Parallel CARLOS-3D Code Development

J. M. Putnam
McDonnell Douglas Corporation
J. D. Kotulski
Sandia National Laboratories

Abstract
CARLOS-3D is a three-dimensional scattering code which was developed under the sponsorship of the Electromagnetic Code Consortium (EMCC), and is currently used by over 80 aerospace companies and government agencies. The code has been extensively validated and runs on both serial workstations and parallel super computers such as the Intel Paragon. CARLOS-3D is a three-dimensional surface integral equation scattering code based on a Galerkin method of moments formulation employing Rao-Wilton-Glisson roof-top basis for triangular faceted surfaces. Fully arbitrary 3D geometries composed of multiple conducting and homogeneous bulk dielectric materials can be modeled. Thin materials can be modeled as either bulk regions or using various approximate boundary conditions such as impedance surfaces or resistive and magnetically conducting sheets. Matrix symmetry and geometric symmetry are used extensively in the code in order to reduce the computational and memory requirements for large scatterers. A unique feature of the code is the Galerkin operator structure of the code which is based on an indexing scheme which relates basis functions to unknown numbers in the matrix equation. All of the integral equation formulations are implemented using this structure, and are based on six Galerkin matrix operators. This operator based structure is independent of the geometry representation and basis functions used.

In this presentation, we will describe some of the extensions to the CARLOS-3D code, and how the operator structure of the code facilitated these improvements. Body of revolution (BOR) and two-dimensional geometries were incorporated by simply including new input routines, and the appropriate Galerkin matrix operator routines. Some additional modifications were required in the combined field integral equation (CFIE) matrix generation routine due to the symmetric nature of the BOR and 2D operators. Quadrilateral patched surfaces with linear roof-top basis functions were also implemented in the same manner. Quadrilateral facets and triangular facets can be used in combination to more efficiently model geometries with both large smooth surfaces and surfaces with fine detail such as gaps and cracks. Since the parallel implementation in CARLOS-3D is at high level, these changes were independent of the computer platform being used. This approach minimizes code maintenance, while providing capabilities with little additional effort.

Results will be presented showing the performance and accuracy of the code for some large scattering problems. Comparisons between triangular faceted and quadrilateral faceted geometry representations will be shown for some complex scatterers.

This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.
DISCLAIMER

 Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
Introduction

CARLOS-3D is a general-purpose method of moments (MM) code [1] for computing the scattering from complex three-dimensional scatterers. It is based on a McDonnell Douglas Aerospace proprietary code CARLOS, which models antenna and scattering problems for 2D and 3D geometries. CARLOS-3D was developed under the sponsorship of the Electromagnetics Code Consortium (EMCC), and is available to qualified organizations through them, subject to US export control laws. The code uses the MM technique, with Galerkin testing to solve the Stratton-Chu surface integral equations for a user specified geometry. All of the surfaces describing the scatterer, consisting of conducting surfaces and boundaries between different dielectric regions are replaced with equivalent electric (J) and magnetic (M) currents. The code solves for these induced equivalent currents, which are then used to compute the scattered far-fields.

Code Description

Arbitrary 3D surfaces are modeled using flat triangular facets, with the electric and magnetic currents expanded in terms of the Rao, Wilton, Glisson roof-top functions [2]. A roof-top function spans the two facets forming each interior edge of a surface. At junction edges, which are formed by the intersection of two or more surfaces, half roof-top expansion functions are used to expand the currents. Current continuity across a junction is enforced by equating the unknowns associated with each of the half roof-top functions for the edge. Junction edges are determined automatically for surfaces which have common node points along a line of intersection between surfaces.

Complex geometries, which are composed of multiple conducting and bulk dielectric regions can be modeled. Various boundary conditions can be imposed separately on each of the surfaces comprising the target. For a given geometry, each dielectric region is given a number, and every surface which forms a boundary between different regions must be entered as a faceted surface. Infinitesimally-thin conducting, resistive, and impedance sheets must also be defined. The user must specify both the interior and exterior regions for each surface, along with the boundary condition to impose. Resistive and magnetically-conducting boundaries can either be embedded in a region, or be defined on the interface between different regions. Tapered resistive and impedance surfaces are modeled by specifying values for each facet forming the surface. Impedance (Leontovich) boundary conditions are modeled as an equivalent combination resistive/magnetically-conducting boundary.

The Stratton-Chu integral equations can be solved using several different formulations. For conducting surfaces, either the electric field integral equation (EFIE), the magnetic field integral equation (MFIE), or the combined field integral equation (CFIE) can be used. The coupling parameter in the CFIE formulation is specified separately on each surface, allowing the formulation to be used for geometries with both open and closed surfaces. For dielectric boundaries, the PMCHW formulation (after Poggio, Miller, Chu, Harrington, and Wu) is used. Galerkin testing, in conjunction with the form of the current expansions, results in a symmetric system of equations whenever the PMCHW/EFIE formulation is used. In addition, the formulations implemented for treated surfaces also
result in symmetric matrices which require only half of the matrix elements to be computed and stored.

Scattering from apertures, cavities, and gaps in a conducting surface can be modeled using an infinite ground plane option. Image theory is used to model both the source and the induced currents on all of the surfaces which are either conducting, dielectric or treated. The resulting system of equations is again symmetric.

**Code Structure**

The CARLOS-3D code has a flexible and modular structure which facilitates the incorporation of new features. The major components of the code are independent of the surface representation (i.e., flat facets, quadrilaterals, curved surfaces, etc.), and of the basis functions which are used to approximate the surface currents, permitting the code to be easily adapted to advanced basis functions. The section of code which generates the system matrix for an arbitrary geometry is written in terms of a generalized Galerkin matrix operator notation [3]. These operators result from testing either the integral operators in the Stratton-Chu equations, the equivalent currents directly, or the incident fields. The subroutines which assemble the system matrix and right-hand-side vector refer to these generic operators. The generic operators then reference routines which are specifically written for a given basis function type. Only the geometry input routine, and these specialized Galerkin operator routines depend upon the surface representation and basis functions used. Symmetry relations, which can be established for the Galerkin matrix operators, are used to efficiently fill the system matrix by eliminating redundant calculations.

A systematic approach [4] is used to generate the matrix equation, \( Z\mathbf{I} = \mathbf{V} \), for an arbitrary geometry with various boundary conditions imposed on the surfaces. This approach is based on a simple indexing scheme, which assigns an index number to each edge (roof-top or half roof-top function) defining the entire geometry. The index number for a particular edge specifies the location within the column vector, \( \mathbf{I} \), of either the \( J \) or \( M \) current coefficient associated with that edge. This indexing is performed in the geometry input routine, and is based on the boundary condition which is imposed on each surface. The boundary condition is used to define the equivalent currents which reside on the surface, and the relationship between the interior and the exterior current coefficients. Current continuity across junction edges connecting separate surfaces is enforced by equating the indices associated with the half roof-top functions for the edge. The matrix assembly routine is based on the index associated with each basis function (edge), and not on the explicit form of the basis function or the surface representation.

**Quadrilateral Patch Implementation**

A quadrilateral patch formulation [5] was incorporated into the parallel version of the code. In the quad-patch formulation, the currents on each patch are represented in terms of local parametric coordinates. For large smooth surfaces, the parametric coordinate directions are nearly orthogonal and the formulation has been shown to require fewer unknowns. This formulation employs a quadrilateral patched mesh to represent the
surface geometry and uses linear roof-top basis functions which span the pair of quad patches which form each interior edge of the grid. These basis function are edge based, and are implemented in manner similar to the Rao-Wilton-Glisson (RWG) roof-top functions. In this implementation, each quad patch is represented by four points on the surface of the scatterer, and a bi-linear surface description over the quad is assumed. The matrix elements can be computed in a manner analogous to the RWG basis functions using a combination of analytic and numerical procedures to compute the self and near-self terms.

The modifications to the parallel code were straightforward. The input routine was modified to accept either triangular or quad patches, and perform the unknown counting and indexing. Galerkin operator routines were written for the linear basis functions on quad patches using the triangular facet routines as templates. The high level operator routines were then modified to call the new operator routines for the quad patched surfaces.

**Hybrid Combinations**

Hybrid combinations of triangular and quad meshed surfaces have also been implemented into the code. The half roof-top functions for triangular and quadrilateral patches can be combined for edges which are formed by adjacent triangles and quads. Current continuity normal to a hybrid edge is maintained, and the representation is free of line charges. Note that the RWG half roof-top function is different than the half roof-top function which occurs for a degenerate quad patch where two points are identical. The RWG basis functions appear to be superior for this degenerate situation. A combination of triangular meshed and quad meshed surfaces is an efficient way to model geometries which have both surfaces with fine detail and large smooth surfaces. Galerkin operator routines were written to compute the interactions between triangular and quad surfaces.

**BOR and 2D Implementation**

The subroutines contained in the CICERO code [6] for both body of revolution (BOR) and two-dimensional (2D) geometries were incorporated into the parallel CARLOS-3D code. This was a straightforward way of implementing a parallel version of a BOR code by taking advantage of the operator structure and unknown indexing of CARLOS-3D. This permitted the modeling of some large missile shapes using an efficient formulation for BORs. In addition, some of the formulations and features of CARLOS-3D which were not contained in the CICERO code were available with little additional effort.

The input routines for BOR and 2D geometries were incorporated into the code with only minor modifications. The Galerkin operator routines were then integrated into the code with some additions added relating to the block matrix fill used in CARLOS-3D for the parallel implementation. The generic matrix assembly routines were then modified to call these new routines for BOR of 2D surfaces. Some additional changes were made to take advantage of the matrix symmetry which results from the BOR and 2D formulations due to the orthogonal components of the basis functions.
Results

The results for a number of selected problems will now be shown. All of the results were generated on the Intel Paragon at Sandia National Laboratories. The first problem to be shown is the scattering of the upper stages of a STRYPI missile at 1 GHz using the triangular patch representation and the BOR representation of the missile. The upper stages of this missile is 5.7m long with a circular cross section of radius 0.39m. The triangular patch model was generated using two symmetry planes resulting in a total of 62172(15798,15543,15543,15288) unknowns. The spatial resolution of this model was 309 facets/\lambda^2. The BOR representation of this scatterer consisted of 54.6 segments/\lambda which resulted in 1124 unknowns per Fourier mode. Thirty Fourier modes were used along with a 61 point Gaussian quadrature. The results of these two calculations are shown in Figure 1. and it is pointed out that the combined field formulation was used for both calculations.

The results clearly show excellent comparison between the two methods with a slight discrepancy at nose on (or tip aspect). The solution times for the two methods are presented in Table 1.
Table 1. Solution Times for BOR and Patch formulations (all times are in hours).

<table>
<thead>
<tr>
<th>Formulation</th>
<th># of Nodes</th>
<th>Matrix Fill</th>
<th>Matrix Solve</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOR</td>
<td>512</td>
<td>4.8e-02</td>
<td>2.7e-02</td>
<td>.1</td>
</tr>
<tr>
<td>Triangular Patch</td>
<td>768</td>
<td>1.0</td>
<td>.26</td>
<td>2.16</td>
</tr>
</tbody>
</table>

The second problem addresses the triangular versus the quad-patch formulation of the scatterer. The test scatterer for this case is the 1 meter almond at 2 GHz. The geometry and unknown information for this model is shown in Table 2.

Table 2. Geometry and unknown information for quad and triangular patch formulation.

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Nodes</th>
<th>Facets</th>
<th>Total Number of Unknowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad</td>
<td>1980</td>
<td>1872</td>
<td>14976</td>
</tr>
<tr>
<td>Triangle</td>
<td>1787</td>
<td>3359</td>
<td>20154</td>
</tr>
</tbody>
</table>

Note that the geometry information is actually for the quarter model of the almond so two planes of symmetry are used and the resulting total number of unknowns is approximately four times the quarter model unknowns. The monostatic rcs is shown in Figure 2 for theta polarization calculated using the EFIE.

Figure 2. RCS of the 1 meter almond at 2 GHz for the two formulations using the EFIE.
The above results clearly show excellent agreement between the two formulations. The solution times for these runs are collected in Table 3. Note that the problems were run on 64 nodes of the Intel Paragon.

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Matrix Fill</th>
<th>Matrix Solve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad</td>
<td>.16</td>
<td>.16</td>
</tr>
<tr>
<td>Triangle</td>
<td>.14</td>
<td>.29</td>
</tr>
</tbody>
</table>

Table 3. Solution times for the quad and triangular patch models (time in hr.).

Conclusions
The previous parallelization effort on CARLOS-3D resulted in a code that permitted new features to be added independent of the parallel implementation[7]. Two new features have been added namely the BOR formulation and the quad-patch version of the code. These features were easily added due to the original structure of the code. These features have been validated for a number of large test cases a few of which have been presented here.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
References


