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

File SFPOSTP.DOC Table of Contents

IX. WSFplot: the Poisson Superfish Plotting Code

WSFplot plots output from programs Automesh, Fish, CFish, Poisson, and Pandira. It reads the binary solution file and plots the problem geometry boundaries plus mesh lines, field contours, arrows, or circles. For complex rf fields, WSFplot plots the real and imaginary parts in different colors. For Superfish problems, WSFplot plots the drive point as a black dot. WSFplot also displays the field components at the location of the cursor. You can explore the field distribution in the problem geometry by moving the mouse or by pressing the arrow keys. The codes run in the following order:

- Automesh
- WSFplot (optional)
- Fish, CFish, Poisson, or Pandira
- WSFplot (optional)
- SFO (optional)
- WSFplot (optional)

For Superfish problems, running WSFplot after SFO will show the field components that correspond to the field normalization applied by SFO. For Poisson and Pandira problems, there is no difference in the WSFplot display before and after running the postprocessors (SFO, SF7, or Force).

A. Starting program WSFplot

To start WSFplot, double-click on the solution file with extension T35, or right-slick on the file and select Open. You can also use a command line (for example, in a batch file) to start the code:

WSFplot *time* *T35file*

Both command-line parameters are optional and they can be in either order if both appear on the line. The *time* parameter specifies the number of seconds to display the problem geometry before stopping automatically. For interactive operation do not include a *time* on the command line. The *time* parameter permits batch control files to run unattended. The filename *T35file* is the name of a binary solution file. When using a command line, it is usually not necessary to enter the name of the binary solution file because the code will find the filename in file TAPE35.INF. You can always omit the *T35file* when starting WSFplot immediately after running Automesh or one of the solver codes. For more details about startup options refer to the section titled Opening input files on startup.

If the solution file does not contain a solution array, then WSFplot displays only the mesh triangles. If the code finds a solution, then it displays contours of constant A_z for rectangular magnet problems, rA_ϕ for cylindrically symmetric magnet problems, V for electrostatic problems, H_z for rectangular rf problems, or rH_ϕ for cylindrically symmetric rf problems. Depending upon SF.INI settings, the code may plot arrows in the direction of the field and length proportional to field strength. For rf problems, the plot may include circles with diameters proportional to field strength perpendicular to the page.

Once the program is running, you can modify the display as described in the following sections.

B. WSFplot display windows

The WSFplot display has two windows. The main window includes the menu bar at the top, status bars at the bottom, and the graphics display. The graphics display area contains the boundary outlines plus any or all of the following items: title, numbered axes, mesh triangles, field contours, field arrows, and field circles.

The field-display window lists the coordinates and interpolated fields at the cursor location when the cursor is over the graphics display window. The cursor shape is a crosshair in the display area. You can move the crosshair with the mouse or the arrow keys.

1. Main display window

The main display always uses the same scale factor for the X and Y axes, thus preserving the relative location of features in the geometry (provided that the window retains the same aspect ratio when resized). For a Superfish problem, the displayed contours are lines of constant H_z (or rH_ϕ if the problem has cylindrical symmetry). For Poisson and Pandira magnetic-field problems, contours are lines of constant A_z (or rA_ϕ for cylindrically symmetric problems). For electrostatic problems the contour lines are equipotentials.

Table IX-1. WSFplot menu items.

Menu item	Function
File, Open File	Read a new solution file.
File, **E**xit	End the program.
Hard**C**opy, Start	Make a hardcopy of the present display.
HardCopy, Options	Select options for the active hardcopy driver.
HardCopy, Driver	Show the list of hardcopy drivers. The active driver is checked (✓).
Display, Line **W**idths	Start a dialog window to select line widths of display items.
Display, Contour **L**ines	Start a dialog window to select the number and range of contours.
Display, **A**rea Boundary	Start a dialog window to define boundaries for next zoom level.
Display, Arrow **G**rid Spacing	Start a dialog window to set arrow and circle grid spacing.
Display, Arrow **K**ey Step Size	Start a dialog window to set step size for arrow keys.
Display, Re**S**et Defaults	Reset all display parameters to SF.INI settings.
Display, **R**edraw Plot	Redraw the plot (e.g., after changing window size by a large amount).
View, Show Title	Display the plot title if checked (✓).
View, File ID Footer	Display the footer containing file path and creation date if checked (✓).
View, Boundary Axes	Display the boundary axes if checked (✓).
View, Triangles	Display the mesh triangles if checked (✓).
View, Field Contours	Display the field contours if checked (✓).
View, Field Arrows	Display the field arrows if checked (✓).
View, Field Circles	Display the field circles if checked (✓) (rf problems only).
View, Origin…	Show locations for X_{Min},Y_{Min}. The active setting is checked (✓).
View, X Axis…	Direction of X axis (**H**orizontal, **V**ertical). Active setting is checked (✓).
Zoom, **F**orward	Go forward one zoom level.
Zoom, **B**ackward	Go back one zoom level.
Zoom, **O**riginal	Go back to the original display (zoom level 1).
Help, About WSFplot	Brief code description, release date, version number, copyright notice.
Help, About Poisson Superfish	Distribution information, release date, version number, copyright notice.

Table IX-1 lists the WSFplot menu items. Underlined letters on menu items indicate the standard Windows Alt-key method to choose the command. A few of the letters indicated in **Bold** font also are available as single-character commands that serve as shortcuts. For example, pressing the "E" key exits the program and so does pressing Alt-F, E. Note that pressing Alt-C brings up the HardCopy pull-down menu, but pressing only "C" actually starts the hardcopy. These character commands are not case sensitive. The number keys 1 through 9 switch to the numbered zoom level, if available. The Display, View and Zoom menu items are unavailable when WSFplot is in the process of redrawing the plot. On large problems, you may notice that these menu items are grayed out temporarily.

a. Zoom levels

There are two methods for expanding the display. The first method uses menu item Display, Area Boundary, which brings up a dialog window requesting X and Y coordinates for the new display area. In the dialog window, you assign these settings to one of the *zoom levels* already in use, or to the next level available. You can define up to nine zoom levels.

The other method uses the mouse to define two corners of a rectangle in the current display area. Click and hold down the left mouse button to anchor one corner of the rectangle. Move the mouse to frame the desired area. When you release the left mouse

button, the display zooms in on the area within the rectangle. The code assigns the expanded area to the next numbered zoom level (overwriting any selection already occupying that level). If you are already at zoom level 9, the new display replaces the level 9 settings.

The status bar displays the current zoom level and the highest zoom level defined so far. Step backwards or forward though zoom levels with the menu items Zoom, Forward (or the "F" key) and Zoom, Backward (or the "B" key). Menu item Zoom, Original (or the "O" key) returns to zoom level 1. You can also type one of the digits 1 through 9 to go directly to a particular level. When you end WSFplot, a preference file contains data describing all the current screens that you have set up. Upon restarting the code on the same problem (even after changing the geometry and rerunning the solver), WSFplot starts with the last screen you had displayed.

b. Plot title, file ID footer, and numbered axes

Menu item View, Show Title toggles the title at the top of the graphic screen on and off. You can change whether or not to display the title at startup by setting SF.INI variable PlotTitle = On or Off. The plot title consists of either one or two lines from the problem title. Set CombineTitleLines = Yes in the [WSFplot] section of SF.INI to concatenate problem title lines 1 and 2 into a single-line plot title.

Menu item View, File ID Footer toggles on and off a line of text at the bottom right of the display. The line contains the complete file specification of the solution file and time and date of the problem's Automesh input file. You can change whether or not to display the footer at startup by setting SF.INI variable FileIDFooter = On or Off.

Menu item View, Boundary Axes toggles the numbered axes along all four edges of the graphic screen on and off. You can change whether or not to display the axes at startup by setting SF.INI variable BoundaryAxes = On or Off. When the title is displayed, the numbers along the top axis are not displayed.

SF.INI variable AxisLineWidth sets the initial line width in pixels of the boundary axes. The default setting is 1 pixel. If the axis line width is 2 pixels or more, then all the text is bold. After the program is running, select menu item Display, Line Widths to change the line width.

c. Boundary segments

WSFplot always displays the boundary segments that outline each region in the problem geometry. SF.INI variable BoundaryLineWidth sets the initial line width in pixels of the boundary segments. After the program is running, use menu item Display, Line Widths to change the line width.

d. Mesh triangles

Menu item View, Triangles toggles the mesh triangles on and off. You can change whether or not to display the mesh at startup by setting SF.INI variable MeshTriangles = On or Off. SF.INI variable MeshLineWidth sets the initial line width in pixels of the mesh lines. After the program is running, use menu item Display, Line Widths to change the line width.

e. Contour lines

Menu item View, Field Contours toggles the field contours on and off. This feature is inactive if the T35 file contains no solution data. If you do not want to display the contours at startup, set SF.INI variable NumberOfContours = 0. SF.INI variable ContourLineWidth sets the initial line width in pixels of the contour lines. After the program is running, select menu item Display, Line Widths to change line width. Select menu item Display, Contour Lines (or press the "L" key) to specify the number of field contour lines and the minimum and maximum contour values. A dialog window appears with the current settings and information about the minimum and maximum values in the solution file. The SF.INI settings in Table IX-2 establish default values for the field contours. Note that the SF.INI entries for contour values are expressed as a percentage of the range from minimum to maximum.

Table IX-2. Contour settings in SF.INI

Keyword	Description	Default
NumberOfContours	Number of contour lines of either algebraic sign.	30
ContourMinimum	Minimum contour value as a percentage of the range.	0
ContourMaximum	Maximum contour value as a percentage of the range.	100

Configuration variables in SF.INI allow you to control the intervals between contour lines in WSFplot. The program does not include a menu item for controlling the interval. The code displays up to a maximum of $2\times N_C + 1$ contour lines, where N_C is the number of contours of either algebraic sign. The actual number of contours plotted depends upon the values of G_{Max} and G_{Min}, the minimum and maximum values of the solution array, where G refers to either A_z, V, or H_z for problems in rectangular coordinates or to rA_ϕ, V, or rH_ϕ for problems in cylindrical coordinates. If either $G_{Max} = 0$ or $G_{Min} = 0$, or if they have the same sign, then the code will plot exactly N_C contours. The G = 0 contour appears only if G_{Min} is negative and G_{Max} is positive. The code plots the maximum number of contours ($2\times N_C + 1$) when $G_{Min} = -G_{Max}$. Other values for G_{Max} and G_{Min} will result in a varying numbers of contours. The configuration keywords listed in Table IX-3 control the intervals between contour-line values $\pm G(n)$. If G_{Max} and G_{Min} have opposite signs or one of them is zero, then the contour values are given by the equations

$$\pm G(n) = \pm \Delta G n^p, \;\; \Delta G = \frac{MAX\left(ABS\left(G_{Max}\right), ABS\left(G_{Min}\right)\right)}{\left(N_C + 1\right)^p}$$

where n is a contour counter that ranges from 0 to N_C. Exponent p is the power-law setting, and ΔG is the base contour interval. In this case, the first nonzero contour plotted has value $+\Delta G$ or $-\Delta G$. If G_{Max} and G_{Min} have the same sign and neither is zero, the code computes the contour values as follows:

$$\pm G(n) = \pm\left(G_0 + \Delta G n^p\right), \;\; \Delta G = \frac{G_{Max} - G_{Min}}{\left(N_C + 1\right)^p}.$$

where G_0 is the minimum of $|G_{Max}|$ and $|G_{Min}|$. The allowed range for the power p is from 0.1 to 5.0. For most problem types, WSFplot uses p = 1, which results in equal intervals of G between adjacent contour lines.

For Superfish problems with cylindrical symmetry (KPROB = 1, ICYLIN = 1), the default is to use p = 2, which places more contour lines near $rH_\phi = 0$. This setting results in more contour lines in the vicinity of the beam axis where H_ϕ has an approximately linear variation with radius r. In multicell rf cavity problems, the concentration of contour lines near $rH_\phi = 0$ facilitates identification of the resonant mode. If equal intervals of rH_ϕ were used instead, there might be no contour lines in the region where the field reverses. In this case, one could not tell the difference between (for example) the zero mode and π mode of a two-cell structure.

Table IX-3. Power-law settings for WSFplot contour lines.

Keyword	Problem type	Default
RF_XY_PowerLaw	Radio-frequency problems in Cartesian coordinates.	1
RF_RZ_PowerLaw	Radio-frequency problems in cylindrical coordinates.	2
Magnetic_XY_PowerLaw	Magnetic field problems in Cartesian coordinates.	1
Magnetic_RZ_PowerLaw	Magnetic field problems in cylindrical coordinates.	1
Electric_XY_PowerLaw	Electrostatic problems in Cartesian coordinates.	1
Electric_RZ_PowerLaw	Electrostatic problems in cylindrical coordinates.	1

Suppose you want equal intervals of A_z, V, and H_z for problems in rectangular coordinates, but quadratic intervals of rA_ϕ, V, and rH_ϕ for problems in cylindrical coordinates. For this case, use the following settings:

```
[WSFplot]
RF_XY_PowerLaw = 1
Magnetic_XY_PowerLaw = 1
Electric_XY_PowerLaw = 1
RF_RZ_PowerLaw = 2
Magnetic_RZ_PowerLaw = 2
Electric_RZ_PowerLaw = 2
```

f. Field arrows

Menu item View, Field Arrows toggles the field arrows on and off. This feature is inactive if the T35 file contains no solution data. You can change whether or not to display arrows at startup by setting SF.INI variable FieldArrows = On or Off. SF.INI variable ArrowLineWidth sets the initial line width in pixels of the field arrows. After the program is running, select menu item Display, Line Widths to change the line width.

WSFplot draws field arrows on a rectangular grid. The default spacing of 3% is the screen width. You may choose a different spacing by setting SF.INI variable GridSpacing. After the program starts use Menu item Display, Arrow **G**rid Spacing to change the spacing. (Field circles for rf problems use the same grid.)

g. Field circles

For rf programs, menu item View, Field Circles toggles the field arrows on and off. This feature is inactive if the T35 file contains no solution data. You can change whether or

not to display arrows at startup by setting SF.INI variable FieldCircles = On or Off. SF.INI variable CircleLineWidth sets the initial line width in pixels of the field circles. After the program is running, select menu item Display, Line Widths to change the line width.

WSFplot draws field circles on a rectangular grid. The default spacing of 3% is the screen width. You may choose a different spacing by setting SF.INI variable GridSpacing. After the program starts use Menu item Display, Arrow **G**rid Spacing to change the spacing. (Field arrows use the same grid.)

h. Display orientation

WSFplot can plot the picture in any of 8 different orientations as selected by the "origin" location and the direction of the X axis. The origin in this context actually refers to the coordinates X_{Min},Y_{Min} for the current *zoom level* described above. To move X_{Min},Y_{Min} to one of the four corners of the display, select menu item View, Origin… and then choose either LowerLeft (the default), LowerRight, UpperLeft, or UpperRight. A check mark (✓) appears next to the current selection. An alternative method is to use the single-key commands listed in Table IX-4 to make the same selections as the corresponding menu items. These keypad keys occupy the four corners of a square. You can change the default orientation in file SF.INI by setting variable PlotOrigin.

The X axis may be either horizontal (the default) or vertical. Select menu item View, X Axis… and then choose either Horizontal or Vertical to make a change. A check mark (✓) appears next to the current selection. As a shortcut, press either the "H" key or the "V" key. You can change the default direction in file SF.INI by setting variable XaxisDirection.

Table IX-4. Plot orientation menu items.

Menu item	Key commands
View, Origin…, LowerLeft	Keypad 1, End
View, Origin…, LowerRight	Keypad 3, Page Down
View, Origin…, UpperLeft	Keypad 7, Home
View, Origin…, UpperRight	Keypad 9, Page Up
View, X axis…, **H**orizontal	H, h
View, X axis…, **V**ertical	V, v

i. Color settings

The SF.INI configuration variables in Table IX-5 control the colors used for WSFplot displays. If you would like to modify the color settings, run program WSFcolor to find the color numbers to enter in file SF.INI.

Table IX-5. The WSFplot color parameters in SF.INI.

Variable	Controls the color of	Default
TitleAndAxisColor	Plot title and numbered axes.	150 (blue)
BoundaryColor	Boundaries of closed regions.	150 (blue)
MeshTriangleColor	Mesh triangles.	140 (cyan)
FieldContourColor	Contour lines or equipotentials (real part if complex).	190 (magenta)
ImaginaryFieldColor	Imaginary part H_z or rH_ϕ for CFish contour lines.	100 (green)
FieldArrowColor	Arrows showing field strength and direction.	16 (red)
FieldCircleColor	Circles showing H_z or H_ϕ for rf problems.	16 (red)

j. Line widths

Select menu item Display, Line Widths (or the "W" key) to specify lines widths in pixels of several display items. The SF.INI configuration variables in Table IX-6 control the initial line widths used for WSFplot displays. The default line width is 1 pixel. If the boundary-axis line width is greater than 1 pixel, then the screen text will use a **bold** font. The WSFplot preference file stores the most recent line-width settings, which the code will use the next time it starts (overriding the SF.INI settings).

The line-width settings affect both the screen display and the line thickness for hardcopy output. The HardCopy, Options menu includes a setting for line thickness in tenths of a point. The default setting of 0.5 points corresponds to approximately 1 pixel on the display screen. When WSFplot starts a hardcopy, the code first checks the current setting for the hardcopy line thickness. Each time the line width of the display screen changes in accordance with the Table IX-6 settings, WSFplot finds the hardcopy line thickness by multiplying the line width times the initial hardcopy line thickness. For example, if the field contours are 2 pixels wide, and the initial hardcopy line thickness is the default 5 tenths of a point, then the contour thickness on the hardcopy will be 1 point (10 tenths).

Table IX-6. The WSFplot line-width parameters in SF.INI.

Variable	Controls the line width of	Default
ContourLineWidth	Contour lines or equipotentials.	1
ArrowLineWidth	Arrows showing field strength and direction.	1
CircleLineWidth	Circles showing field strength and direction.	1
BoundaryLineWidth	Boundary segments around closed regions.	1
MeshLineWidth	Mesh triangles.	1
AxisLineWidth	Numbered boundary axes.	1

2. Field-display window

The WSFplot field-display window contains information about the problem geometry and the solution at the location of the cursor in the graphics display window. When the mouse cursor is inside the graphics display window it has the crosshair shape. Besides the mouse, you also can move the crosshair with the arrow keys. This method provides a convenient way to move along a horizontal or vertical line. Select menu item Display, Arrow Key Step Size (or the "K" key) to change the step size for each press of an arrow key. The step size is a percentage of the screen horizontal width. If you make the step

smaller than the pixel spacing on your screen, the cursor may not move on each arrow key, but it will eventually move after repeated steps. You can change the default step size in file SF.INI by setting variable ArrowKeyStepSize.

The information included in the field-display window depends upon the type of problem and whether or not the binary solution file includes a completed solution. If only program Automesh has run, then the display includes the following information:

- Material number of the triangle containing the cursor,
- Logical K,L coordinates of the nearest mesh point, and
- Physical X,Y (or R,Z) coordinates.

If a solution exists, then the field-display window includes the field components. The number after "Fn" in the upper right corner of the field-display window indicates the fitting function used to interpolate the fields. For a list of all the fitting functions, see the separate section on the field interpolator. The following sections describe the field components displayed for each problem type.

Because the field interpolator uses first and second nearest neighbors and possibly third nearest neighbors, you may see erroneous values in the field-display window if the mesh is too coarse in the vicinity of thin objects. For example, near a metal plate that is only one row of mesh triangles thick, the code may include points in the fit from the other side of the plate. These interpolated fields will be incorrect. The only solution is to make this part of the mesh finer. You can check the fits by looking at the chi-squared goodness of fit parameter printed by program SF7.

a. Field components for magnetic-field problems

For magnetic-field problems, the program displays field components B_x, B_y, (or B_r, B_z), field magnitude B, and the vector potential A_z (or A_ϕ). Because of symmetry, the vector potential A can have only one nonzero component. The codes store a single number at each mesh point: A_z for X,Y problems or rA_ϕ for cylindrically symmetric R,Z problems. The code lists both A_ϕ and rA_ϕ in the field-display window.

b. Field components for electric-field problems

For electric-field problems, WSFplot displays field components E_x, E_y (or E_r, E_z), the magnitude E, and the electric potential V. The vector sum of the two electric-field components will be perpendicular to the equipotential lines.

c. Field components for rf problems with real fields

For rf problems solved by Fish, Autofish, or one of the tuning programs, WSFplot displays the electric field components E_x, E_y (or E_z, E_r), the magnitude E, the magnetic field H, and the product rH. There is only one nonzero component of H. It is H_z for rectangular X,Y problems, and H_ϕ for cylindrically symmetric Z,R problems. If you have run program SFO to normalize the field to a particular value of E_0, then the field components appear with units that correspond to the normalization. Otherwise, the fields correspond to a value of H = 1000.0 A/m at the drive point.

The field-display window will include the Slater-perturbation frequency shift if the SF.INI variable SlaterTerm = Yes. The code computes the frequency shift df (in kHz) using the following equation for a perturbation volume $\delta V = 1\ mm^3$:

$$df = \frac{k_H \mu H^2 - k_E \varepsilon E^2}{4U} \delta V f\ ,$$

where μ and ε are the material permeability and permittivity, H and E are the magnetic and electric fields, f is the cavity frequency, U is the cavity stored energy, and k_H and k_E are the shape factors for the magnetic and electric terms. For an empty cavity, μ and ε are equal to μ_0 and ε_0. For cylindrically symmetric problems U is the stored energy for the problem geometry. For problems in Cartesian coordinates, U is the stored energy in a 1-meter-long cavity having the cross section of the problem geometry.

In WSFplot, the field display window shows a frequency shift term labeled “Sphere” except when the cursor is near a metal boundary where it displays the “Surface” term. The perturbing volume is assumed to be a metallic object that completely excludes the original electromagnetic fields computed by Superfish. The Surface term corresponds to a perturbation that does not significantly distort the fields, for example, a small displacement of a cavity wall. In a cylindrically symmetric cavity, you could move the cursor along a wall and interpret the frequency shift as a 1-mm wide indentation of the wall (into the cavity) of thickness $1/2\pi R$ around the cavity azimuth, where R is the radial location of the indentation. Away from the cavity boundary, the Sphere term gives the frequency shift produced by a metallic spherical bead of radius 0.620 mm. Program SF7 has an option to include three columns of frequency-shift data for different shape perturbations in Tablplot input files.

d. Field components and power losses for rf problems with complex fields

For rf problems solved by CFish, the field components in the data window are the same ones described above for real fields, but they include both the real and the imaginary parts. By default, the data window includes the power losses in materials if the problem has complex fields and complex permittivity and permeability. These power losses correspond to the region enclosed by the current display boundaries. Because this calculation may take a while for a large number of mesh points, you may wish to turn it off until you are satisfied with the design of the cavity. To turn off the material power-loss calculation, do one of the following:

- set ICCP = 0 or MaterialPowerCode = 0 in the SFO input file, or
- set ICCP = 0 in the REG namelist of the Automesh input file.

The default value of ICCP is 1. If ICCP > 0, then WSFplot calculates power losses in dielectric and magnetic materials. The value of ICCP in Table IX-7 indicates which points on a mesh triangle to use in calculating the average fields for the triangle.

Table IX-7. Options for calculating material power losses.

ICCP	Meaning
0	Do not calculate stored energy and material power integrals.
1	Use the interpolated fields at the triangle center.
2	Average fields at triangle center and edge midpoints.
3	Average fields at triangle center and corners.
>3	Average fields at triangle center, edge midpoints, and corners.

C. Output from WSFplot

WSFplot writes text output file OUTWSF.TXT. The code writes file OUTWSF.TXT if either CreateOutputTextFile = Yes or WriteFieldContours = Yes in the [WSFplot] section of file SF.INI. This file lists the plotted equipotential values in the order WSFplot draws them on the screen. Each output line also includes the symbol and units associated with the contour value.

The code also writes graphics files using one of the available hardcopy drivers and a *preference file* containing saved screen settings to be used the next time WSFplot starts.

1. Writing contour coordinates to OUTWSF.TXT

WSFplot displays lines of constant A_z for rectangular magnet problems, rA_ϕ for cylindrically symmetric magnet problems, V for electrostatic problems, H_z for rectangular rf problems, and rH_ϕ for rf cylindrically symmetric problems. The WriteFieldContour configuration variable in SF.INI specifies whether to write the coordinates of plotted contour lines to file OUTWSF.TXT. The default is not to write contour coordinates. To make WSFplot write this information, use the following settings in SF.INI:

```
[WSFplot]
WriteFieldContours = Yes.
```

WSFplot will write the contour coordinates in OUTWSF.TXT the first time it draws the screen, and once after any change to the number of contours plotted or the range of contours plotted. To stop writing contours, set WriteFieldContours = No. A word of caution is in order regarding these contour coordinates. WSFplot uses a very simple linear interpolation algorithm to find locations on triangle legs for a particular value of H, V, or A. If the mesh is very coarse, you may notice unphysical variation in the direction of a contour line. If you need to trace an equipotential accurately, use as fine a mesh as you can. For an example of the type of problems you might encounter, look at lines of constant A_ϕ near R = 0 for the simple solenoid magnet problem in file MSE2.AM in the directory LANL\Examples\Magnetostatic\StoredEnergy.

2. Hardcopy of graphics screens

The HardCopy, Driver menu lists several software drivers for producing hardcopy graphics files or printing directly to a printer using the Windows Print Manager. Select HardCopy, Start (or the “C” key) to create the hardcopy output. WSFplot re-plots the present screen using the active driver. You can set a preference for the hardcopy driver in

file SF.INI. A check mark (✓) appears next to the currently selected driver. The software drivers have configurable options that you can edit from the HardCopy, Options menu.

The HardCopy, Options menu also includes the full path and name of the graphics output file (if any). You can choose a different name, if desired, but we recommend that you retain the suggested filename extension. In the case of BMP/PCX/PNG bit-image driver, the filename extension will correspond to one of these three graphic types. If you select a different graphic type in the Options menu, and if you do not alter the default filename in any way, then the code will generate the a new default filename with the proper extension. However, the new name will not immediately appear in the File field. If you close the Options dialog and reopen it, the new filename will appear. If you modify the filename, be sure that the extension corresponds to the correct graphic type. The code does not check the extension if you override the default filename.

If the active driver is the BMP/PCX/PNG bit-image driver, then WSFplot sets the default image size to match the current screen size. If you resize the window, the image size also changes. If you want a different size image, you can override the settings HardCopy, Options menu and start a hardcopy before re-sizing the image.

Each hardcopy driver's Options menu includes a setting for the line thickness in tenths of a point. Both this setting and the screen line widths that you select for the various display items (field contours, field arrows, field circles, segment boundaries, mesh triangles, and axis boundaries) affect the hardcopy line thickness as discussed in section B.1.j.

The following sections describe the graphics export formats supported by WSFplot. Much of the information has been supplied with ***Winteracter*** development package of Interactive Software Services Ltd.

a. BMP (Windows Bitmap Format)

Windows Bitmap Format (BMP) is the Windows bitmap format, so many Windows packages support it. Commonly, BMP files are uncompressed, but 16 and 256 color BMP files can also be compressed (but not 16/24/32 bit color files). A few Windows packages do not support the compressed format. Most mainstream packages support compressed files, but the uncompressed format is slightly more universal. Uncompressed files can be very large, so use of the compressed format (or one of the other bitmap formats, PCX or PNG) is usually recommended. The Options dialog allows selection of both compressed and uncompressed BMP formats

b. CGM (Computer Graphics Metafile)

The Computer Graphics Metafile (CGM) standard was created to help standardize graphics output. In practice, it has proved sufficiently ambiguous that each graphics importer or exporter interprets the standard in a different way, thereby adding to the problem instead of solving it. CGM is potentially a good graphics export option since files are compact, it is widely supported as an import format, and its image description capabilities are relatively good. CGM is a vector based format. Only five line types are supported, which is only a limitation with non-Windows packages (because Windows itself only supports 5 line types).

Font support among CGM importers is somewhat variable. The ***Winteracter*** CGM driver used in WSFplot does embed full font information in CGM files. Poisson Superfish codes write Windows-style font names in the CGM files (e.g. “Courier New,” “Arial,” etc.) because they are understood by the widest range of importers among Windows based applications.

c. DXF (AutoCAD Drawing Exchange Format)

The AutoCAD Drawing Exchange Format (DXF) is a vector oriented graphics file format, mainly intended for use with CAD and other drawing packages such as AutoCAD and AutoSketch. It is also supported as an import format by a number of other word processors. The DXF format is extremely verbose, so output files tend to be very large. Font support is erratic amongst different importers, so only one standard “hardware” font is used. DXF provides some basic fill support (3 or 4 sided areas only). The ***Winteracter*** DXF driver supports the full AutoCAD set of 255 colors. Some packages support a more limited palette of 14 distinct colors, of which only the first 7 colors are compatible with the full AutoCAD set. The DXF driver Options dialog can select either 255, 14, or 7 colors depending on the target importer software capabilities.

d. PS (PostScript) and EPS (Encapsulated PostScript)

The PostScript (PS) driver can generate either standard PostScript files or Encapsulated PostScript (EPS) files. The EPS format is most useful when the finished output is destined for a PostScript printer. Most EPS importers make no attempt to actually decode the contents of an Encapsulated PostScript file. Instead, the EPS file is read “as-is”, on the assumption that the PostScript output device will make sense of the file. Usually, an EPS file appears as an empty frame in most word processing programs. Two notable exceptions are Corel Draw 7 and GhostScript/GSView, both of which can interpret and display PS and EPS files.

e. HP-GL and HP-GL/2 (Hewlett Packard Graphics Language)

HP-GL is the graphics language used by Hewlett Packard pen plotters. Many packages claim to be able to read HP-GL, but each has its own interpreter and some are better than others. Some even vary between releases (e.g. WordPerfect for Windows v6.0 is completely unusable with virtually any HP-GL file). Despite this, HP-GL is a good format to try for simple line graphics of the type generated by WSFplot. Other formats are better if solid fills are required. The most common problem encountered when importing HP-GL is color selection. WSFplot translates color numbers 0 through 255 as indicted in Table IX-8. There is no standard for what HP-GL pen numbers mean. They are simply numbered from 1 upwards. On most devices the pen numbers will produce the color shown in the table.

Table IX-8. HP-GL Color Translation.

Color number	HP-GL pen	Color
16 to 47	2	red
48 to 79	4	yellow
80 to 111	3	green
112 to 143	7	cyan
144 to 175	5	blue
176 to 207	6	magenta
all other	1	black

HP-GL/2 is a later variant of HP-GL widely implemented on modern HP laser printers and high end inkjet printers. We do not recommend the use of HP-GL/2 as an export format, since most applications are very poor at importing this newer implementation of HP-GL. HP-GL/2 is best used as a hardcopy (rather than export) format, where it can produce reasonably high quality output.

f. PCX (ZSoft PC Paintbrush Format)

The Paintbrush PCX format is probably the one of the most reliable formats. The color depth of the saved file affects its size. Monochrome images are smallest. A few software packages do not recognize the 24-bit PCX format.

g. PNG (Portable Network Graphics)

Portable Network Graphics (PNG)is a bit image format developed as a replacement for the GIF file format. It supports much better compression than either PCX or BMP, but is currently (late 2001) less widely supported. PNG is a supported standard in Internet Explorer and Netscape. Thus, PNG is a good choice when publishing images on the web or on an intranet.

h. SVG (Scalable Vector Graphics)

The Scalable Vector Graphics (SVG) format has been defined by the World Wide Web Consortium (W3C) to allow web browser display of re-scaleable vector based images. As such, SVG is effectively a web equivalent of the PostScript or CGM formats. For viewing SVG files in your web browser download the free plug-in from the Adobe web site. SVG is an ASCII format that appears to be fully featured, offering a comprehensive and reliable set of 2D graphics capabilities.

i. WMF (Windows Metafile) and EMF (Enhanced Windows Metafile)

Windows Metafile (WMF) is the “native” Windows metafile format. Logically, WMF files are similar to CGM files, but their internal structure is very closely linked to that of the Windows application programming interface (API). As with most areas of Windows programming WMF exists in several variants. The basic 16-bit “Standard WMF” format is the most widely supported, with the notable exception of some versions of Word and Excel, which require a slightly different format known as an Aldus Placeable Metafile. Since most importers also accept Aldus format WMF files, the ***Winteracter*** WMF driver generates Aldus format files by default. In the HardCopy Option menu, you can select the Standard WMF format or a third EMF (“Enhanced WMF”) format. Microsoft introduced

EMF for the Win32 API. Some popular applications (e.g. Word 7.0) cannot read EMF files. Most Windows packages support at least one variant of WMF as an import format, though their ability to read them correctly varies. The least reliable importers are those that need to understand the precise contents of the WMF file. Such applications (typically drawing packages) provide their own WMF interpreter. The more reliable importers are those that simply include the picture as an object, relying upon the appropriate Windows API function to replay the metafile.

In common with the rest of the Windows Graphics Device Interface (GDI), WMF files have just 5 line types, rather than the 7 nominally supported in ***Winteracter***. Certain line types will be indistinguishable from one another. Clipping of circles, arcs, ellipses and non-hatched polygons only works in Enhanced WMF files. Clipping is not handled correctly in 16-bit WMF files, so it is disabled in the ***Winteracter*** WMF driver when generating Standard or Aldus format files. A Windows API restriction limits 16-bit Standard WMF files to short (8.3 format) filenames. The Aldus variant of this format is not subject to this limitation, but the Windows API limits the Aldus file size to 16 MB.

3. WSFplot preference file

WSFplot writes a *preference file* containing saved screen settings to be used the next time the program starts. The name and location of the file depends upon the SF.INI variable SaveSettingsTo. The choices are:

SaveSettingsTo = Local Directory
SaveSettingsTo = Individual File
SaveSettingsTo = *filename*
SaveSettingsTo = *directory*

where *filename* is the name of an actual file (including path) to be created and *directory* is the name of an existing directory. If variable SaveSettingsTo = Local Directory (the default), then the code creates file WSFPRF.TXT, in the current directory. If variable SaveSettingsTo = Individual File , then the code writes an output file named after the Automesh input file with characters PRF appended and with extension TXT. For any setting other than “Local Directory” or “Individual File,” WSFplot checks first to see if the entry is a valid directory name. If so, it reads or creates preference file WSFPRF.TXT in that directory. If the entry is not a directory, but is an existing file (or the file can be created), then that file is the preference file. Otherwise, WSFplot assumes SaveSettingsTo = Local Directory.

The preference file is especially useful when you need to look at the same area of the geometry after repeated runs of the solver program (Fish, CFish, Poisson, or Pandira), for example, after making small changes in electrode shapes. You can change the geometry and rerun the solver program, and then view the same detailed portion of the mesh viewed the last time WSFplot ran. If you are working on more than one problem, choose SaveSettingsTo = Individual File to retain settings for each problem. The preference file includes the following information:

1. window size and location,
2. currently active hardcopy graphics driver,

3. currently active graphic elements (title, number axes, mesh triangles, field contours field arrows, field circles),
4. field and circle grid spacing,
5. number of contours,
6. line widths of field contours, field arrows, field circles, segment boundaries, mesh triangles, and axis boundaries,
7. problem filename and first title line,
8. plot orientation,
9. boundaries for all zoom levels, and
10. current zoom level.

If SF.INI variable UseSavedSettings = Yes, then WSFplot will attempt to use information stored in the preference file. WSFplot always uses items 1 through 6, but it uses items 8 though 11 only if the present solution filename and title match the information in item 7. If you start WSFplot and would rather not use previously saved screen settings, select menu item Display, ReSet Defaults (or press the "S" key). You also can simply delete the preference file before starting WSFplot to accomplish the same thing. Deleting the preference file also restores the original window sizes and locations.

D. WSFplot error messages

Table IX-9 lists the WSFplot error messages. The error numbers also are available as exit error codes to batch control files..

Table IX-9. WSFplot 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.
205	File entered from File, Open 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 starting with Automesh.
221	An error occurred reading a record from the solution file. Please report this error.
292	Insufficient memory, cannot allocate MT namelist arrays for a Poisson problem.
293	Insufficient memory, cannot allocate MT namelist arrays for a Superfish problem.
299	Insufficient memory, cannot allocate arrays for binary solution file data.

X. SFO: Poisson Superfish Postprocessor

SFO stands for Superfish Output. For Superfish problems, SFO reads the electromagnetic field solution written by Fish or CFish and calculates parameters of interest to the accelerator designer. For Poisson or Pandira problems, SFO reads the vector potential or the scalar potential solution and calculates field components along boundary segments.

SFO must run after Fish, CFish, Poisson, or Pandira. The codes run in the following order:

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

For Superfish problems, 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). For Poisson and Pandira problems, there is no difference in the field components before and after running SFO.

SFO was originally designed for drift-tube-linac cells, but it also provides useful information for coupled-cavity linac cavities and other rf cavity geometries such as RFQs. Most features in the code are for rf accelerating cavities.

A. Starting program SFO

There are several ways to start SFO. Double-click on a file with the SEG extension, right-click on the SEG file and select Open, or right-click on the Poisson Superfish solution file with extension .T35 and select “Postprocess (SFO).” To open the SEG file in Notepad, right click on the file and choose Edit. You can also use a command line to start the code:

SFO *segfile* *T35file*

The filenames on the command line are optional. The parameter *segfile* is the name of an input file, with extension SEG, and *T35file* is the name of a binary solution file. Both *segfile* and *T35file* have the same name as the Automesh input file, but different extensions. The code will read the *segfile* even if does not appear on the command line. It is usually not necessary to enter the name of the T35 solution file immediately after running a solver program (Fish, CFish, Poisson, Pandira) because the code will find the name of the solution file to open in file TAPE35.INF. Launching a program from the Windows Start menu does not supply any information on the command line. 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.

If you have supplied the name of a valid *segfile*, or if SFO finds the default input file, then entries in the input file control operation of the program. Lines in the *segfile* contain the list of segment numbers on which to compute fields, other tables of input data and problem variables that you wish to change from the values set by Automesh.

B. SFO files and filename conventions

SFO reads the binary solution file written by Fish, CFish, Poisson, Pandira, or SFO itself. It rewrites the file after calculating the requested fields and other data. The code writes an output file named after the Automesh input file, but with extension SFO. The length of the SFO output file depends upon details of the problem geometry. The file can be very long for a problem with many surfaces and a fine mesh. You can supply an input *segfile* containing new values for problem variables.

SFO writes an optional Tablplot file called Transit.TBL, which contains the integrands of the transit-time factor integrals. Another option writes the Tablplot file TBeta.TBL (or TBeta01.TBL, TBeta02.TBL, etc. for multiple cells). These files contain transit-time data as a function of particle velocity. If the ParmilaData option in SF.INI has been activated, then SFO writes the PMI file for the ion-linac design code Parmila. For more information, see the section titled Files and filename conventions.

C. Problem variables for SFO

Table X-1 lists problem variables that SFO can read from file *segfile*, the SFO input file. Except for the table entries, you can supply the data as variables in the first REG namelist of the Automesh input file. When entering variables upon starting SFO, use either the variable name or the more descriptive input file keyword from Table X-1. SFO may need one or more of the data tables listed in Table X-2. All tables end with the keyword EndData.

Table X-1. Problem variables used by SFO.

Variable	Input file keyword	Default value	Description
ALPHAT	ResistanceCoefficient	3.93×10^{-3}	Temperature coefficient of resistance at TEMPR (C^{-1}).
ASCALE	ScalingFactor	$4\pi c \times 10^{-8}$	Scale factor (default sets drive H = 1000 A/m)
BETA	ParticleVelocity	0	Particle velocity (used if KMETHOD = 1).
BETA1	StartingBeta		Starting BETA for table of transit-time data.
BETA2	EndingBeta		Ending BETA for table of transit-time data.
CCLMINK	CouplingMinimum	1	Lowest coupling for coupling-slot power table.
CCLMAXK	CouplingMaximum	6	Highest coupling for coupling-slot power table.
CCLDELK	CouplingIncrement	1	Increment in the coupling-slot power table.

Table X-1. Problem variables used by SFO. (continued)

Variable	Input file keyword	Default value	Description
CLENGTH	CavityLength	0	User-supplied cavity length for defining E_0.
DBETA	BetaIncrement		BETA increment for table of transit-time data.
DPHI	GeometryPhaseLength	180	Phase length (degrees) (used if KMETHOD $\neq 1$).
ENORM	En	10^6	Normalize to ENORM in V/m when NORM = 4.
EZERO	E0	10^6	Normalize to E_0 in V/m when NORM = 0.
EZEROT	E0T	10^6	Normalize to E_0T in V/m when NORM = 1.
FREQD	DesignFrequency		Design frequency (MHz).
HPHI	SegmentH	5.0×10^3	Normalize to H_ϕ (A/m) on segment end if NORM = 2.
IBETA	TransitTimeTable	0	If>0, write transit-time table from BETA1 to BETA2.
ICCP	MaterialPowerCode	1	>0 for material power losses.
ICORNER1	FirstCornerArcSegment	0	First segment of corner arc for finding average H.
ICORNER2	LastCornerArcSegment	0	Last segment of a corner arc for finding average H.
IOBSEG	OuterBoundarySegment	–1	Starting segment of the CCL outer boundary.
IRTYPE	RsOption	0	Rs option: 0: normal; 1: superconductor; 2: RS; 3: RHO.
ISLOT	SlotPowerCode	0	If 1, estimate coupling-slot power losses.
ITFILE	TplotCode	0	If 1 in SFO, write a plot file of transit-time data.
KMETHOD	WaveNumberCode	0	Compute wave number from DPHI (0), BETA (1).
LINT	LogicalRow	1	Logical-mesh coordinate for E_z dz integration.
NHSTEM	NumberOfHalfStems	1	Number of half stems on the symmetry plane.
NORM	NormalizationCode	0	0: E_0; 1: E_0T; 2: HPHI; 3: ASCALE; 4: point to point.
NRMSEG	NormalizationSegment	1	Normalization segment when NORM = 2.
PLCELL	MulticellPhaseLength	360	Phase length (deg) per cell for a multicell problem.
RESIDR	ResidualResistance	10^{-8}	Residual resistance for superconductors (Ω).
RFMU	Permeability	1.0	Relative permeability for rf surface resistivity.
RHO	Resistivity	1.7241×10^{-6}	User-supplied material bulk resistivity (Ω-cm).
RHOR	ReferenceResistivity	1.7241×10^{-6}	Reference resistivity (Ω-cm) at T = TEMPR.
RMASS	RestMassEnergy	–2 (H^+)	Rest-mass energy (MeV) or particle-type code.
RS	SurfaceResistance	0	RF surface resistance (Ω).
RSTEM	StemRadius	1	Stem radius for stems along boundary segments.
SLOSS	SlotPowerIncrease	0.03	Coupling-slot power increase per percent coupling.
TC	SCcriticalTemperature	9.2	Critical temperature for superconductors (K).
TEMPC	Temperature	20	Operating temperature for normal conductors (C).
TEMPK	SCtemperature	2	Operating temperature for superconductors (K).
TEMPR	ReferenceTemperature	20	Reference temperature for normal conductors (C).
XNORM1	Xstart		Starting X coordinate for NORM = 4 option.
YNORM1	Xend		Starting Y coordinate for NORM = 4 option.
XNORM2	Ystart		Ending X coordinate for NORM = 4 option.
YNORM2	Yend		Ending X coordinate for NORM = 4 option.
ZCTR	ZCenterLocation	0	Electric center of a cell (usually where field peaks).

The most commonly used data entry in SFO is that of the boundary segment numbers on which to compute the power and fields. If you do not specify a FieldSegments list in the SFO input file for Superfish problems, then SFO tries to identify the metal boundaries for

a typical cavity geometry. You should always check whether the included segments make sense for your problem if you allow the code to find the segments.

1. Static-field solvers Poisson and Pandira use the FieldSegments table

SFO calculates field components along boundary segments for Poisson and Pandira problems. You supply the list of boundary segments in the same way that it is done for rf field problems. The keyword FieldSegments followed by the segment numbers and keyword EndData defines the boundary segments on which to calculate the fields. If you do not specify a FieldSegments list in the SFO input file for Poisson problems, then SFO calculates the fields along all segments. File OUTAUT.TXT lists the boundary segment numbers for each region (after any line regions have been added by Automesh).

Table X-2. Table headings in the SFO input file.

Input file table-heading keyword	Description
FieldSegments	Boundary segments for field calculations.
CellData	Cell properties for a multicell problem.
StemData	Stem and post-coupler properties in a multicell tank.
ResistanceSegments	Unique surface-resistance data for defined segments.

2. Running SFO more than once on the same problem

SFO uses variables stored in the binary solution file by Automesh, the solver code (Fish CFish, Poisson or Pandira) or by a previous run of SFO unless you override a value in the input file. Thus, if you wish to change a variable back to its default value, it many not be sufficient to simply remove the entry from the SFO input file. For example, suppose you started SFO with the input file shown in Figure X-1. The second line sets CLENGTH = 5.25 cm, which may differ from ZLONG, the length of the mesh problem. The length CLENGTH defines E_0 for field normalization. Ordinarily, SFO would use the calculated length ZLONG to compute E_0. Now, suppose that you run SFO again after removing (or commenting) the CavityLength line. This run would still use CLENGTH, and not ZLONG, for normalization because the stored value of CLENGH is nonzero after the first run. To make SFO use ZLONG, you must reset the value of CLENGTH to zero as in Figure X-2.

```
StemRadius = 0.635          ; Stem radius in cm
CavityLength = 5.25         ; Cavity length for normalization in cm
GeometryPhaseLength = 180   ; Phase length of the problem geometry in degrees
FieldSegments               ; Start of list of segments included in power and field calculations
2 -3 4 5 6 7 8              ; Segment numbers
EndData                     ; End of FieldSegments list
End                         ; End of input file
```

Figure X-1. SFO input file for first code run.

```
StemRadius = 0.635              ; Stem radius in cm
CavityLength = 0                ; Seting Cavity length = 0 forces use of geometry length ZLONG
GeometryPhaseLength = 180       ; Phase length of the problem geometry in degrees
FieldSegments                   ; Start of list of segments included in power and field calculations
2 -3 4 5 6 7 8                  ; Segment numbers
EndData                         ; End of FieldSegments list
End                             ; End of input file
```

Figure X-2. SFO input file for second code run.

D. Field normalization in SFO

Table X-3 lists variables that affect how SFO scales the fields for power and peak surface field calculations in Superfish problems. Table X-4 gives the available normalization methods. Poisson and Pandira problems do not require normalization. Several examples for the pillbox cavity show how to supply these normalization options to program SFO after having solved the problem in Fish. One of the five variables ASCALE, EZERO, EZEROT, HPHI, or ENORM sets the scale for power and field calculations in SFO. ASCALE is the scaling factor for the magnetic field array. Automesh sets the default value for ASCALE to $4\pi c \times 10^{-8}$ where c is the speed of light in cm/s. This value of ASCALE corresponds to the field H_ϕ = 1000 A/m at the drive point. SFO recalculates the value of ASCALE unless you use the NORM = 3 method.

Table X-3. Variables that affect the Superfish field normalization.

Variable	Default value	Description
ASCALE	$4\pi c \times 10^{-8}$	Scaling factor for fields used in SFO and SF7.
CLENGTH	0	User-supplied cavity length for defining E_0.
ENORM	10^6	Normalize to ENORM in V/m when NORM = 4.
EZEROT	10^6	Normalize to E_0T in V/m when NORM = 1.
EZERO	10^6	Normalize to E_0 in V/m when NORM = 0.
LINT	1	Logical coordinate for E_z integration.
HPHI	5.0×10^3	Normalize to H_ϕ in A/m on segment end when NORM = 2.
NORM	0	0: EZERO; 1: EZEROT; 2: HPHI on segment; 3: ASCALE 4: Normalize to an integral between specified points.
NRMSEG	1	Segment number for normalization when NORM = 2.
XNORM1		Starting X coordinate for NORM = 4 normalization option.
YNORM1		Starting Y coordinate for NORM = 4 normalization option.
XNORM2		Ending X coordinate for NORM = 4 normalization option.
YNORM2		Ending X coordinate for NORM = 4 normalization option.
ZLONG		E_0 integration length, and cavity length if CLENGTH = 0.

Table X-4. The field normalization methods.

NORM	Description
0	SFO uses EZERO to determine ASCALE (the default).
1	SFO uses EZEROT to determine ASCALE.
2	HPHI on segment NRMSEG determines ASCALE.
3	ASCALE is not changed by SFO.
4	Normalize the integral from XNORM1,YNORM1 to XNORM2,YNORM2 to ENORM.

1. Using the NORM=0 option: normalize to E_0

The usual method (NORM = 0) for defining the field normalization is to adjust the value of ASCALE so that

$$E_0 = \frac{1}{L}\int_{Z_s}^{Z_e} |E_z| dz .$$

where L (= ZLONG = $Z_e - Z_s$) is the cavity length, E_z is the longitudinal component of electric field, and E_0 is the value of EZERO. SFO calculates ZLONG from the problem geometry. If you supply a value for CLENGTH, then SFO performs the integration over length ZLONG, but divides by CLENGTH to calculate E_0. If the problem geometry includes a bore-tube extension, you can supply CLENGTH to define E_0 in terms of a "cell" length instead of the problem-geometry length. If CLENGTH differs from ZLONG, then E_0 is given by:

$$E_0 = \frac{1}{L_c}\int_{Z_s}^{Z_e} |E_z| dz .$$

where ZLONG = $Z_e - Z_s$, and L_c is CLENGTH. Using the CLENGTH option affects only the field normalization. It does not affect transit-time factor integrals. The value of CLENGTH also affects the calculation of shunt impedance. The shunt impedance is defined as:

$$Z = \frac{E_0^2}{P/L} \qquad \text{if CLENGTH = 0,}$$

or as

$$Z = \frac{E_0^2}{P/L_c} \qquad \text{if CLENGTH} \neq 0.$$

2. Using the NORM=1 option: normalize to E_0T

For cylindrically symmetric problems in which SFO calculates E_0 along the axis (R = 0), you can specify the product E_0T, where T is the transit-time factor. To use this option enter a value for EZEROT and set NORM = 1. The same options involving the length CLENGTH (discussed in section 1 above) apply to the definition of E_0. If NORM = 1, SFO sets EZERO = EZEROT/T after calculating T.

3. Using the NORM=2 option: normalize to a value of H_ϕ

If NORM = 2, then SFO scales the fields so that H_ϕ equals the value HPHI at the end of segment NRMSEG. The NORM = 2 option does not require cylindrical symmetry, whereas the calculation of E_0 for the NORM = 0 and NORM = 1 options requires both cylindrical symmetry and nonzero axial E_z. If NORM = 2, and the problem has cylindrical symmetry and nonzero axial E_z, then SFO will report the computed values of EZERO and EZEROT.

4. Using the NORM=3 option: use default or user-supplied ASCALE

When NORM = 3, SFO uses the current value of ASCALE for field normalization. For all other normalization options, SFO computes the value of ASCALE. Program Automesh sets the default value for ASCALE to $4\pi c \times 10^{-8}$, which corresponds to a field of 1000 A/m at the drive point. If NORM = 3, and the problem has cylindrical symmetry and nonzero axial E_z, then SFO will report the computed values of EZERO and EZEROT.

5. Using the NORM=4 option: normalize to a field integral between points

The NORM = 4 option adjusts ASCALE to make the electric-field integral between two physical points in the geometry equal ENORM. The following equation states the requirement:

$$\text{ENORM} = \frac{1}{L_p}\int_{p_1}^{p_2} E_p dp\,,$$

where E_p represents the component of E parallel to the straight-line path from point p_1 (XNORM1,YNORM1) to point p_2 (XNORM2,YNORM2); dp is a distance interval along the path; and L_p is the total path length [= SQRT((XNORM2 – XNORM1)^2 + (YNORM2 – YNORM1)^2)]. This option is convenient for a cylindrically symmetric problem that has no axial component of electric field. For example, in a coaxial waveguide the electric field direction is always radial (for the cutoff mode). However, like the NORM = 2 and NORM = 3 options, SFO will report the computed values of EZERO and EZEROT if the problem does have nonzero axial E_z. The NORM = 4 option is also applicable to problems with Cartesian coordinates.

SFO chooses the number of steps based upon the average mesh interval between the two points. The number of steps will be approximately 4 times the total number of mesh triangles between the points. For a very coarse mesh, the code uses at least 20 steps. The two end points and all the intervening steps must be inside the problem geometry. SFO will report an error if it encounters a point along the path that is out of bounds or inside a conductor. Directory LANL\Examples\RadioFrequency\QuarterWaveResonator contains an example file that uses the NORM = 4 option for normalization.

6. Off-axis integration of E_z

For the NORM = 0 and NORM = 1 options, SFO integrates the electric field component E_z along the logical row LINT. The default value is LINT = 1 and corresponds to the cavity axis. When using an off-axis integral, add a Y line region along the row of interest (and also set LINEY = 1 for that region) to ensure that all points along the line have the same radial coordinate.

E. Options for calculating surface resistance

Program SFO has several options for specifying how to determine the surface resistance for rf power calculations. The variable IRTYPE selects one of the methods listed in Table X-5. The default value for IRTYPE is zero. The variables in Table X-6 specify other parameters used to compute the surface resistance. For the resistivity of cavities not made

of copper, refer to Table X-8 for common metals and to Table X-9 for some brazing alloys. If NRESIST is nonzero, then SFO reads a table of surface-resistance data for individual segments.

In SFO, the default surface corresponds to the 100% IACS resistivity for normal-conducting copper at 20 degrees C, standard ASTM B193-95. IACS is the International Annealed Copper Standard. [Note: versions of SFO before September, 1995 used the nominal value 1.7 μΩ-cm for the resistivity of room-temperature copper.]

1. Surface resistance for normal conductors

Program SFO uses variables RHOR, TEMPR, ALPHAT, TEMPC, and RFMU from Table X-6 to calculate the rf surface resistance of a normal conductor if IRTYPE is 0, which is the default. Temperatures are in degrees C, RHOR is in Ω-cm, and ALPHAT is in reciprocal degrees C. For normal conductors, SFO calculates the surface resistance using the skin depth δ as follows:

$$R_s = \frac{1}{\sigma\delta},$$

where σ is the dc conductivity, the reciprocal of the resistivity ρ. The skin depth is

$$\delta = \sqrt{\frac{2}{\sigma\mu\omega}},$$

where ω = 2πf and f is the resonant frequency, and μ is the permeability given by

$$\mu = \mu_r\mu_0,$$

where μ_r is the relative permeability RFMU, and $\mu_0 = 4\pi \times 10^{-7}$ T-m/A. For rf cavity problems, a typical magnetic field H_ϕ is of the order of 10^4 A/m, which corresponds to a magnetic induction B ~ 120 Gauss. Thus, magnetic materials generally do not reach saturation when subjected to the rf-cavity fields. A low-field value of the permeability for a magnetic material (in the linear portion of the material's B-H curve) is sufficient for computing the skin depth. The code computes the resistivity RHOC at a specified temperature according to the formula:

$$\text{RHOC} = \text{RHOR}\,[1 + \text{ALPHAT}\,(\text{TEMPC} - \text{TEMPR})],$$

where RHOR is the reference resistivity at temperature TEMPR, and ALPHAT is the temperature coefficient of resistance at temperature TEMPR. If a copper cavity will operate at a temperature other than 20 degrees C, enter the operating temperature as TEMPC. For other normal-conducting materials, you can enter values for the resistivity and the temperature coefficient of resistance at a known temperature. Refer to Table X-8 for the resistivity of some common metals and to Table X-9 for the resistivity of several brazing alloys. In the tuning programs, change a normal conductor's surface resistance with control-file keyword TEMPerature.

Table X-5. Methods for computing surface resistance.

IRTYPE	Description
0	Use normal-conductor parameters ALPHAT, RFMU, RHOC, RHOR, TEMPC, and TEMPR.
1	Use superconductor parameters RESIDR, TC, and TEMPK.
2	User has supplied a value for the surface resistance RS.
3	Calculate RS from user-supplied resistivity RHO and permeability RFMU.

Table X-6. Variables that affect the surface resistance.

Variable	Default value	Description
ALPHAT	3.93×10^{-3}	Temperature coefficient of resistance at TEMPR (C^{-1}).
IRTYPE	0	0: normal; 1: superconductor; 2: user RS; 3: user RHO.
NRESIST	(calculated)	Number of segments with different surface resistance.
RESIDR	10^{-8}	Residual resistance for superconductors (Ω).
RFMU	1.0	Relative permeability for rf surface resistivity.
RHO	1.7241×10^{-6}	User-supplied material bulk resistivity (Ω-cm).
RHOC	Calculated	Calculated resistivity (Ω-cm) for normal conductors.
RHOR	1.7241×10^{-6}	Reference resistivity (Ω-cm) at temperature TEMPR.
RS	0	RF surface resistance (Ω).
TC	9.2	Critical temperature for superconductors (K).
TEMPC	20	Operating temperature for normal conductors (C).
TEMPK	2	Operating temperature for superconductors (K).
TEMPR	20	Reference temperature for normal conductors (C).

2. Surface resistance for superconductors

Program SFO uses variables TEMPK, TC, and RESIDR from Table X-6 to calculate the rf surface resistance of a superconductor if IRTYPE is 1. By default, IRTYPE = 0 and SFO assumes a normal conducting surface. If IRTYPE = 1, SFO uses the Piel semi-empirical formula to calculate the surface resistance:

$$R_s = 9 \times 10^{-5} \frac{f^2(\mathrm{GHz})}{T} \exp\left(-1.83 \frac{T_c}{T}\right) + R_{\mathrm{residual}},$$

where T is the operating temperature entered as TEMPK, T_c is the critical temperature entered as TC, and $R_{residual}$ is the residual resistance entered as RESIDR. Temperatures are in degrees K, and RESIDR is in ohms. In the tuning programs, enter superconductor surface-resistance data with control-file keyword SUPERConductor.

3. Entering a known surface resistance

Variable RS is the surface resistance for power calculations in SFO. If IRTYPE is 0, 1, or 3, then RS is a calculated quantity. You can specify a known value RS by using the IRTYPE = 2 option. When IRTYPE = 2, there is no default setting for RS; you must supply its value. The default surface is room-temperature copper, which has a bulk resistivity of 1.7241 μΩ-cm. In the tuning programs, use control-file keyword R_Surface to enter a known surface resistance.

4. Entering a known value for the bulk resistivity

If you know from measurements or published data the resistivity for a cavity and you do not want SFO to compute the temperature effect, set IRTYPE = 3 and enter the material's resistivity as RHO. The default value for RHO corresponds to the SFO default surface of room-temperature copper, which has resistivity

$$\rho = 1.7241\ \mu\Omega\text{-cm}.$$

For other normal-conducting materials, set IRTYPE = 3 and change RHO to the appropriate value. The calculation for IRTYPE = 3 includes the effect of the material's magnetic permeability supplied in RFMU. The default value for RFMU is 1.0. In the tuning programs, use control-file keyword RESISTivity to enter a known resistivity.

5. Entering surface resistance data for individual segments

The surface-resistance calculated or entered as discussed in sections E1 to E4 above becomes the default value of R_s for all the boundary segments in the problem. In some cases, it may be necessary to change the surface resistance for one or more individual segments. For example, the beam pipe attached to a superconducting cavity may have a normal-conducting section in a region of low field. Another example would be to include the effect of a braze joint in the cavity. Table X-9 lists some measurements of braze-alloy resistivities made in 1981 at the Chalk River Nuclear Laboratories.

Use a ResistanceSegments table to change the surface resistance settings for one or more segments. Each segment requires one line of data starting with the segment number ISEG and the surface-resistance option IRTYPE. The rest of entries on a line are all optional and they depend upon the value of IRTYPE as defined in Table X-7. If you omit any of the variables, the code uses the problem variables of the same name that have already been defined. For example, consider the SFO input file in Figure X-3.

```
FieldSegments              ; Start of list of segments included in power and field calculations
3 4 5 6 7 8 9 10 11 12     ; Segment numbers
EndData                    ; End of segment list
ResistanceSegments         ; Start of surface resistance table for individual segments
10 0 24                    ; Segment 10 data
11 0 28                    ; Segment 11 data
12 0 32                    ; Segment 12 data
EndData                    ; End of surface resistance table
End                        ; End of SFO input file
```

Figure X-3. SFO input file containing surface resistance data.

The first three lines list segments that SFO should include in the power and field calculations. The ResistanceSegments table includes three segments have special values of some surface resistance parameters. The entries specify IRTYPE = 0 for segments 10, 11, and 12. The temperature of segment 10 is 24 degrees, segment 11 is 28 degrees, and segment 12 is 32 degrees. The default operating temperature of 20 degrees C applies to all the other segments. Because no entries for RHOR, TEMPR, ALPHAT, or RFMU appear on the three lines of the table, SFO uses the same surface-resistance variables as it uses for the other segments when calculating R_s for these segments. You can omit only trailing parameters on the surface-resistance data lines. For example, if you need to

change RHOR, you must supply a value for TEMPC, which precedes RHOR on the line. Do not use zero as a placeholder.

Table X-7. Variables for each method of computing surface resistance.

IRTYPE	Variables
0	ISEG,IRTYPE(=0),TEMPC,RHOR,TEMPR,ALPHAT,RFMU
1	ISEG,IRTYPE(=1),TEMPK,TC,RESIDR
2	ISEG,IRTYPE(=2),RS
3	ISEG,IRTYPE(=3),RHO,RFMU

6. Resistivity of common metals and brazing alloys

For reference, Table X-8 lists the resistivity, the temperature coefficient of resistance, and the magnetic permeability (if available) for some common metals. The data are from: Robert C. Weast, editor-in-chief, CRC Handbook of Chemistry and Physics, 64th Edition (1983-1984). The skin depth for highly permeable materials can be very much smaller than for most normal-conducting metals because the rf magnetic field tends to be excluded from the material. SFO includes the effect of the permeability in computing the skin depth.

One reason to enter resistivity data for individual segments in SFO is to include the effects of brazing alloys. Table X-9 lists, in order of decreasing liquidus temperature, some measurements of resistivities made at the Chalk River Nuclear Laboratories [R. B. Turner and J. Ungrin, “Brazing Techniques and Alloys for Accelerator RF Components,” Proceedings of the 1981 Linear Accelerator Conference, Los Alamos National Laboratory report LA-9234C, Santa Fe, NM (October 19-23, 1981)]. The measurements reported in Table X-9 used small foil or wire samples obtained from the Western Gold and Platinum Company, indicated by “(W)” after the trade name, or from the Hardy and Herman Company indicated by “(H)” after the trade name. The resistivities in parentheses are values supplied by the manufacturer.

Turner and Ungrin estimate the error in the resistivity at less than 10%. The authors note that the composition and resistivity of an alloy may change during the braze. The properties after brazing can depend on the metals joined, the braze temperature, and on subsequent braze cycles. For example, although the small amount of lithium increases substantially the measured resistivity of the Lithobraze 720 BT alloy (compared to Cusil or Braze 720), the brazed joint may have a lower resistivity because the Li is consumed as a deoxidizer.

The Chalk River measurements were made at room temperature. No attempt was made to measure the temperature coefficient of resistance nor the magnetic permeability of these alloys. Copper, gold, and silver, which make up the bulk of most of these braze alloys, all have a temperature coefficient of resistance in the range 0.0034/C to 0.0039/C. Alloys containing nickel may be magnetic, which may seriously degrade their electrical conductivity in an rf cavity environment. For the rest of the alloys, it is probably safe to assume that $\mu_r = 1$.

Table X-8. Properties of common metals.

Metal	Resistivity at 20° C (μΩ-cm)	Temperature Coefficient/°C at 20° C	Relative Permeability at 20 Gauss
Aluminum	2.824	0.0039	1
Bismuth	120	0.004	1
Brass	7	0.002	1
Cadmium	7.6	0.0038	1?
Cobalt	9.8	0.0033	~160
Constantan	49	0.00001	
Copper, annealed	1.7241	0.00393	1
Copper, hard-drawn	1.771	0.00382	1
Gold	2.44	0.0034	1
Iron, 99.98% pure	10	0.005	250
Lead	22	0.0039	1
Magnesium	4.6	0.004	1
Mercury	95.783	0.00089	1
Molybdenum, drawn	5.7	0.004	
Monel metal	42	0.0020	
Nichrome	100	0.0004	
Nickel	7.8	0.006	100
Palladium	11	0.0033	
Phosphor bronze	7.8	0.0018	1
Platinum	10	0.003	1
Silver	1.59	0.0038	1
Tantalum	15.5	0.0031	
Tin	11.5	0.0042	1
Tungsten, drawn	5.6	0.0045	1
Zinc	5.8	0.0037	1

Table X-9. Brazing alloy resistivities.

Trade name	Composition	Liquidus (°C)	Solidus (°C)	Resistivity (μΩ-cm)
50-50 (W)	50% Au, 50% Cu	970	955	9.7
Nioro (W) Permabraze 130 (H)	82% Au, 18% Ni	950	950	31.2 (29.3)
Nicoro 80 (W)	81.5% Au, 16.5% Cu, 2% Ni	925	910	17.2
80-20 (W)	80% Au, 20% Cu	910	908	12.3
Palcusil 15 (W)	65% Ag, 20% Cu, 15% Pd	900	850	6.0
Palcusil 10 (W)	58% Ag, 32% Cu, 10% Pd	852	824	4.2
Silcoro 60 (W)	60% Au, 20% Cu, 20% Ag	845	835	13.0
Palcusil 5 (W)	68% Ag, 27% Cu, 5% Pd	810	807	3.8
Nicusil 3 (W)	71% Ag, 28% Cu, 0.75% Ni	795	780	2.6
Cusil (W) Braze 720 (H)	72% Ag, 28% Cu	780	780	2.3 (2.2)
Lithobraze 720 BT (W)	71.8% Ag, 28% Cu, 0.2% Li	760	760	4.2 (3.39)
Incusil 10 (W)	63% Ag, 27% Cu, 10% In	730	685	7.1
Incusil 15 (W) Permabraze 615 (H)	61.5% Ag, 24% Cu, 14.5% In	705	630	9.6 (10.7)

F. Power calculations in SFO

One of the main uses of SFO is to compute the quality factor or Q of resonant cavities plus a number of other accelerator figures of merit. To do this calculation, SFO needs the total power dissipated in the walls of the cavity. The code can also include the power losses in dielectric and magnetic materials for CFish problems. SFO uses the Slater perturbation technique to compute the power dissipated on drift-tube stems and post couplers. The Slater-perturbation calculations also provide frequency-shift data for small displacements of boundary surfaces and for stems and post couplers immersed in the fields computed for drift-tube linac structures. SFO also estimates the additional power losses near coupling slots in accelerator cavities from empirically determined data.

1. Power calculations on boundary segments

For Superfish problems, two variables control the way SFO calculates the power and fields along boundary segments. Variable NHSTEM is the number of half stems for boundary segment numbers with a negative value, and RSTEM is the stem radius in the same units used for boundary-point data in the Automesh input file. If you do not include the FieldSegments list for a Poisson problems, then SFO calculates the fields along all segments. If you do not include the FieldSegments list for a Superfish problem, the code tries to identify the metal boundaries for a typical cavity geometry. You should always check whether the included segments make sense for your problem if you allow the code to find the segments.

a. Segment numbering

Segment numbering can be confusing. Program Automesh initially numbers segments in the order of the PO namelist end points. Automesh adds more segments if you include

"line regions" to change the mesh size. One new segment appears for each intersection of a line region with a cavity boundary line. Automesh stores the end points and segment numbers for all the segments in binary solution file for use by SFO and the tuning programs.

One method for finding the segments on which to calculate power is to start with all the segments and then omit those that do not correspond to a metal surface or to a drift-tube stem. For example, look in file OUTAUT.TXT for the last section titled "Logical segment boundary end points" and note the number of the last segment. Include all the segments in the FieldSegments list in the SFO input file. (SFO will automatically exclude Dirichlet boundaries from the power calculations unless IncludeDirichletSegs = Yes in either the [Global] section or the [SFO] section of SF.INI.) Run SFO and then inspect the "Wall segments" table near the end of the SFO output file. Look at the physical coordinates of the boundary segments to decide which segments to exclude from the FieldSegments list.

b. Power and rf fields on Dirichlet boundaries

For Superfish problems, SFO omits Dirichlet boundaries from power and field calculations. There are several reasons for this default setting. Usually the cavity designer is interested in the power dissipated on the metal surfaces and the peak magnetic and electric fields on these surfaces. Segments that satisfy a Dirichlet boundary condition cannot contribute to the power losses since $H_\phi = 0$ on the boundary. Metal surfaces must satisfy the Neumann boundary condition, so the peak surface electric field will not occur on a Dirichlet boundary.

Occasionally, the need arises for listing the fields along the Dirichlet boundary segments in a Superfish problem. For example, one might use the complementary fields to find the solution for a TE cavity mode in cylindrical coordinates. In this case, the roles of E and H are interchanged. The listed values for E in the SFO output file actually correspond to magnetic field H in the TM mode. You can prevent SFO from automatically skipping the Dirichlet boundary segments by setting IncludeDirichletSegs = Yes in either the [Global] section or the [SFO] section of SF.INI.

c. Complementary solutions

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. Program SFO was written primarily for TM modes in cylindrically symmetric cavities (ICYLIN = 1). The only nonzero components of the field are of H_ϕ, E_r, and E_z. For problems in Cartesian coordinates (ICYLIN = 0), the nonzero components are H_z, E_x , and E_y. These solutions for a waveguide cross section are transverse electric or TE modes. Because the form of Maxwell's equations is identical (except for an algebraic sign) in the respective cases, 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. However, in this case program SFO will not calculate correctly the power along metal surfaces. A section in the Automesh chapter describes a method for computing Q for the complementary solution.

2. Power estimates on stems

SFO uses the Slater perturbation technique to compute the power dissipated on drift-tube stems and post couplers and also the frequency shift caused by immersing such objects in the fields computed for drift-tube linac structures. The Slater perturbation calculation for stems takes into account field enhancements caused by distortion of the cavity field. Though SFO uses both electric and magnetic fields in the Slater-perturbation calculation, drift-tube stems typically reside in predominantly magnetic field regions. The field varies sinusoidally with angle around the circular stem. At two points on the stem (fore and aft longitudinally) the magnitude of H is twice the unperturbed field H_ϕ computed by SFO in the absence of the stem. At the two points 90 degrees away, H drops to zero. Thus the integral of the power per unit length on the stem is twice the value one would obtain on a thin flat conducting strip with the same surface area as the stem, where one assumes that the conducting strip is aligned so it does not disturb the magnetic field. Ginzten (see References) refers to a "shape factor" in discussions of the Slater perturbation calculation. For a cylinder, the shape factor is 2 for both electric and magnetic field effects. The surface resistance used for these calculations depends upon the settings discussed in Section E "Options for calculating surface resistance."

a. Negative segment numbers indicate a stem

A negative value for a segment number indicates that SFO should calculate the power dissipation and frequency shifts for NHSTEM half drift-tube stems instead of for the cavity wall. Automesh sets the default value of NHSTEM to one. This calculation assumes that the negative-numbered segments lie along a plane of symmetry that contains one or more drift-tube stems, but no conducting wall.

b. Stems in multiple-cell problems

For multiple-cell problems, use a StemData table to supply the number and locations of the additional drift-tube stems or post couplers. Stem information entered in the StemData table is independent of the stem-power calculation done for negative segment numbers.

c. Stem proximity effect

Besides the field enhancement discussed above for cylindrical stems, there can be an additional increase in frequency shift and power dissipation from the stem proximity effect discussed by R. K. Cooper and J. H. Billen (Los Alamos National Laboratory internal report AT-6:ATN-89-5, Feb., 1989). SFO does not include this effect in any of its calculations. We present here a short summary of the results for drift-tube linac designers.

The treatment considers a periodic array of cylinders of diameter 2c with centers separated a distance 2b immersed in an otherwise uniform magnetic field of intensity H_0. For an isolated stem, the distribution of field on the stem surface is $2H_0\sin\alpha$. For the period array of stems the distribution may be written $[(1+\lambda)/\lambda]H_0\sin\alpha$, where λ is the enhancement factor listed in Table X-10.

Table X-10. Field enhancement factor λ versus c/b.

c/b	λ	(1+λ)/λ
0.00	1.000	2.000
0.05	0.996	2.004
0.10	0.984	2.017
0.15	0.964	2.038
0.20	0.936	2.068
0.25	0.902	2.109
0.30	0.862	2.160
0.35	0.816	2.225
0.40	0.766	2.306
0.45	0.711	2.406
0.50	0.653	2.531
0.55	0.592	2.689
0.60	0.528	2.894
0.65	0.461	3.168
0.70	0.392	3.551
0.75	0.320	4.123

For example, if the stem centers are separated by 4 stem radii, the magnetic field on the stem surface would be $(2.109)H_0\sin\alpha$. That is, H is enhanced by a factor 2.109 rather than the factor 2.000 for an isolated stem. Because the stored energy and power are proportional to the square of the field, the frequency shift and power dissipation in this case would be about 10% higher than for an isolated stem.

3. Total surface area used for power calculations

Variable SAREA provides additional information about your problem geometry. SFO reports a value for SAREA, the total surface area of the cavity on which it calculates power. Note that this number will not correspond to the total cavity surface area unless all the appropriate segments are included in the FieldSegments list.

4. Coupling-slot effects

Program SFO includes two calculations related to the presence of a coupling slot in certain types of rf cavities. One calculation computes the average magnetic field H_ϕ along segments in the corner of the cavity where the coupling slot is normally located. The other calculation is an estimate of the additional power loss in a cavity containing a coupling slot. Accelerators with coupling slots include the CCL and CCDTL operating in the $\pi/2$ structure mode.

Note that the coupling slot itself cannot be modeled in a two-dimensional code such as Superfish. However, analyses based upon the cavity stored energy and the fields near the future location of the coupling slot can yield considerable useful information.

a. Average field at the coupling slot

A paper by J. Gao ["Analytical formulas for the resonant frequency changes due to opening apertures on cavity walls," Nuclear Instruments and Methods in Physics Research A311, p. 437-443 (1992)] describes a Slater-perturbation calculation that can be

used to obtain the coupling between cavities in CCL or CCDTL. Application of the method requires the stored energies in the cavities and the unperturbed magnetic field at the location of the coupling slot. By unperturbed field we mean the field in the original cavities before the slot has been cut.

Program SFO uses variables in Table X-11 to compute an average of the field H_ϕ along the segments spanning the future location of the coupling slot. Figure X-4 shows the essential features of a portion of a coupled-cavity drift-tube linac, in this case at the transition from one type of cavity to another type. Coupling slots are formed at intersections between the accelerating cavities and the coupling cavities. In the accelerating cavities, the coupling slot is along the circular arc connecting the outer radius of the cavity to the vertical wall.

For the accelerating cavities designed by programs CCLfish and CDTfish, these codes both determine values for ICORNER1 and ICORNER2 and include them in the SEG file for SFO to trigger this calculation. The input variables in Table X-11 can also appear in the first REG namelist of the Automesh input file. The reason that CCLfish and CDTfish do not include them in the Automesh input file (or REG file) is that the segment numbers are not known when the code generates the REG file because of the influence of the X and Y line regions. For input files created with an editor (for example, for the coupling cavities), the user may find it more convenient to include settings for ICORNER1 and ICORNER2 in the Automesh input file.

The computed value of HCORNER appears near the end of the SFO output file in the section titled “Problem variables calculated by this code.”

Table X-11. Variables for the average field near the coupling slot.

Variable	Default value	Description
HCORNER	none	Average field H_ϕ along segments ICORNER1 and ICORNER2.
ICORNER1	0	First segment number for computing the average field.
ICORNER2	0	Last segment number for computing the average field.

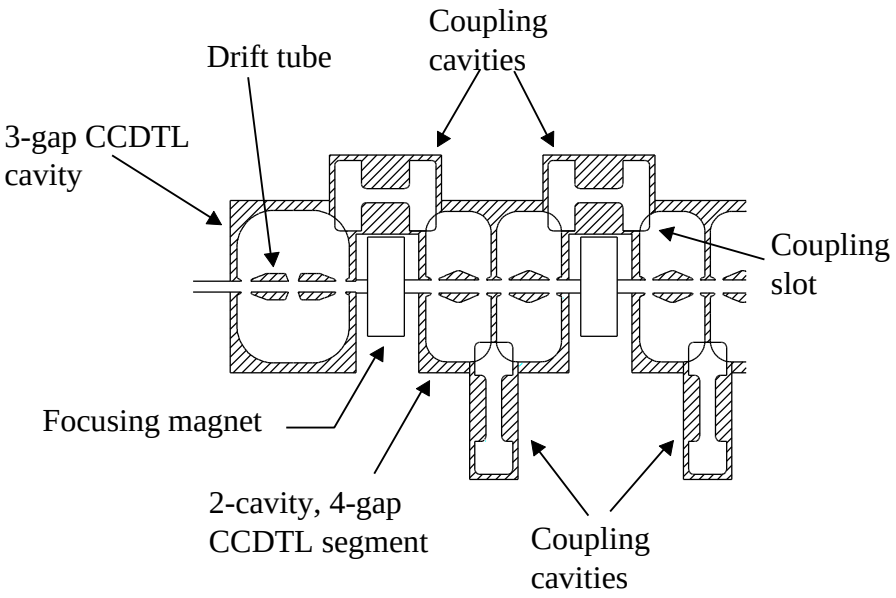

Figure X-4. Coupling slots in a CCDTL.
This figure of revolution shows how coupling slots are formed at the intersections between accelerating cavities and side-mounted coupling cavities.

b. Estimating coupling-slot power losses

Program SFO uses variables in Table X-12 to estimate the additional power dissipation near the coupling slot that connects CCL or CCDTL cavities to an adjacent cavity. The calculation is based upon the empirical observation that the Q of CCL cavities decreases by about 3% for each percent of coupling for a full cavity with two slots connecting it to the adjacent cavities. Side-coupled linacs such as the 805-MHz LAMPF accelerator, other 805-MHz linacs at the Fermi National Accelerator Laboratory and at CERN, and many electron linacs manufactured commercially typically have cell-to-cell coupling of 5% to 7%. The extra power dissipation comes from the field distortion in the neighborhood of the coupling slot.

Variable SLOSS has a default value of 0.03, which reflects the empirical result of 3% loss for each percent of coupling. You can enter your own measured value for SLOSS, if available. SFO calculates the extra power dissipated near the coupling slot and tabulates the results for a range of coupling values from CCLMINK to CCLMAXK in steps of CCLDELK.

Table X-12. Variables affecting coupling slot power estimates.

Variable	Default value	Description
CCLMINK	1.0	Lowest coupling for coupling-slot power table.
CCLMAXK,	6.0	Highest coupling for coupling-slot power table.
CCLDELK	1.0	Increment in the coupling-slot power table in SFO.
IOBSEG	–1	Starting segment of the CCL outer boundary.
ISLOT	0	Selects the coupling-slot power option if equal to 1.
SLOSS	0.03	Coupling-slot power increase per percent coupling.

Programs CCLfish and CDTfish both set ISLOT = 1 in the Automesh input file to trigger this calculation. CDTfish also sets IOBSEG to the first segment on the outer boundary of the problem geometry. Setting IOBSEG is necessary to exclude any drift tubes or stems from the portion of the total power that increases because of the slot. Setting IOBSEG is not necessary for ordinary CCL half cells, which contain no drift tubes, so CCLfish does not include a value for IOBSEG. If IOBSEG has its default value of −1, then SFO determines the first segment of the outer boundary and reports it in the output file.

Coupling-slot effects for power of 3141.368 W on boundary segments 16 through 33 for 3.000% increase in outer wall power per percent coupling:

Coupling (%)	Slot Power (W)	Total Power (W)	Q	Z (MΩ/m)	ZT^2 (MΩ/m)
1.0000	94.241	5083.873	32255.93	70.202	39.313
2.0000	188.482	5178.114	31668.88	68.924	38.597
3.0000	282.723	5272.355	31102.81	67.692	37.907
4.0000	376.964	5366.596	30556.62	66.503	37.242
5.0000	471.205	5460.837	30029.29	65.356	36.599
6.0000	565.446	5555.078	29519.85	64.247	35.978

Figure X-5. Coupling-slot data in the SFO output file.

Program SFO finds the last segment of the outer boundary by looking for the last one of the contiguous set of segments starting with IOBSEG. Figure X-5 shows a sample of the coupling-slot data from the SFO output file. If ParmilaData = Yes in file SF.INI, then SFO includes in the PMI file a less extensive table than the one shown in Figure X-5. The PMI-file table contains only the shunt impedance Z versus coupling. Program Parmila uses this data to interpolate the rf cavity power for the design value of the coupling between cavities.

5. Power losses in dielectric and magnetic materials for CFish problems

Variable ICCP indicates whether SFO should calculate power losses in dielectric and magnetic materials for complex-field problems solved by CFish. A value of ICCP > 0 tells the code to calculate these power losses. Because this calculation may take a while for a large number of mesh points, you may wish to turn it off until you are satisfied with the design of the cavity. The value of ICCP in Table IX-7 indicates which points on a mesh triangle to use in calculating the average fields for the triangle. If the problem has complex fields and complex permittivity and permeability, then SFO tabulates in the output file the power dissipated by electric and magnetic fields in each material for the entire cavity. CFish supplies a value of 1/2Q (half the loss tangent) for the entire cavity in variable Q2I.

Other postprocessors also compute power losses in materials. Program SF7 makes a table in file OUTSF7.TXT similar to the one produced by SFO, but only for the materials within a rectangular region. Program WSFplot displays the total power loss for all materials within the display boundaries.

G. Field-interpolation algorithm in SFO

Program SFO uses the Poisson Superfish field interpolator for all types of problems: rf, magnetostatic, and electrostatic. The field interpolator fits one of several polynomial

functions of X and Y (or R and Z) to the first and second nearest neighbors of the closest mesh point. Some of the functions satisfy either Neumann or Dirichlet boundary conditions on or near vertical or horizontal boundaries.

H. Transit-time-factor integrals

Beam-dynamics codes such as Parmila use the transit-time factor and some related integrals to compute the effects of the accelerating gap on a beam of charged particles. The latest version of Parmila (developed from 1995 to 1997 by H. Takeda) uses transit-time-factor data computed by SFO. These data appear in the SFO output file (and in the PMI file if file SF.INI includes the setting ParmilaData = Yes). For compatibility with older versions of Parmila, the output file also includes the SFDATA line for Parmila.

The treatment presented here is based upon the article "Numerical methods. Acceleration by a gap." by A. Carne, B. Schnizer, P. Lapostolle, and M. Prome, pages 747 to 783 in Linear Accelerators, edited by Pierre M. Lapostolle and Albert L Septier, North-Holland Publishing Company (1970). Note that there are some differences between the transit-time parameters that appear in the Carne et al. article and those reported by SFO. As discussed below, the derivatives T′, S′, etc. reported by SFO include factors of $\beta\lambda$ in the denominator to make dimensionless quantities. Also, the algebraic signs are not the same as in Carne et al. in some cases. These differences were introduced for convenience in coding the equations in Parmila. The parameters reported by SFO are consistent with their use in Parmila.

The synchronous particle gains energy δW through a cell:

$$\delta W = qV_0\mathbf{T}\cos\phi_s,$$

where q is the charge, V_0 is the rf axial voltage, **T** is the transit-time factor, and ϕ_s is the synchronous phase. The rf axial voltage is given by:

$$V_0 = \int_{Z_s}^{Z_e} E_z dz,$$

where Z_s and Z_e are the starting and ending z coordinates for the cell. For convenience, we define the average axial electric field amplitude as $E_0 = V_0/L$, where L is the cell length ($= Z_e - Z_s$). The transit-time factor is

$$\mathbf{T} = \frac{1}{V_0}\left[\int_{Z_s}^{Z_e} E_z(z)\cos\omega t(z)dz - \tan\phi_s\int_{Z_s}^{Z_e} E_z(z)\sin\omega t(z)dz\right].$$

We make the usual approximation that the change in particle velocity is small compared to the initial velocity: $\omega t \approx 2\pi z/\beta\lambda = kz$, k is the wave number and $\beta\lambda$ is the distance a particle of velocity $v = \beta c$ travels in one rf period. In the discussion to follow, we also replace z with the quantity $z - Z_c$, where Z_c is a reference location for computing the integrals. The equation above then can be written:

$$\mathbf{T} = \frac{1}{V_0}\left[\int_{Z_s}^{Z_e} E_z(z)\cos k(z - Z_c)\,dz - \tan\phi_s \int_{Z_s}^{Z_e} E_z(z)\sin k(z - Z_c)\,dz\right].$$

Introducing Z_c facilitates identifying the center of the gap in the Automesh geometry, which may not have z = 0 in the center of the gap. If the center of the rf gap is at $Z_c = 0$, then the synchronous particle arrives at z = 0 at time t = 0, and the phase of the rf field relative to the peak is ϕ_s. In practical applications (specifically, in program Parmila), one chooses the reference position (or cell center) Z_c to coincide with the cell's electrical center Z_{ec} (see below) so the second term vanishes and the transit-time factor is independent of the synchronous phase.

For program SFO to report the transit-time-factor integrals we introduce variables T and S, and write $\mathbf{T} = T - (\tan\phi_s)S$. If you use the multiple cell option, then SFO reports these integrals for each cell. The limits of integration, Z_s and Z_e, are the starting and ending positions of the cell or cavity. A summary near the end of the SFO output file tabulates the integrals T, T′, T″, S, S′, and S″ which are defined below. If Z_c corresponds to the gap center, the odd integrals should be zero unless there are significant field asymmetries. Program SFO reports both halves of the integrals on either side of the cell center Z_c. (These equations are more general than the ones found in section C.1.2 of the 1987 Reference Manual. The equations in the Reference Manual applied only to cells that were 1βλ in length with their electrical center at z = 0.) The even integrals are $T = T_L + T_R$, $T' = T'_L + T'_R$, and $T'' = T''_L + T''_R$, and the odd integrals are $S = S_L + S_R$, $S' = S'_L + S'_R$, and $S'' = S''_L + S''_R$, where the subscripts L and R refer to the left and right halves of the cell. Following are the definitions:

$$T_L = \frac{1}{V_0}\int_{Z_s}^{Z_c} E_z(z)\cos[k(z - Z_c)]dz,$$

$$T_R = \frac{1}{V_0}\int_{Z_c}^{Z_e} E_z(z)\cos[k(z - Z_c)]dz,$$

$$S_L = \frac{1}{V_0}\int_{Z_s}^{Z_c} E_z(z)\sin[k(z - Z_c)]dz,$$

$$S_R = \frac{1}{V_0}\int_{Z_c}^{Z_e} E_z(z)\sin[k(z - Z_c)]dz,$$

$$T'_L = \frac{1}{V_0\beta\lambda}\int_{Z_s}^{Z_c} (z - Z_c)E_z(z)\sin[k(z - Z_c)]dz,$$

$$T'_R = \frac{1}{V_0\beta\lambda}\int_{Z_c}^{Z_e} (z - Z_c)E_z(z)\sin[k(z - Z_c)]dz,$$

$$S'_L = \frac{1}{V_0 \beta\lambda} \int_{Z_s}^{Z_c} (z - Z_c) E_z(z) \cos[k(z - Z_c)] dz ,$$

$$S'_R = \frac{1}{V_0 \beta\lambda} \int_{Z_c}^{Z_e} (z - Z_c) E_z(z) \cos[k(z - Z_c)] dz ,$$

$$T''_L = \frac{1}{V_0 (\beta\lambda)^2} \int_{Z_s}^{Z_c} (z - Z_c)^2 E_z(z) \cos[k(z - Z_c)] dz ,$$

$$T''_R = \frac{1}{V_0 (\beta\lambda)^2} \int_{Z_c}^{Z_e} (z - Z_c)^2 E_z(z) \cos[k(z - Z_c)] dz .$$

$$S''_L = \frac{1}{V_0 (\beta\lambda)^2} \int_{Z_s}^{Z_c} (z - Z_c)^2 E_z(z) \sin[k(z - Z_c)] dz ,$$

and

$$S''_R = \frac{1}{V_0 (\beta\lambda)^2} \int_{Z_c}^{Z_e} (z - Z_c)^2 E_z(z) \sin[k(z - Z_c)] dz .$$

The factors of $\beta\lambda$ that appear in the denominators of the first and second derivatives reflects a particular choice or convention for making the reported quantities dimensionless. Also, by convention, we do not include algebraic signs in the definitions. Beam-dynamics codes that use SFO transit-time-factor integrals must take these conventions into account. The coding in program Parmila is consistent with the definitions presented here. Other laboratories may use different conventions. For example, another common convention used to make the transit-time-factor derivatives dimensionless is to multiply them by appropriate powers of $k = 2\pi/\beta\lambda$. For the definitions presented above the following relationships hold:

$$T' = -\frac{\partial T}{\partial k} \frac{1}{\beta\lambda} , \qquad S' = +\frac{\partial S}{\partial k} \frac{1}{\beta\lambda} ,$$

$$T'' = -\frac{\partial^2 T}{\partial k^2} \frac{1}{(\beta\lambda)^2} , \qquad S'' = -\frac{\partial^2 S}{\partial k^2} \frac{1}{(\beta\lambda)^2} .$$

The center of the gap occurs at the longitudinal position Z_c, which is also the electrical center of a symmetric cell. In the Automesh and SFO input files use variables ZCTR or ZCENTER specify Z_c. For full-cavity calculations, Z_c is ZCTR. For multiple-cell calculations, Z_c for each cell is the value of ZCENTER. The equation $S_L + S_R = 0$ defines the electrical center Z_{ec}:

$$S_L(Z_c = Z_{ec}) = -S_R(Z_c = Z_{ec}).$$

For symmetric full cells, Z_{ec} is midway between Z_s and Z_e. In this case, Z_c and Z_{ec} would be equal. The limits of integration may include either a full cell or (for symmetric

problems) a half cell. For full cells, SFO calculates both separate integrals for the left and right halves of T, T′, T″, S, S′, and S″. For half cells, the code assumes that $T_L = T_R$, $T'_L = T'_R$, $T''_L = T''_R$, $S_L = -S_R$, $S'_L = -S'_R$, and $S''_L = -S''_R$. If variable ITFILE = 1, then SFO creates a plot file of the transit-time-factor integrands versus z position. Run program Tablplot to view the plot.

1. Variables used with whole-cavity transit-time integrals

For all problems, SFO uses the variables in Table X-13 to compute the transit-time-factor integrals for the full geometry. (If NCELL is 1 or greater, then SFO also uses variables for multiple-cell problems listed in Table X-15.) The default values in the table correspond to half of a symmetric 1βλ drift-tube linac cell. Some variables in Table X-13 are user-supplied and some are calculated by SFO. The variable CAPK is the wave number k used in the transit-time integrals:

$$k = \frac{2\pi}{\beta\lambda} = \frac{k_0}{\beta},$$

where β is the value of variable BETA, and the wavelength $\lambda = c/f_D = 2\pi/k_0$. The frequency f_D is variable FREQD, which is assumed to be the design frequency f_D of the rf linac. Automesh input files that originate with one of the tuning programs will have a value for FREQD defined in the first REG namelist. If the problem has no entry for f_D, then SFO sets f_D equal to the resonant frequency f computed by Superfish and found in variable FREQ.

Variable KMETHOD (default value = 0) determines the method used to calculate the wave number k. If KMETHOD ≠ 1, then SFO computes k and β in the following order:

$$k = \frac{\Delta\phi}{\Delta z}, \text{ then } \beta = \frac{k_0}{k},$$

where Δϕ is the change in phase (in radians) for a particle of velocity β over the distance Δz. The phase change Δϕ in degrees appears in variable DPHI. The default value of 180 degrees for DPHI corresponds to half of a symmetric drift-tube linac cell of length βλ. If KMETHOD = 1, then SFO computes k and Δϕ in the following order:

$$k = \frac{k_0}{\beta}, \text{ then } \Delta\phi = k\Delta z,$$

where β is the value of BETA. For both methods, the default value for the distance Δz is ZLONG, which is a computed value. Variable CLENGTH is a user-supplied value for Δz that takes precedence over ZLONG. If CLENGTH is positive, SFO uses it for Δz in the above equations. Usually, CLENGTH is less than ZLONG, for example when fields penetrate into a bore tube on one end of the cavity. Integrals are still performed over the length ZLONG in order to include all of the fields. SFO also uses variable ZLONG (or CLENGTH, if supplied) to define E_0 for field normalization.

Table X-13. Variables used with whole-cavity transit-time integrals.

Variable	Default value	Description
BETA	see text	Particle velocity used if KMETHOD = 1.
CAPK	$2\pi/\beta\lambda$	Wave number k in transit-time-factor integrals.
DPHI	180	Phase length (degrees) used if KMETHOD ≠ 1.
CLENGTH	0	User-supplied cavity length for computing E_0.
FREQ	(computed)	Resonant frequency f computed by Fish (MHz).
FREQD	FREQ	Design frequency f_D (MHz).
KMETHOD	0	if 1, compute CAPK from BETA, else compute CAPK from DPHI,
LINT	1	Logical coordinate for E_z integration.
ZCTR	0	Gap center used in transit-time integrals.
ZLONG	(computed)	Problem-geometry length used for E_0 integration.

2. SFDATA line for older versions of Parmila

The PMI file written by SFO is a short text file containing information used in the ion-linac design code Parmila (the 1996 version written by H. Takeda). The traditional version of Parmila that has been in use since the 1960s cannot use all the information in the PMI file. The old Parmila uses data from a line in the SFO output file known as the SFDATA line. In older versions of SFO the SFDATA line appeared after the summary of cavity parameters near the end of the file.

For compatibility with older versions of Parmila, this information is still included in the transit-time-factor summary. The SFDATA line contains the data in Table X-14. It does not include the second derivatives T″ and S″. A word of caution is in order. For the SFDATA line, the code assumes that the field distribution for half cell geometries is symmetric about the left edge of the geometry. The results reported on this line will not be correct for an antisymmetric field distribution (as, for example, half of a structure containing an even number of cavities operating in the π mode).

Table X-14. SFDATA parameters for old versions of Parmila.

Position	Parameter	Description
1	β	Particle velocity relative to light.
2	T	T integral, sum of T_L and T_R.
3	T′	T′ integral, sum of T'_L and T'_R.
4	S	S integral, sum of S_L and S_R magnitudes.
5	S′	S′ integral, sum of S'_L and S'_R magnitudes.
6	$g/\beta\lambda$	Gap length relative to βλ.
7	Z	Shunt impedance in MΩ/m.

I. Multicell calculations in SFO

Program SFO supports several calculations for multicell accelerating structures such as the drift-tube linac (DTL). The code will include in the total power the power dissipated on stems and post couplers. It also calculates the frequency effects of the stems and post couplers using the Slater perturbation technique. SFO calculates the transit-time-factor integrals for each cell. As an option, the code can tabulate transit-time-factor data as a

function of particle velocity. These tables are in the form of Tablplot input files, one file for each cell.

SFO can perform these calculations only for Superfish problems (KPROB = 1) with cylindrical symmetry (ICYLIN = 1). The next few sections describe how to use the multicell features.

1. Multicell transit-time-factor integrals

For multicell problems, SFO uses the variables in Table X-15 and the CellData entries listed in Table X-16 to compute the transit-time-factor integrals for each cell. The code also computes the full-geometry transit-time factor as discussed in section H.1 above.

Table X-15. Variables used with multiple-cell transit-time integrals.

Variable	Default value	Description
BETA	see text	Particle velocity relative to light.
FREQ	none	Resonant frequency f computed by Fish (MHz).
FREQD	FREQ	Design frequency f_D (MHz).
KMETHOD	0	If 1, compute wave number from BETA, else use phase lengths.
LINT	1	Logical coordinate for E_z integration.
PLCELL	360	Phase length (degrees) per cell in degrees.
ZCTR	none	Gap center supplied in the CellData table.

Table X-16. CellData columns for multicell transit-time integrals.

Column	Description
ZSTART	Longitudinal position of the start of the cell.
ZCENTER	Center of the cell.
ZSTOP	End of the cell.
CPHASE	Phase length of the cell [optional, default = PLCELL].

For multicell problems, Z_c is the value of ZCENTER supplied in the CellData table. ZCENTER is usually the physical center of the gap. SFO calculates the actual electrical center Z_{ec} and reports the shift from Z_c caused by any field asymmetries. The code reports this shift for your information. The code does not use Z_{ec} in any of the transit-time-factor integrals reported in the SFO output file. (You can set IBETA = 2 to make SFO use Z_{ec} for Z_c in the Tablplot files of transit-time factors versus β.) If your application requires it, you can rerun SFO using revised values of ZCENTER in CellData table. The tuning program CDTfish can use the reported values of Z_{ec} to adjust the gap geometry on subsequent runs so that Z_{ec} actually coincides with Z_c.

Variable KMETHOD determines the method used to calculate the wave number k_I used in the transit-time integrals for the cell I:

$$k_I = \frac{2\pi}{\beta_I \lambda} = \frac{k_0}{\beta_I},$$

where β_I is the particle velocity in cell I, and the wavelength $\lambda = c/f_D = 2\pi/k_0$. The frequency f_D is variable FREQD, which is assumed to be the design frequency f_D of the rf

linac. Automesh input files that originate with one of the tuning programs will have a value for FREQD defined in the first REG namelist. If the problem has no entry for f_D, then SFO sets f_D equal to the resonant frequency f computed by Superfish and found in variable FREQ.

If KMETHOD = 0 (the default), then SFO computes k for each cell as follows:

$$k_I = \frac{\Delta\phi_I}{\Delta z_I}, \text{ then } \beta_I = \frac{k_0}{k_I}$$

where $\Delta\phi_I$ = CPHASE(I) × π/180 is the change in phase (in radians) for a particle of velocity β_I over the distance Δz_I = ZSTOP(I) − ZSTART(I) (see Table X-16). The phase change in degrees appears in variable CPHASE(I) for each cell. If you omit CPHASE from the CellData table, then SFO uses variable PLCELL for the phase length of every cell. The default value for PLCELL is 360 degrees. The value of β_I computed for each cell appears in the transit-time-factor summary near the end of the SFO output file. If KMETHOD = 1, then SFO uses a fixed particle velocity and wave number for all the cells and computes k and $\Delta\phi_I$ in the following order:

$$k = \frac{k_0}{\beta}, \text{ then } \Delta\phi_I = k\Delta z_I,$$

where β is the value of BETA. In the tuning programs CDTfish and MDTfish, KMETHOD has the default value of zero. Therefore, SFO always computes the value of β_I for each cell.

Using KMETHOD = 1 can help characterize the performance of short independently-phased sections of a linac when such cavities accelerate particles whose velocity differs from the design velocity (or geometrical β). For 0 < BETA ≤ 1, the cell phase length is a calculated quantity and the code ignores values of CPHASE supplied in the CellData table.

2. Transit-time-factor data versus β

A cavity designed for a particular particle velocity (sometimes called the geometrical β) might be used to accelerate particles having a range of β. You can use variables in Table X-17 to compute transit-time-factor data as a function of β for such a cavity. This option applies to single-cavity problems and to multicell cavities. In this context, a single cavity may be a multicell cavity, but one for which you have not entered data for the individual cells (i.e., no CellData table in the SFO input file). The type of output depends upon the setting of IBETA according to Table X-18. As mentioned in the table, the value of IBETA affects the value of Z_c used in the transit-time-factor integrals. If IBETA = 1 or 3 , then Z_c corresponds to ZCTR for single-cavity problems, or to ZCENTER for each cell, which is supplied in the CellData table for multicell problems. If IBETA = 2 or 4, then $Z_c = Z_{ec}$, where Z_{ec} is the electrical center of the cavity or cell computed by SFO.

Table X-17. Variables for plotting transit-time data versus β.

Variable	Default value	Description
IBETA	0	If > 1, make transit-time table from BETA1 to BETA2.
BETA1	0.10	Starting BETA for table of transit-time data.
BETA2	0.95	Ending BETA for table of transit-time data.
DBETA	0.05	BETA increment for table of transit-time data.

Table X-18. IBETA settings for SFO.

IBETA	Description
0	No T versus beta calculations performed.
1	Generates Tablplot file of T, S, etc. versus beta with Z_C at the geometric center of the cavity or individual cells. Option applies to SFO, Autofish, and ELLfish.
2	Generates Tablplot file of T, S, etc. versus beta with Z_C at the electrical center of the cavity or individual cells. Option applies to SFO, Autofish, and ELLfish.
3	Same as 1, plus generate a family of PMI files for use in Parmila. Option applies only to the stand-alone SFO program.
4	Same as 2, plus generate a family of PMI files for use in Parmila. Option applies only to the stand-alone SFO program.

a. Tablplot files of T versus β

For IBETA = 1 though 4, programs SFO and Autofish create filenames of the form TBETAxx.TBL, where xx stands for two integers that identify the cell number. For example, if the CellData table has 3 lines, then SFO creates three files: TBeta01.TBL, TBeta02.TBL, and TBeta03.TBL. If the multicell option is not in use then SFO and Autofish make only one file named TBETA.TBL Use program Tablplot to plot the data in the TBETAxx.TBL files. Since these files get overwritten each time you run SFO with the IBETA > 0, you may wish to rename the files for long-term storage.

The control file for program ELLfish includes several keywords that you can use to specify variables IBETA, BETA1, BETA2, and DBETA. The code will create a Tablplot file for each cavity in the control file. ELLfish names the file after the Automesh input file, but with extension TBL. There is only one file for each ELLfish problem because the code runs on only a half cell or a full cell (i.e., there is no multicell option in ELLfish).

Although ELLfish does not deal with multicell cavities, you can use the ELLCAV program to generate an Autofish input file for a multicell cavity. When it creates the Autofish input file, this utility code also generates an SFO input file containing a CellData table for all the cells in the problem. The input file has some helpful comments on how to exercise the numerous transit-time-factor options in SFO. ELLCAV sets the value of IBETA to either 3 or 4, anticipating that the user will want to generate PMI files for Parmila as described in the next section. If IBETA is originally 0 or 1, ELLCAV sets IBETA = 3, and if IBETA is originally 2, ELLCAV sets IBETA = 4.

b. PMI files for Parmila for each β

In SFO, if IBETA = 3 or 4, the program creates a family of PMI files for supplying transit-time-factor data to program Parmila. Each PMI file corresponds to a particular

particle velocity over the range from BETA1 to BETA2 (see Table X-17). This family of files is similar to the series of files created by the tuning programs CCLfish, DTLfish, or CDTfish, where each file corresponds to a separate cavity designed for a particular velocity. SFO also creates a control file for program ReadPMI (distributed with Parmila) for reading the PMI files and creating the required data tables for Parmila. The ReadPMI program automatically opens file with extension RDP. SFO creates file Input.RDP.

We recommend that you use the ELLCAV program to generate both an Autofish input file and a corresponding SEG file for SFO. The name of each Autofish input file consists of two characters that indicate the number of cells in the full cavity (e.g., 6C, 7C, etc.) followed by the first 6 characters of the filenames specified in the ELLfish control file. Be sure to use short enough filename prefixes in ELLfish to avoid generating the same name for different problem files. When SFO runs, it will name the PMI files using the first 4 characters of the Autofish filename followed by the characters "_Bnn" where nn is a sequence number. (Note that the sequence numbers will correspond to 100 times the β value if you set DBETA = 0.01 (see Table X-17); otherwise the sequence numbers start with 01.) Do not request a set of more than 99 files on a single run. Usually increments of the order of 0.01 in β are sufficient. If you do require more detail, run SFO more than once over different ranges and rename files as necessary to avoid duplication.

You can edit the SFO input file in various ways to produce different results. We will discuss two useful applications using the cavity in Figure X-6 as an example. Figure X-7 is the input file created by ELLCAV, which instructs SFO to create PMI files containing data for three cells. This form of the output is appropriate for input to the code Parmila, which transports the beam through each cell, one at a time. For other studies, you may need the form of T versus β for the entire cavity taken as a unit. The file is easily modified to generate this full-cavity data. As indicated by comments in the file itself, insert a semicolon comment indicator on all lines of CellData table. Lastly, set IBETA = 1 so that SFO uses Z_C = ZCTR and does not generate a set of unneeded PMI files.

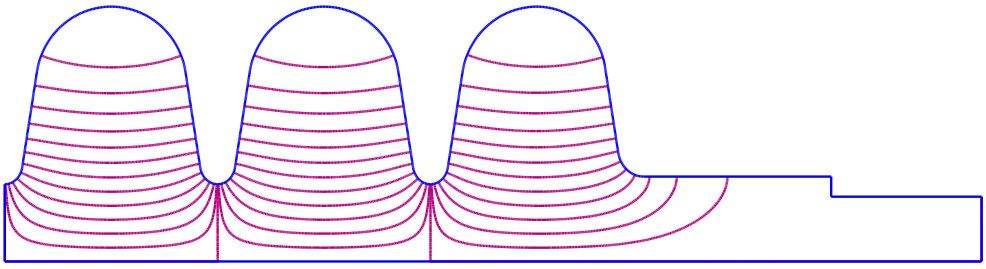

Figure X-6. Half of a six-cell cavity.
This cavity has cell lengths designed for β = 0.76, but it will be used for acceleration over a range somewhat narrower than the limits β = 0.6 to 0.99 used in the SFO input file.

```
CavityLength = 42.45508                 ; Normalization length
; On multicell problems, to get the full-cavity T and S integrals,
; insert a comment indicator (;) on all lines of the CellData table.
MulticellPhaseLength = 180.0000         ; PLCELL, default cell phase length
ParticleVelocity= 0.7600000             ; BETA, design particle beta
; Settings for IBETA
; 0 no tables of T vs beta
; 1 create T vs beta table using geometric center
; 2 create T vs beta table using electrical center
; 3 same as 1, and also makes family of PMI files
; 4 same as 2, and also makes family of PMI files
; ELLCAV default is either 3 or 4
IBETA = 4
BETA1 = 0.6000000                       ; Starting velocity for table
BETA2 = 0.9900000                       ; Ending velocity for table
DBETA = 0.1000000E-01                   ; Velocity increment for table
CellData                                ; Columns are Zstart,Zcenter,Zstop,Phase
0.0000000       7.0758468       14.1516937      180.0000000
14.1516937      21.2275405      28.3033873      180.0000000
28.3033873      35.3792342      42.4550810      180.0000000
EndData
FieldSegments
6   7   8   9   10  11  12  13  14  15  16  17  18  19  20
21  22  23  24  25  26  27  28
EndData
End
```

Figure X-7. SFO input file created by ELLCAV

This file shows the initial settings that instruct SFO to create PMI files containing data for three cells. The file is easily modified to generate transit-time-factor for the full cavity.

A word of caution is in order for Parmila users. Although SFO will generate PMI files with Z_C at the geometrical centers of the cells (IBETA =3), this data is not of use in Parmila, which expects to receive data with Z_C at the electrical center (IBETA = 4). The PMI files will contain the distance between geometric and electrical centers so that Parmila can compute the phase at the point where the beam particles receive their impulse.

3. Stems and post couplers in a multicell tank

Program SFO calculates frequency shifts for the stems and post couplers in a multicell cavity using the Slater perturbation formula. The code also computes the power dissipated on each stem or post coupler and includes this power in the cavity Q and other accelerator figures of merit. Stem information entered in a StemData table is independent of the stem-power calculation done for negative segment numbers. The output summary near the end of the SFO output file tabulates the computed power dissipation and frequency shifts for the stems and post couplers.

Lines in the StemData table contain the entries listed in Table X-19. If LENGTH = 0, then SFO assumes a stem or half stem. For a stem, the code computes the power dissipated and the frequency shifts for a cylindrical body extending all the way from the cavity radius to the drift tube. For drift tubes mounted on two or more stems, include duplicate lines with the same longitudinal position. Some DTL designs have dummy half

stems and half post couplers on the end walls of the cavity. Use a negative value for the radius to indicate a half stem or half post coupler. For half stems and posts, SFO calculates the net power added to the total power. (The half stem or half post coupler covers up a portion of the end wall.)

Table X-19. Stem and post-coupler parameters.

Variable	Description
POSITION	Longitudinal position of the stem or post coupler.
LENGTH	Stem or post length (can be zero for stems).
RADIUS	Stem or post radius (negative value for half stem or half post).

The power dissipated on stems and post couplers includes only the effect of the magnetic field. The calculation correctly accounts for the sinusoidal variation of the magnetic field on the stem surface. The longitudinal electric field produces an azimuthal surface current on the stem. However, the magnitude of the electric effect is so small that it can be safely ignored. Let $P_E(r)$ and $P_H(r)$ be the power from the electric-field and magnetic-field effects as a function of radial position in the DTL tank. Typically, P_E/P_H has a maximum of about 0.006 at a radius approximately one-third of the distance from the drift tube toward the tank wall. Neglecting the electric effect underestimates the total stem power by less than 0.5%.

J. The SFO output summary

For Superfish problems, a summary near the end of the SFO output file contains a number of quantities of interest to the accelerator designer. SFO does not write this summary for static problems solved by Poisson or Pandira. The summary starts with name, date, and time of the Automesh input file and a reminder that calculated values in the summary refer to the mesh geometry only. Some older versions of this Superfish postprocessor included such things as power and stored energy under the assumption that the geometry corresponded to a symmetric half cavity. This practice led to so much confusion that it has been discontinued. Next, the summary lists the method used to normalize the fields and the resonant frequency calculated by Fish or CFish. The summary continues with the charged-particle properties, some accelerator figures of merit, peak fields and power densities, and several tables listing power dissipation and frequency shifts.

1. Particle rest mass, velocity, and kinetic energy

For Superfish problems with cylindrical symmetry, SFO reports the kinetic energy of a charged particle with velocity βc. The value of β, which appears as variable BETA, is also used in the transit-time-factor integrals. The code computes kinetic energy W using the following relation:

$$W = mc^2(\gamma - 1), \text{ where } \gamma = \frac{1}{\sqrt{1-\beta^2}},$$

and mc^2 is the rest-mass energy. The default rest-mass energy is that of a proton (H^+). Use variable RMASS to enter a different rest-mass energy. A negative code number for RMASS selects a value from Table X-20. To specify the particle type in the tuning codes, use control-file keywords PARTICLE or REST_mass. These keywords in the control file affect the value of RMASS that the tuning code writes in the Automesh input file. The rest-mass energies (except H^- and D^-) listed in Table X-20 are the 2002 CODATA (Committee on Data for Science and Technology) recommended values. For more information refer to the National Institute of Standards and Technology (NIST) web pages (www.nist.gov). The negative-ion hydrogen and deuterium rest mass energies are computed from the other masses as follows:

$$\left(mc^2\right)_{H^-} = \left(mc^2\right)_{H^+} + 2\left(mc^2\right)_{e^-} - 13.5984\text{eV} - 0.699\text{eV},$$

$$\left(mc^2\right)_{D^-} = \left(mc^2\right)_{D^+} + 2\left(mc^2\right)_{e^-} - 13.5984\text{eV} - 0.699\text{eV},$$

where 13.5984 eV is the hydrogen ionization potential and 0.699 eV is the electron affinity of neutral hydrogen.

Table X-20. Code numbers and rest masses for some charged particles.

Particle	RMASS	Rest-mass energy (MeV)	Uncertainty (eV)
$e^{\pm}$	–1	0.510998918	0.044
H^+	–2	938.272029	80
H^-	–3	939.294012	~80
D^+	–4	1875.61282	160
D^-	–5	1876.63480	~160
$^3He^{2+}$	–6	2808.39142	240
$^4He^{2+}$	–7	3727.37917	320
$\mu^{\pm}$	–8	105.6583692	9.4

2. Accelerator figures of merit

For Superfish problems with cylindrical symmetry, SFO reports a number of accelerator figures of merit derived from the solution. These figures of merit depend upon many parameters that have been entered in the Automesh input file, computed by Fish, or entered in the SFO input file. In particular, the total power dissipation reported by SFO and used in many of the figures of merit depends on correctly specifying which boundary segments to include in the calculation (in the FieldSegments list).

For the specified normalization method and length L, SFO calculates the average axial electric field E_0. It also calculates the stored energy U and the power P dissipated on the metal surfaces whose surface resistance is R_s. One figure of merit is the transit-time factor T discussed in Section H “Transit-time-factor integrals.” Given these quantities, SFO also reports the cavity Q, shunt impedance per unit length Z, and the effective shunt impedance per unit length ZT^2 defined as follows:

$$Q = \frac{\omega U}{P},$$

$$Z = \frac{E_0^2}{P/L},$$

and

$$ZT^2 = \frac{\left(E_0 T\right)^2}{P/L},$$

where ω is the angular frequency equal to 2πf. The code also lists a few other quantities of interest in certain applications. These include the product $R_s Q$ and the so-called "r over Q" parameter or r/Q, where r is the effective shunt impedance:

$$r = \frac{\left(E_0 TL\right)^2}{P} = \left(ZT^2\right)L,$$

and

$$\frac{r}{Q} = \frac{\left(V_0 T\right)^2}{\omega U},$$

where V_0 is the voltage drop seen by a particle at the crest of the rf wave (equal to $E_0 L$). The r/Q parameter measures the efficiency of acceleration per unit stored energy. It is independent of the surface properties that determine the power losses and depends only on the cavity geometry. Another figure of merit, usually only of interest for relativistic electron accelerators, is the wake loss parameter k. Typical units for k are V/pC. It is defined as:

$$k = \frac{\omega r}{4Q}.$$

3. Peak magnetic and electric fields and the Kilpatrick factor

SFO reports the maximum magnetic and electric fields along all the boundary segments included in the calculation. The output summary also lists the coordinates in the geometry where these maxima occur. The code also lists the average magnetic field along the outer wall of the cavity. For convenience, the code converts the average and peak magnetic fields to power densities using the following relation:

$$\frac{P}{A} = \frac{1}{2} R_s H^2,$$

where P/A refers to a power per unit area, R_s is the surface resistance and, H is the magnetic field.

Many people find it useful to relate the peak surface electric field to the Kilpatrick field (also called the "Kilpatrick limit"). For more information, see the paper by W. D. Kilpatrick "A Criterion for Vacuum Sparking Designed to Include Both R. F. and D. C."

in Review of Scientific Instruments **28**, page 824 (1957). Kilpatrick observed experimentally that the field level in resonant cavities at which breakdown would occur was related to the resonant frequency. T. J. Boyd of Los Alamos expressed Kilpatrick's result in a convenient formula, which is now widely used. The formula relates frequency F in MHz to the Kilpatrick limit E_k in MV/m:

$$F = 1.643E_k^2 \exp\left(\frac{-8.5}{E_k}\right).$$

Though the Kilpatrick field is no longer considered a limit on the maximum achievable field levels, it is still commonly used in reference to the design field level for accelerating cavities. Cleaner surfaces and better vacuum systems are now available compared to ~40 years ago. Field levels well over $2E_k$ have been achieved in some applications. Design field levels for modern accelerating cavities are typically in the range from $1.0E_k$ to $2.0E_k$. Program SFO calculates the Kilpatrick field and reports the ratio of the maximum surface field to the Kilpatrick field in the output summary. The Poisson Superfish code distribution includes a stand-alone utility program KILPAT for calculating the Kilpatrick field as a function of frequency.

4. Power and frequency-shift tables

The SFO output summary includes one or more tables listing computed power and frequency-shift information. A table labeled "Wall segments" simply summarizes most of the data at the ends of the "Power and fields…" sections found earlier in the output file. Table X-21 describes the column headings for the wall-segment table. The first row of the table contains only the starting coordinates of the first segment. The starting coordinates of a segment always appear in the previous row. After a segment that was omitted from the FieldSegment list, and when starting a new boundary region, the table contains another row of starting coordinates.

Note that the wall-segment table lists the peak surface electric field, but the average power per unit area. The more detailed sections earlier in the file include the peak surface magnetic field and the corresponding peak power density. The frequency shifts are from a Slater-perturbation calculation for 1-mm displacements of the segments. These displacements are assumed to increase the cavity volume.

If the FieldSegment list included any negative segment numbers, then the output summary includes another table after the heading "Stem segments." This heading includes the number of half stems and the stem radius. The stem-segment table is just like the wall-segment table, but without the Emax column (see Table X-21) and with only one column of frequency-shift data. Both the power and the frequency shifts reported for stems take into account field enhancements caused by distortion of the cavity field. The magnetic field varies sinusoidally with angle around the circular stem. Stems raise the frequency because they displace cavity volume containing predominantly magnetic field.

Table X-21. Data columns in the table of wall segments.

Column	Heading	Description
1	Segment	Boundary segment number assigned by Automesh.
2	Zend or Xend	Z or X coordinate at the end of the segment.
3	Rend or Yend	R or Y coordinate at the end of the segment.
4	Emax	Peak surface electric field on the segment.
5	Power	Total power dissipated on the segment.
6	P/A	Average power per unit area.
8	dF/dZ or dF/dX	Frequency shift for a 1-mm displacement in Z or X.
9	dF/dR or dF/dY	Frequency shift for a 1-mm displacement in R or Y.

A second table lists power and frequency shift data for multicell cavities with stems and post couplers. Table X-22 lists the column headings for this table, which appears in the output summary if the SFO input file included a StemData table. The Z position, radius, and length have the same units used for boundary-point data in the Automesh input file. Like the calculations discussed above for negative segment numbers, the stem-and-post-coupler table also accounts for the enhancement and sinusoidal variation of the magnetic field around the stems and post couplers.

If the SFO output file contains one or both of the tables for stem data, then the output summary also reports the “Cavity frequency corrected for stems and post couplers.” This line appears after the stem-and-post-coupler table, or if there was no StemData table in the input file, it appears after the stem-segment table. The corrected frequency reported here includes the effects of all the stems and post couplers from both tables.

Table X-22. Data columns for multicell stems and post couplers

Column	Heading	Description
1	(none)	Line number in the StemData table.
2	Type	Identifies a full or half stem or full or half post coupler.
3	Z Position	Location of the center of the stem or post from StemData table.
4	Radius	Stem or post-coupler radius from StemData table.
5	Length	Stem or post-coupler length from StemData table.
6	Power (kW)	Total power dissipated on the stem or post.
7	dF (MHz)	Total frequency shift caused by the stem or post.

Finally, another table in the output summary contains the coupling-slot effects computed by SFO if ISLOT is 1. An example of this table appears in Section F4 “Coupling slot effects.”

K. SFO error messages

Table X-23 lists the SFO error messages. The error numbers also are available as exit error codes to tuning programs and batch control files.

Table X-23. SFO error messages.

Error	Description
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.
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).
217	The solution file for a Superfish problem contains only mesh data and no solution arrays.
218	The solution file for a Poisson problem contains only mesh data and no solution arrays.
220	Reached the end of the solution file unexpectedly. Rerun problem starting 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 starting with Automesh.
292	Insufficient memory, cannot allocate MT namelist arrays for a Poisson problem.
293	Insufficient memory, cannot allocate MT namelist arrays for a Superfish problem.
600	Insufficient memory, cannot allocate arrays for boundary point data.
601	Insufficient memory, cannot allocate arrays for binary solution file data.
610	Unable to read the SFO input file.
611	The SFO input file contains an invalid entry..
612	A line in the CellData table contains nonnumeric data.
613	Too many entries in the CellData table.
614	The CellData table contains bad geometry for a cell.
615	A line in the StemData table contains nonnumeric data.
616	Too many entries in the StemData table.
617	A line in the FieldSegments list contains nonnumeric data.
618	Too many entries in the FieldSegments list.
619	A segment number in the FieldSegments list is outside the range of segments present.
620	A zero segment number has been entered in the FieldSegments list.
621	The FieldSegments list specifies an invalid segment number.
622	A line in the ResistanceSegments table contains nonnumeric data.
623	Too many entries in the ResistanceSegments table.
624	A segment number in the ResistanceSegments table is outside the range of segments present.
625	A line in the ResistanceSegments table contains an invalid segment number.
626	A line in the ResistanceSegments table contains an invalid option number IRTYPE.
627	Magnetic field is zero at the end of the segment specified for field normalization.
628	The starting point of the NORM = 4 integration path is outside the problem geometry.
629	The ending point of the NORM = 4 integration path is outside the problem geometry.
630	An intermediate point of the NORM = 4 integration path is outside the problem geometry.
631	The integration path for normalization option NORM = 4 has zero length.
632	The field integral is zero along the integration path for the NORM = 4 option.
633	The cavity length or the integration path is not correctly defined for NORM = 0 or 1.

Table X-23. SFO error messages. (continued)

Error	Description
634	The field integral is zero along the integration path specified for NORM = 0 or 1.
635	No cavity boundaries along the integration path used to compute transit-time integrals.
636	The integral of Ez is zero trying to compute transit-time-factor integrals..
637	The first cell of a multicell problem has zero or negative length.
638	The SFO input file contains an unrecognized keyword.

XI. SF7: Poisson Superfish Postprocessor for Field Interpolation

Program SF7 reads the solution computed by solver programs Fish, CFish, Poisson, or Pandira and tabulates the fields on lines, arc, user-supplied curves, or within rectangular regions. The code also can create input files for programs Parmela and EGUN. The Poisson Superfish codes run in the following order:

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

For Superfish problems, run SFO before running SF7 to apply the field normalization. If you do not run SFO first, then SF7 uses the default normalization that corresponds to H_ϕ = 1000 A/m at the drive point (or to a user-defined value of ASCALE). For Poisson and Pandira problems, there is no difference in the field components before and after running SFO, so SF7 can run immediately after the solver program Poisson or Pandira.

A. Starting program SF7

There are several ways to start SF7. With or without a separate input file with extension IN7, you can right-click on the Poisson Superfish solution file with extension .T35 and select “Interpolate (SF7).” If you have an input file with extension IN7, you also can double-click on the IN7 file, or right-click on the IN7 file and select Open. To open the IN7 file in Notepad, right click on the file and choose Edit. You can also use a command line (for example, in a batch file) to start the code:

SF7 *in7file* *T35file*

The filenames on the command line are optional. The parameter *in7file* is the name of an optional input command file with extension IN7, and *T35file* is the name of a binary solution file. Both *segfile* and *T35file* have the same name as the Automesh input file, but different extensions. It is usually not necessary to enter the name of the T35 solution file immediately after running a solver program (Fish, CFish, Poisson, Pandira) because the code will find the name of the solution file to open in file TAPE35.INF. Launching a program from the Windows Start menu does not supply any information on the command line. 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.

The code will read the default input file even if *in7file* does not appear on the command line. The default input file is *filename*.IN7, where *filename* is the name of the Automesh input file. For example, if you start Automesh with input file PROB1, then SF7 will look

for the file PROB1.IN7. If this file exists in the default directory (and you do not specify another file on the command line), the code reads it automatically. If the default input file does not exist, SF7 looks for a file named IN7.

1. Input using the dialog window

If there is no input command file, SF7 displays a dialog window where you can enter the information required for the job. The dialog window is useful for an occasional run on a problem. For repetitive jobs, it may be advantageous to create a command file as described in the next section. The dialog window includes a Select region where you can choose the type of interpolation (Line, Arc, Grid, Curve , or File). The File option lets you switch to the input command file method by supplying the name of an input file. You can either type the name of a file or click the Browse button to locate the file.

At the bottom of the dialog window, for your convenience you will find the name of the problem file and the title lines from the Automesh input file. As you change the choice in the Select region, other parts of the dialog window become active, indicating which input is required. For lines, arcs, and grids, the coordinates (X1,Y1) and (X2,Y2) are required. The default values for these coordinates and most other entries in the dialog window are in the [SF7] section of file SF.INI. An arc also requires the arc radius. Lines and arcs require the number of interpolation increments and a grid requires the number of increments along both coordinate directions.

One or more check boxes may be active, depending upon the type of problem and the choice in the Select box. For lines, arcs, and curves you can check the box to make an input file for program Tablplot. Variable MakePlotFile in the [SF7] section of file SF.INI determines the default setting for this check box.

For grids, if the problem has cylindrical symmetry you can create a Parmela input file. The default setting for the box labeled “Force E0 = 1 MV/m” can be set in file SF.INI by variable Force1MVperMeter. For Poisson or Pandira problems with cylindrical symmetry, you can check the box to create an EGUN input file. In this case, you need to supply a value for the EGUN mesh size rather than the X Steps and Y Steps values. Because of the different settings used to define the grid, SF7 will not make both a Parmela input file and an EGUN input file simultaneously. Run the code twice if you need both files.

2. Input using the command file

The input command file is useful for repetitive jobs that use the same input data. The first entry is a command keyword from Table XI-1 that selects the interpolation type. Optional keywords from Table XI-2 can appear on the same line after a command keyword. Most of the command keywords in Table XI-1 require input of physical coordinates on the second line and the number of interpolation points on the third line. The Curve keyword reads only one more line that gives the name of the *CurveFile* containing the curve coordinates. For Line and Arc commands, two coordinate pairs, (X1,Y1) and (X2,Y2), define the end points of the interpolation path. The Arc command also requires the arc radius R. For the Grid, Parmela, and EGUN commands, the coordinate pairs are diagonally opposite corners of a rectangle. The third data-entry line is the number of increments between the interpolation bounds. One increment is required for a line or an

arc and two for a rectangular grid. For the EGUN command, only the EGUN mesh interval δx is needed.

When the code finishes the first task, it reads the next line in the command file, which may contain data for another task. An End line stops the program.

Table XI-1. SF7 command keywords and data entry lines.

Keyword	Description	Second line	Third line
Line	Straight line from X1,Y1 to X2,Y2.	X1,Y1,X2,Y2	N
Arc	Arc of radius R from X1,Y1 to X2,Y2.	X1,Y1,X2,Y2,R	N
Curve	Curve with coordinates from a file.	*CurveFile*	(none)
Grid	Rectangular grid with corners X1,Y1 and X2,Y2.	X1,Y1,X2,Y2	N_x,N_y
Parmela	Grid with output to *filename*.T7 for Parmela.	X1,Y1,X2,Y2	N_x,N_y
EGUN	Grid with output to *filename*.EGN for EGUN.	X1,Y1,X2,Y2	δx

Table XI-2. SF7 optional keywords.

Keyword	Description
Plot	Create a Tablplot input file for Line, Arc, or Curve commands.
X, Y, Z, R	Use the indicated mesh coordinate as the abscissa in the initial Tablplot display.
A, H, V, E, B, Ex, Ey, Bx, By, Er, Ez, Br, Bz	If a potential, field magnitude, or field component from this list appears on a Line, Arc, or Curve command line, that quantity, if available, is included in the initial Tablplot display.

a. Line keyword for interpolation along a line

Figure XI-1 shows SF7 entries for interpolation along a line. Optional keyword Plot creates a Tablplot input file for displaying field components. The second line in Figure XI-1 gives the end-point coordinates of the line, and the third line is the number of increments along the line. In this example, the entries request 50 increments along a line from X,Y = 0.0,1.0 to X,Y = 10.0,1.0.

```
Line      Plot
0.0   1.0   10.0   1.0
50
End
```

Figure XI-1. Sample SF7 input for interpolation along a line.

For Poisson and Pandira magnet problems, SF7 computes the integral H·dl when interpolating fields along straight lines. At present, this feature assumes air for the material. The code writes a warning message if any part of the path enters another material. To do this calculation correctly for other materials requires the permeability data, which is not currently available in program SF7.

b. Arc keyword for interpolation along a circular arc

Figure XI-2 shoes SF7 entries for interpolation along an arc. Optional keywords Plot, Bx and By after keyword Arc creates a Tablplot input file for displaying the indicated field components. The second line in Figure XI-2 gives the end-point coordinates and radius of

the arc, and the third line gives the number of increments along the arc. In this example, the entries request 20 increments along an arc of radius 1.0 from X,Y = 0.0,1.0 to X,Y = 1.0,1.0.

```
Arc      Plot       Bx    By
0    1    1    0    1
20
End
```

Figure XI-2. Sample SF7 input for interpolation along an arc.

c. Curve keyword for interpolation along a user-supplied curve

Figure XI-3 shows SF7 entries for interpolation along a user-supplied curve. The second line CDATA is the name of a *CurveFile* containing a list of interpolation points. One use of the Curve command is finding the fields along an equipotential curve in electrostatic Poisson problems. You can get the coordinates along an equipotential from the WSFplot output file.

```
Curve
CDATA
End
```

Figure XI-3. Sample SF7 input for interpolation along a user-supplied curve.

d. Grid keyword for interpolation on a rectangular grid

Figure XI-4 shows SF7 entries for interpolation on a rectangular grid. SF7 does not create a plot file when interpolating on rectangular grids. The second line in Figure XI-4 defines the X range and Y range of the grid. The line contains two X,Y coordinate pairs at diagonally opposite corners of a rectangle. The third line contains the number of increments in X and Y directions. In this example, the entries request 14 increments along X from 0.0 to 1.5 and 7 increments along Y from 0.0 to 0.2.

```
Grid
0    0    1.5    0.2
14   7
End
```

Figure XI-4. Sample SF7 input for interpolation on a rectangular grid.

e. Parmela keyword to create input for program Parmela

The Parmela command interpolates static magnetic or electric fields from Poisson and Pandira problems and electromagnetic fields from Superfish problems for input to program Parmela. (Older version of SF7 called this the “Tape” command. You also can select this feature by checking the appropriate box in the SF7 dialog window.) Like the Grid command, the fields are interpolated on a rectangular grid. However, the code creates a second output file named *filename*nn.T7, where *filename* is the name of the Automesh input file and nn is a 2-digit sequence number. For example, if you start Automesh with namelist input file PROB.AM or PROB.AF, then SF7 will create files PROB01.T7, PROB02.T7, etc.

Parmela reads the T7 files for rf problems after CField and TRWCField lines in its input file. Parmela reads T7 files for magnet or electrostatic problems after the Poisson line in its input file. (Note: Programs Poisson and Pandira also create static 2-D field maps for Parmela if variable ParmelaFields = Yes setting in [Poisson] or [Pandira] section of file SF.INI when running these codes.)

See Data files for program Parmela for a description of file *filename*nn.T7. The Parmela documentation also describes the file. For Superfish problems, before using SF7 to make a Parmela input file, be sure to run SFO to normalize the fields. You can use any normalization that results in a nonzero value for average axial field EZERO. If you fail to normalize the fields, SF7 will stop with an error message. SF7 will scale the results to EZERO = 1.0 MV/m unless the setting Force1MVperMeter = No appears in file SF.INI.

Since Parmela expects 2-D field maps in cylindrically coordinates, SF7 allows the Parmela command only for problems with cylindrical symmetry.

f. EGUN keyword to create input for program EGUN

The EGUN command interpolates static magnetic or electric fields from Poisson and Pandira problems for input to program EGUN. The fields are on a rectangular grid like the Grid command. SF7 writes file OUTSF7.TXT as it does for the Grid command, but it also creates file *filename*nn.EGN, where *filename* is the name of the Automesh input file and nn is a 2-digit sequence number. For example, if you start Automesh with namelist input file PROB.AM.AF, then SF7 will create the files PROB01.EGN, PROB02.EGN, etc.

Program EGUN uses a constant mesh with $\delta x = \delta y$ for problems in Cartesian coordinates or with $\delta r = \delta z$ for cylindrically symmetric problems. Unlike the Grid command which requires the number of X and Y increments, SF7 needs only the EGUN mesh interval δx. The code then computes the number of increments in the X and Y (or R and Z) directions. If necessary to preserve a square mesh, SF7 will reduce the X and Y physical upper limits by less than one mesh interval.

The section titled Files and filename conventions includes a brief description of EGUN and lists the person to contact for more information.

B. SF7 files and filename conventions

SFO reads the binary solution file written by Fish, CFish, Poisson, Pandira, or SFO. The code writes the OUTSF7.TXT output summary. You can supply an input file that contains the usual keyboard entries for SF7. This option permits batch control files and tuning programs (such as DTLfish) to run unattended. The default name for this file is the name of the Automesh input file plus the extension IN7.

For each command specifying interpolation along a line, arc or curve, SF7 writes a Tablplot input file of the interpolated fields if the optional Plot keyword appears on the command line. The Parmela command writes file *rootname*nnn.T7, where *rootname* is first several characters (see discussion on naming plot files) of the name of the Automesh input file and nnn is a sequence number of one or more digits. This output file contains electromagnetic field data on a grid for the program Parmela. Similarly, the EGUN command writes files *rootname*nnn.EGN, which contain electromagnetic field data on a

grid for the program EGUN. See Files for other programs in the Poisson Superfish documentation for more information about these files.

C. SF7 output text file

The code writes file OUTSF7.TXT, which contains a table of fields at the requested X and Y (or R and Z) coordinates. 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.

For rf problems in Cartesian coordinates the code reports E_x, E_y, E, and H_z. For rf problems in cylindrical coordinates the code reports E_z, E_r, E, and H_ϕ.

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 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. For electrostatic problems in cylindrical coordinates the code reports V, E_r E_z, and E.

The data table can be expanded to include additional data from the fitting algorithm. the fitting function number used by the field interpolator and the χ^2 per degree of freedom for the fit. For more information refer to section F, “Field-interpolation algorithm in SF7.”

D. Creating Tablplot files of the interpolated fields

For interpolations along lines, arcs, or curves you can make SF7 produce input files for program Tablplot as it writes information to the standard output file OUTSF7.TXT. The dialog window includes a check box to activate this feature. If selected, the Tablplot file contains standard settings chosen by SF7 for the initial plot. A command file allows more flexibility than the dialog window for defining the initial screen when Tablplot starts. If keyword Plot appears on the Line, Arc, or Curve lines, or if you check the appropriate box in the dialog window, then SF7 creates a Tablplot input file for displaying field components.

Tablplot filenames have extension TBL. SF7 writes the Tablplot files to the current directory. The filenames will consist of the first several characters of the Automesh input filename plus a sequence number (starting with 1). By default the code uses the first 6 characters. You can instruct SF7 to use more characters (up to the full length of the name) by increasing variable RootNameLength in the SF7 section of SF.INI. Suppose the Automesh input file was PROBABC.AM and you later run SF7 with input from the file shown in Figure XI-5. This example produces two Tablplot files named PROBAB1.TBL and PROBAB2.TBL. The first file contains 50 points along a line at R = 0 from Z = 0.0 to 5.0. Another file PROBAB2.TBL contains 200 points along a line at Z = 10 from R = 0.0 to 25.0. There will be no plot file for the second Line command, but this data appears on the screen as the program calculates it.

```
Line     Plot
0.0  0.0  5.0  0.0
50
Line
0.0  1.0  5.0  1.0
20
Line     Plot
10.0    0.0    10.0    25.0
200
End
```

Figure XI-5. Sample SF7 input containing several commands.

The files written by SF7 contain the same data columns that appear in OUTSF7.TXT plus the necessary title and label sections required by Tablplot. The default setting in the Tablplot file is to plot E_z and E_r (or E_x, E_y; B_x, B_y; B_r, B_z as appropriate) versus Z (or X). If the Z (or X) coordinate does not change, the code changes the abscissa to R (or Y). To change the initial display to a particular field component, put the name of the component on the SF7 command line. For example, to display only the magnitude of electric field for the first command in the example above, the command line would be:

```
Line Plot E
```

To display both field components and the magnitude of E the command would be:

```
Line Plot E Ex Ey
```

You also can specify which mesh coordinate to use as the abscissa when Tablplot starts by including the letter X, Y, R, or Z on the command line. However, Tablplot will ignore this choice if that component is constant over the range of the interpolated data. Naming a coordinate axis or field components on the SF7 command line affects only the initial plot displayed by Tablplot. You can plot other columns by using menu item Data after the code is running. Tablplot can also integrate a data column or a curve fitted to a data column.

Variable MakePlotFile in the [SF7] section of file SF.INI determines the default setting for plot-file check box in the dialog window. For command file, the default in SF7 is not to write Tablplot files. For files with multiple interpolations (more than one Line, Arc, or Curve section), you must include the Plot keyword on each command line for which you want a plot file.

1. Plotting Slater-Perturbation Frequency Shifts

For rf cavities, SF7 has an option selected by SF.INI variable SlaterTerm to include frequency-shift data in the Tablplot input file. The code computes three different frequency shifts df (in kHz) using the following equation for a perturbation volume $\delta V = 1\ mm^3$:

$$df = \frac{k_H \mu H^2 - k_E \varepsilon E^2}{4U} \delta V f\ ,$$

where μ and ε are the material permeability and permittivity, H and E are the magnetic and electric fields, f is the cavity frequency, U is the cavity stored energy. For cylindrically symmetric problems U is the stored energy for the problem geometry. For problems in Cartesian coordinates, U is the stored energy in a 1-meter-long cavity having the cross section of the problem geometry.

The three frequency shifts correspond to the different choices for shape factors k_H and k_E listed in Table XI-3 selected by SF.INI variable SlaterTerm. Program WSFplot has an option to show Slater-perturbation frequency shifts in the field-display window. To use this option in SF7, set SlaterTerm = Yes or use one of the values in the first column of Table XI-3. The code will include all three terms in the Tablplot input file. If you specified a keyword from Table XI-3, then your choice will be one of the initial columns of data plotted when Tablplot starts. You can select the other columns using the menu item Data after the program is running.

Table XI-3. Shape factors for settings of variable SlaterTerm.

SlaterTerm	Description	k_H	k_E
Surface	Perturbing volume does not distort the fields.	1.0	1.0
Sphere	Perturbing volume is a sphere.	1.5	3.0
Cylinder	Perturbing volume is a circular cylinder.	2.0	2.0

For an empty cavity, μ and ε are equal to μ_0 and ε_0. The perturbing volume is assumed to be a metallic object that completely excludes the original electromagnetic fields computed by Superfish. The Surface term corresponds to a perturbation that does not significantly distort the fields, for example, a small displacement of a cavity wall. In a cylindrically symmetric cavity with this setting, you interpolate the fields along a wall and interpret the frequency shift as a 1-mm wide indentation of the wall (into the cavity) of thickness $1/2\pi R$ around the cavity azimuth, where R is the radial location of the indentation. The Sphere term gives the frequency shift produced by a metallic spherical bead of radius 0.620 mm. The Cylinder term gives the frequency shift for a 1-mm thick slice of a cylinder (e.g., a drift-tube stem) of radius 0.564 mm.

E. Power and stored energy integrals in SF7

When using the Grid option, problem variable ICCP indicates whether SF7 should calculate electric and magnetic stored energies, magnetic flux, and (for complex problems solved by CFish) power losses in dielectric and magnetic materials. A value of ICCP > 0 tells the code to calculate these integrals. Because this calculation may take some time for a large number of mesh points, you may wish to turn it off until you are satisfied with the design of the cavity. The value of ICCP in Table IX-7 indicates which points on a mesh triangle to use in calculating the average fields for each mesh triangle.

You can enter a value for ICCP in the REG namelist section of the Automesh input file. If you run SFO before running SF7, and enter a value for variable MaterialPowerCode in the SFO input file, the code sets ICCP = MaterialPowerCode for subsequent calculations in SF7 and WSFplot. SF7 makes a table in file OUTSF7.TXT for each material within a

rectangular region. At the end of the table of interpolated fields, the code writes the following quantities:

$$U_E = \frac{1}{2}\int_{Z_1}^{Z_2}\int_{R_1}^{R_2} 2\pi\varepsilon E^2 r\,dr\,dz \ \text{[Joules] or } \frac{1}{2}\int_{X_1}^{X_2}\int_{Y_1}^{Y_2} \varepsilon E^2\,dx\,dy\ \text{[Joules/m]},$$

$$U_H = \frac{1}{2}\int_{Z_1}^{Z_2}\int_{R_1}^{R_2} 2\pi\mu H^2 r\,dr\,dz \ \text{[Joules] or } \frac{1}{2}\int_{X_1}^{X_2}\int_{Y_1}^{Y_2} \mu H^2\,dx\,dy \ \text{[Joules/m]},$$

$$\Phi = \int_{Z_1}^{Z_2}\int_{R_1}^{R_2} \mu H\,dr\,dz \ \text{[Webers] or } \int_{X_1}^{X_2}\int_{Y_1}^{Y_2} \mu H\,dx\,dy \ \text{[Webers/m]},$$

$$P_E = \omega U_E \frac{\varepsilon_2}{\varepsilon_1} \quad \text{[Watts or Watts/m]},$$

$$P_H = \omega U_H \frac{\mu_2}{\mu_1} \quad \text{[Watts or Watts/m]},$$

where $\varepsilon = \varepsilon_1 + i\varepsilon_2$ and $\mu = \mu_1 + i\mu_2$, and the ratios $\varepsilon_2/\varepsilon_1$ and μ_2/μ_1 may be identified as the loss tangents for dielectric and magnetic materials, respectively. The output includes values for the limits of integration and the material properties. If the rectangular region includes the entire cavity, then the results for the two stored-energy integrals should be in reasonable agreement with one another. These values also should agree approximately with the stored energy reported by program SFO. The difference between the electric and magnetic stored-energy integrals will depend upon the mesh size. Finer meshes result in better agreement.

SFO makes a similar table of the power dissipated by electric and magnetic fields in each material for the entire cavity. WSFplot displays the total power loss for all materials within the display boundaries.

F. Field-interpolation algorithm in SF7

The field interpolator for all types of problems (rf, static magnetic, and static electric fields) fits one of several polynomial functions of X and Y (or R and Z) to the first and second nearest neighbors of the closest mesh point. Different functions satisfy either Neumann or Dirichlet boundary conditions on or near vertical or horizontal boundaries.

Because the field interpolator uses first and second nearest neighbors, you may see erroneous values if the mesh is too coarse in the vicinity of thin metal objects. For example, near a metal plate that is only one row of mesh triangles thick, the code may include points in the fit from the other side of the plate. These interpolated fields will be incorrect. The only solution is to make this part of the mesh finer. You can check the fits by looking at the chi-squared goodness of fit parameter printed by SF7 when it runs with the expanded table option. To include additional data from the fitting algorithm in file OUTSF7.TXT insert the following line in the [SF7] section of SF.INI:

ExpandedTable = Yes

The expanded table includes the new column headings defined in Table XI-4. The Chi^2 parameter includes no estimate of uncertainties. It is the sum of the absolute squares of the magnetic field differences between the data and the fit at each included mesh point divided by the number of degrees of freedom. The number of degrees of freedom is the number of data points in the fit minus the number of fitted coefficients. The chi-squared is normalized to the square of the average of the absolute values of solution array (H, A, or V) for the points used in the fit.

Table XI-4. SF7 expanded data-table columns.

Column	Description
D	T (for true) means the point is on or near a Dirichlet boundary.
N	T means the point is on or near a Neumann boundary.
Chi^2	Chi-squared per degree of freedom.
F#	The function number used to fit the data.
K,L	Logical coordinates of the nearest mesh point (two columns).
Num	Number of data points in the fit.
P(I)	Fitted coefficients P_1 through P_8.

Fits have from two to eight coefficients depending upon whether the point is on or near Dirichlet or Neumann boundaries. Near boundaries the code either excludes or constrains some terms. Good fits typically have Chi^2 much less than 10^{-7}. The column label F# refers to a function number listed with the discussion of the field interpolator.

When interpolating along a line or curve, if you request more than one interpolation point per mesh interval, a plot of the curve may show raggedness under some circumstances. A coarse mesh often exhibits this ragged behavior in the vicinity of rapid changes in the field. The only real solution is to make the mesh finer. An unfortunate choice of the path can cause unphysical looking spikes in a field-component curve. For example, consider a horizontal path through a mesh of equilateral triangles with legs of length s. If the path is between about 0.29s and 0.71s above one row of mesh points, then the nearest point to the interpolation points along the line will alternate between the two rows as illustrated in Figure XI-6. This effect causes the interpolator to select a different set of nearest neighbors for points closely spaced on the path. If the field has significant second derivatives in this vicinity, the interpolated values can differ appreciably.

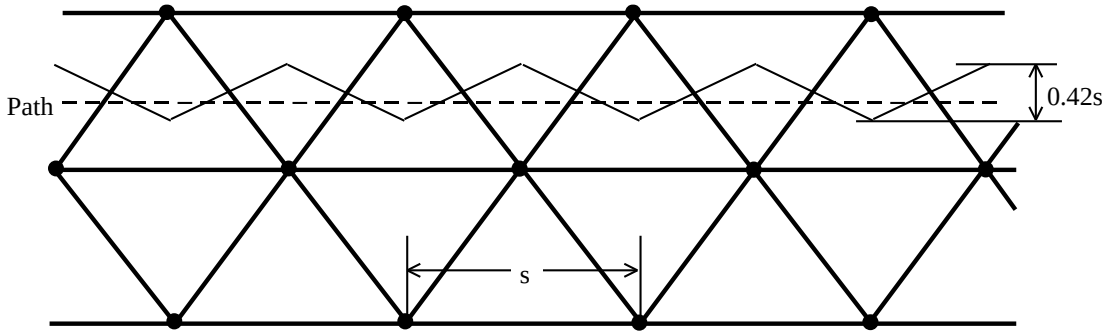

Figure XI-6. Interpolation path between mesh rows.
When the path is below the zigzag line, the nearest mesh point is on the middle row in this figure. When the path is above the zigzag line the nearest point shifts to the top row.

G. Writing real and imaginary CFish fields

The ComplexFields configuration variable tells SF7 whether to write both real and imaginary components of H_ϕ, E_r, and E_z for Superfish problems the complex fields. This setting affects only problems solved with CFish. The default setting for ComplexFields is No. If you need this detailed information, insert the following line in the [SF7] section of SF.INI before running SF7:

ComplexFields = Yes

H. SF7 error messages

Table XI-5 lists the SF7 error messages. The error numbers also are available as exit error codes to batch control files.

Table XI-5. SF7 error messages.

Error	Description
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.
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).
217	The solution file for a Superfish problem contains only mesh data and no solution arrays.
218	The solution file for a Poisson problem contains only mesh data and no solution arrays.
220	Reached the end of the solution file unexpectedly. Rerun problem starting 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 starting with Automesh.
292	Insufficient memory, cannot allocate MT namelist arrays for a Poisson problem.
293	Insufficient memory, cannot allocate MT namelist arrays for a Superfish problem.
601	Insufficient memory, cannot allocate arrays for binary solution file data.
647	Insufficient memory, cannot allocate arrays for Parmela fields.
648	Unexpected end of file occurred reading the SF7 input file.
649	Unable to read the SF7 input file.
650	When creating field data for Parmela, program SFO must run before SF7 to normalize fields.
651	A line in the SF7 input file contains an unrecognized keyword command.
652	The line of grid corner coordinates in the SF7 input file contains insufficient data.
653	The line of grid corner coordinates in the SF7 input file contains nonnumeric data.
654	The number of grid increments for EGUN data is not a valid number.
655	The number of grid increments for EGUN data is zero or negative.
656	The line of grid increments in the SF7 input file contains insufficient data.
657	The line of grid increments in the SF7 input file contains nonnumeric data.
658	The line of arc center coordinates and radius in the SF7 input file contains insufficient data.
659	The line of arc center coordinates and radius in the SF7 input file contains nonnumeric data.
660	The number of increments along an arc is not a valid number.
661	The number of increments along an arc is zero or negative.
662	The line of end-point coordinates in the SF7 input file contains insufficient data.
663	The line of end-point coordinates in the SF7 input file contains nonnumeric data.
664	The number of increments along a line is not a valid number.
665	The number of increments along a line is zero or negative.

XII. Force: Poisson Postprocessor for Calculating Magnetic Forces

Force calculates magnetic forces on coils or iron regions. Force must run after Poisson or Pandira for static magnetic field problems. The codes run in the following order:

- Automesh
- WSFplot(optional)
- Poisson, or Pandira
- WSFplot (optional)
- SFO or SF7 (all problems); or Force (Poisson/Pandira problems only)

No documentation for Force appears in 1987 Reference Manual and User's Guide. The rough draft titled "Chapter 10 Examples, 10.12 Force – Infinite Wire" shows how to set up a Force problem for older UNIX code in version 4. However, the rough draft does not address some problems we have found with the code. We have provided enough information here to run the code. The present version of Force has more options for entering the integration path and it does more checking to be sure data entries are consistent with the problem geometry.

A. Starting program Force

To start Force, right-click on the Poisson Superfish solution file with extension .T35 and select "Run Force," or double-click on the Force n file with extension FCE file, or right-click on the FCE file and select Open. To open the FCE file in Notepad, right click on the file and choose Edit. You can also use a command line (for example, in a batch file) to start the code:

Force *forcefile* *T35file*

The filenames on the command line are optional. The parameter *forcefile* is the name the Force input file with extension FCE, and *T35file* is the name of a binary solution file. Both *forcefile* and *T35file* have the same name as the Automesh input file, but different extensions. It is usually not necessary to enter the name of the T35 solution file immediately after running Poisson or Pandira because the code will find the name of the solution file to open in file TAPE35.INF. Launching a program from the Windows Start menu does not supply any information on the command line. 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.

The code will read the default input file even if a *forcefile* does not appear on the command line. The default input filename consists of the name of the Automesh input file plus the extension FCE. For example, if you start Automesh with namelist input file PROB1, then Force will look for the file PROB1.FCE. If this file exists in the default directory (and you do not specify another file on the command line), the code reads it automatically. If the default FCE file does not exist, the program looks for a file named FORDATA.

The first entry in a Force input file is the region number of the material on which you wish to calculate the force. The code will check that the region you enter actually exists and if it makes sense to calculate a force on it. For example, you cannot calculate the force on a region that contains only open space with no current density. Line regions between areas of fine and coarse mesh fall into this category.

After the region number you will define an outline of the material on which to find the force. We refer to this as the force element. The force element need not be a material's entire region. Force reads physical (X,Y) coordinates. If the force element includes the whole region, then enter the coordinates of any point on the boundary followed by the word COUNT (or any abbreviation). Force will find the rest of the coordinates around the region. If the force element does not include the whole region, then use as many lines as necessary to enter the end coordinates of connected line segments. Enter the word COUNT at the end of the last line.

B. Force files and filename conventions

Force reads an input file whose default name consists of the name of the Automesh input file plus the extension FCE. The code reads the solution written by Poisson or Pandira in the binary solution file. Force writes its output summary in the OUTFOR.TXT file. For more information, refer to Files and filename conventions.

C. Output from Force

The OUTFOR.TXT file contains a complete list of problem constants followed by a list of the logical-coordinate path around the force element. Calculated forces appear at the end of the file. Subdirectories of LANL\Examples\Magnetostatic include several example problems for which the force on an object is computed.

1. Output for Cartesian coordinates

For Cartesian coordinates, the code reports the X and Y components of the force and the magnitude of the resultant force. These quantities have dimensions of force per unit length. For convenience, the code reports the results in N/m, kg/m, and pounds/inch.

For rectangular meshes, the code also reports information about torque or moments about the origin. This data appears in terms of the line-of-action parameters r and θ, where |r| is the perpendicular distance of the line of action from the origin (X,Y = 0,0), and θ is direction of the force (θ = 0 is along X, θ = 90 is along Y). A positive value for r corresponds to a positive torque, which means a vector along +Z in a right-handed coordinate system.

2. Output for cylindrical coordinates

For problems with cylindrical symmetry (ICYLIN = 1), the abscissa is the R axis and the ordinate is the Z axis. The code reports the radial and longitudinal force components per radian, the net force per radian, and the total force in the z direction. The radial component of the total force is zero because of the cylindrical symmetry. Again, the code reports these force in several units (N, kg, and lb).

D. Iron or coil region on which to calculate the force

To run Force, you must supply the region number of the material on which to calculate the net force and an outline of the material, unless the force element is the entire region. For the case of an entire region, only a single coordinate on the region's boundary is needed in an input file. If you use the pop-up menu (simply by not supplying an input file), you need only supply the region number of the force element. The code will find the boundary points from data supplied by Automesh.

When supplying a path in an input file, the outline consists of a closed path of physical (X,Y) coordinates. Closed means that the first and last coordinates must be the same. The program constructs a complete integration path around the force element by connecting the points you supply with line segments. If you supply just one point on the region boundary, the code will find the path that surrounds the entire region.

The code obtains nonzero force components only along the boundaries between regions. If you specify a path entirely within iron or empty space, the force will be zero. Thus, at least part of the integration path should lie along a region boundary. Consult file OUTAUT.TXT for the problem at hand for lists of coordinates along all the region boundaries.

E. Other considerations when using Force

1. Force does not make use of symmetry planes

Program Force does not compute the force components correctly if part of the path lies on a plane of symmetry. In order to use this code, the problem space must include the full geometry.

2. Meshing the problem geometry

Program Force does not try to optimize the step size along the integration path. The code uses the mesh triangles adjacent to the force element that were generated by Automesh. The user must ensure that the mesh is fine enough for accurate results. Also, the user should avoid using sharp corners on objects. Such features can lead to large local fluctuations in the field near the force element. Since the force per unit area depends on the square of the fields, anomalous fields at a sharp corner can dominate the entire force calculation.

3. Possible sign errors in the force components

The sign of force components reported by Force may be wrong in some cases. In tests we have run, it appears that the direction of the path around the force element makes a difference. If you supply just one point on the boundary of a region, then the code finds the integration path around the whole region. For the limited number of cases we have run, we have not observed an incorrect sign if the code finds the integration path. The code always chooses a path that goes counterclockwise around the region. It is tempting to say that signs are correct if the path is counterclockwise, but we are not sure this is true in all cases.

F. Force error messages

Table XII-1 lists the Force error messages. The error numbers also are available as exit error codes to batch control files.

Table XII-1. Force error messages.

Error	Description
190	A line in the Force input file contains nonnumeric data.
191	Unknown error reading the Force input file.
192	Unexpected end of file occurred reading the Force input file.
193	Unexpected character encountered in the Force input file.
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.
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).
218	The solution file for a Poisson problem contains only mesh data and no solution arrays.
220	Reached the end of the solution file unexpectedly. Rerun problem starting 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 starting with Automesh.
292	Insufficient memory, cannot allocate MT namelist arrays for a Poisson problem.
601	Insufficient memory, cannot allocate arrays for binary solution file data.
670	Requested force on a region that does not exist.
671	Requested force on an unclosed line region.
672	Requested force on a region containing open space with no current density.
673	A path requires more points than the longest path set up by Automesh.
674	No data for the path coordinates have been entered.
675	The number of input data values for a path is not an even number.
676	A point on the path is outside the problem geometry.
677	A supplied path of multiple points is not closed.
678	The logical coordinate L for a point on the path is outside the problem geometry.
679	The logical coordinate K for a point on the path is outside the problem geometry.
680	Consecutive points are inconsistent with a valid logical path.
681	The starting mesh point for a path is not on the boundary of the specified region.
682	A region has too many boundary points. Please report this error.
683	A leg of the integration path is outside the problem geometry.