An object oriented system for damage tolerance design of stiffened panels

N.K. Salgado\textsuperscript{a}, M.H. Aliabadi\textsuperscript{b,}\textsuperscript{*}

\textsuperscript{a} EMBRAER S.A., Brazil
\textsuperscript{b} Department of Engineering, Queen Mary and Westfield College, University of London, Mile End Road, London E1 4NS, UK

Abstract

This paper describes an interactive, integrated system for damage tolerance design of aircraft stiffened panels. The tasks supported by the system range from the initial definition of the stiffened panel configuration to the elaboration of the residual strength and crack growth diagrams, including the mesh design and the incremental crack extension analysis. The structural configuration is represented in an object oriented fashion, using a high level of abstraction and in terms of object components well-known to the designer, such as stiffeners, cracks and boundaries. Input data is performed interactively, through a user-friendly windows-based interface and the boundary element meshes are automatically designed using a rule-based strategy. Crack propagation analysis is performed with the dual boundary element method which allows for the solution of problems involving single or multiple cracks under mixed mode conditions. © 1998 Published by Elsevier Science Ltd. All rights reserved

1. Introduction

Aerospace components must adhere to airworthiness regulations and satisfy damage tolerance requirements \cite{1}. A structural component is damage tolerant if it can withstand reasonable loads without catastrophic failure or excessive deformation after the occurrence of serious fatigue damage. Fatigue crack growth analysis is central to damage tolerance assessment. It involves determining how cracks, that may be present since the very moment the aircraft is put into service, will propagate during the aircraft’s operational life.

Stiffened panels are metal sheets reinforced by stringers. They are widely used in the aerospace industry where light, highly resistant and damage tolerant structures are required. Fig. 1 illustrates a stiffened panel configuration where a sheet is reinforced by two stringers discretely attached to it by means of fasteners. The original role of stiffeners was to provide stability and static strength. However, with the introduction of damage tolerance design and the need to provide fail-safe structures, that role has been considerably extended \cite{2}. The stiffeners have a great deal of influence on the crack propagation life and on the residual strength of the panel, and design changes to the stiffeners and their attachments are quite often required to optimize the damage tolerance performance.

Airframe engineers must design high performance structures that comply with damage tolerance certification requirements and meet the demands for light weight and low operational cost in an increasingly competitive market. Finding the proper balance between these conflicting requirements is a difficult task.

Numerical crack growth analysis via the finite element method (FEM) or the boundary element method (BEM) has been used extensively to predict the damage tolerance characteristics of stiffened panels. Several numerical analyses, each simulating a different crack length, have to be performed. The important results from the analyses are the stress intensity factors and the maximum stress levels at the sheet, stiffener and stiffener attachments. The complete set of results is used to construct the residual strength diagram, which is basically a comparison of the stress level at which each component would fail, assuming all other components remain intact.

The residual strength diagram is used to assess the fail-safe character of the panel, e.g. should a crack reach a critical length and a skin fracture start, the load must be effectively transferred from the skin to the stiffeners and the crack arrested. The residual strength of the other components must remain above the critical level. Another important piece of information obtained from the residual strength diagram is the critical crack length which is used to set up an inspection programme to detect fatigue cracks before they become critical.
Regardless of the numerical technique selected, the mathematical model used in the analysis is a complex one. Due to the presence of a stress singularity, great care is required when modelling cracks. Special crack tip elements must be employed in the FEM. The discretization around the crack tips has to be fine and the meshes generated can require a large number of elements. Another source of complexity in the numerical model regards the stiffeners and their attachments. Accurate modelling of these elements is particularly important for obtaining reliable residual strength predictions.

Crack growth processes are simulated with an incremental crack extension analysis. For each increment, a stress analysis is performed, the stress intensity factors are evaluated and new crack tip positions are calculated. The discretization (often referred to as the mesh) then needs to be modified in order to reflect the changes to the crack tip positions. Except for simple cases where the crack trajectory can be predicted, re-meshing can be quite cumbersome, especially with the FEM.

Current damage tolerance design practice requires the use of many different software systems and fragmented sources of information. It is not unusual for an engineer to proceed in the following manner: gather design information using several database systems; use that information to create the numerical model; perform various stress analyses, changing the model manually to simulate different crack sizes; select the important results; import them (or even worse, type them) into a spreadsheet system; perform further calculations; and finally produce the residual strength diagram and the crack growth diagram.

The above procedure allows the designer to assess the damage tolerance behaviour of the structure. If it is not satisfactory, either because it does not comply with certification requirements or the component is too heavy or the inspection intervals are not long enough then some of the design variables have to be modified and the structure re-analysed. This cycle has to be repeated until an optimum design solution is found.

Damage tolerance design can be a time consuming, error prone and therefore expensive task. There is a clear need for higher productivity and quality in damage tolerance design. Design tools must be developed which allow engineers to concentrate on the real design problems instead of the complexities of the numerical analysis or the peculiarities of disparate computer systems.

This paper presents an object oriented computer system for damage tolerance design of stiffened panels. The system is referred to as I-DTD system. Its ultimate goal is to free the engineer from most of the tedious and error prone tasks, so that he or she can concentrate on the actual design problem and find optimum design solutions by experimenting with the design variables in a straightforward, intuitive manner.

2. System description

The ideas that led to the development of the I-DTD system are based on two fundamental concepts: interaction and integration.

Interaction, because design activities are interactive by nature. When designing structural components engineers seek answers to "what if?" type questions. They want to be able to modify the value of a design variable and observe the effect of that change in the structural response. However, the number of design cycles that can be executed is restricted in practice by the time taken for a single analysis to be performed. An interactive system for damage tolerance design would be impractical some years ago due to the computational cost of the crack propagation analysis. However, due to modern digital computers and state-of-the-art numerical techniques such as the recently developed dual boundary element method (DBEM) [3-4], interactive systems for fracture mechanics analysis are now feasible.

Integration, because it is an effective way of reducing the time and the possibility of errors in the analysis process. All distinct phases such as definition of the structural model; mesh design; crack propagation analysis (with implicit remeshing); and construction of the residual strength and crack growth diagrams should ideally be supported by the same computer system.

Modern software engineering concepts such as object orientation are the key for achieving the high level of integration and user friendliness required for the development of engineering systems that go beyond the usual "number crunching" and can therefore be regarded as real design tools.

2.1. Software development and object orientation

The software development process can be generally divided into three stages: analysis of the problem domain; system design; and system programming. The problem domain represents the real-world situation at which the software is aimed. Traditionally, the analysis of the problem domain is based on its decomposition into a series of functional modules. The functional modules receive some data, transform them and pass them on to other modules. In that
manner, input data are gradually transformed into output data. This approach, known as the procedural approach, is well suited for certain types of problem domains, such as algorithmic problems, which can be represented naturally by functional decomposition. The procedural approach, even if not used in a formal and systematic manner, is quite popular in the scientific community. The main programming language used in engineering codes is FORTRAN (formula translation language) which provides good support for implementation based on a procedural problem domain representation.

However, many problem domains are too complex to be naturally represented by functional decomposition. Let us take the example of a system aimed at damage tolerance design of stiffened panels. The tasks to be supported by such a system range from the initial definition of the stiffened panel configuration to the elaboration of the residual strength and crack growth diagrams, including the mesh design and the incremental crack extension analysis. While the last task is suitable for representation with a procedural approach, the same cannot be said of the other tasks. Therefore, for such a system, another representation paradigm, which can be used across the whole problem domain, needs to be adopted.

Object orientation is a new way of thinking about problems using models organized around real-world concepts. The fundamental construct is the object, which combines both data structure and behaviour in a single entity [5]. Software systems can be analysed and designed as a collection of such objects allowing for more challenging problem domains to be modelled and resulting in systems which are more stable in the long run. The object oriented approach focuses first on the identification of the objects of the problem domain and then on fitting procedures (methods) around them. Object oriented technology emphasizes what an object is rather than how it is used.

Most of the recent interest in object oriented systems is due to the growing need for interactive, user oriented systems. In such systems, a great deal of attention is devoted to the user interface. According to Coad and Yourdon [6], up to 75% of the code in a modern interactive system may be concerned with the user interface. The use of object oriented technology provides a practical, productive way to develop interactive, user oriented systems for complex problem domains.

2.2. System characteristics

In the I-DTD system, the structural configuration is represented in an object oriented fashion, using a high level of abstraction and in terms of object components well-known to the designer such as stiffeners, cracks and boundaries. Input data is performed interactively, through a user-friendly windows-based interface.

The system proposes an extensive use of pre-defined information stored in integrated databases. Whenever possible, the user is requested to select a value (or code) from a database instead of having to manually type the parameter. One obvious example of this application is to allow the user to input material properties by selecting a material code instead of having to produce and type the mechanical properties. This simple practice saves time and prevents the user from making mistakes. In the I-DTD system, this idea is extended to other design variables such as stiffener cross-sections, fasteners and adhesives. Every panel is associated with one project code. The project code is used to filter information, so that only options available for the panel’s project are presented for selection. Therefore, the possibility of mistakes, such as selecting a type of fastener not available for a particular project, is eliminated.

The I-DTD system provides some innovative capabilities. It is, for example, possible to Cut, Paste and Copy panel components such as stiffeners, cracks and boundaries in a manner similar to that frequently used in text processors and CAD systems. Such copy and paste operations can be performed between two different panels, creating a powerful tool to increase productivity and quality since panel components successfully used in previous studies can be re-used.

Once the panel is completely defined, the numerical model is automatically generated, analysed and the residual strength and crack growth diagrams calculated and displayed. The engineer can produce new designs in an intuitive, straightforward manner. New design variables can be selected from a range of alternatives available for the current project. Once new design variables have been selected, the model is updated, analysed and the new results displayed, so that they can be readily compared with the results from the previous design option.

Crack growth simulation is performed with the DBEM [4]. The BEM mesh design is very important as it affects both the accuracy of the results and the computational cost of the analysis (which, in the context of damage tolerance design, is even more relevant, because the analysis is performed several times). Moreover, mesh generation is itself time consuming, tedious and requires BEM expertise. The I-DTD system provides an automatic mesh design facility implemented in a knowledge-based fashion.

2.3. Implementation

From a software engineering point of view the I-DTD system can be classified as a hybrid system. Object orientation provides the framework for representing the objects of the structural configuration such as stiffeners and cracks and for the objects of the user’s graphical interface. Specialized tasks such as the DBEM analysis and the mesh generation are represented using alternative paradigms: procedural and rule-based, respectively. The system is implemented using the C++ programming language with the exception of the DBEM analysis, which is implemented with FORTRAN.

The DBEM analysis is performed with an extensively tested FORTRAN code. Although there are, undoubtedly,
benefits in designing scientific code in an object oriented manner [7-8], they do not seem attractive enough yet to compensate for the extra work and loss of run-time efficiency [9].

A large number of knowledge-based expert systems have been designed to infer upon knowledge represented as production rules. In the rule-based approach, production rules [10] are used to represent problem solving knowledge. A rule-based approach, combined with object oriented concepts, is used in the I-DTD system to create the BEM meshes.

In the following sections the I-DTD system is described in detail. The object model of the problem and implementation domains are explained. The rule-based strategy used to perform the DBEM mesh design is introduced and the basic ideas and equations of the DBEMethod are presented. Examples of damage tolerance design of stiffened panels are included in order to illustrate the functionality of the system.

2.4. The object model of the problem domain

The object class diagram captures the static structure of the system by showing the objects in the problem domain and the relationship between them. Concise object class diagrams are presented in the notation introduced by Rumbaugh et al. [5]. In what follows, a description of the problem domain is presented where names associated with object classes in the model are printed in bold type, the first time they appear in the text.

The most important object in the problem domain is the Panel. A panel has at least one Boundary and Material properties as illustrated in the object model presented in Fig. 2. Stiffeners and embedded cracks (EmbCrack) are optional panel Components. If a stress Analysis has been performed for the panel, it is associated with it.

The panel class has methods to deal with the insertion and removal of the components as well as to ensure that they are compatible among themselves. For example, an embedded crack must lie completely inside the external boundary and outside internal boundaries. Another important method provided by the Panel class is Mesh, which controls the generation of the boundary element mesh in a rule-based fashion.

The panel geometry is defined by a series of boundaries (Boundary). As shown in Fig. 3, boundaries are components formed by a collection of Segments, Rectangles, Circles, Polygons and PolySegs are special types of closed boundaries. A boundary can have edge cracks (EdgeCrack) associated with it. The edge crack cannot exist without the boundary. If the boundary is deleted, moved or copied, so are the associated edge cracks.

Segments can be Line or Arc segments. A line segment is defined by its two extreme Points, while an arc segment requires three points: start, end and centre. During the discretization process, the segments are sub-divided into smaller segments of the same type (line or arc), to form the BEM Mesh. Fig. 4 presents the related object class model. Each segment can have an associated Discretization and boundary conditions (BoundaryConditions). A discretization is defined by a series of points which subdivide the associated segment. Segment is an abstract class which provides, among many other things, the functionality required for its own sub-division into smaller segments.

Fig. 5 presents the object class model for the stiffeners and associated classes. Stiffeners are also panel components. They have their own boundary, which is made up of one or more co-linear segments. They have a cross-section (CrossSection) which has properties such as area and second moment of inertia, as well as material properties. The stiffener’s Attachment to the sheet can be of one of
three types: Integral, Continuous (bonded and integral stiffeners) or Discrete (by means of fasteners). Axial forces, shear forces and in-plane moments can be applied to the stiffener’s end points (EndLoads).

Cracks can be of two types: embedded cracks (EmbdCrack) or edge cracks (EdgeCrack) as shown in Fig. 6. The main difference between them is that an edge crack starts at a boundary while an embedded crack can exist without being associated with any boundary. Embedded cracks have two tips while edge cracks have only one tip. Cracks have their own boundary, formed of line segments.

A boundary element analysis can be performed based on a Model made of Nodes and Elements representing the discretized panel geometry. Fig. 7 presents the object class model for the analysis and related classes. Nodes are points with special identification attributes and quadratic elements are defined by three nodes. The Model class encapsulates the knowledge required to create the BEM model from the panel data. The analysis class has a method called Run, which invokes the execution of the DBEM simulation. If an analysis is successfully performed then it has Results associated. Crack propagation (CrackPropagation) is a special type of analysis. It requires more information about the fatigue crack growth properties of the panel material than required by normal stress analysis.

Boundaries, Cracks and Stiffeners are different objects; however, they share some important properties. They can all be added to and removed from the panel. They can also be edited to change properties or location. The component class, from which they are all derived, captures that commonality. A final comment about the object model is that aggregation seems to be the most common type of object association. Unfortunately, inheritance is the only type directly supported by C++ (and most object oriented languages).

2.5. The crack growth simulation task

Stress analysis and crack propagation analysis are implemented as methods provided by the Analysis and CrackPropagation classes, respectively. In the DBEM, the so-called displacement integral equation is used when collocating at one crack surface and the traction integral equation when collocating on the other crack surface. A comprehensive description of the application of the DBEM to the analysis of stiffened panels, including examples demonstrating the accuracy of the stress intensity factors calculated, can be found elsewhere [4-11]. A brief review is presented next.

2.5.1. DBEM formulation for problems involving stiffeners

The boundary integral displacement equation, for a source point \( \mathbf{x} \) at the boundary \( \Gamma \) of a finite sheet, subjected to a set of boundary tractions and displacement constraints and under the action of body forces \( b_j(\mathbf{X}) \) applied at field points \( \mathbf{X} \) continuously distributed over \( n \) straight lines \( L_n \) inside the domain is given by:

\[
c_{ij}(\mathbf{x})u_j(\mathbf{x}) + \int_{\Gamma} T_{ij}(\mathbf{x}', \mathbf{x})u_i(\mathbf{x})d\Gamma(\mathbf{x}) = \int_{\Gamma} U_{ij}(\mathbf{x}', \mathbf{x})d\Gamma'(\mathbf{x}) + \sum_{n} \int_{L_n} U_{ij}(\mathbf{x}', \mathbf{X})b_j(\mathbf{X})dL_n(\mathbf{X})
\]

where \( T_{ij}(\mathbf{x}', \mathbf{x}) \) and \( U_{ij}(\mathbf{x}', \mathbf{x}) \) are the Kelvin traction and

![Fig. 5. Object class diagram for Stiffener and related classes.](image)

![Fig. 6. Object class diagram for Crack and related classes.](image)
and $U_{ij}$ connection lines. These interaction forces can be treated in the and the stiffener will share interaction forces along the con-

of boundary loads and displacement constraints, the sheet

Cracked sheet, and this configuration is subjected to a set

collocating at one of the crack surfaces.

over the body forces loci are evaluated analytically. If a

elements and are evaluated numerically. The line integrals

values are in the form of integrals over the boundary

structed. The coefficients of the unknown nodal boundary

Eq. (1), a system of simultaneous linear equations is con-

tized into a set of elements. By taking the collocation point

movement considerations.

S ijk

where

source point

component of the unit outward normal to the boundary at the

is a coefficient that can be determined by rigid body

displacement fundamental solutions, respectively, $u_i(x)$ and $t_i(x)$ are displacements and tractions at boundary field points $x$, $\delta$ stands for principal value integral and $c_i$ is a coefficient that can be determined by rigid body movement considerations.

The corresponding traction boundary integral equation, presented below, can be obtained by differentiation of Eq. (1), application of Hooke’s law and multiplication by

Eq. (1), application of Hooke’s law and multiplication by

is the strain energy per unit volume, $n_1$ is the

relation:

while$ = \sigma_i n_i$) and $U_i$ are the components of the interior tractions and displacements and $m$ stands for mode I and mode II. The $J$-integral is related to the stress intensity factor; under plane stress conditions, the relationship is:

$$J^m = \frac{K^2}{E}$$

and $J^u = \frac{K^2}{E}$

2.5.3. Crack growth propagation

Crack growth directions are calculated using the prin-

Fig. 7. Object class diagram for Analysis and related classes.

displacement fundamental solutions, respectively, $u_i(x)$ and $t_i(x)$ are displacements and tractions at boundary field points $x$, $\delta$ stands for principal value integral and $c_i$ is a coefficient that can be determined by rigid body movement considerations.

The corresponding traction boundary integral equation, presented below, can be obtained by differentiation of Eq. (1), application of Hooke’s law and multiplication by

Eq. (1), application of Hooke’s law and multiplication by

is the strain energy per unit volume, $n_1$ is the

relation:

while$ = \sigma_i n_i$) and $U_i$ are the components of the interior tractions and displacements and $m$ stands for mode I and mode II. The $J$-integral is related to the stress intensity factor; under plane stress conditions, the relationship is:

$$J^m = \frac{K^2}{E}$$

and $J^u = \frac{K^2}{E}$

2.5.3. Crack growth propagation

Crack growth directions are calculated using the prin-

principal stress criteria [12], and re-meshing after each crack
increment is reduced to adding new elements to the crack boundary. No change is required in the existing mesh. Only matrix coefficients due to the new elements have to be calculated and added as new rows and columns to the existing (LU-decomposed) matrix. Only the new rows and columns have to be decomposed during the current increment and the computational time is substantially reduced. In multiple-site damage problems it is necessary, after each iteration, to determine not only the direction of the crack extension but also the relative sizes of the increments (i.e. the relative growth rates between the various crack tips). For the sake of simplicity, constant amplitude loading and the Paris propagation law [13] are used.

2.6. The mesh generation task

Generally speaking, generating a BEM mesh means choosing the size of the elements used in the discretization of each boundary segment. The mesh design is very important because it affects both the accuracy of the results and the computational cost of the analysis. As shown in the previous section, re-meshing after each crack increment in the DBEM is straightforward. Therefore, only the design of the initial mesh has to be addressed.

Connor [14] proposed a knowledge-based approach to design boundary element meshes for two-dimensional problems. In his work, the concept of continuity of mesh size at boundary points is combined with pre-defined solution patterns associated with boundary conditions and local geometric features, such as corners, notches and cut-outs, that require additional mesh refinement. Identifying the regions containing the geometric features was a key issue. Later, Portela et al. [15] extended that work using a similar strategy in a more object oriented implementation. The heuristics used for the mesh generation at sharp and round notches were substantiated with analysis results. However, in most practical cases, it is not necessary or desirable that every region containing boundary conditions or geometric features, such as the ones mentioned above, is discretized in a refined manner. In the case of crack problems, the single feature that really requires special mesh refinement is the crack tip.

The approach adopted here uses the same basic concepts of continuity of mesh size at boundary points and pre-defined solution patterns. Nevertheless, only the crack tip is assumed to require special discretization and, alternatively, the user has the possibility of specifying other boundary segments of particular interest. The mesh generation strategy is described by a set of production rules. An inference engine is used to apply those rules to the panel geometry in a manner that resembles a production system. The order in which the rules are placed in the rule set dictates their priority (i.e. the conflict resolution strategy is to fire the first rule that qualifies for firing).

Rule inferencing can be provided by a third party product such as CLIPS [16]. However, in order to keep the impedance between the object representation and the rule base low, it was decided to build an inference engine using object oriented concepts. An inference domain class library was developed. Its use allows objects of the problem domain to take part in the inferencing process. Object member functions can be invoked from within a rule. Only minor changes to the original object classes are required. A complete description of the mesh generation strategy and implementation can be found elsewhere [17].

2.7. The document/view model

The I-DTD system is implemented on the basis of the document/view model [18]. Fig. 8 presents the relevant object classes used in the implementation. According the document/view model, the application program (TApplication) has a document manager (TDocManager) associated. The document manager deals with user requests to open, close and save documents (TDocument). The user does not see or communicate directly with documents. Instead, each document has one or more views (TView) associated. Views are windows that display document information in a special manner. User requests to open and close views are also controlled by the program manager.

A panel document (PanelDoc) is a special type of document that implements additional functionality particular to the Panel class. The main panel document view displays the panel geometry (GeomView). Most interaction with the user is held through the geometry view. Boundaries, cracks, stiffeners, and points can be selected by clicking the left mouse button while pointing the cursor at the desired component. Selected components can then be edited, cut, deleted or copied. The Application class maintains the clipboard that is used as temporary storage in such operations. After the panel has been analysed, other views become available, such as the model view, deformed model view, stress intensity factors view (SIFView), crack growth diagram view and the residual strength diagram view (RSView).

![Fig. 8. Object class diagram for the document/view model as used in the I-DTD system.](image-url)
3. Applications

The concepts presented in the preceding sections will be demonstrated with the help of two examples analysed using a personal computer.

3.1. Stiffened panel with central crack

The first example comprises a typical stiffened panel configuration, containing a central crack and a broken central stringer, loaded with uniformly distributed tension applied in the direction of the stiffeners. All stiffeners are discretely attached to the sheet. The main steps required to generate the panel and analyse three different design alternatives will be illustrated.

After starting the system, the user must open a new panel document by selecting File/New in the main command menu. The system will then present the user with a list of available project names for selection. Fig. 9 presents the system main window at that stage. The system of units also has to be selected at this point. Two options are available in the current implementation: SI (MPa, mm) and US (ksi, in). After that, a window containing the empty geometry view for the panel will be created.

The definition of the panel must start with the external boundary. Any geometry made up of line and arc segments is supported. Rectangular and circular boundaries can be created directly; other types of geometries can be imported from pre-defined external text files. A rectangular boundary is created by selecting the command Sheet/External Boundary/Rectangle. A dialogue box will be displayed and the user must provide the boundary width, height and the coordinates of the origin. The geometry view will now display the panel with the external boundary. After the external boundary is in place, the sheet material code and thickness value can be selected.

A stiffener is created by invoking the Stiffener command. A dialogue box is displayed (see Fig. 10). The user must then define the start and the end point of the stiffener, select the attachment type and properties, select the cross-section and define the end loads. Fig. 11 shows the dialogue box used to input the cross-section properties of the stiffener. A list-box with pre-defined stringers and frames presented for selection can be seen in the figure.

After the first intact stiffener is successfully created, it can be selected (by placing the cursor at the stiffener locus and clicking the left mouse button), copied and pasted three times at the positions where the other intact stiffeners lie.

![Fig. 9. I-DTD system main screen with dialogue box for the creation of a new panel.](image)

![Fig. 10. Stiffener main input dialogue box.](image)
The central broken stiffener is created as two independent stiffeners.

The crack is introduced by clicking Crack/Embedded and providing the coordinates for the start and end point. Boundary conditions are associated with the boundary segments in a similarly simple manner. Fig. 12 presents the main system window with the geometry view after the panel has been defined and the mesh generated (by clicking Mesh/Generate). It would have taken an experienced user about a quarter of an hour to get to this point.

We are now ready to create the boundary element model and perform the crack propagation analysis. Due to the loading and geometry of the panel, the crack will propagate in a straight line. Fifteen crack increments are performed and once the analysis is complete, new views can be created, which allow the results to be studied. In Fig. 13 a view displaying the deformed model can be seen on the left of the main window while the residual strength diagram is on the right. The diagram presents normalized values of the residual strength against the number of the crack growth increment. The normalized critical value, under which any component would fail, is represented by the horizontal line (residual strength = 1.0). From the diagram we can see that the panel does not exhibit the required fail-safe character. A skin fracture starts, when the residual strength due to the stress intensity factors at the crack tips falls below the normalized critical level, at point A. We can see from the diagram that the fracture is not arrested.

The damage tolerant behaviour of stiffened panels relies on load being transferred from the skin to the stiffeners as the crack tips approach them. The stiffener attachment plays a crucial part in that process. One way of improving the efficiency of the attachment, is to reduce the distance between the fasteners. That can be done by performing a simple sequence of operations for each stiffener. Because discretely attached stiffeners must have nodes at the fastener positions (and the system knows that!) changing the fastener pitch will cause the existing mesh to be deleted. Having the new design solution re-meshed and analysed requires clicking a few more buttons. Fig. 14 presents the new residual strength diagram side-by-side with the old one. We can see (on the right) that the fail-safe character of the panel has been established. The skin fracture will stop at point B. We can also observe, by comparing the two solutions, that the stiffeners are taking a bigger share of the load. Unfortunately, the fastener curve is dangerously close to its limiting value near point B.

One way of improving the shear bearing capacity of the
attachment is to increase the fastener diameter. A new design solution can be generated in a similarly easy way. New fasteners can be selected from the database. Fig. 15 presents the residual strength diagram for the last solution compared with the prior one. The new solution (on the right) presents improved damage tolerance characteristics, as the fastener curve moved up. Note that the stiffener load has increased even more. The reason is that the new rivets are less flexible and consequently more efficient than the former ones.

3.2. Stiffened panel with cracked internal boundary

In this example the I-DTD system is used to analyse a
panel reinforced by two stringers and containing an oval inspection door with cracks emanating from its frame. The panel is subjected to tensile stresses, acting in the direction of the stiffeners, linearly distributed at the top and bottom edges and uniform shear stresses acting at the same edges. A schematic representation of the panel and its loads is presented in Fig. 16. Fatigue crack propagation is simulated assuming constant amplitude loading cycles with stress ratio equal to zero.

The panel representation is constructed in a manner similar to that presented in the previous example, except that the oval opening is imported from a library containing a selection of pre-defined cracked oval boundaries. The library is actually another panel document which can be opened simultaneously with the panel being created. The desired boundary is copied from the library panel to the I-DTD system clipboard and then pasted into the desired location in the new panel. Fig. 17 shows the system screen with the incomplete panel geometry and three other library panels containing typical boundaries.

The crack propagation analysis is more complex in this case than the previous example. Due to the presence of the shear forces in the panel, the cracks will not grow in straight lines. Furthermore, there are four different crack tips which can grow at different rates. Fig. 18 presents the I-DTD system screen displaying the deformed model and the crack growth paths.

Fig. 19 shows the crack growth diagram (crack lengths against the number of loading cycles) and the stress intensity factors (equivalent mode I stress intensity factor against the number of loading cycles). It can be observed that, although the ratio between the highest and lowest value for the stress intensity factors at the initial configuration is around 1.6, two of the cracks grow very little in comparison with the others. This is due to the exponential nature of the relation between the stress intensity factors and crack growth rates.

4. Conclusion

An object oriented integrated system for damage tolerance design of stiffened panels was presented. The system is user friendly, interactive and windows based. It supports all
the tasks required to perform residual strength assessment, from the object oriented definition of the panel to the residual strength diagram. Numerical crack growth simulation is performed using the DBEM. Data input is made as easy and error free as possible by using information from integrated databases and by re-using component objects from within the same panel or from other panels or libraries. The user is free to experiment with the design variables. They can be easily changed and new results are obtained in a matter of minutes, on an inexpensive personal computer. Optimum design solutions can be obtained from such experiments.

The system was designed predominantly in an object
oriented manner. Specific tasks are performed using alternative paradigms. Procedural paradigm is used for the crack growth simulation and rule inferencing is used to generate BEM meshes.

Object oriented design and implementation allows for the panel to be defined in terms of its real features instead of the BEM model features that represent them. The user can manipulate the panel components in a simple and intuitive manner using well-known mouse activated operations. The knowledge that allows the numerical model to be constructed from the panel objects is encapsulated in the panel objects themselves. The system provides an enhanced level of abstraction that makes it unique when compared with BEM or FEM systems with integrated pre- and post-processing facilities.

Acknowledgements

This work was sponsored by CNPq—Conselho Nacional de Desenvolvimento Científico e Tecnológico and EMBRAER—Empresa Brasileira de Aeronáutica S.A., Brazil.

References

boundary element mesh design. Engineering with Computers, in press.

