

A Simple Tool for the Design and Analysis of Multiple-Reflector Antennas in a Multi-Disciplinary Environment

Daniel S. Katz, Andrea Borgioli[†], Tom Cwik, Chuigang Fu,
William A. Imbriale, Vahraz Jamnejad, and Paul L. Springer

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, MS 168-522
Pasadena, CA 91109-8099
d.katz@jpl.nasa.gov

[†]University of California at Santa Barbara
Electrical and Computer Engineering Department, Engineering-1
Santa Barbara, CA 93106

I. Introduction

The process of designing and analyzing a multiple-reflector system has traditionally been time-intensive, requiring large amounts of both computational and human time. At many frequencies, a discrete approximation of the radiation integral¹ may be used to model the system. The code which implements this physical optics (PO) algorithm was developed at the Jet Propulsion Laboratory. It analyzes systems of antennas in pairs, and for each pair, the analysis can be computationally time-consuming. Additionally, the antennas must be described using a local coordinate system for each antenna, which makes it difficult to integrate the design into a multi-disciplinary framework in which there is traditionally one global coordinate system, even before considering deforming the antenna as prescribed by external structural and/or thermal factors. Finally, setting up the code to correctly analyze all the antenna pairs in the system can take a fair amount of time, and introduces possible human error.

The use of parallel computing to reduce the computational time required for the analysis of a given pair of antennas has been previously discussed². This paper focuses on the other problems mentioned above. It will present a methodology and examples of use of an automated tool that performs the analysis of a complete multiple-reflector system in an integrated multi-disciplinary environment (including CAD modeling, and structural and thermal analysis) at the click of a button. This tool, named MOD Tool (Millimeter-wave Optics Design Tool), has been designed and implemented as a distributed tool, with a client that runs almost identically on Unix, Mac, and Windows platforms, and a server that runs primarily on a Unix workstation and can interact with parallel supercomputers with simple instruction from the user interacting with the client.

II. Automation of multiple-antenna analysis

When using the PO code to analyze a system involving multiple antennas, multiple runs are needed. First, the currents on the first antenna are calculated, based on the feed horn, and the relative locations and orientations of the horn and the antenna. In the same run, these currents are used to calculate currents on the second antenna, again based on the relative locations and orientations of the two antennas. If this is the last antenna, this same code run can be used to calculate far-field patterns, using a coordinate system related to the second antenna. If there are more than two antennas, the currents on the second antenna are written to a file. Then, the code must be run again for the second and third antennas, which again must be described by their relative locations and orientations. This process continues until currents on the last antenna have been computed, at which time far-field patterns can be computed as was done for the simple dual-antenna system.

Using this code to model a system with n antennas therefore requires $n-1$ PO code runs, where for each run, relative translation between that pair of antennas and the Eulerian angles that describe the relative rotation between them are needed. In addition, the translation and rotation between the feed and the first antenna and the last antenna and the coordinate system used for the far-field calculations are needed. Because the calculation of the required Eulerian angles is non-intuitive, this, as well as the calculations of relative displacements introduces potential error. Additionally, either doing the $n-1$ runs or setting up a script to do them also adds potential error, which can lead to having to re-run portions of the calculation.

MOD Tool uses a compiled Fortran program on the server to automatically compute the $n+1$ sets of relative location and orientation data, and then to create the input files for the $n-1$ PO runs, as well as a single script file that can be executed to perform the complete system analysis. After the user has entered the geometry in a single global coordinate system, and tested it using a ray-tracing analysis package also interfaced into MOD Tool, a click of a button starts the complete PO analysis, with codes either running on the server, or on a user-specified parallel supercomputer. (The user is required to provide a login and a password on the supercomputer, but no other action is required.)

III. Integration into multi-disciplinary environment

When designing an antenna system that must meet various requirements (beamwidth, gain, pointing, etc.) while operating in a known thermal/structural environment, the analysis must take the environment into account when predicting performance. This implies that the analysis must include deformations of the antenna system caused by the thermal/structural environment. This requires that PO analysis be integrated with structural and thermal analyses, so that 1) the structural and thermal models use the identical initial geometry as designed in the PO code, and 2) the PO code can analyze the geometry after structural and thermal deformations. This is shown in Figure 1.

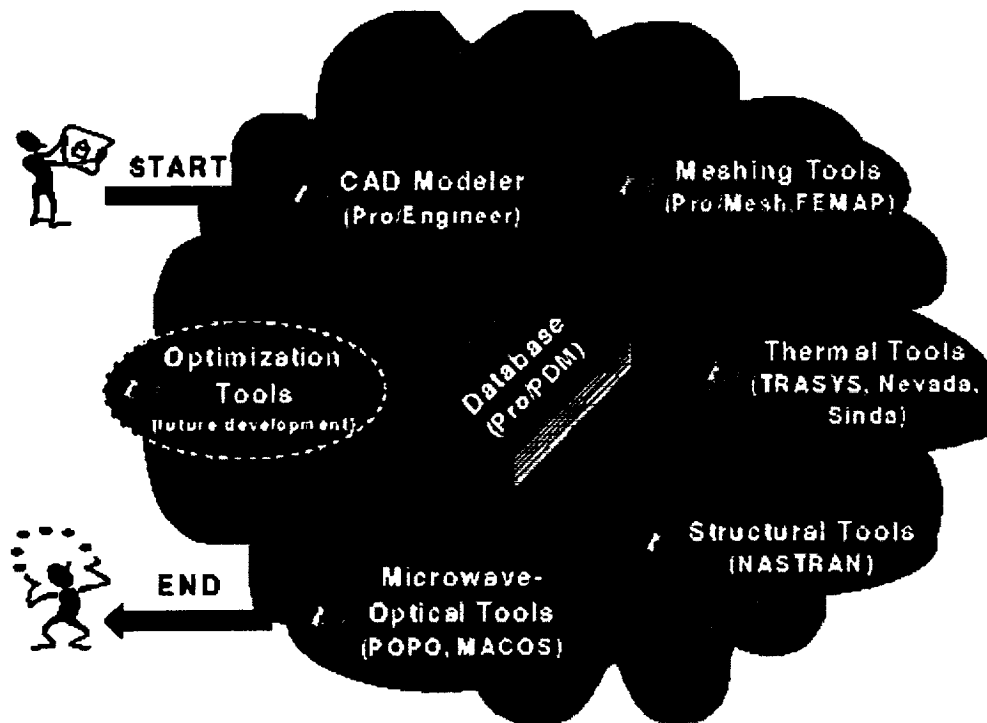


Figure 1. The MOD Tool environment.

An example of this is the Microwave Instrument for the Rosetta Orbiter (MIRO), currently being designed at JPL. This antenna system will be placed on a spacecraft that will follow a comet as it travels from 3.2 AU to 1.1 AU. As it moves closer to the Sun, MIRO will heat up and expand. The antenna designer can propose an initial antenna shape, that then can be condensed to a limited number of points in a global coordinate system and a small number of labels for these points. Geometric Optics analysis³ of this data can be performed to roughly verify the design. The data can be read into a CAD tool using macro commands written in that tool, in order to create a solid model of the antenna systems. The CAD designer and structural analyst can then add structure to support the antenna system. Thermal analyses can take this model (mesh) and use ephemeris information to predict the complete range of temperatures over various parts of MIRO. Structural analyses can take these temperatures and predict the complete range of shapes of the mirrors and support structures of MIRO (or deformations of the mesh at its nodes). Using MOD Tool, a pair of files describing a mesh and the displacements of each node of the mesh are submitted from the client to the server. If these files are chosen during the setup for a PO run, the server runs a pair of scripts to strip out the data for the antenna surfaces, and then runs Matlab to create a bi-polynomial surface that approximates the deformation to each surface. The coefficients of this polynomial are then used in the PO run. This allows MIRO to be designed so that it meets its requirements over the complete mission.

IV. Description of the distributed tool

MOD Tool was written using Tcl/Tk. This is a scripting language that includes a graphics toolkit which can be used for building fairly complicated graphical user interfaces (GUIs). The client GUI, with which the user directly interacts, is a single Tcl/Tk script that can be run through a Tcl/Tk interpreter. There are versions of the interpreter freely available on many platforms, including most common versions of Unix, Windows and Macintoshes. The MOD Tool server is an Expect script that runs on a Sun workstation. Expect is a scripting language written on top of Tcl/Tk, which allows automation of tasks that would normally be done interactively, such as transferring files and running compiled codes on one or many machines. Communication between the MOD Tool client and server, which are invisible to the user, are implemented using sockets.

V. Conclusions

A tool named MOD Tool was developed to meet the needs of multiple-reflector antenna systems designers. These needs, and the capabilities that have been build into MOD Tool, include:

- Simplification of the analysis of systems with more than two reflectors by automating the set-up of the PO code for each pair of reflectors, and automating the runs of the PO code.
- Incorporation of parallel supercomputing resources without the active involvement of the user.
- Output of antenna system design data in a single global coordinate system for the use of analysts in other disciplines (CAD, structural, thermal, etc.).
- Input of the deformation caused by structural and thermal environments into the PO analysis process.

Using these new capabilities, an experienced engineer can design a system and perform multiple analyses (including tolerancing studies) in hours instead of the days or weeks that were required before MOD Tool was developed. MOD Tool also allows automatic interfacing of structure and thermal data with the PO code in a matter of seconds, while this had previously been a laboriously manual process involving pieces of paper and inter-office mail.

VI. References

¹ W. A. Imbriale and R. E. Hodges, "The Linear-Phase Triangular Facet Approximation in Physical Optics Analysis of Reflector Antennas," *App. Comp. Electromag. Soc. J.*, v. 6, pp. 52-73, 1991.

² D. S. Katz and T. Cwik, "Large-Scale, Low-Cost Parallel Computers Applied to Reflector Antenna Analysis," *USNC/URSI Nat. Rd. Sc. Mtg.*, Atlanta., GA, 1998.

³ *Modeling and Analysis for Controlled Optical Systems User Manual*, version 2.4.1, Jet Prop. Lab., Cal. Inst. of Tech., April 1997.

