POISSON/SUPERFISH on PC Compatibles*

James H. Billen and Lloyd M. Young
Los Alamos National Laboratory
Los Alamos, NM 87545

Abstract

We have adapted the POISSON/SUPERFISH codes to run on 486 or 386 PCs. The PC version includes features not found in the standard version, including programs for automatically tuning RFQ, DTL, and CCL cavities; a complex version of the rf field solver; memory allocation for temporary data; many line regions for dividing the mesh into fine or coarse sections; full support for multiple-cell DTL cavities; plotting of resonance-search and transit-time data; and on-line documentation. We modified AUTOMESH to generate self-consistent logical and physical coordinates. This new, more robust code greatly reduces the number of crashes in LATTICE caused by ill-formed mesh triangles along boundaries. The codes solve arbitrarily large problems. Each program allows free-format entry of CON array elements, and provides error checking of user entries. Standard release 4 now uses our root finder and convergence criteria.[1]

Introduction

In 1985, one of us (Young) adapted the Cray version of these codes to run on IBM-PC-compatible computers. He improved several algorithms that gave trouble in the original version. In 1992, we modified the codes extensively. The goal was to do larger problems, but we also improved the user interface, the field-interpolation algorithm in post processors, and the root finder for resonance searches. We added DOS exit error codes, control programs to tune cavities, post processor support for multiple-cell cavities, and on-line documentation. We use the Lahey FORTRAN compiler F77L-EM/32.[2] The Lahey debugging tools greatly aided our development work.

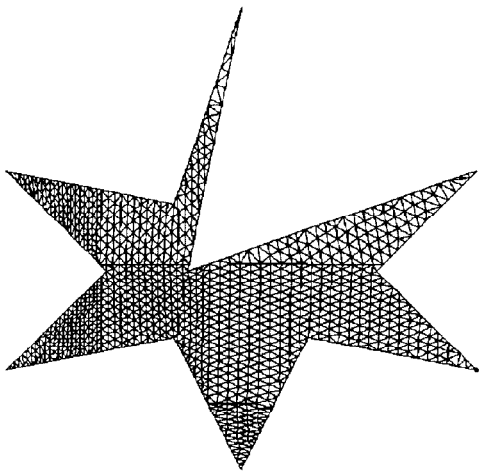

Figure 1. A "star cavity" to test the AUTOMESH code.

* Work supported by the U. S. Department of Energy.

The User Interface

AUTOMESH, the first code run on a problem, uses NAMELIST input. We retain the CON array as the method for codes to share information. The PC version allows entry of more variables (including most CON elements) in the NAMELIST file. CON element input is truly free-format and the codes check and warn of possible errors in input values. New CON elements control the field normalization, surface resistance, and accelerated particle mass in SFO. Some CON elements cannot be changed by the operator, and others can change only in appropriate codes. The codes prevent user input of unchangeable CON elements. For example, boundary conditions must not change after program LATTICE. The starting frequency can change in AUTOMESH, LATTICE, or FISH, but not in post processor SFO. Output files list CON values used in their respective programs and show where the user last changed a value.

AUTOMESH

AUTOMESH lists logical coordinates along boundaries for LATTICE. Integer coordinates K,L describe the logical overlay mesh and X,Y are the physical coordinates. Older AUTOMESH routines did little checking for consistency among points, sometimes resulting in triangles for which LATTICE calculates a negative or zero area. We limit choices for K,L for boundary points, line-region intersections with the boundary, and points along the connecting path. For example, all points with identical Y must use the same L, and likewise for X and K. These checks prevent most "negative-triangle" crashes in LATTICE. Figure 1 shows a sample geometry that we use to test AUTOMESH. This "star cavity" has several challenging features. Sharp corners are often hard to mesh. Line regions that intersect near a boundary were usually too much for the old version of AUTOMESH.

A New Root Finder

In SUPERFISH, resonances occur at slope=−1 roots of the $D(k^2)$ function.[3] Wave number k corresponds to frequency f ($k=2\pi f/c$). Subroutine FROOT finds a slope=−1 root using polynomial approximations. The original root finder, though used for many years, was not very robust. Occasionally, it did not converge to the nearest mode and sometimes could not find a mode at all. It stored $D(k^2)$ and k^2 at each iteration, but sometimes it swapped a new point with a previous point. It never used new data to update the slope of previous points. Derivatives of $D(k^2)$ were usually wrong because the calculation used adjacent points in unsorted arrays. Because it made little use of these derivatives, SUPERFISH usually converged to a resonance near the initial frequency. Our new

version of the FROOT subroutine sorts the data by k^2 and recalculates first and second derivatives after every iteration. It analyzes the data to place upper and lower bounds on the location of the desired root. Taking a smaller first step toward resonance gives a more reliable estimate of the local slope and reduces the chances of missing a mode where the function has a narrow local minimum.

After four or more iterations, FROOT selects three points for a parabolic interpolation. Carefully chosen points speed convergence to the root. We use the last chronological point and lower or upper bounds, if available. Other points must be adjacent to or between the bounds. Given a choice, FROOT favors points with slope close to −1. With only two points, the code examines the two parabolas that go through both points and have slope equal to −1 at $D(k^2)=0$. It picks the k^2 within the current bounds. If parabolic approximations fail to find a new k^2 consistent with the upper and lower bounds, FROOT uses one of several contingencies that give more information about the local shape of the function.

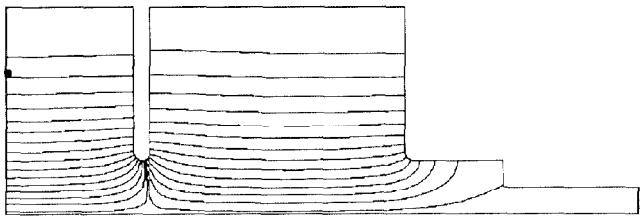

Figure 2. Electric field lines for the lowest π mode of a three-cell cavity. The left edge is a symmetry plane and r = 0 is at the bottom of the figure. The dot at left shows the drive-point location.

We tested the new root finder on many cavity shapes, sometimes deliberately placing the drive point in unfavorable locations. Figure 2 shows electric field lines for the 499.58-MHz π mode of a three-cell coupled cavity. Close proximity to the zero mode, 3.7 MHz lower in frequency, was a problem for the old version. The new code always found the nearest mode for 60 starting frequencies between 490 and 505 MHz. FISH makes a plot file of mode-searches and frequency-scan data. Figure 3 shows such a $D(k^2)$ plot over some higher-order modes of this cavity. If the drive is in weak field for a

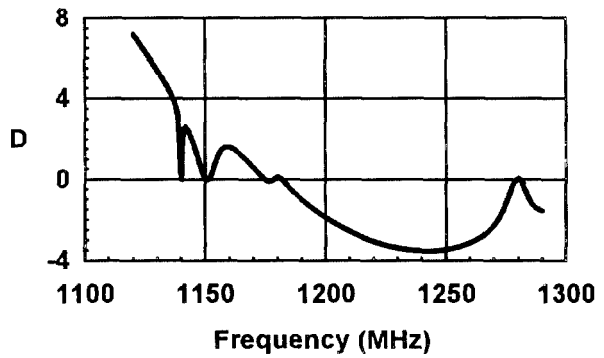

Figure 3. The function $D(k^2)$ plotted versus frequency over the range of 5 higher-order modes.

mode, the function barely crosses zero and challenges the root finder. The new FROOT never failed to find a nearby mode for a wide range of starting frequencies. For example, it found the 1140-MHz mode for 12 starting frequencies from 1139.3 to 1140.4 MHz. The old version found this mode only when the starting frequency was within 40 kHz of resonance.

Convergence Criteria

For convergence to a resonant frequency, SUPERFISH requires that $D(k^2)$ is small enough and has the correct slope:

$$\frac{|D(k^2)|}{k^2} < \varepsilon; \quad \frac{|\delta(k^2)|}{k^2} < \varepsilon; \quad \text{and} \quad -1.02 < \frac{dD}{d(k^2)} < -0.98.$$

The convergence parameter ε is typically 0.0001, and $\delta(k^2)$ is the next proposed change in k^2 after an iteration. Standard versions of SUPERFISH through version 3.0 used only the second of these criteria. The code did not actually require a small $D(k^2)$ and it did not check for a negative slope. Newer releases of the standard version use our convergence criteria. Failure to check the slope is not a problem unless the starting frequency is near a positive-slope root. After one iteration, the code does not know the slope. LATTICE initializes the slope to +1, forcing a second iteration. If the starting frequency is known to be near resonance, then the operator (or a control program) can set the slope to −1 to allow the code to converge in only one iteration.

Large Problems

Mesh dimensions in the SUPERFISH codes can be as large as the computer resources allow. The original codes were new when memory was scarce, so they shifted and masked bits to store multiple pieces of information in one computer word. The "INDEX" array allotted 5 bits for region numbers and 15 bits for the mesh-point counter. In all, INDEX stored seven items for each point. The PC codes eliminate mask and shift operations. We use a four-byte integer array for the mesh-point counter, two-byte integer arrays for region numbers, and one-byte integer and logical arrays for other data previously coded in the INDEX array. (Standard version 4 now uses a similar strategy.)

FISH and POISSON run faster if the mesh is fine only where it needs to be. AUTOMESH allows many divisions in the mesh, called line regions, for changing the mesh size. Figure 4 shows fields from an example calculation for an entire drift-tube linac tank. The dot above the sixth full drift tube from the left is the location of the drive point. The mesh had about 130,000 points. The mesh resolution near the curved surfaces on drift tubes was about one-fourth of the radius of curvature. For this problem, an iteration took 35 minutes on a 66-MHz 486 computer. Allocating memory for temporary arrays also speeds execution. While solving the tri-diagonal matrix, program FISH stores data for each mesh row, which it later reads back in reverse order. Instead of storing this data on disk, the program can use available memory. For the problem

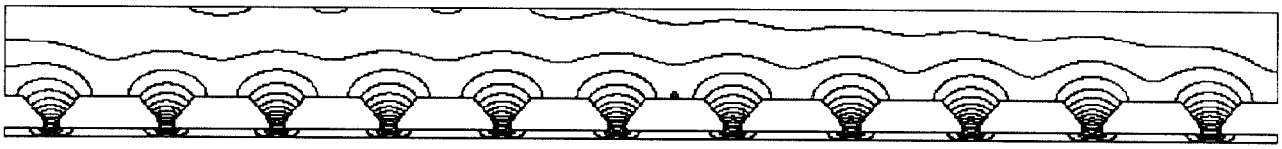

Figure 4. Electric field lines in an 11-cell drift-tube linac. The cavity design includes a ramped field. This calculation used only 130,000 mesh points. Mesh spacing near the drift-tube noses was 25% of the radius of curvature.

shown in Fig. 4, the computer had insufficient memory to hold the temporary data arrays, so the program wrote them to a 120-megabyte disk file.

Post Processors

Post processor codes include SFO, SF7, and VGAPLOT. SFO calculates parameters of interest to the accelerator designer, such as axial field integrals, transit-time factor, power dissipated on metal surfaces, shunt impedance, and frequency shifts from perturbations. SF7 interpolates electric and magnetic fields on lines, arcs, and rectangular grids. VGAPLOT is the SUPERFISH plotting code. Figures 1, 2, and 4 in this paper were drawn by VGAPLOT.

We added new code in SFO for multiple drift-tube linac cells. The code calculates transit-time integrals for all cells and power and frequency shifts for single or multiple stems and post couplers. SFO also now has more field-normalization options. For example, the designer can specify a value for either E_0 or E_0T. The code can calculate rf surface resistance R_s for normal or superconducting materials, or the user can enter a known value of R_s. SFO includes an option to write a plot file of the transit-time integrands.

The field-interpolation algorithm in SFO and SF7 is new. The old version selected points poorly, resulting in large point-to-point fluctuations along lines. The new algorithm first finds the mesh triangle containing physical point x_0,y_0 at which it will calculate fields. The area of this triangle is A_0. It fits a doubly quadratic polynomial to first and second nearest neighbors of x_0,y_0. The least-squares fit weights points by their physical proximity to x_0,y_0. Points within a range $\sqrt{A_0}$ have unity weight. Other mesh points x_i,y_i have weight W_i given by

$$W_i^2 = \frac{A_0}{(x_i - x_0)^2 + (y_i - y_0)^2}.$$

The general fitting polynomial is:

$$H(x,y) = P_1 + P_2(x-x_0) + P_3(y-y_0) + P_4(x-x_0)^2 + P_5(y-y_0)^2 + P_6(x-x_0)(y-y_0).$$

The interpolated magnetic field $H(x_0,y_0) = P_1$. Electric field components E_x and E_y (or E_z and E_r) are proportional to coefficients P_3 and P_2 respectively. The new code imposes boundary conditions by eliminating appropriate terms from the fitting function. For example, on the axis of a cavity with cylindrical symmetry, H and E_r must be zero. For this boundary $P_1 = P_2 = P_4 = 0$. On Dirichlet vertical boundaries and Neumann horizontal boundaries E_z is zero, and on Dirichlet horizontal boundaries and Neumann vertical boundaries E_r is zero. The code constrains the electric field to be normal to sloping Neumann boundaries.

New Programs

CFISH is a version of FISH that uses complex fields. It is useful for designing rf windows and calculating power losses in dielectric and magnetic materials. Post-processor codes SFO, SF7, and VGAPLOT are compatible with both real and complex versions of the code. VGAPLOT shows the real and imaginary field components in different colors. AUTOFISH is a combined version of programs AUTOMESH, LATTICE, FISH, and SFO. It saves time because it loads only one program and avoids several disk reads and writes.

DTLFISH, MDTFISH, CCLFISH, and RFQFISH are control codes for tuning DTL cells, multiple-drift-tube cavities, CCL cells, and RFQ cavities. An input file may contain starting parameters for many problems. These programs set up the cavity geometry and run the SUPERFISH codes repetitively, varying the cell geometry to tune each problem to within tolerance of a specified frequency. DTLFISH and CCLFISH assume symmetric half cells. DTLFISH adjusts either the tank diameter, drift-tube diameter, gap, or face angle. CCLFISH adjusts the cell diameter, septum thickness, gap, or cone angle; it also tunes buncher cavities and quasi-elliptical cavities. MDTFISH tunes DTL cavities by varying the tank diameter or the average drift-tube gap. Drift tubes have the same diameter and nose radii, but can have different face angles. Output includes the distribution of E_0. RFQFISH tunes an RFQ cross section by varying cavity volume or the shape of the vane tip.

Acknowledgments

We benefited from discussions with Robert Ryne of the Accelerator Code Group. We thank Richard K. Cooper for helpful suggestions and for several of the test cavities.

References

[1] Los Alamos Accelerator Code Group, "Reference Manual for the POISSON/SUPERFISH Group of Codes," Los Alamos National Laboratory document LA-UR-87-126.

[2] Lahey Computer Systems, Inc., 865 Tahoe Blvd., Incline Village, NV 89450.

[3] K. Halbach and R. F. Holsinger, "SUPERFISH — A Computer Program for Evaluation of RF Cavities with Cylindrical Symmetry," *Particle Accelerators* 7 (1976) 213–222.