THE AMOS (AZIMUTHAL MODE SIMULATOR) CODE*

J. F. DeFord, G. D. Craig, and R. McLeod
Lawrence Livermore National Laboratory
Livermore, CA 94550

Abstract

AMOS is a newly developed computer code that is finding application in the design of accelerator cavities. The program simulates the temporal evolution of electromagnetic fields in rotationally symmetric structures. An explicit Fourier series expansion is assumed in the azimuthal coordinate, which allows different multipole modes to be simulated independently. Nonuniform electrical properties (ϵ, μ, σ) are permitted, and both impedance and radiation boundary conditions are available. A geometric editor (DRAGON) allows for simple and rapid development/modification of models, and a direct CAD system–DRAGON data link is being used to facilitate the study of existing and developmental linac accelerator cells. A wakefield postprocessor (ANDY) allows the calculation of wake potentials, cavity impedances, etc., from AMOS simulation data. Code verification has consisted of comparisons with exact and approximate analytic results for simple geometries.

Introduction

A "wakefield," as it will be used herein, is the field left by a charged particle as it traverses a structure. Consider a particle moving down a smooth, perfectly conducting beampipe at the speed of light. Although this cannot be achieved exactly, it is nevertheless a reasonable assumption for high energy beams and will serve as an illustration. For such a particle the field lines that emanate from the charge are Lorentz-contracted in the laboratory reference frame to lie in the plane perpendicular to the direction of motion. In this situation it is evident that the source charge would exert no force on a test charge q that trails along behind it. If, however, the source charge passes a perturbation in the beampipe wall, such as a gap, then its field will be diffracted, leaving a wakefield that will interact with the test charge. This interaction can be quantified using the concept of the wake potential, which is simply the integral of the force on the test particle due to the field left by the source charge as the two particles travel the length of the beampipe, i.e.,

$$\vec{W}(s) = \frac{1}{Q'} \int_{-\infty}^{\infty} \vec{F}(z,t)dz|_{t=\frac{z+s}{c}} = -\frac{\partial w_0(s)}{\partial s}\hat{z} + w_1(s)r'\hat{x} + \cdots,$$

(1)

where $\vec{F} = \vec{E} + \vec{v} \times \vec{B}$ is the Lorentz force on a unit test charge due to the wakefields left by the source charge Q', s is the distance q is behind Q', r' is the radial position of Q', and the right hand side of Eq. (1) is a moment expansion of the wake potential in terms of the *characteristic* wake potentials w_n.

The wake potential is useful because it allows one to characterize the effect that structures have on charged particle beams [1,2]. Particle accelerators typically accelerate charge in a bunch consisting of many particles. The leading edge of the bunch will leave wakefields that interact with the charge further back in the bunch. This interaction can result in oscillations of the trailing edge, or tail, of the bunch, which can severely impact the performance of the accelerator. Knowledge of how the wake potential changes as modifications to the
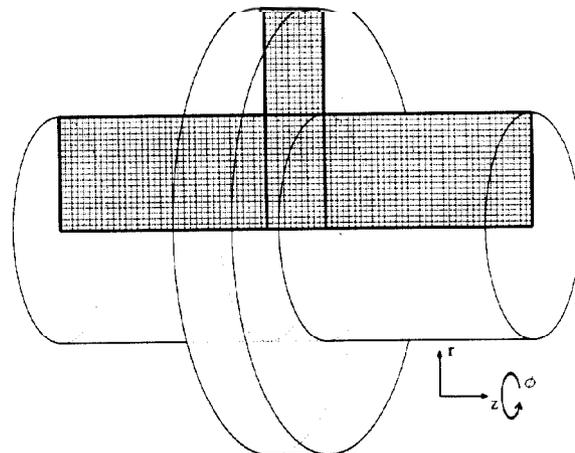
Fig. 1. AMOS models the electromagnetic fields in a cylindrical volume by distributing the field components on an $r - z$ grid and assuming a known ϕ-variation that depends on the mode being simulated.

structure are made may be used to optimize the structure for beam transport.

Unfortunately, only the simplest structures allow the wake potential to be calculated analytically. This fact provides the motivation for writing simulation software that will yield the wake potential, and related quantities such as cavity impedance, for any geometry of interest. From the standpoint of numerical simulation, the calculation of wake potentials is actually a two-part problem. The fields generated by the source charge must first be calculated, and then these fields must be manipulated in such a way as to produce the wake potential. The former task falls most naturally to an EM field simulator (AMOS); the latter to a wakefield postprocessor (ANDY). These two codes, and a geometric editor (DRAGON) which acts as a preprocessor, have all been through their first release in the last year and are described in the following sections.

AMOS: $2\frac{1}{2}$-D EM Simulator

AMOS is a finite-difference time-domain (FDTD) code that solves the time-dependent Maxwell equations in cylindrically symmetric volumes, or "bodies of revolution." Ordinarily, simulating the fields using the FDTD technique would require a 3-D finite-difference grid inside the volume. However, by decomposing the fields into a sum of modes, each with a given azimuthal, or ϕ, variation, the 3-D problem can be reduced to a 2-D one. In AMOS, the fields are distributed on a regular, 2-D, r-z grid (Fig. 1). The field equations are expressed in cylindrical coordinates (r-z-ϕ), and an explicit ϕ-dependence of the form

$$f_n = f'_n \left\{ \begin{array}{c} \cos n\phi \\ \sin n\phi \end{array} \right\}$$

(2)

is assumed, where f_n represents any of the six field components, i.e., E_ρ, E_z, E_ϕ, H_ρ, H_z, or H_ϕ. For a given n, AMOS yields the mode coefficients, f'_n, as functions of r, z, and time t, for the field components represented on the finite-difference grid. The actual field value as a function of r, z, ϕ, and t may be obtained simply by substituting the mode coefficient calculated by AMOS back into Eq. (2). Fields resulting from sources

with more complicated azimuthal variations may be treated as a sum over modes of the form in Eq. (2), and in these cases AMOS may be used to simulate each mode separately to obtain the total field.

To run AMOS, a complete problem definition is necessary, which begins with a description of the problem geometry. The area to be simulated, the simulation space, is composed of a set of regions (dark lines on grid in Fig. 1), each containing a portion of regular grid. The user defines the regions by listing the cell indices (i,j) of the lower left and upper right cells in each region. By sequentially storing the field components that are represented on each of the subgrids one obviates the need to allocate memory for grid cells outside the simulation area, while maintaining the indexing advantages of a logically regular grid.

With the simulation grid defined, sources for the EM fields may be specified. Both volume-distributed electric and magnetic source currents are supported. Sources in general have both a temporal variation and a spatial variation within the portion of the grid on which they are active. The user may either choose the temporal and spatial functions from a set of functions provided within AMOS, or they may be read in as a series of points from a file.

I/O operations are defined by specifying the particular type of diagnostic one would like to invoke, such as **field_value**, **power_flux**, **wakefield**, etc., and then supplying the necessary parameters for that diagnostic. The appropriate field quantities are either dumped to the screen, if that was requested, or dumped to the output data file for postprocessing by ANDY.

Postprocessing commands are also included in the AMOS input command file. These commands are passed along to the postprocessor ANDY in the form of a separate input command file that AMOS writes. The various commands that may be issued to the postprocessor reference the I/O requests by number, which allows the postprocessor to obtain information about exactly what fields were dumped without the user having to specify these fields explicitly. The nature and execution of the postprocessing commands will be described below.

DRAGON: Geometric Editor for Model Generation

The primary purpose of DRAGON is to simplify the definition of the simulation space. DRAGON is a screen-oriented, menu-driven geometric editor which allows the user to 'draw' the object he/she wants to simulate and then edit the model as required. Using a 'mouse' to move a cursor on the terminal screen [Fig. 2(a)] the user enters points that DRAGON connects with lines as the points are input, and these line segments collectively define the model. Other modes of input are also supported such as point specification via coordinates supplied interactively by the user, or read in from a file. The latter format allows for a direct interface between AMOS and CAD/CAM systems that are typically used in drafting engineering drawings of devices.

To edit the model, points may be added, deleted, or moved, and line segments, possibly containing many points, may be translated, rotated, or deleted. In addition, view manipulation features such as enlargement, panning, printing to a hardcopy device, etc. may be used as necessary.

DRAGON groups the line segments that make up a model into sets defining closed polygons, and each polygon in the model may have a different material type associated with it [Fig. 2(b)].

Finally, once the model has been defined and the material types specified, DRAGON prompts the user for the grid spacings in the radial (r) and axial (z) directions and produces the grid region definitions that AMOS needs to define the problem geometry [Fig. 2(c)]. This set of region definitions, together with the associated material type for each region, is written out in a data file and is used to complete the AMOS input command file.
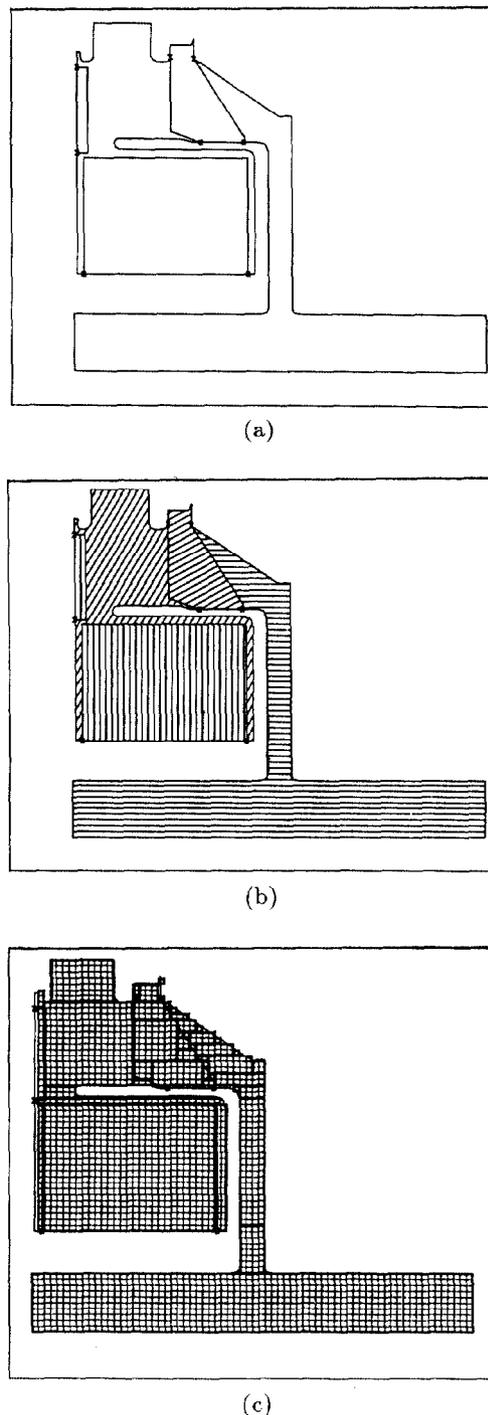

(a)

(b)

(c)

Fig. 2. Model development example using DRAGON: (a) Line segment definition of model boundaries. Nodes are entered via mouse, keyboard, or input file. The figure may be edited as necessary. (b) Material specification is made by placing the graphics cursor in each region and entering the material type for that region. (c) Finally, the finite-difference grid is generated to fill the simulation area. The grid is composed of rectangular subgrids, each with a particular material type. These subgrid definitions are output to the AMOS input file.

ANDY: Wakefield Postprocessor

The program ANDY takes the time-domain field data provided by AMOS and massages it to produce wake potentials, field plots, cavity impedances, etc. ANDY executes commands passed to it from AMOS, and it acts on data written in the simulation data file. Various input operations may be executed, such as reading in a particular field time history from the data file or reading in a block of time histories. Once the data has been read into ANDY it may be displayed, which at present simply produces a file that may be plotted, or a Fourier transform may be taken, or the data may also be manipulated algebraically to compute some quantity of interest.

The most powerful part of the postprocessor is the capability to calculate wake potentials and impedances from the time-domain data. This is done with the execution of a set of simple user commands in which the user specifies which I/O request is to be worked on and which component of wake potential or impedance he/she would like to calculate. File names appropriate to each request are generated automatically, depending on the overall simulation problem name. The rather complex set of operations required for these calculations takes place in a dynamically allocated scratch memory area within ANDY. Each macro contains information about the minimum and maximum storage requirements of its particular task, as a function of time history series length, and each macro requests blocks of memory from the pool as necessary. This memory-management strategy eliminates the need to redefine the vector lengths in ANDY, and the number of vectors in the scratch area, each time the program is run.

Example Application

AMOS simulation results have compared well with the approximate analytic theory of ATA (Advanced Test Accelerator) cell response developed by Briggs, et al. [3,4], and with TBCI on a more complicated geometry [5]. Other simpler tests have also been performed on closed cavities, and these also have shown excellent agreement with theory. In this section the authors describe the simulation of a geometrically accurate model of an ATA cell and compare these results with the simplified cell geometry studied in Refs. 3 and 4. The simplified cell geometry consists of a beampipe intersecting a radial transmission line that is terminated by an impedance boundary condition $Z = 2Z_o = 754\Omega$ at radius $r = 27$cm. The model of the ATA cell used in this comparison is shown in Fig. 2. Free-space impedance boundary conditions were used at the beampipe ports. Impedance boundary conditions were also used at entrances into ferrite media in the cell to approximate the rf absorption of the ferrite, and at the input power port at the top of the cell. AMOS was used to compute the time-domain fields that are generated by a charge bunch, with a Gaussian profile in the axial direction, moving across the gap at the speed of light. The time-domain E_z field at 12 points across the gap, and J_z at the edge of the gap, are passed on to ANDY, which uses these values to calculate the first moment of the transverse wake potential, i.e., the characteristic wake potential w_1 of Eq. (1). The details of this calculation, and in particular how one may compute the transverse component of the wake potential with knowledge of only E_z will not be discussed in this paper; the interested reader is referred to Ref. 6. ANDY uses the wake potential to compute the first moment of the transverse impedance, which is defined by the relation

$$Z_{\perp 1}(\omega) = \frac{-i}{c} \int_{-\infty}^{\infty} w_1(s) e^{\frac{i\omega s}{c}} ds. \tag{3}$$

$\Re(Z_{\perp 1})$ is plotted, as a function of frequency, in Fig. 3, along with the analytic results described in Refs. 3 and 4 for the
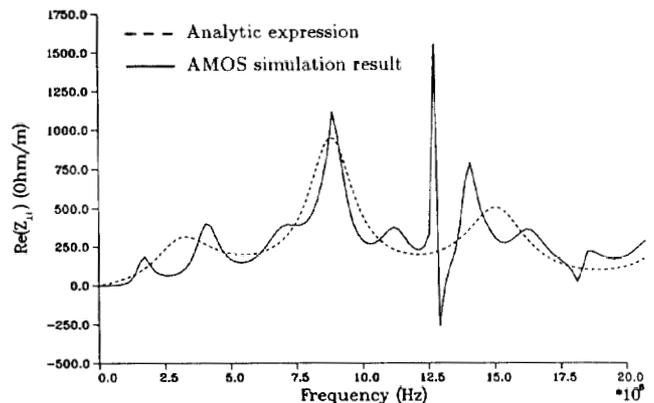

Fig. 3. Comparison of transverse impedance functions for the ATA cell, as generated by AMOS, with the analytic theory for the simplified cell geometry developed in Ref. 3.

simplified cell geometry. The existence and location of the important beam breakup (BBU) mode around 830 MHz is common to both models. As is to be expected, some of the fine structure of the actual cell response is missed by the analysis of a simplified geometry. Also, a large resonance is seen just below the beampipe cutoff that is not present in the earlier work. The ATA model contained $\approx 13\ 000$ finite difference cells, and the simulation covered 80 ns in $\approx 21\ 000$ iterations using about 45 min. of cpu time on a CRAY-1 computer.

Conclusions

The AMOS-ANDY-DRAGON code package has been developed in response to the simulation requirements of induction linac cell design and a broad range of other simulation problems. These requirements include the capability of modeling spatially varying electrical properties, various boundary conditions at ports into the cells and in the beampipe, and postprocessing software to calculate relevant quantities from the time-domain simulation data such as wake potentials and cavity impedances. AMOS has been designed to be easy to learn and use, and particular attention has been given to the model generation software and postprocessing capabilities. The programs have been tested satisfactorily against analytic theory and other simulation codes, and they are currently being used for induction-linac cavity design and other cavity problems.

References

1. P. B. Wilson, "Introduction to wakefields and wake potentials," *SLAC-PUB-4547, SLAC/AP-66*, Stanford Linear Accelerator Center, 1989.

2. K. L. F. Bane, "Wakefield effects in a linear collider," *Physics of Particle Accelerators, AIP Conf. Proc.*, **153**, 1986, p. 971.

3. R. J. Briggs, et al., "Theoretical and experimental investigation of the interaction impedances and Q values of the accelerating cells in the Advanced Test Accelerator," *Part. Accel.*, **18**, 1985, p. 41.

4. G. D. Craig and J. F. DeFord, "The connection between SLAC-DESY and LLNL wakefield formalisms," to be published as *UCID-21641*, Lawrence Livermore National Laboratory, 1989.

5. D. U. L. Yu, G. D. Craig, and J. F. DeFord, "Relativistic klystron wakefields," presented this conference.

6. T. Weiland, "Comment on wake field computation in time domain," *Nuclear Instruments and Methods*, **216**, 1983, pp. 31-34.