SIMULATION OPTIMIZATION SYSTEMS
Research Laboratory

PARAMETERIZATION OF ARBITRARY GEOMETRICAL STRUCTURES FOR AUTOMATED ELECTROMAGNETIC OPTIMIZATION

J.W. Bandler, R.M. Biernacki and S.H. Chen

SOS-98-30-R

September 1998

McMASTER UNIVERSITY
Hamilton, Canada L8S 4L7
Department of Electrical and Computer Engineering

PARAMETERIZATION OF ARBITRARY GEOMETRICAL STRUCTURES FOR AUTOMATED ELECTROMAGNETIC OPTIMIZATION

J.W. Bandler, R.M. Biernacki and S.H. Chen

SOS-98-30-R

September 1998

Parameterization of Arbitrary Geometrical Structures for Automated Electromagnetic Optimization

John W. Bandler *

Simulation Optimization Systems Research Laboratory and Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7

Radoslaw M. Biernacki and Shao Hua Chen

HP EEsof Division, Hewlett-Packard Company, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799

ABSTRACT

For the first time, this paper reveals and discusses the theoretical foundation of the Geometry Capture technique. Geometry Capture facilitates user-parameterization, through graphical means, of arbitrary 2D and 3D geometrical structures. This makes it possible to optimize the shape and dimensions of geometrical objects in an automated electromagnetic design process by adjusting the user-defined parameters subject to explicit numerical bounds and implicit geometrical constraints.

---

* Also affiliated with Bandler Corporation, P.O. Box 8083, Dundas, Ontario, Canada L9H 5E7.

All authors were formerly affiliated with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada L8S 4L7, and with Optimization Systems Associates Inc., Dundas, Ontario, Canada, before acquisition by HP EEsof.

I. INTRODUCTION

The most significant features of EM simulators include their unsurpassed accuracy, extended validity ranges, and the capability of handling fairly arbitrary geometrical structures. In order to take full advantage of these features the structures may need to be simulated in their entirety. This means that the microwave designer expects to be able to optimize increasingly more complex structures. This paper addresses the critical issue of parameterization of arbitrary geometrical structures [1-3] for the purpose of layout-based design, in particular automated EM optimization.

Automated EM optimization has raised a number of challenges. Some have already been successfully addressed [4-6] including geometrical interpolation and modeling, reconciling and exploiting the discrete nature of numerical EM solvers with the requirement of continuous variables and gradients by the optimizers, as well as parallel computation combined with efficient data base handling. Techniques, such as Space Mapping [7,8] play a pivotal role in effective utilization of EM design tools. Advances in computer hardware make automated EM optimization feasible, though still very CPU intensive. The potential and importance of EM-based optimal design have been fully endorsed by a panel session [9] and workshop [10] and we expect widespread use of this approach in the future. The number of reported applications is growing rapidly (e.g., [11,12]).

As the optimization process proceeds, revised structures must be automatically generated. Moreover, each such structure must be physically meaningful and should follow the designer's intention w.r.t. allowable modifications and possible limits. It is, therefore, of utmost importance to leave the parameterization process to the user. In our earlier work (*Empipe Version 1.0, 1992*) [13] we created a library of predefined elements (lines, junctions, bends, gaps, etc.), that were already parameterized and ready for optimization. The applicability of that approach is, however, limited to structures that are decomposable into the available library elements. Even a comprehensive library would not satisfy all microwave designers, simply because of their creativity in devising new structures. Moreover, the library approach inherently omits possible proximity couplings between the elements since they are individually simulated by an EM solver and connected by a circuit-level simulator.

To provide a tool for parameterizing arbitrary structures, we created the user-friendly Geometry Capture™ technique (*Empipe Version 2.0, 1994*) [2,13]. Here, we examine theoretical and implementational concepts and present the mathematical foundation of that technique.

II. THEORY

Mathematical Description of Geometrical Objects

Every structure to be simulated by an EM solver consists of a number of 2D or 3D objects. Each object must be uniquely defined by its attributes and a finite ordered set of numerical values. The attributes determine the class of objects into which a particular object falls (for example, a polygon or a polytope) as well as how the numerical values are interpreted by the specific EM solver. These numerical values typically represent absolute coordinates of points which form a defining set for the object. For example, a specific polygon can be defined by a sequence of its vertices, with the assumption that each pair of consecutive vertices determines an edge, i.e., they are connected by a line segment (with the last vertex connected to the first one). In contrast, a purely mathematical description of objects such as defining its boundary by a (possibly implicit) function may not be quite practical, could limit available shapes, etc.

It is possible for some of the numerical values defining an object to represent parameters such as the length or width of a rectangle, or the angle or radius of a radial stub. Such parameters are of direct interest to the designer. If all numerical values represented such parameters and if they were readily available then there would be no need for parameterization. However, some of the values (if not all) must represent absolute geometrical coordinates for the simple reason of indicating relative placement of an individual object w.r.t. to all other objects, as required to handle arbitrary structures. Therefore, in the following discussion we concentrate on those absolute coordinates only, assuming for simplicity that they represent vertices.

Consider an ordered set of vertices of an object as described by

$$\boldsymbol{x}_{v1},\ \boldsymbol{x}_{v2},\ \ldots,\ \boldsymbol{x}_{vm} \tag{1}$$

where m is the total number of vertices and each $\boldsymbol{x}_{vi}$ is the vector of vertex coordinates. Depending

on the object, x_{vi} is either a two- or three-dimensional vector. All the vertices can be conveniently represented by a single vector

$$x = [\, x_{v1}{}^T \quad x_{v2}{}^T \ldots \; x_{vm}{}^T \,]^T \tag{2}$$

which combines all the coordinates in an ordered manner. The space of all vectors x will be denoted by X (it can be either R^{2m} or R^{3m}).

Implicitly Constrained Coordinate System

Considering x in (2) as a vector of unconstrained optimization variables can easily lead to unacceptable results. This is illustrated by Fig. 1. Starting from the object shown in Fig. 1(a) it is possible for the optimizer to suggest the values leading to the situation depicted in Fig. 1(b). This may pose serious difficulties - the best an EM solver can do is to employ a sophisticated rule checker and to dismiss the suggested values. Such a rule checker will not be normally available to the optimizer.

In order to impose constraints on the movement of the vertices we consider a function T mapping certain designable or optimizable parameters ϕ into X as

$$x = T(\phi) \tag{3}$$

and assume that the parameters are allowed to vary within an orthotope specified by

$$\phi_{i\,\min} \le \phi_i \le \phi_{i\,\max}, \quad i = 1, 2, \ldots, n \tag{4}$$

There will normally be very few parameters ϕ as compared with the number of vertices ($n \ll m$). The process of parameterizing an object consists of selecting the parameters ϕ, defining and determining the function T, and establishing the constraints (4). Finally, discretization of the parameters ϕ needs to be considered.

Defining Structure Parameters and Object Evolution

As already mentioned, defining the parameters ϕ should be left to the designer who knows best what changes to the object are desired and allowable. Although this process is quite intuitive, a few rules should be followed. First, there should be as few parameters as possible. Secondly, the

parameters must be consistent. In other words, if independently changed, they must not contradict each other. For example, attempting to define all partial lengths as well as the total length as independent variables is incorrect. A clear understanding of how the object evolves when a parameter is varied is crucial. The limits to be specified by (4) are particularly important in preserving the physical meaning of the object. *It is worth emphasizing that the parameter values, as seen by the optimizer, are intermediate to the process of generating actual layouts.* Therefore, parameter transformations such as scaling or normalization can be used to link those optimizable parameters with the actual layout design parameters.

Object evolution and defining the parameters are illustrated by Fig. 2. Assuming that the location of the left edge is fixed, the evolution of the object can be described by just one parameter in all cases. However, its definition and impact on the location of the vertices will be different. One may select the overall length, the distance between the slit and the left edge, or the distance between the slit and the right edge in the case of Fig. 2(b), (c), or (d), respectively. The importance of the limits (4) is particularly evident in the case of Fig. 2(c): the distance between the slit and the left edge must not exceed the overall length. Another example is shown in Fig. 3 where one parameter controls the length of the right edge in a symmetric manner.

Defining the Mapping

The mapping (3) will be defined w.r.t. to the starting, or nominal, object $\boldsymbol{x}^0$

$$\boldsymbol{x}^0 = \boldsymbol{T}(\boldsymbol{\phi}^0) \tag{5}$$

where $\boldsymbol{\phi}^0$ represents the nominal values of the parameters $\boldsymbol{\phi}$. In other words, we consider the following form of $\boldsymbol{T}$

$$\boldsymbol{T}(\boldsymbol{\phi}) = \boldsymbol{T}(\boldsymbol{\phi}^0) + \boldsymbol{F}(\boldsymbol{\phi}-\boldsymbol{\phi}^0) \tag{6}$$

As long as the vectors $\boldsymbol{x}^0$ and $\boldsymbol{\phi}^0$ are known (specified by the designer), only the function $\boldsymbol{F}$ in (6) needs to be identified. The movement of individual vertices w.r.t. the nominal object is then

$$x_{vi} = x_{vi}^0 + f_i(\boldsymbol{\phi}-\boldsymbol{\phi}^0) \tag{7}$$

where f_i, i = 1, 2, ..., m, are the subvectors of $\boldsymbol{F}$ such that

$$\boldsymbol{F} = [\, f_1^T \quad f_2^T \ldots \; f_m^T \,]^T \tag{8}$$

A principal assumption we make about the mapping is that the functions f_i^T are additive w.r.t. the contributions due to incremental changes in individual parameters. This is expressed mathematically as

$$f_i(\boldsymbol{\phi}-\boldsymbol{\phi}^0) = \sum f_{ij}(\phi_j-\phi_j^0) \tag{9}$$

The case of two parameters is illustrated in Fig. 4. Under this assumption, defining the mapping (3) can be carried out by identifying the functions f_{ij} in (9). Each such function determines the trajectory of the movement of a specific vertex due to a change in one parameter alone. An example of such a trajectory is shown in Fig. 5. An important consequence of (9) is that $\boldsymbol{F}$ in (6) can be expressed as

$$\boldsymbol{F}(\boldsymbol{\phi}-\boldsymbol{\phi}^0) = \sum \boldsymbol{F}_j(\phi_j-\phi_j^0) \tag{10}$$

where each term on the RHS indicates the evolution of the whole object due to a change in one parameter alone. This means that the process can be split into steps in which the user characterizes the evolution of the whole structure in response to changes in one parameter at a time.

Specific Forms of the Mapping

Mathematically speaking, each function f_{ij} in (9) is a parametric description of the ith vertex trajectory corresponding to changes in ϕ_j when all other parameters are kept at their nominal values. Theoretically, any form of f_{ij} can be handled. From the implementational point of view, however, we may want to limit the available forms to some predefined functions. One also needs to make a decision whether an analytical (through some coefficients) or numerical (through some function values) description is to be entered. An important consideration in limiting the available forms is to understand how much arbitrariness in the overall structure evolution we may loose. Finally, it is strongly desirable to keep the functions as simple as possible.

The latter consideration turns our attention to linear functions f_{ij} of the form

$$f_{ij}(\phi_j-\phi_j^0) = (\phi_j-\phi_j^0)\alpha_i \tag{11}$$

where $\boldsymbol{\alpha}_i$ is a fixed two- or three-dimensional vector. Correspondingly, the ith vertex trajectory is a straight line. To use this form we need either to enter $\boldsymbol{\alpha}_i$ directly, or to specify another point on the line in addition to the already defined nominal point.

The next choice could be a polynomial form

$$f_{ij}(\phi_j-\phi_j^0) = \sum (\phi_j-\phi_j^0)^k \boldsymbol{\alpha}_{ik} \tag{12}$$

where $\boldsymbol{\alpha}_{ik}$, $k = 1, 2, ..., N$, are fixed two- or three-dimensional vectors, and N is the polynomial degree. To use this form we need either to provide directly N vectors $\boldsymbol{\alpha}_{ik}$, or to specify another N points on the trajectory (in addition to the already defined nominal point). If the defining data is entered using those additional points then two (or three) corresponding systems of N simultaneous linear equations need to be solved to determine the vectors $\boldsymbol{\alpha}_{ik}$.

Trigonometric functions could be useful in defining circular trajectories. For example, if $\phi_j = \theta$ is the angle of a radial planar stub then the movement of the ith vertex of this stub can be described by

$$f_{ij}(\theta-\theta^0) = \boldsymbol{\alpha}_i - \boldsymbol{\alpha}_i cos(\theta-\theta^0) + \boldsymbol{\alpha}_i^{\perp} sin(\theta-\theta^0) \tag{13}$$

where $\boldsymbol{\alpha}_i = [\alpha_{i1} \quad \alpha_{i2}]^T$ is a fixed two-dimensional vector, and $\boldsymbol{\alpha}_i^{\perp} = [\alpha_{i2} \quad -\alpha_{i1}]^T$. Here, either $\boldsymbol{\alpha}_i$ is entered directly, or two additional non-colinear (w.r.t. the nominal point) points on the trajectory need to be specified.

From the foregoing discussion one can have an impression that we need a variety of specific forms of the functions f_{ij} in order to accommodate most of the practical situations. Even so, such forms do not have to be directly included as an integral part of capturing the object evolution provided that an expression processor is available to pre-process the object parameters $\boldsymbol{\phi}$. This can be illustrated by an alternative approach to (13). Let θ be the parameter of interest to the designer. Instead of defining (13) directly we may introduce two intermediate variables, say ϕ_1 and ϕ_2. Using just the linear function form (11) we can then define two functions $\mathbf{f}_{i1}$ and $\mathbf{f}_{i2}$ simply as

$$\mathbf{f}_{i1}(\phi_1-\phi_1^0) = (\phi_1-\phi_1^0)\,[1 \quad 0]^T \tag{14}$$

$$\mathbf{f}_{i2}(\phi_2-\phi_2^0) = (\phi_2-\phi_2^0)\,[0 \quad 1]^T \tag{15}$$

To enforce the vertex trajectory the variables ϕ_1 and ϕ_2 are not permitted to be independent.

Rather, they are both controlled by θ through pre-processing as

$$\phi_1 = \phi_1^0 + \alpha_{i1} - \alpha_{i1}cos(\theta-\theta^0) + \alpha_{i2}sin(\theta-\theta^0) \quad (16)$$

$$\phi_2 = \phi_2^0 + \alpha_{i2} - \alpha_{i2}cos(\theta-\theta^0) - \alpha_{i1}sin(\theta-\theta^0) \quad (17)$$

After adding both functions $\mathbf{f}_{i1}$ and $\mathbf{f}_{i2}$ according to (9) the effect of θ on the vertex movement is identical to that described by (13).

We believe that the linear mapping (11), occasionally supported by an expression pre-processor, should be adequate for the majority of arbitrary structures that microwave designers may want to consider. Note, that all the cases of Figs. 2 and 3 can be handled by (11) without any need for pre-processing. Further examples are given in Section III.

Discretization of Structure Parameters

Discretization of structure parameters may or may not be enforced by the EM simulator. In any case it is advantageous to use it or at least to have the option to use it. There exist techniques facilitating significant improvement of efficiency when the parameters are discretized. These techniques include the utilization of a data base of already simulated structures in conjunction with efficient interpolation and modeling [4-6]. The benefits of these techniques include efficient gradient evaluation, handling of tolerances, efficient model evaluation in Monte Carlo analysis and yield-driven design.

In cases when parameter discretization is not enforced by the EM simulator we have total flexibility in setting the discretization grid. The only factor to consider is the trade-off between the accuracy of interpolation and the CPU time saving. Actual EM simulations are invoked on an as needed basis at the on-grid parameter values only. Their results are stored in a data base for future reference. For off-grid values the structure responses are interpolated from the results of the neighbouring on-grid simulations.

Parameter discretization may be enforced by the EM simulator if it is a fixed grid solver. If this is the case, all the (user-defined) parameters must be discretized in such a manner that for on-grid parameter values the mapped vertices are also on the grid. More precisely, we need to deal

with two grids. The first one is the already mentioned *parameter grid* created in the space of the parameters ϕ by their discretization. The second one, due to meshing, is imposed on the structure by the EM solver. We call it the *layout grid*.

We concentrate our discussion on the simplest and prevailing case of a rectangular uniform layout grid, which is assumed to be fixed when the structure evolves. The vertices (1) of any object have to be on this grid which can be mathematically expressed as

$$\boldsymbol{x}_{vi} = [\, k_{i1}\,\Delta x_1 \quad k_{i2}\,\Delta x_2 \quad k_{i3}\,\Delta x_3 \,]^T \tag{18}$$

where k_{i1}, k_{i2} and k_{i3} are some positive integers, and Δx_1, Δx_2 and Δx_3 are the grid sizes in the respective dimensions. When an attempt is made to supply a vertex $\boldsymbol{x}_{vi}$ which does not satisfy (18) then it will normally be automatically snapped to the nearest grid point.

It is clear from (3) that when the optimizer changes the parameters $\boldsymbol{\phi}$ then the vertices $\boldsymbol{x}_{vi}$ are likely to be off the layout grid. Thus, snapping may occur for various settings of the parameters $\boldsymbol{\phi}$ making the structure responses severely discontinuous (constant over certain regions with sudden jumps between the regions). Here, interpolation has to be applied to circumvent this problem.

Since the optimizer deals with the parameters $\boldsymbol{\phi}$ only, both grids need to be consistent, namely for on-grid parameters $\boldsymbol{\phi}$ the corresponding vertices, determined according to (3), have to be on the layout grid. This is simply required because no snapping error should be allowed in EM simulations for on-grid values of $\boldsymbol{\phi}$. To reconcile both grids is not trivial and, in general, may not be feasible. This is perhaps theoretically the most limiting aspect of the whole process of structure parameterization.

Grid Reconciliation: General Case

Fortunately, subject to some limitations, reconciliation of both grids can be assured in an intuitive way if a graphical editor is used to define the mapping (3). In defining the functions $\boldsymbol{F}_j$ in (10) one simply needs to provide a number of points such that the corresponding structures are drawn on the grid, which is quite natural. Consider the parameter ϕ_j and a set of its values

$$J_j = \{ \phi_j^0 \quad \phi_j^1 \quad \dots \quad \phi_j^{Nj} \} \tag{19}$$

We assume that each of the corresponding N_j+1 structures is drawn on the grid. An underlying assumption is that only ϕ_j is varied while all other parameters are kept at their nominal values. Also consider one of the coordinates of the vertex x_{vi}. The N_j+1 locations of the vertex corresponding to the parameter values (19) establish a set of the coordinate values, which, by definition, can be expressed as

$$x^l = k_j^l \Delta x, \quad l = 0, 1, \dots, N_j \tag{20}$$

The vertex and coordinate indices are dropped for simplicity. The mapping is constructed to satisfy

$$x^l - x^0 = f_{ij}(\phi_j^l - \phi_j^0), \quad l = 1, \dots, N_j \tag{21}$$

Of course, the form of each function f_{ij} in (9) must be capable of accommodating (21), e.g., cubic splines, or a polynomial of the N_jth degree. Then we can formulate the following lemma.

Lemma

For any point in the Cartesian product of all the sets (19)

$$\phi \in J_1 \times J_2 \times \dots \times J_n \tag{22}$$

the corresponding structure defined by any mapping (3) satisfying (5)-(10), (20) and (21) is on the grid, i.e., each vertex of the structure can be expressed as in (18).

The proof of the lemma follows immediately from (20) by noticing that, when the partial results of the form of the RHS of (21) are combined together through (9) and (7) for all parameters and each coordinate, Δx is coordinate specific and common for all the terms. Thus the resulting x_{vi} can be expressed as in (18) and, therefore, is on the layout grid. This is true for all the vertices, which shows that the two grids are indeed reconciled.

It is worth pointing out that since the user of the EM simulator may have control over initially setting the grid sizes Δx_1, Δx_2 and Δx_3 the limitation of drawing the perturbed structures on the grid is not severe.

Note that, by definition, the grid (22) is finite. The lemma can be extended to other points in the parameter space if they are mapped to on-grid structures. However, there may not exist such points simply because a vertex trajectory, once determined according to (19)-(21), may never

pass through the layout grid points other than those implied by (20). If this is the case, the selected structures need to be drawn in the region of the expected solution since extrapolation outside of the region determined by (22) may lack sufficient accuracy. Also note that the user needs to supply ΣN_j perturbed structures, in addition to the nominal one. In the case of a large number of defining points (19) this might be tedious, and for practical reasons may limit the applicability of the lemma.

Grid Reconciliation: The Linear Case

The possible limitations of the general case can be avoided if the linear mapping (11) is used. First, we only need to assure that the direction of the straight line trajectory is such that it passes through some of the layout grid points. It will then pass repeatedly through some other layout grid points, thus extending the parameter grid infinitely. Secondly, only one perturbed on-grid structure (drawing) needs to be specified for each parameter (N_j = 1). Specifically, if the values of the parameter ϕ_j corresponding to the nominal and perturbed structures are denoted by ϕ_j^0 and ϕ_j^{pert} then ϕ_j can be discretized as

$$\phi_j = \phi_j^0 + m_j \Delta\phi_j \tag{23}$$

where m_j is an integer number (positive, zero or negative),

$$\Delta\phi_j = (\phi_j^{pert} - \phi_j^0)/M \tag{24}$$

and M-1 is the number of additional layout grid points on the trajectory between the two vertex locations $\boldsymbol{x}_{vi}^0$ and $\boldsymbol{x}_{vi}^{pert}$ corresponding to ϕ_j^0 and ϕ_j^{pert}, respectively. The expression (24) establishes an n-dimensional unbounded uniform rectangular grid in the space of the parameters $\boldsymbol{\phi}$.

Again, considering one of the coordinates of the vertex $\boldsymbol{x}_{vi}$ at the nominal and perturbed locations ($\boldsymbol{x}_{vi}^0$ and $\boldsymbol{x}_{vi}^{pert}$), denoted by x^0 and x^{pert} we have, by construction

$$x^0 = k_j^0 \Delta x \tag{25}$$

and

$$x^{pert} = k_j^{pert} \Delta x \tag{26}$$

Since the layout grid is rectangular and uniform the number of grid sizes Δx in the difference x^{pert}-x^0 must an integer multiple of M, i.e., k_j^{pert}-k_j^0 = KM. K is a vertex/coordinate/parameter

specific non-zero integer. It can then be shown that the corresponding component of the mapping (11) can be expressed as

$$x - x^0 = (\phi_j - \phi_j^0)K\Delta x/\Delta\phi_j \tag{27}$$

Therefore, if ϕ_j is on the (parameter) grid, i.e., represented as in (23), then, from (27) and (25), x can be expressed as

$$x = (k_j^0 + m_j K)\Delta x \tag{28}$$

Clearly, (28) takes the form of (18) although some safeguards may need to be applied to ensure a positive value of the coefficient. As in the general case, the partial results of the form of (28) are combined together through (9) and (7) for all parameters and each coordinate, and through (8) for all vertices, leading to an on-grid structure.

III. EXAMPLES

The Double Folded Stub Microstrip Filter

Consider the double folded stub microstrip filter (see for example [4]) shown in Fig. 6. We consider changing the overall length of the filter, the length of the folded segments of the stubs, the spacing of the folded segments of the stubs, and the width of the main line and of the stubs as allowable modifications to the structure. This can be controlled by the parameters L_1, L_2, S, W_1 and W_2 marked in the diagram. Here, the parameterization process is implemented by Empipe [13] and ***xgeom*** [14]. First, a nominal structure is fully characterized using ***xgeom***, as shown in Fig. 7. This includes drawing, specifying the grid, box, substrate, etc., and entering all material constants. Then, the structure is sequentially edited to reflect changes w.r.t. to each optimizable parameter. Every such modified structure needs to be fully characterized. Fig. 8 shows the structures corresponding to S=4.8 mil and to S=11.2 mil, respectively.

Similar structures reflecting modifications due to the remaining parameters need to be drawn. Then Empipe's Geometry Capture tool captures the absolute coordinates of all the vertices for the nominal and modified structures. Finally, Empipe prompts the user to provide the values of parameters (4.8 mil and 11.2 mil in the case of Figs. 8(a) and 8(b)) corresponding to all drawings,

as well as certain data needed for discretization. As a result, after performing all mathematical calculations, Empipe generates a new optimization-ready library element that can be stored and reused.

Independent and Interdependent Parameters

This example illustrates various options in selecting structure parameters. In Fig. 9(a), the incremental changes represent two separate and independent parameters, namely *L1* and *L2*. Fig. 9(b), however, implies that the parameters *L1* and *L2* are strongly correlated: an increase in *L1* implies a decrease in *L2* and vice versa. This is necessary in order to maintain the overall length (*L1* + *L2*) constant, which is a commonly encountered constraint in layout within a fixed enclosure. In fact, for the structure depicted in Fig. 9(b), there is really only one degree of freedom, and therefore it will suffice to define just one parameter (*L1* or *L2*).

The WR-75 Waveguide Two-Section Mitered Bend

Consider the waveguide two-section mitered bend depicted in Fig. 10. Optimization of this structure is reported in [15]. The miter consists of two sections, symmetrically placed as shown in Fig. 11. The allowable change to the miter is the location of the edge connecting the two sections: along the dashed line in Fig. 11. This can be controlled by the parameter *d*, also shown in Fig. 11. Note, that the trajectories of vertex movement are straight lines (though not parallel to any axes) so the linear form of mapping is suitable to handle this case.

Here, the parameterization process is implemented by Empipe3D [13] and the 3D Solid Modeler of Maxwell Eminence [16] (the 3D Solid Modeler of HFSS [17] could be used as well). First, a nominal structure is fully characterized using Maxwell Eminence, as shown in Fig. 12. Maxwell Eminence allows exploitation of geometrical symmetry to reduce computation time. For the waveguide bend structure, we can set up a "Perfect H Boundary" so that only half of the structure needs to be analyzed by the 3D solver. In Fig. 12, the waveguide appears to have a square cross section, since the solid model is one half of the actual structure. The actual waveguide

dimension is 0.75 × 0.375 inch, as illustrated in Fig. 10.

In order to parameterize the bend, we need to create a perturbed project representing an incremental change in d. Fig. 13 shows the cross-section of the bend for two values: d = 0.1 inch and d = 0.05 inch. Then Empipe3D's Geometry Capture tool captures the information necessary for translating parameter values to a corresponding solid model. Similarly to the first example, Empipe3D generates a new optimization-ready library element that can be stored and reused.

IV. CONCLUSIONS

We have examined theoretical concepts and formulations relevant to parameterization of arbitrary geometrical structures for automated layout-based optimization using EM tools. This is to facilitate friendly user-parameterization of geometrical objects. Once a structure has been parameterized with user-defined parameters controlling its dimensions (size as well as shape), it becomes available for automated optimization. Significantly, the structure can be saved and reused, thus augmenting a customized library of elements.

Our theoretical derivations are not bound by any particular EM solver. Certain assumptions have been made to keep the technique simple and manageable. We expect that our innovations will become widely used in optimization-oriented layout-based applications, including MMIC design, VLSI design, high-speed interconnect design, particularly in conjunction with EM simulators.

ACKNOWLEDGEMENT

The authors would like to thank Dr. James C. Rautio, President, Sonnet Software, Inc., Liverpool, NY, for making his software available for this work. The authors would also like to thank Ansoft Corp. of Pittsburgh, PA, and HP EEsof Division, Santa Rosa, CA, for providing their 3D finite element solvers. This work was carried out and supported in part by Optimization Systems Associates Inc. before acquisition by HP EEsof, and supported in part by the Natural Sciences and Engineering Research Council of Canada under Grants OGP0007239, OGP0042444 and STR0167080 and through the Micronet Network of Centres of Excellence.

REFERENCES

1. J.W. Bandler, R.M. Biernacki and S.H. Chen, "Parameterization of arbitrary geometrical structures for automated electromagnetic optimization," *IEEE MTT-S Int. Microwave Symp. Dig.* (San Francisco, CA), 1996, pp. 1059-1062.

2. J.W. Bandler, R.M. Biernacki, Q. Cai, S.H. Chen and P.A. Grobelny, "Integrated harmonic balance and electromagnetic optimization with Geometry Capture," *IEEE MTT-S Int. Microwave Symp. Dig.* (Orlando, FL), 1995, pp. 793-796.

3. M.A. Schamberger and A.K. Sharma, "A generalized electromagnetic optimization procedure for the design of complex interacting structures in hybrid and monolithic microwave integrated circuits," *IEEE MTT-S Int. Microwave Symp. Dig.* (Orlando, FL), 1995, pp. 1191-1194.

4. J.W. Bandler, R.M. Biernacki, S.H. Chen, D.G. Swanson, Jr., and S. Ye, "Microstrip filter design using direct EM field simulation," *IEEE Trans. Microwave Theory Tech.*, vol. 42, 1994, pp. 1353-1359.

5. J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny and S. Ye, "Yield-driven electromagnetic optimization via multilevel multidimensional models," *IEEE Trans. Microwave Theory Tech.*, vol. 41, 1993, pp. 2269-2278.

6. J.W. Bandler, R.M. Biernacki, S.H. Chen and P.A. Grobelny, "A CAD environment for performance and yield driven circuit design employing electromagnetic field simulators," *Proc. Int. Symp. Circuits and Systems* (London, England), vol. 1, 1994, pp. 145-148.

7. J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny and R.H. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Trans. Microwave Theory Tech.*, vol. 42, 1994, pp. 2536-2544.

8. J.W. Bandler, R.M. Biernacki, S.H. Chen, R.H. Hemmers and K. Madsen, "Electromagnetic optimization exploiting aggressive space mapping," *IEEE Trans. Microwave Theory Tech.*, vol. 43, 1995, pp. 2874-2882.

9. F. Arndt, J.W. Bandler, W.J.R. Hoefer, A.M. Pavio, J.C. Rautio, R. Sorrentino, D.G. Swanson, Jr., S.H. Talisa, R.J. Trew, "Circuit Design with Direct Optimization-driven Electromagnetic Simulators," Panel Session, IEEE MTT-S Int. Microwave Symp. (San Diego, CA), 1994.

10. F. Arndt, S.H. Chen, W.J.R. Hoefer, N. Jain, R.H. Jansen, A.M. Pavio, R.A. Pucel, R. Sorrentino and D.G. Swanson, Jr., *Automated Circuit Design using Electromagnetic Simulators*. Workshop WMFE (J.W. Bandler and R. Sorrentino, Organizers and Chairmen), IEEE MTT-S Int. Microwave Symp. (Orlando, FL), 1995.

11. P.P.M. So, W.J.R. Hoefer, J.W. Bandler, R.M. Biernacki and S.H. Chen, "Hybrid frequency/time domain field theory based CAD of microwave circuits," *Proc. 23rd European Microwave Conf.* (Madrid, Spain), 1993, pp. 218-219.

12. F. Alessandri, M. Dionigi, R. Sorrentino and M. Mongiardo, "A fullwave CAD tool of waveguide components using a high speed direct optimizer," *IEEE MTT-S Int. Microwave Symp. Dig.* (San Diego, CA), 1994, pp. 1539-1542.

13. *Empipe*™ and *Empipe3D*™, formerly Optimization Systems Associates Inc., P.O. Box 8083,

Dundas, Ontario, Canada L9H 5E7, now HP EEsof Divison, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403.

14. ***xgeom***™, Sonnet Software, Inc., 1020 Seventh North Street, Suite 210, Liverpool, NY 13088.

15. J.W. Bandler, R.M. Biernacki, S.H. Chen, L.W. Hendrick and D. Omeragić, "Electromagnetic optimization of 3D structures," *IEEE Trans. Microwave Theory Tech.*, vol. 45, 1997, pp. 770-779.

16. *Maxwell® Eminence*, Ansoft Corporation, Four Station Square, Suite 660, Pittsburgh, PA 15219.

17. *HFSS*, HP EEsof Divison, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403.

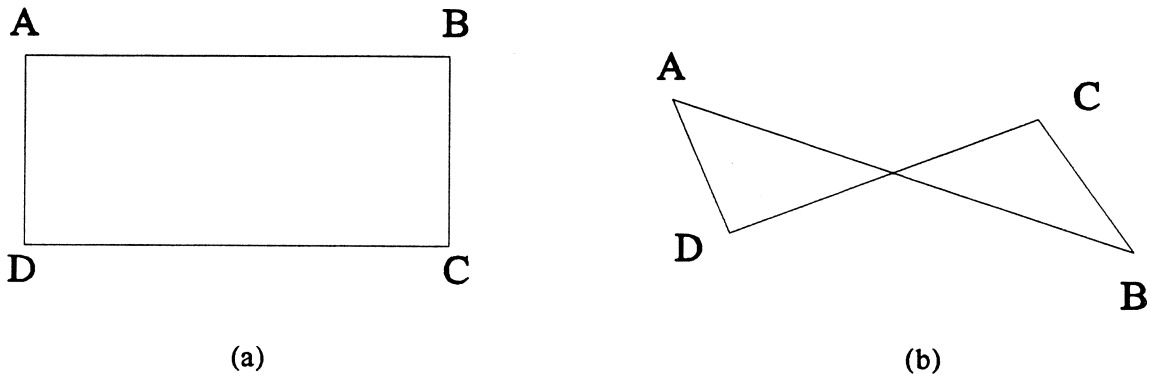


Fig. 1. Arbitrary movement of vertices of a polygon: (a) the initial geometry, and (b) an unwanted result due to an arbitrary and independent movement of vertices.

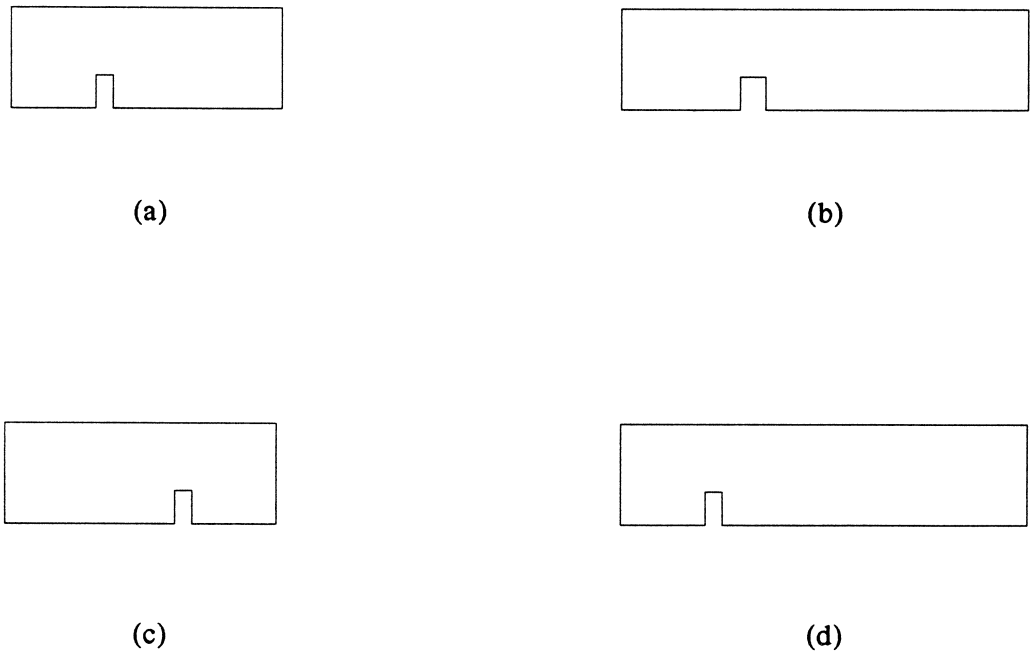


Fig. 2. Various evolutions of a microstrip line with a slit: (a) the initial geometry, (b) proportional expansion of the whole structure along the x axis, (c) only the location of the slit in the fixed line is allowed to change, and (d) only the segment to the right of the slit is allowed to expand.

Fig. 3. Evolution of a rectangle (a) to a tapered line (b).

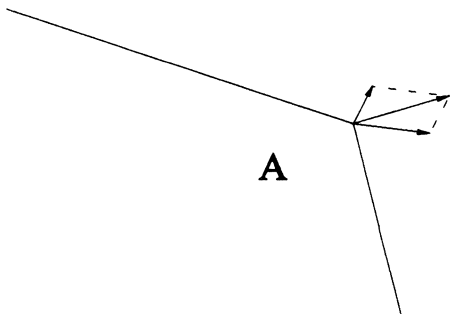


Fig. 4. Additiveness of the vertex movement w.r.t. the changes in individual parameters.

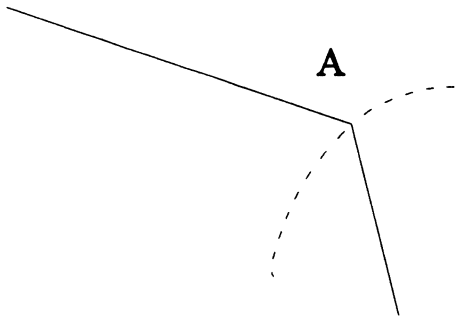


Fig. 5. Possible trajectory of the movement of a vertex with a change in a parameter.

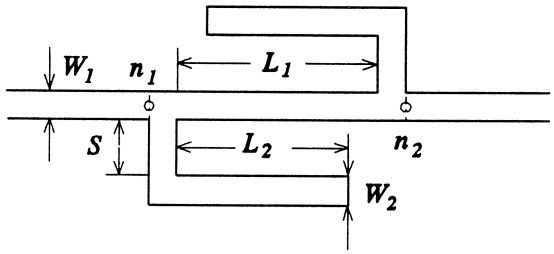


Fig. 6. Double folded stub filter.

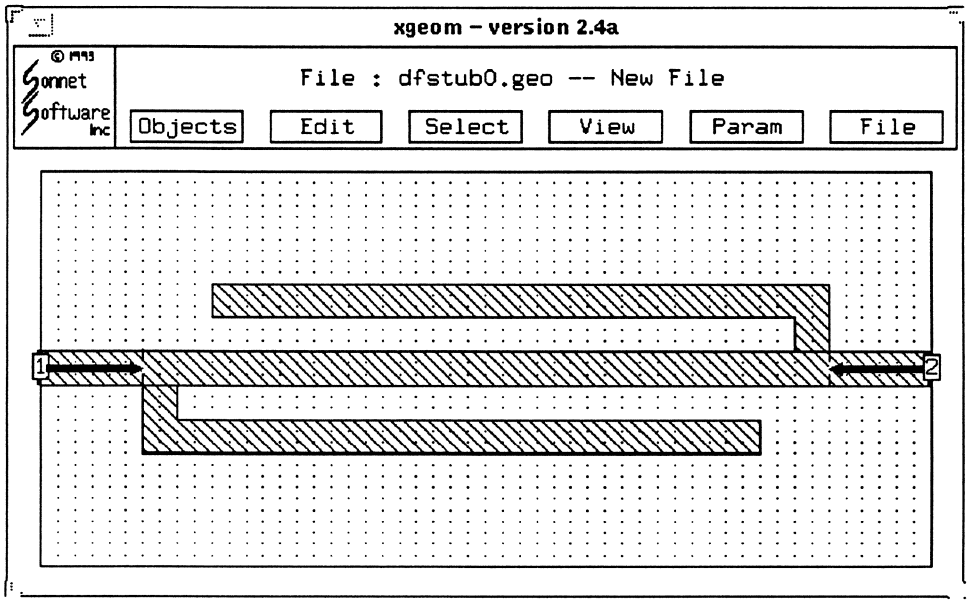
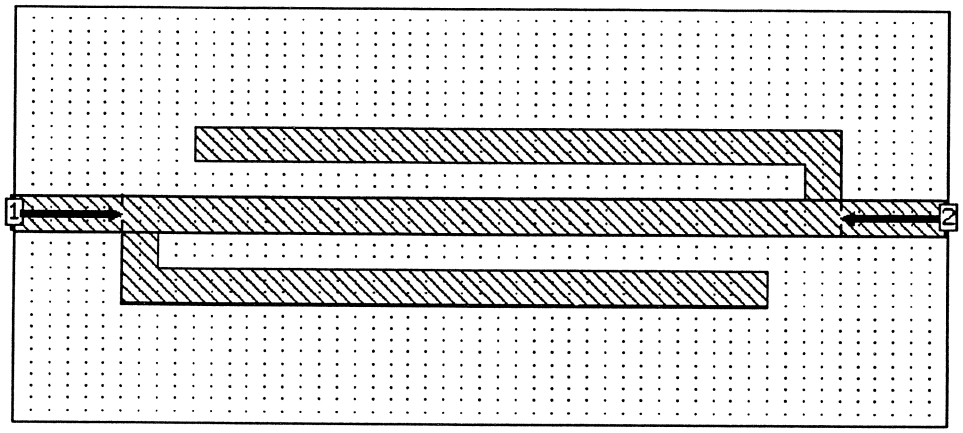
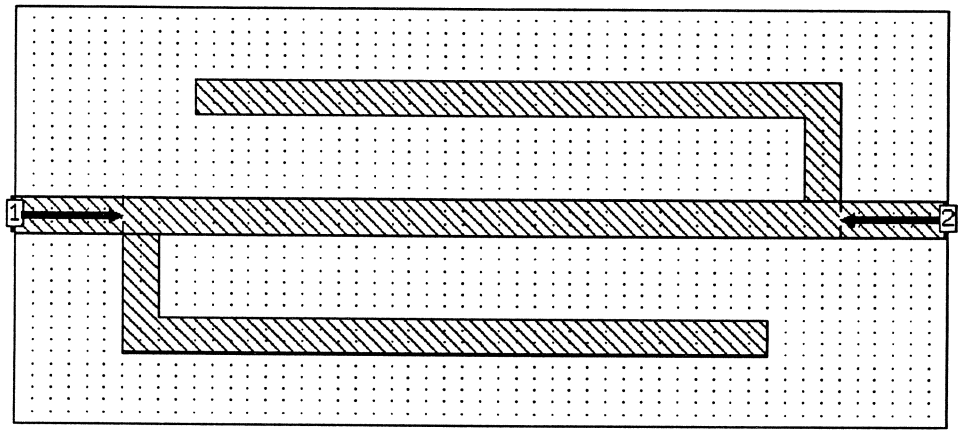


Fig. 7. Parameterization of the double folded stub filter: the nominal geometry.

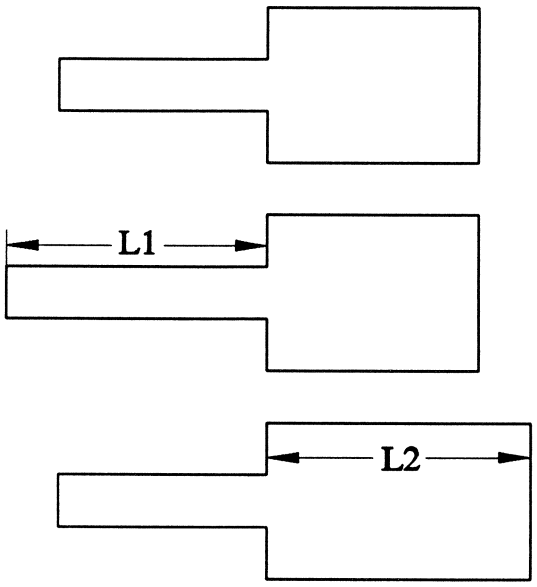


(a)

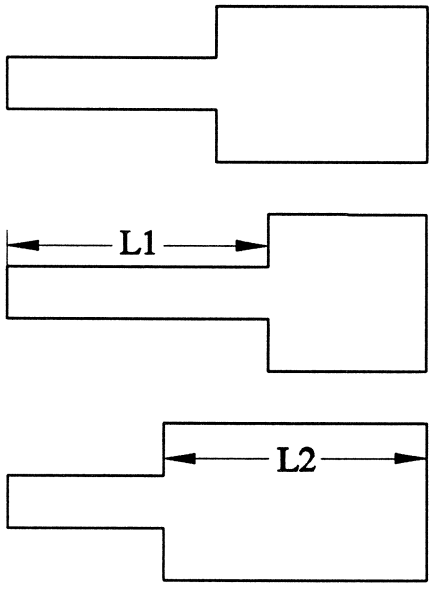


(b)

Fig. 8. Parameterization of the double folded stub filter: (a) the filter structure for S=4.8 mil, and (b) the filter structure for S=11.2 mil.



(a)



(b)

Fig. 9. The parameters L1 and L2 are (a) independent, and (b) interdependent.

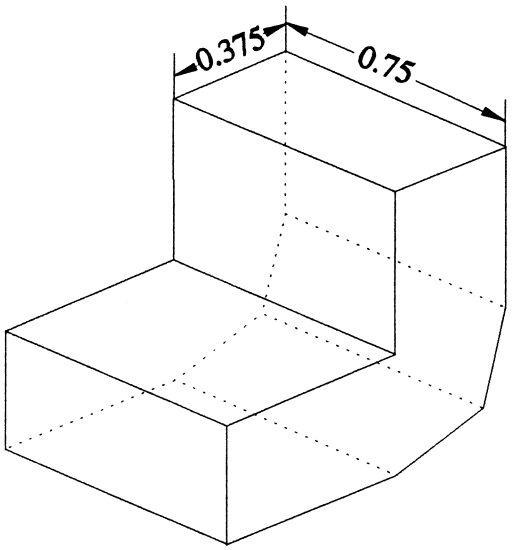


Fig. 10. The WR-75 waveguide two-section mitered bend.

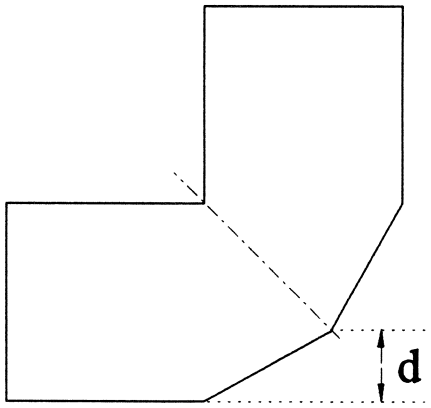


Fig. 11. The optimizable parameter *d* for the two-section mitered bend.

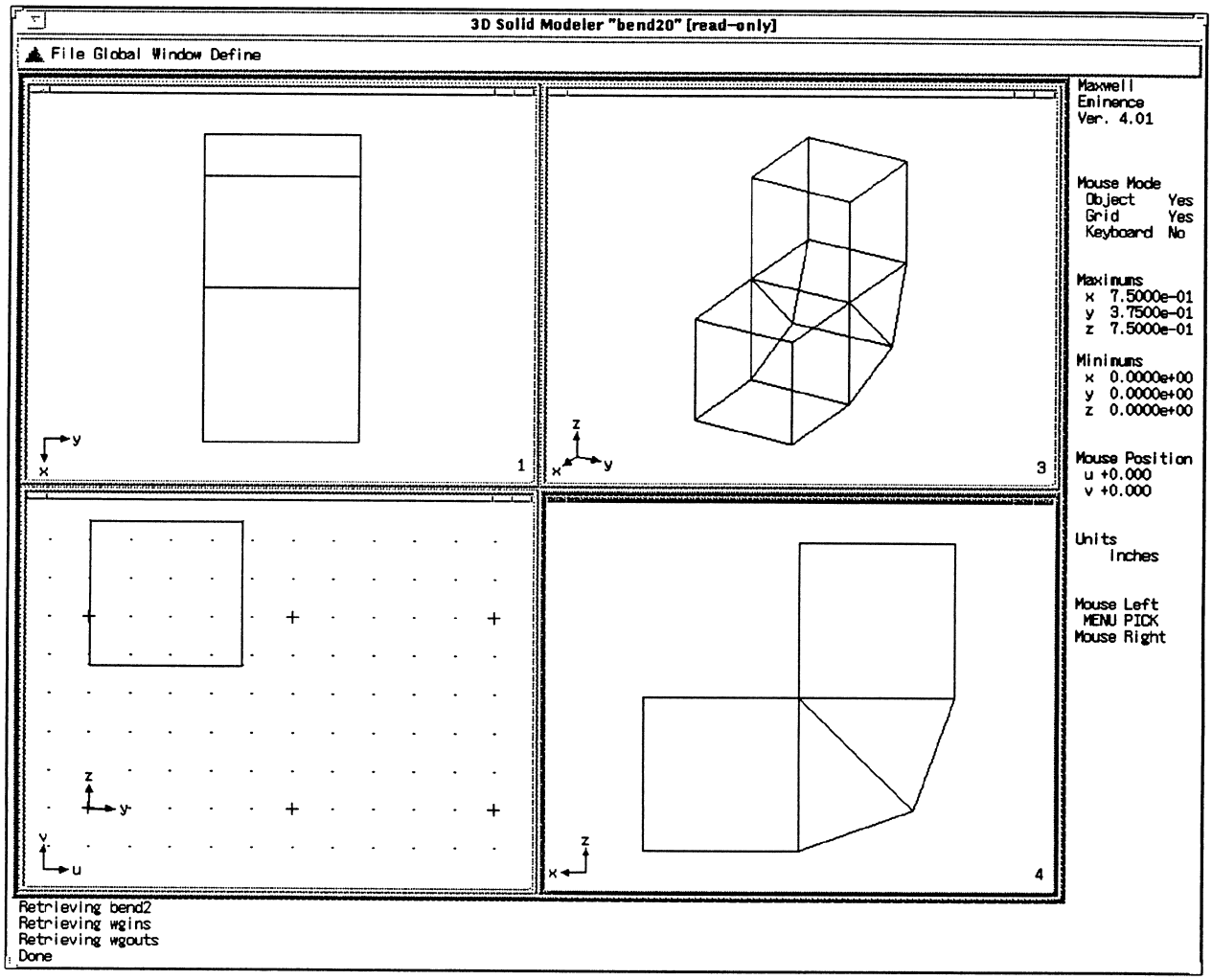
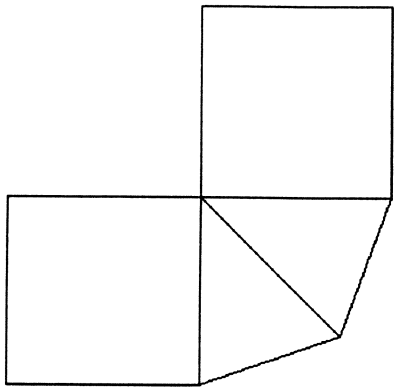
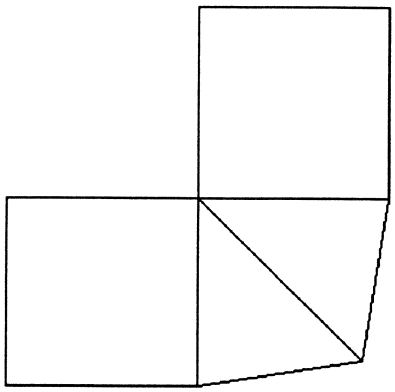


Fig. 12. Solid model for the waveguide two-section mitered bend.



project for d = 0.1 inch



project for d = 0.05 inch

Fig. 13. Cross-section of the waveguide mitered bend for two distinct values of the parameter d.