File SFCODES.DOC contains the following bookmarks. Select the topic you are trying to find from the list and double click the highlighted text.

AnisotropicMagneticMaterials
Autofish
Automesh
AutomeshErrorMessages
BadTriangles
BoundaryConditionVariables
C_ArrayInTAPE36
CFish
CFishErrorMessages
ComplementarySolution
ComplexRootFinder
ConformalMapping
ConvergenceCriteriaInFish
ConvergenceCriteriaInPoisson
CONVsettingsForLengthUnits
CoordinateSystem
DrivePointRequirements
ElectrostaticFields
Fish
FishErrorMessages
FixedPotential
FloatingElectrode
FrequencyScans
HarmonicAnalysis
HarmonicAnalysisImplementation
HarmonicAnalysisTheory
IBOUND_SpecialBoundaryIndicator
IncrementalLineRegions
InternalPermeabilityTables
IsotropicMagneticMaterials
LineRegions
MeshModification
MeshSizeControl
MeshSizeGuidelines
MaterialsForElectrostaticProblems
MaterialsForRFproblems
MT_namelist
NamelistVariables
NumberOfMeshIntervals
OverlappingRegions
Pandira
PandiraErrorMessages
PermanentMagnetProperties
PermanentMagnetSetup
PO_namelist
POA_namelist
Poisson
PoissonErrorMessages
PoissonFieldOutput
PoissonStoredEnergy
ProblemBoundaries
REG_namelist
RootFinder
SampleInputFiles
ScalingToDesiredMagneticField
StackingFactorCorrection
TriangleType
VariablesDXandDY
VariableEasyAxisDirection
VariablesInAnyREG
VariablesInFirstREG

File SFCODES.DOC Table of Contents

V. Autofish: Combined Automesh, Fish, and SFO ____ 147

VI. Automesh: the Poisson Superfish Mesh Generator ____ 148

A. Starting program Automesh ____ 148

1. Using the Esc key to stop Automesh early ____ 148

B. Automesh files and filename conventions ____ 149

C. Input file of namelist variables ____ 149

1. REG namelist: Input of region data and problem parameters ____ 153
   a. Variables that can only appear in the first REG namelist ____ 153
   b. Entry of line-region array variables ____ 154
   c. Default values for the mesh increments DX and DY ____ 154
   d. Variables that can appear in the first and any other REG namelist ____ 155
   e. Material numbers ____ 155
   f. The MODE setting for declaring material properties ____ 155
2. PO namelist: Input of boundary-point data in Automesh ____ 156
   a. Problem boundaries, requirements for region 1 ____ 157
   b. Straight line segments ____ 158
   c. Circular and elliptical arc segments ____ 158
   d. Hyperbolic segments ____ 158
   e. Obsolete PO namelist variable NEW ____ 159
3. MT namelist: Input of material data in Automesh ____ 159
   a. Material data for rf problems ____ 162
   b. Material data for electrostatic problems ____ 163
   c. Isotropic magnetic materials ____ 164
   d. Anisotropic and permanent-magnet materials ____ 166
   e. Anisotropic material with variable easy axis direction ____ 168
   f. Permanent magnets and real currents ____ 168
4. POA namelist: Input of single-point boundary values ____ 169

D. Sample Automesh input files ____ 170

1. Superfish pillbox cavity ____ 170
2. Example with an elliptical segment ____ 171
3. Poisson and Pandira H-shaped magnet ____ 171

E. Solving for rf and electrostatic fields in the same geometry ____ 172

F. Entering coordinate data in units other than centimeters ____ 173

G. Specifying boundary conditions and fixed potentials ____ 175

1. Boundary conditions at the edges of the problem geometry ____ 175
2. Using IBOUND to indicate boundary conditions ____ 175
3. Using IBOUND to indicate fixed-potential values ____ 176
4. Using the POA namelist to indicate a single-point fixed-potential ____ 176
5. Simulating a floating electrode ____ 176

H. Specifying rectangular or cylindrical coordinate systems ____ 176

I. Complementary solutions for rf cavities ____ 176

J. SF.INI settings for Automesh 179

K. Mesh optimization considerations 179
1. Using line regions to control the mesh size in Automesh 179
2. Choosing the mesh size 181
3. Other tips related to meshing the problem 182
4. Mesh modification option 182

L. Drive points for Superfish problems 184
1. Specifying a single drive point 184
2. Line drives for CFish problems 185
3. Wavelength and starting phase for a sinusoidal drive line 185

M. Overlapping and adjacent regions 185

N. Indicating the triangle type used to fill the mesh 186
1. Bad triangles: overlapping, zero-area, and near zero-area triangles 187
   a. Fatal errors, overlapping triangles or zero-area triangles 187
   b. Nonfatal errors, near zero-area triangles 187

O. Indicating the cavity type for program Parmila 188

P. Controlling the relaxation order for the mesh points 189

Q. How Automesh initializes the potential 189

R. Variables calculated by Automesh 191
1. NREG and MAXPPR, number of regions and maximum points per region 191
2. XMING, XMAXG,YMING, and YMAXG, the problem boundaries 191
3. KMAX and LMAX, the number of mesh intervals along each coordinate 191
4. ITOT, the total number of mesh points in the problem 192
5. DX1, the X mesh interval in the first region 192
6. DXMIN and DYMIN, the smallest mesh intervals 192
7. XDRI,YDRI, the drive-point location for Superfish problems 192
8. XBZERO,YBZERO, physical point where B = BDES in Poisson 192
9. KBZERO,LBZERO, logical point where B = BDES in Poisson 193
10. IPERM, indicates if a Pandira problem includes real currents 193
11. NSEG, the number of boundary segments 193
12. NPBOUND, the number of points along all boundary segments 193
13. NAIR, the number of mesh points in empty space 193
14. NFE, the number of mesh points in iron 193
15. NINTER, the number of mesh points on an interface 193
16. NBND, the number of mesh points on Dirichlet boundaries 193
17. NSPL, the number of mesh points with user-defined fixed-potential 193
18. NPINP, the number of mesh points in the problem 194
19. NWMAX, the number of points for recalculating couplings 194
20. NGMAX and NGSAM, coupling array counters 194
21. NPONTS, the number of unknown relaxation points 194
22. LCYCLE, the number of relaxation iterations 194
23. CLIGHT and PI, the speed of light and the number π 194
24. EPS0 and FMU0, permittivity and permeability of free space 195
25. SPOSG, SNEGG, STOTG, currents at generation for magnet problems 195
26. XYAREA and VOLUME, area and volume in the problem geometry 195

27. TRIMIN, TRIMAX, and TRIAVG, size of mesh triangles 195

S. Automesh error messages 195

VII. Fish and CFish, Radio-Frequency Field Solvers 201

A. Starting and stopping Fish and CFish 201
1. Using the Esc key to stop Fish or CFish early 201

B. Fish and CFish files and filename conventions 202

C. Data entries for Fish and CFish 202
1. Controlling pivoting in matrix inversion routines 202
2. Setting up a frequency scan 203

D. The Superfish root finder 203
1. Root finder in program Fish 204
2. Root finder in tuning programs 205
3. Root finder in program CFish 206

E. Convergence criteria in the Fish and CFish solvers 207

F. Variables calculated by Fish and CFish 208
1. DSLOPE, the slope of the $D(k^2)$ function 208
2. FREQ, the resonant frequency 209
3. XK0 and XKSQ, the wave number k and its square k^2 209
4. DKSQ, the proposed change in k^2 at the last iteration 209
5. ERG, the Fish and CFish stored-energy integral 209
6. RESIDA, the residual of the solution matrix 210
7. RESIK, the ratio $|D(k^2)|/k^2$ at the end of the last iteration 210
8. ICYCLE, the number of the last completed iteration 210
9. Q2I, the ratio 1/2Q calculated by CFish for lossy materials 210

G. Fish and CFish exit error codes 210

VIII. Poisson and Pandira, Static Field Solvers 212

A. Starting and stopping Poisson and Pandira 212
1. Using the Esc key to stop Poisson or Pandira early 212

B. Files and filename conventions 212

C. Data entries for Poisson and Pandira 213

D. Input options for Poisson and Pandira 214
1. Permeability tables 215
2. Listing the fast-interpolation material tables 217
3. Using the stacking-factor correction 217
4. Adjusting fields for a particular value on a mesh point 219
5. Computing a series of solutions for a range of current multipliers 220
6. Some properties of permanent-magnet materials 221
7. Initialization for permanent-magnet problems 223
8. Controlling pivoting in Pandira matrix inversion routines 223
9. Mesh-point initialization in coil regions 223
10. H·dl integrals during the Poisson calculation 224
11. Electrostatic fields 224

E. Output options in Poisson and Pandira _____225
1. Defining boundaries for computing the field and gradient _____ 225
2. Harmonic analysis of the field's multipole content_____ 227
a. Mathematical theory of harmonic analysis _____ 227
b. Implementing the harmonic analysis _____ 230
3. Conformal transformations _____ 232
4. Stored energy calculations in Poisson and Pandira _____ 233

F. Convergence in the Poisson and Pandira solvers _____234

G. Quantities calculated by Poisson and Pandira _____238
1. SPOSA, SNEGA, and STOTA, currents after completing the solution _____ 238
2. XJFACT, factor by which currents have been scaled _____ 238
3. RATIO, the quantity |BZERO|/XJFACT_____ 239
4. SNOLDA and SNOLDI, old values of the summed squares of the residuals _____ 239
5. RHOAIR, optimized over-relaxation parameter _____ 239
6. ENERGY, computed stored energy_____ 239
7. RESIDA and RESIDI, the residuals for air and iron points _____ 240
8. ICYCLE, the number of the last completed iteration _____ 240
9. ETAAIR and ETAFE, rates of convergence parameters_____ 240

H. Poisson and Pandira error messages _____241

V. Autofish: Combined Automesh, Fish, and SFO

Autofish combines the calculations done by Automesh, Fish, and SFO into a single program. It saves the extra load time and disk reads necessary to run all the codes separately. Autofish writes all the same output files as the original programs. After the code finishes, you can run any of the other codes using files written by Autofish. For example, you can start either Fish or SFO and the code will read the appropriate records from the binary solution file.

To start Autofish, double-click on a file with the AF extension. Double-clicking on files with extension AM launches Automesh. To open the AF or AM file in Notepad, right click on the file and choose Edit. You can start Autofish on an AM file using the right-click menu. You can also use a command line to start the code:

Autofish *InputFile*

where the *InputFile*, which has default extension AF, contains REG, PO, and MT namelist variables that define the problem parameters and the geometry. If no *InputFile* appears on the command line (e.g., when using the Windows Start menu), the program opens the standard Open dialog window.

Autofish also reads an input file for SFO if it exists. The filename consists of the Automesh input file as the root name plus extension SEG.

VI. Automesh: the Poisson Superfish Mesh Generator

Automesh sets up the mesh data, and then generates and optimizes the triangular mesh for all the solvers including Fish, CFish, Poisson, Pandira, and the tuning programs. Automesh is the first code for both Superfish and Poisson problems. The codes run in the following order:

- Automesh
- WSFplot
- Fish, CFish, Poisson, or Pandira
- WSFplot
- SFO, SF7 (all problems); or Force (Poisson and Pandira problems only)

A. Starting program Automesh

To start Automesh, double-click on a file with the AM extension. Double-clicking on files with extension AF launches Autofish. You can start Automesh on an AF file using the right-click menu. To open the AM or AF file in Notepad, right click on the file and choose Edit. You can also use a command line (for example, in batch files) to start the code:

Automesh *InputFile*

where the *InputFile*, which has default extension AM, contains REG, PO, and MT namelist variables that define the problem parameters and the geometry. For command line entries, consider the following example:

Automesh PROB1

If file PROB1 exists in the current directory, Automesh will open it as the input file. If file PROB1 (with no extension) does not exist, but file PROB1.AM does exist, then PROB1.AM becomes the input file. (The code will also open PROB1.AF if it did not find either of the first two files first.) If no *InputFile* appears on the command line (e.g., when using the Windows Start menu), the program opens the standard Open dialog window.

1. Using the Esc key to stop Automesh early

During the mesh optimization, you can press the Esc key to stop the program. Automesh gives you three options:

F Finish on cycle xx, save the solution, then stop normally.
C Continue the calculation.
S Stop now without saving the solution (the default).

The code inserts the present cycle number for xx above. Choosing Stop returns exit error code 46. Tuning programs can prompt the operator for further instructions.

B. Automesh files and filename conventions

Automesh reads an input file of namelist variables. The default extension for the input file is AM to launch Automesh or AF to launch Autofish. The program writes the TAPE36 file, which it later rereads to generate the mesh, and the OUTAUT.TXT file, which contains an output summary. More information about these files may be found in the separate section titled Files and filename conventions.

C. Input file of namelist variables

Automesh reads namelist-like entries, but the code does not actually use the Fortran implementation of namelist. Instead, Automesh parses every line and checks for possible errors in the input parameters. This feature results in better diagnostic messages and can help save time debugging an input file. You can use either a dollar sign ($) or an ampersand (&) as the namelist delimiter. Comments can appear on any line in the input file after a semicolon (;) or exclamation mark (!).

Table VI-1. REG namelist variables.

Variable		Description
ALPHAT	S	Temperature coefficient of resistance at TEMPR (C^{-1}).
ANGLE	P	Extent of an arc for interpolating the potential .
ANGLZ	P	Initial point on arc for interpolating the potential.
ASCALE	S	Normalization factor used in SFO and SF7.
BDES	P	Adjust XJFACT so $\|B\|$ = BDES at KBZERO,LBZERO.
BETA	S	Particle velocity, used in SFO to compute k if KMETHOD = 1.
BNDMULT		Increase number of fixed boundary points by this factor.
CCDTL	S	Indicates number of gaps in a coupled-cavity drift-tube linac.
CCL	S	Indicator for a coupled-cavity linac problem.
CCLDELK	S	Increment in the coupling-slot power table in SFO.
CCLMAXK	S	Highest coupling for coupling-slot power table in SFO.
CCLMINK	S	Lowest coupling for coupling-slot power table in SFO.
CHARGE		Total charge (Coulombs/length), a synonym for CUR.
CLENGTH	S	User-supplied cavity length for defining E_0.
CONV		Conversion factor (in cm/unit) for length units.
CUR		Total current (Amps) or charge (Coulombs/length).
CURRENT	P	Total current (Amps), a synonym for CUR.
DELFR	S	Step size for a frequency scan (MHz).
DEN		Current or charge density (A/L^2 or C/L^3 areas; A/L or C/L^2 lines).
DENSITY	P	Current or charge density, a synonym for DEN.
DIAGDLL		If 1, DLL functions write diagnostics to file DiagDLL.txt.
DPHI	S	Phase length (degrees), used in SFO to compute k if KMETHOD ≠ 1.
DSLOPE	S	Initial value for the slope of the $D(k^2)$ function.
DSTOLER	S	Tolerance on DSLOPE for convergence.
DTL	S	Indicates number of cells in a drift-tube linac.
DX		X mesh size in the first region [default = 2% of X range].
DY		Y mesh size in the first region (for default, see Table VI-2).

Table VI-1. REG namelist variables. (continued)

Variable		Description
ENORM	S	Normalize to ENORM in V/m when NORM = 4.
EPSIK	S	Convergence parameter in Fish.
EPSILA	P	Convergence parameter for air and interface points.
EPSILI	P	Convergence parameter for iron points.
EPSO		Convergence parameter during mesh optimization.
EZERO	S	E_0 normalization if NORM = 0 (default = 1.0E6 V/m).
EZEROT	S	E_0T normalization if NORM = 1.
FIXEPS	P	Permittivity if MODE = –1 (XJFACT = 0).
FIXGAM	P	Reluctivity if MODE = –1 (XJFACT ≠ 0).
FREQ	S	Resonant frequency (also starting frequency) (MHz).
FREQD	S	Design frequency (MHz), used in SFO if KMETHOD = 0.
HDRIVE	S	Field H at the Superfish drive-point.
HDRIVEI	S	Imaginary part of H for a CFish drive-point.
HPHI	S	Normalize to H_ϕ in A/m on segment NRMSEG when NORM = 2.
IBOUND		Special boundary or fixed-potential indicator.
ICAL	P	Selects formula in Automesh for computing mesh-point currents.
ICCP	S	>0 for stored energy, material power loss in SFO, SF7, WSFplot.
ICYLIN		Coordinate system (0: rectangular; 1: cylindrical).
ICYSEN	P	If zero, Poisson does not print H·dl integrals.
IENERGY	P	Calculate stored energy if 1.
IHDL	P	Number of cycles in Poisson between H·dl integrals.
IOBSEG	S	Starting segment of the CCL outer boundary.
IPDIAG		Automesh and other codes print special diagnostics if nonzero.
IPERM	P	0 = problem contains real currents; 1 = permanent magnets only.
IPIVOT		Controls pivoting in matrix inversion in Fish and Pandira.
IRMAX		Cycles between optimization of the mesh SOR parameter.
IPRFQ	P	Print frequency during Poisson iterations.
IRTYPE	S	Rs option: 0: normal; 1: superconductor; 2: user RS; 3: user RHO.
ISKIP	P	Poisson cycles between recalculation of permeabilities.
ISLOT	S	If 1, estimate coupling-slot power losses in SFO.
ITRI		Triangle type: default = 0 (2 if XJFACT = 0 and ICYLIN = 1).
IVERG	P	Number of cycles between Poisson convergence tests.
KBZERO	P	Logical K coordinate where $\lvert B \rvert$ = BDES.
KMAX		Number of mesh points in the horizontal direction.
KMETHOD	S	If 1, SFO uses BETA to compute wave number, else uses DPHI.
KMIN	P	Lower K bound for computing the field and gradient.
KPROB		1 for Superfish problems, 0 for Poisson or Pandira problems.
KREG(I)		Logical coordinates for vertical line regions, I = 0 to MXLR.
KREGi		Synonyms for the first few KREG(I) for i = 1 to 8.
KTOP	P	Upper K bound for computing the field and gradient.
KTYPE	P	Symmetry indicator for harmonic analysis.
LBZERO	P	Logical L coordinate where $\lvert B \rvert$ = BDES.
LINES		Default for LINESX and LINESY.
LINESX		Default for LINEX(I).
LINESY		Default for LINEY(I).
LINEX(I)		0 = X line region can move; 1 = fixed X line region, I = 1 to MXLR.

Table VI-1. REG namelist variables. (continued)

Variable		Description
LINEXi		Synonyms for the first few LINEX(i) for i = 1 to 8.
LINEY(I)		0 = Y line region can move; 1 = fixed Y line region, I = 1 to MYLR.
LINEYi		Synonyms for the first few LINEY(i) for i = 1 to 8.
LMAX		Number of mesh points in the vertical direction.
LMIN	P	Lower L bound for computing the field and gradient.
LREG(I)		Logical coordinates for horizontal line regions, I = 0 to MYLR.
LREGi		Synonyms for the first few LREG(I) for i = 1 to 8.
LTOP	P	Upper L bound for computing the field and gradient.
MAP	P	Number of poles for conformal transformation.
MAT		Material number.
MAXCY		Maximum number of iterations or cycles.
MODE	P	Material property indicator for the problem.
MODT36		If 1, modify mesh using data from T36 file.
MSHAPE	P	Material property indicator for a particular material number.
MTID	P	Pointer to the MT namelist with the same value of MTID.
NBSLF		Left-side boundary condition.
NBSLO		Lower boundary condition.
NBSRT		Right-side boundary condition.
NBSUP		Upper boundary condition (0: Dirichlet; 1: Neumann).
NCELL	S	Number of cells for a multicell problem.
NORM	S	Field normalization option in SFO.
NOTE	P	Determines relaxation order for the mesh points.
NPOINT	**	Number of points in a region (calculated by Automesh).
NPTC	P	Number of arc points for interpolating the potential.
NREG	**	Number of regions (calculated by Automesh).
NRMSEG	S	Segment number for normalization when NORM = 2.
NPTBL	P	Selects permeability table (default is 1010 steel).
NSTEM	S	Number of stems and post couplers in a multicell tank.
NSTEP	S	Number of steps for a frequency scan.
NTERM	P	Number of coefficients in the harmonic analysis.
OMEGAM		Parameter used in SOR procedure that optimizes the mesh.
OMEGAP	P	Parameter used in Poisson SOR procedure.
PLCELL	S	Phase length per cell for a multicell problem.
RESIDR	S	Residual resistance for superconductors (Ω).
RFQ	S	Indicator for a radio-frequency quadrupole problem.
RHO	S	User-supplied material bulk resistivity in Ω-cm.
RHOR	S	Reference resistivity (Ω-cm) at temperature TEMPR.
RHOXY		Over-relaxation factor during mesh optimization.
RINT	P	Radius of the arc for interpolating the potential.
RMASS	S	Rest mass energy (MeV) or particle-type code number.
RNORM	P	Aperture radius used in the harmonic analysis.
RS	S	RF surface resistance (Ω).
RSTEM	S	Stem radius for DTL stems along boundary segments.
RX	S	Scale factor between successive X regions.
RY	S	Scale factor between successive Y regions.
SINEDRV	S	Wavelength and starting phase for a sinusoidal drive line.

Table VI-1. REG namelist variables. (continued)

Variable		Description
SLOSS	S	Coupling-slot power increase per percent coupling.
STACK	P	Default stacking or fill factor for all iron regions.
STACKING	P	Stacking or fill factor for a particular iron region.
TC	S	Critical temperature for superconductors (K).
TEMPC	S	Operating temperature for normal conductors (C).
TEMPK	S	Operating temperature for superconductors (K).
TEMPR	S	Reference temperature for normal conductors (C).
TNEGC	P	Total negative current after conformal transformation.
TPOSC	P	Total positive current after conformal transformation.
RZERO	P	Magnet bore radius for conformal transformations.
VOLTAGE	P	Voltage on a boundary for electrostatic problems (Volts).
XAZERO	P	Physical X coordinate where A = 0 for harmonic analysis.
XBZERO	P	Physical X coordinate where $\|B\|$ = BDES.
XDRI	S	X coordinate of the drive point.
XJFACT	P	0 for electric fields; or scale factor for currents.
XJFEND	P	Ending value XJFACT for XJSTEPS additional calculations.
XJSTEPS	P	Number of steps between XJFACT and XJFEND.
XJTOL	P	Tolerance on XJFACT when setting |B| = BDES.
XMAXF	P	Upper X bound for computing the field and gradient.
XMINF	P	Lower X bound for computing the field and gradient.
XNORM1	S	Starting X coordinate for NORM = 4 normalization option.
XNORM2	S	Ending X coordinate for NORM = 4 normalization option.
XREG(I)		X coordinates for line regions, I = 1 to MXLR.
XREGi		Synonyms for the first few XREG(I) for i = 1 to 8.
YAZERO	P	Physical Y coordinate where A = 0 for harmonic analysis.
YBZERO	P	Physical Y coordinate where $\|B\|$ = BDES.
YDRI	S	Y coordinate of the drive point.
YMAXF	P	Upper Y bound for computing the field and gradient.
YMINF	P	Lower Y bound for computing the field and gradient.
YNORM1	S	Starting Y coordinate for NORM = 4 normalization option.
YNORM2	S	Ending X coordinate for NORM = 4 normalization option.
YREG(I)		Y coordinates for line regions, I = 1 to MYLR.
YREGi		Synonyms for the first few YREG(i) for i = 1 to 8.
ZCTR	S	Electric center of a cell (usually where field peaks).

Problem descriptions of up to ten 80-character title lines are supported in all the codes. Enter these lines before the first REG namelist in the Automesh input file. The code ignores any blank lines or comment lines (starting with a semicolon or an exclamation mark) interspersed among the title lines. The binary solution file stores the title lines and each code prints the problem description in its respective output file. Automesh will write a warning message if the input file contains no title lines or more than ten title lines.

Automesh reads four sets of namelist variables named REG (for region) and PO (for point), MT (for material table), POA (for single-point boundary values). Namelist sections start with &REG, &PO, &MT, or &POA (or with $REG, $PO, $MT, or $POA)

and end with &END or just & (or with $END or $). It is not necessary to indicate the number of regions (NREG) or the number of points in each region (NPOINT) since these parameters are calculated in the codes. You can simulate the end of the file (for example, to exclude the last region in the file) with the keyword STOP. Automesh will not read beyond the STOP line. Synonyms for STOP are ENDFile and ENDOFFILE.

The REG namelist defines properties of a region that contains a series of connected points. The connected points, defined by consecutive PO namelist entries, usually forms a closed region. An unclosed "region" is a series of connected points on which the user defines a boundary condition. The PO points must follow their associated REG namelist. A new REG namelist starts a new region. The order of occurrence of REG and PO sections is important. If regions overlap or share a boundary, subsequent regions replace properties defined by previous regions.

The MT and POA namelist sections are all independent of one another. They may appear at any location in the input file except within another namelist of any type. We recommend placing all MT and POA namelist entries after the last REG and PO section.

1. REG namelist: Input of region data and problem parameters

Table VI-1 lists all the REG namelist variables in the input file for Automesh or Autofish. Variables that are arrays have an index "(I)" after the name. When entering an array, use the array name only and separate all elements by comma or spaces. Do not include an index in parentheses for these elements. For example "XREG = 1.0, 5.0, 10.0" but not "XREG(1) = 1.0, XREG(2) = 5.0, XREG(3) = 10.0." In the second column, a letter P means that the variable is used only for Poisson or Pandira problems. Letter S is for Superfish problems. Those with no letter are used in both Poisson and Superfish problems.

Variable KPROB must appear in the first REG namelist. It specifies the problem type: KPROB = 1 is Superfish problem, KPROB = 0 is a Poisson or Pandira problem.

NPOINT and NREG (marked ** in the second column) are obsolete REG namelist variables. The code calculates these quantities and ignores entries for them in the input file.

a. Variables that can only appear in the first REG namelist

Several variables can only appear in the first REG namelist. These include the X and Y line region entries XREGi, KREGi, LINEXi, YREGi, LREGi, and LINEYi as well as KPROB, ICYLIN, NBSUP, NBSLO, NBSRT, NBSLF, XJFACT, DX, DY, RX, RY, CONV, MODE, XDRI, and YDRI.

Other variables that are not used by Automesh are passed to the solver codes through the binary solution file. These variables can appear in any REG namelist, but we recommend that they appear in the first region. Automesh stores in the solution file only the last entry REG namelist variables that are not region specific.

b. Entry of line-region array variables

Some namelist entries, which have an index "(I)" after the name in Table VI-1, are arrays of numbers. These array variables control the locations and other properties of line regions that divide the problem area into regions of different mesh size. The "Description" column refers to the dimensions MXLR and MYLR. The array variables are XREG(1:MXLR), KREG(0:MXLR), and LINEX(1:MXLR) for the X line regions, and YREG(1:MYLR), LREG(0:MYLR), and LINEY(1:MYLR) for the Y line regions. Variables MXLR and MYLR are the total number of X and Y line regions supplied. The X and Y line regions use arrays XREG and YREG for physical coordinates, and KREG and LREG for logical coordinates. Arrays LINEX and LINEY indicate if the line regions are fixed throughout the mesh or just on the boundaries of closed regions. The points on a line region (but not on boundaries) are allowed to move during mesh optimization if the LINEX or LINEY value is zero.

In the logical coordinate arrays, the first element is either 0 or –1. If KREG(0) = 0 or LREG(0) = 0, then the rest of the KREG or LREG entries are the actual logical coordinates that correspond to supplied XREG and YREG array elements. (Note that XREG and YREG do not contain elements with index zero.) If KREG(0) = –1 or LREG(0) = –1, then the rest of the KREG or LREG entries are intervals between logical mesh columns or rows rather than actual logical coordinates. Section K.1 below contains more information about this use.

The LINEX and LINEY arrays start with element 1 rather than 0. Consider the following example, which has XMIN = 0, and X line regions at four locations. Suppose that we wish to allow the line at X = 8.5 to move during mesh optimization, but the other lines are to remain fixed. The following entries in the first REG namelist would achieve this result:

```
XREG =     4.5,  5.5,  8.5,  10.5,
KREG = 0,  20,   30    45    60,
LINEX =    1,    1,    0,    1
```

Notice that the second line has a placeholder variable as the first element. The code does not allow specification of individual array elements using parentheses, such as "XREG(2)". You must include the entire array of elements up to the maximum number of line elements you intend to specify. Section K.1 below includes a more detailed discussion of the use of line regions.

c. Default values for the mesh increments DX and DY

Automesh selects a default value for DX equal to 2% of the X range determined by Automesh by analyzing the first region. (All other regions must fit within the first region, which defines the domain for the problem geometry.) However, we recommend that users actually specify a value for DX based upon the size of objects in the problem geometry. For problems with both fine and coarse details, use line regions to control the mesh size in different parts of the geometry.

If the increment DY is not specified in the input file, Automesh chooses it value based upon the value of DX and the type of triangle used in the mesh. Table VI-2 lists variables

in the first REG namelist that affect the default value for DY. For equilateral triangles (ITRI=0 or ITRI=1), the default value is DY = DX sin(60°). For right triangles (ITRI = 2), the default is DY = DX, except for cylindrically symmetric electrostatic problems that use equilateral triangles, for which it is DY = 1.2DX. Electrostatic problems have KPROB = 0 and XJFACT = 0. (By default, Automesh uses right triangles for cylindrically symmetric electrostatic problems.)

Table VI-2. REG variables that affect the default value of DY.

KPROB	XJFACT	ICYLIN	ITRI	Default DY
0	>0	0,1	0,1	DX sin(60°)
0	>0	0,1	2	DX
0	0	0	0,1	DX sin(60°)
0	0	0,1	2	DX
0	0	1	0,1	1.2DX
1	not used	0,1	0,1	DX sin(60°)
1	not used	0,1	2	DX

d. Variables that can appear in the first and any other REG namelist

Each new region requires a new REG namelist containing variables specific to that region. These variables may include: ITRI, CUR, CURRENT, CHARGE, VOLTAGE, DEN, DENSITY, IBOUND, and MAT. For Poisson problems, the REG namelist for material numbers MAT higher than 1 can include three more parameters that define the material properties. STACKING is the stacking factor associated with the material, MTID is a pointer to the MT namelist that has the same value of MTID. The MT namelist contains definitions of the material properties. MSHAPE defines the shape of the B-H curve associated with the material.

e. Material numbers

The REG namelist variable MAT is the material number for the region enclosed by the region's boundary points. Table VI-1 lists the meaning of the material numbers for each type of problem. Every new region uses the default value MAT = 1 if the variable does not appear in the REG namelist. For more information about defining material properties for MAT ≥ 2, refer to the following sections:

- VI.3.a. Material data for rf problems
- VI.3.b. Material data for electrostatic problems
- VI.3.c. Isotropic magnetic materials
- VI.3.d. Anisotropic and permanent-magnet materials

f. The MODE setting for declaring material properties

The parameter MODE determines the method for specifying the iron permeability in magnet problems. If MODE = –2 (the default), then all materials with MAT > 1 have infinite permeability. If MODE = –1, then permeabilities have a finite fixed value. Each material can have a different value of the permeability and a different stacking factor. If

MODE = 0, then at least some materials have a variable permeability. You must define the material properties using MT namelist sections in the Automesh input file.

For electrostatic problems, for which XJFACT = 0, the default setting is MODE = –1. This setting means that all materials with MAT = 2 or higher have a finite fixed value of the permittivity ε_r. Automesh does not allow the setting MODE = –2 for electrostatic problems because it implies the unphysical case that $\varepsilon_r = 0$.

Table VI-3. Material numbers.

MAT	Radio frequency problem	Electrostatic problem	Magnetostatic problem
0	Unmeshed metal region ($\varepsilon_r = \infty$ and $\mu_r = \infty$).	Unmeshed metal region ($\varepsilon_r = \infty$).	Unmeshed region.
1	Air or vacuum ($\varepsilon_r = 1$ and $\mu_r = 1$).	Air or vacuum ($\varepsilon_r = 1$).	Air, vacuum, and coil regions ($\mu_r = 1$).
≥2	User defined ε and μ.	User defined ε.	User defined μ.

2. PO namelist: Input of boundary-point data in Automesh

Table VI-4 lists the PO namelist variables in the input file for Automesh or Autofish. Variables in the PO namelist following each REG namelist specify the boundary points for the present region. Each point corresponds to a separate PO data set. Each succeeding PO data set is a new point and includes the type of curve that connects it to the last point. Automesh can connect points with straight lines, circular arcs, elliptical arcs, or hyperbolic curves.

Table VI-4. PO namelist variables.

Variable	Description
NT	Type of curve: NT = 1 for a straight line; NT = 2 for a circular or elliptical arc centered on X0,Y0; NT = 3 for a section of an hyperbola symmetric about the slope = 1 line that passes through the point X0,Y0; NT = 4 for a counterclockwise circular arc of a given RADIUS; and NT = 5 for a clockwise circular arc of a given RADIUS. (Variable R is a not a synonym for RADIUS for NT = 4 or 5.) The default is NT = 1 if the variable does not appear in a PO namelist.
X0,Y0	Origin for the coordinates X,Y in the present PO data set. Default values are X0,Y0 = 0,0. Supplying X0,Y0 is optional for straight lines (NT = 1), for hyperbolas (NT = 3), and for two types of circular arcs (NT = 4 and 5). For NT = 2 arcs, X0,Y0 is the center of the circle or ellipse and must be supplied. For hyperbolas (NT = 3), X0,Y0 is the center of the hyperbola.
X,Y	Cartesian coordinates of a point relative to X0,Y0. For problems with cylindrical symmetry, the correspondence between coordinates R,Z and X,Y depends upon the type of problem. For Superfish problems, X,Y refers to Z,R. For Poisson and Pandira problems, X,Y refers to R,Z. [Note: The "R" in coordinate pairs R,Z or Z,R is NOT the polar coordinate R in PO data sets, the next item in this list.]
R	For NT = 1, 2, 4, or 5 R is the radial polar coordinate relative to point X0,Y0. For NT = 3, R is the minimum distance from the center of the hyperbola (X0,Y0) to the hyperbolic branch of $(X-X0)(Y-Y0) = R^2/2$ in the first quadrant.
RADIUS	A synonym for R for NT = 1, 2, or 3. For NT = 4 or 5, the radius of the circular arc drawn to point X,Y or to R,THETA.
THETA	For NT = 1, 2, 4, or 5 THETA is the angular polar coordinates of a point relative to X0,Y0, measured counterclockwise from the +X axis. For NT = 3, THETA is not used.
A,B	For NT = 2, A and B are the ellipse parameters for the semiaxes in the equation $(X-X0)^2/A^2 + (Y-Y0)^2/B^2 = 1$. Not used for other values of NT.
AOVRB	For NT = 2, AOVRB is the ratio A/B of the ellipse parameters. For circular arcs AOVRB = 1.0, the default value. Not used for other values of NT.

The PO namelist in Table VI-4 includes more variables than used in Poisson Superfish version 4 and earlier. The present code does not support the variable NEW (see below). Parameters not found in the 1987 Reference Manual are A, B, and AOVRB for specifying an elliptical arc segment.

For line segments and arcs (NT = 1, 2, 4, or 5) Automesh ignores polar coordinates R and THETA unless neither X nor Y appears in the PO namelist. If either X or Y appears without the other, the code assumes a default value of zero for the missing coordinate. For Cartesian coordinate problems (ICYLIN = 0), X0 + X and Y0 + X may have positive or negative values. In cylindrical coordinates (ICYLIN = 1), the radial coordinate must be positive (or zero), but the Z coordinate can be positive or negative. For Superfish problems, the radial coordinate would be Y0 + Y. For Poisson and Pandira, it would be X0 + X.

a. Problem boundaries, requirements for region 1

The PO namelist points for the first region must define a closed area that defines the domain for the problem geometry. This region need not be rectangular, but you should ensure that the maximum extent of the region 1boundary points in all four directions contains all other supplied regions. Automesh determines the limits Be sure that the boundary conditions are specified correctly at all four edges for the type of problem you

are solving. All other regions must fit within the problem geometry defined by region 1. Points in other regions may lie along the region 1 boundary. However, these points must traverse the boundary in the same direction used for region 1 to avoid potential meshing errors. Also, note that variable IBOUND has no effect along the problem boundaries.

b. Straight line segments

If NT = 1, then Automesh draws a straight line segment from the previous point to the point defined by entries X0,Y0 plus either X,Y or R,THETA. The default setting for X0,Y0 is the origin 0,0. If you enter the coordinates X,Y, then the next end point has coordinates X0+X, Y0+X. If you enter the coordinates R,THETA, then the coordinates are X0 + R cos(THETA), Y0 + R sin(THETA). Because NT = 1 is the default setting, you can omit it from the PO namelist. Most straight line segments include only two entries: either X and Y or R and THETA.

c. Circular and elliptical arc segments

You can use one of three settings (NT = 2, 4, or 5) to draw circular arcs. Only one setting (NT =2) can draw an elliptical arc. Circular and elliptical arc segments specified with NT = 2 require the coordinates X0,Y0 of the center of the circle or ellipse. In this case, the code checks your entries of the arc end points for consistency, which can help to diagnose errors in the problem geometry. For circular arcs entered with NT = 2, it is not necessary to enter the parameters A, B, or AOVRB. Automesh calculates the radius as the distance from X0,Y0 to the point X,Y. If you specify the end point of a circular arc using polar coordinates R and THETA, then R is also the radius of the circle. For elliptical arcs, you must supply two of the parameters A, B, and AOVRB. If both A and B appear, the code ignores any supplied value for AOVRB. Arc segments take the shorter path to the specified end point, or for 180-degree arcs, the counterclockwise direction. For a clockwise 180-degree arc, use two 90-degree arcs. Using two 90-degree arcs also eliminates any ambiguity in 180-degree counterclockwise arcs.

The entries NT = 4 or 5 define circular arcs of a given radius specified by the variable RADIUS. The setting NT = 4 corresponds to a counterclockwise arc to the point X0+X, Y0+Y; and NT = 5 indicates a clockwise arc. You can specify the end point of the arc by the set of entries X0, Y0, X, and Y or by the set X0, Y0, R, and THETA. Note that X0,Y0 is not necessarily the center of curvature of the arc. Automesh converts NT = 4 and NT = 5 parameters to the corresponding set of NT = 2 parameters.

d. Hyperbolic segments

Automesh supports only one very particular type of hyperbolic segment (with NT = 3). This feature was added to an early version of the code to model the pole tips of magnetic quadrupole lenses. The hyperbola must be of the form

$$(X - X0)(Y - Y0) = R^2/2,$$

where R is the minimum distance from the origin to the hyperbolic branch in the first quadrant. The center of this hyperbola is displaced to the coordinates X0,Y0 relative to the origin of the coordinate system. The code ignores variable THETA for hyperbolic

segments. You must specify the end point of the hyperbola using Cartesian coordinates X,Y relative to X0,Y0.

The method used to mesh hyperbolic segments is less robust than the methods used for lines and arcs. We recommend avoiding the use of line regions that intersect hyperbolic segments. For an area that includes a hyperbolic segment, we recommend that the aspect ratio of the mesh triangles lie in the range from 0.5 to 2.0.

e. Obsolete PO namelist variable NEW

Older versions of Automesh referred to a PO namelist variable called NEW. If NEW was +1 or –1, the code (supposedly) would not allow points on the logical path for a region to coincide with any previous region's path (except for the end points for NEW = –1). According to the 1987 Reference Manual, part C, chapter 3, section 3, page 22, the NEW option was a fix for certain types of logical mesh "glitches." The present code version does not support this variable because it conflicts with the extensive checking for consistency among points that Young and Billen added to Automesh.

3. MT namelist: Input of material data in Automesh

The MT namelist in the Automesh input file defines material properties for magnetostatic, electrostatic, and radio-frequency problems. MT stands for Material Table (or equivalently, Material Type). We use the terminology "material table" or "material type" to represent data that defines properties of a material. The data may consist of a single number such as permittivity or permeability, an actual table such as the B-H table for variable permeability iron, or a set of parameters such as those that define a permanent-magnet material. In fact, a single material type may contain any or all of these definitions.

Please note the distinction between a *material* and a *material table*. (In this manual, *material type* is a synonym for *material table*.) A material table contains a set of definitions that the user sets up as available to the problem. A material is an actual substance (air, iron dielectric, etc.) associated with one or more enclosed boundary regions in the problem. Each numbered material derives its properties from one of the material tables. A material table may or may not be used by any of the materials in the problem.

Program Automesh sets the maximum number of materials and material tables according to the requirements found in the input file. The upper limit depends only on available computer memory.

Table VI-5 lists all the MT namelist variables in the input file for Automesh or Autofish. Automesh writes the MT namelist data to the binary solution file. The solver programs and postprocessors read from the solution file the material data along with other information. The codes use this data to initialize material-related parameters for the problem. Automesh performs extensive checking of your material-data input and offers advice when it detects insufficient or inconsistent entries.

Table VI-5. MT namelist variables.

Variable	Description
MTID	Material table ID number. By default, each MT namelist has an ID number assigned by its location in the file. Default MTID numbers start with 2 in Superfish problems, in which a region's MAT number connects the material to the MT namelist with the same ID number. User-defined MTID numbers start with 1 in Poisson problems. A region has a material number MAT and a pointer MTID to an internal table (0, −1, etc.) or to an MT namelist (1, 2, etc.).
GAMMA	Reluctivity γ for magnet problems with fixed permeability. For magnetostatic problems (KPROB = 0 and XJFACT ≠ 0), GAMMA is the fixed reluctivity of a material when MODE = −1. When MODE = 0, programs Poisson and Pandira use GAMMA to initialize the arrays of reluctivity data. If a value for MU appears in the MT namelist, then Automesh computes GAMMA = 1/MU. GAMMA is also $\gamma_{\|}$, the reluctivity parallel to the easy axis for anisotropic materials with HCEPT = BCEPT = 0. The default value of GAMMA is FIXGAM from the first REG namelist.
EPSILON	Permittivity ε_r for rf and electrostatic problems. For electrostatic problems (KPROB = 0 and XJFACT = 0), EPSILON is a single number. For Poisson problems, the default value of EPSILON is FIXEPS from the first REG namelist. For Superfish problems (KPROB = 1), EPSILON is a single number for real ε_r and two numbers for complex ε_r. The second number is the imaginary part of ε_r and has a default value of 0.0.
MU	Permeability μ_r for rf and magnetostatic problems. For magnetostatic problems (KPROB = 0 and XJFACT ≠ 0), MU is a single number. If the MT namelist includes values for both GAMMA and MU, the value of MU takes precedence when XJFACT ≠ 0. In this case, Automesh computes GAMMA = 1/MU. MU is also the fixed permeability ($1/\gamma_{\|}$) parallel to the easy axis for anisotropic materials with HCEPT = BCEPT = 0. For Superfish problems (KPROB = 1), MU is a single number for real μ_r and two numbers for complex μ_r. The second number is the imaginary part of μ_r and has a default value of 0.0.
BGAM, BMU, BH	Table of (B,γ), (B,μ_r), or (B,H) data for materials with variable permeability. Automesh converts a (B,μ_r) or (B,H) table into a (B,γ) table. Only one of these arrays should appear in an MT namelist. If more than one appears, the last one takes precedence. Poisson and Pandira use the permeability tables when MODE = 0. To use a particular set of variable permeability data, at least one REG namelist with the same MTID must have MSHAPE = 0, which is the default when MODE = 0.

Table VI-5. MT namelist variables. (continued)

Variable	Description
HCEPT, BCEPT	H_c and B_r intercepts for an anisotropic or a permanent-magnet material. Figure VI-2 shows typical straight-line B-H curves for this type material. Permanent magnets have nonzero B_r. Both HCEPT and BCEPT must appear in an MT namelist to define an anisotropic or a permanent-magnet material. For a permanent magnet, HCEPT must be negative and BCEPT must be positive. For an anisotropic (but non-permanent) magnetic material set BCEPT = HCEPT = 0 and enter a value for GAMMA or MU to define $\gamma_{\|}$, the reluctivity parallel to the easy axis. Pandira uses the anisotropic permeability data when MODE = 0. To use a particular set of anisotropic data, at least one REG namelist with the same MTID must set MSHAPE = 1 for fixed permeability along both the easy and hard axes.
AEASY	Easy axis direction φ_E in degrees for an anisotropic or permanent-magnet material. Zero degrees is in the direction of the positive X axis. The default value is $\varphi_E = 0.0$. The easy axis has the fixed direction φ_E unless a value for PHIA appears in the MT namelist.
GAMPER	Relative reluctivity $\gamma_{\perp}$ perpendicular to the easy axis for an anisotropic or permanent-magnet material. The default value is $\gamma_{\perp} = 1.0$.
X0A,Y0A	Coordinates X_A,Y_A of the center of the circle used with the option that allows a variable direction of the easy axis. This option is selected by specifying a value for PHIA. The default center is $X_A,Y_A = 0,0$.
PHIA, MULTA	Angle φ_A in degrees and polar-angle multiplier m that give the easy axis direction $\varphi_A + m\theta$ for the option in which the easy axis is a function of position, where angle θ is the polar coordinate for a point relative to the circle centered at coordinates X_A,Y_A. PHIA has no default value. If a value for φ_A appears in the MT namelist, then the code ignores the value of AEASY and the easy axis is a function of position around the circle centered at X_A,Y_A. The default value for MULTA is m = 2, which corresponds to a dipole field pattern.

For Poisson and Pandira problems, the material data includes the MT namelist entries plus additional information in Table VI-6 from the REG namelist sections. The variable STACK in the first REG namelist has default value of 1.0 and serves as the default stacking factor s for all materials with MAT > 1. The variable STACKING is the stacking factor for a particular material supplied in the REG namelist for a region. Suppose a problem contains several regions with material number MAT = 3 and this material has s = 0.95. The setting STACKING = 0.95 on any region with MAT = 3 sets the stacking factor for all regions that use material 3. The last region with a given MAT and STACKING takes precedence. Note that multiple materials numbers MAT can point to the same MT namelist data, but each material can use a different value of the stacking factor.

Similar flexibility results from using different values of MSHAPE with the same MT namelist. For example, suppose one material with MSHAPE = 0 uses the variable permeability arrays in an MT namelist. Another material can use the fixed permeability (MSHAPE = –1) or the anisotropic data (MSHAPE = 1) from the same MT namelist.

Table VI-6. REG namelist material arrays in the binary solution file.

Variable	Description
MTID	Pointers to the MT namelist with the same value of MTID.
MSHAPE	Material property indicators for a each material number.
STACKING	Stacking or fill factors for each iron region.

All regions that have the same value of MAT have identical material properties. If an input file contains multiple regions with the same MAT, but different MTID settings, the last one of these regions sets the properties for all regions with the same MAT. An example may help to clarify this important point. Consider the input-file fragment in Figure VI-1, which contains two MT namelist sections and four regions whose material properties need defining. Regions 1 and 2 would appear to be using two different values of fixed (MSHAPE = –1) permeability, but that is not the case. Region 1 demonstrates an incorrect use of the MAT and MT namelist entries because the settings in Region 2 redefine the properties for MAT = 2. Both region 1 and region 2 will MU = 10 in this example. Regions 3 and 4 use the same value of MTID and different values of MAT. Region 3 will have the fixed permeability MU = 20 and region 4 will use the variable permeability table that starts with BGAM = 0, 0.00175.

```
&REG MAT=2, MTID=6, MSHAPE=–1&                    ; Region 1
; PO namelist entries for this region would appear here.

&REG MAT=2, MTID=7, MSHAPE=–1 &                   ; Region 2
; PO namelist entries for this region would appear here.

&REG MAT=3, MTID=6, MSHAPE=–1&                    ; Region 3
; PO namelist entries for this region would appear here.

&REG MAT=4, MTID=6, MSHAPE=0 &                    ; Region 4
; PO namelist entries for this region would appear here.

&MT MTID = 6 MU = 20
BGAM = 0.000E+00 0.00175
0.114E+04       0.00175
; lines of this material table have been omitted here.
0.280E+05       0.2518 &

&MT MTID = 7 MU = 10
BGAM = 0.000E+00 0.00275
0.134E+04       0.00275
; lines of this material table have been omitted here.
0.310E+05       0.2905 &
```

Figure VI-1. Sample input file illustrating the use of MAT.

a. Material data for rf problems

Radio-frequency problems solved by Fish or CFish can include the effects of dielectric and permeable materials. These codes use the MT namelist variables EPSILON and MU listed in Table VI-5. For the real solver Fish, EPSILON and MU are both real and they require only a single entry for each parameter. For CFish, these parameters can include both real and imaginary parts. For example, the complex permittivity of quartz would be EPSILON = 3.78, 0.0006426. If a problem contains only real values of EPSILON and

MU, then either CFish or Fish can solve for the fields. If the problem contains any complex materials, then only CFish can solve it.

Material MAT = 0 is assumed to be metal ($\varepsilon_r = \infty$ and $\mu_r = \infty$), and material MAT = 1 is assumed to be air or vacuum ($\varepsilon_r = 1$ and $\mu_r = 1$), where ε_r is the relative permittivity and μ_r is the relative permeability defined as follows:

$$\varepsilon_r = \frac{\varepsilon}{\varepsilon_0} = \frac{1}{\varepsilon_0}\left(\varepsilon_1 + i\varepsilon_2\right) \text{ and } \mu_r = \frac{\mu}{\mu_0} = \frac{1}{\mu_0}\left(\mu_1 + i\mu_2\right),$$

where the ratios $\varepsilon_2/\varepsilon_1$ and μ_2/μ_1 may be identified as the loss tangents for dielectric and magnetic materials, respectively. If you use material numbers other than 0 or 1 in a Fish or CFish problem, then you must use MT namelist sections to define the permittivity and permeability of each material. File FTEST.AM in directory FerriteCavity under Examples\RadioFrequency includes a sample problem that requires entry of material properties. This file contains MT namelist sections to define real values of permittivity and permeability for three materials:

```
&MT EPSILON = 14.5, MU = 1.5 &
&MT EPSILON = 9, MU = 1 &
&MT EPSILON = 10, MU = 1 &
```

These MT namelist sections do not include the MTID variable, so Automesh automatically numbers them 2, 3, and 4. The first line defines the properties of regions with MAT = 2, and so forth.

For CFish problems with complex materials you can supply both real and imaginary parts of ε_r and μ_r in the MT namelist. Example file COAXWG.AM in subdirectory CFish contains the complex permittivity and permeability for one material with MAT = 3:

```
&MT MTID=3
EPSILON = 0.6, 0.8
MU = 0.6, 0.8 &
```

The quantities $\varepsilon_r = \mu_r = 0.6 + 0.8i$ represent a very lossy material with magnitudes of ε_r and μ_r equal to unity ($0.6^2 + 0.8^2 = 1.0$). This is a convenient way to ensure that there will be no reflected wave from the end of the waveguide that is filled with this "artificial" material.

b. Material data for electrostatic problems

Electrostatic problems solved by Poisson or Pandira can include the effects of dielectric materials. When KPROB = 0 and XJFACT = 0, signaling an electrostatic problem, the MT namelist variable EPSILON listed in Table VI-5 is the relative permittivity ε_r. defined as follows:

$$\varepsilon_r = \frac{\varepsilon}{\varepsilon_0}.$$

Material MAT = 0 is assumed to be metal ($\varepsilon_r = \infty$) and material MAT = 1 is assumed to be air or vacuum ($\varepsilon_r = 1$). When variable MODE does not appear in Automesh input file,

the code sets MODE = –1 for electrostatic problems. Variable FIXEPS is the default value of ε_r for materials with MAT = 2 and higher. A value of EPSILON in the MT namelist section for a material takes precedence over FIXEPS.

Though seldom used, the codes can (for MODE = 0) use variable permittivity defined by user-supplied tables. The MT namelist array names BGAM, BMU, and BH refer to the magnetostatic tables of (B,γ), (B,μ_r), and (B,H). You would use the same MT namelist variables in an electrostatic problem to represent tables of (E,ε_r), (E,1/ε_r), and (E,D).

File 2EPSILON.AM in the Dielectrics directory under Examples\Electrostatic includes a sample problem that contains MT namelist sections to define the permittivity for two materials:

```
&MT MTID=3
EPSILON=2

&MT MTID=4
EPSILON=9
```

Note that MODE = –2 is not allowed for electrostatic problems. To simulate a material with infinite permittivity, set FIXEPS to a very large number. If MODE = 0, the code expects to find user-supplied tables of material properties.

c. Isotropic magnetic materials

Poisson and Pandira can use three different types of isotropic magnetic materials depending upon the value of MODE specified in the first REG namelist section of the Automesh input file. Table VI-7 lists the options. When MODE is –2 (the default), the codes ignore any data supplied in the MT namelist sections and assume infinite permeability for magnetic materials. When MODE is –1, then FIXGAM is the default value of the reluctivity γ for materials with MAT = 2 and higher. A value of GAMMA (or MU) supplied in the MT namelist section for a material takes precedence over the FIXGAM setting. When MODE is zero, then the codes use variable permeability defined by either the internal tables or user-supplied tables. The anisotropic or permanent-magnet materials discussed in the next section also require the MODE = 0 setting.

Table VI-7. The three possible values for MODE.

MODE	Description
–2	All materials have infinite permeability.
–1	All materials have finite, but fixed, permeability.
0	Some materials have variable permeability.

The REG namelist variable MSHAPE declares one of three possible shapes for a material's B-H curve according to the list in Table VI-8. The default value of MSHAPE is equal to MODE. For example, if MODE = 0, then MSHAPE = 0 unless the REG namelist sets it to a different value. Thus, it is sufficient to just set MODE = 0 to use tables of variable permeability.

When MODE is –2 or –1, the value of MSHAPE is irrelevant: all materials have fixed permeability, either infinite or finite. When MODE is zero, a problem can use a mixture

of material types. Materials with MSHAPE = 0 have variable permeability and use either an internal table or a user-defined table. The setting MSHAPE = –1 declares a fixed-permeability material. The setting MSHAPE = 1 declares an anisotropic or permanent-magnet material with fixed (but different) values of the permeability along the easy axis and the hard axis. Permanent magnets have MSHAPE = 1 and nonzero entries for BCEPT and HCEPT. For anisotropic materials (but not permanent magnets), the permeability along the easy axis is variable if MSHAPE = 2. Problems with anisotropic or permanent-magnet materials can only be solved by Pandira.

The REG namelist variable MTID connects the material number MAT to the appropriate material table ID number MTID. If there are multiple regions that use the same material, the setting for MSHAPE need only appear in one REG namelist for that material. For multiple settings, the last one takes precedence. The variable MTID has the default value of zero, which connects the material to the internal table for 1010 steel. If a setting for MSHAPE or MTID appears in more than one REG namelist for the same material, the last one takes precedence.

You can supply an unlimited number of your own tables of permeability data in the MT namelist sections of the Automesh input file. Each table can contain information about an isotropic material used in Poisson and Pandira, and about an anisotropic material used only in Pandira. The default table for material numbers 2 and higher is the internal material table with MTID = 0 for 1010 steel.

Table VI-8. Values of MSHAPE when MODE = 0.

MSHAPE	Description
–1	Isotropic material with finite, but fixed, permeability.
0	Isotropic material with variable permeability.
1	Anisotropic material with straight-line BH curves.
2	Anisotropic material with variable permeability for the easy axis.

Three different formats are available for user-defined tables. The variable BGAM supplies pairs of (B,γ) points, BMU supplies (B,μ_r) data, and BH supplies (B,H) data. The material table can contain up to 200 data pairs. Figure VI-1 shows an example MT namelist for a (B,γ) material table. The entries need not use one line per data pair, though this format is probably the most convenient. The code converts entries of (B,μ_r) or (B,H) data into the (B,γ) format. The magnetic induction B is in Gauss, and the magnetizing force H is in Oersteds. The reluctivity γ and the relative permeability μ_r are dimensionless and are related as follows:

$$\mu_r = \frac{\mu}{\mu_0} \text{ and } \gamma = \frac{1}{\mu_r}.$$

```
&REG MAT=2, MTID=3, MSHAPE= 0, STACKING=1 &
; PO namelist entries for this region would appear here.

&MT MTID=3,
BGAM=0.00000E+00 0.0017513135
0.11420E+04     0.0017513135
0.29530E+04     0.0010159504
0.51140E+04     0.0007821666
; About 40 lines of this material table have been omitted here.
0.25885E+05     0.1888580000
0.26854E+05     0.2186950000
0.28019E+05     0.2517840000 &
end
```

Figure VI-1. A sample MT namelist with a (B,γ) material table.
Several lines have been omitted for brevity. The REG namelist for material 2, which will use this data, includes the settings MTID = 3, MSHAPE=0, and STACKING = 1.

The first point in the table is the only valid location of an entry with B = 0. For (B,H) data, Automesh extrapolates γ at B = 0 from the first two nonzero entries. The code also extrapolates γ at B = 0 if you omit this point from (B,γ) or (B,μ_r) data. Automesh checks for bad material tables and generates an error if a table

- is too long,
- contains negative entries,
- has an odd number of entries, or
- does not have monotonically increasing B.

d. Anisotropic and permanent-magnet materials

Pandira can solve problems that include anisotropic or permanent-magnet materials with straight-line B-H curves like those shown in Figure VI-2. The magnetic induction B is in Gauss, and the magnetizing force H is in Oersteds. The first REG namelist must set MODE = 0 to use these materials. Table VI-9 lists variables in the MT namelist related to anisotropic and permanent-magnet materials. The REG namelist that uses the material connected to an MT namelist for an anisotropic material must set MSHAPE = 1 or MSHAPE = 2. The setting MSHAPE = 1 declares an anisotropic or permanent-magnet material with fixed (but different) values of the permeability along the easy axis and the hard axis. Permanent magnets have MSHAPE = 1 and nonzero entries for BCEPT and HCEPT. For anisotropic materials (but not permanent magnets), the permeability along the easy axis is variable if MSHAPE = 2.

If there are multiple regions that use the same material, then at least one of the REG namelist sections for that material must set MSHAPE = 1 or MSHAPE = 2. If a setting for MSHAPE appears in more than one REG namelist for the same material, the last one takes precedence.

Table VI-9. MT namelist variables for anisotropic materials.

Variables	Symbol	Default	Description
HCEPT	H_c	none	Negative H intercept.
BCEPT	B_r	none	Positive B intercept.
AEASY	φ_E	0.0	Angle in degrees of the easy axis to the horizontal.
GAMPER	$\gamma_\perp$	1.0	Relative reluctivity perpendicular to the easy axis.
GAMMA	$\gamma_\parallel$	FIXGAM	Reluctivity parallel to the easy axis if $B_r = H_c = 0$.
X0A,Y0A	X_A, Y_A	0.0,0.0	Center of circular arc for the easy axis.
PHIA	φ_A	none	Easy axis direction at polar angle $\theta = 0$.
MULTA	m	2.0	Multiplier for the polar angle θ.

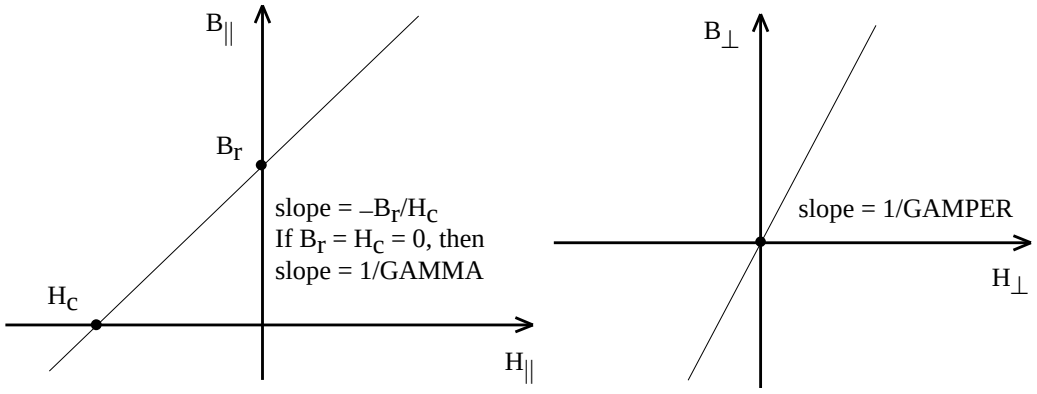

Figure VI-2. B-H relationships for an anisotropic permanent-magnet material. The left figure shows the coercive force H_c (in Oersteds) and the remanent field B_r (in Gauss) for fields parallel to the easy axis. For a non-permanent (but anisotropic) magnetic material, the curve goes through the origin like it does for the direction perpendicular to the easy axis shown at right.

```
&REG MAT=2, MTID=6, MSHAPE= 1 &
; PO namelist entries for this region would appear here.

&REG MAT=3, MTID=7, MSHAPE= 1 &
; PO namelist entries for this region would appear here.

&MT MTID = 6
AEASY = 90, GAMPER = 1.0
HCEPT = -10000, BCEPT = 10000 &

&MT MTID = 7
AEASY = -90, GAMPER = 1.0
HCEPT = -10000, BCEPT = 10000 &
```

Figure VI-3. Sample input file defining permanent magnet materials

Figure VI-3 shows a sample implementation that defines two permanent-magnet materials. The example sets $\varphi_E = +90$ for material MAT = 2, which is connected to the MT namelist with MTID = 6. It sets $\varphi_E = -90$ for material MAT = 3, which is connected to the MT namelist with MTID = 7. For both materials $\gamma_\perp = 1$,

$H_c = -10000$, and $B_r = +10000$. The code computes the value $\gamma_{||} = -H_c/B_r = 1$. No other anisotropic material variables appear in this MT namelist. Thus, the code will ignore X_A, Y_A, φ_A, and m because no value for φ_A (PHIA) appeared to make the easy axis a function of position as discussed in the next section.

e. Anisotropic material with variable easy axis direction

Pandira includes an option that allows the easy axis direction to be a function of angle around the center of a circle. For a detailed description refer to the section titled "Easy axis on an off-center circle" in Chapter XX. When a value of PHIA appears in the MT namelist, then the easy axis direction varies with position and is given by

$$\varphi_E = \varphi_A + m\theta,$$

where φ_A is the entry for PHIA, θ is the polar angle to the centroid of a mesh triangle with respect to the offset origin (X_A, Y_A), and m is the multiplier MULTA. For a mesh triangle whose centroid is at location (x, y), the polar angle is

$$\theta = \tan^{-1}\left(\frac{y - Y_A}{x - X_A}\right).$$

There are no natural materials for which the easy axis direction changes with location in the material. However, wedge-shaped slabs of permanent magnet material can be assembled to approximate this behavior. Example files in the Examples\Magnetostatic subdirectories use this feature. File PMDVAR in the PMDipoles directory has multiplier m = 2, which corresponds to a dipole field pattern. File PMQUAD2 in the PMQuads directory uses the setting m = 3 to produce a quadrupole field pattern.

f. Permanent magnets and real currents

A Pandira problem that includes permanent-magnet materials may or may not also include real currents. Variable IPERM has one of two possible values listed in Table VI-10. The IPERM setting determines how Pandira will initialize the potential. If IPERM = 0, the problem must contain at least one current region, but it may also include permanent-magnet materials. The current region may be a single point, a line of current, or an area. If IPERM = 1, then user-supplied current regions are ignored and the problem must contain at least one permanent-magnet material.

Users of older versions of Poisson Superfish may recall when it was necessary to include a current region in the input data in order to make Pandira initialize the potential correctly when IPERM = 1. This "dummy" current region is no longer required. If the problem contains any materials with the settings MSHAPE = 1 or MSHAPE = 2, then Pandira with initialize the potential correctly for a permanent-magnet problem, unless you intentionally set IPERM = 0.

Table VI-10. The meaning of IPERM.

IPERM	Description
0	The problem contains real currents.
1	The problem does not contain real currents.

Automesh will set the value of IPERM correctly for most problems by checking whether the input file contains any current regions and any permanent-magnet materials. The code will set IPERM = 1, if a problem contains permanent-magnet material and no current regions. For problems with current regions, Automesh sets IPERM = 0. If these settings are not what you want, then include a setting for IPERM in the first REG namelist section of the Automesh input file.

4. POA namelist: Input of single-point boundary values

Table VI-11 lists the POA namelist variables in the input file for Automesh. Variables in the POA namelist define a single point boundary value at location XA, YA. The boundary value ASET corresponds to the vector potential A_z (or A_ϕ) for magnet problems, voltage V for electrostatic problems, or the magnetic field H_z (or H_ϕ) for rf problems. Each point corresponds to a separate POA data set, which is independent of all the other POA points. Automesh does not connect them with a curve of any type, although a region in the problem geometry may contain the same points along one or more segments.

This feature restores a capability previously found in older version of Poisson and Pandira. In version 4 and earlier, users could provide a list of logical coordinates and potential values directly to these solver programs. The present implementation defines the points using their physical coordinates instead of their logical coordinates, and the entries are part of the Automesh input file.

Table VI-11. POA namelist variables.

Variable	Description
XA,YA	Coordinates of a point on which to define a boundary value. Both entries are required on each POA namelist.
ASET	Value of the potential on point XA,YA. This entry is required on each POA namelist. The value corresponds to the vector potential A_z (or A_ϕ) for magnet problems, voltage V for electrostatic problems, or the magnetic field H_z (or H_ϕ) for rf problems. For complex problems solved by CFish, ASET is the real part of H_z (or H_ϕ).
ASETI	Optional entry equal to the imaginary part of the magnetic field H_z (or H_ϕ) on point XA,YA. This entry applies only to complex problems solved by CFish. The default value is zero.

The main use of the POA namelist is to define a current filament or a varying potential along one or more boundary segments in the problem geometry. For an example of this type problem, see the POAsample subdirectory under Examples\Magnetostatic.

You must exercise caution when using the POA namelist feature to be sure the code does what you intend. Automesh processes the POA namelist entries last, after assigning intermediate points along all other segments in the geometry. The code then creates a

single-point region for each POA namelist, using the IBOUND = –1 setting to define fixed potential ASET on the point XA, YA. For example, the entry

```
&POA XA=8.66025404, YA=5.00000000, ASET=647.0588 &
```

is equivalent to the following one-point region:

```
&REG IBOUND=–1, CUR=647.0588 &
&PO X=8.66025404, Y=5.00000000 &
```

However, the REG and PO method will not work satisfactorily when setting the potential to different values on all points along a boundary segment. The reason is that Automesh first reads all the user-supplied PO points in the input file. When filling in points along segments, Automesh will assign the logical coordinates K,L of any stored point at the same physical location X,Y. Trouble occurs because an individual one-point region always chooses the closest available unused mesh point K,L, which often is not the correct choice when tracing a path of logically connected points. By using POA namelist entries instead, the code ignores these individual points on the first pass, and only assigns their logical coordinates after processing all the other regions.

For a constant potential along a boundary, do not use POA namelist entries. Instead, use REG namelist variables IBOUND and CUR (or one of its synonyms CURRENT, CHARGE, HDRIVE, or VOLTAGE) to indicate the boundary condition.

D. Sample Automesh input files

The Examples subdirectories contain many sample input files for running Poisson Superfish. These examples include almost all of the sample problems from the 1987 Reference Manual and User's Guide.We include here a few short examples to illustrate the structure of the input file. In these examples, namelist variables start with &REG or &PO and end with &. These examples do not include any MT namelist sections to define material properties.

Four chapters of this manual contain more detailed discussions of the examples. Chapter XVI discusses rf cavity problems solved by the Fish, CFish, or Autofish programs. Chapters XVII and XVIII are on Poisson and Pandira static magnetic field and electrostatic field examples, respectively. Examples problems for the tuning programs CCLfish, CDTfish, DTLfish, ELLfish, MDTfish, RFQfish, and SCCfish are in Chapter XIX.

1. Superfish pillbox cavity

Figure VI-4 shows a sample input file for a simple pillbox cavity for Superfish. The section on example problems includes more details about similar input files for pillbox cavities.

```
2.4-GHz TM010 Short (Kmax < Lmax) Pillbox Cavity
&REG KPROB = 1,DX = .3,FREQ = 2400
XDRI = 0.,YDRI = 4.7 &
&PO X = 0.,Y = 0. &
&PO X = 0.,Y = 4.7 &
&PO X = 3.0,Y = 4.7 &
&PO X = 3.0,Y = 0. &
&PO X = 0.,Y = 0. &
```

Figure VI-4. Automesh input file for a pillbox cavity.

2. Example with an elliptical segment

Figure VI-5 shows the use of an elliptical segment in a superconducting accelerating cavity.

```
Accelerating cavity with an elliptical segment
&REG KPROB = 1,FREQ = 700.0,DX = .1,
NBSUP = 1,NBSLO = 0,NBSRT = 1,NBSLF = 0,
ICYLIN = 1,CCL = 1
IRTYPE = 1,RMASS = –2.0,TEMPK = 2.0,TC = 9.2,RESIDR = 0.10E–07,
XDRI = 5.1350,YDRI = 18.600 &

&PO X = 0.00,Y = 0.00 &
&PO X = 0.00,Y = 5.00 &
&PO NT = 2,X0 = 0,Y0 = 8,A = 1.5,B = 3,X = 1.4918,Y = –0.3127 &
&PO X = 1.903,Y = 15.533 &
&PO NT = 2,X0 = 5.1350,Y0 = 15.36350,X = 0.00,Y = 3.23650 &
&PO X = 5.1350,Y = 0.00 &
&PO X = 0.00,Y = 0.00 &
```

Figure VI-5 Automesh input file for a cavity with an elliptical segment.

3. Poisson and Pandira H-shaped magnet

Figure VI-6 is a sample input file for a simple H-shaped magnet problem solved by Poisson or Pandira. This example file has multiple regions. The first region is the outline of the problem geometry. The second region defines the boundaries of the iron, and the third region sets up the current carrying coil. The section on example problems includes more details about similar input files for H-shaped magnets.

```
H-Shaped Magnet (no special options) [Reference Manual B.2.1]
&REG KPROB = 0,DX = .3, MODE = –1 &
&PO X = 0.,Y = 0. &
&PO X = 22.,Y = 0. &
&PO X = 22.,Y = 13. &
&PO X = 0.,Y = 13. &
&PO X = 0.,Y = 0. &

&REG MAT = 2 &
&PO X = 0., Y = 2. &
&PO X = 5.1,Y = 2. &
&PO X = 5.5,Y = 2.4 &
&PO X = 5.5,Y = 6. &
&PO X = 15.,Y = 6. &
&PO X = 15.,Y = 0. &
&PO X = 22.,Y = 0. &
&PO X = 22.,Y = 13. &
&PO X = 0.,Y = 13. &
&PO X = 0.,Y = 2. &

&REG MAT = 1,CURRENT = -25455.791 &
&PO X = 6.,Y = 0. &
&PO X = 14.5,Y = 0. &
&PO X = 14.5,Y = 5.5 &
&PO X = 6.0,Y = 5.5 &
&PO X = 6.,Y = 0. &
```

Figure VI-6. Automesh input file for an H-shaped magnet.

E. Solving for rf and electrostatic fields in the same geometry

For problems that have both static and radio-frequency fields, you can use the static solvers (Poisson or Pandira) to get the electrostatic or magnetostatic fields, and the rf solvers (Fish, CFish, or Autofish) to get the rf fields. There will be several differences in the input file even though the problems have the same geometry. We recommend making two copies of the input file. The input file for the static-field problem has KPROB = 0 in the first REG namelist; the rf problem has KPROB = 1. You must start each problem with Automesh, then run the appropriate solver program and postprocessors.

For rectangular coordinates (ICYLIN = 0), all the PO namelist entries can be the same. However, if the problem has cylindrical symmetry (ICYLIN = 1), you will need to interchange X and Y for one of the files. Radio-frequency problems use the horizontal (X) axis for the axis of symmetry, but static problems use the vertical (Y) axis. Here is a list of entries to check when setting up your input files:

- Use KPROB = 0 for static problems, KPROB = 1 for rf problems
- If ICYLIN = 1, interchange X and Y in all PO entries. You may need to make other changes on circular arcs that use the THETA variable.
- If ICYLIN = 1, also interchange any REG namelist variables that refer to X and Y line regions and their corresponding K and L regions.

- Check the settings NBSUP, NBSLO, NBSRT, NBSLF that specify boundary conditions on the edges of the problem geometry.
- Be sure the rf problem has a drive point XDRI,YDRI and starting frequency FREQ.
- Enter appropriate settings for MODE, FIXEPS, and XJFACT for the static problem.
- Use IBOUND = –1 to enter fixed potentials on metal electrodes for the static problem.
- When defining material properties, use the appropriate MT namelist sections in each code. If you use CFish for the rf problem, enter complex numbers for the permittivity and permeability. Use real numbers for the material properties in the other codes.
- For the rf problem, include settings needed in SFO, such as surface-resistance options, particle mass, and field normalization parameters.

F. Entering coordinate data in units other than centimeters

If CONV has a value other than unity it changes the scale for coordinate dimensions in the PO namelist sections and for variables that include any length dimensions. The value of CONV is the number of centimeters per unit. For example, to enter data in inches, set CONV = 2.54. The CONV value also affects REG namelist variable DEN, which has units of Amps/length^2 for magnet problems or Coulombs/length^3 for electrostatic problems. (For lines or curves of unclosed regions, the units are Amps/length or Coulombs/length^2.) All codes work internally with coordinates in cm, but you can enter the data in any units. The codes scale coordinates back to original units on output. Dimensions of calculated quantities such as electric and magnetic fields are not influenced by CONV. Thus, an electric field will be reported in MV/m whether you use cm, mm, inches, or any other unit for the input coordinates.

If you use the CONV option, the codes write a reminder in their respective output files and on the screen about using the appropriate units when later entering variables in postprocessors. Headings of tables and some other output lines will use unit abbreviations from Table VI-13 if you set CONV to one of the recognized standard units.

Table VI-12 lists variables that have dimensions of length, area, or volume and use the units specified by CONV in the Automesh input file. If the problem type is blank in Table VI-12, then the variable appears in both Superfish and Poisson problems.

Table VI-12. Variables affected by the choice for CONV.

Variable	Problem type	Description
XAZERO	Poisson	Physical X coordinate where A = 0 for harmonic analysis.
YAZERO	Poisson	Physical Y coordinate where A = 0 for harmonic analysis.
XBZERO	Poisson	Physical X coordinate where $\|B\|$ = BDES.
YBZERO	Poisson	Physical Y coordinate where $\|B\|$ = BDES.
XMINF	Poisson	Lower X bound for computing the field and gradient.
XMAXF	Poisson	Upper X bound for computing the field and gradient.
YMINF	Poisson	Lower Y bound for computing the field and gradient.
YMAXF	Poisson	Upper Y bound for computing the field and gradient.
XDRI	Superfish	X coordinate of the drive point.
YDRI	Superfish	Y coordinate of the drive point.
CLENGTH	Superfish	User-supplied cavity length for defining E_0.
ZLONG	Superfish	E_0 integration length, and cavity length if CLENGTH = 0.
RSTEM	Superfish	Stem radius for stems along boundary segments.
XMING		Lower X bound of the problem geometry.
XMAXG		Upper X bound of the problem geometry.
YMING		Lower Y bound of the problem geometry.
YMAXG		Upper Y bound of the problem geometry.
XORG	Poisson	X coordinate of arc center for harmonic analysis.
YORG	Poisson	Y coordinate of arc center for harmonic analysis.
ZCTR	Superfish	Electrical center of a cell (usually where field peaks).
XNORM1	Superfish	Starting X coordinate for NORM = 4 normalization option.
YNORM1	Superfish	Starting Y coordinate for NORM = 4 normalization option.
RINT	Poisson	Radius of the arc for interpolating the potential.
XNORM2	Superfish	Ending X coordinate for NORM = 4 normalization option.
YNORM2	Superfish	Ending X coordinate for NORM = 4 normalization option.
RNORM	Poisson	Aperture radius used in the harmonic analysis.
XYAREA		Total cross sectional area in the problem.
VOLUME		Cavity volume for cylindrically symmetric problems.
SAREA		Total surface area used for power calculations.
DXMIN		Minimum X mesh interval.
DYMIN		Minimum Y mesh interval.
DX1		X mesh spacing in the first region.
TRIMIN		Area of the smallest (positive-area) mesh triangle.
TRIMAX		Area of the largest mesh triangle.
TRIAVG		Average area of the mesh triangles.

Table VI-13. Length units.

CONV	Units
1.00	cm
0.10	mm
100.00	m
2.54	in
30.48	ft

G. Specifying boundary conditions and fixed potentials

When setting up a problem, you must supply the boundary conditions at the edges of the problem geometry or accept the default values assigned by Automesh. You also can require the solution to satisfy Neumann or Dirichlet boundary conditions along other boundary segments in the geometry. Variable IBOUND in REG namelist sections serves two purposes. A positive or zero value for IBOUND specifies the boundary condition along a path that is not part of the outer rectangular edge of the problem geometry. The setting IBOUND = –1 declares a region of fixed-potential points.

1. Boundary conditions at the edges of the problem geometry

Variables NBSUP, NBSLO, NBSRT, and NBSLF describe the boundary conditions along the edges of the problem geometry. A value of 0 indicates a Dirichlet boundary and 1 indicates a Neumann boundary. In the WSFplot display, contour lines will tend to be parallel to Dirichlet boundaries and they will intersect Neumann boundaries perpendicular to the boundary. The default boundary conditions are different for Superfish and Poisson problems. Automesh uses the default settings in Table VI-14. Enter these values in the first REG namelist section of the Automesh input file. They cannot be changed for a problem after running Automesh.

Table VI-14. Default boundary conditions at the edges of the geometry.

Variable	Superfish	Poisson
NBSUP	1	0
NBSLO	0	1
NBSRT	1	0
NBSLF	1	0

2. Using IBOUND to indicate boundary conditions

Table VI-15 gives the default boundary conditions for region boundaries. IBOUND = 1 indicates a Neumann boundary, and IBOUND = 0 indicates a Dirichlet boundary. The default value depends on the type of problem and whether the region is the first one in the file. Automesh initializes IBOUND to the value –2 and then applies the default value from this table unless the input file includes a valid entry. The order in which regions appear in the Automesh input file affects the final boundary conditions. This order is especially important for overlapping and adjacent regions. You cannot use IBOUND to set the boundary conditions on the problem boundaries at the edges of the geometry. For examples that use IBOUND see (among others) the sample files in the Quadrupoles subdirectory under Examples\Magnetostatic.

Table VI-15. Default boundary conditions for region boundaries.

Default values in:	Poisson/Pandira	Superfish
First region	IBOUND = 0	IBOUND = 1
Other regions	IBOUND = 1	IBOUND = 1

3. Using IBOUND to indicate fixed-potential values

The setting IBOUND = –1 indicates a fixed potential equal to the value of REG namelist variable CUR (or synonyms CURRENT, CHARGE, HDRIVE, or VOLTAGE). A common use of this technique is to set the voltage on a conductor for electrostatic problems. Several LANL\Examples\Electrostatic subdirectories contain files that illustrate setting the voltage on a conductor.

Another use of IBOUND specifies a driving term for the field H in CFish problems (see the sample files in directory LANL\Examples\RadioFrequency\CFish).

Though less commonly used, you also can specify a fixed vector potential for magnet problems. In this case, CUR actually refers to the value for A. The sample problem in directory LANL\Examples\Magnetostatic\Shielding illustrates this approach.

4. Using the POA namelist to indicate a single-point fixed-potential

The POA namelist defines a fixed potential value on a single point. You may use multiple POA points to set a varying potential along a previously defined boundary segment. The POAsample subdirectory under Examples\Magnetostatic shows an example of this type problem.

5. Simulating a floating electrode

An electrostatic problem may contain a "floating" electrode, that is, one whose potential is unknown at the start of the problem because it is not connected to any conductor at a known potential. You cannot use IBOUND = –1 to indicate this constant, indeterminate potential. However, you can simulate the floating electrode using a high-dielectric material. Instead of a metal region (material number MAT = 0), use MAT = 2 (or a higher number) for the electrode region and enter a very large, but finite, dielectric constant for the material. This material will exclude nearly all the electric field and very nearly satisfy the Neumann boundary condition at the interface. This method has one minor disadvantage: the code must compute the fields at all points inside the dielectric region.

H. Specifying rectangular or cylindrical coordinate systems

Variable ICYLIN determines the type of coordinate system. For rectangular coordinates (ICYLIN = 0), the abscissa is the X axis and the ordinate is the Y axis. Superfish problems and Poisson problems differ in the way they implement cylindrical symmetry (ICYLIN = 1). Superfish problems use the horizontal (X) axis for the axis of symmetry, while in Poisson and Pandira problems the vertical (Y) axis is the axis of symmetry.

The default coordinate system for Superfish problems has cylindrical coordinates. The default coordinate system for Poisson and Pandira problems has rectangular coordinates. Enter the value of ICYLIN in the REG namelist section of the Automesh input file. You cannot change ICYLIN for a problem after running Automesh.

I. Complementary solutions for rf cavities

The discussion under Maxwell's equations in cylindrical coordinates describes how the codes can be used to solve for both TM and TE modes in an rf cavity. The Superfish code

and postprocessors were written primarily for problems with cylindrical symmetry. In cylindrical coordinates (ICYLIN = 1), the codes Fish and CFish find solutions for the transverse magnetic or TM modes. The solution array is H_ϕ, the only nonzero component of the rf magnetic field. The only other nonzero field components are E_r and E_z. For problems in Cartesian coordinates (ICYLIN = 0), the solution array corresponds to the component H_z, and the other nonzero field components are E_x and E_y. These solutions for a waveguide cross section are transverse electric or TE modes at cutoff.

Superfish also can solve for TE modes in cylindrical coordinates or for TM modes in Cartesian coordinates if the user simply interchanges the roles of **E** and **H** and applies the appropriate boundary conditions. We call this the complementary solution. Consider the two Maxwell's equations valid in charge-free regions for the curl of **E** and the curl of **H**:

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t},$$

and

$$\nabla \times \mathbf{H} = \varepsilon \frac{\partial \mathbf{E}}{\partial t}.$$

The form of the equations is identical except for an algebraic sign. One can show that the equations remain unchanged after making the following substitutions:

$$\mathbf{E} \rightarrow -c\mathbf{B}^* = -c\mu\mathbf{H}^* = -Z\mathbf{H}^*,$$

and

$$\mathbf{H} \rightarrow \frac{\mathbf{E}^*}{c\mu} = \frac{\mathbf{E}^*}{Z},$$

where **E** and **H** are the electric and magnetic fields computed in Fish or CFish, the starred symbols are the complementary fields in which we are interested, c is the velocity of light in the medium, and Z is the medium's impedance defined by the equation

$$Z = \sqrt{\frac{\mu}{\varepsilon}}.$$

In the absence of any materials, $\mu = \mu_0$, $\varepsilon = \varepsilon_0$, and Z becomes the impedance of free space Z_0, which is equal to 376.7303 Ω. In cylindrical coordinates the code will report values of E_r and E_z in MV/m and H_ϕ in A/m. To convert these field components to the corresponding complementary field components, we have

$$E_\phi^*[\mathrm{MV/m}] = (Z \times 10^{-6}) H_\phi[\mathrm{A/m}],$$

$$H_r^*[\mathrm{A/m}] = \left(-\frac{10^6}{Z}\right) E_r[\mathrm{MV/m}],$$

and $$H_z^*[\mathrm{A/m}] = \left(-\frac{10^6}{Z}\right) E_z[\mathrm{MV/m}].$$

Similarly, in Cartesian coordinates the code will report values of E_x and E_y in MV/m and H_z in A/m. To convert these field components to the corresponding complementary field components, we have

$$E_z^*[\mathrm{MV/m}] = \left(Z \times 10^{-6}\right) H_z[\mathrm{A/m}],$$

$$H_x^*[\mathrm{A/m}] = \left(-\frac{10^6}{Z}\right) E_x[\mathrm{MV/m}],$$

and $$H_y^*[\mathrm{A/m}] = \left(-\frac{10^6}{Z}\right) E_y[\mathrm{MV/m}].$$

Unfortunately, the user must perform the task of reinterpreting the complementary-solution results. The rf solvers Fish and CFish and the postprocessors SFO, SF7, and WSFplot all assume field components for TM modes if ICYLIN = 1 or for TE modes if ICYLIN = 0. Thus, program SFO will not calculate correctly the power along metal surfaces for the complementary solution. To compute the cavity Q, use program SF7 to find the fields along metal surfaces, interchanging the roles of E and H as discussed above. After interpolating the fields along each line segment, write a short program or use a spreadsheet to integrate the power P. (Provide SF7 with the physical coordinates of the end points of line segments and arcs plus a number of steps to take along the each such path.) Then $Q = \omega U/P$, where U is the stored energy, which is calculated correctly by SFO.

In WSFplot, lines of constant H_z or rH_ϕ need to be reinterpreted as lines of constant E_z or rE_ϕ. When using the mouse in WSFplot to probe for magnitudes of fields, remember to interpret **E** components as –**H** components, and vice versa.

Generally speaking, the only difference in setting up a problem for the complementary solution is in the boundary conditions and material properties. When specifying material properties, be sure to reverse the roles of permittivity ε and permeability μ. The SPLITTER.AM problem is an example of a complementary field solution in which the fields **E** and –**H** are interchanged. In the SPLITTER problem, the setting NBSLO = 1 in the first REG namelist makes the lines of constant electric field E_z perpendicular to the lower boundary. All the other edges are metal surfaces with settings NBSUP = 0, NBSLF = 0, or NBSRT = 0. If your complementary-solution problem has metal surfaces in the interior of the mesh use the setting IBOUND = 0 for the interior boundary segments. Remember that NBSLO, NBSUP, NBSLF, and NBSRT take precedence over the IBOUND setting at the edges of the geometry, so both types of boundary-condition entries will usually be needed.

J. SF.INI settings for Automesh

Automesh determines the number of regions and fixed boundary points needed from the input file. It also estimates the length of internal arrays necessary to store information about interpolated points along boundaries. The required length increases with the number of regions, the number of points per region, the number of line regions, and the fineness of the mesh. For some unusual geometries, a problem may exhaust these resources. The code will ask you to double the value of InternalArrays in file SF.INI if a problem exhausts the available resources.

K. Mesh optimization considerations

The mesh size in your problem geometry affects the size of the solution file, the amount of memory needed to run the codes, the time needed to calculate the solution, and the accuracy with which you can interpolate the fields. You will find that it is almost always worth the few minutes spent tailoring the mesh to your problem geometry, especially if you will run many similar cases.

Problems with roughly equal-sized objects or features throughout the geometry can use constant mesh-size increments DX and DY. If the increment DY is not specified in the input file, Automesh chooses it value based upon the value of DX and the type of triangle used in the mesh (see Table VI-2 and Table VI-18). If your problem has fine details in one area compared to the rest of the geometry, then you should use X and Y line regions to divide the mesh into fine and coarse regions. The tuning programs automatically add line regions depending upon the settings of the control-file keywords MESH_size and INCrement. You also can use a line region to fix a row of mesh points in the geometry for some purpose. For example, when computing an off-axis integral of the electric field E_z in SFO, a Y line region ensures that all points along the line have the same radial coordinate

1. Using line regions to control the mesh size in Automesh

Line regions use the REG namelist arrays XREG(1:MXLR) and YREG(1:MYLR) for physical coordinates, and KREG(0:MXLR) and LREG(0:MYLR) for logical coordinates. Note that these logical coordinate arrays start with index 0. KREG has MXLR + 1 elements, and LREG has MYLR + 1 elements. MXLR and MYLR are the total number of X and Y line regions supplied to Automesh, Autofish, or a tuning program.

The first element in the arrays KREG and LREG is not a logical coordinate, but an indicator of how to treat the following array elements. Automesh uses the special value KREG(0) = −1 (or LREG(0) = −1) to indicate that all KREG (or LREG) entries are intervals between logical mesh columns (or rows) rather than actual logical coordinates. In addition to the line-region entries themselves, the variables KMAX (or LMAX) also refer to incremental values if the first element in the KREG (or LREG) array is −1.

Consider an example. In this discussion, we will assume that Automesh finds the problem boundaries $X_{min} = 0$ and $X_{max} = 15$ after analyzing the first region. Suppose the following entries appear in the Automesh input file:

```
XREG = 4.5, 5.5,
KREG = –1, 29, 15,
KMAX = 20
```

For these entries, physical coordinate X_{min} = 0 will have logical coordinate K = 1; physical X = 4.5 will have logical K = 1+29 = 30; physical X = 5.5 will have K = 30+15 = 45; and physical X_{max} = 15 will have logical K = 45+20 =65. The following equivalent entries would specify the absolute logical coordinates:

```
XREG = 4.5, 5.5,
KREG = 0, 30, 45,
KMAX = 65
```

When using the arrays KREG and LREG to specify absolute logical coordinates, include any number other than –1 as a placeholder for the first element. For problems with many line regions, you may find it more convenient to supply the increments rather than the actual coordinates. This method allows repeated runs to fine tune the mesh without the need to edit every KREG or LREG element after the one you changed. The choice of the incremental option for the K coordinates is independent of the choice for the L coordinates. You can use the method on either or both sets of KREG or LREG elements. When you use the incremental option, Automesh computes the logical-coordinate values from the increments supplied and reports the actual logical coordinates in file OUTAUT.TXT.

You can specify the first eight line-region parameters by using the names listed in Table VI-16 substituting an integer from 1 to 8 for the lower-case letter i. The nonzero variables of each set increase from 1 to 8. For example, X_{min} < XREG1 < XREG2 < XREG3 < XREG4 < XREG5 < XREG6 < XREG7 < XREG8 < X_{max}, where X_{min} and X_{max} are the minimum and maximum X coordinates in the problem. If the single-valued entry KREG = –1 appears, then the KREGi variables correspond to increments instead of actual logical coordinates. Similarly, if LREG = –1 appears, then the LREGi variables correspond to increments. Several files in subdirectories under LANL\Examples illustrate the use of line regions. See, for example, the HFULL.AM input file for a full-geometry H-shaped magnet, and the FTEST.AM input file for the ferrite-tuned cavity.

When entering line region data, use the notation described in Table VI-16, or use the array method with a string of entries of the required length for your problem. For more than 8 line regions, you must use the array method. Consider an example problem with minimum X coordinate X_{min} = 0, and suppose we wish to use line regions at these four X locations: X = 4.5, X = 5.5 X = 8.5, and X = 10.5. Using the array method, the following entries should appear in the first REG namelist:

```
XREG = 4.5, 5.5, 8.5, 10.5
```

Table VI-16. REG namelist variables for the first 8 line regions.

Variable	Description
XREGi	Physical X location of a line-region boundary.
KREGi	Logical mesh-point number of XREGi.
LINEXi	Indicator for including line region XREGi in the mesh.
YREGi	Physical Y location of a line-region boundary.
LREGi	Logical mesh-point number of YREGi.
LINEYi	Indicator for including line region YREGi in the mesh.

With individual variables from Table VI-16, we can supply the same information:

XREG1=4.5, XREG2=5.5, XREG3=8.5, XREG4=10.5

The second line has an explicit variable for each line-region coordinate. The code does not (yet) allow specification of individual array elements using parentheses, such as the entry XREG(2). To use the array method you must include the entire array up to the maximum number of line regions you intend to specify.

Arrays LINEX(MXLR) and LINEY(MYLR) indicate if the line regions are fixed throughout the mesh or just at intersections with physical boundaries. The points on a line region are allowed to move during mesh optimization if the LINEX or LINEY value is zero. Note that these arrays start with index 1. The first element LINEX(1) corresponds to the X line region with physical coordinate XREG(1) and logical coordinate KREG(1). To hide all lines that define different regions, set REG namelist variable LINES = 0. To show the lines (fixing them in the mesh), set LINES = 1. Use LINESX or LINESY to refer to all X lines or all Y lines. These variables serve as the default values for the arrays LINEX(MXLR) and LINEY(MYLR). You can override the default value for individual lines by specifying a value for an element in LINEX or LINEY. Example files for RFQ cavities use the LINES = 0 option.

You can make the mesh interval increase by a factor RX or RY by supplying only the physical coordinates (XREG or YREG) of the line regions and omitting logical coordinates (KREG or LREG). For example, suppose the problem geometry has $X_{min} = 0$ and $X_{max} = 20$ and the input file specifies DX = 0.1, RX = 2 (the default value), XREG1 = 5, and XREG2 = 10. Starting from $X_{min} = 0$, the mesh will have 50 intervals to X = 5, another 25 intervals to X = 10, and another 25 intervals to $X_{max} = 20$. You cannot use DX and RX (or DY and RY) to increase and decrease the mesh size at different points. In this situation you would use both logical and physical line regions, which override the settings in DX or DY.

2. Choosing the mesh size

Here are some guidelines for choosing a mesh size. For circular arcs, try to use a mesh size smaller than one-fifth the arc radius. Small features should include at least three rows of mesh points in both X and Y directions. Having a large enough number of mesh triangles is particularly important if you need to interpolate fields near a thin object. The field interpolator fits a polynomial to the nearest mesh point and all its first and second nearest neighbors that are in the same material (a maximum of 19 points). There are

fewer available points near boundaries between materials, but usually enough for a good fit. However, the code does not try to determine if a mesh point in the same material is on the other side of an intervening material. If a material region has only one or two rows of triangles, you can expect to see interpolated fields with very poor accuracy.

For problems with cylindrical symmetry, be sure to include plenty of mesh triangles near R = 0 (that is, near X = 0 for Poisson and Pandira problems, or near Y = 0 for Superfish problems). This mesh density is particularly important for cylindrically symmetric electrostatic problems. If you set XJFACT = 0 and ICYLIN = 0 in the first REG namelist, then Automesh will analyze the geometry and warn if the radial mesh size is too large (compared to the longitudinal mesh interval) for an accurate solution.

3. Other tips related to meshing the problem

Some types of geometries prove to be particularly nettlesome for the meshing algorithm in Automesh. For example, we have noticed that some features of ion-source geometries are often hard to mesh. In our efforts to assist users having difficulties setting up a mesh for their problem, we have learned some things that may also help other users:

- Don't allow more than about a factor of 2 or 2.5 change in the mesh interval across line regions.
- If a line region will lie very close to a boundary point of your geometry, make the line region actually coincide with the boundary point. This approach can often prevent a problem that results when Automesh finds too few logical points available in an area.
- Electrostatic problems with cylindrical symmetry use right triangles by default. If you get the message about the mesh containing overlapping triangles, try using equilateral triangles in some of the regions if the problem area is far from the axis (r = 0). Setting ITRI = 0 in the REG namelist for a region overrides the default of ITRI = 2, which specifies right triangles.
- For large problems of ~10^6 mesh points or more, the successive over-relaxation (SOR) procedure in Automesh that adjusts the sizes of the internal mesh triangles may not converge. More than likely, the mesh is good enough after a few cycles of the SOR procedure. Therefore, if the residuals reported by the code fail to improve (or actually start to increase), simply stop the procedure manually. Press the Esc key and then select "Finish" on the next cycle.
- For large problems you also may wish to set more stringent termination criteria for the rf solver in Fish, CFish, and Autofish. The MDTfish tuning program already sets EPSIK = 10^{-8}.

4. Mesh modification option

Automesh has a feature added in April, 2000 that allows modification of the mesh boundaries to reflect distortions computed by a thermal or structural analysis code. The main use of this feature is to calculate the resonant frequency and fields for an rf cavity in its distorted shape. This section discusses how to implement the mesh modification

option. Using an rf cavity (Superfish) problem as an example, the following steps are involved.

1. Run Autofish (or Automesh, Fish, and SFO) on the original problem.
2. Extract the necessary rf field data along boundary surfaces for input into a thermal/structural mechanical analysis code (e.g. ANSYS, ABAQUS, COSMOS, ALGOR, etc.). You can extract the fields along all the boundary segments by running the code SegField. Include a line with keyword NodesOnly in the SegField input file to eliminate all the points at midpoints of the triangle legs.
3. Perform the structural analysis, which computes a distorted shape of the cavity.
4. Output data about the displaced nodes to a file for input to Automesh as discussed below.
5. Run Automesh with the setting ModT36 = 1 in the first REG namelist. The Automesh input file should otherwise be identical to the original file for this cavity shape.
6. Check the section of file OUTAUT.TXT that lists the displaced nodes (look for the phrase "Adjusting boundary-point locations …"). Be sure the code did what you expected.
7. Run Fish to compute the resonant frequency and fields for the distorted shape.

When ModT36 = 1, Automesh expects to find a T36 file, which is a text file containing node displacement data. This file has the same name as the Automesh input file, but with extension T36. The string "T36" refers to the fact that the displacements are applied to coordinates in temporary file TAPE36.

There are three methods for supplying information about the displaced nodes to program Automesh. The program selects a method automatically based upon the type of data found in the T36 file. Table VI-17 describes the three methods. All of the lines in the T36 file must be of the same type. You cannot mix the methods. The code selects the method by looking at the format of the first data line. For example, after discovering that the first data line contains two integers followed by two real numbers, the code assumes method 1, and will then look only for more lines of that type.

For methods 1 and 2, data in the T36 file may be in any order. For each node found in the original TAPE36 file, Automesh searches the T36 file for an entry for that node. If the code finds such an entry, then the new coordinates in the T36 file replace the original coordinates in TAPE36. For method 2, values of X_{old}, Y_{old} must be accurate enough for the comparison with entries in TAPE36. The code looks for matches within 1% of the smallest mesh interval in the problem geometry.

For method 3, the displacements refer to consecutive boundary points in file TAPE36. Program SegField generates the list of coordinates and fields in the same order found in the TAPE36 file. If the first boundary point to be modified is not the first one in TAPE36, then the T36 file must define the first point using the FirstNode keyword, which can appear anywhere in the file. For example, if the first modified node's original coordinates are X = 5.5 and Y = 1.0, then the following line should appear in the file:

FirstNode = 5.5,1.0

Table VI-17. Formats for displaced nodes.

Method	Description	Line contents
1	Lines contain (in any order) the logical coordinates and the new physical coordinates for the boundary points.	K, L, X_{new}, Y_{new}
2	Lines contain (in any order) the old and new physical coordinates for the boundary points.	X_{old}, Y_{old}, X_{new}, Y_{new}
3	Lines contain physical displacements for consecutive boundary points. Optional keyword line FirstNode specifies the coordinates for the first node to be displaced.	δX, δY

The coordinates in file TAPE36 are always in centimeters. If REG namelist parameter CONV in the original Automesh input file was not equal to 1.0, then the coordinates in the T36 file will not be centimeters. Parameters X_{old}, Y_{old}, X_{new}, Y_{new}, δX, and δY in the T36 file must correspond to the units used in the original Automesh input file.

The recommended method for extracting fields along all the boundary segments is to run the code SegField, which reads the data in the SFO output file. The input file for SegField allows specification of the field normalization. To extract data only for the mesh-point nodes, include the line NodesOnly in the SegField input file. If you extract the coordinates and fields from the SFO output file by some other method, be sure to remove duplicate nodes that occur at junctions between segments in the SFO output. Program SegField does this task automatically.

L. Drive points for Superfish problems

Superfish will not run without a properly defined drive point. The drive point is a mesh point with a nonzero value of the magnetic field H_z (or H_ϕ for cylindrically symmetric problems). The code cannot change the drive-point field during the solution. Automesh checks that the drive has been properly assigned and will not allow a drive point out of bounds of the problem geometry. However, this checking does not guarantee that the drive is not inside a conductor. If there is not a valid drive point after generating the mesh, Automesh warns of this situation, but terminates normally so you can view the mesh with WSFplot. Programs Fish and CFish also check that the drive point is in the field region of the cavity. If there is no drive point, the program stops with an error message.

1. Specifying a single drive point

For resonance searches in Fish and CFish, the geometry must contain exactly one drive point. Use XDRI and YDRI in the first REG namelist to specify the X and Y location of the drive point for Superfish problems. If you omit the drive point, Automesh finds a location for XDRI and YDRI. Usually, it is best to place the drive point in the problem yourself so you know it is in a place with appreciable magnetic field for the modes of interest. If you place the drive point in a region with very weak magnetic field compared to the rest of the geometry, the root finder may take longer than necessary to converge to

a resonant mode. There is also a chance it will converge to some other mode that happens to have higher field at the drive location.

2. Line drives for CFish problems

You can use line regions with IBOUND = –1 and appropriate values of HDRIVE and HDRIVEI to initialize both real and imaginary parts of the magnetic field on multiple drive points for CFish problems. On boundaries along a radius in coaxial lines, H_ϕ will be proportional to 1/r. These features can be used for single-iteration problems to simulate traveling waves in a waveguide or coaxial line. You cannot use the Fish root finder to search for resonance if the problem contains multiple drive points.

3. Wavelength and starting phase for a sinusoidal drive line

The SINEDRV array in a REG namelist defines a sinusoidal drive term for CFish problems. This type of drive is intended for rectangular waveguide problems in Cartesian coordinates. SINEDRV is an array of two terms. The first term is the wavelength of the sinusoidal pattern in the same units used for X and Y entries. The second term is the starting phase in degrees. A problem can contain up to 5 separate sinusoidal line-drive regions.

M. Overlapping and adjacent regions

Many problem geometries, especially magnet problems, have multiple regions, some of which overlap and some of which are adjacent to one another. Electrostatic problems may have adjacent regions of metal and dielectric material. Automesh now prevents many potential problems with regions automatically. For example, it is not necessary (as in some older versions of the code) to traverse common line or arc segments in the same direction. Automesh scans all the regions and looks for segment end points that will appear on other boundaries. The code adds these physical points in all the regions that share the common line or arc segment.

In some cases, when several regions share common boundary points and line segments, Automesh may be unable to mesh the geometry without distorting a physical boundary. These cases usually involve four or more segments that share the same mesh point. Because the mesh is triangular, each point is surrounded by six other points. Thus, an absolute maximum of six lines can logically terminate on the same point. In practice, Automesh handles intersections of three lines reliably; it occasionally has trouble with four lines; it fails frequently with five lines; and it would probably require a miracle to mesh the intersection of six lines.

The best practice is to avoid multiple overlapping regions when possible. Line regions defined using the XREG and YREG variables also use up available logical points in the mesh. Thus you should always avoid adding X or Y line regions at points in the mesh that already are shared by multiple segments. Sometimes eliminating overlapping regions actually makes the Automesh geometry a truer representation of the physical geometry. However, you may need a very fine mesh to include (for example) a small air gap between two pieces of iron.

Some problems include unclosed regions with IBOUND = 1 for a Neumann boundary or IBOUND = 0 for a Dirichlet boundary. These regions should appear near the end of the Automesh input file. Metal regions in electrostatic (and also rf) problems also define a Dirichlet boundary. If an adjacent material such as a dielectric appears later in the input file, it may remove the Dirichlet boundary condition along the shared boundary segments. To see the effects of incorrect boundary conditions you need to calculate and display equipotential lines. For example, in electrostatic problems look for equipotential lines that incorrectly terminate perpendicular to a metal surface.

N. Indicating the triangle type used to fill the mesh

Variable ITRI in a REG namelist determines the type of triangle, according to Table VI-18, used to fill interior regions of the mesh. The default is ITRI = 0 for all problems except Poisson and Pandira electrostatic problems with cylindrical symmetry. For these problems, which have XJFACT = 0 and ICYLIN = 1, the default setting is ITRI = 2. The triangle type is one of several REG namelist variables that influence the default values listed in Table VI-2 for the mesh interval DY relative to DX.

Table VI-18. The three possible values for triangle type ITRI.

ITRI	Triangle type
0	Equal-weight, equilateral triangles.
1	Equilateral triangles.
2	Right triangles.

In Table VI-18, the term equilateral really mean approximately equilateral to the extent possible given the mesh spacing in the X and Y directions. The term “equal-weight” for ITRI = 0 means that all six nearest neighbors of a point contribute equally to the computed value of the point’s X,Y coordinates. For equal-weight triangles, the new X coordinate depends only on the average of the X coordinates of the six nearest neighbors. Similarly, the new Y depends only on the other Y coordinates. For ITRI = 1, the new X (or Y) coordinate is a function of both X and Y coordinates of the nearest neighbors. The setting ITRI = 1 produces a slightly better mesh than does ITRI = 0 in the sense that a triangle is more similar to nearby triangles. However, mesh optimization usually takes more iterations for ITRI =1, and an iteration takes about 30% longer than for ITRI =0.

For most problems, especially those without line regions and for which Automesh uses the default DY, there is very little difference between ITRI = 0 and ITRI = 1. For geometries with complex features, or for meshes with large aspect ratios, you should consider using ITRI = 1. One way to check whether the ITRI setting makes much difference is to compare the computed stored energies for both cases. A smoother (i.e., better) mesh will result in a slightly lower stored energy.

You can specify the triangle type for each closed region, if necessary. If regions overlap, then Automesh uses the ITRI setting for the last region that covers a portion of the geometry. We have observed some dependence on the triangle type of the quality of fits found by the field interpolator. For example, for cylindrically symmetric magnet problems, the interpolator usually does better in the interior part of the mesh with

equilateral triangles. But the fitted chi-squared per degree of freedom is better with right triangles near a vertical Dirichlet boundary with R > 0 (that is, not on the symmetry axis).

Cylindrically symmetric electrostatic problems always give more accurate results with right triangles. Older versions of Poisson and Pandira could get very poor solutions for cylindrically symmetric electrostatic fields. The problem is related to finding the coupling term for triangle legs on the axis R = 0 (see Eq. B.1.2.2 in the 1987 Reference Manual.) The solutions are far more accurate if the mesh has right triangles (ITRI = 2) instead of approximately equilateral triangles (ITRI = 0 or 1). The default triangle type in Automesh is ITRI = 2 for electrostatic problems with cylindrical symmetry and ITRI = 0 for other problem types.

If you set XJFACT = 0 and ICYLIN = 1 in the Automesh input file to indicate a cylindrically symmetric electrostatic problem, then Automesh will warn if you use ITRI = 0 or ITRI = 1. If you do set ITRI = 0 or ITRI = 1, then Automesh analyzes the aspect ratio of the mesh triangles in every section between XREG-defined regions. The code warns if the X mesh spacing is too large for accurate results.

1. Bad triangles: overlapping, zero-area, and near zero-area triangles

Automesh now eliminates many of the errors of previous versions that lead to crashes caused by bad mesh triangles. However, problems can still occur occasionally. [Older versions of Automesh referred to "negative-area triangles." The term derives from the algorithm used to calculate the area of a mesh triangle. If the code obtained a negative value, it indicated an inconsistent ordering of the logical mesh points.] If the logical mesh has been corrupted, you must try to resolve the problem in the Automesh input file before proceeding any further.

a. Fatal errors, overlapping triangles or zero-area triangles

Automesh may report that the mesh contains overlapping triangles or zero-area triangles. This message indicates a serious problem with the points assigned by Automesh along boundary segments. The solver programs will not run, but you can use WSFplot to look at the generated mesh.

A common cause of the error is a mesh spacing coarser than some feature in the geometry. An example is an arc whose radius is only 2 or 3 times the mesh spacing. Trying to add a line region where the mesh is already too coarse also might cause this error. In a coarse mesh, the code may have too few logical mesh points for all the physical coordinates needed for boundary points and line regions. Thus, the first thing to try is a finer mesh. If Automesh still fails with a finer mesh, there may be an error in the problem geometry. Use WSFplot to look only at the boundary lines (that is, without the mesh). Expand the scale near the bad triangles if the problem is not immediately obvious.

b. Nonfatal errors, near zero-area triangles

Automesh will check for and may warn of "near zero-area" triangles. A near-zero-area triangle has

$$\text{Area} < 0.01 * \text{DXMIN} * \text{DYMIN},$$

where DXMIN and DYMIN are the smallest X and Y mesh intervals used in the problem. Though less serious than overlapping triangles or zero-area triangles, abnormally small triangles also indicate possible problems with the boundary-point data along segments. The solver codes will run, but the results may be unreliable. A slightly finer mesh will often eliminate the warning message. Intersecting line regions very near a physical boundary may also result in near-zero-area triangles.

O. Indicating the cavity type for program Parmila

For Superfish cylindrically symmetric rf cavity problems, SFO will write a special file for the beam-dynamics code Parmila. The file has the same name as the Automesh input file, but with extension PMI. For multicell cavities, Parmila may use up to three different tables of transit-time factor data that correspond to different boundary conditions at interfaces between cells or at the end of a structure. The second line of the PMI file contains information about the cavity type that program ReadPMI (distributed with Parmila) uses to generate the appropriate tables.

For single-cell cavities, this cavity-type setting is not important. The fact that the PMI file contains the line "Cavity type: Unknown RF Cavity" will not affect the results. The table created by ReadPMI will be the same type used for drift-tube linac (DTL) structures.

This cavity information usually comes from the tuning program (CCLfish, CDTfish, DTLfish, ELLfish, MDTfish, or SCCfish) that generated the Automesh input file. However, the user also can supply the information using the REG namelist variables CCL, DTL, and CCDTL. These variables are common acronyms for the rf structures listed in Table VI-19. The tuning programs include the appropriate REG namelist variable in the Automesh input file to indicate the cavity type for Parmila. If none of the indicators appear, then Automesh or Autofish would identify the cavity type as "Unknown RF Cavity" (see the comment above). Program DTLfish sets DTL = 1 and MDTfish sets DTL equal to the number of cells or rf gaps in the problem. CCLfish and SCCfish both set CCL = 1. CDTfish sets CCDTL equal to the number of rf gaps in a coupled-cavity drift-tube linac cavity. Program ELLfish sets SCCAV = 1, and program ELLCAV sets SCCAV equal to the value of NumberOfCells that appears in the ELLfish control file.

Table VI-19. REG namelist variables that specify a cavity type.

Variable	Acronym for	Possible values
DTL	Drift-Tube Linac.	0: not a DTL problem; >0: number of DTL cells.
CCL	Coupled-Cavity Linac.	0: not a CCL problem; 1: a CCL cavity.
CCDTL	Coupled-Cavity Drift-Tube Linac.	0: not a CCDTL problem; >0: number of gaps.
SCCAV	Superconducting cavity.	0: not a SCCAV problem; >0: total number of cells in a superconducting cavity.

Only the last one of these variables to appear in the file has any effect. Each time Automesh encounters one of these variables, it sets all the other indicators to zero.

P. Controlling the relaxation order for the mesh points

Variable NOTE determines the order in which Poisson calculates new values for the mesh points during each iteration. The default is NOTE = 1, which gives the relaxation order: ("air" + interface) points, then iron points. Following the usage in the 1987 Reference Manual, the term "air" point refers to a mesh point in empty space. The setting NOTE = 0 gives the order: air points, interface points, then iron points. In previous versions of the codes, program Pandira required that NOTE = 0. Automesh now sets up a separate ordered relaxation array for Pandira, so NOTE = 0 is not required. You can still set NOTE = 0 in the first REG namelist to change the order in Poisson, but the order makes little or no difference in the code's performance. The value of NOTE cannot be changed after running Automesh.

Q. How Automesh initializes the potential

While processing REG namelist sections in the input file, Automesh stores, for later generating the mesh, the region number IREG and the variables listed in Table VI-20. The 1987 Reference Manual refers to the first six parameters collectively as the "C-array" (see sections B.3.2 and C.3.2). Values of these parameters appear on the heading line for a region's coordinates in the temporary file TAPE36.

Automesh no longer allows user input of the variable IREG. Instead, Automesh assigns sequential values to IREG for each region. Variables CUR and DEN have synonyms that are useful for readability when they refer to the Superfish drive point or to a charge density or voltage in an electrostatic problem. Table VI-20 lists the synonyms plus default values. In addition to the original six values, the C-array includes other elements assigned by Automesh. For example, C(7) is a variable JBOUND that distinguishes between line regions and regions that are actual boundary surfaces. Elements C(8) through C(13) contain information about line-drive regions used in CFish problems.

Automesh resets each parameter to its default value before every new region. (In previous versions of Automesh, variables DEN and CUR retained their values from the last region unless they were explicitly reset in the REG namelist for the next region.) The new default values will prevent the problem some users have experienced when they forgot to reset a current or current density to zero following a region with a nonzero value.

Table VI-20. The C-array parameters that determine the initial potential.

Variable	C	Synonyms in CAPS or description	Default value
(IREG)	1	Sequential region number	(assigned by Automesh)
MAT	2	MATERIAL	1
CUR	3	CURRENT, CHARGE, HDRIVE, VOLTAGE	0.0
DEN	4	DENSITY, HDRIVEI	0.0
ITRI	5	Triangle type indicator	0 (2 if XJFACT = 0 and ICYLIN = 1)
IBOUND	6	Boundary type indicator	(see Table VI-14)
(JBOUND)	7	Region type indicator	0 for boundaries, –1 for line regions
SINEDRV(1)	8	Wavelength of sine pattern	(none)
SINEDRV(2)	9	Starting phase of sine pattern	0
(none)	10,11	Line-drive starting X,Y coordinates	(assigned by Automesh)
(none)	12,13	Line-drive ending X,Y coordinates	(assigned by Automesh)

Variable IBOUND specifies the boundary condition along a path that is not part of the outer rectangular edge of the problem geometry, it also indicates fixed-potential points. Refer to section G for more information.

When CUR or DEN is nonzero, Automesh calculates values for all points in the region and initializes the vector potential on these mesh points. Note that the region might be a single point, several lines or curves, or a closed path that encloses an area. If the region encloses an area, then DEN has units of Amps/length^2 for magnet problems or Coulombs/length^3 for electrostatic problems. For a line or curve, the units are Amps/length or Coulombs/length^2, and for a single point the units are Amps or Coulombs/length. The default length unit is centimeters (that is, when CONV = 1.0).

For electrostatic problems, the field interpolator gives incorrect results on or near point regions or line regions containing charge. The interpolation is correct in charge-free regions provided that the point of interest is at least two mesh divisions from a point or line region with charge. The interpolator also gives correct results inside a closed area containing charge and across the interface between the closed charge area and charge-free regions.

The code now initializes current (or charge) terms slightly differently from older UNIX release 4.12. In the older version, a new source term writes over any previously initialized values. Automesh adds the calculated source term to the present value of the vector potential already set by previous regions. However, when you use IBOUND = –1 (for example, to set the voltage for an electrostatic problem), Automesh replaces previously initialized values with the new value (just like older versions).

An example best illustrates the reason for this change. Suppose you want to enter a surface current on a circular region. If you try to do it with a single region in the Automesh input file, the code will interpret it as an area and distribute the current among all the enclosed mesh points. The solution is to break the assignment into two unclosed paths, say one from 0 to 180 degrees and another from 180 to 360 degrees. However, for each unclosed region, the code allocates the current among the points based upon the length of the line. Each point gets a fraction of the total current equal to the length of the

adjacent legs within the region divided by the total length of the line. For equal intervals along the line, the two end points get half the current of the other points. After both regions have been added, all points around the circle will have equal source terms. If instead, Automesh writes over the previous assignments, then there would be two points on this surface with incorrect source terms. A potential problem to be aware of is that you cannot overlap coil regions unless you take into account how the code will superimpose the source terms.

For Pandira problems, remember that currents regions are treated as real currents only if IPERM is zero. The setting IPERM = 0 is the default if the input file includes any regions with nonzero current.

R. Variables calculated by Automesh

Automesh calculates a number of quantities that are used by the solver programs or by postprocessors. These parameters appear in a list in each code's output file for your information. The serious user should examine these output files.

1. NREG and MAXPPR, number of regions and maximum points per region

Automesh counts the total number of REG and POA namelist sections in the input file and stores the value as NREG for use by subsequent programs. Previous versions of these codes required a user-supplied value of NREG in the first REG namelist. NREG may be larger than the number of REG namelist sections. Line regions for changing the mesh interval count as separate regions. The Superfish drive point and the point XBZERO,YBZERO in Poisson problems also generate a separate region.

MAXPPR is the maximum number of points used by Automesh to hold all the interpolated boundary points for any one region. Automesh uses it to allocate the required space for its boundary-point arrays.

2. XMING, XMAXG,YMING, and YMAXG, the problem boundaries

Automesh analyzes the first region boundary points, which must define a closed area, to determine values for X_{min}, X_{max}, Y_{min}, and Y_{max}. Automesh and all subsequent Poisson Superfish codes report these values as XMING, XMAXG,YMING, and YMAXG in file OUTAUT.TXT and other output files that list the problem variables.

3. KMAX and LMAX, the number of mesh intervals along each coordinate

Automesh calculates values for KMAX and LMAX if you do not supply these values in the first REG namelist in the input file. KMAX and LMAX are the number of logical points along X and Y, respectively. You can enter these values along with logical coordinates of line regions [KREG(i), LREG(i)] to control mesh size. If omitted, then Automesh determines a value based upon the input data and the mesh intervals DX and DY.

If you use the special option to enter logical-coordinate increments for X or Y line regions, then input values for variables KMAX or LMAX must also be incremental values. For example, if the first element in the KREG array is KREG(0) = –1, then

KMAX is an incremental value instead of the actual maximum logical K coordinate. Automesh computes the logical-coordinate values from the increments and reports the actual logical coordinates in file OUTAUT.TXT.

The solver programs Fish, CFish, and Pandira allocate temporary data arrays using the dimension NROW, which is the minimum of LMAX and KMAX.

4. ITOT, the total number of mesh points in the problem

Automesh calculates for use by all the codes the total number of mesh points in the problem according to the equation:

ITOT = (KMAX+2)*(LMAX+2)

The code adds an extra logical row and column around the region of interest defined by KMAX and LMAX and stores material number MAT = 0 for the rows of triangles outside the problem geometry. Automesh computes ITOT and passes the value to other codes. You cannot change KMAX, LMAX, or ITOT after values have been set in Automesh. All programs allocate ITOT memory locations for each mesh-point coordinate, the solution array, and several other arrays.

5. DX1, the X mesh interval in the first region

Automesh computes DX1, the size of the X mesh interval at the left edge of the geometry. For electrostatic problems with cylindrical symmetry, Poisson and Pandira use one-third of this value for the "average radius" when setting coupling terms for triangle legs on R = 0. This approach can cause errors in the electrostatic fields near R = 0 unless the mesh consists of right triangles. The Automesh default is ITRI = 2 for electrostatic problems with cylindrical symmetry, which selects right triangles for a region. For other problems, the default is ITRI = 0 (equal-weight, equilateral triangles).

6. DXMIN and DYMIN, the smallest mesh intervals

DXMIN and DYMIN are the smallest X and Y mesh intervals used in the problem computed by Automesh. They may help to diagnose problems with Automesh. The code also checks for and may warn of "near zero-area" triangles.

7. XDRI,YDRI, the drive-point location for Superfish problems

For Superfish problems, if the user does not supply a drive point in the first REG namelist, then Automesh finds a location for XDRI,YDRI. After generating the mesh, the code finds the logical coordinates KDRI,LDRI associated with XDRI,YDRI. The physical coordinates Automesh also reports NDRI, which is the index in the mesh-point arrays for the drive point.

8. XBZERO,YBZERO, physical point where B = BDES in Poisson

If you enter the logical coordinates KBZERO and LBZERO, at which Poisson should make B = BDES, then Automesh computes the physical coordinates of the point XBZERO,YBZERO.

9. KBZERO,LBZERO, logical point where B = BDES in Poisson

If you enter the physical coordinates XBZERO and YBZERO, at which Poisson should make B = BDES, then Automesh computes the logical coordinates of the point KBZERO,LBZERO.

10. IPERM, indicates if a Pandira problem includes real currents

For Pandira problems, current regions are treated as real currents only if IPERM is zero. The setting IPERM = 0 is the default if the input file includes any regions with nonzero current. If IPERM = 1, then the problem contains no real currents and must contain at least one region of permanent-magnet material.

11. NSEG, the number of boundary segments

Automesh counts the total number of boundary segments and stores the result in NSEG for use by subsequent programs. SFO allocates arrays of length NSEG to store the results of power and field calculations along the boundary segments.

12. NPBOUND, the number of points along all boundary segments

Automesh counts the total number of mesh points along all the boundary segments and stores the result in NPBOUND for use by subsequent programs. SFO allocates arrays of length NPBOUND to trace the path along each segment for power and field calculations.

13. NAIR, the number of mesh points in empty space

Automesh counts the total number of mesh points in the “air” or empty space region and stores the value as NAIR for use by subsequent programs. All air points correspond to material MAT = 1.

14. NFE, the number of mesh points in iron

Automesh counts the total number of iron points and stores the value as NFE for use by subsequent programs In this context, “iron” also can refer to dielectric materials or magnetic materials with μ not equal to μ_0. A better definition might be non-air points having a value of MAT other than unity.

15. NINTER, the number of mesh points on an interface

Automesh counts the total number of points on an interface between different materials and stores the value as NINTER for use by subsequent programs.

16. NBND, the number of mesh points on Dirichlet boundaries

Automesh counts the total number of mesh points on Dirichlet boundaries and stores the value as NBND for use by subsequent programs.

17. NSPL, the number of mesh points with user-defined fixed-potential

Automesh counts the total number of points containing fixed potential values supplied by the user and stores the value as NSPL for use by subsequent programs. For Poisson

problems, NSPL corresponds to the number of points affected by regions with IBOUND = –1. Poisson and Pandira will increase the value of NSPL if you use one of the options in these codes for entering fixed potentials. For a typical Superfish problem, NSPL = 1 and corresponds to the Superfish drive point. For CFish problems with multiple drive points, the value of NSPL is the number of drive points.

18. NPINP, the number of mesh points in the problem

Automesh adds the values it calculates for the number of points in empty space, in iron, on interfaces, on boundaries, and containing user-supplied fixed potentials and stores the value as NPINP for use by subsequent programs. Thus, the value of NPINP is equal to

NPINP = NAIR + NFE + NINTER + NBND + NSPL

The variable name comes from the phrase "Number of Points IN the Problem." Poisson uses this sum and the variables that contribute to the sum in the successive over-relaxation calculation. NPINP will always be smaller than ITOT, the total number of mesh points because of the extra row and column of points used for bookkeeping, and because some interior mesh points may not be part of the problem geometry.

19. NWMAX, the number of points for recalculating couplings

NWMAX is the number of points for recalculating couplings during the solution in Poisson and Pandira. The solvers use it to allocate arrays of proper length for a problem.

20. NGMAX and NGSAM, coupling array counters

NGMAX is the total number of points in the coupling arrays of length NWMAX for which the solvers Poisson and Pandira will recalculate the reluctance at each iteration. NGSAM is the number of points in a subset of NGMAX that have the same material in the point's upper and lower triangles.

21. NPONTS, the number of unknown relaxation points

NPONTS is the total number of mesh points in the problem that Automesh is free to move in optimizing the mesh.

22. LCYCLE, the number of relaxation iterations

LCYCLE is the number of relaxation iterations used by Automesh in optimizing the mesh.

23. CLIGHT and PI, the speed of light and the number π

Automesh initializes variables CLIGHT and PI for use by subsequent programs. CLIGHT is the exact value of the speed of light in cm/sec. This value is 29979245800.0000. To get PI, the code uses the statement:

```
PI = 4.D0*ATAN(1.D0)
```

On Pentium processors running Lahey Fortran with double precision, the numerical value is 3.14159265358979.

24. EPS0 and FMU0, permittivity and permeability of free space

Automesh stores the value of the physical constants ε_0 and μ_0 (the permeability and permittivity of free space) for use in other codes. The constants are defined in MKS units as follows by the following lines of code:

```
FMU0 = PI*4.0D-7
EPS0 = 1.D0/(FMU0*(0.01D0*CLIGHT**2))
```

FMU0 has units of Tesla-meter/Amp and EPS0 has units of Farads/meter.

25. SPOSG, SNEGG, STOTG, currents at generation for magnet problems

SPOSG is the total positive current in the problem geometry, SNEGG is the total negative current, and STOTG is the net current.

26. XYAREA and VOLUME, area and volume in the problem geometry

XYAREA is the calculated total cross-sectional area in the problem found by summing the areas of all the mesh triangles. VOLUME is the cavity volume for cylindrically symmetric problems. These variables provide additional information about your problem geometry.

27. TRIMIN, TRIMAX, and TRIAVG, size of mesh triangles

TRIMIN is area of the smallest positive-area mesh triangle, TRIMAX is the area of the largest mesh triangle, and TRIAVG is the average triangle area. Automesh finds these values during its checks for overlapping triangles and abnormally small triangles. These parameters may help to diagnose problems with Automesh.

S. Automesh error messages

Table VI-21 lists the Automesh error messages. The error numbers also are available as exit error codes to tuning programs and batch control files.

Table VI-21. Automesh error messages.

Error	Description
101	(Autofish) Cannot run Autofish on Poisson or Pandira problem.
102	Input file does not contain a REG namelist section.
103	Input file has no REG namelist before the first PO namelist.
104	Cannot read REG namelist data because of possible damaged file.
105	Cannot read PO namelist data because of possible damaged file.
106	Cannot read MT namelist data because of possible damaged file.
107	Cannot read POA namelist data because of possible damaged file.
108	Unknown single-character entry in REG namelist.
109	Unknown single-character entry in PO namelist.
110	Unknown single-character entry in MT namelist.
111	Unknown single-character entry in POA namelist.
112	A REG namelist entry contains no data or an invalid numerical entry.
113	A PO namelist entry contains no data or an invalid numerical entry.
114	An MT namelist entry contains no data or an invalid numerical entry.
115	A POA namelist entry contains no data or an invalid numerical entry.
116	A REG namelist variable contains too many numerical entries.
117	A PO namelist variable contains too many numerical entries.
118	An MT namelist variable contains too many numerical entries.
119	A POA namelist variable contains too many numerical entries.
120	Variable can only appear in the first REG namelist.
121	Material number is outside defined range. Please report this error.
122	Material ID number is too large. Please report this error.

Table VI-21. Automesh error messages. (continued)

Error	Description
123	Material ID number is smaller than lowest internal material table number.
128	XJFACT and XJFEND cannot have opposite sign to avoid confusion if XJFACT = 0 occurs.
129	Incompatible options specified involving current multiplier XJFACT.
130	Exceeded maximum of 400 entries for a B-Gamma table.
131	Exceeded maximum of 400 entries for a B-Mu table.
132	Exceeded maximum of 400 entries for a B-H table.
140	Input file has no PO namelist after the first REG namelist.
141	The first region has too few PO namelist entries to make a closed area.
142	The first region is not closed.
144	The Automesh input file contains an invalid entry for variable KPROB.
145	The Automesh input file contains multiple entries for variable KPROB.
146	Variable KPROB does not appear in first REG namelist.
147	A REG namelist line contains an invalid format.
148	A PO namelist line contains an invalid format.
149	An MT namelist line contains an invalid format.
150	A POA namelist line contains an invalid format.
151	Invalid entry following a REG namelist array name.
153	Invalid entry following an MT namelist array name.
155	A variable name is not a REG namelist variable.
156	A variable name is not a PO namelist variable.
157	A variable name is not an MT namelist variable.
158	A variable name is not a POA namelist variable.
160	Exceeded maximum number of XREG entries. Please report this error.
161	Exceeded maximum number of KREG entries (must agree with number of XREG entries).
162	Exceeded maximum number of LINEX entries (must agree with number of XREG entries).
170	Exceeded maximum number of YREG entries. Please report this error.
171	Exceeded maximum number of LREG entries (must agree with number of YREG entries).
172	Exceeded maximum number of LINEY entries (must agree with number of YREG entries).
176	Exceeded maximum of 2 entries for SINEDRV.
177	Exceeded maximum of 2 entries for EPSILON.
178	Exceeded maximum of 2 entries for MU.
180	Error reading first title line from Automesh input file (could be empty file).
181	Error reading remaining title lines from Automesh input file.
182	Input file contains no namelist data or has incorrect namelist indicator.
190	A line in file Tape36 contains nonnumeric data. Please report this error.
191	Unknown error reading temporary file Tape36. Please report this error.
192	Unexpected end of temporary file Tape36. Please report this error.
193	Unexpected character encountered in temporary file Tape36. Please report this error.
201	Input file specified by on command line does not exist.
204	File entered in browse dialog window does not exist.
290	Insufficient memory, cannot allocate MT namelist arrays for Automesh setup.
291	Insufficient memory, cannot allocate MT namelist arrays for a tuning code.
292	Insufficient memory, cannot allocate MT namelist arrays for a Poisson problem.
301	Insufficient memory, cannot allocate line region arrays.
302	Insufficient memory, cannot allocate boundary-point and region arrays.

Table VI-21. Automesh error messages. (continued)

Error	Description
304	Insufficient memory, cannot allocate internal arrays for points on boundary segments.
305	Insufficient memory, cannot allocate internal arrays for reading displaced node coordinates.
306	Error allocating temporary arrays while tracing logical path. Please report this error.
307	Insufficient memory, cannot allocate material arrays to start mesh optimization.
308	Insufficient memory, cannot allocate setup arrays for mesh optimization.
310	Insufficient memory, cannot allocate mesh-point arrays that are stored in the solution file.
311	Insufficient memory, cannot allocate Automesh setup arrays.
319	Too many MT namelist sections containing user-defined materials.
320	Boundary-point array overflow while reading PO namelist data.
321	Boundary-point array overflow while saving boundary point data for later output.
322	Boundary-point array overflow while inserting segment end points from other regions.
323	Boundary-point array overflow while inserting points at line-region intersections.
324	Internal array overflow while saving region data for later output.
325	Internal array overflow while storing boundary segment data.
326	Internal array overflow while adding X line regions.
327	Internal array overflow while adding Y line regions.
328	Internal array overflow while tracing logical path along a boundary segment.
329	Internal array overflow while saving logical path information for a segment.
330	PO namelist variable NT, which specifies the type of segment, is out of allowed range.
331	The first point on a boundary cannot indicate a curved segment.
332	Insufficient data supplied in a PO namelist for a line or arc segment.
333	Insufficient data supplied in a PO namelist for hyperbolic segment.
334	Insufficient data supplied in a PO namelist, variable RADIUS missing for NT = 4.
335	Insufficient data supplied in a PO namelist, variable RADIUS missing for NT = 5.
336	Bad PO namelist data, points are too far apart to be connected by an arc with specified radius.
337	Bad PO namelist data, the radius of an arc cannot be zero.
338	Bad PO namelist data, for elliptical arcs, the area of the ellipse cannot be zero.
339	Bad PO namelist data, data supplied for an arc segment is inconsistent.
340	Bad PO namelist data, variable R for a hyperbolic segment must be positive.
341	Bad PO namelist data, data supplied for a hyperbolic segment is inconsistent.
342	Two consecutive boundary points have the same physical coordinates.
343	The center of an arc coincides with the end of the last segment.
344	There is no available logical point for a physical point.
345	There is no available logical point for an X line region intersection with the current segment.
346	There is no available logical point for a Y line region intersection with the current segment.
351	No PO namelist sections found for a region.
352	A PO namelist describes a point that is outside of the problem geometry.
353	The drive point for an rf problem is outside the problem geometry.
354	The point for scaling the field in a Poisson problem is outside the problem geometry.
355	The point at which A = 0 in the harmonic analysis is outside the problem geometry.
356	A POA namelist point is outside the problem geometry.
357	A POA namelist contains insufficient data.
360	Logical path finding failed in both directions for a segment.
361	Unable to trace the logical path along an arc segment (mesh may be too coarse).
362	Cannot find logical coordinates for the next point on an arc segment (mesh may be too coarse).

Table VI-21. Automesh error messages. (continued)

Error	Description
363	A segment has zero length after removing superfluous points. The mesh is too coarse.
364	Could not assign end points for an X line region.
365	Could not assign end points for a Y line region.
366	The first and last points of a segment have the same logical coordinates.
372	Missing the KREG logical coordinate increment for an XREG line region.
373	Missing the LREG logical coordinate increment for a YREG line region.
374	The logical KMAX increment that corresponds to X_{max} is missing.
375	The logical LMAX increment that corresponds to Y_{max} is missing.
378	XREG line-region physical coordinates do not increase monotonically.
379	KREG line-region logical coordinates do not increase monotonically.
380	YREG line-region physical coordinates do not increase monotonically.
381	LREG line-region logical coordinates do not increase monotonically.
382	At least one XREG physical coordinate exceeds X_{max}.
383	At least one KREG logical coordinate exceeds KMAX.
384	At least one YREG physical coordinate exceeds Y_{max}.
385	At least one LREG logical coordinate exceeds LMAX.
386	There are no mesh divisions between two X line regions (mesh is too coarse).
387	There are no mesh divisions between two Y line regions (mesh is too coarse).
388	Y_{min} is negative for Superfish problem with cylindrical symmetry.
389	X_{min} is negative for Poisson or Pandira problem with cylindrical symmetry.
390	An error occurred reading the file containing modified (displaced) node coordinates.
400	Too many sinusoidal drive lines for a CFish problem.
401	Too many boundary points sent to mesh optimizer. Please report this error.
402	Number of table entries for a region not a multiple of 4. Please report this error.
403	An interior mesh point has negative or zero K. Please report this error.
404	An interior mesh point has negative or zero L. Please report this error.
405	An interior mesh point's K coordinate exceeds KMAX. Please report this error.
406	An interior mesh point's L coordinate exceeds LMAX. Please report this error.
407	Two consecutive points on a segment have the same K,L coordinates. Please report this error.
408	Two consecutive points on a segment have the same X,Y coordinates. Please report this error.
409	Consecutive points are inconsistent with a valid logical path. Please report this error.
410	Not enough memory is allocated for problem arrays. Please report this error.
420	An MT namelist has an invalid identification (MTID) number.
421	An MT namelist has incomplete permanent-magnet data.
422	An MT namelist has BCEPT = 0 and nonzero value of HCEPT.
423	An MT namelist has a negative value for BCEPT.
424	An MT namelist has HCEPT = 0 and nonzero value of BCEPT.
425	An MT namelist has a positive value for HCEPT.
426	Data in an MT namelist table (after BGAM, BMU, or BH) has an odd number of entries.
427	An entry in an MT namelist table (after BGAM, BMU, or BH) is negative.
428	An entry in an MT namelist table (after BGAM, BMU, or BH) has B = 0.
429	B does not increase monotonically in an MT namelist table (after BGAM, BMU, or BH).
431	An MT namelist has a negative value for GAMMA.
432	An MT namelist has a negative value for EPSILON.
433	An MT namelist has a negative value for MU.

Table VI-21. Automesh error messages. (continued)

Error	Description
440	A material with MSHAPE = 1 does not refer to a user-defined material table.
441	A material with MSHAPE = 1 refers to an MT namelist with no permanent magnet data.
442	A material with variable permeability refers to an MT namelist without a material table.
443	Material data has not been defined for an rf problem.
444	A material has an unphysical value for the permittivity in an electrostatic problem.
451	Cannot read the logical boundary segment end points. Please report this error.
452	Cannot specify a boundary value at r = 0 where for cylindrically symmetric rf problems.
453	The mesh has points with undefined coordinates, probably because of unclosed regions.
454	Cannot compute the angle between mesh triangle legs. Leg 1 ends have the same X,Y.
455	Cannot compute the angle between mesh triangle legs. Leg 2 ends have the same X,Y.
456	Cannot compute the angle between mesh triangle legs. Arccosine argument out of range.
457	There are no interior points in the mesh, probably because no regions are closed.
493	The mesh has overlapping triangles or zero-area triangles.

VII. Fish and CFish, Radio-Frequency Field Solvers

Fish and CFish are the radio-frequency field solvers. Fish solves problems with real fields. CFish is a version of Fish that uses complex variables for the rf fields, permittivity, and permeability. They must be run after Automesh for radio-frequency field problems. The codes run in the following order:

- Automesh
- WSFplot (optional)
- Fish or CFish
- WSFplot (optional)
- SFO
- WSFplot (optional)
- SF7

Run SFO before running the other postprocessors (SF7 or WSFplot) to apply the field normalization. If you do not run SFO first, then SF7 and WSFplot use the default normalization that corresponds to H_ϕ = 1000 A/m at the drive point (or to a user-defined value of ASCALE.

A. Starting and stopping Fish and CFish

To start Fish or Cfish, right-click on the binary solution file, which has extension T35 and select either "Run Fish" or "Run Cfish.". You also can use a command line (for example, in batch files) to start the code:

Fish *T35file*
CFish *T35file*

where *T35file* is the name of a binary solution file. It is usually not necessary to enter the name of the *T35file* immediately after running Automesh because the code will find the name of the solution file to open in file TAPE35.INF. If the file listed in TAPE35.INF either does not exist or was not created recently enough, the code opens the standard Open dialog window. For more details about startup options refer to the section titled Opening input files on startup.

1. Using the Esc key to stop Fish or CFish early

Type END to stop Fish at any program prompt. During a frequency scan or a resonance search, you can press the Esc key to stop the program. Fish gives you three options:

- Finish this iteration, save the solution, then stop normally.
- Continue the calculation.
- Stop now without saving the solution (the default).

Choosing Stop returns exit error code 56. Tuning programs can prompt the operator for further instructions.

B. Fish and CFish files and filename conventions

Fish and CFish read the binary solution file created by Automesh. They append the solution to this file. The codes also write the OUTFIS.TXT output summary. The codes also write FishScan.TBL, a Tablplot input file containing results of mode searches and frequency scans. See Files and filename conventions for more information.

Depending upon your computer's memory configuration, programs Fish and CFish may write a temporary file on one of your hard drives. The program will use memory for temporary data arrays if possible. If there is insufficient memory, the program creates the scratch file on the hard drive with the most free space. If it cannot find enough free disk space, the program stops with a message telling you how much extra space it needs on one of your drives. For more information, refer to the section on Temporary data arrays.

C. Data entries for Fish and CFish

Table VII-1 lists the variables used in programs Fish and CFish supplied in the first REG namelist of the Automesh input file. For the complete list of variables see Superfish Variables.

Table VII-1. Variables used by Fish and CFish.

Variable	Default value	Description
CONV	1.0	Length conversion (number of cm/unit).
ICYLIN	1	Coordinate system (0: rectangular; 1: cylindrical).
DSLOPE	1.0	Slope of the $D(k^2)$ function.
DSTOLER	0.02	Tolerance on slope of $D(k^2)$ for convergence.
MAXCY	19	Maximum number of cycles to find resonance.
NSTEP	I	Number of steps for a resonance search.
DELFR	I	Frequency step size (in MHz) for a resonance search.
FREQ	I	Resonant frequency (also starting frequency) (in MHz).
IPIVOT	1	Controls pivoting in matrix inversion routines.
EPSIK	1.0D–4	Frequency convergence occurs when $\|D(k^2)\|/k^2 <$ EPSIK.
IRESID	0	Calculate residuals for the potential if IRESID = 1.

1. Controlling pivoting in matrix inversion routines

Variable IPIVOT sets the pivoting strategy used to solve the tridiagonal matrix. No pivoting (IPIVOT = 0) is the fastest, but least accurate, of the methods. No pivoting works reasonably well for most rf cavity problems of moderate size (a few 10K mesh points). More accurate solutions are obtained by the use of either partial pivoting (IPIVOT = 1) or complete pivoting (IPIVOT = 2). In partial pivoting only rows are interchanged, and in complete both rows and columns are interchanged. The accuracy of the solution with partial pivoting is usually very nearly the same as that obtained by complete pivoting. The default value is IPIVOT =1 or partial pivoting.

As an example, we consider an rf cavity problem with 27,430 mesh points, for which the Fish solver obtained RESIK = 4.720×10^{-7} using no pivoting, where RESIK is equal to $|D(k^2)|/k^2$. For the same problem with pivoting, the results were RESIK = 1.730×10^{-11}

for partial pivoting and RESIK = 1.726×10^{-11} for complete pivoting. Relative to the no-pivoting solution, the partial pivoting took 7% longer and complete pivoting took 45% longer. The searching overhead and additional memory requirements make complete pivoting is a more costly strategy. Because partial pivoting works so well, complete pivoting is rarely used in practice.

2. Setting up a frequency scan

Fish and CFish create a plot file named FishScan.TBL containing the results of mode searches and frequency scans. A frequency scan steps through a range of frequencies evaluating the $D(k^2)$ function at each step. Plotting the $D(k^2)$ function after the scan helps locate the frequencies of resonant modes. Resonances occur at roots of $D(k^2)$ where the slope equals −1. The chapter on RF Field Examples discusses a frequency scan example. Another recommended use of this feature is to determine starting frequencies for all the modes of a passband in a multicell cavity.

To set up the scan, enter values for NSTEP, DELFR, and FREQ in the first REG namelist in the Automesh input file and then programs Automesh and Fish or run the Autofish code.

Table VII-2. Frequency-scan variables.

Variable	Description
NSTEP	Number of steps for a frequency scan.
DELFR	Frequency step size for a frequency scan.
FREQ	Starting frequency.

Use program Tablplot to plot the results. If the solution array is real, then frequency is the initial abscissa and $D(k^2)$ is the initial ordinate. If the solution array is complex, then the default initial axis are $\mathrm{Im}[dH_1]$ versus $\mathrm{Re}[dH_1]$ and the plot includes labels at every point giving the frequency corresponding to the point. If you want the initial plot to be $D(k^2)$ versus frequency set variable ScanFormatComplex = No in the [CFish] section of SF.INI. For complex problems, the quantity $\mathrm{Re}[D(k^2)]$ in file FishScan.TBL is not the same as $D(k^2)$ used in by root finder after 3 cycles of a resonance search. The values $\mathrm{Re}[D(k^2)]$ and $\mathrm{Im}[D(k^2)]$ are simply the real and imaginary components of the drive-point field error dH_1 multiplied by the same factors ordinarily used to compute $D(k^2)$. To find resonances for complex problems, look in the complex plane for points dH_1 on the resonance circle that are nearest to the origin.

D. The Superfish root finder

Resonances occur at roots of the $D(k^2)$ function where the slope equals −1 [see the paper by K. Halbach and R. F. Holsinger, “Superfish - A Computer Program for Evaluation of RF Cavities with Cylindrical Symmetry”, Particle Accelerators 7 (1976) 213-222.], where k is the wave number corresponding to frequency f: $k = \omega/c = 2\pi f/c$. The Superfish root finder attempts to locate a slope = −1 root of $D(k^2)$ using polynomial approximations. The root finder has seen several improvements over earlier versions.

1. Root finder in program Fish

If the solution array is real, the quantity $D(k^2)$ is proportional to the field error at the driving point $\delta H_1 = H_{Drive} - H_1$ where H_{Drive} is the driving field and H_1 is the field implied by the six neighboring mesh points surrounding the driving point.

After the first evaluation of $D(k^2)$, the root finder has no information about the slope. It assumes it is near a resonance and extrapolates a slope = –1 line to $D(k^2) = 0$. If the resulting step in k^2 is large, it limits the step to 1% of the present k^2. This 1% step gives reliable information about the local slope of the curve and reduces the chance of missing a mode where the function has a narrow local minimum. After each evaluation of $D(k^2)$, the root finder sorts the available data by k^2 and recalculates first and second derivatives. The first derivative is a weighted average of the slope obtained from the points on either side of each value of k^2. Second derivatives are calculated when enough data becomes available.

Before making any polynomial approximations for the next value of k^2, the code analyzes the available data to place upper and lower bounds on the location of the desired root. Points with k^2 that lie out of bounds are discarded. From the remaining points, the code selects up to three points to calculate a quadratic polynomial. Carefully chosen points speed convergence to the root. Points that qualify for selection include the upper and lower bounds, the last chronological point (if not already selected as an upper or lower bound), and points adjacent to or between the bounds. If there is a choice, the code selects the point whose slope is closest to –1. With three points, it uses a three-point-parabola formula. With only two points available, the code uses a parabola defined by two points plus a slope of –1 at $D(k^2) = 0$. The two-point-plus-slope formula results in four equations with four unknowns. These equations reduce to a single quadratic equation for k^2, and so has two solutions. The code selects a solution based upon properties already learned about the $D(k^2)$ function. For example, it rejects solutions above the upper bound or below the lower bound.

For the three-point-parabola, the root finder evaluates the properties of the parabolic solution. For example, if all the $D(k^2)$ values are positive, a parabola with positive curvature may have no real roots. The minimum of this parabola is likely to be near the desired root of $D(k^2)$. The same is true if the real root with a negative slope is below the lower bound. In these cases the code selects the minimum of the parabola for the next k^2. If both parabolic approximations fail to find a new k^2 consistent with the upper and lower bounds, then the root finder uses one of several contingency procedures. If the direction to the root is known, it simply takes a step toward the root. If upper and lower bounds have been established, it tries the midpoint between the bounds. If the current k^2 is near a slope = +1 root, then the code makes an arbitrary 0.5% step toward smaller k^2. These and other procedures produce new information about the function at the next iteration.

During the resonance search, the code reports information about the method used to generate the next frequency (i.e. wave number k) estimate. Table VII-3 lists the 10 available methods. When using method 10, the code has already determined that $D(k^2)$ crosses the origin with slope = –1. Since the lowest mode must be at a higher frequency, Fish starts the resonance search over at cycle 2 with a new (higher) starting frequency. When using method 4, the code sometimes finds where the line crosses zero, and then steps k^2 by a certain percentage from that point. On early iterations, the code may also

restrict the size of the step in k^2 to avoid losing information about the local slope of the $D(k^2)$ function.

Table VII-3. Methods for estimating the frequency at the next iteration.

Method	Description
1	One-point line with slope of –1 (sometimes –2).
2	The upper (or lower) root of a three-point parabola.
3	The two-point parabola with slope = –1 at $D(k^2) = 0$.
4	Zero crossing of a two-point line between points 1 and 2.
5	Step k^2 by some percentage to get away from the slope = +1 root.
6	The minimum of the three-point parabola.
7	The midpoint between two points.
8	Step k^2 by some percentage from a previous point.
9	The lower root of an inverted three-point parabola.
10	Double the maximum k^2 to get away from the zero-frequency root.

2. Root finder in tuning programs

The automated tuning programs (CCLfish, CDTfish, DTLfish, ELLfish, MDTfish, RFQfish, and SCCfish) use the same procedures described above for program Fish with some modifications described in this section. The tuning codes write information to the log file indicating which method Fish used at each iteration in the following format:

Cycle	FREQ	FREQERR	EPSIK	RESIK	DSTOLER	DSLOPE	Method
1	419.55362	5.44638	2.0E–02	2.5E–02	±0.40	1.00000	1
2	419.26865	5.73135	2.0E–02	1.4E–03	±0.40	–0.94816	2

On the first iteration, DSLOPE is whatever value was supplied by the tuning code (usually +1 on the first Superfish run on a problem, and –1 on subsequent runs). FREQ is the current estimate of the resonant frequency. FREQERR is the absolute difference between FREQ and the target frequency for the problem. RESIK is equal to $|D(k^2)|/k^2$. EPSIK and DSTOLER are used in tests for termination. EPSIK and DSTOLER have default values of 0.0001 and 0.02, respectively. However, the tuning codes relax these values when FREQERR lies within the ranges listed in Table VII-4 provided that the root finder has used method 1, 2, or 3. These three methods are likely to produce a reasonably accurate estimate of the frequency, whereas the other methods are attempts to get close enough to resonance eventually to use either method 2 or method 3. In Table VII-4, δf is the frequency tolerance specified in the tuning code's control file.

We have empirically determined that this scheme saves considerable computation time while the tuning code adjusts the geometry in an effort to tune a cavity. The farther the cavity is from the target frequency, the less accurately the code needs to know the resonant frequency at intermediate steps.

Table VII-4. Relaxed convergence parameters used by the tuning programs.

FREQERR Range	EPSIK	DSTOLER
FREQERR > 1000δf	0.2	0.8
1000δf ≥ FREQERR > 100δf	0.02	0.4
100δf ≥ FREQERR > 10δf	0.002	0.2
10δf ≥ FREQERR > 2δf	0.0004	0.1
FREQERR ≤ 2δf	0.0001	0.02

3. Root finder in program CFish

When the solution array is complex then the field error at the driving point δH_1 is also complex. Near an isolated resonance δH_1 lies on a circle in the complex plane and resonance occurs at the point on the circle closest to the origin. In the complex root finder $D(k^2)$ for a point δH_1 in the complex plane is the perpendicular distance from the point to the line containing both (0,0) and the center of the circle. $D(k^2)$ versus k^2 has a negative slope at the intersection of the circle with this line. The complex root finder cannot evaluate $D(k^2)$ properly until it has some information about the resonance circle, which requires at least 3 points for different values of k^2.

When CFish starts a resonance search, it first checks whether the solution will be real or complex (you can run CFish on real problems). If there are no complex materials in the problem, then the solution is real and the code follows the same procedures described above for the Fish root finder. If the solution is complex, then the code completes at least 3 cycles before the first test for convergence. For the first two cycles, CFish uses the real part of δH_1 to compute an approximate value of $D(k^2)$. After cycle 3, CFish fits a circle to the δH_1 data from all completed cycles and calculates $D(k^2)$ for each point. The root finder then makes polynomial approximations to find the point where $D(k^2)$ crosses zero with a negative slope in the same manner as the real root finder.

Note that $D(k^2)$ of a given point will change from one cycle to the next if the fitted circle parameters change. For each fitted circle, CFish writes the center of the circle and its radius to file OUTFIS.TXT. For a diagnostic, CFish can write a plot file showing all the fitted resonance circles from a mode search. If variable CFishCirclePlot = Yes in the [CFish] section of SF.INI, CFish writes an input file for program Quikplot called CircleFit.QKP. The code discards any existing copy of this file and starts a new one for each resonance search. File CircleFit.QKP contains points $(R\cos\theta, R\sin\theta)$ along the fitted circle of radius R between θ =45 degrees and θ = 180 degrees. Each curve is in a separate Data/EndData section labeled by the cycle number in the search. The file also contains a separate curve for each fitted circle showing the line between (0,0) and the center of the resonance circle. The file does not contain the δH_1 data from the search, but you can add a separate Data/EndData section containing the points $[\mathrm{Re}(\delta H_1), \mathrm{Im}(\delta H_1)]$ from the resonance search or from previous frequency scans to see where these points lie relative to the fitted resonance circles. If a search needs more than a few cycles, you may need to increase the value of MaxDataSets in the [Quikplot] section of SF.INI to view all the fitted circles.

Finding a resonance successfully for a complex problem requires some user oversight of the procedure. Watch the resonance search progress reported in file OUTFIS.TXT. The

code reports the percent RMS error in the radius for the points used in the circle fit to $\mathrm{Im}(\delta H_1)$ versus $\mathrm{Re}(\delta H_1)$. This quantity is typically very small ($<10^{-5}$). A large RMS radius error may indicate that there are multiple resonances in the frequency range and the root finder has moved from one resonance circle to another. We recommend that you complete a frequency scan over the region of interest and examine the plot of $\mathrm{Im}(\delta H_1)$ versus $\mathrm{Re}(\delta H_1)$ contained in the Tablplot file FishScan.TBL. Tablplot can label individual points by frequency to help identify the locations of resonant modes and choose a starting frequency for a resonance search. Because CFish does not require that the slope of $D(k^2)$ be within a certain tolerance of −1 (see the next section), you may wish to use a smaller setting for variable EPSIK than the default value 0.0001.

E. Convergence criteria in the Fish and CFish solvers

Resonances occur at roots of the $D(k^2)$ function where the slope equals −1. For convergence to a resonant frequency, the Superfish root finder requires that the $D(k^2)$ function is small enough and has the correct slope. If the solution array is real, the convergence requirements are:

$$\frac{\left|D\left(k^2\right)\right|}{k^2} < \mathrm{EPSIK},$$

$$\frac{\left|\delta\left(k^2\right)\right|}{k^2} < \mathrm{EPSIK},$$

and

$$-1.02 < \frac{dD}{d\left(k^2\right)} < -0.98,$$

where the default value of EPSIK is 0.0001. The notation $dD/d(k^2)$ refers to the first derivative (or slope) of the $D(k^2)$ function. The term $\delta(k^2)$ is the proposed change in k^2 calculated before the next evaluation of $D(k^2)$. RESIK is equal to the quantity $|D(k^2)|/k^2$. Parameters in Table VII-5 affect the rate of frequency convergence when the Fish and CFish solvers are performing a resonance search. Enter values for these parameters in the REG namelist section of the Automesh input file.

The solver relaxes slightly the requirement on the slope if the magnitude of $\delta(k^2)/k^2$ is already many times smaller than EPSIK. Consider a case using the default tolerance of ±2% on the requirement that the slope be −1. If $\delta(k^2)/k^2$ is already 10 to 100 times smaller than EPSIK, the code uses ±4% tolerance on the slope, and if $\delta(k^2)/k^2$ is more than 100 times smaller, it uses ±8% tolerance. This addition to the tests allows the code to stop one iteration earlier when it is clear that solution is already well within the requested accuracy. This situation can arise when the solver gets very close to the final frequency using previous points relatively widely spaced in frequency and the accumulated data is not finely spaced enough for an accurate calculation of the slope. In more detail, the termination criteria consist of:

$\frac{\left|D\left(k^2\right)\right|}{k^2} < \mathrm{EPSIK}$, and satifying both conditions on one of the following lines:

$$\frac{\left|\delta\left(k^2\right)\right|}{k^2} < \text{EPSIK}, \text{ and } -1-\text{DSTOLER} < \frac{dD}{d\left(k^2\right)} < -1+\text{DSTOLER},$$

or $$\frac{\left|\delta\left(k^2\right)\right|}{k^2} < \text{EPSIK}/10, \text{ and } -1-2(\text{DSTOLER}) < \frac{dD}{d\left(k^2\right)} < -1+2(\text{DSTOLER}),$$

or $$\frac{\left|\delta\left(k^2\right)\right|}{k^2} < \text{EPSIK}/100, \text{ and } -1-4(\text{DSTOLER}) < \frac{dD}{d\left(k^2\right)} < -1+4(\text{DSTOLER}).$$

If the solution array is complex, the convergence criteria differ from the above in two ways. First, CFish completes at least 3 cycles before the first test for convergence, and second, CFish does not require that the slope of $D(k^2)$ be within a certain tolerance of –1. The code ignores the quantity DSTOLER, but it still requires that $dD/d(k^2)$ is negative.

For certain types of problems, the default setting of EPSIK = 0.0001 may not be demanding enough. An example of such a situation is a very long, multicell drift-tube linac cavity generated by the code MDTfish. If the Fish solver completes only one iteration (see the discussion below on setting DSLOPE = –1) the field distribution may be in error by a small amount and can also depend on placement of the drive point. To ensure an accurate result, you can include a smaller value (e.g. 10^{-6} or 10^{-8}) for EPSIK in the Automesh input file. There are several ways to check the accuracy of the resulting fields. One suggestion for a multicell MDTfish problem is to compute the cell-by-cell E_0 integrals for two runs with different drive-point locations and plot the results versus cell number. Another possibility would be compare SF7-interpolated fields along the same path, again for two cases with different drive points.

F. Variables calculated by Fish and CFish

Fish and CFish calculate a few quantities that it lists near the end of OUTFIS.TXT. These values will appear in the variable list in each subsequent code's output file for your information.

1. DSLOPE, the slope of the $D(k^2)$ function

Fish and CFish report DSLOPE, which is the slope of the $D(k^2)$ function. If the root finder completed more than one iteration, then DSLOPE is the calculated value at the end of the last iteration. If the code finished only one iteration, then DSLOPE retains its initial value. There are two common circumstances that result in only one iteration. Tuning programs set DSLOPE = –1 when the starting frequency is known to be near resonance. If the other convergence criteria were satisfied after the first iteration, the code will stop. Also, if MAXCY = 1, then the code stops after one cycle regardless of the convergence criteria. A typical use of this method is in CFish waveguide problems.

Table VII-5. Parameters that affect convergence in Superfish.

Variable	Description
EPSIK	Convergence parameter for frequency convergence. The default value of EPSIK is 0.0001.
MAXCY	Maximum number of cycles or evaluations of the $D(k^2)$ function allowed before giving up on a resonance search. The default value is 19. You can stop the search at fewer cycles or allow more cycles by changing MAXCY.
DSLOPE	Slope of the $D(k^2)$ function. During the resonance search, the code does not know the slope after only one iteration. Automesh initializes DSLOPE to +1. Since the root finder does not calculate the slope until the second iteration, this procedure forces Fish or CFish to perform another iteration. If you know that the starting frequency is near resonance, set DSLOPE = –1 to allow the code to converge in only one iteration. The example input file SPHERE for a spherical cavity sets DSLOPE = –1 to stop after the first iteration.
DSTOLER	Tolerance on slope of $D(k^2)$ for convergence. The default value is 0.02. That is, the value of DSLOPE must be between –1.02 and –0.98 for convergence. You can tighten or relax this requirement by specifying a value for DSTOLER within the range from 0.002 to 0.5. For example, to accept the solution when DSLOPE is within 15% of –1, set DSTOLER = 0.15. Refer to section VII.E for more details on how the code uses variable DSTOLER. CFish ignores DSTOLER if the solution array is complex and instead only requires a negative slope of $D(k^2)$ in addition to the other convergence criteria.

2. FREQ, the resonant frequency

Variable FREQ is the frequency in MHz at the end of the last iteration. The code recalculates the frequency after adjusting k^2 before each iteration of the root finder. FREQ is also the starting frequency for a resonance search or a frequency scan.

3. XK0 and XKSQ, the wave number k and its square k^2

Variable XKSQ is the value of k^2, the square of the wave number, at each iteration of the resonance search. XK0 is k (calculated in the code by taking the square root of XKSQ). The code uses the following statement to get XKSQ:

```
XKSQ = (2.0D0*PI*FREQ/CLIGHT)**2.
```

4. DKSQ, the proposed change in k^2 at the last iteration

Variable DKSQ is the term $\delta(k^2)$ used in the tests for convergence. During the resonance search, DKSQ is the proposed change in k^2 calculated before the next evaluation of $D(k^2)$.

5. ERG, the Fish and CFish stored-energy integral

Variable ERG is a number proportional to the rf field energy after the last iteration. Program SFO uses ERG to calculate the stored energy after it finds the scale factor ASCALE that corresponds to the desired field normalization. SFO reports the stored energy in Joules as ENERGY. For problems in Cartesian coordinates (ICYLIN = 0),

$$\text{ERG} = \iint H^2 dx\, dy\,, \text{ and} \qquad \text{ENERGY} = \frac{(\text{ASCALE})^2\,\text{ERG}}{4\times 10^{-9}(\text{CLIGHT})^2}\,.$$

For cylindrically symmetric problems (ICYLIN = 1),

$$\text{ERG} = \iint H^2 r\, dr\, dz\,, \text{ and} \qquad \text{ENERGY} = \frac{(\text{ASCALE})^2\,\text{ERG}}{8\pi\times 10^{-9}(\text{CLIGHT})^2}\,.$$

In these equations CLIGHT is the speed of light in cm/sec. Fish and CFish perform the integration over the entire problem geometry.

6. RESIDA, the residual of the solution matrix

Variable RESIDA is the residual of the solution matrix for the field H. The default is not to calculate this quantity. If you want the code to calculate the residual for the solution, set IRESID = 1.

7. RESIK, the ratio $|D(k^2)|/k^2$ at the end of the last iteration

Variable RESIK is equal to $|D(k^2)|/k^2$ at the end of the last iteration the quantity. It is one of the quantities used in the tests for convergence.

8. ICYCLE, the number of the last completed iteration

Variable ICYCLE is the iteration number on which the root finder stopped. If ICYCLE is equal to MAXCY, then the solution may not have converged to resonance.

9. Q2I, the ratio 1/2Q calculated by CFish for lossy materials

Q2I is equal to the reciprocal of 2Q calculated by CFish, where Q is the quality factor related to the power losses in materials with complex permittivity or complex permeability. Program SFO uses Q2I to calculate the power dissipated in the lossy materials after it finds ENERGY, the stored energy in Joules. SFO uses the field interpolator to perform an independent calculation of the complex material power losses. SFO reports the following result for the CFish power loss for comparison with the result using the field interpolator:

PLOSS = 2*Q2I*XK0*CLIGHT*ENERGY.

G. Fish and CFish exit error codes

Table VII-6 lists the Fish and CFish error messages. The error numbers also are available as exit error codes to tuning programs and batch control files.

Table VII-6. Fish and CFish error messages.

Error	Description
200	Solution file specified on command line does not exist.
202	File specified on command line does not appear to be a Poisson Superfish solution file.
204	File entered in browse dialog window does not exist.
207	An error occurred reading the first record from the Poisson Superfish solution file.
208	Solution arrays have not been properly declared. Please report this error.
209	The dimension of at least one solution array is too small. Please report this error.
210	Mesh point arrays are not large enough. Please report this error.
211	The problem has no drive point.
212	Drive point is on a Dirichlet boundary where there is no magnetic field.
213	The mesh has overlapping or zero-area triangles.
214	Subroutine RDTAPE35 expects to return a REAL solution array, but calling code expects a COMPLEX array. Please report this error.
215	Subroutine RDTAPE35 expects to return a COMPLEX solution array, but calling code expects a REAL array. Please report this error.
216	Unable to read triangular mesh data (possibly incompatible code versions).
220	Reached the end of the solution file unexpectedly. Rerun problem stating with Automesh.
221	An error occurred reading a record from the solution file. Please report this error.
222	The Poisson Superfish solution file is obsolete. Rerun problem stating with Automesh.
224	Cannot run Fish or CFish on Poisson problems.
225	Missing material data. This error should usually be detected by Automesh.
226	Cannot run Fish on problems with complex material properties.
293	Insufficient memory, cannot allocate MT namelist arrays for a Superfish problem.
501	Insufficient memory, cannot allocate arrays for binary solution file data.
502	Insufficient memory, cannot allocate arrays for weighting factors.
503	Insufficient memory, cannot allocate arrays for the tridiagonal submatrices.
504	Insufficient memory, cannot allocate arrays for the root finder results.
505	Insufficient memory or disk space for temporary data and no valid drives available.
506	Insufficient memory or disk space for temporary data.
508	Insufficient memory, cannot allocate an array needed to invert the tridiagonal matrix..
510	No starting frequency has been supplied.
511	The starting frequency is zero.
512	The starting frequency is negative.
513	The drive point weighting factor is zero.
514	Exceeded the current maximum number of cycles for the resonance search.
515	The cavity stored energy is zero. Possible problem with drive point location.

VIII. Poisson and Pandira, Static Field Solvers

Poisson and Pandira solve static magnetic field problems or electrostatic problems. Poisson uses the method of successive over relaxation. Pandira uses a direct matrix inversion method and can handle permanent magnet material. Poisson and Pandira must be run after Automesh. The codes run in the following order:

- Automesh
- WSFplot (optional)
- Poisson or Pandira
- WSFplot(optional)
- SFO, SF7, or Force

A. Starting and stopping Poisson and Pandira

to start the static solver programs, right-click on the binary solution file, which has extension T35, and select either "Run Poisson" or "Run Pandira." You also can use a command line (for example, in batch files) to start the code:

Poisson *T35file*
Pandira *T35file*

where *T35file* is the name of a binary solution file. It is usually not necessary to enter the name of the *T35file* immediately after running Automesh because the code will find the name of the solution file to open in file TAPE35.INF. If the file listed in TAPE35.INF either does not exist or was not created recently enough, the code opens the standard Open dialog window. For more details about startup options refer to the section titled Opening input files on startup.

1. Using the Esc key to stop Poisson or Pandira early

Type END to stop Poisson or Pandira at any program prompt. During the over-relaxation calculation in Poisson or during the tridiagonal matrix solution in Pandira, you can press the Esc key to stop the program. You have three options. The choices are:

- Finish on cycle XX (Poisson), or after the present iteration (Pandira), save the solution, then stop normally.
- Continue the calculation.
- Stop now without saving the solution (the default).

Poisson replaces XX above with the next cycle number at which it checks for convergence (usually a multiple of 10). Choosing Stop returns exit error code 56.

B. Files and filename conventions

Poisson and Pandira read the binary solution file created by Automesh. They append the solution to this file. The codes also write an output summary to file OUTPOI.TXT or file

OUTPAN.TXT. If the ParmelaFields option in configuration file SF.INI has been selected, then Poisson or Pandira also writes a file with extension T7, which contains fields interpolated for input to the Parmela program. (Note: Running postprocessor SF7 provides a more convenient method for producing this file.) The codes also write Quikplot input files EFLD.QKP (for electrostatic field problems) or BFLD.QKP (for magnetostatic field problems). See Files and filename conventions for more information.

Depending upon your computer's memory configuration, Pandira may write a temporary file on one of your hard drives. (Poisson does not create the temporary data arrays because it solves the problem by successive over-relaxation, and not by inverting the tridiagonal matrix.) Pandira will use memory for temporary data arrays if possible. If there is insufficient memory, the program creates the scratch file on the hard drive with the most free space. If it cannot find enough free disk space, the program stops with a message telling you how much extra space it needs on one of your drives. For more information, refer to the section on Temporary data arrays.

C. Data entries for Poisson and Pandira

Table VIII-1 lists the variables used in programs Poisson or Pandira. Supply these variables in the first REG namelist of the Automesh input file. For the complete list, see Poisson Variables.

Table VIII-1. Variables used by Poisson and Pandira.

Variable	Default value	Description
ANGLE	0.0	Extent of the arc for interpolating the potential.
ANGLZ	0.0	Initial point on arc for interpolating the potential.
BDES	1.0D15	If not default value, adjust XJFACT so |B| = BDES.
CONV	1.0	Length conversion (number of cm/unit).
EPSILA	5.0D–7	Convergence parameter for air and interface points.
EPSILI	5.0D–7	Convergence parameter for iron points.
FIXEPS	9.0	Permittivity if MODE = –1 (XJFACT = 0).
FIXGAM	0.004	Reluctivity if MODE = –1 (XJFACT ≠ 0).
ICAL	0	Selects formula for computing initial mesh-point currents.
ICYLIN	0	Coordinate system (0: rectangular; 1: cylindrical).
ICYSEN	0	If zero, Poisson does not print H·dl integrals.
IENERGY	–1	Calculate stored energy if 1.
IHDL	100000	Number of cycles in Poisson between H·dl integrals.
IPERM	0 or 1	0 = real currents; 1 = permanent magnets only.
IPIVOT	1	Controls pivoting in Pandira matrix inversion.
IPRFQ	0	Print frequency during Poisson iterations.
ISKIP	1	Poisson cycles between recalculation of permeabilities.

Table VIII-1. Variables used by Poisson and Pandira. (continued)

Variable	Default value	Description
IVERG	10	Number of cycles between Poisson convergence tests.
KBZERO	1	Logical K coordinate where |B| = BDES.
KMIN	1	Lower K bound for computing the field and gradient.
KTOP	KMAX	Upper K bound for computing the field and gradient.
KTYPE	2	Symmetry indicator for harmonic analysis.
LBZERO	1	Logical L coordinate where |B| = BDES.
LMIN	1	Lower L bound for computing the field and gradient.
LTOP	1	Upper L bound for computing the field and gradient.
MAXCY	100000	Maximum number of cycles (default 20 in Pandira).
MODE	–2, –1	Material property indicator.
NOTE	1	Determines relaxation order for the mesh points.
NPTC	0	Number of arc points for interpolating the potential.
NTERM	5	Number of coefficients in the harmonic analysis.
OMEGAP	0.001	Used in optimizing the over-relaxation parameter.
RHOPT1	1.9	If equal to RHOAIR, causes optimization of RHOAIR.
RHOAIR	1.9	Over-relaxation factor for air and interface points.
RHOFE	1.0	Over-relaxation factor for iron points.
RHOGAM	0.08	Under-relaxation factor for the reluctivity in Poisson.
RINT	0.0	Radius of the arc for interpolating the potential.
RNORM	RINT	Aperture radius used in the harmonic analysis.
STACK	1.0	Stacking or fill factor for iron regions.
XAZERO	none	Physical X coordinate where A = 0 for harmonic analysis.
XBZERO	0	Physical X coordinate where |B| = BDES.
XJFACT	1.0	0 for electric fields; or scale factor for currents.
XJFEND	1.0	Ending XJFACT for XJSTEPS additional calculations.
XJSTEPS	0	Number of steps between XJFACT and XJFEND.
XJTOL	1.0D–7	Tolerance on XJFACT when setting |B| = BDES.
XMAXF	0.0	Upper X bound for computing the field and gradient.
XMINF	0.0	Lower X bound for computing the field and gradient.
YAZERO	none	Physical Y coordinate where A = 0 for harmonic analysis.
YBZERO	0	Physical Y coordinate where |B| = BDES.
YMAXF	0.0	Upper Y bound for computing the field and gradient.
YMINF	0.0	Lower Y bound for computing the field and gradient.

D. Input options for Poisson and Pandira

The next few sections discuss some of more commonly used input options for Poisson and Pandira. The features include:

- using the internal permeability tables,
- using the stacking-factor correction,
- entering current-filament data,
- scaling to a desired value of the magnetic field somewhere in the mesh,
- initializating permanent-magnet problems, and
- solving for electrostatic fields.

1. Permeability tables

Table VIII-2 lists variables that control the use of permeability data in Poisson and Pandira. The default value for MODE depends on the value of XJFACT. For nonzero XJFACT, which indicates a magnet problem, the default value is MODE = –2. For electrostatic problems with XJFACT = 0, the default value is MODE = –1. The codes use the relative permittivity ε_r (for electrostatic problems) and the relative permeability μ_r (for magnetostatic problems), which are dimensionless numbers defined as follows:

$$\varepsilon_r = \frac{\varepsilon}{\varepsilon_0} \text{ and } \mu_r = \frac{\mu}{\mu_0}.$$

When MODE is –2, the codes assume infinite permeability for magnetic materials. The setting MODE = –2 is not allowed for electrostatic problems. When MODE is –1, then for materials 2 and higher, FIXGAM is the default value of the reluctivity γ (= $1/\mu$) in magnet problems, and FIXEPS is the default value ε_r in electrostatic problems. A value of GAMMA or EPSILON in the MT namelist for a material takes precedence over FIXGAM or FIXEPS, respectively. When MODE is zero, then the codes can use material properties from either the internal tables or from user-supplied tables. You can enter your own table of permeability data in various formats in MT namelist sections in the Automesh input file.

Table VIII-2. Variables related to permeability in Poisson and Pandira.

Variable	Default value	Description
MODE	–2, –1	Specifies permeability data in Poisson and Pandira.
STACK	1	Default stacking factor for all iron regions.
FIXGAM	0.004	Default reluctivity for all materials (XJFACT ≠ 0).
FIXEPS	9.0	Default permittivity for all materials (XJFACT = 0).

There are currently for internal permeability tables available in Poisson and Pandira. Internal tables are numbered in reverse order starting at zero. Figure VI-1 shows a plot of the B-H curves on a linear scale for the tables with MTID = 0, –1, –2, and –3. The table with MTID = –1 for “decarburized iron” has been in the code for many years. According to comments in the code, it came from someone with the initials J. D. The highest magnetic field in this table is about 75 kG, which is well into the saturated region. We doubt if the last 10 or so points on this curve represent actual measurements. Older versions of Poisson and Pandira did not correctly extrapolate beyond the end of a permeability table and the code would sometimes crash if the fields were high enough to saturate the iron. We suspect that someone computed a few points by linear extrapolation from the measurements and added them to the internal table. This expediency probably prevented problems when running the codes.

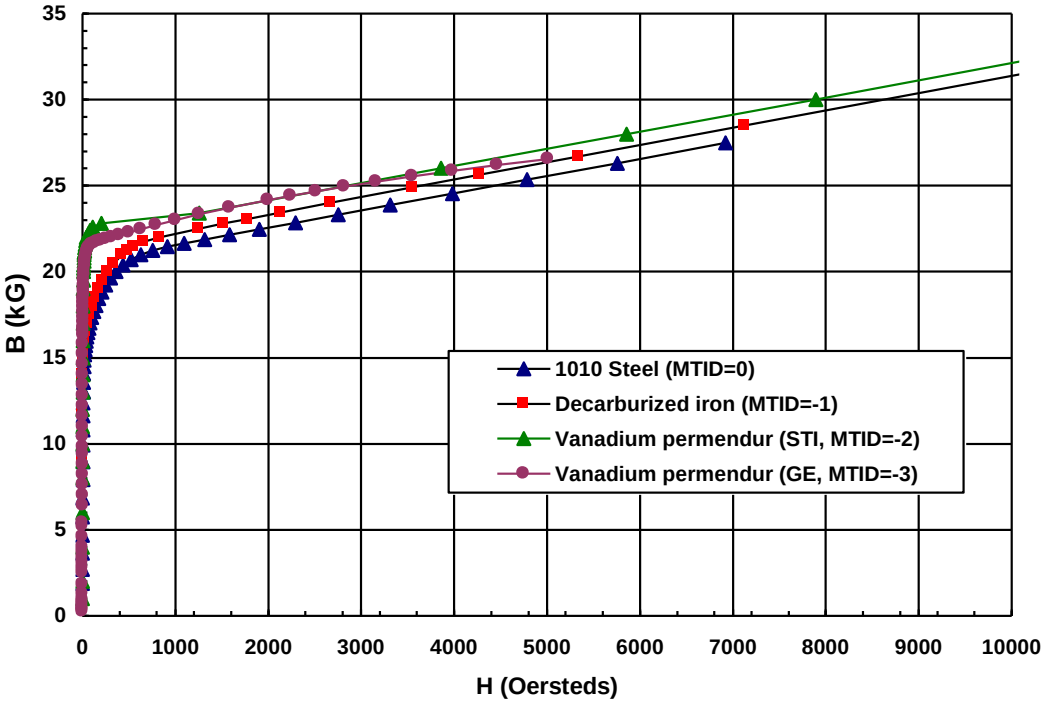

Figure VIII-1 B-H curves (linear scale) for the internal permeability tables.

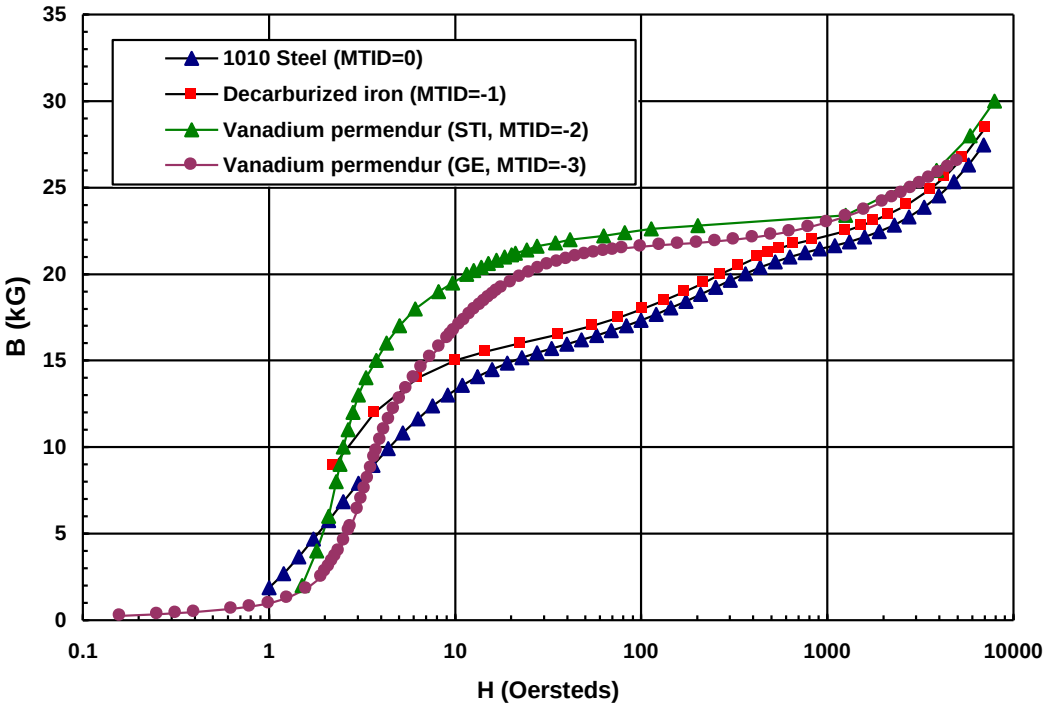

Figure VIII-2. B-H curves (log scale) from Figure VIII-1.

Figure VIII-2 shows the data from Figure VIII-1 with H plotted on a log scale. Small differences are apparent between the MTID = –1 table and the newer MTID = 0 table for 1010 steel. The data for 1010 steel were supplied to the Los Alamos Accelerator Code Group in 1990 by Dick Early at the Stanford Linear Accelerator Center (SLAC). The highest magnetic field in the 1010 table is 27.4772 kG. The codes use the 1010-steel data if MODE is zero and the REG namelist for the material contains no MTID number.

The vanadium permendur table with MTID = –2 came from STI Optronics Inc. (2755 Northup Way, Bellevue, WA 98004-1495). STI Optronics anneals their samples at 840 C for at least 10 hours in an inert atmosphere or vacuum. Parts are cooled in the same environment at a rate no larger than 100 C/hour. The furnace remains closed until the temperature reaches 150 C or lower. Steve Gottschalk of STI points out that if the temperature exceeds 870 C, then an austenitic phase forms that could ruin the parts.

The vanadium permendur table with MTID = –3 came from a widely used graph of B-H curves published by General Electric Co. (Schenectady, NY) in 1955.The graph is titled "DC Magnetization Curves for Various Magnet Materials" and has the identifying number MMS-55-0730.

2. Listing the fast-interpolation material tables

A setting in file SF.INI allows you to plot the material tables that the codes produce for fast interpolation during the solution. The codes will produce Tablplot input files with names of the form BGammaxx.Tbl, where xx is the material ID number containing the data. The filenames for the internal tables with negative material ID numbers will be of the form BGamma_xx.Tbl, where xx is the absolute value of the ID number. These tables have equal B intervals generated either from an internal table or from a user-supplied table. To produce these plot files while running Poisson. use the following lines in SF.INI:

```
[Poisson]
PrintBGammaTables = Yes
```

The fast-interpolation tables are linear interpolations of the (B,γ) data. For tables supplied originally as either (B,γ) or (B,μ_r) data, the codes interpolate γ at a given B. For tables supplied as (B,H) data, the codes interpolate H at a given B and then calculate γ = H/B.

If you are supplying your own material table, this feature provides a way to check the data for potential problems. For example, when running Tablplot, select H for the abscissa and μ_r for the ordinate. If the curve is double valued, it indicates a problem with the input data.

3. Using the stacking-factor correction

The stacking factor provides a method for dealing with regions of iron that contain some nonpermeable material. Poisson and Pandira treat these regions as if they had an effective value of B and γ derived from the properties of the solid material. Solid materials have stacking factor s = 1 and laminated materials have s < 1. The value of s specifies the fraction of the volume occupied by the permeable material. An example is a series of laminated slabs of iron with thin air spaces between the slabs. The stacking factor correction assumes that the magnetic field is parallel to the plane of the iron laminations.

The parameter STACK is the default value of the stacking factor for all materials with MAT >2. The default value of STACK is 1.0. The REG namelist variable STACKING is the stacking factor for a particular material. For example, suppose a problem contains several regions with material number MAT = 3 and this material has s = 0.95. The setting STACKING = 0.95 on any region with MAT = 3 sets the stacking factor for all regions that use material 3. The last region with a given MAT and STACKING takes precedence.

In a solid permeable material, the following relationships hold among the field H, the magnetic induction B, the magnetization M, and the reluctivity γ:

$$B = H + M(H), \text{ and } B = H/\gamma.$$

A solid material's permeability table consists of a series of (B,γ) pairs. For a partially filled region with stacking factor s, Poisson and Pandira convert the (B,γ) table into a table of (B,γ_m) pairs, where γ_m is a modified value of γ:

$$\gamma_m = \frac{1}{s/\gamma + (1-s)}.$$

The following derivation of this correction was provided by Richard K. Cooper: Consider a stack of iron laminations, all of width d, each lamination separated from the next by a gap g. Now further imagine that each lamination carries a magnetic field of strength B_i. The fields are assumed to be in the plane of the laminations. Inside the iron the H field is B_i/μ, or

$$B_i = \mu H = \frac{\mu_0 H}{\gamma},$$

where $\gamma = \mu_0/\mu$. But tangential H is continuous, so in the gaps (assumed to be air) the value of H is the same, implying that in the gaps the magnetic field is

$$B_{gap} = \frac{\mu_0 B_i}{\mu} = B_i \gamma,$$

Now one can calculate the average B field as

$$B_{avg} = \frac{B_i d + B_{gap} g}{g + d}.$$

Now let the stacking factor s = d/(g + d). Then g/(g + d) = 1 – s. The average magnetic field can be written

$$B_{avg} = s B_i + (1-s) B_{gap}.$$

Writing the right-hand side in terms of H we have

$$B_{avg} = s\mu H + (1-s)\mu_0 H = \mu_0 \left[s/\gamma + (1-s) \right] H,$$

or $$B_{avg} = \frac{\mu_0 H}{\gamma_m},$$

where $$\gamma_m = \frac{1}{s/\gamma + (1-s)}.$$

So to get the correct relationship between H and the average value of B all one has to do is to modify γ in the permeability table. These equations have the correct asymptotic behavior, namely:

for $$s \to 1, \quad \gamma_m \to \gamma, \quad \text{and} \quad B_{avg} \to B_i,$$

and for $$s \to 0, \quad \gamma_m \to 1, \quad \text{and} \quad B_{avg} \to B_{gap}.$$

Note that Poisson does not store or use H in the calculations. A permeability table consists only of (B,γ) pairs. In the MT namelist section of the Automesh input file, you can enter other data such as (B,H) or (B,μ_r) where $\mu_r = \mu/\mu_0$. However, the code converts these entries to (B,γ) for use in the calculations.

The manner in which Poisson uses the modified permeability table is as follows. At each iteration, the code has the present estimate of A (the vector potential). From A, it calculates values of B_{avg} using the appropriate derivatives for the curl of A. For a material with s < 1, the actual B in the iron is higher than B_{avg} and the actual B in the air is lower than B_{avg}. For a better estimate of γ (in the iron) at the next iteration, the code looks up γ_m at $B = B_{avg}$ in the modified table. This procedure returns γ for the present best estimate of the field in the iron, such that H is constant in both air and iron. The value of γ_m computed for a given B_{avg} is higher than the original γ in the table for solid material because B in the iron is higher than B_{avg}.

For example, suppose that a high-permeability magnet is made with stacking factor s = 0.5. The code would look up a value for γ_m that would be approximately 2γ for $\gamma << 1$. This is the correct γ for the iron in the laminations, which corresponds to $B_i/2B_{avg}$ and a negligibly small value of B_{gap} compared to B_i.

4. Adjusting fields for a particular value on a mesh point

Table VIII-3 lists variables that are concerned with scaling the solution to a desired value of the magnetic field on a particular point in the mesh. BDES is desired value of the magnetic field at the physical point XBZERO, YBZERO, which has logical coordinates KBZERO, LBZERO. An example file HTEST2.AM shows how to use this feature. Older versions of the code only allowed entry of the logical coordinates KBZERO, LBZERO. Poisson adjusts the scale factor XJFACT to achieve the result. (Note that XJFACT = 0.0 signals an electrostatic problem with a scalar potential.) This field adjustment feature is not available in Pandira.

Table VIII-3. Variables for scaling the Poisson fields.

Variable	Default value	Description
BDES	FINDEF	Desired value of B at logical coordinates KBZERO,LBZERO.
XBZERO	FINDEF	Physical X coordinate where a particular B is desired.
YBZERO	FINDEF	Physical Y coordinate where a particular B is desired.
KBZERO	1	Logical K coordinate where a particular B is desired.
LBZERO	1	Logical L coordinate where a particular B is desired.
XJFACT	1	Factor multiplying currents and current densities.
XJTOL	1.0D−7	Tolerance on XJFACT when adjusting for a desired B.

Automesh initializes variables BDES, XBZERO, and YBZERO to the unphysical value FINDEF = −1.0D+99, which the codes use to tell whether the user has supplied a value. The default mesh point is KBZERO, LBZERO = 1, 1. If you specify a value for BDES, then Poisson will begin adjusting XJFACT after the residual RESIDA falls below 20*EPSILA. The default value of EPSILA is 5.0×10^{-7}. When XJFACT changes by less than XJTOL, Poisson stops adjusting XJFACT, but continues the over-relaxation calculation until it satisfies the usual convergence criteria. The default value of XJTOL is 1.0×10^{-7}. Enter values for these variables in the first REG namelist of the Automesh input file. Directory LANL\Examples\Magnetostatic\H-Magnet includes the file HTEST2, which illustrates entries for the parameters BDES, XBZERO, and YBZERO.

5. Computing a series of solutions for a range of current multipliers

For magnet problems, Poisson and Pandira can compute a series of solutions for different values of the current multiplier XJFACT. Variable XJFACT itself serves as the starting value for the series. REG namelist variable XJFEND is the ending value and XJSTEPS is the number of steps in the series after the first calculation.

This feature is incompatible with the option that adjusts XJFACT discussed in the previous section. If entries in the first REG namelist in the Automesh input file select both of these options, Automesh will stop with an error message. Also, Automesh will not allow XJFACT and XJFEND to have opposite signs, thus eliminating the possibility that one of the runs will have XJFACT = 0 signaling an electrostatic problem.

The first calculation results in the usual set of output files. Files OUTPOI.TXT or OUTPAN.TXT contain a list all the problems constants, a log of each iteration, and a table of requested interpolated and fields. If the requested interpolation was along a line, then the codes produce files OUTPOI.TBL or OUTPAN.TBL, which are input files for plotting program Tablplot. The solution file has the same name as the Automesh input file, but with extension .T35. If PROB.AM is the same of the Automesh input file, then files for other calculations in the series have the following form:

PROB_1.TXT, PROB_2.TXT, PROB_3.TXT, …
PROB_1.TBL, PROB_2.TBL, PROB_3.TBL, …
PROB_1.T35, PROB_2.T35, PROB_3.T35, …

6. Some properties of permanent-magnet materials

Readers may find the following information about permanent magnets useful. We quote from a Hitachi Magnetics Corp. brochure from the 1970s.

Figure VIII-3 illustrates some relationships among properties of a permanent magnet. The rectangular hysteresis loop (dashed line) represents the relationship between the magnetic potential or field strength H and the resulting intrinsic magnetization or flux density B. The other hysteresis loop (solid line) is the material's normal induction curve, which is obtained by adding the value of the magnetic potential to the value of the intrinsic magnetization at every point. When a permanent magnet creates a field in an air gap, the induction curve determines the volume and geometry of the magnet. The magnet operates in the second quadrant of the induction hysteresis loop. When a permanent magnet reacts with an external field, the intrinsic magnetization curve gives the force or torque on the magnet, because the force results from an interaction between the magnetization and the external field.

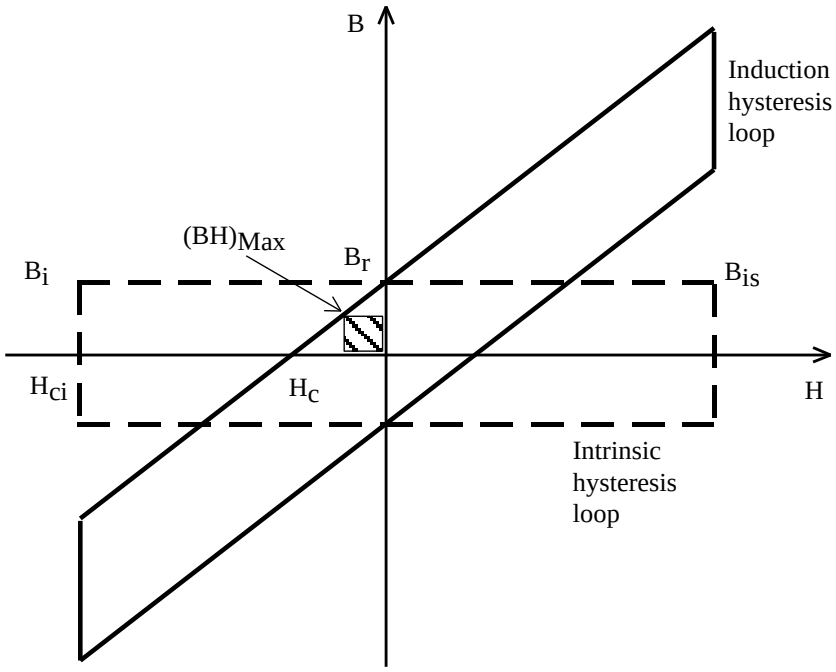

Figure VIII-3. Flux density B versus field strength H.
The dashed line is the intrinsic hysteresis loop and the solid line is the induction loop.

Several important material properties appear in Figure VIII-3. The residual magnetization B_r is the magnetization level at zero applied field. The intrinsic level of resistance to complete demagnetization is called its intrinsic coercive force H_{ci}. The coercive force H_c is the induction level of resistance to complete demagnetization. The maximum available energy product $(BH)_{Max}$ is a useful volumetric figure of merit. It represents the point on the induction curve where the product of B and H is a maximum. The level of the intrinsic saturation magnetization B_{is} establishes the material's basic boundaries. The coercive force can approach the residual magnetization as a limit, since the material's intrinsic magnetization cannot increase along with increasingly negative values of

magnetizing force. In a bulk magnet the level of intrinsic saturation is the composite contribution of all the domains or crystal sites in the material. It is important to pack together as many sites as possible with each one aligned to makes a maximum contribution. Ideally, the ratio B_r/B_i should be unity.

Permanent magnets can be divided into two basic types with respect to their resistance to demagnetization measured by the coercive force H_c and their residual magnetization B_r. For the Type I magnets, the intrinsic coercive force $H_{ci} << B_r$, and for the Type II magnets, $H_{ci} \geq B_r$. Rare-earth permanent magnets are Type II magnets. This dissimilarity of the two characteristics means that the behavior of the two classes of magnets will be different, and they generally cannot be used interchangeably in the same application.

The first distinction between the two classes of magnet concerns their resistance to demagnetization. For the Type II magnets, the point of irreversibility of induction moves from the second quadrant of the induction hysteresis loop to the third quadrant. A magnet in this class can encounter levels of demagnetization greater than its own maximum self-demagnetizing influence without the need for re-magnetization. Also, their reversible permeability is near unity, and the major and minor hysteresis loops tend to have the same slope. When a Type II magnet is loaded and unloaded over a dynamic work cycle, as would happen in a torque drive or a latching magnet, it transforms energy efficiently. The magnet stores potential energy within its volume, and this energy can be converted back into mechanical work. For a Type I magnet, such as Alnico-S, most of the potential energy is stored in the leakage field external to the magnet's volume, and only a small part of this energy can be converted back into mechanical work. For the dynamic-work-cycle sort of service, the significant property of the magnet is its useful energy $(BH)_u$, as opposed to its total available energy $(BH)_{Max}$. A comparison of the useful energy of common types of permanent magnets shows that rare-earth magnets are 8 to 10 times more efficient than Alnico or ceramic magnets.

A second point of distinction between the two classes is the greater disparity between the intrinsic and normal demagnetization curves for the Type II magnets. This disparity means that a heavily self-demagnetized Type II magnet can be placed in an external field, and yet very large forces or torques per unit mass may still be obtained. The torque is directly proportional to the intrinsic magnetization of the magnet and the field in which it is immersed. Rare-earth magnets have high levels of intrinsic magnetization, making them practically immune to magnetization reversal when placed in very high fields.

The third important distinction between the two classes concerns leakage. For the Type II magnets, the permeability of the magnetic materials is high, making it essentially the same as the permeability of the surrounding space. When a magnet's permeability is high, the lines of flux leaving the magnet's pole are practically perpendicular to the surface of the pole. Leakage is almost nil, and all of the flux at the pole can be used. For example, the low leakage of samarium-cobalt magnets improved the performance of traveling-wave tubes. To focus the beam of these tubes, the most efficient type of magnetic circuit is a periodic reversing structure in which the focusing cells are in repulsion. The peak field seen by the electron beam is limited by the coercive force of the magnet. The large coercive force of samarium-cobalt allows high fields, resulting in an efficient high-power tube.

7. Initialization for permanent-magnet problems

For Pandira problems, currents regions are treated as real currents only if IPERM is zero. The setting IPERM = 0 is the default if the input file includes any regions with nonzero current. If IPERM = 1, then the problem contains no real currents and must contain at least one region of permanent-magnet material.

Older versions of Pandira required the user to set IPERM = 1 for a permanent-magnet problem containing no real currents. In addition, the user needed to supply a line-current region somewhere in the problem to initialize the potential. These settings are no longer necessary. Automesh now sets the value of IPERM correctly for most problems by checking whether the input file contains any current regions and any permanent-magnet materials. Automesh will set IPERM = 1, if a problem contains permanent-magnet material and no current regions. For problems with current regions, Automesh sets IPERM = 0.

8. Controlling pivoting in Pandira matrix inversion routines

Variable IPIVOT sets the pivoting strategy used to solve the tridiagonal matrix in Pandira. No pivoting (IPIVOT = 0) is the fastest, but least accurate, of the methods. No pivoting works reasonably well for most problems of moderate size (a few 10K mesh points). More accurate solutions are obtained by the use of either partial pivoting (IPIVOT = 1) or complete pivoting (IPIVOT = 2). In partial pivoting only rows are interchanged, and in complete both rows and columns are interchanged. The accuracy of the solution with partial pivoting is usually very nearly the same as that obtained by complete pivoting. The default value is IPIVOT =1 or partial pivoting.

9. Mesh-point initialization in coil regions

For magnet problems, program Automesh initializes the vector potential A_z or A_ϕ on mesh points within current regions by summing contributions from the six mesh triangles. Each triangle's contribution depends on its area, current density, and reluctivity. This calculation has an option to use an "angle formula" that takes into account the angles between the triangle legs that meet at the node. There is very little difference between the angle formula and the formula that uses only triangle areas except near the edges of coil regions. The 1987 Reference Manual claims that the angle formula is more accurate.

Variable ICAL in the first REG namelist of the input file selects which option to use. If ICAL = 0, the code uses triangle areas only, and if ICAL = 1, it uses the angle formula. As best we can determine, the default setting has always been ICAL = 0.

To illustrate the effect, consider the standard H-magnet problem HTEST1.AM as an example. For mesh points at the coil edges the two methods differ substantially (up to ~50%) in the initial value assigned to A_z. Just one mesh interval inside the coil the differences were below 0.1%, and well inside the coil the differences were smaller than 1 part per million. The component B_y along y = 0 is identical in the two resulting solutions. At the coil edge where x = 6 cm, $B_y \approx 11{,}000$ G. Component B_x is nearly zero along y = 0, but at x = 6 cm, B_x falls to about −110 G for ICAL = 0, and rises to about +110 G for ICAL = 1.

10. H·dl integrals during the Poisson calculation

For magnet problems, Poisson can optionally make a correction to the solution matrix at regular intervals based on a calculation of the integral H·dl around the Dirichlet boundary. The 1987 Reference Manual claims that making this correction may speed convergence of the solution. Variable IHDL in the first REG namelist of the input file is the number of cycles between calculations of H·dl. The default value is IHDL = 100,000, effectively turning off this option for most problems. Select the option by entering a value smaller than the total number of cycles typically required for a solution. If the option is in use, then setting REG namelist variable ICYSEN = 1 causes Poisson to write each calculated integral value in the output summary.

On example problems HTEST1.AM, HFULL.AM, and DIPOLE1.AM we have found marginally better performance for some settings of IHDL, but we have also seen convergence take much longer for other settings. With the default setting where the H·dl calculation is not done, the HTEST1 problem takes 690 cycles. The best result (520 cycles) had IHDL = 60. However, for IHDL = 40, convergence took 840 cycles; IHDL = 30 took 1430 cycles; and IHDL = 90 took 780 cycles. With the default setting, the HFULL problem takes 2950 cycles. Setting IHDL = 500 was the only value we found that improved convergence (2770 cycles). Several other settings from IHDL = 50 to 1000 required more than 3200 cycles, sometimes much more. The DIPOLE1 problem takes 4040 cycles and we saw only slight changes for IHDL = 100 (4070 cycles), 500 (4000 cycles), and 1000 (4010 cycles). For most problems, probably the best advice is to ignore the IHDL option.

11. Electrostatic fields

Though Poisson and Pandira were originally written to solve magnet problems, they can also solve for electrostatic fields. The code can include the effects of dielectric materials. Though seldom used, the codes can use a variable permittivity. Setting variable XJFACT = 0 in the first REG namelist section of the Automesh input file signals an electrostatic problem.

Use MT namelist sections in the Automesh input file to enter material properties. Variable EPSILON listed in Table VI-5 is the relative permittivity ε_r. defined as follows:

$$\varepsilon_r = \frac{\varepsilon}{\varepsilon_0}.$$

Material 0 is metal ($\varepsilon_r = \infty$) and material 1 is air or vacuum ($\varepsilon_r = 1$). The default setting is MODE = –1 for electrostatic problems, which means that all materials with MAT = 2 or higher have a finite fixed value of the permittivity ε_r. Automesh does not allow the setting MODE = –2 for electrostatic problems because it implies the unphysical case that $\varepsilon_r = 0$. Variable FIXEPS is the default value of ε_r. A value of EPSILON in the MT namelist section for a material takes precedence over FIXEPS. For variable permittivity, set MODE = 0 and supply a table of (E,ε_r), $(E,1/\varepsilon_r)$, and (E,D) as arrays BGAM, BMU, and BH, which refer to the magnetostatic tables of (B,γ), (B,μ_r), and (B,H).

In directory LANL\Examples\Electrostatic\Dielectrics, the example file 2EPSILON contains MT namelist sections to define the permittivity for two materials:

```
&MT MTID=3
EPSILON=2

&MT MTID=4
EPSILON=9
```

To set the voltage on a conductor, set IBOUND = –1 and enter a value for VOLTAGE in the REG namelist for the region. The setting indicates a fixed potential. Several LANL\Examples\Electrostatic subdirectories contain files that illustrate setting the voltage on a conductor. To simulate a floating electrode, enter a region of high-dielectric material instead of a metal region. Use MAT = 2 (or a higher number) for the electrode region and enter a very large, but finite, dielectric constant EPSILON for the material.

E. Output options in Poisson and Pandira

After solving for the field on all the mesh points, Poisson and Pandira can produce several types of output in files OUTPOI.TXT and OUTPAN.TXT. Variables in the Automesh input file control these options. One commonly used option produces a table of the interpolated field and field gradient along a line or on a rectangular grid. Another popular option is the harmonic analysis of the magnetic field.

The codes no longer support the very old printer plot type of output from previous versions. Instead, we recommend the SF7 program for interpolating fields on lines, arc, grids, or user-supplied curves. Also, the WSFplot program is a faster and more convenient way to explore the field distribution. If you need the numerical values of the nonzero component of the vector potential on individual mesh points, use the List35 program.

1. Defining boundaries for computing the field and gradient

Table VIII-4 lists the variables that define a grid or line in the problem geometry for which Poisson or Pandira prints field and gradient information. The code generates a table of fields at the requested X and Y (or R and Z) coordinates. Each line includes the logical coordinates K and L of the nearest mesh point. For magnet problems in Cartesian coordinates the code reports A_z, B_x, B_y, B, $\partial B_y/\partial y$, $\partial B_x/\partial y$, and $\partial B_y/\partial x$. It does not supply a value for $\partial B_x/\partial x$ because (in the absence of a magnetic monopole) $\partial B_x/\partial x = -\partial B_y/\partial y$. In regions of zero current density there are only two unique second derivatives since $\partial B_x/\partial y = \partial B_y/\partial x$. For conformally mapped problems the output includes coordinates and field components in both planes (u-v and x-y).

For magnet problems in cylindrical coordinates the code reports A_ϕ, B_r B_z, B, $\partial B_z/\partial r$, $\partial B_r/\partial z$, and the field index $n = (r/B_z)(\partial B_z/\partial r)$. For electrostatic problems in Cartesian coordinates the code reports V, E_x, E_y, and E, and for electrostatic problems in cylindrical coordinates the code reports V, E_r E_z, and E.

Each line in the table contains the fitting function number used by the field interpolator and the χ^2 per degree of freedom for the fit. The precision of the printed fields depends on SF.INI variable DecimalPlaces. The allowed range for DecimalPlaces is 3 to 12. The default setting is 6.

The default settings print data along the bottom edge of the problem geometry. The two coordinates are treated independently. If XMINF and XMAXF are both zero, then the abscissa selection is by logical coordinates in KMIN and KTOP. If YMINF and YMAXF are both zero, then the ordinate selection is by logical coordinates in LMIN and LTOP.

Table VIII-4. Variables defining the field-interpolation region.

Variable	Default Value
KMIN	1
KTOP	KMAX
LMIN	1
LTOP	1
XMINF	0
XMAXF	0
YMINF	0
YMAXF	0

For a nonzero value in XMINF or XMAXF, the value of KTOP defines a grid size DX (KMIN is ignored). Similarly, for a nonzero value in YMINF or YMAXF, the value of LTOP defines a grid size DY (LMIN is ignored). The relationships are:

$$DX = (XMAXF - XMINF)/(KTOP - 1),$$

and

$$DY = (YMAXF - YMINF)/(LTOP - 1).$$

The names XMINF, XMAXF, YMINF, and YMAXF differ from the older UNIX-version names to avoid conflict with other Automesh namelist variables.

If you want Poisson or Pandira to output the fields at every grid point you need to set to zero all the physical coordinates for the grid and specify the logical coordinates. For example, include the following settings in the first REG namelist of the Automesh input file:

```
XMINF=0,XMAXF=0,YMINF=0,YMAXF=0,
KMIN=1,KTOP=50,LMIN=1,LTOP=100,
```

where we have requested a 50-by-100 array of values. In this case, the grid spacings are not uniform, but follow the actual locations of the mesh points. When you use the physical coordinates you get a uniform grid spacing. Both types of interpolation use the same method for field interpolation used in the SF7 code. One of the implications of this procedure is that the value of A_z or A_ϕ reported on each grid point will be the result of the interpolation and, in general, will differ slightly from the value of the potential in the solution array. If you want the actual values of the solution array on all the grid points you can use the LIST35 program. However, you will not get values for the B components from LIST35 because LIST35 does not use the field interpolator.

2. Harmonic analysis of the field's multipole content

Harmonic analysis of the field gives a numerical estimate of the multipole content of the field in the gap of a magnet. Figure VIII-4 illustrates the parameters involved for an H-shaped dipole magnet. One chooses the point x_0,y_0 about which to do the analysis and the radius r of a circle about that point. Usually, the point x_0,y_0 is the origin of coordinates for the problem. The following discussion is in two sections. As an option, one can specify the coordinates x_A,y_A where the vector potential is zero. The first section is based upon the chapter titled "Harmonic Analysis" in Part B, Chapter 13.3 by John L. Warren in the 1987 publication Reference Manual for the Poisson/Superfish Group of Codes, LA-UR-87-126. We reproduce it here in essentially its original form. The second section discusses how to implement the harmonic analysis in Poisson or Pandira.

a. Mathematical theory of harmonic analysis

The mathematical theory is based on the idea that the vector potential A(x, y) can be thought of as the real part of a complex function F(z) where z = x + iy. The function F(z) can be expanded in a power series

$$F(z) = A(x,y) + i\,V(x,y) = \sum_{n=0}^{\infty} \frac{c_n}{r_{norm}^n} z^n \,, \qquad \text{(VIII-1)}$$

where r_{norm} is a normalization radius. With this definition, the coefficients c_n all have the same units, for example, gauss-cm. If we let

$$c_n = a_n + i\,b_n \,, \qquad \text{(VIII-2)}$$

and

$$z^n = u_n + i\,v_n \,, \qquad \text{(VIII-3)}$$

then the vector potential is given by

$$A(x,y) = \sum_{n=0}^{\infty} \frac{(a_n u_n - b_n v_n)}{r_{norm}^n} \,. \qquad \text{(VIII-4)}$$

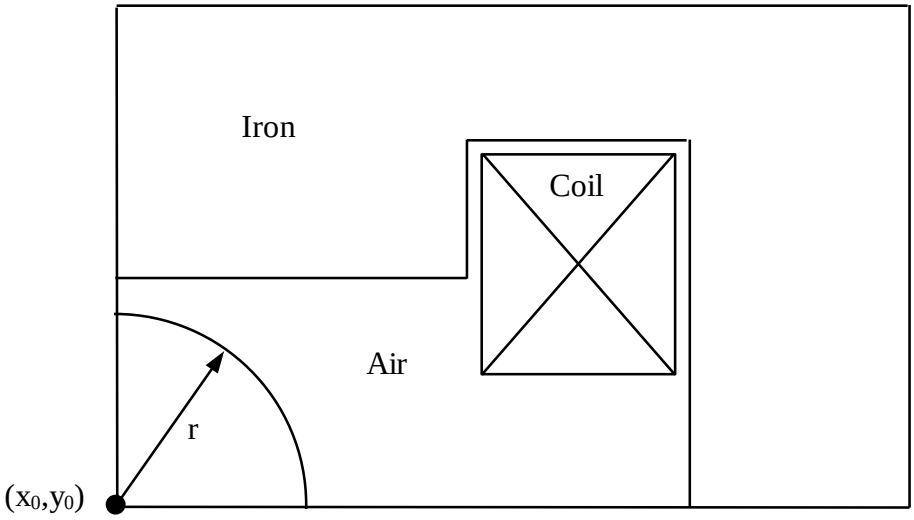

Figure VIII-4. One quarter of a symmetric H-shaped magnet.
The harmonic analysis will be done about the center of the gap on a circular arc of radius r.

The quantities u_n and v_n are called harmonic polynomials (see Chapter XXI). These polynomials generate the regular and skew multipole fields, which means that the representation of the potential in terms of this power series is equivalent to a multipole decomposition of the potential. To implement this decomposition, we find the coefficients a_n and b_n by evaluating the potential on the circle. A simple method uses polar coordinates in the complex plane, expressing z by the formula

$$z = re^{i\varphi}, \tag{VIII-5}$$

and the coefficients c_n by the formula

$$c_n = |c_n| e^{i\alpha_n} = |c_n| \left(\cos\alpha_n + i \sin\alpha_n\right), \tag{VIII-6}$$

When these relations are substituted into the formula for the vector potential, one obtains

$$\begin{aligned} A(r,\varphi) &= \mathrm{Re}\left[\sum_{n=0}^{\infty} \left(\frac{r}{r_{norm}}\right)^n |c_n| e^{i\alpha_n} e^{in\varphi}\right] \\ &= \sum_{n=0}^{\infty} \left(\frac{r}{r_{norm}}\right)^n |c_n| \cos\left(\alpha_n + n\varphi\right), \\ &= \sum_{n=0}^{\infty} \left(\frac{r}{r_{norm}}\right)^n \left(|c_n| \cos\alpha_n \cos n\varphi - |c_n| \sin\alpha_n \sin n\varphi\right) \\ &= \sum_{n=0}^{\infty} \left(\frac{r}{r_{norm}}\right)^n \left(a_n \cos n\varphi - b_n \sin n\varphi\right). \end{aligned} \tag{VIII-7}$$

Equation VIII-7 has the form of a Fourier series in the variable φ. Fourier analysis theory tells us that

$$a_0 = \frac{1}{2\pi}\int_0^{2\pi} A(r,\varphi)d\varphi, \quad b_0 = 0, \tag{VIII-8}$$

$$a_n = \frac{1}{\pi}\left(\frac{r_{norm}}{r}\right)^n \int_0^{2\pi} A(r,\varphi)\cos(n\varphi)d\varphi, \quad n \geq 1, \tag{VIII-9}$$

$$b_n = -\frac{1}{\pi}\left(\frac{r_{norm}}{r}\right)^n \int_0^{2\pi} A(r,\varphi)\sin(n\varphi)d\varphi, \quad n \geq 1. \tag{VIII-10}$$

The circle of integration should not include any part of an iron or coil region, because this would be inconsistent with the assumptions underlying the use of the complex potential F(z). We can take advantage of the magnet symmetry. Consider the magnet shown in Figure VIII-4. Because of the field symmetry one does not need to integrate over the entire range from φ = 0 to φ = 2π. In this case, the integrals from φ = 0 to φ = 2π are four times the value of the integrals from φ = 0 to φ = π/2. We can generalize the limits of integration to be from φ = ϕ_z to φ = ϕ, where φ is measured from the horizontal (x or r) axis. The definitions of the harmonic coefficients become

$$a_0 = \frac{1}{\phi - \phi_z}\int_{\phi_z}^{\phi} A(r,\varphi)d\varphi, \quad b_0 = 0, \tag{VIII-11}$$

$$a_n = \frac{2}{\phi - \phi_z}\left(\frac{r_{norm}}{r}\right)^n \int_{\phi_z}^{\phi} A(r,\varphi)\cos(n\varphi)d\varphi, \quad n \geq 1, \tag{VIII-12}$$

$$b_n = -\frac{2}{\phi - \phi_z}\left(\frac{r_{norm}}{r}\right)^n \int_{\phi_z}^{\phi} A(r,\varphi)\sin(n\varphi)d\varphi, \quad n \geq 1. \tag{VIII-13}$$

The coefficients a_0 and b_0 are not needed to calculate the magnetic field. The integrals are computed numerically by converting them to summations. To make this conversion, we let

$$d\varphi \rightarrow \Delta\varphi = \frac{\phi - \phi_z}{N_{ptc} - 1}, \tag{VIII-14}$$

where N_{ptc}, is the number of equidistant interpolated points on the arc of radius r. One cannot, of course, obtain all the coefficients a_n and b_n, but must limit oneself to $n \leq N_{term}$, where the integer N_{term} is the number of coefficients to be obtained.

The harmonic analysis of the magnetic induction starts with the series relation

$$B_x - iB_y = i\frac{dF}{dz} = \sum_{n=1}^{\infty} in\left(\frac{r}{r_{norm}}\right)^{n-1} \frac{c_n}{r_{norm}} e^{i(n-1)\varphi} . \qquad \text{(VIII-15)}$$

which can be equated to a Fourier series for the induction written in the form

$$B_x - iB_y = \sum_{m=0}^{\infty} B_m e^{im\varphi} . \qquad \text{(VIII-16)}$$

The strengths B_m of the harmonic components of the magnetic induction are directly related to the coefficients a_n and b_n found in analyzing A(x, y). The relationship is

$$B_m = -\frac{(m+1)}{r_{norm}}\left(\frac{r}{r_{norm}}\right)^m \left(b_{m+1} - ia_{m+1}\right). \qquad \text{(VIII-17)}$$

Poisson and Pandira print the quantities a_n, b_n, na_n/r_{norm} and nb_n/r_{norm}. The program also gives absolute values of the complex quantities c_n and nc_n/r_{norm}, which do not have a simple interpretation.

b. Implementing the harmonic analysis

Table VIII-5 lists variables that control the harmonic analysis performed by Poisson and Pandira. To request the harmonic analysis, specify proper values for variables in Table VIII-5. The table includes the mathematical symbols used in Equations VIII-1 through VIII-17 above. The HTEST1.AM example file includes the parameters in the first REG namelist in the Automesh input file. Variables NTERM, RNORM, and ANGLZ have default values of 5, RINT, and 0.0, respectively. Variables XORG and YORG, which define the center of the arc, both have default value of 0.0. There are no default values for NPTC, RINT, and ANGLE. Variable RINT is the radius of an arc of extent ANGLE along which the code will interpolate the potential at NPTC evenly spaced locations. ANGLZ is the starting angle on the arc measured counterclockwise from a line parallel to the horizontal axis. Optional variables XAZERO and YAZERO define the location x_A,y_A where the vector potential A_z (or A_ϕ) is zero. If you supply values for XAZERO and YAZERO, then the code will subtract the value of A_z or A_ϕ from all interpolated points on the arc.

NPTC, the number of points on the interpolation arc, must be at least 3. We recommend setting NPTC to approximately the number of mesh triangles intersected by the arc. There is no practical upper limit on NPTC because the code allocates memory for points on the arc. The maximum number of coefficients (NTERM) is 14. The distance between the circle and the pole piece should be larger than the mesh increment DY. Select the radius RINT so that the arc does not come within a row of mesh triangles of any coil or iron regions.

Table VIII-5. Variables that control the harmonic analysis.

Variable	Symbol	Description
KTYPE		Symmetry indicator for harmonic analysis.
XORG	x_0	Arc-center X coordinate for the harmonic analysis.
YORG	y_0	Arc-center Y coordinate for the harmonic analysis.
NTERM	N_{term}	Number of coefficients in the harmonic analysis.
NPTC	N_{ptc}	Number of arc points for interpolating the potential.
RINT	r	Radius of the arc for interpolating the potential.
ANGLE	$\phi - \phi_z$	Extent of the arc for interpolating the potential.
RNORM	r_{norm}	Aperture radius used in the harmonic analysis.
ANGLZ	ϕ_z	Initial point on arc for interpolating the potential.
XAZERO	x_A	Physical X coordinate where A = 0.
YAZERO	y_A	Physical Y coordinate where A = 0.

The code checks for proper values of these input variables before it will proceed. Warning messages tell if the interpolation arc includes points outside the problem geometry. Table VIII-6 lists the various symmetry options specified by the value of KTYPE. The second through fourth columns list the order of the starting coefficient, the interval between coefficients, and the median-plane symmetry option. Table VIII-7 explains the meaning of median-plane symmetry column. For example, KTYPE = 6 for an H-magnet dipole would fit to coefficients A(1), A(3), A(5), etc. For symmetry types other than these default values for KTYPE = 1 through KTYPE = 9, you can enter a 3-digit number for KTYPE constructed from the three columns labeled First coefficient, Interval, and Median-plane symmetry. For example, KTYPE = 241 is equivalent to the KTYPE = 4 selection in Table VIII-6.

Though the harmonic analysis for cylindrically symmetric problems has doubtful application, the codes will generate the fits if requested. For magnetic-field problems in cylindrical coordinates, KTYPE = 1 means no symmetry, and KTYPE = 2 means mid-plane symmetry with the magnetic field lines perpendicular to the R axis. For scalar potentials (electrostatic problems), KTYPE = 1 means no symmetry, KTYPE = 2 means mid-plane symmetry with the lines of constant potential perpendicular to the R axis, and KTYPE = 3 means that the R axis is a line of constant potential.

Table VIII-6. Symmetry options for the harmonic analysis.

KTYPE	First coefficient	Interval	Median-plane symmetry	Used for
1	1	1	2	No symmetry.
2	1	1	1	Mid-plane symmetry.
3	2	2	1	Elliptical aperture quadrupole.
4	2	4	1	Symmetric quadrupole.
5	2	2	0	Skewed elliptical aperture quad.
6	1	2	1	"H" dipole or elliptical aperture sextupole.
7	3	6	1	Symmetric sextupole.
8	2	2	1	Elliptical aperture octupole.
9	4	8	1	Symmetric octupole.

Table VIII-7. Meaning of median-plane symmetry in Table VIII-6.

Value	Meaning
0	A = 0, the Cn coefficients are pure imaginary, $B_y(x,0) = 0$.
1	V = 0, the Cn coefficients are pure real, $B_x(x,0) = 0$.
2	no symmetry, the Cn coefficients are complex.

3. Conformal transformations

Conformal transformations have been used to trim multipole magnets for as uniform as possible field in a given region. The most common application is to remove higher order multipole components of the field from a quadrupole magnet. The procedure involves conformally transforming the quadrupole field to a dipole field, making the dipole field more uniform, and transforming back to the quadrupole field. After running Poisson or Pandira on the resulting quadrupole geometry, the user can perform a harmonic analysis to evaluate the field purity.

Klaus Halbach described the theory in "Application of Conformal Mapping to Evaluation and Design of Magnets Containing Iron with Nonlinear B(H) Characteristics," Nuclear Instruments and Methods **64,** p. 278 (1968). In the mid 1980s, Robert J. Lari at Argonne National Laboratory (ANL) summarized using the conformal mapping features in Poisson (or Pandira). We quote here from Lari's September 12, 1985 letter to the LAACG and an ANL internal report: R. J. Lari, "Harmonic analysis errors in calculating dipole, quadrupole, and sextupole magnets using Poisson," ANL Light Source note LS-32 (September 10, 1985).

The following equation transforms a coordinate (x,y) of higher pole magnet to coordinates (u,v) of a dipole magnet:

$$W = \frac{Z^{p/2}}{(p/2)R_0^{(p/2)-1}},$$

where W = u + iv, Z = x + iy, p is the number of poles and R_0 is the magnet bore radius. For a quadrupole magnet, this transformation becomes

$$u = \frac{x^2 - y^2}{2R_0}, \text{ and } v = \frac{xy}{R_0}.$$

The procedure requires that the user supply the transformed geometry to Automesh along with the values of p, I_-, I_+, and R_0 (see Table VIII-8). Specify the u,v coordinates as x,y points in Automesh. When setting up the triangular mesh, Automesh uses the supplied values of I_- and I_+ to apply a correction to make the currents the same in both the x-y and u-v planes. You cannot use line regions with conformally mapped problems.

Table VIII-8.Variables that control conformal mapping.

Variable	Symbol	Description
MAP	p	Number of poles for the magnet.
TNEGC	I_-	Total negative current after conformal transformation.
TPOSC	I_+	Total positive current after conformal transformation.
RZERO	R_0	Magnet bore radius.
XJFACT		Must be nonzero, signaling a magnet problem.
ICYLIN		Must be zero, signaling Cartesian coordinates.

If variable MAP is 2 or higher for magnet problems (XJFACT ≠ 0) in Cartesian coordinates (ICYLIN = 0), the Poisson or Pandira output field table includes coordinates and field components in both planes (u-v and x-y).

We are not sure whether the conformal mapping approach is as useful as it was several decades ago when problems were limited to a few hundred mesh points. It is now possible to finely mesh very detailed geometries and perform the harmonic analysis. Note that Automesh does not perform the conformal transformation of the geometry; that is up to the user. Given a conformally mapped geometry, Poisson and Pandira will print fields in both the u-v and x-y planes. After having "tuned" a geometry in the conformally mapped (e.g., dipole) geometry, it may be difficult to generate an accurate mesh for the multipole (e.g., quadrupole) geometry because the pole shapes may not be composed of simple line segments and arcs.

4. Stored energy calculations in Poisson and Pandira

If IENERGY = 1, then Poisson and Pandira calculate the stored energy in the fields for both magnet problems and electrostatic problems. The ability to compute stored energy in permanent-magnet or anisotropic materials has not yet been added to Pandira. This feature may be available in a future release. The present code will write a warning message if you request a stored-energy calculation in problems with permanent-magnet or anisotropic materials.

The default in the codes is not to calculate the stored energy automatically. You can change this default using the SF.INI setting ComputeStoredEnergy. For example, to make Poisson calculate the stored energy without having to supply IENERGY for each problem add the following line in the [Poisson] section of SF.INI:

```
[Poisson]
ComputeStoredEnergy = Yes
```

A user-supplied IENERGY in the Automesh input file takes precedence over the SF.INI setting. For magnet problems, the section of code that computes stored energy comes from a routine added to the UNIX version in 1985. In December, 1994 we corrected the stored-energy calculation for magnet problems to account for the saturation magnetization if the B field exceeds the highest entry in the permeability table. The code now extends the internally generated table of magnetic energy density by assuming the last point (for highest B) in the permeability table corresponds to saturated iron.

Stored-energy calculations use the field interpolator to evaluate the fields in each mesh triangle at the average position of the three triangle vertices. We have verified the accuracy of the calculations using several simple geometries for which one can calculate the stored energy analytically. Some subdirectories under LANL\Examples\Electrostatic and LANL\Examples\Magnetostatic contain some of these example problems. The examples include:

- Parallel plate capacitor in both rectangular and cylindrical coordinates
- Current carrying coaxial cable in rectangular coordinates
- Magnetic solenoid in cylindrical coordinates
- Coaxial capacitor in both rectangular and cylindrical coordinates

F. Convergence in the Poisson and Pandira solvers

Poisson uses the method of successive over relaxation (SOR) and Pandira uses a direct matrix inversion method. If a problem contains no iron, the permeability is infinite (MODE = –2), or the permeability is finite but constant (MODE = –1), then Pandira finishes in only one iteration and does not need to test for convergence.

Table VIII-9. Variables related to Poisson and Pandira convergence tests.

Variable	Default value	Description
MODE	–2, –1	Material property indicator.
MAXCY	100000	Maximum number of cycles (default is 20 in Pandira).
OMEGAP	0.001	Parameter used in optimizing the over-relaxation parameter.
SNOLDA	Calculated	Square of residuals for “air” points.
SNOLDI	Calculated	Square of residuals for “iron” points.
RHOPT1	1.9	If equal to RHOAIR, causes optimization of RHOAIR.
RHOAIR	1.9	Over-relaxation factor for air and interface points.
RHOFE	1.0	Over-relaxation factor for iron points.
RHOGAM	0.08	Under-relaxation factor for the reluctivity in Poisson.
ISKIP	1	Cycles between calculation of reluctivity in Poisson.
EPSILA	5.D–7	Convergence parameter for air and interface points.
EPSILI	5.D–7	Convergence parameter for iron points.
IVERG	10	Number of cycles between convergence tests in Poisson.
RESIDA	Calculated	Residual of the solution for air points.
RESIDI	Calculated	Residual of the solution for iron points.
ICYCLE	Calculated	Iteration (or cycle) number.
ETAAIR	Calculated	Rate of convergence for air points at cycle ICYCLE.
ETAFE	Calculated	Rate of convergence for iron points at cycle ICYCLE.

Poisson tests for convergence after every IVERG iterations of the SOR calculation. If the problem contains iron of finite and variable permeability (MODE = 0), then Poisson performs separate tests for convergence. One test is for "air" and interface points, and the other test is for "iron" points (material number MAT equal 2 or higher). If MODE = 0, Pandira tests for convergence in the iron after every iteration. Table VIII-9 lists all the variables in Poisson and Pandira that either affect the tests for convergence in some way or are quantities computed during the tests for convergence. Table VIII-10 describes in more detail the parameters that you can change to affect the convergence.

In the following discussion, we outline the procedure used in Poisson for calculating the parameters needed in the test for convergence in "air." The computation and test for iron points in Poisson and Pandira is similar. Both convergence tests must be satisfied if the iron test is done at all. After IVERG iterations the code evaluates A_{norm}, the average value of the vector potential for the air and interface points:

$$A_{norm} = \frac{1}{NAIR} \sum_{I=1}^{NAIR} A\big(IRLAX(I)\big),$$

where NAIR is the number of such points. The relaxation-order array IRLAX contains the indices to all the mesh points in the problem geometry. The summation from 1 to NAIR implies that we pick out only those points that correspond to air and interface mesh points. Next, the code calculates SUMA, the sum of the squares of the differences between A and A_{norm} at each point:

$$SUMA = \sum_{I=1}^{NAIR} \Big[A\big(IRLAX(I)\big) - A_{norm} \Big]^2 .$$

The code also calculates SUMDA, the sum of the squares of all the changes to be made in the vector potential at the current iteration:

$$SUMDA = \sum_{I=1}^{NAIR} \Big[\delta A\big(IRLAX(I)\big) \Big]^2 .$$

The residual for "air" points is defined as

$$RESIDA = \sqrt{\frac{SUMDA}{SUMA}} .$$

If SNOLDA is the value of SUMDA calculated IVERG iterations earlier, the convergence rate for "air" points is then defined as

$$ETAAIR = \left(\sqrt{\frac{SUMDA}{SNOLDA}} \right)^{\frac{1}{IVERG}} .$$

For convergence, the current value of RESIDA must be less than the value of EPSILA. If the problem contains iron of finite and variable permeability, then the similarly computed

RESIDI also must be less than EPSILI. If the solution has not yet converged, then Poisson recalculates RHOAIR, the over-relaxation parameter for air points. The code performs this optimization only if the initial value of RHOAIR was equal to the value of RHOPT1. The default value for both of these parameters is 1.9. To optimize RHOAIR, Poisson first computes the parameter λ:

$$\lambda = \frac{\text{RHOAIR} - \text{ETAAIR} - 1}{\text{RHOAIR}\sqrt{\text{ETAAIR}}}.$$

Poisson reports the current value of λ on the screen and in the file OUTPOI.TXT whenever it updates the value of RHOAIR. If the value of λ is less than 1, then the optimized RHOAIR is given by

$$\text{RHOAIR} = \frac{2}{1+\sqrt{1-\lambda^2}} - \text{OMEGAP},$$

where OMEGAP has the default value of 0.001. If λ is 1 or larger, then the over-relaxation parameter has gotten too large. In an attempt to reduce it, Poisson makes the new RHOAIR the larger of 1.0 and the quantity RHOAIR – 10*OMEGAP.

Table VIII-10. Parameters that affect convergence in Poisson or Pandira.

Variable	Description
MAXCY	Maximum number of cycles allowed before giving up. The default values are 100000 for Poisson and 20 for Pandira. You can stop the search at fewer cycles or allow more cycles by changing MAXCY.
OMEGAP	Parameter used for optimizing the over-relaxation parameter RHOAIR in Poisson. The default value is 0.001. Pandira does not use OMEGAP.
RHOPT1	Determines whether Poisson optimizes RHOAIR. Poisson optimizes RHOAIR if the initial values of RHOAIR and RHOPT1 are equal. The default value is 1.9. Pandira does not use RHOPT1.
RHOAIR	Over-relaxation parameter for air points in Poisson. The default value is 1.9. Poisson optimizes RHOAIR every so often if the initial value of RHOAIR and RHOPT1 are equal. RHOAIR must be start with a value between 1.0 and 2.0. Poisson resets any value 2.0 or higher to 1.9999 and any value below 1.0 to 1.0. Pandira does not use RHOAIR.
RHOFE	Over-relaxation parameter for iron (points with MAT = 2 or higher). The default value is 1.0, which implies no over-relaxation. The initial value of RHOFE must be between 1.0 and 2.0. Poisson and Pandira reset any value 2.0 or higher to 1.9999 and any value below 1.0 to 1.0.
RHOGAM	Under-relaxation parameter for reluctivity in Poisson. RHOGAM prevents too-rapid a change in the reluctivity in a single iteration. The default value is 0.08. Smaller values typically are better for materials such as vanadium permendur. Poisson uses RHOGAM only for problems with finite and variable permeability (MODE = 0). Pandira does not use RHOGAM.
ISKIP	Number of cycles between recalculation of reluctivity in Poisson. The code uses ISKIP only for problems with finite and variable permeability (MODE = 0). The default value is 1. Pandira does not use ISKIP.
EPSILA	Convergence parameter for air in Poisson. The default value is 5×10^{-7}. Pandira does not use EPSILA.
EPSILI	Convergence parameter for iron (points with MAT = 2 or higher). The default value 5×10^{-7}.
IVERG	Number of cycles between convergence tests in Poisson. The default value is 10. Pandira does not use IVERG.

You can change Table VIII-10 parameters to modify the convergence tests in the Poisson and Pandira solvers. Please be forewarned that it is very easy to make things considerably worse by tinkering with certain of these parameters. For example, not allowing the code to optimize RHOAIR can increase the number of iterations for convergence by a factor of 3 or more even if RHOAIR has a fairly reasonable (but constant) value between 1.8 and 1.99. Unless you have a good reason for changing RHOAIR or OMEGAP, it is best to leave them alone. Consider the H-magnet example file. This problem converges in 690 cycles if RHOAIR starts at 1.90 and Poisson optimizes it at cycles 50, 100, 200, and 400. Table VIII-11 shows the value of ICYCLE at convergence for several fixed values of RHOAIR. By manually setting RHOAIR to 1.97 or 1.98 (and making RHOPT1 a different value), Poisson converges only slightly faster than it would using its default initial setting of 1.9. However, the convergence is much worse at other nearby fixed values.

Table VIII-11. Number of cycles to converge versus RHOAIR.

RHOAIR	ICYCLE
1.80	4580
1.85	3460
1.90	2330
1.95	1160
1.96	900
1.97	600
1.98	650
1.99	1050

On the other hand, it is not unreasonable to reduce the size of EPSILA or EPSILI listed in Table VIII-10 for a more accurate solution, provided that the mesh spacing is fine enough to warrant it. Reducing these values much below about 10^{-10} is probably overtaxing the ability of the SOR solver using Fortran double-precision coding.

In Poisson, RHOGAM is very important for problems with finite and variable permeability (MODE = 0). For each mesh triangle i, the under-relaxation parameter ρ_γ limits the rate of change of the reluctivity γ_i in a single iteration:

$$\gamma_i^{n+1} = \rho_\gamma \gamma_i^{T} + (1-\rho_\gamma)\gamma_i^{n}$$

where superscript n refers to the cycle just completed, and superscript T refers to the new value derived from the permeability table for the magnetic field B at cycle n. The default value of 0.08 is usually near optimum for 1010 steel, but it is generally too large for materials such as vanadium permendur. Some problems may not converge if RHOGAM remains at the default value. When using the internal table for vanadium permendur, or when supplying your own permeability tables, we recommend that you try a few values for RHOGAM in the range 0.0005 to 0.1 to optimize performance.

G. Quantities calculated by Poisson and Pandira

Poisson and Pandira calculate a number quantities that are reported at the end of each code's output file for your information.

1. SPOSA, SNEGA, and STOTA, currents after completing the solution

After the solution has converged, SPOSA is the total positive current in the problem geometry, SNEGA is the total negative current, and STOTA is the net current. If XJFACT remains at its default value of 1.0 for magnet problems, then these quantities will be equal to the initial values calculated by Automesh: SPOSG, SNEGG, and STOTG.

2. XJFACT, factor by which currents have been scaled

Poisson reports a calculated value for XJFACT if it has adjusted the fields for a particular magnitude of B at the physical point XBZERO,YBZERO (logical coordinates KBZERO,LBZERO). The default location is the logical point 1,1. You can select this

option by specifying a value for the field magnitude BDES. Pandira does not report a calculated value for XJFACT.

3. RATIO, the quantity |BZERO|/XJFACT

Poisson reports the value RATIO, which is equal to |BZERO|/XJFACT, where BZERO is the interpolated field at the physical point XBZERO,YBZERO (logical coordinates KBZERO,LBZERO). See the discussion of XJFACT in paragraph 2.

4. SNOLDA and SNOLDI, old values of the summed squares of the residuals

SNOLDA and SNOLDI are parameters used by Poisson to test for convergence. They are both sums of the squares of the residuals IVERG iterations previous to the current iteration. IVERG is the number of cycles between convergence tests. The default value for IVERG is 10 iterations. SNOLDA corresponds to "air" and interface points, and SNOLDI corresponds to points in "iron" (material number MAT = 2 or higher). For problems with finite but constant permeability (MODE = –1), Poisson includes the iron points in the value for SNOLDA.

Pandira does not calculate SNOLDA. It calculates a value for SNOLDI for problems with finite and variable permeability (MODE = 0). Pandira tests for convergence in iron after every iteration if MODE = 0, thus behaving as if IVERG = 1.

5. RHOAIR, optimized over-relaxation parameter

RHOAIR is the optimized over-relaxation parameter calculated by Poisson for "air" and interface points during the successive over-relaxation procedure. Pandira does not use this parameter and so does not report a calculated value for it. Poisson automatically optimizes RHOAIR if its initial value is equal to the value of RHOPT1. If this condition is not met, Poisson does not report a calculated value for RHOAIR.

The codes use two other parameters that do not change during the course of a calculation. Therefore, these parameters appear only in the original list of problem constants in OUTPOI.TXT and OUTPAN.TXT. RHOFE is the over-relaxation parameter for points in "iron" (material number MAT = 2 or higher). Both Poisson and Pandira use the value of RHOFE. The default value of RHOFE is 1.0, which implies no over-relaxation. RHOGAM is an under-relaxation parameter used only by Poisson to prevent too-rapid a change in the reluctivity in a single iteration. The default value of RHOGAM is 0.08. It is used only for problems with finite and variable permeability (MODE = 0).

6. ENERGY, computed stored energy

If IENERGY is 1, then Poisson and Pandira calculate the stored energy in the fields for both magnet problems and electrostatic problems. The codes report the result as variable ENERGY. The default is not to automatically calculate the stored energy. You can change this default using the SF.INI setting ComputeStoredEnergy.

7. RESIDA and RESIDI, the residuals for air and iron points

Variable RESIDA is the residual of the solution matrix for the vector potential A for points in "air" or open space. RESIDI is the residual for points in "iron" (material number MAT = 2 or higher). Poisson, which uses the method of successive over-relaxation, calculates a value for RESIDA. Pandira does not calculate RESIDA. Both programs calculate RESIDI for problems with finite and variable permeability (MODE = 0). If MODE is −1 or −2, Pandira converges in only one iteration and does not need RESIDA to test for convergence. In Poisson, the iron points are included in RESIDA for problems with finite but constant permeability (MODE = −1).

8. ICYCLE, the number of the last completed iteration

Variable ICYCLE is the iteration number on which the solver stopped. In Poisson, ICYCLE is the number of iterations of the successive over-relaxation procedure. In Pandira, ICYCLE is the number of times the code inverts the tridiagonal matrix. If ICYCLE is equal to MAXCY, then the solution may not have converged to the accuracy specified by EPSILA and EPSILI.

9. ETAAIR and ETAFE, rates of convergence parameters

Variable ETAAIR is the rate of convergence of the over-relaxation solution calculated by Poisson for "air" and interface points. Pandira does not use this parameter and so does not report a value for it. Poisson uses ETAAIR in the calculation of an optimized value for RHOAIR, the over-relaxation parameter.

ETAFE is a parameter similar to ETAAIR, but for "iron" points, which have material number MAT = 2 or higher. Both Poisson and Pandira calculate a value for ETAFE for problems with finite and variable permeability (MODE = 0).

H. Poisson and Pandira error messages

Table VIII-12 lists the Poisson and Pandira error messages. The error numbers also are available as exit error codes to tuning programs and batch control files.

Table VIII-12. Error messages for Poisson and Pandira.

Error	Description
200	Solution file specified on command line does not exist.
202	File specified on command line does not appear to be a Poisson Superfish solution file.
204	File entered in browse dialog window does not exist.
207	An error occurred reading the first record from the Poisson Superfish solution file.
208	Solution arrays have not been properly declared. Please report this error.
209	The dimension of at least one solution array is too small. Please report this error.
210	Mesh point arrays are not large enough. Please report this error.
213	The mesh has overlapping or zero-area triangles.
216	Unable to read triangular mesh data (possibly incompatible code versions).
220	Reached the end of the solution file unexpectedly. Rerun problem stating with Automesh.
221	An error occurred reading a record from the solution file. Please report this error.
222	The Poisson Superfish solution file is obsolete. Rerun problem stating with Automesh.
223	Cannot run Poisson or Pandira on Superfish problems.
230	Cannot run Poisson on problems with anisotropic or permanent-magnet materials.
280	The relative permittivity for an electrostatic problem is not positive.
281	The relative reluctivity for a magnet problem is not positive.
292	Insufficient memory, cannot allocate MT namelist arrays for a Poisson problem.
501	Insufficient memory, cannot allocate arrays for binary solution file data.
503	Insufficient memory, cannot allocate arrays for the tridiagonal submatrices in Pandira.
505	Insufficient memory or disk space for Pandira temporary data and no valid drives available.
506	Insufficient memory or disk space for Pandira temporary data.
507	Insufficient memory, cannot allocate arrays for harmonic analysis interpolation points.
508	Insufficient memory, cannot allocate an array needed to invert the tridiagonal matrix..
520	A negative value for KTYPE is not a valid symmetry type.
521	The sum of the weights is zero for at least one mesh point, likely a material problem.
522	The sum of the vector potential on interface and iron points is zero, likely a material problem.
523	Decoded 3-digit KTYPE is not a valid symmetry type.
524	The harmonic analysis has attempted to use a point that lies outside the problem geometry.
525	The potential (in Poisson) is zero at all air points indicating no valid source terms.
526	The potential (in Poisson) is zero at all iron points. Iron may be too far from source terms.
527	The point at which A = 0 for the harmonic analysis is not part of the solution.