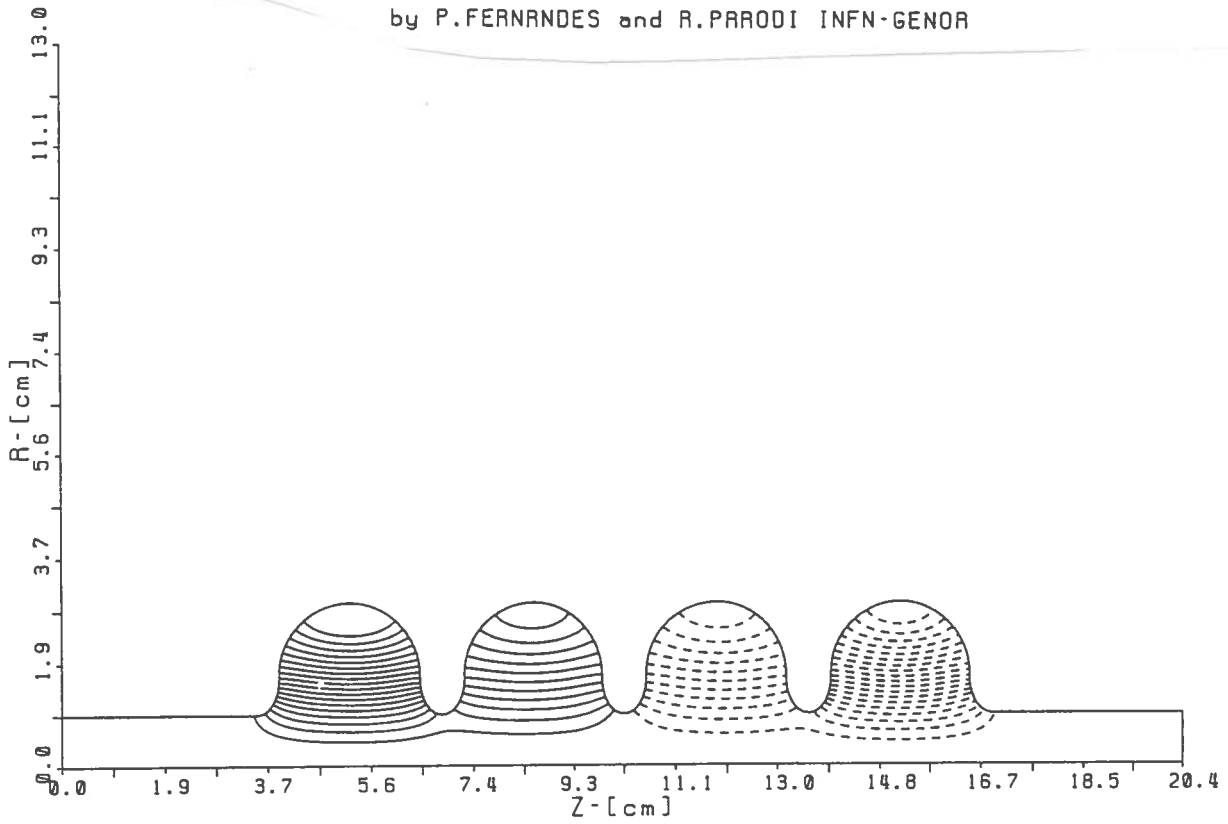
••••• 4-CELL STRUCTURE (8 LOWEST TM MODES) •••••

file EXAMPLM02 frequency = 4512.82 Mhz computed by the OSCAR2D code written by P.FERNANDES and R.PARODI INFN-GENOA



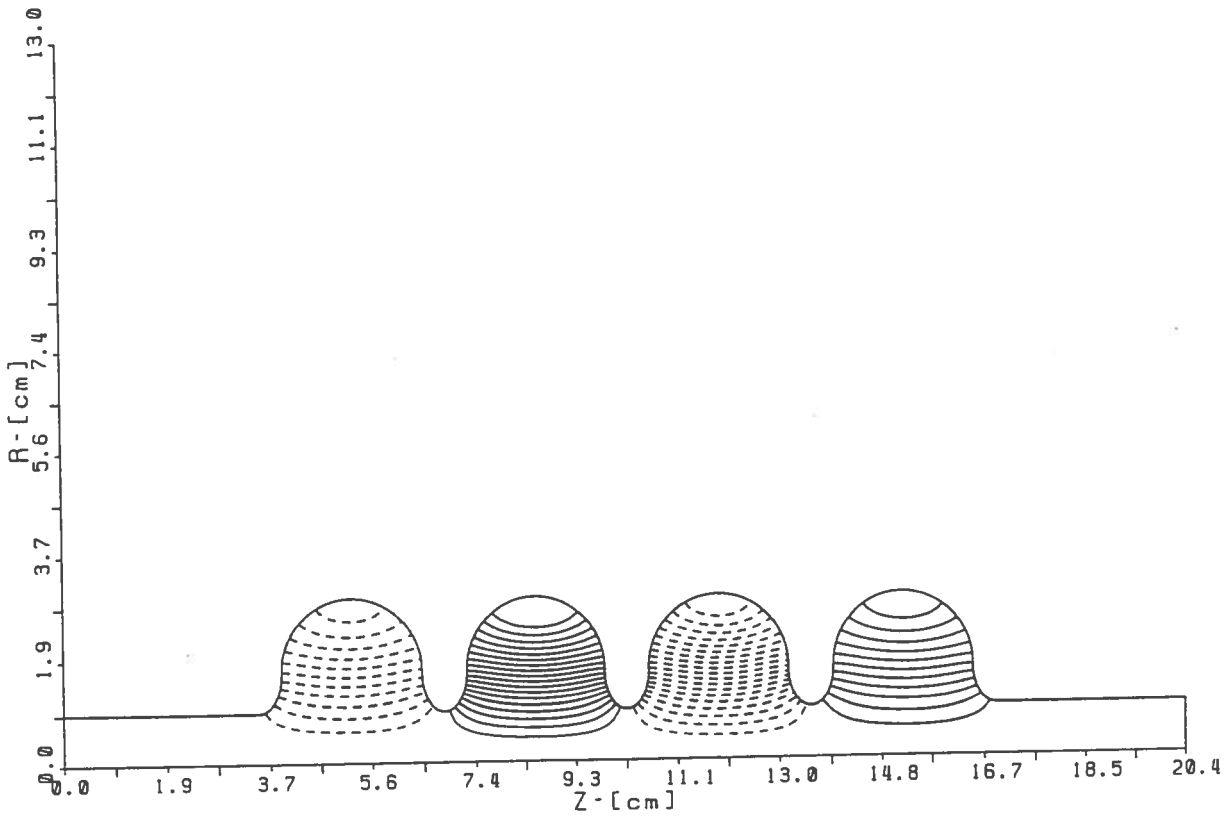
••••• 4-CELL STRUCTURE (8 LOWEST TM MODES) •••••

file EXAMPLM03 frequency = 4493.96 Mhz computed by the OSCAR2D code written by P.FERNANDES and R.PARODI INFN-GENOA

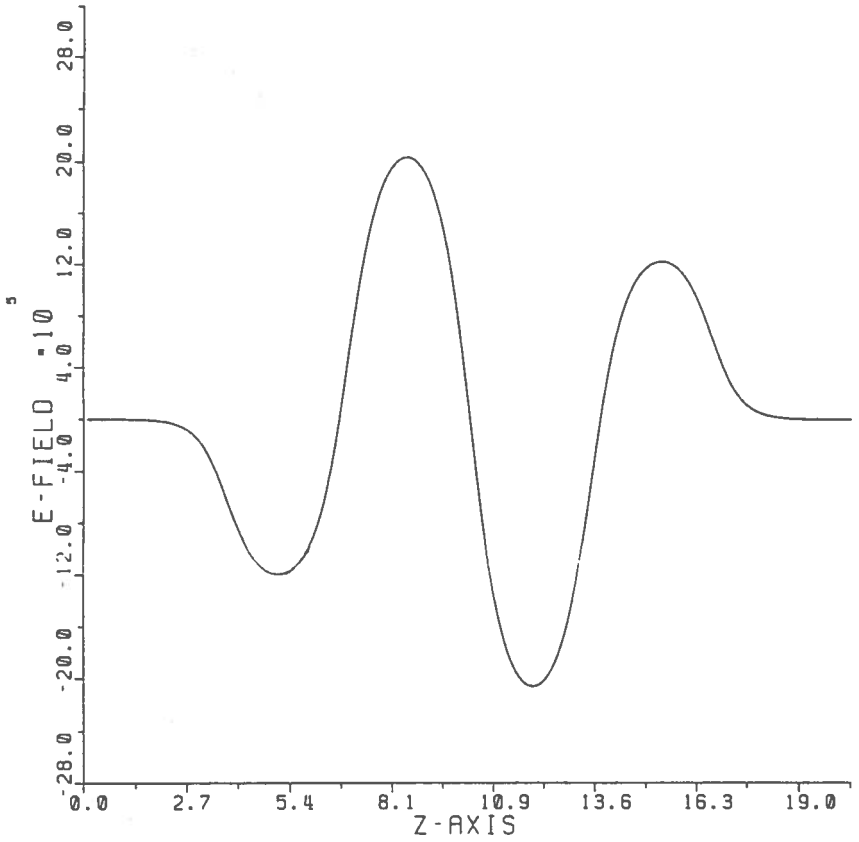


••••• 4-CELL STRUCTURE (8 LOWEST TM MODES) •••••

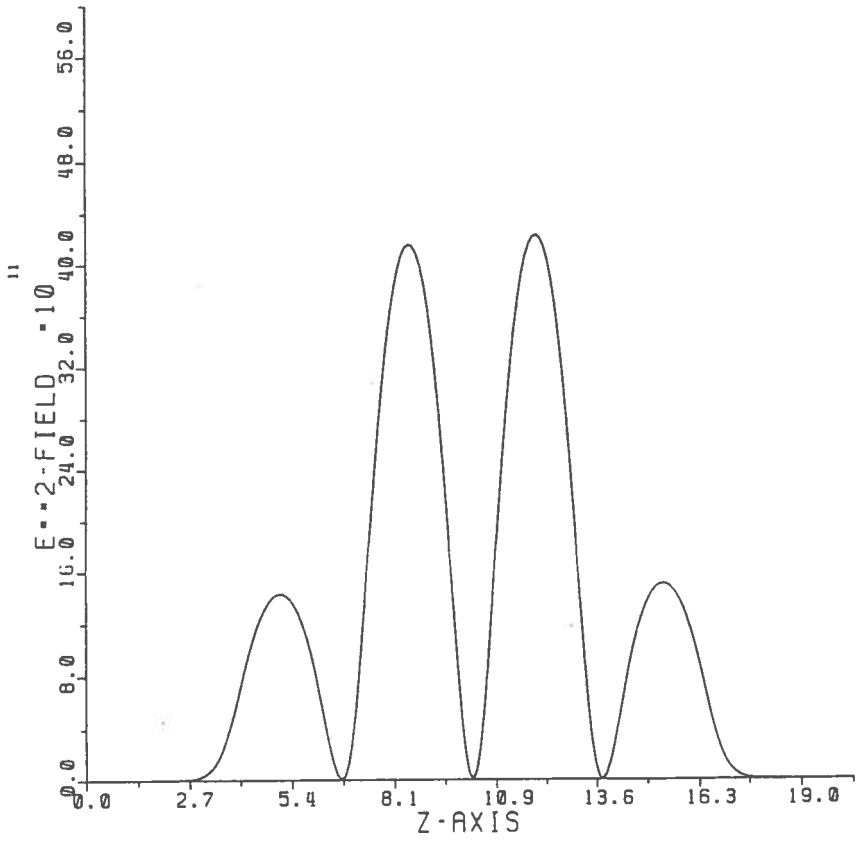
file EXAMPLM04 frequency = 4528.69 Mhz computed by the OSCAR2D code written by P.FERNANDES and R.PARODI INFN-GENOA

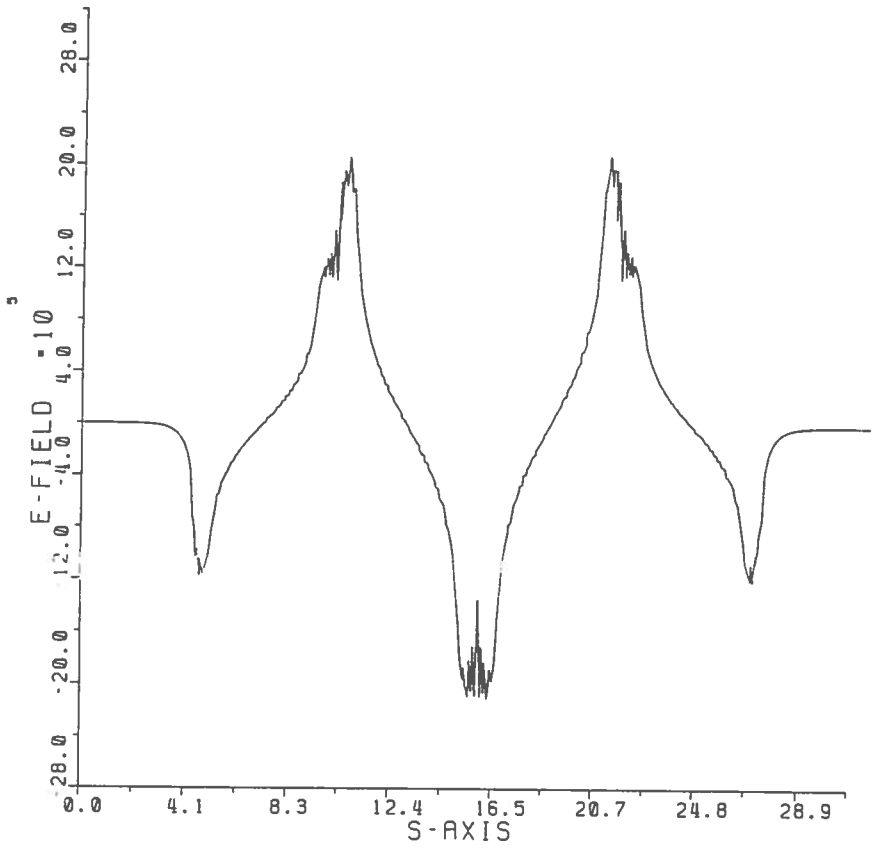


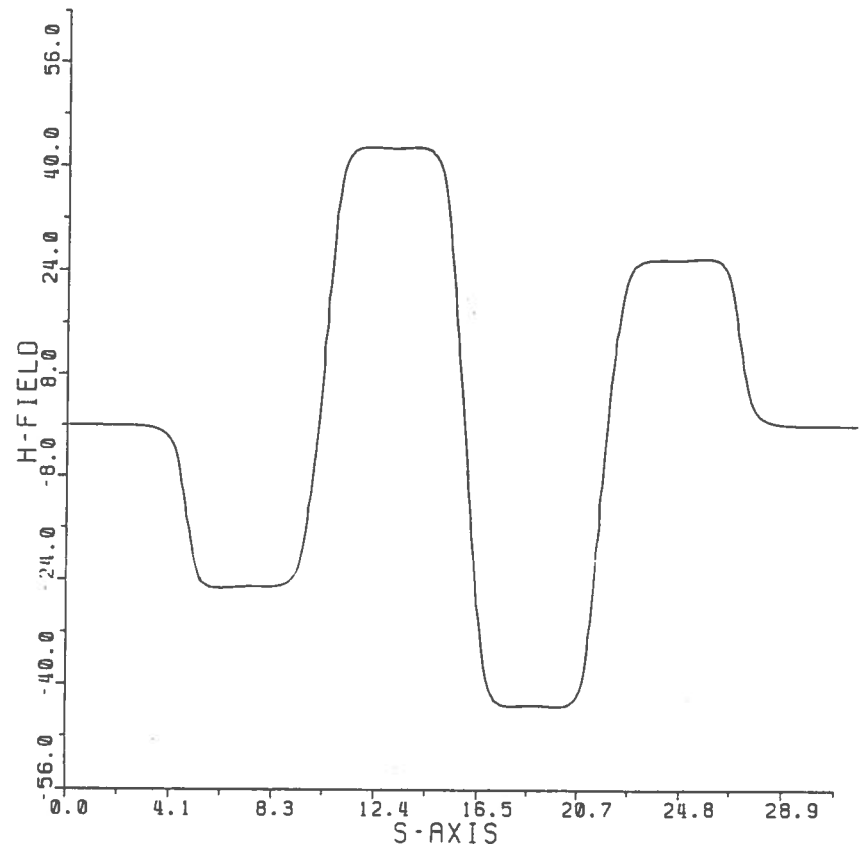
EXAMPLM04
AXIAL FIELD

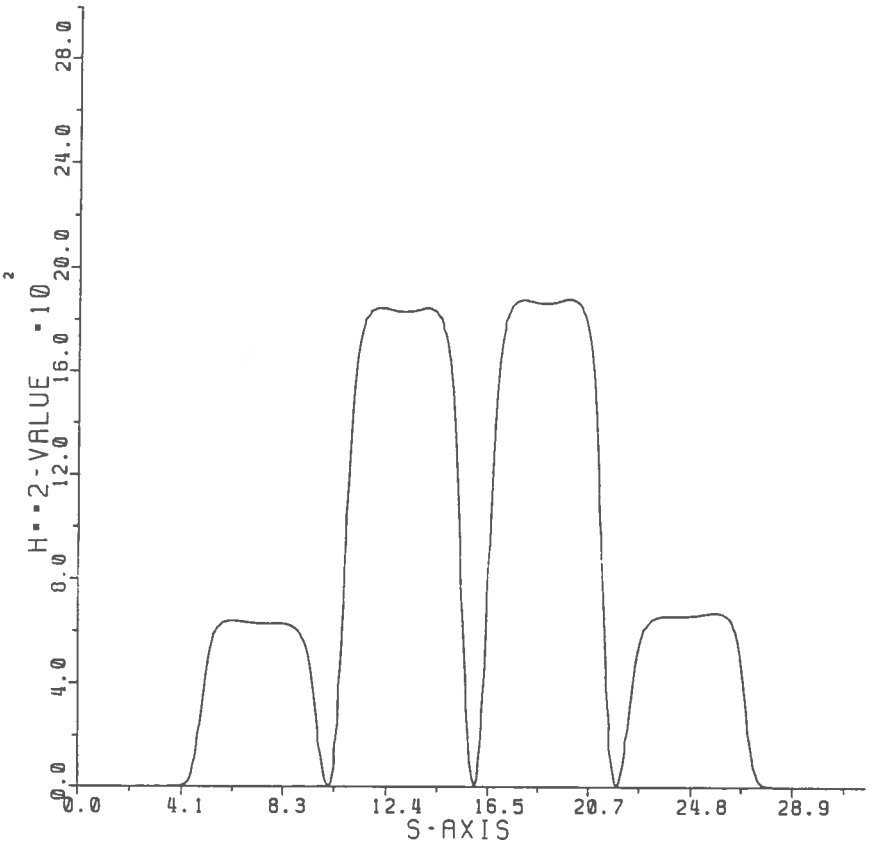


EXAMPLM04
AXIAL FIELD



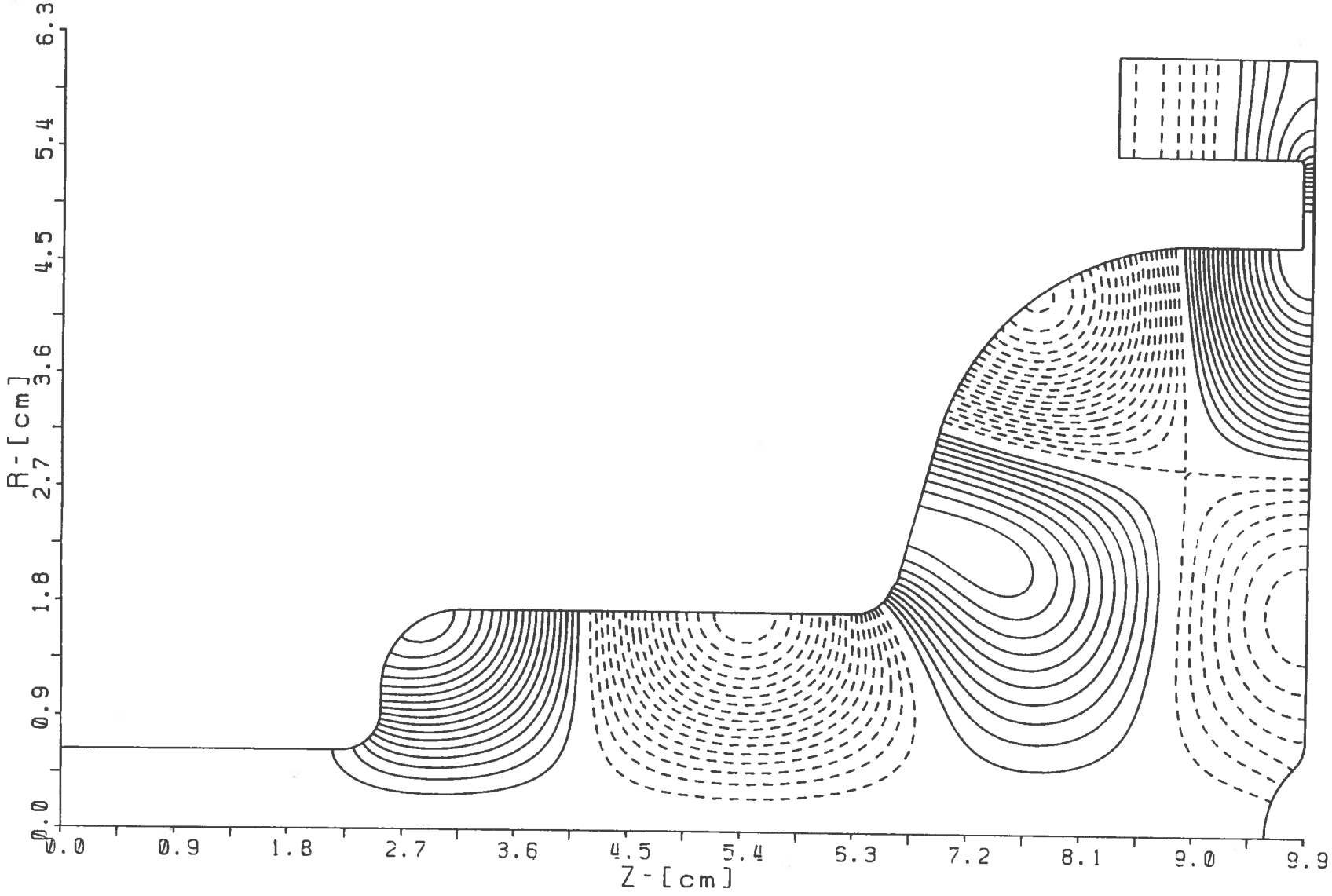

EXAMPLM04
ELECTRIC FIELD
ON BOUNDARY
E-FIELD *10^3
28.0
20.0
12.0
4.0
-4.0
-12.0
-20.0
-28.0
0.0
4.1
8.3
12.4
16.5
20.7
24.8
28.9
S-AXIS

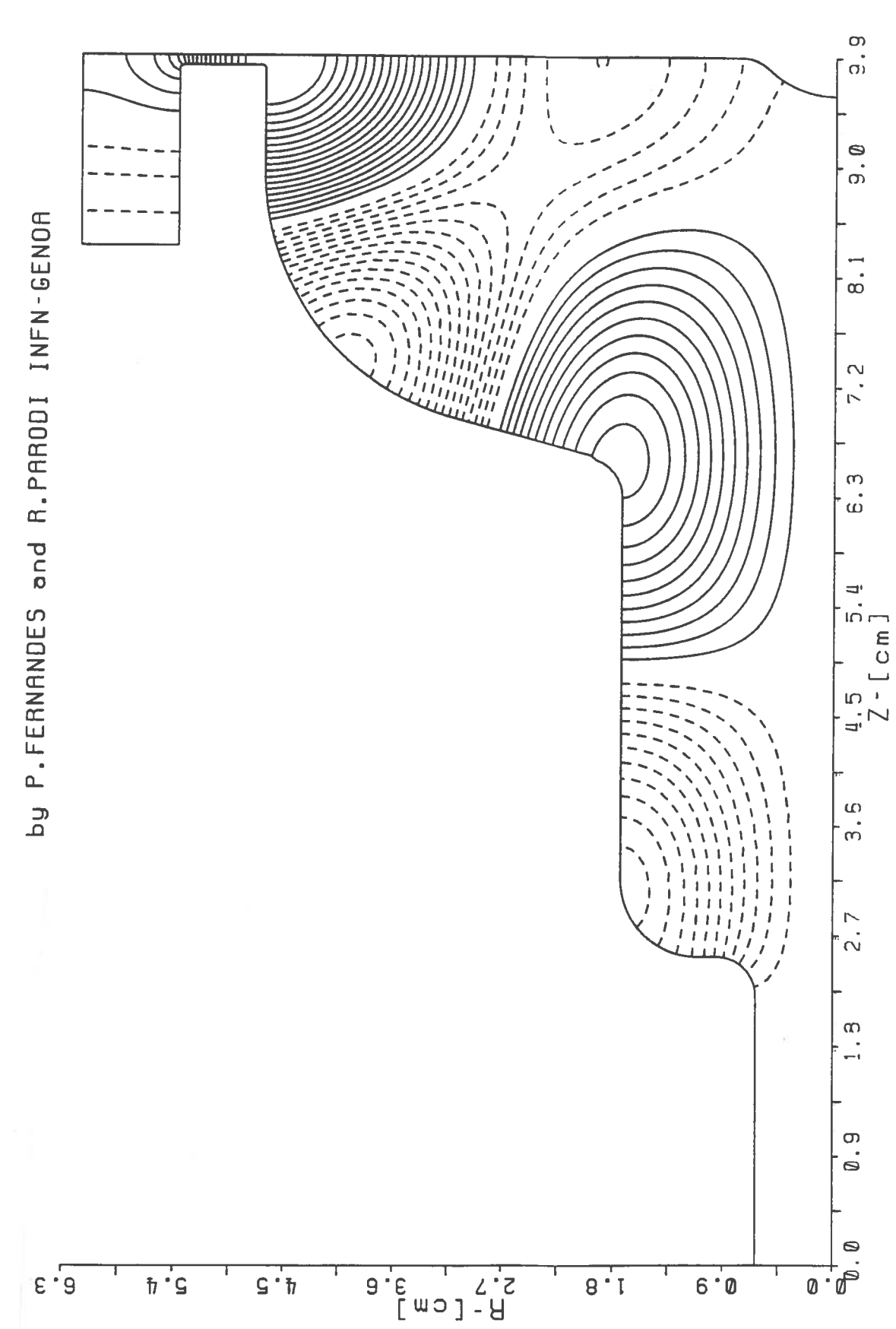

EXAMPLM04
MAGNETIC FIELD
ON BOUNDARY
H-FIELD
56.0
40.0
24.0
8.0
-8.0
-24.0
-40.0
-56.0
0.0
4.1
8.3
12.4
16.5
20.7
24.8
28.9
S-AXIS

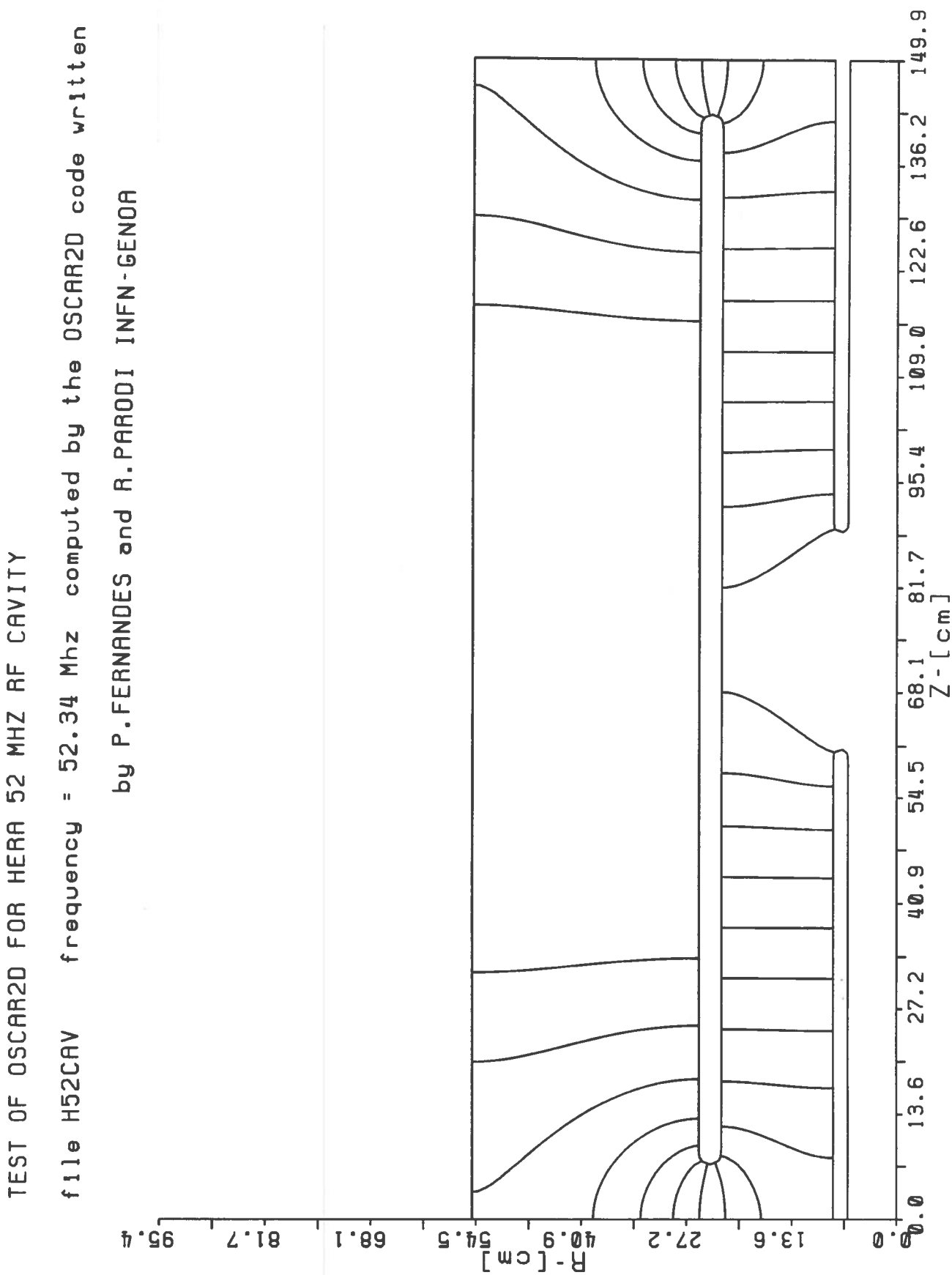

EXAMPLM04
POWER DENSITY
ON BOUNDARY
H**2-VALUE *10**2
0.0
4.0
8.0
12.0
16.0
20.0
24.0
28.0
0.0
4.1
8.3
12.4
16.5
20.7
24.8
28.9
S-AXIS

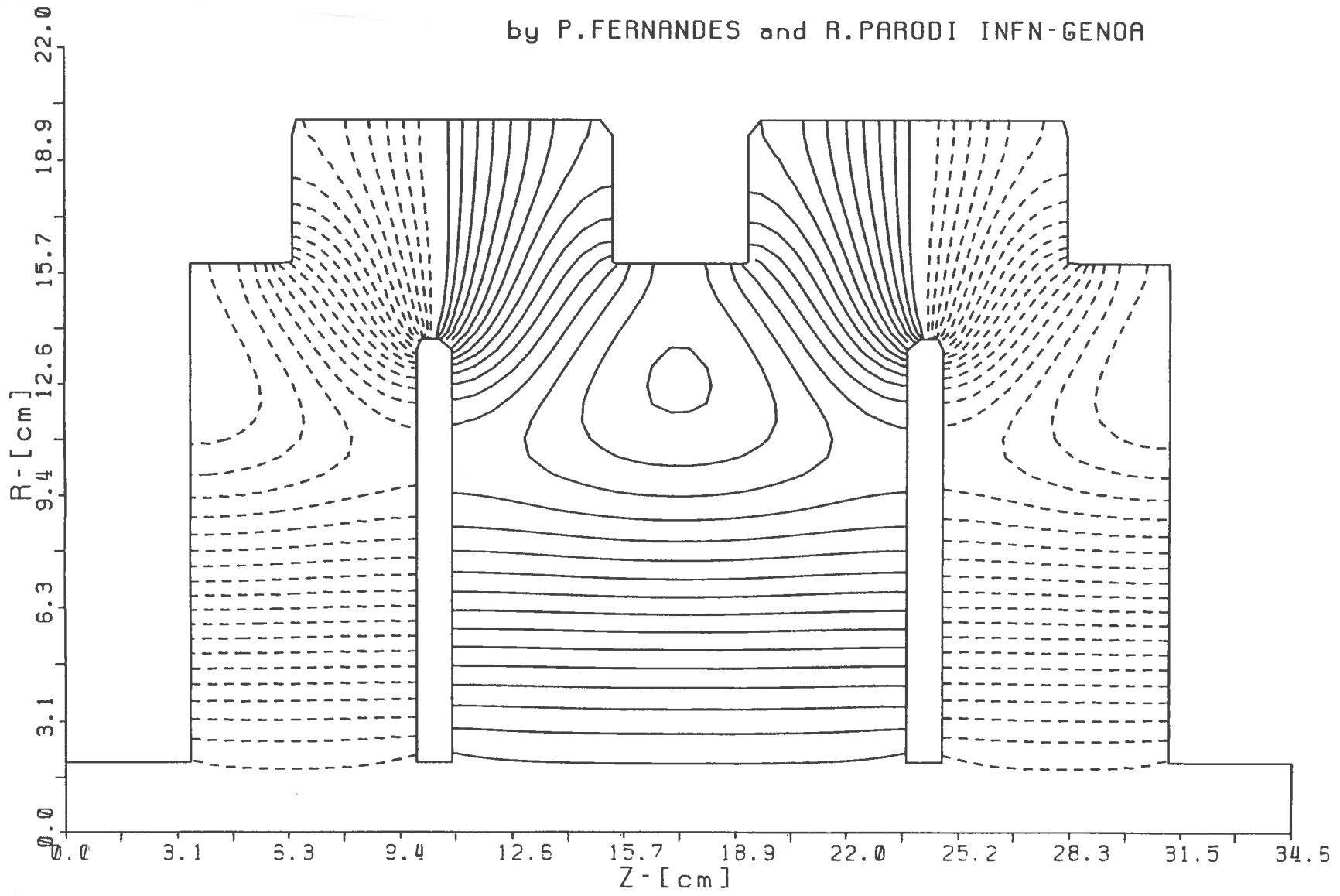
***** MOFFAT CAVITY 8 MODES *****

file MOFFATM08 frequency = 8764.27 Mhz computed by the OSCAR2D code written by P.FERNANDES and R.PARODI INFN-GENOA




***** MOFFAT CAVITY 8 MODES *****
file MOFFATM06 frequency = 7776.07 Mhz computed by the OSCAR2D code written
by P.FERNANDES and R.PARODI INFN-GENOA
R- [cm]
0.0
0.9
1.8
2.7
3.6
4.5
5.4
6.3
Z- [cm]
0.0
0.9
1.8
2.7
3.6
4.5
5.4
6.3
7.2
8.1
9.0
9.9


TEST OF OSCAR2D FOR HERA 52 MHZ RF CAVITY
file H52CAV frequency = 52.34 Mhz computed by the OSCAR2D code written
by P.FERNANDES and R.PARODI INFN-GENOA
R - [cm]
Z - [cm]
0.0
13.6
27.2
40.9
54.5
68.1
81.7
95.4
109.0
122.6
136.2
149.9


••• DAW STRUCTURE (MULTIPLY CONNECTED) •••
file DAW frequency = 1012.82 Mhz computed by the OSCAR2D code written
by P.FERNANDES and R.PARODI INFN-GENOA
R-[cm]
0.0
3.1
6.3
9.4
12.6
15.7
18.9
22.0
Z-[cm]
0.0
3.1
6.3
9.4
12.6
15.7
18.9
22.0
25.2
28.3
31.5
34.5

APPENDIX B:

A COMPLETE OUTPUT.

In the following pages the complete output (both printed and graphic) of EXAMPLE06 is reported.
GRAPH=7 was added in the EXAMPLE06.INP input file to generate all the intermediate files needed for graphic postprocessing (i.e. TM031.CHN, TM031.AXI and TM031.BOU).

The complete output consists of the following items:

1. Printed output with default options
   TM031.OUT output file produced by running EXAMPLE06.INP (or EXAMPLE06.OUT output file in the distribution kit)

2. Plot of the electric field lines
   Obtained by processing the TM031.CHN file by the FLF and FLT postprocessors

3. Graphs of the field along the axis
   Obtained by processing the TM031.AXI file by the FLDAXI postprocessor

   i Electric field

   ii Electric field squared

4. Graphs of the fields on the boundary
   Obtained by processing the TM031.BOU file by the FLDBOU postprocessor

   i Electric field

   ii Magnetic field

   iii Magnetic field squared (power density)

Some information about graphic postprocessors and a wide set of graphic output samples can be found in Appendix A.

***** SINGLE HIGHER TM MODE BY RQI+BCG ***** PROGRAM OSCAR2D, VERSION 2.0

DATE: 6-FEB-90 TIME: 16:58:54

CYLINDRICAL COORDINATES

TM MODE

NUMBER OF LINE SEGMENTS = 11 REQU. MESH SIZE H = 0.07500

RB	ZB	RC	ZC	R	R2	TYPE	METAL	SIGMA(MHO/M)
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1	0	5.91016E+07
6.50000E-01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0	1	5.91016E+07
6.50000E-01	9.30000E-01	1.35000E+00	9.30000E-01	4.00000E-01	7.00000E-01	0	1	5.91016E+07
1.35000E+00	1.33000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0	1	5.91016E+07
1.80000E+00	1.33000E+00	1.80000E+00	2.33000E+00	-1.00000E+00	-1.00000E+00	0	1	5.91016E+07
2.80000E+00	2.33000E+00	1.80000E+00	2.33000E+00	-1.00000E+00	-1.00000E+00	0	1	5.91016E+07
1.80000E+00	3.33000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0	1	5.91016E+07
1.35000E+00	3.33000E+00	1.35000E+00	3.73000E+00	4.00000E-01	7.00000E-01	0	1	5.91016E+07
6.50000E-01	3.73000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0	1	5.91016E+07
6.50000E-01	4.66000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1	0	5.91016E+07
0.00000E+00	4.66000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1	0	5.91016E+07

USED MESH SIZE H = 0.07516

FULL CAVITY PROBLEM
BETA (RELATIVE VELOCITY) = 0.8550000E+00
AUTOMATIC COMPUTATION OF THE DRIFT LENGTH

SECOND ORDER APPROXIMATION OF BOUND. COND. (2 DIM.)

ONE STEP OF "RAYLEIGH QUOTIENT ITERATION + BICONJUGATE GRADIENT"

NO SECOND STEP

ONE MODE ONLY
NO SYMMETRY FORCED

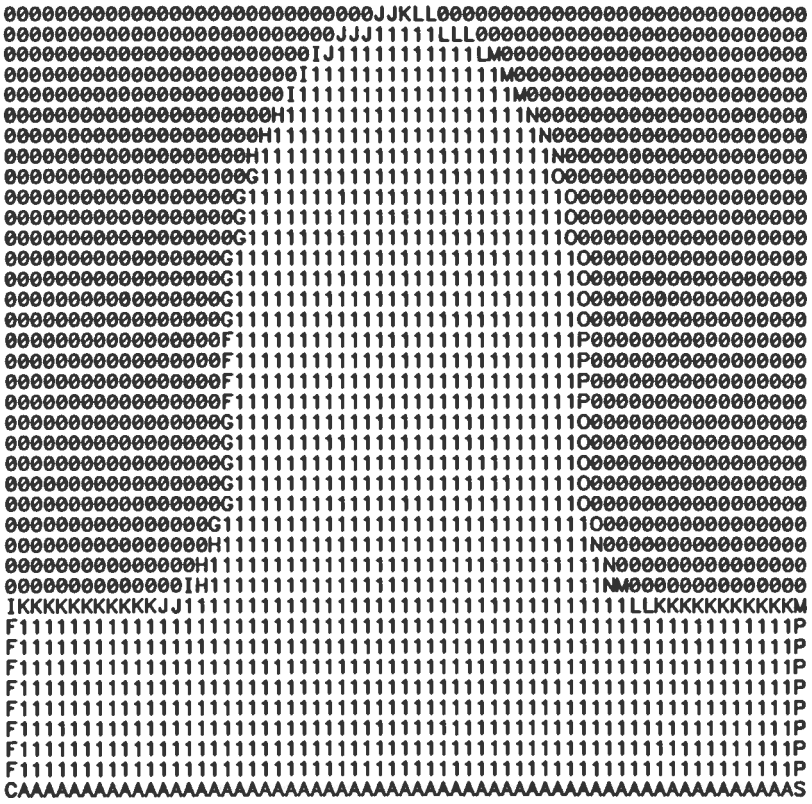
INITIAL FREQUENCY = 18000.00

BESSEL LOAD WITH 2 R NODE(S) AND 1 Z NODE(S)
Z NODES:
0.2330000E+01

LAMBDA CONVERGENCE = 0.1000000E-04 F CONVERGENCE = 0.1000000E-04
ITERATION TIME ALLOWED = 1800 SECONDS

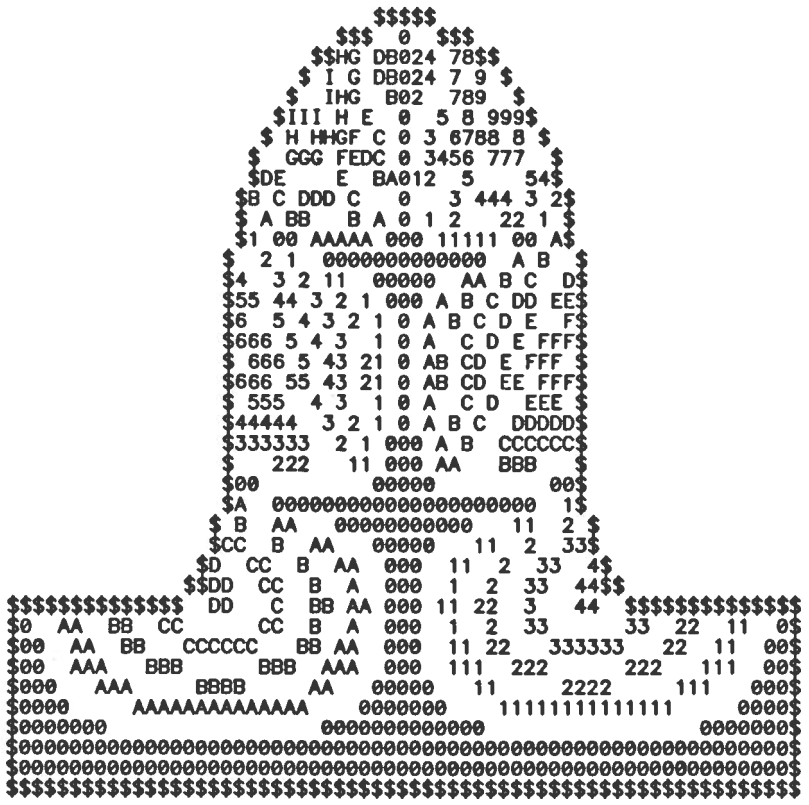
OUTPUT FILE NAME: TM031

THIS RUN REQUIRES 94866 WORDS OF DYNAMIC MEMORY
(REQU. BOUNDARY POINTS = 312)

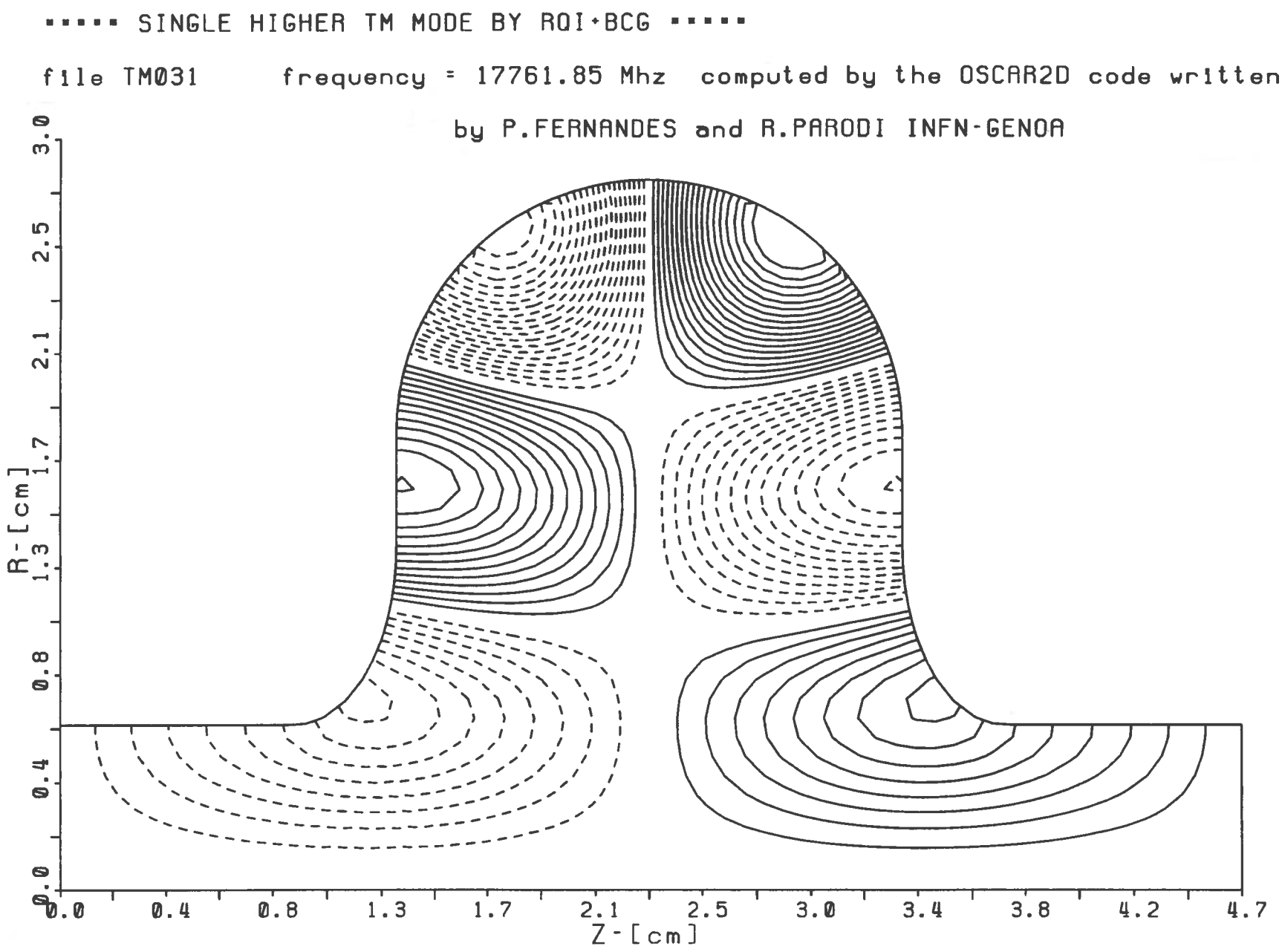


MESH POINTS ALONG Z = 63
MESH POINTS ALONG R = 39
TOTAL MESH POINTS = 2457
BOUNDARY POINTS = 202

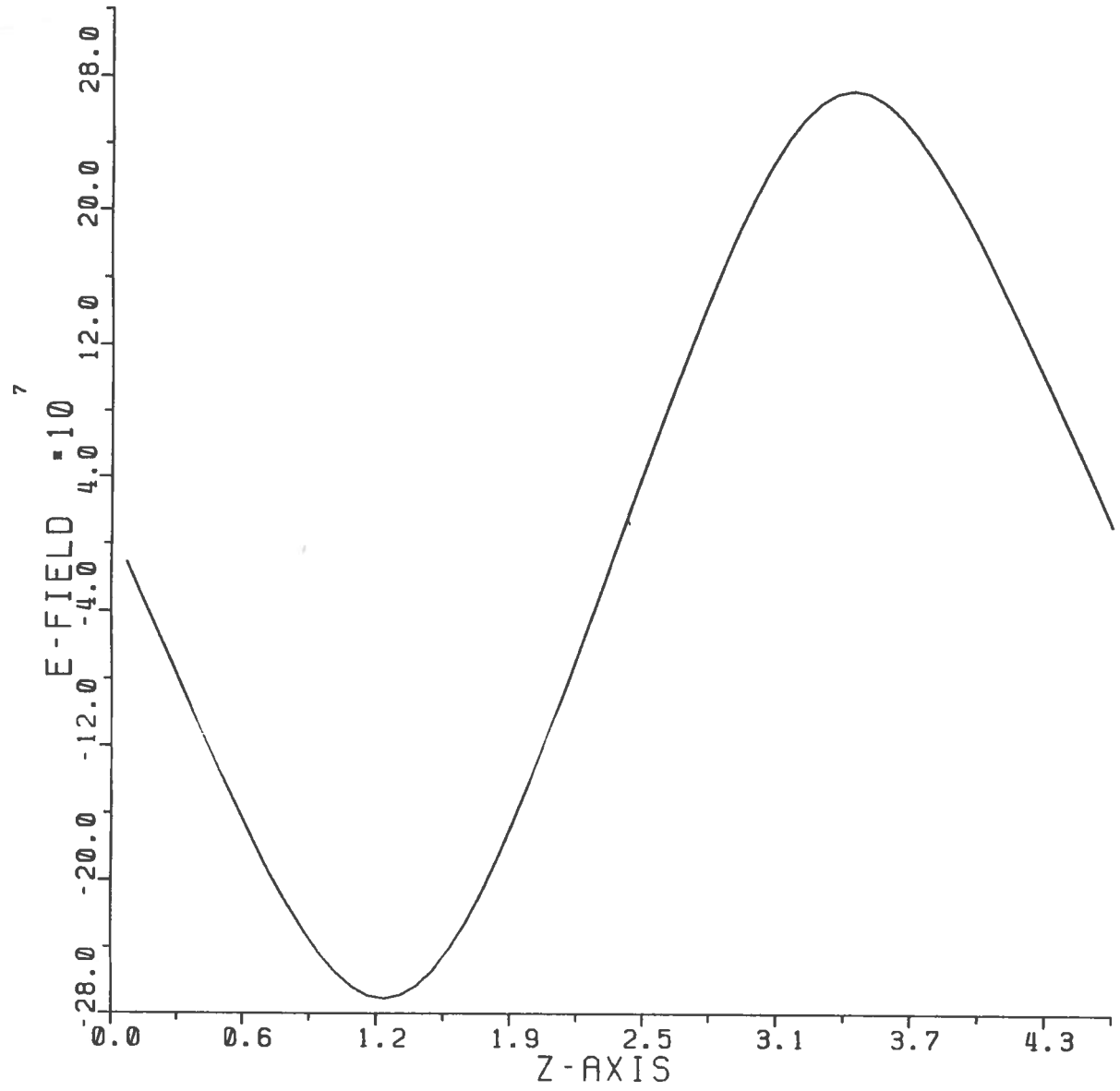
```
LAMBDA =  1.3952617E+01  RESIDL =  1.598E-03  DL/L = -1.963E-02  IN  114 BCG ITERATIONS
LAMBDA =  1.3907908E+01  RESIDL =  2.456E-04  DL/L = -3.204E-03  IN  128 BCG ITERATIONS
LAMBDA =  1.3907756E+01  RESIDL =  8.913E-07  DL/L = -1.092E-05  IN  141 BCG ITERATIONS

ITERATION TIME =     46

FINAL LAMBDA =    0.1385776E+02          (FREQUENCY =   17761.83280 MHZ)



 ***** SINGLE HIGHER TM MODE BY RQI+BCG *****

NORMALIZATION FACTOR =  9.03741E+03

STRUCTURE LENGTH L =  4.66000E+00 CENTIMETERS

FREQUENCY =   17761.83280 MEGA-HERTZ

W (=V*T/L) =  1.00000E+00 MEGA-VOLTS/METER

AXIAL VOLTAGE V =  7.65074E+06 VOLTS

BETA (RELATIVE VELOCITY) = 0.855000
PERIODIC LENGTH =    0.7215549E+00 CM,  DRIFT LENGTH =    0.1608445E+01 CM

PARTICLE ENERGY =  4.74310E-01 MEV

TRANSIT TIME FACTOR T =  6.09092E-03

ENERGY STORED IN CAVITY U =  9.19617E-01 JOULES   POWER LOSS P =  5.78716E+06 WATTS

MAXIMUM SURFACE ELECTRIC FIELD =  1.29507E+08 VOLTS/METER AT R =  1.08984E+00  Z =  3.35937E+00  E/SQRT(P*Q) = 4.04255E+02

MAXIMUM AXIAL ELECTRIC FIELD = -2.70957E+08 VOLTS/METER AT R =  0.00000E+00  Z =  1.20258E+00

MAXIMUM SURFACE MAGNETIC FIELD = -5.01328E+03 GAUSS       AT R =  7.46405E-01  Z =  1.13249E+00  H/SQRT(P*Q) =-1.56489E-02

MAXIMUM AXIAL MAGNETIC FIELD =  0.00000E+00 H GAUSS     AT R =  0.00000E+00  Z =  0.00000E+00

SHUNT IMPEDANCE Z (=V**2/P) =  1.01144E+07 OHMS

Z/L =   2.17048E+08  Z/(L*Q) =   1.22390E+04  Z*T**2 =   3.75238E+02  Z*T**2/L =   8.05231E+03  Z*T**2/(L*Q) =   4.54058E-01

Q FACTOR =  1.77341E+04
MAGNETIC GEOMETRIC FACTOR =  6.10848E+02 OHMS
ELECTRIC GEOMETRIC FACTOR =  1.16853E+03 OHMS

                              POWER LOSS
SURFACE (LINE SEGMENT) #  2     3.10640E+05
SURFACE (LINE SEGMENT) #  3     5.60704E+05
SURFACE (LINE SEGMENT) #  4     5.67702E+05
SURFACE (LINE SEGMENT) #  5     1.45721E+06
SURFACE (LINE SEGMENT) #  6     1.52833E+06
SURFACE (LINE SEGMENT) #  7     5.38600E+05
SURFACE (LINE SEGMENT) #  8     5.78590E+05
SURFACE (LINE SEGMENT) #  9     2.45393E+05
```

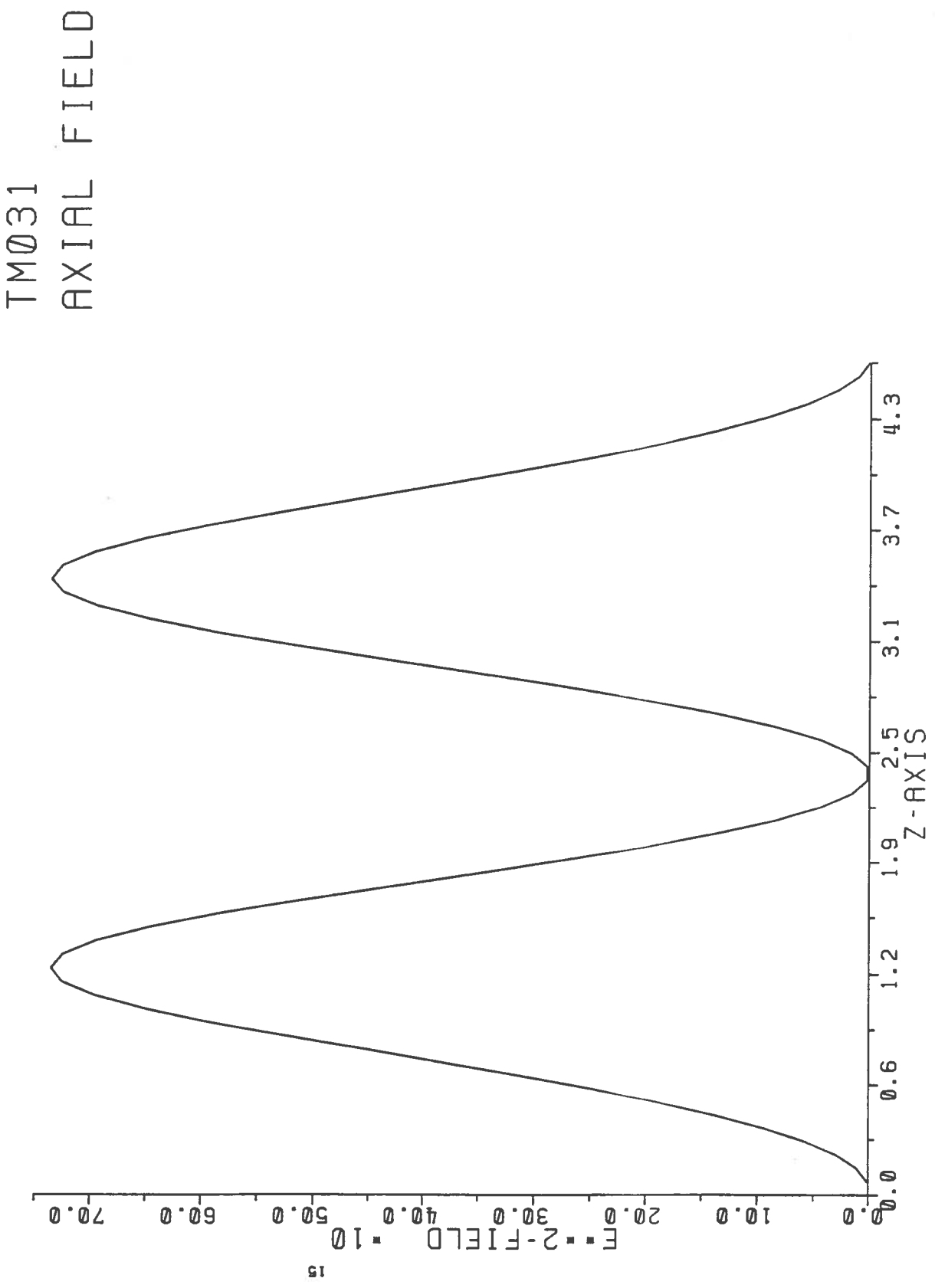


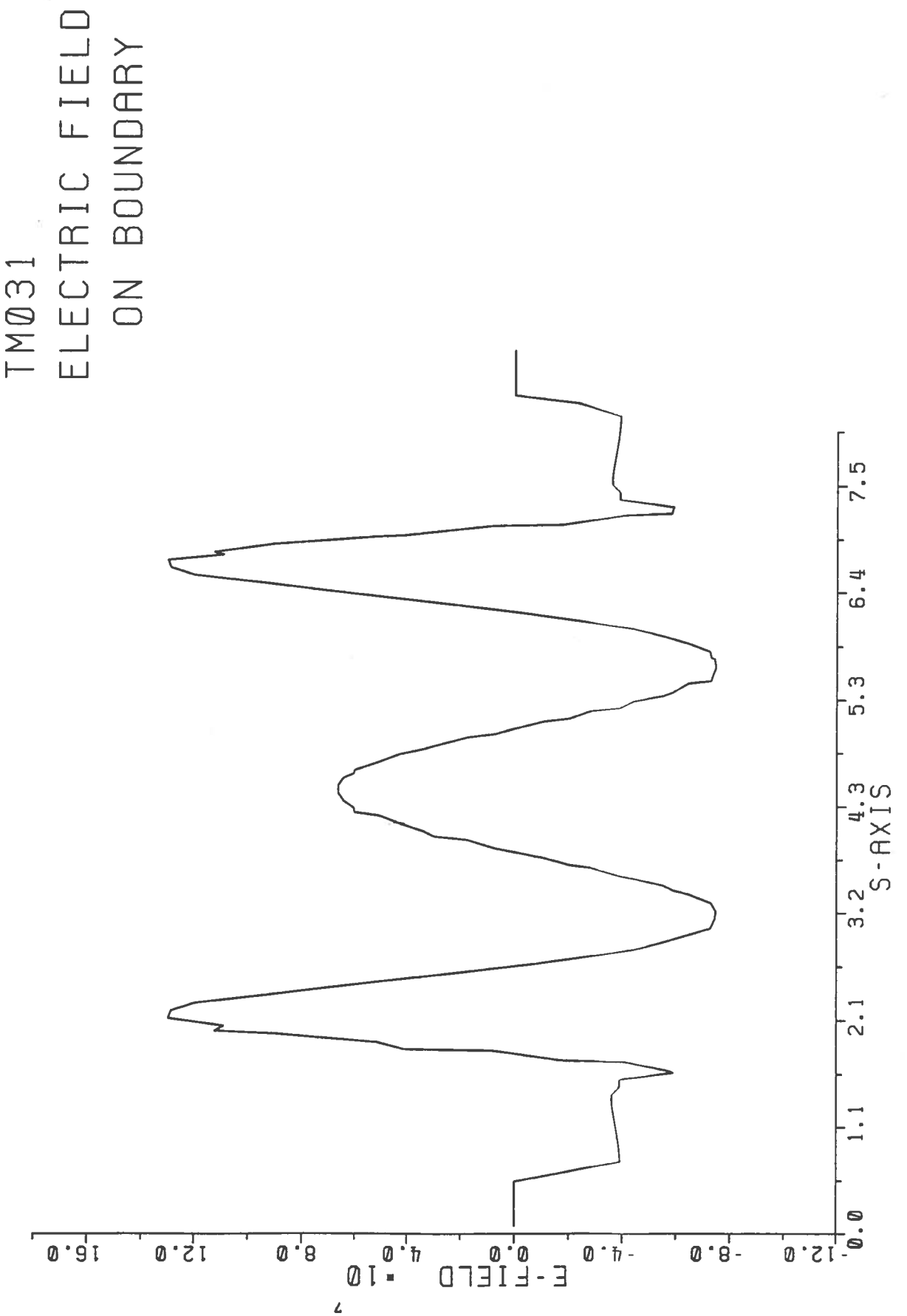
```
MESH POINTS ALONG Z =   63
MESH POINTS ALONG R =   39
TOTAL MESH POINTS =     2457
BOUNDARY POINTS =   202

TOTAL TIME =     50
```

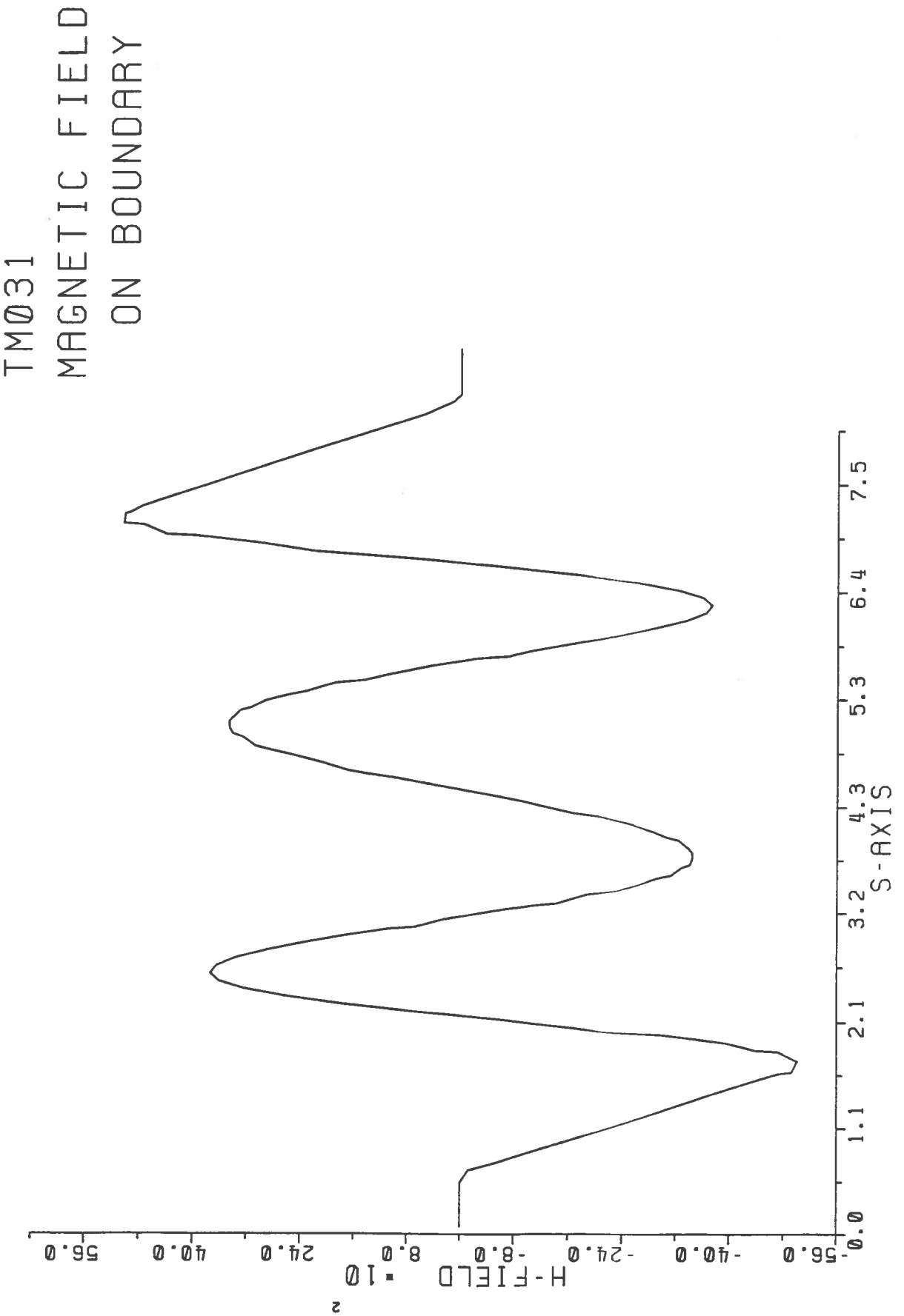

***** SINGLE HIGHER TM MODE BY ROI+BCG *****
file TM031 frequency = 17761.85 Mhz computed by the OSCAR2D code written
by P.FERNANDES and R.PARODI INFN-GENOA
R-[cm]
0.0
0.4
0.8
1.3
1.7
2.1
2.5
3.0
Z-[cm]
0.0
0.4
0.8
1.3
1.7
2.1
2.5
3.0
3.4
3.8
4.2
4.7

TM031
AXIAL FIELD




TM031
AXIAL FIELD
E**2-FIELD *10
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
0.0
0.6
1.2
1.9
2.5
3.1
3.7
4.3
Z-AXIS


TM031
ELECTRIC FIELD
ON BOUNDARY
E-FIELD *10
16.0
12.0
8.0
4.0
0.0
-4.0
-8.0
-12.0
0.0
1.1
2.1
3.2
4.3
5.3
6.4
7.5
S-AXIS


TM031
MAGNETIC FIELD
ON BOUNDARY
H-FIELD *10
56.0
40.0
24.0
8.0
-8.0
-24.0
-40.0
-56.0
0.0
1.1
2.1
3.2
4.3
5.3
6.4
7.5
S-AXIS