

# Advances in Simulation of Arbitrary 3D Crack Growth using FRANC3D/NG

P.A. Wawrzynek,<sup>1</sup> B.J. Carter,<sup>1</sup> and A.R. Ingraffea<sup>2</sup>

<sup>1</sup> *Fracture Analysis Consultants, Inc., Ithaca, NY, USA;*

<sup>2</sup> *Cornell University, Ithaca, NY, USA;*

## 1. Introduction

The FRacture ANalysis Code 3D / Next Generation (FRANC3D/NG or F3D/NG for short) is designed to simulate crack growth in engineering structures where the component geometry, local loading conditions, and the evolutionary crack geometry can be arbitrarily complex. It is designed to be used as a companion to a general purpose Finite Element (FE) package. Currently, interfaces to the ANSYS, ABAQUS, and NASTRAN commercial programs are supported.

F3D/NG is a successor to the original FRANC3D program (now referred to as FRANC3D/Classic), which was developed at Cornell University in the late 1980's. While the two codes share a name, the next generation code benefits from over 20 years of experience developing and using the Classic code. The NG version is a complete rewrite employing different approaches for geometrical modeling and deformation analysis.

The typical workflow for a F3D/NG analysis is shown in Figure 1. An analyst creates an uncracked FE mesh using the standard tools available for the commercial FE package. Typically, the analyst then defines a sub-model of the crack growth region. F3D/NG reads the sub-model mesh file and remeshes the sub-model to incorporate the geometry of a crack. The crack geometry and location can be prescribed either interactively using the Graphical User Interface (GUI) or programmatically using F3D/NG extensions to the Python programming language.

The "cracked" sub-model is reintegrated into the remainder of the model and an analysis is performed. The resulting displacements are read back into F3D/NG, which then computes Stress Intensity Factors (SIF's) for all node points along the crack front. The SIF's are used to predict the direction and relative amount of growth of the crack front points. The crack is extended, the sub-model remeshed, and another stress analysis is performed. This process is repeated for the number of crack steps specified by the analyst.

This paper is organized into five sections. The following section describes aspects of the crack growth modeling approach used within F3D/NG. The third section is a brief overview of F3D/NG's geometrical modeling and meshing pipeline. This is followed by an illustrative example and a brief summary.

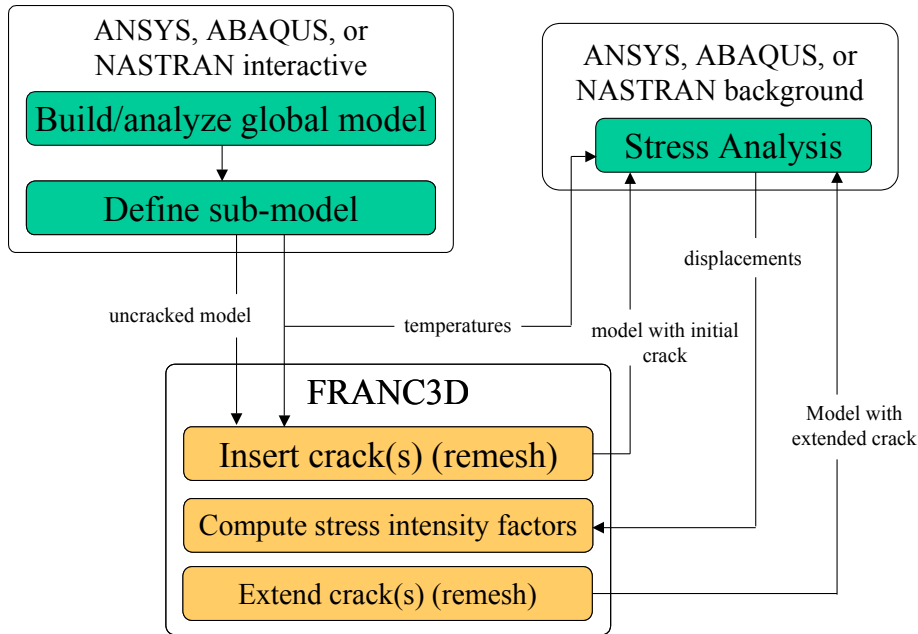


Figure 1. Typical workflow for a F3D/NG crack growth analysis.

## 2. Modeling Approach

In this section, key aspects of how F3D/NG operates and interacts with other programs are described.

### 2.1 Sub-models

F3D/NG supports the notion of a sub-model; remeshing for crack growth is confined to this sub-model. The sub-model approach is illustrated in Figure 2. All the supported FE packages have tools to define a sub-model.

In a "real world" analysis, the size of a crack is small relative to the size of the structure. Confining the remeshing for crack growth to the sub-model greatly reduces the amount of data that needs to be transferred to, and processed by, F3D/NG, thus speeding the crack growth process. It also allows the analyst to leave intact portions of a model with different structural idealizations (e.g. shell elements), complex boundary conditions (e.g., contact), or are just naturally and easily meshed with brick elements (F3D/NG remeshes with predominantly tetrahedral elements, as described below).

Sub-modeling is used for mesh modification only; it does not affect the analysis strategy. That is, the remeshed sub-model is "plugged" back into the global model and the stress and deformation analysis is performed for the full composite model. This approach is not a sub-structuring or local/global analysis

methodology. The sub-model can be redefined at any step of a crack growth analysis.

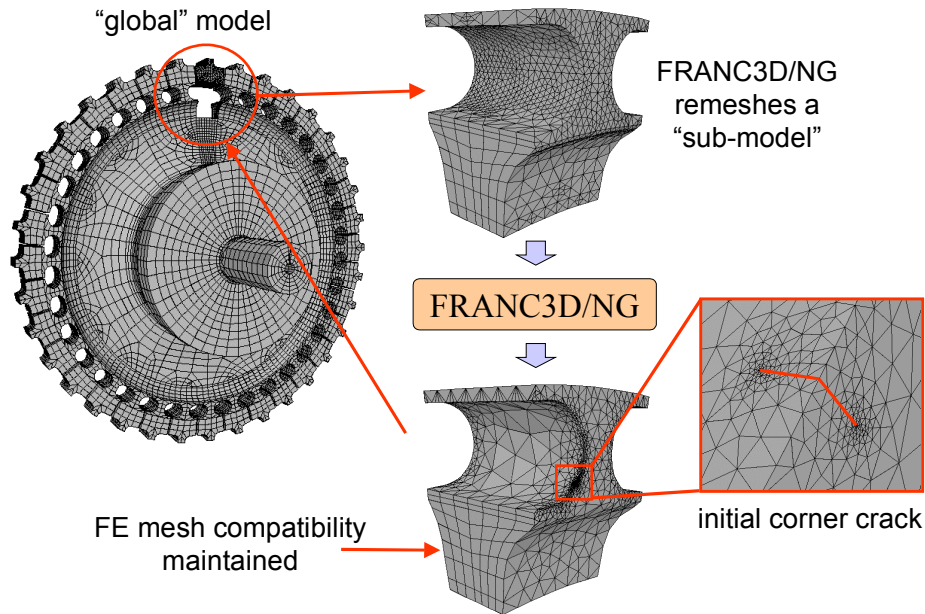


Figure 2. The sub-modeling approach used in F3D/NG.

## 2.2 Mesh File Interface

The main form of communication between F3D/NG and FE programs is through ASCII mesh description files. These are human-readable files that define a mesh, node coordinates, materials, and boundary conditions. These files use proprietary formats defined by the FE package vendors (.cdb, .inp, and .bnf formats for ANSYS, ABAQUS, and NASTRAN, respectively). F3D/NG expects a mesh file as input that describes an uncracked model or sub-model. It may also request a file of nodal temperatures in the case of thermal/mechanical loading, and possibly a file of nodal stresses if initial or residual stress fields are to be considered. The output from the program is a new file that describes a mesh for a model or sub-model containing a newly inserted or newly extended crack (and possibly a file of interpolated nodal temperatures).

Transferring a mesh description of a component between F3D/NG and a FE program is sub-optimal because mesh descriptions encode geometrical information incompletely. For example, if a portion of the surface of a component is curved, then the mesh model will have replaced that surface with a collection of planar or polynomial patches. This means that F3D/NG must use heuristic algorithms to reconstruct a description of the local geometry from the mesh data (a procedure described below). In most cases, the reconstructed geometry will be approximate.

In theory, a better approach would be for the FE package to send F3D/NG geometrical data with the mesh data. In practice, however, this would introduce unwanted complexities. The format of mesh files varies among vendors, but the main information they contain (node coordinates and element descriptions) is essentially the same among popular FE packages, and they describe a relatively simple data model. True geometrical information, in general, is much more complex and can vary markedly among vendors. Frequently, an analyst tasked with performing a crack growth investigation does not have easy access to a solid model description of a component and can more easily work with a mesh model.

On balance, even though exchanging FE data introduces geometrical approximations, doing so makes F3D/NG a more flexible and useful tool than if it were tied to the geometrical information demanded by a specific FE package.

### 2.3 Initial Flaw Geometry

F3D/NG can be used to insert both zero volume flaws (cracks) and finite volume flaws (voids) into a model. Both types of flaws are defined as a collection of triangular cubic Bézier spline patches. Using spline patches to describe flaws means that very complex, doubly curved crack surfaces can be modeled. In most cases, however, an analyst will start by inserting an initial flaw with a relatively simple geometry and have it grow into a more complex shape.

F3D/NG provides a "wizard" to specify an initial flaw shape, orientation, and location. One first selects from a small library of parameterized flaw shapes (e.g. elliptical crack, part-through crack, center crack, ellipsoidal flaw). Then one specifies translations and rotations to position the flaw.. The wizard provides visual feedback so that one can confirm the flaw is in the proper location. An image crack insertion wizard is shown in Figure 3.

Notice in figure 3 that the wizard allows the analyst to work with the full elliptical crack shape, even though only about half the ellipse will ultimately be inserted into the part. The program computes the intersection of the flaw with the (in this case, doubly curved) surface and trims the unused portion of the flaw.. This means the same parameterized elliptical crack model can be used to define a wide range of fully embedded, surface, or corner crack geometries.

### 2.4 Crack Region Meshing

A variety of element types is used within F3D/NG for meshing near cracks. As illustrated in figure 4, 15-nodes wedge elements are used adjacent to a crack front. By default, eight wedge elements are used circumferentially around the crack front and these elements have the appropriate side-nodes moved to the quarter points, which allows the element to reproduce the theoretical  $1/\sqrt{r}$  stress distribution.

The crack-front elements are surrounded by "rings" of 20-noded brick elements (two rings by default). Together, the wedge and brick elements comprise what is referred to as the crack front "template". The template is extruded along the crack front as shown in figure 5. This regular pattern of elements in the template is exploited when computing conservative integrals (e.g.,  $J$ -integral and  $M$ -integral).

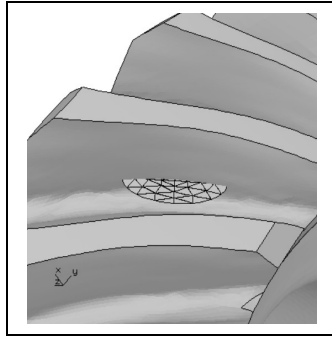


Figure 3. An image from the flaw insertion wizard showing an elliptically shaped crack being positioned in a gear tooth.

The bulk of the sub-model is meshed with 10-noded tetrahedral elements. The triangular faces of the tetrahedral elements are not compatible with the quadrilateral faces of the brick elements; 13-noded pyramid elements are used to transition from the template to the tetrahedra. Not all finite element packages support pyramid elements, so as an option the pyramid elements can be divided into two tetrahedra with the "hanging" node constrained.

#### 2.4 Computing Crack-front Parameters

F3D/NG computes stress intensity factors using either a displacement correlation approach or an  $M$ -integral. It can also compute elastic strain energy release rates by way of a  $J$ -integral.

In the displacement correlation approach, the finite-element-computed displacements for nodes on the crack faces are substituted into the theoretical expressions for the crack-front displacements fields written as functions of the stress-intensity factors.

For a linear-elastic analysis, the  $J$ -integral is equivalent to the total strain energy release rate. F3D/NG uses an equivalent domain formulation of the  $J$ -integral where the conventional contour formulation is replaced with a volume integral, which can be evaluated more accurately in a finite element context.

The  $M$ -integral, sometimes called the interaction integral, computes the energy release rates segregated by modes so the modal SIF's ( $K_I$ ,  $K_{II}$ , and  $K_{III}$ ) can be computed. F3D/NG has  $M$ -integral formulations for both isotropic and

orthotropic materials where the material axes can be oriented arbitrarily relative to the crack front [1].

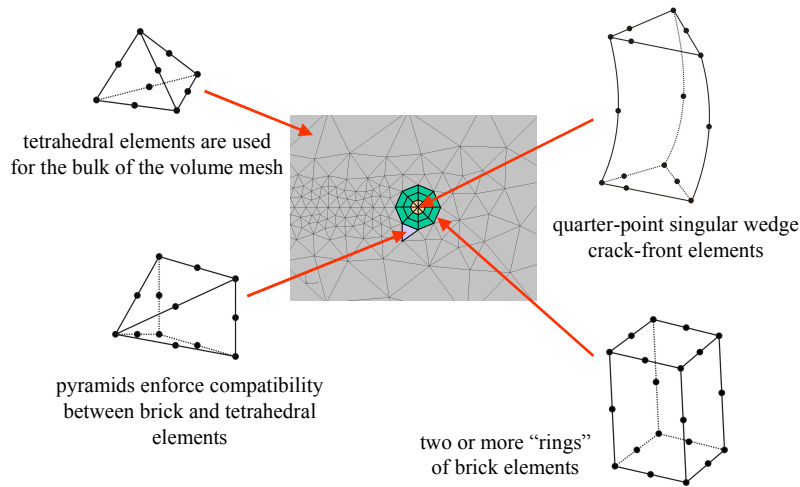


Figure 4. Element types used to mesh the crack-front region.

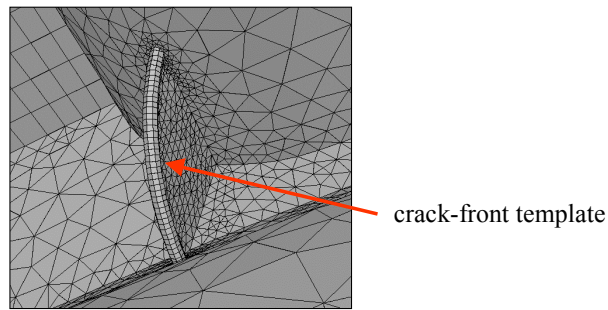


Figure 5. A crack-front template

### 2.5 Crack Growth

Within F3D/NG, crack growth is a five-step process:

1. SIF's are computed for all node points along the crack front.
2. At each such point the direction and extent of growth is determined.
3. A space curve is fit through the new crack-front points and, for the case of a surface crack, extrapolated, if necessary, to extend outside of the body.
4. New Bézier patches are added to the crack surfaces.
5. The extended crack is inserted into an uncracked mesh.

By default, the Maximum Tensile Stress (MTS) criterion is used to predict the local direction of crack growth. One can also select a transition to a Maximum Shear Stress (MSS) criterion when the ratio of Mode II to Mode I is above a defined threshold.

The relative amount of crack growth for points along the front, by default, is a ratio of the corresponding SIF's raised to an analyst specified power. This is analogous to evaluating the Paris crack growth rate equation for two points where both points are subjected to the same number of load cycles. In addition to a Paris type model, the NASGRO or an analyst supplied (piecewise linear in log/log space) crack growth rate equation can be used to determine relative amounts of growth.

### 3. The FRANC3D/NG Geometry/Meshing Pipeline

The core of the F3D/NG program is the geometrical modeling and meshing pipeline that adapts a FE mesh to insert a crack. This is a four-step process: surface facets and edge detection, geometry reconstruction, intersection computations, and meshing. These steps are described briefly in the following sections.

#### 3.1 Surface Facets and Edge Detection

As mentioned above, F3D/NG takes as input a mesh description of model. The first step in the pipeline is to determine the FE facets that either make up the outer boundary of the sub-model or fall on a bi-material interface. Depending on the solid element type, the surface facets will have either a triangular or quadrilateral shape. Only element corner nodes are considered. Quadrilateral element facets are further divided into two triangular facets yielding a planar-faceted, approximate geometrical description of the sub-model external and bi-material interface surfaces.

The next step is to determine collections of the surface facets that can be grouped to form planar or gently curving regions that can be logically seen as a single surface, figure 6. The logical surfaces are bounded by "hard" edges. These are patch boundaries that one would perceive as making up a portion of a "sharp" edge of the model.

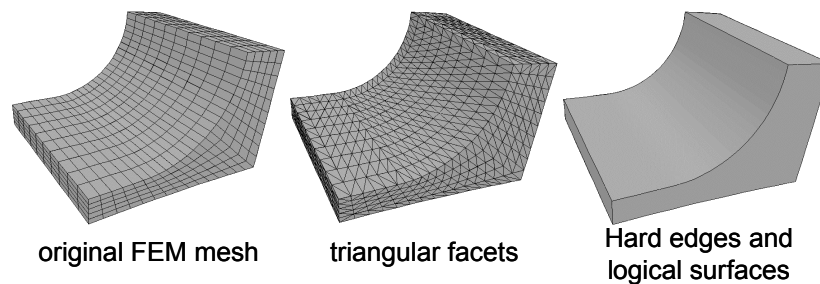


Figure 6. Preliminary surface and edge detection.

The primary heuristic used to identify sharp edges is to examine the angle between two adjacent surface patches. If the angle is below a threshold, the

common boundary segment is flagged as a hard edge. Once all edges have been identified, strings of adjacent edges are determined and grouped to form circuits that form boundaries of logical surfaces.

Exceptions to the patch grouping procedure are surfaces that fall on the interface between the sub-model and global model. These surfaces are retained as distinct patches so that nodal compatibility can be maintained when the sub-model is reinserted into the global mesh.

### 3.2 Geometry Reconstruction

The FE facets give an approximate description of the surface geometry of the sub-model. A more accurate, but still approximate, geometry description is obtained by replacing the planar triangular facets with triangular cubic Bézier patches, which can represent doubly curved surfaces.

These patches have ten control points so ten conditions must be given to define the geometry. Three conditions are given by the coordinates of the corners of the patches. Three more conditions are given by computing normals at all patch corners. Normals are computed as a weighted average of the planar facet normals for facets sharing the corner. The weights are given by the included patch angle at the corner point. If the corner falls on a hard edge, care is taken so that normals are not averaged across the edge. The final control point is selected to maintain quadratic precision of the surface interpolation [2].

### 3.3 Intersection Computations

At this stage of the pipeline, both the component and the crack are represented as collections of triangular cubic Bézier patches. To "insert" the crack into a component, a search is made to find all intersections between crack and component patches [3,4]. For intersected crack patches, those parts of the patch inside the component and those outside are determined. The external parts are "trimmed", and the internal parts are added to the component model to form a composite crack/component model. The intersection curves, which are the crack mouth, become additional hard edges in the model.

### 3.3 Meshing

The final stage in the pipeline is generating a FE mesh. Meshing is performed in four steps. First, FE nodes are generated along all hard edges. Second, triangular surface meshes are generated for all logical surfaces. Third, pyramid elements are generated adjacent to all quadrilateral boundary facets. Finally, a tetrahedral volume mesh is generated for the full sub-model.

Generating nodes along the hard edges is an iterative process. First, a characteristic element size is determined for the ends of all hard edges. The



characteristic size will be the smaller of the shortest of all the adjacent hard edge lengths and the size of the finite element that was at this location before crack insertion. Node distributions along the edges are then determined such that there will be a linear variation in element sizes from one end of the edge to the other. If the rate of change of nodal spacing along any edge is greater than a threshold, then the longer of characteristic lengths at this edge's endpoints is reduced until the spacing is within tolerance. As the edge endpoint will also be an endpoint for other hard edges, the size reduction will require a regeneration of nodes for these edges. This might possibly cause these edges to violate the spacing transition tolerance, and the process is repeated iteratively until the node distributions along all edges fall within the tolerance.

An advancing front meshing algorithm is used to generate surfaces meshes. One of two forms of this algorithm is used, depending on the geometry of the surface. If the surface is planar or nearly so, the surface boundary is mapped to a least squares fit plane, a 2-D mesh is generated in this plane, and the internally generated node points are mapped back so that they fall on the appropriate Bézier patches. If the surface has significant curvature, then a computationally more expensive advancing front procedure is used where all the computations and intersection checks are performed using the 3-D surface geometry [5]. An advancing front meshing algorithm is used for volume meshing also. In this case tetrahedral elements are generated [6].

#### 4. Example

For a simple demonstration analysis, a "half-penny" surface crack has been inserted into a thick-walled cylinder subjected to combined tension and torsion loading. The left image of figure 7 shows the FE mesh with a high concentration of elements near the crack mouth. The center image shows a detail of the mesh on the surface of the crack. The right image shows the crack surface after one step of crack growth. The crack extension is non-planar due to the mixed-mode loading.

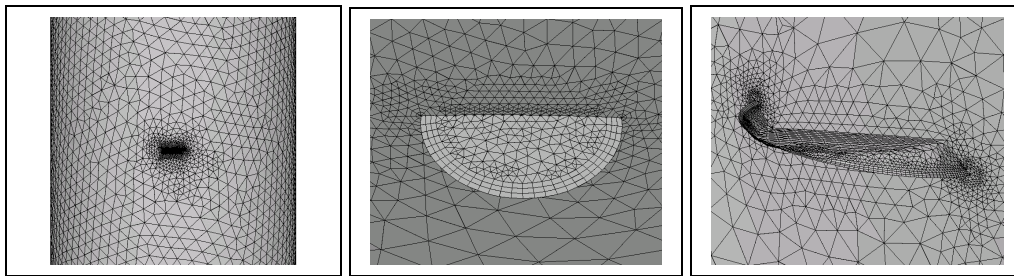


Figure 7. Crack growth in a thick-walled cylinder: initial mesh (left), initial crack surface mesh (middle), crack surface mesh after one step of growth (right).

Figure 8 shows a sequence of "cut-away" images illustrating the evolving crack geometry. The penultimate image shows that the crack front is automatically split

into two fronts as the crack grows to the inner bore. The final image shows the final trace of the crack mouth on the surface of the cylinder.

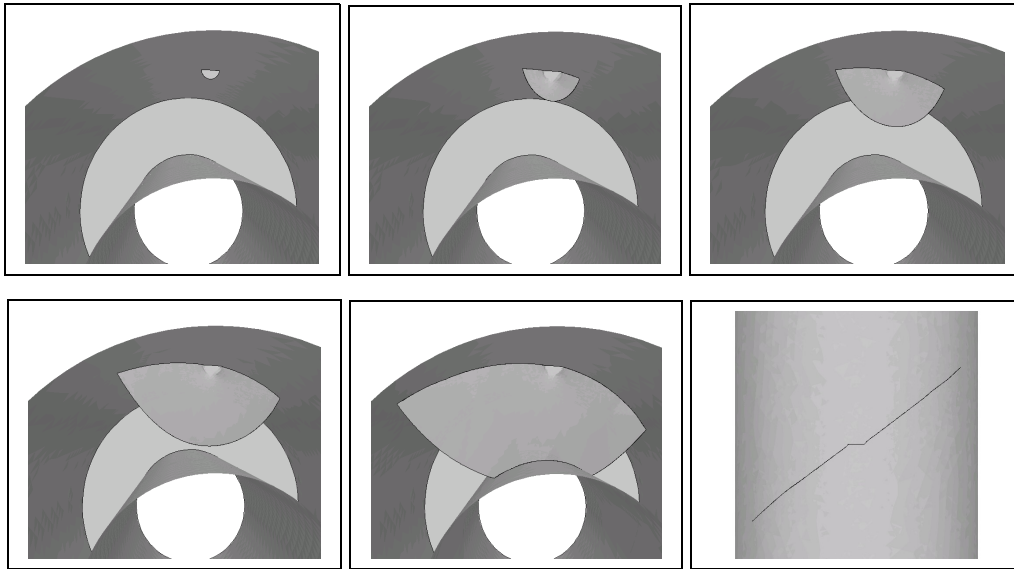


Figure 8. The geometry of the evolving non-planar crack geometry.

## 5. Summary

F3D/NG is a program designed to simulate crack growth in engineering structures where the component geometry, local loading conditions, and the evolutionary crack geometry can be arbitrarily complex. It is designed to be used as a companion program to a general purpose FE package. Because stress analysis is performed by capable commercial packages, complex mechanics, such as contact and advanced material models, can be included in crack growth analyses.

This paper has provided a brief overview of F3D/NG. Because of length restrictions many aspects of the program have been treated very superficially. Little mention has been made of the graphical user interface, which allows an analyst to be productive with a shallow learning curve. The paper does show, however, how geometrical modeling, computational fracture mechanics, and meshing capabilities have been combined to create a tool that provides an analyst with the capability to model realistically shaped cracks in real engineering structures subjected to realistic loads.

## Acknowledgements

We gratefully acknowledge that the development of FRANC3D/NG was partially supported by AFRL/RXLMN, contract numbers FA6650-04-C-5212 and FA8650-07-C-5216, Drs. Pat Golden and Reji John technical monitors, and by

NAVAIR, contract number N68335-08-C-0011, Mr. Ken Barlow technical monitor.

## References

- [1] L. Banks-Sills, P. Wawrzynek, B. Carter, A. Ingraffea, I. Hershkovitz, Methods for computing stress intensity factors in anisotropic geometries: Part II – arbitrary geometry, *Eng Fracture Mech* 74 (8)(2007) 1293-1307
- [2] G. Farin, *Curves and Surfaces for CAGD*, Morgan Kaufmann Publishers, San Francisco, 2002
- [3] G. Müllenheim, On determining start points for a surface/surface intersection algorithm, *Comp Aided Geometric Design* 8 (1991) 401-408
- [4] C. Bajaj, C. Hoffmann, R. Lynch, J. Hopcroft, Tracing surface intersections, *Comp Aided Geometric Design* 5 (1988) 285-307
- [5] A. Miranda, L. Martha, P. Wawrzynek, A. Ingraffea, Surface mesh regeneration considering curvatures, *Eng with Comp*, Submitted December, 2007
- [6] J. Cavalcante Neto, P. Wawrzynek, M. Carvalho, L. Martha, and A. Ingraffea, An algorithm for three-dimensional mesh generation for arbitrary regions with cracks, *Eng with Comp* 17 (2001) 75-91