DOT/FAA/TC-17/15

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

Probabilistic Integrity and Risk Assessment of Turbine Engines

April 2018

Final Report

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

Technical Report Documentation Page

1. Report No.

DOT/FAA/TC-17/15

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

PROBABILISTIC INTEGRITY AND RISK ASSESSMENT OF TURBINE ENGINES

5. Report Date

April 2018

6. Performing Organization Code

7. Author(s)

R. Craig McClung[1]
Michael P. Enright[1]
Yi-Der Lee[1]
Jonathan P. Moody[1]
James C. Sobotka[1]
Vikram Bhamidipati[1]
Simeon H. K. Fitch[2]
Benjamin Guseman[2]
Jonathan P. Dubke[3]
Nikki Howard[4]
Michael E. McClure[5]
Alonso D. Peralta[6]

8. Performing Organization Report No.

18-16853

9. Performing Organization Name and Address

1 Southwest Research Institute®
6220 Culebra Road
San Antonio, TX 78238

2 Elder Research, Inc.
300 West Main Street #301
Charlottesville, VA 22903

3 Rolls-Royce Corporation
450 South Meridian Street
Indianapolis, IN 46225

4 GE Aviation
1 Neumann Way
Cincinnati, OH 45215

5 Pratt & Whitney
400 Main Street
East Hartford, CT 06108

6 Honeywell
111 South 34th Street
Phoenix, AZ 85072

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

11-G-009

12. Sponsoring Agency Name and Address

U.S. Department of Transportation
Federal Aviation Administration
Engine and Propeller Directorate
12 New England Executive Park
Burlington, MA 01803

13. Type of Report and Period Covered

Final Report
August 2011 – May 2016

14. Sponsoring Agency Code

AIR-6A1

15. Supplementary Notes

The FAA William J. Hughes Technical Center Aviation Research Division CORs were Joseph Wilson (retired) and David Galella.

16. Abstract

This grant supported the efforts of the FAA to develop an enhanced life management process, based on probabilistic damage tolerance principles, to address the threat of material or manufacturing anomalies in high-energy rotating components of aircraft engines. Major research products included automated methods for integrity and reliability assessment of production rotors using both 2-D and 3-D finite element models; completion of first-generation capabilities for high-temperature crack growth; methods for incorporating local residual stress effects; new capabilities for fleet risk computations; improved methods for fracture mechanics characterization of cracks in rotors; and multiple new versions of the DARWIN® computer code for technology transfer to industry and the FAA. The efforts facilitate implementation of official advisory material for circular holes and support evolving methods for surface damage analysis of attachment slots and turned surfaces, while also developing improved analysis methods for inherent anomalies in all materials.

17. Key Words

Aircraft gas turbine engines, Disks, Rotors, Low-cycle fatigue, Fracture mechanics, Probabilistic damage tolerance, Risk assessment, Surface damage, DARWIN

18. Distribution Statement

This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161. This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov.

19. Security Classif. (of this report)

Unclassified

20. Security Classif. (of this page)

Unclassified

21. No. of Pages

275

22. Price

ACKNOWLEDGEMENTS

Successful completion of this large, multidisciplinary research project would not have been possible without the significant contributions and assistance of many different people. Some are named as authors on this report, and many others deserve special mention.

Special appreciation is extended to the FAA employees and consultants who provided sustained support, encouragement, and guidance throughout the project. Heartfelt thanks go to Tim Mouzakis of the Engine and Propeller Directorate in the New England Regional Office, Joe Wilson and Dave Galella of the William J. Hughes Technical Center, and FAA consultant Jon Bartos.

The entire project was motivated and guided by the Rotor Integrity Steering Committee of the Aerospace Industries Association, led by current chair Nick Denton (GE Aviation) and previous chair Michael Gorelik (Honeywell).

Direct leadership was provided by the industry Steering Committee comprising representatives from the four partner companies: GE Aviation, Honeywell, Pratt & Whitney, and Rolls-Royce Corporation (RRC). Current members are chair Mike McClure (Pratt & Whitney), Brad Robinson (RRC), Brijesh Kumar (Honeywell), and Rajiv Sampath (GE Aviation). Previous members during the term of this program include former chair Jon Dubke (RRC), Alonso Peralta (Honeywell), and Bob Maffeo, Nikki Howard, and Jay Kelly (GE Aviation).

General DARWIN® development was led by Michael Enright of Southwest Research Institute®, (SwRI®), with substantial support from Vikram Bhamidipati (SwRI), James Sobotka (SwRI), and Wuwei Liang (formerly SwRI). Yi-Der Lee (SwRI) performed all development activities for the Flight_Life fracture mechanics module, including generating all stress intensity factor solutions. Jonathan Moody (SwRI) led DARWIN testing and validation activities with assistance from John McClure (SwRI). Simeon Fitch, Ben Guseman, Colin Thomas, Zach Brown, and Ben Hocking (all Elder Research) performed all graphical user interface development for DARWIN.

The partner companies provided extensive evaluation and testing of new DARWIN releases throughout the program. Key contributors included Mike Hartle, Matt Oakland, and Michael Krauss (GE Aviation); Alonso Peralta and Brijesh Kumar (Honeywell); Jon Dubke, Jason Baker, Justin Gilman, and Doug Olsen (RRC); and Mike McClure (Pratt & Whitney).

Mike McClure, Brett Ziegler, and Keith Kersey led the thermomechanical fatigue mission crack growth testing activity at Pratt & Whitney. Those tests were performed at Derivation Research Laboratory. Professor Jim Newman (Mississippi State University) performed the investigation of strip-yield modeling for load-time-temperature effects on crack growth. Youri Lenets (Honeywell) and Ross Miller (formerly Honeywell) led the small-crack testing activities. Jim Neumann was the project manager at Honeywell.

Birdie Matthews (SwRI) provided superlative clerical assistance in preparing this large final report. Gerry Leverant (SwRI consultant) is thanked for his careful peer review of the report.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

Figure Page

LIST OF TABLES

LIST OF ACRONYMS

AC	Advisory Circular
AFLCMC	Air Force Life Cycle Management Center
AFRL	Air Force Research Laboratory
ALA	As-large-as
AM	Additive manufacturing
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CF	Correction factor
CPCA	Compression precracking constant amplitude
CPLR	Compression precracking load-reduction
C(T)	Compact tension
CTOD	Crack tip opening displacements
DARWIN®	Design Assessment of Reliability With INspection
DEFORM™	Design Environment for FORMing
DOF	Degrees-of-freedom
DRL	Derivation Research Lab
EDM	Electro-discharge machined
EGO	Efficient Global Optimization
FASTRAN	FAtigue crack growth STructual ANalysis computer program
FCG	Fatigue crack growth
FE	Finite element
FEA	Finite element analysis
FEM	Finite element model
FIB	Focused ion beam
FRANC3D	FRacture ANalysis Code 3D
GP	Gaussian process
GUI	Graphical user interface
HA	Hard alpha
HDF5	Hierarchical Data Format
IR	Infrared
LEFM	Linear-elastic fracture mechanics
MicroFaVa	Micromechanical Fatigue Variability
MSU	Mississippi State University
NAVAIR	Naval Air Systems Command
NDE	Nondestructive evaluation
OEM	Original equipment manufacturer
PDRI	Probabilistic Design for Rotor Integrity
PIRATE	Probabilistic Integrity and Risk Assessment of Turbine Engines
POD	Probability of detection
PWF	Point weight function
R&D	Research and development
RAC	Risk Assessment Code
RISC	Rotor Integrity Steering Committee (originally Rotor Integrity Sub-Committee)
RRC	Rolls-Royce Corporation
RS	Residual stress(es)

SBIR	Small Business Innovative Research
SC(T)	Surface crack tension
SFTC	Scientific Forming Technologies Corporation
SIESTA	System for Integrated Engineering Structural and Thermal Analysis
SIF	Stress intensity factor
SwRI®	Southwest Research Institute®
TMF	Thermo-mechanical fatigue
TRMD	Turbine Rotor Material Design
USAF	United States Air Force
V&V	Verification and validation
WF	Weight function
XML	Extensible Markup Language
XRSIM	X-ray simulator

EXECUTIVE SUMMARY

The FAA is currently working with the aircraft engine industry to develop an enhanced life management process, based on probabilistic damage tolerance principles, to address the threat of material or manufacturing anomalies in high-energy rotating components. The multiyear Probabilistic Integrity and Risk Assessment of Turbine Engines (PIRATE) research grant supported this effort by developing enhanced predictive tool capability and supplementary material/anomaly behavior characterization and modeling. Grant activity included the development of methods to treat surface anomalies at attachment slots and on turned surfaces for all rotor materials, inherent material anomalies in all rotor materials, and issues in fleet risk assessment and continued airworthiness. Southwest Research Institute® led the effort; industry partners GE Aviation, Honeywell, Pratt & Whitney, and Rolls-Royce Corporation served as major subcontractors and Steering Committee; and Elder Research played a key subcontracting role. Major PIRATE accomplishments included the following:

- New Design Assessment of Reliability With INspection (DARWIN®) versions 7.2, 8.0, 8.1, 8.2, and 9.0 were released to the FAA and industry. Each new version was evaluated thoroughly by the development team, the project steering committee, and Rotor Integrity Steering Committee (RISC) before being released for production use.
- Eighteen commercial DARWIN licenses were active at the end of this grant, including 14 manufacturers of gas turbine engines for aircraft applications. Licensing revenue was used to fund additional enhancements of DARWIN and user support, including training workshops for licensees.
- A comprehensive 3-day DARWIN training workshop was conducted in April 2013. Forty-two people from 27 different organizations in the U.S. and six foreign countries attended in person, and an additional 59 people from 12 organizations participated remotely in the first day of the workshop through a web meeting. Seven other 2-day DARWIN training workshops were presented in the U.S. and three foreign countries for licensee companies and the U.S. Air Force Research Laboratory (AFRL). More than 100 people participated in those seven workshops.
- Growing interest in DARWIN among other U.S. government agencies led to a number of DARWIN spin-off projects funded by AFRL, Naval Air Systems Command, NASA, and individual DARWIN licensee companies. These projects typically result in significant new capabilities in the production DARWIN software delivered to all users.
- A new binary hierarchical input/output file format called Hierarchical Data Format (HDF5) was implemented in DARWIN that enables a reduction in the amount of computer memory required for execution. This enables DARWIN to support very large file sizes (file sizes are theoretically unlimited) without a substantial increase in memory use.

- Optimal autozoning algorithms for 2-D and 3-D finite element (FE) models were implemented in DARWIN to minimize the total number of automatically created zones that are required to compute the fracture risk of a component. The newest such algorithm uses Gaussian process pre-zoning to combine FEs with similar properties into groups called “pre-zones” for use in autozoning computations. The resulting algorithm is 1,000–10,000 times faster than a previous algorithm that created zones at individual FEs. The pre-zoning-based algorithm also requires less memory than previous algorithms (about 50% less memory for the example problems considered), enabling it to solve much larger models. The graphical user interface (GUI) was enhanced to enable the user to assign material properties and inspection parameters as property regions directly on the FE model to support autozoning life and fracture risk assessments.
- The DARWIN GUI was enhanced to enable the user to view non-hoop stress planes including maximum principal stress planes. The DARWIN auto-modeling algorithms (e.g., autoplate, life contours, and autozoning) were also enhanced to support non-hoop stress planes. The maximum principal stress plane is used as the default non-hoop stress plane.
- DARWIN was enhanced to facilitate the creation of surface and line zones to support RISC blade slot damage studies. Surface and line zones are defined via selection of element faces and edges, respectively. DARWIN was also enhanced to support anomaly distributions for line zones based on the number of anomalies per unit length.
- The DARWIN GUI was restructured to provide a centralized location for optional features and to accommodate additional new features. This information is stored in three new menu windows (i.e., “configuration,” “optional features,” and “global factors”) to enable efficient user input of general analysis parameters.
- A 64-bit version of the GUI was implemented in DARWIN to support memory-intensive operations such as the display and manipulation of 3-D FE models. The memory of the 64-bit GUI is no longer limited by the operating system, and it can be increased directly by the user via hardware improvements to meet future computational demands.
- DARWIN was enhanced to provide support for 3-D sector models. This new capability enables users to import and display 3-D FE sector models and to visualize a wireframe outline of the full component geometry. It enables a crack to extend beyond the sector boundary and into the full component model for use in life and risk assessments.
- DARWIN was enhanced to provide treatment for large FE models with large numbers of load steps. The HDF5 binary data structure was implemented for storage of FE data, enabling fast retrieval of selected portions of results files using significantly less memory compared to the previous format. A new user option to disable viewing of stress and temperature contours can further reduce the time required to import large FE models by a factor of 100 or more, depending on the size of the FE model.
- Full GUI support for cycle-dependent and time-dependent overload crack growth retardation models was added to DARWIN, allowing users to use optional modified Willenborg methods in life and risk calculations.
- A new thermo-mechanical fatigue crack growth option was added to DARWIN that enables crack propagation assessment based on material properties corresponding to the maximum temperature during the flight, in support of AC 33.70-2. Improved methods for temperature interpolation of cycle-dependent fatigue crack growth rates were also developed and implemented in DARWIN.

- Several new weight function (WF) stress intensity factor (SIF) solutions that accept bi-variant stress gradients were developed and implemented in DARWIN. The specific SIF solutions included an offset surface crack in the bore of an offset circular hole in a plate (SC29), a crack spanning a chamfered corner (CC12), and a crack at a corner with an angle of 135° (CC18). Improved univariant (SC30) and bivariant (SC31) WF SIF solutions for an offset surface crack in a plate were also developed and implemented.
- New capabilities were added to DARWIN to allow users to provide local residual stress gradients (e.g., due to surface enhancements such as peening) and combine them with fatigue stresses for 2-D or 3-D FE models using univariant SIF models.
- The DARWIN Fleet Assessment Module was enhanced to provide a new option for disk replacement following a simulated inspection event and to provide a new capability for corrective actions that enables replacement of existing disks with new disks that may have completely different geometries.
- A new DARWIN capability was developed to assess the risk associated with the introduction of new anomalies at shop visits, such as surface damage occurring at attachment slots during disassembly or assembly.. This includes treatment of cracks that may form sometime after an anomaly is initially introduced, and it also addresses the variability associated with nonzero crack formation lives.
- Experimental fatigue crack growth rate data were generated at 600°F for small fatigue cracks in Alloy 718. The small-crack data were compared with large-crack data generated previously under the same conditions and used to evaluate further an engineering model for small-crack effects. This test temperature was high enough to emulate realistic rotor service conditions but low enough (and the test frequency fast enough) to avoid time-dependent effects.
- Significant results from this research program and related spin-off projects were disseminated to a much broader audience in the gas turbine engine community or the international technical community through conference presentations or publications in archival technical journals. Twenty-three published papers (most of them presented at international conferences) and 19 additional presentations without manuscript occurred during the term of the current grant.

1. INTRODUCTION

1.1 BACKGROUND

The traditional design practice for high-energy aircraft gas turbine rotors, the so-called "safe-life" method, implicitly assumes that all material or manufacturing conditions that may influence the fatigue life of a rotor have been captured in laboratory coupon and full-scale component fatigue testing. In addition, the final design is usually based conservatively on minimum properties. This methodology provides a structured approach for design and life management that ensures high levels of safety. However, industry experience has shown that certain material and manufacturing anomalies can potentially degrade the structural integrity of high-energy rotors. These anomalies generally seldom occur and, therefore, are not typically present in laboratory test articles. On those rare occasions when anomalies are present in manufactured products in service, they represent a significant departure from the assumed nominal conditions and can result in incidents such as the Sioux City accident in 1989.

As a result of Sioux City, the FAA requested that industry, through the Aerospace Industries Association Rotor Integrity Sub-Committee (RISC) – since renamed the Rotor Integrity Steering Committee - review available techniques to determine whether a damage tolerance approach could be introduced to produce a reduction in the rate of uncontained rotor events. The industry working group concluded that additional enhancements to the conventional rotor life management methodology could be developed that explicitly addressed anomalous conditions. During the development of this probabilistic damage tolerance approach, it became apparent to RISC that the capabilities and effectiveness of the emerging technology could be significantly enhanced by further research and development (R&D). In early 1995, Southwest Research Institute® (SwRI®), in partnership with four major U.S. engine manufacturers and with guidance from RISC, proposed a multiyear R&D program and was awarded an FAA grant to address identified shortfalls in technology and data. This program, titled "Turbine Rotor Material Design" (TRMD), developed enhanced predictive tool capability and supplementary material/anomaly behavior characterization and modeling with a particular focus on hard alpha (HA) anomalies in titanium rotors.

One of the key outcomes of this work was a probabilistic damage tolerance computer code called Design Assessment of Reliability With INspection (DARWIN®). DARWIN integrates finite element models (FEMs) and stress analysis results, fracture mechanics models, material anomaly data, probability of anomaly detection, and uncertain inspection schedules with a user-friendly graphical user interface (GUI) to determine the probability-of-fracture of a rotor disk as a function of operating cycles with and without inspections. DARWIN was developed to provide an acceptable and publicly available (but not the only) means to comply with the damage tolerance requirements in Section 33.14 (now Section 33.70) of Title 14, Part 33 of the Code of Federal Regulations. The DARWIN version developed during TRMD received an R&D100 Award from R&D Magazine as one of the 100 most technologically significant new products of the year in 2000.

Other major accomplishments of the first grant included development of advanced probabilistic methods for risk assessment of components with rare inherent material anomalies [1], three versions of DARWIN [2], and verification and validation (V&V) of DARWIN against industry software and experience. Other accomplishments that supported ongoing RISC investigations included generation of fatigue crack growth (FCG) data in vacuum for three titanium rotor alloys [3], experimental and analytical characterization of the constitutive and damage properties of bulk titanium HA [4–5], experimental characterization of HA cracking in titanium alloy matrix material under monotonic and cyclic loading [6], and development of a forging microcode capable of predicting the fracture and change of location and shape of HA during reduction from ingot to billet and from billet to final forged shape, and forging experiments to validate the microcode. Further details are available in the TRMD Final Report [7].

An accident at Pensacola, Florida in 1996 called attention to surface anomalies induced by manufacturing activities. With guidance from the FAA, RISC began to apply and extend the insights and methods developed for inherent material anomalies in titanium rotors to the broader problem of induced surface anomalies in all rotor materials. SwRI, in continuing collaboration with the industry, proposed and was awarded a second FAA grant (i.e., TRMD—Phase II [TRMD-II]). This program began to address the surface anomaly challenge while completing the titanium HA work.

Major accomplishments of TRMD-II directly related to DARWIN included accurate new weight function (WF) stress intensity factor (SIF) solutions for select crack geometries under univariant and bivariant stressing [8–9]; advanced probabilistic methods to improve the efficiency and accuracy of risk computations [10–13]; a sophisticated GUI to facilitate analysis of fully 3-D models [8]; new DARWIN versions 4.x, 5.x, and 6.0 developed to implement these and other technology advances; V&V of each DARWIN version by comparison with engine company software and experience; and an infrastructure for formal software configuration management, code licensing, distribution, and user support, so that engine companies can employ DARWIN for official FAA and company purposes.

Additional supporting technology resulting from TRMD-II included:

- A mathematical model and computer code to describe the diffusion of nitrogen or oxygen in titanium from an inclusion during metal forming and heat treatment.
- Detailed nondestructive evaluation (NDE) and metallography of forgings with known HA anomalies to validate the HA forging microcode.
- Analytical characterization of the nitrogen contents, temperatures, strain rates, and orientations associated with cracking of HA anomalies during the forging operation.
- Experimental investigations of the effects of oxygen on tensile, fatigue, and dwell fatigue behavior of Ti-17.
- Experimental measurement of the coefficient of thermal expansion of bulk HA with different nitrogen contents.
- Evaluation of the potential effects of thermally-induced residual stresses (RS) on fatigue crack initiation and growth at HA inclusions [14].
- Spin pit tests and coupon fatigue tests, including periodic nondestructive inspection, along with post-test fractography and metallography, performed with material from the TRMD-Phase I forgings containing natural and synthetic HA anomalies.
- Vacuum FCG data for one titanium rotor alloy, two nickel-rotor alloys, and one powder metallurgy nickel alloy.

- Thermo-mechanical FCG data for IN-718 generated with diagnostic stress-temperature histories.
- A comprehensive literature survey on the stability and significance of RS in fatigue [15].

Further details are available in the TRMD-II Final Report [16].

Additional work on surface damage and inherent anomalies was performed in a third grant (i.e., "Probabilistic Design for Rotor Integrity" [PDRI]). Major accomplishments of PDRI included:

- New univariant and bivariant WF SIF solutions, including novel integration methods to substantially reduce execution time without reducing accuracy [17].
- A novel method to perform elastic-plastic shakedown calculations for bivariant fields.
- New software to translate commercial finite element (FE) models into DARWIN-compatible formats.
- New algorithms for thermo-mechanical FCG and time-dependent crack growth.
- New analysis mode to address surface damage on turned surfaces in 2-D models.
- New DARWIN capabilities that directly support certification calculations and reports for FAA advisory material on damage tolerance of hole features [18].
- Novel auto-modeling methods for inherent anomaly analysis in 2-D models that substantially reduce total analysis time while significantly reducing the potential for user error and user-to-user variability [19–21].
- Time-dependent FCG data for IN-718 used to evaluate predictive engineering methods.
- A comprehensive literature survey of small-crack effects in rotor alloys.
- New algorithms to calculate the service life associated with a user-specified probability of fracture.
- New capabilities to input user-supplied tabular SIF solutions.
- New DARWIN versions 6.1, 6.2, 7.0, 7.1, and 7.2 to implement these and other technology advances.
- V&V of each DARWIN version.

These accomplishments were documented in the PDRI Final Report [22].

The broad FAA/RISC vision for enhanced life management of high-energy rotors is summarized in figure 1. Here damage tolerance is a supplement to the existing safe-life methodologies. The FAA/RISC vision embraces both inherent anomalies introduced during production of the rotor materials and induced surface anomalies introduced during manufacturing or maintenance of the rotors themselves. All rotor materials are addressed—titanium alloys, conventional cast and wrought nickel alloys, and advanced (powder metallurgy) nickel alloys. The red check mark by Titanium HA indicates that the methods to address that threat have been developed and are now defined in FAA Advisory Circular (AC) 33.14-1 [23]. The check mark by circular holes indicates the completion of that methodology and the release of FAA AC 33.70-2 [24]. Current RISC activities are focused on attachment slots and turned surfaces. Some preparatory work is also underway to address inherent anomalies in nickel alloys.

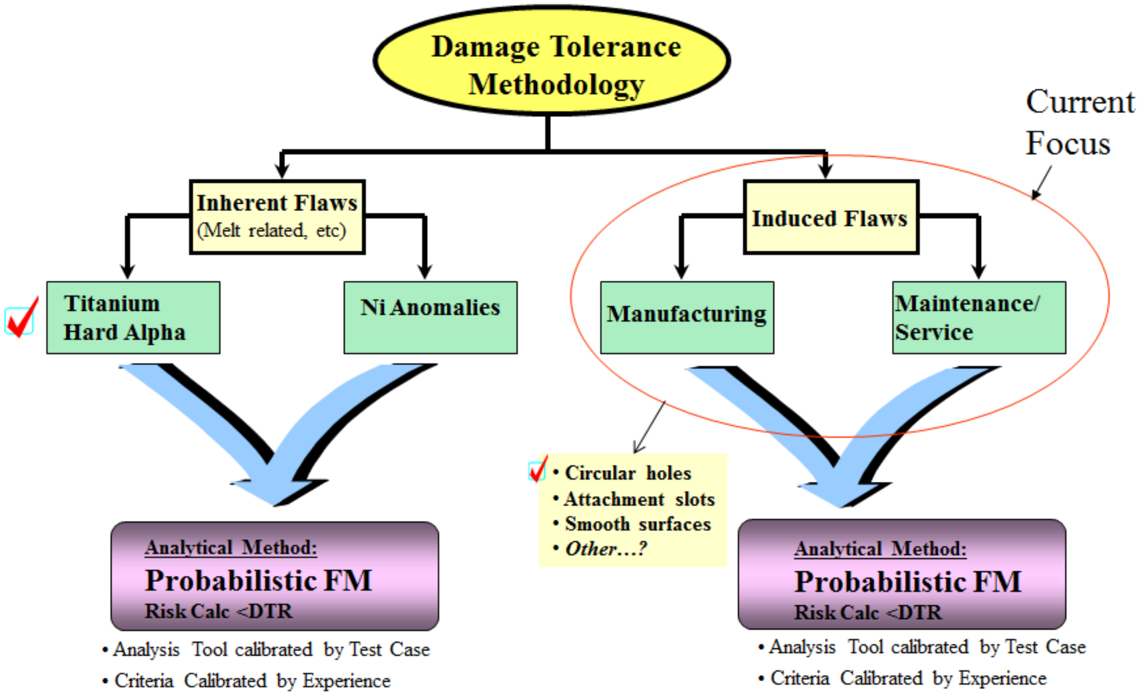


Figure 1. The FAA/RISC vision for an enhanced rotor life management process based on damage tolerance

The previous grants mirrored this incremental realization of the FAA/RISC vision. TRMD-I supported AC 33.14-1 for titanium HA anomalies. The primary focus of TRMD-II was the development and implementation of probabilistic damage tolerance methods for induced surface anomalies at circular holes. The PDRI grant continued to support the advisory material for circular holes and began developing methods to address surface damage at attachment slots and turned surfaces, while also developing enhanced methods for inherent anomalies in all materials.

The FAA has addressed assessment methods for continued airworthiness issues through AC 39-8 [25]. A DARWIN Fleet Assessment module that addresses some of the provisions of AC 39-8 was developed with funding from an engine company and is available to all DARWIN users.

A new grant, Probabilistic Integrity and Risk Assessment of Turbine Engines (PIRATE), was awarded in 2011 to continue this support of the FAA and the aircraft engine industry as they worked together to address the next steps in the comprehensive rotor-integrity vision described in figure 1. Three initiatives were pursued.

One initiative supported the continuing development of methods to treat surface anomalies at attachment slots and on turned surfaces for all rotor materials. This required attention to a number of topics, including elevated-temperature crack growth, near-surface RS, zoning of surfaces and edges in 3-D models, cracks at non-normal corners, in-service and maintenance-induced anomalies, and small cracks.

A second initiative supported the enhancement of methods to treat inherent material anomalies in all rotor materials. This initiative addressed methods for automatically generating fracture mechanics models, life contours, zones, and risk contours; the treatment of non-hoop maximum principal stresses; probabilistic methods with improved efficiency; and the use of 3-D models for analysis of inherent anomalies.

A third initiative expanded the scope of the project to address issues in fleet risk assessment and continued airworthiness. This initiative provided the capability for replacement of disks that are removed following an inspection or failure of a disk, or at user-specified intervals. The initiative also permits the application of corrective actions in the DARWIN fleet-assessment module.

SwRI led the effort, and industry partners GE Aviation, Honeywell, Pratt & Whitney, and Rolls-Royce Corporation served as both major subcontractors and Steering Committee. Elder Research played an important subcontracting role in support of DARWIN development. RISC continued to provide oversight and guidance.

This document is a comprehensive final report of the investigations conducted and results obtained under the PIRATE grant. The main body of the report is a summary of the major activities and key results from the project. Some additional details are contained in appendices.

1.2 ORGANIZATION OF RESEARCH

The PIRATE grant was organized in terms of technologies rather than applications, because some planned technology advances are applicable to multiple applications and because the investigating team is primarily organized along technological lines. Therefore, the project was expressed as eight major tasks described below:

1. Task 1: Advanced Capabilities for 2-D and 3-D FEMs–further-developed auto-modeling capabilities for 2-D (axisymmetric) FEMs; addressed cracks on non-hoop maximum principal stress planes in both 2-D and 3-D models; developed new capabilities for surface and line (edge) zoning in 3-D models, volumetric zoning of 3-D models for inherent anomaly analysis, and 3-D sector models; and developed new methods for the treatment of large FEMs with large numbers of load steps.
2. Task 2: High-Temperature Crack Growth Methods–completed the implementation, evaluation, and refinement of first-generation, high-temperature crack growth models in DARWIN, while exploring potential second-generation models.
3. Task 3: RS Methods–implemented tabular superposition methods for local RS gradients to be incorporated in univariant SIF solutions for both 2-D and 3-D FEMs.
4. Task 4: Fleet Risk and Probabilistic Methods–developed new capabilities for risk and reliability computations, including disk replacement, the application of corrective actions in the DARWIN fleet assessment module, and accommodation of new anomaly appearance at shop visits or in service.
5. Task 5: Advanced Fracture Analysis–focused on the development of new SIF solutions for common but complex surface geometries including angled and chamfered corners.

6. Task 6: Small Cracks–generated experimental FCG-rate data for small fatigue cracks in Alloy 718, a common rotor alloy for which no appropriate small-crack data were previously available. These data were used to evaluate further a previously proposed engineering model for small-crack effects.
7. Task 7: DARWIN Testing and Evaluation–included formal testing and evaluation of new DARWIN versions by both the primary developer (SwRI) and the engine companies that compared DARWIN against their own company codes, fleet experience, and test data.
8. Task 8: Technology Transfer–included all SwRI and engine-company activities associated with meetings, telecons, and written reporting, along with program management functions; publication and presentation of research results to the broader gas turbine engine and international technical communities; presentation of a DARWIN training workshop; and transfer of DARWIN and DARWIN technology to industrial users and other U.S. government agencies.

Brief summaries of DARWIN-related projects funded by other government agencies are provided in section 9.6 of this report as a courtesy.

2. ADVANCED CAPABILITIES FOR 2-D AND 3-D FEMS

FEMs provide the core input structure for life and risk assessments in DARWIN. They provide the geometry, stresses, and temperatures that are required for fatigue crack growth life prediction and associated risk assessments. The models continue to grow in size, and users have become accustomed to obtaining results quickly and viewing them directly on 2-D and 3-D FEMs.

DARWIN has evolved to support these expectations. New advancements in auto-modeling (see section 2.1) have enabled automated risk assessments of large FEMs that run approximately 100 times faster than previous exhaustive algorithms that applied a zone within each FE. DARWIN has been enhanced to provide treatment for specialized FE geometries and stress conditions required by RISC and the turbine engine industry, such as non-hoop maximum principal stress planes (see section 2.2), line and surface zoning of 3-D FE models (see section 2.3), and 3-D sector models (see section 2.4). Additional advancements in auto-modeling have enabled risk assessment of inherent anomalies in fully 3-D geometries that run at least 1000 times faster than exhaustive algorithms (see section 2.5). Efforts have also been made to improve computation speed and memory requirements for massive FEMs (see section 2.6).

2.1 FURTHER DEVELOPMENT OF 2-D AUTO-MODELING CAPABILITIES

A number of new capabilities were introduced in DARWIN to enable automated risk assessments. A new binary data format was implemented to reduce the computer memory required to support realistic engine disk models (see section 2.1.1). Computation times for automated risk assessment were dramatically reduced via the introduction of an optimal autozoning algorithm that was originally developed for uninspected components (see section 2.1.2) and later extended in include the effects of inspection (see section 2.1.3). The GUI was redesigned to accommodate additional capabilities and to help the user quickly locate them (see section 2.1.4). The GUI was upgraded to support 64-bit system architectures as described in section 2.1.5. The property regions concept that was originally developed for automated zoning was adapted for use in manual zoning (see section 2.1.6).

2.1.1 Hierarchical Data Format File Format

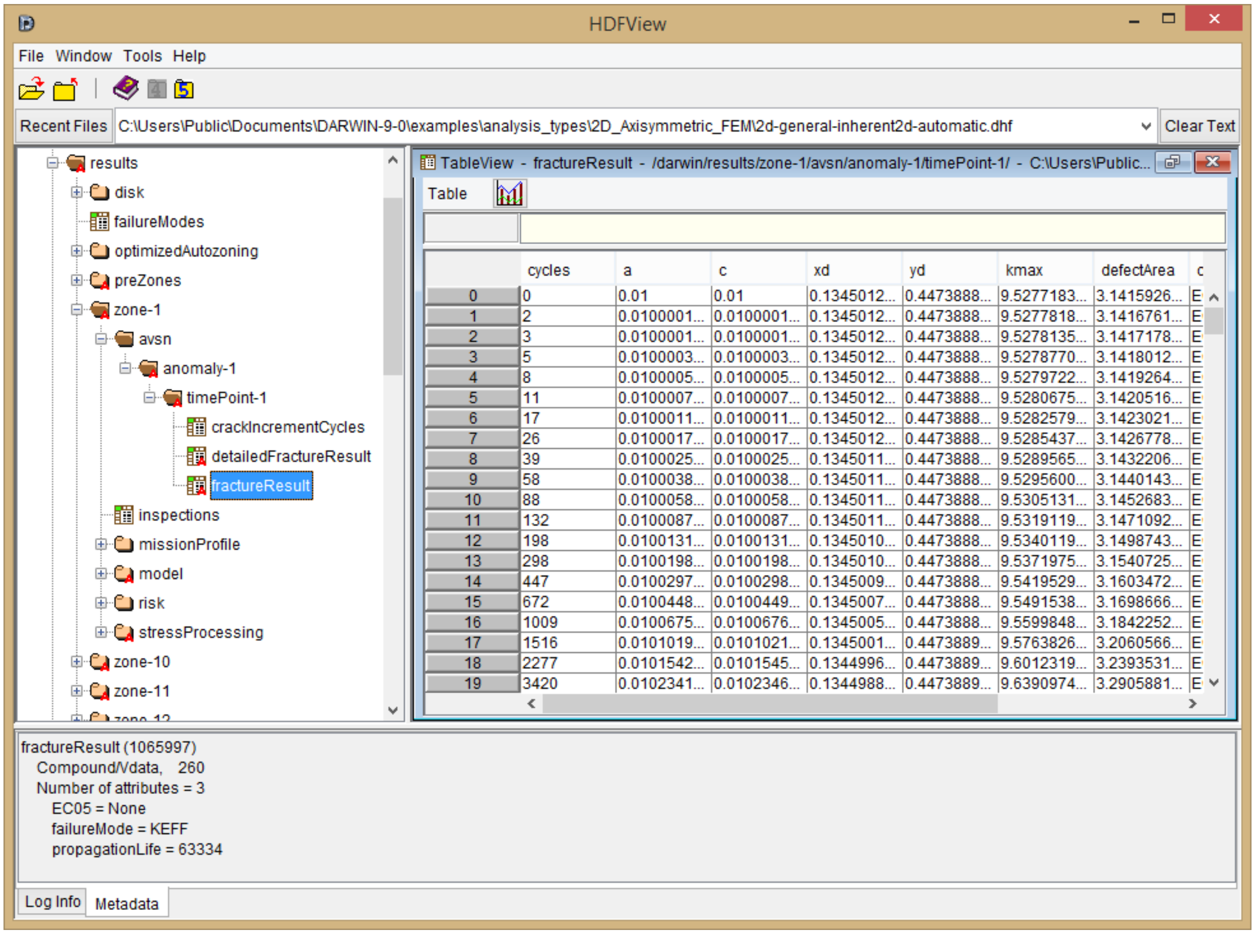
In previous versions of DARWIN, Extensible Markup Language (XML)-formatted files were used for storage of input and results data. This format was adequate for the small amount of data typically associated with routine risk assessments involving univariant crack types and small numbers of zones, FEs, and load steps. However, DARWIN capabilities have been expanded in recent years to include data-rich features such as bivariant crack types and autozoning. This development significantly increased file sizes for routine analysis. For example, the number of stress points required to model bivariant crack types is approximately three orders of magnitude greater than univariant types. Execution of the life-contours algorithm also requires computations at each FE node. In addition, as computer speeds increase, analysts have been developing FEMs with significantly more elements and load steps. All of these factors have contributed to increases in file sizes that have become difficult to manage using the XML file format in previous DARWIN versions.

One of the major drawbacks of the XML format was that it required the entire file to be loaded into memory even when only a small portion of the data is needed. Memory requirements for large files were substantial and could even exceed the available memory. Another drawback of the XML file format was that floating-point numbers were stored in American Standard Code for Information Interchange (ASCII) format in files, whereas they are stored in binary format in memory. This required a conversion procedure that not only required more computing power but also caused precision losses that were dependent on the number of significant digits used when stored in ASCII format. These precision losses lead to discrepancies between the GUI and Risk Assessment Code (RAC), particularly for computational-geometry results.

To resolve these issues, a file format called Hierarchical Data Format (HDF5) was implemented in DARWIN 8.0. HDF5 is a binary hierarchical file format that was specifically designed for complex high-volume data. It supports direct random access to specific locations within a file without the need to load the entire file into memory, which significantly reduces memory requirements. It supports a number of data types in binary format (including floating-point numbers), eliminating the inefficiency and errors associated with the conversion between binary and text formats. It supports multiple languages, including C, C++, FORTRAN, JAVA, PYTHON, and MATLAB. It is a mature technology with more than 25 years of development history, and it is actively developed and maintained by the HDF Group at the University of Illinois at Urbana-Champaign.

HDF5 organizes data in a tree-like, hierarchical structure that is similar to the data structure used for XML-formatted data (see figure 2). This structure makes it convenient to navigate large files with complex data. Each node of the structure is defined as either a group or dataset. A group is similar to a folder in a file system, which may contain subgroups or datasets. A dataset consists of data in a tabular form, which is essentially a single or multidimensional array. Both groups and datasets can include attributes of any supported data type such as integer, floating-point numbers, strings, and arrays. HDF5 files can be easily viewed and edited using a free, open-source tool called HDFView, which provides capabilities to add, delete, copy, and paste data. The DARWIN

GUI was recently enhanced to include the HDF5 viewer for display and editing of HDF5-formatted files. The tabular data in HDFView are compatible with Microsoft® Excel®, which enables the analyst to directly copy tabular data into Excel spreadsheets for post-processing.



C:\Users\Public\Documents\DARWIN-9-0\examples\analysis_types\2D_Axisymmetric_FEM\2d-general-inherent2d-automatic.dhf

TableView - fractureResult - /darwin/results/zone-1/avsn/anomaly-1/timePoint-1/ - C:\Users\Public...

	cycles	a	c	xd	yd	kmax	defectArea	c
0	0	0.01	0.01	0.1345012...	0.4473888...	9.5277183...	3.1415926...	E
1	2	0.0100001...	0.0100001...	0.1345012...	0.4473888...	9.5277818...	3.1416761...	E
2	3	0.0100001...	0.0100001...	0.1345012...	0.4473888...	9.5278135...	3.1417178...	E
3	5	0.0100003...	0.0100003...	0.1345012...	0.4473888...	9.5278770...	3.1418012...	E
4	8	0.0100005...	0.0100005...	0.1345012...	0.4473888...	9.5279722...	3.1419264...	E
5	11	0.0100007...	0.0100007...	0.1345012...	0.4473888...	9.5280675...	3.1420516...	E
6	17	0.0100011...	0.0100011...	0.1345012...	0.4473888...	9.5282579...	3.1423021...	E
7	26	0.0100017...	0.0100017...	0.1345012...	0.4473888...	9.5285437...	3.1426778...	E
8	39	0.0100025...	0.0100025...	0.1345011...	0.4473888...	9.5289565...	3.1432206...	E
9	58	0.0100038...	0.0100038...	0.1345011...	0.4473888...	9.5295600...	3.1440143...	E
10	88	0.0100058...	0.0100058...	0.1345011...	0.4473888...	9.5305131...	3.1452683...	E
11	132	0.0100087...	0.0100087...	0.1345011...	0.4473888...	9.5319119...	3.1471092...	E
12	198	0.0100131...	0.0100131...	0.1345010...	0.4473888...	9.5340119...	3.1498743...	E
13	298	0.0100198...	0.0100198...	0.1345010...	0.4473888...	9.5371975...	3.1540725...	E
14	447	0.0100297...	0.0100298...	0.1345009...	0.4473888...	9.5419529...	3.1603472...	E
15	672	0.0100448...	0.0100449...	0.1345007...	0.4473888...	9.5491538...	3.1698666...	E
16	1009	0.0100675...	0.0100676...	0.1345005...	0.4473888...	9.5599848...	3.1842252...	E
17	1516	0.0101019...	0.0101021...	0.1345001...	0.4473889...	9.5763826...	3.2060566...	E
18	2277	0.0101542...	0.0101545...	0.1344996...	0.4473889...	9.6012319...	3.2393531...	E
19	3420	0.0102341...	0.0102346...	0.1344988...	0.4473889...	9.6390974...	3.2905881...	E

```
fractureResult (1065997)
  Compound/Vdata,   260
  Number of attributes = 3
    EC05 = None
    failureMode = KEFF
    propagationLife = 63334
```

Figure 2. Illustration of hierarchal data structure in HDF5

The implementation of the HDF5 file format in DARWIN has enabled a reduction in the amount of computer memory required for execution, particularly for large files. To fully realize the memory reduction, future enhancements are planned to retrieve data from the HDF5 file on an as-needed basis. When the HDF5 format is used, memory requirements are based on the amount of data that are used rather than the amount of data that are present in the input files. Therefore, it is anticipated that DARWIN will be able to support even larger file sizes in the future without a substantial increase in memory use. File sizes are theoretically unlimited, but the hardware memory required to store them is finite. In general, file sizes actually become smaller when the HDF5 format is used due to the relatively compact binary format associated with HDF5.

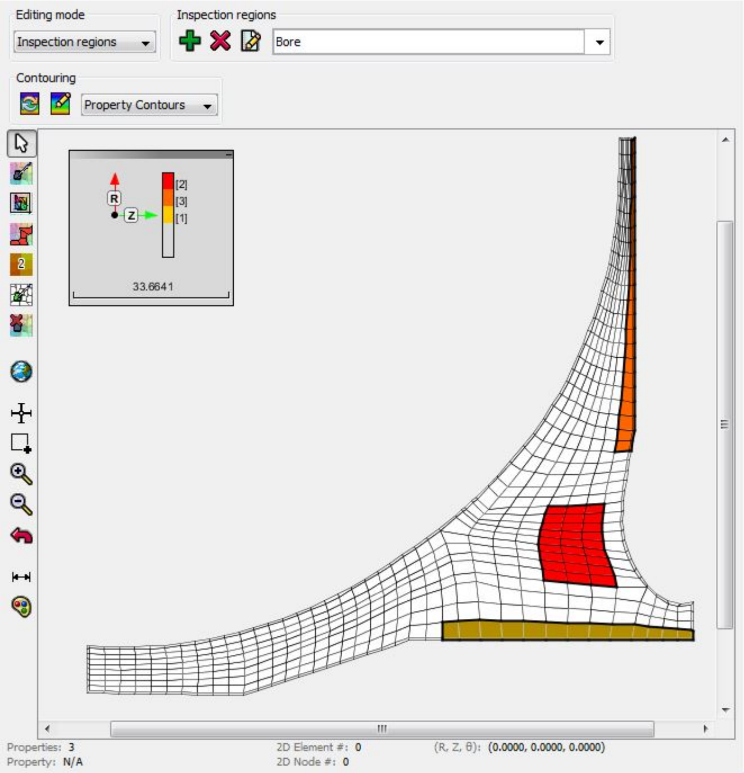
The HDF5 data format provides additional capabilities that are applicable to DARWIN. For example, the random data access capability associated with HDF5 enables DARWIN to read only the required data such as effective load steps, nodes, and elements. Because file size does not affect memory use, HDF5 enables consolidation of multiple DARWIN analysis files (e.g.,*.DAR, *.DDB, *.UIF, *.UOF) into a single file without a significant increase in memory or execution time. HDF5 also allows files to be written and read simultaneously, providing potential improvements in both computational speed and memory usage.

2.1.2 Optimal Autozoning

DARWIN Version 7.2 included an automatic zone-creation algorithm in which zones were assigned to each FE in a component model. This provided a robust risk solution, but the process required significant computer time. An optimal autozoning algorithm was implemented in DARWIN 8.0 that minimized the total number of automatically created zones that are required to compute the fracture risk of a component. The algorithm uses a relatively coarse response surface method to estimate the conditional risk value at each node in a FE model. Zones are defined for each FE in the model, and the algorithm identifies and merges zones based on their contribution to component risk. The process continues until all of the zones have been merged into a single zone. The resulting zone sequence is applied in reverse order to define the optimal allocation of FEs to zones. Testing confirmed that the number of zones (and associated computation times) can be dramatically reduced using the new optimal autozoning algorithm. This feature did not provide treatment for in-service inspections (this was addressed in DARWIN 8.1, see section 2.1.3). Further details regarding the approach are provided in appendix A.

2.1.3 Optimal Autozoning with Inspection

In DARWIN 8.0 (and previous versions), users specified inspections for zones by manually assigning inspection schedules to the interior and edges of manually-created rectangular plate fracture mechanics models. This approach was not supported by the autozoning algorithm described in the previous section because it required that the fracture mechanics model be available prior to execution of DARWIN. A new approach was developed in which the user defines multiple inspection regions directly on the FE model geometry rather than on the fracture mechanics model (see figure 3). To support autozoning, an algorithm was implemented in DARWIN 8.1 that automatically transfers the inspection events from the model geometry to fracture mechanics models that are created by the autoplate algorithm.



Editing mode: Inspection regions

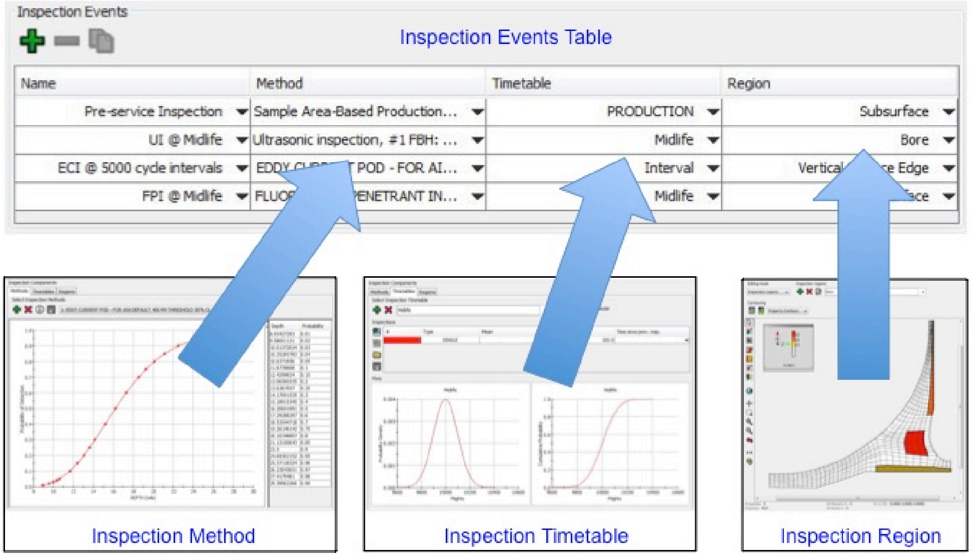
Inspection regions: Bore

Contouring: Property Contours

Properties: 3 — Property: N/A — 2D Element #: 0 — 2D Node #: 0 — (R, Z, θ): (0.0000, 0.0000, 0.0000)

Figure 3. Example of inspection regions defined on an FEM

Definition of inspections was consolidated into a single Inspection Events menu (see figure 4) in which the user specifies inspection methods (probability of detection [POD] curves), timetables (schedules) and regions (geometry). The menu is automatically hidden from view when inspections are not in use. This capability is used to define inspections for zones that are created both manually and via autozoning.



Inspection Events

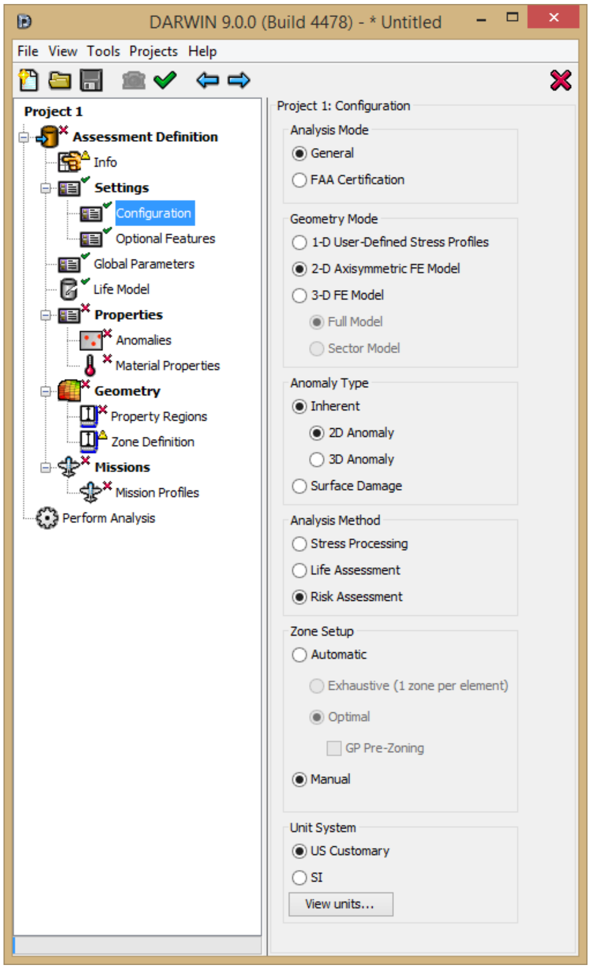
Name	Method	Timetable	Region
Pre-service Inspection	Sample Area-Based Production…	PRODUCTION	Subsurface
UI @ Midlife	Ultrasonic inspection, #1 FBH: …	Midlife	Bore
ECI @ 5000 cycle intervals	EDDY C[illegible]T POD - FOR AI…	Interval	Vertical [illegible]e Edge
FPI @ Midlife	FLUO[illegible]ENETRANT IN…	Midlife	[illegible]ce

Figure 4. Consolidation of inspection definition into a single menu

2.1.4 GUI Restructuring to Accommodate New Features

In DARWIN 8.1 (and previous versions), the user had to specify general analysis parameters via a single menu window. The GUI contains a number of new features, and the associated general parameters did not fit in a single menu window. In addition, several general parameters were located in various submenu windows or were otherwise hidden from view. To address these issues, the GUI was restructured in DARWIN 8.2 to provide three new menu windows (i.e., Configuration, Optional Features, and Global Factors) for user input of general analysis parameters.

The Configuration menu window (see figure 5) contains high-level project options that affect the availability of features that are provided in other menu windows. The Optional Features menu window (see figure 6) contains activation buttons for optional features (i.e., capabilities that are not required for an analysis). When an optional feature is disabled, the input fields associated with it are hidden or disabled, which greatly simplifies the data entry task. The Global Factors menu window (see figure 7) provides a single location for input parameters that apply to the entire component.



Project 1: Configuration

Analysis Mode
● General
○ FAA Certification

Geometry Mode
○ 1-D User-Defined Stress Profiles
● 2-D Axisymmetric FE Model
○ 3-D FE Model
● Full Model
○ Sector Model

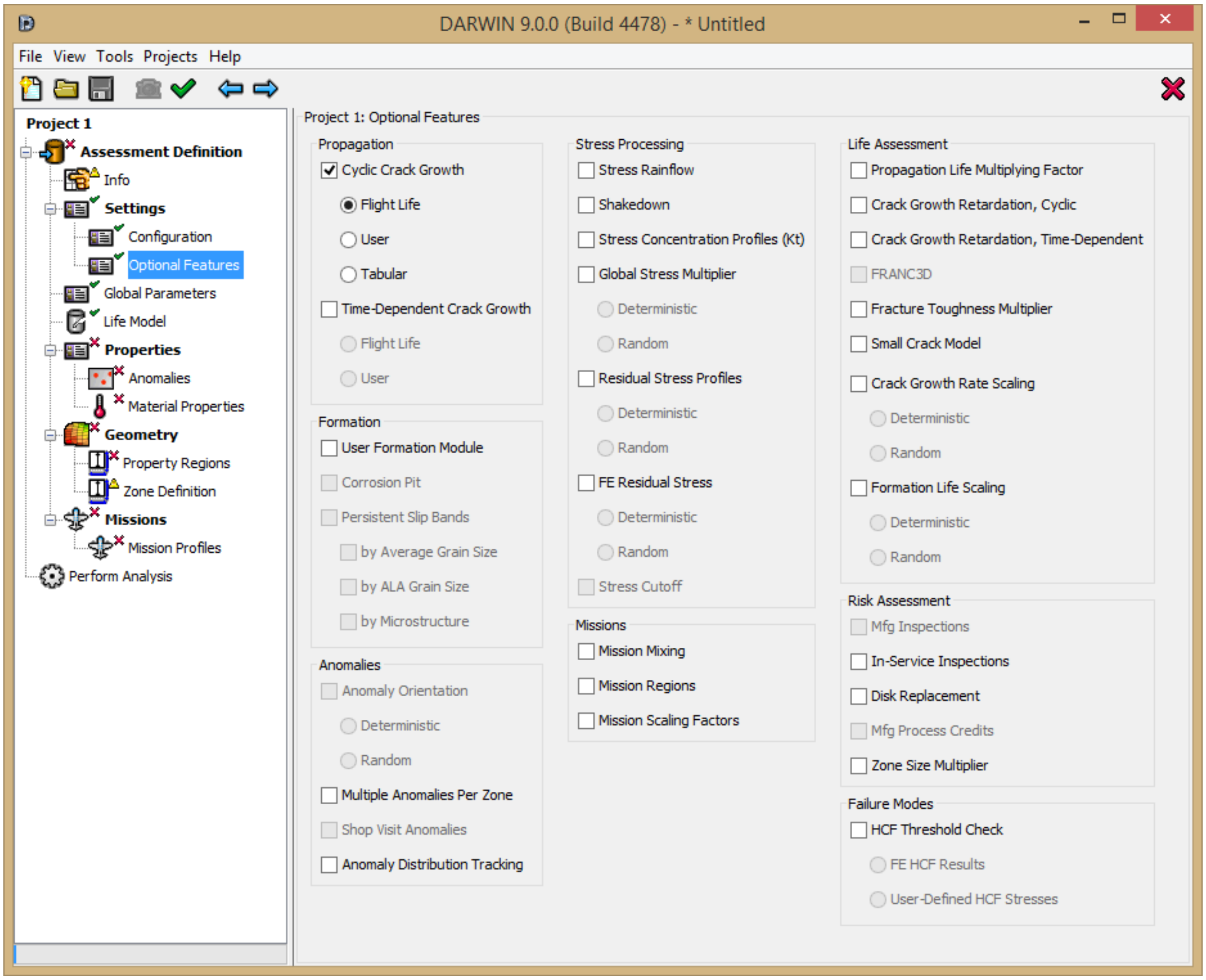
Anomaly Type
● Inherent
● 2D Anomaly
○ 3D Anomaly
○ Surface Damage

Analysis Method
○ Stress Processing
○ Life Assessment
● Risk Assessment

Zone Setup
○ Automatic
○ Exhaustive (1 zone per element)
● Optimal
□ GP Pre-Zoning
● Manual

Unit System
● US Customary
○ SI

Figure 5. GUI configuration menu implemented in DARWIN 8.2



Project 1: Optional Features

Propagation
☑ Cyclic Crack Growth
● Flight Life
○ User
○ Tabular
□ Time-Dependent Crack Growth
○ Flight Life
○ User

Formation
□ User Formation Module
□ Corrosion Pit
□ Persistent Slip Bands
□ by Average Grain Size
□ by ALA Grain Size
□ by Microstructure

Anomalies
□ Anomaly Orientation
○ Deterministic
○ Random
□ Multiple Anomalies Per Zone
□ Shop Visit Anomalies
□ Anomaly Distribution Tracking

Stress Processing
□ Stress Rainflow
□ Shakedown
□ Stress Concentration Profiles (Kt)
□ Global Stress Multiplier
○ Deterministic
○ Random
□ Residual Stress Profiles
○ Deterministic
○ Random
□ FE Residual Stress
○ Deterministic
○ Random
□ Stress Cutoff

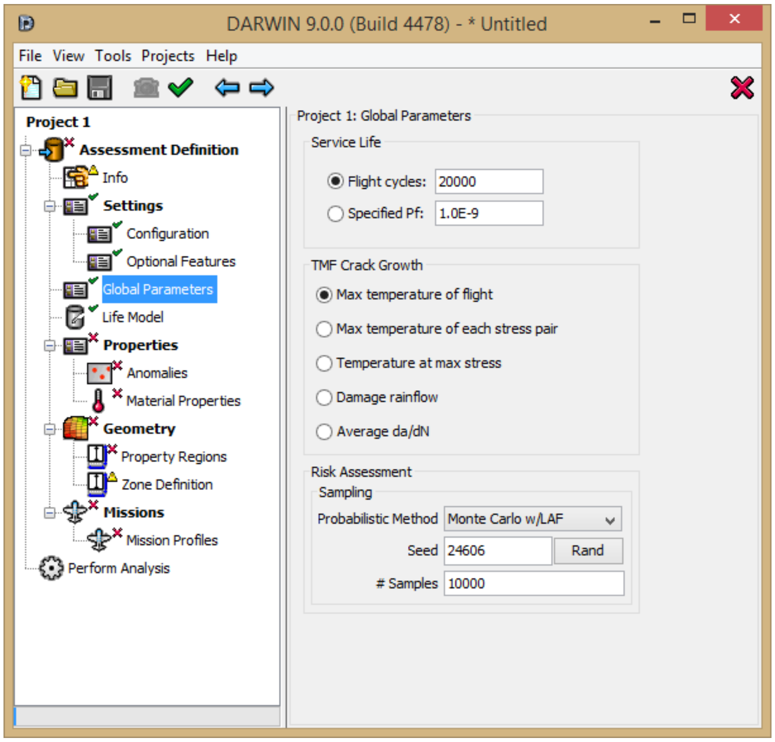
Missions
□ Mission Mixing
□ Mission Regions
□ Mission Scaling Factors

Life Assessment
□ Propagation Life Multiplying Factor
□ Crack Growth Retardation, Cyclic
□ Crack Growth Retardation, Time-Dependent
□ FRANC3D
□ Fracture Toughness Multiplier
□ Small Crack Model
□ Crack Growth Rate Scaling
○ Deterministic
○ Random
□ Formation Life Scaling
○ Deterministic
○ Random

Risk Assessment
□ Mfg Inspections
□ In-Service Inspections
□ Disk Replacement
□ Mfg Process Credits
□ Zone Size Multiplier

Failure Modes
□ HCF Threshold Check
○ FE HCF Results
○ User-Defined HCF Stresses

Figure 6. GUI optional features menu implemented in DARWIN 8.2



Project 1: Global Parameters

Service Life

● Flight cycles: 20000

○ Specified Pf: 1.0E-9

TMF Crack Growth

● Max temperature of flight

○ Max temperature of each stress pair

○ Temperature at max stress

○ Damage rainflow

○ Average da/dN

Risk Assessment

Sampling

Probabilistic Method: Monte Carlo w/LAF

Seed: 24606 Rand

Samples: 10000

Figure 7. GUI global factors menu implemented in DARWIN 8.2

2.1.5 Implementation of 64-Bit GUI

In DARWIN 8.1 (and previous versions), the GUI used a 32-bit operating system in which the memory available for the GUI was limited to approximately 1.0–1.5 GB. This was related to the available addressable memory associated with 32-bit Windows® operating systems, which are limited to 3.5 GB. This limit was adequate for smaller 2-D FEMs, but was insufficient for visualization of the larger 2-D and 3-D models that are now commonly used in the aircraft gas turbine engine industry.

Most users have upgraded to 64-bit operating systems that are limited only by the allocable memory of the hardware (typically 8 GB) rather than by the addressable memory of the operating system (theoretically 4x2^32 GB for 64-bit systems). Most memory-intensive software programs have provided 64-bit versions for years and have discontinued support for 32-bit versions (ANSYS 15 is the last version with 32-bit support, according to their literature).

A 64-bit version of the GUI was implemented in DARWIN 8.2. This enables the user to allocate significantly more memory to support memory-intensive operations, such as the display and manipulation of 3-D FEMs. The memory of the 64-bit GUI is no longer limited by the operating system, so the user will have the capability to increase it (via hardware improvements) to meet future computational demands.

2.1.6 Zone Property Regions Enhancement

In DARWIN 7.1 (and previous versions), zones were created manually and component properties (e.g., anomaly exceedance curves) were assigned to the zones. This approach could not be used for autozoning (the zones are initially undefined), so the GUI was enhanced in DARWIN 7.2 to enable the user to assign properties directly to FEs. Based on this information, DARWIN identifies property regions or groups of FEs containing identical properties (see figure 8).

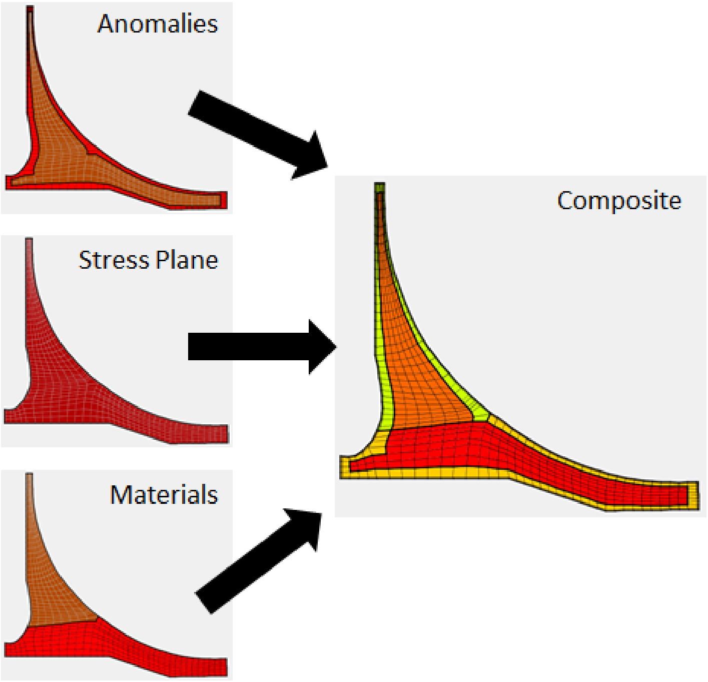


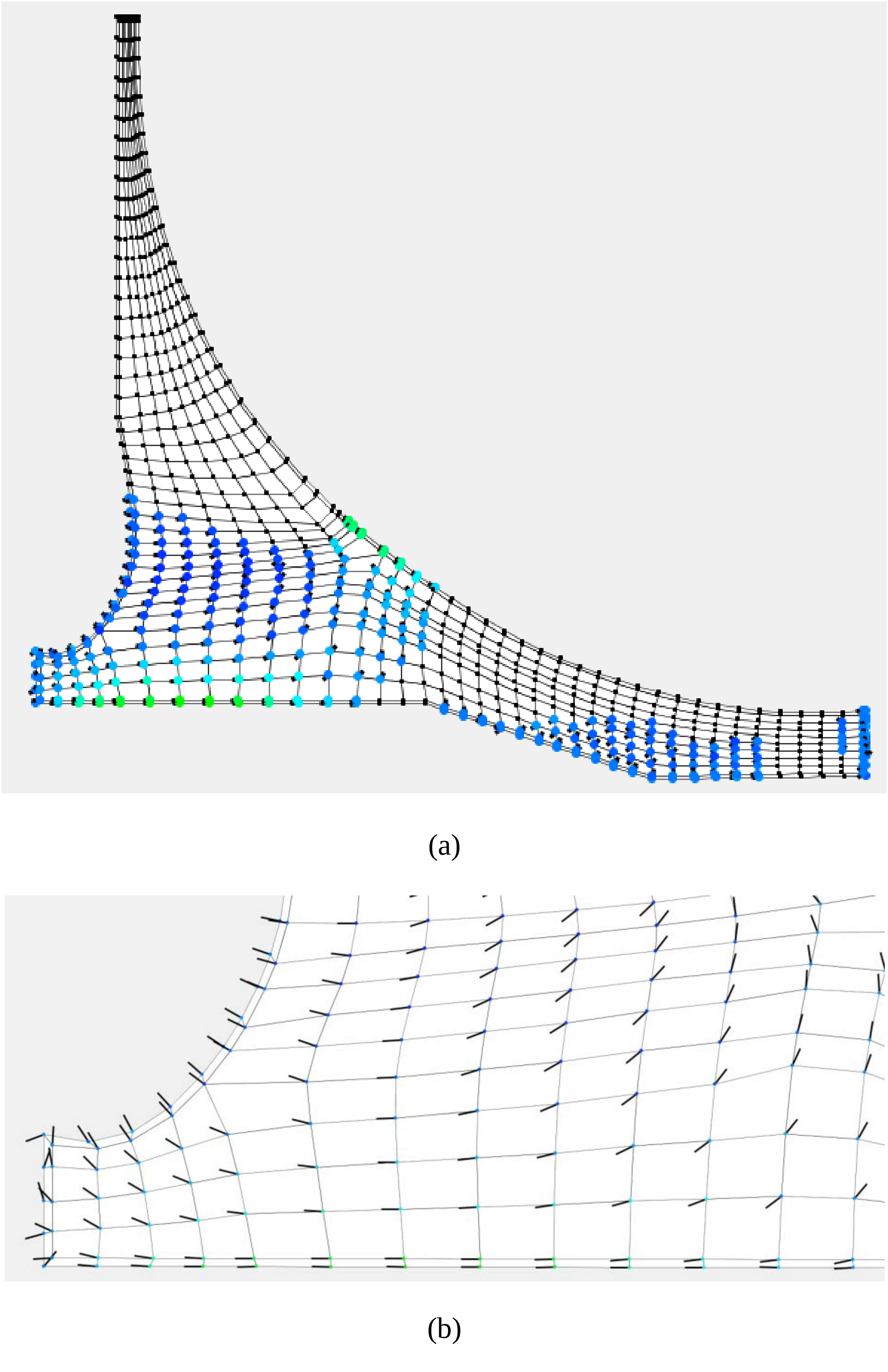
Figure 8. Schematic representation of the DARWIN zone property regions capability

The same approach was applied to manual zoning in DARWIN 8.2. Rather than defining zones first and then applying properties to them, the user first applies the properties to the FEM and then defines zones based on the property regions identified by DARWIN. This improvement not only simplified the manual zone definition process, but it has provided a more consistent framework for the assignment of properties in all analysis modes.

2.2 NON-HOOP MAXIMUM PRINCIPAL STRESS PLANES

In DARWIN 7.2 (and previous versions), fracture analysis was limited to hoop, axial, and radial planes for 2-D axisymmetric models, and visualization was limited to hoop planes. A new capability was developed to define fracture models in general non-hoop stress planes (including maximum principal stress planes) for 2-D axisymmetric FEMs. This ensures that the analysis is performed for the life-limiting orientation.

The DARWIN GUI was enhanced to enable the user to view non-hoop stress planes, including maximum principal stress planes. The GUI displays the orientation of the maximum principal stress plane associated with each FE node. When the user views the entire component, the hoop and non-hoop principal stress plane orientations are displayed as black and colored dots at each node, respectively (see figure 9(a)). When the GUI zoom button is applied, the plot provides further detail regarding the principal stress plane orientation at each node (see figure 9(b)). For non-hoop stress planes, the fracture plate is displayed as a single line that is coincident with the fracture-plate stress gradient (see figure 10). This feature was implemented in DARWIN 8.0 and is limited to univariant SIF solutions in the 2-D inherent analysis mode (i.e., embedded crack case EC05, surface crack case SC30, and corner crack case CC11).



(a)

(b)

Figure 9. DARWIN GUI display of maximum principal stress plane orientation at each FE node: (a) full-component view and (b) zoom view

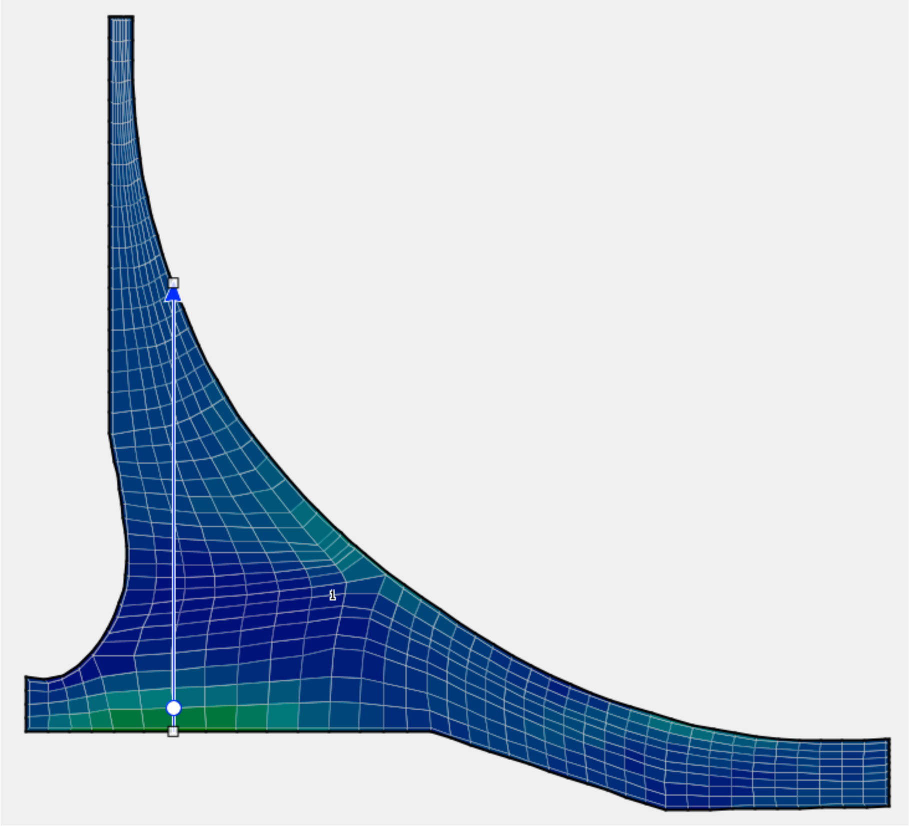
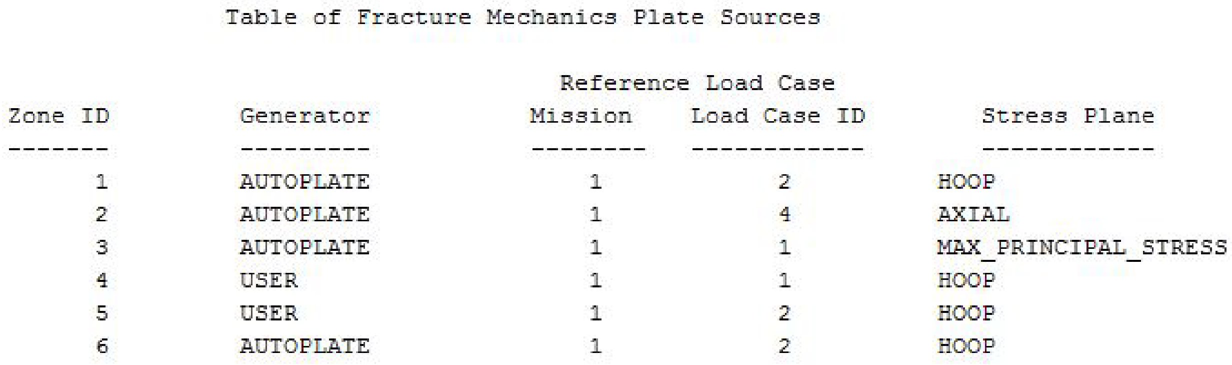


Figure 10. DARWIN 8.0 GUI display for non-hoop stress planes

The DARWIN auto-modeling algorithms (e.g., autoplate, life contours, and autozoning) were also enhanced to support non-hoop stress planes, in which the maximum principal stress plane is used as the default non-hoop stress plane.

In DARWIN 7.2 (and previous versions), the user had two options for creating fracture plates: Autoplate and Create Plate. Autoplate generated a ready-to-use plate via the DARWIN Autoplate algorithm, and Create Plate generated a simple plate that required user manipulation. In DARWIN 8.0, these two buttons were combined into a single Create Plate button that builds plates using the Autoplate algorithm. The user can adjust the resulting plate if desired. In rare instances in which Autoplate execution fails, the plate is generated using the previous Create Plate algorithm. Plate information is provided in the DARWIN output file (*.out) in the form of a table (see figure 11).



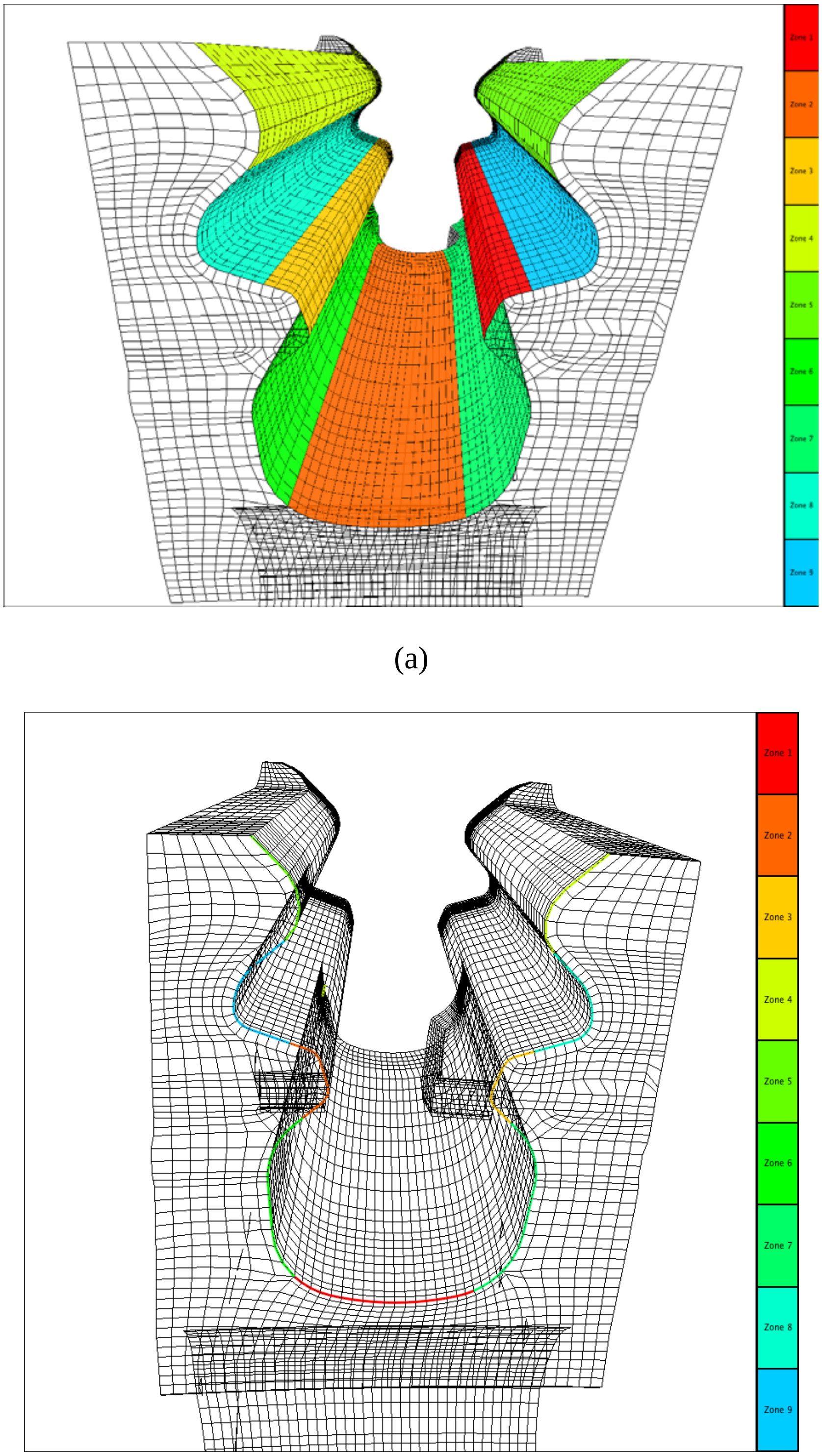
```
               Table of Fracture Mechanics Plate Sources

                                        Reference Load Case
Zone ID         Generator             Mission    Load Case ID        Stress Plane
-------         ---------             --------   ------------        ------------
      1         AUTOPLATE                1            2           HOOP
      2         AUTOPLATE                1            4           AXIAL
      3         AUTOPLATE                1            1           MAX_PRINCIPAL_STRESS
      4         USER                     1            1           HOOP
      5         USER                     1            2           HOOP
      6         AUTOPLATE                1            2           HOOP
```

Figure 11. Table in DARWIN output file describing plate information for each zone

2.3 3-D SURFACE AND LINE ZONING

RISC has been investigating surface damage anomalies that can be introduced in the blade-slot region of engine disks during routine blade removal and replacement procedures. The anomalies may appear anywhere on the mating surfaces of the slot, but may sometimes occur more frequently on the surfaces near or on the edges of the slot. The 3-D surface damage capability in DARWIN 8.0 (and previous versions) did not support the introduction of anomalies on the edges of a component or at specific locations of a surface. DARWIN 8.1 was enhanced to enable the user to define zones in 3-D FEMs by selecting element faces (surface-based zones) (see figure 12(a)) or edges (line-based zones) (see figure 12(b)). During zone creation, the faces or edges associated with a zone are highlighted, and the cumulative surface area or line length is automatically computed and displayed for the zone. DARWIN 8.1 was also enhanced to support anomaly distributions for line-based zones that are based on the number of anomalies per unit length. This new capability enables RISC to complete blade-slot damage studies and provides a production-ready tool to enable users to perform blade-slot risk assessments.



(a)

(b)

Figure 12. Examples of zones for blade-slot damage assessment associated with (a) element faces or (b) element edges

2.4 3-D FE SECTOR MODELS

DARWIN 8.2 (and previous versions) supported the use of 3-D FEMs in life and risk assessments to describe the geometry and loads associated with a component. This capability was limited to 3-D FEMs that contain the full component geometry. With the increase in analysis complexity and a desire for greater accuracy, FEM sizes have increased substantially. As a result, the use of 3-D FEMs that analyze a full component geometry have become undesirable, and the use of sector models has become more prevalent among gas turbine engine manufacturers.

DARWIN 9.0 was enhanced to provide support for 3-D sector models. FE2NEU (DARWIN's native FE results file conversion tool) was modified to acquire sector model specifications from ANSYS and ABAQUS results files. Sector model specifications include the axis of rotation and the number of sectors represented by the sector model. During execution, FE2NEU stores the sector model specification in the FE geometry neutral file (*.uif). When the 3-D sector model is imported into the DARWIN GUI, it displays the sector geometry and results, and also a wireframe outline of the full component geometry based on the provided sector model specifications (see figure 13).

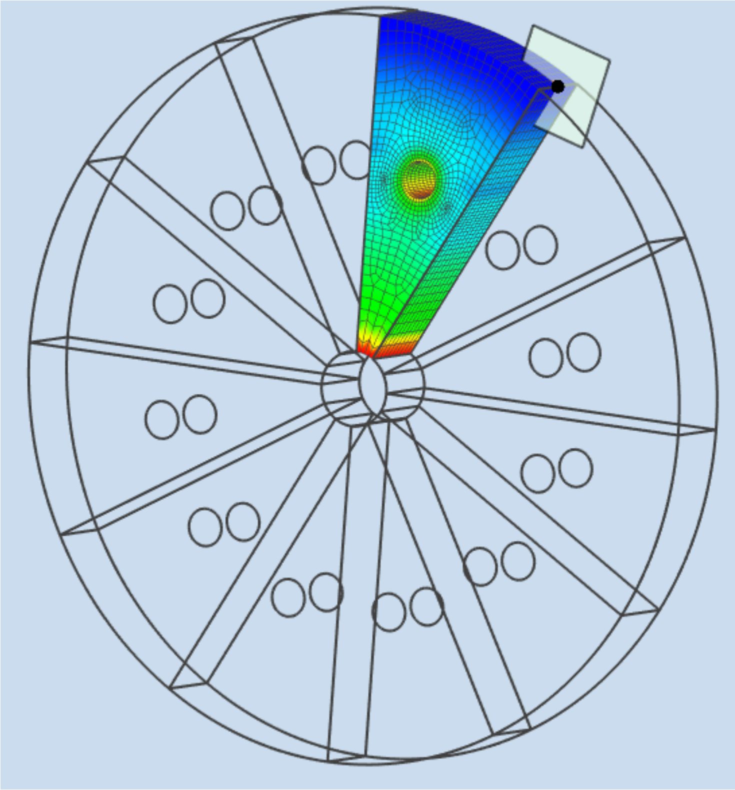


Figure 13. Display of the full component geometry based on the provided sector model specifications

In DARWIN 8.2, life and risk assessments performed using 3-D sector models were limited to cracks that were within the 3-D sector. The 3-D sector model capability in DARWIN 9.0 enables the crack to extend beyond the sector boundary and into the full component model. The stresses throughout the non-sector area in the crack growth field are based on the stresses in the 3-D sector model.

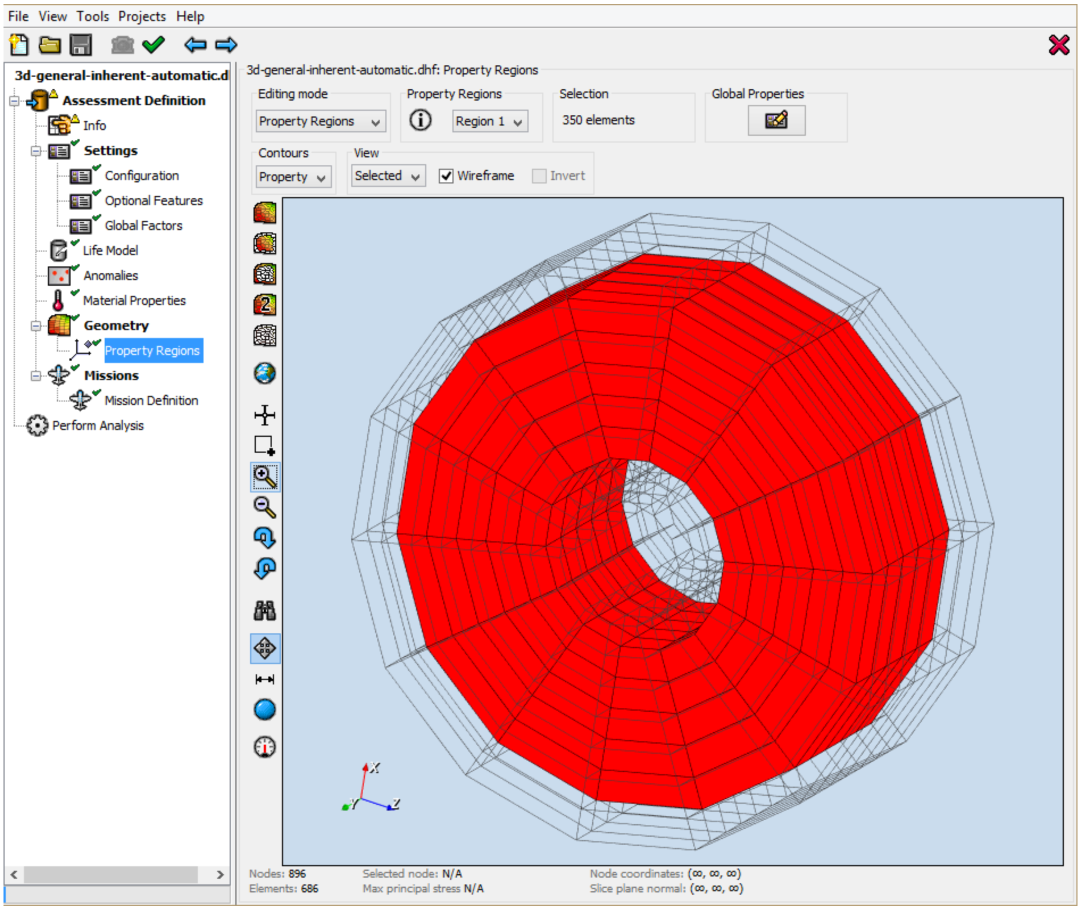
2.5 3-D VOLUMETRIC ZONING FOR INHERENT ANOMALIES

In DARWIN 8.1 (and previous versions), analysis of inherent anomalies was limited to 2-D axisymmetric FEMs. New autozoning capabilities have been developed for inherent anomaly assessments using 3-D FEMs. DARWIN 8.2 uses a process called exhaustive autozoning that creates a zone in each FE in a 3-D FEM to assess inherent anomaly risk (see section 2.5.1). DARWIN 9.0 uses 3-D optimal autozoning in which multiple FEs are automatically placed in zones for inherent anomaly risk assessment (see section 2.5.2).

2.5.1 3-D Exhaustive Autozoning Algorithm

New capabilities in DARWIN 8.2 support 3-D analysis of life prediction and fracture risk assessment for general inherent anomalies (e.g., HA inclusions). DARWIN 8.2 enables 3-D geometries imported from 3-D FEMs to be analyzed with internal anomalies in addition to the existing 3-D surface damage analysis capability.

DARWIN 8.2 supports inherent anomalies in 3-D geometries within the exhaustive autozoning framework. This capability is similar to existing capabilities for 2-D models. GUI enhancements enable user definition of property region information (i.e., material response, anomaly distributions, inspection schedules, and mission regions) directly on the 3-D FEM. Properties may be defined for the entire model, surface, and/or internal elements only (see figure 14), or element by element. This functionality simplifies the definition of components where surface regions have different anomaly distributions than interior regions, or dissimilar materials are fused together in the component.



3d-general-inherent-automatic.dhf: Property Regions

Editing mode: Property Regions

Property Regions: Region 1

Selection: 350 elements

Global Properties

Contours: Property

View: Selected

☑ Wireframe

□ Invert

Nodes: 896
Elements: 686

Selected node: N/A
Max principal stress N/A

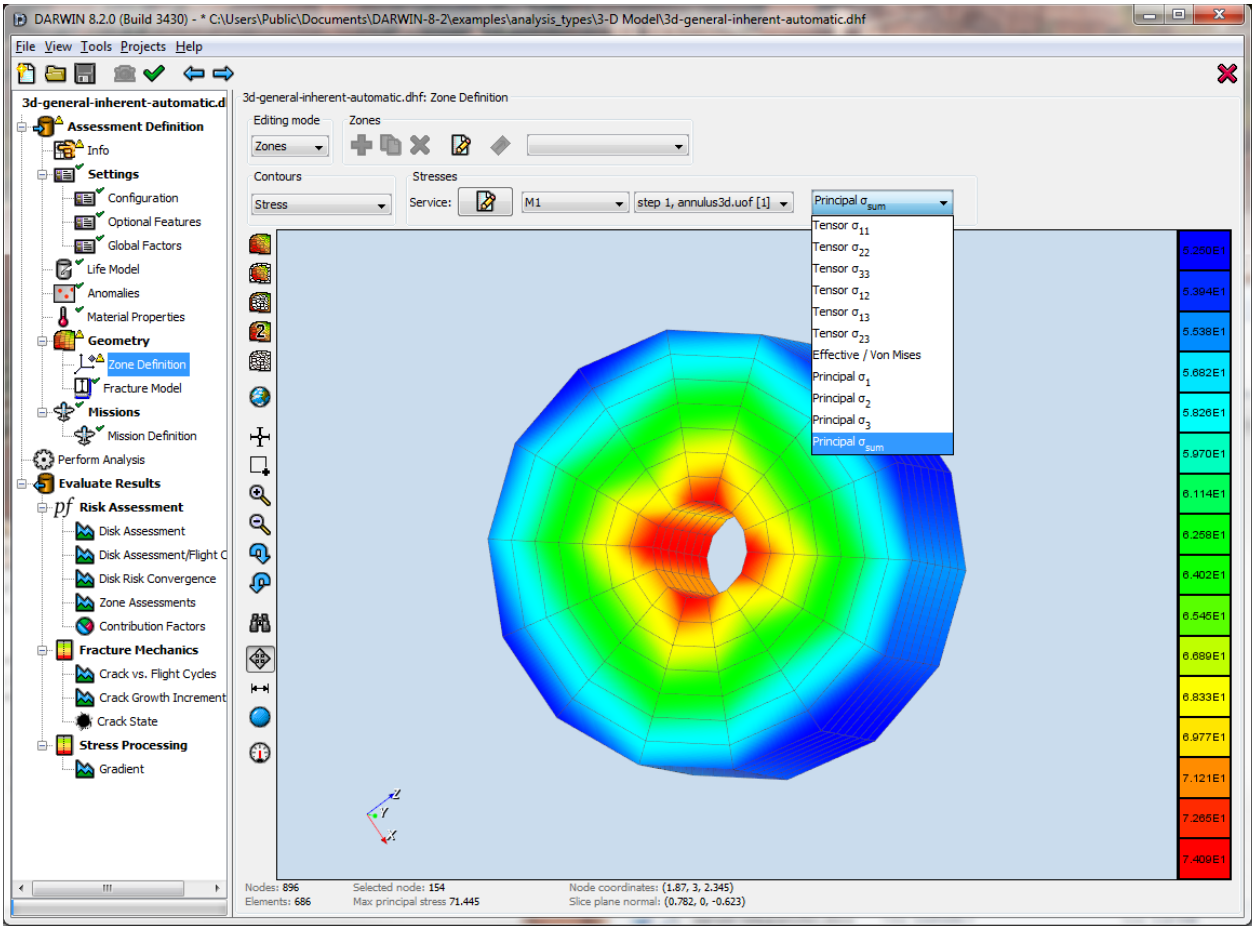
Node coordinates: (∞, ∞, ∞)
Slice plane normal: (∞, ∞, ∞)

Figure 14. Example of specification and viewing of property regions in 3-D FEMs

Autozoning separates zones by property regions when computing life and risk values. DARWIN uses property-region information to determine the 3-D zones in the model using the exhaustive autozoning algorithm that was previously developed for 2-D FEMs and associated zones. This algorithm has been extended to support Hex and Tet elements (i.e., cubic- and tetrahedral-shaped elements, respectively) with linear and quadratic displacement fields.

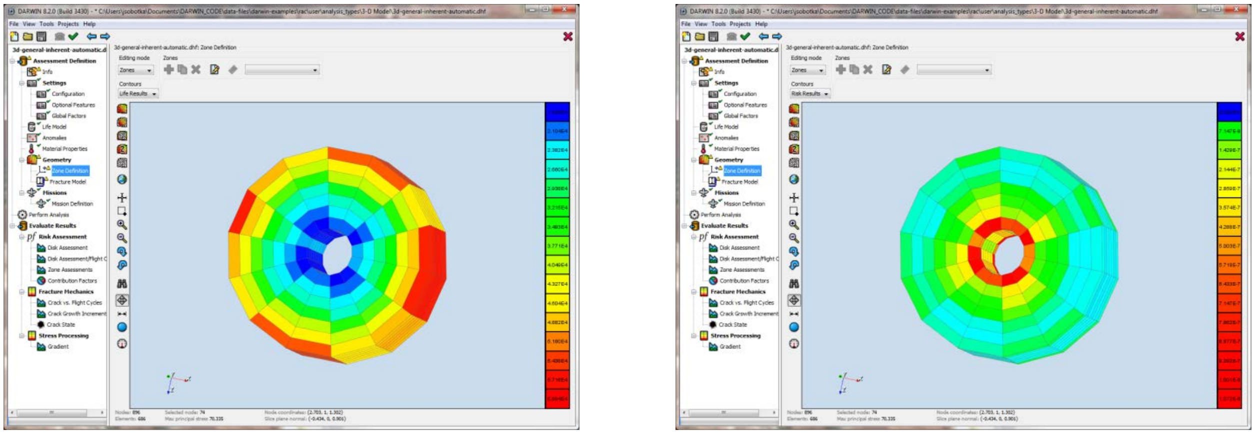
Exhaustive autozoning adopts a simple heuristic to locate cracks to maximize risk within the zone. A crack is placed at a node located on surface corners (if any exist in the element), then at a node on the surface of the model (again, if any exist in the element), and at the node closest to the surface of the model if the element has no surface nodes.

Three-dimensional autozoning results include stress, life, and risk information similar to results from a 2-D analysis of general inherent anomalies. Enhanced 3-D visualization techniques enable users to view contours of the stress variations, life predictions, and risk assessments. Figure 15 shows the stress variation for a 3-D model in DARWIN 8.2 and displays the available stress components and invariants. Figure 16 shows life and risk values based on the 3-D exhaustive autozoning algorithm.



Nodes: 896
Elements: 686
Selected node: 154
Max principal stress 71.445
Node coordinates: (1.87, 3, 2.345)
Slice plane normal: (0.782, 0, -0.623)

Figure 15. Visualization of stress tensor components and principal values in 3-D FEMs



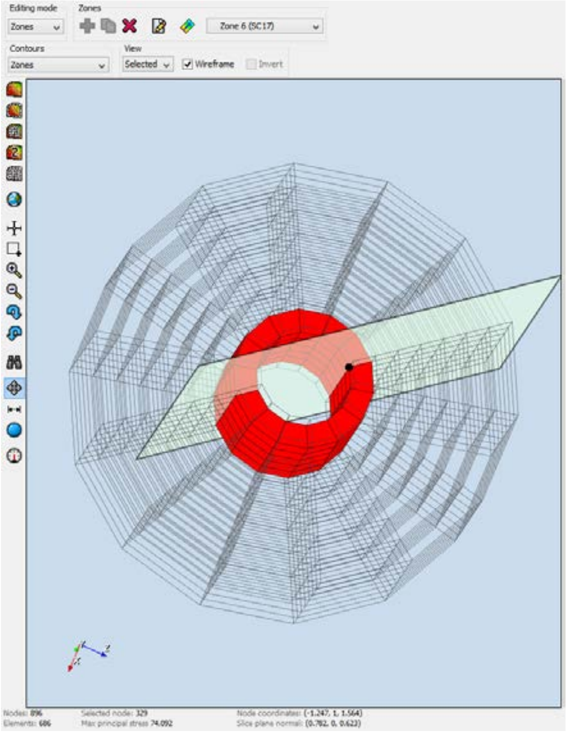
(a)

(b)

Figure 16. Examples of (a) life estimates and (b) risk assessment based on the DARWIN exhaustive autozoning algorithm for 3-D models

Additional information, such as crack location, plate definition, and the slice geometry (now stored within the analysis file), can be visualized zone by zone (see figure 17). DARWIN 8.2

also supports user redefinition of these features following an autozoning analysis. For example, the crack origin can be translated to different locations within a zone. DARWIN 8.2 enables reanalysis of the modified input model.



Nodes: 896
Elements: 686
Selected node: 329
Max principal stress 74.092
Node coordinates: (-1.247, 1, 1.564)
Slice plane normal: (0.782, 0, 0.623)

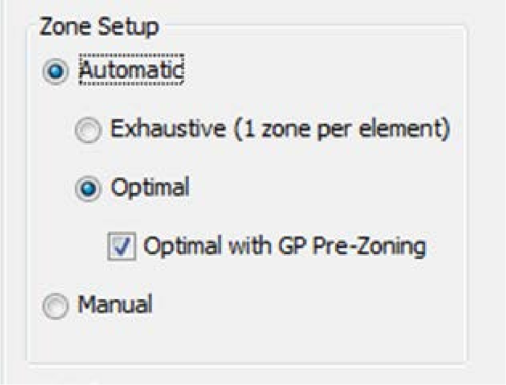
Figure 17. Example view of the 3-D elements and fracture mechanics plate associated with an individual zone

The exhaustive autozoning algorithm is impractical for production use with large 3-D FEMs, just as the 2-D exhaustive autozoning algorithm is impractical for production use with large 2-D FEMs. Development of the 3-D exhaustive algorithm was an essential first step for several reasons. First, the GUI functionality and work flow associated with the exhaustive algorithm were directly applicable to 3-D optimal autozoning. Second, the exhaustive autozoning method provided baseline results for computational verification of the optimal algorithm. Third, the optimal autozoning method would have been too large of an effort to complete in one step.

2.5.2 3-D Optimal Autozoning Algorithm

Creating zones manually is time consuming and requires human judgment. Algorithms have been developed to automatically create zones based on individual FEs, but the associated computation times increase exponentially with the number of FEs. Three-dimensional FEMs typically contain millions of FEs. Computation of component risk associated with 3-D FEMs using individual FE-based automated zoning algorithms may take days or even weeks to complete. An improved optimal autozoning methodology called “Optimal Autozoning with GP Pre-Zoning” has been developed that substantially reduces the computation time associated with fracture risk assessments. It combines FEs with similar properties (i.e., stress, temperature, and proximity to the surface) into groups called “pre-zones”. An automated zone-creation algorithm is applied to pre-zones rather than individual FEs, reducing the overall number of computations. In this section, the optimal autozoning methodology is presented and illustrated for FE geometries in both 2-D and 3-D gas turbine engine components. Based on the demonstration problem results, it is shown that the computation speed associated with the optimal autozoning algorithm is expected to be three to four orders of magnitude faster than a previous algorithm that created zones at individual FEs. The pre-zoning-based algorithm also requires less memory (about 50% less memory for the example problems considered) than previous algorithms, enabling it to solve much larger models. The resulting algorithm provides a feasible and realistic solution for fracture risk assessment of 2-D and 3-D component FEMs. Complete details of the algorithm are provided in appendix B.

The DARWIN GUI has been enhanced to support the new optimal autozoning framework with pre-zoning. The configuration screen has a new zone setup option, “Optimal with GP Pre-Zoning” (see figure 18). Toggling this option activates the new algorithm. Users then define the analysis using the same commands that they would use for an exhaustive autozoning analysis. The GUI displays risk contours and pre-zone boundaries based on the estimates from the associated Gaussian process (GP) models (see figure 19) as well as the uncertainties of the models (see figure 20). Additional information regarding the risk drivers associated with pre-zone breakup is provided via a new information button (see figure 21). Pre-zone contours can be shown with the zone boundaries to identify the pre-zones that have been combined into zones (see figure 22). Users can also modify the crack location and rerun the analysis using the manual zoning approach.



Zone Setup

● Automatic

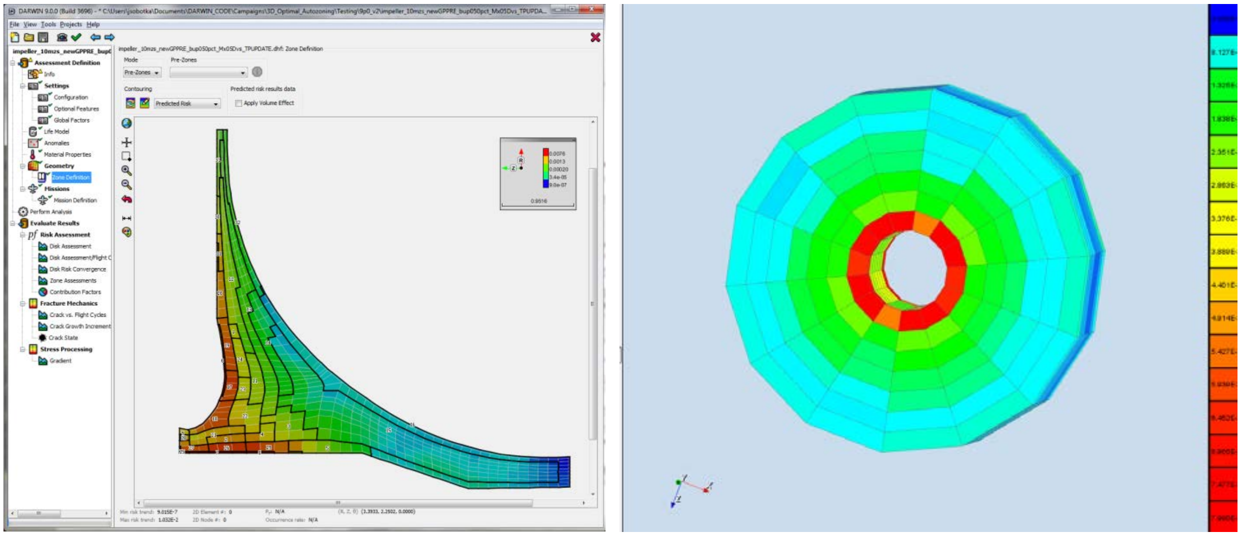
○ Exhaustive (1 zone per element)

● Optimal

☑ Optimal with GP Pre-Zoning

○ Manual

Figure 18. Enhanced selection options for the optimal GP pre-zoning methodology



DARWIN 9.0.0 (Build 3696) - * C:\Users\jsobotka\Documents\DARWIN_CODE\Campaigns\3D_Optimal_Autozoning\Testing\9p0_v2\impeller_10mzs_newGPPRE_bup050pct_Mx05Dvs_TPUPDA...

impeller_10mzs_newGPPRE_bup050pct_Mx05Dvs_TPUPDATE.dhf: Zone Definition

Mode: Pre-Zones

Pre-Zones:

Contouring: Predicted Risk

Predicted risk results data

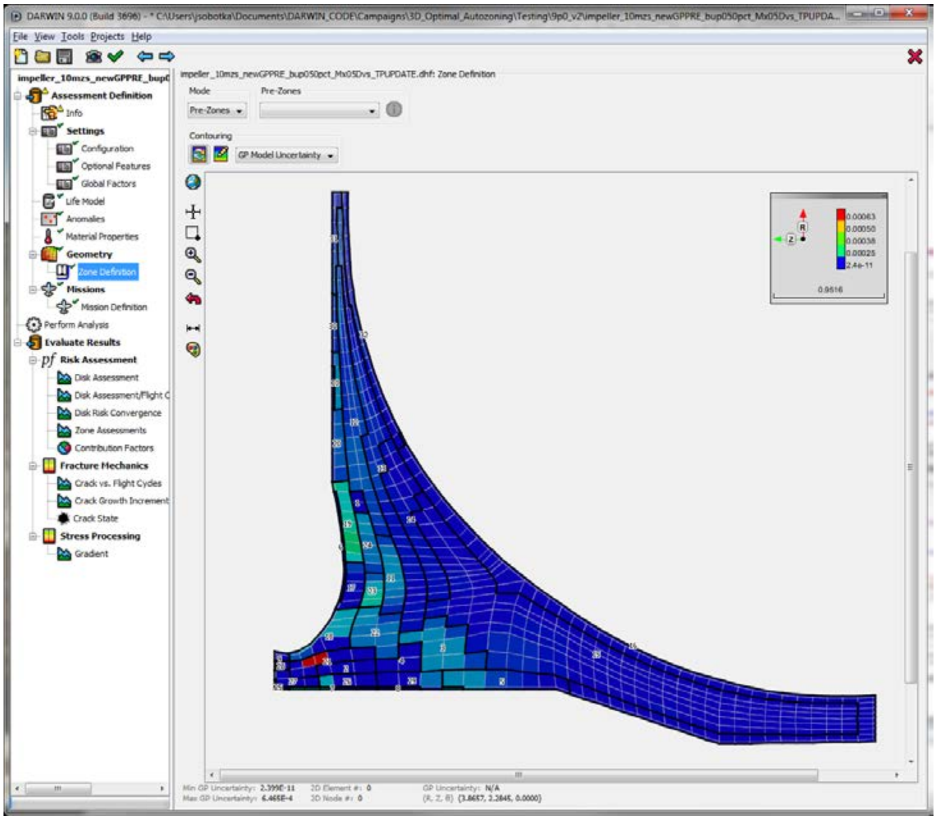
□ Apply Volume Effect

Min risk trend: 9.015E-7   2D Element #: 0 P_f: N/A   (R, Z, θ) (3.3933, 2.2502, 0.0000)

Max risk trend: 1.032E-2   2D Node #: 0   Occurrence rate: N/A

(a) (b)

Figure 19. GUI display of risk estimates and boundaries associated with pre-zones: (a) 2-D FEM and (b) 3-D FEM



DARWIN 9.0.0 (Build 3696) - * C:\Users\jsobotka\Documents\DARWIN_CODE\Campaigns\3D_Optimal_Autozoning\Testing\9p0_v2\impeller_10mzs_newGPPRE_bup050pct_Mx05Dvs_TPUPDA...

impeller_10mzs_newGPPRE_bup050pct_Mx05Dvs_TPUPDATE.dhf: Zone Definition

Mode: Pre-Zones

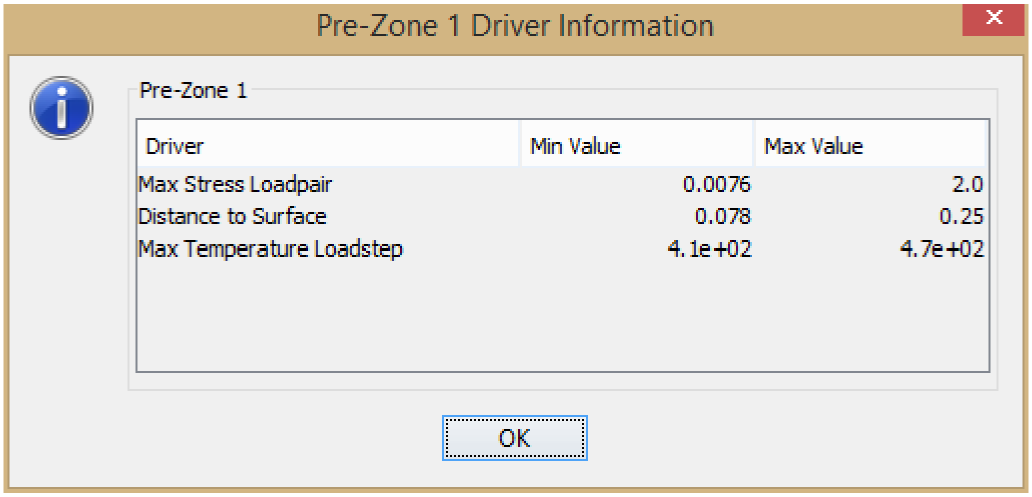
Pre-Zones:

Contouring: GP Model Uncertainty

Min GP Uncertainty: 2.399E-11   2D Element #: 0   GP Uncertainty: N/A

Max GP Uncertainty: 6.465E-4   2D Node #: 0   (R, Z, θ) (3.8657, 2.2845, 0.0000)

Figure 20. GUI display of uncertainty associated with GP model estimates



Pre-Zone 1

Driver	Min Value	Max Value
Max Stress Loadpair	0.0076	2.0
Distance to Surface	0.078	0.25
Max Temperature Loadstep	4.1e+02	4.7e+02

Figure 21. GUI display of information regarding the risk drivers associated with GP pre-zones.

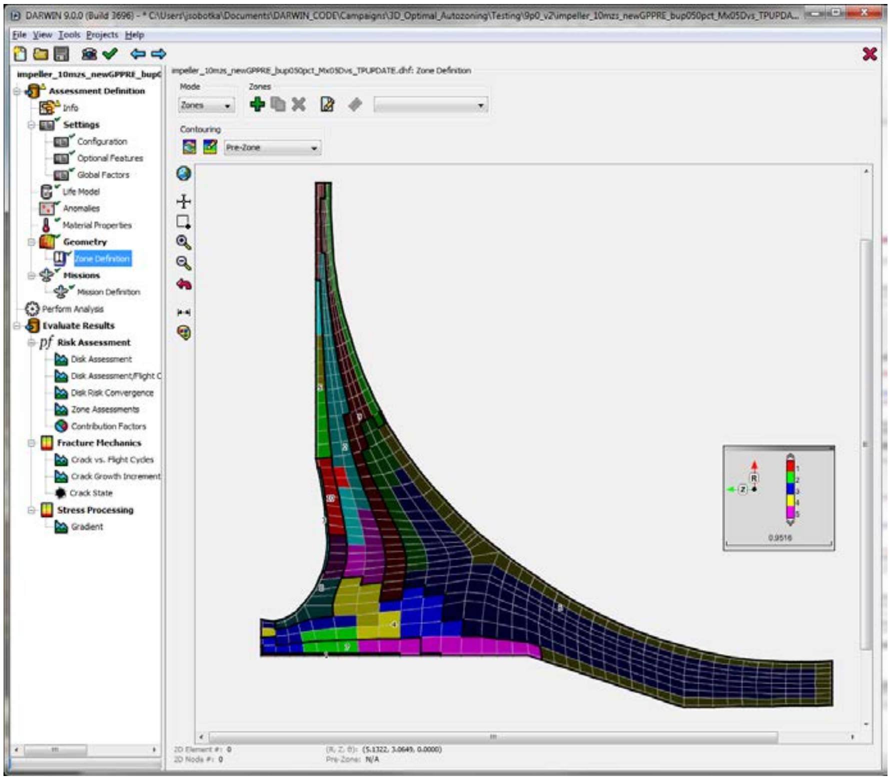
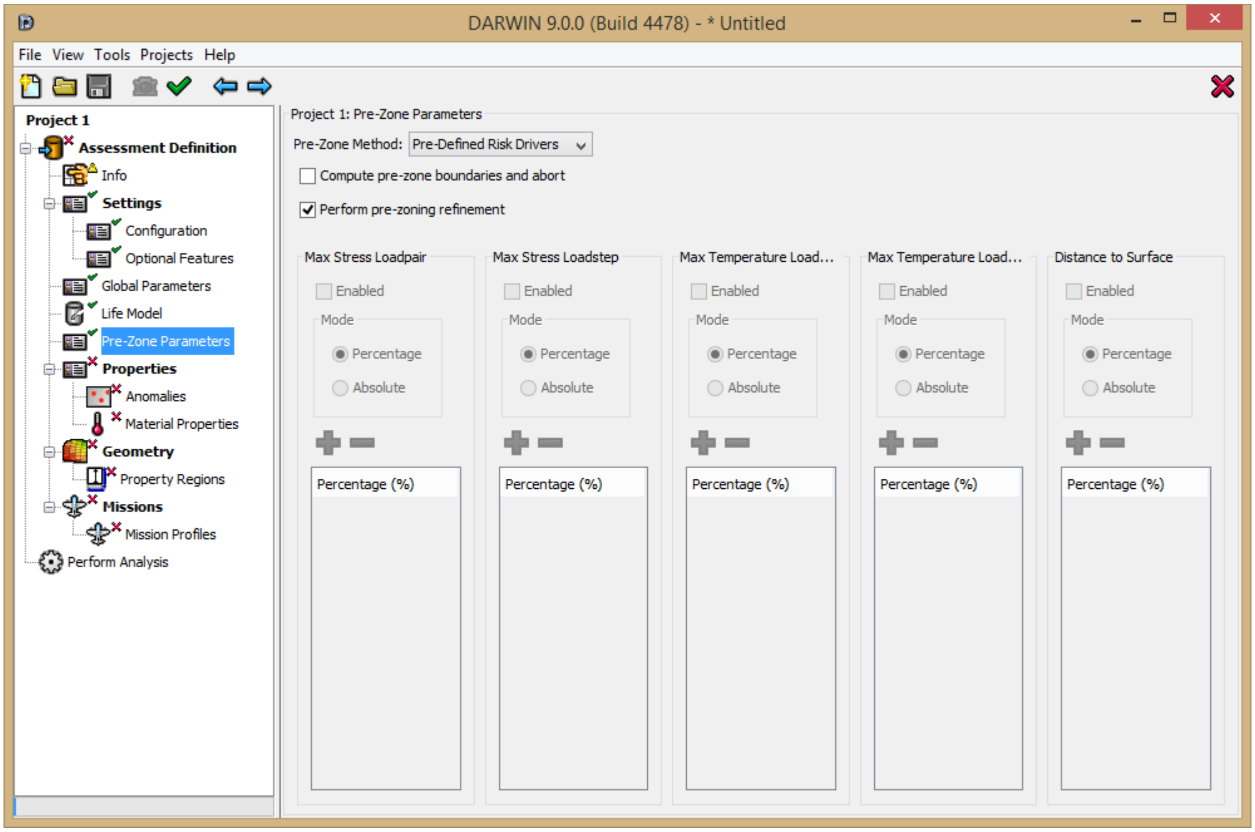


Figure 22. GUI display of GP pre-zone contours

The risk drivers play an important role in the definition of pre-zones. By default, the values for the risk drivers are not accessible by the user. However, a hidden feature ("riskDrivers") is provided in the GUI that enables the user to specify these values. When the hidden feature is enabled, the GUI displays the pre-zone definition screen (see figure 23). This screen enables users to select their own combination of risk drivers and how each risk driver shall be broken up. Five risk drivers are supported: the maximum stress load pair and load step, the maximum temperature load pair and load step, and distance to surface. Any or all of these drivers may be enabled by the user. With this option, users select the values that serve as boundaries for the pre-zone. For example, the user could specify boundaries for pre-zones at 10 ksi, 20 ksi, 30 ksi, and 40 ksi stress values. These values may be entered at absolute values or as percentages of the minimum/maximum values. DARWIN 9.0 supports the addition of any number of divisions of the model into pre-zones.



DARWIN 9.0.0 (Build 4478) - * Untitled

Project 1: Pre-Zone Parameters

Pre-Zone Method: Pre-Defined Risk Drivers

□ Compute pre-zone boundaries and abort

☑ Perform pre-zoning refinement

Max Stress Loadpair	Max Stress Loadstep	Max Temperature Load...	Max Temperature Load...	Distance to Surface
□ Enabled	□ Enabled	□ Enabled	□ Enabled	□ Enabled
Mode ● Percentage ○ Absolute	Mode ● Percentage ○ Absolute	Mode ● Percentage ○ Absolute	Mode ● Percentage ○ Absolute	Mode ● Percentage ○ Absolute
Percentage (%)	Percentage (%)	Percentage (%)	Percentage (%)	Percentage (%)

Figure 23. DARWIN 9.0 optional pre-zone definition screen

2.6 TREATMENT OF LARGE FEMS WITH LARGE NUMBERS OF LOAD STEPS

Continuing growth in the size of production FEMs introduced memory and speed problems that could have ultimately limited the use of DARWIN within the commercial aircraft engine industry, if not addressed. In DARWIN 9.0 (and previous versions), FE geometry and results were stored in ASCII formatted files. Upon retrieval, the entire geometry and stress results files were read and stored in random access memory. For large FEMs, the required memory may exceed available computer memory. Excessive memory use can also significantly increase the computation time associated with assessments involving large FEMs.

For example, consider a FEM with approximately 1,000,000 nodes and 1,000 load steps. Every node in the model has six stress scalars and one temperature assigned to every load step. Every stress scalar/temperature requires 8 bytes to store as a double precision number. To store the loading history, 56 GB of memory is required, excluding any other data structures needed for additional computations. DARWIN 8.0 subprograms (i.e., RAC, GUI, and Autoplate) read the data from an ASCII-formatted file. These subprograms can become unresponsive (i.e., freeze or hang) when reading the data from large files. When this occurs, users can neither see any useful visualizations or set up their models in the GUI, nor can they track the progress in the RAC. Though this example is extreme, original equipment manufacturer (OEM) FEMs are approaching these sizes.

Efforts have been directed toward short-term and long-term solutions that treat large FEMs with large numbers of FEs/load steps. For the short-term, the DARWIN 9.0 GUI was enhanced to provide a user option to disable viewing of stress and temperature contours. This option is available in the "Preferences" menu and can be selected independently for 2-D and 3-D geometries. This feature reduces the time required to import large FEMs. For an example FE model with 500,000 elements, the GUI response time decreased from 1 min 50 seconds to 1 second (over 100x reduction) when using the new option disabling viewing of stress and temperature contours. The DARWIN 9.0 RAC was enhanced to provide a memory check for large FEMs. During runtime, the RAC computes the memory required to store FEM results and compares it to the available system memory. Analysis runs with FEMs that fail the memory check are terminated, and the RAC indicates the additional memory needed to execute the analysis.

The long-term solution for large FEMs is to store the FE results data in binary format. This format allows for fast retrieval of selected portions of results files using significantly less memory compared with the current ASCII format. The HDF5 was selected as the data model for storage of FE data in binary format. HDF5 is designed to handle large and complex data sets on a variety of computer platforms. The DARWIN input and results data are currently stored in HDF5 formatted files, and the RAC and GUI already include a number of algorithms for processing HDF5 data. HDF5 also features free data-visualization software (the HDF5 viewer). This software can be used to view and edit the binary file.

In DARWIN 9.0, FE geometry and results are stored in a data structure that is defined via the System for Integrated Engineering Structural and Thermal Analysis (SIESTA) standard previously developed by GE Aviation. SIESTA is an ASCII file format with support for major DARWIN features (e.g., node locations, element connectivity, and nodal stress/temperature data by load step). It is a flexible framework—in many ways, too flexible. SIESTA requires extensive parsing of the file to read in any line of data (e.g., nodal stresses can be listed in any order). Consequently, reading SIESTA takes additional time in excess of the time required to read ASCII data into memory.

The SIESTA standard has been extended for application to the HDF5 data format. The new file format, entitled HSIESTA, specifies the data structure for binary data files containing output data from finite element analysis (FEA). It represents the new interface between DARWIN and commercial FEA codes (e.g., ANSYS). HSIESTA is a new format that was developed with the following considerations:

1. Rapid retrieval of information stored in HSIESTA by the GUI and RAC
2. Incorporation and maintenance of file format in the existing computational framework
3. Long-term storage of binary data
4. Data format supported by a large user community
5. Extensibility of file format to current and future challenges
6. Stability of underlying file format

7. Accessibility of information by users via open-source tools
8. Support for parallel data access

The complete HSIESTA standard is provided in appendix C. It has the following major features:

- Hierarchical Binary File Format (HDF5)–The hierarchical format stores data frameworks that are similar to internal file managers on desktop computers (e.g., Windows Explorer). DARWIN currently uses the ASCII format of SIESTA to represent FE results as a continuous text file demarcated by keywords. In contrast, HSIESTA separates related data into related groups. For example, SIESTA indicates nodal locations by the keyword NODE, followed by line-after-line of nodal locations. HSIESTA separates all nodal geometric data into a single table that is located in the group node. This group may be expanded or contracted as needed using the HDF5 editor to enable visualization of data stored in HSIESTA files. The HDF5 file currently acts as the binary file structure for the DARWIN input file (*.dhf). Consequently, both the GUI and RAC software development teams have extensive experience with the HDF5 file format.
- Random-Access Capability–The GUI and the RAC may access information in the HSIESTA file format on an as-needed basis. For example, users may only need to display stresses for the 100th load case. In HSIESTA, the GUI reads the stresses for the 100th load case and nothing else. However, the ASCII file format of SIESTA imposes additional computational costs on the GUI: The GUI must read in steps 1–99 before reading in stresses for the 100th load case, though load cases 1–99 will not be used. SwRI expects significant computational savings to be achieved by exploiting the random-access capability of HDF5.
- Self-documentation–The hierarchical format adopted by HSIESTA is somewhat self-documenting. For example, nodal coordinates are assigned a unit system, located in the node group and stored in the coordinates table. Names of objects in the HSIESTA file use complete words/phrases rather than cryptic strings that may be abbreviated (e.g., SIESTA). Comments may be added to any group or table as an attribute that begins with "$", similar to the existing convention in SIESTA files.
- Combination of *.uif/uof files to single *.fea files–Currently, FE results are often stored as two files: *.uif files for the FE geometry, and *.uof files for the FE stresses and temperatures. The HSIESTA file format combines geometry and field (stress, temperature, etc.) information into a single binary file format (*.fea). As a result, only a single FE results file needs to be stored and maintained for archival purposes.
- Distinction of light and heavy data–Light data refers to information that is read once and used multiple times (e.g., nodal locations), whereas heavy data refers to information that may be read, used, and discarded (e.g., nodal stresses for a particular load case). SIESTA uses the *.uif (light) and *.uof (heavy) files to separate light and heavy data. HSIESTA instead creates a group for the light data (geometry) and for the heavy data (fields) within the same file. This file structure enables the RAC and the GUI to optimize memory usage for each type of data type and based on the overall size of the model.

- Support for large FEMs–The HDF5 file format has been designed to provide long-term storage for massive datasets (terabyte scale and above) from inception. HSIESTA (built with HDF5) should accommodate complex 3-D models (millions of elements and thousands of load cases). ASCII files may prove unmanageable at these file sizes.

HSIESTA has additional features not present in SIESTA:

- Hierarchical ordering of components–Structures are rarely composed of just one component (e.g., an engine might have a core rotor and blades). Currently, SIESTA has limited support to define multiple components within an assembly. HSIESTA orders the entire assembly into a hierarchical set of components. Components may represent a single part (e.g., a turbine blade) or a collection of parts (several turbine blades). All components may be traced to a single assembly. Each component has separate geometric definitions and field data. This feature should assist DARWIN to support named components in upcoming releases.
- Improved data structures–ASCII files (e.g., SIESTA) store large datasets (e.g., stresses) as consecutive lines of separated numbers (e.g., 1 2 3 4 5 …). This data structure requires that all data be recast as 2-D tables. HDF5 features more flexible data structures that are more appropriate to tensorial quantities. For example, HSIESTA uses a single 3-D table to store stresses by load case, index, and component. HDF5 supports data slicing in 3-D to more rapidly access data. HDF5 is supported by a GUI (i.e., the HDF5 viewer) that enables users to rearrange, plot, and copy data from 3-D tables.
- New element topologies–DARWIN supports 2-D and 3-D solid elements. SIESTA includes a limited element library of solid elements (e.g., wedges are not supported). SIESTA also contains several transition elements that require significant parsing while adding little information to the analysis. HSIESTA supports two 2-D topologies (triangles and quadrilaterals) and four 3-D topologies (tetrahedrons, pyramids, wedges, and bricks). These topologies cover the topologies in the major FE codes ABAQUS and ANSYS. HSIESTA defines the linear elements and quadratic serendipity elements for all topologies. HSIESTA defines the dozen element types it supports. Transition elements are downgraded to linear elements. These elements cover the major use cases. Reducing the number of elements will assist SwRI to develop routines that will certify FEMs. These routines will help users to identify errors in FEMs before they are imported into DARWIN.

Note that because the HSIESTA standard was developed solely by SwRI (with the guidance and review of the Steering Committee), it can be changed to incorporate new features without the permission of a third party. This flexibility will improve the long-term usability of HSIESTA.

The HSIESTA format was implemented in DARWIN 9.1. This has several consequences for future versions of DARWIN (9.1+):

- FE2NEU 6.0 has been modified to output neutral file data in the HSIESTA format only. FE2NEU 5.0 previously outputted neutral files in the SIESTA format (*.uif/uof files). To support backward compatibility, FE2NEU has been enhanced to convert files formatted in SIESTA to the HSIESTA format. This feature was used to convert files in the DARWIN verification model database from SIESTA to HSIESTA format.
- The RAC and GUI have been modified to read FEM data from the HSIESTA format only. Legacy SIESTA-formatted files will need to be converted to the HSIESTA format (via FE2NEU 6.0) prior to use in DARWIN 9.1.
- An application programming interface (API) has been developed to enable the RAC to read information from HSIESTA files. The API will simplify access of data. For example, the native HDF5 commands require developers to navigate to a group before accessing information stored in that group. The API will perform the navigation automatically. These API functions should minimize future computational efforts in the RAC and optimize the input/output operations. SwRI will make the API available to DARWIN users to simplify their data-processing routines. Appendix D documents in detail the current routines available in the API.
- The Autoplate algorithm has been modified to accept FE data from the RAC. Autoplate previously read the full FEM into memory whenever it was executed. Now, the RAC reads the data and sends it to Autoplate. This approach will yield a reduced memory footprint during autozoning. Autoplate accesses HSIESTA through the API functions designed for the RAC.

SwRI and Steering Committee evaluation of speed improvements associated with HSIESTA will be performed in a follow-on grant.

3. HIGH-TEMPERATURE CRACK GROWTH

3.1 GUI INTERFACE FOR CRACK-GROWTH RETARDATION MODELS

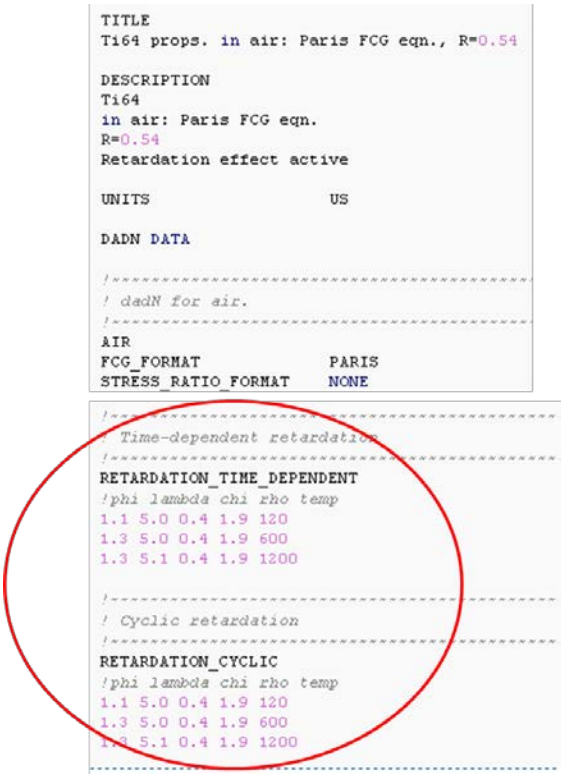
It has long been understood that a single overload cycle could retard crack growth in subsequent fatigue cycling at lower peak loads. This phenomenon can be particularly significant for aircraft structures, in which highly irregular variable amplitude loads arising from mission variations or wind gusts can have a profound impact on FCG rates. This effect is generally less significant for low-cycle fatigue of engine rotors, because the major cycle (which typically has the peak stress) is often the dominant contribution to damage on each mission. Therefore, this effect is often neglected in FCG analysis of rotors, and models to address retardation were not previously included in DARWIN. Neglecting this effect is conservative.

However, it has more recently been observed that overloads can have a profound impact on static crack growth if the overload occurs immediately prior to a slow ramp or dwell period [26]. Even a small overload can substantially reduce the ensuing static crack growth rate. Van Stone and Slavik [26] developed a modified Willenborg model (adapted from the generalized Willenborg model commonly used for load interaction effects in cyclic crack growth) to address dwell overload retardation.

During the previous research grant [22], GE Aviation performed a series of IN-718 crack-growth experiments to demonstrate the cyclic and static retardation effect and to evaluate the proposed modified Willenborg models. These two overload retardation models—one for cycle-dependent retardation, and one for time-dependent retardation—were also implemented in DARWIN near the end of that grant. However, because of resource limitations, corresponding GUI support was not implemented at that time. Therefore, the models were inaccessible to routine DARWIN users.

Development of a GUI interface for the crack-growth retardation models was completed under the PIRATE grant, and this interface was first released in DARWIN 8.0. The new feature enables the analyst to include crack retardation effects for both cycle- and time-dependent crack-growth life and risk assessments.

The interface required modifications to the DARWIN material file to provide definition of the cycle-dependent and time-dependent crack retardation properties. An example file that illustrates the revised format of the material properties data file is shown in figure 24.



```
TITLE
Ti64 props. in air: Paris FCG eqn., R=0.54

DESCRIPTION
Ti64
in air: Paris FCG eqn.
R=0.54
Retardation effect active

UNITS                   US

DADN DATA

!*****************************************
! dadN for air.
!*****************************************
AIR
FCG_FORMAT              PARIS
STRESS_RATIO_FORMAT     NONE
```

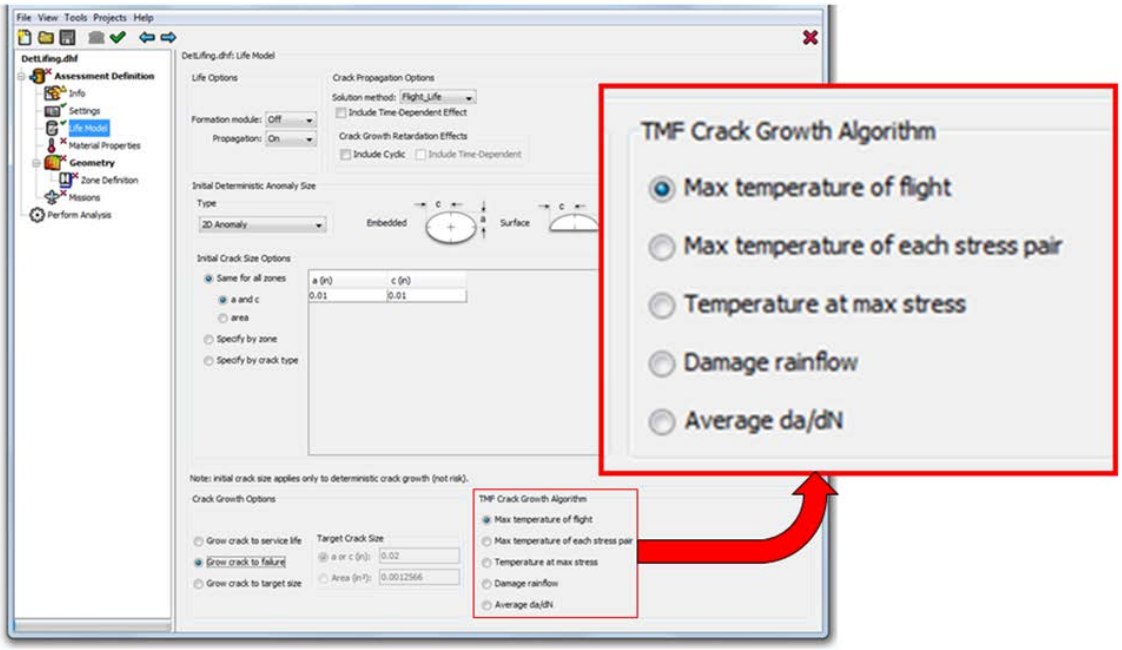
```
!*********************************************
! Time-dependent retardation
!*********************************************
RETARDATION_TIME_DEPENDENT
!phi lambda chi rho temp
1.1 5.0 0.4 1.9 120
1.3 5.0 0.4 1.9 600
1.3 5.1 0.4 1.9 1200

!*********************************************
! Cyclic retardation
!*********************************************
RETARDATION_CYCLIC
!phi lambda chi rho temp
1.1 5.0 0.4 1.9 120
1.3 5.0 0.4 1.9 600
1.3 5.1 0.4 1.9 1200
```

Figure 24. Example material properties data file associated with the crack-retardation feature in DARWIN

3.2 ADDITIONAL TEMPERATURE OPTION FOR THERMO-MECHANICAL FATIGUE CRACK GROWTH

AC 33.70-2 [24] provides design guidelines for treatment of crack propagation under out-of-phase temperature and stress profiles in which the temperature is based on either: a) the temperature at the maximum stress in each cycle, or b) the maximum temperature during the flight. Previous versions of DARWIN included the option to specify the temperature at the maximum stress in each cycle. A new thermo-mechanical fatigue (TMF) crack-growth option was introduced in DARWIN 8.0 that enables crack-propagation assessment based on the material properties corresponding to the maximum temperature during the flight (see figure 25).



DetLifing.dhf: Life Model

Life Options

Formation module: Off

Propagation: On

Crack Propagation Options

Solution method: Flight_Life

□ Include Time-Dependent Effect

Crack Growth Retardation Effects

□ Include Cyclic □ Include Time-Dependent

Initial Deterministic Anomaly Size

Type

2D Anomaly

Initial Crack Size Options

● Same for all zones

● a and c

○ area

○ Specify by zone

○ Specify by crack type

a (in)	c (in)
0.01	0.01

Note: initial crack size applies only to deterministic crack growth (not risk).

Crack Growth Options

○ Grow crack to service life

● Grow crack to failure

○ Grow crack to target size

Target Crack Size

● a or c (in): 0.02

○ Area (in²): 0.0012566

TMF Crack Growth Algorithm

● Max temperature of flight

○ Max temperature of each stress pair

○ Temperature at max stress

○ Damage rainflow

○ Average da/dN

TMF Crack Growth Algorithm

● Max temperature of flight

○ Max temperature of each stress pair

○ Temperature at max stress

○ Damage rainflow

○ Average da/dN

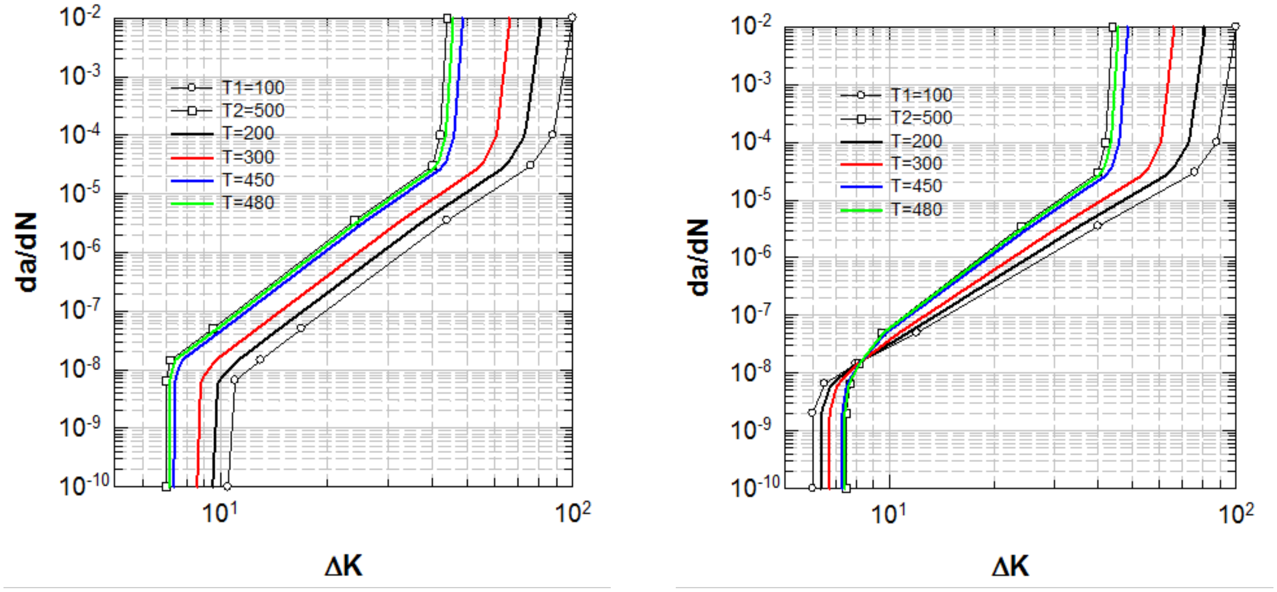
Figure 25. DARWIN 8.0 life model screen showing TMF crack-growth option using the maximum temperature during the flight, in support of AC 33.70-2

3.3 IMPROVED METHODS FOR TEMPERATURE INTERPOLATION OF FCG RATES

Methods for interpolation of cycle-dependent FCG rates as a function of temperature have been available in DARWIN for many years. These methods are necessary because FCG properties are only available at a few discrete temperatures for each material, but service conditions require prediction of FCG rates at other temperatures. The legacy methods in DARWIN are based on interpolation between FCG rate curves for different temperatures at specified *R* values along lines of constant Δ*K*. These methods are computationally efficient and robust in the middle region of FCG curves, where most life is consumed in traditional low-cycle fatigue engine applications. However, it was discovered that these legacy methods could give non-smooth results at the upper and lower ends of the FCG rate curves due to the asymptotic behavior associated with threshold and instability. These difficulties are of relatively little consequence for the majority of traditional DARWIN applications, but the growing use of DARWIN to solve a wider range of problems motivated a decision to implement an improved temperature interpolation method with more robustness.

The new DARWIN temperature interpolation methods for cycle-dependent FCG, which were first introduced in DARWIN 8.2, are based on interpolation along lines of constant *da/dN*. This can be more cumbersome computationally, because FCG equations are written as functions of Δ*K*. The computationally efficient solution path chosen was to convert the FCG equations into multidimensional tables with appropriately fine resolution. This tabulation is performed for all appropriate temperatures and *R* values prior to calling the Flight_Life module for FCG computations. If the original FCG properties were defined in terms of tabular data, some further expansion of this table is still required. The one exception to this protocol is the NASGRO® FCG equation, because in this equation, the threshold can be a function of the crack size, which changes throughout the calculation. Therefore, the NASGRO equation is tabulated inside the Flight_Life module.

The performance of the new interpolation method is shown in figure 26. Here, the interpolation method was originally supplied with fictitious tabulated crack growth rate curves at two temperatures (100°F and 500°F). The method was then used to generate full curves at four intermediate temperatures. The interpolated results are smooth and stable, even when the originally supplied curves cross near threshold.



(a)

(b)

Figure 26. Illustration of new method for temperature interpolation of FCG rates: (a) when bounding curves are approximately parallel, and (b) when bounding curves cross

Note that this method is an empirical one. The user still has the responsibility to confirm that the model behavior is an acceptable representation of actual material response, because different materials may exhibit different mechanisms of temperature sensitivity under different conditions.

3.4 STUDIES OF NEW APPROACHES FOR TIME-DEPENDENT CRACK GROWTH

Fatigue crack growth at elevated temperatures, including time-dependent effects, is a complicated subject. Traditional fracture mechanics methods can sometimes be non-conservative in this regime. First-generation superposition methods for time-dependent crack growth were developed and implemented under a previous grant [22]. First-generation time-dependent retardation methods were implemented during the first year of the current grant. However, the search for improved methods that more accurately predict these high-temperature, time-dependent effects has continued.

Some exploratory work was done in this area under the previous PDRI grant [22]. Though several valuable lessons were learned from that work, and valuable data were generated that can be used going forward, it was ultimately determined that the method proposed by Pratt & Whitney in the PDRI final report would not be a practical way to analyze real aircraft engine hardware. One of the main impediments to practically deploying that method was the requirement that the crack itself needed to be included in the FEM to get accurate values for the various coefficients in the crack-growth model. Though this method can be used to analyze a particular location in a part, it would not be feasible to analyze an entire part, specifically probabilistically, in this way.

Based on the lessons learned under the PDRI grant, work began to simplify the process and to focus on capturing first-order effects. These investigations lead to discussions with Prof. James C. Newman, Jr., at Mississippi State University. Newman's research is centered on experimental and computational aspects of crack growth and fracture behavior of metallic materials to develop material databases, models, and theories for fatigue life, durability and damage tolerance analyses of aging commercial aircraft, and future aerospace materials and structures. He is particularly well-known for his research in SIF solutions and crack closure behavior. Newman is the developer of the well-known FAtigue crack growth STructual ANalysis computer program (FASTRAN) [27] for FCG analysis based on a modified strip-yield model to describe the stress-strain history around the crack tip and the resulting crack closure history.

Newman, through a subcontract from Pratt & Whitney, was tasked to investigate the development of an enhanced strip-yield modeling approach that could incorporate load-time-temperature effects. Though the effort was focused on FASTRAN, the methodology (if successful) could ultimately be ported to DARWIN. This exploratory research effort is documented in appendix E.

The effort identified a path forward for implementation of a time-dependent fatigue crack growth model in FASTRAN, and some preliminary steps to modify FASTRAN for this purpose were completed. The effort also identified some additional material property inputs (i.e., experimental data) that would be needed to feed the new model.

However, several questions remain about the viability of this modeling approach. It is not yet clear if the closure-based model in FASTRAN would be more successful in predicting crack growth rates than the first-generation superposition methods currently used in DARWIN. If time-dependent effects on high-temperature crack growth in nickel-based superalloys are driven more by environmental effects than by creep effects, then the incorporation of time-dependent deformation terms into the crack-tip fields may be less significant.

The project steering committee has decided not to pursue additional efforts in the grant to develop second-generation models for time-dependent crack growth at this time. Individual companies will continue their own investigations, and the issue will be revisited in the future. However, some efforts to generate additional materials data that could be used to evaluate existing or proposed models may be useful.

3.5 TMF MISSION CRACK GROWTH TESTING TO SUPPORT RISC TEST CASE

During the RISC evaluation of the test case for axial blade slots, it became evident that there were some differences of opinion among the different companies about ways of incorporating temperature into a pairing routine to account for out-of-phase temperature loading. In addition, the duty cycle of the RISC test case goes into compression before a tensile dwell and a maximum tensile load are applied. This loading history raises questions about the appropriate method to use to account for these effects.

Because the RISC test case is a notional loading cycle, crack growth data were not readily available that could be used to referee the various methods used to analyze the test case. Pratt & Whitney began work with Georgia Tech that involved, in part, generating test data based on the RISC test-case duty cycle that could be used to referee the various methods OEMs used. Some grant funds were allocated to supplement the effort with a small test program that had been developed by Pratt & Whitney and Georgia Tech.

The experimental procedures and the results of the test program are summarized in appendix F. In short, a standard Pratt & Whitney TMF crack growth test specimen geometry was used. This specimen is a hollow cylinder with a 20-mil electro-discharge machined hole through one side of the specimen. The test plan called for four specimens to be tested under two loading conditions. One set of tests contained a 30-second compressive dwell and 2-minute tensile dwell test with out-of-phase temperature loading. This mission profile most closely mimics the mission currently being used for the RISC slot-bottom test case. The second set of tests contained a 30-second compression dwell test with out-of-phase temperature loading with the strains shifted down such that the maximum strain is zero, resulting in an $R = -\infty$ condition. This test was proposed to bound the problem and see what the physical limits are of this mission profile loading. The specific test loading conditions were based on the surface flaw location identified for the RISC slot bottom test case. The material for these tests is INCO-718. These tests were conducted at Derivation Research Lab (DRL) in Ottawa, Ontario, Canada (which also conducted the other strain-controlled tests in the joint Pratt & Whitney /GA Tech project).

4. RS METHODS

DARWIN has direct access to the FEA of the mechanical and thermal stresses in the uncracked component due to engine operation and can easily use these data when calculating SIF to determine FCG lifetimes. However, other stresses may also be present in the component that may also affect the crack driving force and the lifetime. For example, engineered RS can be introduced on selected surfaces by performing shot peening or laser shock peening. These peening processes generally introduce compressive RS on the surface of the part. The compressive RS tend to delay the initiation of fatigue cracks and/or may tend to decrease the growth rates of fatigue cracks that are initiated on the surface. These are generally desirable outcomes from the perspective of component life management. It is sometimes useful to quantify the impact of these RS on the life and fracture risk.

Usually these RS are not included in the original FEM, but the analyst may have independent quantitative descriptions of the RS that have been induced through peening. DARWIN has now been enhanced to permit the analyst to specify these local RS through the DARWIN GUI. The RS are then linearly superimposed on the service stresses to determine the total stresses. These combined stresses are then used to calculate the SIFs.

Note that RS are static stresses: They do not change during the service cycling, and it is currently assumed that they do change during the lifetime of the component. Therefore, the same RS gradient is added to the maximum and the minimum service stress gradient. This means that the RS changes both the minimum and the maximum SIF (K_{min} and K_{max}), but it does not change the range of the SIF (ΔK). The RS does change the R-ratio (= $K_{\text{min}} / K_{\text{max}}$), and this generally changes the calculated FCG rate.

4.1 LOCAL RS PROFILES FOR 2-D MODELS

An RS feature had been implemented previously in DARWIN for modeling local RS associated with surface treatments in 2-D FEMs, but this feature was originally limited to univariant surface cracks (SC17). In DARWIN 8.0, this feature was extended to the remaining univariant crack-in-plate types associated with 2-D FEMs, including EC05, CC11, TC15, CC08, and SC18. The RS capability was also added to SC30 when SC30 was first introduced in DARWIN 8.1. The RS gradient direction must be aligned with the service stress gradient in these univariant solutions. In the case of the crack-at-hole solutions (e.g., CC08 and SC18), this means that the RS gradient is applied in the bore of the hole and has a radial orientation relative to the hole; it cannot be applied on the surface adjacent to the hole. Univariant through-crack solutions TC01, TC02, and TC13 are not selectable as initial crack types, but if one of the initial univariant solutions transitions to one of these through crack solutions, the specified RS will be included in the post-transition analysis.

The DARWIN GUI permits the visualization of the location where the RS are applied and shows the gradient direction (see figure 27). This “RS region” is characterized as a rectangular box within which RS are defined, and superposition is performed. The RS gradient is aligned with the rectangular axes. Multiple RS regions may be required to represent an RS on a curved surface.

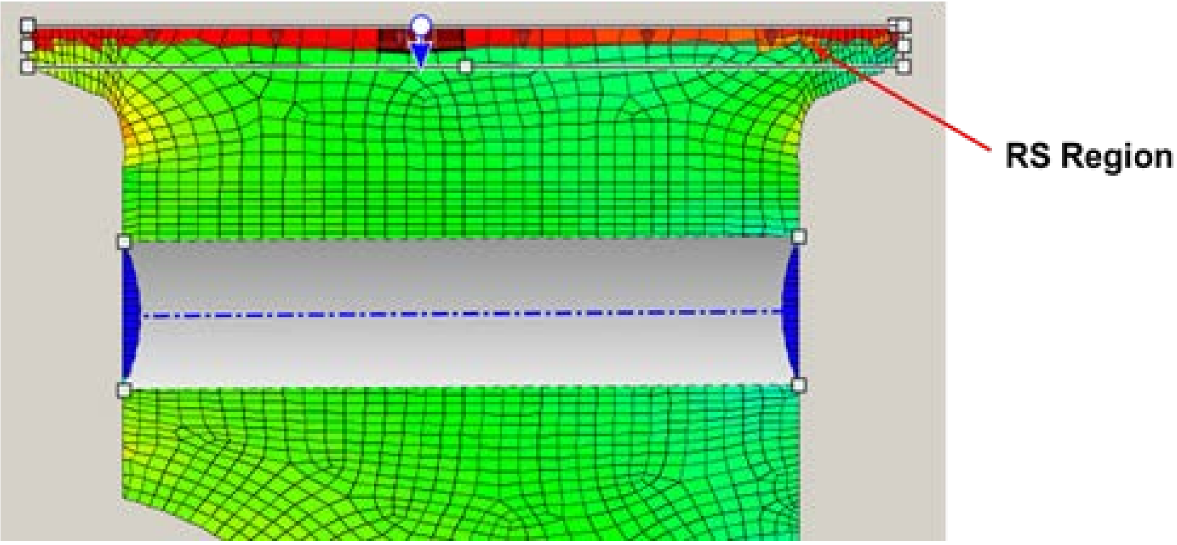
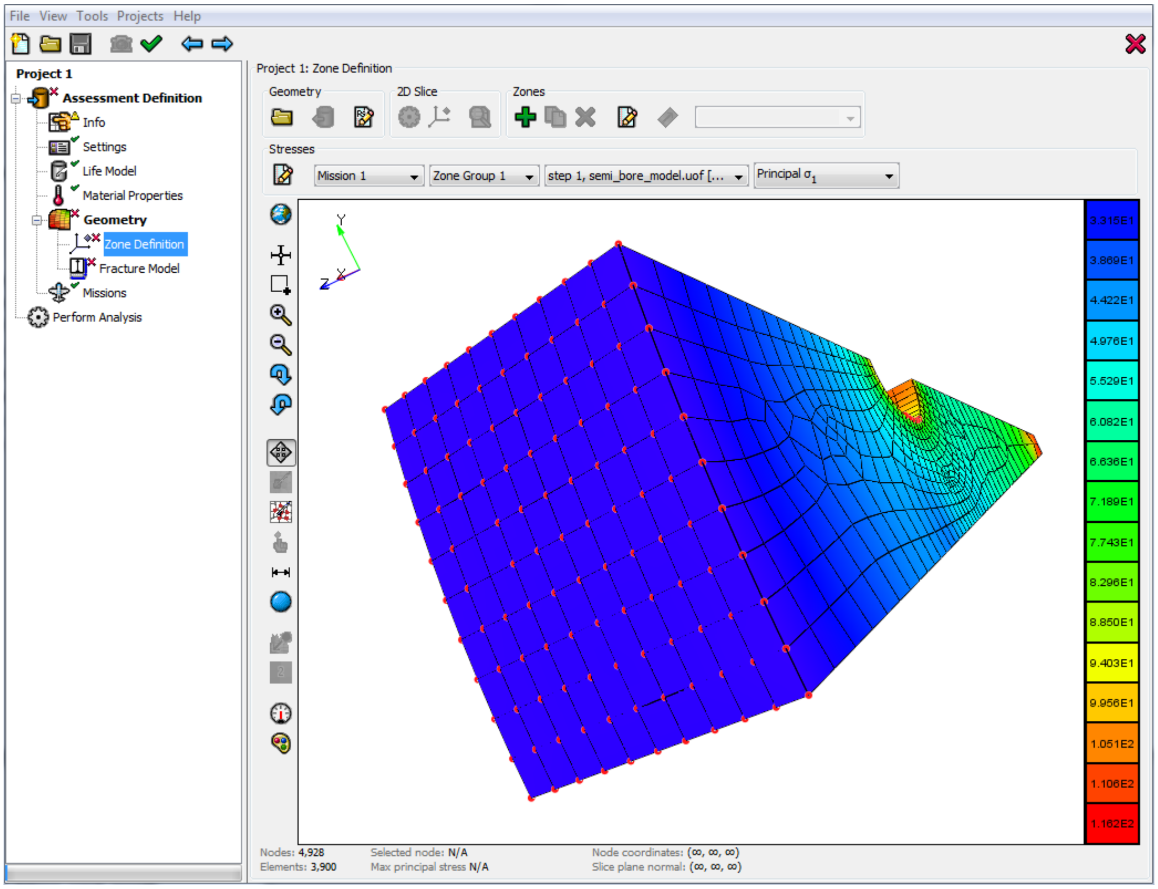


Figure 27. DARWIN GUI visualization of local RS regions

4.2 LOCAL RS PROFILES FOR 3-D MODELS

An RS feature was previously implemented in DARWIN for modeling RS associated with surface treatments in 2-D FEMs based on univariant residual gradients provided by the user. In DARWIN 8.1, this capability was enhanced to support 3-D FEMs. It enables users to define univariant RS gradients and assign them to selected surface nodes in 3-D FEM geometries. The GUI was enhanced to display the selected surface nodes as red dots (see figure 28). The orientation of the RS gradient applied to a node is set to the average orientation of the surface normal vectors associated with the faces of the FEs adjacent to the node. When the user places the crack center at one of these nodes, the RS gradient associated with the node is projected onto the crack plane and superimposed with service stresses for use in FCG life computations. This capability is currently available in the 3-D General Surface Damage mode for univariant crack types SC30, CC11, EC05, and TC15.



Mission 1 | Zone Group 1 | step 1, semi_bore_model.uof [... | Principal σ_1

Nodes: 4,928
Elements: 3,900
Selected node: N/A
Max principal stress N/A
Node coordinates: (∞, ∞, ∞)
Slice plane normal: (∞, ∞, ∞)

Figure 28. RS gradients (represented by red dots) assigned directly to 3-D FE surface nodes

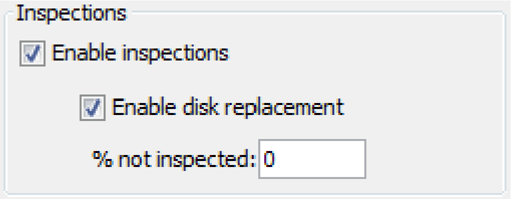
5. FLEET RISK AND PROBABILISTIC METHODS

New capabilities were implemented in DARWIN to support replacement of disks following in-service inspections or as part of a corrective action. These new capabilities are described in sections 5.1 and 5.2, respectively. DARWIN was also enhanced to assess the risk of fracture associated with anomalies that are introduced during shop visits (see section 5.3).

5.1 DARWIN DISK REPLACEMENT CAPABILITY

DARWIN includes a fleet assessment module that can be used to assess the risk of a fleet of aircraft based on probability of fracture results from DARWIN. The fleet assessment computation is based on a fleet of aircraft in which engine components may be replaced during the service life. In previous versions of DARWIN, the risk computation was based on a population of disks rather than aircraft. Disks containing cracks that were detected during nondestructive inspection simulations were removed from the population before failure and were not replaced. This approach neglected the small (but non-zero) additional risk associated with the replacement engine inserted in the aircraft during the remaining part of the aircraft service life.

DARWIN was enhanced to provide a new option for disk replacement following a simulated inspection event. The new GUI feature is shown in figure 29. This new feature enables DARWIN computation of the additional risk associated with replaced components for use in fleet-assessment computations.



Inspections

☑ Enable inspections

☑ Enable disk replacement

% not inspected: 0

Figure 29. Enabling the disk replacement option in the DARWIN GUI

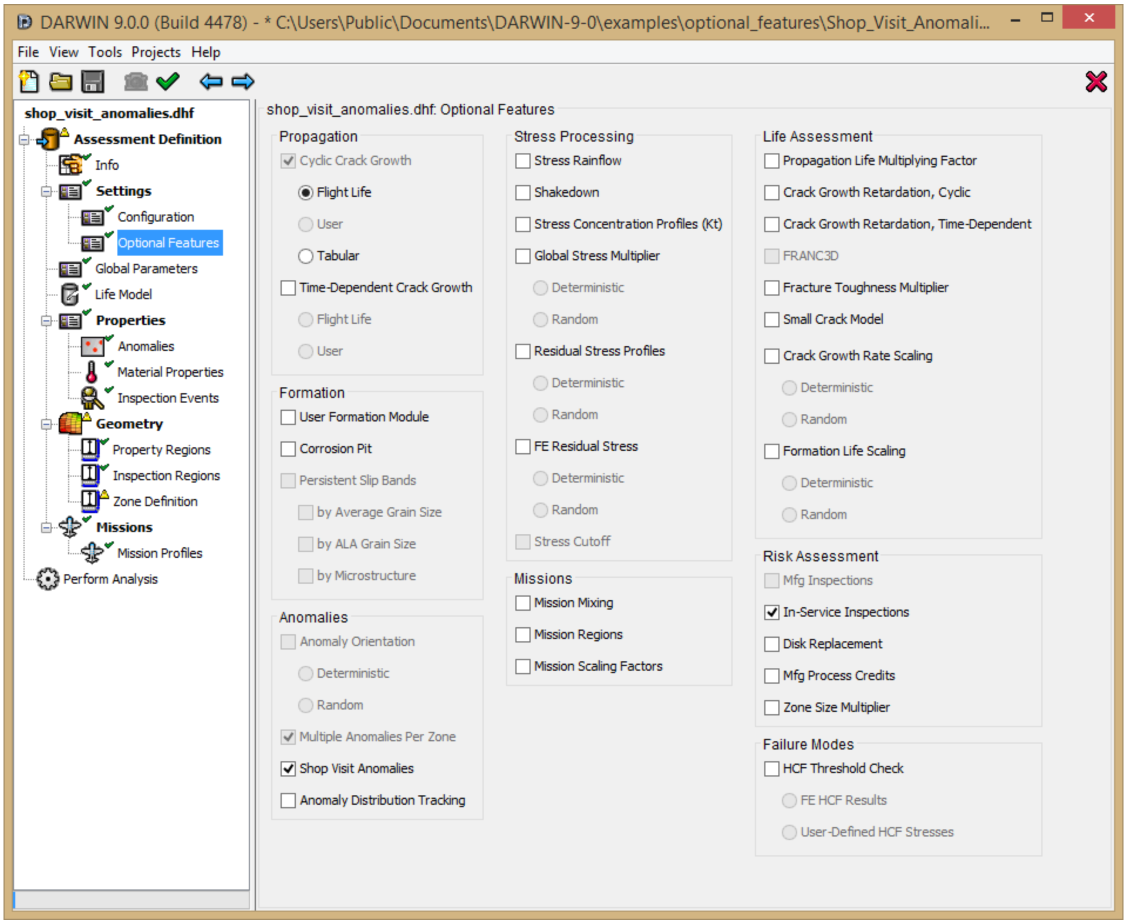
5.2 APPLY CORRECTIVE ACTIONS IN DARWIN FLEET ASSESSMENT MODULE

A new capability was implemented for application of corrective actions in the DARWIN Fleet Assessment Module. In particular, this new capability enables replacement of existing disks with new disks that may have completely different geometries. Unlike the previous version, in which disks were replaced only when they failed an inspection, the new version enables new disks to be placed in service based on the schedule associated with a disk-replacement corrective action. Additional DARWIN runs are required to assess the risk of disks associated with each time point in the corrective action replacement schedule, and the risk results are combined to display the associated fleet risk.

5.3 ACCOMMODATE NEW ANOMALY APPEARANCE AT SHOP VISIT OR IN SERVICE

Anomalies may sometimes be introduced on the surfaces of a component that is handled or inspected during routine maintenance, such as assembly or disassembly operations at attachment slots. For example, an anomaly might not exist until cycle 5000, when it is introduced during an inspection. Introducing these anomalies analytically at the beginning of the service life may result in an overestimate of the risk of fracture. Shop-visit anomalies are produced on the surface (not the interior) of components and may require some non-zero number of cycles to form a crack. Furthermore, the timing of shop-visit anomaly creation may or may not be aligned with inspection schedules.

DARWIN 8.1 (and previous versions) was limited to anomalies present before the part entered service. In DARWIN 8.2, users have the option to define shop-visit anomalies for improved risk assessments. This feature is available for surface damage investigations of one dimensional, 2-D, and 3-D geometries. Figure 30 shows available options for shop-visit anomaly investigations. By default, multiple anomalies are enabled in each zone when shop-visit anomalies are active. In-service inspections remain an optional feature with shop-visit anomalies.



shop_visit_anomalies.dhf: Optional Features

Propagation

- ☑ Cyclic Crack Growth
- ● Flight Life
- ○ User
- ○ Tabular
- ☐ Time-Dependent Crack Growth
- ○ Flight Life
- ○ User

Formation

- ☐ User Formation Module
- ☐ Corrosion Pit
- ☐ Persistent Slip Bands
- ☐ by Average Grain Size
- ☐ by ALA Grain Size
- ☐ by Microstructure

Anomalies

- ☐ Anomaly Orientation
- ○ Deterministic
- ○ Random
- ☑ Multiple Anomalies Per Zone
- ☑ Shop Visit Anomalies
- ☐ Anomaly Distribution Tracking

Stress Processing

- ☐ Stress Rainflow
- ☐ Shakedown
- ☐ Stress Concentration Profiles (Kt)
- ☐ Global Stress Multiplier
- ○ Deterministic
- ○ Random
- ☐ Residual Stress Profiles
- ○ Deterministic
- ○ Random
- ☐ FE Residual Stress
- ○ Deterministic
- ○ Random
- ☐ Stress Cutoff

Missions

- ☐ Mission Mixing
- ☐ Mission Regions
- ☐ Mission Scaling Factors

Life Assessment

- ☐ Propagation Life Multiplying Factor
- ☐ Crack Growth Retardation, Cyclic
- ☐ Crack Growth Retardation, Time-Dependent
- ☐ FRANC3D
- ☐ Fracture Toughness Multiplier
- ☐ Small Crack Model
- ☐ Crack Growth Rate Scaling
- ○ Deterministic
- ○ Random
- ☐ Formation Life Scaling
- ○ Deterministic
- ○ Random

Risk Assessment

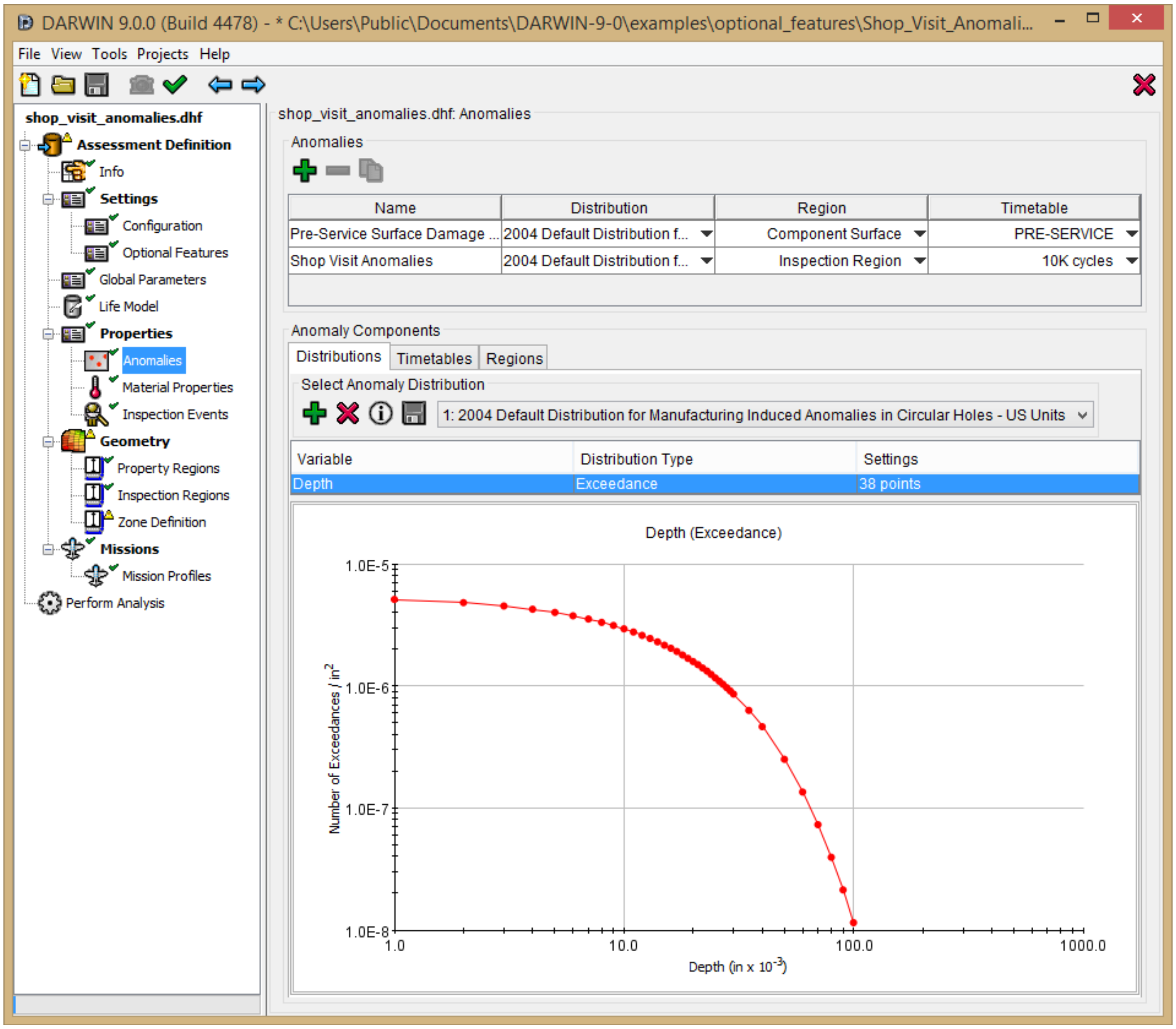
- ☐ Mfg Inspections
- ☑ In-Service Inspections
- ☐ Disk Replacement
- ☐ Mfg Process Credits
- ☐ Zone Size Multiplier

Failure Modes

- ☐ HCF Threshold Check
- ○ FE HCF Results
- ○ User-Defined HCF Stresses

Figure 30. Available optional features for use with shop-visit anomalies

A new GUI menu has been introduced that enables the user to specify the size distribution and timetables associated with shop-visit anomalies. When shop-visit anomalies are enabled, each anomaly requires an identifier, size distribution, and a timetable. Figure 31 shows the new anomaly distribution menu where size distributions are imported and visualized. Figure 32 shows the timetable menu that has been added to the anomaly distribution tab for shop-visit anomalies. Shop-visit timetables may be linked to timetables of inspection schedules if desired. The user can specify whether an inspection is performed before or after a shop-visit anomaly via the Global Factors menu. When cracks form sometime after the anomaly is introduced (i.e., the crack formation option is active), then the anomaly definition requires the information regarding both formation case and formation scatter (see figure 33).



DARWIN 9.0.0 (Build 4478) - * C:\Users\Public\Documents\DARWIN-9-0\examples\optional_features\Shop_Visit_Anomali...

shop_visit_anomalies.dhf: Anomalies

Anomalies

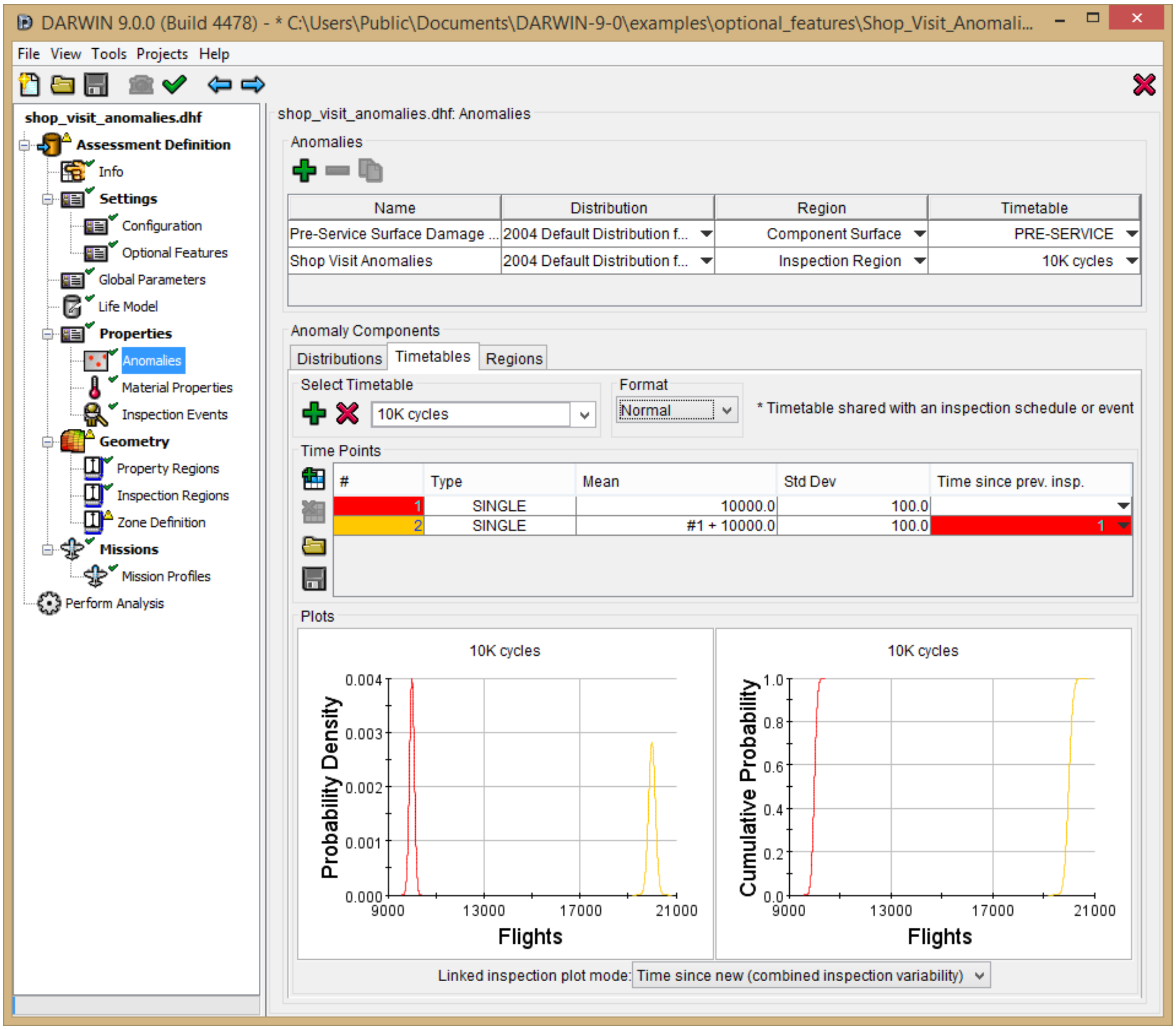
Name	Distribution	Region	Timetable
Pre-Service Surface Damage ...	2004 Default Distribution f...	Component Surface	PRE-SERVICE
Shop Visit Anomalies	2004 Default Distribution f...	Inspection Region	10K cycles

Anomaly Components

Select Anomaly Distribution: 1: 2004 Default Distribution for Manufacturing Induced Anomalies in Circular Holes - US Units

Variable	Distribution Type	Settings
Depth	Exceedance	38 points

Figure 31. Enhanced GUI supports definition of shop-visit anomalies



Name	Distribution	Region	Timetable
Pre-Service Surface Damage ...	2004 Default Distribution f...	Component Surface	PRE-SERVICE
Shop Visit Anomalies	2004 Default Distribution f...	Inspection Region	10K cycles

Select Timetable: 10K cycles

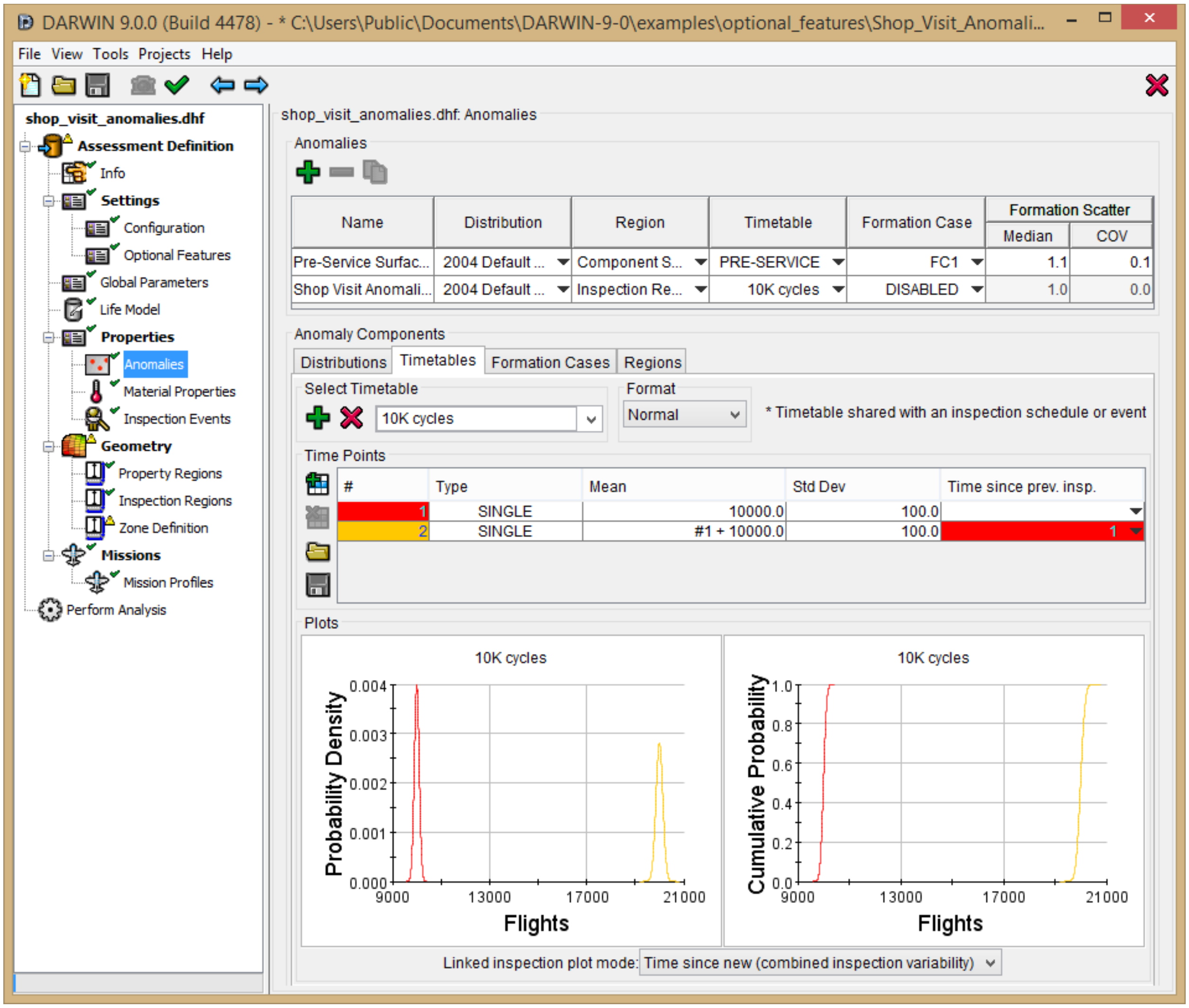
Format: Normal

* Timetable shared with an inspection schedule or event

#	Type	Mean	Std Dev	Time since prev. insp.
1	SINGLE	10000.0	100.0	
2	SINGLE	#1 + 10000.0	100.0	1

Linked inspection plot mode: Time since new (combined inspection variability)

Figure 32. Enhanced timetable definition capability associated with shop-visit anomalies



Name	Distribution	Region	Timetable	Formation Case	Formation Scatter	
					Median	COV
Pre-Service Surfac...	2004 Default ...	Component S...	PRE-SERVICE	FC1	1.1	0.1
Shop Visit Anomali...	2004 Default ...	Inspection Re...	10K cycles	DISABLED	1.0	0.0

Select Timetable: 10K cycles

Format: Normal

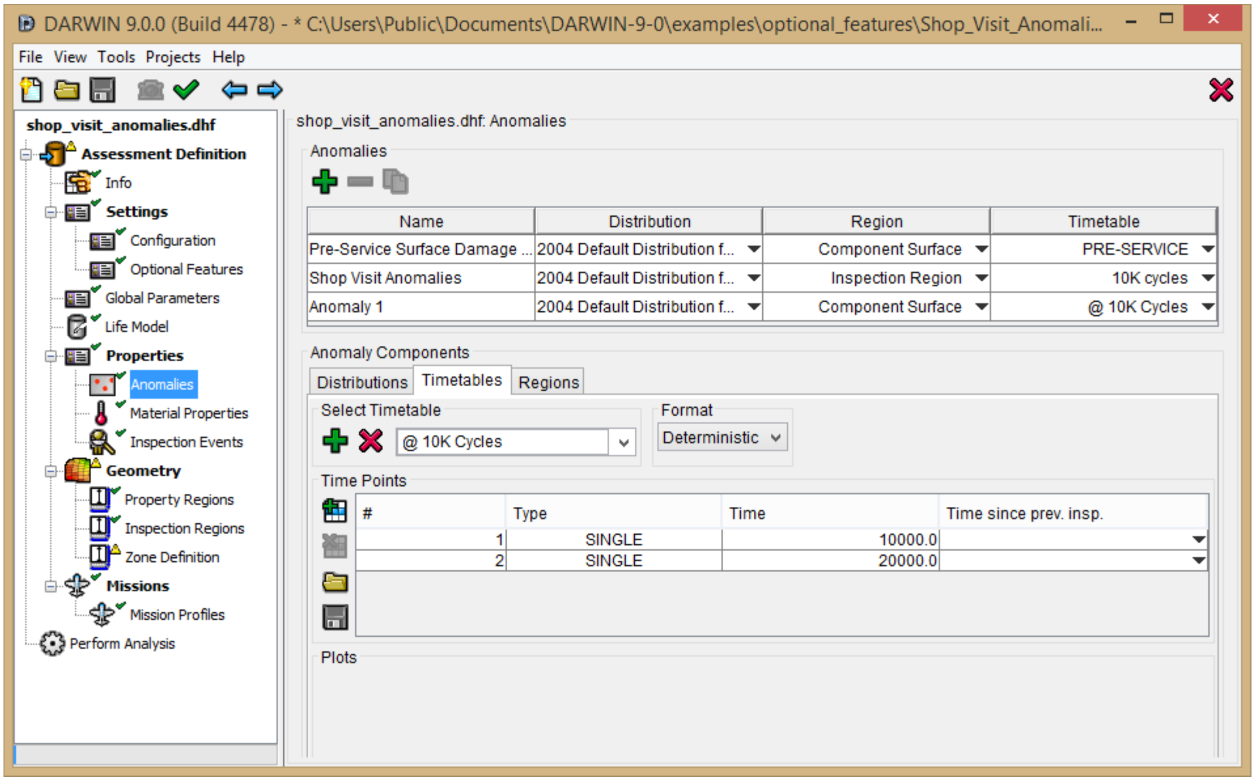
* Timetable shared with an inspection schedule or event

#	Type	Mean	Std Dev	Time since prev. insp.
1	SINGLE	10000.0	100.0	
2	SINGLE	#1 + 10000.0	100.0	1

Linked inspection plot mode: Time since new (combined inspection variability)

Figure 33. Shop-visit crack formation case and formation scatter definition

The timetable menu has been enhanced in DARWIN 8.2 to provide new methods to describe inspection events/shop visits. This enables users to define a deterministic schedule (i.e., events occur at known times in the flight history) (see figure 34). As in previous DARWIN versions, subsequent events can be scheduled relative to prior events. Figure 35 shows four deterministic events in which the second event occurs at a fixed number of flights from the first event. The third and fourth events are linked to the second and third events similarly. Figure 36 demonstrates a similar schedule with Weibull distributions used to represent the probability of an inspection event or shop visit occurring in a certain timeframe. The GUI indicates when a timetable is shared with an inspection (see figure 32). Here, the timetable for the anomaly “Shop Visit Anomaly” matches a timetable used for inspections. This is indicated by the message located to the right of the format dropdown box. Timetables that are not shared with an inspection schedule do not display this message (see figures 34–36).



shop_visit_anomalies.dhf: Anomalies

Anomalies

Name	Distribution	Region	Timetable
Pre-Service Surface Damage ...	2004 Default Distribution f...	Component Surface	PRE-SERVICE
Shop Visit Anomalies	2004 Default Distribution f...	Inspection Region	10K cycles
Anomaly 1	2004 Default Distribution f...	Component Surface	@ 10K Cycles

Anomaly Components

Distributions | Timetables | Regions

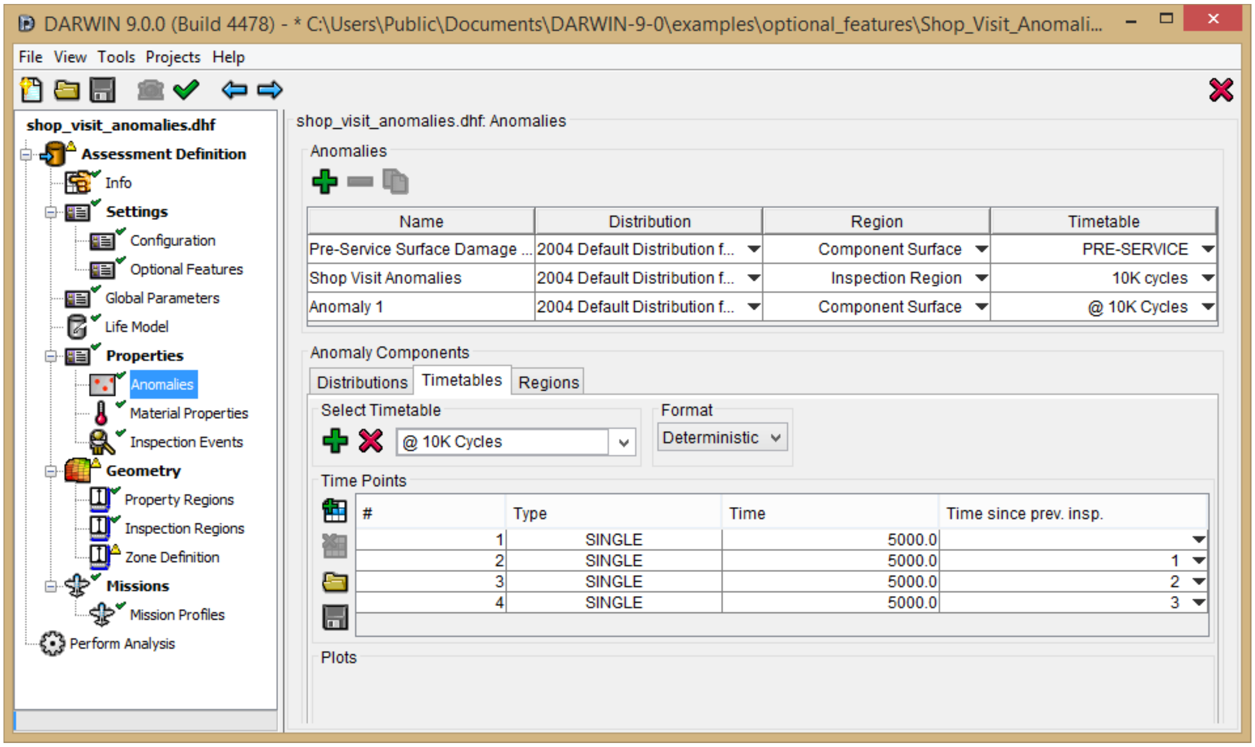
Select Timetable: @ 10K Cycles

Format: Deterministic

Time Points

#	Type	Time	Time since prev. insp.
1	SINGLE	10000.0	
2	SINGLE	20000.0	

Figure 34. Example timetable showing unlinked deterministic shop-visit times



shop_visit_anomalies.dhf: Anomalies

Anomalies

Name	Distribution	Region	Timetable
Pre-Service Surface Damage ...	2004 Default Distribution f...	Component Surface	PRE-SERVICE
Shop Visit Anomalies	2004 Default Distribution f...	Inspection Region	10K cycles
Anomaly 1	2004 Default Distribution f...	Component Surface	@ 10K Cycles

Anomaly Components

Distributions | Timetables | Regions

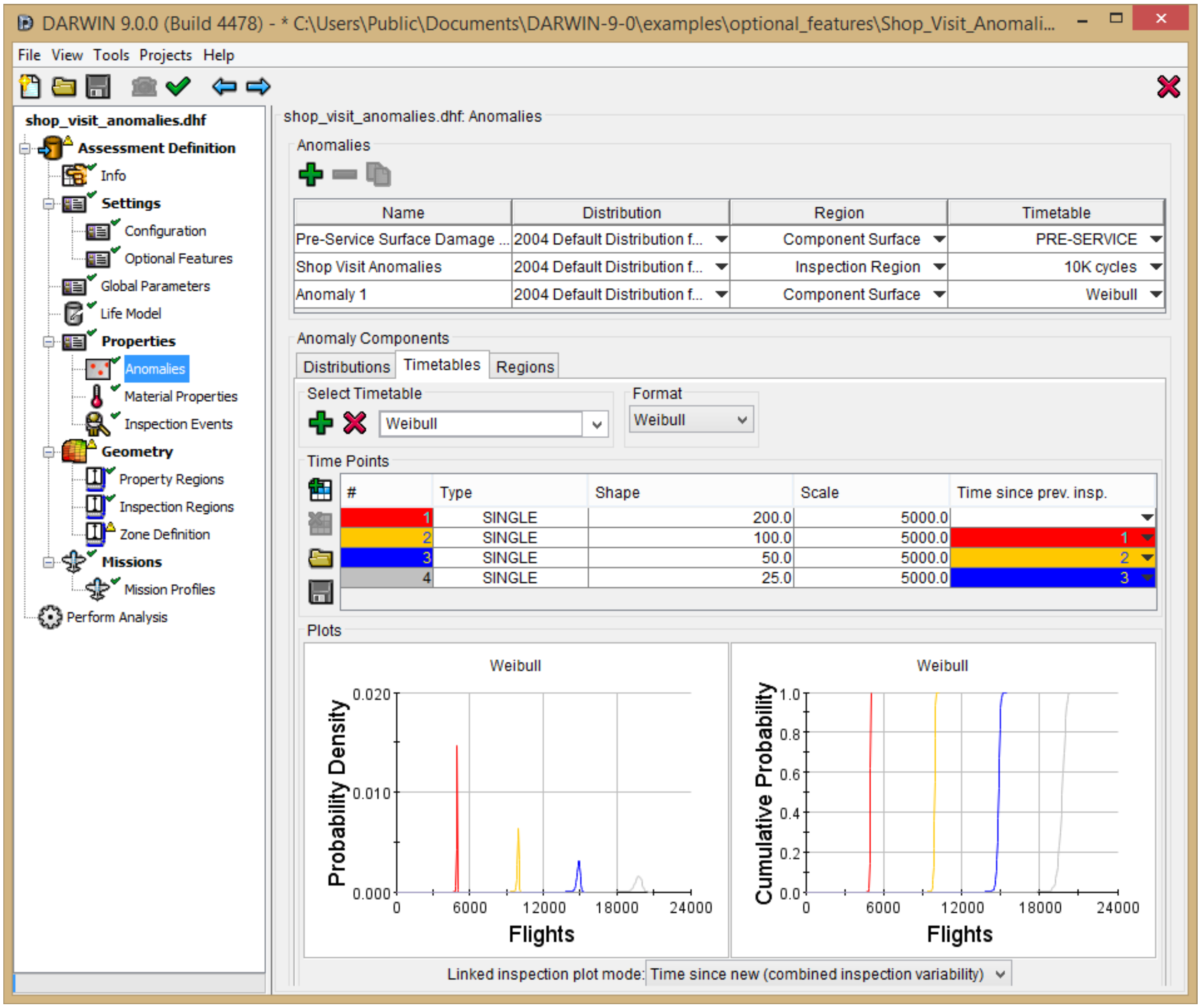
Select Timetable: @ 10K Cycles

Format: Deterministic

Time Points

#	Type	Time	Time since prev. insp.
1	SINGLE	5000.0	
2	SINGLE	5000.0	1
3	SINGLE	5000.0	2
4	SINGLE	5000.0	3

Figure 35. Example timetable showing linked deterministic shop-visit times



shop_visit_anomalies.dhf: Anomalies

Anomalies

Name	Distribution	Region	Timetable
Pre-Service Surface Damage ...	2004 Default Distribution f...	Component Surface	PRE-SERVICE
Shop Visit Anomalies	2004 Default Distribution f...	Inspection Region	10K cycles
Anomaly 1	2004 Default Distribution f...	Component Surface	Weibull

Anomaly Components

Distributions | Timetables | Regions

Select Timetable: Weibull

Format: Weibull

Time Points

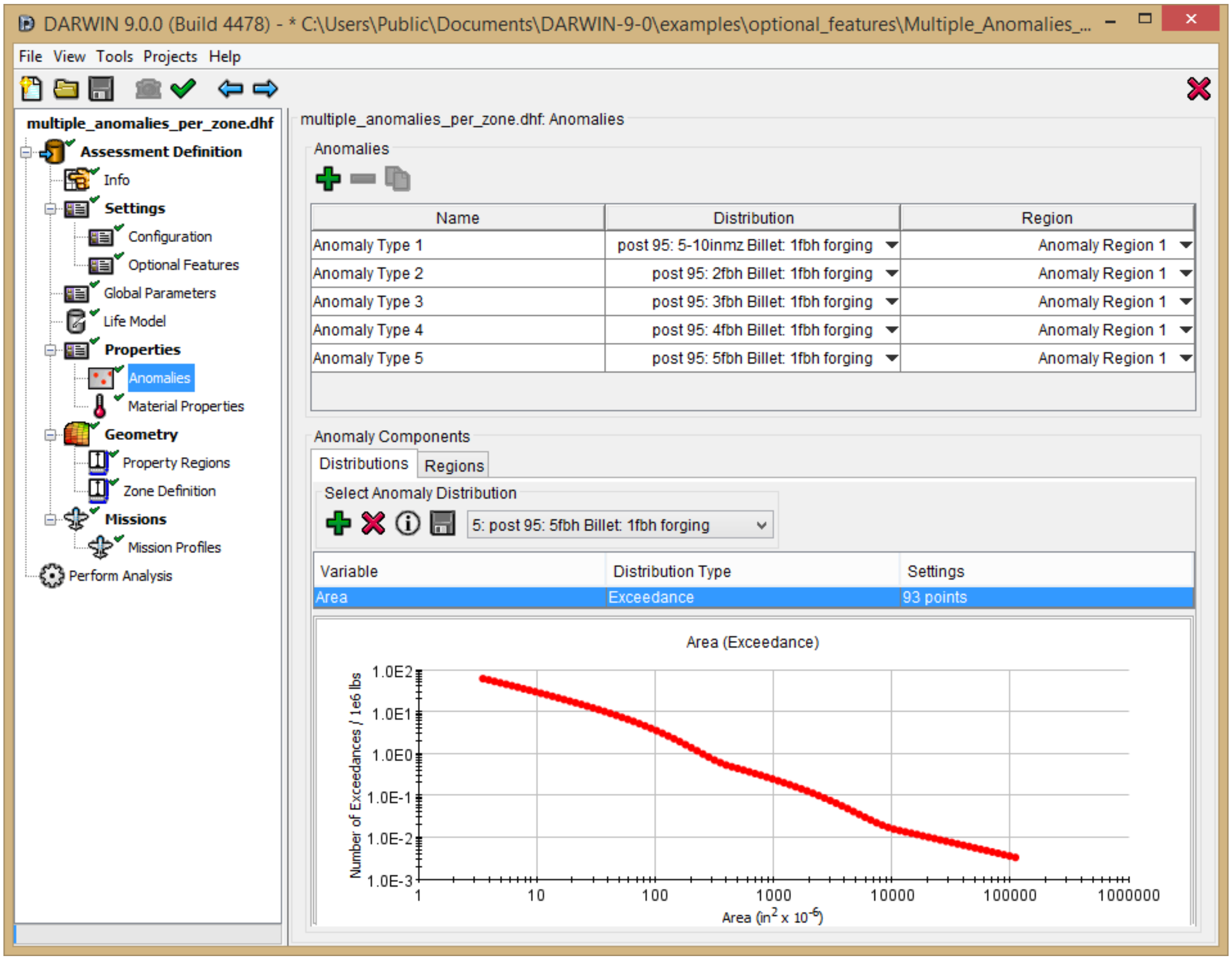
#	Type	Shape	Scale	Time since prev. insp.
🟥 1	SINGLE	200.0	5000.0	
🟨 2	SINGLE	100.0	5000.0	🟥 1
🟦 3	SINGLE	50.0	5000.0	🟨 2
4	SINGLE	25.0	5000.0	🟦 3

Plots

Linked inspection plot mode: Time since new (combined inspection variability)

Figure 36. Example timetable showing Weibull distributions

As a result of this flexibility, DARWIN 8.2 enables several combinations of inspection and shop-visit anomaly events. Users might define unrelated schedules for inspections/shop visits. For example, inspections might occur every 5000 cycles, whereas shop-visit anomalies are introduced every 8000 cycles using a deterministic schedule. Alternatively, users could establish two schedules: one for inspections and a second for maintenance. The inspection schedule could be linked to a shop-visit anomaly with a size distribution calibrated for anomalies introduced during inspection. The maintenance schedule might reflect the introduction of shop-visit anomalies of a different size distribution (calibrated for anomalies introduced during maintenance) but without any associated inspections. Finally, the component might have an initial anomaly distribution before the part enters service. All three anomaly distributions can be included in the same zone through the anomaly mixer (see figure 37).



Name	Distribution	Region
Anomaly Type 1	post 95: 5-10inmz Billet: 1fbh forging	Anomaly Region 1
Anomaly Type 2	post 95: 2fbh Billet: 1fbh forging	Anomaly Region 1
Anomaly Type 3	post 95: 3fbh Billet: 1fbh forging	Anomaly Region 1
Anomaly Type 4	post 95: 4fbh Billet: 1fbh forging	Anomaly Region 1
Anomaly Type 5	post 95: 5fbh Billet: 1fbh forging	Anomaly Region 1

Select Anomaly Distribution: 5: post 95: 5fbh Billet: 1fbh forging

Variable	Distribution Type	Settings
Area	Exceedance	93 points

Figure 37. Multiple anomaly distributions can be assigned to the same zone

DARWIN tracks risk associated with shop visit anomalies and quantifies its influence on the overall probability of fracture. Figure 38 highlights total risk, risk from the pre-service anomaly, and risk from the shop-visit anomaly. Fracture probabilities are computed with and without inspection. These results can be broken up zone by zone to focus on hotspots, with risk results plotted by zone, anomalies, and times (see figure 39). Again, the effect of inspections can be plotted in these results. Finally, DARWIN plots life information associated with cracks resulting from shop-visit anomalies. Figure 40 shows values of the SIF for a crack resulting from a shop-visit anomaly introduced at two time points. In this example, the anomalies have the same initial size and, therefore, lead to identical but shifted driving force curves.

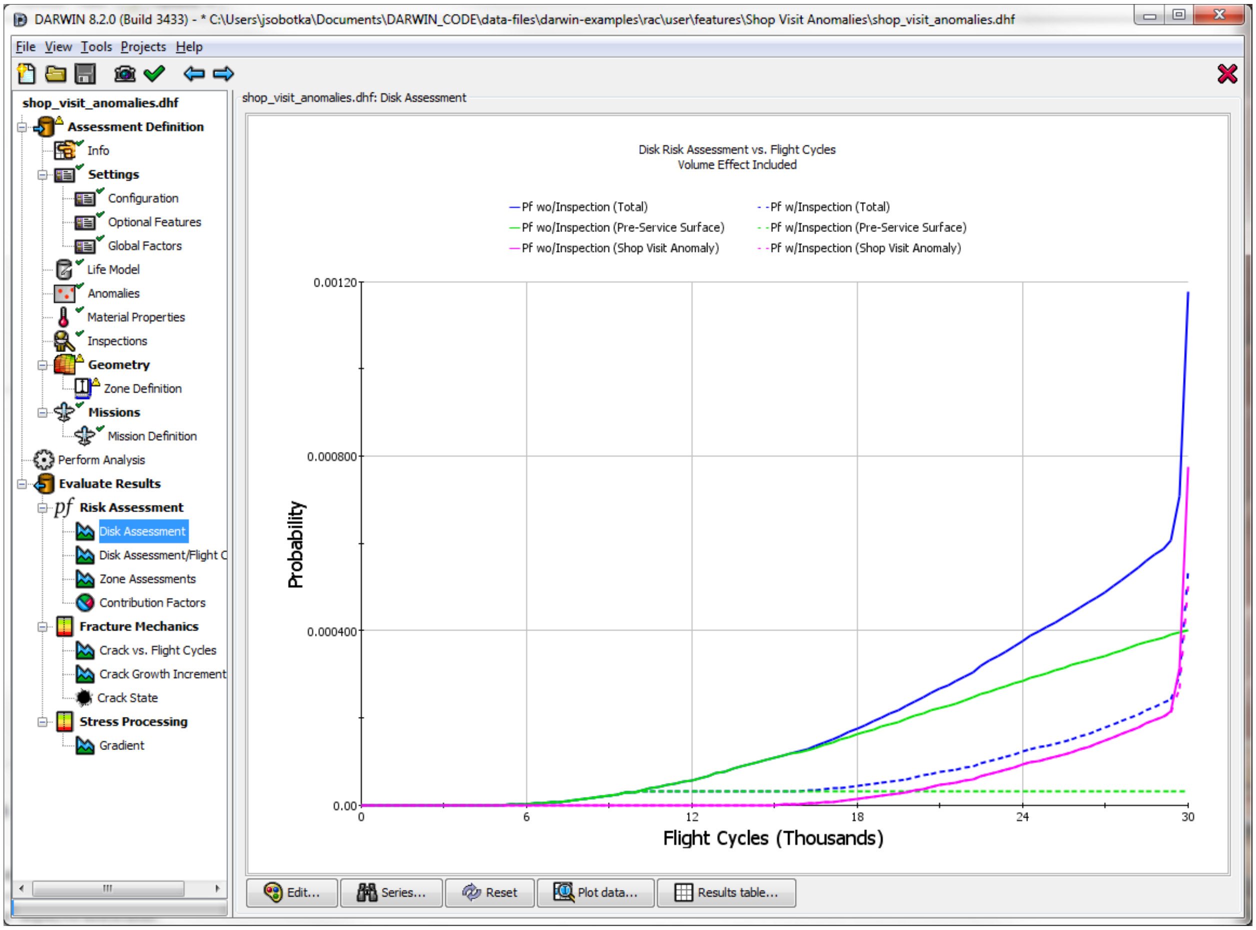
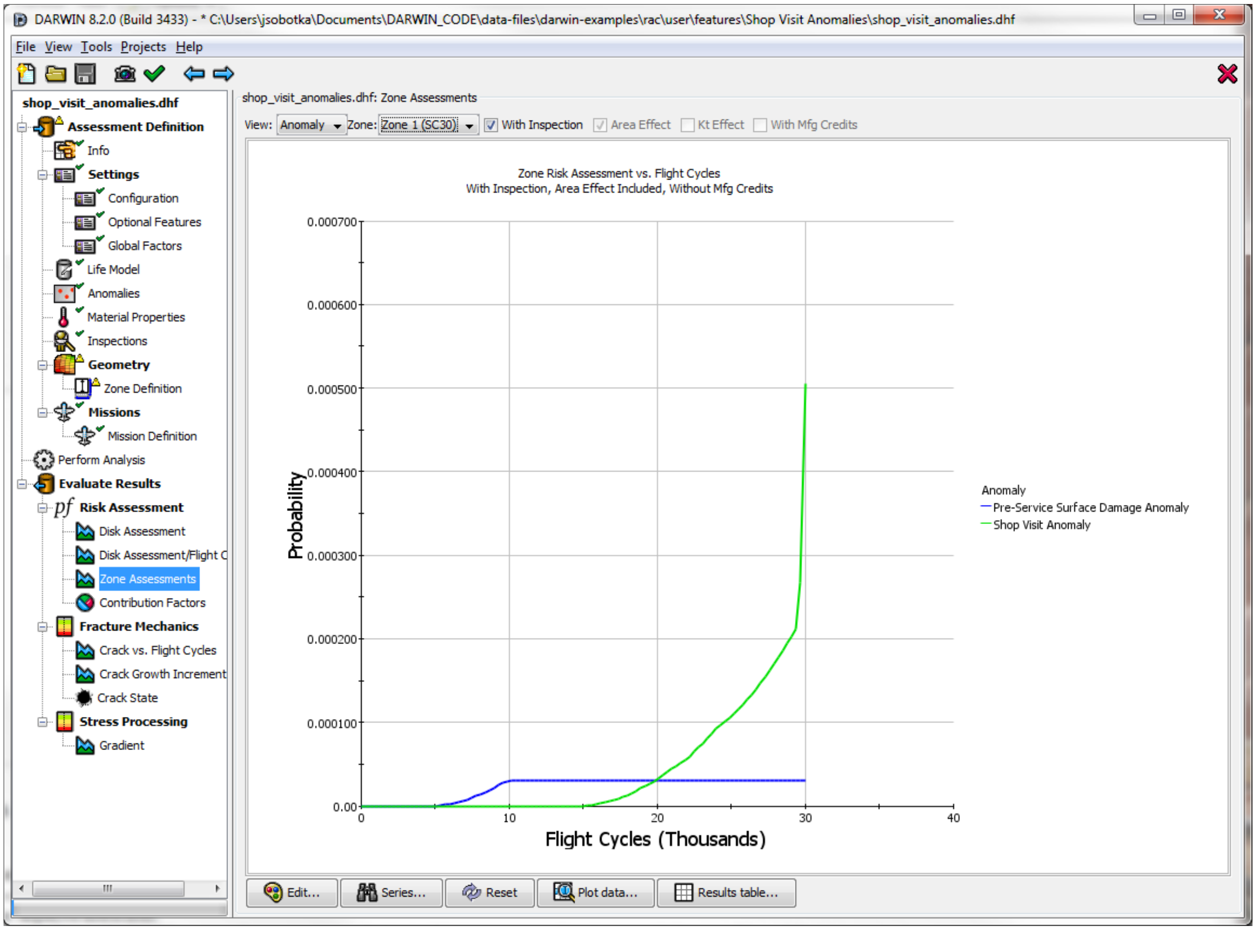


Figure 38. DARWIN 8.2 displays the probability of fracture with and without shop-visit anomalies

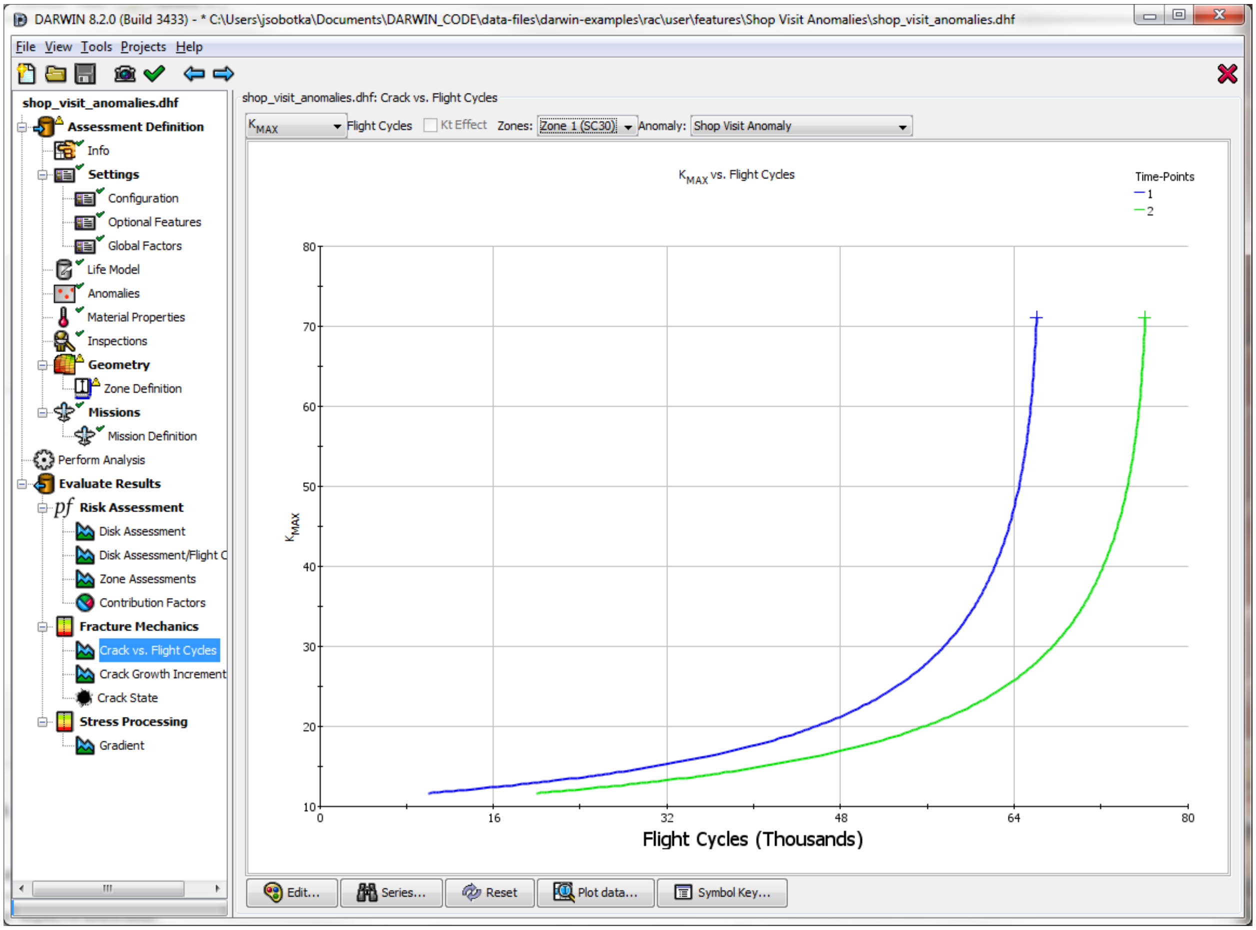


DARWIN 8.2.0 (Build 3433) - * C:\Users\jsobotka\Documents\DARWIN_CODE\data-files\darwin-examples\rac\user\features\Shop Visit Anomalies\shop_visit_anomalies.dhf

shop_visit_anomalies.dhf: Zone Assessments

View: Anomaly Zone: Zone 1 (SC30) ☑ With Inspection ☑ Area Effect □ Kt Effect □ With Mfg Credits

Figure 39. DARWIN 8.2 enables users to visualize contributions of fracture probabilities by zone, anomalies, and times



DARWIN 8.2.0 (Build 3433) - * C:\Users\jsobotka\Documents\DARWIN_CODE\data-files\darwin-examples\rac\user\features\Shop Visit Anomalies\shop_visit_anomalies.dhf

shop_visit_anomalies.dhf: Crack vs. Flight Cycles

K_{MAX} Flight Cycles □ Kt Effect Zones: Zone 1 (SC30) Anomaly: Shop Visit Anomaly

Figure 40. In DARWIN 8.2, life estimates are broken down by anomaly and by time when the anomaly is introduced

6. ADVANCED FRACTURE ANALYSIS

Progress in advanced fracture analysis included the development and implementation in DARWIN of five new SIF solutions.

6.1 NEW SIF SOLUTION FOR A SURFACE CRACK AT AN OFF-CENTER HOLE UNDER BIVARIANT STRESSING

A new bivariant crack model SC29 was developed to determine the SIF at both surface tips and the maximum depth tip location of a surface crack in the bore of a round hole. The surface crack is described by a semi-ellipse with surface tips along the bore and the deepest tip along the plate width direction. Figure 41 shows the geometric configuration for this crack model, where the center location is designated by the crack offset along the bore, (x_d, y_d), in reference to the local *x*-*y* coordinate system and the offset of the bore, $h_x + R$, in reference to the side surface of the plate, where R is the radius of the bore and h_x is the smaller net section width resulting from the hole. Note the crack is always assumed to initiate at the smaller net section. The remaining geometric parameters are the thickness of the plate (h_y) and the larger net section width (h_{x2}). There are six dimensionless geometric parameters describing this cracked configuration: R/h_y, a/c, $(h_{x2} – h_x)/(h_{x2} + h_x +2R)$, a/h_x , y_d /h_y , and c/y_d.

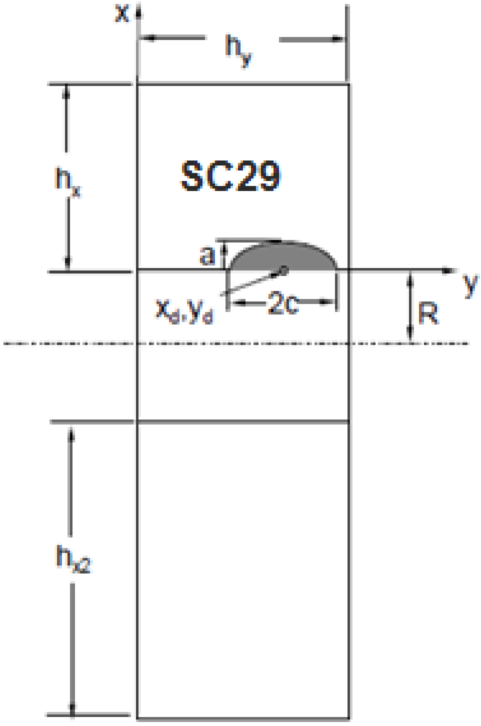


Figure 41. Geometry configuration for SC29 crack model

The applicable load condition is the bivariant stress variation across the net section plane prior to cracking. This is the variation of the stress component perpendicular to the net section plane (crack opening plane). This stress variation is determined from numerical models without cracks. Figure 42 is a schematic description of the stresses on the uncracked net section.

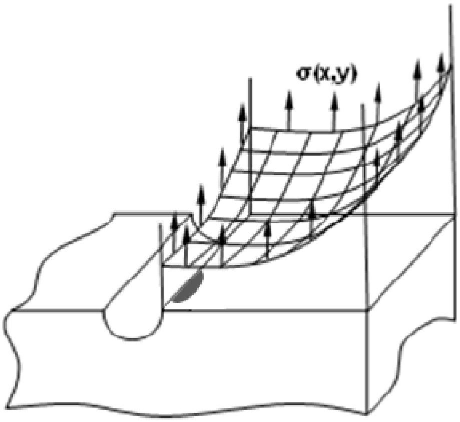


Figure 42. Schematic representation of bivariant stressing for SC29 crack model

The SC29 SIF solution is based on a WF approach that requires a point weight function (PWF) and a set of reference solutions derived based on three reference stresses. The PWF is the same as used by the SC31 crack model for a bivariant surface crack in a rectangular cross-section of a plate. The mathematical form of the PWF is given in the DARWIN Theory Manual. The PWF determines the effect of the stress applied at the point of interest on the crack surface to the crack tip location along the crack front. The function contains three undetermined variables that require using the reference solutions to calibrate for various geometric aspect ratios. The bivariant SIF solution can theoretically be a function of the parametric crack-tip location along the crack front. However, because the primary interest is focused on the two surface tip locations and the maximum depth tip location, and the discrete reference solutions are consistently extracted at these three crack-tip locations, the computed SIF formulation given in equation 1 is for two surface tips and the maximum depth tip.

$$K = \int_0^a \int_{-c\sqrt{1-y^2/a^2}}^{c\sqrt{1-y^2/a^2}} \sigma(x,y) \cdot W_{QQ'} dxdy \tag{1}$$

In the above expression, the crack-tip locations are designated by Q' and the location at the crack surface is designated by $Q(x,y)$. $W_{QQ'}$ is the PWF contributing its effect with the stress at Q to the crack tip location Q', and $\sigma(x,y)$ is the stress at $Q(x,y)$. Note the coordinate system defined in the above WF solution is different from the one depicted in figure 41. In this local coordinate system, the center of the surface crack defines the origin with both the deepest tip (a-tip) and the surface tip (c-tip) designating the *y*- and *x*-axes (contrary to the *x*- and *y*-axes in figure 41).

The reference solutions consist of normalized SIF results determined at the three crack-tip locations (one maximum depth and two surface tips) for a surface crack subjected to three different reference loads. The solutions were numerically determined using FADD3D, a numerical program based on boundary element method specifically for fracture mechanics usage with the aid of a modeling tool, PATRAN, and a utility software developed by SwRI to convert input files from ABAQUS format to FADD3D format. The reference loads are all in terms of simple loading and applied on the crack surface. Respectively, they are: 1) uniform tension, $\sigma(x,y) = 1$; 2) unit bending along the plate width direction, $\sigma(x,y) = 1 - y/a$; and 3) unit bending along the thickness direction, $\sigma(x,y) = 1 + x/c$. The solution space is discrete, resulting from various combinations of geometric aspect ratios. Because of the large number of models resulting from the six independent geometric aspect ratio parameters to account for every geometric combination, not all reference solutions were determined, and only select geometric aspect ratios were used. Table 1 lists the geometric aspect ratios at which the reference solutions were determined. The parameters used in table 1 were slightly different from those in figure 41. They were defined and used to facilitate numerical modeling; for example, two offsets (i.e., *T*, the offset of the crack center from the mid-plane of the plate, and *B*, the offset of the hole from the center of the plate) are in reference to the central locations. Relating them to the geometric parameters in figure 41 is straightforward:

$$t = \frac{h_y}{2} \tag{2}$$

$$W = h_x + 2R + h_{x2} \tag{3}$$

$$b = \frac{w}{2} \tag{4}$$

$$T = y_d - t \tag{5}$$

$$B = b - (h_x + R) \tag{6}$$

Eight hundred numerical models were generated (see table 1). After accounting for the three reference stresses, the total number of computations amounted to 2400. The reference solutions whose geometric aspect ratios are not specifically included in table 1 were determined through interpolation. To use the discrete reference solutions for SIFs during fatigue crack growth computation, the first step is to reduce the six degrees-of-freedom (DOFs) embedded within the reference stress solution arrays to four DOFs with finer resolution for easier interpolation. The reduction is based on the given R/t and B/b values associated with the initial crack definition, because these two aspect ratios remain unchanged during the computation. The interpolated reference solutions are then used to define the unknowns in the PWF, and a 2-D pre-integration routine evaluates the surface integral for the SIF results. For a new configuration (i.e., new offset or radius of the hole), a new set of finer reference solutions with four DOFs needs to be regenerated to facilitate the computation.

Table 1. Geometric aspect ratios at which reference solutions were determined

$\frac{R}{t}$	$\frac{c}{a}$	$\frac{B}{b}$	$\frac{c}{b-(B+R)}$	$\frac{T}{t}$	$\frac{a}{t-T}$	Number of Models
0.25	0.2 0.4 1.0 2.0	0	0.05 0.1 0.2 0.5 0.8	0 0.5 0.8 0.95	0.1 0.5 0.8 0.95	320
0.25	0.2 0.4 1.0 2.0	0.8	0.1 0.8	0 0.8 0.95	0.1 0.5 0.8 0.95	96
1.0 2.0	0.2 0.4 1.0 2.0	0 0.8	0.1 0.8	0 0.8 0.95	0.1 0.5 0.8 0.95	384

The validity of the SIF solutions is bounded by the solution limits associated with the reference solutions. Some may be slightly extended in view of the solution behavior near the solution limits. Respectively, the validity limits for the six geometric ratios are listed as follows:

$$0.25 \le \frac{R}{t} = \frac{2R}{h_y} \le 2 \tag{7}$$

$$0.2 \le \frac{a}{c} \le 2 \tag{8}$$

$$0 \le \frac{B}{b} = \frac{h_{x2} - h_x}{h_x + 2R + h_{x2}} \le 0.8 \tag{9}$$

$$\frac{a}{h_x} = \frac{a}{b-(B+R)} \le 0.8 \tag{10}$$

$$-0.95 \le \frac{T}{t} = \frac{2y_d}{h_y} - 1 \le 0.95 \tag{11}$$

$$\frac{c}{t-|T|} \le 0.95, \text{ or } \frac{c}{min(h_y - y_d, y_d)} \le 0.95 \tag{12}$$

The SC29 SIF solution was first implemented in DARWIN 8.0.

6.2 UPDATED SIF SOLUTIONS FOR A SURFACE CRACK IN A PLATE

Previous versions of DARWIN included univariant (SC17) and bivariant (SC19) WF SIF solutions for a semi-elliptical surface crack offset in a rectangular plate. These solutions, which were developed in 2003 and first implemented in DARWIN 5.1, were also used in the NASGRO fracture mechanics software (which is also developed by SwRI). Further studies of these solutions found that they were over-conservative for certain ranges of crack geometry. In particular, the solutions were conservative for crack aspect ratios (*a/c*) less than 1.0, where *a* is the maximum depth of the crack and *c* is half of the total surface length. The conservatism could be as high as 20% for *a/c* values at approximately 0.2, but the solutions are gradually more accurate as *a/c* approaches 1, and there are no accuracy issues for *a/c* ≥ 1. This inaccuracy has limited consequence for many life predictions Because the stable shape of a growing fatigue crack under predominately tension loading is generally approximately 1, this inaccuracy has limited consequence for many life predictions. However, the conservatism could be more significant for cracks that are growing primarily in linearly decreasing stress fields.

These solutions were updated to provide more accurate values for crack aspect ratios less than 1. New WF reference solutions were generated using the commercial FE fracture mechanics software FEACrack. The same WF formulation was used. To avoid confusion, the updated univariant and bivariant solutions were named SC30 and SC31, respectively. These updated solutions were first implemented in DARWIN 8.1. Note that SC30 and SC31 are identical to SC17 and SC19 for *a/c* ≥ 1. The legacy SC17 and SC19 solutions were maintained in DARWIN for a few years for comparison purposes but have now been phased out. These new solutions were developed with NASGRO funding and were made available to DARWIN at no cost.

6.3 NEW SIF SOLUTION FOR A CRACK SPANNING A CHAMFERED CORNER

Previous versions of DARWIN included SIF solutions for quarter-elliptical cracks located at a corner defined by a simple angle of 90° (CC09 and CC11), and the CC09 solution allows the corner angle to vary from 90° by ±5°. However, some corners in actual component geometries have more complex shapes. For example, a corner could be chamfered. A SIF solution for a corner crack spanning a chamfered corner (designated as CC12) was first implemented in the NASGRO fracture mechanics software in NASGRO v6.0 (2009). This geometry is summarized in figure 43 along with the relevant geometry limits of the associated SIF solution.

The CC12 formulation is derived from the similar WF solution implemented for the CC09 crack case (bivariant quarter-elliptical corner crack). In reference to figure 44(b), the solution requires that: 1) the corner crack must completely span the chamfer; 2) the two chamfer depths, d_x and d_z, defined respectively along the x- and z-axes, must be equal (i.e., $d_x = d_z = d$); and 3) the crack center with the quarterly elliptical corner crack tip perimeter is the location from which chamfer depths are referred (see the origin *O*).

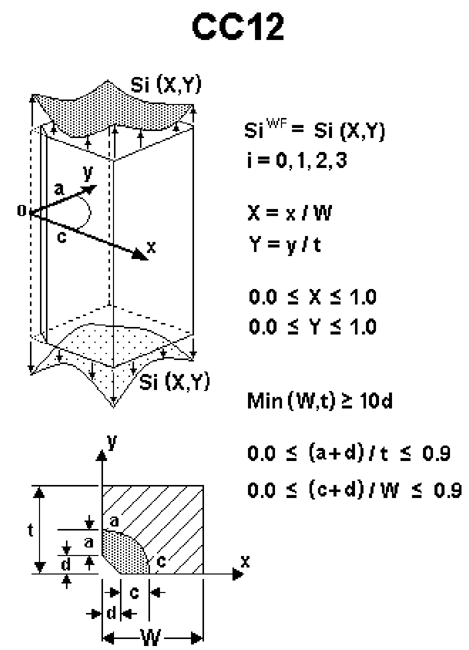
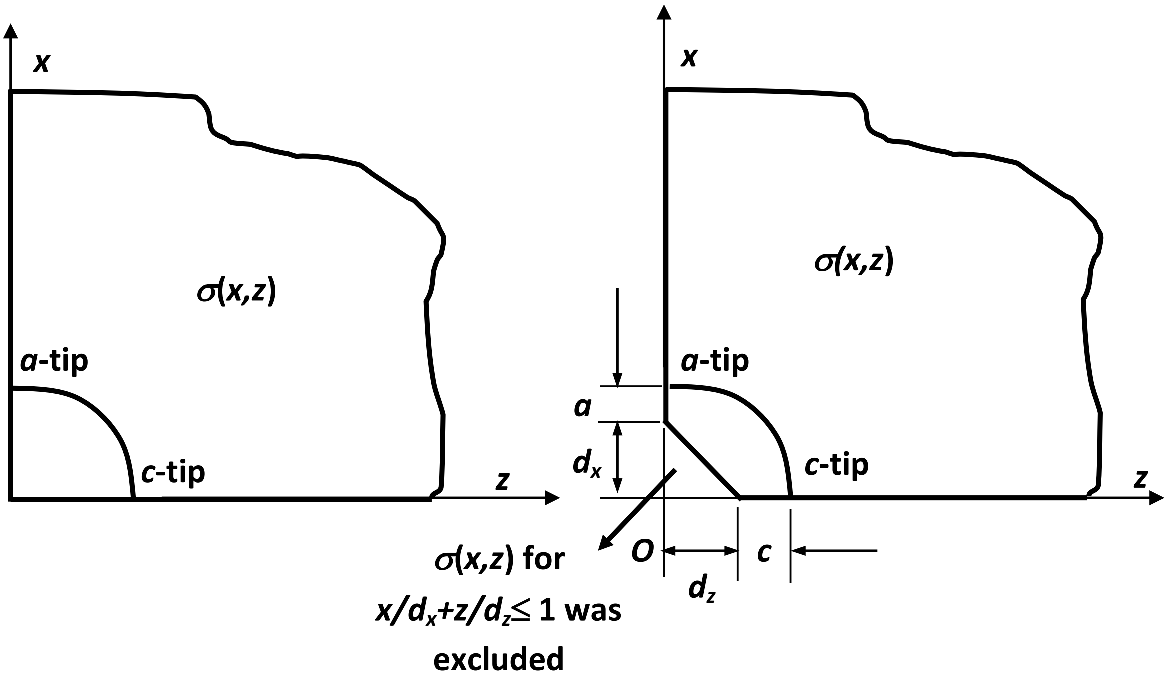


Figure 43. NASGRO CC12 solution for a crack spanning a chamfered corner



(a) (b)

Figure 44. The ratio of two different corner crack solutions was used for determining the SIFs for chamfered corner: (a) the integration for WF solution over the quarterly elliptical corner crack surface, and (b) the integration over the domain confined by the corner crack perimeter and the chamfer

In terms of the geometric dimensions designated in figure 44(b), the current solution validity range for CC12 is given by:

$$0 \leq \frac{a+d}{h_x} \leq 0.9 \tag{13}$$

$$0 \leq \frac{c+d}{h_z} \leq 0.9 \tag{14}$$

and the constraint imposed upon the chamfer depth is:

$$\frac{\min(h_x, h_z)}{d} \geq 10 \tag{15}$$

where the two geometric parameters, h_x and h_z, denote the thickness and width of the cross section and are measured from the crack center O. Note the crack depths a and c, respectively defined along the x- and z-axes, are measured from two chamfer corners.

To account for the effect of the chamfer on the force reduction in the corner crack area, a correction factor (CF) approach was applied. It serves as a multiplication factor to "correct" the corresponding CC09 solutions. This CF is based on the ratio between two quarter-infinite domain solutions, as shown in figure 44. One is the bivariant corner crack solution obtained by integrating the WF over the quarter-elliptical crack surface as shown in figure 44(a), and the second one is the bivariant corner crack solution by integrating the WF over the domain confined by the corner crack perimeter and the chamfer as shown in figure 44(b). The WF in both cases was for the corner crack in a quarterly infinite domain. Using the designation of the first corner crack solution by K_{corner}^{∞} and the second one by $K_{chamfer}^{\infty}$, the SIF for the crack at the chamfered corner is defined by:

$$K^{CC12} = K_{chamfer} = \frac{K_{chamfer}^{\infty}}{K_{corner}^{\infty}} K^{CC09} = C.F. \times K^{CC09} \tag{16}$$

in which K^{CC09} is the SIF computed using the CC09 module for finite dimensions. The CF thus depends on the bivariant stress variation in addition to the crack tip location and must be applied to both crack tips separately.

To verify this approach, two sets of SIF solutions were determined numerically using the 3-D fracture mechanics program FADD3D [28]. One is for the corner crack in the rectangular plate, and the other is for the corner crack in the chamfered rectangular plate. Two stress variations were applied for validation: one is unit tension ($\sigma = 1$), and the other is unit bending ($\sigma = 1 - x/a^{*}$). The depth parameter, a^{*}, is in the plate-thickness direction, or x-axis. Note that what this parameter refers to is slightly different in figures 44(a) and 44(b). For a 90° corner depicted in figure 44(a), it denotes the actual crack depth with a-tip, whereas for a chamfer corner in figure44(b), it represents the actual crack depth plus chamfer depth (i.e., $a + d$).

The ratios of the solutions based on these two cracked configurations correspond to $K_{chamfer}/K^{CC09}$, and they are used to compare with the above-defined CFs $K_{chamfer}^{\infty}/K_{corner}^{\infty}$. Notice the former is for a finite domain, whereas the latter is for a quarter-infinite domain. The comparison of these factors for both crack tips is displayed in figure 45(a) for unit tension and in figure 45(b) for unit bending. The excellent agreement shown in these two figures validates the approach of multiplying the CC09 solution by the CF to account for the effect of reduction in crack area due to chamfering.

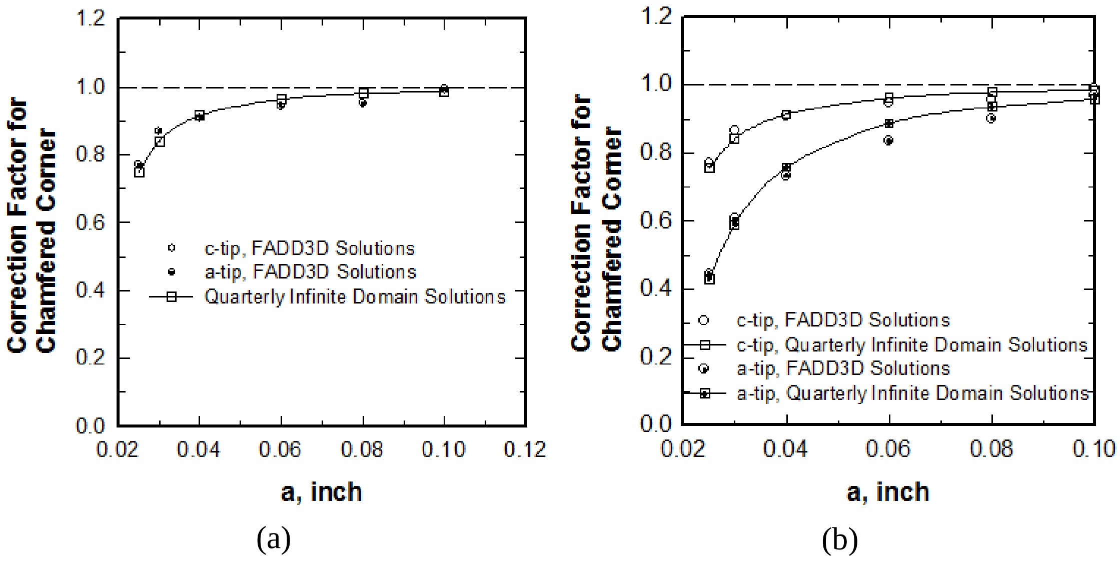
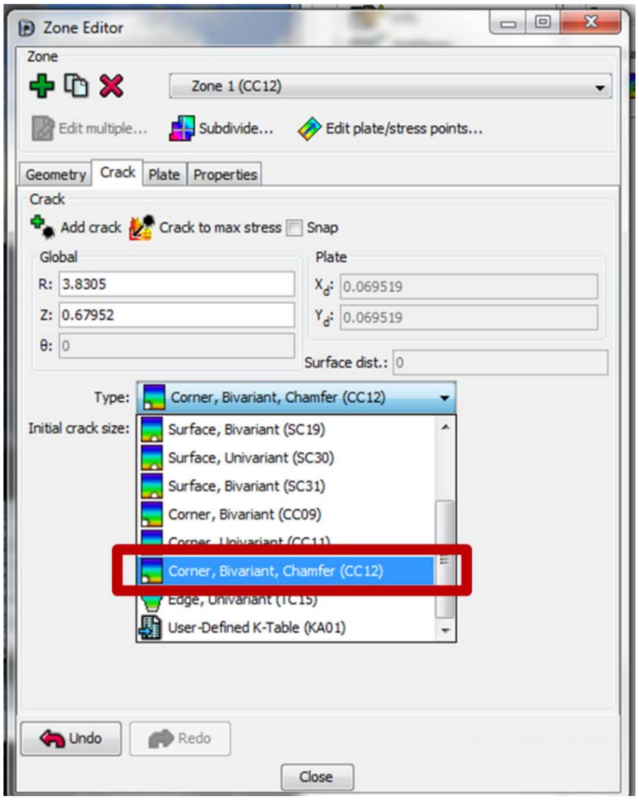


Figure 45. Comparison of FADD3D results for finite plate dimensions and WF solutions for quarter-infinite domain with chamfer corrections: (a) unit tension and (b) unit bending

The CC12 solution was imported from NASGRO into Flight_Life, and the DARWIN GUI and risk-assessment codes were modified to support its use, beginning with DARWIN 8.2. Figures 46 and 47 show the GUI procedure to define a crack at a chamfer. The user manually defines a zone in a 2-D (axisymmetric) model using the zone editor. The crack definition tab has been supplemented to include solution CC12 in the drop-down selection menu (see figure 46). For a chamfered crack, the crack location represents the quarter-ellipse center, and chamfer edges are located at a distance of *d*/2 in the primary and secondary plate directions. The GUI provides a first estimate of the chamfer depth based on the geometry during the plate generation step. The user can adjust the chamfer depth in the zone editor (see figure 47) and change plate dimensions via the zone editor or GUI to match the geometry. The current implementation allows slight deviations of the plate from 90°. The GUI checks that the user-defined plate satisfies the geometric limits for this solution. Users can define both real chamfer geometries (in which the geometry contains a chamfered corner) and virtual chamfer geometries (in which DARWIN uses the CC12 solution on a corner that is not explicitly chamfered in the FEM). The CC12 model is currently available only in the deterministic mode of DARWIN.



Zone Editor

Zone: Zone 1 (CC12)

Crack

☐ Snap

Global

R: 3.8305

Z: 0.67952

θ: 0

Plate

X_d: 0.069519

Y_d: 0.069519

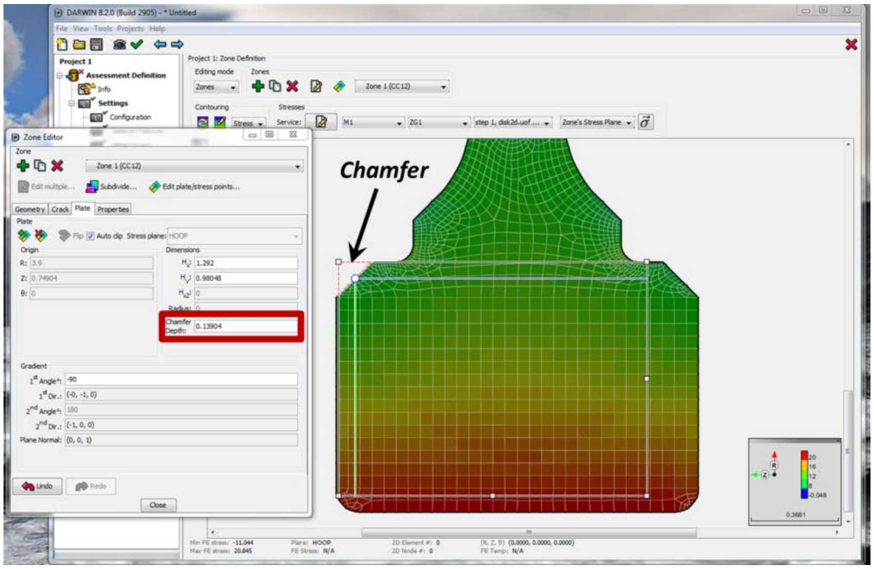
Surface dist.: 0

Type: Corner, Bivariant, Chamfer (CC12)

Initial crack size:

Edit multiple...
Subdivide...
Edit plate/stress points...
Geometry
Crack
Plate
Properties
Add crack
Crack to max stress
Surface, Bivariant (SC19)
Surface, Univariant (SC30)
Surface, Bivariant (SC31)
Corner, Bivariant (CC09)
Corner, Univariant (CC11)
Corner, Bivariant, Chamfer (CC12)
Edge, Univariant (TC15)
User-Defined K-Table (KA01)
Undo
Redo
Close

Figure 46. DARWIN zone editor showing CC12 in the dropdown selection menu



DARWIN 8.2.0 (Build 2905) - * Untitled

Project 1: Zone Definition

Editing mode: Zones

Zones: Zone 1 (CC12)

Contouring: Stress

Stresses

Service: M1 ZG1 step 1, disk2d.uof ... Zone's Stress Plane

Zone Editor

Zone: Zone 1 (CC12)

Plate

☑ Auto clip Stress plane: HOOP

Origin

R: 3.9

Z: 0.74904

θ: 0

Dimensions

H_x: 1.292

H_y: 0.98048

H_{x2}: 0

Radius: 0

Chamfer Depth: 0.13904

Gradient

1st Angle°: -90

1st Dir.: (-0, -1, 0)

2nd Angle°: 180

2nd Dir.: (-1, 0, 0)

Plane Normal: (0, 0, 1)

Min FE stress: -11.044

Max FE stress: 20.845

Plane: HOOP

FE Stress: N/A

2D Element #: 0

2D Node #: 0

(R, Z, θ) (0.0000, 0.0000, 0.0000)

FE Temp: N/A

Figure 47. GUI display of the user-defined chamfer dimensions associated with CC12

6.4 NEW SIF SOLUTION FOR A CRACK AT AN ANGLED CORNER

A new SIF solution has been developed and implemented for a crack at a corner with an angle of 135°. This new solution complements the CC12 solution for a crack spanning both corners of a chamfered feature, by treating a crack that spans only one corner of the chamfer, and transitioning to CC12 as the crack reaches the other chamfer corner. The solution also addresses isolated 135° corners (in the absence of a full chamfer).

The new crack model is designated as CC18 and is applicable to bivariant stressing acting normal to the crack plane. The "center" of a CC18 crack is always defined at the angular corner, while the "center" of a CC12 crack is at the imaginary corner of the rectangular cross section in the absence of the chamfer. The crack-tip perimeter of the CC18 crack is assumed to maintain a part-elliptical shape (see figure 48).

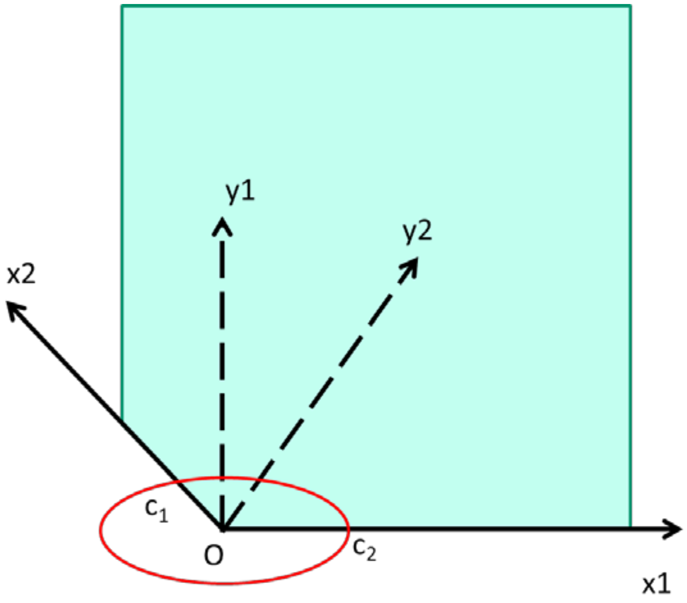


Figure 48. Schematic representation of new SIF solution for crack at angled corner

To define a unique crack tip perimeter, two coordinate systems are used. The selection depends on the relative sizes of the two surface crack lengths, c1 and c2 (see figure 49). These schematic drawings show that the crack center is located at the bottom corner of the chamfer (Point O). The two surface crack lengths are measured in reference to the crack center, and the *x*-axis lies along the larger of the two lengths, c1 and c2. In both scenarios shown in figures 49(a) and 49(b), the surface crack tip along the chamfer is denoted by c1 tip, and the surface crack tip along the plate surface by c2 tip. Figure 49(a) shows the crack length with c1 tip is shorter than the crack length with c2 tip. The designated coordinate system defines the *x*-axis to be along the direction with the c2 tip, and the *y*-axis to be perpendicular to *x*-axis, pointing into the interior of the component. A separate coordinate system is depicted in figure 49(b), where the crack length with c1 tip is longer than the crack length with c2 tip, in which case the *x*-axis is defined along the c1 tip from the crack center. The choice of coordinate system determines the description of the crack depth, *a*, which is defined along the *y*-axis. Therefore, any set of c1 and c2 surface crack lengths defines a unique crack.

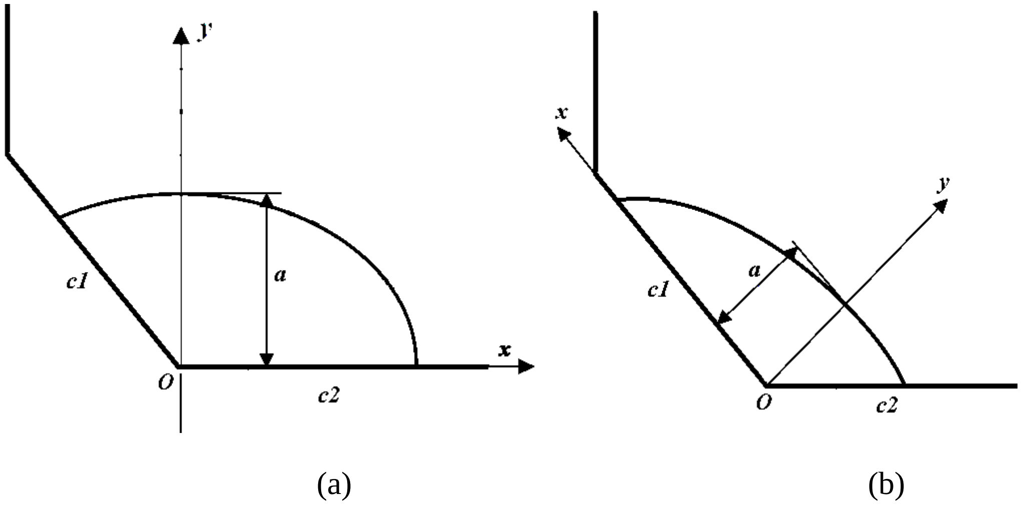


Figure 49. Designation of coordinate system for CC18 angular corner crack depending on the relative sizes of c1 and c2 crack lengths: (a) c1 ≤ c2 and (b) c1 > c2

The SIF formulation of CC18 is based on a WF approach. Its PWF is the same as the one used by SC31, which is a surface crack in a rectangular section subjected to bivariant stressing. The difference is in the SIF formulation, which accounts for the interactive effect of PWF and crack opening stress as a result of the reduction in effective crack area from the 45° chamfer. For reference, the PWF is given as follows:

$$W_{QQ'} = \frac{\sqrt{\ell^2_{Q_RQ^*} - \ell^2_{QQ^*}}}{\pi \ell^2_{QQ'}\sqrt{\pi \ell_{Q_RQ^*}}}\left(1 + \frac{\ell^2_{QQ'}}{\ell^2_{\bar{Q}_xQ'}}\right)\left\{\begin{array}{c} 1 + \Pi_1\sqrt{1 - \left(\frac{\ell_{QQ^*}}{\ell_{Q_RQ^*}}\right)^2} + \Pi_2\left(1 - \frac{y}{y'}\right)^2 \\ +\Pi_3\left[1 - \frac{x}{\mathrm{sign}(Q')x'}\right]^{1.5}\sqrt{1 - \frac{y}{a}} \end{array}\right\} \tag{17}$$

Figure 50 is provided to facilitate identifying the point locations indicated in the above PWF equation. Their respective definitions in the elliptical coordinate system (i.e., represented by elliptical radius ξ and angle η), can be found in table 2. The elliptical radius corresponding to the crack tip perimeter is denoted by ξ_0. The script *l* denotes the distance between the two points indicated in its subscript (e.g., $\ell_{QQ'}$ is the distance between two points Q and Q'). The interceptions of the axes of the local *xy*-coordinate system with the crack-tip perimeter defines two crack lengths, *a* and *c*. The chord lengths x' and y' are derived by fixing *y* and *x* values with any interior point, *Q*. The remaining unknowns in the above PWF (Π_1, Π_2, and Π_3) will be determined by the reference solutions described later.

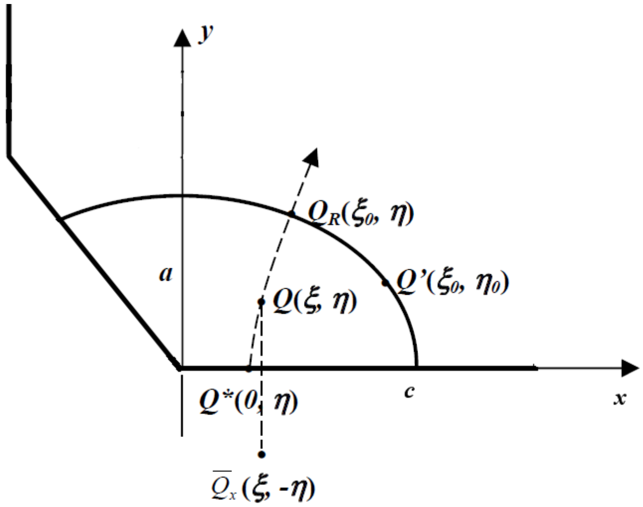


Figure 50. Geometry configuration and nomenclature for CC18 angular corner crack: point locations, crack-length designation, and coordinate system referenced in PWF

Table 2. Definition of point locations indicated in PWF

Point Designation	Location Coordinate in Elliptical System
Q	(ξ, η)
Q_R	(ξ_0, η)
Q'	(ξ_0, η_0)
Q^*	$(0, \eta)$
$\bar{Q}_x$	$(\xi, -\eta)$

The CC18 SIF solution is computed by integrating the product of the PWF and the crack opening stress over the crack area. The formulation in equation form for the SIF at the crack-tip location, Q', is given by:

$$K(\xi_0, \eta_0) = \int_0^{\xi_0} \int_0^{\eta_e} W_{QQ'}(\xi, \eta, \xi_0, \eta_0) \cdot \sigma(\xi, \eta) \cdot (\sin^2 \eta + \sinh^2 \xi) d\eta d\xi \tag{18}$$

where the crack opening stress applied at any point, $Q(x, y) = Q(\xi, \eta)$, on the crack surface is denoted by $\sigma(\xi, \eta)$. The upper limit of the internal integral η_e is a dependent of elliptical radius ξ and is given by, based on the assumption of a 45° chamfer,

$$\eta_e = \pi - \tan^{-1} \frac{1}{\tanh \xi} \tag{19}$$

The reference solutions are discrete SIF results at c1 and c2 tips of the angular corner crack subjected to three separate sets of reference stress. In reference to the local coordinate system depicted in figure 50, the three sets of reference stress are given by $\sigma_1 = 1$, $\sigma_2 = 1 - y/a$, and $\sigma_3 = 1 - x/c$. The geometric dimensions used for the determination of discrete SIF results are in terms of the ratios of c1/c2 and c1/*d,* where *d* is the chamfer length (the distance between the two corners of the chamfer). The width and the thickness of the chamfered cross section have been defined large enough to minimize the side-surface effect. Numerical models were generated using Patran in accordance with discrete sets of geometric combinations. Their model representations were in terms of surface patches in 3-D domain and text files in ABAQUS input deck format. A software converter was then used to convert the text files in ABAQUS format to input files in a format compatible with the boundary element fracture mechanics program, FADD3D, that determined the variation of SIF along the crack-tip perimeter. It should be noted that the crack length *c* denoted in figure 50 could be associated with either c1 or c2 tip pending on the coordinate system being selected resulting from the relative magnitudes of c1 and c2.

The post-transition crack model for CC18 is CC12 when the c1 tip grows near or beyond the chamfer length. The onset of the transition is when c1 ≥ 0.9*d*, and the new crack depth, *a*, with the transitioned CC12 crack is given by:

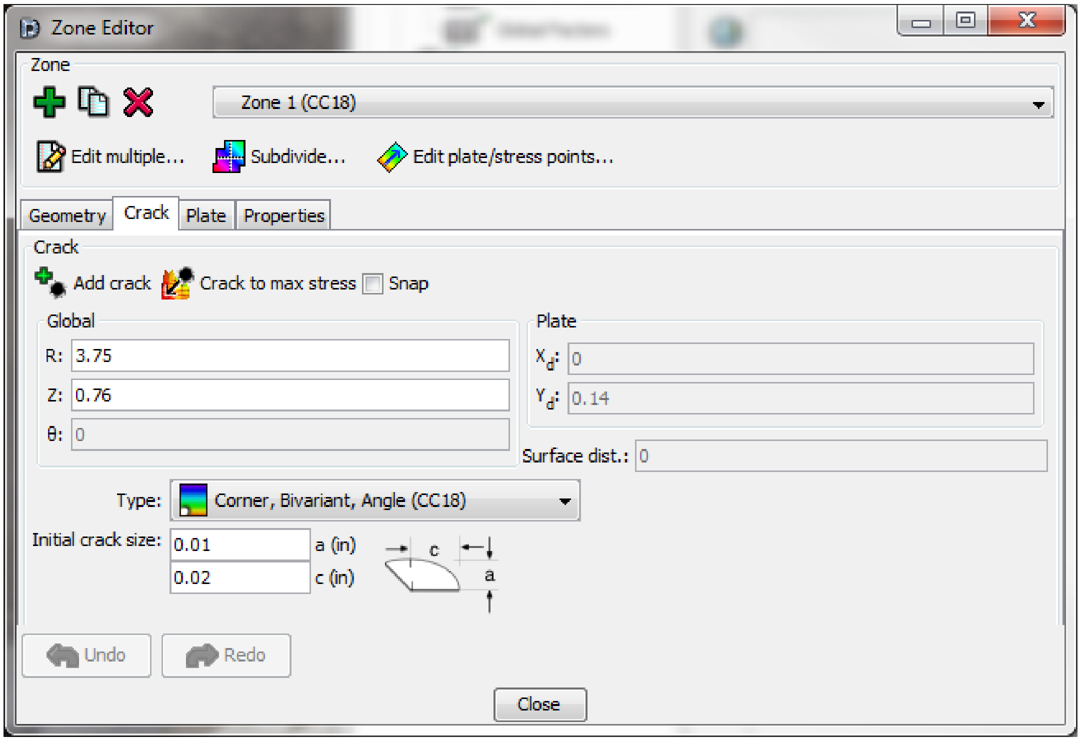
$$a = \begin{cases} \dfrac{d_S^2\left(1-\frac{\pi}{4}\right)}{d_S+c_2} & ,c_1 \leq c_2 \\ \dfrac{\frac{1}{2}d_S^2+\frac{d_S a}{\cos\frac{\pi}{4}}+\frac{1}{2}a^2-\frac{\pi}{4}d_S(c_2+d_S)}{\frac{\pi}{4}(c_2+d_S)} & ,c_1 > c_2 \end{cases} \tag{20}$$

Note that the crack depths for CC12 are measured from the two chamfer corners. The crack length c2 remains the same and is used to denote the crack length *c* with the transition crack model. The quantity d_S is the chamfer depth (the distance from the chamfer corner to the virtual corner of the plate in the absence of the chamfer).

Future plans include the development of an appropriate transition criterion and process as a surface crack (SC30 or SC31) approaches an angled corner and becomes a CC18 crack.

The DARWIN GUI has been enhanced to support CC18 for deterministic life analyses. During zone definition, users can select the new crack type (see figure 51). For non-circular crack fronts (i.e., *a* < *c*), the GUI positions an elliptical icon over the crack location. The GUI includes a button to enable users to switch the chamfer leg where the long crack length is located. This crack type also requires that the user input a chamfer dimension, as is already done for the previous chamfer solution, CC12. The chamfer dimension is indicated in the GUI screen (see figure 52), and the crack "center" is indicated by the small circle.

New SIF solution CC18 was first implemented in DARWIN 9.0.



Zone 1 (CC18)

Crack

□ Snap

Global

R: 3.75

Z: 0.76

θ: 0

Plate

X_d: 0

Y_d: 0.14

Surface dist.: 0

Type: Corner, Bivariant, Angle (CC18)

Initial crack size: 0.01 a (in)

0.02 c (in)

Figure 51. DARWIN GUI zone editor for new angled corner crack solution CC18

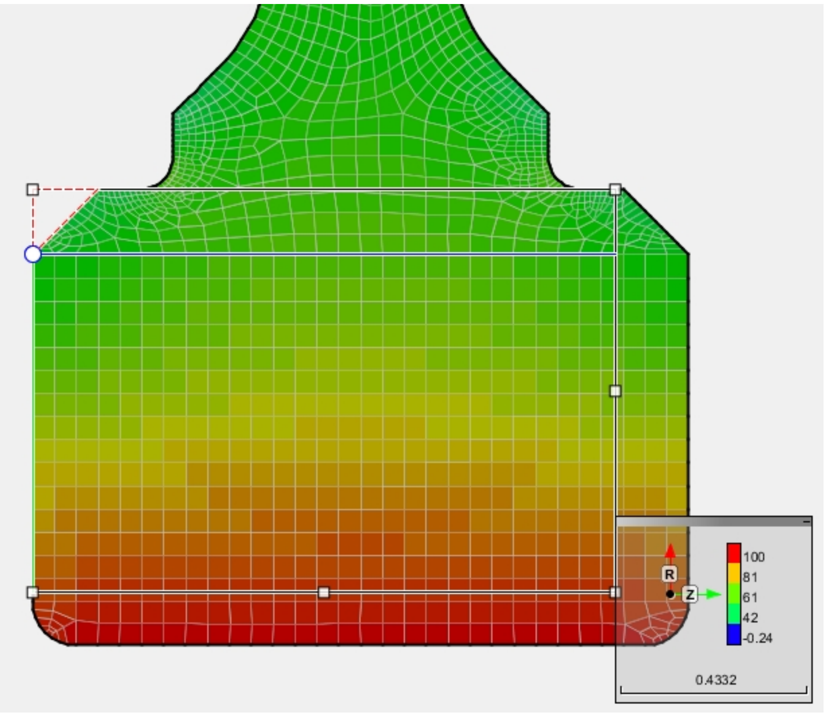


Figure 52. DARWIN GUI display of CC18 showing chamfer dimensions and crack center

7. SMALL CRACKS

Damage-tolerant approaches to induced surface-anomaly problems may involve growth-rate analysis of fatigue cracks much smaller than are typically treated by established (long crack) fracture-mechanics methods. Under some conditions, small fatigue cracks exhibit anomalously fast growth rates in comparison to large cracks at nominally the same applied SIF range. As a result, life calculations for small cracks based on large-crack fracture mechanics can be non-conservative under some conditions, particularly when initial anomaly sizes are extremely small. Special methods for small fatigue cracks should increase the accuracy of reliability calculations. In a previous grant [22], available small-crack data for several rotor materials were collected from the open literature, and a simple engineering method was identified to predict the growth rate behavior of small fatigue cracks nominally 0.001″ and larger.

No available small-crack data were identified in the previous study for the common rotor material Alloy 718. Therefore, a test program was performed at Honeywell to generate such data on a fine-grained 718 at elevated temperature (600°F). Some preliminary work to develop an appropriate specimen preparation method and test protocol was also performed in the previous grant, but data were not successfully generated at that time for cracks sufficiently small to exhibit a distinct small-crack effect (or to determine if they would exhibit such an effect). Therefore, additional small-crack testing was planned and performed at Honeywell in the current grant. A detailed description of the test program and its results is provided in appendix G. For convenience, a short summary is provided here.

The test program was performed on a Delta-processed Inconel 718 alloy (DP718) with typical ASTM grain sizes 10-12, yield strength of 169 ksi, and an ultimate tensile strength of 205 ksi. All specimens were excised from a rotor forging. Small-crack data were generated with surface cracks in rectangular, tension-loaded coupons under $R = 0$ cycling. The small cracks were initiated at tiny artificial notches created by focused ion beam (FIB) machining. Multiple FIB notches were placed on each specimen to generate additional data. FIB notches with an elliptical shape and a total surface length ranging from 0.001″–0.003″ were used. The smallest notches were not successful in initiating fatigue cracks, such that specimen failure occurred from fatigue cracks that formed elsewhere on the coupon. Three successful tests were completed, two of these at $R = 0$ with a maximum stress of 130 ksi, and one at $R = -1$ with a maximum stress of 90 ksi. Crack growth data at $R = 0$ were successfully collected on 12 different cracks. Total surface crack lengths were measured with optical microscopy during repeated test interruptions. The evolution of the crack shape was determined by performing heat tinting during the test and then performing post-test fractography, and this shape information was used to estimate the crack depth from the surface length measurements as the crack grew. Some additional FCG tests were performed with compact tension specimens to determine large-crack growth rates and the large-crack threshold for comparison. Different test methods were explored to obtain near-threshold data, including ASTM load shedding and compression precracking.

SwRI performed additional analysis of the Honeywell test data to investigate whether small-crack effects were observed at the smallest crack sizes. ASTM incremental polynomial methods (second-order, five-point moving window) were used to calculate *da/dN* and *dc/dN* to perform some limited data smoothing. DARWIN crack cases were used to calculate the SIF.

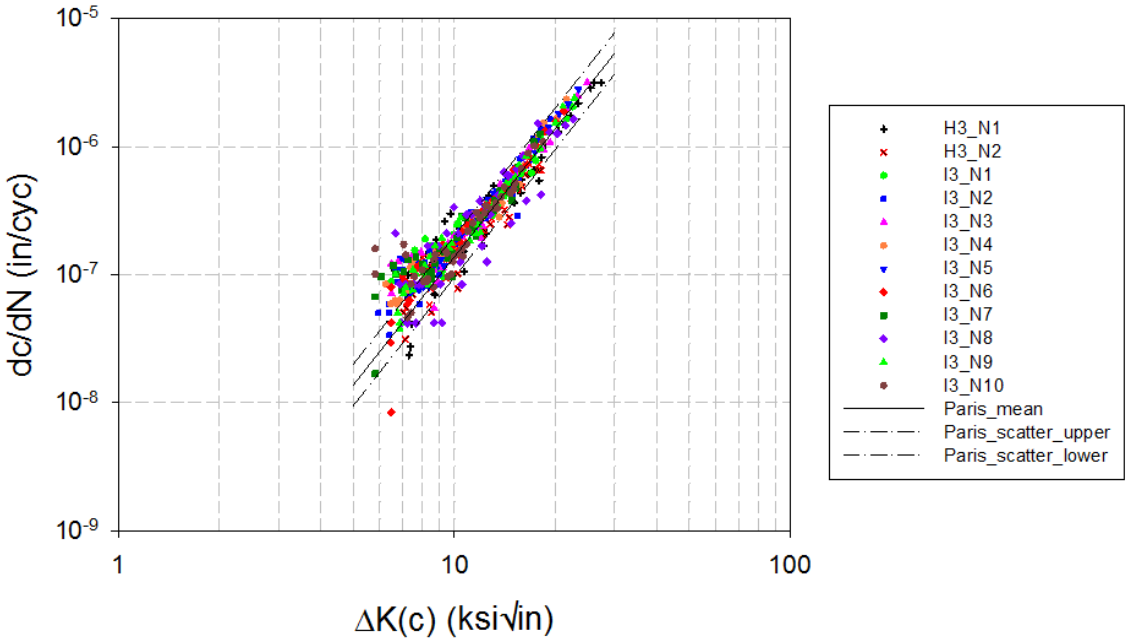


Figure 53. Growth rate data for small fatigue cracks in DP718, plotted as a function of the nominal Δ*K* values

SwRI analysis of the 12 cracks grown at $R = 0$, $S_{max} = 130$ ksi, is summarized in figure 53 (only the data at the surface tips are shown here). The Paris line shown in figure 53 was based on a regression of the data with $\Delta K_c > 12$ ksi√in (total surface length > 0.008″). The scatter bands in this figure are provided only as an aid to visualization and are purely qualitative. The data indicate a small bias towards faster growth rates at the smallest crack sizes.

A previous study of small fatigue cracks conducted in an earlier grant [22, 29] had identified a simple engineering model to predict the growth rates of small cracks. The model, originally proposed by El Haddad [30], suggests that the actual "equivalent" driving force for crack growth of a small crack can be calculated by replacing the physical crack size, *a*, with the sum ($a + a_0$) according to the relationship:

$$\Delta K_{eq} = F(a)\Delta S\sqrt{\pi(a + a_0)} \tag{21}$$

The length parameter a_0 can be calculated by the equation:

$$a_0 = \frac{1}{\pi}\left(\frac{\Delta K_{th}}{F\Delta\sigma_e}\right)^2 \tag{22}$$

when the long-crack threshold, ΔK_{th}, and the endurance limit, $\Delta\sigma_e$, are available. The previous study found that the El Haddad model was relatively successful in correlating small-crack data from the literature for several rotor materials, including titanium alloys and nickel-based superalloys.

Returning to the present study, values for ΔK_{th} and $\Delta\sigma_e$ were not available for DP718, so it was not possible to perform a prediction using the El Haddad model. Instead, different empirical values of a_0 were evaluated to determine if this general form could be used to correlate small-crack data with the Paris line trends of longer cracks. The calibrated value a_0 = 0.0006″ appeared to be successful in accomplishing this goal (see figure 54).

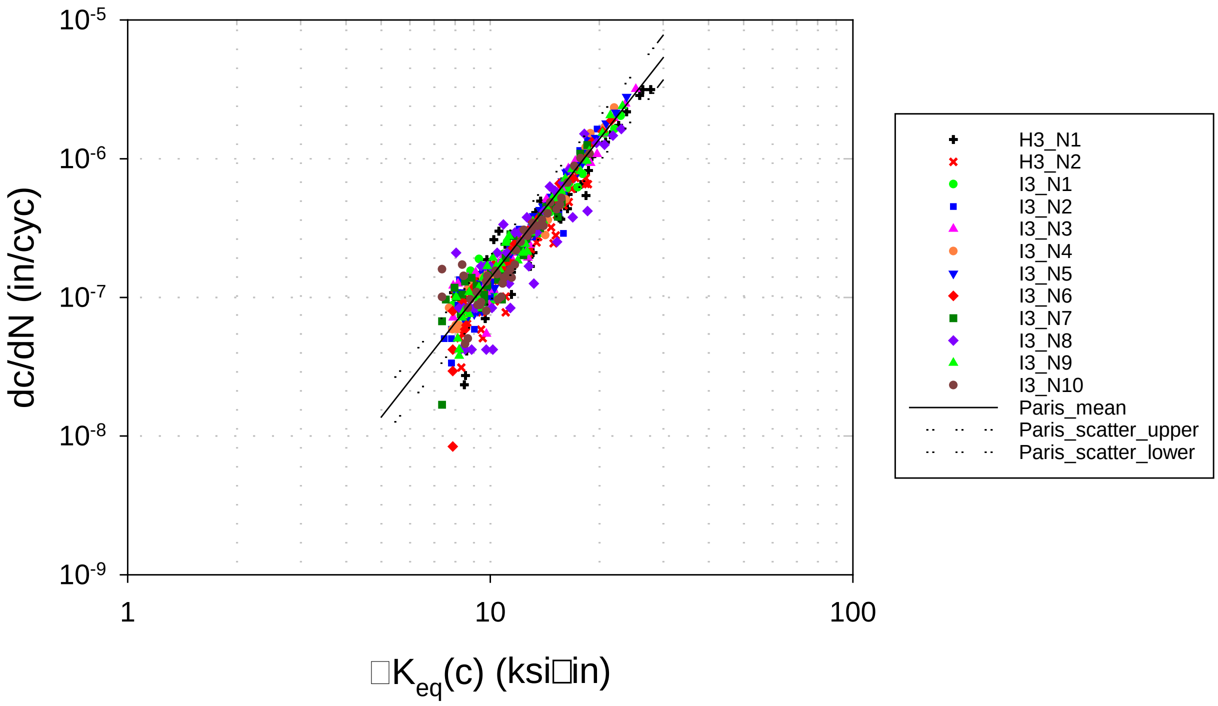


Figure 54. Growth rate data for small fatigue cracks in DP718, plotted as a function of the adjusted ΔK_{eq} values

This calibrated value of a_0 is slightly smaller than values of a_0 (0.001″–0.002″) that had been successful in correlating small-crack behavior for other nickel-based superalloys [2]. Those El Haddad parameters were usually on the order of five times the average grain diameter. The typical grain size of the DP718 was stated as ASTM 10-12, which corresponds (inversely) to mean intercept grain sizes of 0.0002″–0.0004″. Five times this value would be somewhat larger than 0.0006″. Therefore, the significance of the agreement in figure 54 is not clear.

In any case, the data show that the small-crack effect in this superalloy is relatively small, consistent with the extremely fine grain size. An empirical El Haddad relationship may be a useful engineering method to predict small-crack growth rates for this material.

8. DARWIN TESTING AND EVALUATION

This task was focused on the formal testing and evaluation of the DARWIN software. New DARWIN versions were evaluated by both the primary developer (SwRI) and OEMs represented on the Steering Committee. DARWIN was first tested internally by SwRI using a variety of methods. The developers of individual new features performed focused testing on those features, and automated algorithms were used to test the overall stability of the software. The software was then provided to the engine companies for review via incremental alpha and beta software releases that occurred throughout the program. These companies evaluated the functionality and stability of DARWIN while also comparing DARWIN results against their own company codes, fleet experience, and test data. Once these evaluations had been completed, and all issues and bugs had been satisfactorily resolved, the software was released for use by licensed DARWIN users.

8.1 DARWIN CODE RELEASES

DARWIN development was performed throughout the project with major software releases occurring approximately once a year. For each major new version, two or more incremental (alpha) versions were submitted to the Steering Committee for review periods lasting at least 3 months. Each of these incremental releases contained only a small number of new features, facilitating a focused review. The official "final" alpha release contained all new features. During these review periods, the Steering Committee provided feedback regarding the accuracy and usability of the software, and this feedback was addressed by SwRI. After the alpha review period was completed and reported issues were resolved, a beta version was released to RISC for review. During the beta review period, RISC and the Steering Committee provided additional feedback, which was also addressed by SwRI. Following these reviews, a final production version was released to licensed users.

A summary of the features associated with the major software releases is provided in this section. Note that some of the new features in each version were funded by sources other than the FAA, such as Small Business Innovation Research (SBIR) projects funded by other U.S. government agencies. Some further information about these other projects is provided in section 9.6 of this report. For completeness, all features (both FAA funded and non-FAA funded) are included in the lists in sections 8.1.1–8.1.5. For clarity, the non-FAA-funded features are denoted in *italics*.

8.1.1 DARWIN 7.2

An initial DARWIN 7.2 alpha version was completed under a previous FAA grant. Under the current grant, this version was extensively verified and enhanced to address bugs identified by SwRI and the Steering Committee. The beta version was first released to RISC for review on February 22, 2012, and a revised version was submitted for review on November 19, 2012. The production version of DARWIN 7.2 was released to licensed users on March 13, 2013. DARWIN 7.2 included the following new features:

- Automatic zone generation for risk assessment of inherent anomalies in 2-D models
- Time-dependent fatigue crack growth assessment
- Parallel processing
- Local RS profiles for 2-D models
- FE2NEU support for ABAQUS models with non-Cartesian coordinate systems
- Mission definition enhancements

8.1.2 DARWIN 8.0

The DARWIN 8.0 alpha final version was released to the Steering Committee for review on August 31, 2012. Following an extensive review period, the beta version was released to RISC for review on March 13, 2013. The production version was later released to licensed users on January 14, 2014. DARWIN 8.0 included the following new features:

- Optimal autozoning for risk assessment of inherent anomalies in 2-D models
- Crack-growth retardation capability
- New SIF solution for bivariant surface crack at off-center hole (SC29)
- Local RS profiles in 2-D models for additional univariant SIF solutions
- Additional temperature option for TMF crack growth
- Support for non-hoop stress planes
- Autoplate enhancements
- HDF5 file format for DARWIN input and output
- *Interface for user-provided crack-propagation module*
- *File-based license manager*

8.1.3 DARWIN 8.1

The DARWIN 8.1 alpha final version was released to the Steering Committee for review on August 20, 2013. After an extensive review period, the beta version was submitted to RISC for review on November 6, 2014. The production version of DARWIN 8.1 was released to licensed users on March 11, 2015. This version included the following capabilities:

- Autozoning with inspection for 2-D models
- Local RS profiles for 3-D models
- Surface and line zoning for 3-D models
- Updated SIF solutions for surface crack in plate (SC30 and SC31)
- Component replacement for fleet assessments
- *Random RS from 2-D FE results*
- *2-D FE input of forging strains to determine 3-D anomaly orientation*
- *New anomaly appearance due to hot corrosion*

8.1.4 DARWIN 8.2

The DARWIN 8.2 alpha final version was released to the Steering Committee for review on September 30, 2014. The beta version was submitted to RISC for review on March 30, 2015. The production version of DARWIN 8.2 was released to licensed users on November 17, 2015. This version included the following capabilities:

- Risk assessment of inherent anomalies in 3-D models
- Introduction of new anomalies at shop visits
- New bivariant SIF solution for crack spanning a chamfered corner (CC12)
- Fleet-assessment enhancements to allow corrective actions involving disk replacement
- Improved *da/dN* temperature interpolation method
- 64-bit DARWIN GUI
- Restructuring of GUI settings menu
- Integration of Visualization ToolKit libraries for 3-D graphic rendering (*mixed funding*)
- *Enhanced mission-mixing capabilities*

- *GP importance sampling for risk calculations*
- *Material-properties scaling module for user-defined equations*
- *Random average grain size and random anomaly orientation*
- *Interface with FRacture ANalysis Code 3D (FRANC3D) software for tabular SIF solutions*
- *Allow file input of high-cycle fatigue stresses from 3-D FEMs*
- *Allow file input of RS from 3-D FEMs*
- *Optional direct link to DARWIN material-properties files*

8.1.5 DARWIN 9.0

The DARWIN 9.0 alpha final was released to the Steering Committee for review on August 27, 2015. The beta version was submitted to RISC for review on December 18, 2015. The production version of DARWIN 9.0 was released to licensed users on May 13, 2016. This version included the following capabilities:

- Optimal GP pre-zoning for inherent anomalies in 2-D and 3-D models
- 3-D sector models
- New bivariant SIF solution for a crack at an angled corner (CC18)
- Enhancements for large FEMs
- *Crack-formation life scaling*
- *Anomaly distribution tracking*
- *Micromechanical crack initiation based on input from microstructural models*

8.2 DARWIN INTERNAL VERIFICATION TESTING

Internal verification of DARWIN consists of the creation and execution of verification test problems to identify bugs and subsequent software implementation to resolve them. Several internal procedures are used to reduce the effort associated with this process and to identify bugs as quickly as possible after code is developed. Examples of these procedures include modular code development in which verification is performed on individual modules prior to integrating them into the main source code, incremental code releases in which the software is periodically provided to the Steering Committee for review prior to a major release, and automated verification in which verification testing is automatically performed whenever a code developer uploads completed source code into the DARWIN source code repository. These practices substantially reduced the number of bugs in the major code releases and reduced the human time associated with the code maintenance effort.

8.2.1 Modular Code Development and Verification

The DARWIN verification procedure provides verification at the module level. Module-based verification focuses the verification effort on new code, in which new or modified modules are verified prior to integration with the rest of the source code. This approach helps to maintain the overall stability of DARWIN.

8.2.2 Incremental Code Releases

Major DARWIN releases typically include a number of features requiring several months of Steering Committee review and bug fixes. To shorten the review period, features are submitted to the Steering Committee for review in smaller incremental releases. This allows the Steering Committee and SwRI to focus on a smaller number of features at a time and enables a more thorough examination of each individual feature. This also improves the quality of the code development process, because the code developers receive more immediate feedback on recently developed features.

8.2.3 Automated Verification Procedure

The DARWIN automated software-verification procedure provides continuous verification of the DARWIN Fortran code throughout the software development process. The automated procedure is shown in figure 55. The verification procedure begins when a new bug is discovered during automated testing (dynamic analysis, regression, internal, or generative) or is reported by an OEM. When a revision is submitted, an internal algorithm automatically retrieves, compiles, and performs regression tests of the revised code. Selected output values are compared to reference results during the Verify Build Stability step. If results are not within established tolerance values, the source of the difference is identified, and either the reference results are updated or a new bug is logged. If results are within tolerances, then the software is automatically prepared for release on the DARWIN website. For bugs that are reported by OEMs or users, the automated procedure notifies the reporter when a new build is available. The automated verification procedure reduces the human time required for verification, helping to identify bugs as they are introduced during the software development period. This process includes testing on Windows, Linux®, and Mac® OS X® operating systems.

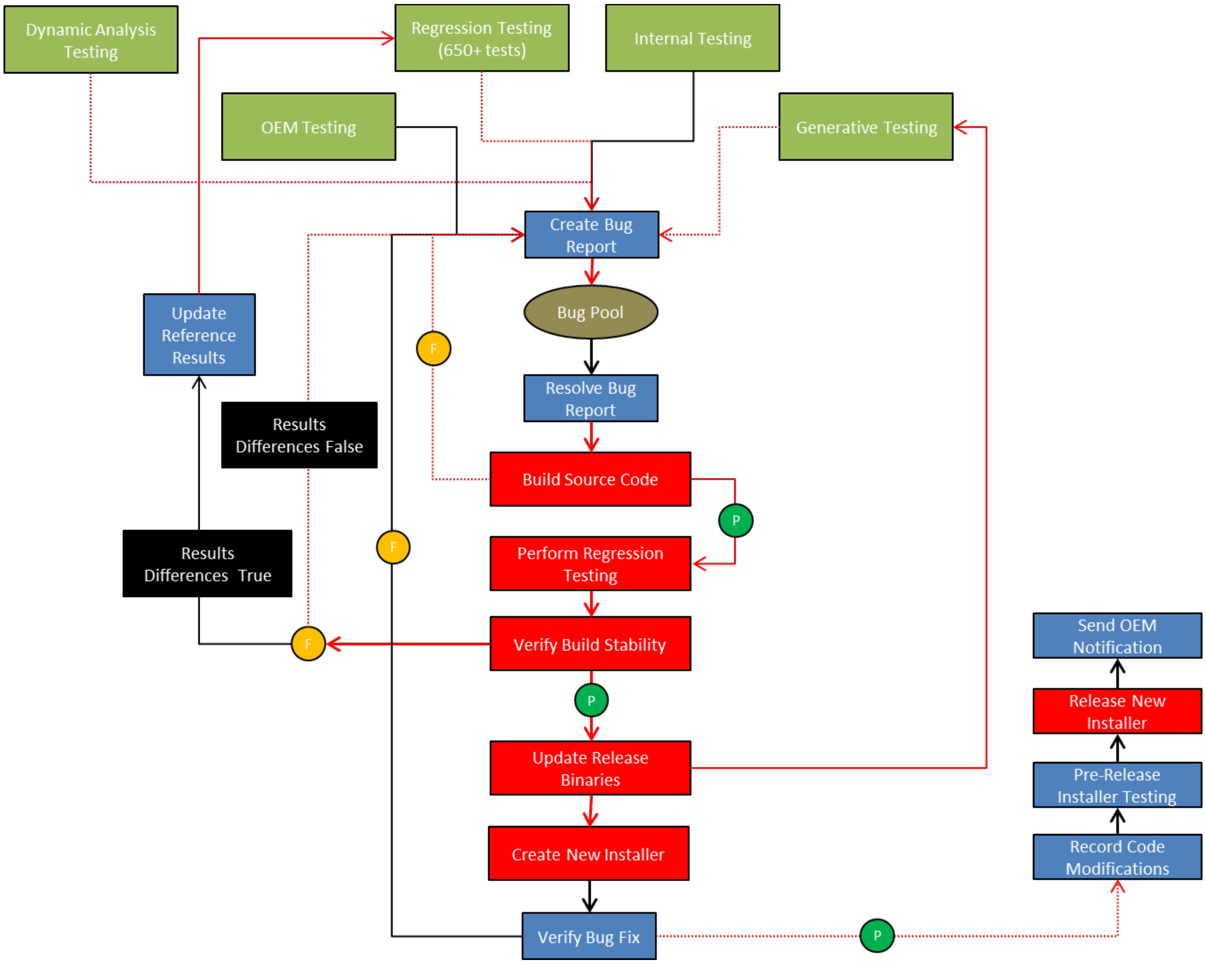


Figure 55. Automated verification procedure flowchart

8.3 OEM EVALUATION

DARWIN was evaluated by each of the four PIRATE steering committee companies throughout the project. Each company participated in substantial code review efforts to verify the numerical accuracy of the DARWIN software. The OEMS provided continuous guidance regarding the content and design of the user interface and provided evaluations of the user interface design associated with each code release.

8.3.1 Numerical Verification

Numerical verification consisted of a series of studies performed by the OEMs focused on comparison of DARWIN predictions to experimental data, field experience, and OEM in-house code results. These studies included an extensive set of test cases that were exercised by multiple users within each of the four OEM companies. The studies were performed using DARWIN versions 7.2 through 9.0. OEM evaluation of DARWIN alpha and beta versions identified a number of bugs that were fixed prior to production release of the code, and also helped to improve user confidence in the quantitative results.

8.3.2 Design Guidance

The Steering Committee was instrumental in the development of the user interface. Prior to the development of each feature, the Steering Committee was consulted regarding both the content of new features and the placement in the user interface. A number of major changes were made to the user interface based on Steering Committee guidance. This upfront guidance resulted in fewer changes to the interface during the evaluation period. User feedback regarding major interface changes has been positive.

8.3.3 Design Evaluation

The PIRATE Steering Committee performed a series of evaluations to assess the practical use of DARWIN in a design environment. Over the course of the PIRATE program, the OEMs provided general feedback regarding their experiences with the DARWIN code. The response was very positive, with additional OEMs electing to license DARWIN for certification of new engine rotors and disks. The OEMs also mentioned that the DARWIN user manual was well-documented and provided substantial assistance to both new and experienced users of the code.

9. TECHNOLOGY TRANSFER

9.1 PROGRESS REPORTS AND REVIEW MEETINGS

Regular communication with the FAA, the project steering committee, and the aircraft engine industry facilitated effective oversight of the research project and timely transfer of important new results. Written reports included brief monthly progress reports and substantially longer interim progress reports, which were issued twice per year to coincide with program review meetings or with major project reports to RISC. One-day program review meetings with the FAA were typically held once per year, with specific schedules directed by the FAA. The project steering committee conducted a 1-day or 2-day working meeting in conjunction with each program review meeting. A short status report on program activities was presented at most meetings of RISC (generally held three times per year). At least once per year, the SwRI program manager and one of the SwRI co-principal investigators attended the RISC meeting in person and made the project presentation. This facilitated additional interaction with the FAA personnel present at the RISC meeting. The Steering Committee often held a working meeting in conjunction with this particular RISC meeting. At other RISC meetings, the SwRI program manager typically made the project presentation via a telephone link. In addition, the project steering committee conducted periodic telephone meetings, typically one or two times per month.

9.2 CONFERENCE PRESENTATIONS AND JOURNAL ARTICLES

Significant results from this research program were disseminated to a much broader audience in the gas turbine engine community or the international technical community in disciplines such as fracture mechanics or reliability through presentations at conferences and symposia or publications in archival technical journals. A number of additional publications and presentations during the term of this grant were based on DARWIN-related research funded primarily by non-FAA sources. However, all of this technology transfer was based on the solid foundation of FAA funding to develop advanced technology and software for probabilistic integrity and risk assessment of turbine-engine components, whether through this grant or through its predecessor grants. Twenty-three published papers (most of them presented at international conferences) and 19 additional presentations without manuscript occurred during the term of the current grant. A complete listing of these presentations and publications is provided in appendix H.

9.3 DARWIN COMMERCIAL LICENSING

At the request of industry and the FAA, an infrastructure was developed previously to support the formal use of DARWIN by engine companies for official FAA and company purposes. The DARWIN software was first offered for commercial licensing in 2000. The licensing activity has grown slowly and steadily since that date. DARWIN license fees collected by SwRI are used exclusively to enhance the DARWIN code and to provide user support for the benefit of engine manufacturers, other licensees, and the FAA. No FAA funds from the current project were used to support the licensing activity.

As of this writing, 18 commercial DARWIN licenses are active, including 14 manufacturers of gas turbine engines for aircraft applications, two manufacturers of gas turbine engines for land-based power generation, and two consulting firms. Discussions are continuing with other prospective licensees.

A number of significant enhancements to DARWIN were funded entirely or substantially with licensing revenue. These enhancements include a new user interface for the user-provided crack propagation module, a file-based license manager, partial funding of micromechanics-based crack formation and growth algorithms, and a host of minor GUI enhancements.

9.4 DARWIN TRAINING

A comprehensive DARWIN training workshop was conducted at SwRI on April 16–18, 2013. A total of 42 people from 27 different organizations in the U.S. and six foreign countries attended in person (including 13 people from the four steering committee companies, eight people from other RISC member companies, one FAA staff member, and one FAA consultant). An additional 59 people from 12 organizations participated remotely in the first day of the workshop through a Web meeting, including more than ten people each from the United States Air Force (USAF), NASA Centers, and the National Research Council of Canada. Other participants (either onsite or remote) included Naval Air Systems Command (NAVAIR), other aircraft engine manufacturers, industrial turbine engine manufacturers, a major aircraft structures manufacturer, and various technology providers. The presentation charts from the workshop were distributed to the FAA and to participants who requested them. The workshop was based on DARWIN 8.0.

The costs of preparing the training curriculum and presenting the April 2013 workshop were supported by the FAA grant. The curriculum was then adapted to present 2-day training workshops to several DARWIN commercial licensees at their company sites. However, all of the costs of preparing and presenting the training to licensees, including all travel costs, were paid by the licensing companies. During the past 3 years, SwRI staff presented the training workshop six different times for licensee companies in Russia, Japan, and France. Approximately 80 people participated in the licensee training.

One additional 2-day DARWIN training workshop was presented in Dayton, Ohio, in December 2014. Participants included the Air Force Research Laboratory (AFRL), the Air Force Life Cycle Management Center (AFLCMC), and AFRL contractors. Twenty-four people participated in this training. All costs for this training and the related travel were covered by AFRL.

9.5 TECHNOLOGY TRANSFER TO OTHER U.S. GOVERNMENT AGENCIES

DARWIN is available royalty-free to all U.S. government agencies. To date, DARWIN has been distributed to several branches of the Department of Defense including the Army, Air Force, and Navy. Under the current grant, DARWIN has been distributed for use at AFRL (Dayton, Ohio), and the AFLCMC (Dayton, Ohio). AFRL is using DARWIN to support their research into fatigue behavior of gas turbine engine materials. The AFLCMC is investigating the use of DARWIN to supplement remaining life predictions and fleet assessments of aircraft gas turbine engines in the USAF.

9.6 DARWIN SPIN-OFF PROJECTS

One of the clearest indications of technology transfer to other government agencies and commercial licensees is the growing number of independent research projects funded by these other organizations for additional development and implementation of new DARWIN technology that is specifically relevant to their own applications. Though most of these independent projects have been modest in size—usually 1 to 2 years in duration—they have each contributed to the growing capabilities of DARWIN. In most cases, the technology resulting from the project has been implemented in the commercial production version of DARWIN and/or published and, therefore, is available to all DARWIN users.

Several additional projects were conducted for individual commercial licensees. Some of the new features developed under those projects were made available to other DARWIN licensees (with the advance approval of the funding licensee). However, other new features developed under these projects were made available only to the funding licensee as password-protected hidden features. These projects were proprietary in nature and are therefore not mentioned here.

For convenience, sections 9.61–9.65 provide thumbnail sketches of many of the government-funded spinoff projects.

9.6.1 Integrated Processing and Probabilistic Lifing Models for Superalloy Turbine Disks

Integrated Computational Materials Engineering aims to merge the computational simulation of materials, manufacturing processes, and component design into a holistic system to optimize components before they are fabricated. The manufacturing process software tool Design Environment for FORMing (DEFORM™), developed by Scientific Forming Technologies Corporation (SFTC), can simulate rotor manufacturing processes including forging, heat treating, and machining. Outputs from DEFORM can then be used as inputs to probabilistic damage-tolerance analysis software such as DARWIN.

AFRL had previously funded an exploratory 9-month Phase I SBIR project conducted jointly by SwRI and SFTC. In this project, an initial link was established to transfer DEFORM calculations of location-specific bulk RS that emerge in the rotor during manufacturing into DARWIN, and then use this information in DARWIN to inform FCG life computations. The effect of the deterministic manufacturing RS on life and risk results was demonstrated for an idealized rotor geometry [31].

Based on the success of the Phase I feasibility study, AFRL awarded SFTC and SwRI a larger 2-year Phase II project and a subsequent 2-year Phase IIE project. The project aimed to extend the DEFORM-DARWIN link to address both microstructure (location-specific average grain size) and anomaly characterization (location-specific anomaly orientation) and also to extend the link from deterministic descriptions to parametric and probabilistic descriptions of the relevant variables. DARWIN was enhanced to provide a GP modeling capability that estimates the relationship among DEFORM model responses and material processing parameters. This enhancement-enabled probabilistic treatment of manufacturing process variables required only a few hundred DEFORM analyses rather than millions of runs that would be required for a brute-force Monte Carlo simulation approach. The GP modeling capability was first applied to probabilistic modeling of random bulk RS [32]. It was then applied to enable probabilistic modeling of residual strain, average grain size, and as-large-as (ALA) unrecrystallized grain size [33]. The forging strain information is used in the prediction of 3-D anomaly orientation. The average grain sizes are used in a newly created material properties scaling module to inform microstructure-based fatigue crack growth models implemented in DARWIN under this effort [33]. Average and ALA grain sizes are used in the DARWIN formation module to inform microstructure-based fatigue crack formation life scaling models (also implemented in DARWIN under this effort [33]). The GP modeling capability was extended for application to probabilistic risk assessments via a new DARWIN capability entitled GP Importance Sampling.

DEFORM includes a capability to track the location of anomaly distributions. Under this effort, DARWIN was enhanced to transfer the resultant anomaly distributions from DEFORM to DARWIN for use in risk assessments.

The random RS and random forging strain capabilities were first introduced in DARWIN 8.1. Random average grain size, material properties scaling module, random 3-D anomaly orientation, and GP importance sampling for risk calculations capabilities were provided in the DARWIN 8.2 release. Micromechanical crack initiation models, crack formation life scaling, and anomaly distribution tracking were introduced in DARWIN 9.0.

OEM review of these new capabilities is currently underway, and future enhancements are planned under separate funding.

9.6.2 3-D Crack Growth Life Prediction for Probabilistic Risk Analysis of Turbine Engine Metallic Components

Most current tools and methodologies (such as DARWIN) that predict the life and reliability of fracture-critical gas turbine engine components rely on SIF solutions that assume highly idealized component and crack geometries. This can lead to highly conservative results in some cases. AFRL had previously funded an exploratory 9-month SBIR project jointly conducted by SwRI and Fracture Analysis Consultants to combine one software tool for creating high-fidelity crack growth simulations FRANC3D with another software tool for performing probabilistic FCG life assessments of turbine engine components (DARWIN). DARWIN uses FEMs of component stresses, whereas FRANC3D performs automatic adaptive re-meshing of these models to simulate crack growth. The main technical objective was to demonstrate that uncertainty and conservatism in probabilistic life predictions could be reduced by using high-fidelity models for 3-D crack growth.

Based on the success of the Phase I feasibility study, a larger two-year SBIR Phase II project was awarded by AFRL to FAC and SwRI for further development and evaluation of the integrated tool. Modifications were performed to both codes to allow them to share and exchange data and to enhance their shared computational capabilities. Most notably, a new methodology was developed to predict the shape evolution and the fatigue lifetime for cracks that are geometrically complex and not easily parameterized by a small number of degrees of freedom. Special features of the new FRANC3D-DARWIN interface include the ability to consider RS or high-cycle fatigue stresses calculated using the FEM. The integrated software system is described elsewhere [34]. The FRANC3D interface was first released to users in DARWIN 8.2.

9.6.3 Life and Reliability Prediction for Turbopropulsion Systems

The objective of this 6-month NAVAIR-funded SBIR Phase I project, performed by Mustard Seed Software (which subsequently merged with Elder Research, Inc.) and SwRI, was to develop an innovative methodology for life and reliability prediction of hot-section components in advanced turbopropulsion systems. The project focused on addressing the effects of competing time-dependent damage modes including creep, stress corrosion, and stress rupture on long-term performance and reliability of engine disks made from nickel-based superalloys that could exhibit location-specific microstructures, microstructural variability, or evolution of microstructures after thermal exposures at elevated temperatures for extended time periods. Building on the time-dependent crack growth framework developed for the current FAA project, a set of three generic time-dependent crack growth models was implemented and integrated into DARWIN, along with GUI enhancements to facilitate entry of material property data and time-dependent mission information. A framework was developed for extending the Phase I methodology to treat crack growth variability due to microstructural variation using the Micromechanical Fatigue Variability (MicroFaVa) probabilistic micromechanics software [35] with a plan for future work and transition to industry.

The coupled cycle- and time-dependent crack-growth methodology framework was further extended in Phase II to treat specific environmental-sensitive damage mechanisms. Major outcomes in the Phase II Program included: 1) development of a micromechanical approach for predicting concurrent creep- and oxidation-induced crack growth rates for nickel-based materials [36–39]; 2) development of a framework for treating time-dependent crack formation and crack growth threshold at grain boundary oxides in nickel-based alloys [37]; 3) development of a time-dependent crack-growth methodology for predicting cold dwell fatigue in titanium-based alloys [40]; 4) implementation of some of these time-dependent fracture models into a standalone software (e.g., MicroFaVa) for predicting time-dependent and cycle-dependent crack growth properties, and the integration of MicroFaVa with DARWIN via a common external module; 5) applications of the probabilistic life-prediction code (i.e., DARWIN) to benchmark nickel-based disk problems to assess fracture risk due to microstructural variability such as grain size and gamma prime size variations [41]; and 6) evaluation of the benchmark computation results by an industrial partner.

9.6.4 Hot Corrosion of Nickel-Based Turbine Disks

Anticipated increases in turbine inlet temperatures will increase the likelihood of Type II corrosion damage, which is typically characterized by localized corrosion pitting caused by melting of sulfur-containing salts. These Type II hot corrosion pits have been shown to decrease the fatigue resistance of superalloys due to initiation of fatigue cracks at hot corrosion pits. However, the rigorous analytical models and tools needed by turbine engine designers to predict Type II corrosion effects are not currently available.

Barron Associates and SwRI were previously awarded SBIR Phase I and Phase II projects by NASA Glenn Research Center to investigate the feasibility of integrating probabilistic models for these effects into DARWIN. The outcome of this project was the demonstration that Type II hot corrosion can be treated as two damage processes involving: 1) time-dependent pit initiation and pit propagation; and 2) pit transition to a single crack that ultimately propagates to a critical length and results in fracture [42]. Empirical models for treating these damage processes were developed and implemented in DARWIN.

Elder Research, Inc. and SwRI were later awarded Small Business Technology Transfer Phase I and II projects by NASA Glenn Research Center to develop physics-based modeling tools for predicting the initiation of hot corrosion and addressing pit and fatigue crack formation in nickel-based alloys subjected to corrosive environments. A probabilistic framework for linking pit initiation and growth, pit transition to crack, and combined cycle- and time-dependent crack growth in the DARWIN platform was developed [43]. The project, currently in Phase II, is focused on: 1) the development of modeling tools for treating the evolution and coalescence of hot corrosion pits and microcracks as a function of time and temperature, as well as corrosion environments; 2) the development of visualization tools for monitoring the progression of hot corrosion damage in a FEM of a gas turbine disk; and 3) the implementation of these modeling tools and visualization tools in DARWIN. This technology will significantly improve the current ability to simulate and avoid corrosion fatigue failure of engine disks or metallic structural components due to prolonged exposure to extreme environments at elevated temperatures.

9.6.5 Rapid Qualification of Additive Manufacturing Processes with Fracture Mechanics and NDE Modeling

Additive manufacturing (AM) offers the prospect of substantial reductions in the time and cost required to design and produce new parts under certain conditions. However, as with any manufacturing process, AM can create defects. For AM processes, potential defects include incomplete fusion, gas porosity, and quench cracking. It is essential that any defects produced that would compromise the integrity of the part can be found by appropriate NDE methods. Conversely, brute force and trial-and-error NDE qualification could compromise the cost-effectiveness of the AM process.

An exploratory 9-month SBIR Phase I project funded by AFRL was performed to confirm the feasibility of combining NDE modeling with probabilistic fracture mechanics modeling (i.e., DARWIN) to address this challenge. SwRI teamed with small business NDE Technologies, Inc.™ to perform the study. The X-ray simulator (XRSIM) software developed by NDE Technologies was used in conjunction with DARWIN.

The XRSIM simulations were used to determine the POD of defects at various locations in a complex 3-D part produced by direct metal laser sintering and inspected by X-ray methods. Fatigue crack growth (DARWIN) simulations were used to determine critical initial defect sizes (to determine what needs to be found) and also to determine the probability that an undetected defect would grow to fracture during the service lifetime of the part. The joint NDE-fracture modeling system could be used to optimize both the design of the part and the design of the NDE inspection plan.

If successful, the paradigm could easily be expanded to include other NDE methods (such as eddy current and ultrasonic techniques) and to address other applications besides AM. DARWIN already uses POD curves to assess fracture risk for various anomaly types, but the POD curve must be selected manually by the DARWIN user from a limited library of standard curves. This new approach could provide a powerful means of calculating location-specific POD curves for significant regions of a mechanical component.

Based on the success of the Phase I project, the team was awarded a 2-year Phase II project to develop and study the modeling system further. The Phase II project is underway as of this writing.

10. SUMMARY

The following are major accomplishments of the PIRATE research grant:

- New DARWIN versions 7.2, 8.0, 8.1, 8.2, and 9.0 were released to the FAA and industry. Each new version was evaluated thoroughly by the development team, the project steering committee, and RISC before being released for production use. Specific enhancements are described below.
- Eighteen commercial DARWIN licenses were active at the end of this grant, including 14 manufacturers of gas turbine engines for aircraft applications. Licensing revenue was used to fund additional enhancements of DARWIN and user support, including training workshops for licensees.
- A comprehensive 3-day DARWIN training workshop was conducted in April 2013. Forty-two people from 27 different organizations in the U.S. and six foreign countries attended in person, and an additional 49 people from 12 organizations participated remotely in the first day of the workshop through a Web meeting.
- Seven other 2-day DARWIN training workshops were presented in the U.S. and three foreign countries for licensee companies and the U.S. AFRL. All costs of these seven workshops were covered by the participating organizations. Over 100 people participated in these workshops.
- The expanding capabilities of DARWIN, its increasing use in industry, and growing interest in DARWIN among other U.S. government agencies have all contributed to a significant number of DARWIN spin-off projects funded by AFRL, NAVAIR, NASA, and individual DARWIN licensee companies. These projects typically result in significant new capabilities in the production DARWIN software delivered to the FAA and all other DARWIN users.
- A new binary hierarchical input/output file format called Hierarchical Data Format (HDF5) was implemented in DARWIN that enables a reduction in the amount of computer memory

required for execution. This enables DARWIN to support very large file sizes (file sizes are theoretically unlimited) without a substantial increase in memory use.

- An optimal autozoning algorithm for 2-D FEMs was implemented in DARWIN that minimizes the total number of automatically created zones required to compute the fracture risk of a component. The GUI was enhanced to enable the user to assign material properties and inspection parameters as property regions directly on the FEM to support autozoning life and fracture risk assessments.
- An improved optimal autozoning algorithm for 3-D FEMs called “optimal autozoning with Gaussian Process pre-zoning” was implemented in DARWIN. It combines FEs with similar properties into groups called “pre-zones” for use in autozoning computations. The resulting algorithm is 1,000–10,000 times faster than a previous algorithm that created zones at individual FEs. The pre-zoning-based algorithm also requires less memory than previous algorithms, enabling it to solve much larger models.
- The DARWIN GUI was enhanced to enable the user to view non-hoop stress planes including maximum principal stress planes. The DARWIN auto-modeling algorithms (e.g., autoplate, life contours, and autozoning) were also enhanced to support non-hoop stress planes, in which the maximum principal stress plane is used as the default non-hoop stress plane.
- DARWIN was enhanced to facilitate the creation of surface and line zones to support RISC blade slot damage studies. This new capability enables definition of surface and line zones via selection of FE faces and edges, respectively. DARWIN was also enhanced to support anomaly distributions for line zones that are based on the number of anomalies per unit length.
- The DARWIN GUI was restructured to provide a centralized location for optional features and to accommodate additional new features. This information is stored in three new menu windows (i.e., Configuration, Optional Features, and Global Factors) to enable efficient user input of general analysis parameters.
- A 64-bit version of the GUI was implemented in DARWIN to support memory-intensive operations such as the display and manipulation of 3-D FEMs. The memory of the 64-bit GUI is no longer limited by the operating system, and it can be increased directly by the user via hardware improvements to meet future computational demands.
- DARWIN was enhanced to provide support for 3-D sector models. This new capability enables users to import and display 3-D FE sector models and to visualize a wireframe outline of the full component geometry. It enables a crack to extend beyond the sector boundary and into the full-component model for use in life and risk assessments. DARWIN was enhanced to provide treatment for large FEMs with large numbers of load steps. A new binary data structure (i.e., HDF5) was implemented for storage of FE data, enabling fast retrieval of selected portions of results files using significantly less memory compared with the previous ASCII format. The DARWIN GUI was enhanced to provide a user with the option to disable viewing of stress and temperature contours, further reducing the time required to import large FEMs by a factor of 100 or more, depending on the size of the FE model.
- Full GUI support for cycle-dependent and time-dependent overload crack growth retardation models was added to DARWIN, allowing users to use optional modified Willenborg methods in life and risk calculations.

- A new TMF crack growth option was added to DARWIN that enables crack propagation assessment based on material properties corresponding to the maximum temperature during the flight. This is one of the optional analysis methods called out in AC 33.70-2.
- Improved methods for temperature interpolation of cycle-dependent FCG rates were developed and implemented in DARWIN.
- New capabilities were added to DARWIN to allow users to provide local RS gradients (e.g., due to surface enhancements such as peening) and combine them with fatigue stresses for 2-D or 3-D FEMs in life and risk calculations involving univariant SIF models.
- The DARWIN Fleet Assessment Module was enhanced to provide a new option for disk replacement following a simulated inspection event. This new feature enables DARWIN computation of the additional risk associated with replaced components for use in fleet assessment computations.
- The DARWIN Fleet Assessment Module was enhanced to provide a new capability for corrective actions that enables replacement of existing disks with new disks that may have completely different geometries. This capability enables new disks to be placed in service based on a schedule associated with a disk-replacement corrective action.
- A new DARWIN capability was developed to assess the risk associated with the introduction of new anomalies at shop visits, such as surface damage occurring at attachment slots during disassembly or assembly. This includes treatment of cracks that may form sometime after an anomaly is initially introduced, and it also addresses the variability associated with nonzero crack formation lives. Several new WF SIF solutions that accept bivariant stress gradients were developed and implemented in DARWIN. The specific SIF solutions included an offset surface crack in the bore of an offset circular hole in a plate (SC29), a crack spanning a chamfered corner (CC12), and a crack at a corner with an angle of 135° (CC18). Improved univariant (SC30) and bivariant (SC31) WF SIF solutions for an offset surface crack in a plate were also developed and implemented.
- Experimental FCG rate data were generated at 600°F for small fatigue cracks in Alloy 718, a common rotor alloy for which no appropriate small-crack data were previously available. This temperature was chosen as it is high enough to emulate realistic rotor service but low enough to avoid time-dependent effects. These data were used to evaluate further a previously proposed engineering model for small-crack effects.
- Significant results from this research program and related spin-off projects were disseminated to a much broader audience in the gas turbine engine community or the international technical community through conference presentations or publications in archival technical journals. Twenty-three published papers (most of them presented at international conferences) and 19 additional presentations without manuscript occurred during the term of the current grant.

11. REFERENCES

1. Wu, Y. T., Enright, M. P., and Millwater, H. R. (2002). Probabilistic methods for design assessment of reliability with inspection. *AIAA Journal, 40*(5), 937–946.

2. Millwater, H., Wu, J., Riha, D., Enright, M., Leverant, G., McClung, C., … Meyer J. (2000). *A probabilistically-based damage tolerance analysis computer program for hard alpha anomalies in titanium rotors.* Paper presented at the 45th ASME International Gas Turbine & Aeroengine Technical Congress, Munich, Germany (Paper 2000-GT-0421).

3. McClung, R. C., Lawless, B. H., Gorelik, M., Date, C., Gill, Y., and Piascik, R.S. (1998). *Fatigue behavior of titanium alloys*: *fatigue crack growth of titanium rotor alloys in vacuum and air*. Paper presented at the TMS Fall Meeting '98, Symposium on Fatigue Behavior of Titanium Alloys, Chicago, IL.

4. Chan, K. S., Perocchi, L., and Leverant, G. R. (2000). Constitutive properties of hard alpha titanium. *Metallurgical and Materials Transactions A, 31A*(12), 3029–3040.

5. Chan, K.S. (2001) Constitutive relations for inelastic deformation and damage accumulation in hard alpha titanium. *ASME Transactions, Journal of Engineering Materials and Technology, 123*(3), 281–286.

6. McKeighan, P. C., Perocchi, L. C., Nicholls, A. E., and McClung, R. C. (1999). *Characterizing the cracking behavior of hard alpha defects in rotor grade Ti-6-4 alloy*. Paper presented at Symposium on Fatigue Behavior of Titanium Alloys, Chicago, IL.

7. FAA Report. (2000). Turbine Rotor Material Design (DOT/FAA/AR-00/64).

8. Enright, M. P., Lee, Y. D., McClung, R. C., Huyse, L., Leverant, G. R., Millwater, H. R., and Fitch, S. K. (2003). *Probabilistic surface damage tolerance assessment of aircraft turbine rotors*. Paper presented at 48th ASME International Gas Turbine & Aeroengine Technical Congress, Atlanta, GA (Paper GT-2003-38731).

9. McClung, R. C., Enright, M. P., Lee, Y. D., Huyse, L., and Fitch, S. (2004). *Efficient fracture design for complex turbine engine components*. Paper presented at 49th ASME International Gas Turbine & Aeroengine Technical Congress, Vienna, Austria (Paper GT-2004-53323).

10. Wu, Y. T., Enright, M. P. and Millwater, H. R. (2000). *Efficient and accurate methods for probabilistic analysis of titanium rotors*. Paper presented at the 8th ASCE Joint Specialty Conference on Probabilistic Mechanics and Structural Reliability, Notre Dame, IN (Paper PMC2000-221).

11. Enright, M. P., and Millwater, H. R. (2002). *Optimal sampling techniques for zone-based probabilistic fatigue life prediction*. Paper presented at the 43rd Structures, Structural Dynamics, and Materials Conference, Denver, CO (AIAA Paper 2002-1383).

12. Huyse, L., and Enright, M. P. (2003). *Efficient statistical analysis of failure risk in engine rotor disks using importance sampling techniques*. Paper presented at the 44th Structures, Structural Dynamics, and Materials Conference, Norfolk, VA (Paper 2003-1838).

13. Enright, M. P., Millwater, H. R., and Huyse, L. (2006). Adaptive optimal sampling methodology for reliability prediction of series systems. *AIAA Journal, 44*(3), 523–528.

14. Laz, P. J., Chan, K. S., McClung, R. C., and Leverant G. R. (2003). Effects of CTE-induced residual stresses around hard alpha particles on fatigue crack growth in Ti-6Al-4V. *Fatigue and Fracture of Engineering Materials and Structures, 26*(12), 1145–1157.

15. McClung, R. C. (2007). A literature survey on the stability and significance of residual stresses during fatigue. *Fatigue and Fracture of Engineering Materials and Structures, 30*(3), 173–205.

16. FAA Report. (2008). Turbine Rotor Material Design—Phase II (DOT/FAA/AR-07/13).

17. Lee, Y. D., McClung, R. C., and Chell, G. G. (2008). An efficient stress intensity factor solution scheme for corner cracks at holes under bivariant stressing. *Fatigue and Fracture of Engineering Materials and Structures, 31*(11), 1004–1016.

18. Enright, M. P., McClung, R. C., Liang, W., Lee, Y. D., Moody, J. P., and Fitch, S. (2012). *A tool for probabilistic damage tolerance of hole features in turbine engine rotors*. Paper presented at the ASME Turbo Expo 2012, Copenhagen, Denmark (GT2012-69968)

19. Moody, J. P., Millwater, H. R., and Enright, M. P. (2007). *Automatic risk assessment methodology for gas turbine engines employing adaptive mesh refinement techniques*. Paper presented at the 52nd ASME International Gas Turbine & Aeroengine Technical Congress, Montreal, Canada (GT2007-27576).

20. Enright, M. P., Liang, W., Moody, J. P., and Fitch, S. (2010). *Development of risk contours for assessment of aircraft engine components*. Paper presented at the 12th AIAA Non-Deterministic Approaches Conference, Orlando, FL (AIAA-2010-2846).

21. McClung, R. C., Lee, Y. D., Liang, W., Enright, M. P., and Fitch, S. H. K. (2010). *Automated fatigue crack growth analysis of components*. Paper presented at the 10th International Fatigue Congress (Fatigue 2010), Prague, Czech Republic. See also *Procedia Engineering, 2*(1), 629–637.

22. FAA Report. (2012). Probabilistic Design for Rotor Integrity. Final Report to Federal Aviation Administration, Grant 2005-G-005.

23. FAA. (2001, January). Advisory Circular 33.14-1, *Damage Tolerance for High Energy Turbine Engine Rotors*.

24. FAA. (2009, August). Advisory Circular 33.70-2, *Damage Tolerance of Hole Features in High-Energy Turbine Engine Rotors*.

25. FAA. (2003, September). Advisory Circular 39-8. *Continued Airworthiness Assessments of Powerplant and Auxiliary Power Unit Installations of Transport Category Airplanes.*

26. Van Stone, R. H., and Slavik, D. C. (2000). Prediction of Time-Dependent Crack Growth with Retardation Effects in Nickel Base Alloys. *Fatigue and Fracture Mechanics, 31st Volume, ASTM STP 1389,* (405-426). West Conshohocken, PA: American Society for Testing and Materials.

27. NASA Report. (1992). FASTRAN II - Fatigue Crack Growth Structural Analysis Program (NASA TM-104159).

28. Li, S., Mear, M. E., and Xiao, L. (1998). Symmetric weak-form integral equation method for three-dimensional fracture analysis. *Computer Methods in Applied Mechanics and Engineering*, *151*, 435–539.

29. McClung, R. C., and Bhamidipati, V. (2016). An investigation of small-crack effects in various aircraft engine rotor materials. *Materials at High Temperatures*, *33*(4–5), 452–464.

30. El Haddad, M. H., Smith, K. N., and Topper, T. H. (1979). Fatigue crack propagation of short cracks. *Journal of Engineering Materials and Technology, Transactions of the ASME*, 101, 42–46.

31. McClung, R. C., Enright, M. P., Liang, Moody, J. P., Wu, W. T., Shankar, R., Luo, W., Oh, J., and Fitch, S. (2012). *Integration of manufacturing process simulation with probabilistic damage tolerance analysis of aircraft engine components*. Paper presented at the 14th AIAA Non-Deterministic Approaches Conference, Honolulu, HI. (AIAA-2012-1528).

32. Enright, M. P., McFarland, J., McClung, R. C., Moody, J. P., Wu, W. T., and Shankar, R. (2013). *Probabilistic integration of material process modeling and fracture risk assessment using Gaussian process models*. Paper presented at the 15th AIAA Non-Deterministic Approaches Conference, Boston, MA (DOI 10.2514/6.2013-1851).

33. Enright, M. P., McClung, R. C., Chan, K. S., McFarland, J., Moody, J. P., and Sobotka, J. C. (2016). *Micromechanics-based fracture risk assessment using integrated probabilistic damage tolerance analysis and manufacturing process models*. Paper presented at the, ASME Turbo Expo 2016, Seoul, South Korea (GT2016-58089).

34. McClung, R. C., Wawrzynek, P. Lee, Y. D., Carter, B. J., Moody, J. P, and Enright, M. P. *An integrated software tool for high fidelity probabilistic assessments of metallic aero-engine components*. Paper presented at the ASME Turbo Expo 2016, Seoul, South Korea (GT2016-57877).

35. Chan, K. S., Enright, M. P, Moody, J. P., Hocking, B., and Fitch, S. H. K. (2012). *Life Prediction for Turbopropulsion Systems Under Dwell Fatigue Conditions, Journal of Engineering for Gas Turbines and Power, 134*(122501), 8. Also published as Paper GT2012-69742, *Proc. of ASME Turbo Expo 2012,* Copenhagen, Demark, June 2012.

36. Chan, K. S., Enright, M. P., Moody, J., and Fitch, S. H. K. (2014). A microstructure-based time-dependent crack growth model for life and reliability prediction of turbopropulsion systems. *Metallurgical and Materials Transactions A*, *45A*, 287–301. (DOI:10.1007/s11661-013-1971-9).

37. Chan, K. S. (2014). Time-dependent crack growth thresholds of Ni-base superalloys. *Metallurgical and Materials Transactions A*, *45A*, 3454–3466. (DOI: 10.1007/s11611-014-2271-8, 05).

38. Chan, K. S. (2015). A grain boundary fracture model for predicting dynamic embrittlement and oxidation-induced cracking in superalloys, *Metallurgical and Materials Transactions A*, *46A*, 2491–2505. (DOI:10.1009/s11661-015-2860-1).

39. Chan, K. S. (2016). Mechanistic modeling of time-dependent fatigue crack growth in Ni-based superalloys, *Materials at High Temperatures*, 33(4-5), 425–438.

40. Chan, K. S. and Moody, J. (2016). A hydrogen-induced decohesion model for treating cold dwell fatigue in Ti-based alloys. *Metallurgical and Materials Transactions A*, *47A*, 2058–2072. (DOI: 10.1007/s11661-016-3367-0).

41. Chan, K. S., Enright, M. P., Moody, J., and Fitch, S. H. K. (2016). Mitigating time-dependent crack growth in Ni-base superalloy components. *International Journal of Fatigue*, *82*, 332–341.

42. Chan, K. S., Enright, M. P., and Moody, J. P. (2014). Development of a probabilistic methodology for predicting hot corrosion fatigue crack growth life of gas turbine engine disks. *Journal of Engineering Gas Turbines and Power*, *136*, Paper 22502 (8 pp). Also published as Paper GT2013-95057, *Proceedings of ASME Turbo Expo 2013*, San Antonio, TX, June 2013.

43. Chan, K. S., Enright, M. P., Moody, J., and Fitch, S. H. K., (2016). *Physics-based modeling tools for predicting type II hot corrosion in Ni-based superalloys*. Paper presented at Superalloys 2016 Symposium, Seven Springs, PA.

APPENDIX A—TWO-DIMENSIONAL OPTIMAL AUTOZONING

A.1 AUTOMATIC ZONE-BASED RISK ASSESSMENT

The fracture risk of a zone $P[F_i]$ is defined as the product of the probability of having an anomaly in the zone $P[A_i]$ and the probability of fracture given the presence of an anomaly in the zone $P[B_i|A_i]$ (equation A-1). $P[B_i|A_i]$ is also referred to as the conditional risk of the zone:

$$P[F_i] = P[B_i|A_i] \cdot P[A_i] \tag{A-1}$$

In Design Assessment of Reliability With INspection (DARWIN®), the conditional risk is computed using a fracture-model plate generation algorithm. This algorithm defines a fracture mechanics geometry model at any crack location based on the geometry of the part, material properties, and the loading profile. The orientation and dimensions of the generated plates are based on the most likely crack growth path. An example of an automatically generated fracture-model plate is shown in figure A-1.

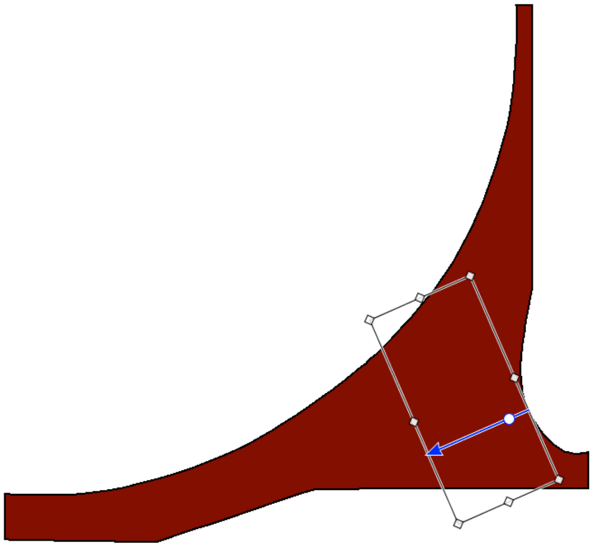


Figure A-1. Automatically-generated fracture-model plate superimposed on a 2-D geometry

Material volume, along with a material and process-dependent anomaly rate, is used to compute the probability of having an anomaly. Therefore, by assigning the material volume associated with each finite element (FE) in a 2-D axisymmetric model to an individual zone, $P[A_i]$ can be automatically computed. In this scheme, the number of zones in the zone-based risk model will equal the number of FEs in the 2-D finite element model (FEM). Figure A-2 shows the breakup of a 2-D FEM into zones, where an individual zone is assigned to each FE.

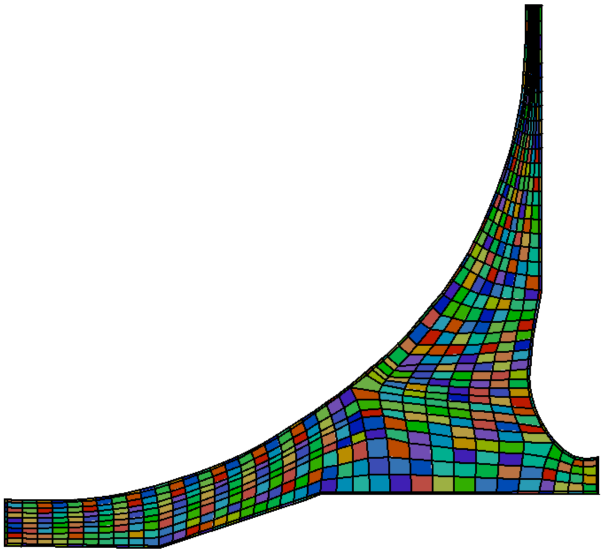


Figure A-2. Zone-based risk model consisting of one FE per zone

For each zone, a fracture-model plate is automatically generated at the location of the highest risk node and the fracture risk is computed. The highest risk node within a zone is referred to as the risk-limiting location. It is critical that the risk-limiting location is used to compute $P[B_i|A_i]$ to avoid anti-conservative fracture risk solutions. An example 2-D contour of fracture risk is shown in figure A-3.

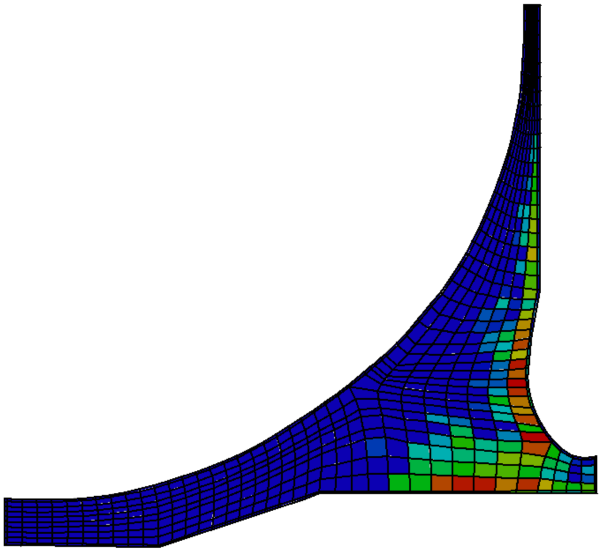


Figure A-3. 2-D contour of fracture risk

The component fracture risk $P\left[F_{component}\right]$ is the union of the individual zone risks. The equation for component fracture risk is given by:

$$P\left[F_{component}\right] = P[F_1 \cup F_2 \cup \cdots \cup F_n] = 1 - \prod_{i=1}^{n}(1 - P[F_i]) \qquad \text{(A-2)}$$

where n is the number of zones in the risk model. Because the number of zones in the risk model equals the number of FEs in the FEM, the equation can be rewritten as:

$$P[F_{component}] = 1 - \prod_{i=1}^{N}(1 - P[F_i]) \tag{A-3}$$

where *N* is the number of elements in the FEM.

The fracture risk within each zone is set to the value at the risk critical location within each zone. This approximation is conservative for relatively large zones where the value of risk may vary significantly throughout the zone. For relatively small zones, the fracture risk within the zone approaches the value at the risk critical location. Component fracture risk therefore decreases with increasing number of zones, and converges to a single value as the fracture risk within each zone approaches the value at the risk critical location. A common approach for quickly identifying a large number of zones for a component is to assign one zone to each FE. This approach typically provides a converged component risk result. However, risk convergence can often be achieved with tens or even hundreds of FEs assigned to some of the zones.

The one-zone-per-FE approach may require significant computation time, particularly for 2-D FEMs with large numbers of FEs. To improve computational efficiency, an FE allocation algorithm was developed to provide an optimal allocation of multiple FEs to zones to minimize the total number of zones required to reach fracture risk convergence.

A.2 OPTIMIZED AUTOMATIC ZONE-BASED RISK ASSESSMENT

The objective of the optimized automatic risk assessment methodology is to minimize the total number of zones *n* required to reach component fracture risk convergence. The methodology is a constrained optimization problem mathematically represented by equations A-4 through A-8:

Minimize:

$$n \tag{A-4}$$

Subject to:

$$P[F_{component}] \leq P[F_{component}]_{target} \tag{A-5}$$

$$\left|P[F_{component}]_k - P[F_{component}]_{k-1}\right| \leq \Delta P[F_{component}]_{th} \tag{A-6}$$

$$1 \leq n \leq N \tag{A-7}$$

$$\text{Contiguous zones} \tag{A-8}$$

The first constraint (see equation A-5) determines whether a converged fracture risk solution is warranted. As analysts are often primarily concerned with satisfying a component reliability standard, the achievement of a target fracture risk $P[F_{component}]_{target}$ constitutes an acceptable risk model. Producing a more accurate risk model that provides a lower fracture risk estimation is not beneficial.

The second constraint (see equation A-6) establishes the achievement criterion for fracture risk convergence. Component fracture risk convergence is achieved when the change in component fracture risk falls below the convergence threshold $\Delta P\left[F_{component}\right]_{th}$. The convergence threshold represents the relative change in fracture risk across a given number of consecutive refinements. For example, a 5% convergence threshold across five refinements means that the relative change in fracture risk must remain below 5% for five consecutive refinements in order for convergence to be achieved.

The third constraint (see equation A-7) indicates that the number of zones in a risk model must be less than or equal to the number of FEs in an FEM and greater than or equal to a single zone. A single-zone risk model assigns all FEs to a single zone. It provides a conservative estimation of fracture risk. A risk model consisting of N zones represents the one-zone-per-FE approach described previously.

The fourth constraint (see equation A-8) prohibits the presence of non-continuous zones, meaning all elements assigned to a zone must be adjacent to at least one other member element. Zone contiguity improves the efficiency of the FE allocation algorithm and provides more visually aesthetic risk models.

The optimized automatic risk assessment methodology is composed of two primary stages. The first stage predicts the optimal risk model (i.e., the risk model that provides the most accurate estimate of the component fracture risk) for any given number of zones using predictive fracture risk values. This stage is referred to as the deconstructive stage because it begins with the one-zone-per-FE model and iteratively breaks it down until a single-zone risk model is obtained. The second stage computes the actual component fracture risk. It begins with the single-zone risk model and iteratively refines it until convergence is achieved. Risk models are refined based on the optimal FE allocations computed in the first stage.

A.3 FE ALLOCATION ALGORITHM

If the outcome of all possible responses to a circumstance is known beforehand, the optimal response (i.e., the response that yields the most favorable outcome) can be determined. Accordingly, by evaluating the component fracture risk for all possible allocations of FEs to zones, the optimal zone-based risk model can be identified. The optimal risk model for any given number of zones is computed using a deconstructive process that breaks down the highest fidelity risk model into a single zone.

To simulate the component fracture risk for all possible allocations of FEs to zones, conditional risk at any location in a 2-D FEM must be available. Actual conditional risk computations would defeat the objective to improve computational expense; therefore, a predictive conditional risk response surface is used. The predictive response surface is used to estimate the conditional risk at any location in a 2-D FEM. The predicted fracture risk of a zone $P^*[F_i]$ is given by:

$$P^*[F_i] = P^*[B_i|A_i] \cdot P[A_i] \qquad \text{(A-9)}$$

where $P^*[B_i|A_i]$ is the predicted conditional risk at the risk-limiting location within a zone. The predicted component fracture risk $P^*\left[F_{component}\right]$ is therefore given by equation A-10:

$$P^*\left[F_{component}\right] = 1 - \prod_{i=1}^{n}(1 - P^*[F_i]) \tag{A-10}$$

Beginning with the highest fidelity risk model, the allocation algorithm iteratively merges two zones at a time until a single-zone risk model is obtained. At each deconstruction step, $P^*\left[F_{component}\right]$ is computed for all possible allocations of FE to zones. The allocation that yields the smallest change in fracture risk represents the optimal risk model for the given number of zones. The change in component fracture risk $\Delta P^*\left[F_{component}\right]$ for a risk model consisting of *n* zones is given by the following equation:

$$\Delta P^*\left[F_{component}\right] = P^*\left[F_{component}\right]_n - P^*\left[F_{component}\right]_N \tag{A-11}$$

where $P^*\left[F_{component}\right]_N$ is the component fracture risk computed using the highest fidelity model, which provides the most accurate fracture risk solution achievable for an FEM containing *N* FEs. The magnitude of $\Delta P^*\left[F_{component}\right]$ increases as the number of zones decreases. Therefore, the lowest fidelity risk model will have the highest fracture risk error, and the highest fidelity risk model will have no fracture risk error.

The fracture risk for a merged zone pair $P^*\left[F_{i+j}\right]$ is given by:

$$P^*\left[F_{i+j}\right] = max\left(P^*[B_i|A_i], P^*\left[B_j|A_j\right]\right) \cdot \left(P[A_i] + P\left[A_j\right]\right) \tag{A-12}$$

As the new zone possesses the material volume of two zones, its anomaly occurrence probability is the union of the merged zones' individual occurrence probabilities $P[A_i]$ and $P\left[A_j\right]$. The conditional risk is the maximum conditional risk value between the merged zones' conditional risks $P^*[B_i|A_i]$ and $P^*\left[B_j|A_j\right]$. The maximum value represents the risk-limiting location in the new zone. The zone merge at each deconstruction step that yields the optimal risk model is referred to as the optimal zone pair.

Once the optimal risk model for any number of zones is computed, the deconstruction sequence is applied in reverse order to compute the actual component fracture risk. This stage makes use of actual conditional risk computations $P[B_i|A_i]$ for each zone. Component fracture risk convergence is achieved when the change in component fracture risk satisfies a convergence threshold or a target component fracture risk is reached.

A.3.1 DEMONSTRATION EXAMPLE

To demonstrate the FE allocation algorithm, consider a simple four-element axisymmetric FEM (see in figure A-4). Initially, a predictive conditional risk response surface is constructed and the risk-limiting location (i.e., the highest risk node) is identified in each element. The nodes N1, N2, N4, and N5 represent the risk-limiting locations for Elements 1, 2, 3, and 4, respectively. To ensure conservatism in the fracture risk solution, the risk at the risk-limiting location is uniformly distributed throughout each element.

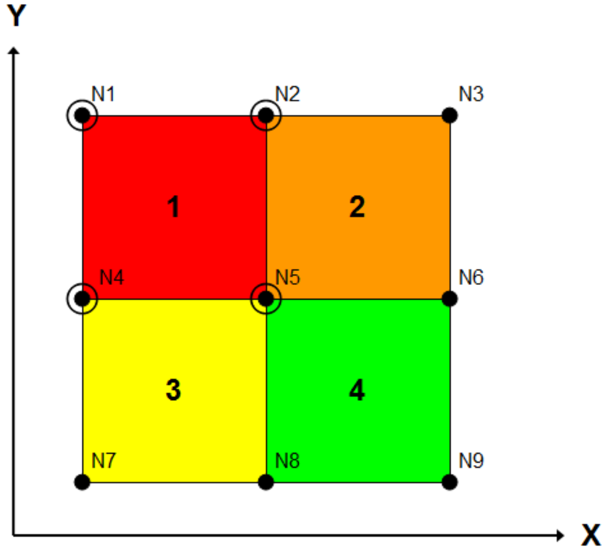
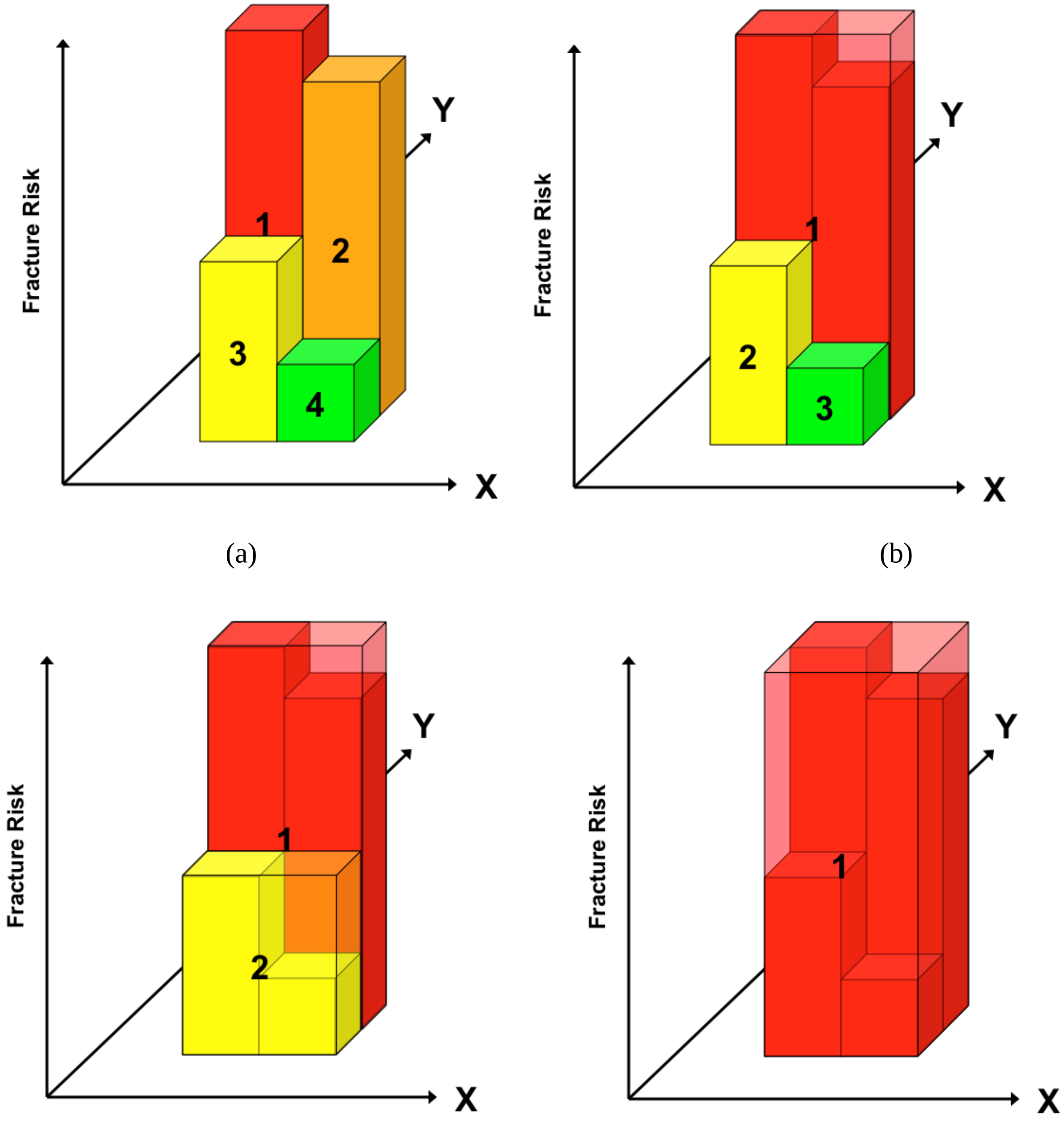


Figure A-4. Example four-element axisymmetric FEM

If each FE is treated as a zone, a four-zone risk model is produced. The fracture risk for each zone is computed and is plotted in figure A-5(a). The fracture risk of the part is the summation of the fracture risks in Zones 1, 2, 3, and 4. The four-zone risk model represents the most accurate risk model possible for a four-element FEM.



(a)

(b)

(c)

(d)

Figure A-5. Optimal risk models consisting of:
(a) 4 zones, (b) 3 zones, (c) 2 zones, and (d) 1 zone

In the first deconstruction step, a single zone is removed from the four-zone risk model by merging two zones. All possible zone pairs and their associated risk model configurations are listed in table A-1. The highlighted rows represent risk models consisting of non-contiguous zone pairs; these risk models are removed from consideration. For the remaining risk models, the fracture risk of the part is computed, and Risk Model 1 is determined to produce the smallest change in fracture risk. Risk Model 1 represents the optimal allocation of FEs to zones for a three-zone risk model (see figure A-5(b)). This process is continued until a single zone is obtained. Figures A-5(c) and A-5(d) show the optimal risk models consisting of 2 and 1 zones, respectively.

Table A-3. Possible allocations of FEs to zones

Risk Model ID	Zone 1	Zone 2	Zone 3	$\Delta \boldsymbol{P}^{*}[\boldsymbol{F}_{\boldsymbol{component}}]$
1	1,2	3	4	5.0E-9
2	1,3	2	4	2.0E-8
3	1,4	2	3	---
4	1	2,3	4	---
5	1	2,4	3	2.5E-8
6	1	2	3–4	1.0E-8

A.3.2 APPLICATION EXAMPLE

A case study is performed to evaluate and illustrate the effectiveness of the FE allocation algorithm. It is based on a practical, real-world model of a gas turbine engine impeller blade represented by a 692-element axisymmetric FEM.

The predictive conditional risk response surface is based on a fast Monte Carlo probabilistic analysis method that uses a series of life-approximating functions to predict the fatigue crack growth life for each sample. The response surface for the impeller blade is shown in figure A-6.

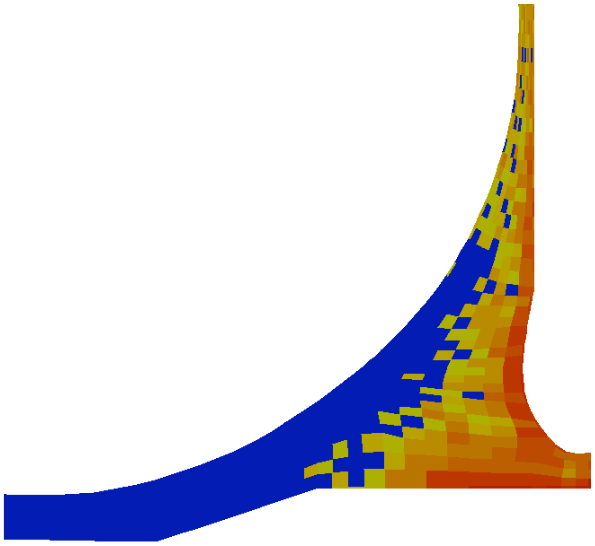
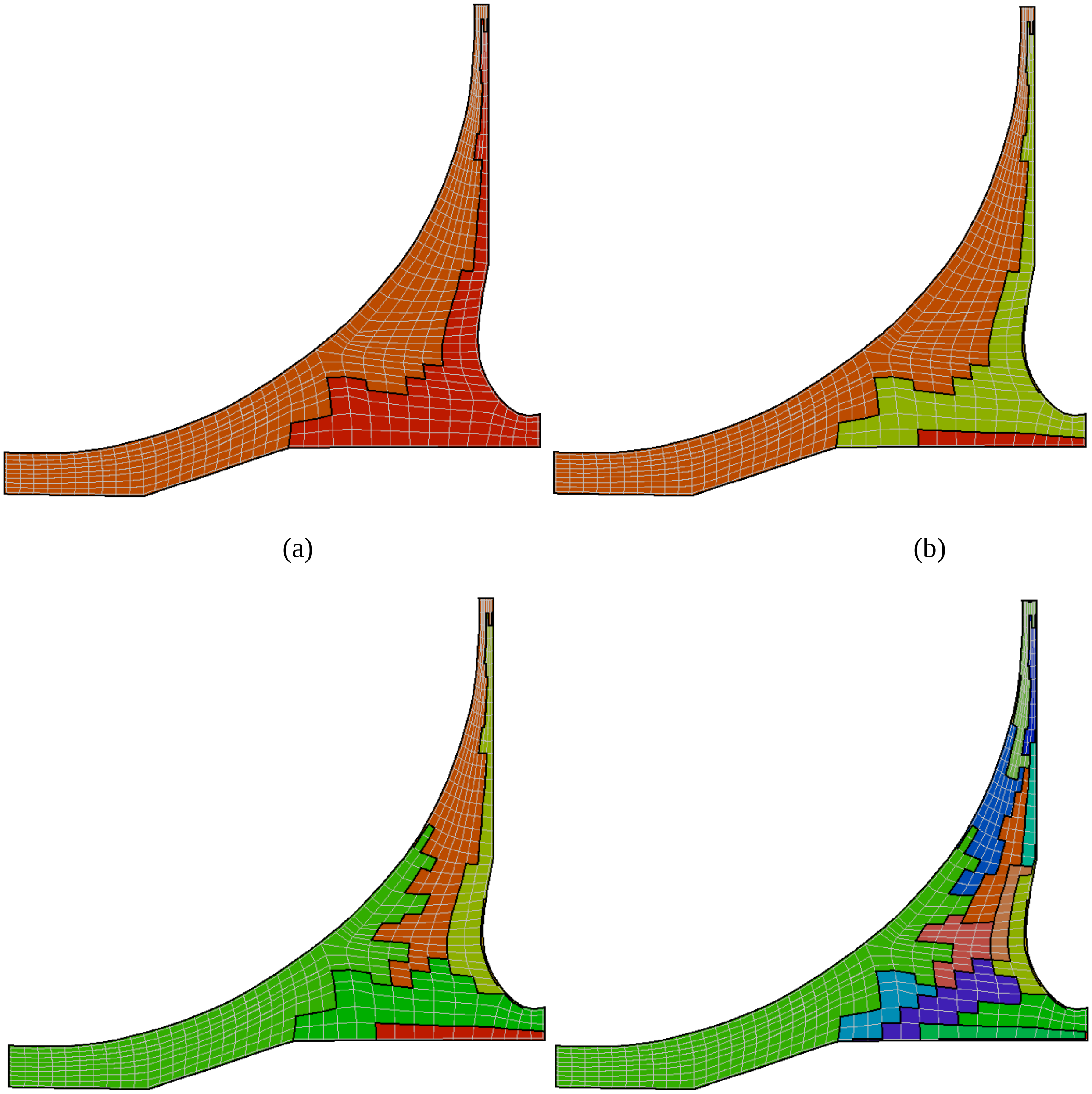


Figure A-6. Predictive conditional risk response surface used to predict the optimal allocation of FEs to zones

The FE allocation algorithm is exercised using the predictive response surface, and the optimal risk model is obtained for any number of zones. The deconstruction sequence is then reversed, and the fracture risk of the part is computed for each optimal risk model beginning with the single-zone model. At each refinement step, a zone is added to the previous risk model and fracture risk is computed. This process continues until the fracture risk converges.

Optimal FE allocations for the two-, four-, six-, and 17-zone risk models are shown in figure A-7. The 17-zone risk model represents the converged fracture risk solution for a 5% convergence threshold across five consecutive refinements. The fracture risk convergence is plotted in figure A-8. For the risk model consisting of a single zone, the predicted risk is approximately 23 times the risk predicted using the one zone per FEM. This conservatism is quickly reduced through each refinement step. For the risk model consisting of 17 zones, the predicted risk is approximately 30% greater than the one zone per FEM. The conservatism can be further reduced by setting a smaller convergence threshold, which will introduce more zones into the optimal risk model.



(a)

(b)

(c)

(d)

Figure A-7. Optimal FE allocation for risk models containing: (a) 2 zones, (b) 4 zones, (c) 6 zones, and (d) 17 zones

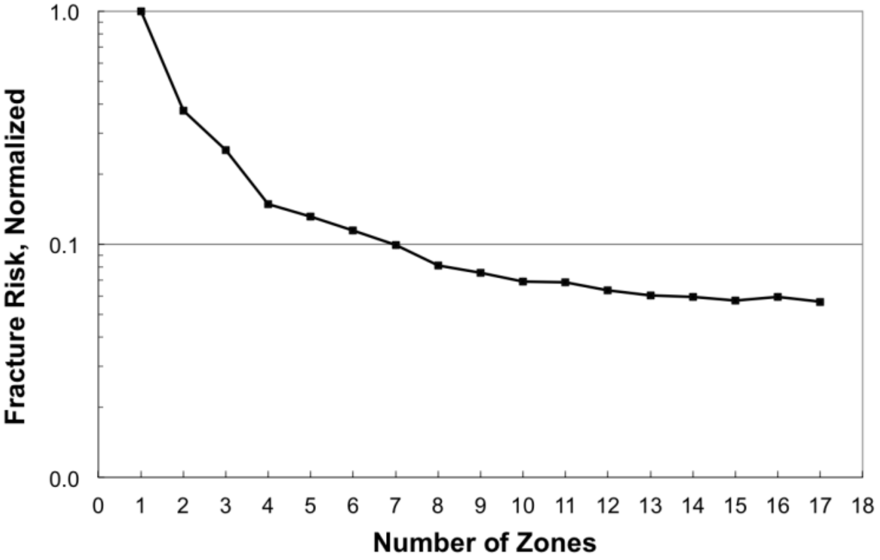
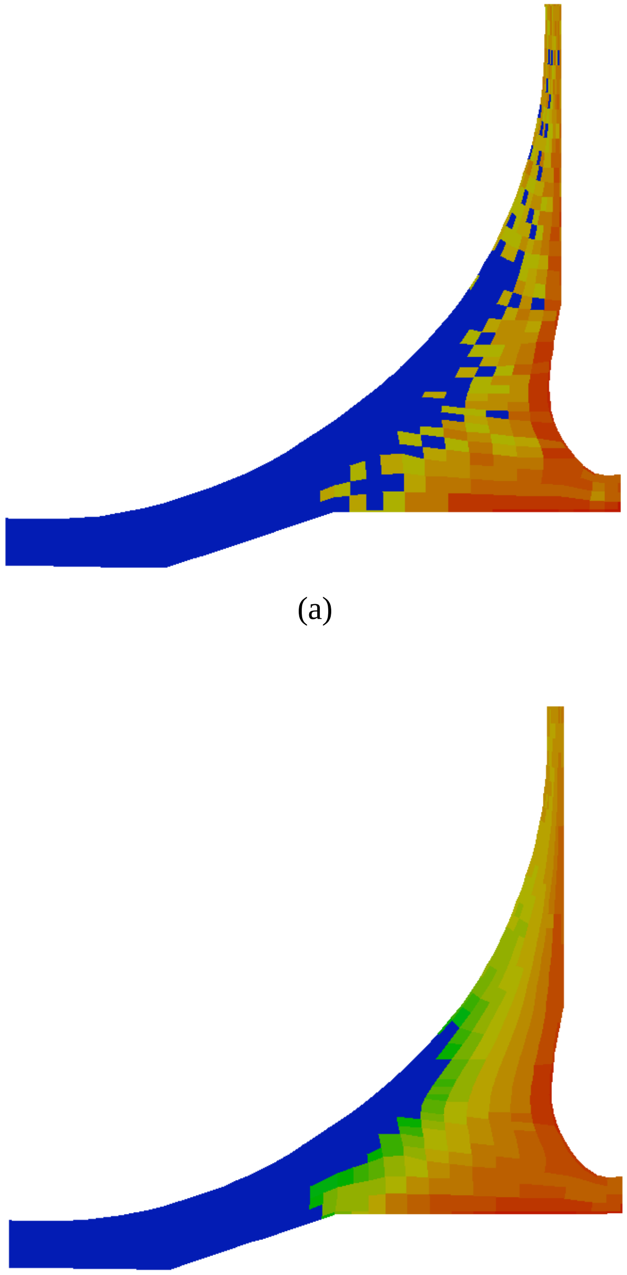


Figure A-8. Convergence of fracture risk

A noteworthy observation regarding the optimal risk models is the lack of smoothness and continuity in the prescribed zone boundaries. As shown in the risk models provided in figure A-7, zone boundaries have a tendency to creep into regions predominantly represented by other zones. This effect is due largely in part to sampling error, which introduces discontinuity in the underlying predictive response surface used to optimize the FE allocation.

To illustrate the effect of sampling error, figure A-9 provides the response surfaces generated using 10,000 samples/zone (see figure A-9(a)) and 1,000,000 samples/zone (see figure A-9(b)). Figure A-9(b) shows a smooth transition of conditional risk values throughout the domain of the impeller blade, whereas figure A-9(a) contains numerous abruptions in the conditional risk trend. As a result, the associated 17-zone risk models display apparent differences in the smoothness of the zone boundaries. Specifically, the zone boundaries for the analysis using 1,000,000 samples/zone (see figure A-10(b)) are much smoother than the zone boundaries for the analysis using 10,000 samples/zone (see figure A-10(a)).



(a)

(b)

Figure A-9. Predictive conditional risk response surface using (a) 1E4 samples/zone and (b) 1E6 samples/zone

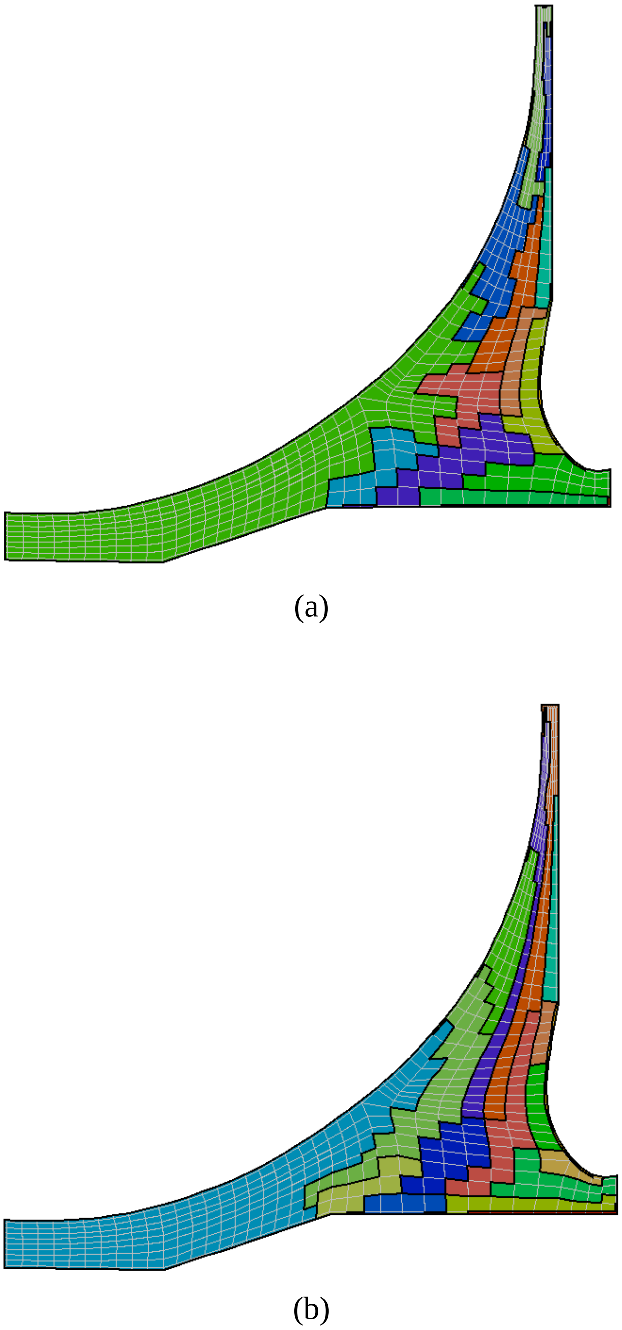


Figure A-10. Optimal FE allocation for a 17-zone risk model based on a predictive response surface using (a) 1E4 samples/zone and (b) 1E6 samples/zone

A.4 CONCLUSIONS

To improve the computational efficiency of an exhaustive automatic risk assessment methodology, a simple and robust FE allocation algorithm was developed. This algorithm provides an optimal allocation of multiple FEs to zones with the objective of minimizing the total number of zones required to reach fracture risk convergence. Using an FEM of an impeller blade, it was demonstrated that the algorithm effectively reduces the number of zones required to obtain a converged fracture risk solution. Further study is needed regarding FE mesh size requirements, risk critical locations, and the influence of inspection on optimal zone configuration.

APPENDIX B—3-D OPTIMAL AUTOZONING ALGORITHIM

B.1 INTRODUCTION

Design Assessment of Reliability With INspection (DARWIN®) uses a zone-based risk-assessment methodology to estimate component fracture risk based on groupings of finite elements (FEs) called zones associated with FE stress and temperature models. Zoning enables DARWIN to compute unconditional fracture risk (i.e., the probability of fracture considering that an anomaly may or may not exist in the component). Independent software has also been developed for fracture risk assessments in the nuclear, aerospace, and power-generation industries [B-1–B-4]. These codes do not use zones or other volume-based techniques and can only perform conditional fracture risk assessments (i.e., the probability of fracture assuming that the component has a pre-existing anomaly).

To compute unconditional fracture risk, the conditional fracture risk and the volume over which it is computed must both be valid. In the lower limit, the unconditional risk associated with an anomaly placed at a random location within a finite element model (FEM) can be evaluated by first computing the conditional risk at each element within the component to form a risk contour. The unconditional risk can then be estimated by integrating the conditional risk over each differential volume in the component. This approach is not computationally tractable for FEMs containing large numbers of FEs. In the upper limit, unconditional risk could be evaluated by computing conditional risk at a single risk-limiting location and then applying this constant value over the entire volume of the component. This approach typically overestimates the unconditional risk.

The zoning concept arose to achieve a balance between speed and accuracy for fracture risk computations. Zones are used to represent regions of a component with similar conditional risk values and are often characterized by groups of individual FEs. The conditional fracture risk is modeled as a single constant value within each zone. If an anomaly is placed at the highest risk location of a zone, the computed conditional risk within each zone will be conservative.

The assignment of FEs to zones is one of the key technologies associated with the zone-based risk assessment methodology. Creating zones manually is time consuming and requires human judgment. Previous papers have presented algorithms for assigning FEs to zones based on adaptive refinement of manually created zones [B-5–B-6] or automatically created zones that require approximate risk estimates at each of the nodes in a component model [B-7]. Though significantly faster and more accurate than manual zoning, the computation times associated with automatically created zones appear to increase exponentially with the number of FEs. Complex 3-D FEMs typically contain up to millions of elements, approximately two to three orders of magnitude more elements than 2-D FEMs. The time required to compute approximate risk values at all nodes in a 3-D FEM could be measured in days or even weeks, which is not a feasible solution.

An improved optimal autozoning methodology has been developed for 3D FEMs based on an initial partitioning of the FEM into pre-zones. Pre-zones are groups of elements that are expected to have similar fracture risk values based on similar values of stress, temperature, and distance from the crack to the nearest surface. Within each pre-zone, the initial crack location is placed at the location of maximum fracture risk, which is identified via a Gaussian process (GP) model fit of estimated risk values at a small number of locations called training points. As pre-zones consist of groups of elements, the number of pre-zones will be significantly lower than the total number of elements in the FEM. For example, an FEM containing millions of elements could be represented using only hundreds of pre-zones. This methodology is applicable to both 2-D and 3-D FEMs.

A brief summary of zone-based fracture risk assessment is provided, followed by a description of a previously reported algorithm [B-7] that identifies the optimal number of zones required for a component risk assessment. A new pre-zoning methodology is presented that significantly reduces the number of zones that are applied to the optimal autozoning algorithm. A description is provided for each of the key features of the methodology, including initial pre-zone breakup, placement of training points, determination of the risk-limiting location, and pre-zone refinement. The methodology is illustrated for a representative aircraft engine gas turbine component.

B.2 ZONE-BASED FRACTURE RISK ASSESSMENT

The fracture risk of a zone $P[F_i]$ is defined as the product of the probability of having an anomaly in the zone, $P[A_i]$, and the probability of fracture given the presence of an anomaly in the zone, $P[B_i|A_i]$ (equation B-1). $P[B_i|A_i]$ is referred to as the conditional risk of the zone:

$$P[F_i] = P[B_i|A_i] \cdot P[A_i] \tag{B-1}$$

In DARWIN, $P[B_i|A_i]$ is computed by virtually placing an initial crack of random size at a specified location and growing the crack to failure. This computation is performed using an automated rectangular fracture mechanics model (plate) generation algorithm [B-8]. The algorithm identifies the orientation of the plate at each location based on the maximum principal stress of the major stress cycle and the local stress gradient. Plate dimensions are based on the intersection of the crack growth plane (coincident with the principal stress plane) and the component geometry as well as the crack size at fracture. Additional details regarding the algorithm are provided in reference [B-8]. Crack growth life predictions include additional random variables such as scatter in the applied stress and material property values, among others.

$P[A_i]$ is based on a material and process-dependent anomaly occurrence rate (i.e., number of anomalies per unit volume) and the volume of a zone. The zone volume is equal to the sum of the volumes of the individual FEs that are assigned to a given zone.

DARWIN includes an automated risk assessment approach in which one element equals one zone. This is referred to as the “exhaustive autozoning” approach. Each zone is assigned a crack that is located on one of its nodes based on the model geometry. Consequently, almost all nodes in the model have a crack attached to them. A fracture mechanics plate is automatically generated at each node using the algorithm described in reference [B-8], and the fracture risk is computed at each node. The conditional risk throughout each zone is set to the value at the node to obtain a conservative estimate of $P[B_i|A_i]$. Because of the high level of discretization, results using exhaustive autozoning are often considered a lower-limit estimate of the component fracture risk.

Once the risk values have been computed for all zones, the results are used to construct a contour of risk values that is superimposed directly on the FE geometry. The component fracture risk $P\left[F_{\text{component}}\right]$ is the probability union of the individual zone risks [B-5]:

$$P\left[F_{\text{component}}\right] = P\left[F_1 \cup F_2 \cup \cdots \cup F_n\right] = 1 - \prod_{i=1}^{n}\left(1 - P\left[F_i\right]\right) \qquad \text{(B-2)}$$

where *n* is the number of zones in the risk model. Because the number of zones in the risk model equals the number of FEs in the FEM, equation (B-2) can be expressed as:

$$P\left[F_{\text{component}}\right] = 1 - \prod_{i=1}^{N}\left(1 - P\left[F_i\right]\right) \qquad \text{(B-3)}$$

where *N* is the number of elements in the FEM.

B.2.1 OPTIMAL AUTOZONING

An optimal autozoning algorithm that identifies the number and placement of zones for fracture risk assessments associated with 2-D FEMs was developed previously [B-7]. To identify zones, the algorithm first identifies a risk surface based on approximate risk estimates at every node in the model. A zone is assigned to each FE, and the node with the highest risk within each FE is referred to as the risk-limiting location for the associated zone. This risk surface is then provided as input to the optimal autozoning algorithm. It compares the risk in each zone with each of its neighbors. The algorithm computes the change in unconditional risk associated with merging various pairs of contiguous zones and identifies the pair that produces the smallest change in unconditional risk. The algorithm then merges these two zones together and re-evaluates the model for the next pair of zones. The algorithm continues until all FEs have been merged into a single zone. If the algorithm is then reversed (starting with the final single zone), it provides an optimal step-by-step discretization of the model into zones. The algorithm identifies the optimal placement of FEs within each zone for a specified number of zones. As the number of zones increases, the expected unconditional risk for the component decreases, but the computation time increases. The algorithm converges to a final zone breakup upon satisfying user-defined convergence criteria. Further details regarding the algorithm are provided in reference [B-7].

The approach discussed in the preceding paragraph is referred to as an “optimal autozoning” algorithm. The entire method is named after the key algorithm that defines the discretization of the model into zones. This autozoning approach is significantly faster that the exhaustive autozoning approach. Computation times for models containing several thousand elements are on the order of several minutes for optimal 2-D autozoning rather than several hours or days for exhaustive autozoning.

Computation times associated with the 2-D optimal autozoning algorithm are directly proportional to the number of elements in an FEM. Fracture plates must be determined for every node in the model, despite only a few nodes being necessary for the final risk calculations. The computational costs of this algorithm increase exponentially with the number of FEs.

Three-dimensional FEMs typically contain hundreds of thousands to millions of elements, or approximately two to three orders of magnitude multiplied by the number of elements in 2-D

FEMs. Because computation times associated with the 2-D optimal autozoning algorithm scale exponentially with the number of elements, it is not feasible for application to autozoning of 3-D FEMs.

B.2.2 PRE-ZONING

An improved optimal autozoning methodology has been developed for 3-D FEMs based on an initial partitioning of the mesh into “pre-zones”. Pre-zones are groups of FEs that are expected to have similar conditional risk values based on similar characteristics such as stress, temperature, and crack type, among others.

Pre-zoning is an extension of the optimal autozoning algorithm. Consider an FEM with E elements. Let the optimal autozoning algorithm merge together a portion of the elements (E’ elements) into zones. This operation yields E–E’ zones. Consider the same FEM divided into E–E’ pre-zones. The optimal autozoning algorithm and the pre-zoning approach will produce identical risk results if they discretize the model into exactly the same E–E’ pre-zones and use the same risk-limiting locations.

In the best case scenario, the zones identified by pre-zoning would be identical to the zones identified by the optimal autozoning algorithm. Pre-zoning requires fewer risk evaluations and fewer zone-merging operations iterations. These features lead to an efficient and robust algorithm. In general, pre-zoning will not generate the same boundaries as optimal autozoning, which will lead to more conservative risk estimates than the optimal autozoning algorithm applied at all FEs.

Pre-zoning may also identify slightly different risk-limiting locations than those obtained via optimal autozoning. This may lead to a non-conservative result if the actual risk-limiting location differs from the one identified by the algorithm. To address this issue, DARWIN sets very tight tolerances when searching for the risk-limiting location and provides graphical information to users enabling them to review the risk-limiting locations identified by the algorithm.

The initial pre-zone boundaries are set based on engineering judgment regarding crack-growth lives associated with anomalies placed at various locations within a component. For example, cracks that initiate in high-stress regions grow more quickly than cracks located in low-stress regions, all other things being equal. Stress is one of several drivers of crack growth, and DARWIN features several drivers: maximum stress at loadsteps/cycles, maximum temperature at loadsteps/cycles, and distance of the FE centroid to the nearest surface.

After defining the drivers, bins are identified that represent various ranges of values and are assigned to each driver. For example, one of the stress driver bins may consider all contiguous FEs with maximum stress levels between 0 and 50 ksi. DARWIN uses a Cartesian or cylindrical grid to define the physical locations within the FEM associated with the bins for each driver. By default, DARWIN sets the number of bins based on the maximum number of zones specified by the user. Figure B-1 shows example pre-zones formed using the individual drivers for stress, temperature, and distance-to-surface. These drivers define independent and preliminary pre-zone boundaries.

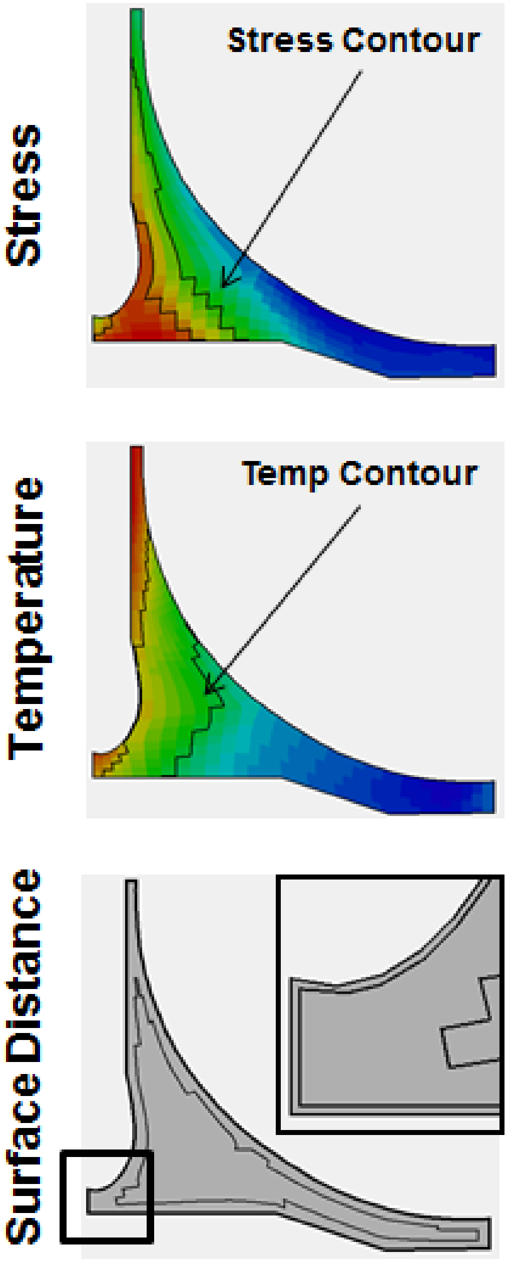


Figure B-1. Contours of stress, temperature, and distance-to-surface values that can be used to define independent preliminary pre-zone boundaries

DARWIN combines the preliminary boundaries to form the final pre-zone boundaries (see figure B-2). Within a single pre-zone, all elements have the same set of driver bins: the same stress levels, the same temperature levels, and the same distance-to-surface. However, the pre-zone boundaries are not the boundaries that appear by simply overlaying the initial boundaries from each individual pre-zone. Elements within a pre-zone are contiguous (i.e., there are no isolated regions within the same pre-zone). Isolated regions are divided into separate pre-zones. Furthermore, all elements within the same pre-zone have the same properties (e.g., crack growth rates). These restrictions further divide the model into pre-zones. Finally, the user may request more zones than there are pre-zones in the model based on the initial pre-zone bins. For this case, DARWIN subdivides the largest pre-zones into small pre-zones until it reaches the user-defined limit.

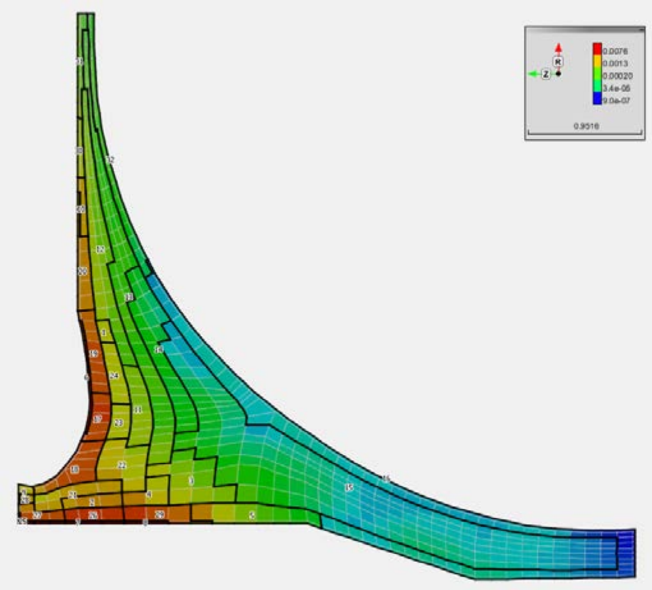


Figure B-2. Final pre-zone boundaries based on combining the preliminary boundaries associated with stress, temperature, and distance-to-surface

The crack location must be set at the location of maximum risk (risk-limiting location) within each pre-zone. To identify the risk-limiting location, a GP model [B-9, B-10] fits an approximate risk surface based on estimated risk values at a small number of locations within each pre-zone. These locations are referred to as training points. The GP approach is well suited for modeling data from deterministic computer simulations because it can create a function that directly interpolates all of the training data. As it is a non-parametric approach, it also has the advantage that it can capture complex nonlinear relationships without needing a priori information regarding functional form.

Given a set of training data (here risk estimates at discrete locations), the GP model prediction at an input location **x** is given by:

$$\hat{G}(\mathbf{x}) = \mathbf{f}^T(\mathbf{x})\boldsymbol{\beta} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{Y} - \mathbf{F}\boldsymbol{\beta}) \tag{B-4}$$

where **Y** is a vector containing the observed outputs from the *n* training points, **F** is an *m* x *q* matrix with rows $\mathbf{f}^T(\mathbf{x})$ (the trend basis functions at each of the training points), **r** is the vector of correlations between **x** and each of the training point locations, **R** is the *n* x *n* matrix of correlations among the training points, and **β** is a vector of trend function coefficients. Typically, the trend function (defined by the first term) is used to capture large-scale variations, whereas the correlation portion of the model (represented by the second term) is used to capture small-scale fluctuations and other, more complex behavior. A linear trend function was used for this effort. Because risk values may vary by a magnitude or more within a very small distance, the GP model used in this work is defined using the natural log of risk values.

The behavior of the model depends on the type of correlation function that is used to describe the degree to which outputs at different input locations are related to each other. The separable, squared-exponential formulation was used, which produces a smooth response function. With this formulation, the correlation, *c*, between any two input locations, **a** and **b,** is computed as:

$$c(\mathbf{a},\mathbf{b}) = \exp\left[-\sum_{i=1}^{d}\xi_i\left(a_i - b_i\right)^2\right] \qquad \text{(B-5)}$$

where *d* is the dimensionality of the input, and the ξ_i are non-negative correlation parameters that must be estimated from the training data.

Latin hypercube sampling is used to set the initial training point locations within each pre-zone. The number of training points is based on the dimensionality of the model, with 10 initial training points in each pre-zone for 2-D geometries and 20 initial training points in each pre-zone for 3-D geometries. Once it has been determined that a sampled training point is within the pre-zone boundaries, it is virtually placed at the nearest FE node.

Probabilistic fracture and fatigue routines are used to compute an approximate risk estimate at each training point location that is based only on the anomaly size random variable. This risk estimate is of lower quality than the final risk estimate that considers all random variables, but it can be computed using significantly fewer computational resources. It is the same initial risk estimate used in the optimal autozoning methodology. These risk estimates are used to define the approximate risk surface.

In general, the initial training points will not contain the risk-limiting location for every pre-zone. Consequently, the risk surface is interrogated to identify the location of maximum risk using Efficient Global Optimization (EGO) [B-11]. For a GP model, the improvement, I, at any point, **x,** is a random variable:

$$I = \max\left(f_{\min} - Y, 0\right) \qquad \text{(B-6)}$$

where $-Y$ represents a random variable from the GP model and $-f_{\min}$ is the current maximum value of risk for all training points in the pre-zone (note that EGO is a minimization technique, so negative signs are applied to the variables to obtain a local maximum). The expected improvement at this point is then:

$$E[I(\mathbf{x})] = \left(f_{\min} - \hat{y}\right)\Phi\left(\frac{f_{\min} - \hat{y}}{s}\right) + s\phi\left(\frac{f_{\min} - \hat{y}}{s}\right) \qquad \text{(B-7)}$$

This expression uses the mean ($\hat{y}$) and standard deviation (*s*) from the GP at the location **x**. The functions ϕ and Φ indicate the standard normal density and standard normal distribution functions, respectively.

Within each pre-zone, DARWIN computes the expected improvement for each node that is not already a training point. Another training point is added when the maximum value of the expected improvement is above a predefined tolerance (based on the number of samples in the approximate risk estimate). Similarly, training points may be added if the standard deviation of the GP model is large (again based on a predefined tolerance). The algorithm then estimates risk at any new training points. The algorithm iterates on every pre-zone until the training point satisfies the predefined tolerances or the maximum number of training points for the pre-zone is reached. The predefined tolerance for convergence requires that the expected improvement is less than the minimum discretization of the conditional risk estimate (i.e., one divided by the number of Monte Carlo samples). The node with the maximum conditional risk represents the risk-limiting location for the pre-zone. After several iterations, this approach yields an approximate risk-limiting location in every pre-zone (see figure B-3). In general, the number of nodes needed to locate the risk-limiting location using EGO is significantly lower than the total number of nodes in the pre-zone.

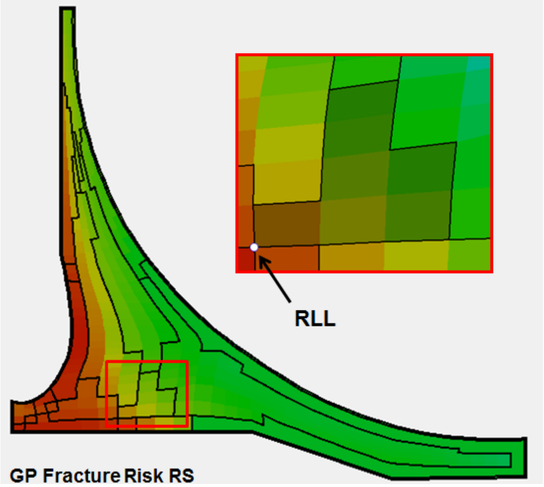


Figure B-3. Example of a risk-limiting location that has been identified using a GP model fit to the conditional fracture risk contour within each pre-zone

Risk values vary location-by-location within a pre-zone and are not known a priori. For example, a group of surface elements might have nearly constant risk values except near a sharp corner that increases risk locally. Lower unconditional risk values would be realized by separating the corner region (high conditional risk over a small volume) from the surface (low conditional risk over a large volume). That is, more accurate pre-zoning leads to less conservative values of risk.

To reduce over-conservatism, DARWIN adopts an iterative approach to further refine pre-zones based on the local risk estimates (from the GP models) within the pre-zone. Based on the risk estimate (provided by the response surface), the algorithm assigns an approximate risk value to every element in the pre-zone. The optimal autozoning algorithm is then invoked to operate over elements in every pre-zone rather than over the entire model. The algorithm subdivides the pre-zone when subdivision would reduce unconditional risk beyond a user-defined threshold. The optimal autozoning algorithm may subdivide any pre-zone into five pre-zones per iteration. Between iterations, new risk-limiting locations are computed for each pre-zone. DARWIN may apply this discretization scheme to the same region recursively. For example, a particularly badly behaved region might be subdivided first from one pre-zone to five pre-zones, and then from five pre-zones to 25 pre-zones. Each subdivision further reduces the unconditional risk value.

Pre-zone refinement triggers the main pre-zoning algorithm to re-initiate on the new set of pre-zones. Pre-zone refinement leads to a redistribution of training points among the new and modified pre-zones. If needed, it samples additional training points to form an approximate risk surface. The pre-zoning algorithm then continues until it satisfies the convergence criteria. Preliminary studies suggest that pre-zone refinement leads to more reliable disk risk values that offset its computational costs.

The result of the pre-zoning strategy is a model broken up into pre-zones based on engineering judgment and risk surfaces. Every pre-zone has a risk-limiting location that represents the maximum risk values for that location. Pre-zones then act as input for the optimal autozoning algorithm. The conditional risk at the approximate risk-limiting location characterizes the entire pre-zone. As pre-zones consist of groups of FEs, the number of pre-zones will typically be significantly lower than the total number of elements in the FEM, thereby reducing computational costs for zone definition. For example, an FEM containing millions of elements could be represented using only hundreds of pre-zones. DARWIN provides visualization of color contours within the boundaries of individual zones to display the pre-zones that are combined to form each zone (see figure B-4).

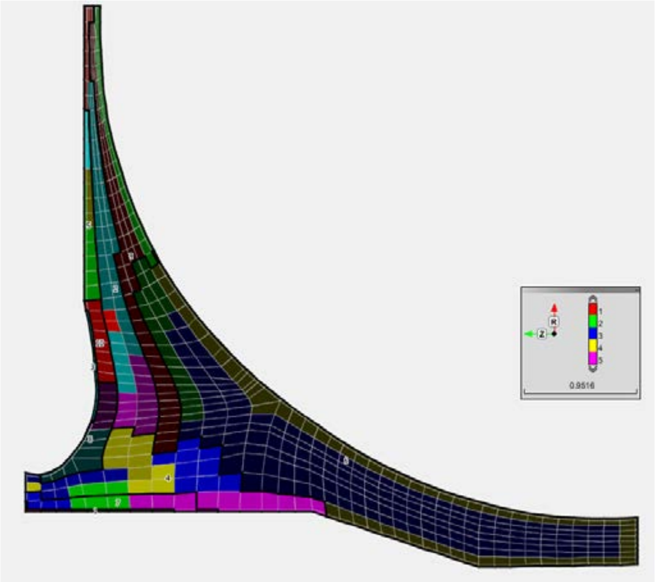


Figure B-4. DARWIN visualization of color contours within the boundaries of individual zones to display the pre-zones that are combined to form each zone

The main limitation of the proposed approach is that the risk-limiting location must be determined within each pre-zone. The pre-zone geometry impacts the number of iterations needed to acquire the risk-limiting location. Slender pre-zones converge less rapidly than bulky pre-zones, particularly on the surface. The EGO algorithm may not converge to the risk-limiting location for all pre-zone shapes. DARWIN provides a warning in these situations and advises that the user re-run the analysis after modifying the pre-zone inputs.

This framework retains several features from the original optimal autozoning methodology: computing a risk surface, determining the risk-limiting location, and executing the optimal autozoning algorithm. The methodology adopted here scales well for increasingly large FE geometries. The major computation cost is computational of risk estimates at training points. Though the algorithm was developed to provide treatment for the large number of elements associated with 3-D FEMs, it can also be applied to improve the computational efficiency of the risk assessment for 2-D FEMs.

B.3 NUMERICAL CHARACTERISTICS OF THE PRE-ZONING ALGORITHM

A case study was performed to illustrate the effectiveness of the new pre-zoning allocation algorithm. It is based on a FEM with simple geometric features: a square axisymmetric disk with linearly varying stresses (see figure B-5(a)) and radially varying temperatures (see figure B-5(b)). The number of elements in this model is allowed to vary to demonstrate the exhaustive, optimal, and pre-zoning methodologies. In this example, optimal autozoning converges once the disk risk changes by no more than 1%. The new pre-zoning methodology targets 50 maximum zones and the same convergence threshold. Exhaustive autozoning sets one zone per element, and may be viewed as an upper bound on the number of zones and an effective lower bound on disk risk.

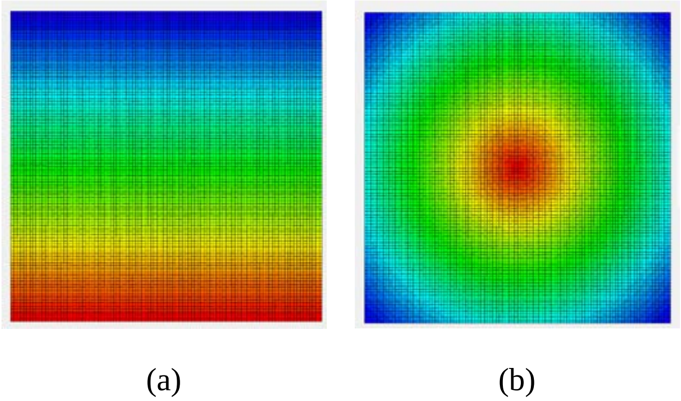


Figure B-5. Variation of (a) stress and (b) temperature in a square axisymmetric component

Figure B-6 shows the performance of the new pre-zoning approach relative to the exhaustive autozoning and the previous optimal autozoning methods. Lines represent best fits to computation times from 100–10,000 elements that are shown as discrete points in figure B-6. At 10,000 elements, optimal autozoning is approximately 10 times faster than exhaustive autozoning, whereas pre-zoning is approximately 100 times faster. For large FEMs (a million or more elements), trends indicate that the computation time associated with the new pre-zoning algorithm is one to two orders of magnitude lower than the exhaustive algorithm, and three to four orders of magnitude lower than the exhaustive algorithm. Note that the trends shown in figure B-6 may underestimate the speed-up associated with pre-zoning. For example, the computation time associated with pre-zoning is very similar for models with 10,000 and 100,000 elements. This suggests that the computational cost associated with pre-zoning may reach a plateau for large models.

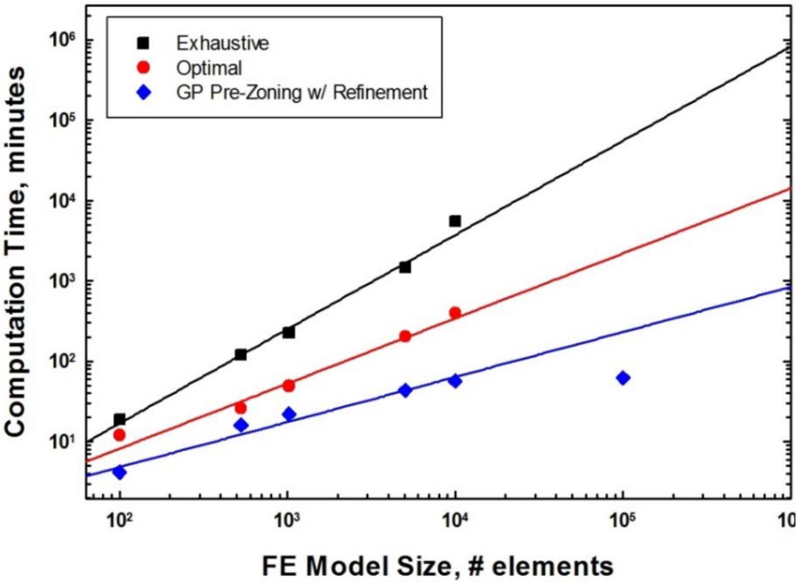


Figure B-6. Computation time as a function of FEM size for three autozoning methods

As the number of FEs increases, the risk associated with each of the methods decreases. However, the risk appears to converge towards a minimum value as the number of elements is increased. Figure B-7 shows the variation of disk risk with model size for the three methods. Trends represent a three-term decaying exponential function. Each method correctly predicts lower risk as the model size increases. However, the more efficient methods suffer from a risk penalty (i.e., over prediction of risk) relative to exhaustive autozoning. The risk penalty varies with the number of elements. Increased mesh refinement leads to an increased risk penalty. At a million elements, the new pre-zoning method predicts approximately 50% higher risk than the risk predicted by exhaustive autozoning. As a reference, manual zoning often leads to disk risk values that vary by orders of magnitude even among experienced users. In contrast, the risk penalty associated with pre-zoning is constant and user independent.

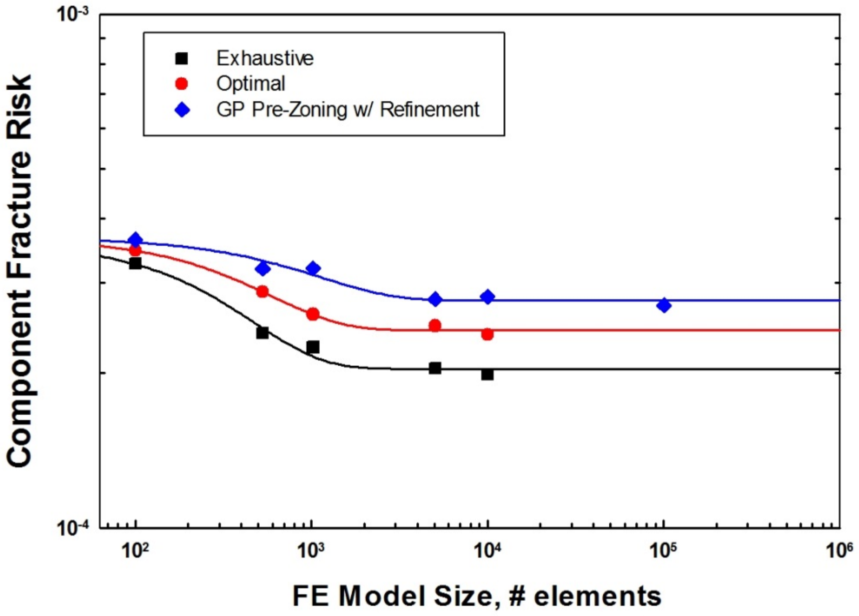


Figure B-7. Disk risk as a function of model size for three autozoning methods

B.4 APPLICATION TO 3-D GAS TURBINE ENGINE DISK

The pre-zoning methodology was applied to an actual engine disk hub model to demonstrate the use of the algorithm for a large 3-D FEM. Figure B-8 shows principal stresses on the 3-D disk hub model with 54,660 elements and 63,540 nodes. The size and complexity of this model significantly exceeds the largest problem executed in the previous examples. As a result, available servers did not have sufficient memory to perform exhaustive autozoning or optimal autozoning on this problem. The pre-zoning methodology enabled this analysis to complete in approximately 2.5 hours (wall-clock) on a desktop machine using eight cores of an Intel Xeon CPU E5-2630 @ 2.30 GHz. In contrast with the previous example, a grid was used to perform the initial pre-zone breakup of the model. The pre-zoning algorithm targeted 100 maximum zones and converged using the default settings in DARWIN. The model has two distinct regions with similar properties: all surface elements and all interior elements.

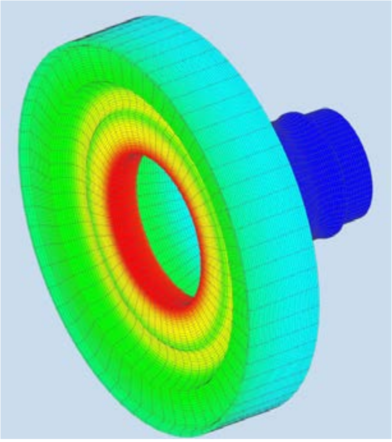


Figure 8. Example 3-D gas turbine engine disk FEM

Figure B-9 shows the final pre-zone boundaries of the model. The grid method leads to an initial pre-zone break-up on the order of 100 zones. However, the pre-zone refinement stage iteratively discretizes these pre-zones to reduce disk risk. The final pre-zone break-up features approximately 800 pre-zones. They are concentrated near the bore of the model where high stresses drive crack growth more rapidly than in other locations of the component.

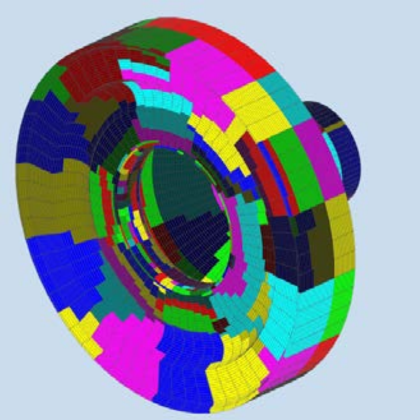


Figure B-9. Final pre-zone boundaries following pre-zone refinement

The approximate risk contour shown in figure B-10 from the pre-zoning analysis suggests higher conditional risk near the center of the bore. This figure displays the GP response surface of approximate conditional risk for each pre-zone. High pre-zone refinement leads to a continuous risk surface for the entire component. DARWIN features a similar contour of the response surface uncertainty, which is quite low for this geometry.

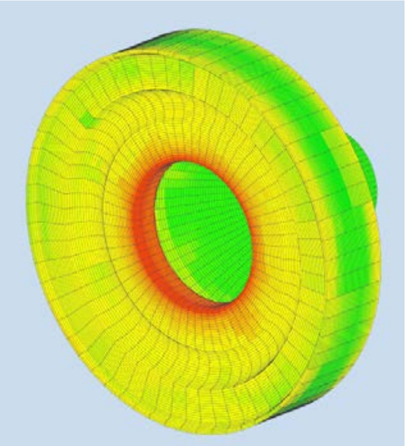


Figure B-10. Approximate conditional risk surface based on the GP response surfaces

Figure B-11 presents the zone with the highest contribution to unconditional risk in the part. This figure shows the internal pre-zones that compose this zone without the surrounding material from it. It is a thin, circular region right on the bore of the model. This zone is largely axisymmetric, even though the initial zone break-up was based on a rectangular grid. High stresses lead to largely uniform and very high conditional risks in this region. This zone has the risk-limiting location on the surface. This sliver of material contributes 13.2% of the final risk to the component.

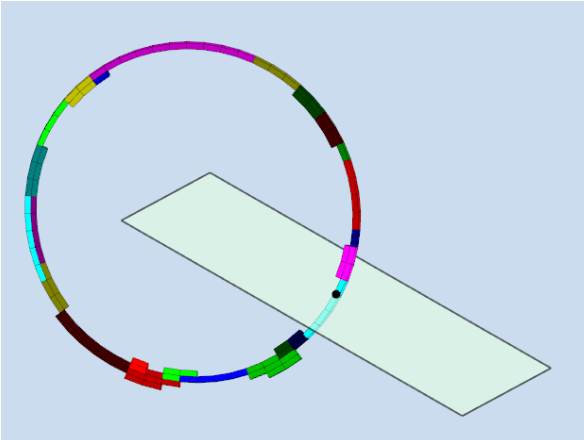


Figure B-11. Zone with highest unconditional risk elements with similar colors were grouped into the same pre-zone; the slice plane and risk-limiting location is also shown

Figure B-12 shows disk risk convergence of the optimal autozoning algorithm as the number of zones increases. Here, zones represent combinations of pre-zones (see figure B-11), which are composed of individual elements. Increasing the number of zones decreases the disk risk. DARWIN predicts the highest disk risk value for the minimum number of zones and predicts lower disk risk values as the number of zones is increased. Optimal autozoning with 33 zones decreases unconditional risk by more than 97% of the initial estimate, based on two zones composed of all surface elements and all non-surface elements. These zones reflect the initial property regions.

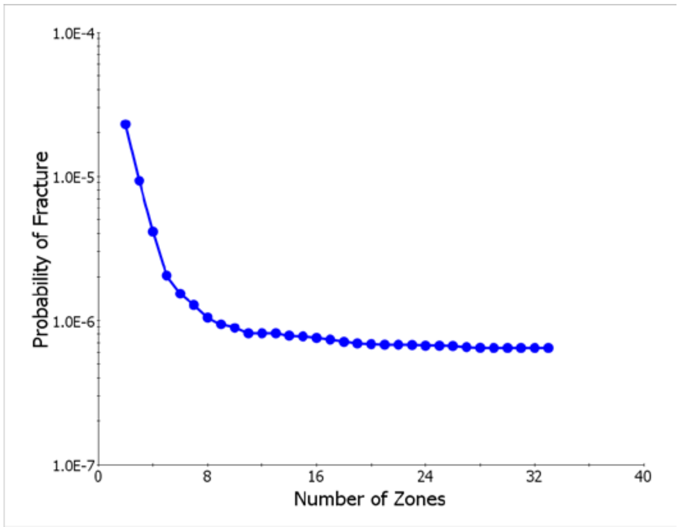


Figure B-12. The optimal autozoning algorithm using pre-zones converges to a near-constant fracture risk as the number of zones increases

Figure B-13 presents deterministic life values computed at the risk-limiting location of each zone. In this figure, all elements in a zone are assumed to have the same life, leading to a non-contiguous variation. In this figure, dark blue indicates that the life equals or exceeds the service life. Results in this figure assume an initial semi-elliptical anomaly that had a nominal size of 10 mils and 1:1 aspect ratio. This indicates that most nominally sized anomalies do not grow to fracture. The life limiting locations appear as local hotspots along the interior of the bore.

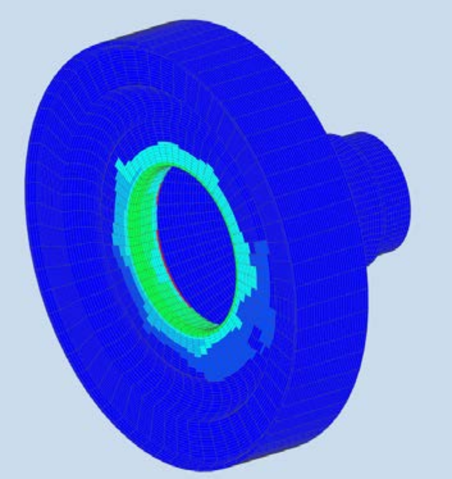


Figure B-13. Life results at risk-limiting locations in each zone

Figure B-14 provides conditional risk values computed at the risk-limiting location of each zone. The conditional risk surface in figure B-14 shows similar hotspots as the approximate risk surface in figure B-10, but without the continuous variation provided by the response surface. Several zones have risk-limiting locations that do not contribute to the actual, as opposed to approximate, conditional risk (i.e., anomalies at these locations do not fracture over the service life). Sampling additional disks during the accurate disk risk calculation may increase the conditional risk in these regions.

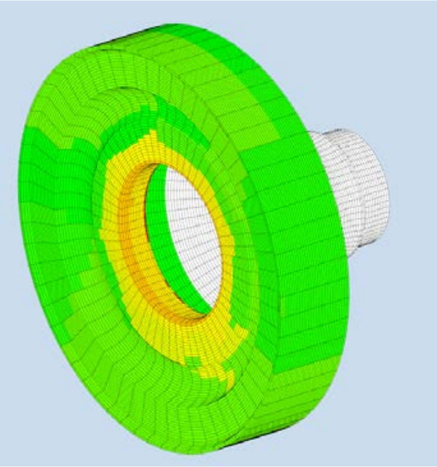


Figure B-14. Conditional risk results at risk-limiting locations in each zone

Figure B-15 incorporates the volume effect to provide the final unconditional risk values in the component. This figure presents relatively constant contours in each zone. This coloring scheme reflects zones with nearly equivalent unconditional risk values, even though individual zones may extend over significantly more material.

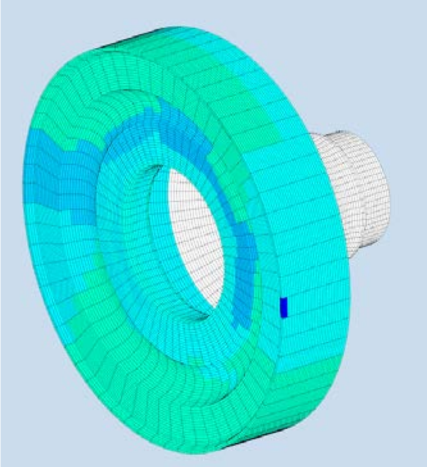


Figure B-15. Unconditional risk results at risk-limiting locations in each zone

B.5 CONCLUDING REMARKS

Zone-based risk assessment of 3-D FEMs provides unconditional risk estimates that represent both the probability of fracture at a location and the probability of having an initial starter crack. However, zone-based estimates using individual FE-based automated zoning algorithms may not be computability tractable (e.g., memory requirements may exceed available resources) or may require any excessively long computation time (days, weeks, or months). An improved autozoning methodology was presented that reduces the computation time by three to four orders of magnitude for large FEMs. It predicts risk values that are conservative compared to methods that require risk assessment at all FEs within an FEM. This new autozoning method combines FEs with similar fracture risk drivers into pre-zones and then determines the optimal zone breakup based on pre-zones rather than individual FEs. The resulting algorithm provides a feasible solution method for fracture risk assessments of realistic components, as demonstrated via application to a realistic 3-D gas turbine engine disk.

B.6 NOMENCLATURE

EGO	=	Efficient Global Optimization
$\mathbf{F}$	=	Matrix with rows $\mathbf{f}^{T}(\mathbf{x})$
$\mathbf{f}^{T}(\mathbf{x})$	=	Trend basis functions at training points
f_{min}	=	Current minimum value of risk for all training points
GP	=	Gaussian process
$\hat{G}(x)$	=	GP response surface
N	=	Number of elements in the FE model
$\mathrm{P}[\mathrm{A_i}]$	=	Probability of having an anomaly in the zone *i*
$\mathrm{P}[\mathrm{B_i} \mid \mathrm{A_i}]$	=	Probability of fracture assuming the presence of an anomaly in the zone *i*
$\mathrm{P}[\mathrm{F_i}]$	=	Fracture risk of zone *i*
$\mathrm{P}[\mathrm{F_{component}}]$	=	Fracture risk of a component
$\mathbf{r}$	=	Vector of correlations between input location and training points
$\mathbf{R}$	=	Matrix of correlations among the training points
S	=	Standard deviation of GP model at $\mathbf{x}$
$\mathbf{x}$	=	Input location for GP model
$\mathbf{Y}$	=	Vector of observed outputs from training points
Y	=	Random variable from GP model
$\hat{y}$	=	Mean value of GP model at $\mathbf{x}$
$\boldsymbol{\beta}$	=	Vector of trend function coefficients
ϕ	=	Standard normal density function
Φ	=	Standard normal distribution function

B.7 REFERENCES

B-1. Dillström, P. (2000). ProSINTAP – A probabilistic program implementing the SINTAP assessment procedure, *Engineering Fracture Mechanics*, *67*, 647–668.

B-2. Cavallini, G., & Lazzeri, R. (2007). A probabilistic approach to fatigue risk assessment in aerospace components, *Engineering Fracture Mechanics*, *74*, 2964–2970.

B-3. Ocampo, J, Millwater, H., Singh, G., Smith, H., Abali, F., Nuss, M., Reyer, M., & Shiao, M. (2011). Development of a probabilistic linear damage methodology for small aircraft, *Journal of Aircraft*, *48*, 2090–2106.

B-4. Beretta, S., Foletti, S., Madia, M., & Cavalleri, E. (2015). Structural integrity assessment of turbine discs in presence of potential defects: Probabilistic analysis and implementation, *Fatigue & Fracture of Engineering Materials & Structures, 38*, 1042–1055.

B-5. Millwater, H. R., Enright, M. P., & Fitch, S. K. (2007). A convergent zone-refinement method for risk assessment of gas turbine disks subject to low-frequency metallurgical defects, *Journal of Engineering for Gas Turbines and Power, 129*, 827–835.

B-6. Millwater, H. R., Enright, M. P., & Fitch, S. K. (2002). *A convergent probabilistic technique for risk assessment of gas turbine disks subject to metallurgical defects*, Paper presented at the 43rd Structures, Structural Dynamics, and Materials Conference, , Denver, Colorado. (Paper 2002-1382).

B-7. Moody, J. P., Enright, M. P., & Liang, W. (2013). *An optimal automated zone allocation approach for risk assessment of gas turbine engine components*, Paper presented at the 58th ASME International Gas Turbine & Aeroengine Technical Congress, San Antonio, Texas. (GT2012-69968).

B-8. McClung, R.C., Lee, Y-D., Liang, W., Enright, M.P., and Fitch, S. (2010). *Automated fatigue crack growth analysis of components*, Paper presented at the 10th International Fatigue Congress (Fatigue 2010), Prague, Czech Republic.

B-9. Rasmussen, C. E., Williams, C. K. I. (2006). *Gaussian processes for machine learning*, MIT Press, Cambridge, Massachusetts

B-10. Diggle, P.J, (2010). *Model-based geostatistics*, Springer, New York.

B-11. Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions, *Journal of Global Optimization, 13*, 455–492.

APPENDIX C—HSIESTA: HDF5 SIESTA FILE FORMAT

C.1 INTRODUCTION

HSIESTA is a binary data file that is used to communicate output data from a structural/thermal analysis. This data file represents a neutral interface between finite element analysis codes (e.g., ABAQUS) and DARWIN® (Design Assessment of Reliability With INspection). This data file employs the Hierarchical Data format (HDF5) file format for data storage. An Application Programming Interface (API) has been created to write and read these data files. Optimally, all data should be read/written using the API program. HSIESTA has been built on top of the legacy System for Integrated Engineering Structural and Thermal Analysis (SIESTA) data structure that has been used for many years in the DARWIN program. Consequently, conventions used in the SIESTA format have been followed (e.g., ordering of element connectivity).

C.2 CONVENTIONS

- HSIESTA shall be defined as exactly as possible for as broad of a use case as possible. It is designed (like the original SIESTA) to be a general purpose FEA container.
- HSIESTA has been defined to handle models with high levels of complexity.
- Attributes are indicated by the "at" sign @, e.g., *@myAttribute*.
- Groups are indicated by a forward slash /, e.g., /myGroup.
- Tables and datasets are indicated by a plus sign +, e.g., **+myTable**.
- The names of attributes, groups, and tables shall use camelCase. For example, the following names are acceptable names: myAttribute, myGroup, and myTable.
- The names of string identifiers shall be UPPER_CASE. For example, the following strings are acceptable string identifiers: "MY_FLAG", "USE_UPPER", and "DO_THIS_NOT_THIS".
- There are special logical flags. One group shall be "TRUE", "YES", and "ON". Another group of these flags shall be "FALSE", "NO", and "OFF".
- HSIESTA shall assume "0" based indexing in all tables. That is, the first row/column shall always start at 0. This row/column shall be followed by 1, 2, 3, …
- Any group listed as "group-X" have the "X" replaced the appropriate integer value in the HDF5 file. For example, suppose there are three groups with header "header-X". They shall be listed in HSIESTA as "header-0", "header-1", and "header-2". Indexing with groups shall begin at zero to be consistent with the overall HSIESTA convention.
- If an attribute, group name, or table is limited to a few possible values, then the possible values shall be enclosed in curly brackets ({ }) and separated by a line (|). For example, *@defaultGeometricUnits*={"INCHES" | "MILIMETERS"}.
- If an attribute, group name, or table may use all non-keyword strings, then the possible values shall be enclosed in square brackets ([ ]). For example, /[componentName].
- If group has a user specified name, then the group shall be given as [description] in path file any references to this group. The description shall indicate the content of this group, though not the actual name which will be variable.

- Comments may be added to any table or group. Comments are attributes that have the form "$-X" in a group or table. A group or table may have any number of comments. Comments are stored as fixed length strings.
- HSIESTA files shall have the file extension *.fea.

C.3 TOP LEVEL GROUPS

HSIESTA files are indicated by the top level group:

/HSIESTA

This group shall have the following required attributes:

> *@dateCreated*="YYYY:MM:DD:HH:MM:SS" which indicates the date on which the file was created using the four digit year : two digit month : two digit date : two digit hour (24-hour format) : two digit minute : two digit second.
>
> *@version*="1.0" is the HSIESTA version number.

HSIESTA files shall have four groups underneath of /HSIESTA:

1. /processing – for processing related information, such as any applied filters
2. /global – for global data, such as component hierarchy in the model.
3. /geometry – for "geometry" data, such as the initial nodal coordinates and element connectivity.
4. /fields – for "fields" data, such as one thousand loadcases of stress.

Each one of these groups is described in detail.

C.3.1 PROCESSING GROUP

The processing group represents data related to the creation of the file. For example, applied filters are listed here. This group serves as a record of the options used to build the HSIESTA file. The processing group does not contain any information critical to running a DARWIN analysis. It is designed to be read by humans while debugging analysis results.

The processing group shall contain a table **+analysis** that contains information specific to a particular analysis program. This table shall have as many rows as needed. It shall have three columns:

1. "name" – a string indicating the name of the analysis program.
2. "attribute" – a string indicating the attribute to be defined for the analysis program.
3. "value" – a string indicating the value of the attribute.

For DARWIN analyses, the analysis group shall contain (at a minimum) the following:

1. Lowest version of DARWIN that can read the analysis file. For HSIESTA version 1.0, this version is 9.0. The analysis row shall be given as: [name:DARWIN, attribute:VERSION, value:9.0].

The HSIESTA file may also contain the following rows as well. This information is not needed by a DARWIN analysis. However, it contains additional processing information.

2. The FEA program that was used to generate the source file. The analysis row shall be given as: [name:FEA, attribute:SOURCE, value:{"ANSYS" | "ABAQUS" | "SIESTA" }].

3. The FEA program version that was used to generate the source file. The analysis row shall be given as: [name:FEA, attribute:VERSION, value:#].

4. The job name of the source file. The analysis row shall be given as [name:FEA, attribute: JOB, value:"JOB NAME"].

5. The stress units of the source file. The analysis row shall be given as [name:FEA, attribute:STRESS,value:{"KSI" | "MPa" }].

6. The length units of the source file. The analysis row shall be given as [name:FEA, attribute:LENGTH,value:{"INCHES" | "MILIMETERS"}].

7. The temperature units of the source file. The analysis row shall be given as [name:FEA, attribute:TEMPERATURE, value: {"FAHRENHEIT" | "CELSIUS"}].

8. Version number of FE2NEU used to generate the file. This analysis row shall be given as [name:FE2NEU, attribute:VERSION, value:#].

9. Indication if **include** filters were used by FE2NEU. The analysis row shall be given as [name:FE2NEU, attribute:INCLUDE, value:{"TRUE" | "FALSE"}].

10. Indication if the **exclude** filters were used by FE2NEU. The analysis row shall be given as [name:FE2NEU, attribute:EXCLUDE, value:{"TRUE" | "FALSE"}].

For analyses built by FE2NEU, the group /processing shall have an additional subgroup: /fe2neu. The group /fe2neu shall have two optional groups:

1. /include – a group with information for the include filters.
2. /exclude – a group with information for the exclude filters.

These groups shall not be present if they are empty.

Both groups have the same possible set of tables:

1. **+materialID –** a table of integer ID's for materials to be included/excluded.

2. **+loadCaseID** – a table of integer ID's for loadcases to be included/excluded.
3. **+elementType** – a table of element type strings to be included/excluded.
4. **+elementID** – a table of integer element ID's to be included/excluded.

The tables **+materialID** and **+loadcase** shall have one column, "ID". These tables shall have as many rows as necessary to define the filter. These tables shall not be present if the filter is not applied.

The table **+elementType** shall have one column, "elementType". This table shall have as many rows as necessary to define the filter. This table shall not be present if the filter is not applied.

The table **+elementID** shall have two columns, "startID" and "endID". This table shall represent a list of elements. This table shall have as many rows as necessary to define the filter. This table shall not be present if the filter is not applied.

C.3.2 GLOBAL GROUP

The global group represents global data that is neither geometric nor fields. For example, the coordinate system will be put here. More importantly, this group contains a series of spatial, temporal, and material relationships from input to the API to the appropriate geometry or field data. For example, if the user requests the stress in a component at time 0.1 seconds, the maps in /global would provide the correct location to the loadcases in the field data. At this time, it is not expected that this framework will be fully utilized.

Within the global group, there shall be four subgroups:

1. /spatial – for a hierarchical ordering of components, etc…
2. /temporal – for temporal ordering of stresses, temperatures, etc…
3. /material – for definition material properties, anomaly identifiers, etc…
4. /relation – for relationships between materials, loads, etc…

C.3.2.1 Spatial Subgroup

The spatial subgroup is used to order the assembly into groups of geometric entities. Sets of geometric entities may be located within a higher order component. This group is located beneath /global and has no limit on the number of subgroups.

The spatial subgroup establishes a hierarchical ordering of the assembly into groupings of geometric entities. This use of the subgroup is similar to the LIST data structure in SIESTA. Finite element codes such as ABAQUS employ a hierarchical design to store multiple parts into a single assembly. For example, a single assembly might have a pin and a sheet that are in contact. It is probable that DARWIN will be asked to store individual parts at some point in the future. For example, DARWIN might be asked to store geometries for a large component and microstructural information. As a result, it was determined to make the specification sufficiently flexible to support this approach.

“Pin-2” are acceptable.

All spatial subgroups are classified as either a “BRANCH” or a “LEAF”. A branch connects one or more leaves to the main assembly – the root of the model. A branch does not contain any geometry, fields, etc., of its own. Instead, it holds together in logical, hierarchical ordering leaves. The leaves contain sets, geometry, fields, etc...

The spatial subgroup shall have at least two groups. The first subgroup is /assembly and shall be the root of any models in the database. The name of the head group (/assembly) may not be changed. The group/assembly is the parent to all components, etc., that will follow. Any additional subgroups must be able to trace their lineage to /assembly.

The name of the component for DARWIN analysis (i.e., the one viewable by the GUI) shall be given a default value by the API, i.e., “darwinComponent”. Other subgroups in assembly have a group name associated with their component (e.g., a component named “myComponent” has group name “myComponent” in the /spatial group).

All groups/[component] have at least two attributes:

> *@parent* = [“NONE” | “someOtherGroupName”]. The token “NONE” is reserved for the group /assembly.
>
> @kind = { “BRANCH” | “LEAF” }. This attribute denotes if the group is a leaf or a branch.

All groups /[component] that are branches have the table:

> **+children** = a table with one column (“name”) and as many rows as there are children. This table may not be empty – otherwise the group is an empty branch and should be pruned.

All groups /[component] that are leaves have additional attributes:

@dimension={2|3}. This attribute's value is a 32-bit integer. It indicates the dimensionality of the component's geometry.

@sectorModel={TRUE | FALSE}. This string's value indicates if the geometry is a 3D sector model or not. It is always FALSE for a 2D model.

+sets = a table with one column ("name") and as many rows as there are sets. If the component has no sets, then this table shall have a single entry: "NONE".

/mySetName = group with a set name that has been given in the table **+sets**. There must be one group for each set.

Other geometric entities than nodes and elements are possible. Possible geometric entities include nodes, elements, faces, edges, integration points, etc… For now, only nodes and elements are supported.

A branch group must not have entities attached to it. In this case, the group joins multiple components together using the parent/child hierarchy. The branch group is not be accessible in the fields/geometry section of the model. Elements and nodes are defined in the geometry section.

A leaf group must have nodes and/or elements attached to it. It must contain either nodes and/or elements. It must be accessible in the geometry group.

The ordering of components is intentionally flat to facilitate access of individual sets during API use. That is, if the user inputs component "darwinComponent" to extract nodes, then the API would not need to scan down groups of components to reach the component.

Sets are internal groupings of elements within a component. Sets are located within a single component. Sets may not span multiple components. For example, a set of elements might represent all elements on the surface of a component, or a set may contain the first ten nodes in a component. Sets are stored in components in the /[mySetName] group. This set has the following attributes:

@hasNodes = {"TRUE" | "FALSE"}. Indicates if nodes are attached to this set.

@hasElements = {"TRUE" | "FALSE"}. Indicates if elements are attached to this set.

Other geometric entities than nodes and elements are possible but not supported at this time.

The set group has one table for every geometric entity included in the set. Currently, these sets are limited to nodes and elements. Consequently, the available tables are **+nodes** and **+elements**. This table contains all entities of the same type attached to the set. Entities are list in compressed form by id, not by index. This table has two columns: "start" for the starting id and "end" for the ending id. As many rows may be added as are needed to define the set.

C.3.2.2 Temporal Subgroup

The /temporal subgroup describes cases in the fields group and links cases through time. For example, quasi-static stress histories are normally defined with respect to some psuedo-time value. The correct loadcase needs to be accessible from the description of the loading and with respect to the psuedo-time. In this initial implementation, we do not consider the psuedo-time parameter in the /temporal subgroup. Instead, we only require that the description of the loadcases be listed.

Descriptions of the loadcases are listed by component. There shall be a table **+description** located at:

/global/temporal/[component]

The table **+description** shall contain the following columns:

1. summary – a string with the description of the loadstep

There shall be one row for every loadcase listed in fields for this component.

C.3.2.3 Material Subgroup

The /material subgroup defines materials. For example, a material for average grain size might be defined to have some processing parameters, some material properties, and have a nodal field with grain size information.

Materials are stored as groups, with one group per material. The name of the group provides the name of the material. Material names must be unique.

Every material has an associated type that is listed in the attribute *@type*. At this time, the HSIESTA standard only supports the type “UNKNOWN”. No additional material properties are needed to define this type.

Materials have regions assigned to them. Regions are defined in the table **+region** (located at /global/material/[materialName]). This table shall have the following columns:

1. component – a string with the bottom (or base) level component.
2. set – a string with the set over which the material is defined.

There shall be one row for every material region. All sets referenced must contain elements. It is assumed that materials apply to elements.

C.3.2.4 Relation Subgroup

The /relation subgroup defines relations between the other subgroups in the global group. For now, the /relation subgroup has one possible subgroup.

Coordinate Systems

The first subgroup is the group to define coordinate systems:

/global/relation /coordinateSystems

The default coordinate system must be specified. It is specified in the following group:

/global/relation/ coordinateSystems /default

With attributes:

@lengthUnits={"INCHES" | "MILIMETERS"}

@angleUnits={"RADIANS" | "DEGREES"}

@type = "CARTESIAN"

The following group and attributes define the origin:

/global/relation/ coordinateSystems / default /origin

Having attributes:

@x = FLOAT64 for CARTESIAN coordinates

@y = FLOAT64 for CARTESIAN coordinates

@z = FLOAT64 for CARTESIAN coordinates

For the default coordinates, these attributes should equal zero.

The following group and attributes define the first, second, and third axes relative to the origin of the coordinate system:

/global/relation/ coordinateSystems / default /axis-X

Having attributes:

@x = FLOAT64 for CARTESIAN coordinates = 1 for axis-1 and 0 otherwise.

@y = FLOAT64 for CARTESIAN coordinates = 1 for axis-2 and 0 otherwise.

@z = FLOAT64 for CARTESIAN coordinates = 1 for axis-3 and 0 otherwise.

The vector of axis-X is defined relative to the origin and the point given by (*@x*,*@y*,*@z*) in the axis definition.

Non-default coordinate systems are specified as follows:

/global/relation/ coordinateSystems / coordinateSystem-X

With attributes:

@LengthUnits={“INCHES” | “MILIMETERS”}

@AngleUnits={“RADIANS” | “DEGREES”}

@type = {“CARTESIAN” | “CYLINDRICAL” }

The following group and attributes define the origin (relative to the default coordinate system):

/global/relation/ coordinateSystems / coordinateSystem-X /origin

Having attributes:

@x = FLOAT64 in the default CARTESIAN coordinates

@y = FLOAT64 in the default CARTESIAN coordinates

@z = FLOAT64 in the default CARTESIAN coordinates

The following group and attributes define the first, second, and third coordinates relative to the origin of the coordinate system:

/global/relation/ coordinateSystems / coordinateSystem-X /axis-X

Having attributes:

@x = FLOAT64 in the default CARTESIAN coordinates

@y = FLOAT64 in the default CARTESIAN coordinates

@z = FLOAT64 in the default CARTESIAN coordinates

For CARTESIAN coordinates:

The first axis refers to the new X-axis;

The second axis refers to the new Y-axis;

The third axis refers to the new Z-axis.

For CYLINDRICAL coordinates:

The first axis provides a new unit radial length scale;

The second axis refers to the break line of the angle, i.e., where the angle breaks from 0 to2π radians.

The third axis refers to the axial direction of the new axis.

Coordinates should be given in the default coordinate system. For example, if the default coordinate system is CARTESIAN and a non-default coordinate system is CYLINDRICAL, then the origin and triad coordinates should be given in CARTESIAN coordinates.

The coordinate system does not need to be in normalized coordinates. Coordinates will be scaled appropriately based on the length of each axis vector.

C.3.3 GEOMETRY GROUP

The group /geometry is located directly beneath /HSIESTA, i.e., /HSIESTA/geometry. It contains initial geometric information that will likely be loaded directly into memory once. For example, nodal locations are located here. This group is the equivalent of the “light” data type in XDMF.

Immediately below the /geometry group is a list of components defined in the global group /assembly. Each group listed in /geometry can be traced to /assembly. Components without nodes or elements do not need to be stored in the /geometry group.

Each component in the geometry group shall have an assigned coordinate system. The coordinate system shall be assigned as an attribute:

@coordinateSystem = a string of a named coordinate system.

Sector models (indicated by the attribute /global/spatial/[componentName]/@sectorModel) have two additional attributes:

@symmetricCoordinateSystem = a string of a named cylindrical coordinate system

@numberOfSectors = an integer equal to the number of sectors.

These two attributes shall not be present for components that are not sector models.

For current projects where geometry data are either for nodes or for elements, there are two subgroups:

1. /node – for nodal coordinate information
2. /element – for element connectivity information

C.3.3.1 Node Subgroup

There is one table in /node for the coordinate data. The coordinate data are stored by index in the default coordinate system. These data are stored in the table **+coordinates** at /geometry/[component]/node.

This table shall have the following columns:

1. “identifier”
2. “x” / “radial”
3. “y” / “angular”
4. “z” / “axial”

The column “identifier” provides the nodal identifier of the node. This identifier does not need to correspond to the index.

The first entry is the column for CARTESIAN coordinates. The second entry is the column for CYLINDRICAL coordinates. The unit for ANGULAR coordinates is radians and spans from the range $[0,2\pi]$. There shall be one row for every node in the model.

C.3.3.2 Element Subgroup

There is one table in the /element group to map the element identifier to the appropriate element type. This table is **+topology** and located at /geometry/element.

There shall be three columns in this table. The first column, “identifier” provides the element identifier of the element. This identifier does not need to correspond to the index of the table. The second column, “index”, shall contain the index number for the appropriate row in the element connectivity table. The third column, “type”, shall be a string with the appropriate SIESTA designation, e.g., EL2D. There shall be one row for every element in the model.

There shall be multiple tables within /element for every element type in the model. These tables shall be named after the element type: **+[ELEMENT TYPE]**. For example, a model with BRICK8 and TETS4 shall have two tables: **+BRICK8** for BRICK8 elements and **+TETS4** for TETS4 elements. The number of rows of each table shall equal the number of elements of that type listed in table **+topology**. The index from **+topology** maps to the index in the appropriate element table. The number of columns in each table equals the number of nodes in the connectivity matrix. The columns are named based on the name provided by the SIESTA format. For example, elements of type QUAD4 will have columns “A”, “B”, “C”, and “D”. Columns are to be arranged in ascending alphabetical ordering. In the previous example, the first column will always be “A”, the second column will always be “B”, etc…

ENTRIES IN THE CONNECTIVITY MATRIX REFERENCE THE NODAL INDICES, NOT THE NODE NUMBER. This requirement intentionally breaks the previous standard of referencing nodes by their identifier. Nodes are referenced by index order to match how they are referenced in fields data. Stresses and temperatures must be referenced by index. Furthermore, nodal indices always begin a known index and increase by a value of one. The nodal identifier may not start at a constant index, change by a constant amount, or even be listed in order. Consequently, it is easier to look up the nodal index in a table than the nodal identifier.

HSIESTA has the following legal element types where the number of nodes in the connectivity matrix is fixed:

- TRI3 for 3 node triangles
- TRI6 for 6 node quadratic triangles
- QUAD4 for 4 node quadrilaterals
- QUAD8 for 8 node quadratic quadrilaterals
- TET4 for 4 node tetrahedrons
- TET10 for 10 node fully quadratic tetrahedrons
- PYRAMID5 for 5 node elements
- PYRAMID13 for 13 node quadratic pyramids
- WEDGE6 for 6 node elements
- WEDGE15 for 15 node quadratic wedges
- BRICK8 for 8 node elements
- BRICK20 for 20 node fully quadratic elements

These elements have a fixed number of nodes, i.e., a four node quadratic element must have four nodes. These nodes must be unique. If the nodes are not unique, then the element will be considered to be in error. Annex C-1 provides a list of these elements with the associated connectivity matrices and faces. Whenever possible, HSIESTA retains the SIESTA ordering.

HSIESTA supports linear element types and certain quadratic types. These element types have a fixed number of nodes. Elements with a variable number of nodes are not supported. These elements should be downgraded to the appropriate linear element.

Nodes referenced in an /element group are the nodes referenced in the /node group of the same component. Consequently, a component can only have elements if it has nodes.

C.3.4 FIELDS GROUP

The group /fields is located directly beneath /HSIESTA, i.e., /HSIESTA/fields. It contains field data that is needed case-by-case during visualization or computations. For example, stresses might be brought temporarily into memory, used, and then discarded during GUI visualization. Nodal stresses might be shown in the GUI for one step, and the other loadsteps in the model do not need to be loaded. The /fields group is equivalent to the “heavy” datatype in XDMF.

Fields data are stored by component. Components without field data do not need to be listed. For example, the /assembly group does not need to be referenced.

The /[component] group contains the table **+parameters** that contain input parameters for all loadcases. This table shall have an unlimited number of rows and columns. Each column shall be named after a parameter (e.g., “loadcase”). Each column shall contain the same type of data. The available data types are 32-bit integers and 64-bit floating point numbers. Each row shall represent a case listed in the table **+description** in the temporal group. The number of loadcases must be the same. They shall not be allowed to vary within a component. The cell value for the index (loadcase X parameter) shall be the parameter for that loadcase and parameter. If the combination of loadcase and parameter does not have a value, then the cell shall contain NaN.

The table **+parameters** may contain any number of parameters. For a DARWIN specific analysis, it shall contain at a minimum the following columns:

- “loadcase” to specify the DARWIN specific loadcase identifier. This column is a 32-bit integer.

- “feSolverID” to specify the finite element solver identifier. This column is a 32-bit integer.

Additionally, the psuedo-time may also be stored using the following column:

- “psuedoTime” which holds the psuedo-time from the finite element analysis. This column is a 64-bit floating point number.

Groups of entities (e.g., node) are within every component containing field data. Entities without field data do not need to be listed. For example, element data does not need to be stored if only nodal stresses/temperatures are listed.

Groups of geometric entities contain data. For example, there may be a geometric entity “node” if the component contains nodal data. Similarly, there might be geometric entities in the future for faces, elements, Gauss points, etc… For the data used in DARWIN, we shall restrict this group to be “node” or “element” at this time.

Each entity group (e.g., /fields/[component]/{entity}), there shall be 3D datasets of field quantities that contain field data. The field data are stored as 64-bit floating point numbers. The ordering of every dataset in the HDF5 file shall be:

Case X Index X Component

Here, case refers to the loadcase, index refers to the entity index (e.g., the node index), and component refers to the field components (if applicable). These indices shall always be referenced as attributes and always attached to the field dataset for all 3D datasets:

- *@dimension-0*=”Case”
- *@dimension-1*=”Index”
- *@dimension-2*=”Component”

Every type of field data shall have a separate 3D dataset. For example, there shall be a dataset for “stress” if nodal stress data are present. There shall be datasets for “stress” and “residualStress” if the analysis contains both types of information. For the data used by DARWIN currently, we shall restrict these datasets to be “stress”, “temperature”, and “data-X”. However, data from

DEFORM (e.g., residual stress, residual strain, average grain size, and ALA grain size) would easily fit into this framework.

The number of indices in an entity table must correspond to the number of rows in the corresponding /geometry/[component]/{entity}/**+[table]**.

- For nodes, the reference table is **+coordinates**.
- For elements, the reference table is **+topology**.

The index dimension in /fields/[component]/{entity}/{dataset} and rows in these geometry tables match each other.

For **+stress**, there shall be an attribute attached to the table: *@units*={“KSI” | “MPA”}.

For **+temperature**, there shall be an attribute table: *@units*={“FAHRENHEIT” | “CELSIUS”}.

The number of components in the dataset shall be determined by the field type.

For **+stress**, there shall be six components. They shall be listed as attributes attached to the table. The components shall be ordered as following:

0. S11 and referenced as *@component-0*=”S11”
1. S22 and referenced as *@component-1*=”S22”
2. S33 and referenced as *@component-2*=”S33”
3. S12 and referenced as *@component-3*=”S12”
4. S13 and referenced as *@component-4*=”S13”
5. S23 and referenced as *@component-5*=”S23”

Further note: For **+stress**, it is assumed that stresses are Cauchy stresses with a coordinate system that coincides with the coordinate system for the component. Consequently, if the coordinate system is cylindrical, then the stresses will be cylindrical as well.

For **+temperature**, there shall be one component. It shall be listed as an attribute attached to the table. The component shall be ordered as follows:

0. Value and referenced as *@component-0*=”Value”

For arbitrary data (either nodal or elemental), there shall be an allowable arbitrary data type **+data-X**. The format shall allow as many different datasets as needed. For example, if there are three different arbitrary data types, then there shall be three data tables: **+data-1**, **+data-2**, and **+data-3**.

This data type shall have attributes attached to the 3D dataset:

1. *@name* – that provides the name for this arbitrary data type
2. *@numberOfComponents* – that provides the number of columns for the data type

3. *@units* – that provides the units for this data type (if applicable). Only units in DARWIN shall be allowable. If units are not provided, then the string NONE shall be used instead.

The dataset shall contain as many components as necessary to define the data type. Each component shall be referenced as attribute *@component-X*=[Description].

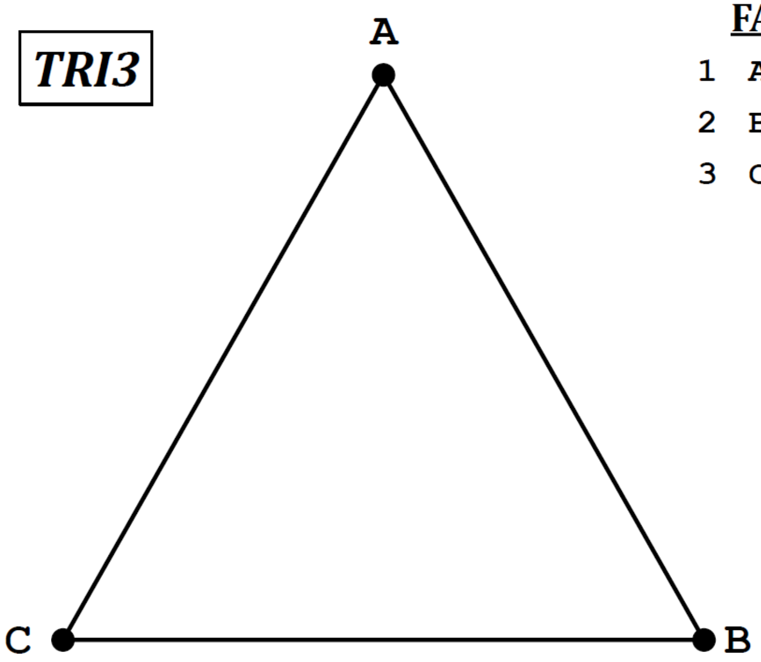
<u>Annex C-1 Legal Elements in HSIESTA</u>.

HSIESTA has the following legal element types where the number of nodes in the connectivity matrix is fixed:

- TRI3 for 3 node triangles
- TRI6 for 6 node quadratic triangles
- QUAD4 for 4 node quadrilaterals
- QUAD8 for 8 node quadratic quadrilaterals
- TET4 for 4 node tetrahedrons
- TET10 for 10 node fully quadratic tetrahedrons
- PYRAMID5 for 5 node elements
- PYRAMID13 for 13 node quadratic pyramids
- WEDGE6 for 6 node elements
- WEDGE15 for 15 node quadratic wedges
- BRICK8 for 8 node elements
- BRICK20 for 20 node fully quadratic elements

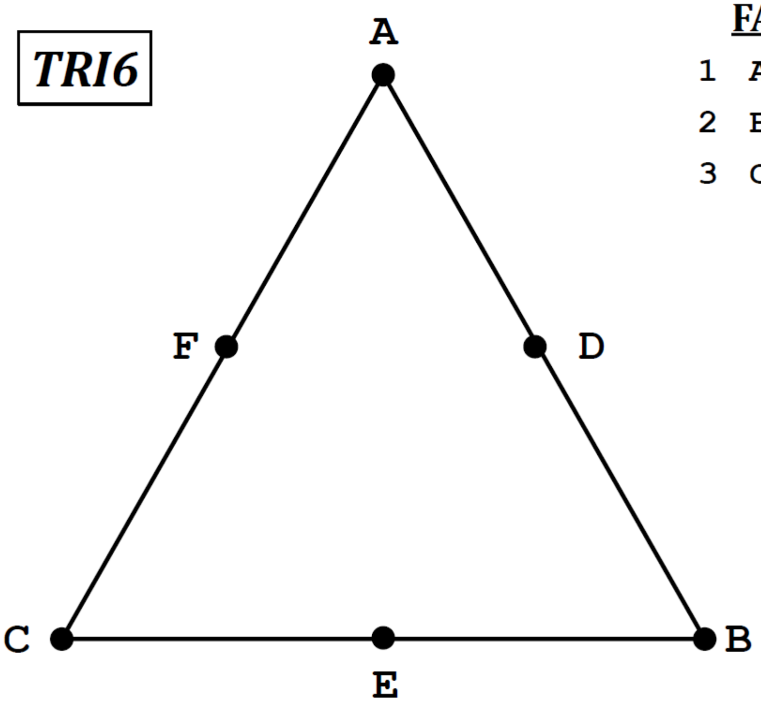
These elements have a fixed number of nodes, i.e., a four node quadratic element must have four nodes. These nodes must be unique. If the nodes are not unique, then the element will be considered to be in error. The following diagrams illustrate the different elements with the associated connectivity matrices and faces. Whenever possible, HSIESTA retains the SIESTA ordering.

TRI3



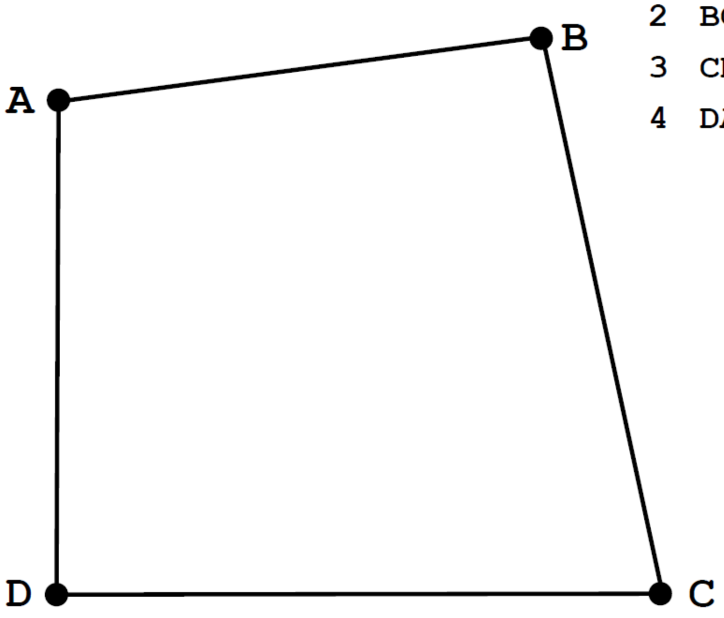
FACE	
1	AB
2	BC
3	CA

TRI6



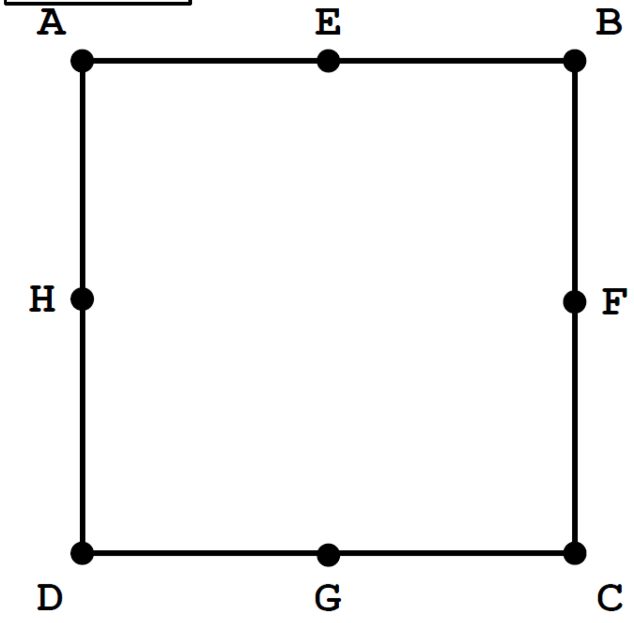
FACE	
1	ADB
2	BEC
3	CFA

QUAD4

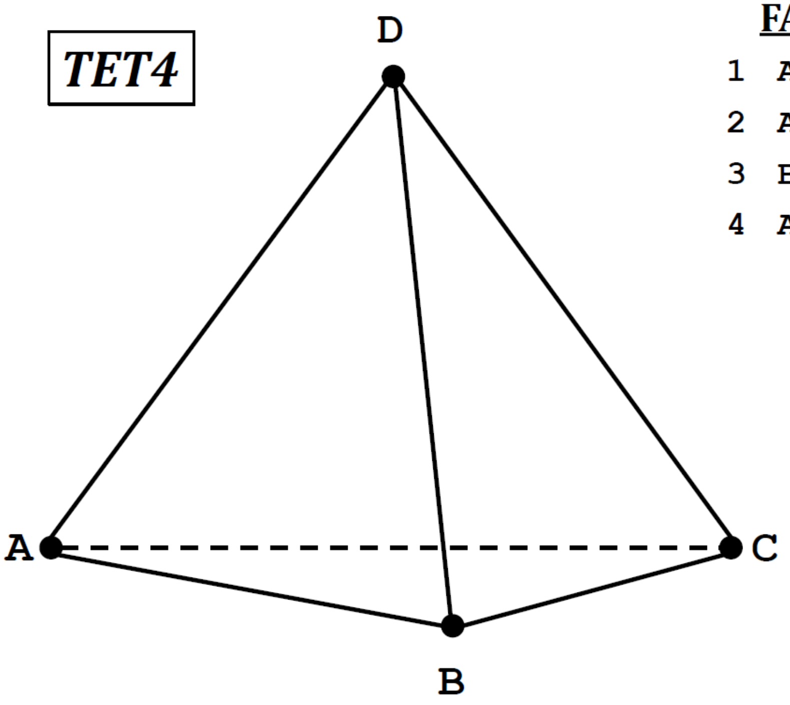


FACE	
1	AB
2	BC
3	CD
4	DA

QUAD8

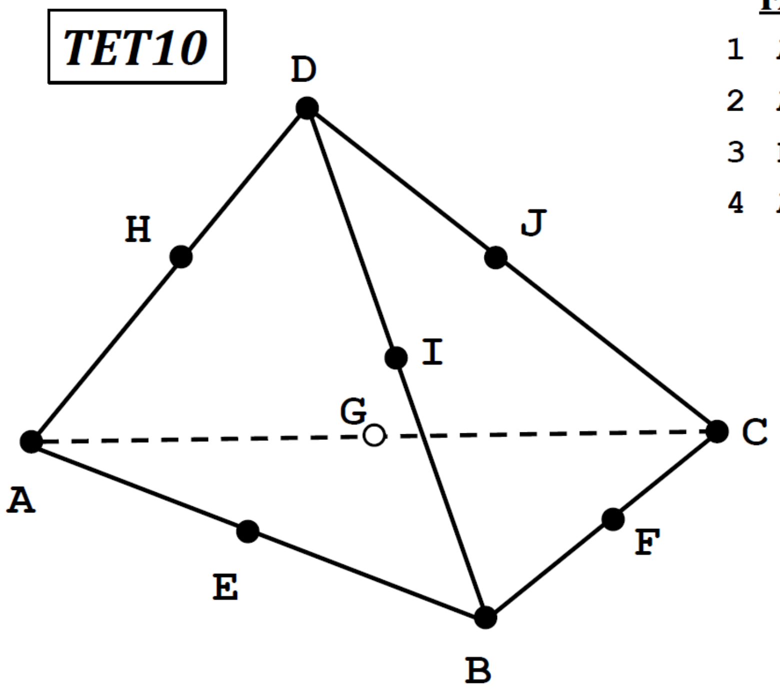


FACE	
1	AEB
2	BFC
3	CGD
4	DHA



FACE

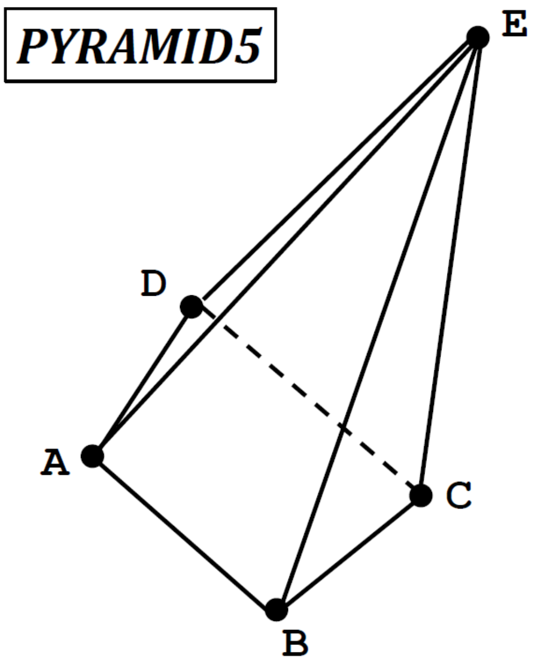
1	ABC
2	ACD
3	BCD
4	ADB



FACE

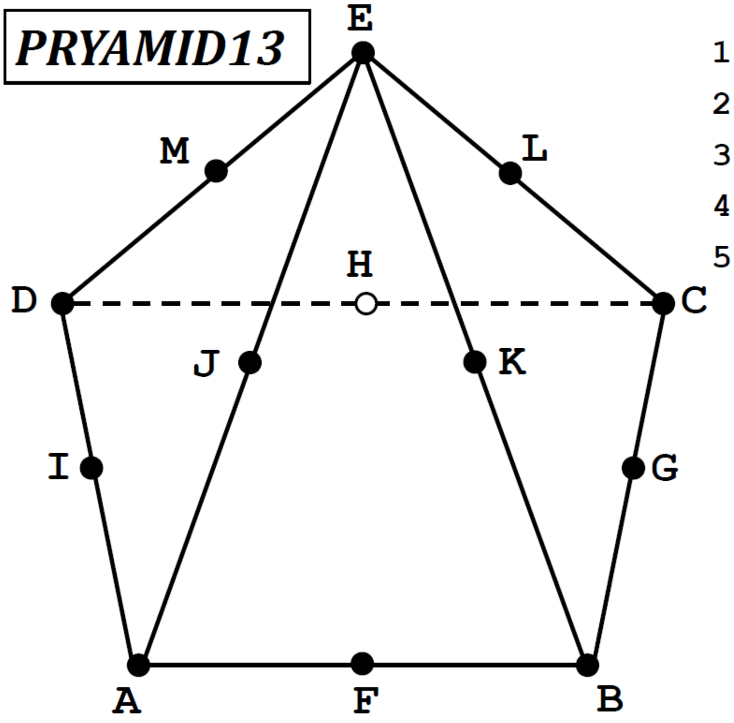
1	AEBFCG
2	AGCJDH
3	BFCJDI
4	AHDIBE

PYRAMID5



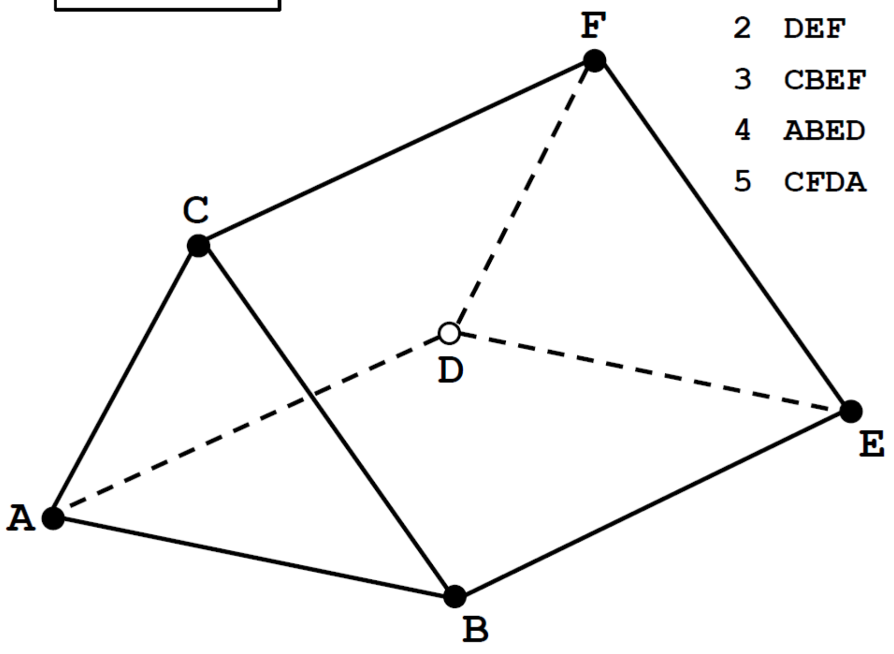
FACE	
1	ABE
2	DCE
3	BCE
4	ADE
5	ADCB

PRYAMID13



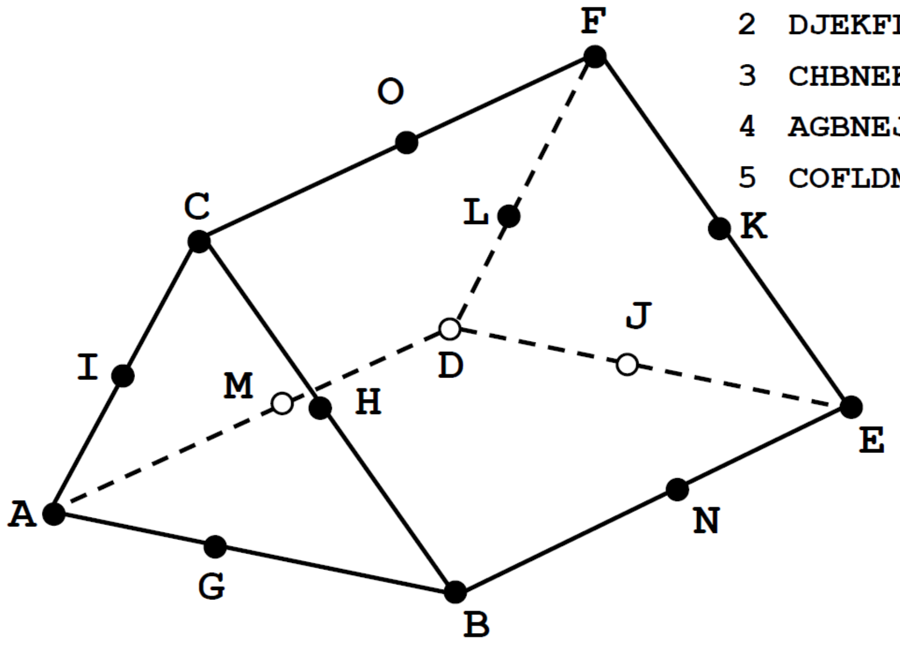
FACE	
1	AFBKEJ
2	DHCLEM
3	BGCLEK
4	AIDMEJ
5	AIDHCGBF

WEDGE6



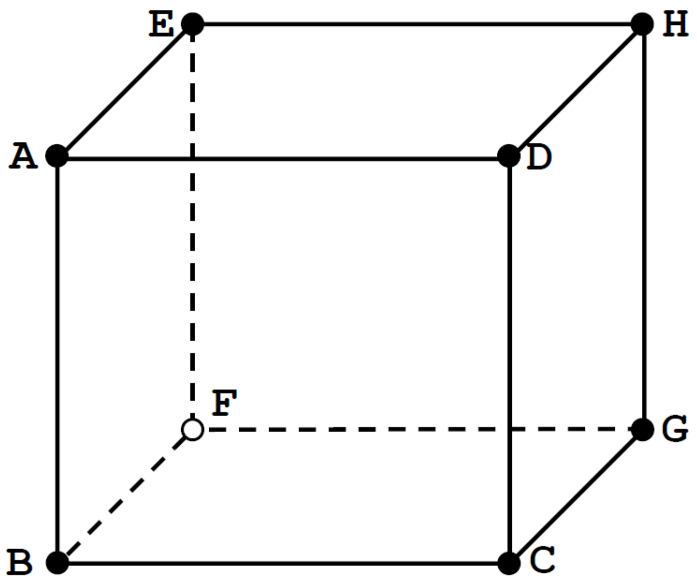
FACE	
1	ABC
2	DEF
3	CBEF
4	ABED
5	CFDA

WEDGE15



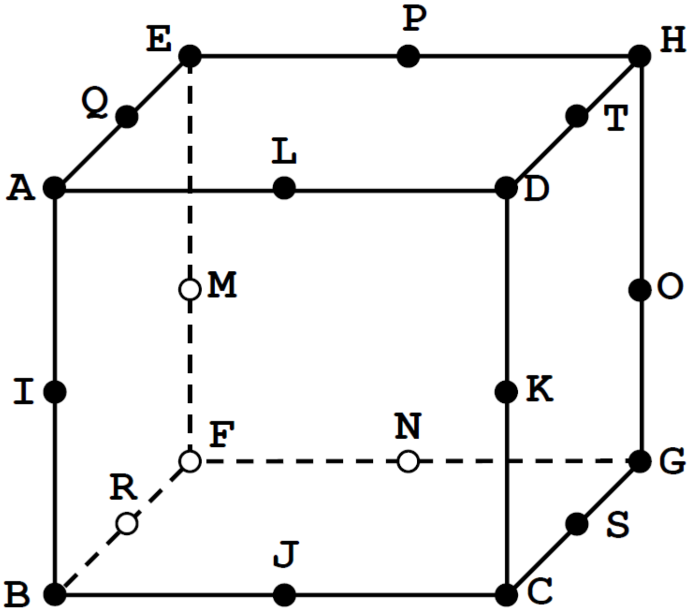
FACE	
1	AGBHCI
2	DJEKFL
3	CHBNEKFO
4	AGBNEJDM
5	COFLDMAI

BRICK8



FACE	
1	ABCD
2	EFGH
3	AEFB
4	DHGC
5	ADHE
6	BCGF

BRICK20



FACE	
1	AIBJCKDL
2	EMFNGOHP
3	AQEMFRBI
4	DTHOGSCK
5	ALDTHPEQ
6	BJCSGNFR

APPENDIX D—HSIESTA API FUNCTIONS

D.1 CONTENTS

1. File Access

Number	Description

API-FA-100	***OPEN_HSIESTA***
	`INTEGER FUNCTION OPEN_HSIESTA( IH, PATH )`
	`   CHARACTER(LEN=80), INTENT(IN)    :: PATH`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	Description: Opens an HSIESTA file to read and returns handle
	Predecessors: None
	Input: *path* is the path to the HSIESTA file
	ih is the integer handle for the API context
	Return: 0 if no error and non-zero otherwise

API-FA-101	***CLOSE_HSIESTA***
	`INTEGER FUNCTION CLOSE_HSIESTA( IH )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	Description: Closes an HSIESTA file
	Predecessors: None
	Input: *ih* is the integer handle for the API context
	Output: None
	Return: 0 if no error and non-zero otherwise

API-FA-102	***GET_HSIESTA_VERSION***
	`SUBROUTINE GET_HSIESTA_VERSION( IH, VERSION )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=4), INTENT(OUT)    :: VERSION`
	Description: Provides the version of the HSIESTA file
	Predecessors: None
	Input: *ih* is the integer handle for the API context
	Output: *version* provides the version number of the HSIEST file.

2. Component Access

Number	Description

API-CA-100	***SET_COMPONENT***
	`INTEGER FUNCTION SET_COMPONENTS( IH, COMPONENT )`
	`    INTEGER(KIND=8), INTENT(IN)      :: IH`
	`    CHARACTER(LEN=80), INTENT(IN)     :: COMPONENT`
	Description: Sets component of interest
	Predecessors: None
	Input: *ih* is the integer handle for the API context
	component is the component name of interest
	Output: None
	Return: 0 if no error and non-zero otherwise
	Note: The 'assembly' component can always be used to obtain the highest level component listing.

API-CA-101	***GET_PARENT***
	`SUBROUTINE GET_PARENT ( IH, PARENT )`
	`    INTEGER(KIND=8), INTENT(IN)      :: IH`
	`    CHARACTER(LEN=80), INTENT(OUT)    :: PARENT`
	Description: Gets the parent of the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	Output: *parent* is the parent of the specified component

API-CA-102	***GET_NUM_CHILDREN***
	`SUBROUTINE GET_NUM_CHILDREN( IH, NUM_CHILDREN )`
	`    INTEGER(KIND=8), INTENT(IN)      :: IH`
	`    INTEGER, INTENT(OUT)              :: NUM_CHILDREN`
	Description: Gets the number of children for the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	Output: *num_children* indicates the number of children

API-CA-103	***GET_CHILD***
	`SUBROUTINE GET_CHILD( IH, CHILD_NUM, CHILD_NAME )`
	`    INTEGER(KIND=8), INTENT(IN)      :: IH`
	`    INTEGER, INTENT(IN)               :: CHILD_NUM`
	`    CHARACTER(LEN=80), INTENT(OUT)    :: CHILD_NAME`
	Description: Gets the child name
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	child_num is the child number
	Output: *child_name* is the name of the child

3. Component Information

Number	Description

API-IC-100	***GET_COORDINATE_SYSTEM***
	`SUBROUTINE GET_COORDINATE_SYSTEM( IH, COORDINATE_SYSTEM )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=80), INTENT(OUT)   :: COORDINATE_SYSTEM`
	Description: Gets the coordinate system of the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	Output: *coordinate_system* is the coordinate system name

API-IC-101	***GET_NUM_NODE***
	`SUBROUTINE GET_NUM_NODE( IH, NUM_NODE )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(OUT)             :: NUM_NODE`
	Description: Gets the number of nodes for the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	Output: *num_node* is the number of nodes

API-IC-102	***GET_NUM_ELEM***
	`SUBROUTINE GET_NUM_ELEM( IH, NUM_ELEM )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(OUT)             :: NUM_ELEM`
	Description: Gets the number of elements for the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	Output: *num_elem* is the number of elements

API-IC-103	***GET_NUM_CASES***
	`SUBROUTINE GET_NUM_CASES( IH, NUM_CASES )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(OUT)             :: NUM_CASES`
	Description: Gets the number of load cases for the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	Output: *num_cases* is the number of load cases

API-IC-104	***GET_SYMM_COORDINATE_SYSTEM***
	`SUBROUTINE GET_SYMM_COORDINATE_SYSTEM( IH, COORDINATE_SYSTEM )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=80), INTENT(OUT)   :: COORDINATE_SYSTEM`
	Description: Gets the symmetric coordinate system of the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	Output: *coordinate_system* is the coordinate system name

3. Component Information

Number	Description

API-IC-105	***GET_NUM_SECTORS***
	`SUBROUTINE GET_NUM_SECTORS( IH, NUM_SECTORS )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(OUT)             :: NUM_SECTORS`
	Description: Gets the number of sectors for the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	Output: *num_sectors* is the number of sectors

4. Coordinate System Information

Number	Description

API-IC-100	***SET_COORDINATE_SYSTEM***
	`INTEGER FUNCTION SET_COORDINATE_SYSTEM( IH, COORDINATE_SYSTEM )`
	`INTEGER(KIND=8), INTENT(IN)     :: IH`
	`CHARACTER(LEN=80), INTENT(IN)   :: COORDINATE_SYSTEM`
	Description: Sets the coordinate system
	Predecessors: None
	Input: *ih* is the integer handle for the API context
	coordinate_system is the coordinate system name
	Output: None
	Return: 0 if no error and non-zero otherwise

API-IC-101	***GET_COORDINATE_SYSTEM_TYPE***
	`SUBROUTINE GET_COORDINATE_SYSTEM_TYPE( IH, TYPE )`
	`INTEGER(KIND=8), INTENT(IN)     :: IH`
	`CHARACTER(LEN=20), INTENT(OUT)  :: TYPE`
	Description: Gets the type of a specified coordinate system
	Predecessors: coordinate system
	Input: *ih* is the integer handle for the API context
	Output: *type* is a string with two possible values: ‘CARTESIAN’ or ‘CYLINDRICAL’

API-IC-102	***GET_LENGTH_UNITS***
	`SUBROUTINE GET_LENGTH_UNITS( IH, UNITS )`
	`INTEGER(KIND=8), INTENT(IN)     :: IH`
	`CHARACTER(LEN=20), INTENT(OUT)  :: UNITS`
	Description: Gets the length units for a specified coordinate system
	Predecessors: coordinate system
	Input: *ih* is the integer handle for the API context
	Output: *units* is a string with two possible values: ‘INCHES’ or ‘MILIMETERS’

API-IC-103	***GET_ANGLE_UNITS***
	`SUBROUTINE GET_ANGLE_UNITS( IH, UNITS )`
	`INTEGER(KIND=8), INTENT(IN)     :: IH`
	`CHARACTER(LEN=20), INTENT(OUT)  :: UNITS`
	Description: Gets the angle units for a specified coordinate system
	Predecessors: coordinate system
	Input: *ih* is the integer handle for the API context
	Output: *units* is a string with two possible values: ‘DEGREES’ or ‘RADIANS’

4. Coordinate System Information

Number	Description

API-IC-104	***GET_XYZ_ORIGIN***
	`SUBROUTINE GET_XYZ_ORIGIN( IH, X, Y, Z )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)             :: X, Y, Z`
	Description: Gets point that specifies the origin
	Predecessors: coordinate system
	Input: *ih* is the integer handle for the API context
	Output: *x* is the x-location
	y is the y-location
	z is the z-location

API-IC-105	***GET_XYZ_FIRST_AXIS***
	`SUBROUTINE GET_XYZ_FIRST_AXIS( IH, X, Y, Z )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)             :: X, Y, Z`
	Description: Gets point that specifies the first axis relative to the origin
	Predecessors: coordinate system
	Input: *ih* is the integer handle for the API context
	Output: *x* is the x-location
	y is the y-location
	z is the z-location

API-IC-106	***GET_XYZ_SECOND_AXIS***
	`SUBROUTINE GET_XYZ_SECOND_AXIS( IH, X, Y, Z )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)             :: X, Y, Z`
	Description: Gets point that specifies the second axis relative to the origin
	Predecessors: coordinate system
	Input: *ih* is the integer handle for the API context
	Output: *x* is the x-location
	y is the y-location
	z is the z-location

API-IC-107	***GET_XYZ_THIRD_AXIS***
	`SUBROUTINE GET_XYZ_THIRD_AXIS( IH, X, Y, Z )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)             :: X, Y, Z`
	Description: Gets point that specifies the third axis relative to the origin
	Predecessors: coordinate system
	Input: *ih* is the integer handle for the API context
	Output: *x* is the x-location
	y is the y-location
	z is the z-location

5. Geometric Nodal Information

Number	Description

API-GN-100	***GET_ALL_NODAL_IDS***
	`SUBROUTINE GET_ALL_NODAL_IDS( IH, NUM_NODE, NODAL_IDS )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)             :: NUM_NODE`
	`   INTEGER, POINTER,INTENT(OUT)    :: NODAL_IDS(:)`
	Description: Gets all nodal identifiers in the component.
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	num_node is the number of nodes in the component
	Output: *nodal_ids* is a pointer to a vector of nodal identifiers. It has length equal to the number of nodes. Nodal identifiers are arranged from the first to the last nodal index.

API-GN-101	***GET_ALL_NODAL_XLOCS***
	`SUBROUTINE GET_ALL_NODAL_XLOCS( IH, NUM_NODE, XLOCS )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)             :: NUM_NODE`
	`   REAL*8, POINTER, INTENT(OUT)    :: XLOCS(:)`
	Description: Gets all nodal x-locations in the component.
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	num_node is the number of nodes in the component
	Output: *xlocs* is a pointer to a vector of nodal x-locations. It has length equal to the number of nodes. X-locations are arranged from the first to the last nodal index.

API-GN-102	***GET_ALL_NODAL_YLOCS***
	`SUBROUTINE GET_ALL_NODAL_YLOCS( IH, NUM_NODE, YLOCS )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)             :: NUM_NODE`
	`   REAL*8, POINTER, INTENT(OUT)    :: YLOCS(:)`
	Description: Gets all nodal y-locations in the component.
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	num_node is the number of nodes in the component
	Output: *ylocs* is a pointer to a vector of nodal y-locations. It has length equal to the number of nodes. Y-locations are arranged from the first to the last nodal index.

5. Geometric Nodal Information

Number	Description

API-GN-103	***GET_ALL_NODAL_ZLOCS***
	`SUBROUTINE GET_ALL_NODAL_ZLOCS( IH, NUM_NODE, ZLOCS )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_NODES`
	`   REAL*8, POINTER, INTENT(OUT)     :: ZLOCS(:)`
	Description: Gets all nodal z-locations in the component.
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	num_node is the number of nodes in the component
	Output: *zlocs* is a pointer to a vector of nodal z-locations. It has length equal to the number of nodes. Z-locations are arranged from the first to the last nodal index.

API-GN-104	***GET_ALL_NODAL_LOCS3D***
	`SUBROUTINE GET_ALL_NODAL_LOCS3D( IH, NUM_NODE, LOCS3D )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_NODES`
	`   REAL*8, INTENT(OUT)              :: LOCS3D(3,NUM_NODE)`
	Description: Gets all nodal locations in a 3D component.
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	num_node is the number of nodes in the component
	Output: *locs3d* is a vector of 3d nodal locations. It has dimensions equal to the number of nodes by the dimension (set equal to 3). Locations are arranged from the first to the last nodal index. The first dimension is X; the second dimension is Y; and the third dimension is Z.

6. Geometric Element Information

Number | Description

API-GE-100

GET_ALL_ELEMENT_IDS

```
SUBROUTINE GET_ALL_ELEMENT_IDS( IH, NUM_ELEM, ELEM_IDS )
   INTEGER(KIND=8), INTENT(IN)      :: IH
   INTEGER, INTENT(IN)              :: NUM_ELEM
   INTEGER, POINTER, INTENT(OUT)    :: ELEM_IDS(:)
```

Description: Gets all element identifiers in the component.

Predecessors: component

Input: *ih* is the integer handle for the API context
num_elem is the number of elements in the component

Output: *elem_ids* is a pointer to a vector of element identifiers. It has length equal to the number of element. Element identifiers are arranged from the first to the last nodal index.

API-GE-101

GET_ALL_ELEM_TYPES

```
SUBROUTINE GET_ALL_ELEM_TYPES( IH, NUM_ELEM, ELEM_TYPE )
   INTEGER(KIND=8), INTENT(IN)      :: IH
   INTEGER, INTENT(IN)              :: NUM_ELEM
   CHARACTER(LEN=20), INTENT(OUT)   :: ELEM_TYPE(NUM_ELEM)
```

Description: Gets the element type.

Predecessors: component

Input: *ih* is the integer handle for the API context
num_elem is the number of elements in the component

Output: *elem_type* is an array of characters that reflect the element type.

API-GE-102

SET_ELEM_TYPE

```
INTEGER FUNCTION SET_ELEM_TYPE( IH, ELEM_TYPE )
   INTEGER(KIND=8), INTENT(IN)      :: IH
   CHARACTER(LEN=20), INTENT(IN)    :: ELEM_TYPE
```

Description: Sets the element type.

Predecessors: component

Input: *ih* is the integer handle for the API context
elem_type is the element type.

Output: None

Return: 0 if no error and non-zero otherwise

API-GE-103

GET_NUM_ELEM_TYPE

```
SUBROUTINE GET_NUM_ELEM_TYPE( IH, NUM_ELEM_TYPE )
   INTEGER(KIND=8), INTENT(IN)      :: IH
   INTEGER, INTENT(OUT)             :: NUM_ELEM_TYPE
```

Description: Gets the number of elements of a particular type.

Predecessors: component; element type

Input: *ih* is the integer handle for the API context

Output: *num_elem_type* is the number of elements of a particular type.

6. Geometric Element Information

Number	Description

API-GE-104	***SET_ELEM_CONNECTIVITY***
	`INTEGER FUNCTION SET_ELEM_CONNECTIVTY( IH, CONNECTIVITY )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=20), INTENT(IN)    :: CONNECTIVITY`
	Description: Sets the connectivity index.
	Predecessors: component; element type
	Input: *ih* is the integer handle for the API context
	connectivity is the element connectivity index.
	Output: None
	Return: 0 if no error and non-zero otherwise

API-GE-105	***GET_ELEM_CONNECTIVITY***
	`SUBROUTINE GET_ELEM_CONNECTIVTY( IH, NUM_ELEM, NODE_INDEX )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_ELEM`
	`   INTEGER, INTENT(OUT)             :: NODE_INDEX(NUM_ELEM)`
	Description: Gets the node index for the selected connectivity.
	Predecessors: component; element type; connectivity
	Input: *ih* is the integer handle for the API context
	num_elem is the number of elements in the component
	Output: *node_index* is the nodal indices located in the element connectivity table at the selected location. This vector is arranged by element index.

API-GE-106	***GET_INDIVIDUAL_ELEM_TYPE***
	`SUBROUTINE GET_INDIVIDUAL_ELEM_TYPE( IH, ELEM_TYPE )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=20), INTENT(OUT)   :: ELEM_TYPE`
	Description: Gets the element type for an individual.
	Predecessors: component; entity type; individual
	Input: *ih* is the integer handle for the API context
	Output: *elem_type* is an array of characters that reflect the element type.

6. Geometric Element Information

Number	Description

API-GE-107	***GET_INDIVIDUAL_ELEM_CONNECTIVITY***
	`SUBROUTINE GET_INDIVIDUAL_ELEM_CONNECTIVITY( &`
	`    IH, NUM_INDICES, ELEM_CONNECTIVITY )`
	`    INTEGER(KIND=8), INTENT(IN)      :: IH`
	`    INTEGER, INTENT(IN)              :: NUM_INDICES`
	`    INTEGER, INTENT(OUT)             :: &`
	`      ELEM_CONNECTIVITY(NUM_INDICES)`
	Description: Gets the element connecitivity for an individual.
	Predecessors: component; entity type; individual
	Input: *ih* is the integer handle for the API context
	num_indices is the number of indices for the element type of the individual. It must match the associated individual.
	Output: *elem_connectivity* is an array of integers that reflects the individual's element connectivity.

7. Field Parameters

Number	Description

API-FP-100	***SET_CASE***
	`INTEGER FUNCTION SET_CASE( IH, CASE )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: CASE`
	Description: Sets the loadcase for the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	case is the loadcase number
	Output: None
	Return: 0 if no error and non-zero otherwise

API-FP-101	***GET_NUM_FIELD_PARAMETERS***
	`SUBROUTINE GET_NUM_FIELD_PARAMETER( IH, NUM_PARAMS )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(OUT)             :: NUM_PARAMS`
	Description: Gets the field parameters for the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	Output: *num_params* is the number of field parameters

API-FP-102	***SET_FIELD_PARAMETER***
	`INTEGER FUNCTION SET_FIELD_PARAMETER( IH, IPARAM )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: IPARAM`
	Description: Sets the field parameter for the specified component
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	iparam is the selected parameter number
	Output: None
	Return: 0 if no error and non-zero otherwise

API-FP-103	***GET_FIELD_PARAMETER_NAME***
	`SUBROUTINE GET_FIELD_PARAMETER_NAME( IH, PARAM_NAME )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   CHARACTER(LEN=80), INTENT(OUT)   :: PARAM_NAME`
	Description: Gets selected field parameter name
	Predecessors: component; field parameter
	Input: *ih* is the integer handle for the API context
	Output: *param_name* is the parameter name

7. Field Parameters

Number	Description

API-FP-104	***GET_FIELD_PARAMETER_TYPE***
	`SUBROUTINE GET_FIELD_PARAMETER_NAME( IH, PARAM_TYPE )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=20), INTENT(OUT)   :: PARAM_TYPE`
	Description: Gets selected field parameter name
	Predecessors: component; field parameter
	Input: *ih* is the integer handle for the API context
	Output: *param_type* is the parameter type. There are two available strings at this time: 'INT32' for 32-bit integers and 'FLT64' for 64-bit floating point numbers.

API-FP-105	***GET_FIELD_PARAMETER_VALUE_FLT64***
	`SUBROUTINE GET_FIELD_PARAMETER_VALUE_FLT64( IH, PARAM_VALUE )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   REAL*8, INTENT(OUT)              :: PARAM_VALUE`
	Description: Gets field parameter value for a 64-bit floating point number
	Predecessors: component; field parameter; loadcase
	Input: *ih* is the integer handle for the API context
	Output: *param_value* is the parameter value

API-FP-106	***GET_FIELD_PARAMETER_VALUE_INT32***
	`SUBROUTINE GET_FIELD_PARAMETER_VALUE_INT32( IH, PARAM_VALUE )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(OUT)             :: PARAM_VALUE`
	Description: Gets field parameter value for an integer
	Predecessors: component; field parameter; loadcase
	Input: *ih* is the integer handle for the API context
	Output: *param_value* is the parameter value

8. Entity Information

Number	Description

API-EI-100	***SET_ENTITY_TYPE***
	`INTEGER FUNCTION SET_ENTITY_TYPE( IH, ENTITY )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=20), INTENT(IN)    :: ENTITY`
	Description: Sets the entity type
	Predecessors: component
	Input: *ih* is the integer handle for the API context
	entity is the name of the entity type
	Output: None
	Return: 0 if no error and non-zero otherwise
	Note: At this time, HSIESTA only supports the following entity types: node and element. Other entity types may follow.

API-EI-101	***GET_STRESS_UNITS***
	`SUBROUTINE GET_STRESS_UNITS( IH, UNITS )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=20), INTENT(OUT)   :: UNITS`
	Description: Gets the stress units
	Predecessors: component; entity type
	Input: *ih* is the integer handle for the API context
	Output: *units* is a string with two possible values: ‘KSI’ or ‘MPA’

API-EI-102	***GET_TEMPERATURE_UNITS***
	`SUBROUTINE GET_TEMPERATURE_UNITS( IH, UNITS )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=20), INTENT(OUT)   :: UNITS`
	Description: Gets the temperature units
	Predecessors: component; entity type
	Input: *ih* is the integer handle for the API context
	Output: *units* is a string with two possible values: ‘FAHRENHEIT’ or ‘CELSIUS’

8. Entity Information

Number	Description

API-EI-103	***SET_LIST_BY_INDICES***
	`INTEGER FUNCTION SET_LIST_BY_INDICES( IH, NUM_ITEMS, ITEMS )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_ITEMS`
	`   INTEGER, INTENT(IN)              :: ITEMS(NUM_ITEMS)`
	Description: Sets a list by the entity indices
	Predecessors: component; entity type
	Input: *ih* is the integer handle for the API context
	num_items is the number of items in the list
	items is the list of items by their indices
	Output: None
	Return: 0 if no error and non-zero otherwise

API-EI-104	***SET_LIST_BY_IDS***
	`INTEGER FUNCTION SET_LIST_BY_IDS( IH, NUM_ITEMS, ITEMS )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_ITEMS`
	`   INTEGER, INTENT(IN)              :: ITEMS(NUM_ITEMS)`
	Description: Sets a list by the entity identifiers
	Predecessors: component; entity type
	Input: *ih* is the integer handle for the API context
	num_items is the number of items in the list
	items is the list of items by their identifiers
	Output: None
	Return: 0 if no error and non-zero otherwise

API-EI-105	***GET_LIST_BY_INDICES***
	`SUBROUTINE GET_LIST_BY_INDICES( IH, NUM_ITEMS, ITEMS )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_ITEMS`
	`   INTEGER, POINTER, INTENT(OUT)    :: ITEMS(:)`
	Description: Gets a list by the entity indices
	Predecessors: component; entity type; list
	Input: *ih* is the integer handle for the API context
	num_items is the number of items in the list
	Output: *items* is a pointer to the list of items by their indices

8. Entity Information

Number	Description

API-EI-106	***GET_LIST_BY_IDS***
	`SUBROUTINE GET_LIST_BY_IDS( IH, NUM_ITEMS, ITEMS )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_ITEMS`
	`   INTEGER, POINTER, INTENT(OUT)    :: ITEMS(NUM_ITEMS)`
	Description: Gets a list by the entity identifiers
	Predecessors: component; entity type; list
	Input: *ih* is the integer handle for the API context
	num_items is the number of items in the list
	Output: *items* is the list of items by their identifiers

API-EI-107	***SET_INDIVIDUAL_BY_INDEX***
	`INTEGER FUNCTION SET_INDIVIDUAL_BY_INDEX( IH, INDEX )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: INDEX`
	Description: Sets an individual
	Predecessors: component; entity type
	Input: *ih* is the integer handle for the API context
	index is the index of single entity
	Output: None
	Return: 0 if no error and non-zero otherwise

API-EI-108	***SET_INDIVIDUAL_BY_ID***
	`INTEGER FUNCTION SET_INDIVIDUAL_BY_ID( IH, ID )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: ID`
	Description: Sets an individual
	Predecessors: component; entity type
	Input: *ih* is the integer handle for the API context
	id is the identifier of single entity
	Output: None
	Return: 0 if no error and non-zero otherwise

9. Stress History

Number	Description

API-SH-100	***GET_ALL_S11***
	`SUBROUTINE GET_ALL_S11( IH, NUM_ENTITY, S11 )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(IN)               :: NUM_ENTITY`
	`   REAL*8, INTENT(OUT)               :: S11(NUM_ENTITY)`
	Description: Gets all stresses for the 11 component
	Predecessors: component; entity type; loadcase
	Input: *ih* is the integer handle for the API context
	num_entity is the number entities
	Output: *s11* is a vector of all stresses with component 11. They are ordered by entity index.

API-SH-101	***GET_ALL_S22***
	`SUBROUTINE GET_ALL_S22( IH, NUM_ENTITY, S22 )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(IN)               :: NUM_ENTITY`
	`   REAL*8, INTENT(OUT)               :: S22(NUM_ENTITY)`
	Description: Gets all stresses for the 22 component
	Predecessors: component; entity type; loadcase
	Input: *ih* is the integer handle for the API context
	num_entity is the number entities
	Output: *s22* is a vector of all stresses with component 22. They are listed by entity index.

API-SH-102	***GET_ALL_S33***
	`SUBROUTINE GET_ALL_S33( IH, NUM_ENTITY, S33 )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(IN)               :: NUM_ENTITY`
	`   REAL*8, INTENT(OUT)               :: S33(NUM_ENTITY)`
	Description: Gets all stresses for the 33 component
	Predecessors: component; entity type; loadcase
	Input: *ih* is the integer handle for the API context
	num_entity is the number entities
	Output: *s33* is a vector of all stresses with component 33. They are listed by entity index.

9. Stress History

Number	Description

API-SH-103	***GET_ALL_S12***
	`SUBROUTINE GET_ALL_S12( IH, NUM_ENTITY, S12 )`
	`INTEGER(KIND=8), INTENT(IN)     :: IH`
	`INTEGER, INTENT(IN)             :: NUM_ENTITY`
	`REAL*8, INTENT(OUT)             :: S22(NUM_ENTITY)`
	Description: Gets all stresses for the 12 component
	Predecessors: component; entity type; loadcase
	Input: *ih* is the integer handle for the API context
	num_entity is the number entities
	Output: *s12*is a vector of all stresses with component 11. They are listed by entity index.

API-SH-104	***GET_ALL_S13***
	`SUBROUTINE GET_ALL_S13( IH, NUM_ENTITY, S13 )`
	`INTEGER(KIND=8), INTENT(IN)     :: IH`
	`INTEGER, INTENT(IN)             :: NUM_ENTITY`
	`REAL*8, INTENT(OUT)             :: S13(NUM_ENTITY)`
	Description: Gets all stresses for the 13 component
	Predecessors: component; entity type; loadcase
	Input: *ih* is the integer handle for the API context
	num_entity is the number entities
	Output: *s13* is a vector of all stresses with component 13. They are listed by entity index.

API-SH-105	***GET_ALL_S23***
	`SUBROUTINE GET_ALL_S23( IH, NUM_ENTITY, S23 )`
	`INTEGER(KIND=8), INTENT(IN)     :: IH`
	`INTEGER, INTENT(IN)             :: NUM_ENTITY`
	`REAL*8, INTENT(OUT)             :: S23(NUM_ENTITY)`
	Description: Gets all stresses for the 23 component
	Predecessors: component; entity type; loadcase
	Input: *ih* is the integer handle for the API context
	num_entity is the number entities
	Output: *s23* is a vector of all stresses with component 23. They are listed by entity index.

9. Stress History

Number	Description

API-SH-106	***GET_LIST_S11***
	`SUBROUTINE GET_LIST_S11( IH, NUM_LIST, S11 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_LIST`
	`   REAL*8, INTENT(OUT)              :: S11(NUM_LIST)`
	Description: Gets stresses for the 11 component in the list
	Predecessors: component; entity type; loadcase; list
	Input: *ih* is the integer handle for the API context
	num_list is the number list items
	Output: *s11* is a vector of stresses with component 11 that are in the list. They are ordered based on the order of the input list.

API-SH-107	***GET_LIST_S22***
	`SUBROUTINE GET_LIST_S22( IH, NUM_LIST, S22 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_LIST`
	`   REAL*8, INTENT(OUT)              :: S22(NUM_LIST)`
	Description: Gets stresses for the 22 component in the list
	Predecessors: component; entity type; loadcase; list
	Input: *ih* is the integer handle for the API context
	num_list is the number list items
	Output: *s22*is a vector of stresses with component 22 that are in the list. They are ordered based on the order of the input list.

API-SH-108	***GET_LIST_S33***
	`SUBROUTINE GET_LIST_S33( IH, NUM_LIST, S33 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_LIST`
	`   REAL*8, INTENT(OUT)              :: S33(NUM_LIST)`
	Description: Gets stresses for the 33 component in the list
	Predecessors: component; entity type; loadcase; list
	Input: *ih* is the integer handle for the API context
	num_list is the number list items
	Output: *s33* is a vector of stresses with component 33 that are in the list. They are ordered based on the order of the input list.

9. Stress History

Number	Description

API-SH-109	***GET_LIST_S12***
	`SUBROUTINE GET_LIST_S12( IH, NUM_LIST, S12 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)             :: NUM_LIST`
	`   REAL*8, INTENT(OUT)             :: S12(NUM_LIST)`
	Description: Gets stresses for the 12 component in the list
	Predecessors: component; entity type; loadcase; list
	Input: *ih* is the integer handle for the API context
	num_list is the number list items
	Output: *s12*is a vector of stresses with component 12 that are in the list. They are ordered based on the order of the input list.

API-SH-110	***GET_LIST_S13***
	`SUBROUTINE GET_LIST_S13( IH, NUM_LIST, S13 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)             :: NUM_LIST`
	`   REAL*8, INTENT(OUT)             :: S13(NUM_LIST)`
	Description: Gets stresses for the 13 component in the list
	Predecessors: component; entity type; loadcase; list
	Input: *ih* is the integer handle for the API context
	num_list is the number list items
	Output: *s13* is a vector of stresses with component 13 that are in the list. They are ordered based on the order of the input list.

API-SH-111	***GET_LIST_S23***
	`SUBROUTINE GET_LIST_S23( IH, NUM_LIST, S23 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)             :: NUM_LIST`
	`   REAL*8, INTENT(OUT)             :: S23(NUM_LIST)`
	Description: Gets stresses for the 23 component in the list
	Predecessors: component; entity type; loadcase; list
	Input: *ih* is the integer handle for the API context
	num_list is the number list items
	Output: *s23* is a vector of stresses with component 23 that are in the list. They are ordered based on the order of the input list.

API-SH-112	***GET_INDIVIDUAL_S11***
	`SUBROUTINE GET_INDIVIDUAL_S11( IH, S11 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)             :: S11`
	Description: Gets stresses for the 11 component of an individual
	Predecessors: component; entity type; loadcase; individual
	Input: *ih* is the integer handle for the API context
	Output: *s11* is the value of stress component 11 for the individual

9. Stress History

Number	Description

API-SH-113	***GET_INDIVIDUAL_S22***
	`SUBROUTINE GET_INDIVIDUAL_S22( IH, S22 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)             :: S22`
	Description: Gets stresses for the 22 component of an individual
	Predecessors: component; entity type; loadcase; individual
	Input: *ih* is the integer handle for the API context
	Output: *s22* is the value of stress component 22 for the individual

API-SH-114	***GET_INDIVIDUAL_S33***
	`SUBROUTINE GET_INDIVIDUAL_S33( IH, S33 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)             :: S33`
	Description: Gets stresses for the 33 component of an individual
	Predecessors: component; entity type; loadcase; individual
	Input: *ih* is the integer handle for the API context
	Output: *s33* is the value of stress component 33 for the individual

API-SH-115	***GET_INDIVIDUAL_S12***
	`SUBROUTINE GET_INDIVIDUAL_S12( IH, S12 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)             :: S12`
	Description: Gets stresses for the 12 component of an individual
	Predecessors: component; entity type; loadcase; individual
	Input: *ih* is the integer handle for the API context
	Output: *s12* is the value of stress component 12 for the individual

API-SH-116	***GET_INDIVIDUAL_S13***
	`SUBROUTINE GET_INDIVIDUAL_S13( IH, S13 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)             :: S13`
	Description: Gets stresses for the 13 component of an individual
	Predecessors: component; entity type; loadcase; individual
	Input: *ih* is the integer handle for the API context
	Output: *s13* is the value of stress component 13 for the individual

API-SH-117	***GET_INDIVIDUAL_S23***
	`SUBROUTINE GET_INDIVIDUAL_S23( IH, S23 )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)             :: S23`
	Description: Gets stresses for the 23 component of an individual
	Predecessors: component; entity type; loadcase; individual
	Input: *ih* is the integer handle for the API context
	Output: *s23* is the value of stress component 23 for the individual

9. Stress History

Number	Description

API-SH-118	***GET_ALL_STRESS***
	`SUBROUTINE GET_ALL_STRESS( IH, NUM_ENTITY, STRESS )`
	`    INTEGER(KIND=8), INTENT(IN)      :: IH`
	`    INTEGER, INTENT(IN)               :: NUM_ENTITY`
	`    REAL*8, INTENT(OUT)               :: STRESS(6,NUM_ENTITY)`
	Description: Gets all stresses for all components
	Predecessors: component; entity type; loadcase
	Input: *ih* is the integer handle for the API context
	num_entity is the number entities
	Output: *stress* is an array with all components of the stress tensor. Stress components are ordered as: 11, 22, 33, 12, 13, 23.

API-SH-119	***GET_LIST_STRESS***
	`SUBROUTINE GET_LIST_STRESS( IH, NUM_LIST, STRESS )`
	`    INTEGER(KIND=8), INTENT(IN)      :: IH`
	`    INTEGER, INTENT(IN)               :: NUM_LIST`
	`    REAL*8, INTENT(OUT)               :: STRESS(6,NUM_LIST)`
	Description: Gets stresses for all components in the list
	Predecessors: component; entity type; loadcase; list
	Input: *ih* is the integer handle for the API context
	num_list is the number of list items
	Output: *stress* is an array with all components of the stress tensor. Stress components are ordered as: 11, 22, 33, 12, 13, 23.

API-SH-120	***GET_INDIVIDUAL_STRESS***
	`SUBROUTINE GET_INDIVIDUAL_STRESS( IH, STRESS )`
	`    INTEGER(KIND=8), INTENT(IN)      :: IH`
	`    INTEGER, INTENT(IN)               :: NUM_LIST`
	`    REAL*8, INTENT(OUT)               :: STRESS(6)`
	Description: Gets stresses for all components of an individual
	Predecessors: component; entity type; loadcase; individual
	Input: *ih* is the integer handle for the API context
	Output: *stress* is an array with all components of the stress tensor. Stress components are ordered as: 11, 22, 33, 12, 13, 23.

10. Temperature History

Number	Description

API-TH-100	***GET_ALL_TEMPERATURE***
	`SUBROUTINE GET_ALL_TEMPERATURE( IH, NUM_ENTITY, TEMPERATURE )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_ENTITY`
	`   REAL*8, INTENT(OUT)              :: TEMPERATURE(NUM_ENTITY)`
	Description: Gets all temperature values
	Predecessors: component; entity type; loadcase
	Input: *ih* is the integer handle for the API context
	num_entity is the number entities
	Output: *temperature* is an array that contains all temperatures

API-TH-101	***GET_LIST_TEMPERATURE***
	`SUBROUTINE GET_LIST_TEMPERATURE( IH, NUM_LIST, TEMPERATURE )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_LIST`
	`   REAL*8, INTENT(OUT)              :: TEMPERATURE(NUM_LIST)`
	Description: Gets all temperature values in the list
	Predecessors: component; entity type; loadcase; list
	Input: *ih* is the integer handle for the API context
	num_list is the number of list items
	Output: *temperature* is an array that contains all temperatures

API-TH-102	***GET_INDIVIDUAL_TEMPERATURE***
	`SUBROUTINE GET_INDIVIDUAL_TEMPERATURE( IH, TEMPERATURE )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   REAL*8, INTENT(OUT)              :: TEMPERATURE`
	Description: Gets the temperature value of an individual
	Predecessors: component; entity type; loadcase; individual
	Input: *ih* is the integer handle for the API context
	Output: *temperature* is a floating point number with the temperatures

11. General Datatype History

Number	Description

API-GD-100	***SET_DATA_TYPE***
	`INTEGER FUNCTION SET_DATA_TYPE( IH, DNUM )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(IN)              :: DNUM`
	Description: Sets the data type number
	Predecessors: component; entity type
	Input: *ih* is the integer handle for the API context
	data_type indicates the specific data type
	Output: None
	Return: 0 if no error and non-zero otherwise
	Note: The data type commands provide a general way to list data that does not have a specific type at this time.

API-GD-101	***GET_DATA_TYPE_NAME***
	`SUBROUTINE GET_DATA_TYPE_NAME( IH, NAME )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=80), INTENT(OUT)   :: NAME`
	Description: Gets the data type name
	Predecessors: component; entity type; data type
	Input: *ih* is the integer handle for the API context
	Output: *name* indicates the specific name of the data type

API-GD-102	***GET_DATA_TYPE_COLUMNS***
	`SUBROUTINE GET_DATA_TYPE_COLUMNS( IH, NUM_COLUMNS )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(OUT)             :: NUM_COLUMNS`
	Description: Gets the number of data columns
	Predecessors: component; entity type; data type
	Input: *ih* is the integer handle for the API context
	Output: *num_columns* indicates the number of data type columns

API-GD-103	***GET_DATA_TYPE_UNITS***
	`SUBROUTINE GET_DATA_TYPE_UNITS( IH, UNITS )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   CHARACTER(LEN=20), INTENT(OUT)   :: UNITS`
	Description: Gets the number of data units
	Predecessors: component; entity type; data type
	Input: *ih* is the integer handle for the API context
	Output: *units* indicates the units of the data type. It is assumed that all columns of a data type will have the same units

11. General Datatype History

Number	Description

API-GD-104	***SET_DATA_COLUMN***
	`INTEGER FUNCTION SET_DATA_COLUMN( IH, DATA_COLUMN )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)             :: DATA_COLUMN`
	Description: Sets the data type column
	Predecessors: component; entity type; data type
	Input: *ih* is the integer handle for the API context
	data_column indicates the data column number
	Output: None
	Return: 0 if no error and non-zero otherwise

API-GD-105	***GET_ALL_DATA_VALUES***
	`SUBROUTINE GET_ALL_DATA_VALUES( IH, NUM_ENTITY, DATA_VALUES )`
	`   INTEGER(KIND=8), INTENT(IN)     :: IH`
	`   INTEGER, INTENT(IN)             :: NUM_ENTITY`
	`   REAL*8, INTENT(OUT)             :: DATA_VALUES (NUM_ENTITY)`
	Description: Gets all data values
	Predecessors: component; entity type; loadcase; data type; data column
	Input: *ih* is the integer handle for the API context
	num_entity is the number entities
	Output: *data_values* is an array that contains all data values

API-GD-106	***GET_LIST_DATA_VALUES***
	`SUBROUTINE GET_LIST_DATA_VALUES ( IH, NUM_LIST, DATA_VALUES )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   INTEGER, INTENT(IN)              :: NUM_LIST`
	`   REAL*8, INTENT(OUT)              :: DATA_VALUES(NUM_LIST)`
	Description: Gets all data values in the list
	Predecessors: component; entity type; loadcase; list; data type; data column
	Input: *ih* is the integer handle for the API context
	num_list is the number of list items
	Output: *data_values* is an array that contains the data values

API-GD-107	***GET_INDIVIDUAL_DATA_VALUES***
	`SUBROUTINE GET_INDIVIDUAL_DATA_VALUES( IH, DATA_VALUES )`
	`   INTEGER(KIND=8), INTENT(IN)      :: IH`
	`   REAL*8, INTENT(OUT)              :: DATA_VALUES`
	Description: Gets the data value of an individual
	Predecessors: component; entity type; loadcase; individual; data type; data column
	Input: *ih* is the integer handle for the API context
	Output: *data_value* is a floating point number with the data values

APPENDIX E—STRIP-YIELD MODELING OF LOAD-TIME-TEMPERATURE (PLASTICITY, CREEP, AND RELAXATION) EFFECTS ON FATIGUE CRACK GROWTH AND CLOSURE BEHAVIOR IN ENGINE MATERIALS

E.1 PREFACE

This project began an effort to incorporate load-time-temperature effects into a strip-yield model, such as FAtigue crack growth STructual ANalysis computer program (FASTRAN). The framework for what modifications and what creep/relaxation test data are needed in the strip-yield model have been established. For fatigue crack growth under mechanical loading, the crack tip opening displacements (CTOD) parameter has been found to be directly proportional to the amount of crack extension under cyclic loading. Past research on creep crack growth has also used the CTOD as the crack-driving parameter. Whether the large amount of crack extension under load-time-temperature effects can be directly related to CTOD has not been established. The research in the past year has identified critical test data that need to be determined. First, the creep and relaxation test data under extreme stress and strain states that develop in the crack-front region has not been measured on an engine material, such as Inconel-718 (project material). The author has conducted some creep/relaxation tests on an aluminum alloy under extreme stress and strain states and found creep and relaxation equations to fit the experimental test data. Currently, this behavior would have to be assumed/extrapolated from lower stress and strain states on engine materials. In addition, the measurement of CTOD during load-time-temperature profiles need to be measured to see if the large amount of crack extension could be characterized by the CTOD parameter.

During the past year, the author has analyzed a large number of materials under a wide variety of load-time-temperature profiles and found that the linear-superposition model worked fairly well using linear-elastic fracture mechanics (LEFM) concepts to model mechanical and time-dependent crack growth for creep-brittle materials. Efforts to model CTOD under creep and relaxation conditions using the strip-yield model may be developed by the author as independent research.

E.2 PROJECT BACKGROUND

Aircraft engine components are designed to operate in high-temperature environments, with high-thermal gradients and mechanical loads. Repeated engine startup and shut down operations subject components to cyclic loading, which are generated both thermally and mechanically. Such complex load-time-temperature histories may lead to material damage from environmental degradation, fatigue, and creep. These thermo-mechanical fatigue (TMF) cycles may cause microstructural material damage to the components, which may lead to fatigue-crack nucleation, crack growth, and fracture. Advanced high-temperature fatigue-life prediction methodologies are needed to decide when to replace engine components to increase reliability and service life and to reduce maintenance costs. Predicting the lives of components subjected to TMF loading continues to be a great challenge [E-1]. In the past 5 decades, much work has been done in modeling fatigue-crack nucleation, crack growth, and fracture in engine materials.

Many years ago, McEvily and Wells [E-2] proposed to model time-dependent cracking with a Dugdale-type model, but nothing was really developed. Shortly thereafter, in the 1977–1978 timeframe, Vitek [E-3, E-4], Riedel [E-5], and Ewing [E-6] developed creep-crack incubation and growth models based on the Dugdale [E-7] or Bilby et al. [E-8] models. These researchers used the CTOD as the crack-growth criterion. All of these models, however, were under constant loading conditions and modeled the effects of creep deformation on cracking behavior. In the past 3 decades, no one has developed a fatigue-creep strip-yield model under cyclic loading. Therefore, the proposed work herein would be the first attempt to develop such a model.

In the past few years, aircraft engine manufacturers have expressed an interest in developing a fatigue-creep strip-yield model under cyclic loading. Though some finite-element analyses [E-9] may be available to model some of the complex material behavior under these load-time-temperature conditions, there is a desire to develop a model that could be applied in a probabilistic manner, which requires faster computations. But the first objective will be to develop an approach that closely models the physics of the process and then to develop a more rapid simulation. The task is very difficult and complex because tensile (and compressive) stress-strain properties, creep properties, and the time-dependent crack-growth properties of engine materials are functions of local strain rates (frequencies), temperature, time-of-load application (hold times), and environment. Therefore, the project will be composed of two phases: 1) initial model development with plasticity and creep deformations, and 2) load-time-temperature data analysis with model refinements. The second phase will be initiated at a later stage, so that load-time-temperature data can be analyzed with the developing model.

For ambient temperature conditions, the plasticity-induced crack-closure model has been very successful in modeling and predicting fatigue behavior (small-crack theory) and fatigue-crack growth under both constant- and variable-amplitude loading for aircraft materials [E-10]. Much work still needs to be done on roughness- and debris-induced crack-closure behavior, accounting for their influence on fatigue and crack-growth behavior, and to incorporate these mechanisms into the strip-yield model. Based on a preliminary review of the literature, the crack-closure concept has not been applied very widely to TMF conditions. It may be caused by the lack of crack-growth and closure modeling under these conditions.

E.3 PROJECT DESCRIPTION

E.3.1 INTRODUCTION

The proposed research is composed of two phases. Phase I is a preliminary study to review the literature on load-time-temperature effects on engine materials and to develop a prototype load-time-temperature dependent plasticity-creep-relaxation model for the life-prediction code, FASTRAN [E-11, E-12]. The latest code will be revised to incorporate both mechanical and creep deformations on crack-surface displacements and used to calculate the crack-closure (opening) behavior under various load-time-temperature histories. This report describes the completed Phase I effort.

Phase II would review TMF life-prediction models for engine materials, study the various crack-tip parameters used to correlate and predict time-dependent creep-fatigue crack growth, develop an improved model to predict crack closure and crack growth under various load-time-temperature histories, and compare these calculations with test data. Phase II efforts are left for future research and are not within the scope of this project.

E.3.2 RESEARCH PLAN

The research centered on modifying the life-prediction code, FASTRAN [E-11, E-12], for plasticity and creep deformations around the crack front and in the wake of the crack (see figure E-1). The revised code will be FASTRAN Version 6.0. Under the maximum applied stress, S_{max}, at temperature, T_1, with a hold-time, t_1, the tensile plastic and creep deformations in front of the crack tip will control the plastic- and creep-zone regions while the material in the plastic wake (residual plastic deformations) will undergo creep recovery (relaxation). At the minimum applied stress, S_{min}, at temperature, T_2, with a hold-time, t_2, the material will undergo compressive plastic and creep deformations. The crack-opening stress, S_o, will then be calculated from the final contact stresses in the plastic wake. The effective stress-intensity factor range (ΔK_{eff}) will be calculated for the next tensile load excursion.

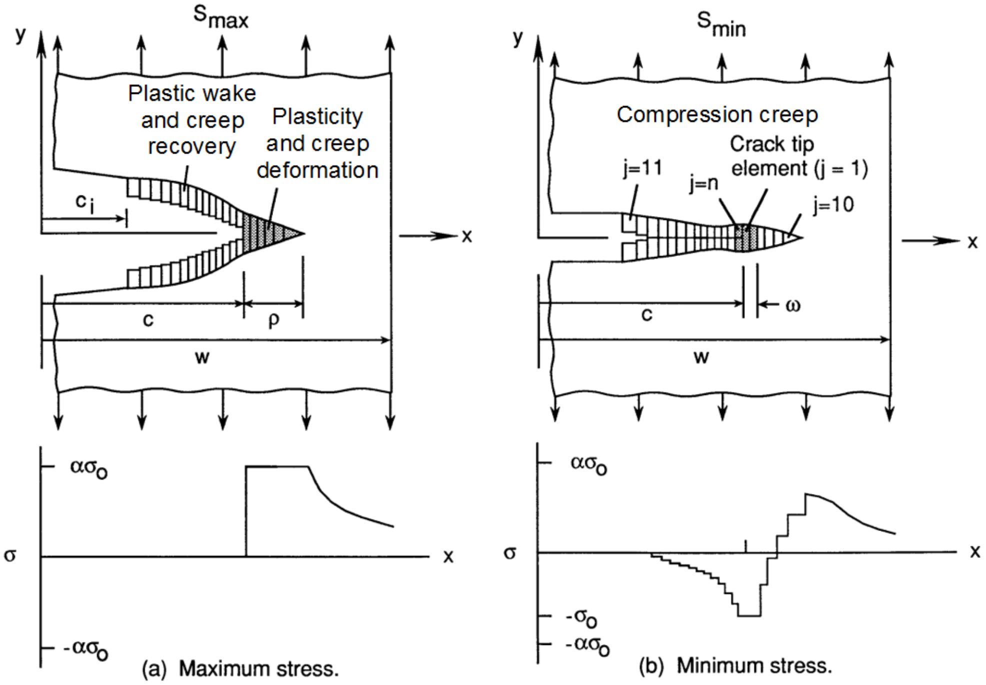


Figure E-1. Strip-yield model for plastic and creep deformations under cyclic loading

Under the various load-time-temperature histories, the ΔK_{eff} values will be a function of stress amplitude and mean stress (R), loading frequencies (f), hold times (Δt), temperatures (T), and constraint (α)—plane strain or plane stress. Whether these values control or have any bearing on creep-fatigue crack growth is a question to be addressed. Further discussions will be made on traditional methods to combine mechanical (cyclic) crack growth and time-dependent creep-crack growth.

Recently, an application of the FASTRAN crack-closure model based on the Dugdale strip-yield model [E-13] was used to evaluate the effective stress-intensity-factor range (ΔK_{eff}) for characterizing crack-tip damage. Figure E-2 shows calculated crack-tip-opening displacements for a crack in an infinite plate under remote uniform applied stress at R = 0 loading (the CTOD values are proportional to the cyclic plastic strains in the crack-tip region). The initial crack length was 2 mm, and the crack was grown in the model until the length had reached 30 mm, and plasticity-induced crack closure behavior had developed and stabilized (constant crack-opening stress, S_o). The solid curves show the loading and unloading records during one cycle of loading. The calculated crack-opening stress is shown by the lower solid symbol along the loading trace. The smaller solid (upper) symbol shows the applied stress level when the crack-tip element went from compression to tensile stress and the beginning of element yielding. The enclosed area is the cyclic hysteresis energy for crack growth, $(W_p)_{cg}$. A second simulation was then made with an initial crack length of 30 mm and applying the same cyclic stress. The dashed curves show the loading and unloading traces with no crack growth or crack closure, but with plastic yielding at the crack tip. The cyclic hysteresis energy with no crack growth, $(W_p)_{ncg}$ is the area traced by the dashed curves. For small-scale yielding, the strain-energy-release rate, G, and the J-integral, are equal to K^2/E. Assuming that the ratio of the cyclic hysteresis energies is proportional to $(\Delta K_{eff}/\Delta K)^2$, the middle circular symbol shows the value of crack-opening stress to satisfy the energy ratio. The ΔK_{eff} values calculated from the model using the contact stresses and the hysteresis energies were within 2%. The current method of calculating crack-opening stress levels from contact stresses, correctly partitions the hysteresis energies. Therefore, the ΔK_{eff} crack-tip damage parameter is equivalent to using cyclic CTOD/hysteresis energies.

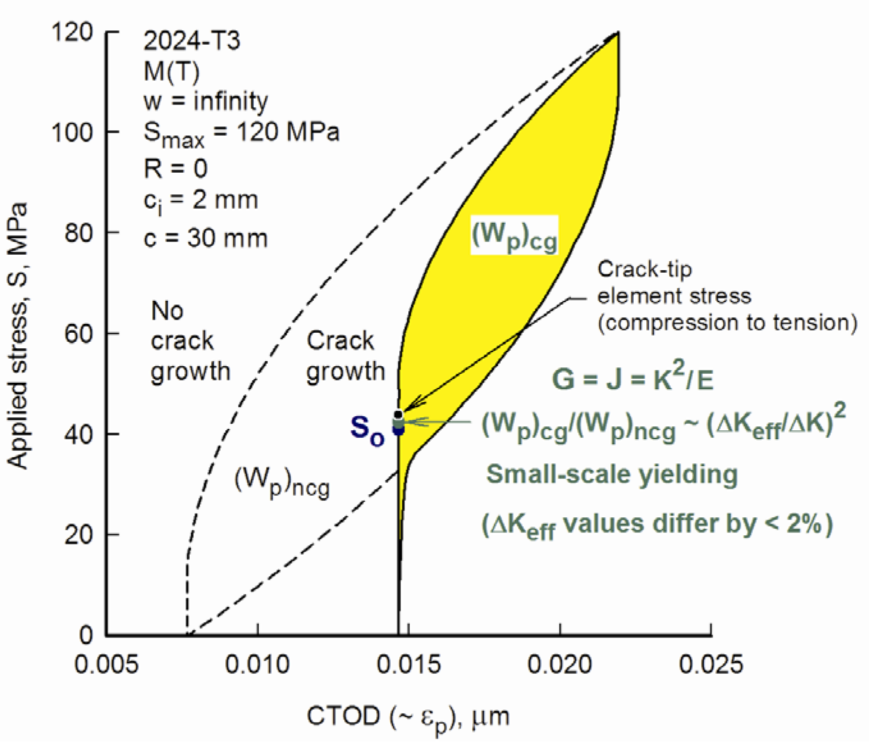


Figure E-2. Cyclic crack-tip-opening displacements with and without fatigue-crack closure

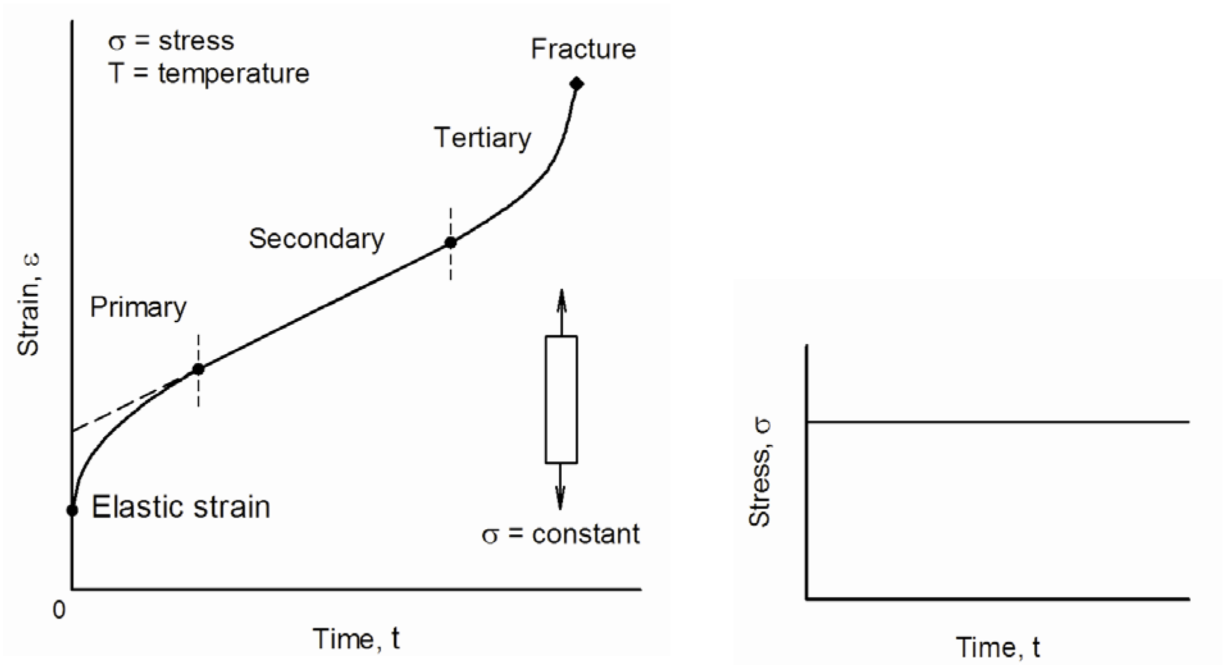
As the very early creep models, which are based on the strip-yield model [E-3–E-6], used the CTOD as a creep-crack-growth criterion, the use of this parameter may be appropriate for TMF conditions. The FASTRAN code calculates the cyclic CTOD that includes the influence of crack closure (see figure E-2). Therefore, a nature quantity calculated from the model may be used as the creep-fatigue crack-growth criterion.

In addition, the FASTRAN code uses a cyclic-plastic-zone corrected effective stress-intensity factor range, which is proportional to the cyclic J-integral. This quantity has also been found to be useful for large-scale plastic deformations and low-cycle fatigue conditions. In addition, CTOD is directly proportional to the J-integral; and the cyclic CTOD would be proportional to the cyclic J-integral. The formulation of the C*-integral, used for creep-ductile materials, comes from a time-dependent integral, like the J-integral. Therefore, the cyclic CTOD under load-time-temperature conditions may be useful for both creep-brittle and creep-ductile materials.

Creep and Relaxation Behavior of Metals. The mechanism of time-dependent deformation [E-14, E-15] is analogous to plastic deformation and the total strain is given by:

$$\varepsilon = \varepsilon_e + \varepsilon_p + \varepsilon_c \qquad \text{(E-1)}$$

where ε_e is the elastic strain, ε_p is the plastic strain, and ε_c is the creep strain. When a tensile bar is subjected to a constant applied stress with time, the initial strain is the elastic strain, and then the material develops a creep strain. The creep process is composed of three regions: 1) primary, 2) secondary, and 3) tertiary (see figure E-3 [E-16]). Primary region is at the early stage of loading where the creep rate decreases rapidly with time, as dislocations slip. As the dislocation density saturates, the secondary creep region is formed. Some researchers have ignored the primary-creep stage and extrapolated back to zero time using the same strain rate as measured in the secondary stage (see dashed line). The secondary region is the steady-state creep stage followed by a rapid increase in strain (tertiary stage). The tertiary creep region is observed when grain boundaries slide and voids form and coalesce, leading to fracture (the previous Dugdale-type models [E-3–E-6] assumed only a secondary creep stage). Tertiary creep occurs rapidly at high stresses, such as around crack tips. The tertiary creep stage is where material damage evolves.

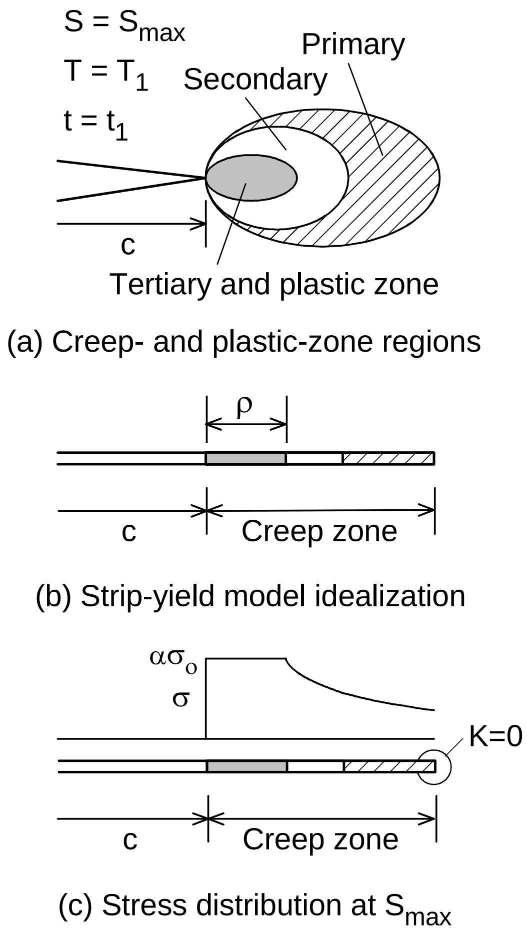


(a) Creep Strain

(b) Stress-Time History

Figure E-3. Creep strain as a function of time at constant applied stress

Because the crack-tip stresses in the plastic-zone region are larger than the flow stress (average of yield stress and ultimate tensile strength) of the material and are very high in the region surrounding the crack tip, all creep stages may be needed to model the deformations around cracks in engine materials (see figure E-4(a)). Here the cracked body is subjected to an applied stress, S_{max}, at time, t_1, and at temperature, T_1. To account for the secondary and tertiary creep damage behavior of materials, the Kachanov-Rabotnov [E-17, E-18] equations for creep rate and damage evolution will be studied. Several authors have developed specialized variations based on this fundamental formulation [E-9, E-19] using finite-element analyses.



(a) Creep- and plastic-zone regions

(b) Strip-yield model idealization

(c) Stress distribution at S_{max}

Figure E-4. Creep- and plastic-zone regions around crack tip and model idealizations

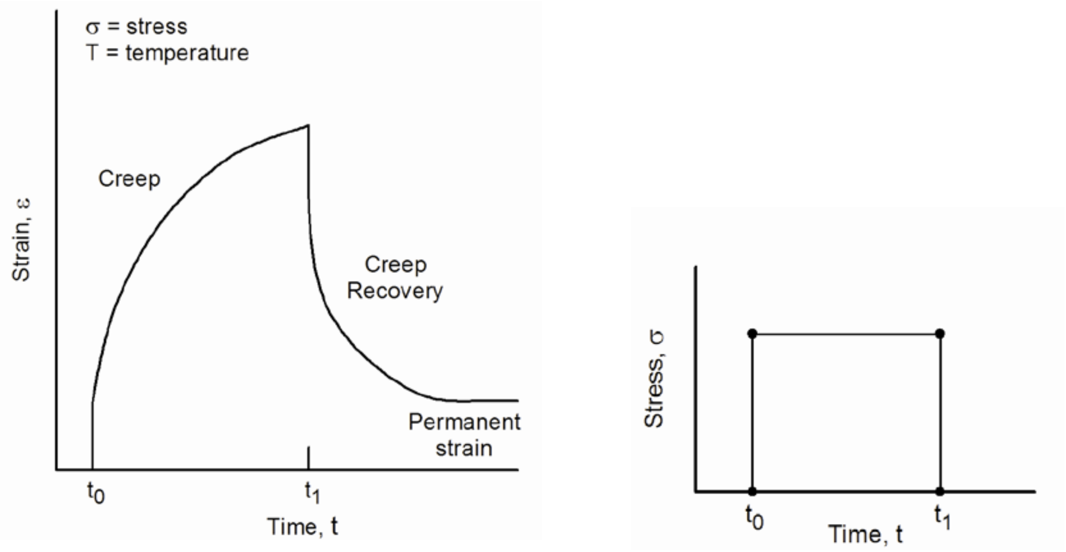
In the strip-yield model idealization (see figure E-4(b)), all deformations due to plasticity and creep would be concentrated along a linear path ahead of the crack tip. Deformations in the plastic- and creep-zone regions are only plastic and creep strains; elastic deformations are only in the surrounding elastic region. However, elastic deformations will be influenced by changes in the modulus of elasticity (E) due to time and temperature exposure. Deformations in the plastic-zone region are controlled by the deformations of the surrounding elastic material, which would also control the amount of creep deformation (tertiary behavior) in the plastic-zone region. Away from the plastic-zone region, both secondary and primary creep deformations will model the material behavior. Like in the original strip-yield model, the Dugdale finiteness condition ($K = 0$) would act on the fictitious crack tip at the end of the creep zone (stresses are finite and less than or equal to $\alpha\sigma_o$), as shown in figure E-4(c). Therefore, the element stresses in the plastic- and creep-zone regions would be determined from displacement compatibility with two constraints: 1) maximum stress equal to $\alpha\sigma_o$, and 2) $K = 0$. The second constraint ensures that the stress at the fictitious crack tip is finite and continuous (no singularity). As the primary- and secondary-creep strains relieve the element stresses, the plastic-zone region (ρ) will increase in size to maintain $K = 0$.

Newman [E-20] used a "ligament" model and displacement compatibility to determine the extent of yielding in front of a crack, similar to the Dugdale-type model, but stresses (less than the flow stress) were determined outside of the plastic-zone region in the elastic region. This approach would be well suited for determining the yielded elements in the plastic region and stresses on elements further from the crack tip, which would develop creep strains (in addition, this approach may allow the model to address a nonlinear or bilinear stress-strain curve behavior instead of perfectly-plastic material behavior). Ideally, the creep zone should extend over the complete cracked body or along the complete net-section, but the major influence of the creep zone on crack-surface deformations would be very local. Therefore, the creep-zone size will be selected as several plastic-zone sizes or some percentage of the crack length. Numerical studies will be made to find an appropriate size.

In addition, the maximum number of elements in the strip-yield model would have to be increased to accommodate the creep-zone elements. Currently, there is a maximum of 60 elements allowed in the model, 20 elements in the plastic zone, and 40 elements allowed in the wake. Because of element lumping away from the crack tip, applications of the model have resulted in approximately 33–38 elements in the wake. The revised model may return to 10 elements in the plastic zone, like the original model, and allow 20 additional elements for the secondary and primary creep regions.

As the crack grows in the model, residual plastic- and creep-deformations will be left along the crack surface in the wake (see figure E-1). The deformations in the wake are only due to plastic or creep strains. In time, some of the creep strains will recover (relaxation) and approach a permanent creep strain.

If the stress on a bar is applied at time (t_0) and released at time (t_1) before creep rupture occurs, an immediate elastic recovery occurs, followed by a period of slow recovery (reduction in creep strain) (see figure E-5). Materials in most cases do not recover to their original size and a permanent creep deformation remains, very much like residual plastic deformations. The magnitude of the permanent deformation from creep strains depends on the length of time, amount of stress applied, and temperature.



(a) Creep-Recovery Strain (b) Stress-Time History

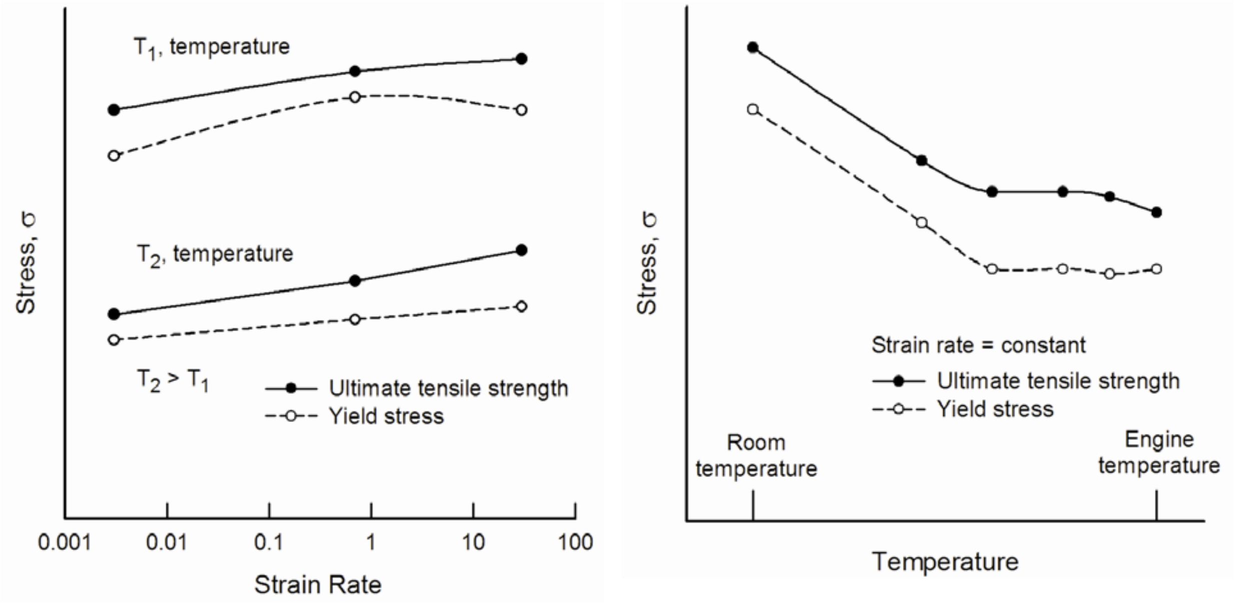
Figure E-5. Creep-strain recovery after one applied stress cycle from time t_0 to t_1

In the strip-yield model (see figure E-1), the material in the plastic wake will have failed under the mechanical loads at the end of the tertiary creep stage with a large amount of plastic deformation. How the deformations on the elements in the plastic wake behave with time and temperature are not well understood; this will require a great deal of study.

At the minimum applied stress, S_{min}, on the strip-yield model (see figure E-1(b)), the elements in the plastic wake will be under compressive stresses and undergo compressive creep [E-21]. While the intact elements in the plastic- and creep-zone regions will be under compressive stresses near the crack tip, and yield in compression, but tensile stresses would develop away from the crack. Depending on the magnitude of the minimum applied stress, the elements in the creep zone will, generally, have low tensile stresses. Therefore, the creep behavior under compressive stresses will also have to be modeled.

E.3.2.1 Strain Rate and Temperature Effects on Metals

The mechanical properties of metals (e.g., yield stress, ultimate tensile strength, and modulus of elasticity) are functions of temperature and strain rate [E-22]. Some typical properties for a titanium alloy (Ti-6Al-4V) as a function of strain rates over five orders of magnitude and temperatures from room to engine temperatures are shown in figure E-6. This alloy is not very sensitive to strain rates, but other engine alloys show larger variations, while the operational temperature had a very strong effect. The temperature reduced the yield stress and tensile strength by a factor of two from room to engine temperatures. It is suspected that for many high-strength engine materials, the differences between the yield stress and ultimate tensile strength are fairly small, so that elastic-perfectly-plastic material behavior with a constant flow stress could be assumed (the flow stress is the average between the yield stress and ultimate tensile strength).



(a) Strain Rate

(b) Temperature

Figure E-6. Typical strain-rate and temperature effects on tensile properties for an engine material

One of the first tasks will be to modify the code to input the material tensile properties (e.g., yield stress, ultimate tensile strength, and modulus) as a function of strain rate and temperature. Because of the variable nature of how strain rate and temperature affects these properties, a table-lookup would be the best option with either a linear or nonlinear interpolation scheme to determine the properties at a particular strain rate and temperature. Extrapolations outside of data limits would not be allowed.

E.3.2.2 Load-Time-Temperature Histories

Current load (or stress) input to FASTRAN can be done with several options. The most general option is to use NFOPT = 8, which is a string of number with S_{max1}, S_{min2}, S_{max2}, …, S_{minn}, where n is the last applied stress input. It is assumed that the initial applied stress $S_{min1} = 0$. There is basically no limit on the size of the string of numbers (n). This option can be used to easily apply constant-amplitude loading or any other type of variable-amplitude loading.

A second task will be to modify the code to input load, time, and temperature histories. It is proposed that the initial input to FASTRAN will use NFOPT = 8, but modified to input "applied stress, time, hold time, and temperature" (see figure E-7). It is assumed that the initial applied stress, $S_{min1} = 0$, at time equal zero with no hold time and at room temperature. The stress (or strain) rate and frequency of loading can be determined from the applied stress and time differences.

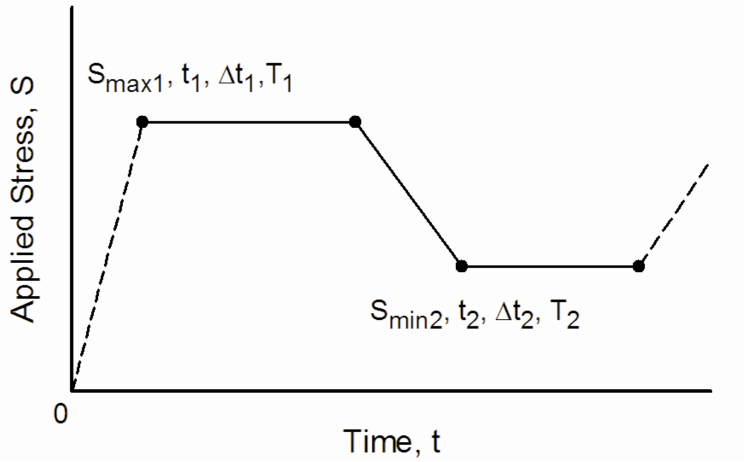

Applied Stress, S
S_{max1}, t_1, Δt_1, T_1
S_{min2}, t_2, Δt_2, T_2
0
Time, t

Figure E-7. Typical stress-time-temperature history input for the life-prediction code

The FASTRAN code will be modified to calculate the plastic-zone size as a function of temperature and simulate crack growth under constant-amplitude loading (no hold times) with only a temperature change at a specified time, t_c. This stress, time, and temperature history is shown in figure E-8. In addition, the fatigue-crack-growth-rate data, like the current ΔK_{eff}-rate table-lookup curve, will be input for various temperatures with either a linear or nonlinear interpolation scheme to determine the properties at a particular temperature.

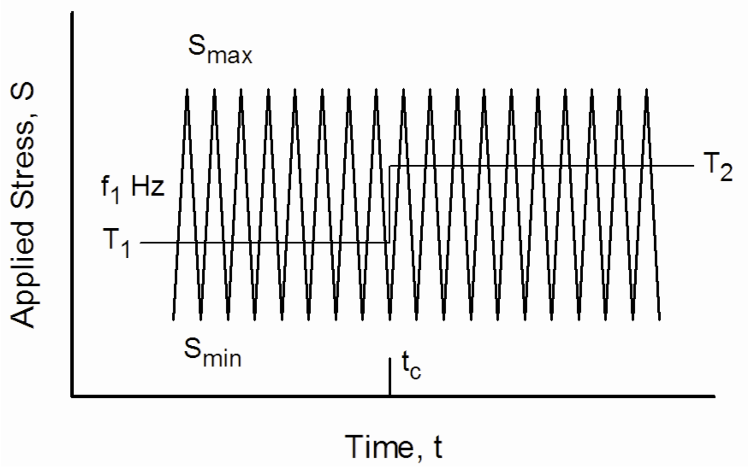


Figure E-8. Constant-amplitude loading at a given frequency with temperature change, T_1 to T_2, at time, t_c

The influence of the changing flow properties with temperature on crack-closure (opening) behavior can be studied with the simple load-time-temperature history (see figure E-8). It is suspected that crack-growth acceleration and retardation will occur because of the changing flow properties and plastic-zone sizes, similar to overloads and under loads.

One of the objectives of the project is to work towards predicting fatigue and creep-crack growth under a more general load-time-temperature history (see figure E-9). This history combined high-frequency cyclic loading with various hold times at maximum and minimum applied stresses at constant temperature. Tests were conducted on Inconel-718 at 650ºC using compact, C(T), specimens; and conducted under K-control instead of load control [E-23]. FASTRAN has a K-control option for C(T) and middle-crack-tension, M(T), specimens.

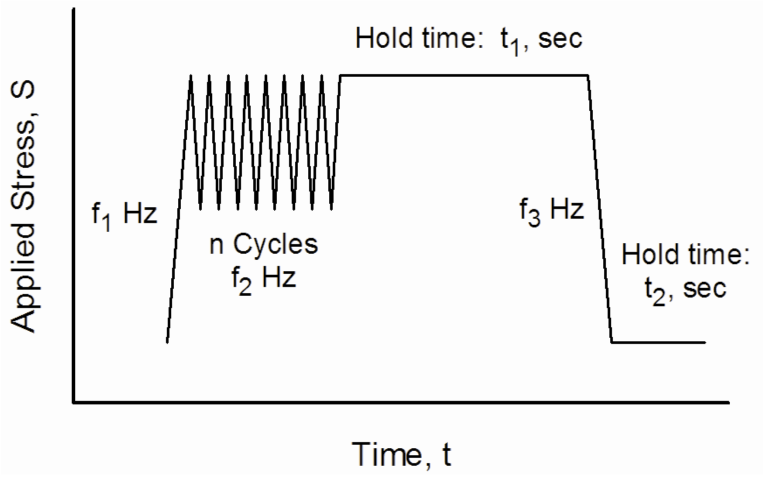


Figure E-9. Typical stress-time history at a constant temperature with cyclic loading at a certain frequencies and hold times at maximum and minimum loads

E.3.2.3 Fatigue- and Creep-Crack-Growth Criteria

The ASTM standard test method for creep-fatigue crack-growth testing [E-24] uses both the stress-intensity factor (ΔK) and the C_t-integral (related to C*-integral, like a time-dependent J-integral) to characterize crack growth in creep-brittle and creep-ductile materials, respectively. Models for creep-fatigue interaction can be separated by ones that account for hold-time effects and those that only apply to continuous cycling situations. The objective of these models is to provide the ability to predict time-dependent crack-growth effects. Most researchers [E-23–E-26] have used the linear superposition method to calculate the crack-growth increment or crack-growth rate for a combined loading of fatigue cycles and creep (hold time) effects. The governing equation for such a model is:

$$dc/dN = (dc/dN)_{fatigue} + L\ (dc/dt)_{creep} \qquad \text{(E-2)}$$

where L is a loading function related to frequencies and hold times. For creep-brittle materials, both terms in equation (E-2) have been uniquely related to ΔK; for creep-ductile materials, the second term in equation (E-2) has been related to the C_t-integral.

As the very early creep models, based on the strip-yield model [E-3–E-6], used the CTOD as a creep-crack-growth criterion, the use of this parameter may be appropriate for TMF conditions. The FASTRAN code calculates the cyclic CTOD that includes the influence of crack closure (see figure E-2). Therefore, a nature quantity calculated from the model (i.e., cyclic CTOD) may be used as the crack-growth criterion for both mechanical and creep conditions. For small-scale yielding, the cyclic CTOD reduces to a function related to $(\Delta K)^2$.

As previously mentioned, the FASTRAN code also uses a cyclic-plastic-zone corrected effective stress-intensity factor range, which is proportional to the cyclic J-integral. This quantity has also been found to be useful for large-scale plastic deformations and low-cycle fatigue conditions. In addition, CTOD is directly proportional to the J-integral, and the cyclic CTOD would be proportional to the cyclic J-integral. The formulation of the C*- or C_t-integrals, used for creep-ductile materials, comes from a time-dependent integral, like the J-integral. Therefore, the cyclic CTOD under mechanical loading and creep conditions may be useful for both creep-brittle and creep-ductile materials.

E.3.3 PHASE I PROGRESS

E.3.3.1 FASTRAN MODELING MODIFICATIONS

FASTRAN version 5.42 [E-27] has been selected to become FASTRAN version 6.0 with the stress-time-temperature option denoted as NFOPT = 11 (NFOPT = 0–10 are only for the plasticity model, as currently programmed). This will allow the current FASTRAN code to be used along with the plasticity-creep-relaxation version (however, if programming the strip-yield plasticity-creep-relaxation model becomes too involved, then this model may be a standalone code, only for stress-time-temperature loading with material plasticity, creep, and relaxation behaviors). All types of loading (e.g., constant-amplitude, dwell loading, and general spectrum loading) are to be input in the same manner using a stress-time-temperature format. A typical trapezoidal stress sequence is shown in figure E-10.

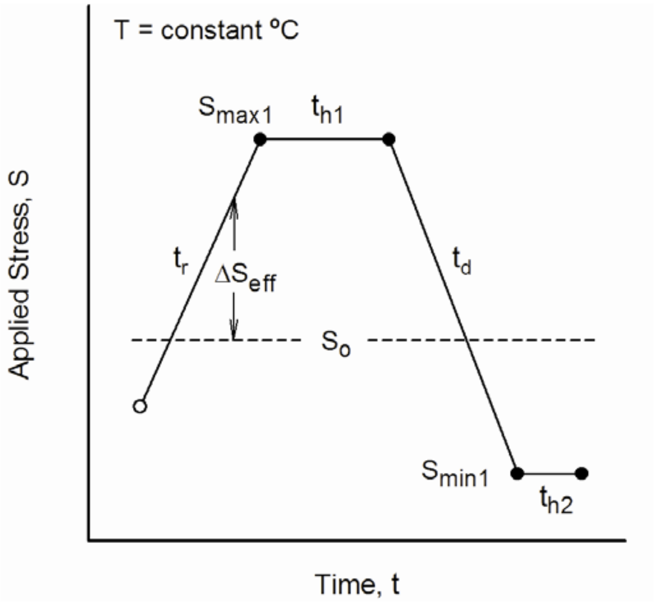


Figure E-10. Typical stress-time-temperature history input for life-prediction code

It is assumed that the initial applied stress S_{min} = 0 at time equal zero with no hold time, and at the test temperature. The stress (or strain) rate, frequency of loading, ramp, and hold times can be determined from the applied stress and time differences. The loading history may be either trapezoidal or sinusoidal, and the temperature may change for each block of constant-amplitude loading cycles, if needed. The loading sequence may be composed of up to 15 different blocks of constant-amplitude loading, but the temperature is held constant during each block of loading.

The crack-opening stress is S_o and $\Delta S_{eff} = S - S_o$, which is the instantaneous effective stress (see figure E-10). Time-dependent damage during the rise time, t_r, is computed from the creep-crack growth relation by integrating from $S = S_o$ to $S = S_{max1}$. This may be the first time that crack-closure theory has been proposed to influence creep-crack growth under trapezoidal loading. Time-dependent damage during the hold time, t_{h1}, and cyclic crack growth uses the traditional $\Delta S_{eff} = S_{max1} - S_o$.

Efforts to modify the code to input applied stress, time, and temperature histories are progressing. In addition, the tensile properties (e.g., modulus of elasticity, yield stress, and ultimate tensile strength) are now input in a table format (including strain-hardening behavior is beyond the scope of the current project). The flow stress, σ_o, is the average between the yield stress and ultimate tensile strength. The flow stress is calculated as a function of temperature (extrapolation beyond the input properties are not allowed). During execution, the flow stress may change as a function of temperature history. The plastic elements in the model will retain the plastic strains as temperature changes during load applications, similar to elastic-plastic finite element models. Plastic deformation can only be modified by further yielding of the elements (the current code does not use the time and temperature information but further code modifications will allow the use of these data in calculating plastic-zone sizes and residual plastic deformations).

Creep-crack-growth (da/dt and/or dc/dt) properties are input in tabular form similar to the fatigue-crack-growth-rate properties. The relation can be a simple power law or a multi-linear relation. One option is based on K_{max}, while the other option is based on CTOD. The first option is for creep-brittle materials, whereas the second option is for creep-ductile materials. (The current code does not use the time and temperature information but these data will be used once the new subroutine on summing cyclic and creep-crack growth damage is added).

E.3.3.2 FATIGUE- AND CREEP-CRACK-GROWTH MODELING

To identify the relevant crack-tip field parameters for characterizing fatigue- and creep-crack growth at elevated temperatures, the following process will be considered. Figure E-11 shows a schematic of the deformation zones in front and along the wake of a moving crack subjected to loading in the creep regime. Upon loading, a plastic zone is formed at the crack tip, and with time a creep zone develops. The size of the plastic zone depends on the applied stress-intensity factor (*K*) level and the effective flow stress ($\alpha\sigma_o$) of the material (the flow stress, σ_o, is elevated by the constraint factor, α, which accounts for the 3-D stress state in the crack-front region). If extensive plastic deformation occurs during initial loading, *K* loses its significance as the dominating crack-tip parameter. But the Dugdale-type strip-yield model has been useful from small- to large-scale plastic deformations. The size of the creep zone depends on the applied stress level, time, and temperature. With time, the stresses in the vicinity of the crack begin to relax due to creep deformation and the size of the relaxation (or creep) zone increases with time. The residual plastic and creep deformations are modeled in the wake of the moving crack tip. With time, creep recovery (strain relaxation) occurs along the crack surfaces. Neither *K* nor the J-integral are expected to uniquely characterize the crack-tip stress relaxation (or redistribution) behavior within the creep zone because creep deformation is not admitted in their formulation. Conversely, if the creep zone is small, *K* or *J* will continue to characterize the crack-tip stresses outside the creep zone. When the creep zone becomes comparable to the dimensions of the cracked body, *K* and *J* completely lose their significance as crack-tip parameters. Therefore, when creep is present, other crack-tip parameters need to be considered. In the past, the C^* and C_t parameters have been used in these situations. As the CTOD and cyclic CTOD are analogous to the J-integral and ΔJ, respectively, the rate of change in CTOD with time may also be analogous to C^*.

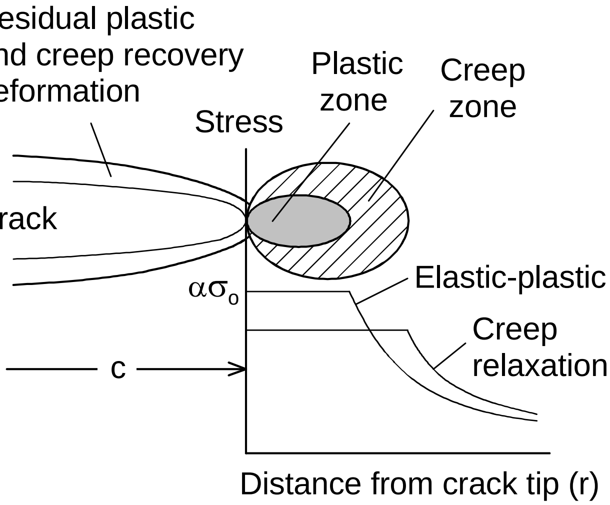


Figure E-11. Deformation zones in front and along the wake of a moving fatigue crack

The material creep and relaxation properties are yet to be incorporated into the FASTRAN code. The author is continuing to study the various methods and equations used to characterize these behaviors. In tests conducted at Mississippi State University (MSU) on an aluminum alloy under extremely high stresses, the material developed creep deformation that correlated well with $\varepsilon_T = \varepsilon_a + \gamma \log(1 + t)$, where ε_T is the total strain, ε_a is the initial strain due to the applied steady stress, and γ is a fitting parameter that is a function of applied stress, and *t* is the time in seconds. Under an applied strain near the fracture strain, the stress relaxation correlated well with $\sigma = \sigma_a + \eta \log(1 + t)$, where σ is engineering stress, σ_a is the resulting stress due to the applied strain, η is a parameter that is a function of the initial strain level, and *t* is the time in seconds. It will be assumed that the fitting parameters, γ and η, are power functions. Therefore, two coefficients are required for each fitting parameter from test data at a given test temperature.

- Stress-Intensity Parameters. When a cracked plate at elevated temperature is continuously cycled at a rapid loading frequency, such as 1 Hz or higher, and the fatigue-crack-growth rate is fast enough that it outpaces the spread of deformation and damage due to creep, and if small-scale yielding conditions are simultaneously maintained, the crack-growth rate (dc/dN) for a constant value of stress ratio (*R*) is uniquely correlated by the cyclic stress intensity factor, ΔK [E-28] or ΔK_{eff}, which accounts for the effects of *R* on fatigue-crack-growth rates.

  When a dwell (or hold) time is added to the fatigue cycle, it has been shown that the relationship between dc/dN and ΔK is no longer unique [E-28, E-29] in creep-ductile materials. However, for creep-brittle materials, Pelloux and Huang [E-30] and Saxena [E-31], have shown that dc/dN and Δ*K* can still be used to correlate the data using time-dependent fracture mechanics concepts.

- J-Integral and Δ-. When cyclic plasticity is no longer negligible at the crack tip (large-scale yielding), the cyclic J-integral (Dowling and Begley [E-32]), can be used to characterize high-temperature cyclic crack growth. Here the FASTRAN strip-yield model can be used to calculate the cyclic crack-tip-opening displacement (ΔCTOD) that is directly related to the cyclic J-integral. Therefore, the strip-yield model is well suited for correlating and predicting crack growth under large-scale plasticity conditions. In fact, most of the early strip-yield model developments used CTOD as the crack-driving parameter for modeling secondary creep-crack growth [E-33–E-34].
- C^* and C_t Parameters. If a cracked plate is subjected to secondary creep deformations, a J-integral like path-independent integral, C^*, can be defined that uniquely characterizes the crack-tip stresses and strains [E-35, E-36]. Bassani et al. [E-37] have shown that C_t is a more general parameter, and it includes all the conditions under which C^* is valid.

E.3.3.2.1 CYCLIC AND TIME-DEPENDENT CRACK GROWTH

The new FASTRAN code is being programmed to have two different fatigue- and creep-crack-growth relations in terms of: 1) ΔK_{eff} and K_{max}; and 2) ΔCTOD and CTOD. In both cases, the crack-growth relation will be:

$$(\mathrm{da/dN})_{\mathrm{Total}} = (\mathrm{da/dN})_{\mathrm{Cyclic}} + [\textstyle\int(\mathrm{da/dt})\ \mathrm{dt}\ ]_{\mathrm{Creep}} \qquad \text{(E-3)}$$

where for NEQN = 0:

$$(\mathrm{dc/dN})_{\mathrm{Cyclic}} = f(\Delta K_{eff}) \text{ and } (\mathrm{da/dt}) = \mathrm{g}(K_{max}) \qquad \text{(E-4)}$$

where ΔK_{eff} is the Elber's effective stress-intensity factor range [E-38], and K_{max} is the maximum applied stress-intensity factor,

and for NEQN = 1:

$$(\mathrm{dc/dN})_{\mathrm{Cyclic}} = f(\Delta\mathrm{CTOD}) \text{ and } (\mathrm{da/dt}) = \mathrm{g}(\mathrm{CTOD}) \qquad \text{(E-5)}$$

where ΔCTOD is the CTOD that automatically accounts for crack closure, and CTOD is the maximum crack-tip-opening displacement.

E.3.3.3.2 SOME ISSUES ON CREEP-CRACK-GROWTH-RATE TESTING

In FASTRAN (NFOPT=11), the stress-time-temperature history can be either trapezoidal (linear) or sinusoidal waveform (the sinusoidal waveform allows no hold times, i.e., t_{h1} and t_{h2} are zero). The linear option is shown in figure E-10 with the crack-opening stress, S_o. Equation (E-3) is evaluated over the complete cyclic history. Total crack extension per cycle is:

$$(\Delta a)_{\text{Total}} = (\Delta a)_{\text{Cyclic}} + (\Delta a)_{\text{Rise}} + (\Delta a)_{\text{Hold}} + (\Delta a)_{\text{Decay}} \qquad \text{(E-6)}$$

Because the crack-tip region develops compressive stresses during any unloading, the crack-tip damage, $(\Delta a)_{\text{Decay}}$, is zero; therefore, there is no damage during unloading. In addition, there is no damage (crack growth) during the hold time at the minimum applied stress. The hold time at minimum applied stress will cause compressive creep deformation in the crack-tip region.

The damage due to cyclic loading, $(\Delta a)_{\text{Cyclic}}$, is calculated from crack-closure theory using ΔK_{eff}. Currently, in FASTRAN, a multi-linear relation is used to describe the ΔK_{eff}-rate relation for a given material, as shown in figure E-13 (note that "a" is the crack depth through the thickness, and "c" is the crack length in the width direction, because FASTRAN deals with either surface or corner cracks in plates or at holes and through cracks). This nomenclature is consistent, but does not agree with the ASTM standards. For fatigue-crack growth, the effective stress-intensity-factor range [E-38] is calculated from the effective stress range, ΔS_{eff} (= $S_{max1} – S_o$). Initially, the cycle-by-cycle option (NMAX = 1) will be used, so that hold time, t_{1h}, will be used to develop the secondary creep zone (stress relaxation) and creep recovery for the material in the plastic wake. At the minimum stress, S_{min1}, reverse yielding will occur in the plastic zone and, maybe, in the plastic wake. Creep recovery under compressive stresses in the creep- and plastic-wake zones will occur. However, creep behavior under compressive stresses and strains has not been studied well, so this will be a topic for further discussions.

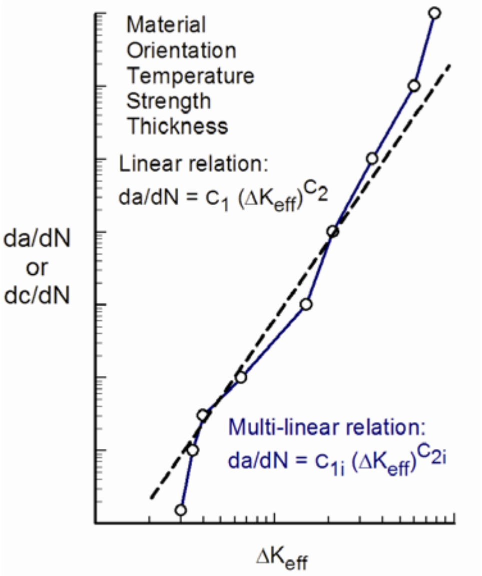


Figure E-13. Cyclic fatigue-crack-growth-rate relationship in terms of effective stress-intensity-factor range

In the multi-linear relationship, C_{1i} and C_{2i} are the constants for each linear segment. The threshold term in the FASTRAN crack-growth equation [E-27] uses C_3, C_4, and C_7; whereas, the fracture term uses C_5 and C_6. Initially, only the linear relationship (see dashed line in figure E-13) will be used; C_1 and C_2 will be selected to fit cyclic fatigue-crack-growth-rate data.

The damage due to the rise time, t_r, is $(\Delta a)_{\text{Rise}}$ and during the hold time, t_{h1}, crack growth is $(\Delta a)_{\text{Hold}}$. These are calculated from the second term in equation E-3 using the K_{max}-da/dt relation (see figure E-14). As for cyclic fatigue-crack growth, a multi-linear relationship will be programmed into FASTRAN. However, most creep-fatigue papers (reviewed) has not included the K_{max}-da/dt data, but only showed total crack-growth data for various frequencies, dwell times, and waveforms at elevated temperatures. These papers have presented time-dependent crack-growth data in terms of C^*, C_t, or $C(t)$ against total da/dN. Initially, the linear relationship will be assumed, and the constants C_8 and C_9 will be determined to best fit the experimental data.

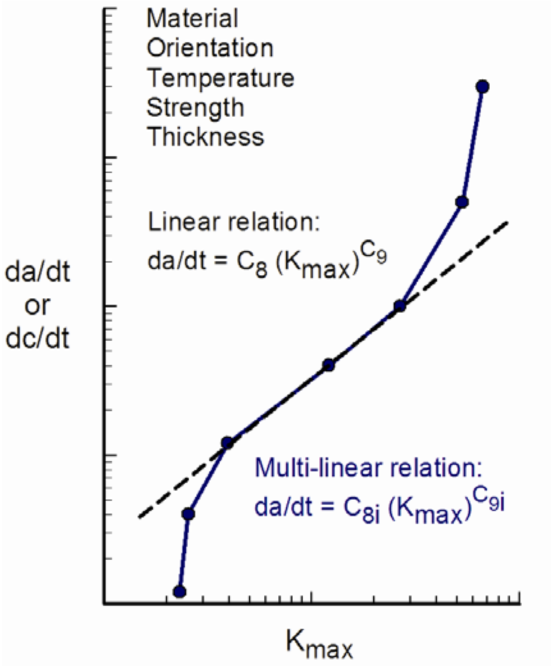


Figure E-14. Time-dependent crack-growth-rate relationship in terms of maximum stress-intensity factor

During a "K_{max} against da/dt" test, the crack is fully open, so that K_{max} could be considered an effective stress-intensity factor, K_{eff}. Under cyclic loading and the development of crack closure, the crack-front region has compressive stresses when the crack is closed, but it develops tensile stresses when the crack front opens. Therefore, the crack front feels a lower stress-intensity factor, like $K_{eff} = K_{max} – K_o$ (note that this term is exactly the same as ΔK_{eff}, but the term will be referred to as K_{eff} to avoid confusion). Therefore, for time-dependent crack growth, the effective stress-intensity factor is used in equation E-3 to compute damage. Newman et al. [E-39] had shown how fatigue-crack-growth and crack-closure behavior affected the fracture toughness of D6AC steel using a similar concept.

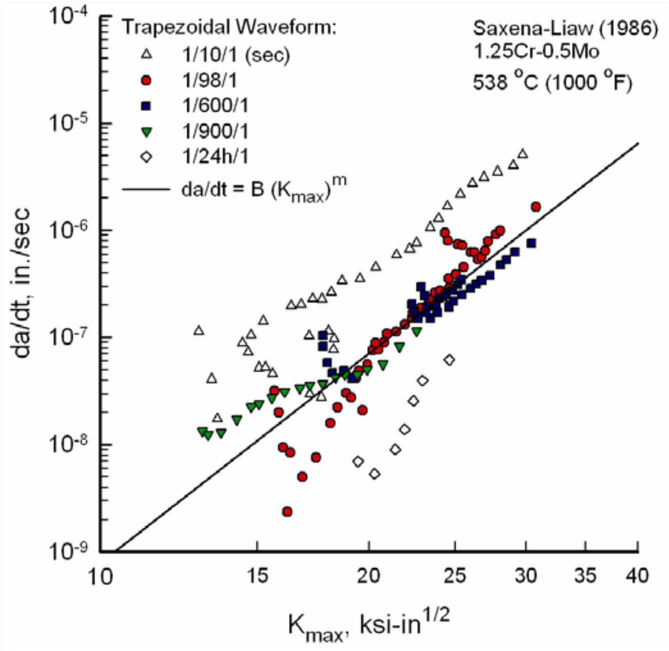
E.3.3.5 FATIGUE- AND CREEP-CRACK-GROWTH MODELING ON A STEEL

The new FASTRAN code is being programmed to have two different fatigue- and creep-crack-growth relations in terms of: 1) ΔK_{eff} and K_{max}, and 2) ΔCTOD and CTOD. In both cases, for the trapezoidal loading (dwell) tests, the crack-growth relation will be:

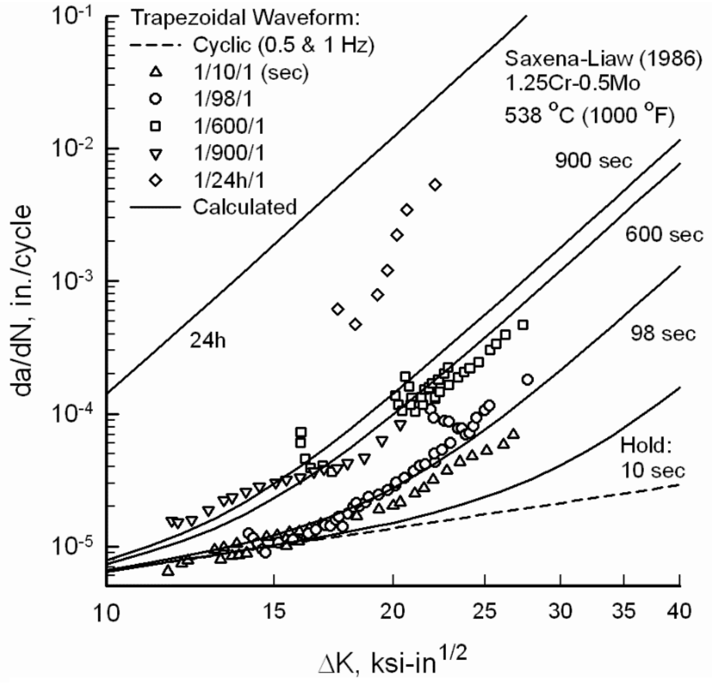
$$(da/dN)_{Total} = (da/dN)_{Cyclic} + t_h\ (da/dt)_{Creep} \qquad \text{(E-7)}$$

where $(da/dN)_{Cyclic}$ is the cyclic crack-growth-rate part, $(da/dt)_{Creep}$ is the creep crack-growth-rate part, and t_h is the hold time (neglecting rise time to maximum load and decay time to minimum load because they are small compared with the hold time on the pressure vessel steel).

Test data from Saxena and Liaw [E-40] and Yoon et al. [E-41] on a 1.25Cr-0.5Mo steel that was tested at elevated temperature (1000°F) are shown in figures E-6(a) and E-6(b). This material is classified as a creep-ductile material, where LEFM parameters were deemed inadequate. The tests were conducted on compact specimens using trapezoidal (dwell) fatigue loading with 1-second rise, t_h hold time (10 sec), and 1-second decay, such as 1/10/1-second loading. Figure E-15(a) is an estimated K_{max}-against-da/dt data to fit equation E-7.



(a)



(b)

Figure E-15. The (a) estimated time-dependent K_{max}-against-da/dt crack growth on 1.25Cr-0.5Mo steel at 1000ºF and (b) calculated time-dependent crack growth under trapezoidal loading on 1.25Cr-0.5Mo steel at 1000ºF

In figure E-15(a), the solid symbols show test data that correlated well with a simple power law as:

$$(\mathrm{da/dt})_{\mathrm{Creep}} = B\ (K_{max})^{m} \qquad \text{(E-8)}$$

as shown by the solid line. The test data from the lowest and highest hold times did not correlate properly, but this may be because of using LEFM parameters.

In Figure 15(b), the time-dependent crack-growth rate is plotted against ΔK. The dashed line is a simple power law as:

$$(\mathrm{da/dN})_{\mathrm{Fatigue}} = A\ (\Delta K)^{n} \qquad \text{(E-9)}$$

that was fit to fatigue-crack-growth test data at 0.5 and 1-hertz at $R = 0.1$. The solid curves are the calculated results using equation (E-9) for each hold time. The smallest hold time results were under predicted, whereas the longest hold time results were over predicted. But the overall agreement was considered fairly good, specifically from 98- to 900-second hold times.

E.3.3.6 FATIGUE- AND CREEP-CRACK-GROWTH MODELING UNDER VARIOUS WAVEFORMS

Some of the common loading waveforms used in the study of creep-fatigue crack growth behavior are shown in figure E-16. The important time parameters characterizing the loading waveforms are: t_r = rise time, t_h = hole time, and t_d = decay time. The amplitude of the loading waveform can be characterized by $\Delta P = P_{max} - P_{min}$, the difference between the maximum and minimum loads, and the load ratio, $R = P_{min}/P_{max}$. The loading frequency is denoted as f.

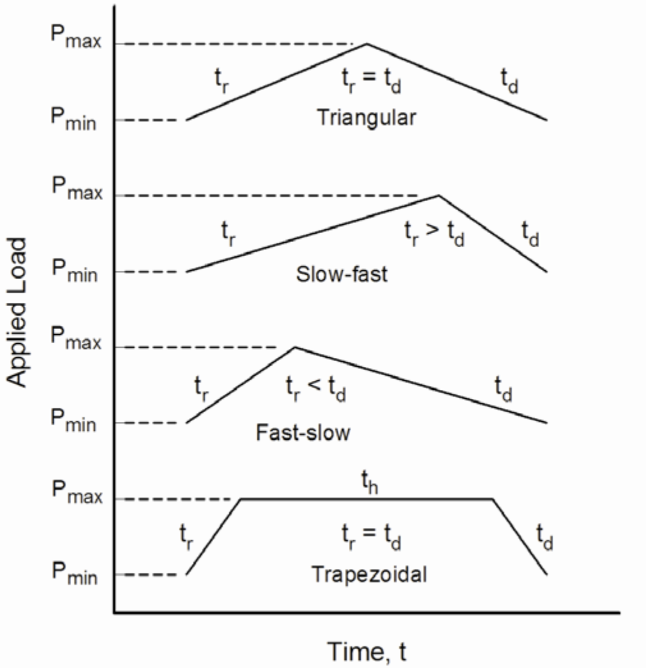


Figure E-16. Loading waveforms typically used in creep-fatigue experiments

In the literature, the sine waveform is also used to study creep-fatigue interactions by varying the frequency from very fast (≥ 10 Hertz) to very slow (~0.001 Hertz). For high frequencies, crack growth is dominated by cyclic crack growth, but for very slow frequencies, crack growth is dominated by creep-crack growth. During cyclic and creep-crack growth, the crack will develop crack closure, which will greatly affect the local crack-tip stress variations. A closed crack will have compressive stresses, whereas at the crack-opening load, P_o, the local crack-tip stresses go from compression to tension. Therefore, the damaging load amplitude should be $\Delta P_{eff} = P_{max} - P_o$. Therefore, the calculation of the crack drive, ΔK_{eff}, during cyclic crack growth will, naturally, include crack-closure effects. In a search of the literature, crack-closure effects appear to not have been considered on creep-crack growth. But the cyclic and (possible) creep damage are affected by crack closure (see figure E-17). The creep damage computed during the rise time from P_o–P_{max} should account for crack-closure effects. In using the second term in equation (E-3), the sinusoidal waveform causes about a factor-of-2 more damage than the triangular waveform, because the sine wave spends more time near the maximum load.

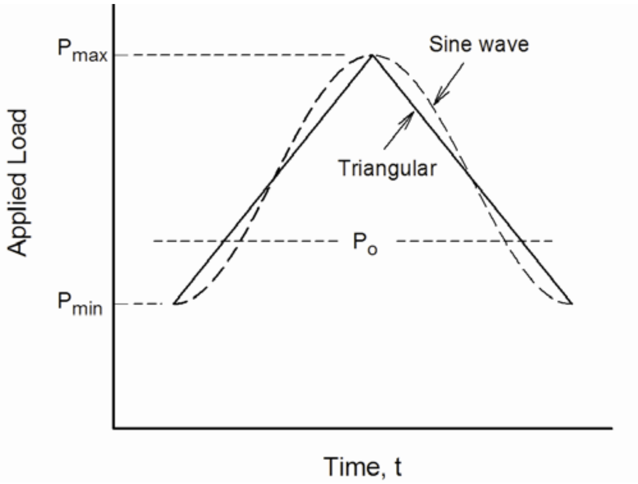


Figure E-17. Schematic diagrams of triangular and sinusoidal loading waveforms used in creep-fatigue experiments

E.3.3.7 FATIGUE- AND CREEP-CRACK-GROWTH DATA AND MODELING ON INCONEL-718

Floreen and Kane [E-42] conducted sinusoidal waveform loading on Inconel-718, which is classified as a creep-brittle material. They conducted tests over a wide range in frequencies (1–0.01 Hertz) in air and helium. The results for air are shown in figure E-18. For cyclic crack growth, the values of C_1 and C_2 were found to fit the test data (circular symbols) at 1 Hz. There was a slight difference between 1 Hz and 10 Hz. The value of C_9 was slightly larger than C_2, and C_8 was chosen to fit the data at 0.1 and 0.01 Hz. It would interesting to compare the K_{max}-against-da/dt relationship with actual test data instead of curve fitting.

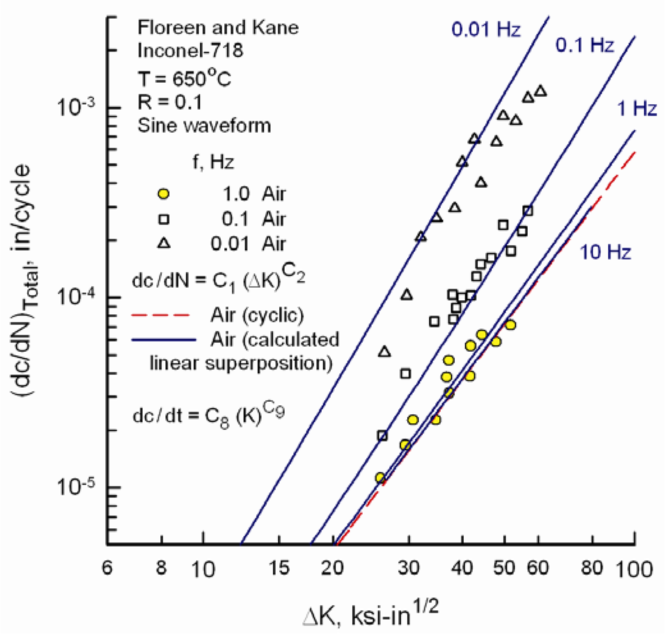


Figure E-18. Creep-fatigue behavior of Inconel-718 in air during elevated temperature tests under sinusoidal loading over a wide range in frequencies

Figure E-19 shows the same type of test data, except these tests were conducted in helium gas. The results were remarkable, in that very little difference in creep-fatigue crack growth was observed at 1 Hz. These results basically indicate the creep-fatigue on the Inconel-718 alloy is basically due to “environment”. Time in the aggressive air environment is attacking the grain boundaries and causing rapid crack growth compared with the more inert environment of helium.

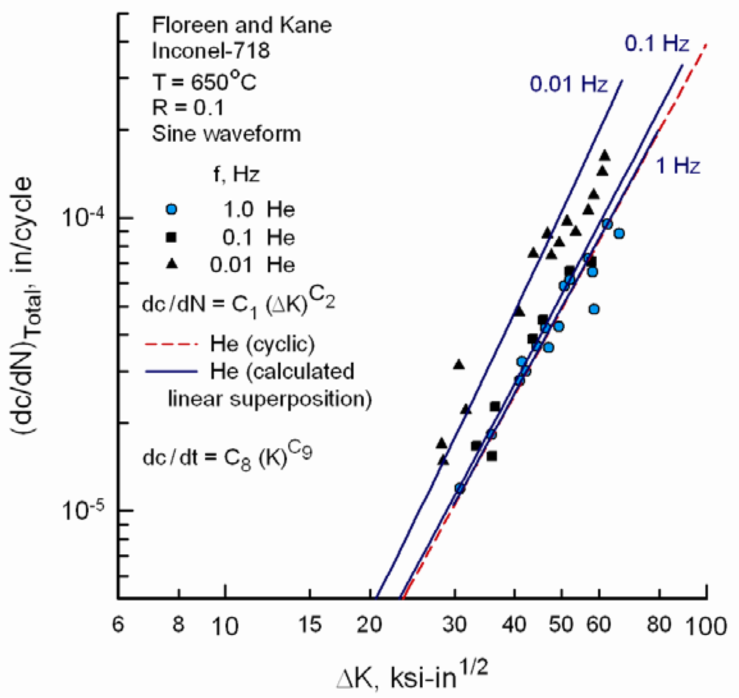


Figure E-19. Creep-fatigue behavior of Inconel-718 in helium during elevated temperature tests under sinusoidal loading over a wide range in frequencies

E.3.3.8 CREEP-FATIGUE CRACK-GROWTH DATA AND MODELING ON A NI-BASE SUPERALLOY

Nikbin et al. [E-43, E-44] tested a Ni-Base Superalloy (Astroloy) under sinusoidal loading over a very wide range in frequencies (10–0.001 Hz) (these data were taken from E-2760 [E-45] and the actual references have not been located). In this case, $C_9 = C_2$ and the value of C_8 was chosen to fit the 0.001 Hz data. The fit for all frequencies was quite good. In figure E-20, some slight discrepancy was observed for the 0.01 Hz data.

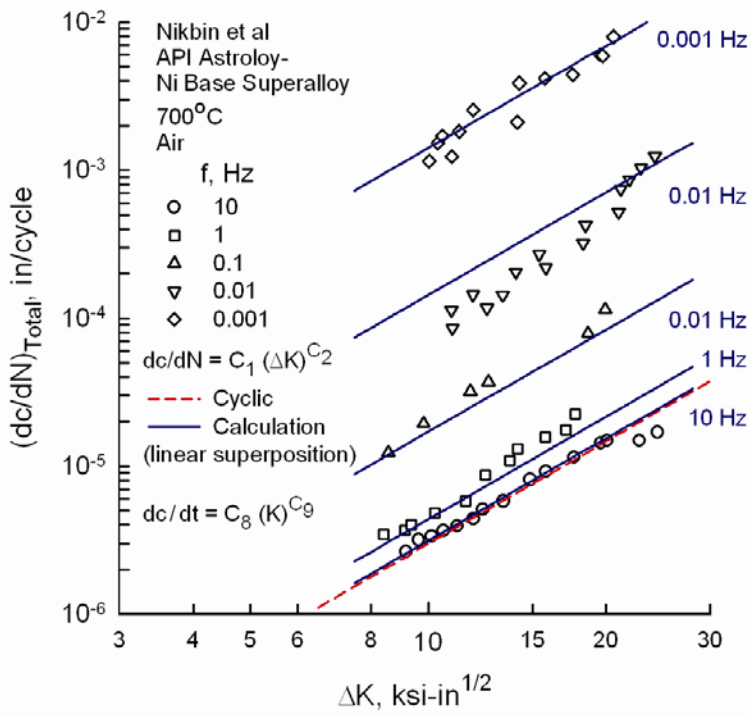


Figure E-20. Creep-fatigue behavior of Astroloy in air during elevated temperature tests under sinusoidal loading over a very wide range in frequencies

E.3.3.9 CREEP-FATIGUE CRACK-GROWTH DATA AND MODELING ON A 304 STAINLESS STEEL

Plumtree and Yu [E-46] tested the 304 stainless steel (see figure E-21) using the trapezoidal loading (see figure E-16). The 304 stainless steel is classified as a creep-ductile material, but the superposition method using LEFM, Δ*K* and K_{max}, were used for cyclic and creep-crack growth, respectively. Because of the slopes on the test data, $C_9 = C_2$ and the value of C_8 was chosen to fit the test data. The fit was excellent, except for one test sequence. These results were surprising because LEFM methods were not supposed to work on a creep-ductile material.

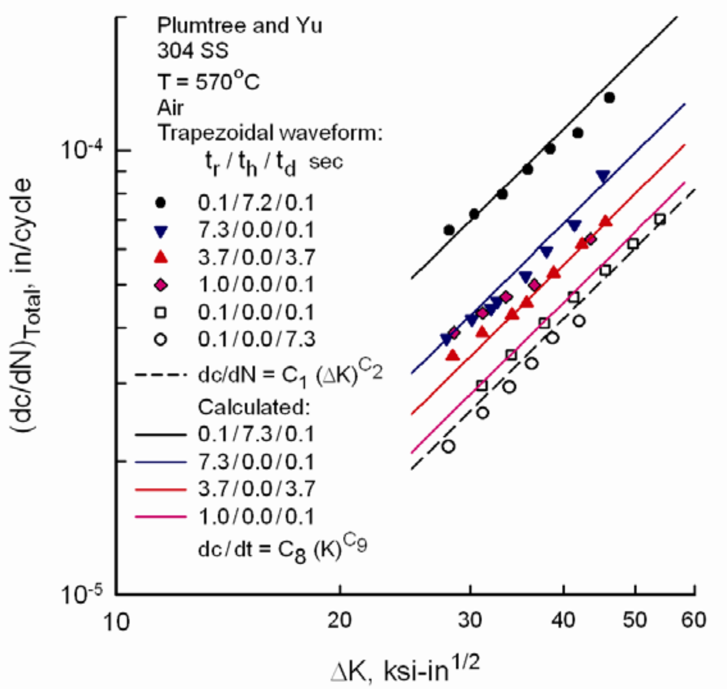


Figure E-21. Creep-fatigue behavior of 304 stainless steel in air during elevated temperature tests under various trapezoidal waveforms

E.3.3.10 FATIGUE-CRACK-GROWTH PROPERTIES OF INCONEL-718

Inconel-718 has been selected as the model material for the plasticity-creep-relaxation project. Fatigue-crack-growth rate tests had previously been conducted on compact specimens made of an Inconel-718 alloy to study the behavior over a wide range of rates from threshold to fracture and over a wide range of load ratios ($0.95 \geq R \geq 0.1$) and constant K_{max} test conditions [E-47]. These tests were conducted under laboratory air and room temperature, and the results are shown in figure E-22. These results include a test conducted at a very high R value of 0.95. Conversely, only the R = 0.1 data were generated using the compression precracking constant amplitude (CPCA) test procedures. This method does not require any load reduction and the stress-intensity level was always increasing. On the other R values (0.4 to 0.95), the compression precracking load-reduction procedure was used, and the load reduction was initiated at a rate approximately 2E-8 to 3E-8 in/cycle.

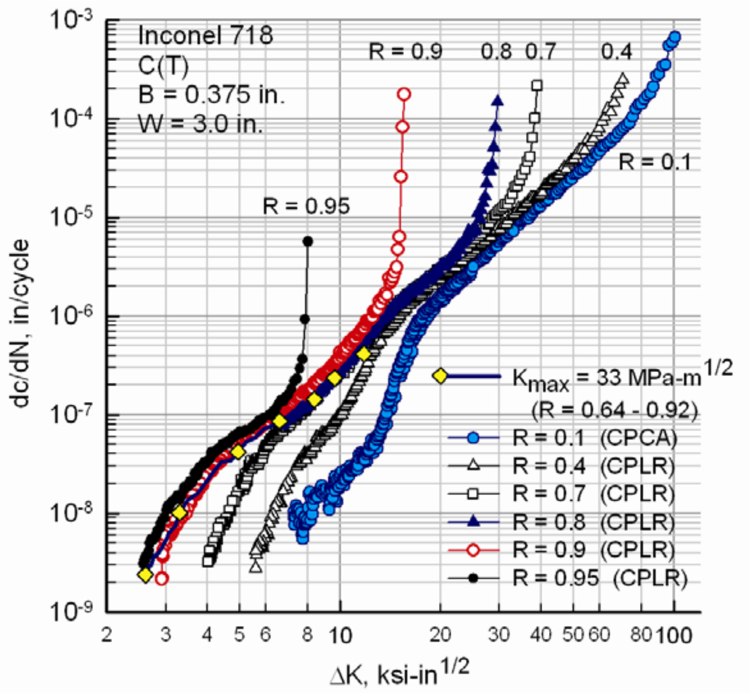


Figure E-22. Fatigue-crack-growth data using compression precracking methods for a wide range in stress ratio, *R*

For the R = 0.1 tests, the lower plateau (ΔK = 7 to 12 ksi-in$^{1/2}$) ends at approximately 4E-8 in/cycle, transitions sharply to about 1E-6 in/cycle, and then reaches another plateau. It would have been interesting to run CPCA tests on the high R values to see if a lower plateau would have developed for these tests. Of the many materials tested at MSU during the past decade, the Inconel-718 was the only material where a significant difference was observed between ASTM E647 load-reduction and CP testing. The R = 0.95 and 0.9 tests show a difference in the threshold regime, and there were indications that the very high R tests were experiencing crack closure [E-47].

The correlation of the ΔK-rate data onto a unique ΔK_{eff}-rate curve in the threshold range is very difficult due to the shape differences between the various R values.

E.4. CONCLUDING REMARKS

The framework for what modifications and what creep/relaxation test data are needed in the strip-yield model have been established. For fatigue-crack-growth under mechanical loading, the crack-tip-opening displacements (CTOD) parameter has been found to be directly proportional to the amount of crack extension under cyclic loading. Past research on creep-crack growth has also used the CTOD as the crack-driving parameter. Whether the large amount of crack extension under load-time-temperature effects can be directly related to CTOD has not been established. The research in the past year has identified critical test data that needs to be determined. First, the creep and relaxation test data under extreme stress and strain states that develop in the crack-front region has not been measured on an engine material, such as Inconel-718 (project material). The author has conducted some creep/relaxation tests on an aluminum alloy under extreme stress and strain states, and found creep and relaxation equations to fit the experimental test data. Currently, this behavior would have to be assumed/extrapolated from lower stress and strain states on engine materials. In addition, the measurement of CTOD during load-time-temperature profiles need to be measured to see if the large amount of crack extension could be characterized by the CTOD parameter.

During the past year, the author has analyzed a large number of materials under a wide variety of load-time-temperature profiles and found that the linear-superposition model worked fairly well using LEFM concepts to model mechanical and time-dependent crack growth for creep-brittle materials. Efforts to model CTOD under creep and relaxation conditions using the strip-yield model may be developed by the author as independent research.

E.5 REFERENCES

E-1. Antolovich, S. D., & Saxena, A. (2002). Failure analysis and prevention. In W.T. Becker & R.J. Shipley (Eds.), *ASM Handbook Vol. 11* (738). Materials Park: ASM International.

E-2. McEvily, A. J., & Wells, C. H. (1973). *On the applicability of fracture mechanics to elevated temperature design*, Proceedings from the International Conference on Creep and Fatigue in Elevated Temperature Applications, Philadelphia, PA,

E-3. Vitek, V. (1977). *International Journal of Fracture, 13.*

E-4. Vitek, V. (1978). A theory of diffusion controlled intergranular creep crack growth. *Acta Metallurgica*, *26*(9), 1345–1356.

E-5. Riedel, H. (1977). A Dugdale model for crack opening and crack growth under creep conditions. *Materials Science and Engineering*, *30*(3), 187–196.

E-6. Ewing, D. J. F. (1978). Strip yield models of creep crack incubation and growth. *International Journal of Fracture, 14*(1), 101–117.

E-7. Dugdale, D. S. (1960). Yielding of steel sheets containing slits. J*ournal of Mechanics and Physics of Solids*, *8*(2), 100–104.

E-8. Bilby, B. A., Cottrell, A. H., and Swinden, K. H. (1963). *Proceedings of the Royal Society*, Vol. A727, 304-314.

E-9. Stewart, C. M. & Gordon, A. P. (2013). *Creep crack growth simulation of Ni-base superalloy*, Paper presented at the ASME Turbo Expo 2013: Turbine Technical Conference and Exposition, San Antonio, TX. (GT2013-96005).

E-10. ASTM STP 1321, 1997, The merging of fatigue and fracture mechanics concepts—A historical perspective, *Fatigue and Fracture Mechanics 10,* pp. 3–51.

E-11. NASA Report. (1992). FASTRAN II—Fatigue Crack Growth Structural Analysis Program (NASA TM-104159).

E-12. Newman, J. C., Jr. (2010). *FASTRAN – A Fatigue Crack Growth Life Prediction Code based on the Crack-Closure Concept, Version 5.3*.

E-13. FAA Report. (2014). Small- and Large-Crack Damage Tolerance Databases for Rotorcraft Materials (DOT/FAA/TC-13/29).

E-14. Riedel, H., (1987) Fracture at high temperatures. *Springer-Verlag*, Berlin, Germany.

E-15. Webster, G. A., & Ainsworth, R. A. (1994) *High temperature component life assessment*. Chapman and Hall, London, England, UK.

E-16. Gittus, J. (1975). *Creep, viscoelasticity and creep fracture in solids*. London, England: Applied Sciences.

E-17. Kachanov, L. M. (1967). *The theory of creep*. Boston, MA: National Lending Library for Science and Technology.

E-18. Rabotnov, Y. N. (1969). *Creep problems in structural members*. Amsterdam: North Holland.

E-19. Stewart, C. M., & Gordon, A. P. (2009). Modeling the temperature dependence of tertiary creep damage of a Ni-base alloy, J*ournal of Pressure Vessel Technology, 131*(5), 1–11.

E-20. Newman, J. C., Jr., "Finite-Element Analysis of Fatigue Crack Propagation Including the Effects of Crack Closure," Ph.D. Thesis, Virginia Polytechnic Institute and State University.

E-21. Carlson, R. L. and Manning, G. K., "A Summary of Compressive-Creep Characteristics of Metal Columns at Elevated Temperatures," WADC Technical Report 57-96, April 1958.

E-22. Defense Metals Information Center. (1961). *Effects of Moderately High Strain Rates on the Tensile Properties of Metals*,(DMIC Memorandum 142), Battelle Memorial Institute, Columbus, Ohio,. Moon, D.P. and Campbell, J.E.

E-23. Dayton Univ OH Research Inst. (1984). *Hold-Time Effects in Elevated Temperature Fatigue Crack Propagation*, (AFWAL-TR-84-4184), Weerasooriya, T. and Nicholas, T.,

E-24. ASTM Standard E-2760, "Test Method for Creep-Fatigue Crack Growth Testing," ASTM International, West Conshohocken, PA, 2010

E-25. Javanroodi, F.D. and Nikbin, K.M. (2006). The Fracture Mechanics Concept of Creep and Creep/Fatigue Crack Growth in Life Assessment, *International Journal of Engineering Science, 17*(3-4), 1-7.

E-26. Gallerneau, F., Kruch, S. and Kanoute, P., “A New Modeling of Crack Propagation with Fatigue-Creep-Oxidation Interaction under Non-Isothermal Loading,” RTO AVT Symposium “Ageing Mechanisms and Control: Part B – Monitoring and Management of Gas Turbine Fleets for Extended Life and Reduced Costs,” Manchester, England, United Kingdom, 2001. Published in RTO-MP-079.

E-27. Fatigue & Fracture Associates, LLC. (2010). *FASTRAN – A fatigue crack growth life prediction code based on the crack-closure concept, Version 5.3* User Guide.

E-28. Grover, P. S. (1996). *Creep-Fatigue Crack Growth in Cr-Mo-V Base Material and Weldments* (Ph.D. Thesis), Georgia Institute of Technology, Atlanta GA.

E-29. Saxena, A. (1998). *Nonlinear Fracture Mechanics for Engineers*, Boca Raton, Florida: CRC Press, LLC.

E-30. Pelloux, R.M. and Huang, J.S. (1980). Creep-Fatigue-Environment Interactions in Astroloy. In R. Pelloux and N. Stoloff, (Eds.), *Creep-Fatigue-Environment Interactions*. (151-164), Warrendale, Pennsylvania: TMS-AIME.

E-31. Saxena, A. (1983) A Model for Predicting the Environment Enhanced Fatigue Crack Growth Behavior at High Temperature,” *Thermal and Environmental Effects in Fatigue-Research Design Interface,* (PVP-Vol. 71), ASME: New York, 171–184.

E-32. ASTM STP 590, “Fatigue Crack Growth under Gross Plasticity and the J-Integral,” *Mechanics of Crack Growth, ,* Philadelphia, Pennsylvania, 82–103.

E-33. Vitek, V. (1977). A Theory of the Initiation of Creep Crack Growth,” *International Journal of Fracture, 13*(1), 39-50.

E-34. Ewing, D.J.F. (1978). Strip Yield Models of Creep Crack Incubation and Growth, *International Journal of Fracture, 14*(1), 101-117.

E-35. ASTM STP 590, 1976, “A Fracture Mechanics Approach to Creep Crack Growth,” *Mechanics of Crack Growth, Proc. of the Eighth National Symposium on Fracture Mechanics*, Philadelphia, Pennsylvania: American Society for Testing and Materials, 128-148.

E-36. Dayton Univ Oh Research Inst.,(2003). *Dwell-Time Fatigue Crack Growth in Ni-Base Superalloys,* (US Govt. Primary Contract No. F33615-98-C-5214), Saxena, A. and Findley, K.,

E-37. ASTM STP 995, 1989, "Evaluation of the Ct Parameter for Characterizing Creep Crack Growth in the Transient Regime," *Nonlinear Fracture Mechanics, Vol. 1, ,* Philadelphia, Pennsylvania, 7-29.

E-38. ASTM STP 486, 1971, “The Significance of Fatigue Crack Closure,” *Damage Tolerance in Aircraft Structures*,” Philadelphia, Pennsylvania: American Society for Testing and Materials, 230-242.

E-39. ASTM STP 1256, 1995, “Fracture Toughness and Critical Crack Sizes for the Space Shuttle Motor D6AC Steel Case,” *Fracture Mechanics: 26th Volume*, , West Conshohocken, Pennsylvania, 799-821.

E-40. Electric Power Research Institute. (1986). *Remaining Life Estimation of Boiler Pressure Parts – Crack Growth Studies*, (EPRI contract RP 2253-7), Saxena, A. and Liaw, P.K.,

E-41. Yoon, K. B., Saxena, A., & Liaw, P.K. (1993). Characterization of creep-fatigue crack growth behavior under trapezoidal waveshape using Ct-parameter, *International Journal of Fracture, 59*(2), 95–114.

E-42. Floreen, S., & Kane, R. H. (1980). An investigation of creep-fatigue-environment interactions in Ni-Base super alloy, *Fatigue of Engineering Materials and Structures, 2*, 401–412.

E-43. ASTM STP 942, 1988, “Prediction of Crack Growth under Creep-Fatigue Loading Condition,” In H. Solomon, G. Halford, L. Kaisand and B. Leis, (Eds.), *Low-Cycle Fatigue*, Philadelphia, Pennsylvania, 281–292.

E-44. Winstone, M. R., Nikbin, K. M., & Webster, G. A. (1985). Modes of failures under creep/fatigue loading of Ni-based super alloy, *Journal of Materials Science, 20*(7), 2471–2476.

E-45. ASTM E2760-10, 2010, “Standard Test Method for Creep-Fatigue Crack Growth Testing” West Conshohocken, Pennsylvania: ASTM International.

E-46. Plumtree, A., & Yu, M., (1983). Influence of Waveshape on High Temperature Crack Growth Behavior in Stainless Steel,. In C. Jaske, S. Hudak, and M. Mayfield (Eds.), *Thermal and Environmental Effects in Fatigue: Research-Design Interface PVP-Vol. 71*, (13-19)., New York, NY: American Society for Mechanical Engineers.

E-47. Yamada, Y and Newman, J.C., Jr. (2009). Crack closure under high load-ratio conditions for Inconel 718 near threshold behavior, *Engineering Fracture Mechanics, 76*(2), 209–220.

APPENDIX F—THERMO-MECHANICAL FATIGURE MISSION TESTING TO SUPPORT ROTOR INTEGRITY SUB-COMMITTEE TEST CASE

F.1 BACKGROUND

During the evaluation of the Rotor Integrity Sub-Committee (RISC) test case for axial blade slots, it became evident there was some disagreement amongst the original equipment manufacturers (OEMs) about how to incorporate temperature into a pairing routine to account for out-of-phase temperature loading. In addition, the duty cycle of the RISC test case goes into compression before a tensile dwell and a maximum tensile load are applied. This loading history raises questions about the appropriate method to use to account for these effects.

As the RISC test case is a notional loading cycle, crack growth data that could be used to referee the various methods used to analyze the test case were not readily available. Pratt & Whitney began work with Georgia Tech (GA Tech) that involved, in part, generating test data based on the RISC test case duty cycle that could be used to referee the various methods OEMs used. Some additional funds from this grant were used to supplement the effort with a few additional tests that had been suggested by Pratt & Whitney and GA Tech.

The Pratt & Whitney project with GA Tech involved another approach to develop a high-temperature, time-dependent crack growth model. This project has a testing phase and a model development phase. The mission developed for the RISC Axial Slot test case was one of the loading profiles used for model validation. The testing program at GA Tech included a variety of testing conditions. These testing conditions include simple in-phase and out-of-phase thermo-mechanical fatigue (TMF) cycles with and without dwell as well as a variety of mission cycles in addition to the RISC Axial Slot test case mission to validate the model. Two test results from the work with GA Tech are shown in section 3.5 of this report to provide context for one of the new test results. The full description of the test program, their results, and the subsequent modeling effort is expected to be published in 2017 by Dr. Andrew Radzicki.

The TMF mission testing proposed was designed to answer two questions. The first question was: What is the impact of an intermediate tensile dwell that is present in the notional RISC mission? In the RISC mission, the dwell is 10,000 seconds. Because this was not possible to recreate with the available budget and schedule, a 120-second dwell was used instead. If there are effects of dwell at the intermediate stress and temperature of the mission, an effect should be seen using 120 seconds of dwell time.

The second question to be investigated with this testing was: What is the impact of the 30-second dwell at the minimum load/maximum temperature condition? In the prescribed RISC mission, the *R*-ratio for the major cycle based on this minimum time is approximately -0.3. The second set of tests shifted the mission profile down in strain such that the maximum strain condition occurred at zero strain, which makes the *R*-ratio = -∞. This was done to maximize the creep in compression at the minimum strain/high-temperature time point while maintaining the ΔK values of the pairs. The results were expected to show whether ΔK was a driver in TMF crack growth, or if $K_{\max}$ was more influential, and the crack would grow after the tensile residual stresses (RS) built up because of creep in compression.

All data generated under this effort, not just the FAA-funded data, will be made publicly available. A summary of the data generated under the FAA grant is provided in this report in section 3.5. Some of the GA Tech data will be shown in this report to provide context for the testing performed under the Probabilistic Integrity and Risk Assessment of Turbine Engines grant. The full data set developed in cooperation with GA Tech will be published when the dissertation of Dr. Andrew Radzicki has been cleared for public release.

All testing discussed in this report was performed by Derivation Research Laboratory, Inc. (DRL) located in Ottawa, Canada. DRL was chosen for this project because of their unique capabilities in infrared (IR) imaging and expertise in TMF testing and TMF crack growth.

F.2 TEST METHOD

F.2.1 EXPERIMENTAL SETUP

A uniaxial servo-hydraulic test machine with a 20,000-pound load capacity was used to apply mechanical loading during the TMF test. The thermal loading was induced with an inductive coil powered by a 5kW high-frequency power supply. The strain was measured using a high-temperature axial extensometer, whereas the temperature was measured using an IR pyrometer. The control system consisted of a digital controller running customized software used for closed-loop control of both strain and temperature, and control of cooling air supplied to the specimen. Figure F-1 shows the TMF test setup used in this research.

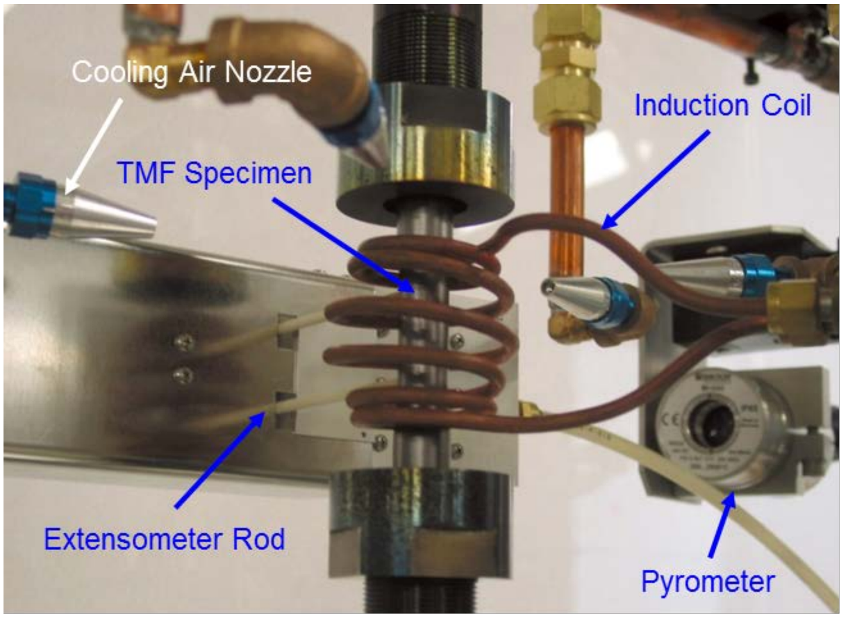


Figure F-1. A photograph of a TMF test setup

The hollow cylindrical geometry of the TMF test specimens were machined from INCO718 material. Prior to starting the TMF test, two black-body targets used for the IR temperature measurements were painted on the specimens using Pyromark® High Temperature Paint 2000 Flat Black. The black body targets have a reduced susceptibility to emissivity changes and therefore assisted in the reduction of temperature variation throughout the TMF test period.

IR imaging was carried out to collect active in situ crack growth data on the test sample using a Jenoptik VarioCAM3® HiRes IR camera. The camera uses a quantum well IR photo detector that has a focal plane array detector of 1024×768 elements, a temperature resolution of 20 mK, and a spectral response in the long-IR region (7.5–14 μm).

Images were generally taken at the peak cyclic load. Data were collected from the IR cameras connected to a computer by a network cable. The visualization, acquisition, and processing of the thermal images were performed using a mix of commercial and in-house software. A photograph of a thermal imaging setup is presented in figure F-2.

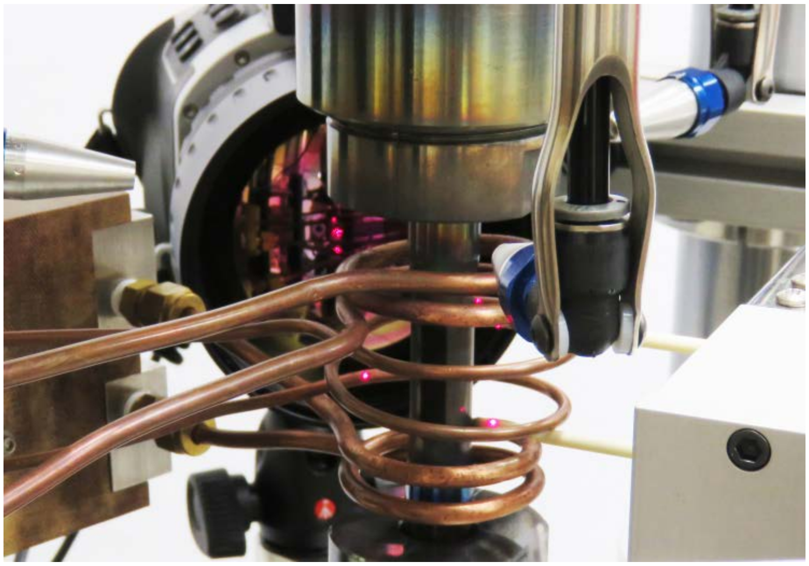


Figure F-2. A photograph of a TMF test setup with the Jenoptik IR camera

F.2.2 TMF TEST SYSTEM DESCRIPTION

During a TMF test, strain and temperature vary simultaneously. Therefore, the total measured strain in a TMF test specimen is the algebraic sum of thermal strain and mechanical strain. These two components must be separated for measurement and control of the mechanical strain as follows: $\varepsilon_{mech} = \varepsilon_{total} - \varepsilon_{thermal}$. The test system compensates for the thermal expansion effects on a dynamic basis so that cycling is accomplished with controlled values of mechanical strain applied to the specimen. The temperature uniformity along the specimen was assessed under static and dynamic temperature conditions with a thermocouple instrumented test sample.

Test time, temperature, and mechnical strain mission endpoints were entered into the computer along with the thermal strain profile prior to transfering control of the test machine to the computer system. The fully automated computer control system then manages all facets of TMF testing, including test initiation, algebraic minipulation of total strain to mechanical strain via the thermal strain calibration, continuous or discontinuous cycling, and test termination. Data on specimen temperature, deflection, and load were measured and recorded logarithmically for cyclic data and every cycle for peak and valley data throughout the duration of the test. Tests were generally discontinued when load on the specimen decreased by 70% to prevent damage to the extensometer and possibly the heating system. Applicable ASTM standards were followed for performing specific parts of the TMF test. These include the following:

- E83: Standard Practice for Verification and Classification of Extensometer Systems
- E220: Standard Practice for Calibration of Thermocouples by Comparison Techniques
- E606: Standard Practice for Strain-Controlled Fatigue Testing
- E647: Standard Test Method for Measurement of Fatigue Crack Growth Rates
- E1012: Standard Practice for Verification of Testing Frame and Specimen Alignment under Tensile and Compressive Axial Force Application
- E2368: Standard Practice for Strain-Controlled Thermomechanical Fatigue Testing

One TMF test frame was used to generate the data. This frame consisted of an MTS Model 810 uniaxial servo-hydraulic test machine with a 22-kip (22,000 lbf) load capacity with digital control system. An MTS air-cooled high temperature axial extensometer with a 25-mm gage length was used to measure strain.

Heating of the specimen during the TMF test was provided by an Ameritherm induction heater (model NovaStar, 5 kW 50-485 kHz) with a three-zone induction coil capable of being adjusted independently. The induction coils were designed to minimize circumferential and axial temperature gradients on the TMF test specimens. Temperature feedback during the test was provided by IR pyrometry to the controller for closed-loop temperature control. Specimen temperature is required to vary periodically in synchronization with mechanical strain during a TMF test. To achieve the desired mechanical strain component of the test setup, free thermal expansion of the sample must be captured for use duing the test. To accomplish this, the control mode of the test frame is set to load control with the force command set to zero, and the sample is thermal cycled between the minimum and maximum temperature with the goal of minimizing thermal hysteresis. Then the optimized thermal cycle is recorded. Part of this adjustment is to manipulate the forced air-cooling of the specimen, which is accomplished via metered and controlled air directed near the specimen surface and through the center of the hollow specimen. The error between the command and the feedback instrument (i.e., the control pyrometer) will call for either heat or cooling. Both internal and external cooling is provided by compressed air. The internal cooling was made through the grips, while the external cooling was applied through four diffusion nozzles strategically placed around the upper portion of the TMF test specimen. The positions of the external nozzles are adjusted to attain the minimum thermal strain hysteresis between heating and cooling portions of the thermal cycle and optimized during the thermal strain calibration of the thermal cycle. Temperature and strain versus time are prescribed in tables F-1 and F-2. Once set, the automated computer system assumes full control of specimen heating and cooling processes during the test.

Table F-1. Mission time points for test profile 1

Test Profile #1 (4.0 min cycle time) with hold			
Delta time	Time (s)	Temp (F)	Strain
	0	500	0.00091
10	10	1110.8	-0.00109
30	40	1110.8	-0.00109
12	52	1050.7	0.00203
120	172	105037	0.00203
12	184	987.8	0.00226
12	196	892	0.00179
20	216	530	0.00325
12	228	500	-0.00018
12	240	500	0.00091

Table F-2. Mission time points for test profile 2

Test Profile #2 (2.0 min cycle time) with hold			
Delta time	Time (s)	Temp (F)	Strain
	0	500	-0.00234
10	10	1110.8	-0.00434
30	40	1110.8	-0.00434
12	52	1050.7	-0.00122
12	64	987.8	-0.00099
12	76	892	-0.00146
20	96	530	0
12	108	500	-0.00343
12	120	500	-0.00234

F.2.3 TMF TEST SPECIMEN

The test specimens used in this program are fabricated from Inconel 718 and are a hollow tube design with button-head grip ends. The grips are designed to allow smooth transition from positive to negative loads in addition to being able to sustain high-compressive loading without buckling. The parallel section of this sample is 1.2″ long. The outside diameter is 0.550″ and the inside diameter is 0.450″, producing a wall thickness of 0.050″ nominally and a cross-sectional area of 0.080 in^2.

The outside diameter is low-stress ground and undergoes a linear polish, producing an 8AA finish or better. The inside diameter is gun drilled, and the last 0.020″ is mechanically honed to produce low RS to the same finish as the outer diameter or better. The inner diameter appears cross-hatched and highly polished as would be expected of a mechanical honing process. This choice of specimen provides the best opportunity for uniform through-wall temperature distribution during heating and cooling cycles. A single notch is plunge electro-discharge-machined through the entire specimen thickness. All TMF specimens mentioned in this report, both the tests performed under Probabilistic Integrity and Risk Assessment of Turbine Engines and the GA Tech tests, were machined at the same time using the same processes. As this was a crack growth test, the recast layer from the electro-discharge-machined process was left intact to help with crack initiation. Leaving the recast layer intact is standard practice for crack growth tests. The dimensions of the notch are on average 0.018″ in diameter, positioned in the center of the gage section as shown in figure F-3.

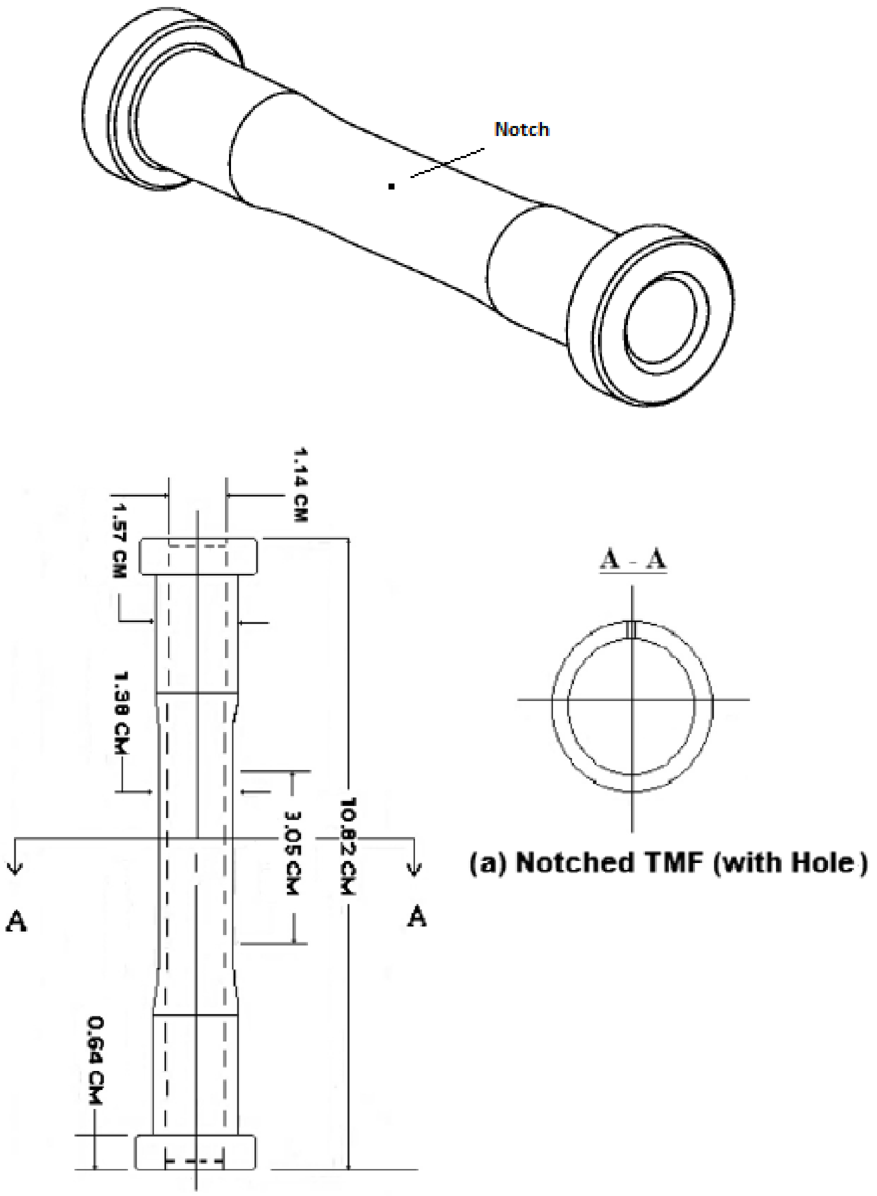


Figure F-3. Pratt & Whitney TMF specimen

F.2.4 TMF CRACK GROWTH MEASUREMENT METHOD

An IR camera was used to monitor crack growth from the hole in the gauge section of the TMF specimens. Images were automatically acquired every cycle throughout the TMF test and then post-test processed to quantify the crack progression.

Because of the fact that the specimen heating is caused by Foucault currents, any hole that causes a field disturbance may result in overheating in the area around the hole. This was analyzed in the region around the hole using IR cameras. Though the IR camera acquires IR images during each test, the actual emissivity is not required to determine the crack length. Therefore, it was not necessary to calibrate the IR camera for this program. IR and optical images taken during the heating portion for a typical specimen are presented in figures F-4(a) and F-4(b). From the IR image shown in figure F-4(a), the scale suggests that the temperature is significantly hotter in the hole. Though from the optical image (see figure F-4(b)), there is no visual evidence of this temperature variation. Because the IR image can only be affected by either temperature or emissivity, it would suggest that the emissivity is different for the hole compared to the bulk of the specimen. From figure F-4, an emissivity of 0.52 and 0.68 was determined for the specimen surface and hole, respectively, at the peak temperature during a temperature hold. Assuming the emissivity is constant throughout the cycle, the emissivity can then be applied to each to determine the actual temperature. Using this corrected emissivity value indicates there is no significant temperature variation between the bulk material, the crack, and the hole. The IR crack measurements results have been verified by comparing with post-test replicas.

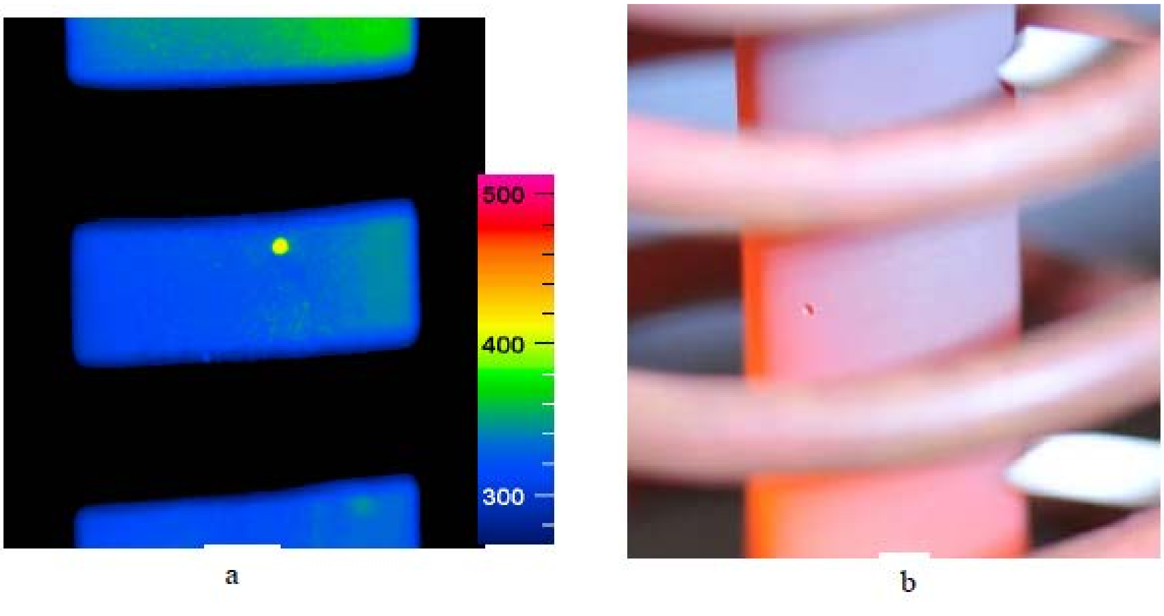


Figure F-4. An (a) IR image around the hole and (b) optical image of the same area

F.3 TEST CONDITIONS

The planned test matrix included two test conditions with two specimens tested at each condition for a total of four tests. The first test condition was designed to mimic the notional mission that RISC is using for the Axial Slot Bottom test case. The stresses used were based on the surface location in the slot that was being considered for the test case. This location was identified by having the highest surface stress of the slot. These stresses were converted to strains based on the material properties provided to analyze the test case. Figure F-5 shows where the stresses were selected to generate the strain profile for these tests. The temperatures used were the temperatures at the same node as was used for the stresses. The mission used for the test case includes a 10,000-second dwell at an intermediate tensile stress and at a moderately high temperature. A 10,000-second dwell was not feasible under the budget and schedule provided, so a 120-second dwell was selected instead. It is expected that if there is an acceleration of the crack growth rate due to the presence of the intermediate tensile dwell, a 120-second dwell time would be sufficient to see some effects. Table F-1 contains the loading conditions for each time step. Figure F-6 is a graph of the strain and temperature profile for the test cycle, whereas figure F-7 shows the mechanical strain versus temperature.

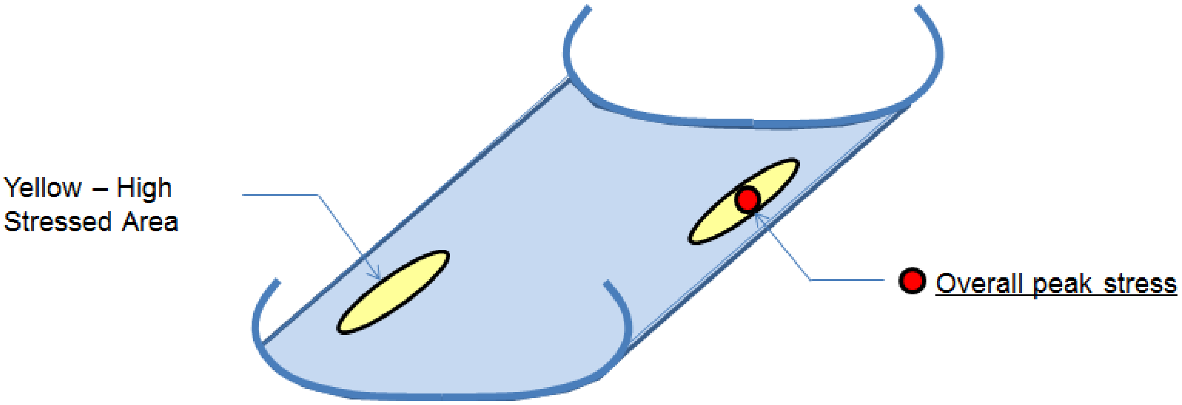


Figure F-5. Illustration of the location used for stress profile used to create loading profile

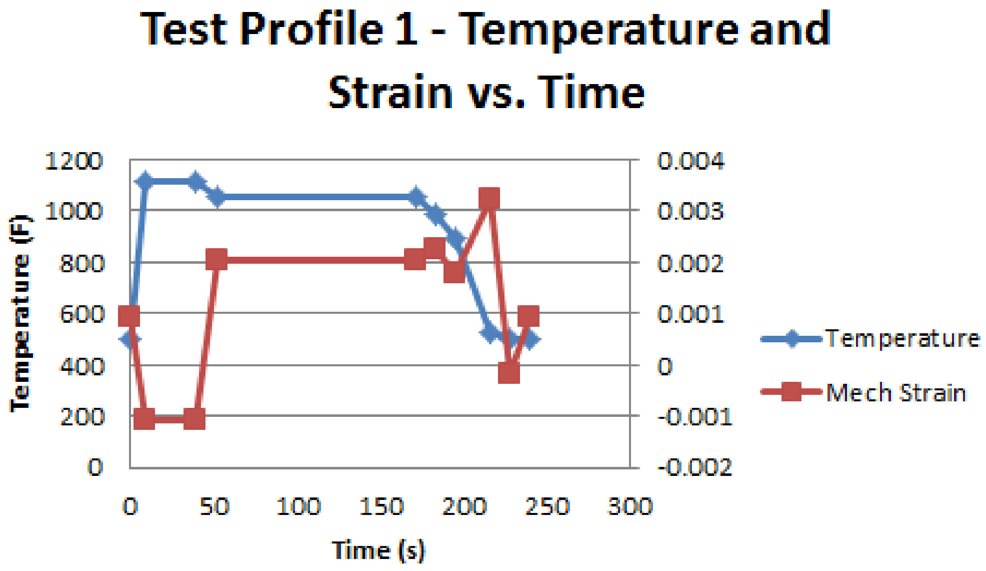


Figure F-6. Temperature and mechanical strain vs. time for test profile 1

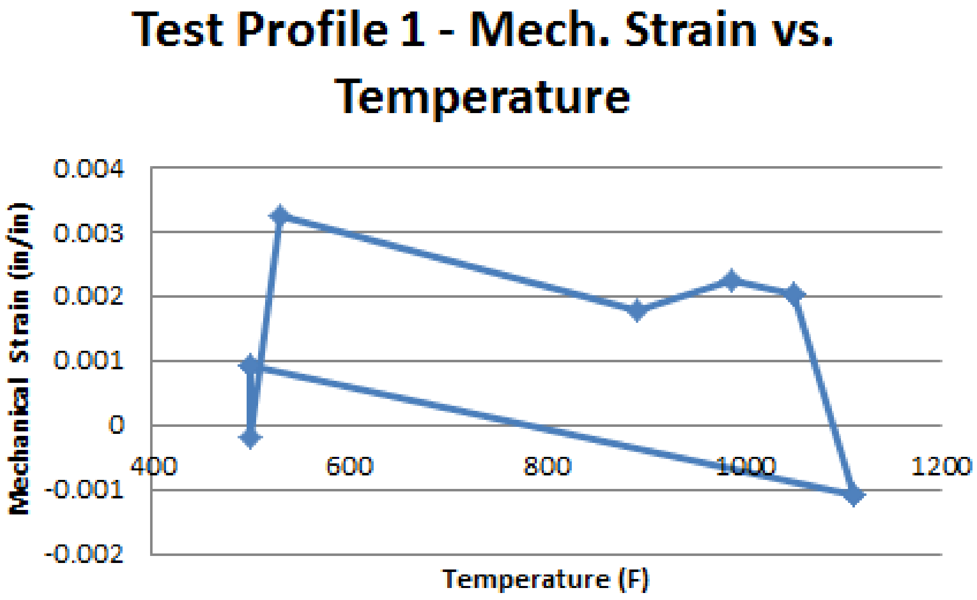


Figure F-7. Mechanical strain vs. temperature for test profile 1

The second test condition is also based on the mission used for the RISC Axial Slot Bottom test case with two key changes. The first modification reduced all the strains of the mission profile by the same amount such that the maximum applied strain was zero. This results in an *R*-ratio equal to negative infinity. The second modification was to eliminate the 120-second dwell to shorten the cycle. The compressive stresses and temperatures for the dwell segment would not be high enough to contribute to time-dependent crack growth or RS due to creep. Table F-2 contains the loading conditions for each time step. Figure F-8 is a graph of strain and temperature profile for the test cycle, whereas figure F-9 shows the mechanical strain versus temperature.

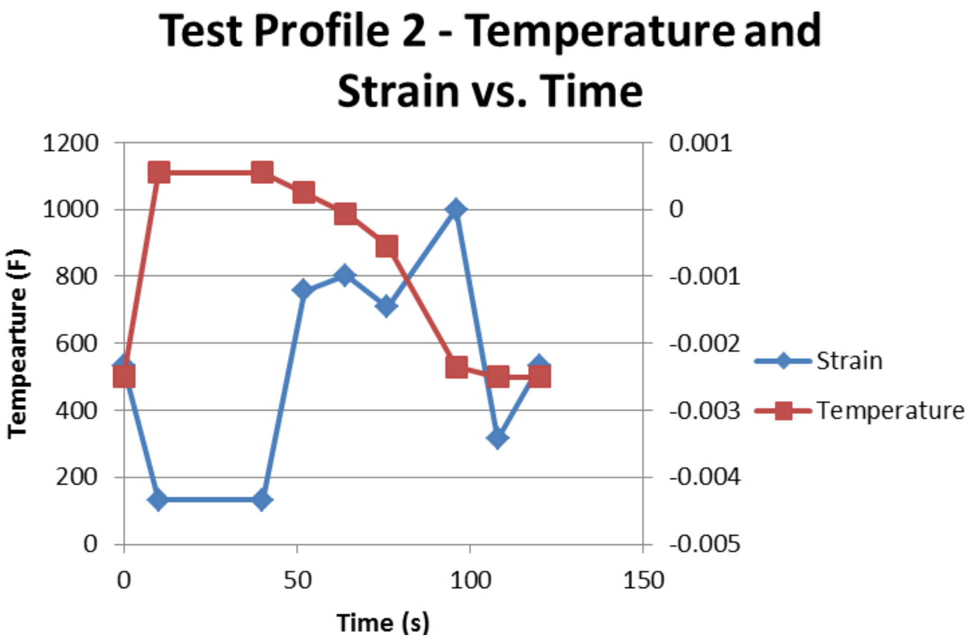


Figure F-8. Temperature and strain vs. time for test profile 2

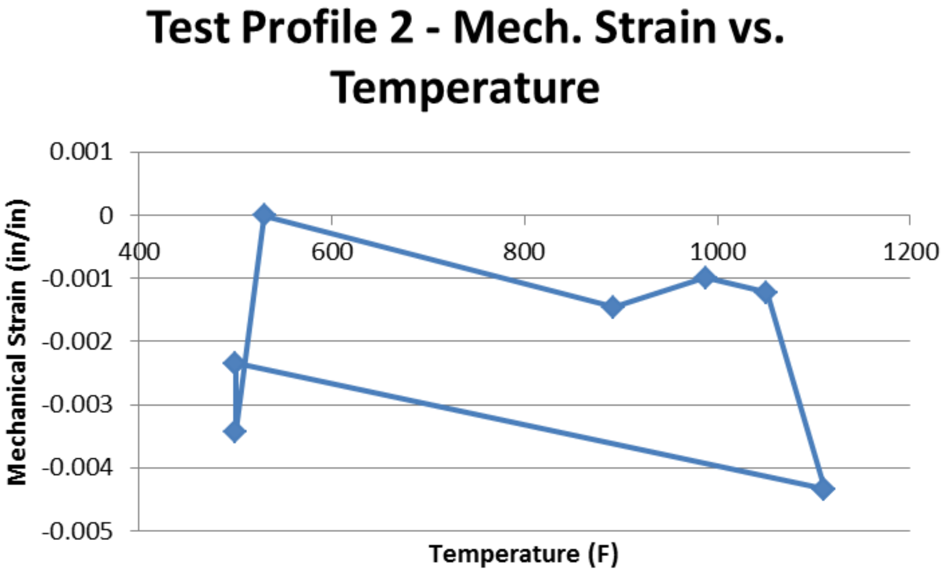


Figure F-9. Mechanical strain vs. temperature for test profile 2

F.4 STRESS INTENSITY FACTOR SOLUTIONS

One way to compare crack growth results between tests is to look at various measures of the stress intensity factor (K), such as K_{max} or ΔK, plotted against the crack growth rate. A literature solution for the exact specimen geometry is not available, but there are several that would provide a very close approximation [F-1, F-2]. However, when these literature solutions were compared, one solution did not agree with the others. Because of this disagreement among the solutions, a FRacture ANalysis Code 3D (FRANC3D) analysis was performed on the specimen geometry to calculate K for various crack lengths. Figure F-10 shows the results of the literature K-solutions in comparison with the results from FRANC3D.

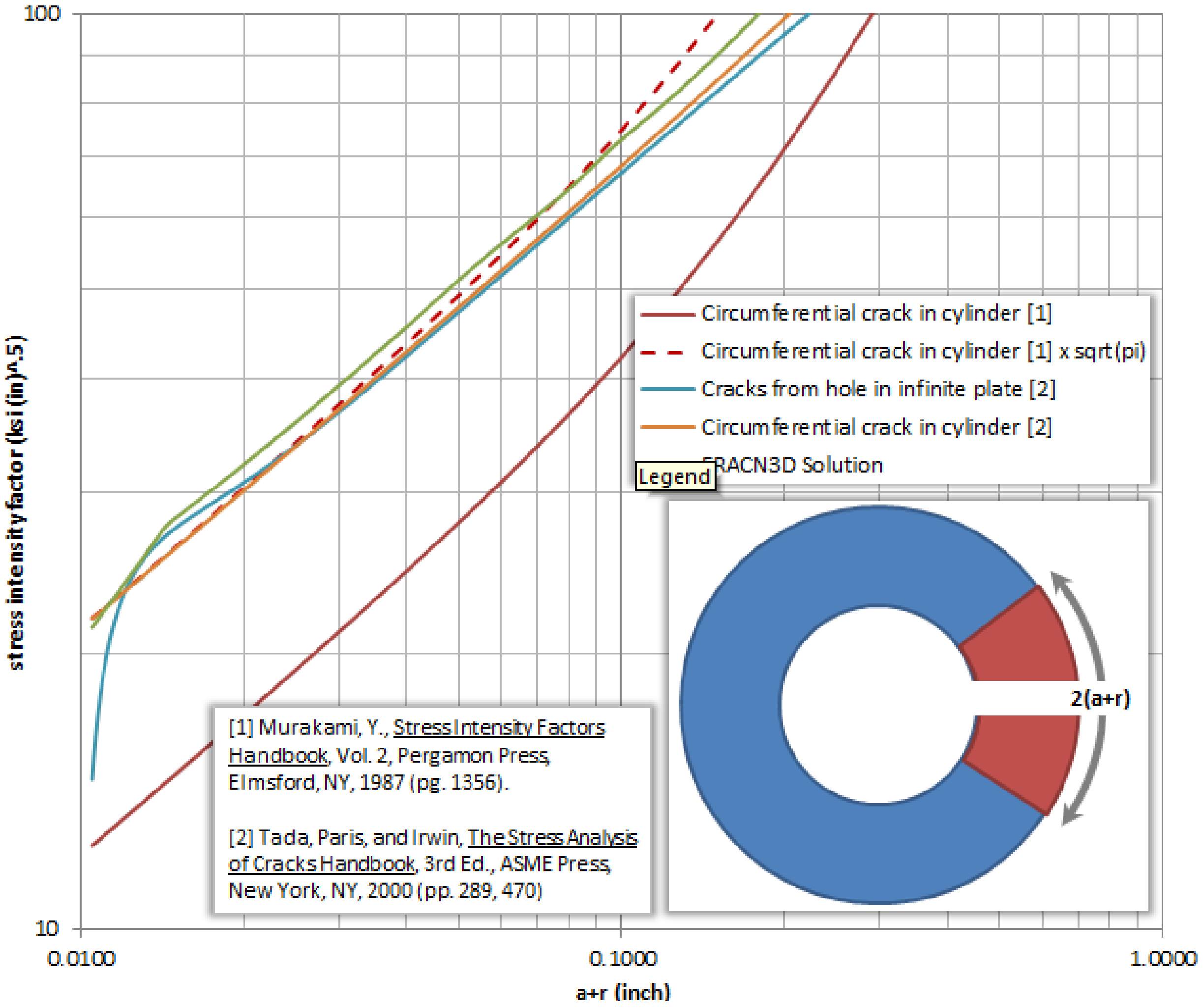


Figure F-10. Comparison of FRANC3D and literature *K*-solutions, including a corrected version of the Murakami solution

The Murakami solution was the obvious outlier. For very small circumferential cracks in a cylinder, the K-solution should approach the K for a crack in an infinite plate, $\sigma\sqrt{\pi a}$. The Murakami solution was apparently missing a $\sqrt{\pi}$ term from the solution. The authors have been notified to correct this mistake. If the formula is corrected in this manner, all the solutions come into good agreement, as also shown in figure F-10.

There is still a slight difference between all the available solutions (<10%). All the literature solutions assume that the stress intensity factor is constant through the thickness. Because the crack is growing faster on the outer diameter compared to the inner diameter, the K must be higher on the outer diameter. The FRANC3D analysis used the K solutions found near the outer diameter surface. Pratt & Whitney recommends that this solution be used when analyzing test data. Representative details are shown in table F-3.

Table F-3. FRANC3D *K* versus *a* table using a maximum stress of 120 ksi

2(a+r) (inch)	K (ksi in^.5)
0.0240	23.7616
0.0288	27.6243
0.0336	29.6665
0.0379	31.3261
0.0422	33.0177
0.0488	35.3753
0.0576	38.4471
0.0689	42.1373
0.0830	46.3797
0.1025	51.9765
0.1282	58.0031
0.1601	64.3455
0.1932	71.7824
0.2379	79.4468

F.5 RESULTS AND DISCUSSION

Results for the three tests performed under this program are summarized in table F-4. Specimen IDs 5725 and 1797 were tested using Test Profile 1, which mimicked the RISC Axial Slot test case mission. Specimen ID 3554 was tested using Test Profile 2, the negative infinity test profile. Because of the length of the first test using Test Profile 2, repeating the second test for the same test profile was not possible under the budget and schedule of this program. Repeating this test condition would not have provided significant value beyond what was learned in the first test.

Table F-4. Summary of test results

Specimen ID	Tmax (F)	Tmin (F)	Dwell Time (sec)	Max Strain (%)	Min Strain (%)	Strain Range (%)	Cycle	Notes
5725	1111	500	120	0.325	-0.109	0.434	6304	Full crack growth curve
1797	1111	500	120	0.325	-0.1109	0.434	26841	Specimen failed in the button head
3554	1111	500	0	0	-0.434	0.434	67293	Test stopped before failure

F.5.1 TEST PROFILE 1 RESULTS

The goal of Test Profile 1 was to show the impact of the 120-second dwell. The results to these tests were compared to two additional tests completed under a Pratt & Whitney funded project at GA Tech. The two tests completed at GA Tech had the same strain and temperature profile except for the 120-second dwell. This would provide four tests run under the same loading conditions, two with the 120-second dwell and two without the 120-second dwell.

Specimen ID 5725 ran successfully to completion. Unfortunately, Specimen ID 1797 failed outside the gage section in the button head of the specimen. No valid crack growth data came from this test. The detailed lab report for Specimen ID 5725 is provided in addendum F-1 at the end of this appendix.

Specimen ID 5725 crack growth data were compared to the results from the GA Tech tests without the 120-second dwell included in the test cycle. Figure F-11 is a graph of the crack length versus the number of cycles. The initial flaw size selected to start the graph was the smallest common crack length measured for all three specimens. As can be seen from figure F-12, the results from Specimen ID 5725 (with the dwell) fell within the scatter of the two crack growth test results without the dwell. This would suggest that the 120-second dwell did not have a significant impact on crack growth at that dwell condition.

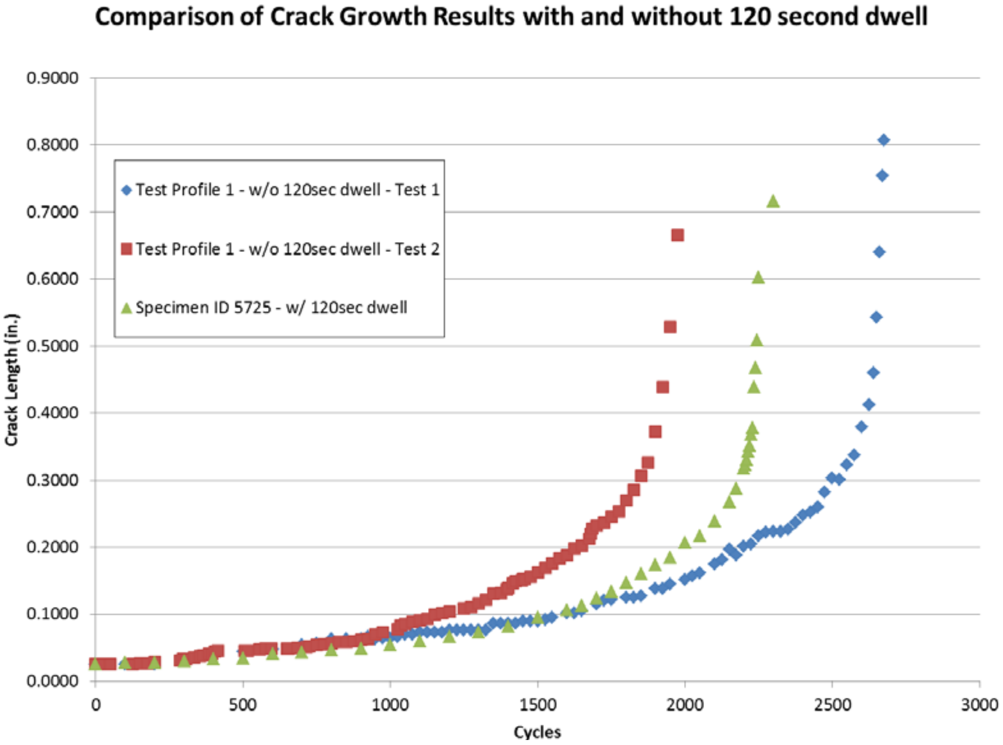


Figure F-11. Crack growth results with and without 120-second dwell

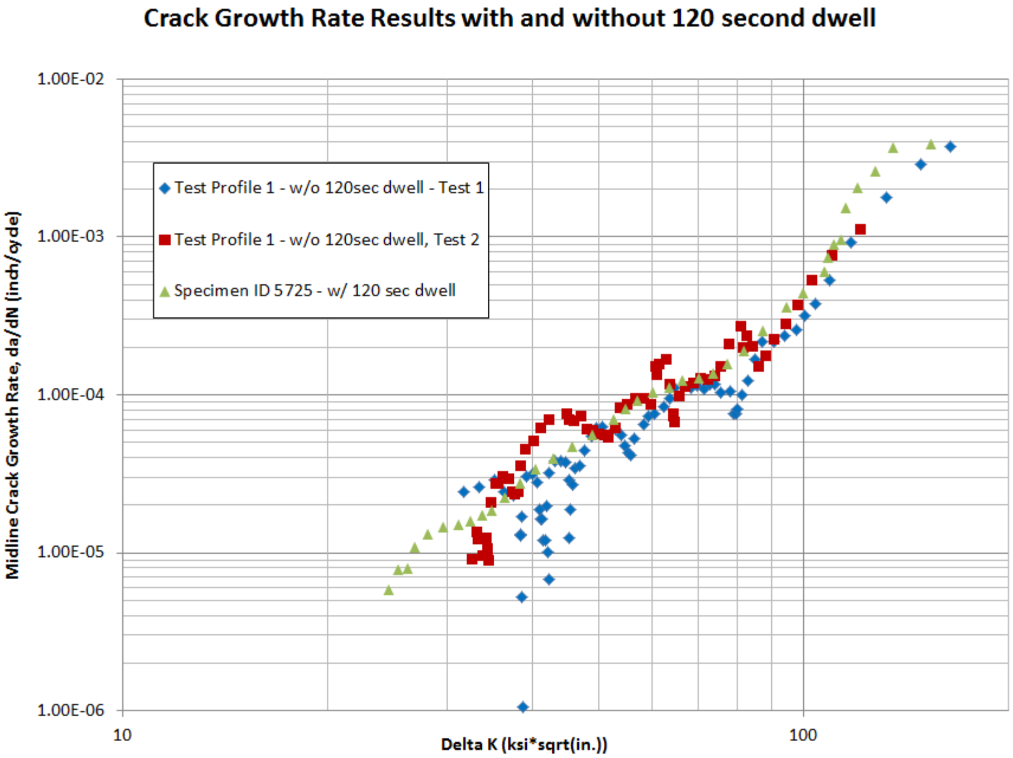


Figure F-12. Delta *K* vs. *da/dN* with and without 120-second dwell

Using the *K* solution discussed previously, a Δ*K* vs. *da/dN* graph of the test data was created. In this graph, K_{max} and K_{min} values were calculated for each crack growth measurement. The number of cycles between each crack growth measurement is known, so *da/dN* for each measurement can be calculated. To take some of the noise out of the data, a seven-point polynomial smoothing routine outlined in ASTM E647 was used. The resulting data are shown in figure F-12.

If the 120-second dwell were to have an impact on accelerating the crack growth, it would be shown in the graph by having a higher *da/dN* value for the same Δ*K*. As can be seen in the graph, this was not the case. The blue diamond and the red squares in the graph are the two tests that did not have a 120-second dwell. The green triangles are from the test that ran the same cycle with the addition of the 120-second dwell. The results of that test agree well with the two tests that did not have the 120-second dwell.

Unfortunately, the test for Specimen ID 1797 did not end successfully. The specimen failed outside of the gage section at the button head of the specimen where the load is applied. As a result the data received from this specimen are not considered valid.

F.5.2 TEST PROFILE 2 RESULTS

The goal of Test Profile 2 was to investigate whether a crack would grow under a fully compressive applied strain profile. The test was designed to show that even if the applied loads are all compressive, tensile RS will build up because of compressive yielding and creep to a level sufficient to grow a crack. The test profile also provides a limit case for model validation. Results from this test could be used for model validation extending the *R*-ratio range for a candidate crack growth model down to an *R*-ratio of negative infinity.

Only one test, Specimen ID 3554, was performed using Test Profile 2 because the test ran much longer than anticipated. After more than 67,000 cycles, the test was halted with the crack growing slowly. The detailed lab report for Specimen ID 3554 is provided in addendum F-2 at the end of this appendix.

When the test conditions were selected, the tensile RS were expected to be higher than occurred during the test. The initial crack growth rates were not surprising, but it was expected that the residual tensile stresses would evolve more rapidly providing the necessary tensile stress to grow the crack more rapidly as the crack grew larger. It is unclear as of this writing why the stresses did not evolve as expected.

Even though the stresses did not evolve as expected and the crack grew slower than expected as a result, the test is still valuable. The test did show that a crack will grow under purely compressive strain conditions. This should provide caution to the designer when performing crack growth analysis. Deep compressive loads cannot be ignored; the full loading profile should be assessed. Secondly, this test will be able to provide a valuable crack growth model verification result that will bound the design space for deep negative *R*-ratios.

F.6 ACKNOWLEDGEMENTS

DRL is a small research laboratory focused on quality testing with specific expertise in thermomechanical fatigue testing of both conventional and non-conventional testing protocols and test geometries. Pratt & Whitney thanks DRL for the donation of rig time over and above the purchase order amount to produce the results in this report. DRL’s professionalism and focus on results were greatly appreciated in this effort.

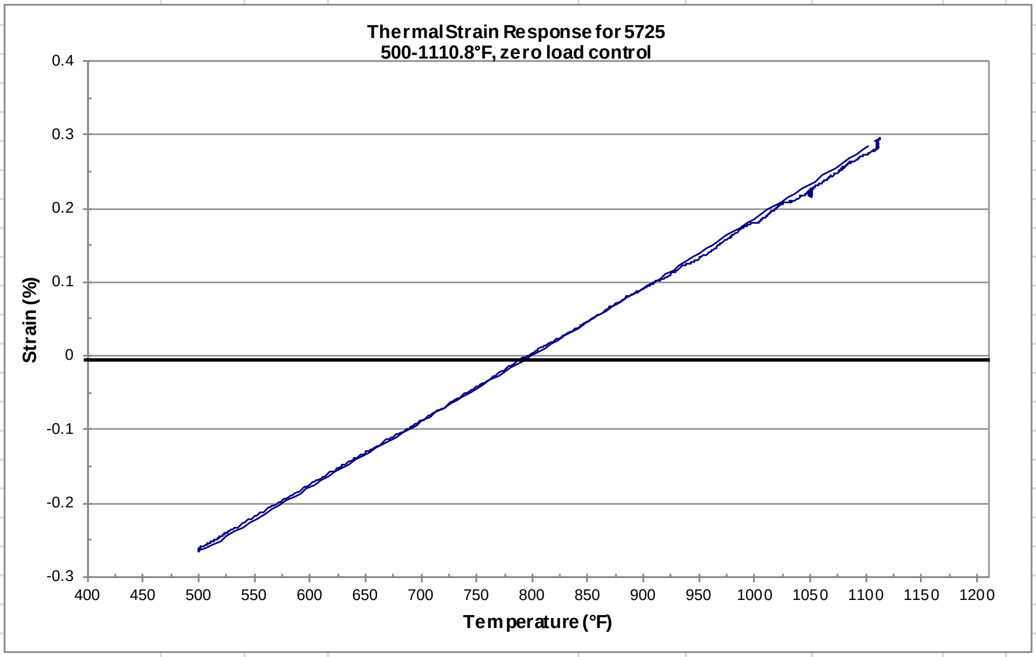
F.7 REFERENCES

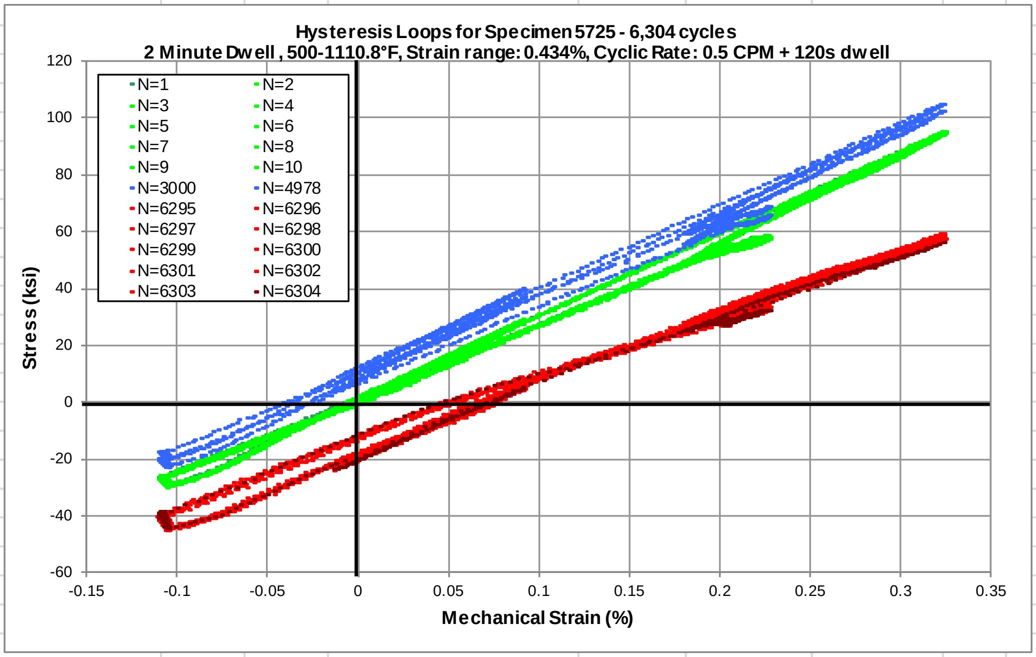
F-1. Murakami, Y. (1987). *Stress Intensity Factors Handbook* (Vol. 2)., Elmsford, NY:, Pergamon Press. pg. 1356.

F-2. Tada, H., Paris, P.C., and Irwin, G.R. (2000).,*The Stress Analysis of Cracks Handbook*, (3rd Ed.). New York, NY: ASME Press, (289, 470).

ADDENDUM F-1—LAB REPORT FOR SPECIMEN ID 5725

Description:					
Test Type:	2 Minute Dwell	**Specimen ID:**	5725	**Material:**	INCO718
Outside diameter:	0.5490 in	**Specimen type:**	MT-41--notched	**Environment:**	Air
Inside diameter:	0.4493 in	**Strain range:**	0.434%	**Tmax:**	1111 °F
Area:	0.0782 in²	**Max strain:**	+0.325%	**Tmin:**	500 °F
		Min strain:	-0.109%	**Dwell:**	120s
				Cyclic Rate:	0.5 CPM
N_f:	6304 cycles	**Reference:**	MMD 20155234		
					Page 1 of 6

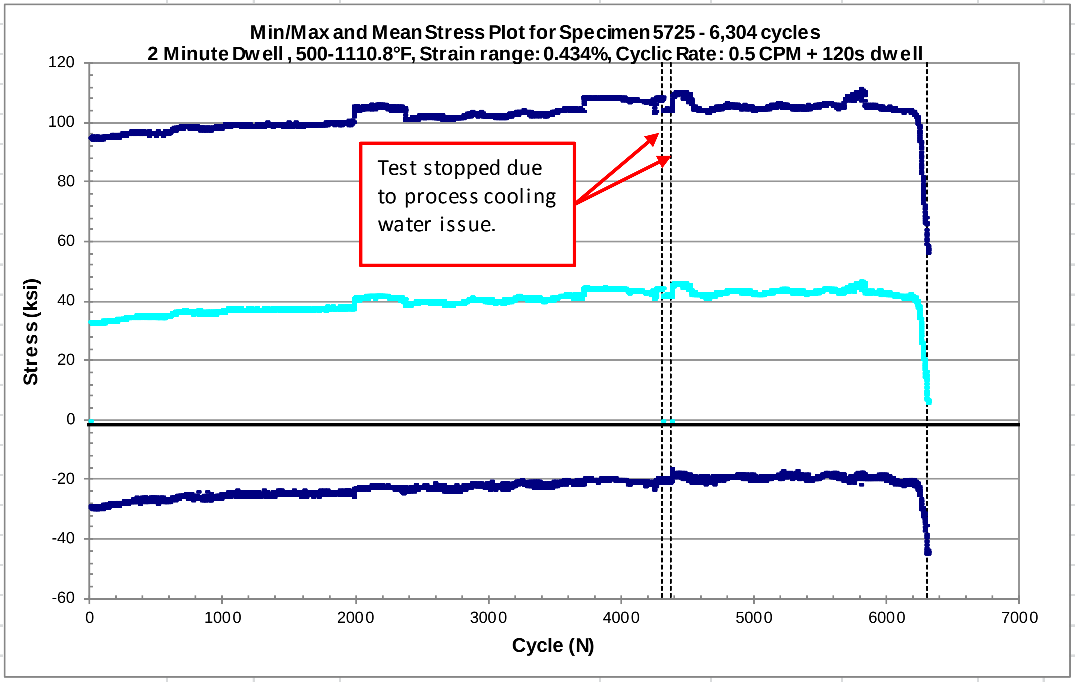




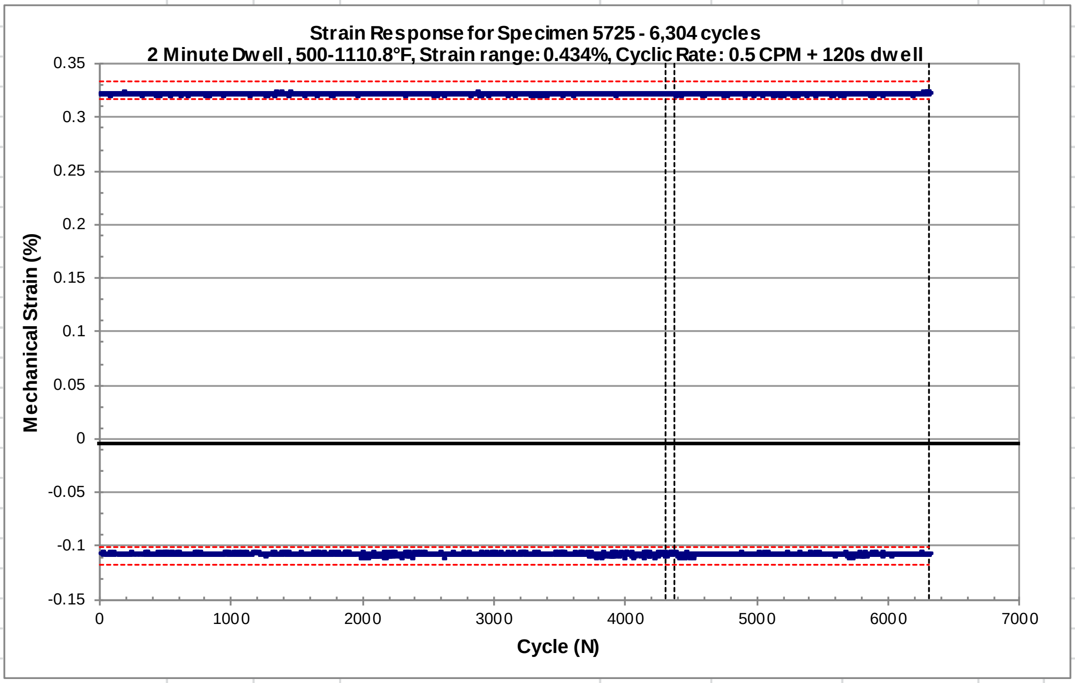
Description:

Test Type:	2 Minute Dwell	**Specimen ID:**	5725	**Material:**	INCO718
Outside diameter:	0.5490 in	**Specimen type:**	MT-41--notched	**Environment:**	Air
Inside diameter:	0.4493 in	**Strain range:**	0.434%	**Tmax:**	1111 °F
Area:	0.0782 in²	**Max strain:**	+0.325%	**Tmin:**	500 °F
		Min strain:	-0.109%	**Dwell:**	120s
				Cyclic Rate:	0.5 CPM
N_f:	6304 cycles	**Reference:**	MMD 20155234		

Page 2 of 6


Min/Max and Mean Stress Plot for Specimen 5725 - 6,304 cycles
2 Minute Dwell , 500-1110.8°F, Strain range: 0.434%, Cyclic Rate: 0.5 CPM + 120s dwell

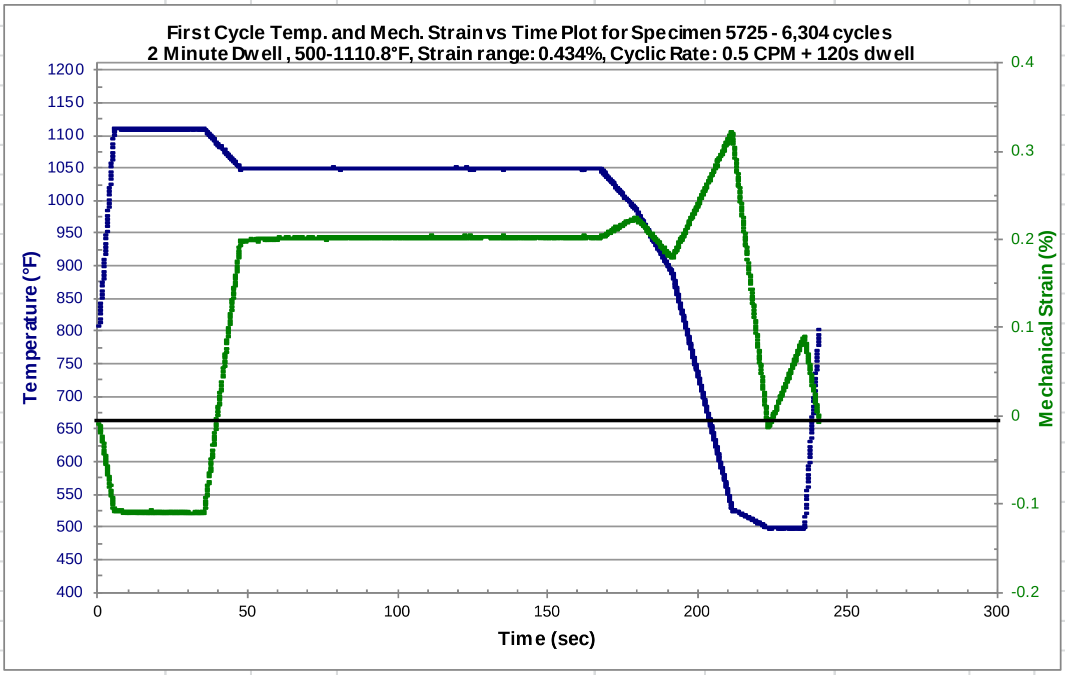
Test stopped due to process cooling water issue.

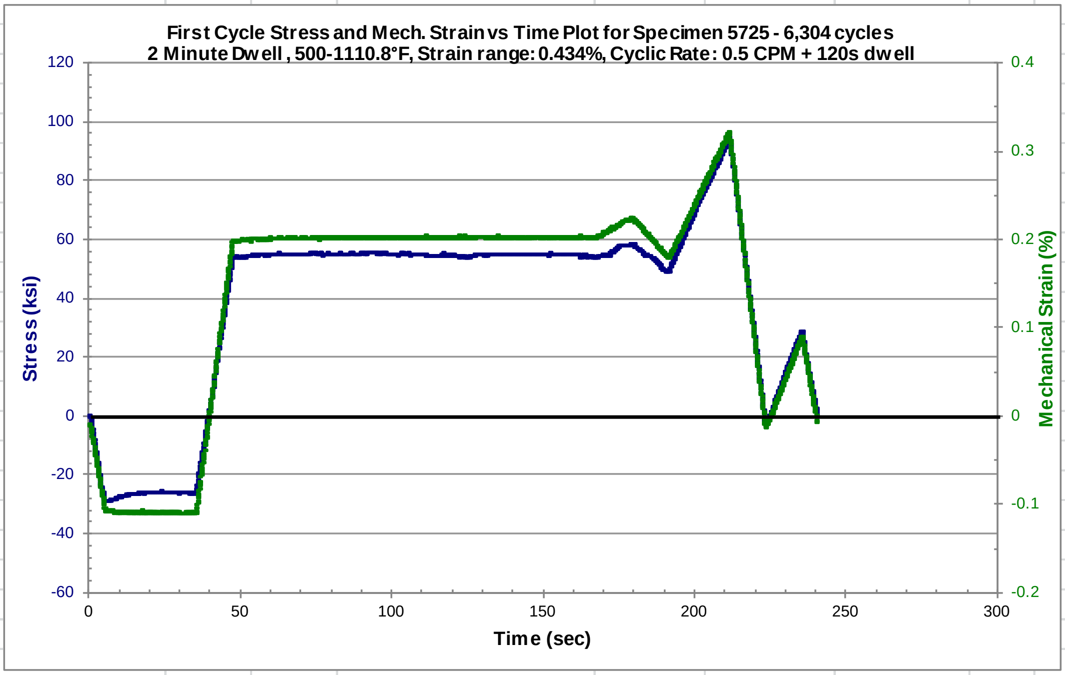

Strain Response for Specimen 5725 - 6,304 cycles
2 Minute Dwell , 500-1110.8°F, Strain range: 0.434%, Cyclic Rate: 0.5 CPM + 120s dwell

Description:

Test Type:	2 Minute Dwell	**Specimen ID:**	5725	**Material:**	INCO718
Outside diameter:	0.5490 in	**Specimen type:**	MT-41--notched	**Environment:**	Air
Inside diameter:	0.4493 in	**Strain range:**	0.434%	**Tmax:**	1111 °F
Area:	0.0782 in^2	**Max strain:**	+0.325%	**Tmin:**	500 °F
		Min strain:	-0.109%	**Dwell:**	120s
				Cyclic Rate:	0.5 CPM
N_f:	6304 cycles	**Reference:**	MMD 20155234		

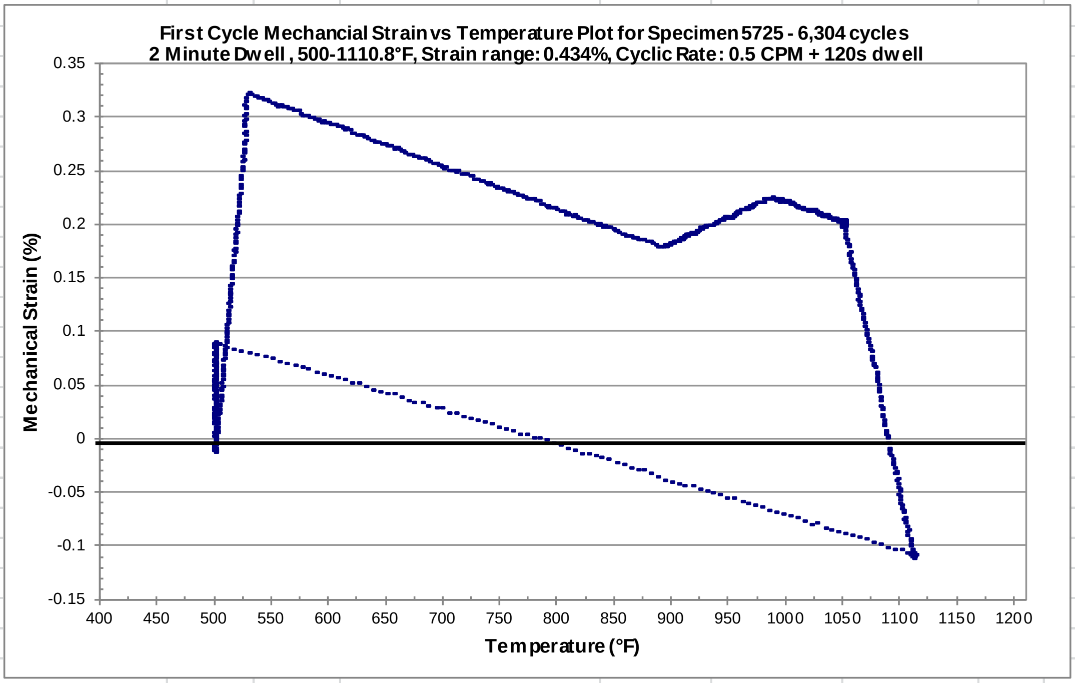
Page 3 of 6

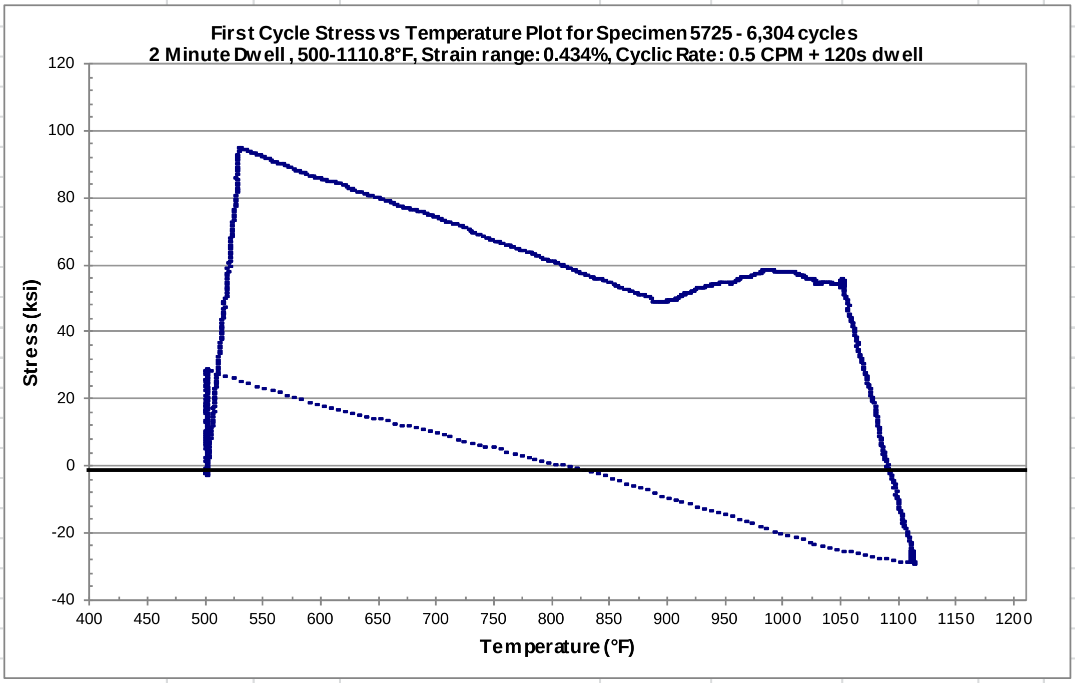

First Cycle Temp. and Mech. Strain vs Time Plot for Specimen 5725 - 6,304 cycles
2 Minute Dwell, 500-1110.8°F, Strain range: 0.434%, Cyclic Rate: 0.5 CPM + 120s dwell


First Cycle Stress and Mech. Strain vs Time Plot for Specimen 5725 - 6,304 cycles
2 Minute Dwell, 500-1110.8°F, Strain range: 0.434%, Cyclic Rate: 0.5 CPM + 120s dwell

Description:						
Test Type:	2 Minute Dwell		Specimen ID:	5725	Material:	INCO718
Outside diameter:	0.5490	in	Specimen type:	MT-41--notched	Environment:	Air
Inside diameter:	0.4493	in	Strain range:	0.434%	Tmax:	1111 °F
Area:	0.0782	in²	Max strain:	+0.325%	Tmin:	500 °F
			Min strain:	-0.109%	Dwell:	120s
					Cyclic Rate:	0.5 CPM
N_f:	6304	cycles	Reference:	MMD 20155234		

Page 4 of 6

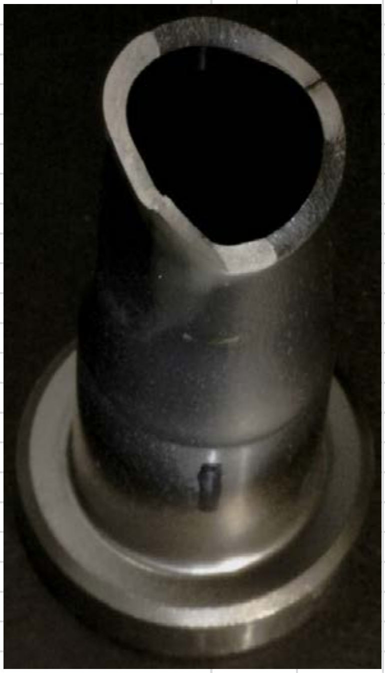


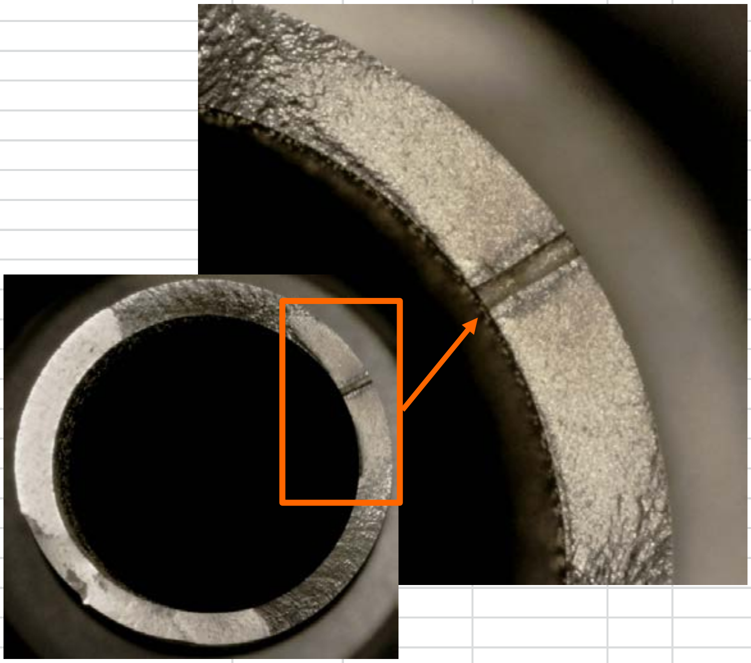


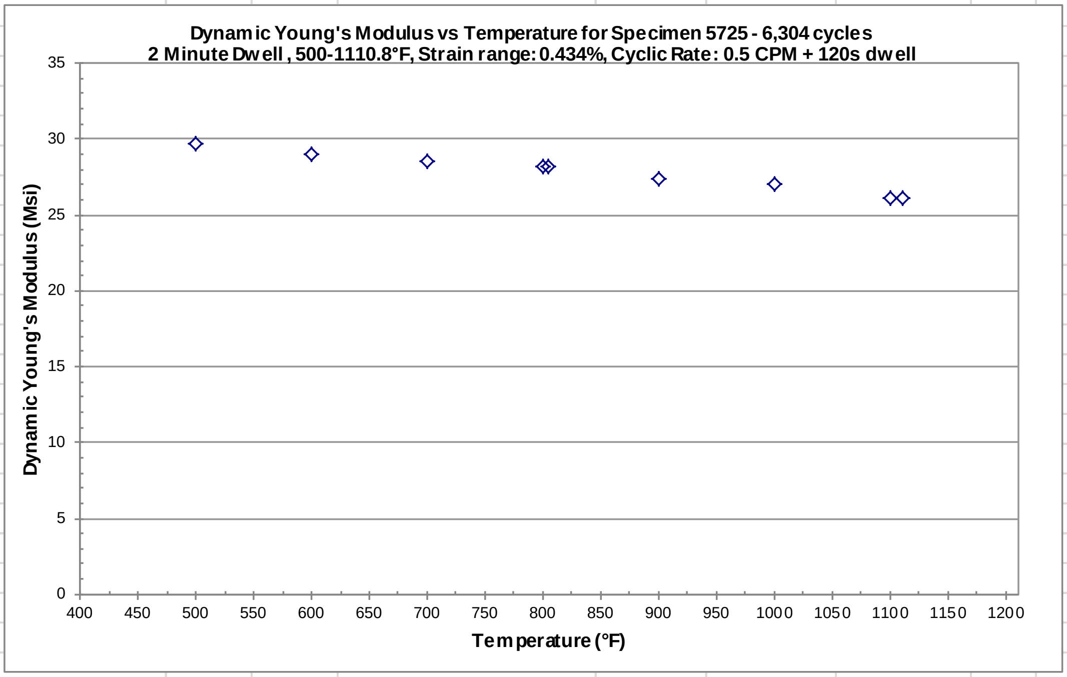
Description:

Test Type:	2 Minute Dwell	**Specimen ID:**	5725	**Material:**	INCO718
Outside diameter:	0.5490 in	**Specimen type:**	MT-41--notched	**Environment:**	Air
Inside diameter:	0.4493 in	**Strain range:**	0.434%	**Tmax:**	1111 °F
Area:	0.0782 in^2	**Max strain:**	+0.325%	**Tmin:**	500 °F
		Min strain:	-0.109%	**Dwell:**	120s
				Cyclic Rate:	0.5 CPM
N_f:	6304 cycles	**Reference:**	MMD 20155234		

Page 5 of 6



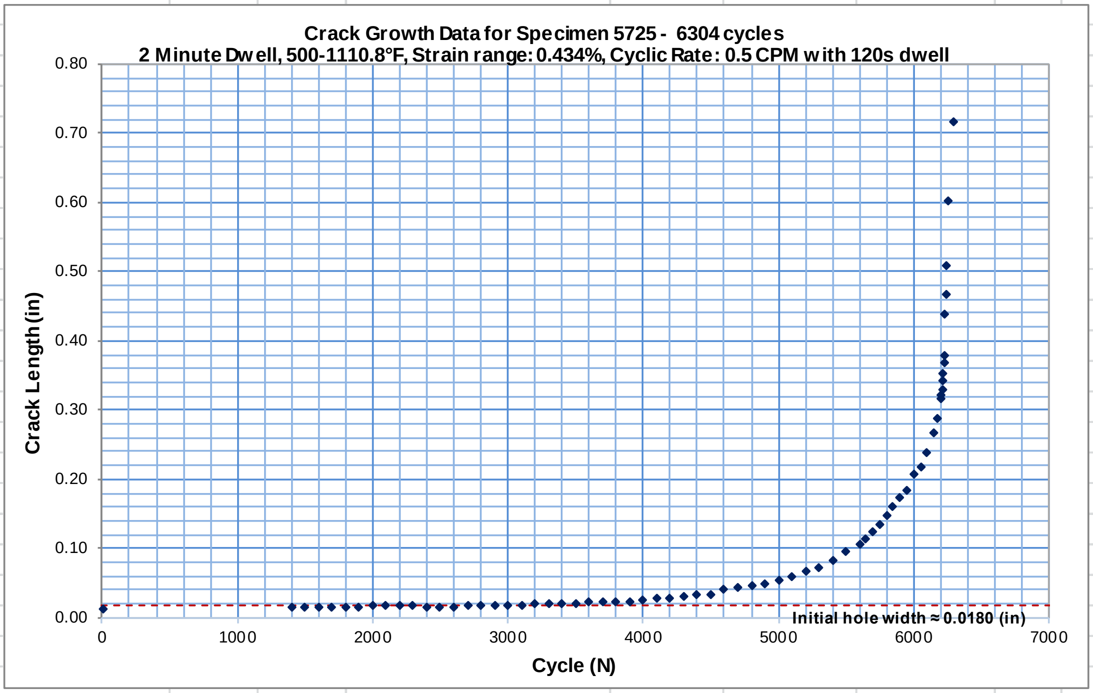


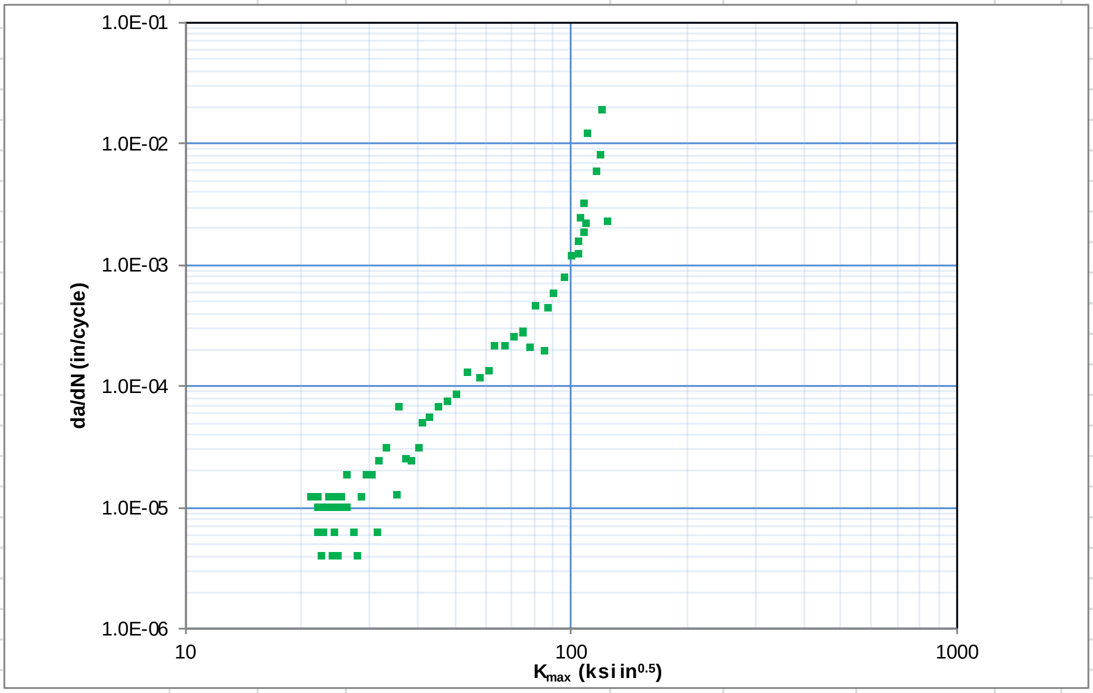

Dynamic Young's Modulus vs Temperature for Specimen 5725 - 6,304 cycles
2 Minute Dwell, 500-1110.8°F, Strain range: 0.434%, Cyclic Rate: 0.5 CPM + 120s dwell
Dynamic Young's Modulus (Msi)
35
30
25
20
15
10
5
0
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
1150
1200
Temperature (°F)

Description:

Test Type:	2 Minute Dwell		**Specimen ID:**	5725	**Material:**	INCO718	
Outside diameter:	0.5490	in	**Specimen type:**	MT-41--notched	**Environment:**	Air	
Inside diameter:	0.4493	in	**Strain range:**	0.434%	**Tmax:**	1111	°F
Area:	0.0782	in²	**Max strain:**	+0.325%	**Tmin:**	500	°F
			Min strain:	-0.109%	**Dwell:**	120s	
					Cyclic Rate:	0.25	CPM
N_f:	6304	cycles	**Reference:**	MMD 20155234			

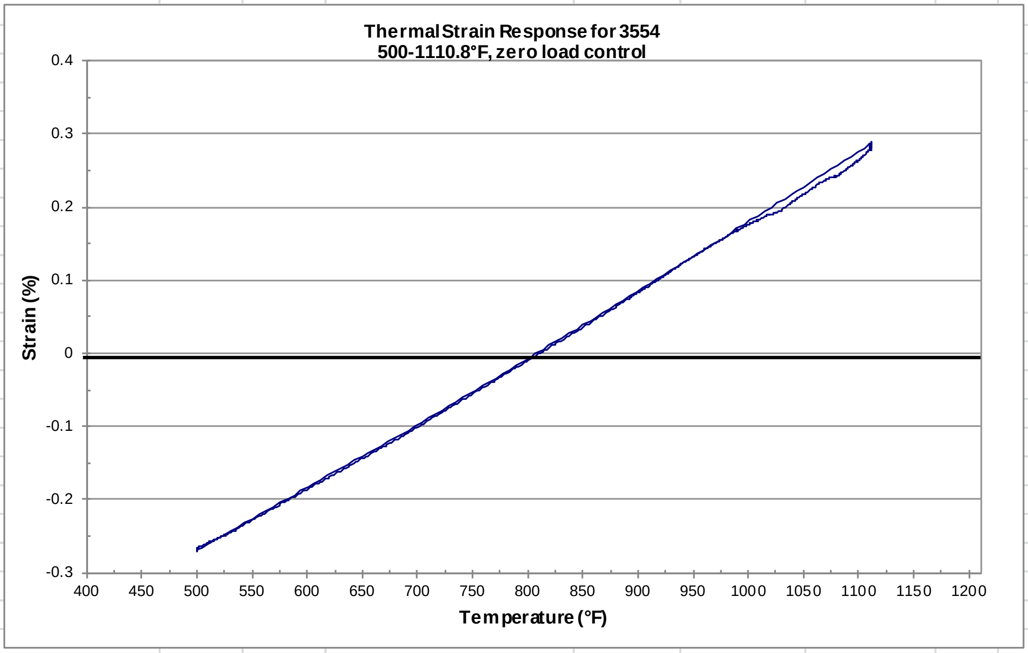
Page 6 of 6

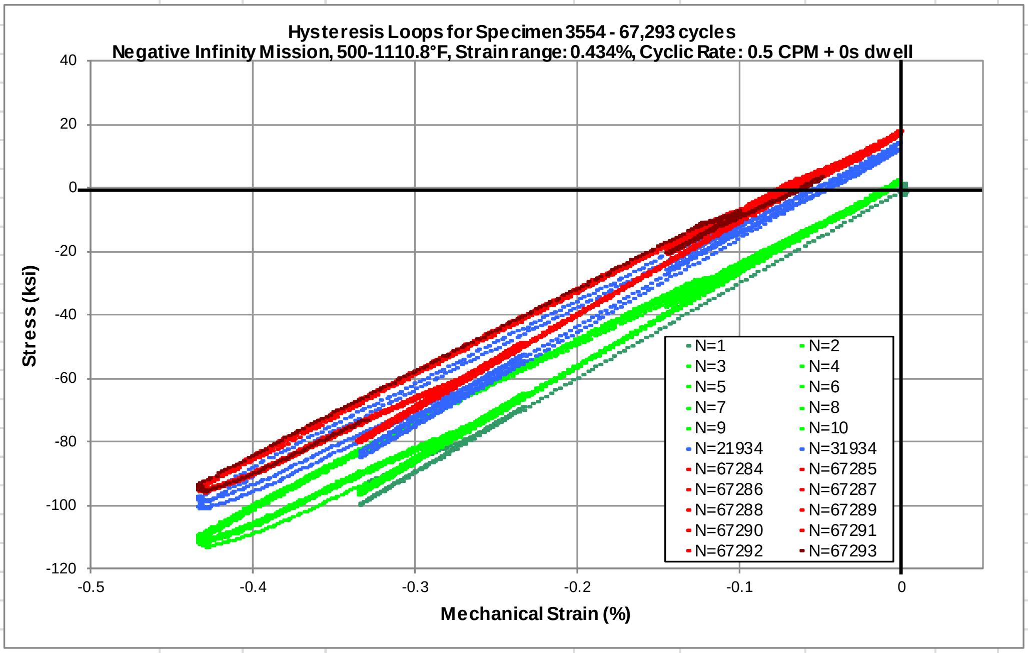




Description:

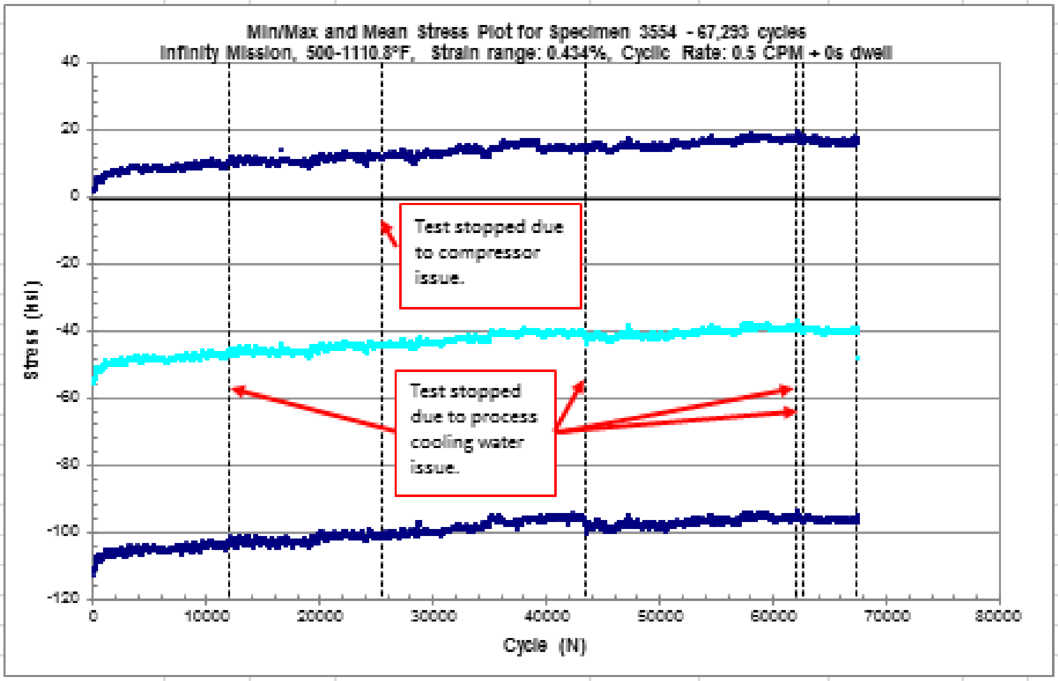
Test Type:	Infinity Mission		**Specimen ID:**	3554	**Material:**	INCO718	
Outside diameter:	0.5505	in	**Specimen type:**	MT-41--notched	**Environment:**	Air	
Inside diameter:	0.4484	in	**Strain range:**	0.434%	**Tmax:**	1111	°F
Area:	0.0801	in²	**Max strain:**	+0%	**Tmin:**	500	°F
			Min strain:	-0.434%	**Dwell:**	0s	
					Cyclic Rate:	0.5	CPM
N_f:	67293	cycles	**Reference:**	MMD 20155234			
	Specimen discontinued at P&W's request					Page 1 of 6	





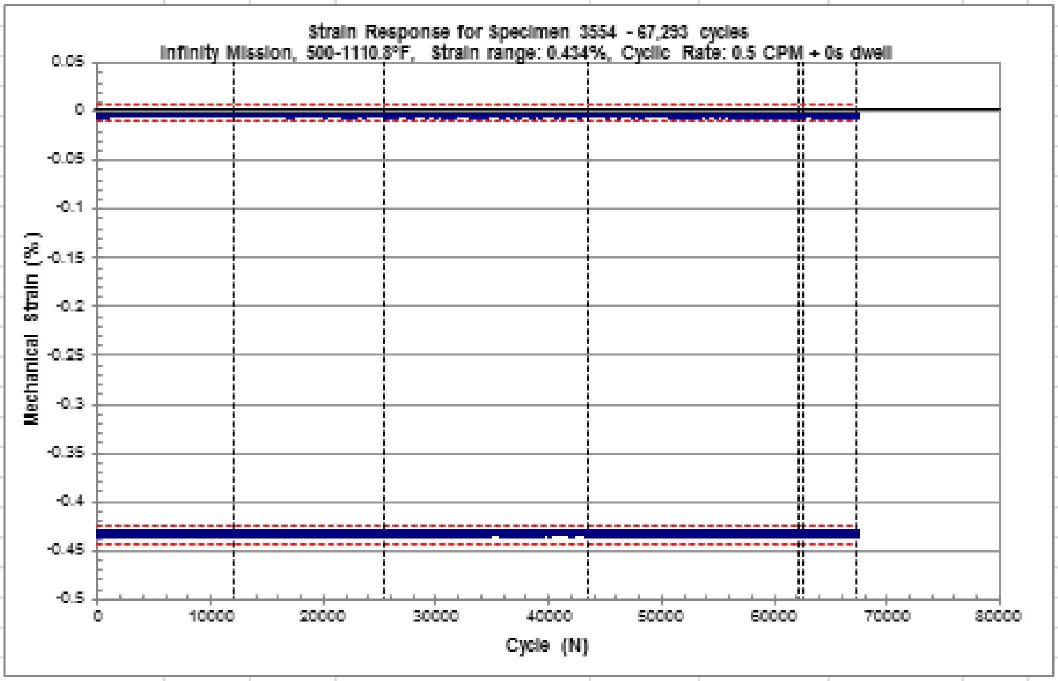
Description:

Test Type:	Infinity Mission		**Specimen ID:**	3554	**Material:**	INCO718	
utside diameter:	#####	in	**Specimen type:**	MT-41--notched	**Environment:**	Air	
Inside diameter:	#####	in	**Strain range:**	0.434%	**Tmax:**	1111	°F
Area:	0.0801	in²	**Max strain:**	+0%	**Tmin:**	500	°F
			Min strain:	-0.434%	**Dwell:**	0s	
					Cyclic Rate:	0.5	CPM
N_f:	67293	cycles	**Reference:**	MMD 20155234			
	Specimen discontinued at P&W's request					Page 2 of 6	


Min/Max and Mean Stress Plot for Specimen 3554 - 67,293 cycles
Infinity Mission, 500-1110.8°F, Strain range: 0.434%, Cyclic Rate: 0.5 CPM + 0s dwell

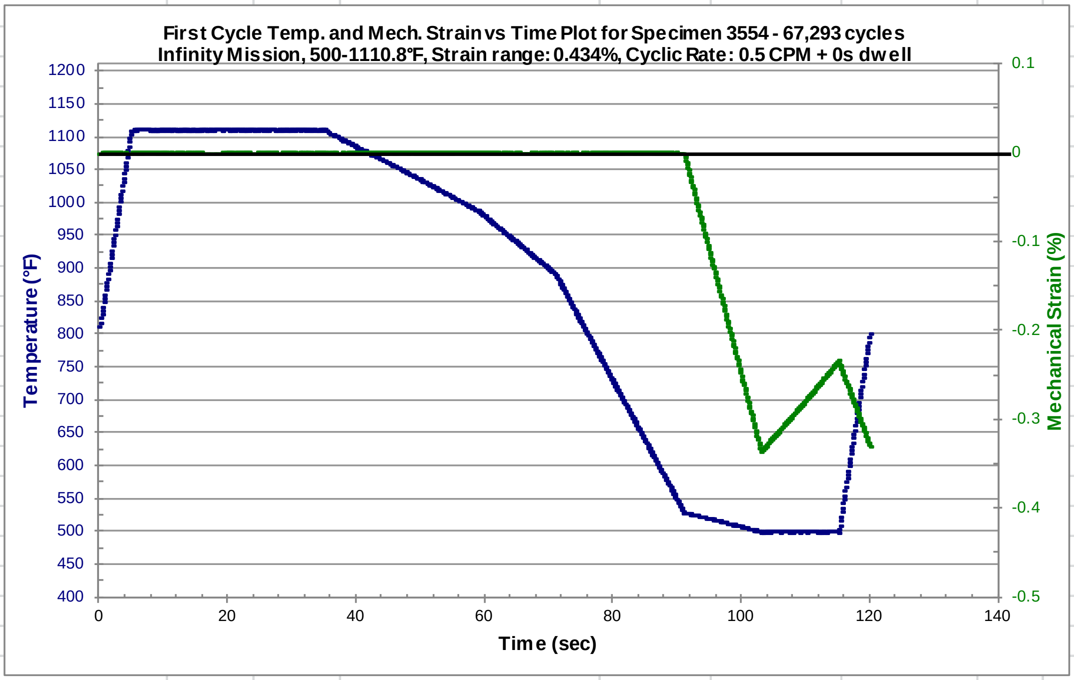
Test stopped due to compressor issue.

Test stopped due to process cooling water issue.

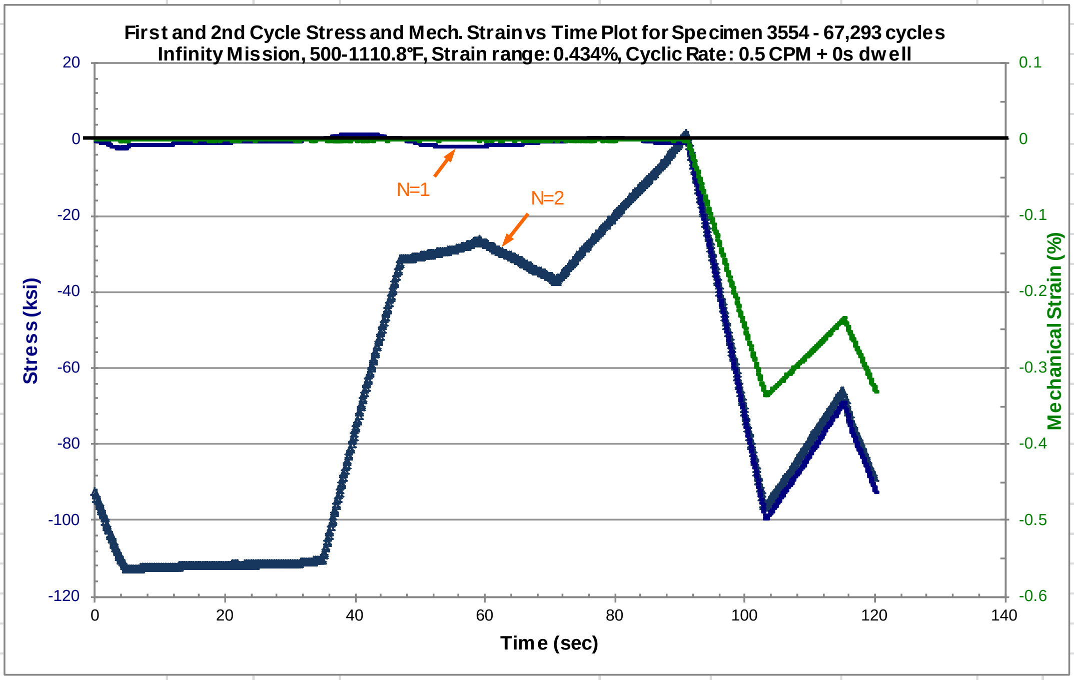

Strain Response for Specimen 3554 - 67,293 cycles
Infinity Mission, 500-1110.8°F, Strain range: 0.434%, Cyclic Rate: 0.5 CPM + 0s dwell

Description:

Test Type:	Infinity Mission	**Specimen ID:**	3554	**Material:**	INCO718
Outside diameter:	0.5505 in	**Specimen type:**	MT-41--notched	**Environment:**	Air
Inside diameter:	0.4484 in	**Strain range:**	0.434%	**Tmax:**	1111 °F
Area:	0.0801 in²	**Max strain:**	+0%	**Tmin:**	500 °F
		Min strain:	-0.434%	**Dwell:**	0s
				Cyclic Rate:	0.5 CPM
N$_f$:	67293 cycles	**Reference:**	MMD 20155234		
	Specimen discontinued at P&W's request				Page 3 of 6

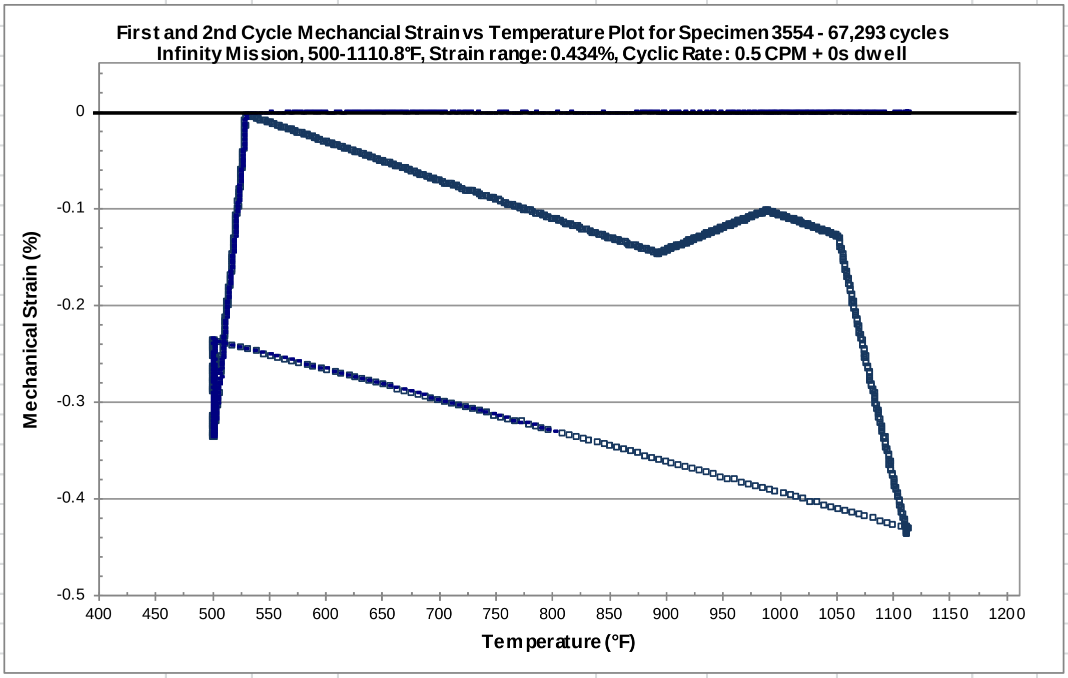


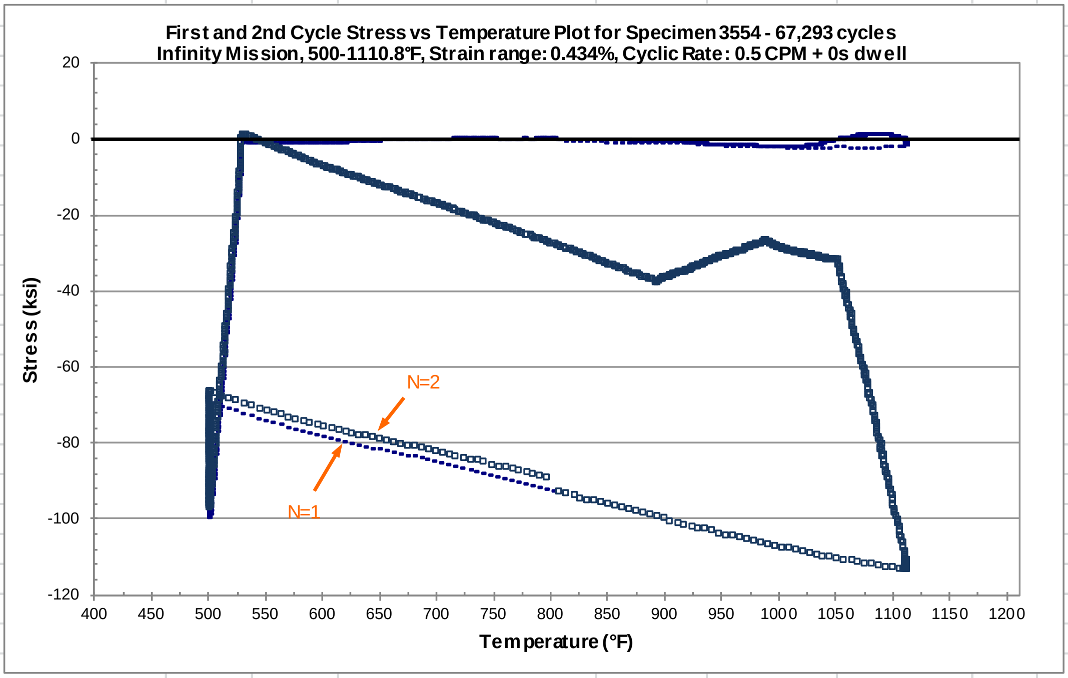
First Cycle Temp. and Mech. Strain vs Time Plot for Specimen 3554 - 67,293 cycles
Infinity Mission, 500-1110.8°F, Strain range: 0.434%, Cyclic Rate: 0.5 CPM + 0s dwell



First and 2nd Cycle Stress and Mech. Strain vs Time Plot for Specimen 3554 - 67,293 cycles
Infinity Mission, 500-1110.8°F, Strain range: 0.434%, Cyclic Rate: 0.5 CPM + 0s dwell

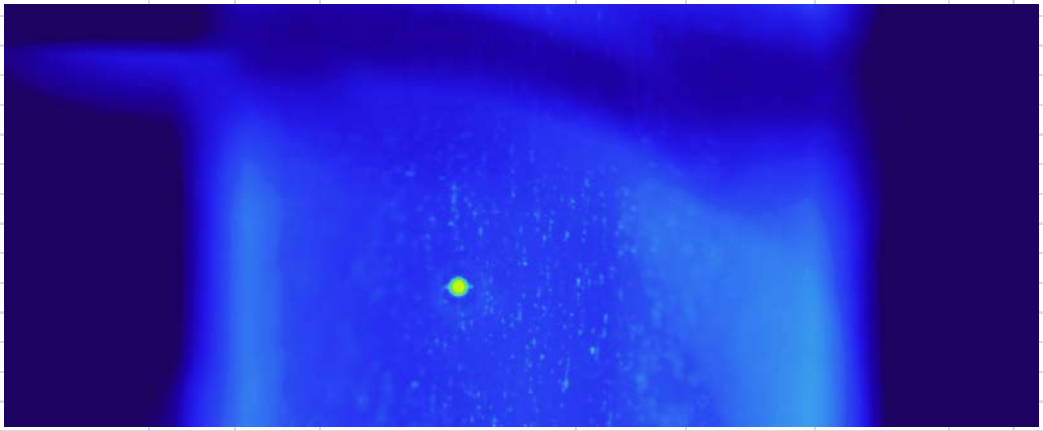
Description:					
Test Type:	Infinity Mission	**Specimen ID:**	3554	**Material:**	INCO718
Outside diameter:	0.5505 in	**Specimen type:**	MT-41--notched	**Environment:**	Air
Inside diameter:	0.4484 in	**Strain range:**	0.434%	**Tmax:**	1111 °F
Area:	0.0801 in²	**Max strain:**	+0%	**Tmin:**	500 °F
		Min strain:	-0.434%	**Dwell:**	0s
				Cyclic Rate:	0.5 CPM
N_f:	67293 cycles	**Reference:**	MMD 20155234		
	Specimen discontinued at P&W's request				Page 4 of 6



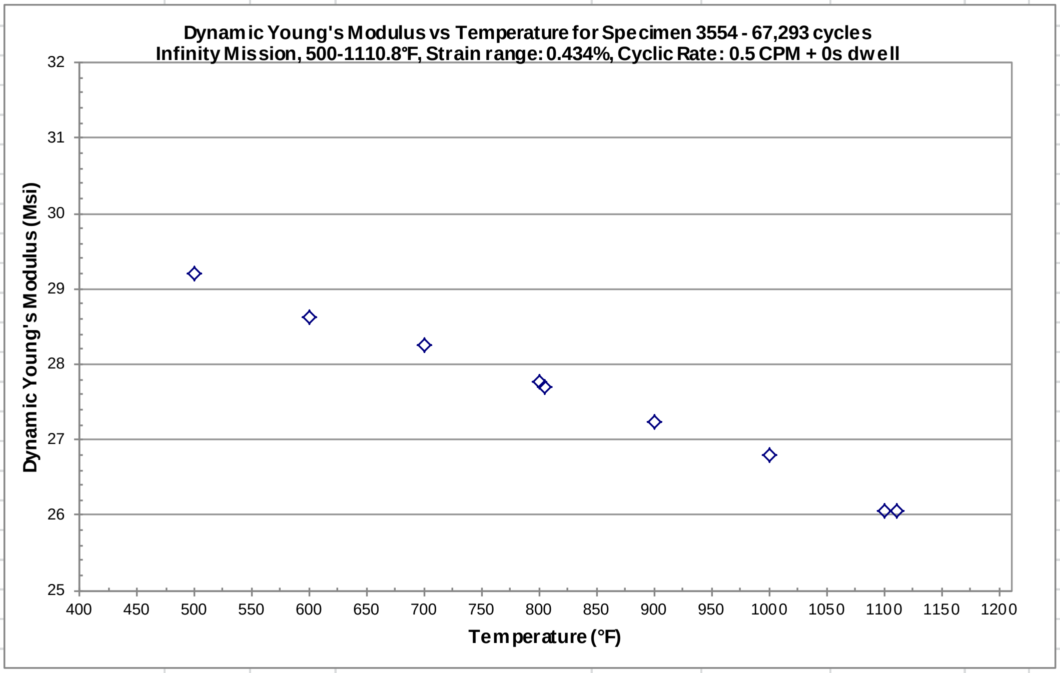


Description:

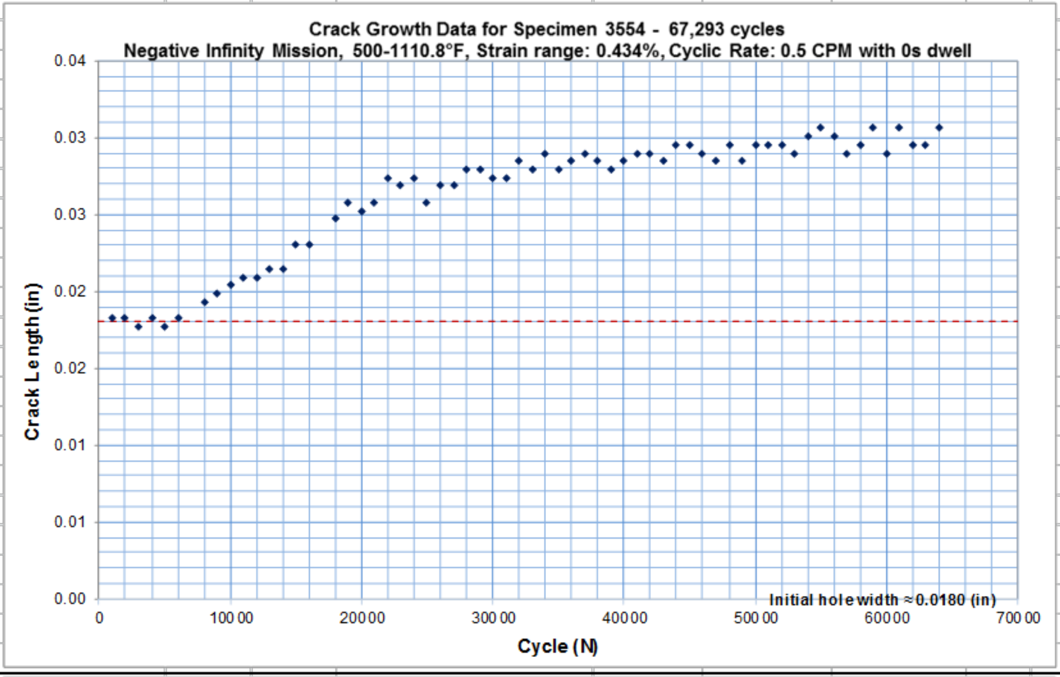
Test Type:	Infinity Mission	**Specimen ID:**	3554	**Material:**	INCO718
Outside diameter:	0.5505 in	**Specimen type:**	MT-41--notched	**Environment:**	Air
Inside diameter:	0.4484 in	**Strain range:**	0.434%	**Tmax:**	1111 °F
Area:	0.0801 in²	**Max strain:**	+0%	**Tmin:**	500 °F
		Min strain:	-0.434%	**Dwell:**	0s
				Cyclic Rate:	0.5 CPM
N_f:	67293 cycles	**Reference:**	MMD 20155234		
	Specimen discontinued at P&W's request				Page 5 of 6

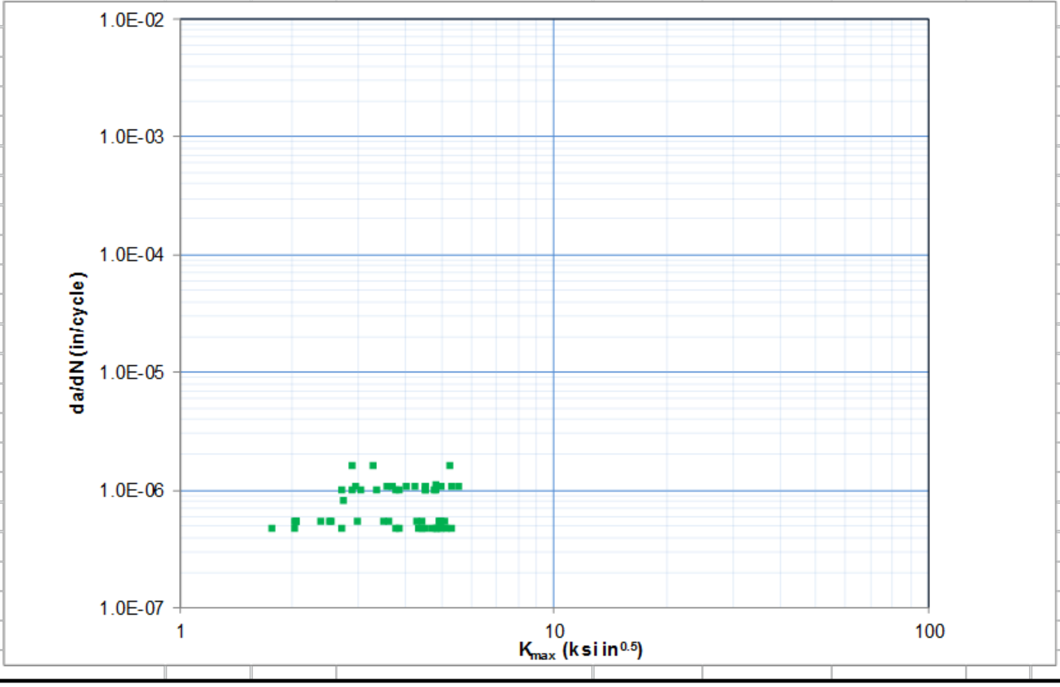


Cycle 67293



Description:							
Test Type:	Infinity Mission		**Specimen ID:**	3554	**Material:**	INCO718	
Outside diameter:	0.5505	in	**Specimen type:**	MT-41--notched	**Environment:**	Air	
Inside diameter:	0.4484	in	**Strain range:**	0.434%	**Tmax:**	1111	°F
Area:	0.0801	in²	**Max strain:**	+0%	**Tmin:**	500	°F
			Min strain:	-0.434%	**Dwell:**	0s	
					Cyclic Rate:	0.5	CPM
N_f:	67293	cycles	**Reference:**	MMD 20155234			
	Specimen discontinued at P&W's request					Page 6 of 6	





APPENDIX G—GROWTH RATE TESTS OF LARGE AND SMALL FATIGUE CRACKS IN DP 718

G.1 BACKGROUND AND CURRENT OBJECTIVE

Previous fatigue crack growth (FCG) testing conducted within the Probabilistic Design for Rotor Integrity (PDRI) program on surface crack tension [SC(T)] specimens of delta-processed superalloy Inconel 718 at 600°F showed no apparent difference in behavior under constant loading conditions ($R = S_{\min}/S_{\max} = 0.05$) between "small" cracks growing from 0.003″ and 0.004″ long notches produced by bombarding the specimen surface with a focused ion beam (FIB) and much larger cracks growing from conventional electro-discharge machined (EDM) notches of approximately 0.020″ surface length. Though the studied crack dimensions were ostensibly outside the so-called microstructurally small-crack regime (average grain size of the material was only approximately 0.0003″) (see figure G-1), a complete absence of any observed physically small crack effect was somewhat unexpected and, therefore, warranted further investigation.

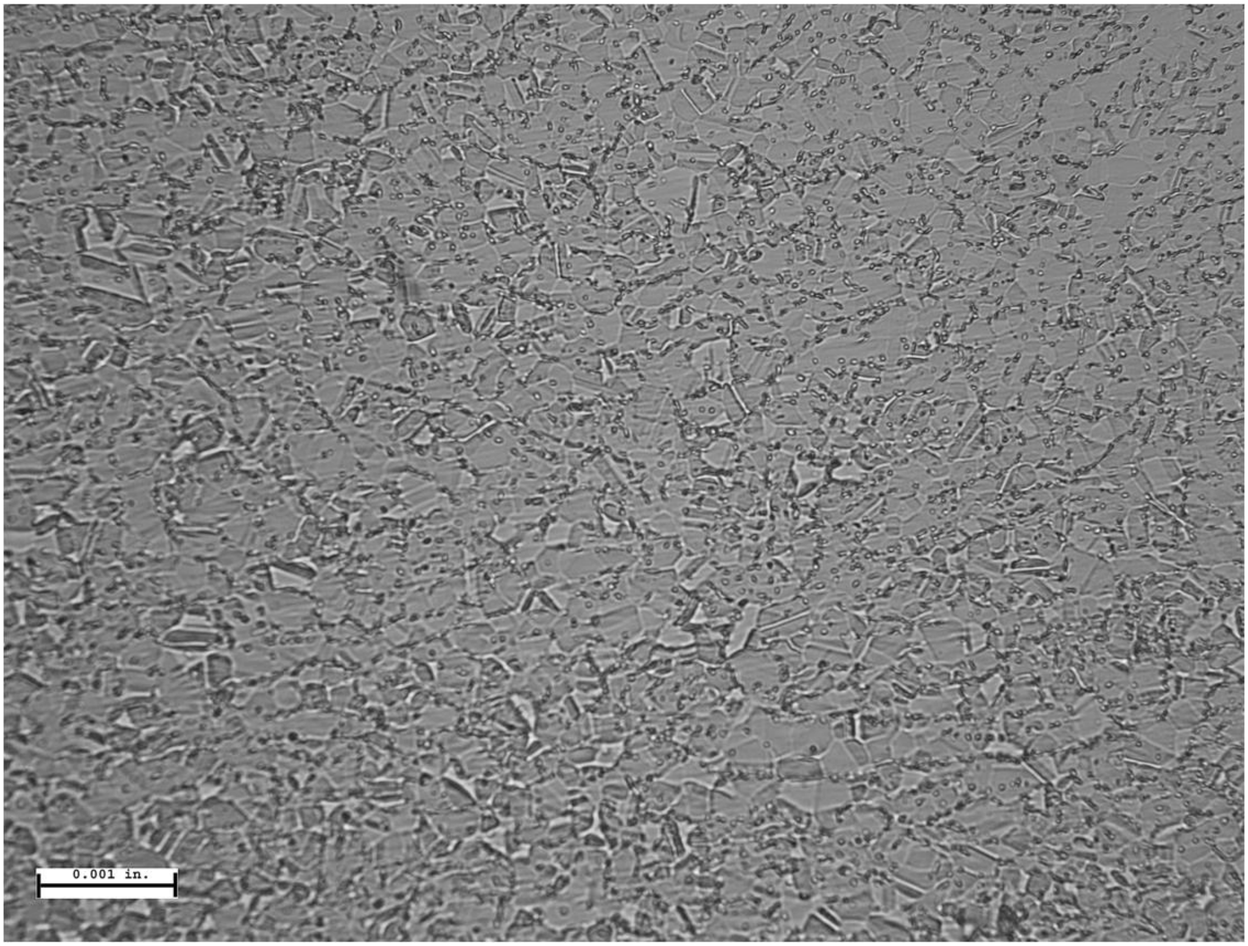


Figure G-1. Microstructure of DP 718

In the current study, additional large-crack and small-crack tests were performed on the same DP718 material at the same testing conditions to compare the resulting growth rates. Because the “large” surface cracks in the previous study were created by unconventional compression pre-cracking methodology that could conceivably have influenced their subsequent behavior, the new tests included conventional (ASTM E647) load-shedding methods along with compression pre-cracking methods, both using the widely used compact tension [C(T)] specimen geometry. The “small” cracks in the current study were initiated at smaller FIB notches with an optimized (semi-elliptical rather than square) shape.

G.2 MATERIAL

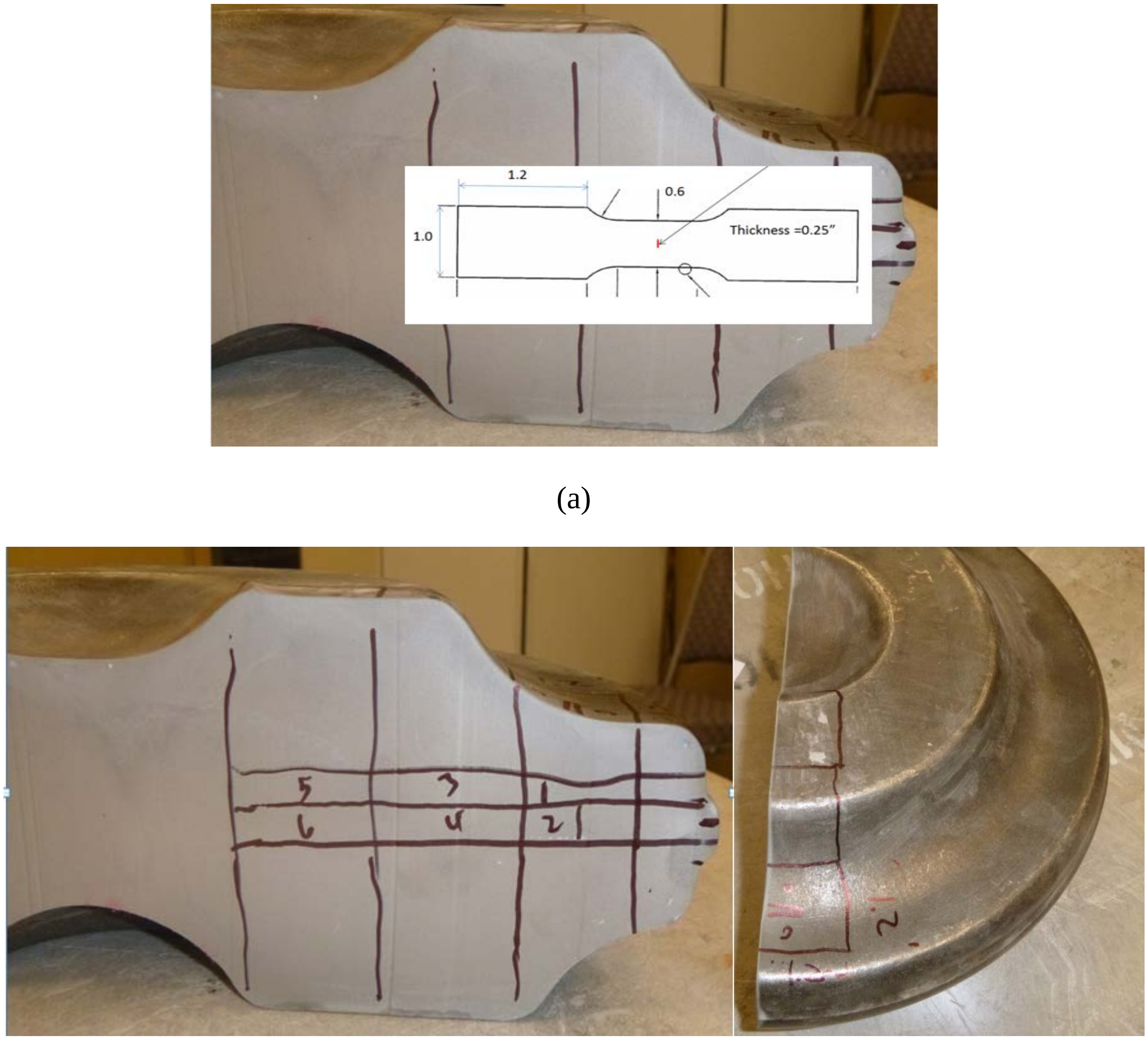
Delta-processed Inconel 718 alloy (DP718, ASTM grain size 10-12, YS = 169.0 ksi, UTS = 205.0 ksi) was used to machine SC(T) and C(T) specimens for this study.

The DP718 billet was made from triple melt ingot through vacuum-induction melting followed by electroslag remelting, and finished by vacuum arc remelting. A DP718 forging derives its strength benefits from a consistently fine microstructure, which is a result of a delta-processing treatment of the raw material when it is being manufactured into billet. The processing occurs during initial ingot breakdown when a special delta phase precipitation heat treatment is added to the normal billet manufacture. This heat treatment essentially consists of a long soak for 24 hours at approximately 1650°F. The delta phase consists of Ni_3Nb precipitates, which act to inhibit grain growth from any potential re-crystallization operations during final billet working. After the delta-phase heat treatment, final billet working is performed at temperatures less than the delta-solvus to ensure fine microstructure. Forging temperatures are generally well below typical delta-solvus temperatures of 1850°F–1875°F, which also ensure that the fine microstructure is maintained.

After forging, the final heat treatment of DP718 consists of a solution treatment below delta solvus followed by a two-step aging process. The solution temperature is typically conducted near the (delta) δ-solvus to avoid grain growth. As δ-solvus is in the range of 1850°F–1875°F, the solution temperatures often range from 1800°F–1850°F to maintain a fine microstructure while achieving the best δ precipitation possible. The first and second age cycles result in precipitates that are the primary strengthening phases, γ’’, with some contribution from the γ’ phase. The resulting microstructure is shown in figure G-1.

G.3 SPECIMEN PREPARATION

All specimens were machined out of the same forging that was used in the previous test program. The SC(T) (see figure G-2(a)) and C(T) (see figure G-2(b) and figure G-2(c)) specimens were excised from the central location of the forging and, therefore, the gauge sections of both had similar microstructure. The crack orientation and propagation direction were also similar.



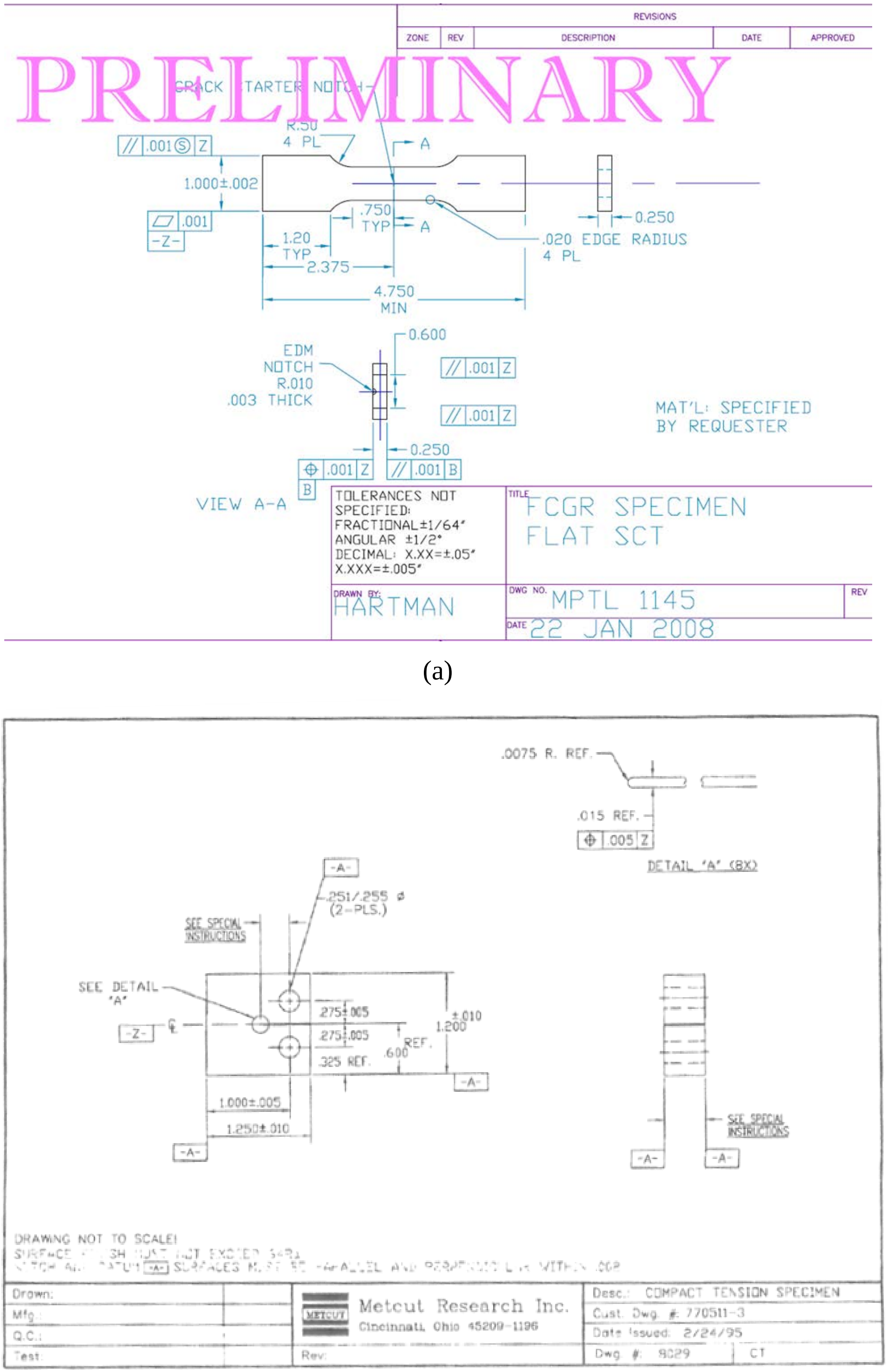
(a)

(b)

(c)

Figure G-2. Specimen location and orientation in the parent forging:
(a) SC(T) specimens, and (b, c) C(T) specimens

Specimen drawings are shown in figure G-3.



(a)

(b)

Figure G-3. Specimen drawings (a) SC(T) specimen and (b) C(T) specimen

In the previous program, the near-surface RS remaining after low-stress grind machining of the SC(T) specimens by Mar-Test were deemed unacceptably high for the purpose of the study. Therefore, specimens were longitudinally polished and then electro-polished by Lambda Technologies to remove at least 0.0005″ and to reduce the surface RS to acceptable level. The latter step complicated the logistics and added considerable time to the specimen machining/preparation stage. In the current program, a multistep polishing procedure, specifically developed at Honeywell, was applied instead with the same result. In addition, the material along the anticipated crack trajectory on the specimen faces was mechanically polished at Honeywell to facilitate optical observation and measurement.

In the previous test program, the FIB notches had a rectangular shape (see figure G-4). Such a shape is easy to create and was used in several earlier studies in the literature that used FIB technology. However, this also creates a crack driving force for early crack growth substantially different from that calculated for a semi-circular defect of the same surface length (see figure G-5). Therefore, in the current study, a conscious effort was made to optimize the shape of the FIB notch (see figure G-6). The instantaneous crack front location was made apparent by periodically heat-tinting the specimen at several stopping points during the test.

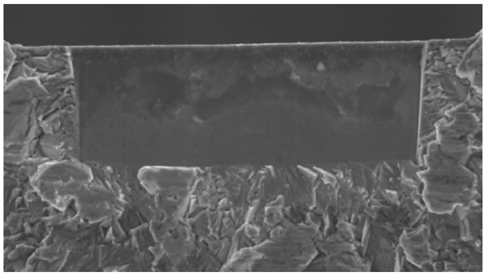


Figure G-4. Rectangular FIB notch shape

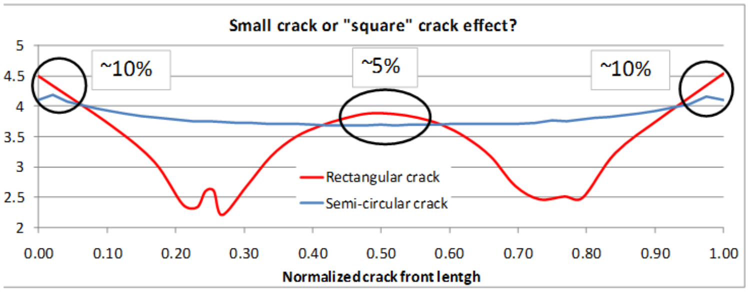


Figure G-5. Stress intensity factors for rectangular and semi-circular cracks

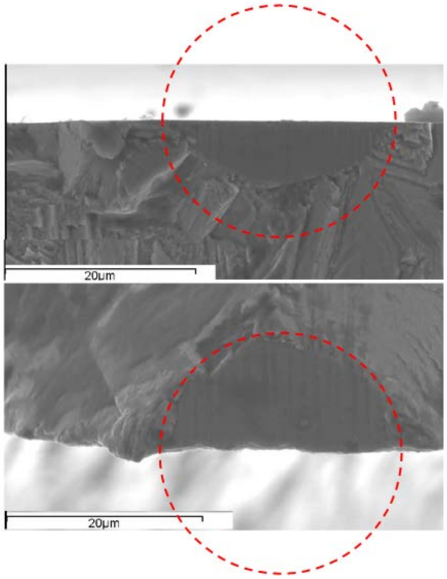


Figure G-6. Optimized FIB notch shape

Figure G-7 shows the location of FIB notches in the SC(T) specimens used in the previous investigation. Such a pattern has two shortcomings that became evident after initial tests: 1) constant temperature over a substantial length of the gauge section of the specimen, and 2) considerable and non-trivial efforts to open non-dominant cracks for post-mortem analysis. In the current study, a different pattern (see figure G-8) was used. In addition to alleviating the above drawbacks, this pattern makes it possible to detect specimen misalignment in the test frame by comparing the behavior of cracks located on the front and rear face of the specimen. Note that the notches and the ensuing small cracks were so small that no interaction occurred between adjacent cracks in the regime of data collected.

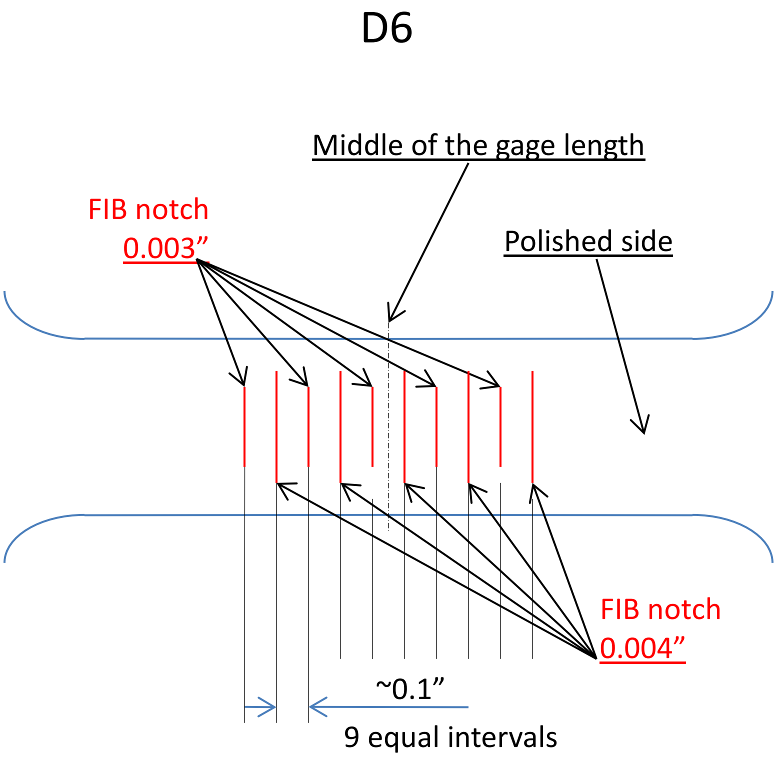
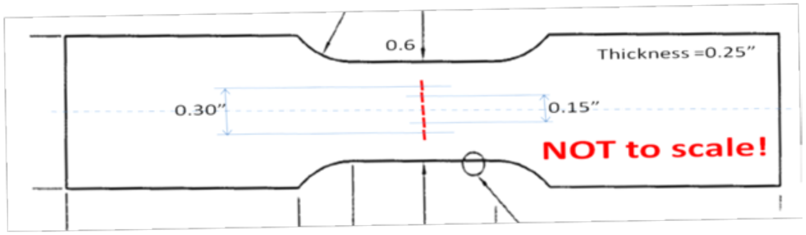


Figure G-7. Location of FIB notches in SC(T) specimens used in the previous program



Note: Five more notches (not shown) were located on the back face of the specimen.

Figure G-8. Location of FIB notches in SC(T) specimens used in the current program

In the case of the C(T) specimens, Metcut had previous experience with cracks originating from the pinholes during compressive pre-cracking (see figure G-9). To eliminate this unwanted phenomenon, several C(T) specimens were pre-cracked in compression between two parallel plates prior to machining the pinholes.

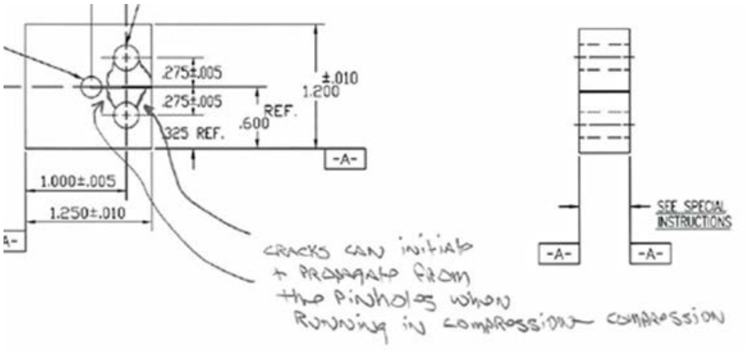


Figure G-9. Unwanted cracks in C(T) specimen subjected to compressive pre-cracking

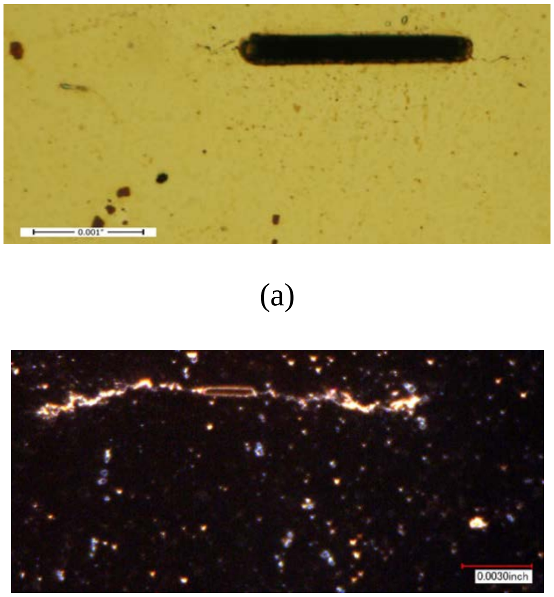
The width of the starter notch was increased to 0.05″ compared to the drawing shown in figure G-3(b). The pinhole surface was polished to prevent cracking during the tension test.

Prior to testing, all C(T) specimens were inspected by Metcut's Quality Assurance Department to ensure that they met the requirements of the specimen drawing and ASTM E-647-11. The material along the anticipated crack trajectory on a specimen middle section was polished by using various grades of silicon carbide abrasive paper and polishing cream to enhance the visibility for crack-length measurements. The specimens were then washed with cotton towels and acetone, followed by ultrasonic cleaning in a detergent bath for approximately 20 minutes.

G.4 CRACK-SIZE MEASUREMENT

The SC(T) specimens were periodically removed from the test frame and subjected to direct measurements/digital photography on a stationary optical microscope (see figure G-10(a)) or SEM (see figure G-10(b)) as the most accurate and reliable method.

The crack length was measured in the C(T) specimens with the direct current electric potential drop (DCEPD) technique as described in ASTM E647-11, sections A1.5.3 and A6. The current and electrical potential drop lead wires were attached to the specimens (see figure G-11). The EPD wires were Alumel and were 0.015″ in diameter. The current wires were also Alumel, approximately 0.030″ in diameter. All four of the wires were kept as short and as straight as possible. The method used in this study uses Johnson's Equation to relate voltage measurements to crack length and the secant method of data reduction as described in ASTM E647-11, section X1.1. The instantaneous crack front location was made apparent by periodically heat tinting the specimen at various stopping points during the test. As the tested material was relatively resistive, a reference voltage source was not used during testing because the required sensitivity and resolution for detecting crack change was available without it. The system used incorporates the practice of switching the polarity of the current from positive to negative during potential measurements to eliminate the possibility of thermoelectric effects. A constant current of 3 amps was applied to a specimen. With this amount of current, the system was able to detect changes in crack length with a resolution of 0.0003″.



(a)

(b)

Figure G-10. Surface crack-size measurements in SC(T) specimens

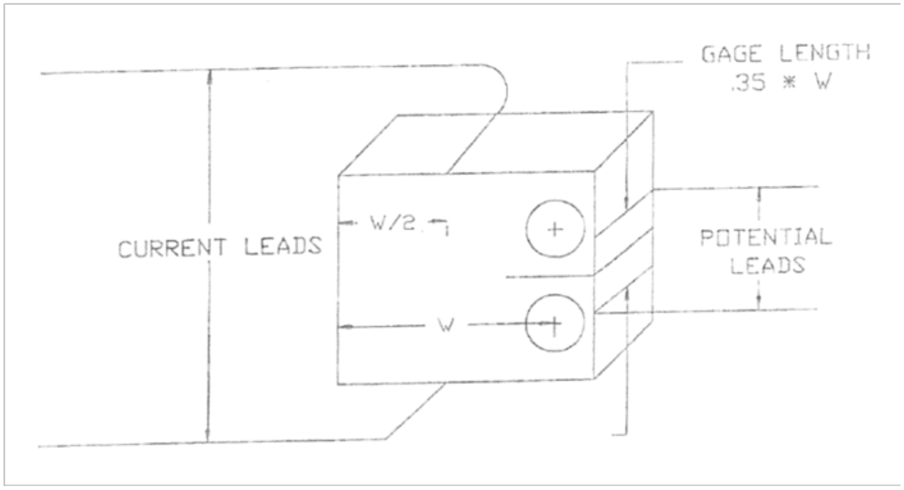


Figure G-11. Specimen instrumentation for DCEPD crack-size measurement

G.5 COMPRESSION PRE-CRACKING

Prior to testing, four C(T) specimens were subjected to pre-cracking in compression. Some basic information about the process is summarized in table G-1. The dimension B in the table is the specimen width, and L represents the length from the tip of the notch to the end of the specimen. Cycling was stopped at convenient intervals to measure crack growth optically on the front and rear surfaces of the specimen.

Table G-1. Summary of compression pre-cracking

Specimen	B (in)	L (in)	Notch Radius	Area (in^2)	Min Stress (Ksi)	Max Stress (ksi)	Min Load (lbs)	Max Load (lbs)	a (front)	a (rear)
B-1	0.24	0.797	0.0077	0.19128	-100	-5	-19128	-956.4	0.04992	0.04914
B-2	0.24	0.7965	0.0074	0.19116	-70	-3.5	-13381.2	-669.06	0.02418	0.01833
C-1	0.239	0.7968	0.0078	0.190435	-80	-4	-	-761.7408	0.02925	0.02847
C-2	0.238	0.797	0.008	0.189686	-60	-3	-11381.16	-569.058	0.02379	0.02418

G.5.1 SPECIMEN B-1

The starting minimum stress for Specimen B-1 was -100 ksi, *R* = 20, with a starting frequency of 1 Hz. The following measurements were recorded:

- 58,670 cycles: Front = 0.04368″, Rear = 0.03822″
- 86,905 cycles: Front = 0.04368″, Rear = 0.04290″

Frequency increased to 2Hz.

- 141,337 cycles: Front = 0.04680″, Rear = 0.04680″
- 289,929 cycles: Front = 0.04992″, Rear = 0.04914″

The pre-cracking process was terminated after the crack growth rate decreased to less than 1 × 10-9 in/cycle.

The final compression pre-crack in Specimen B-1 is shown in figure G-12.

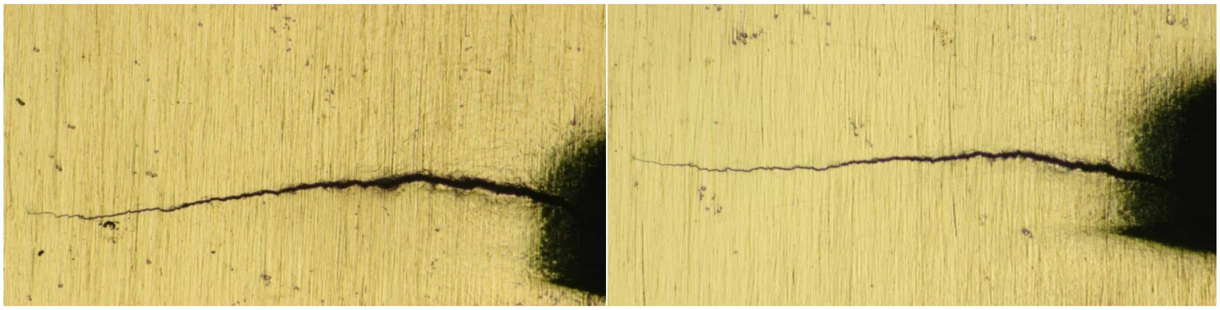


Figure G-12. Compression pre-crack in B-1

G.5.2 SPECIMEN C-1

The initial minimum stress was -90 ksi, R = 20, with a frequency of 6.19 Hz. The following measurements were recorded:

- 127,262 cycles: Front = 0.02808″, Rear = 0.02612″

Frequency increased to 7.43 Hz (attempted further increase caused vibrations).

- 1,000,000 cycles: Front = 0.02925″, Rear = 0.02847″
- 1,200,002 cycles: Front = 0.02925″, Rear = 0.02847″

The final compression pre-crack in Specimen C-1 is shown in figure G-13.

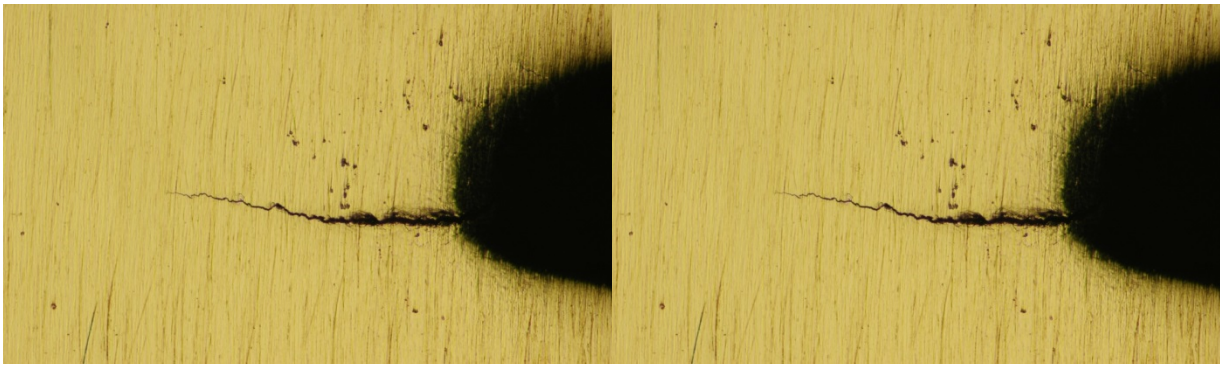


Figure G-13. Compression pre-crack in C-1

G.5.3 SPECIMEN B-2

The initial minimum stress was -70 ksi, R = 20, with a frequency of 8.0 Hz. The following measurements were recorded:

- 183,239 cycles: Front = 0.02418″, Rear = 0.01716″
- 643,051 cycles: Front = 0.02418″, Rear = 0.01833″
- 800,002 cycles: Front = 0.02418″, Rear = 0.01833″

The final compression pre-crack is shown in figure G-14.

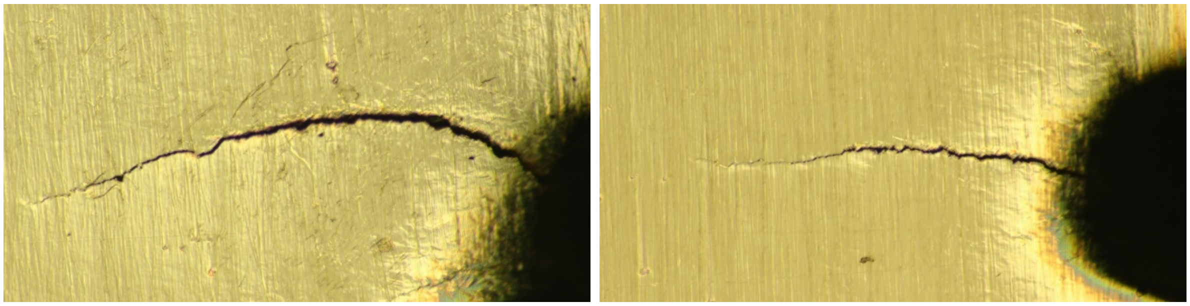


Figure G-14. Compression pre-crack in B-2

G.5.4 SPECIMEN C-2

The starting minimum stress was -60 ksi, $R = 20$, with a frequency of 8.0 Hz.

- 474,280 cycles: Front = 0.02184″, Rear = 0.01755″
- 528,045 cycles: Manual shutdown because of planned weekend power outage
- 628,047 cycles: Front = 0.02379″, Rear = 0.02223″
- 778,047 cycles: Front = 0.02379″, Rear = 0.02418″
- 928,047 cycles: Front = 0.02379″, Rear = 0.02418″

The final compression pre-crack is shown in figure G-15.

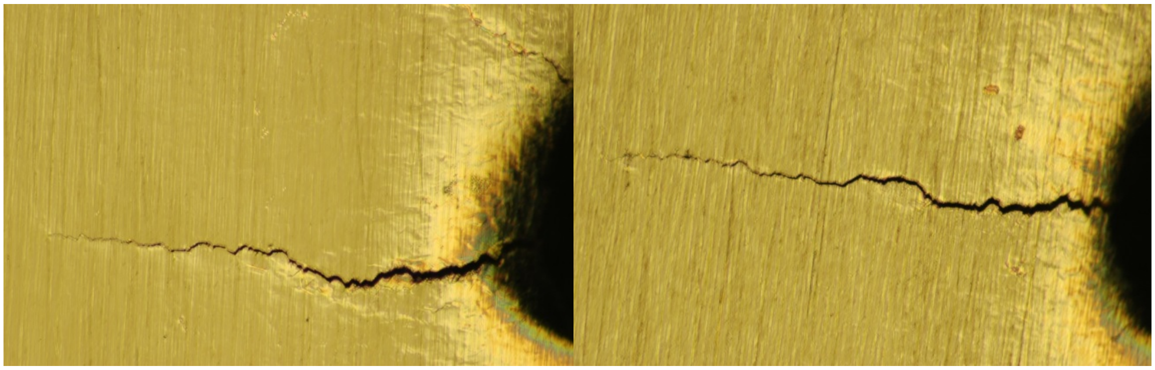


Figure G-15. Compression pre-crack in C-2

The remaining C(T) specimens were pre-cracked in tension as per ASTM E-647-11.

Compression pre-cracking was not used for the SC(T) specimens with FIB notches. They were all pre-cracked in tension.

G.6 TESTING AND RESULTS

All specimens were tested in laboratory air at 600°F using servo-hydraulic MTS frames with FTA software. A triangular wave shape of constant load amplitude was applied with a frequency of 1 Hz. This test temperature was high enough to emulate realistic rotor service conditions but low enough (and the test frequency fast enough) to avoid time-dependent effects.

G.7 C(T) SPECIMENS

The purpose of testing C(T) specimens was to provide a common baseline for “small” crack test results and to compare the results of the ASTM-recommended load-shedding procedure with the compression pre-cracking method. Two different test laboratories were used to make the baseline more general. A summary of the C(T) specimens is provided in table G-2.

Table G-2. Summary of C(T) Specimens

	Spec ID	Pre-crack type	Pre-crack vendor	Test type	Test R	Test vendor	Note
1	A-1	Tens	Metcut	ASTM Load shed	0.05	Metcut	Successful test
2	A-2	Tens	HON	ASTM Load shed	0.05	HON	Successful test
3	B-1	Compr	HON	Const load	0.05	Metcut	Unsufficient pre-crack, invalid threshold
4	C-2	Compr	HON	Const load	0.05	HON	Successful test
5	B-2	Compr	HON	Spare, not tested			
6	C-1	Compr	HON	Spare, not tested			

Figure G-16 compares the ASTM load shed results from two different laboratories. A reasonable agreement between two data sets is observed with an apparent FCG threshold of approximately 5.0 ksi√in.

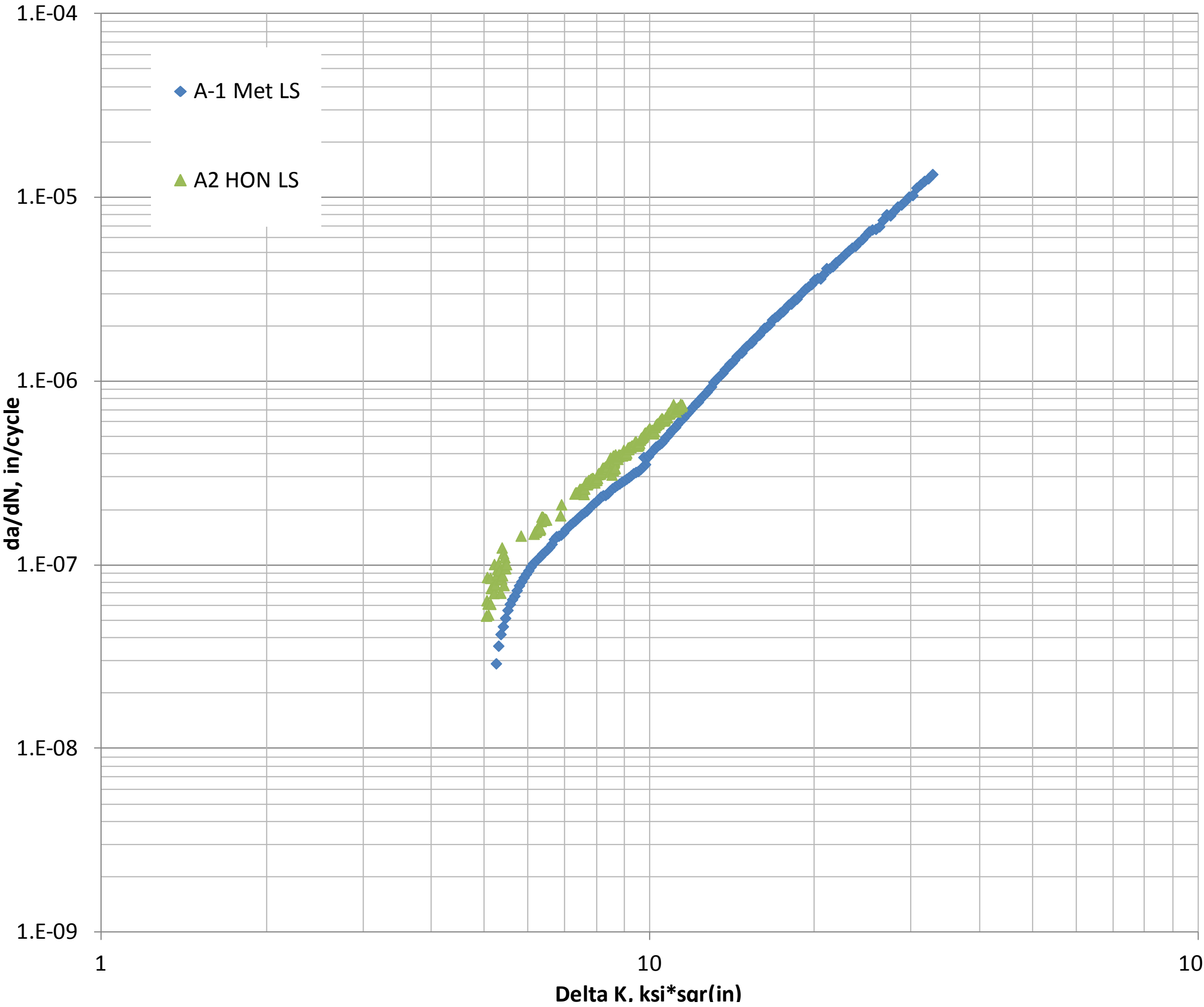
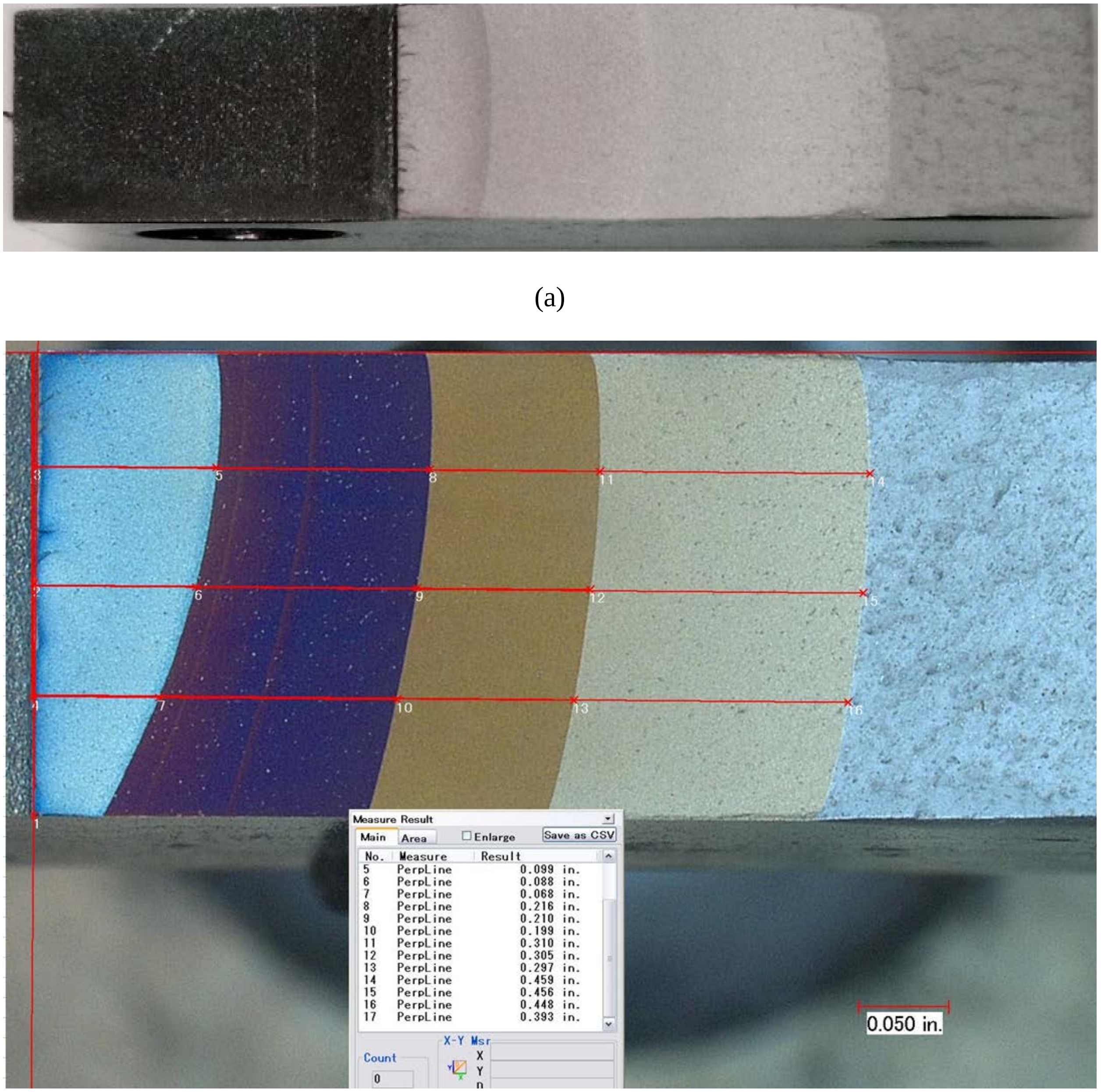


Figure G-16. ASTM load shed results produced by Metcut (MET) and Honeywell (HON)

Consistently higher FCG rates recorded in the Honeywell test may be attributable to some noticeable differences in the crack front curvature (see figure G-17), a different number of intermittent heat-tints used for the post-test analysis, or simply to experimental scatter.


(a)

(b)

Measure Result

No.	Measure	Result
5	PerpLine	0.099 in.
6	PerpLine	0.088 in.
7	PerpLine	0.068 in.
8	PerpLine	0.216 in.
9	PerpLine	0.210 in.
10	PerpLine	0.199 in.
11	PerpLine	0.310 in.
12	PerpLine	0.305 in.
13	PerpLine	0.297 in.
14	PerpLine	0.459 in.
15	PerpLine	0.456 in.
16	PerpLine	0.448 in.
17	PerpLine	0.393 in.

Count 0

Figure G-17. Fracture surface appearance: (a) specimen A1 tested by Metcut and (b) specimen A2 tested by Honeywell

Figure G-18 compares the results from both laboratories after compression pre-cracking. In the case of the Honeywell test, an apparent FCG threshold was also detected at approximately 5.0 ksi√in.

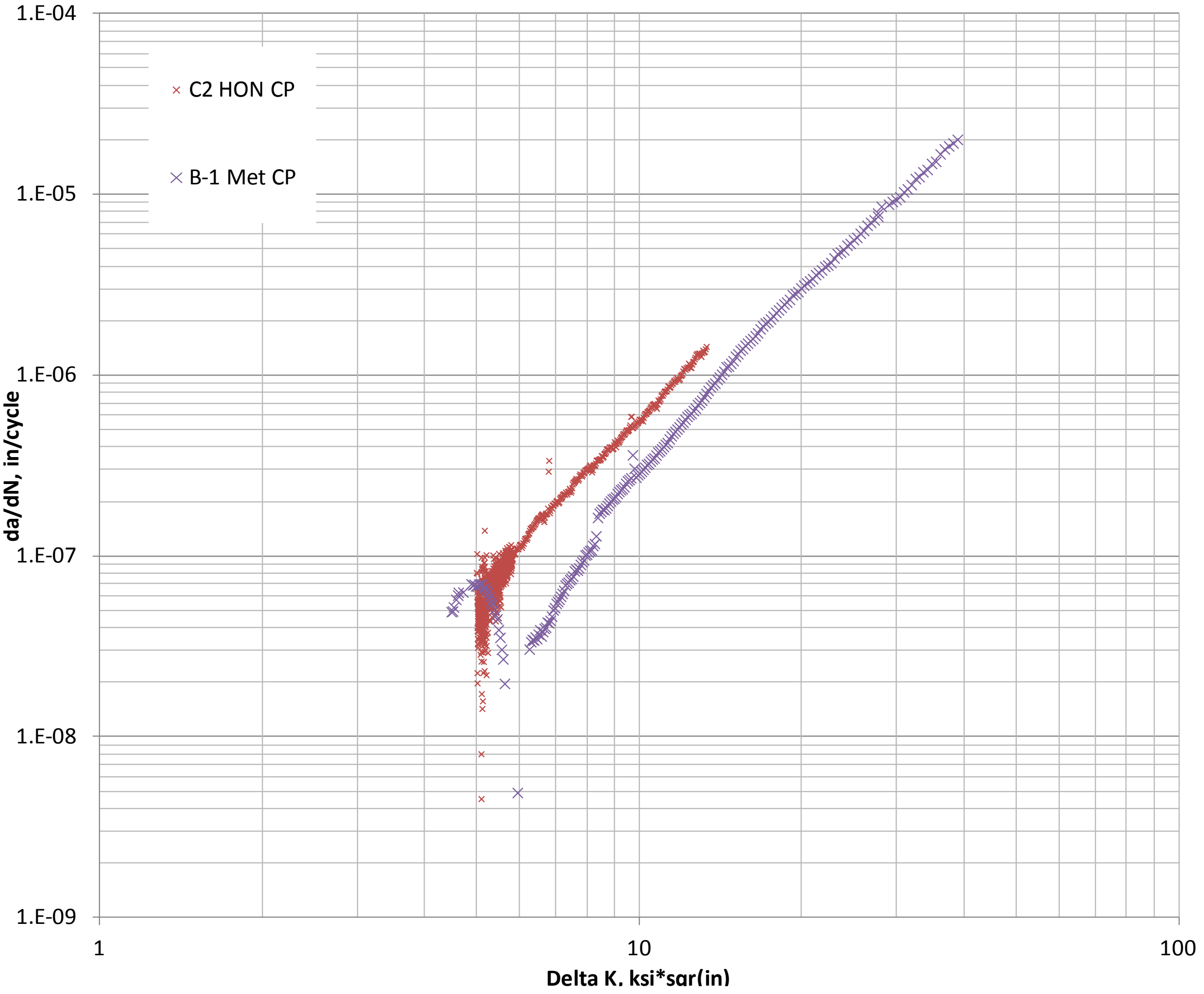
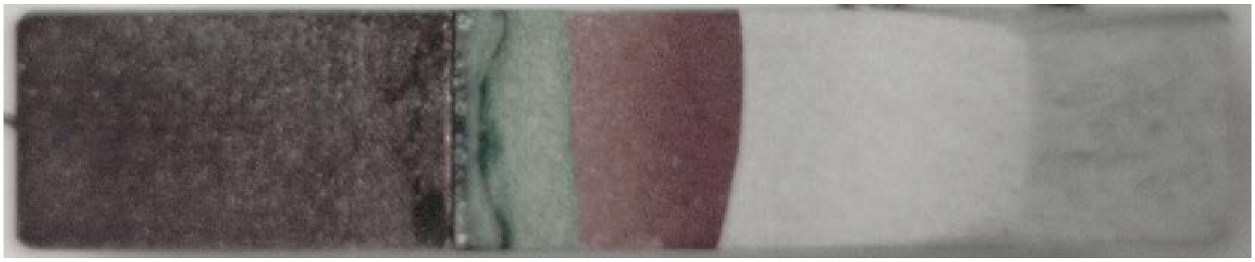


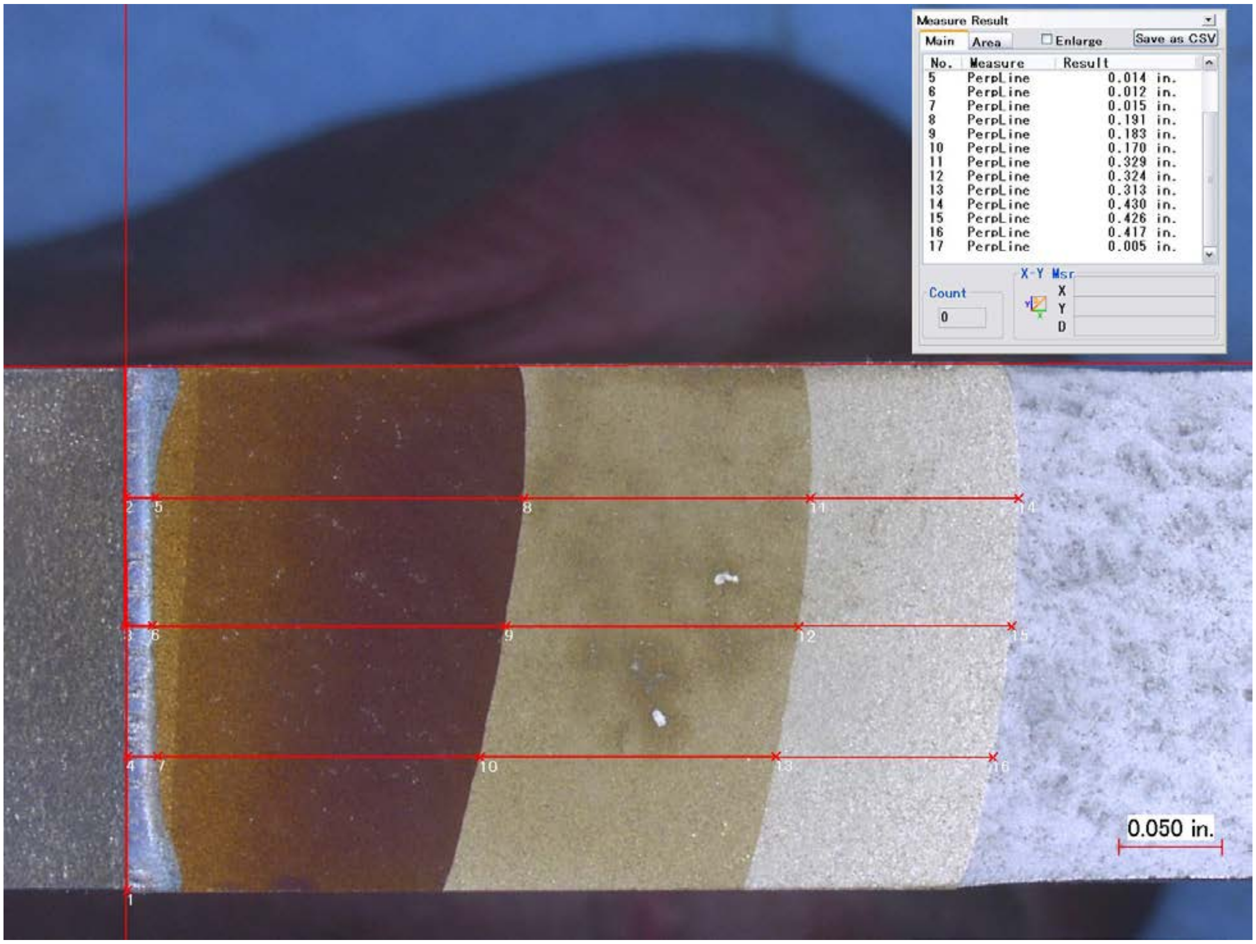
Figure G-18. FCG results produced by Metcut (MET) and Honeywell (HON) after compression pre-cracking

The behavior of specimen B1, tested by Metcut, was quite different and warranted closer consideration. This was the first C(T) specimen subjected to compression pre-cracking in this study. A loading frequency of 1 Hz was used, and the pre-cracking was terminated after only 289,929 cycles when the crack growth rate decreased to less than 1 × 10-9 in/cycle. This was not enough for the crack to grow out of the compressive plastic zone associated with the notch. Because of the premature termination of the pre-cracking process, some tensile RS was left ahead of the crack tip. This may explain a “stable” crack propagation observed in this specimen immediately after applying tensile loading with nominal ΔK values of approximately 4.5 ksi√in, lower than the FCG threshold established by the ASTM load shed results from both vendors and by the Honeywell results generated after compression pre-cracking. As the crack continued to grow, it was getting less and less “help” from this tensile RS, and therefore the FCG abnormally decreased under increasing ΔK values (see figure G-18).

The very slow crack propagation immediately following the compression pre-cracking resulted in noticeable fretting-corrosion of the fracture surface, particularly in the specimen mid-thickness, which would have influenced the subsequent FCG behavior due to oxide-induced crack closure (see figure G-19(a)).



(a)



No.	Measure	Result
5	PerpLine	0.014 in.
6	PerpLine	0.012 in.
7	PerpLine	0.015 in.
8	PerpLine	0.191 in.
9	PerpLine	0.183 in.
10	PerpLine	0.170 in.
11	PerpLine	0.329 in.
12	PerpLine	0.324 in.
13	PerpLine	0.313 in.
14	PerpLine	0.430 in.
15	PerpLine	0.426 in.
16	PerpLine	0.417 in.
17	PerpLine	0.005 in.

Count: 0

(b)

Figure G-19. Fracture surface appearance of: (a) specimen B1 tested by Metcut, and (b) specimen C2 tested by Honeywell

Figure G-20 compares all C(T) test results generated in this study. Based on the foregoing deliberations, specimen B1 should be excluded from further consideration, whereas the three remaining specimens show reasonably consistent behavior, with a FCG threshold at approximately 5.0 ksi√in. In particular, comparison of the two Honeywell tests shows no substantial difference between the ASTM load shed results and the results generated after compression pre-cracking.

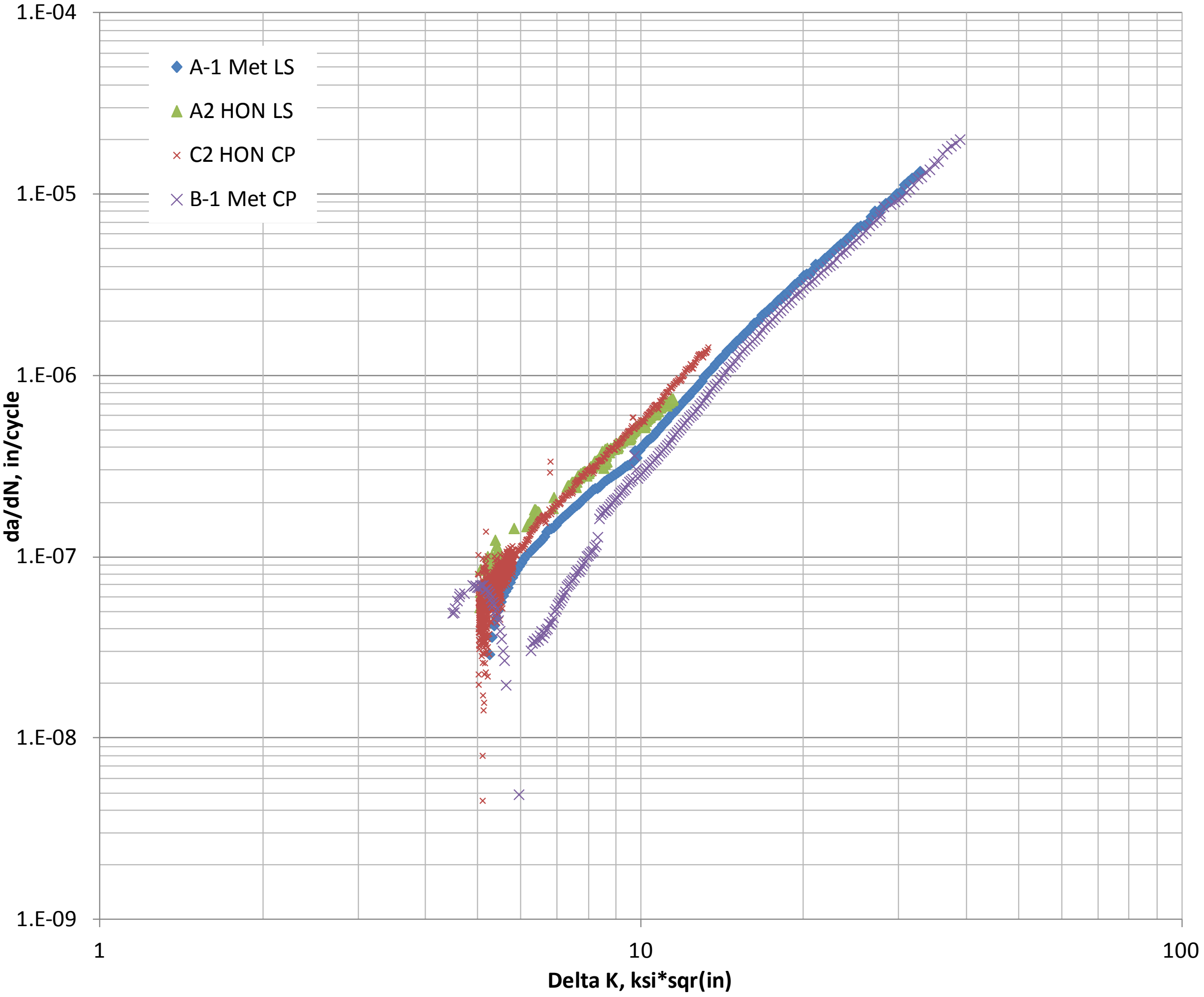


Figure G-20. FCG Results produced by Metcut (MET) and Honeywell (HON) after ASTM load shed (LS) and compression pre-cracking (CP)

Figure G-21 compares all valid results generated in this study with the C(T) geometry with the previous (i.e., PDRI) results for SC(T) specimens with “large” cracks initiated from 0.020″ long EDM notches. At higher ΔK values, the two geometries produced similar results, but the SC(T) specimens exhibited slightly slower growth rates at lower ΔK values.

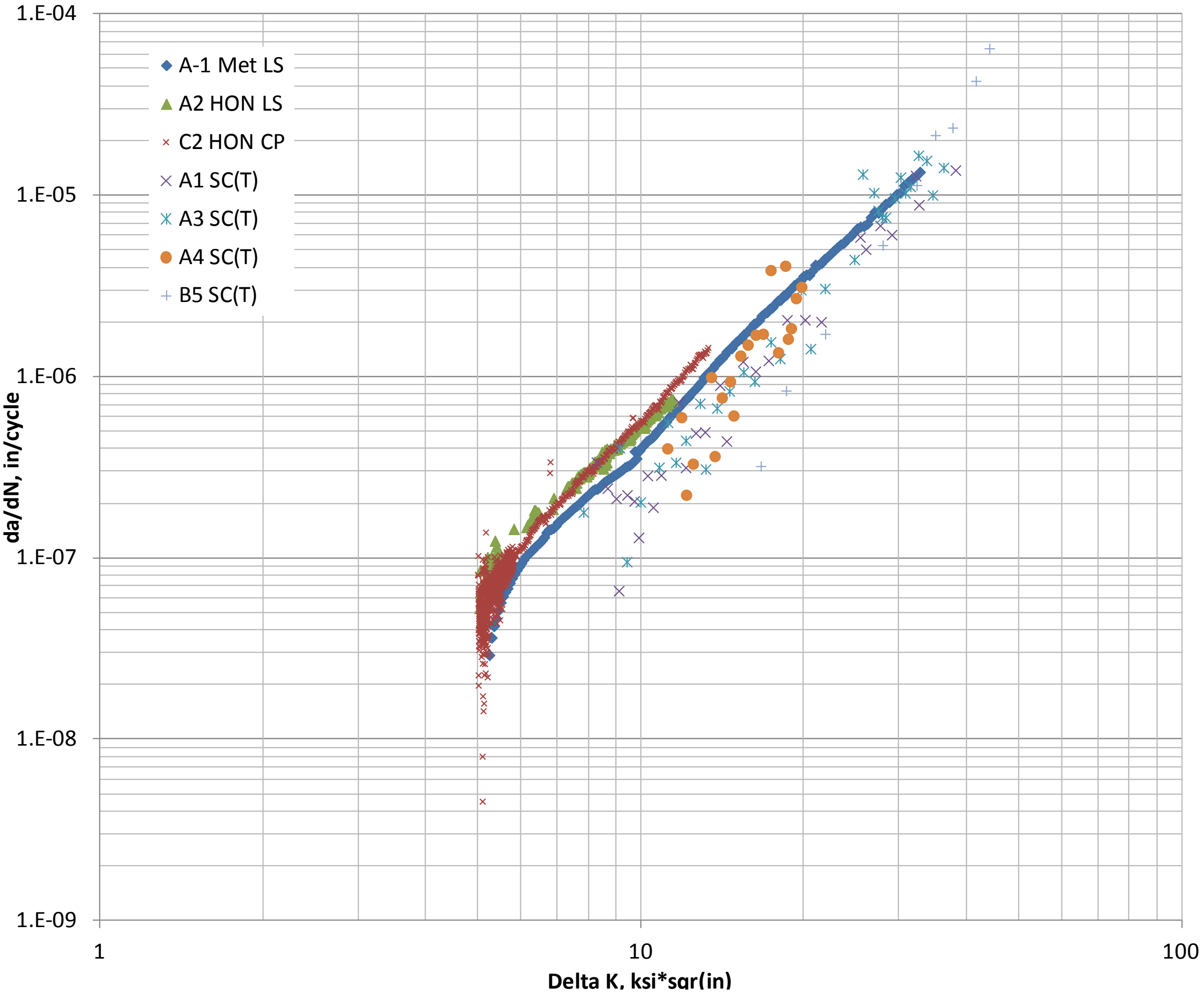


Figure G-21. FCG results for “large” cracks generated using C(T) and SC(T) specimen geometry

G.7.1 SC(T) SPECIMENS WITH FIB NOTCHES

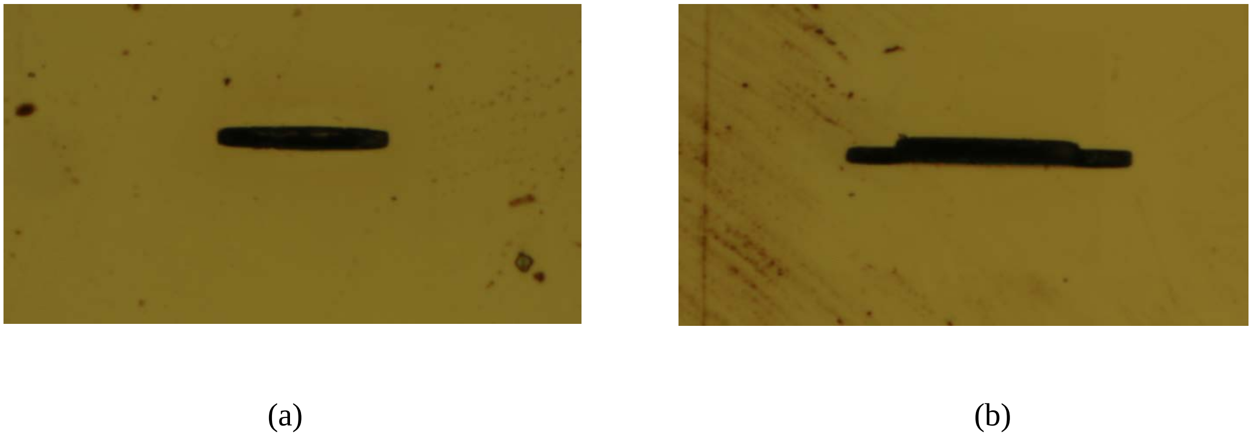
The purpose of testing SC(T) specimens with FIB notches was to evaluate “small” crack behavior. All such tests were conducted by Honeywell after conventional pre-cracking in tension. A summary of SC(T) specimens is provided in table G-3.

Table G-3. Summary of SC(T) specimens

	Spec ID	Notch surf length, in	# of notches	Test type	Test R	Note
1	F-2	0.001 (0.0016)	2	Const load Smax=120, 130, 140 ksi	0(-0.5)	Failed after 9,192 w/o cracking from notches
2	H-3	0.002	2	Const load Smax=130 ksi	0	Successful test
3	K-3	0.001	10	Const load Smax=130 ksi	0	Unwanted crack after 45,000 cycles–discontinued
4	I-3	0.002	10	Const load Smax=130 ksi	0	Successful test
5	J-3	0.002	10	Const load Smax=65, 75, 90, 100, 105 ksi	-1	Failed after 33,000 cycles from shoulder
6	K-2	0.003	10	Const load Smax=90 ksi	-1	Successful test

G.7.2 SPECIMEN F-2

The purpose of this test was to find out if a 0.001″ long and approximately 0.0005″ deep FIB notch will initiate a fatigue crack. Figure G-22 shows FIB notches in this specimen.



(a)

(b)

Figure G-22. FIB notches in specimen F-2

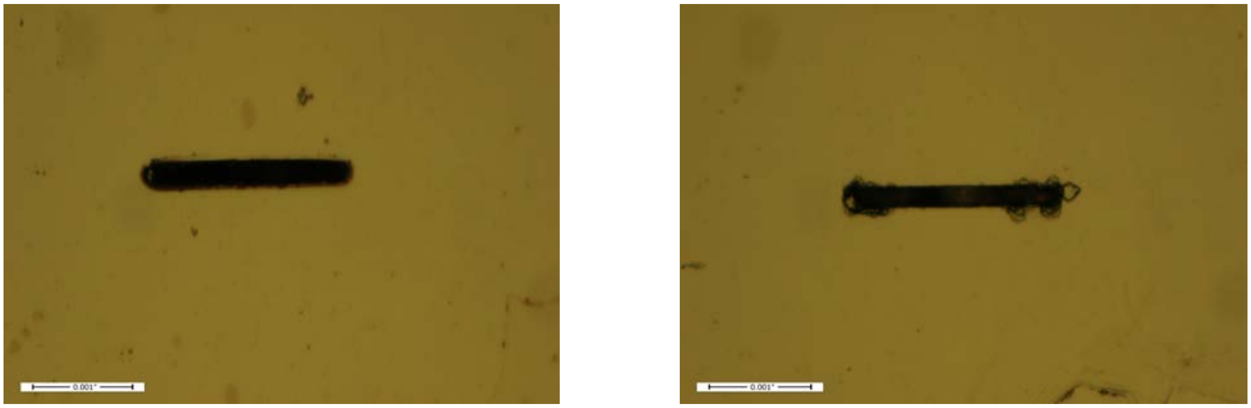
The first 15,000 cycles were tested with a maximum stress of 120 ksi (*R*= 0.00). The specimen was removed from the machine every 5000 cycles to determine crack initiation from the FIB notches. No cracking was noticed at the notches. The maximum stress was then increased to 130 ksi (*R* = 0.00). After 21,932 cycles, the specimen slipped in the grips. The grip pressure and the amount of the specimen being gripped increased. No cracking was noticed at the notches after 15,000 cycles (checking every 5,000 cycles). One of the notches was enlarged to 0.0016″ surface length. The maximum stress was then increased to 140 ksi (*R* = 0.00). The specimen slipped after 30,682 cycles; 31,536 cycles; and 34,053 cycles. The grips were cleaned, and the specimen was reinserted with fresh gripping compound. No cracking was noticed at the notches after 15,000 cycles (checking every 5,000 cycles). The maximum stress was held at 140 ksi, but the *R* ratio was decreased to *R* = -0.5. The specimen failed in the upper radius after 9192 cycles. A total of 54,192 cycles were run on specimen F2.

This test showed that the FIB notches as small as 0.0010″–0.0016″ were inefficient in initiating fatigue cracks in the given geometry without approaching net-section yielding (YS = 158.0 ksi at 600ºF).

G.7.3 SPECIMEN H-3

The next specimen (H-3) was prepared with two FIB notches of ~0.002″ surface length each (figure G-23) located on the opposite sides of the specimen.

The pre-cracking and testing of the SC(T) specimen H3 with two FIB notches of ~0.002″ surface length each located on the opposite sides of the specimen was conducted at *Smax* = 130 ksi and *R* = 0.00. No cracks at the FIB notches were visible after 5000 cycles. After 10,000 cycles, ~0.0003-0.0004″ cracks were present at both tips of one FIB notch and a ~0.00016″ crack was present at one tip of the second FIB notch. From that moment on the test was interrupted every 1000 cycles for optical crack measurements.



(a) (b)

Figure G-23. FIB notches in specimen H-2

The results, generated from this specimen containing two 0.002″-long FIB notches (symbols), were compared to the previous PDRI results for 0.003″ and 0.004″-long FIB notches (lines) in figure G-24. In this figure, the crack growth rates were calculated with simple secant methods (average growth rate between two consecutive measurements). No obvious differences in crack propagation behavior between the two sets of results can be seen. Because the previous testing conducted within PDRI program showed no apparent difference in behavior between cracks propagating from 0.003-0.004″-long FIB notches and much larger cracks, figure G-25 shows an absence of the small-crack effect.

It should be noted; however, that the data for specimen H-3 shown in figure G-24 were subjected to a simplified post-test analysis (i.e., the crack surface length, recorded during the test, was used to calculate *da/dN* and delta *K*, while the crack aspect ratio (*a/c*) was assumed to be constant and equal to 1.0). Alternatively, knowing the actual aspect ratio of the FIB notches (see figure G-25) and of the propagating cracks after two heat-tints (see figure G-26), it was possible to re-analyze the data using a better estimate of the crack aspect ratio (a regression fit of *a/c* versus *c* using the available measurements).

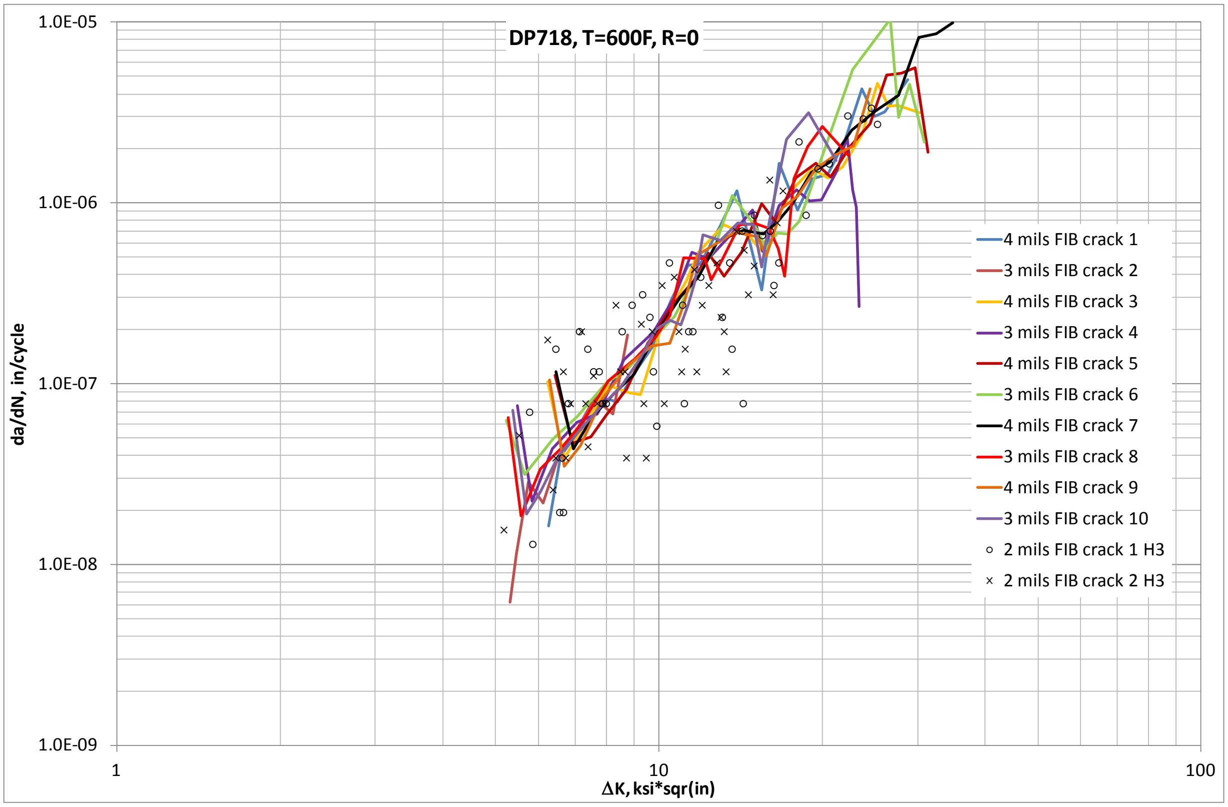


Figure G-24. FCG results for "small" cracks generated using SC(T) specimens with different FIB notches–simplified analysis for H-3

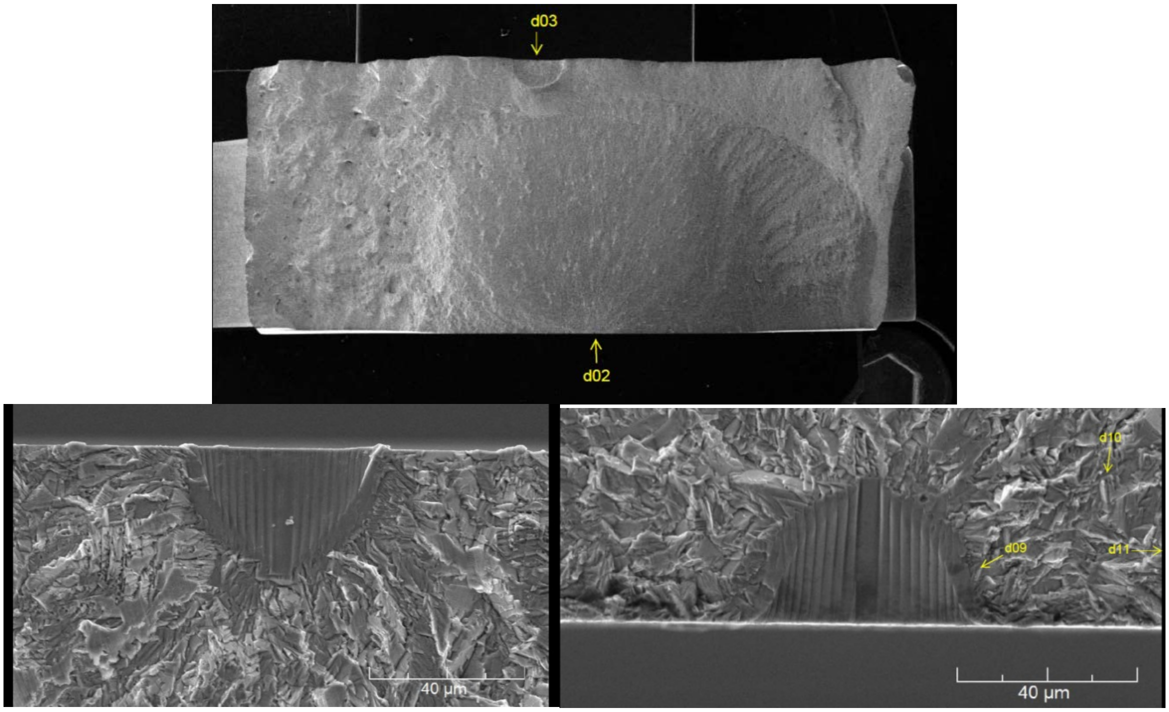


Figure G-25. FIB notches in H-3

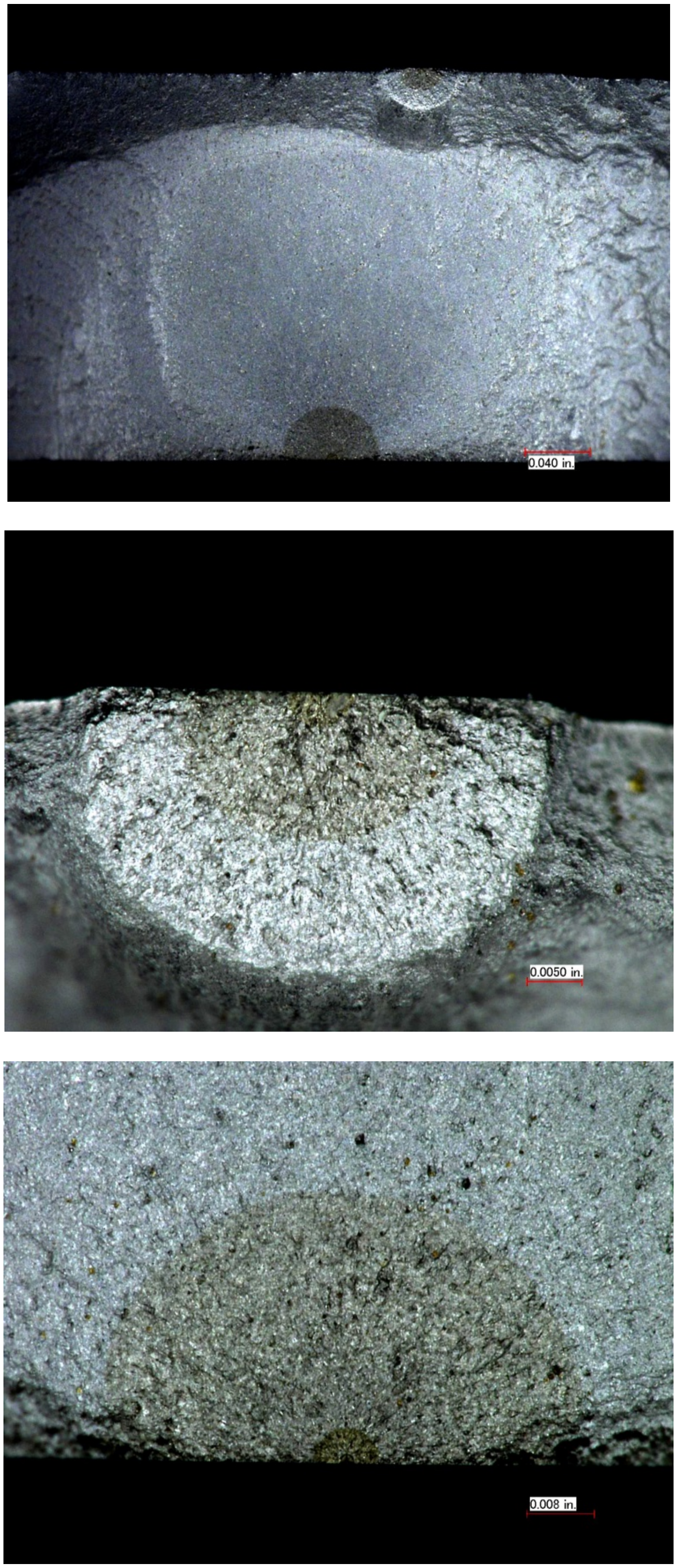


Figure G-26. Heat-tints in specimen H-3

The re-analyzed results for Specimen H-3 are shown in figure G-27. There are some shifts in the H-3 data, but the scatter makes it difficult to draw definitive conclusions. Remember that the earlier testing (colored lines) used an initially rectangular FIB notch. In addition, note that the first crack measurements were taken at 2c = 0.0035″ and 0.0032″.

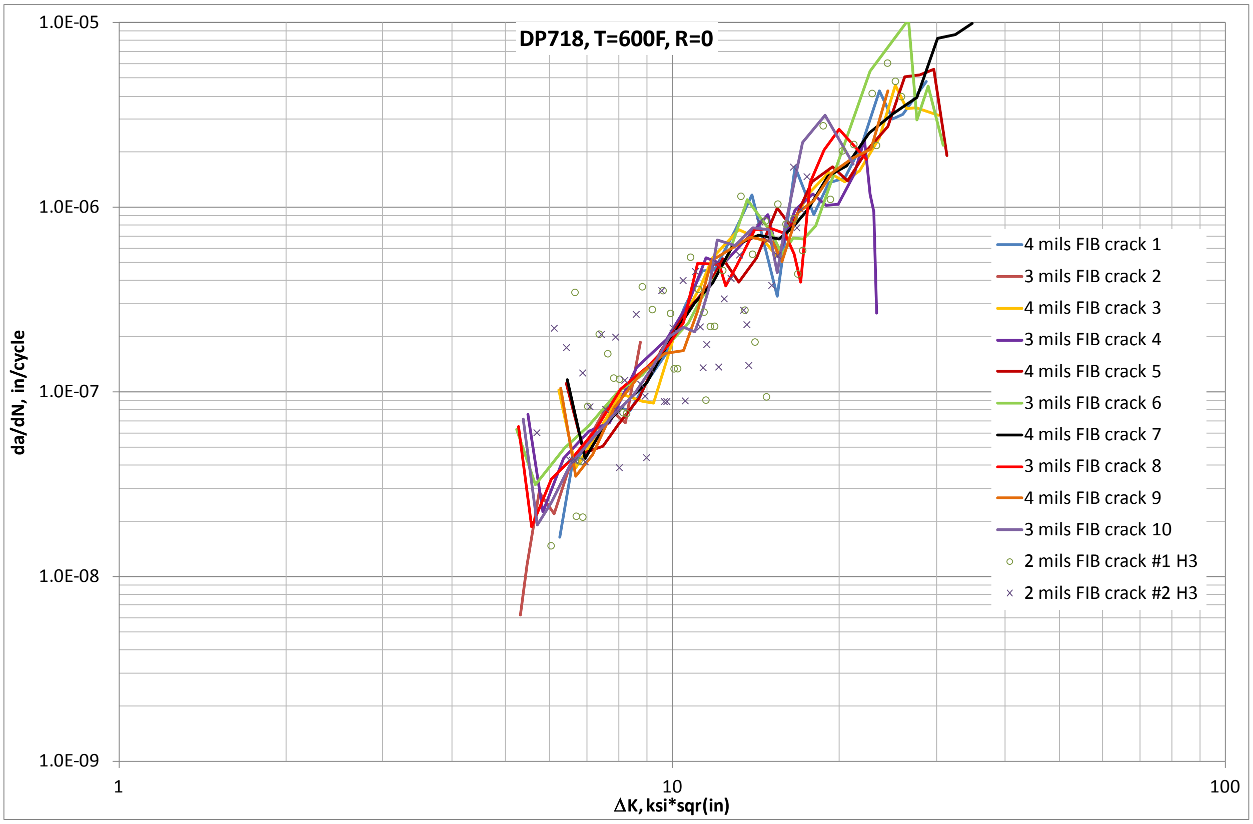


Figure G-27. FCG results for "small" cracks generated using SC(T) specimens with different FIB notches (H-3 re-analyzed for actual crack aspect ratio)

G.7.4 SPECIMEN K-3

By mistake, the FIB vendor prepared 10 notches of 0.001″ surface length in Specimen K3. The mistake was discovered as soon as the test started; however, because another specimen was not immediately available, it was decided to continue testing. Specimen K3 developed an unwanted crack in a shoulder after 45,000 cycles without any evidence of cracking at the notches. The test was discontinued. This was another confirmation that a FIB notch with 0.001″ surface length (0.0005″ depth) is not large enough to efficiently initiate a fatigue crack at reasonable stress levels.

G.7.5 SPECIMEN I-3

The next specimen (I-3) was prepared with 10 FIB notches of ~0.002″ surface length located on both sides of the specimen. Pre-cracking and testing was conducted at *Smax* = 130 ksi and *R* = 0.00. Representative results (note that the initial FIB notches were 0.002″ long and the initial data points were taken at 2c = 0.0022″) are compared to the previous PDRI results for 0.003″- and 0.004″-long FIB notches (colored lines) in figure G-28. This analysis used the best available estimates of the actual crack shape from the post-test fractography. Note that only two arbitrarily chosen cracks (#3 and #8) of the ten total cracks are plotted. Some differences in the early crack propagation behavior can again be seen.

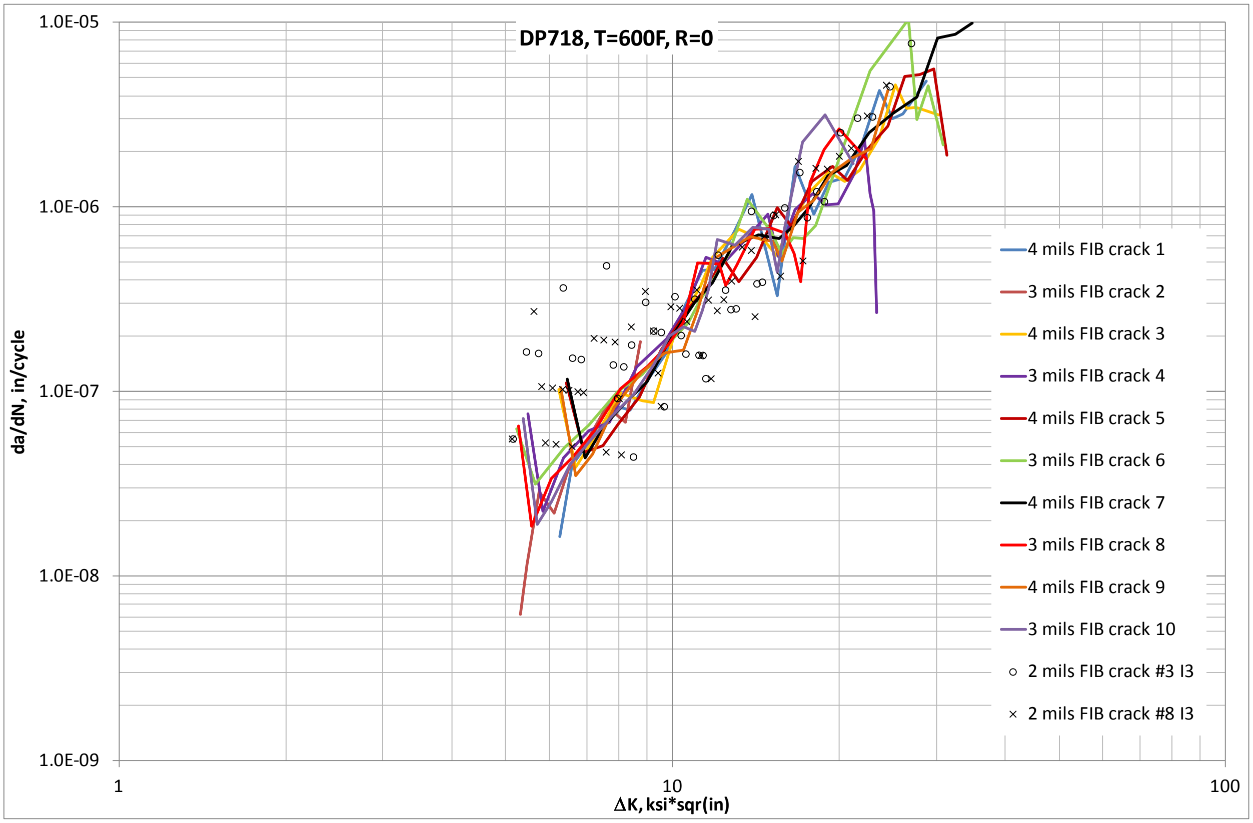


Figure G-28. FCG results for "small" cracks generated using SC(T) specimens with different FIB notches (I-3 analyzed for actual crack aspect ratio)

G.7.6 SPECIMEN J-3

Specimen J3 was prepared with 10 FIB notches of up to 0.002″ surface length located on both sides of the specimen. It was pre-cracked at the same parameters as the previous specimens (R = 0 and Smax = 130 ksi), but it was tested at R = -1. The Smax value was progressively increased from 65 ksi until at Smax = 105 ksi a more-or-less steady crack propagation was observed. Specimen J3 failed from the shoulder radius after up to 33,000 cycles before adequate crack-growth data were collected from the notches.

G.7.7 SPECIMEN K-2

The next specimen, K2, was prepared with 10 notches of 0.003″ surface length. It was pre-cracked at the same parameters as the previous specimens (R = 0 and Smax = 130 ksi), but tested at R = -1 and Smax = 90 ksi.

The results, generated from this specimen containing 10 of 0.003″-long FIB notches, were compared to the previous PDRI results for 0.003″ and 0.004″-long FIB notches (at a stress ratio of R = 0.05) in figure G-29. Note that only two arbitrarily chosen cracks (#3 and #8) are plotted. The results clearly indicate faster crack growth at *R* = -1 compared with *R* = 0. However, no suitable large crack data are currently available at *R* = -1 for comparison.

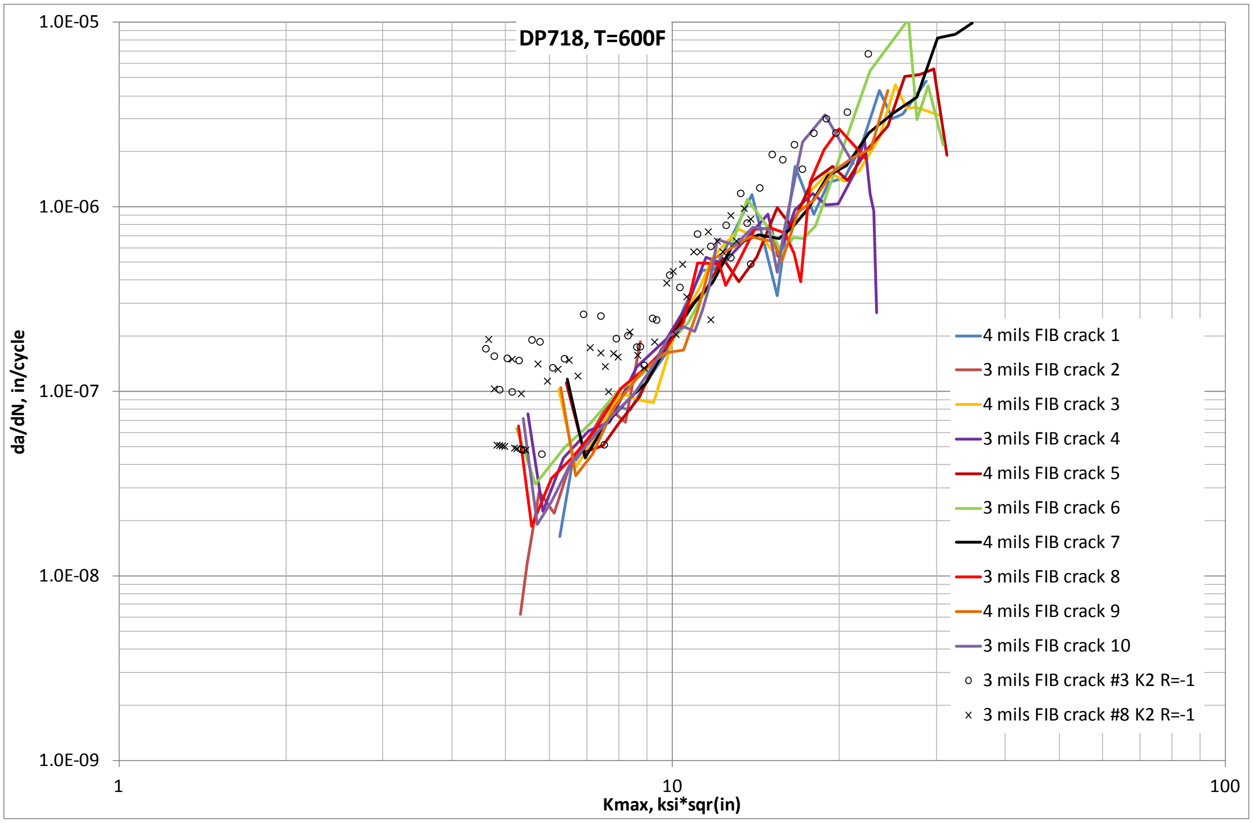


Figure G-29. FCG results for "small" cracks generated using SC(T) specimens with different FIB notches (K2)

G.8 SUMMARY AND CONCLUSIONS

The FCG rate behavior of “large” cracks was studied using compression pre-cracking and ASTM E647 load-shedding methods on a compact tension specimen geometry. After correct application of the compression pre-cracking, the two approaches showed a reasonably consistent behavior with an FCG threshold of approximately 5.0 ksi√in.

These C(T) results were compared with the previous PDRI results for SC(T) specimens with “large” cracks initiated from 0.020″-long EDM notches. At higher ΔK values, the two geometries produced similar results but exhibited some differences at lower ΔK values.

The FCG behavior of “small” cracks was studied using SC(T) specimens containing very small FIB notches with an optimized (semi-elliptical rather than square) shape and the smallest dimensions that produced growing fatigue cracks at reasonable stress levels. FIB notches as small as 0.0010″–0.0016″ were inefficient in initiating fatigue cracks in the given geometry without approaching net-section yielding (YS = 158.0 ksi at 600°F).

Initial comparison of growth rates for cracks initiated from 0.002″-long FIB notches to the previous PDRI results for 0.003″- and 0.004″-long FIB notches showed some differences in the earlier crack propagation behavior. Analysis of these cracks used the best-available estimates of the actual crack aspect ratio based on post-test fractography, and this was shown to have a small but perceptible impact on the results.

APPENDIX H—PIRATE/DARWIN® PUBLICATIONS AND PRESENTATIONS (SEPTEMBER 2011–PRESENT)

Significant results from this research program were disseminated to a much broader audience in the gas turbine engine community or the international technical community in disciplines such as fracture mechanics or reliability through presentations at conferences and symposia or publications in archival technical journals. A number of additional publications and presentations during the term of this grant were based on DARWIN-related research funded primarily by non-FAA sources. This appendix lists all of these publications and presentations, whether based directly on the research in this grant, or based on research funded by other organizations.

H.1 PUBLICATIONS

- McClung, R. C., Enright, M. P., Liang, W., Chan, K. S., Moody, J. P., Wu, W. T., . . . Fitch, S. H. K. (2012). *Integration of manufacturing process simulation with probabilistic damage tolerance analysis of aircraft engine components.* Proceedings from the 14th AIAA Non-Deterministic Approaches Conference, Honolulu, HI.

- Enright, M. P., McClung, R. C., Liang, W., Lee, Y. D., Moody, J. P., & Fitch, S. (2012). *A tool for probabilistic damage tolerance of hole features in turbine engine rotors.* Proceedings from the ASME Turbo Expo 2012, Copenhagen, Denmark.

- McClung, R. C., Lee, Y. D., Enright, M. P., & Liang, W. (2012). New methods for automated fatigue crack growth and reliability analysis. *ASME Journal of Engineering for Gas Turbines and Power, 136*(6), 7 pp. Also published as Proceedings from the ASME Turbo Expo 2012, Copenhagen, Denmark.

- Ma, F. (2012). *A superposition approach for time-dependent fatigue crack growth.* Proceedings from the ASME Turbo Expo 2012, Copenhagen, Denmark.

- Lenets, Y. N. (2012). *Practical aspects of fatigue crack growth in Aero-GTE applications.* Proceedings of ASME Turbo Expo 2012, Copenhagen, Denmark.

- Chan, K. S., Enright, M. P, Moody, J. P., Hocking, B., & Fitch, S. H. K. (2012). Life prediction for turbopropulsion systems under dwell fatigue conditions. *Journal of Engineering for Gas Turbines and Power, 134*, 8. Also published as Proceedings from the ASME Turbo Expo 2012, Copenhagen, Denmark.

- Enright, M. P., Moody, J. P., Chandra, R., & Pentz, A. C. (2012). *Influence of mission variability on fracture risk of gas turbine engine components.* Proceedings from the ASME Turbo Expo 2012, Copenhagen, Denmark.

- Enright, M. P., McFarland, J., McClung, R. C., Moody, J. P., Wu, W. T., & Shankar, R. (2013). *Probabilistic integration of material process modeling and fracture risk assessment using Gaussian process models.* Proceedings from the 15th AIAA Non-Deterministic Approaches Conference, Boston, MA.

- Enright, M. P., McFarland, J., McClung, R. C., Wu, W. T., & Shankar, R. (2013). A *Probabilistic framework for incorporating material processing parameters into fracture risk computations*. Proceedings from the 11th International Conference on Structural Safety and Reliability (ICOSSAR), New York, NY.

- Moody, J. P., Enright, M. P., & Liang, W. (2013). An *optimal automated zone allocation approach for risk assessment of gas turbine engine components*. Proceedings from the ASME Turbo Expo 2013, San Antonio, TX.

- Chan, K. S., Enright, M. P., & Moody, J. P. (2014). Development of a probabilistic methodology for predicting hot corrosion fatigue crack growth life of gas turbine engine disks, *Journal of Engineering Gas Turbines and Power*, *136,* 8. Also published as *Proceedings of ASME Turbo Expo 2013*, San Antonio, TX.

- McClung, R. C., Lee, Y. D., Cardinal, J. W., & Guo, Y. (2013). *The pursuit of K: Reflections on the current state of the art in stress intensity factor solutions for practical aerospace applications.* Proceedings from the 27th Symposium of the International Committee on Aeronautical Fatigue and Structural Integrity, Jerusalem, Israel.

- McClung, R. C., Enright, M. P., Moody, J. P., Lee, Y. D., & McFarland, J. (2014). *Integrating fatigue crack growth into reliability analysis and computational materials design*. Proceedings from the 11th International Fatigue Congress, Melbourne, Australia. Also published in *Advanced Materials Research*, *891-892*, 1009–1014.

- Chan, K. S., Enright, M. P., Moody, J., & Fitch, S. H. K. (2014). A microstructure-based time-dependent crack growth model for life and reliability prediction of turbopropulsion systems. *Metallurgical and Materials Transactions A*, *45A*, 287–301. doi: 10.1007/s11661-013-1971-9.

- McClung, R. C. (2015). Integrating fracture mechanics into the material and structural design process. Proceedings from the *Jerry L. Swedlow Memorial Lecture, 15th International ASTM/ESIS Symposium on Fatigue and Fracture Mechanics (40th National Symposium on Fatigue and Fracture Mechanics),* Anaheim, CA.

- McClung, R. C., & Bhamidipati, V. (2016). An investigation of small-crack effects in various aircraft engine rotor materials. *Materials at High Temperatures, 33*(4-5), 452–464.

- McClung, R. C., Wawrznyek, P., Lee, Y. D., Carter, B. J., Moody, J. P., & Enright, M. P. (2016) *An integrated software tool for high fidelity probabilistic assessments of metallic aero-engine components.* Proceedings from the ASME Turbo Expo 2016, Seoul, South Korea.

- Enright, M. P., Moody, J. P., & Sobotka, J. C. (2016). *Optimal automated risk assessment of 3D gas turbine engine components*. Proceedings from the ASME Turbo Expo 2016, Seoul, South Korea.

- Enright, M. P., McClung, R. C., Chan, K .S., McFarland, J., Moody, J. P., & Sobotka, J. C. (2016). *Micromechanics-based fracture risk assessment using integrated probabilistic damage tolerance analysis and manufacturing process models*. Proceedings from the ASME Turbo Expo 2016, Seoul, South Korea.

- Chan, K. S., Enright, M. P., Moody, J., & Fitch, S. H. K. (2016). Mitigating time-dependent crack growth in Ni-base superalloy components. *International Journal of Fatigue, 82*, 332–341.

- Chan, K. S., & Moody, J. (2016). A hydrogen-induced decohesion model for treating cold dwell fatigue in Ti-based alloys. *Metallurgical and Materials Transactions A, 47A*, 2058–2072. doi: 10.1007/s11661-016-3367-0.

- Chan, K. S. (2016). Mechanistic modeling of time-dependent fatigue crack growth in Ni-based superalloys. *Materials at High Temperatures*, *33*(4-5), 425–438.

- Chan, K. S., Enright, M. P., Moody, J., & Fitch, S. H. K. (2016). *Physics-based modeling tools for predicting type ii hot corrosion in Ni-based superalloys*. Manuscript submitted for publication..

H.2 PRESENTATIONS (NO MANUSCRIPT)

- McClung, R. C. (2011). *A new computational framework for component life and reliability*., Proceedings from the Winter Meeting Integrative Materials Design Center, Worcester, MA.

- McClung, R. C., Lee, Y. D., Chan, K. S., Enright, M. P., & Fitch, S. H. K. (2012). *Methods for temperature effects in probabilistic damage tolerance analysis.* Proceedings from the Propulsion—Safety and Affordable Readiness (P-SAR) Conference, Jacksonville, FL.

- McClung, R. C., & Bhamidipati, V. (2012). *An investigation of small-crack effects in various aircraft engine rotor materials*. Proceedings from the Propulsion—Safety and Affordable Readiness (P-SAR) Conference, Jacksonville, FL.

- Wawrzynek, P., McClung, R. C., Lee, Y. D., & Carter, B. (2012). *A three-dimensional crack growth tool for probabilistic fatigue life predictions.* Proceedings from the Propulsion—Safety and Affordable Readiness (P-SAR) Conference, Jacksonville, FL.

- Enright, M. P., McClung, R. C., Shankar, R., Wu, W. T., Chan, K. S., & McFarland, J. (2012). *Integrating probabilistic lifing with manufacturing process modeling*. Proceedings from the Propulsion—Safety and Affordable Readiness (P-SAR) Conference, Jacksonville, FL.

- Chandra, R., Enright, M. P., Pentz, A., & Moody, J. P. (2012). *Mission variability effects on probabilistic risk of gas turbine engine components*. Proceedings from the Propulsion—Safety and Affordable Readiness (P-SAR) Conference, Jacksonville, FL.

- McClung, R. C., Enright, M. P., Wu, W. T., & Shankar, R. (2013). *Integrating computational materials engineering into probabilistic damage tolerance analysis for component design.* Proceedings from the TMS Annual Meeting & Exhibition, San Antonio, TX.

- Chan, K. S., Enright, M. P., Moody, J. P., & Fitch, S. (2013). *A microstructure-based time-dependent crack growth model for life and reliability prediction of turbopropulsion systems*. Proceedings from the TMS Annual Meeting & Exhibition, San Antonio, TX.

- McClung, R. C., Enright, M. P., McFarland, J., Chan, K. S., Wu, W. T., & Shankar, R. (2014). *Linking ICME to component life management during design*. Proceedings from the TMS Annual Meeting, Symposium on ICME: Linking Microstructure to Structural Design Requirements, San Diego, CA.

- Chan, K. S., Enright, M. P., Moody, J., & Fitch, S. H. K. (2014). *Mitigating time-dependent crack growth in Ni-Base superalloy components*. Proceedings from the International Conference on Fatigue Damage of Structural Materials X, Hyannis, MA.

- McClung, R. C., & Enright, M. P. (2015). *Current trends in probabilistic damage tolerance analysis with DARWIN*. Proceedings from the Propulsion—Safety and Affordable Readiness (P-SAR) Conference, Baltimore, MD.

- Moody, J. P., Sobotka, J. C., Enright, M. P., & McClung, R. C. (2015). *An optimized automated methodology for the assessment of fracture risk in gas turbine engine components*. Proceedings from the Propulsion—Safety and Affordable Readiness (P-SAR) Conference, Baltimore, MD.

- Enright, M. P., McClung, R. C., McFarland, J., Moody, J. P., Chan, K. S., Sobotka, J. C., Shankar, R., & Wu, W. T. (2015). *Micromechanics-based fracture risk assessment using integrated probabilistic damage tolerance analysis and manufacturing process models.* Proceedings from the Propulsion—Safety and Affordable Readiness (P-SAR) Conference, Baltimore, MD.

- Wawrzynek, P., Lee, Y. D., Moody, J. P., Carter, B. J., McClung, R. C., & Enright, M. P. (2015). *A software tool for high fidelity probabilistic fatigue life assessments of metallic aero-engine components*. Proceedings from the Propulsion—Safety and Affordable Readiness (P-SAR) Conference, Baltimore, MD.

- Chan, K. S., Enright, M. P., Moody, J. P., & Fitch, S. H. K. (2015). *Modeling variability of time-dependent fatigue crack growth in Ni-base superalloy disks*. Proceedings from the Propulsion—Safety and Affordable Readiness (P-SAR) Conference, Baltimore, MD.

- McClung, R. C. (2015). *Don't let it break: Using probabilistic fracture mechanics to transform the material and structural design process*. Proceedings from the Distinguished Seminar, Department of Mechanical, Aerospace and Biomedical Engineering, University of Tennessee, Knoxville, TN.

- Enright, M. P., McClung, R. C., Chan, K. S., McFarland, J., Moody, J. P., & Sobotka, J. C. (2016). *Micromechanics-based fracture risk assessment using integrated probabilistic damage tolerance analysis and manufacturing process models.* Proceedings from the Propulsion Safety and Sustainment Conference, Grapevine, TX.

- Enright, M. P., Moody, J. P., & Sobotka, J. C. (2016). *Optimal automated fracture risk assessment of 3D gas turbine engine components.* Proceedings from the Propulsion Safety and Sustainment Conference, Grapevine, TX.

- Gray, J., Gray, I., Enright, M.P., McClung, R.C., McFarland, J., & Moody, J.P. (2016). Integration of simulated NDE POD curves with structural reliability assessments. Proceedings from the Aircraft Airworthiness and Sustainment Conference, Grapevine, TX.