The Validation of EM Modelling Codes
A User Viewpoint

Stanley J. Kubina and C. W. Trueman
Department of Electrical Engineering
Concordia University
7141 Sherbrooke Street West
Montreal, Canada H4B 1R6

Abstract

Computer codes for solving radiation and scattering problems have become powerful and widely available. The user of such a code must initially convince himself that his copy of the code executes correctly on his specific computer with his particular compiler. He must then build up expertise in constructing models for solution by the code which obey the restrictions imposed by the "modelling guidelines" accompanying the code in a user's manual. Users often want to solve "real" problems that do not conform to the problem geometry envisaged by the code writer. The user replaces the "real" structure with a computer model solvable by the code, and which obeys the modelling guidelines. The user must then carry out a "model validation" in which the computer model is tested against full-scale or scale model measurements. A successful "model validation" contributes to the user community's "experience base" and lends confidence to both the computer code and to the modelling process. Sometimes an unsuccessful attempt at modelling exposes a genuine limitation of the code. Then a new "modelling guideline" can be formulated to aid other users in avoiding the same difficulties.

This paper reviews code development to highlight the origin of "modelling guidelines", and how they are extended by the user community. From the user's point of view, the "experience base" is augmented whenever a successful "model validation" is carried out and reported. Several examples are presented of the difficulties that may be encountered in computer modelling, and how such difficulties lead to further "modelling guidelines" aimed at aiding others in solving similar problems.

Introduction

The last two decades have seen the development of powerful computer codes capable of analyzing a broad class of electromagnetic radiation and scattering problems. Such codes often rely on one of two basic techniques: integral-equation based moment-methods, and ray-optics based GTD methods. The problem of "code validation" is much the same in both cases: a data-base must be built of problems which have been analysed with the code resulting in currents or fields which are in agreement with independently-obtained answers. One source of "independent" data is the small class of problems which can be solved analytically, resulting in a closed-form solution, or in a series expansion converging quickly enough for actual evaluation. The best source of "validation" results is direct calibrated measurement. This paper discusses code validation and model validation from the user's point of view. Code development is reviewed as a source of restrictive "modelling guidelines" for the use of the code. When a code is first developed, a limited "code validation data base" is built, and is often designed to test the validity of the modelling guidelines. The user community exploits the code to solve "real" problems, and expends considerable effort in "model validation" to verify the results of specific models against measured data. This experience feeds back into the "validation data-base" which comes to represent the cumulative experience of the user community. Sometimes specific problems identified by users result in further "modelling guidelines".

Code Development

The top row in Fig. 1 summarizes the development of a computer code. A hypothetical problem is posed, which is tractable by analytic methods. For example, an antenna consisting of an interconnection of electrically-thin wires constitutes the "hypothetical problem" for the wire-antenna part of the LLL NEC[1] code. Step 1, "mathematical modelling", leads to a set of equations which embody a unique solution to the hypothetical problem. For the wire antenna case, an integral equation is obtained, accompanied by certain constraints on the current and the charge density at wire junctions and free ends. At this stage, certain assumptions must be made which lead to "modelling guidelines" restricting the range of problems solvable with the resulting computer code. For instance, the wires must be "electrically thin" because no circumferential current flow has been allowed. Wire junction assumptions restrict the range of wire radii which can be connected to a given junction. In some wire antenna codes the "mathematical modelling" of the source leads to further "guidelines" for modelling the source region of the antenna and ultimately to difficulties in computing antenna impedance.

Step 2 takes the mathematical statement and recasts it into a form suitable for solution by numerical computation. In the wire antenna example, the "moment method" is used to reduce the integral equation to a matrix equation. Thus "basis functions" are chosen for representing the current and "testing functions" are used to match the two sides of the integral equation in a least-square error sense. This process leads to further restrictions on the structure that can be modelled. At this stage restrictions reflect the approximations used to obtain a numerical solution of the integral equation, and form "artificial" considerations for the user. Thus the user must concern himself with subdividing the wires of his antenna into "segments", a consideration which is not part of the physical antenna being represented. "Modelling guidelines" restrict the segment length to 1/10 or 1/20 wavelength, restrict the segment length to diameter ratio, and concern the location of the centre of a segment which is adjacent to a wire junction, which must be kept outside the physical volume of other wires at the junction.

Step 3 of the code development process concerns the actual coding of the formulas obtained in step 2 into a computer language such as BASIC or FORTRAN. There is a considerable amount of "book-keeping" at this stage, for example, to impose junction constraints correctly, and so broad scope exists for committing coding errors and creating "bugs". Sometimes the effects of a "bug" are subtle and not readily seen in the results obtained.

Code Validation

Validation should be capable of examining the work of Steps 1, 2 and 3 individually but this is often not possible. Instead the computer program is run for certain specific structures conforming as closely as possible to the "hypothetical problem" and satisfying the assumptions made "en route", that is, obeying the modelling guidelines for the code. The resulting "solutions" are compared with analytic results and with measured data. Thus a wire antenna code might be tested for simple structures such as a dipole antenna, and top-loaded dipoles such as L-wires or tees. More complex structures conforming to the "hypothetical problem" would be examined: wire-grid versions of spheres for instance. "Bugs" leading to gross errors are readily found but many "bugs" are more subtle and elusive. Specific configurations for "code validation" are often chosen to test the "modelling guidelines" to determine how "hard" they are. Thus can a guideline on segment length be mildly violated and still lead to meaningful results? Self-consistency tests are performed. For example, as the number of "segments" is increased the solution is expected to "converge" to a unique answer. In this way an "experience-base" is built up which includes a data-base of specific examples, the "modelling guidelines", and a certain feel for the performance of the code giving insight into the design of suitable models for analysis by the code.

A good example of the failure of a computer code due to assumptions at Step 1 is that of the internal resonances of a closed structure modelled with a wire-antenna program. The integral equation does not distinguish between the "internal" and the "external" solution and the currents flowing on the wires are a superposition of those supporting the internal field and those radiating the external field. If the cavity is resonant at the frequency of operation then large currents supporting a large internal field are computed and swamp out the currents supporting the external field. Thus a "modelling guideline" is required cautioning the user against internal resonances.

The "code validation" process gives rise to a modest experience-base often written up in the documentation accompanying the program. The user runs his copy of the program both to verify that it generates similar results to those reported, and as a "confidence check" in his understanding of the required input and modelling techniques. Further problems can be encountered at this stage. Differences in the compiler and in the "environment" (large computer vs. microcomputer) can make modifications to the code necessary and sometimes extensive. Computer word-length differences and differences in the details of handling the arithmetic on dissimilar machines can pose formidable problems. Identifying and overcoming these difficulties is a basic "code validation" task that the user must deal with before he can exercise the code with confidence.

GTD-Based Codes

Ray-optical formulations invest a great deal of analytic effort in the "mathematical model" of certain canonical problems which together constitute the "hypothetical problem". Asymptotic expansion techniques give rise to diffraction coefficients with clearly specified restrictions, forming part of the "modelling guidelines" for the method. The "numerical implementation" step is complex for a GTD-based program because the code is responsible for the "accounting" required to trace rays from their source to the observer, through possibly multiple reflections and diffractions. It is difficult to be certain that the path is correctly computed without explicitly verifying it with computer graphics. A good example of this was encountered in the early development of the AAPG code.

The Antenna-to-Antenna Propagation with Graphics(AAPG) [2] computer code provides a responsive EMC analysis tool for evaluating the EMI margin for avionics systems. AAPG uses computer graphics to display the individual factors which comprise EMI margin, including transmitter and receiver frequency characteristics, and antenna-to-antenna coupling. In addition, the geodesic path between two antennas is displayed. The aircraft model is approximated by cylinders and cones for the fuselage and with planes for wings and stabilizers. Geodesic paths can include diffraction from wing edges. AAPG computes two paths around an aircraft cylindrical fuselage, one clockwise and the other counterclockwise. Both paths are examined and the path with the lowest coupling loss is used in evaluating EMI margin. Fig. 2 illustrates an erroneous path arrived at because of a "bug" in the path-finding algorithm. The graphics display makes the error strikingly obvious, yet without a graphics processor directly tied to the path generator, the user might never be aware of the erroneous path. The graphics display was designed to show the route of the coupling path over the simplified model of the aircraft used by the code, giving an appreciation for the differences to be expected in comparison with measurements on the actual aircraft.

Computer Modelling and Model Validation

The user wants to solve a specific problem of engineering interest, and assumes that the code he is using has been validated for structures adhering to the "modelling guidelines". But the user's "real" problem rarely conforms to the code's "hypothetical problem". A "computer modelling" step is necessary. Fig. 1 indicates that the real physical problem must be replaced with a "computer model" which the code is capable of analyzing. The user's task is then to demonstrate that the computer model is electrically-equivalent to the given structure by comparing computed results with measured data. This step can be termed "model validation". For example, the solid surface of a helicopter is "modelled" with the "wire-grid" of interconnecting wires shown in Fig. 3. The user must: (i) establish that the grid conforms to the helicopter's shape with no gross errors; (ii) make reasonable choices for the wire radii, based on murky guidelines; (iii) verify that the grid conforms to the various restrictions embodied in the code's "modelling guidelines"; and finally (iv) complete the "model validation" by demonstrating agreement with measured data. At the present state-of-the-art, completing steps (i), (ii) and (iii) does not ensure success at step (iv). Adherence to the "modelling guidelines" is thus a necessary but not a sufficient condition.

For example, the wire grid model of the helicopter shown in Fig. 3 satisfies steps (i) to (iii) above, but the agreement in Fig. 4(a) between the measured and the computed patterns at 8.1 MHz is unsatisfactory. Several alterations to the geometry of the model were tried without appreciable improvement in the agreement. The specific problem with the computer model was found by examining the current amplitude and phase of selected primary paths on the helicopter[3]. A resonant length path was identified which was "tuned" to a slightly different frequency in the wire grid computer model than in the measurement model. By adjusting the path length, the resonant frequencies were aligned, and much better agreement was obtained, as illustrated in Fig. 4(b). A "modelling guideline" can thus be formulated stating that the wire grid must be so constructed that its resonant frequencies agree with those of the "real" problem.

If measured data is not available for the adjustment of the wire grid, then the user must accept the notion that the resonant frequencies of the computer model may not align precisely with those of the actual structure. Radiation patterns, impedances and current distributions often change very rapidly with frequency near a resonant frequency. Hence it is risky to base conclusions on a "run" of a computer model at a single frequency. If the computer model's resonant frequency is not aligned with that of the actual structure, then, for example, the computer model's radiation patterns may be quite "wrong" at some specific frequencies. The user would do well to "run" the model over a range of frequencies to determine how rapidly the radiation patterns of his model are changing with frequency. If a resonant frequency is close by, then computed results must be interpreted with caution.

The display of the computed current distribution on the actual model plays a vital validation role as well since it shows whether the current distributions are realistic or spurious. Thus a large circulating current will sometimes be found on a mesh of a wire grid. A small adjustment in the grid geometry often eliminates the problem. Once the grid has been established for a given antenna, it can be used to study the performance of other antennas, provided that care is taken with the details of the grid in the source region of each antenna studied.

A second example of "model validation" and the subsequent exploitation of the model for a full-scale real world site, concerns the reradiation of commercial AM radio broadcast signals from high-voltage power lines. A model of the power line tower and the "skywires" interconnecting the towers was set up and tested against scale model measurements. An omnidirectional broadcast antenna was used in a 1:600 scale model to illuminate a power line model with 13 evenly-spaced towers. Scattering of the omni's signal from the power line causes lobes and minima in the "omnidirectional" pattern, and the ratio of the largest field value in the pattern to the smallest, or "max-to-min ratio", is a convenient perturbation parameter. By plotting max-to-min ratio against frequency, the power line is found to have a resonance near 860 kHz, as shown in Fig. 5. The computer model's resonant frequency is quite dependent upon the radius of the wire used to represent the tower. The best choice of radius was found to conform to a theoretical study of electromagnetic equivalence[4] as well as heuristic investigations[5,6,7]. This best choice is the mean between that for the equivalent perimeter and equivalent area for the cross-section of the tower structure being modelled. This criterion has proven to be a useful "modelling guideline" in itself.

This "modelling guideline" was applied to develop a computer model of a power line tower of quite different geometry. No scale model measurements were available for "model validation" purposes. A computer model of the "real" site shown in Fig. 6 was developed, encompassing a directional broadcast array, and two power line segments with a total of 25 towers. The model was analysed with the NEC code. Fig. 7(a) shows that the resulting tower currents agree reasonably with the currents measured at the tower bases on the actual power line, providing "model validation" against full-scale measured data. Subsequently the computer model was used to choose towers for isolation from the overhead "skywire" to open-circuit the resonant spans, thus achieve a large reduction in the tower currents and the reradiated field. Fig. 7(b) shows the computed tower currents against full-scale measured values, verifying that the expected reductions were in fact achieved on the actual site. Such comparisons with full-scale measured data lend considerable confidence to the modelling procedure.

"Model Validation" Feeds Back to "Code Validation"

Each time the "model validation" step is completed for a specific real problem, the result becomes part of the community's "code validation experience base" for that particular computer program. A "model validation" showing good agreement with measured results augments the community's confidence in the code and the modelling process.

Sometimes an unsuccessful model validation process cannot be attributed to a specific cause. Thus agreement between computed and measured impedance can be elusive. The exact conditions of the measurement must be well known[8] and details of the feed region may have to be accounted for in the computer model. A "modelling guideline"[8] cautions users that reliable impedance values for complex structures are difficult to obtain with a wire-grid code. An even greater contribution is sometimes made when a "model validation" fails. Thus attempts to model internally-resonant structures identified a serious limitation of moment-method codes, leading to further restrictions on the use of the code embodied in further "modelling guidelines". A mature computer code is thus backed with an extensive experience base built up by many users, consisting both of problems successfully solved, and cautionary "guidelines" arising from knottier problems.

Acknowledgments

The work described in this paper was supported by the Canadian Defence Research Establishment Ottawa, the Communications Research Centre and the Canadian Electrical Association.

References

[1] G.J. Burke, A.J. Poggio, J.C. Logan and J.W. Rockway, "NEC - Numerical Electromagnetics Code for Antennas and Scattering", 1979 IEEE International Symposium on Antennas and Propagation Digest, IEEE Publication No. 79CH1456-3AP, Seattle, Washington, June, 1979.

[2] S.J. Kubina, "Numerical Modelling Methods for Predicting Antenna Performance on Aircraft", AGARD Lecture Series No.131, The Performance of Antennas in their Operational Environment, October 1983.

[3] Y. Bahsoun, "Evaluation of an HF Helicopter Antenna - Measurements and Numerical Techniques", M. Eng. Thesis, Faculty of Engineering, Concordia University, Montreal, Que., Canada, March 1982.

[4] K.S.H. Lee, L. Marin and J.P. Castillo, "Limitations of wire-grid modelling of a closed surface", IEEE Transactions on Electromaagnetic Compatibility, Vol. EMC-18, No.3 pp.123-129, Aug. 1976.

[5] J. Moore and R. Pizer, "Moment Methods in Electromagnetics - Techniques and Applications", Research Studies Press Ltd.(Wiley), 1984.

[6] C.W. Trueman and S.J. Kubina, "Numerical computation of the Re-Radiation from Power Lines at MF Frequencies", IEEE Transactions on Broadcasting, Vol. BC-27, No.2, pp.39-45, June 1981.

[7] A.C. Ludwig, "Wire Grid Modelling of Surfaces", IEEE Transactions on Antennas and Propagation, Vol.AP-35, No.9, pp.1045-1048, Sept. 1987.

[8] S.J. Kubina and C.W. Trueman, "Selected Applications of Method of Moments Modelling", Proceedings of the NATO Advanced Study Institute, August 1987.

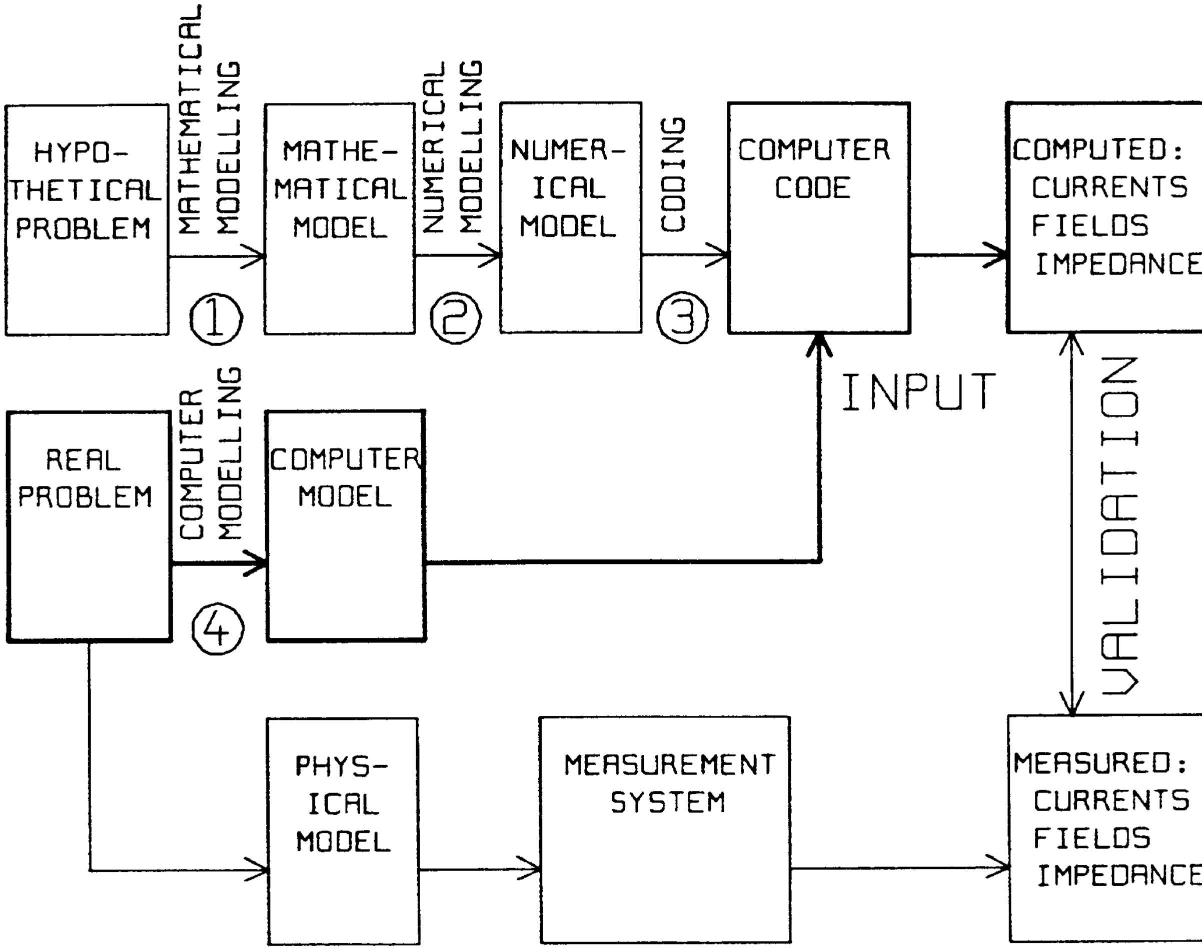

Figure 1 - The top row illustrates key steps in the development of an antenna analysis computer code. The user follows the path in the middle row. Both developers and users rely on scale model measurements for validation.

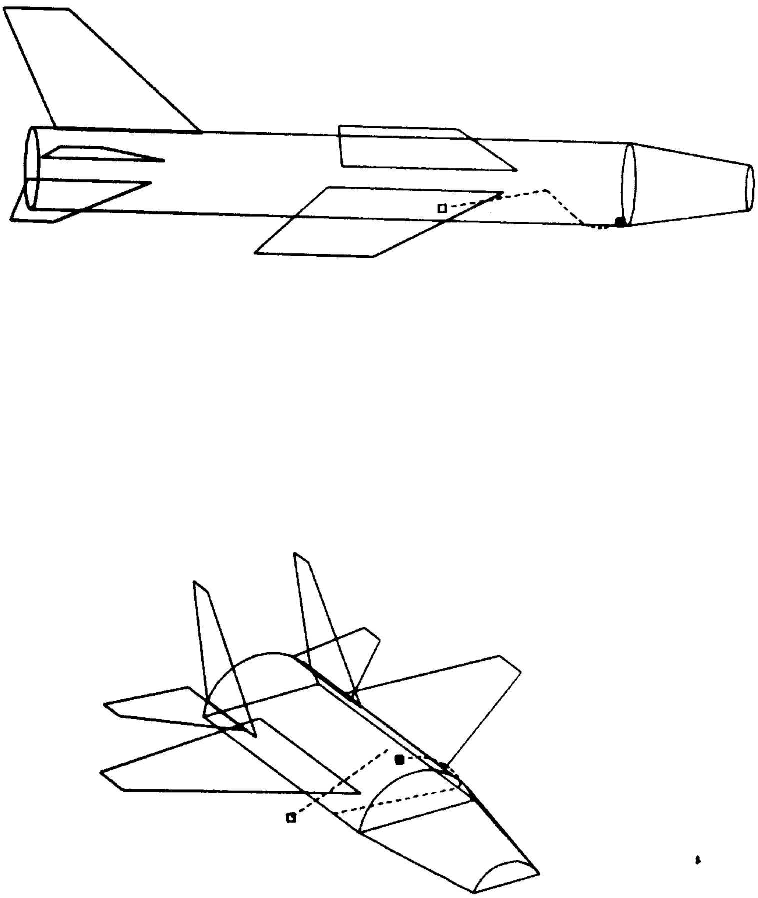

Figure 2 A computer graphics display of a geodesic path on a simplified aircraft highlights an error due to a "bug" in the path-finding algorithm. The filled square represents the transmit antenna, and the open square denotes the receive antenna.

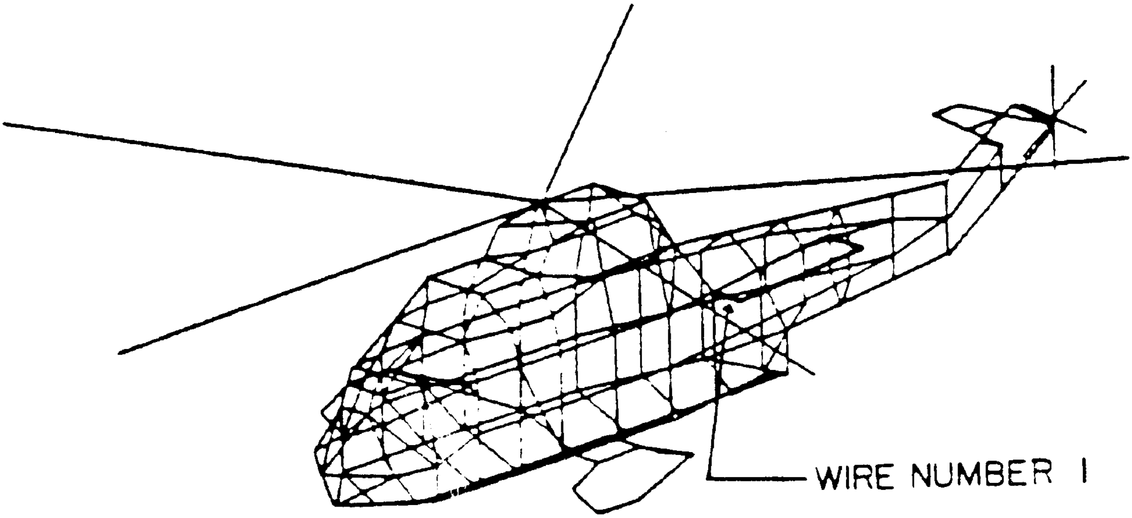

Figure 3 - The CHSS-2/Sea King helicopter has a continuous metallic skin, but must be replaced by a wire-grid computer model to conform to the "hypothetical problem" of a wire antenna analysis computer program.

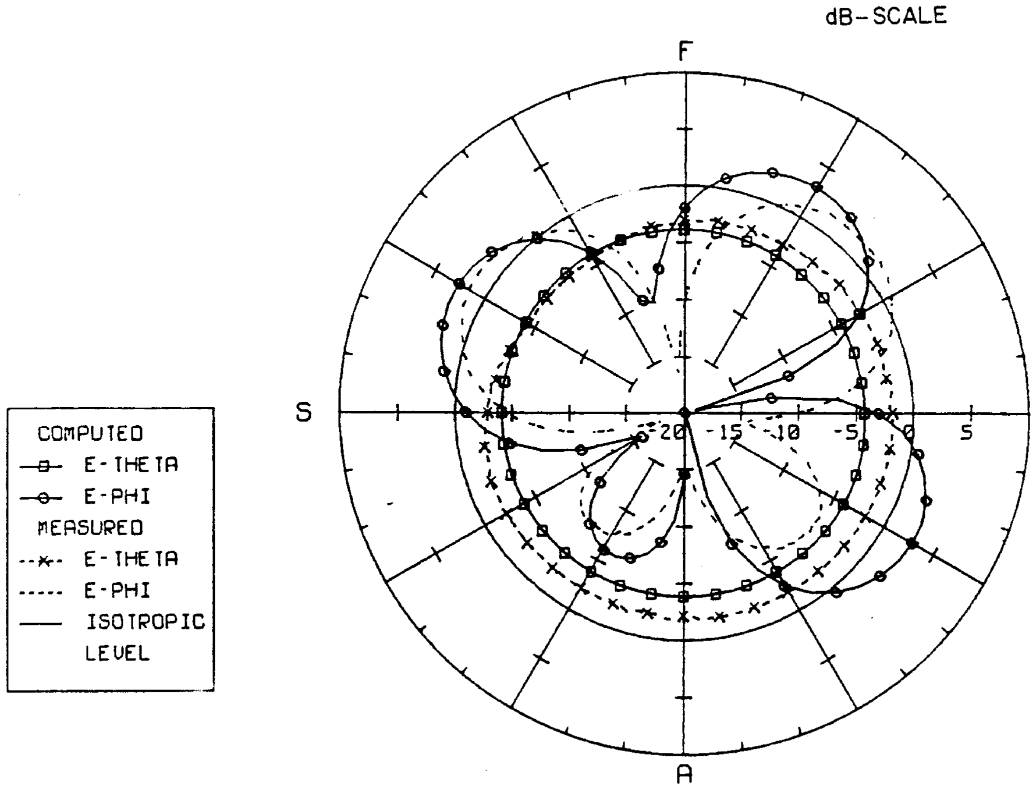

Figure 4(a) The "untuned" helicopter wire-grid model's patterns agree poorly with measured data near a resonant frequency.

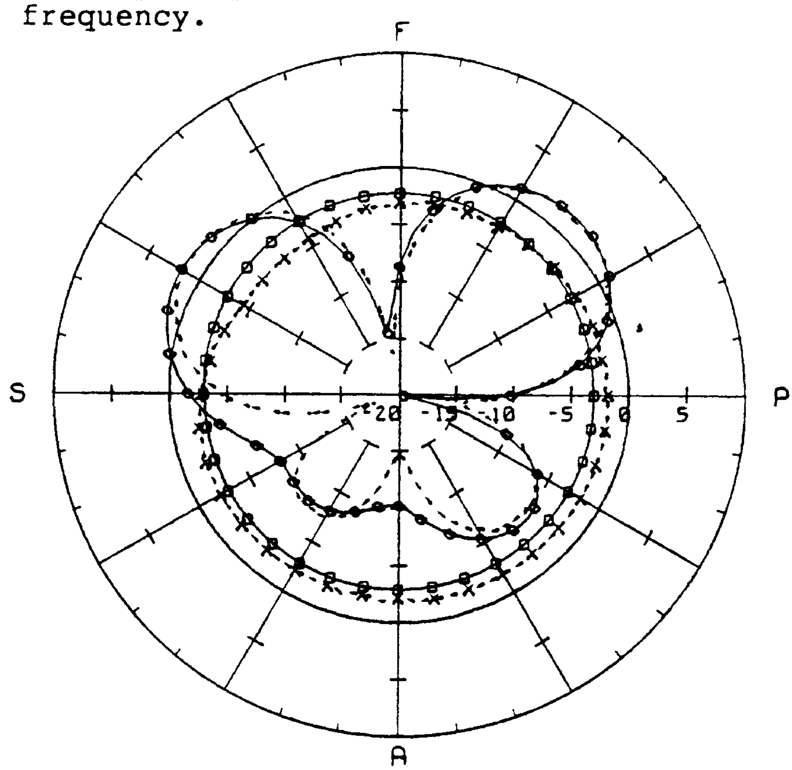

Figure 4(b) When the wire-gird is "tuned" by adjusting a path length involving the rotor blades, the model's patterns agree well with the measured data.

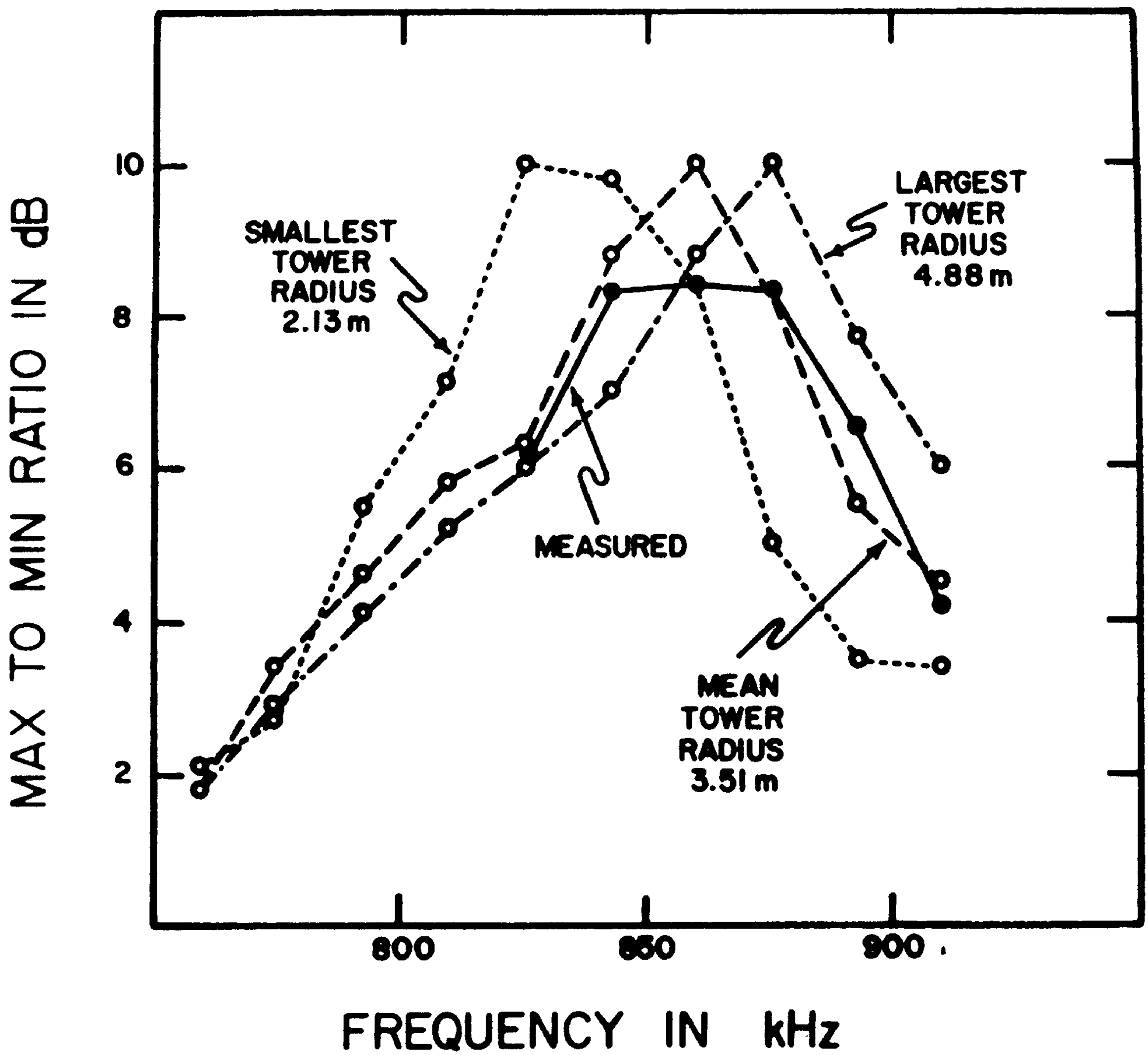

Figure 5 - A computer model of a power line must be adjusted to agree with the measured resonant frequency of a scale model.

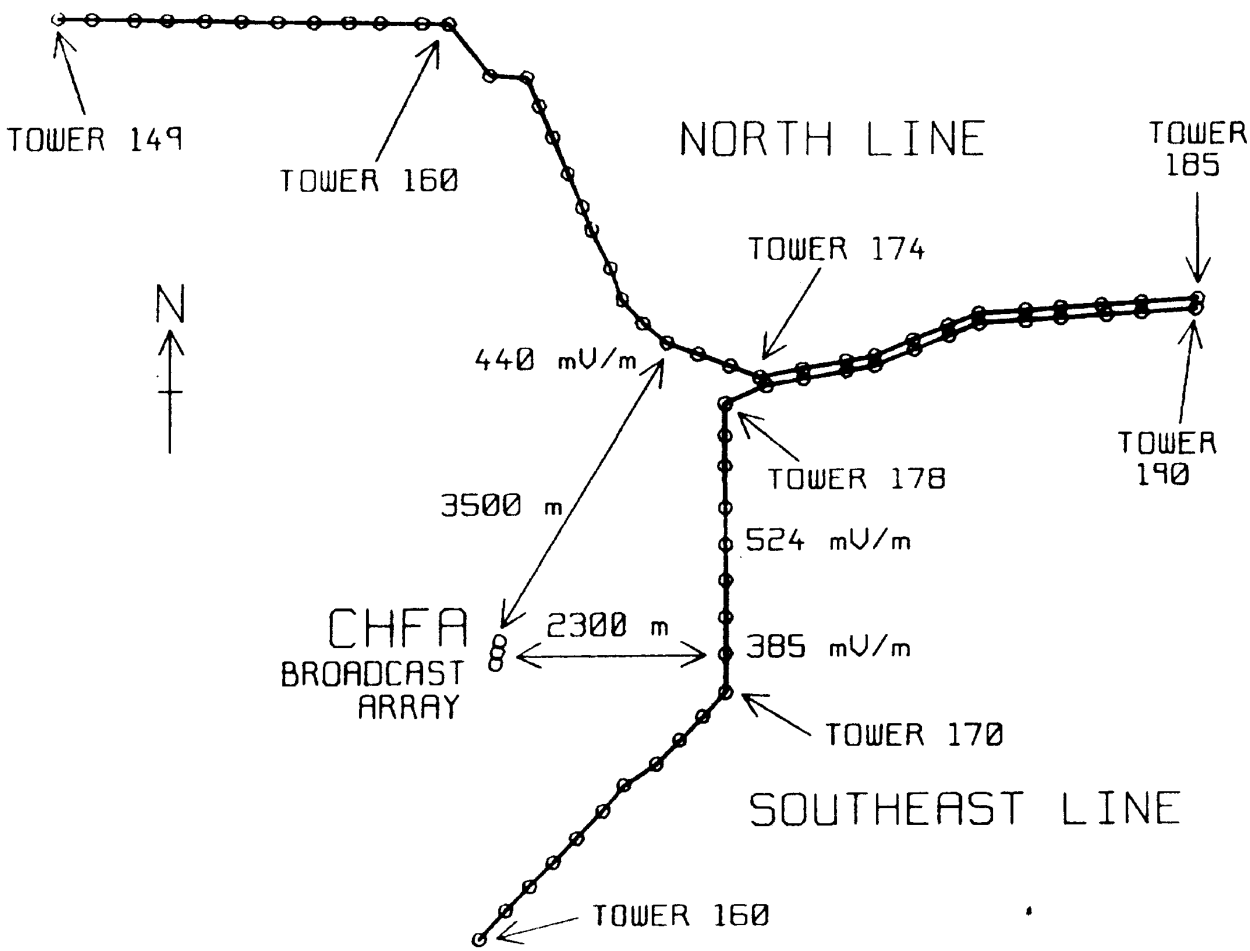

Figure 6 - The CHFA broadcast directional array has two power lines nearby. The "north" line lies in the main lobe of CHFA's pattern. The "southeast" line runs from the pattern minimum south of the antenna into the main lobe north and east of the antenna.

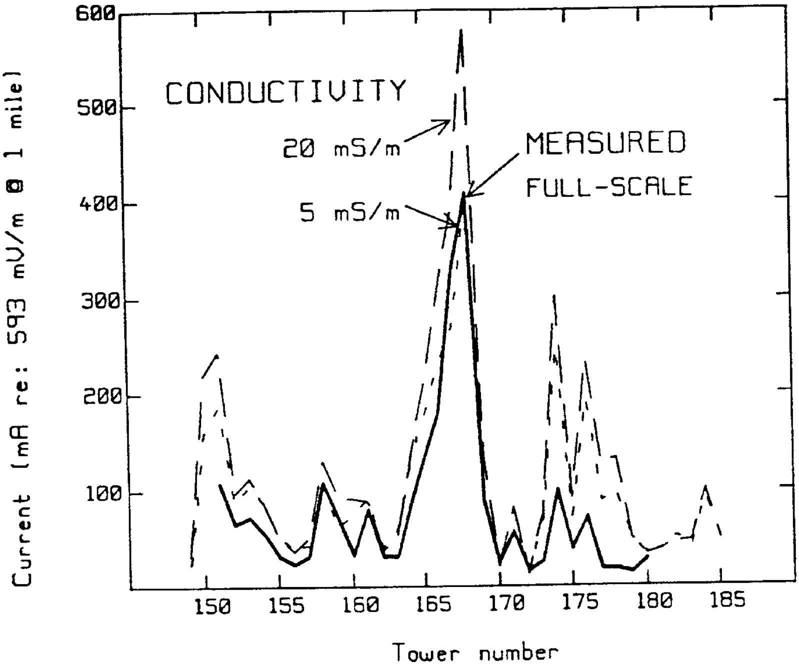

Figure 7(a) - The currents on the towers of a computer model of a power line agree well with the full-scale measured data when the appropriate value of ground conductivity is used.

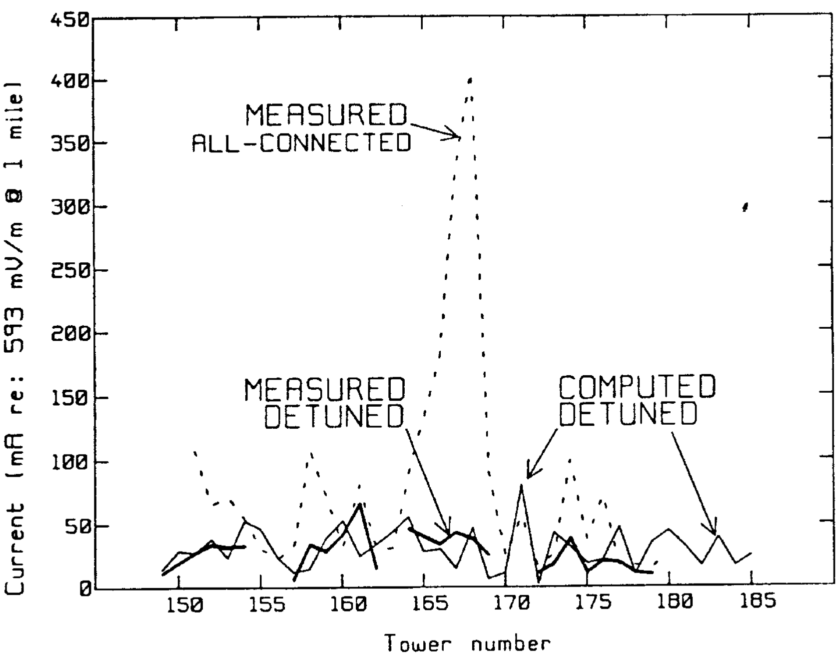

Figure 7(b) - When the computer model is used to design "detuning" to suppress the RF currents on the power line towers, the predicted reductions in current are found in a measurement of the actual currents flowing on the power line towers.