A TOPOLOGY BASED SYSTEM FOR SIMULATING 3D CRACK GROWTH IN SOLID AND SHELL STRUCTURES

B.J. CARTER, C.-S. CHEN, A.R. INGRAFFEA, and P.A. WAWRZYNEK
Cornell Fracture Group, 431 Rhodes Hall,
Cornell University, Ithaca, NY, USA 14853

ABSTRACT

A software framework, FRANC3D, is developed to simulate arbitrary crack growth in both solid and shell structures. A conceptual model of this crack growth simulator defines the major components of the software and suggests some necessary requirements for any computer implementation. The software makes use of a constrained hierarchy of topology-based geometrical models using a boundary representation and a radial-edge database to define the solid model. Crack propagation involves updating the geometry and topology of the model based on a criterion for crack growth and the current equilibrium state information. Equilibrium state information can be obtained from any stress analysis code provided that an appropriate software interface exists. Changes to the mesh model due to crack growth can be localized and minimized to the regions around the crack which implies that an incremental crack growth process can be modeled quickly and efficiently even in three dimensions.

KEYWORDS

Three dimensional crack propagation, topology, radial edge database, boundary representation.

INTRODUCTION

Cracking is a major problem in many engineering fields. From microscopic cracks in computer chips to megascopic cracks in the earth's crust, it has been seen that cracks can have a detrimental effect on the performance of engineered structures. There are also instances, such as hydraulically induced fractures in oil wells, where cracks are created purposely. In these cases, however, if the crack growth is not in the desired location, direction and proper shape, the crack can do more harm than good.

Although many crack growth problems can be simplified and analyzed in two dimensions, an efficient and reliable three dimensional crack growth simulator is required for analyzing more complex problems. Current practice generally relies on highly idealized models of observed behavior, usually simplifying the problem to either a planar or two dimensional crack. The accuracy of predictions based on these idealizations has not been well characterized, however, and it is important that a fully three dimensional simulation tool be available for cases where such simplifications are not possible.

Most commercial stress analysis software packages are not suitable for the purpose of simulating arbitrary non-planar three dimensional crack growth. Although, many programs can perform a stress analysis of a cracked structure, the subsequent propagation of the crack usually is not a simple process.

This paper presents a conceptual model of a software framework that allows a user to efficiently simulate three dimensional crack propagation in complex engineering structures. An abstract model of the representational aspects of a crack growth simulation is discussed in detail; some of the additional mechanisms that are needed to simulate crack growth are summarized briefly. Two examples of arbitrary crack growth in real engineering structures illustrate some of the concepts of the software framework and the conceptual model.

A CONCEPTUAL MODEL OF CRACK GROWTH SIMULATION

There are two primary aspects to the numerical simulation of arbitrary crack growth; these are representation and physics. Representation includes the details of storing the geometry of a cracked body in a computer and updating the geometric description to reflect crack growth. Physics includes stress analysis, extraction of relevant crack growth parameters, and determination of the shape, extent, and direction of crack growth. A conceptual model that focuses on the representation of a discrete crack growth simulation process is presented below.

Crack growth simulation is an incremental process, where a series of steps is repeated for a progression of models. Each iteration in the process relies on previously computed results and represents one crack configuration. There are four primary collections of data, or databases, required for each iteration. The first is the representational database, denoted $\mathbf{R}_i$, (where the subscript identifies the iteration or increment number). The representational database contains a description of the solid model geometry, including the cracks, the boundary conditions, and the material properties. The representational database is transformed by a discretization (meshing) process to a stress analysis database, $\mathbf{A}_i$. The analysis database contains a complete, but approximate description of the body, suitable for input to a solution procedure, usually a finite or boundary element stress analysis program.

The solution procedure is used to transform the analysis database to an equilibrium database, $\mathbf{E}_i$, which consists of primary (loads and displacements) and secondary (stresses and strains) field variables that define the equilibrium solution for the analysis model, $\mathbf{A}_i$. The equilibrium model should contain field variables and material state information for all locations in the body, and in the context of a crack growth simulation, should also contain values for stress-intensity factors, or other fracture parameters, $\mathbf{F}_i$, for all crack fronts. The equilibrium database is used in conjunction with the current representational database to create a new representational model, $\mathbf{R}_{i+1}$, which includes the incremental growth of the crack.

The simulation process is described symbolically as follows. A meshing function, $\mathbf{M}$, transforms a representational description of a cracked body to a discrete model suitable for stress analysis,

$$\mathbf{M}(\mathbf{R}_i) \rightarrow \mathbf{A}_i .$$

A solution procedure, $\mathbf{S}$, computes unknown field variables, $\mathbf{E}_i$, and fracture parameters, $\mathbf{F}_i$,

$$\mathbf{S}(\mathbf{A}_i) \rightarrow \mathbf{E}_i, \mathbf{F}_i .$$

A function which updates the representational model, $\mathbf{U}$, takes the equilibrium state field variables, the existing representation, and a function which predicts crack shape evolution, $\mathbf{C}$, and creates a new representational database,

$$\mathbf{U}(\mathbf{E}_i, \mathbf{R}_i, \mathbf{C}(\mathbf{F}_i)) \rightarrow \mathbf{R}_{i+1} .$$

This process is performed incrementally (Fig 1), and is repeated until a suitable termination condition is reached. Results of such a simulation might include one or more of the following: a final crack geometry, a loading versus crack size history, a crack opening profile, or a history of the crack-front fracture parameters.

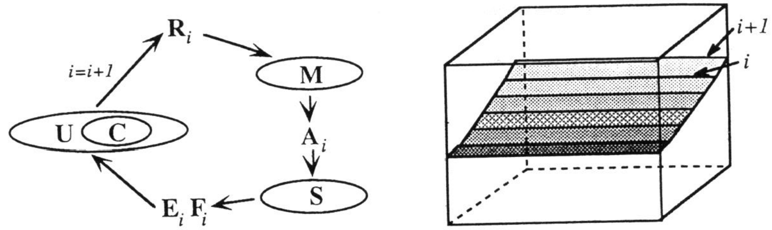

Fig 1. Incremental crack growth simulations; *i* denotes the increment of crack growth.

This conceptual model is organized within a software framework called FRANC3D (Fig 2). FRANC3D encompasses all components of the conceptual model except for the stress analysis procedure. The individual components consists of unique databases and functions that operate on the databases. These are described in more detail in the following sections.

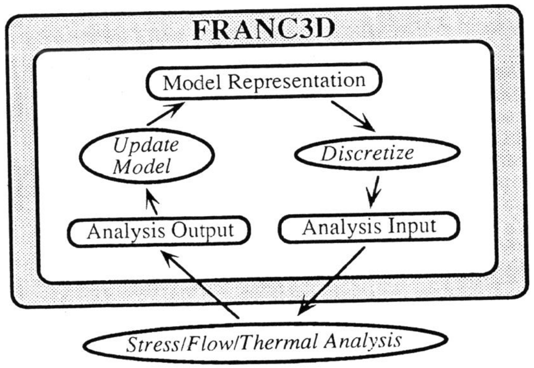

Figure 2. A conceptual model for crack growth simulation incorporated into a software framework called FRANC3D.

MECHANISMS AND METHODOLOGIES FOR SIMULATING CRACK GROWTH

Before useful engineering simulations can be performed, the abstract databases, $\mathbf{R}_i$, $\mathbf{A}_i$, $\mathbf{E}_i$, and $\mathbf{F}_i$, and the abstract functions, $\mathbf{M}$, $\mathbf{S}$, $\mathbf{C}$, and $\mathbf{U}$, must be defined in terms of data structures and algorithms. Much of this has been done for other applications of computational mechanics and will not be detailed here. This section focuses on aspects of the mechanisms and methodologies that differ in the context of fracture mechanics from other applications. These include aspects of geometrical modeling, the use of computational topology, and model hierarchies and constraint.

Solid Modeling For Crack Growth Simulations

Simulation of crack growth is more complicated than many other applications of computational mechanics because the geometry and topology of the structure evolve during the simulation. For this reason, a geometric description of the body that is independent of any mesh should be maintained and updated as part of the simulation process. The geometry database should contain an explicit description of the solid model, including the crack. The three most widely used solid modeling techniques, boundary representation (B-rep), constructive solid geometry (CSG), and parametric analytical patches (PAP) (Hoffmann, 1989; Mäntylä, 1988; Mortenson; 1985), are capable of representing uncracked geometries. Of the three, boundary representation is found to be the most suitable for modeling cracks.

Crack faces are two surfaces that share a common geometric description. They can be modeled quite readily with a B-rep modeler, which stores surfaces and surface geometries explicitly. If explicit topological adjacency information (defined below) is available as well, two topologically distinct surfaces can share a common geometric description. Another fundamental capability of any solid modeling system that is used for simulating crack growth is the classification of points on a crack. Conventional point classification determines if the point is in a body, outside a body, or on the surface of a body. A point on a crack surface cannot be classified as any one of these three. It is simultaneously on the surface of the body at two distinct independent locations, with no adjacent points that lie outside the body. Point classification on crack surfaces presents no difficulty if both crack surfaces and explicit topological adjacency information exist, another reason for choosing a B-rep model.

Structural Idealizations

To this point, the focus has been three-dimensional cracks in three-dimensional bodies. It is common when performing stress analyses, however, to idealize portions or all of a body as a dimensionally degenerate form, such as a plate or a shell. Such idealizations may provide great savings in analysis time and effort, and a comprehensive simulator should be able to model crack growth in these degenerate forms even though it adds extra constraints to the solid modeler. A B-rep modeler is capable of storing and manipulating such degenerate forms which is another reason for choosing boundary representation over other solid modelers.

Computational Topology as a Framework for Crack Growth Simulation

As mentioned above, explicit topological information is an essential feature of the representational database for crack growth simulations. The topology of an object is the information about relationships, proximity, and order among features of the geometry—incomplete geometric information. These are the properties of the actual geometry that are invariant with respect to geometric transformations (Fig 3). A topology framework serves as an organizational tool for the data that represents an object and the algorithms that operate on the data.

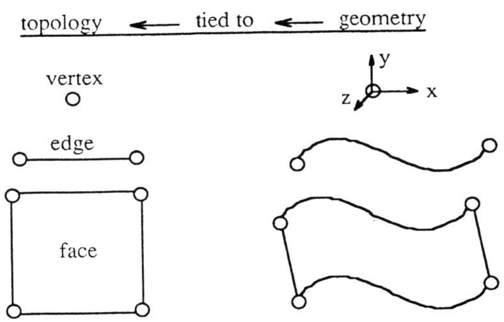

Fig 3. Relationship between topology and geometry; a topological entity can have any number of geometric descriptions.

Explicit topological information is not essential for crack growth simulation. However, there are at least four compelling reasons for using a topological representation:

1. Topological information, unlike geometrical information, can be stored exactly, with no approximations or ambiguity.
2. The theoretical background supporting the concepts of topology and boundary graphs can be used to develop formal and rigorous procedures for storing and manipulating these types of data (Mäntylä, 1988; Hoffmann, 1989; Weiler, 1986).
3. Any topological configuration can represent an infinite number of geometrical configurations.
4. During crack propagation, the geometry of the structure changes with each crack increment whereas the topology changes much less frequently.

Previous investigations into the use of data structures for storing information needed for crack propagation simulations (Wawrzynek, 1986, 1987a&b) showed that topological databases were a convenient and powerful organizing agent. The specific advantages of using topological data structures are:

1. Local modifications can be made without the need to perform global reorganizations of the rest of the data.
2. The data structure maintains a consistent representation during all phases of modeling. This is helpful during the intermediate stages of remeshing; the regions in the mesh where elements are deleted are still consistently represented and recorded as topological entities.
3. The use of graph theoretical operators hides complexities in manipulating the actual data. This encourages a modular system that aids in the development and maintenance of the software.
4. Efficient topological adjacency queries make this type of data organization ideally suited for interactive modeling.

With respect to the abstract model for simulating crack growth, explicit topological information is useful as a framework for the representational database, $\mathbf{R}_i$, and will aid in implementation of the meshing function, $\mathbf{M}$, and the updating function $\mathbf{U}$. In particular, if a topological database is used in conjunction with a B-rep modeler to implement the geometry model, topological entities can serve as the principal elements of the database, with geometrical descriptions and all other attributes (such as boundary conditions and material properties) accessed through the topological entities.

Several topological data structures have been proposed for manifold objects. These include the winged-edge (Baumgart, 1975), the modified winged-edge, the face-edge, the vertex-edge (Weiler, 1985), and the half-edge (Mäntylä, 1988) data structures. The so-called Euler operators, that enforce the *Euler-Poincaré* formulae for entities in a graph, are used to manipulate all of these structures, making the external interfaces identical. From a practical point of view, the difference among the structures is the efficiency with which the various Euler operators and queries can be implemented.

In computational mechanics, it is often desirable to work with idealizations of real objects (Shephard, 1985, Potyondy, 1993), some of which may contain components with no volume (e.g., shells and plates). Such idealizations cannot be represented by the data structures mentioned above because the idealizations create non-manifold topologies (Fig 4). That is, the topologies cannot exist on a two-manifold (see Mortenson, 1985). Other features that introduce non-manifold conditions include internal surfaces, such as bi-material interfaces, some crack configurations, and many transient configurations encountered during the construction or modification of a topological representation of a body.

Weiler (1986) presented another edge-based data structure for storing non-manifold objects, called the radial-edge, and outlined the corresponding generalized non-manifold Euler operators. The basic topological entities used for modeling are vertices, edges, faces, and regions. An internal crack, for example, consists of vertices, edges, and faces with a null volume region between the crack surfaces. The edge entity is the object through which topological relationships are maintained and queried (Fig 5). As the name implies, the edge uses are ordered radially about the edge. Each face has two face uses and each face use has a corresponding edge use. The radial ordering is used to efficiently store, query and manipulate the model topology.

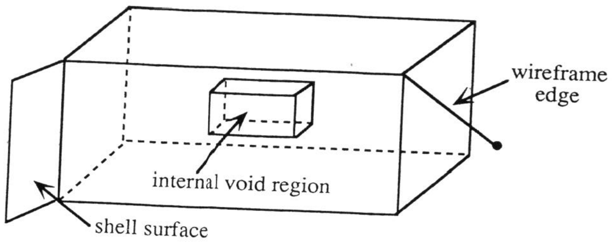

Fig 4. Non-manifold topologies can include internal voids, shell surfaces, and wireframe edges.

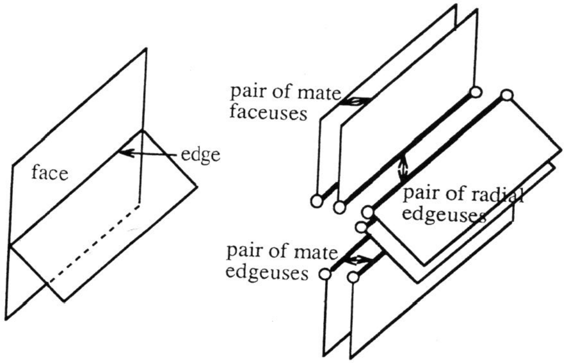

Fig 5. The radial-edge database relies on the radial ordering of edge uses about each edge. A face has two face uses and the edge has a use with respect to each face use (after Weiler, 1988).

A Hierarchy of Models and Constraints

Thus far, only two independent representations of a body, the geometry and the analysis, or mesh, models were mentioned. However, a hierarchy of five independent models is found to ease the implementation, particularly for the meshing function **M** (Martha, 1989). Within the model hierarchy is a strong notion of constraint. That is, entities at any level of the hierarchy are constrained by those in the levels above (Fig 6).

The geometry (Fig 6a) and mesh (Fig 6e) description represent the highest and lowest levels in the five-level hierarchy, respectively. The second highest level is a volume decomposition level (Fig 6b) where regions of the geometric model can be divided into subregions. This level is useful for defining regularly shaped regions for the volume meshing algorithm. The third

highest level is a surface decomposition level (Fig 6c). This step is useful when decomposing an irregularly shaped surface into subsurfaces for surface meshing purposes. The next level is the edge subdivision level (Fig 6d). This level has two purposes. The spline space-curves of the three higher levels are approximated by piecewise linear line segments, and the subdivision edge size defines a metric for the local mesh density. The lowest level in the hierarchy is the mesh (whether it's a surface or volume mesh, or a combination). All levels are constrained by the levels above it, meaning that the lowest mesh level has element edges that correspond to the edge subdivision level edges, element faces cannot span surface subdivision boundaries, element volumes cannot span region subdivision boundaries, and the entire mesh must be constrained to the original geometry.

There are two primary purposes for such a hierarchy. One is to provide constraints and inheritance for interactive modeling, and the other is to aid in minimizing changes to the model during incremental analysis.

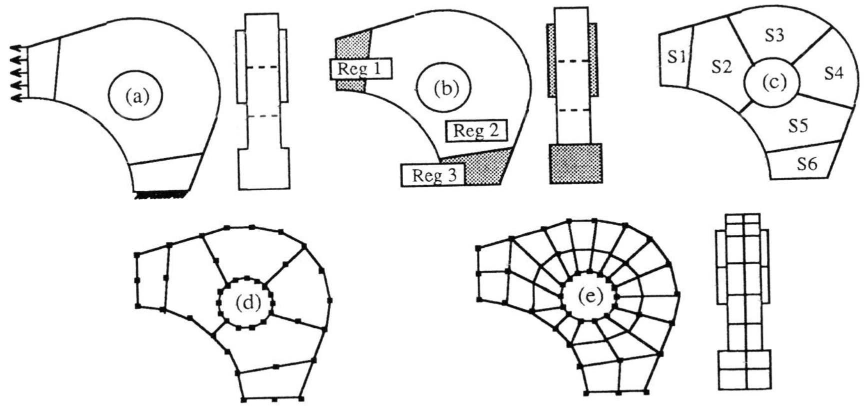

Fig 6. Constrained model hierarchy, (a) geometry, (b) volume decomposition, (c) surface decomposition, (d) edge subdivision, and (e) mesh.

Clearly, it is important that the mesh model is constrained to the geometry of the structure. The hierarchy allows the mesh level model to inherit not only the geometry, but the simulation attributes as well. Simulation attributes, which consist of such things as boundary conditions and material properties are part of the original geometry model. The stress analysis procedure, **S**, requires this data meaning that the mesh model must inherit the data from the geometry. Each level can have its own data, but lower levels automatically inherit the data from the levels above.

In incremental simulations, it is desirable that the changes to the model between increments be kept to a minimum. When a portion of a model is modified, a certain amount of related information becomes obsolete; to speed up the simulation process, the amount of lost information should be minimized. For example, in the case of crack propagation, some modifications to the geometry are made in the region near the crack front which invalidates the mesh in this region. However, portions of the model remote from the crack should not be affected. The total simulation time can be reduced significantly if only a small portion of the model requires remeshing after each crack increment. The five levels of model representation provide a convenient hierarchical framework for enforcing this concept.

The Abstract Functions and Analysis and Equilibrium Databases

There are four abstract functions specified in the abstract model for crack growth, **M**, **S**, **U**, and **C**. The purpose of the **M** function is to transform a geometry database to an analysis database that meets the input requirements of a particular stress analysis procedure. The major portion of this task is the creation of a surface or volume mesh for the body. An automatic or semi-automatic mesh generation capability should be employed, and these are well described in the literature (Baehmann *et al.*, 1987; Cavendish *et al.*, 1985; Kela *et al.*, 1987; Lo, 1989; Potyondy *et al*,. 1995, Shephard, 1985).

However, there are two aspects of mesh generation that are important in the context of crack growth simulation that may be less important in other applications. The first arises due to the geometric coincidence of crack faces. A meshing algorithm used to mesh a surface or volume containing crack faces cannot rely on geometrical checks exclusively while generating elements. This is because nodal points on opposing crack faces are distinct, but share a common location. It cannot be determined from geometrical checks alone if a candidate node is on the proper side of a crack. Algorithms that mesh such regions properly must resort to topological information to select the proper node.

The second aspect of mesh generation is important when considering the concept of minimized change to the model. In that case, often only a small portion of the body near the crack front needs remeshing. However, the new mesh must conform to the remaining unchanged portions of the mesh. Some meshing algorithms might not be capable of honoring such constraints, as they generate nodes along the boundary of the region during the meshing process.

The stress analysis function, **S**, can be any numerical analysis procedure which takes in the analysis database, $\mathbf{A}_i$, and produces the equilibrium state information, $\mathbf{E}_i$ and the required fracture parameters, $\mathbf{F}_i$. The main requirement of the numerical method is that it is able to accurately calculate the displacements and stresses near the crack front. Both finite and boundary element procedures have been developed for this purpose (Aliabadi and Rooke, 1991). The type of analysis will depend on the need to model geometric and material non-linearites as well.

The crack growth model, **C**, takes the field values at or near the crack front and evaluates the potential for further crack growth. The implementation in the current software framework has been described by Martha *et al.* (1993). Essentially, the displacements on the crack surface, near the crack front are obtained from the equilibrium state database, $\mathbf{E}_i$, and converted to stress intensity factors, $\mathbf{F}_i$. The stress intensity factors along the crack front are then compared with the material fracture toughness to determine whether the crack front will extend. If the stress intensity factors are greater than the material toughness, the crack will propagate, with the direction and amount of extension defined by existing theories (see Aliabadi and Rooke, 1991). Crack growth invalidates the current representational model, requiring updates to the geometry and sometimes the simulation attributes (e.g., if tractions are applied to the crack surface).

The update function, **U**, primarily involves modifying the geometry, and possibly the topology, due to crack growth and then remeshing of the locally modified portions of the model. The procedure for propagating surface and internal cracks has been described by Martha *et al.* (1993). Crack growth is constrained by the geometry as well as the crack growth model. In other words, the crack surface cannot extend beyond the boundaries of the structure even if the crack growth model predicts such an occurrence.

ILLUSTRATIVE EXAMPLES

The capability of the above model for simulating arbitrary crack propagation in three-dimensional structures and in dimensionally degenerate forms such as shells is best shown by practical example. Two examples are given; the first shows crack growth simulation in a portion of a turbine fan blade and the second shows crack growth in a portion of a pressurized fuselage. FRANC3D is used to simulate the crack growth process in both models; however, the stress analysis software differs. In the first case, a three dimensional boundary element code, BES (Lutz, 1991) is used, and in the second case, a general purpose shell finite element analysis program, STAGS (Brogan *et al.*, 1994), is used.

Example 1: Crack Growth Simulation in a Turbine Fan Blade

The FRANC3D geometric model ($\mathbf{R}_1$) of a typical turbine fan blade is shown in Fig 7. A small starter crack is placed in the root, and the geometric model is discretized using the meshing function (**M**) to produce the mesh model shown in Fig 8. BES, the stress analysis function (**S**), provides the equilibrium state information for the mesh model ($\mathbf{E}_1$). Stress intensity factors ($\mathbf{F}_1$) are computed at points along the crack front. The crack growth function (**C**) uses the mode I and II stress intensity factors to compute the crack growth increment. The model is updated by propagating the crack and remeshing any portion of the model that is affected by the changing crack geometry.

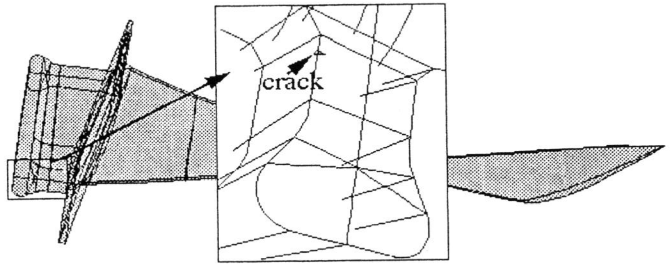

Fig 7. Geometry model of the turbine fan blade with an expanded view of the crack.

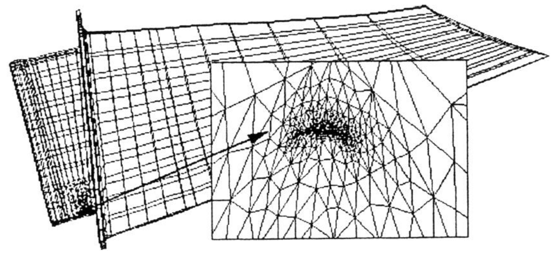

Fig 8. Initial mesh model of the turbine fan blade with an expanded view near the crack.

Fig 9a shows the mesh model after the crack has been propagated. Note that only a small portion of the model requires remeshing. The crack shape is truly three dimensional (Fig 9b); it is non-planar with an arbitrary crack front shape that is governed by the geometry and the loading. The analyses were stopped after twelve steps of propagation; the mode I stress intensity factor history indicates that the crack growth is beginning to accelerate (Fig 9c).

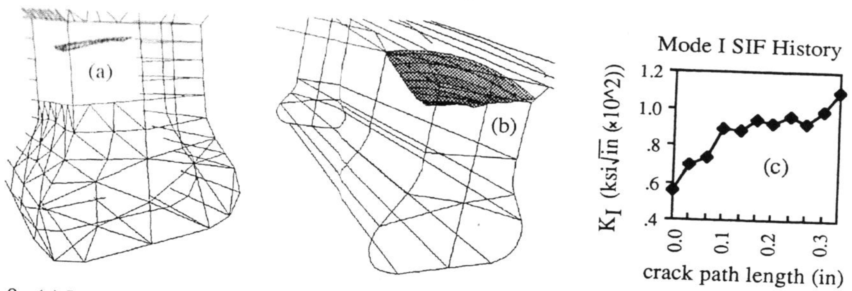

Fig 9. (a) Intermediate stage of crack growth showing the portion of the mesh that was deleted, (b) final stage of crack growth, and (c) mode I stress intensity factor history.

Example 2: Crack Growth Simulation in a Pressurized Fuselage

Crack growth in a pressurized fuselage panel that is representative of a typical narrow-body aircraft is simulated. The panel, comprised of skin, frames, stringers, tear straps, and shear clips, is illustrated in Fig 11. All the components are idealized as shell structures. The FRANC3D geometric model (**R**$_1$), as represented by a 2x2 bay panel with an initial saw cut along an edge of a stringer in the skin of the panel, is shown in Fig 12a. The FRANC3D mesh model (**A**$_1$), which applies the meshing function (**M**) described in Potyondy et al. (1995) to each topological face of the geometric model, is shown in Fig 12b.

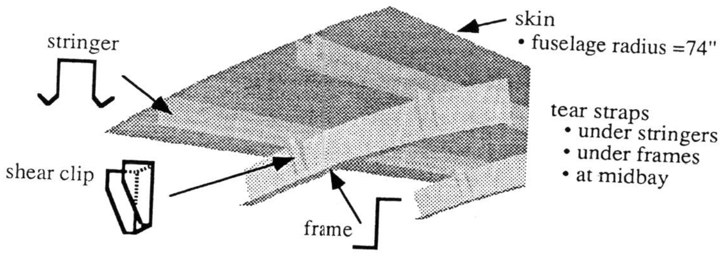

Fig 11. The structural components of a typical narrow-body aircraft fuselage.

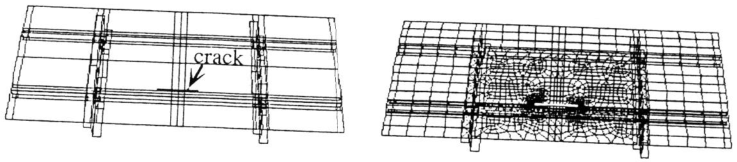

Fig 12. (a) The FRANC3D geometric model, and (b) the mesh model with an initial crack along an edge of a stringer in the skin of the panel.

The STAGS geometric nonlinear shell analysis program, the stress analysis function (**S**), is used to obtain the equilibrium state information (**E**$_1$). Four stress intensity factors (**F**$_1$), which

account for both the in-plane membrane loading and out-of-plane bending loading experienced by a crack in the fuselage panel (Hui and Zehnder, 1993), are computed based on **E**$_1$. A crack growth model (**C**) that considers crack propagation in an anisotropic medium (Boone *et al.*, 1987) is used to propagate the crack.

Since new edges are created to accommodate the crack growth, the topology of the geometric model is altered and the representational database needs to be updated from **R**$_1$ to **R**$_2$. With the aid of the model hierarchy and the concept of minimum change, only the region near the crack tip needs to be remeshed (**M**) while other areas remain unaltered. Fig 13a shows the regions of localized mesh deletion after growing the crack and Fig 13b shows the mesh model after remeshing.

The crack trajectories and the stress intensity factor history are determined for each crack growth increment. Fig 14 compares the numerically predicted crack trajectories with the experimental measurements from a full scale panel test (Miller *et al.*, 1992).

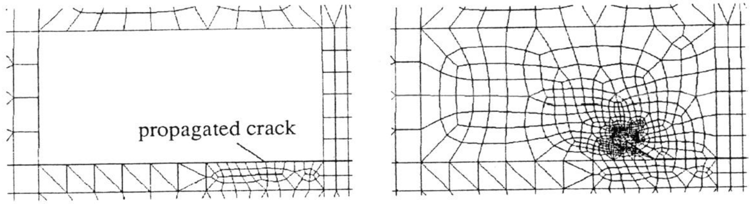

Fig 13. (a) The regions of localized mesh deletion after growing the crack, and (b) the FRANC3D mesh model after remeshing.

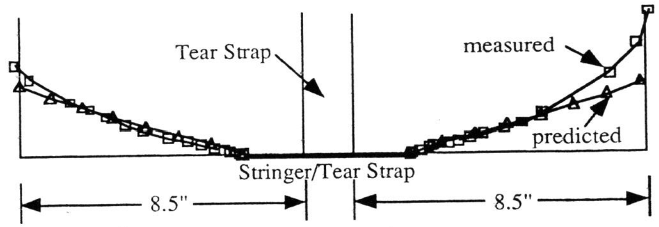

Fig 14. Comparison among computed and measured crack trajectories.

SUMMARY AND CONCLUSIONS

The software framework based on the described abstract model for crack propagation is ideally suited for modeling arbitrary crack growth in three-dimensional solid and shell structures. The use of a boundary representation to describe the structure, a topological database to store and manipulate the data, and the constrained hierarchy of models to simplify the discretization all serve to make the abstract model for crack growth simulations a versatile and powerful tool.

ACKNOWLEDGMENTS

Funding for the creation of FRANC3D has been provided in part by NASA (NAG-1-1184), NSF (PYI 8351914), Boeing, Northrup Grumman, and Schlumberger.

REFERENCES

Aliabadi, M.H. and Rooke, D.P. (1991) *Numerical Fracture Mechanics.* Kluwer Academic Publishers, The Netherlands.

Baehmann, P., Wittchen, S., Shephard, M.S. (1987) Robust geometrically based, automatic two-dimensional mesh generation, Int. J. Num. Meth. Engng., **24**, pp. 1043-1078.

Baumgart, B.G. (1975) A polyhedron representation for computer vision, AFIPS Proc., **44**, pp. 589-596.

Boone, T. J., Wawrzynek, P. A., and Ingraffea, A. R. (1987). Finite element modeling of fracture propagation in orthotropic materials. Engng. Fract. Mech., **26**(2), pp. 185-201.

Brogan, F. A., Rankin, C. C., and Cabiness, H. D. (1994). *STAGS User Manual Version 2.0. LMSC P032594*, Lockheed Palo Alto Research Laboratory, Palo Alto, CA.

Cavendish, J. C., Field, D. A., and Frey, W. H. (1985) An approach to automatic three-dimensional finite element mesh generation, Int. J. Num. Meth. Engng., **21**, pp. 329-347.

Finnigan, P.M., Kela, A., and Davis, J.E. (1989) Geometry as a basis for finite element automation, Engng. Comput., **5**, pp. 147-160.

Hoffmann, C.M. (1989) *Geometric and Solid Modeling: An Introduction.* Morgan Kaufmann Publishers, Inc., San Mateo, California.

Hui, C. Y., and Zehnder, A. T. (1993). A theory for the fracture of thin plates subjected to bending and twisting moments. Int. J. Fract., **61**, pp. 211-229.

Kela, A., Saxena, M., and Perucchio, R. (1987) A hierarchical structure for automatic meshing and adaptive FEM analysis, Engng. Comput., **4**, pp. 104-112.

Lo, S.H. (1989) Delaunay triangulation of non-convex planar domains, Int. J. Num. Meth. Engng., **28**, pp. 2695-2707.

Lutz, E.E (1991) *Numerical Methods for Hypersingular and Near-Singular Boundary Integrals in Fracture Mechanics*, Ph.D. Thesis, Cornell University.

Mäntylä, M. (1988) *An Introduction to Solid Modeling.* Computer Science Press, Rockville, Maryland.

Martha, L.F. (1989) *A Topological and Geometrical Modeling Approach to Numerical Discretization and Arbitrary Fracture Simulation in Three-Dimensions*, Ph.D. Thesis, Cornell University.

Martha, L.F., Wawrzynek, P.A. and Ingraffea, A.R. (1993) Arbitrary crack representation using solid modeling. Engineering with Computers, 9, 63-82.

Miller, M., Kaelber, K.N., and Worden, R.E. (1992) Finite element analyis of pressure vessel panels, in *Durability of Metal Aircraft Structures*, Proc. International Workshop on Structural Integrity of Aging Airplanes, Atlanta Technology Publications, Atlanta, GA.

Mortenson, M.E. (1985) *Geometric Modeling.* John Wiley & Sons, New York.

Potyondy, D.O. (1993) *A software framework for simulating curvilinear crack growth in pressurized thin shells.* PhD Thesis, Cornell University, Ithaca, NY.

Potyondy, D.O., Wawrzynek, P.A. and Ingraffea, A.R. (1995) An algorithm to generate quadrilateral or triangular element surface meshes in arbitrary domains with applications to crack propagation. Int. J. Numer. Meth. in Engng., **38**, pp.2677-2701.

Shephard, M.S. (1985) Finite element modeling within an integrated geometric modeling environment: Parts I and II, Engng. Comput., **1**, pp. 61-71.

Wawrzynek, P.A. (1986) *Interactive Finite Element Analysis of Fracture Processes: an Integrated Approach*, M.S. Thesis, Cornell Univ., Univ. Microfilms Intl., Ann Arbor, MI.

Wawrzynek, P.A. and Ingraffea, A.R. (1987a) Interactive finite element analysis of fracture processes: an integrated approach, Theor & Appl Fract Mech, **8**, pp. 137-150.

Wawrzynek, P.A. and Ingraffea, A.R. (1987b) An edge-based data structure for two-dimensional finite element analysis, Engrng with Comp, **3**, pp. 13-20.

Weiler, K. (1985) Edge-based data structures for solid modeling in curved-surface environments, IEEE Comp Graph & Appl, **5**, pp. 21-40.

Weiler, K. (1986) *Topological Structures for Geometric Modeling*, Ph.D. Dissertation, Rensselaer Polytechnic Institute, Troy, NY, Univ. Microfilms Intl., Ann Arbor, MI.