UNCLASSIFIED

Australian Government
Department of Defence
Defence Science and
Technology Organisation

Guide on the Effective Block Approach for the Fatigue Life Assessment of Metallic Structures

Marcus McDonald

Air Vehicles Division
Defence Science and Technology Organisation

ABSTRACT

Simpler empirical characterisation of the mechanics of metal fatigue crack growth can offer an alternative to physics-based models, where the latter can be challenged by complex material behaviour of commercial metals when loaded by variable amplitude spectra. Such characterisation is anticipated to be more appropriate for cases where the loading consists of repeating blocks of spectra or other scenarios that result in near steady-state crack growth rates. This report examines the many types of characteristic block approaches available, and recommends a particular type that is considered suitable for fatigue life assessments, herein titled the Effective Block Approach, or EBA. Guidance is given on methods and implementation, and several case studies are reviewed where the EBA has been used to assess the life of RAAF airframes. In addition to computational efficiency, significant accuracy gains were achieved when the approach was underpinned by representative coupon tests. This capability is anticipated to give the RAAF more options to better assess and manage the safety, availability and cost of ownership of its air vehicles.

RELEASE LIMITATION

Approved for public release

UNCLASSIFIED

APPROVED FOR PUBLIC RELEASE

Guide on the Effective Block Approach for the Fatigue Life Assessment of Metallic Structures

Executive Summary

Most RAAF aircraft are life-limited due to the metal fatigue failure mode. Predictive modelling is necessary for effective structural integrity management of these life limits. Mechanistic type approaches that attempt to model the physical and chemical behaviour of the metal fatigue mechanism can be difficult to develop to a sufficient level of reliability in a given timeframe. Variable amplitude loadings that cause retardation/acceleration effects are particularly difficult to simulate. This can significantly restrict the RAAF's options available for managing the structural integrity of its air vehicles. However, if the variable amplitude loading consists of regularly repeating blocks, or otherwise is considered to produce steady-state crack growth rates, then a characteristic block type modelling approach may offer an alternate path for conducting life assessments. This is anticipated to be particularly useful for full-scale test interpretation where the applied loading usually consists of repeating spectrum blocks.

Characteristic block type approaches consider the average crack growth rates for a spectrum block, and characterise these against the predominant influential parameters (*e.g.*, crack size, stress or stress intensity factor). This report examines the many types of characteristic block approaches described in the literature, and highlights that the assumptions and efficacy of each can vary significantly. Guidance is given on distinguishing amongst them, and a recommendation is given for a particular type that is considered suitable for conducting fatigue life assessments, herein titled the Effective Block Approach, or EBA. This guide provides steps for using an EBA; considers the benefits and limitations; and reviews a number of case studies. It was found that, when the crack growth rates are underpinned by representative fatigue tests, an EBA can provide significant accuracy, efficiency and insights toward prediction of metal fatigue crack growth.

DSTO has significant full-scale, component and coupon test facilities that it uses to assess RAAF airframes. The efficient and accurate modelling of crack growth is a necessary capability for translating the test outputs into relevant life results for structural integrity management purposes. The EBA enables DSTO to continue to offer leading structural integrity advice, even when mechanics-based models are still being developed. Already, the EBA has been used to assess numerous airframes including Macchi wings, F/A-18 Hornet centre barrels and F-111 wings. This capability is anticipated to give the RAAF additional options to better assess and manage the safety, availability and cost of ownership of its air vehicles.

This page is intentionally blank

Author

Marcus McDonald
Air Vehicles Division

Mr McDonald completed a Bachelor of Engineering (Mechanical) at the University of Queensland in 1994. Prior to joining DSTO in 1998, he spent 3 years working in the oil & gas, steel and railway industries for structural design projects and failure investigations. He has since worked a further 14 years in the aeronautical field at DSTO as an Aircraft Structural Integrity engineer providing advice to the RAAF on numerous aircraft including F-111, F/A-18F Super Hornet, F 35 Lightning II and the F/A-18A/B Hornets. He was attached to the US Naval Air Command (2005-2008) as an airworthiness engineer for the new F-35 combat aircraft. Marcus is currently the technical lead for novel airframe lifing and test interpretation methods, and leads a task to support the airworthiness of RAAF trainer aircraft including the PC-9 and Hawk lead-in fighter.

This page is intentionally blank

Contents

Abbreviations

AA	Aluminium Alloy
CA	Constant Amplitude
CG	Crack Growth
CGAP	Crack Growth Analysis Package, incorporating the FASTRAN model
CGM	Crack Growth Model
CGR	Crack Growth Rate
CPLT	Cold Proof Load Test
EBA	Effective, or Equivalent, Block Approach
EDM	Electrical Discharge Machine
EICS	Equivalent Initial Crack Size
EPFM	Elastic Plastic Fracture Mechanics
FEM	Finite Element Model
FWELD	F-111 Wing Economic Life Determination program
Kt	Stress concentration factor
LEFM	Linear Elastic Fracture Mechanics
LITV	Load Interpretation Truncation Validation coupon test program
NDI	Non-Destructive Inspection
QF	Quantitative Fractography
RAAF	Royal Australian Air Force
USAF	United States Air Force
VA	Variable Amplitude

This page is intentionally blank

1. Introduction

The assessment of the metal fatigue life of an aircraft type, from a full-scale article tested using a variable amplitude (VA) *test*[1] load-sequence, to an *in-service* or other desired sequence, can be done by a comparative coupon test or a validated analysis. This is usually appropriate if the sequences are considered to be similar, *i.e.,* the airframe failure modes are not altered. Despite advances in state-of-the-art mechanistic (physics-based) crack modelling, representative coupon test results are still widely regarded as the more reliable method for determining the effect of load-sequence differences. Where crack modelling is used, validation by coupon tests is still usually a requirement for aircraft structural integrity. This reliability may be pertinent for managing a military capability to a planned withdrawal date or a specified life-of-type.

Modelling of metal fatigue has an important role in providing the analyst with the flexibility to analyse a wider range of conditions, and hence limiting the number of comparative coupon tests required for calibration purposes. In other words, the coupon tests provide data to support a basic understanding of the load-sequence effects, and the resulting calibrated model may provide reasonable extrapolations for untested parameter values, *e.g.,* other similar spectra, stress scale levels, geometries, crack sizes, crack shapes. Characterising fatigue against these parameters in terms of crack growth (CG) or CG rate (CGR) has become more popular[2], particularly when studying load-sequence effects.

Past practice has demonstrated, usually during the the calibration process, that it can sometimes be difficult to achieve confidence in a mechanistic CG model [1-4].. A notable example is the F-111, whereby roughly 15 years were devoted to the calibration of numerous mechanistic CG models without reliable success [3]. This difficulty was attributed to the complex material mechanics, *i.e.,* significant CG-retardation effects caused by the regular Cold Proof Load Test (CPLT) applied to the wings; different observed behaviours for different materials (steel *vs.* aluminium); and relatively short cracks subjected to high notch stress levels. Many of these factors usually exceeded the bounds of validity of linear-elastic fracture mechanics (LEFM)[3].

The F/A-18 Hornet program also experienced similar issues; however, rather than pursuing a mechanistic model calibration effort, it adopted an alternate more-direct use of its coupon test data by characterising the CG rates using simplified, pseudo-ductile (*i.e.,* non-linear) fracture mechanics relationships [5]. These empirical relationships were simple, yet they were able to provide reasonable extrapolations for many useful parameters, *e.g.,* crack sizes and stress scale levels, including trends that could not be simulated by LEFM or constant amplitude (CA)-based models [6]. Importantly, it allowed

---

[1] considerations may also be given to alterations made to the applied load sequence for more efficient application to the test article, *e.g.* approximations (discretisation) of the aero and inertia load distributions, truncation filtering, make-up cycles, or increased load scale levels in-lieu of cycles.

[2] even more-so with the advent of quantitative fractography.

[3] LEFM forms the basis of most state-of-art CG models.

the coupon test program to focus on characterising a wider range of different load-sequences. The F-111 wing life assessment program later adopted a similar approach [7], which also allowed a larger number of different load-sequences to be characterised by testing *pro* analysis predictions.

A common feature of the F/A-18 Hornet and F-111 test programs was that the load spectra consisted of repeating blocks of load-sequences, *i.e.*, relatively short[4] periods of flying that represent a mix of combat missions. This allowed a characteristic block type approach to be used, whereby the load-sequence effect was quantified by the average CG rate for a given block of loading. Block-type characteristic approaches appear to have been available in the fatigue lifing community for as long as VA spectra have been considered in lifing structures. However, there are many types of approaches purported in the literature; and not all are credible. This report serves as a guide for the selection and use of a block-spectrum characteristic approach, for the purpose of improving or facilitating the conduct of life assessments on RAAF aircraft.

Firstly, a number of characteristic type approaches are distinguished in Section 2, and those with more serious limitations are highlighted. Section 3 details the particular approach that has been adopted by DSTO, referred to as the *Effective Block Approach* or *EBA*. This has already been used for a number of aircraft life re-assessments including Macchi [8], F/A-18 [2, 6, 9], F-111 [7, 10], an Infantry Mobility Vehicle life assessment [11] and other applied research [12-22]. A number of CGR relationships, referred to as models, are presented in Section 4, along with some advice on which is best suited for particular scenarios. Section 5 provides more detail on the individual implementation steps, including some methods for incorporating supplementary analyses or adjustments – importantly, how to incorporate predictions of load-sequence effects for untested spectra. Several case studies are then examined in Section 6, in particular where the EBA has been used to provide structural integrity advice to the RAAF, and thus has been subject to extensive peer-review and additional validation testing. Finally, a discussion is given on the efficacy of the EBA for general use.

---

[4] compared to the total aircraft life, *i.e.*, a single block usually represents 100-1000 flight hours.

2. Characteristic Approaches for Variable Amplitude Load Spectra

There are a variety of characteristic approaches that have been used to model CGRs for VA spectra in the literature. Generally, these approaches attempt to characterise relatively complex load sequence effects using very simple, computationally-efficient models or algorithms, *i.e.*, they usually avoid mechanistic crack-tip material modelling. The supposition of each however can vary significantly - some attempt to use numerical or statistical analyses to characterise the VA sequence, while others rely on test results; some apply relatively arbitrary curve fits to CGR data, while others apply similitude and/or material property constraints. Therefore some may be better suited than others for the purpose of improving or facilitating the assessment of the life of an aircraft.

To further complicate matters, there are numerous titles attributed to these approaches - and they are not always consistent or accurate descriptors. The approaches can be referred to as either an effective stress approach [23]; characteristic stress intensity approach [23-25]; root-mean-square approach [23, 26, 27]; equivalent K method [28, 29]; effective block approach [2, 7, 12, 13, 15, 22], equivalent block method [30], mini-block approach [31], block-by-block approach [20], flight-by-flight approach [9, 32], normalised crack growth concept [33], or in many cases no nomenclature is attributed [8, 34, 35].

The following considerations can be used to help distinguish between characteristic approaches:

- **Load-sequence effect determination method (test or analysis)**
- **CGR models and similitude parameter (arbitrary or mechanics-based)**
- **Load-sequence effect incorporation (similitude parameter or material property)**

Distinguishing these approaches is important, because some types have been criticised as inappropriate or limited [2, 23, 36, 37], whereas others have been deemed practical [6, 24, 25, 31, 33]. Other reviews reasonably conclude that there is a limited range of applicability (to be determined) for all these types of block approaches [1, 38]. It becomes important to rationally distinguish between these approaches in order to avoid generalising either praise or criticism.

2.1 Load-sequence effect determination method

The determination of the load-sequence effect by a representative test is regarded as the most reliable method to determine CGRs. Conversely, simple VA spectrum characteristic analysis approaches (*e.g.*, root-mean-square, load counting) have had limited success [37] and are not likely to offer an advantage compared to a mechanistic CG model. While there are examples of success in using simple analysis approaches in the literature, particularly

for random type spectra, major limitations tend to occur for those with occasional over/under loads, which can lead to significant retardation/acceleration effects. Therefore, only approaches that determine CGRs by testing are considered further.

2.2 CGR models and similitude parameters

Approaches that are based on accepted fracture mechanics-based CGR models and similitude parameters are considered more likely to provide a useful range of predictability. The LEFM stress intensity factor, K (for small-scale crack-tip plasticity) appears to have been adopted regularly. However, there does not appear to be common use, if any, of the Elastic Plastic Fracture Mechanics (EPFM) *J-integral* parameter (for larger scale crack-tip plasticity).

Oddly, there are examples that consider <u>very</u> large scale crack-tip plasticity using pseudo-ductile[5] characteristic approaches, *i.e.,* where the plastic zone size may well-exceed the size of the fatigue crack. This is of interest to the assessment of military airframes because some structural components can be highly stressed - often exhibiting very large scale crack-tip plasticity conditions. Furthermore, as most traditional state-of-the-art mechanistic models are still based on the LEFM parameter K, these pseudo-ductile characteristic approaches have the potential to offer an advantage compared to traditional mechanistic CG models.

2.3 Load-sequence effect incorporation

After the load sequence effect of a VA spectrum has been determined, *e.g.,* by test, then it must be attributed or incorporated some way into the characteristic CGR model. Some approaches attribute it to the similitude parameter, *e.g.,* $K_{effective(spectrum1)}$. Another method is to incorporate it as an adjustment to an existing material property, *e.g.,* a Paris-type coefficient $C_{spectrum1}$ [31, 33]): this is similar to incorporating an environmental effect, *e.g.,* salt air, by adjusting an existing so-called material CGR constant.

The former method has been presented in many literature examples, including recent textbooks [23, 29]. They consider treating the load sequence effect as an effective similitude parameter, *i.e.,* a value of the similitude parameter that gives an equivalent CGR. However, this method has received considerable criticism primarily because it has been demonstrated that the load-sequence effect can significantly vary with the magnitude of the similitude parameters [2, 7, 36, 38] – therefore these approaches are incapable (without adding further complex relationships) of characterising trends against the similitude parameter.

The latter method, where the sequence effect may be incorporated into a constant, or multiple constants within a CGR model, has the potential to capture more complex load

---

[5] Perhaps unknowingly – since there are examples of characteristic approaches being applied to high-stress cases seemingly without appreciation of the strain conditions at the crack tip.

sequence trends, which can readily be designed to vary with the magnitude of the similitude parameter. This method has been adopted for use by DSTO.

Some examples incorporate more complex numerical computations within the CGR models, *e.g.*, [30].

2.4 Summary

There are many types of characteristic approaches, and the aim is to down-select to one or several that maintain simplicity and yet offer a useful alternative or advantage, either in support, or in lieu of, a calibrated mechanistic approach. This can be achieved, firstly by limiting to approaches that use testing (not analysis) to determine CGRs, secondly by adopting generally accepted CGR relationships and similitude parameters, and finally, by judicious incorporation of the load-sequence effects into parameters or constants such that significant trends can be simulated. The particular approach adopted by DSTO follows this guidance, and is detailed in the following section.

3. The Effective Block Approach

The *Effective Block Approach*, or EBA[6], is a structured CGR analysis methodology for the purpose of interpreting, analysing and extrapolating CGRs for metallic materials subjected to **relatively short and repeated blocks of VA load sequences.** With careful consideration, it may also be appropriate for assessing general types of flight loading or mission mixes that are considered to generate steady CGRs, such as life extrapolation using a relatively constant future mission-mix. It has similarities with the mini-block approach developed by Gallagher [31] for the analysis of steady-state spectra.

The EBA relies on representative fatigue testing to provide the CGR data for the given VA spectra. The EBA does not utilise a cycle-by-cycle type model for determining load-sequence effects, therefore it has no predictive ability to estimate the CGR behaviour of a particular VA spectrum - it relies on representative test data to provide this information, or other supplementary analysis to provide an estimate.

The EBA fits simple models[7] (empirical equations) to the VA test data to represent the *near average[8]* CGRs. The models are based on well-established existing CGR relationships (*e.g.*, Paris, Tomkins, Frost & Dugdale) based on linear-elastic or pseudo-ductile (elastic-plastic) fracture mechanics, therefore it is more disciplined than say curve fitting using arbitrary equations. This allows it to provide more reasonable estimates of CGRs for other untested combinations of the correlating parameters, thus enabling more accurate interpolations and also extrapolations within reasonable limits. Even though the CGR models are considered simple, it will be later shown that some of them can readily fit CGR observations that cannot be simulated by traditional LEFM and CA-based models, particularly where CA-based approaches may be disadvantaged by limited CGR data for short cracks.

The EBA may require supplementary adjustments to consider scenarios either not covered by the testing program or not represented by the model correlating parameters. One important scenario is the predictions of CGRs for untested VA spectra – other important scenarios are listed in the dot points further below.

The EBA inherently incorporates load-sequence effects into the so-called material constants[9] during the fitting process, *e.g.*, into the C and m constants in a Paris type model.

---

[6] The EBA is named as such to contrast it from other characteristic type approaches whereby attempts are made to characterise the severity of a VA spectra by a *statistic* or *analysis* of a load-sequence.

[7] There are a number of CGR models that can be selected - shown later in this report.

[8] The actual method described in this report uses a least squares regression fit.

[9] This is in contrast to the *effective stress approach* often referred to in handbooks, whereby the effects of the variable amplitude spectrum are incorporated into a so-called effective stress or effective stress intensity factor. This however cannot account for situations where the VA sequence effect on the CGR actually varies with the magnitude of the effective stress or stress intensity factor.

It will also incorporate other characteristics into these constants depending on the mechanic-ability of the chosen CGR model and also the design of the test used to generate the CGR. Examples include:

- residual stress effects (material manufacture process, assembly fit-up stresses, cold-expansion, *etc*)
- notch geometry (high or low Kt)
- notch-plasticity (*e.g.*, fastener holes under high stress)
- scale of crack-tip yielding (*c.f.* small-scale yielding limits)
- short and long crack behaviours
- stable-tearing
- material grain orientation
- stress-state or thickness constraint (plane stress, plain strain, finite thickness)
- loading constraints (clamped, pinned)
- surface condition (machined, etched, anodised, *etc*)
- environment (temperature, humidity, fuel, salt fog, *etc*)
- load rate (1 Hz, 20 Hz, *etc*)

Note that many of these also affect current CA-based mechanistic models, in that they are incorporated into the so-called material constants. Some of these effects have lead to the development of supplementary adjustments – and it can be expected that these would likely also apply to an EBA. Of particular importance to the EBA is that some are known to significantly influence load-sequence effects, *e.g.*, a difference in the notch geometry may result in a different CGR relationship, even after correcting for the geometry effects in accordance with LEFM practice. This places great importance on the design of *representative* fatigue tests in order to provide CGR data that is accurate and relevant for the component of interest. Note that this as *equally important* for the validation of CA-based models as it is for input to an EBA.

For the EBA, diligent quality control of the derived model constants is required as they are explicitly tied to the load sequence and potentially the other factors listed above. Due to the recommendation that the EBA is based on test data under representative airframe conditions (as opposed to CA-based methods that are usually based on data from simplified and well-limited CA CGR test standards) then it is more likely that many of these conditions will need to be considered as *part of* the EBA model. Hence this requires additional discipline in the use of the model and the interpretation of its results when applied to different scenarios.

A number of combat aircraft fatigue programs (F-4 [33], Macchi [8], F/A-18 [39] and F-111 [7]) have shown that CGR trends for commercial metal alloys subject to VA spectra tend to be fitted well by relatively simple models. Examples of such models are given in the next section. It will also be shown later that the fits tend to hold well over a useful range of crack sizes and stress conditions, even when the load sequences are complex (*e.g.*, overloads).

Quick Review

- The Effective Block Approach is a framework for characterising CGRs for <u>repeated blocks of VA load spectra</u>

- The CGRs are based on <u>representative</u> fatigue test data

- It utilises simple models (equations) based on well-established CGR relationships and correlating parameters

- Load sequence and other effects are incorporated into the so-called material CGR constants

- Supplementary adjustments may be required for untested VA load sequences and other factors

- Diligent quality control of the model constants is required

4. Models

The EBA can use any type of model or correlating parameters to suit a particular problem or project, and it is not limited to those described in this report. There are numerous CGR models for CA loading in the literature that can be readily adopted for use in a VA loading scenario. The models that are considered to be the most practical are ones that use **well-established correlating parameters** and have a **simple form**, yet are able to **fit a wide range of test data**. There are two broad groups of models that are considered here based on the correlating parameters. The first are models that simply relate the CGR to a function of *crack size* and *applied stress*. The second are models based on LEFM, where the CGR model has a primary relationship to the *stress intensity factor*.

4.1 Crack size and stress

The general form of this model is given by Tomkins [40] and was originally used to describe CA CGRs. It has been re-formatted[10] here as:

$$\frac{da}{dt} = A a^{j} \sigma_{ref}^{k} \tag{1}$$

where a is the crack size, t is the number of flight hours (flthr), σ_{ref} is the stress at the VA reference condition, and A, j and k are the empirical constants fitted to test data. The constants j and k are non-dimensional and the preferred working units for A are [metre]$^{(1-j)}$·[flthr]$^{-1}$·[MPa]$^{-k}$. Note that the units for A depend on the values of j and k, therefore comparing spectra severity can be done by comparing A values alone, but only when the values of j and k are the same respectively for each spectra.

Figure 1 shows the effect of varying each of the constants about a baseline example, $A = 1E\text{-}11$ flthr^{-1}·MPa^{-3}, $j = 1$, $k = 3$ at an applied reference stress of 350 MPa. In this figure the constants have been varied individually to illustrate how much change is required to roughly halve or double the fatigue life. Figure 1(c) illustrates that as k increases then the CGR becomes more sensitive to changes in stress level. Figure 1(c) also shows that if data is only available for one stress level then the curve fit will be non-unique thus any k value could be chosen. Here the 350 MPa example for $k = 2$ is replicated using $k = 3$, which is possible by simply adjusting the A constant. In this case additional data for a different stress, or other multi-stress data for the same material (different spectrum for example), would be required to estimate an appropriate value for k.

It has been found that this type of model is well suited to cracks where there are no significant changes in crack shape or loading conditions (*e.g.,* load shedding). It is particularly well-suited to stress levels that approach or exceed the material yield stress, as

---

[10] Tomkins' original formula was for CA spectra and so contained a stress *range* term – this has been replaced here with a *reference* stress term for purpose of assessing VA spectra.

this is when CGRs tend to approach an *exponential* relationship[11] (*i.e.,* CGR is proportional to the crack size, or $j = 1$). Tomkins shows a strong dependence (proportionality) between CGR and the crack size when metals are subject to high stress [41]. It can be seen that there is no parameter to account for Kt, hence once the model is fitted to a particular data set it is usually limited to similar Kt scenarios [16].

Frost and Dugdale found that the constants $j = 1$ and $k = 3$ were able to fit test data for a range of materials under CA loading [42]. These values respectively imply that the CGRs are proportional to crack size and proportional to the cube of the applied stress level. These proportionality relationships have been confirmed by a significant amount of test data for the F/A-18 aircraft (see Section 6.2), and are otherwise known as the *exponential* or *log-linear rule* and the *stress-cubed rule* respectively. At DSTO, the following *exponential* version has been used in numerous lifing assessments [2, 5, 8, 39, 43]:

$$\frac{da}{dt} = \lambda a \sigma_{ref}^{\alpha} \tag{2a}$$

where λ and α are empirical constants. Note that this can also be re-arranged into a log(a)-linear(t) form as follows:

$$\frac{d[\ln(a)]}{dt} = \lambda \sigma_{ref}^{\alpha} \tag{2b}$$

---

[11] It is thought that an exponential CGR relationship tends to occur where there is a relatively uniform stress distribution ahead of the crack tip. This is highly relevant for combat aircraft where parts tend to be optimised for relatively even distributions of internal loads and stresses. It is also thought that locally high stresses, where the material yield strength is approached or exceeded (e.g. fastener holes), localised stress re-distributions can occur that can also produce conditions that result in an exponential CGR relationship.
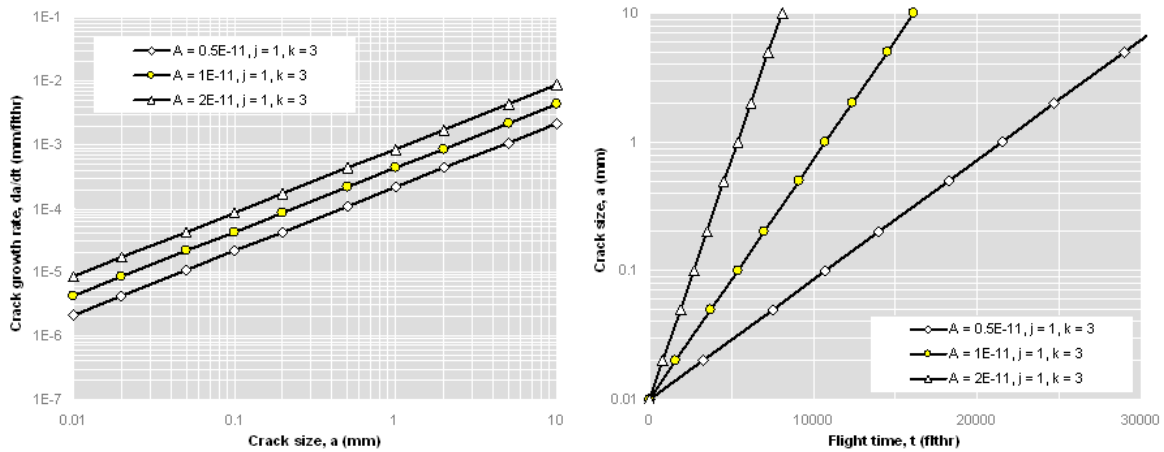
(a) varying constant A
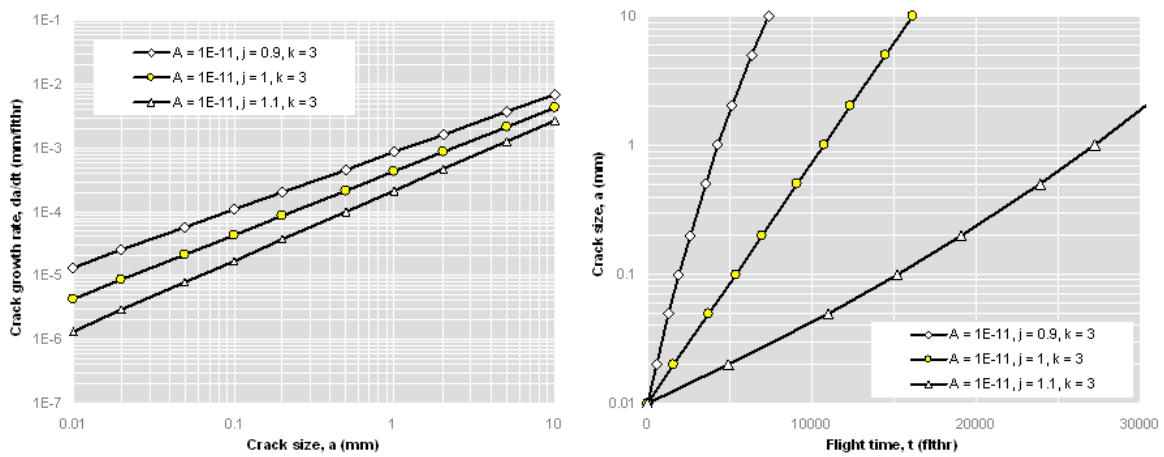
(b) varying constant j

(c) varying constant k

Figure 1: Effect of varying the constants A, j and k in Eqn (1) about a baseline example of: A = 1E-11 flthr^{-1}·MPa^{-3}, j = 1, k = 3.

4.2 Stress intensity factor

This simple model relates the CGR primarily to the *stress intensity factor* and also the *applied stress* (from [17]):

$$\frac{da}{dt} = H\,K_{ref}^{p}\,\sigma_{ref}^{q} \tag{3}$$

where K_{ref} is the reference stress intensity factor, σ_{ref} is reference stress, and H, p and q are the empirical constants. Note that σ_{ref} can be defined as either the gross or net-section stress - however, since its purpose is to adjust the CGR based on the general stress experienced in the crack section, it is usually defined as the net-section stress.

The constants p and q are non-dimensional and the preferred working units for H are $[metre]^{(1-p/2)} \cdot [flthr]^{-1} \cdot [MPa]^{-(p+q)}$. Note that the units for H depend on the values of p and q, therefore comparing spectra severity can be done by comparing H values alone, but only when the values of p and q are the same respectively for each spectra.

The reference stress intensity factor term is approximated by LEFM theory as:

$$K_{ref} = \sigma_{ref}\,\beta\sqrt{\pi a} \tag{4}$$

where β is a geometry function that usually accounts for crack shape, component geometry, and loading boundary conditions - see [44-48] for a range of solutions. Note that β is usually defined based on the gross-section stress. The use of K as a correlating parameter makes this model consistent with most LEFM-based CGR models, and also with fracture strength data for calculating the critical crack size.

Figure 2 shows the effect of varying each of the constants about a baseline example, $H = 1E-9$ flthr^{-1} $\cdot$MPa^{-2}, $p = 2$, $q = 0$ at an applied reference stress of 350 MPa with $\beta = 1$. In this figure the constants have been varied individually to illustrate how much change is required to roughly halve or double the fatigue life. Figure 2(c) shows that as q increases then the crack growth rate becomes more sensitive to changes in stress level. Figure 2(c) also shows that if data is only available for one stress level then the curve fit will be non-unique thus any q value could be chosen. Here the 350 MPa example for $q = 0$ is replicated using $q = 1$, which is possible by simply adjusting the H constant. In this case additional data for a different stress, or other multi-stress data for the same material (different spectrum for example), would be required to estimate an appropriate value for q.

This type of model ought to be well suited to cracks where there are significant differences in geometry (Kt), changes in crack shape, and more complicated loading conditions (*e.g.,* load shedding). This is due to the presence of the β factor which can be used to account for these effects[12]. The stress term on the right most side of Eqn (3) represents an additional factor on the simpler Paris-type model.

---

[12] However, it will be shown later that for high stress cases from the F/A-18 Hornet program, the test data only partially support the use of the β factor to account for Kt effects.

The Paris-type model is given by:

$$\frac{da}{dt} = C K_{ref}^{m}$$
(5)

where C and m are empirical constants. This model has been used in numerous VA spectrum lifing assessments at DSTO [2, 7, 11].

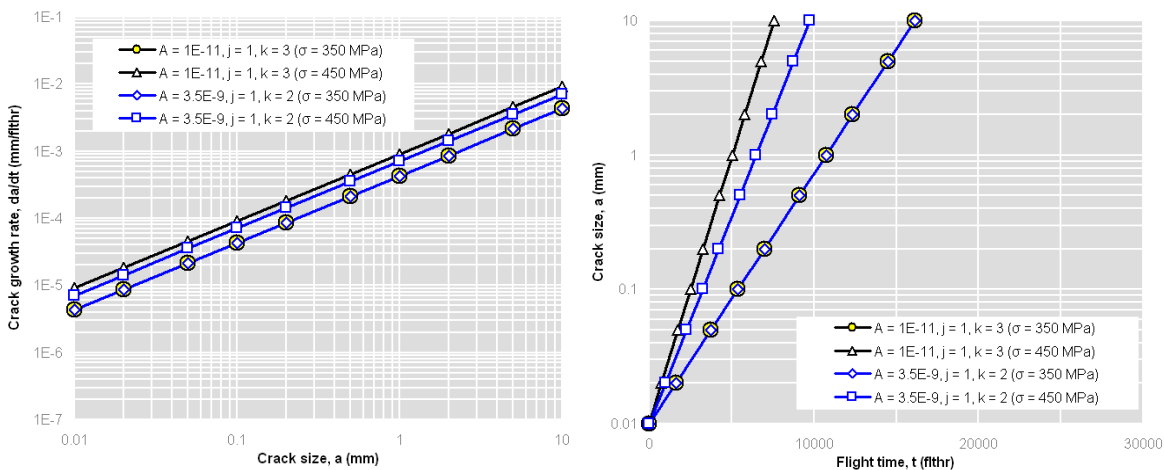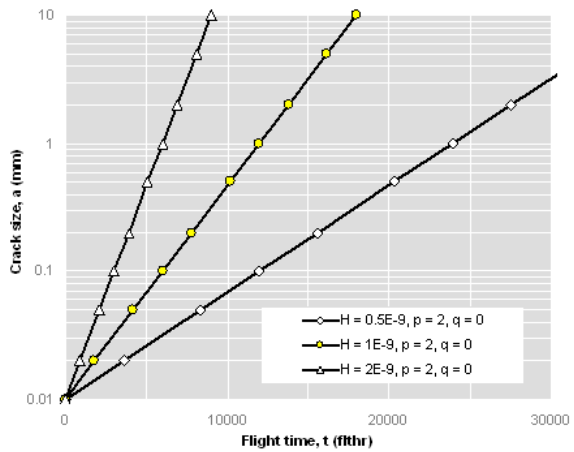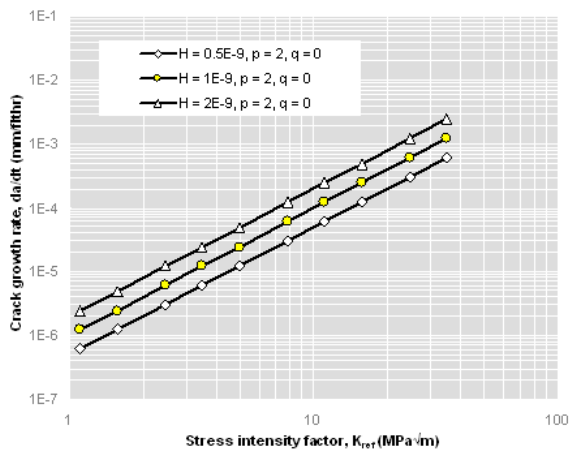*(a) varying constant H*
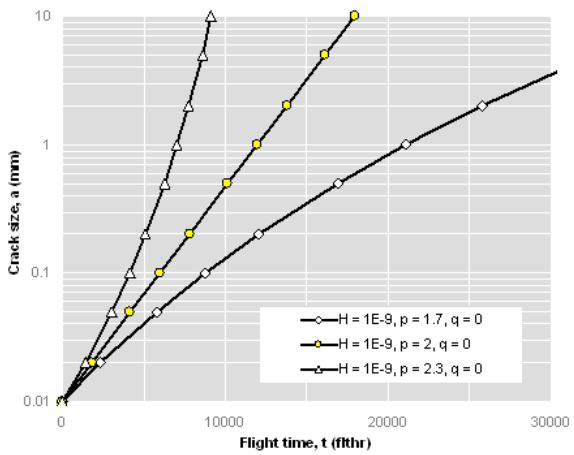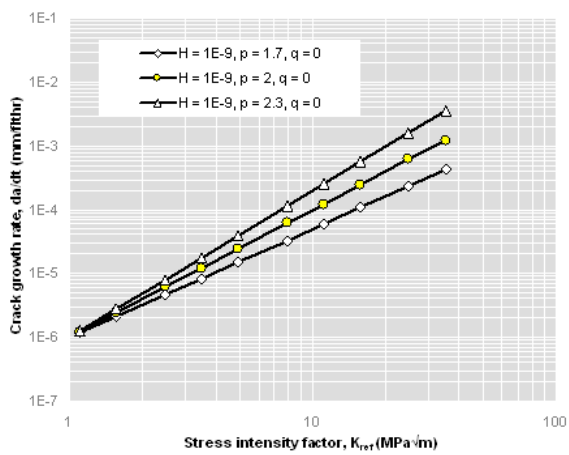
(b) varying constant p
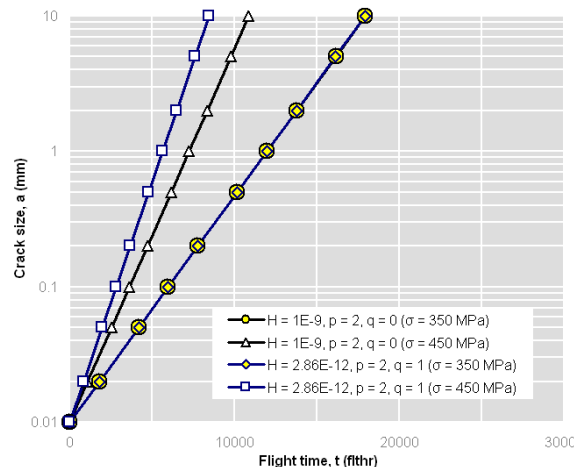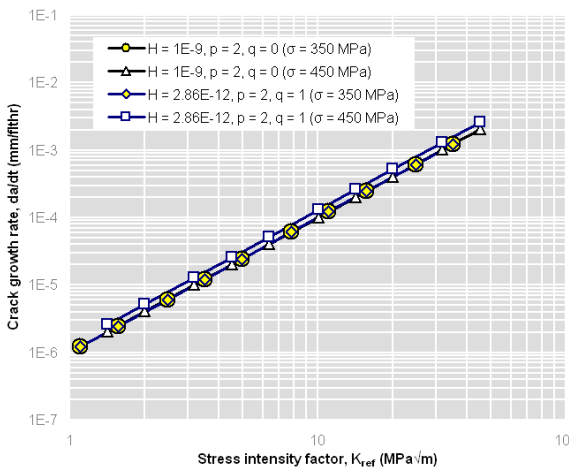
(c) varying constant q

Figure 2: Effect of varying the constants H, p and q in Eqn (3) about a baseline example of: H = 1E-9 flthr⁻¹ ·MPa⁻², p = 2, q = 0.

5. Implementation

The first step to implementing an EBA is to determine if it will fit the CGR data of interest accurately enough to be fit-for-purpose. Comparisons against existing validation data may be useful for this purpose, as long as the data is *representative* of the component of interest, including (but not limited to): material, surface finish, geometry, stress level, crack size range and VA loading sequence. It is not necessary to implement an EBA to determine if it is suitable. Rather, simply plot existing CGR data against the model's correlating parameter (usually stress intensity factor). If traditional model predictions are available then these can also be plotted. An example is given in Figure 3.
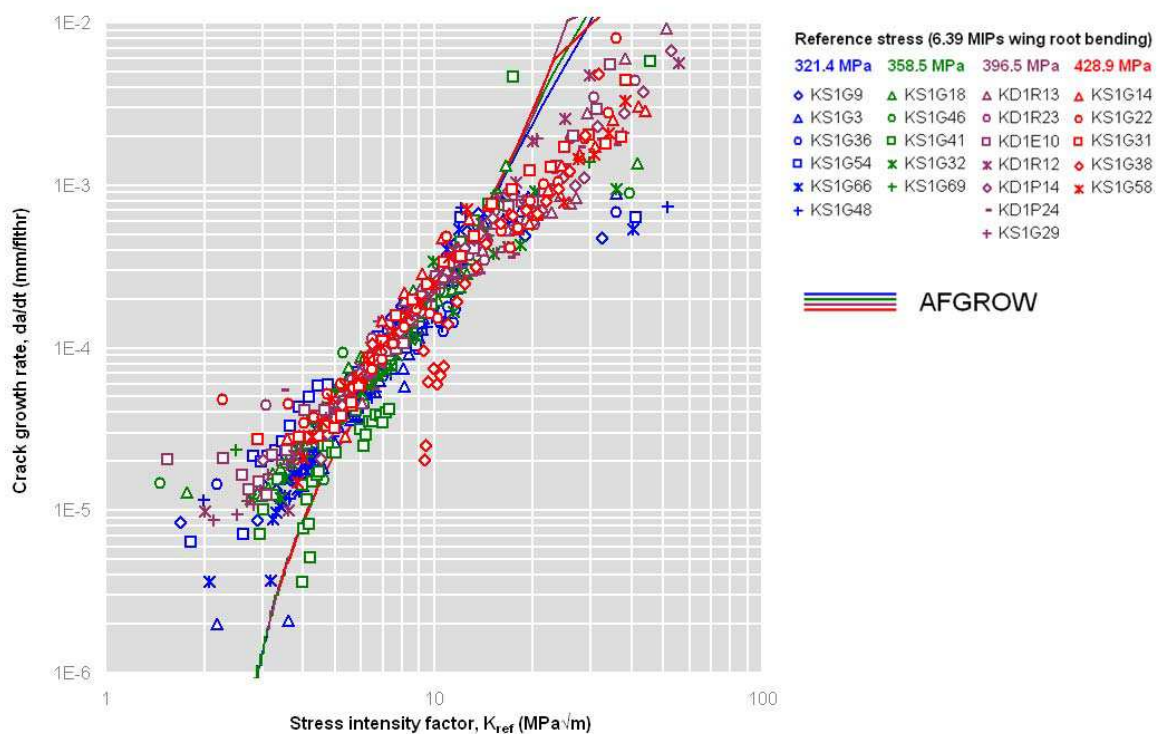
Figure 3: Example comparison of CGR validation data against a cycle-by-cycle (AFGROW) model

This figure shows F/A-18 AA7050-T7451 centre barrel validation data compared to an AFGROW model that was being considered as a lifing model [6]. The validation CGR data covers the full range of life, *i.e.*, CGR data ranges from the initiating feature through to final fracture. In this example it can be seen that the model predicts reasonably accurate CGRs for an intermediate range of K_{ref} values, but tends to under-predict the CGRs for lower K_{ref} and over-predict the CGRs for higher K_{ref} values. This model was considered to be unsuitable for the purpose of modelling the total life, since most of the total life is spent at relatively low K_{ref} values, and in this case the model had a tendency for significant under-prediction.

Casual observation indicates that a simple linear trend could be fitted to the validation data, and thus it appears to be suitable for an EBA. The following sections step through the basic processes of data preparation, model selection, parameter fitting, and using the model to calculate CG curves.

5.1 Data reduction

The preferred method for obtaining the CGR data is by **quantitative fractography** (QF). This method is well suited to small cracks where other measurement methods (*e.g.*, in-situ surface optical, direct current potential drop, *etc*) tend to have difficulty measuring accurate crack sizes at that scale. The **correlating parameter** data (*e.g.*, K_{ref}) can be readily calculated using a beta solution for the coupon specimen and Eqn (4).

The **crack growth rate** values can be obtained by calculating the local slopes of the crack growth data. Several methods are available - Figure 4 shows a comparison of the **secant** method [49], **2nd-order polynomial** method [49] and an **exponential** method (developed at DSTO) applied to an example crack data point and its neighbours. In Figure 4 the derivative, *i.e.*, the CGR, is calculated for the middle point. Note that for the secant and polynomial methods, the CGR values and associated K values may not necessarily match that of the actual point of interest. For example, the secant method associates the values for a position midway between the point of interest and one of its neighbours (in this case the point ahead). By contrast, the exponential method always produced values at the point of interest. Further details on the exponential method are given in Appendix A.

The 2nd-order polynomial and the exponential methods are preferred as they account for points either side of the middle point, which tends to increase accuracy and also provides a smoothing effect. The 2nd-order polynomial method is well suited to a wider range of crack growth trends, including non-exponential trends. The exponential method is very stable as it inherently presumes the CGR is exponential, thus it is well-suited to 'noisy' crack growth data. It also usually calculates a more accurate CGR value for end-points, *i.e.*, where one of the neighbouring points is absent. This is demonstrated in Figure 5 where the right neighbouring point is removed: comparing Figure 5b with Figure 4b shows that the exponential method produces a CGR of near equal value in this example, whereas the secant and 2nd-order polynomial methods estimate significantly different CGR values when one of the neighbouring points is missing.

(a)   *(b)*
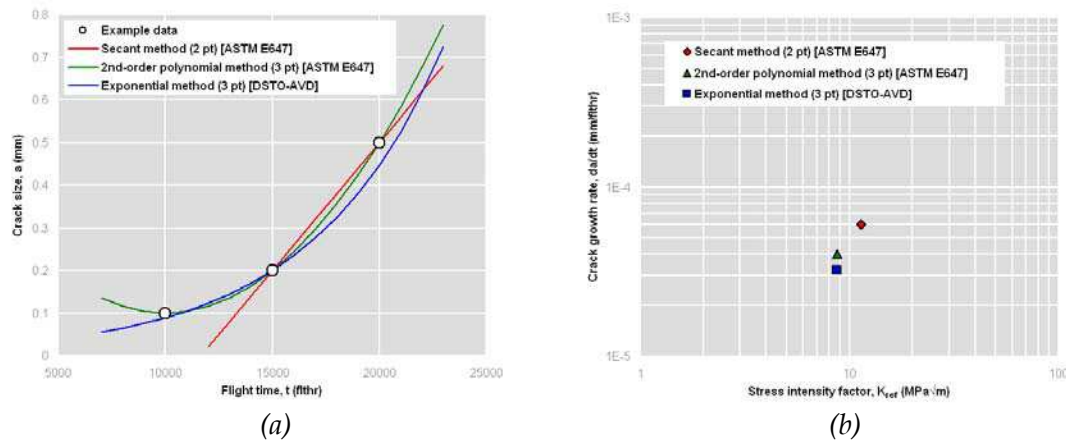
Figure 4: Example data reduction comparing several methods of calculating da/dt values: (a) input crack size data and (b) resulting da/dt values for the middle (15000 flthr) point

(a)   *(b)*
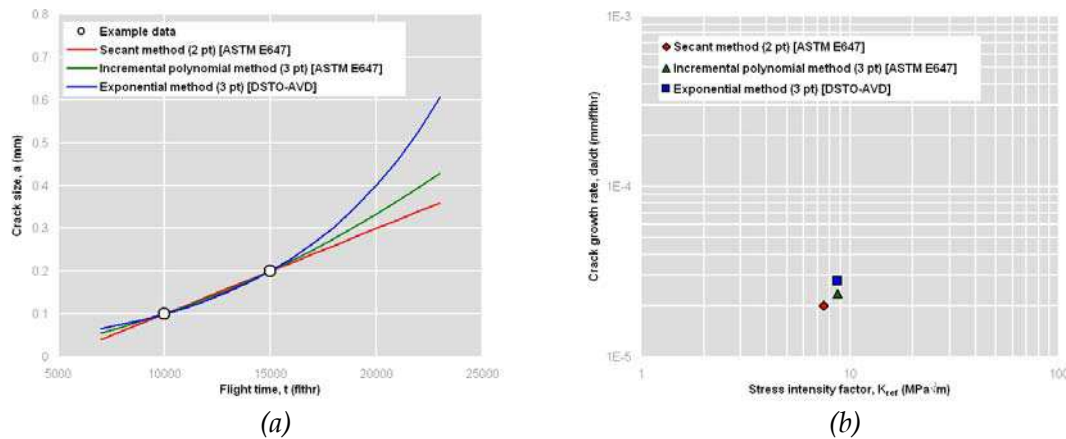
Figure 5: Example data reduction comparing several methods of calculating da/dt values at an end point: (a) input crack size data and (b) resulting da/dt values for the middle (15000 flthr) point

5.2 Reference condition

It is useful to define a **reference condition** for the load spectrum. While this is not necessary, it becomes highly recommended when two or more different load spectra are to be compared.

For aircraft, the reference condition is usually a design load, a test condition or a point-in-the-sky that has already been defined to help assess the structural area of interest. For example: the F/A-18 Hornet centre barrel program used a wing bending moment design reference; the F-111 wing recovery program used a CPLT condition. The reference condition should be stated with all quoted constants to help prevent inadvertent misuse.

To help explain the benefit of using a reference condition, consider the two aircraft load histories in Figure 6, showing typical slight differences in flight profiles. The model CGR parameters could be fitted with respect to the peak load events, resulting in one set of parameters associated with a 3 g event (Aircraft 1), and another set associated with a 3.5 g event (Aircraft 2). Subsequent CG curve calculations and individual spectrum severity comparisons would need to carry on consistent use of these individual-aircraft peak conditions. The alternative is to define a reference condition, in this case a 1 g condition, and then fit the model CGR parameters to this condition. This way, the model parameters and the relative CGRs may be easier to compare.
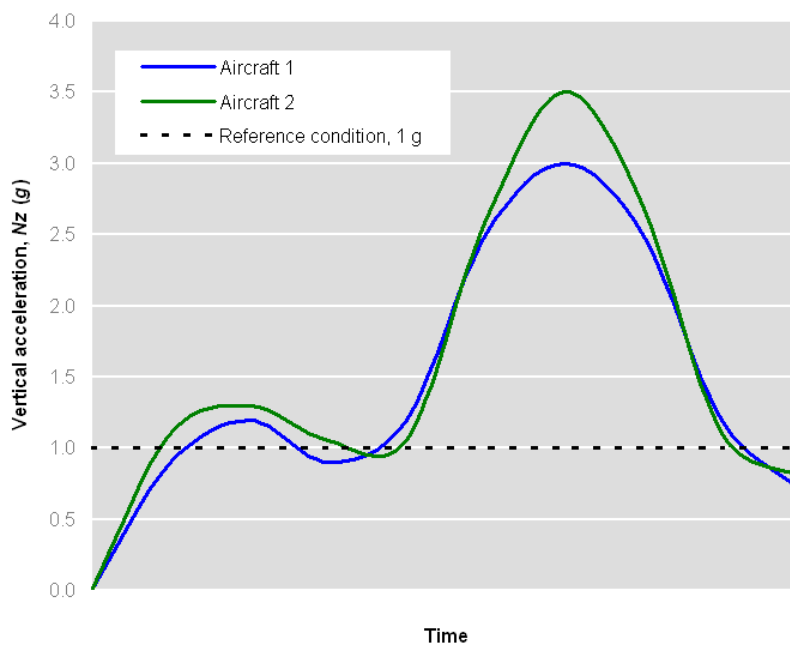
Figure 6: Example schematic of two aircraft load histories compared by a common defined reference condition

5.3 Model selection

The EBA can use any type of model that suits a particular problem or project, *i.e.*, it is not limited to the particular models described in this report. This choice between the models presented in this guide is not an important up-front decision, since the parameter fitting and crack growth calculation tools recommended herein are all based on the *General* model. As the name suggests, this means it can replicate all the other models described herein, thus making it simple to switch from one to another.

Model selection usually depends on the type of CG scenario, and also the broader analysis methodology used for the aircraft. In general, for short cracks (critical size < ~10 mm)

where there is no significant load path changes as the crack grows, then a model based simply on the crack size and stress may be sufficient. This may also allow the use of a closed-form solution – offering considerable productivity advantages. For other more complex cases where, for example: the critical crack size may be long compared to the surrounding geometry; load path changes become more significant; crack shape transitions occurs; or different geometries are being compared, then a model based on K may be more suitable.

5.4 Parameter fitting

The model parameters can be fitted using a least-squares regression [31, 33], which is widely available in most data-fitting software. Equations (1), (2), (3) and (5) can be re-arranged into a linear form to facilitate linear least-squares regression, as given in Appendix B. While the least-squares regression is simple to implement, it effectively treats each *da/dt* point with equal importance, or weighting. This might not be appropriate in some situations. For example, when total life is considered, the lower *da/dt* values contribute a more significant proportion to the total life, and hence it is reasonable that they ought to have more influence over the model's fit. In this case it may be more appropriate to fit the model parameters using a method that gives more weighting to lower *da/dt* data[13].

A parameter fitting template (Microsoft Excel) has been developed at DSTO, which provides least-squares-regressed parameters for the *General* model. Parameters for all the other models described in this guide are also derived by this template. The parameter relationships for each model against the *General* model are given in Table 1.

Table 1: Relationship between the parameters of the General model and other model types

Model	Parameter relationship with the *General* model, Eqn (3)	Conditions required
Tomkins, Eqn (1)	$A = H$ $j = p\,/\,2$ $k = p + q$	$\beta = 1/\sqrt{\pi}$ $\sigma_{ref} = \sigma_{ref\_net\text{-}section}$ in Eqn (3) and (4)
Frost & Dugdale, Eqn (2)	$\lambda = H$ $\alpha = q + 2$	$p = 2$ $\beta = 1/\sqrt{\pi}$ $\sigma_{ref} = \sigma_{ref\_net\text{-}section}$ in Eqn (3) and (4)
Paris, Eqn (5)	$C = H$ $m = p$	$q = 0$

Figure 7 to Figure 10 show the fits of selected CGR models against the example F/A-18 data. Despite the various model types used, each fit the data set reasonably well - this might not be the case for other data sets.

---

[13] No further progress has been made at DSTO to pursue this - however, it is noted as a future recommendation.

Table 2 compares the fitted parameter values. Also included are the fitted coefficient (A, λ, H) values when one of the exponent parameters is rounded or fixed, which can result in an apparently significant change in the coefficient values. However, this does not necessarily affect the crack growth prediction significantly because the equations are self-balanced by the least-squares regression fit. It is important then to realise that, firstly, the coefficient and exponent parameters are inextricably paired to give the resultant CGR relationship. Secondly, it is important when comparing two different parameter sets (perhaps for several different VA load spectra), that the differences in the coefficient values alone may not represent the severity of the CGR behaviour *unless* the exponent values are equal.

It can also be seen in Table 2 that the Tomkins model can be used to replicate the Frost & Dugdale model such that $A = \lambda$ and $k = \alpha$ when $j = 1$. Also, the General model is shown here to replicate the Frost & Dugdale model ($H = \lambda$ and $p + q = \alpha$ when $p = 2$ and $\beta = \pi^{-1/2}$) and the Paris model ($H = C$ and $p = m$ when $q = 0$).

Table 2: Comparison of the least-squares-regressed parameters for each crack growth rate model

Model type	Parameters (units)			Plot
Tomkins All variable $j = 1$	$A = 5.73\text{E-}13$ (flthr^{-1}.metre$^{-0.12}$.MPa$^{-2.82}$) $A = 4.96\text{E-}12$ (flthr^{-1}.MPa$^{-2.82}$)	$j = 1.12$ $j = 1$	$k = 2.82$ $k = 3.07$	Figure 7 *
Frost & Dugdale All variable $\alpha = 3$	$\lambda = 4.96\text{E-}12$ (flthr^{-1}.MPa$^{-2.82}$) $\lambda = 7.47\text{E-}12$ (flthr^{-1}.MPa^{-3})		$\alpha = 3.07$ $\alpha = 3$	* Figure 8
General All variable $p = 2$ $q = 0$	$H = 2.47\text{E-}11$ (flthr^{-1}.metre$^{-0.08}$.MPa$^{-2.82}$) $H = 4.96\text{E-}11$ (flthr^{-1}.MPa$^{-3.07}$) with $\beta = \pi^{-1/2}$ $H = 1.19\text{E-}11$ (flthr^{-1}.metre$^{-0.09}$.MPa$^{-2.19}$)	$p = 2.16$ $p = 2$ $p = 2.19$	$q = 0.67$ $q = 1.07$ $q = 0$	Figure 9 * *
Paris All variable $m = 2$	$C = 1.19\text{E-}09$ (flthr^{-1}.metre$^{-0.09}$.MPa$^{-2.19}$) $C = 1.76\text{E-}09$ (flthr^{-1}.MPa^{-2})	$m = 2.19$ $m = 2$		* Figure 10

* figure not shown as it looks similar
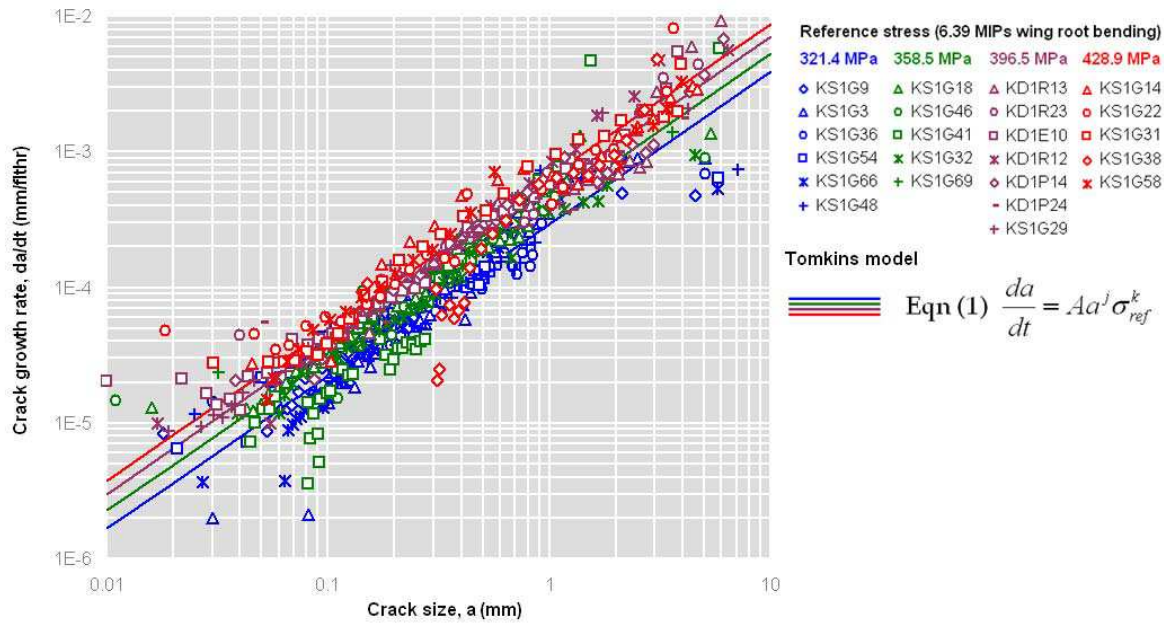
Figure 7: Tomkins model fit of the example F/A-18 data.
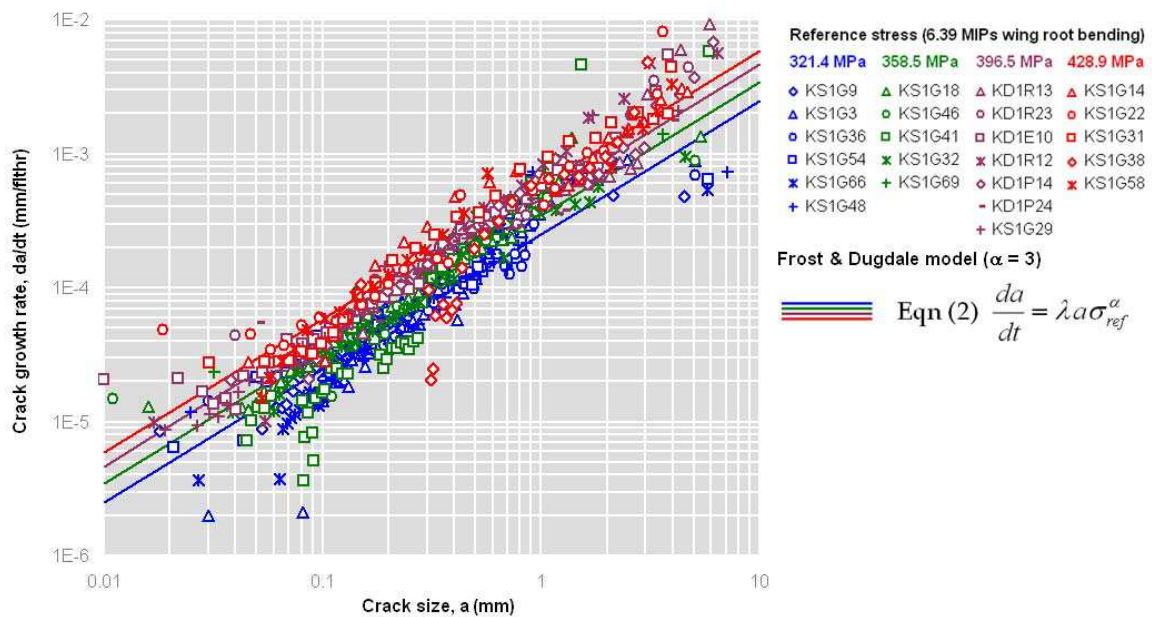
Figure 8: Frost & Dugdale model (with α fixed at 3) fit of the example F/A-18 data
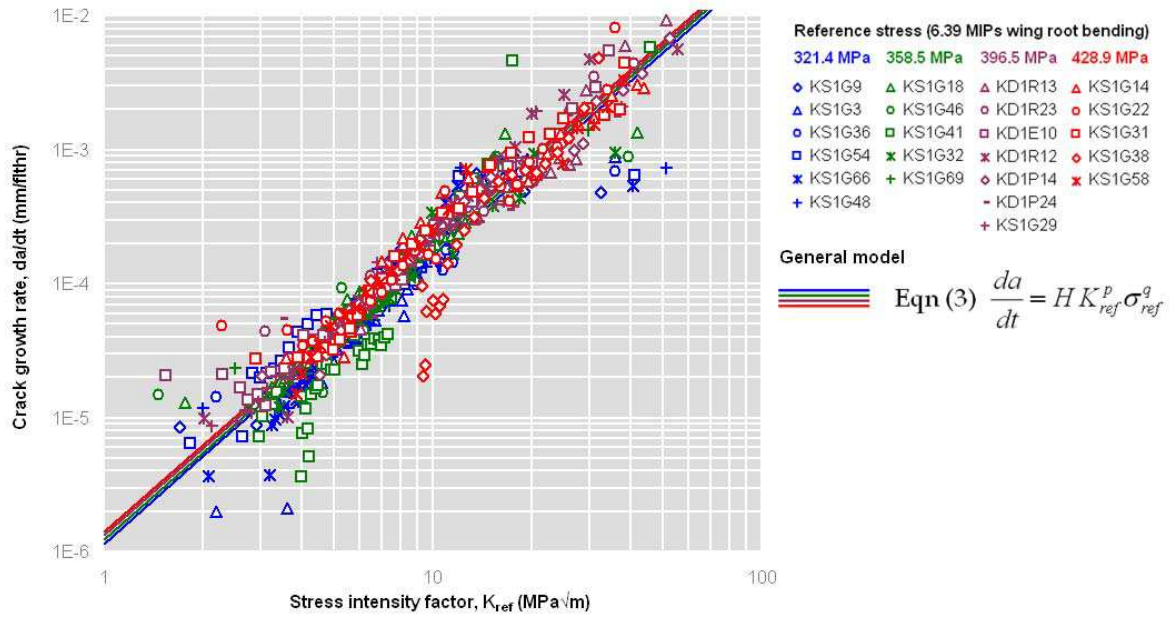
Figure 9: General model fit of the example F/A-18 data
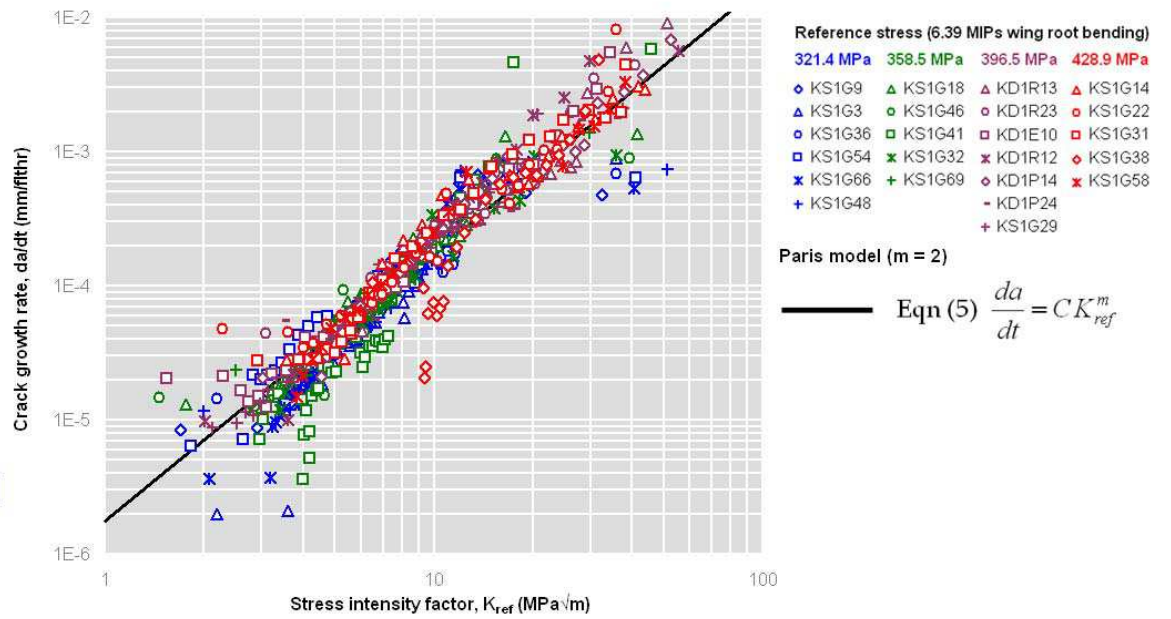
Figure 10: Paris model (with m fixed at 2) fit of the example F/A-18 data

5.5 Calculating crack growth curves

Calculating the CG curve using the model and its parameters is determined by integrating the CGR equation. This can sometimes be achieved by closed form solutions, as given in Appendix B. However, as usual problems tend to be more complex (*e.g.,* the geometry solution β usually varies with crack size), then a numerical solution is required. A numerical solution has been developed at DSTO, which has been implemented in two software codes: Matlab (Appendix C) and Microsoft Office Excel (Appendix D). These two codes are very similar in structure and produce the same accurate results; however, they do have a few implementation and output differences, as shown in Table 3.

Table 3: Summary of Matlab and Microsoft Office Excel implementations of the crack growth rate numerical integration function

Feature	Matlab	Microsoft Office Excel
Function name	IntegrateGeneralCGM() *(CGM = Crack Growth Module)*	IntegrateGeneral()
Input values and descriptions	See Appendix C	See Appendix D
Input beta solution format	Piece-wise linear segments	Piece-wise linear segments
Input stress ratio solution format	Piece-wise linear segments	Piece-wise linear segments
Integration method	adaptive Lobatto quadrature (Matlab function quadl)	Bi-section
Calculation speed	Very fast	Fast
Accuracy	Generally < 0.1%	Generally < 0.1%
Supported start points	Either of (user-requested): Initial crack size OR Initial stress intensity factor	Initial crack size
Supported stop points	Either of (user-requested): Final crack size OR Final stress intensity factor OR Time difference	Either of (user-requested): Final crack size OR Final stress intensity factor OR Time difference
Supported outputs	A single 3-value array of: Crack size AND Stress intensity factor AND Time difference	A single value of either (user requested): Final crack size OR Final stress intensity factor OR Time difference
Input checks and warning messages	Yes, all inputs	Yes, all inputs
Installation	Read in the module file *IntegrateGeneralCGM.m*	Copy an existing workbook that already contains the module, or copy the Visual Basic module between workbooks.
Help file	Basic input help is contained in the header comments in the module file.	Basic input help is available when using the Excel *Insert/Function* menu only. (IntegrateGeneral will be found under the *User Defined* group)
Known issues:	May abort and return an error message if the stress intensity factor solution decreases with crack size (likely if the Kt is high)	May abort and return an error message if the stress intensity factor solution decreases with crack size (likely if the Kt is high)
	Does not support Matlab array operations	May return a "Program execution " error message if any of the variables exceed their numerical precision limits (double precision)

5.6 Calculating the equivalent initial crack size

The **equivalent initial crack size (EICS)** is defined here simply as **the size of an initial (flight time = 0) crack that gives a desired CG life**. This seemingly ambiguous definition is necessary to cover the wide variety of methods that could be adopted for platform life management. For example, the F-111 wing test interpretation program calculated EICS values based on a standardised CGR model to meet the aim of a deterministic lifing assessment framework where life analyses are based on pragmatic nominal conditions [50]. In other research on characterising material discontinuities, the EICS values were determined using unique CGR parameter fits to individual test specimens, in order to provide results that more closely characterised the physical micro-structural sizes [19].

The EICS value can be calculated based on any chosen CGR model. It can be expected that if a different CGR model is used, then the derived EICS value would also differ. It can be important then to quote the EICS result along with the model used to derive it. Figure 11 shows a number of EICS values calculated for a single F/A-18 specimen chosen from the example data set[14]. The EICS values derived here range from about 0.008 to 0.03 mm; *i.e.*, almost an order of magnitude. In this example the main difference is due to the choice of fit, *i.e.*, pass through the final-fracture point or pass through a best-fit[15] of all CG data. There is little difference between the EICS values derived by the Paris and Frost & Dugdale type models in this case because both these models fit the average CGR behaviour of our example data set similarly well - however, this may not always occur for other data sets.

---

[14] Note that the CG curve shape of this particular specimen deviates somewhat from the general trend represented by the models, which could be due to some difference between this particular specimen and the general trend of all the specimens (*e.g.* a different crack shape aspect ratio).

[15] The best-fit was calculated by a least-squares regression in the time domain.
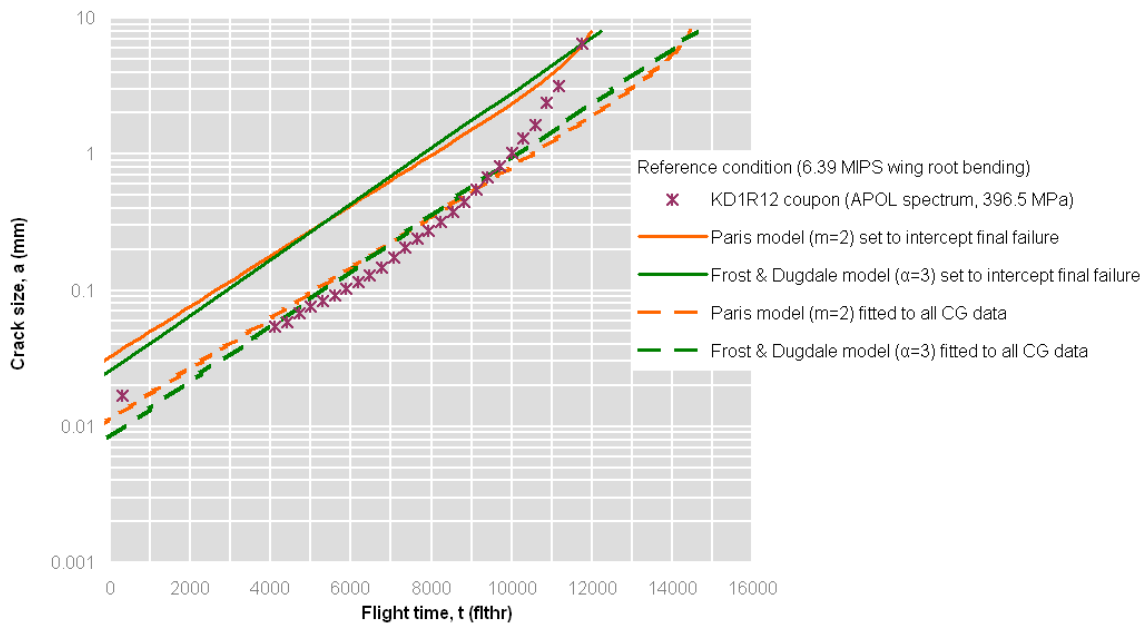
Figure 11: Determination of EICS values for specimen KD1R12 using two CGR model types (Paris and Frost & Dugdale) and two alternate fitting methods (i.e., pass through the final fracture, or provide a best fit through all CG data).

Note that the EICS will be affected by idealisations of the CG modelling approach. For example, many aircraft structural integrity programs idealise the shape (aspect ratio) of the crack as a single semi-circular surface crack, or a single radial corner crack. This was done in the example data set here. However, the actual cracking may include multiple cracks of different shapes that can behave relatively independently, or coalesce into a reducing-number of larger cracks with different shapes. Taking into account these more complex aspects that may be important for research purposes or for more detail comparisons between disparate data sets. Generally though, these may be pragmatically nominalised and incorporated into EICS values, hence the term *equivalent*.

Surface conditions and fatigue enhancements (*e.g.,* peening) treatments may affect the EICS values. Other conditions that may affect or be inherited by the EICS values include:

- material
- residual stress effects (material manufacture process, assembly fit-up stresses, cold-expansion, *etc*)
- notch geometry (high or low Kt)
- notch-plasticity (*e.g.,* fastener holes under high stress)
- scale of crack-tip yielding (*c.f.* small-scale yielding limits)
- material grain orientation
- stress level
- stress-state or thickness constraint (plane stress, plain strain, finite thickness)
- environment (temperature, humidity, fuel, salt fog, *etc*)

- load spectrum
- load rate (1 Hz, 20Hz)

Importantly, the CG model may attempt to account for some of these, and so the relationship between the CG model and the EICS values is inextricably linked. While it seems daunting that there can potentially be many different EICS values for many different conditions, in most applications where representative testing has been performed on a particular fatigue-critical area, this can usually be reduced down to a change in a couple of variables, *e.g.,* load spectrum and stress level.

Plotting populations of EICS values against various conditions can be useful to determine if it is appropriate to transgress lifing calculations across these conditions. Examples are shown later in the case studies (Section 6). While it might be more obvious that the EICS values will differ for different surface conditions or fatigue enhancements, it might not be so obvious if a different spectrum or stress scale level would result in a different EICS value - perhaps due to an effect of the material microstructure. This is an important consideration if the derived EICS value is then subsequently used to make life prediction for a different condition.

5.7 Reverse calculation of beta solutions

The Paris-type model and stress intensity factor relation can be re-arranged such that the geometry beta solution for a particular feature in a structure can be estimated from the measured CGR data at that location, as follows:

Combining (4) and (5)
$$\beta = \frac{1}{\sigma_{ref}\sqrt{\pi a}}\left(\frac{1}{C}\frac{da}{dt}\right)^{\frac{1}{m}}$$
(6)

where the *C* and *m* constants are determined from coupon tests, and the CGR data is from the feature of interest in a structure. Numerous CGR measurements over a range of crack sizes can then be used to construct an estimate of the beta solution at the feature.

Usually the beta factor is considered to be strictly a linear-elastic quantity representing only geometry and boundary condition effects, however, here it is considered to take on other effects as bulleted earlier in Sections 3 and 5.6, which may include significant non-linear effects (*e.g.,* notch plasticity, interference-fit contact). Comparisons of this CGR-derived beta against theoretical linear elastic solutions may give a useful measure of how complex a feature is behaving in reality. This may be pertinent for the validation of a traditional mechanistic model. A case study example is given in Section 6.4.4.

5.8 Supplementary adjustments for different load spectra

Supplementary adjustments may be required for variables that are not present in the CGR models. A list of potential considerations was given earlier in Section 3. The most obvious for the EBA is that, unlike traditional cycle-by-cycle or mechanistic models, the EBA has no predictive ability by itself to account for the effects of *different* VA load sequences. However, other comparative data or analysis can be used to determine load spectra effects. Such data may be from other testing, or the predictions from an individual aircraft tracking lifing model, *e.g.,* a model that has been specifically calibrated to provide good *relative* life predictions.

An EBA can be used to help assess and interpret load spectra effects by plotting the data in a block CGR format, which can be more insightful than simply comparing CG curves. Figure 12 shows an example of some test and analysis data for two different aircraft wing bending moment histories. The CGR data indicates that the difference between the spectra may be characterised by a vertical shift in the CGR from one spectrum to another – this is equivalent to scaling the CGR coefficient parameters, *e.g.,* the *C* parameter of the Paris-type model. Therefore, a spectrum scaling method is required for the EBA, in order to make predictions based other comparative data or analyses.

Two scaling methods are described in Appendix E. The first and simplest method scales CGRs based on a constant relationship between the CGRs for two spectra over the range of the similitude parameter. This is analogous to the linear scaling of initiation or total life results. Examples of this method are given in Figure 12 and in the F/A-18 case study (Section 6.3.4.). The second and more complex method is based on a linearly varying relationship between the CGRs for two spectra over the range of the similitude parameter. This can be used to account for relatively simple trends that may be observed against the similitude parameter. The F-111 wing test case study provides an example of its use (see Section 6.4.3). Some good critical reviews of these two scaling methods are also given in [6, 7, 38].

Figure 12 shows an example where the CGR fit to the test results of one aircraft history (A21-055) is scaled to make a prediction for another (A21-103). The scale factor was calculated using the first method in Appendix E. The factor was based on the relative difference between CGRs estimated by an uncalibrated AFGROW model. The results of the model are plotted on same block CGR chart with the test data to check the complexity of the relative CGR trend – in this case there was a relatively steady vertical shift in the CGR for the two spectra over the range of stress intensity factors. This indicates that that a simple scaling method may be appropriate. In this example the scaled result (dotted blue line) compares well to the best-fit result (solid blue line). Further details of this scaling example is given in [6]. Note that the uncalibrated AFGROW model provided considerably poor predictions of the general CGR behaviour (underestimating for smaller K, and over-estimating for larger K), yet provide a reasonable prediction or the relative difference between the two spectra.
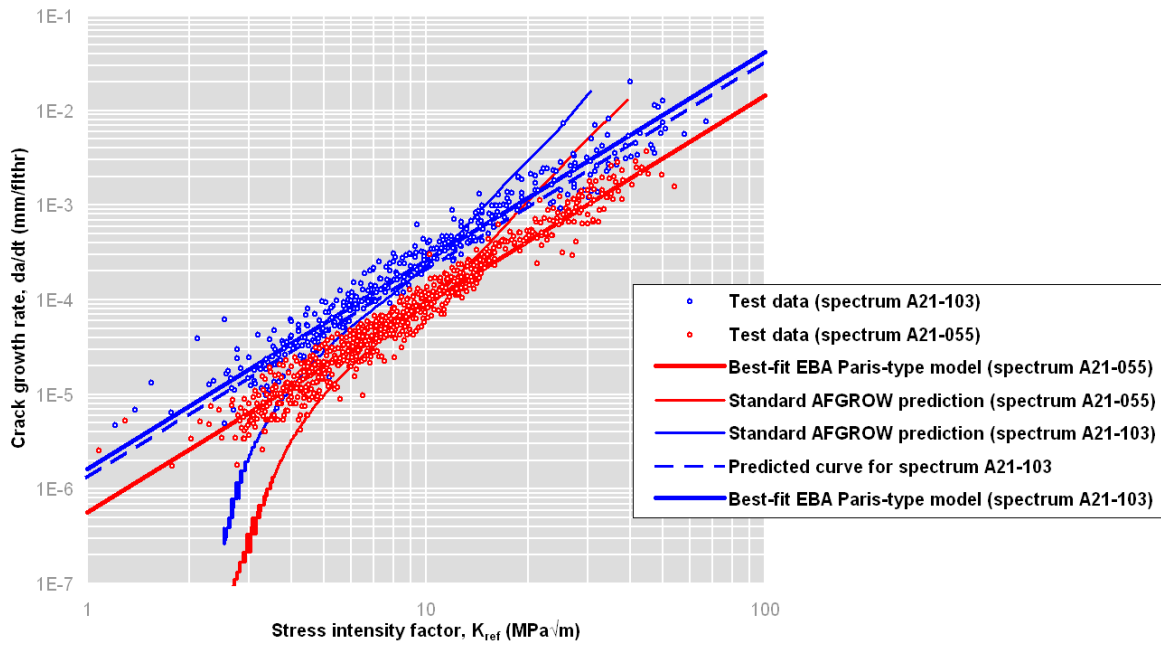
Figure 12: Comparison of CGRs for two aircraft load histories, showing an example of an EBA fit to one (red) can be scaled, using the results of an AFGROW analysis, to predict the CGR for another (blue) spectrum.

6. Case Studies

6.1 Macchi Accident Investigation

The Macchi accident is a pointed example where the crack growth rate knowledge was used to identify the cause of failure and provide ASI management options for the fleet [8, 51]. It is noteworthy that this process specifically did not attempt to match the failure observations using a mechanistic CG model; rather a simple empirical model was developed based directly on the measured crack growth rates and an assumption of similar usage[16] on-average over the aircraft's life. The particular model used in that work was:

$$\text{“crack depth} = A \text{ . } \exp(B \text{ . } time)\text{”}$$

which is equivalent to the herein so-called EBA with a model based on a simple crack size proportionality relationship, as given by Eqn (2). In effect, the entire traditional CG modelling process was completely bypassed, and yet satisfactorily provided the RAAF with an effective management tool.

Significant observations from the Macchi accident investigation were:

- crack growth was observed to proceed without a significant 'initiation' phase

- the importance of the initial defect size in determining the 'safe' life

- CGRs were relatively insensitive to variations in individual aircraft load histories

These are important because they each questioned aspects of the original safe-life 'initiation' philosophy that the aircraft was designed to.

6.2 F/A-18 Horizontal Tail Spindle Fatigue Test Interpretation

The test interpretation of induced damage-tolerant fatigue cracking in the F/A-18 aft fuselage horizontal tail spindle required the consideration of the applied VA test spectrum at different stress scale levels. AF1410 steel coupon data at a number of stress levels were available however, in order to account for stress distribution (Kt) differences between these and the full-scale article, an LEFM-based approach was taken using the *stress intensity factor* as the correlating parameter [2].

---

[16] Variations in fleet usage were accounted for empirically, by simply manipulating a parameter of the model.

Figure 13: Location of the horizontal tail spindle in the F/A-18 aft fuselage

Firstly, a traditional (AFGROW) CA-based retardation calibration effort was initially undertaken, trialling three different CA-based retardation models: shown in Figure 14. The closure retardation model appears to have performed the best out of the traditional models. An EBA Paris type model is also shown for comparison. It can be seen that the CA-based closure retardation model, and the EBA Paris type model, tended to under-predict the highest (1076 MPa net) stress coupon data, most likely because the material yield was exceeded in the notch region in this case.

The EBA was finally selected as the CG model for this test interpretation based on its accuracy, efficiency, and relative steadiness at larger crack sizes and stress levels: this steadiness was an important deciding factor since the calculation of inspection intervals were being considered at the time. It can be seen in Figure 14 that the closure model tended to produce unsteady CG at the longer crack sizes at the higher stress levels.

DSTO-TR-2850

*(a) AFGROW, **No** retardation*
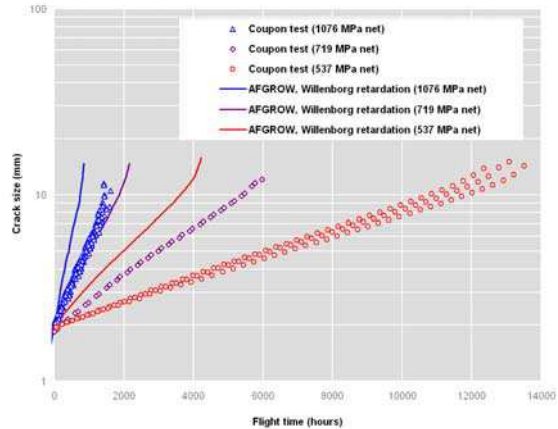
*(b) AFGROW, **Willenborg** retardation*
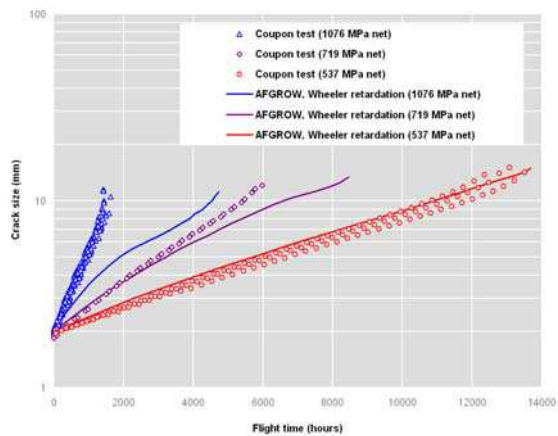
*(c) AFGROW, **Wheeler** retardation*
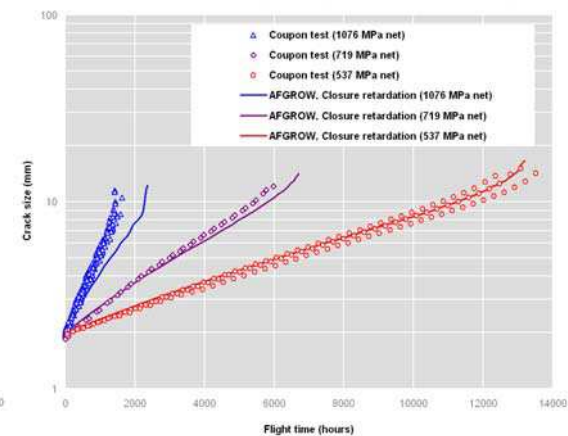
*(d) AFGROW, **Closure** retardation*
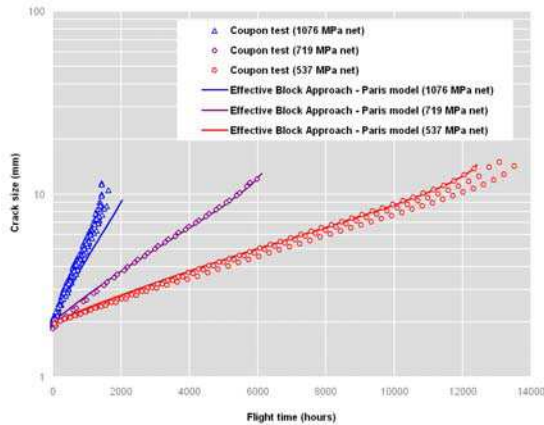
*(c) **EBA, Paris** type model*

Figure 14: Comparison of various traditional (AFGROW) CA-based retardations models, and the EBA Paris type model, against the AF1410 steel coupon test results. Each model was fitted to the low stress (537 MPa net) coupon data[17], and then used to predict the CG at the higher stress levels.

---

[17] The Willenborg model was unable to fit the coupon data, even considering the complete range of adjustment of its retardation parameter.

A CGR model for AF1410 steel was only required for one load spectrum, thus avoiding the need for any supplementary analysis that would otherwise be required to consider other spectra. A detailed finite element model (FEM) of the horizontal tail spindle was developed, which was used to identify and quantify high-stress locations, including the location where an electrical discharge machined (EDM) notch was induced. Post-test comparisons of the CG at the EDM notch provided additional substantiation of the EBA framework, as shown in Figure 14.
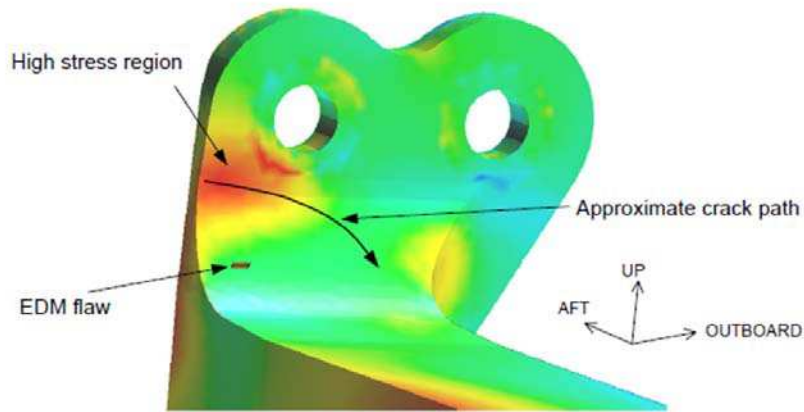
(a)
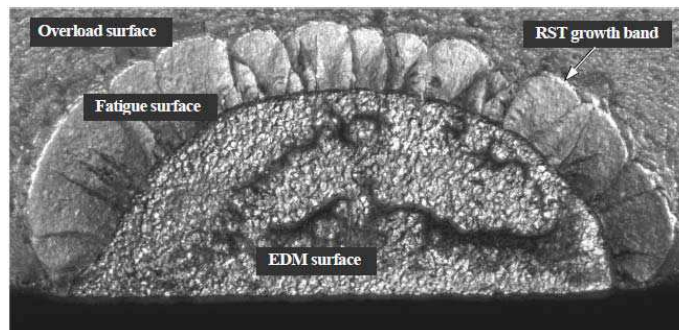
(b)
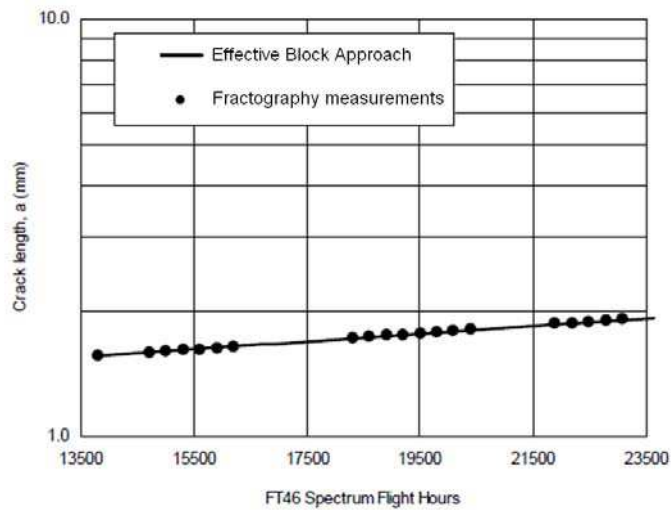
(c)

Figure 15: F/A-18 aft fuselage horizontal tail spindle: (a) view of the upper aft attachment lug showing FEM stress contours, which were used to estimate stress intensity factor distributions; (b) fracture surface at the EDM notch, and; (c) comparison of the EBA crack growth prediction against the full-scale test (FT46) quantitative fractography measurements at the EDM notch.

6.3 F/A-18 Centre-Barrel Fatigue Test Interpretation

The F/A-18 Hornet centre barrels are typified by relatively early multi-site cracking, primarily because of its optimised design in combination with a relatively uniform distribution of etch pits (crack promoters) on the aluminium alloy 7050-T7451 surfaces due to the application of the corrosion protection system [5].

A feature of the optimised design is relatively uniform stress distributions, which encourage relatively short critical crack sizes, ~4-10 mm. Therefore the behaviour of the cracking can be characterised generally by local conditions only (*e.g.,* load shedding and other long-crack effects can be ignored). This has allowed a simple crack growth model, based on *crack size* and *stress* (Eqn 2) to be used with very good characterisation of the CGRs.

Fitting EBA models, and their parameters, to the F/A-18 Hornet centre barrel data has revealed some persistent trends, namely:

1. an exponential crack-size relationship [6];

2. a stress-cubed relationship [6], and;

3. stress concentration (Kt) effects [16].

These trends are of interest because they are not readily simulated by LEFM-based CG models. It has only been through the application of an EBA that these trends can be identified: and this has lead to a re-consideration of theories about ductile or semi-ductile fracture mechanics as a significant contributing explanation of short *pro* long crack CGRs [18], as well as the development of new lifing frameworks [52, 53].

6.3.1 Exponential crack-size and stress-cubed relationships

The F/A-18 program has generated a substantial body of test data demonstrating that the CGRs of short cracks in the 7050-T7451 material (and longer cracks under prescribed conditions) can be accurately characterised by an exponential relationship with the crack size [6, 54]. Similarly, the CGRs are generally proportional to the cube of the stress level – resulting in the adoption of the *stress-cube* rule-of-thumb in DSTO and the RAAF F/A-18 Hornet lifing methodology [55]. The example data supports these trends as shown in Figure 16 and Figure 17 respectively.

These relationships also appear in earlier constant-amplitude work on a range of materials [42]. These relationships are currently thought to be explainable by ductile or semi-ductile fracture mechanics, since detailed elastic-plastic finite element modelling of crack tips appear to support both the exponential crack-size and stress-cubed relationships for low-stress-gradient (Kt) conditions [18, 56].

Based on these evidences, the centre barrel program has adopted the EBA with Eqn (2) as the underlying model for short crack and low-stress-gradient (Kt) conditions [5].
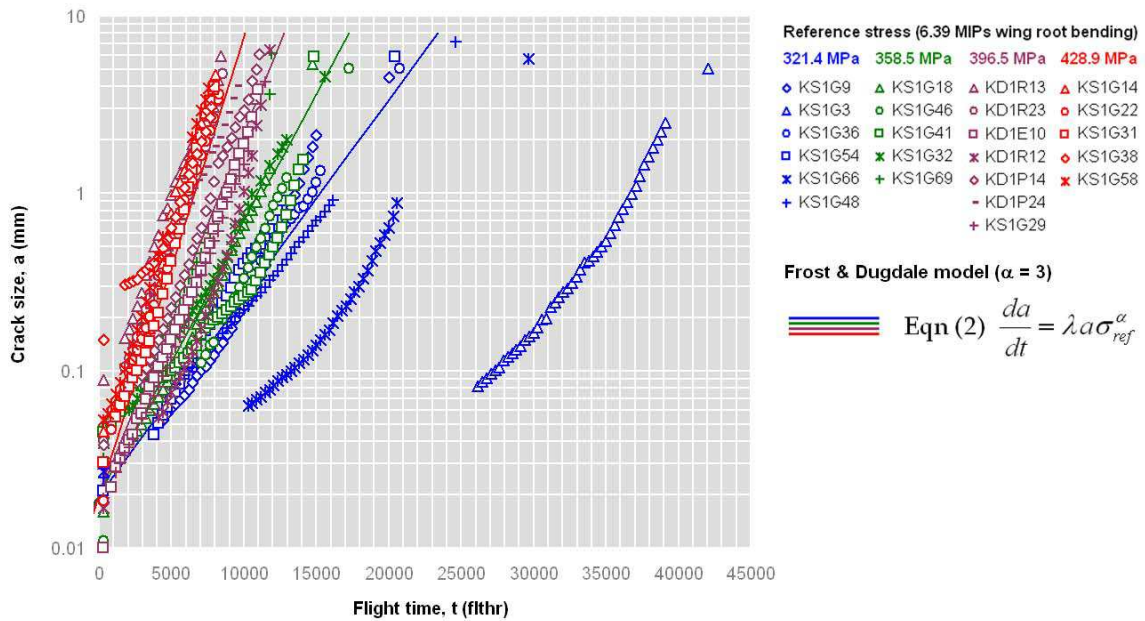
Figure 16: Example F/A-18 data showing that the crack size is generally exponential (i.e., straight crack growth behaviour on a log-linear scale) - which equates to CGR being proportional to crack size – see complementary trend in Figure 8.
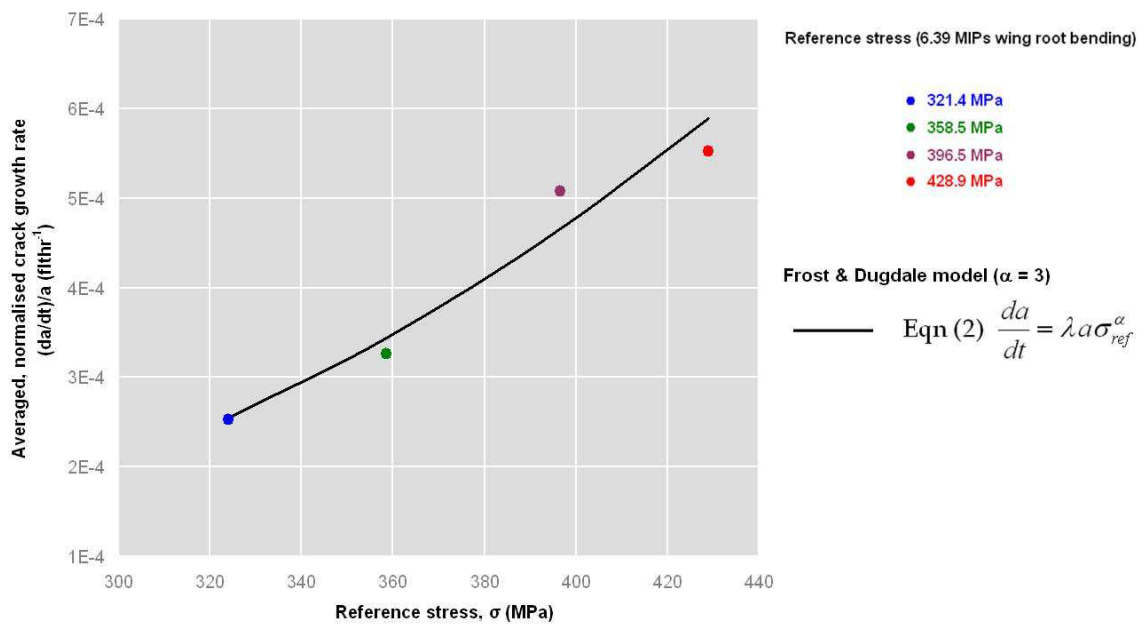
Figure 17: Example F/A-18 data showing that the averaged CGRs are generally proportional to the cube of the stress level ($\alpha = 3$).

6.3.2 Notch geometry effect - High *vs* Low Kt

Stress concentration effects, as measured by test specimens subject to the FT55m5 fighter load spectrum, have been examined using an EBA by DSTO [16, 57]. Figure 18 shows the low and high Kt specimen shapes along with the associated LEFM-based geometry factor (β) solutions. The assessments concluded that a simple LEFM-based EBA model (*i.e.*, Paris-type model) was unable to satisfactorily fit both the Low and High Kt data sets, as indicated by the two separate lines of best fit in Figure 18(c). The test results reveal that the Low-Kt CGRs were relatively higher (about a factor of 4) compared to the High-Kt CGRs where the K level is lower (~2 MPa$\sqrt{m}$).

It is as-yet unclear on the cause of the discrepancy. One postulation is that it is due to a limitation of LEFM due to the high-stress conditions tested. If this is the case, then discrepancies like this may also raise questions about the reliability of traditional LEFM-based mechanistic CG models.
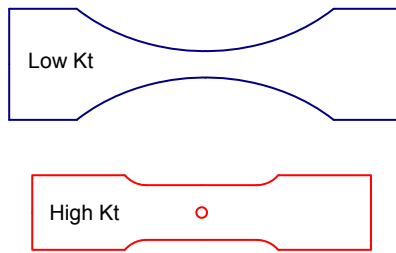
(a) specimen shapes
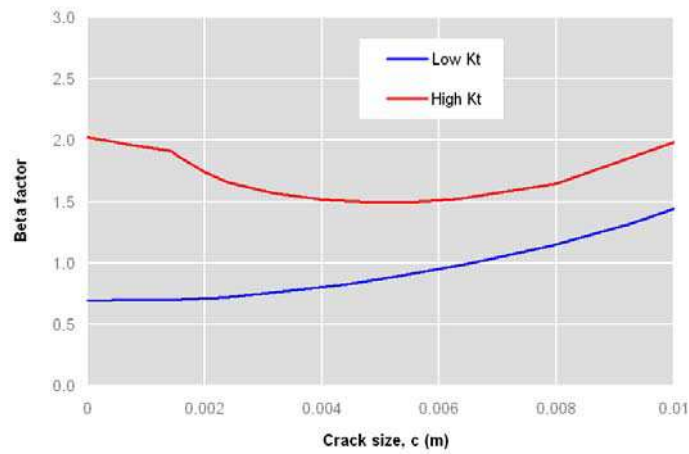
(b) beta factors
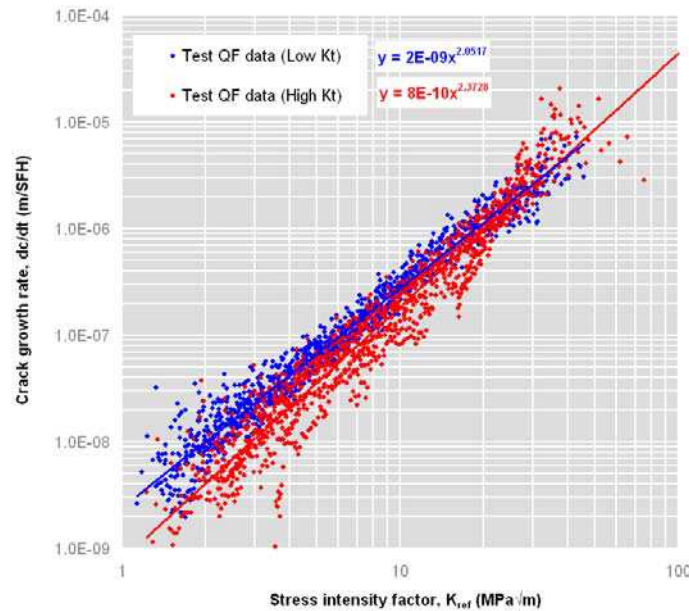
(c) crack growth test results

Figure 18: Comparison of High Kt and Low Kt test results for AA7050-T7451 specimens both tested under the F/A-18 FT55m5 variable amplitude load spectrum [57], showing that the specimen with the higher Kt growing cracks at a slower rates (for the same K magnitude), particularly for shorter cracks.

6.3.3 Back-extrapolations to time zero – Equivalent Initial Crack Sizes

Crack growth model predictions can be highly sensitive to EICS values, therefore is pertinent to determine if there are any sensitivities in using EICS values in a lifing approach. Apart from surface conditions effects, the F/A-18 program also considered CGR model sensitivities, two of which are examined here - namely that the EICS may be dependant on 1) the stress scale level and 2) the VA load spectrum.

The F/A-18 Hornet program has CG data available for a range of stress scale levels and numerous VA spectra. Figure 19 and Figure 20 show the EICS values calculated using two alternate EBA models, the Paris-type (Eqn 5) and the Frost & Dugdale-type (Eqn 2) respectively. A trend line is shown for the most comprehensively tested VA spectrum, FT55m5. Figure 19 shows that the Eqn 5-derived EICS values are sensitive to the stress scale level – suggesting that it may not be appropriate to use the Paris-type model to transgress stress scale levels. However, Figure 20 shows that the Eqn 2-derived EICS values are relatively insensitive to stress scale levels, indicating it may be appropriate to use a Frost & Dugdale model.

Although a detailed statistical analysis is not provided here, the plots also suggests that the EICS values are relatively insensitive to the VA spectrum applied, which may support the use of an EICS methodology to translate between these types of spectra.
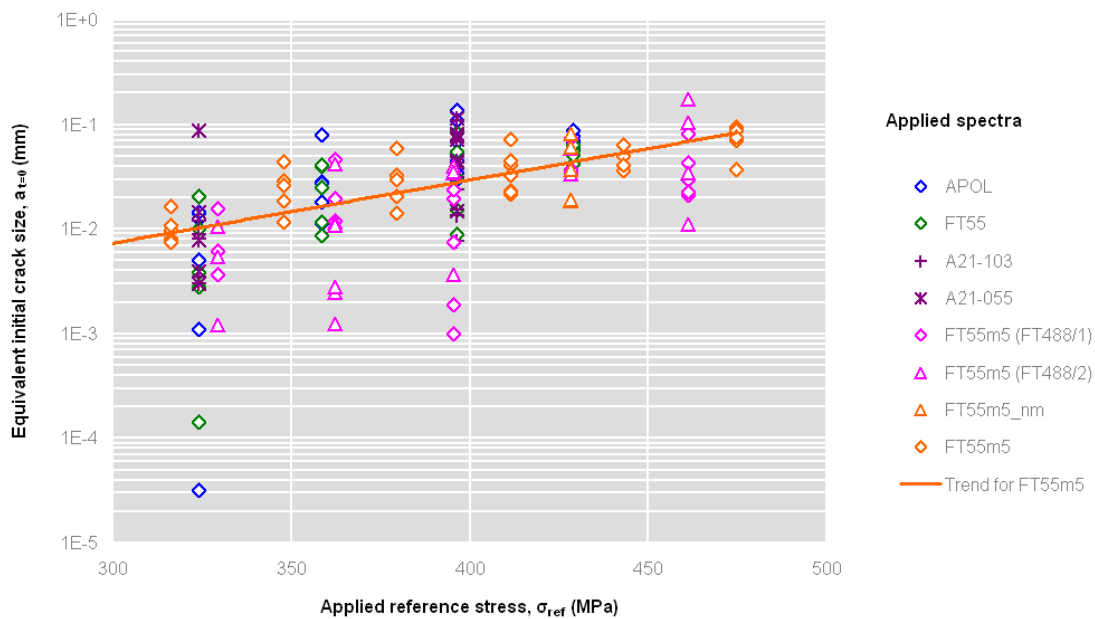
Figure 19: Equivalent initial crack sizes for F/A-18 data using a Paris type CGR model (m=2)
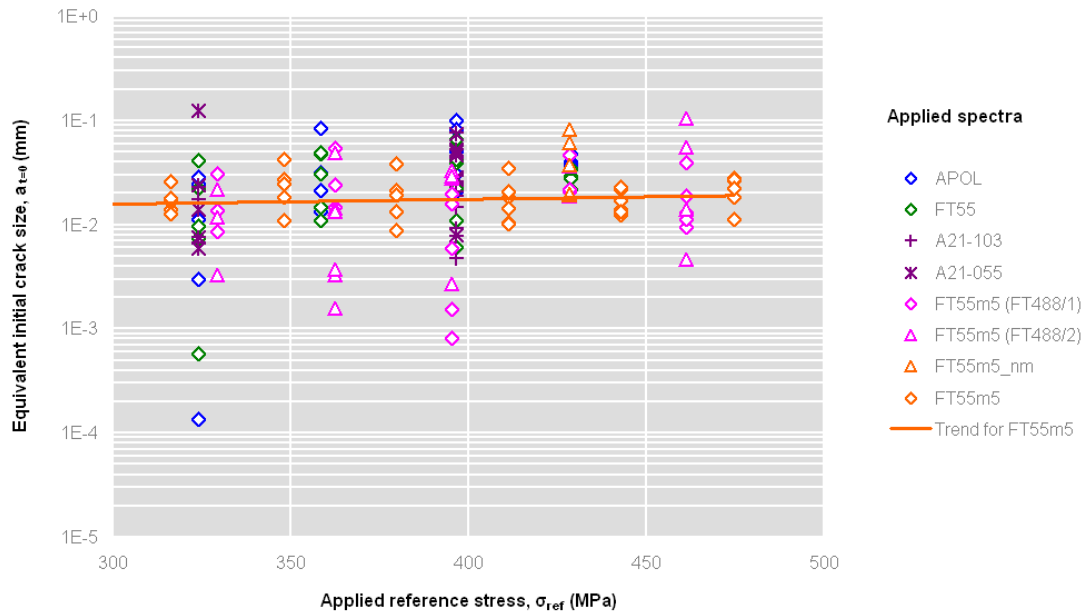
Figure 20: Equivalent initial crack sizes for F/A-18 data using a Frost & Dugdale type CGR model (α=3)

6.3.4 Spectrum severity comparisons

The use of traditional mechanistic type CG models as a means to adjust EBA model parameters in order to predict CG for different (untested) spectra was studied using F/A-18 wing root bending moment spectra [6]. This study assessed the ability of uncalibrated AFGROW and FASTRAN models to rank the severity of different wing root bending load spectra. One of the implied considerations of the study was that, while an mechanistic model may provide poor predictions of the CG curve, there is potential that it may provide useful *relative* spectra severity predictions.

EBA model parameters were fitted to the five spectra: including the test data, as well as analysis predictions from both AFGROW and FASTRAN. These parameters were then compared on a correlation plot, Figure 21, where the model results are given on the vertical scale, and the respective test results on the horizontal scale. The 1:1 correlation line indicates perfect relative predictions. It can be easily seen that the AFGROW model appears to perform better than the FASTRAN model in this example.

This provides a useful example of using simple empirical model constants with an EBA to reduce quite a lot of test and analysis CG data down to the required trends of interest. Note that by selecting a correlation plot as the comparison tool, a simple scaling method is assumed, which will hide any complex CGR trends. Plotting the CGR parameter fits may provide more insight into the efficacy of AFGROW and FASTRAN predictions – and thus guide potential calibration efforts.
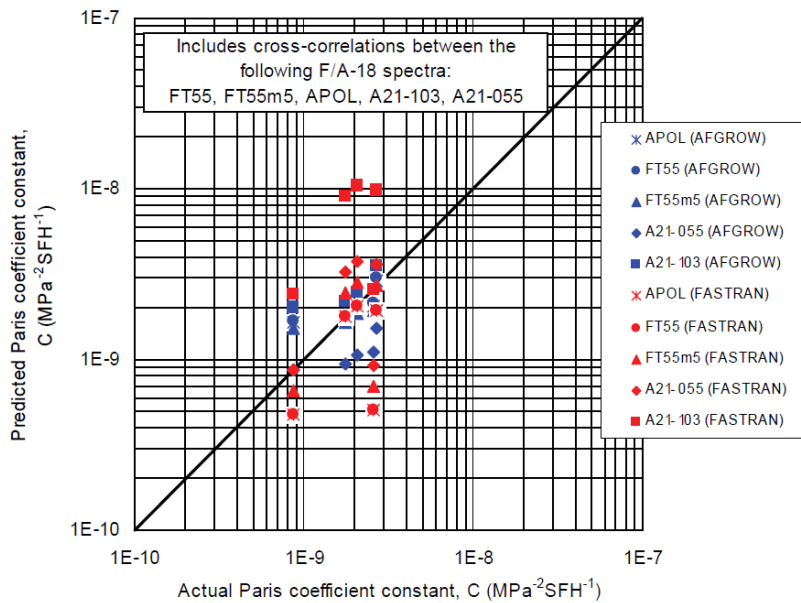
Figure 21: *Analysis ("Predicted") vs test ("Actual") correlation plot of CGRs for several F/A-18 wing root bending spectra. Here the EBA Paris-type model C parameter (with m = 2) is used to represent the CGR for each case (from [6]). The 1:1 correlation line indicates perfect relative predictions. The AFGROW model appears to perform better than the FASTRAN model in this example.*

6.4 F-111 Wing Fatigue Test Interpretation

6.4.1 Wing lifing overview

The F-111 Wing Economic Life Determination (FWELD) program was devised to assist the RAAF in managing the structural integrity of the F-111D and F model wing fleet through to the planned withdrawal date [10, 50]. The program consisted of a full-scale wing fatigue test, structural FEMs, a mechanistic-based Crack Growth Analysis Package (CGAP, incorporating the FASTRAN CG model), and a comparative coupon test program known as Load Interpretation Truncation Validation (LITV)[18] [58]. The aim was to translate the full-scale test findings, which was tested using a USAF spectrum, into life limits for estimated RAAF spectra.

The wings were subject to a regular (every 2000 hours) application of CPLT[19] to help manage fatigue cracking in the carry-through and wing root components that were made from D6ac steel. This caused a significant CG retardation effects in the lower sections of the wing, including the aluminium alloy 2024-T851 structure. This retardation effect was found to be difficult to reliably simulate using mechanistic CG models – several models

---

[18] Initially the LITV program was intended to assess load truncation effects, however, the program was significantly expanded to provide CG data for a wider range of VA load spectra.

[19] Applied at ambient temperature, not -40°C

were trialled including ADAMSys, METLIFE, and most recently FASTRAN – as evidenced by a re-occurring calibration effort that lasted well over a decade [3], and is still ongoing today for the benefit of other platforms[20] [4]. This encouraged a view in DSTO that a traditional test interpretation framework, *i.e.*, one underpinned by a validated CG model, carried a risk of being unsuccessful within the time frame of the planned withdrawal date.

To mitigate this risk, the LITV coupon test program was significantly expanded to provide the necessary CGR data for the various load spectra required for the full-scale test interpretation. Furthermore, since the spectra typically consisted of repeated blocks of loading, the EBA with a Paris-type model (Eqns 4 & 5) was considered for the CG analysis [59].

6.4.2 EBA benefits and challenges

EBA benefits

- Simple empirical models were found to be able to fit CG trends of the LITV data even though there were significant retardation effects caused by the CPLT loading sequence. The EBA was suitable only because the CPLT sequence was repeated regularly.

- Compatible with all of the RAAF F-111 lifing philosophies, *i.e.*, Safe Life (with an appropriate EICS methodology), Damage Tolerance and Safety-by-Inspection.

- Could grow cracks backwards and hence resolved test interpretation difficulties due to multiple applied spectra to the test article and a maturing non-destructive inspection (NDI) capability, *e.g.*, the EBA could readily accounted for:
  o the pre-test in-service spectra already experienced by the full-scale test article,
  o updates to the NDI capability, *i.e.*, the maximum undetectable crack size, based on the ongoing NDI validation efforts.

- Could quickly reverse-calculate stress levels in the wing via QF data, and thus provided additional stress correlation data for the structural FEMs.

- Readily grew cracks based on multiple time-periods of different RAAF usages, *i.e.*, previous + current + future (note: this is not unique to the EBA).

- The EBA non-mechanistic model caused the promotion of a more comprehensive coupon test program, thus reducing risk compared to otherwise relying on perceived accuracy on a traditional mechanistic-type CGR models.

- Very fast analyses resulted in new benefits that have not been practical with other mechanistic-type CGR models, such as:
  o the wing test interpretation, and the life estimations for all F-111 wing sets in the fleet, could be set up on a single quality-controlled spreadsheet and run on demand with a calculation of time of less than 10 seconds,

---

[20] the RAAF F-111 fleet was retired from service circa 2010.

- o more regular, and more accurate, client reporting resulting in near 'live' test interpretation for the client based on the results of the day
- o vastly greater number of *what-if* scenarios could be considered (*e.g.,* higher stresses, lower build-quality, damage tolerance, NDI sensitivities) thus providing additional risk reduction and more reliable management decisions, and more effective resource management.

EBA challenges

- individual CGR constants for individual spectra → additional quality control effort.

- maintaining/proving the representativeness of the coupon tests → simple coupon specimen design *vs* interference fit fastener (note: this issue is not unique to the EBA).

- the estimation of Safe Life estimates required growing cracks backwards to time-zero using one spectrum, and then forward again using a different spectrum → a standardised, nominalised CG methodology was adopted which resulted in resulted in some EICS values that did not compare reasonably against physical microstructure dimensions → required additional effort to conduct sensitivity analyses, explain and justify (note: this issue is not unique to the EBA).

- The EBA required supplementary CGR analysis (*i.e.,* FASTRAN) to estimate the CGRs for untested spectra.

6.4.3 Spectra severity scaling

An example plot of some test and model predictions in a block CGR format is given in Figure 22. It shows that the mechanistic model (*i.e.,* uncalibrated FASTRAN) predictions were significantly unconservative, however, it also showed that model still seemed to have some capability in predicting the relative difference between these spectra – as indicated by the non-parallel trends between the red and blue spectra. This indicated that it may be appropriate in some cases to estimate offset and scaling factors (for the method described in Appendix E.2) using results from the FASTRAN code. Due to the non-parallel trends observed in the LITV coupon data, the F-WELD test interpretation methodology adopted the linear offset scaling method (Appendix E.2).
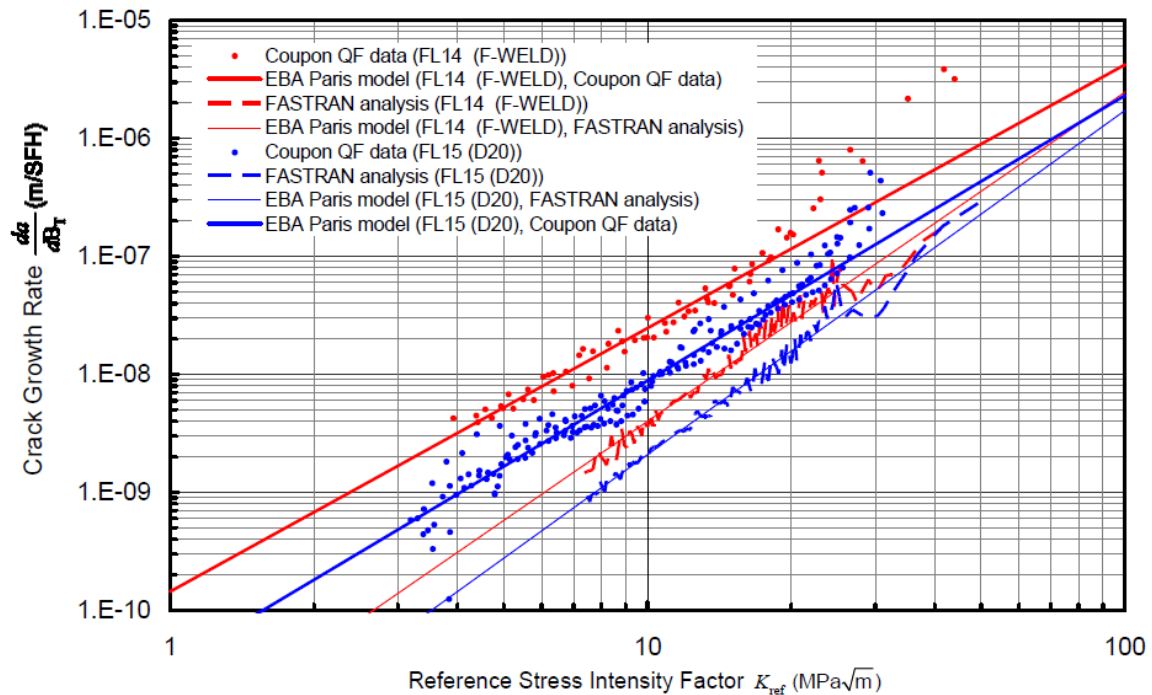
Figure 22: Example plot of test results and mechanistic-based model predictions in an EBA CGR format, allowing a more comprehensive assessment of the differences between them (as compared to traditional CG or total life comparisons). FL14 (FWELD) refers to the wing fatigue test spectrum and FL15 (D20) refers to the estimated RAAF usage spectrum at the same wing location. Figure reproduced from [38].

6.4.4 Reverse calculation of beta solutions

The F-111 wing skins are joined to the spars predominantly using interference-fit fasteners. The test interpretation methodology considered the reverse-calculation of beta solutions at these fasteners, firstly to assess if the idealised (linear-elastic open hole) case was an appropriate approximation for an interference fit, and secondly to provide an alternate method (*pro* FEM) for the determination of stress levels at the fastener locations.

The effective beta solutions for cracked fastener holes in a previous F-111C model wing test (wing no. A15-5) were reversed-calculated from QF measurements in [59]. These were also compared to an idealised (linear elastic open hole) solution in Figure 23. The trends of the effective beta solutions indicate that the interference-fit fasteners can behave similarly to an idealised open-hole case. Note however, that the beta derived for hole no. AAS 078 appears to be <u>consistently</u> higher than the idealised solution by about 20%. This consistency suggests the stress level at this fastener location was underestimated by about 20%. In this case the stress level was estimated by an FEM; and therefore it appears that the effective beta result may be compensating for this.

The use of effective beta solutions was adopted as an integral part of the F-WELD test interpretation methodology. This was done to ensure that complex and unique trends measured at each cracked fastener in the full-scale test article were identified and

DSTO-TR-2850

accounted for in the calculation of life estimates for RAAF load spectra. Furthermore, this was done so that FEM stress estimates were not required for the life calculation, *i.e.,* the stress level could be treated as an unknown along with the beta solution (such that the EBA was actually used to reverse-calculate a *βσ* term). This interpretation methodology provide a more direct life re-assessment approach, as compared to nominal, idealised model that may need to be adjusted to make reasonable sense of a full-scale test article result.
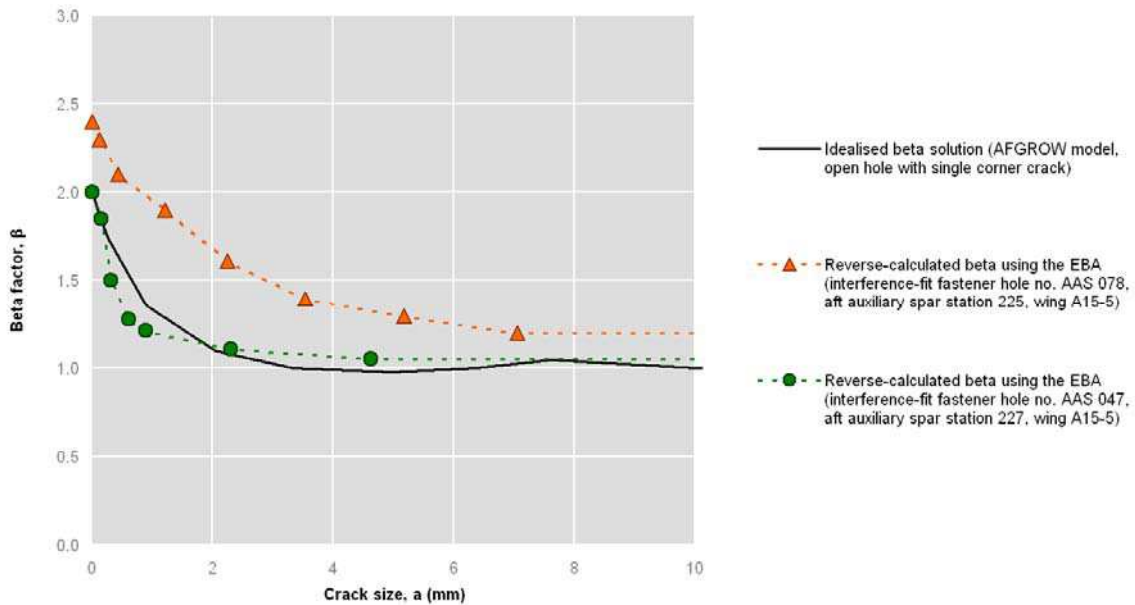
Figure 23: Comparison of effective beta solutions reverse-calculated from QF measurements at interference-fit fasteners against the idealised case (figure adapted from [59]).

6.4.5 Back-extrapolations to time zero – Equivalent Initial Crack Sizes

An EICS methodology was used to enable safe life (*i.e.,* total life with appropriate scatter factor) estimates to be made during the FWELD test interpretation [50]. Such a philosophy assumes that the fatigue life mechanism does not change significantly with crack size, even as the size reaches characteristic dimensions of the alloy micro-structure. A related assumption is that different load spectra will not alter the fatigue crack initiation mechanism. Pragmatically, this assumption can be checked by comparing EICS values for different spectra. If the EICS populations for two different spectra are shown to be similar, then the above assumptions may be considered sound.

Figure 24 shows EICS values for different spectra from numerous locations along the F-111 wing, showing two different spectra of interest at each location (identified by colour) [60]. It can be seen at EICS values within colour groups (same locations) appear to belong to similar populations, however, comparisons across locations tend to deviate more so. In this case, the FWELD test interpretation was only interested in same-location spectra

comparisons, however, the comparisons across locations does seem to indicate that there may be limitations with the EICS approach as spectra become more different.

It is noted that the EICS population here for AA2024-T851 under F-111 load spectra, appears to be similar to that derived for the AA7050-T7451 material under F/A-18 load spectra (Figure 20). This indicates some similarity in the fatigue crack initiation for these two materials.
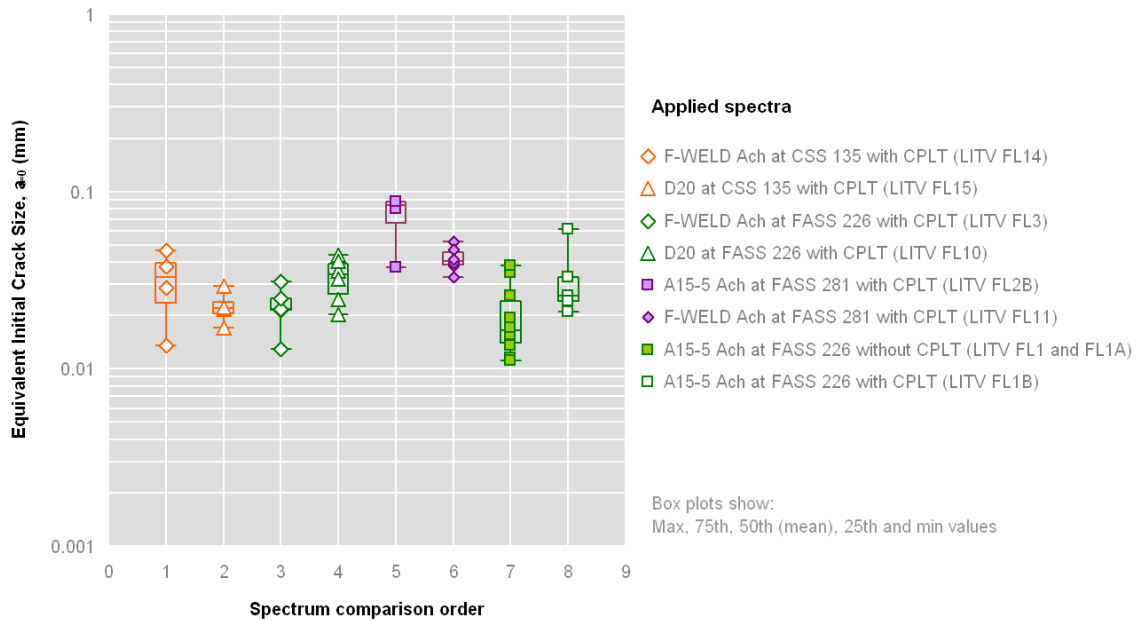
Open symbols: CPLT load is significantly greater than the maximum flight load
Filled symbols: CPLT load is close to or less than the maximum flight load

Figure 24: Comparison of EICS values for numerous F-111 wing bending moment spectra. Figure reproduced from [60].

7. Discussion

It has been observed that a regularly repeating block of VA loading tends to produce apparent steady-state CGRs. Trends of CGR against crack size and stress scale levels (or stress intensity factors) not only appear to be steady, but also relatively simplistic. It has been observed that this steadiness, and lack of complexity, tends to continue even when traditionally accepted bounds of validity are exceeded (*e.g.*, when the stress levels approach or exceed the material yield strength, and when crack sizes are relatively small). These observations not only imply that a characteristic type model is a viable methodology for some repeated-block loading metallic fatigue life assessments, but also that the range of predictability may be enhanced by considering pseudo ductile (elastic-plastic) CGR relationships. Importantly, it may provide solutions for a range of conditions for which traditional LEFM-based mechanistic type models may be found to be ill-suited.

Characteristic type models are usually considered because they are simple and efficient. This can allow very rapid test interpretations and CG estimates to be calculated. Yet they have also been considered here because, when they are used with pseudo ductile (elastic-plastic) CGR relationships, they can offer higher accuracy. The case studies show that this appears to increase the predictability of small crack and high stress effects. These scenarios tend to challenge traditional mechanistic models, likely due to long-crack threshold effects inherent in ASTM-type crack growth tests, and small-scale yielding bounds of validity for LEFM. This is not an indictment of mechanistic approaches, rather a reflection that today's mechanistic models are usually still based on linear-elastic theory – which may be inadequate for military airframes that are usually more highly stressed to improve manoeuvring performance.

This guide also highlights that there are many characteristic approaches in the literature. It is concerning that some of the approaches that are portrayed in current structural integrity text books were considered here to be unsuitable. It is possible that this may be contributing to an unnecessarily poor reputation toward characteristic block approaches in general. This guide recommends attributes that are considered necessary for an approach to be suitable for conducting fatigue life assessments. Such attributes include: repeated-block spectra; determining crack growth rates by representative fatigue tests; characterising these rates against well recognised similitude parameters (*e.g.*, crack size, stress or stress intensity factor); and incorporating the load sequences effects into constants that are separated from the similitude parameters (*i.e.*, the use effective similitude parameters are not recommended, as this prevents matching trends that can occur against these parameters). These attributes help to maintain accuracy, increase predictability, and provides flexibility toward fitting a wider range of observed CGR behaviours. These attributes are the basis for recommending the Effective Block Approach, or EBA, presented in this report.

There are a number of unique advantages to using an EBA to model repeated-block VA load spectra, namely: the ability to characterise CGR observations that are difficult to simulate using a cycle-by-cycle mechanical model; very fast calculations (typically << 0.1 sec); ability to grow cracks backwards; independent validation of stress intensity factors

and/or stress levels within full-scale structures (*e.g.*, through the reverse calculation of beta solutions). Furthermore, by plotting the average CGRs against the similitude parameters, important complex trends can be identified that might otherwise have been missed by the fatigue practitioner (*i.e.*, *pro* simple comparisons of initiation life or total life results). Note that the EBA can also be used on the results of a mechanistic model, thus providing similar detailed insights of the model's performance against the similitude parameters.

The main limitation of the EBA is that it has no ability to predict CGR for different spectra, without supplementary testing or analysis, or making further assumptions about the relative difference in CGRs for different spectra. Some CGR scaling methods are presented that allow such supplementary data to be used within an EBA framework. Importantly, the methods presented herein can allow more complex scaling, *e.g.*, for CGRs that vary against the similitude parameters differently for different VA spectra. This can provide greater accuracy compared to simpler scaling based on initiation or total life values only.

An EBA framework can support an EICS methodology, thus allowing characterisation of different surface conditions, *e.g.*, machined, etched, anodised, *etc*. This can also enable total life assessments to be considered, by extrapolating a crack backwards to time zero using the tested load spectrum, and then forward again using a different spectrum. However, this methodology implies an assumption of steady-state CGR down to crack sizes that usually approach the characteristic dimensions of the alloy micro-structure. A related assumption is that different load spectra will not alter the general fatigue crack initiation mechanism. It is reasonable to expect that these assumptions may only hold true in limited circumstances due to the effect that the microstructure may have on formation of a fatigue crack. Pragmatically, this assumption can be tested by comparing EICS values for specimens tested under different spectra. Two of the case studies presented herein have done so and appear to show favourable comparisons – *i.e.,* EICS values were calculated for numerous coupon specimens covering a range of stress scale levels and different load spectra types, and these values appeared to compare well for similar spectra. This may be because the case studies presented examples that were relatively highly stressed and hence exhibited short, if any, so-called initiation phases.

The EBA potentially allows DSTO to offer a greater range of options for the fatigue life management of RAAF airframes. Case studies presented herein highlight examples of this for Macchi, F/A-18 Hornet and F-111 aircraft. EBA frameworks have also been used to characterise EICS values to assess the effects of different surface conditions.

8. Conclusion

Simpler empirical characterisation of the mechanics of metal fatigue crack growth have been investigate to provide an alternative or supplement to mechanistic (physics-based) models. Such characterisation was found to be appropriate for cases where the loading consists of repeating blocks of spectra or other scenarios that result in near steady-state crack growth rates.

Numerous types of characteristic block approaches were examined, and it was found that the assumptions and efficacy of each can vary significantly. This report provides guidance on distinguishing between these types, and provides a recommendation of the particular attributes that are required for an approach to be considered suitable for conducting fatigue life assessments. Such attributes include: determining crack growth rates by representative fatigue tests; characterising these rates against well recognised similitude parameters (*e.g.,* crack size, stress or stress intensity factor); and incorporating the load sequences effects into constants that are separate from the similitude parameters (*i.e.,* the use of effective similitude parameters are not recommended). These attributes help to maintain accuracy, increase predictability, and provide flexibility toward fitting a wider range of observed crack growth behaviours. A particular approach that adopts these attributes, herein titled the Effective Block Approach, or EBA, is detailed in this report.

A number of case studies were reviewed where the EBA was used to conduct life assessment of RAAF aircraft. The studies support the suitability of the EBA particularly for test interpretation where test articles were loaded by repeating blocks of spectra. The studies also highlighted some useful capabilities such as: the accurate fatigue life predictions for more complex behaviours (*e.g.,* short crack and high-stress effects); characterising surface treatment effects by equivalent initial crack sizes; and the reverse-calculation of stress levels in full-scale assemblies without the need of a finite element internal load model. Furthermore, it was demonstrated that unanticipated benefits could be realised by the extraordinarily fast calculation times, such as more comprehensive sensitivity studies and near real-time updates and client reporting. This provides DSTO with additional capabilities that can either supplement traditional fatigue life assessment approaches (*i.e.,* finite element model stress estimates input into a mechanistic crack growth model), or bypass such traditional approaches completely.

DSTO has significant full-scale, component and coupon test facilities that it uses to assess RAAF airframes. The efficient and accurate modelling of crack growth is a necessary capability for translating the test outputs into relevant life results for structural integrity management purposes. The EBA enables DSTO to continue to offer leading structural integrity advice, even when mechanics-based models are still being developed. Already, the EBA has been used to assess numerous airframes including Macchi wings, F/A-18 Hornet centre barrels and F-111 wings. This capability is anticipated to give the RAAF additional options to better assess and manage the safety, availability and cost of ownership of its air vehicles.

9. Acknowledgements

The author wishes to sincerely thank the contributions, advice and support given by the following:

- Russell Wanhill of the National Aerospace Laboratory, NLR, Amsterdam, The Netherlands.
- Min Liao of the Canada Research Council Institute for Aerospace Research, National Research Council Canada.
- Loris Molent, Wyman Zhuang, Matthew Phillips, Chris Wallbrink, Kevin Walker, Phil Jackson and Rob Boykett of DSTO.

10. References

1. Hu, W., et al., *A Review and Assessment of Current Airframe Lifing Methodologies and Tools in Air Vehicles Division*. 2006, DSTO-RR-0321. Defence Science and Technology Organisation: Fishermans Bend.

2. McDonald, M. and L. Molent, *Fatigue Assessment of the F/A-18 Horizontal Stabilator Spindle*. 2004, DSTO-TR-1620. Defence Science and Technology Organisation: Fishermans Bend.

3. Swanton, G., *A Critical Review of the METLIFE Software in Support of the RAAF F-111 Durability and Damage Tolerance Analysis (DADTA) Program*. 2005, DSTO-TR-1713. Defence Science and Technology Organisation: Fishermans Bend.

4. Lubacz, J., et al., *FASTRAN Development and Re-Calibration in Support of the RAAF P-3 2010 Structural Management Plan Review*. 2011, DSTO-TR-2654. Defence Science and Technology Organisation: Fishermans Bend.

5. Molent, L., et al., *Enhanced Teardown of Ex-Service F/A-18A/B/C/D Centre Fuselages*, in *25th International Conference on Aeronautical Fatigue Symposium*. 2009: Rotterdam.

6. McDonald, M., L. Molent, and A.J. Green, *Assessment of Fatigue Crack Growth Prediction Models for F/A-18 Representative Spectra and Material*. 2006, DSTO-RR-0312. Defence Science and Technology Organisation: Fishermans Bend.

7. Zhuang, W., et al., *Effective Block Approach for Damage Tolerance Analyses of F-111 D/F Model Wing Structures*. 2008, DSTO-TR-2124. Defence Science and Technology Organisation: Fishermans Bend.

8. Clark, G., G.S. Jost, and G.D. Young, *Recovery of the RAAF Macchi MB326H - The Tale of an Aging Trainer Fleet*, in *Proceedings of the 19th International Committee on Aeronautical Fatigue (ICAF) Symposium*. 1997: Edinburgh, Scotland, UK. p. 39-58.

9. Zhuang, W., S. Barter, and L. Molent, *Flight-by-flight fatigue crack growth life assessment*. International Journal of Fatigue, 2007. **29**(9–11): p. 1647-1657.

10. Walker, K., M. McDonald, and G. Swanton, *F-WELD Fatigue Test and Associated Data Interpretation Plan*. 2007, DSTO-TR-2035. Defence Science and Technology Organisation.

11. McDonald, M., *Fatigue Assessment of the Bushmaster Infantry Mobility Vehicle Hull*. 2005, DSTO-TR-1739. Defence Science and Technology Organisation: Fishermans Bend.

12. Zhuang, W.Z., et al., *Effective Block Approach for Aircraft Damage Tolerance Analyses*. Journal of Aircraft, 2009. **46**(5): p. 1660-1666.

13. Zhuang, W.Z., et al. *Applications of Effective Block Approach for Aircraft Damage Tolerance Analyses*. in *Australian International Aerospace Congress*. 2009. Melbourne.

14. Zhuang, W. and L. Molent, *Fatigue Crack Growth Analysis for Notched Specimens under Flight Spectrum Loading*, in *Structural Failure Analysis and Prediction Methods for Aerospace Vehicles and Structures*, S.-Y. Ho, Editor. 2010, Bentham Science: Australia. p. 85-95.

15. Huynh, J. and L. Molent, *Application of the Effective Block Approach to Al7050-T7451 Coupon Results Tested Under Seven Different F/A-18 Spectra*. 2008, DSTO-TR-2179. Defence Science and Technology Organisation: Fishermans Bend.

16. Huynh, J., L. Molent, and S. Barter, *Experimentally derived crack growth models for different stress concentration factors.* International Journal of Fatigue, 2008. **30**(10-11): p. 1766-1786.

17. Molent, L., M. McDonald, and S. Barter, *Using Full-Scale Fatigue Testing Crack Growth Results to Life Aircraft Fleets*, in *Proceedings of the 30th International Committee on Aeronautical Fatigue (ICAF) Symposium*. 2007: Naples, Italy.

18. Molent, L., et al., *Evaluation of spectrum fatigue crack growth using variable amplitude data.* International Journal of Fatigue, 2008. **30**(1): p. 119-137.

19. Molent, L., Q. Sun, and A.J. Green, *Characterisation of equivalent initial flaw sizes in 7050 alloys.* Fatigue & Fracture of Engineering Materials & Structures, 2006. **29**(11): p. 916-937.

20. Zhuang, W. and L. Molent, *Block-by-block approach for spectrum fatigue crack growth prediction*, in *The Eighth International Conference on Fundamentals of Fracture*. 2008: Hong Kong & Guangzhou, China.

21. Molent, L., et al., *Spectrum fatigue crack growth evaluation using variable amplitude data*, in *12th Australian International Aerospace Congress*. 2007: Melbourne.

22. Barter, S., M. McDonald, and L. Molent, *Fleet fatigue life interpretation from full-scale and coupon fatigue tests - a simplified approach*, in *Proceedings of the USAF Aircraft Structural Integrity Program (ASIP) Conference*. 2005: Memphis TN, USA.

23. Grandt, A.F., *Fundamentals of Structural Integrity - Damage Tolerant Design and Non-Destructive Evaluation*. 2004: John Wiley & Sons Inc.

24. Wanhill, R.J.H., *Characteristic stress intensty factor correlations of crack growth in high strength alloys: reviews and compilation of NLR investigations 1985-1990*. 2009, NLR-TP-2009-256. National Aerospace Laboratory NLR: Amsterdam.

25. Bahai, H., G. Glinka, and I.I. Esat, *Numerical and Experimental Evaluation of SIF for Threaded Connectors.* Engineering Fracture Mechanics, 1996. **54**(6): p. 835-845.

26. Kim, S.T., D. Tadjiev, and H.T. Yang, *Fatigue Life Prediction under Random Loading Conditions in 7475-T7351 Aluminium Alloy using the RMS Model.* International Journal of Damage Mechanics, 2006. **15**: p. 89-102.

27. Hudson, C.M., *A Root-Mean-Square Approach for Predicting Fatigue Crack Growth under Random Loading*, in *Methods and Models for Predicting Fatigue Crack Growth under Random Loading*, J.B. Chang and C.M. Hudson, Editors. 1982, ASTM STP 748. p. 41-53.

28. Barsom, J.M. and S.T. Rolfe, *Fracture and Fatigue Control in Structures: Applications of fracture mechanics.* 3 ed. 1999, West Conshohocken, PA: American Society for Testing and Materials.

29. Kondo, Y., *Fatigue Under Variable Amplitude Loading*, in *Comprehensive Structural Integrity*, I. Milne, R.O. Ritchie, and B. Karihaloo, Editors. 2003, Pergamon: Oxford. p. 253-279.

30. Jones, R., L. Molent, and K. Krishnapillai, *An equivalent block method for computing fatigue crack growth.* International Journal of Fatigue, 2008. **30**(9): p. 1529-1542.

31. Gallagher, J.P., *Estimating Fatigue-Crack Lives for Aircraft: Techniques.* Experimental Mechanics, 1976. **16**(11): p. 425-433.

32. Barter, S., W. Zhuang, and L. Molent, *Flight-by-flight fatigue crack growth life assessment.* International Journal of Fatigue, 2007. **29**: p. 1647–1657.

33. Gallagher, J.P. and H.D. Stalnaker, *Developing Normalized Crack Growth Curves for Tracking Damage in Aircraft.* Journal of Aircraft, 1978. **15**(2): p. 114-120.

34. Dainty, R.V., *The use of 'marker blocks' as an aid in quantitative fractography in full-scale aircraft fatigue testing - A case study.* 1982, LTR-ST-1374. National Research Council Canada.

35. Liao, M. and G. Renaud, *Fatigue Analysis for CF-18 Component: Wing Fold Shear-Tie Lug.* Procedia Engineering, 2010. **2**(1): p. 1673-1682.

36. Paris, P.C., H. Tada, and J.K. Donald, *Service load fatigue damage -- a historical perspective.* International Journal of Fatigue, 1999. **21**(Supplement 1): p. 35-46.

37. Barsom, J.M., *Fatigue-Crack Growth Under Variable-Amplitude Loading in ASTM A 514B Steel.* 1973, ASTM STP 536. ASTM.

38. Wallbrink, C., et al., *An Evaluation of the Effective Block Approach Using P-3C and F-111 Crack Growth Data.* 2008, DSTO-TR-2195. Defence Science and Technology Organisation: Fishermans Bend.

39. McDonald, M., *Fatigue Crack Growth Methodologies in Structural Life Assessment - F/A-18 Case Study.* 2005, DSTO-DP-1045. Defence Science and Technology Organisation: Fishermans Bend.

40. Tomkins, B., *Fatigue Crack Propagation - An Analysis.* The Philosophical Magazine, 1968. **18**(155): p. 1041-1066.

41. Tomkins, B., *Micromechanisms of fatigue crack growth at high stress.* Metal Science Journal, 1980. **14**(8-9): p. 408-417.

42. Frost, N.E. and D.S. Dugdale, *The Propagation of Fatigue Cracks in Sheet Specimens.* Journal of the Mechanics and Physics of Solids, 1958. **6**(2): p. 92-110.

43. Barter, S., et al., *An experimental evaluation of fatigue crack growth.* Engineering Failure Analysis, 2005. **12**(1): p. 99-128.

44. Sih, G.C., *Handbook of Stress Intensity Factors for Researchers and Engineers.* 1973, Bethlehem, Pennsylvania: Institute of Fracture and Solid Mechanics, Lehigh University.

45. Tada, H., P. Paris, and G. Irwin, *The Stress Analysis of Cracks Handbook.* 1985, St. Louis, Missouri: Paris Productions Inc.

46. Rooke, D.P. and D.J. Cartwright, *Compendium of Stress Intensity Factors.* 1976, London: Her Majesty's Stationary Office.

47. *Stress Intensity Factors Handbook*, ed. Y. Murakami. 1987, New York, New York: Pergamon.

48. Gravina, R., *Beta Solution Determination Using FEA - Method Development.* 2005, ER-P3-51-APM195. AeroStructures Technologies Pty Ltd Port Melbourne.

49. International, A., *Standard Test Method for Measurement of Fatigue Crack Growth Rates.* 2007.

50. Phillips, M., et al., *Fatigue Life Validation for D- and F-model Wings on F-111 Aircraft.* 2010, DSTO-TR-2461. Defence Science and Technology Organisation: Fishermans Bend.

51. Goldsmith, N.T., G. Clark, and S.A. Barter, *A Growth Model for Catastrophic Cracking in an RAAF Aircraft.* Engineering Failure Analysis, 1996. **3**(3): p. 191-201.

52. Molent, L., S.A. Barter, and R.J.H. Wanhill, *The Lead Crack Fatigue Lifting Framework.* 2010, DSTO-RR-0353. Defence Science & Technology Organisation: Fishermans Bend.

53. Molent, L., S.A. Barter, and R.J.H. Wanhill, *The lead crack fatigue lifing framework.* International Journal of Fatigue, 2011. **33**(3): p. 323-331.

54. Molent, L. and Q. Sun, *The Compendium of F/A-18 Hornet Crack Growth Data Version II.* 2010, DSTO-TR-2488. Defence Science and Technology Organisation: Fishermans Bend.

55. *F/A-18 A/B Structural Analysis Methodology, Issue 2, AL3.* ASI/2006/1114755 Pt1 (20). Directorate General Technical Airworthiness, Royal Australian Air Force.

56. McDonald, M., *Correlating parameter for High and Low Kt F/A-18 coupon data.* 2008, Defence Science & Technology Organisation: Fishermans Bend.

57. Yu, X., M. McDonald, and W. Hu, *Numerical & Experimental Evaluations of the Geometric (Beta) Factor - Part I: F/A-18 High Kt and Low Kt coupon specimens.* 2010, File B2/129 Pt(4). Defence Science and Technology Organisation: Melbourne.

58. Diab, H. and R. Goldsmith, *Fractography Results of F-111 Loads Interpretation and Truncation Validation (LITV) Coupon Test Program.* 2007, DSTO-TR-2000. Air Vehicles Division, Defence Science and Technology Organisation.

59. Boykett, R., et al., *Structural Integrity Assessment of RAAF F-111C wings.* 2007, DSTO-TR-2045. Defence Science and Technology Organisation: Fishermans Bend.

60. McDonald, M., *F-111 Equivalent Initial Crack Size (EICS) Comparison.* 2010, Defence Science and Technology Organisation: Fishermans Bend.

Appendix A: Exponential method for the calculation of crack growth rates

The exponential method for the calculation of the crack growth rate, da/dt, uses a linear regression to calculate an estimate of the crack growth rate at a particular crack size a. However, the method uses the natural log of the crack size values as an input, so overall it is effectively an exponential regression method. The method requires at least 2, preferably 3, consecutive data points comprising crack length and time, *e.g.*, $[a_{i-1}, t_{i-1}]$, $[a_i, t_i]$, $[a_{i+1}, t_{i+1}]$. If only 2 points are available then the data at the point of interest $[a_i, t_i]$ is required plus either of the adjacent data points $[a_{i-1}, t_{i-1}]$ or $[a_{i+1}, t_{i+1}]$ can be used. The exponential method does not require that the data points be evenly spaced with respect to either crack size or time.

The traditional linear least-squares regression method is used to determine a best-fit slope, m in the expression $y = mx + b$, as follows:

$$m = \frac{n\sum(xy) - \sum x \sum y}{n\sum(x^2) - (\sum x)^2}$$

Eqn (A.1)

where n is the number of data sets, x are the time values and y are the natural log of the crack size a values. Using the natural log of the crack size values inherently forces the slope to take the form of an exponential crack growth relationship, therefore:

$$m = \frac{d(\ln(a))}{dt} = \frac{da}{dt}\frac{1}{a}$$

Eqn (A.2)

therefore the crack growth rate at crack size a is given as

$$\frac{da}{dt} = \left(\frac{d(\ln(a))}{dt}\right) a = m\,a$$

Eqn (A.3)

Table A.1 shows the input data and calculated results used in the example presented earlier in Section 5.1 (Figure 4).

Table A.1: Example input data and resulting crack growth rate calculated by the exponential method

	a	$ln(a)$	t	$m = \dfrac{d(\ln(a))}{dt}$	$\dfrac{da}{dt} = m\,a$
	(mm)	(ln(mm))	(flthr)	(flthr^{-1})	(mm/flthr)
i - 1	0.1	-2.303	10000		
i	0.2	-1.609	15000	0.0001609	3.219E-05
i + 1	0.5	-0.693	20000		

DSTO-TR-2850

Appendix B: Summary of crack growth equations

CRACK GROWTH RATE (DERIVATIVE)		CRACK GROWTH (INTEGRAL)
Traditional form	Natural log form (for parameter regression)	
Tomkins $$\frac{da}{dt} = A a^{j} \sigma_{ref}^{k}$$	$$\ln\!\left(\frac{da}{dt}\right) = \ln(A) + j\ln(a) + k\ln(\sigma_{ref})$$	Closed-form solution ($j \neq 1$) $$a_2 = \left[a_1^{(1-j)} + \left(\tfrac{2}{1+j}\right) A \sigma_{ref}^{k} \right]^{\left(\frac{1}{1-j}\right)}$$ or, for $j = 1$ use the Frost & Dugdale solution
Frost & Dugdale $$\frac{da}{dt} = \lambda a \sigma_{ref}^{\alpha}$$ or $$\frac{d[\ln(a)]}{dt} = \lambda \sigma_{ref}^{\alpha}$$	$$\ln\!\left(\frac{da}{dt}\frac{1}{a}\right) = \ln(\lambda) + \alpha \ln(\sigma_{ref})$$ or $$\ln\!\left(\frac{d[\ln(a)]}{dt}\right) = \ln(\lambda) + \alpha \ln(\sigma_{ref})$$	Closed-form solution $$a_2 = a_1 \exp\!\left[\lambda \sigma_{ref}^{\alpha} (t_2 - t_1) \right]$$
General $$\frac{da}{dt} = H K_{ref}^{p} \sigma_{ref}^{q}$$	$$\ln\!\left(\frac{da}{dt}\right) = \ln(H) + p\ln(K_{ref}) + q\ln(\sigma_{ref})$$	Numerical solution IntegrateGeneral() function
Paris $$\frac{da}{dt} = C K_{ref}^{m}$$	$$\ln\!\left(\frac{da}{dt}\right) = \ln(C) + m\ln(K_{ref})$$	For constant β and $m = 2$ only, can use the closed-form solution: $$a_2 = a_1 \exp\!\left[C\pi(\sigma_{ref}\beta)^2 (t_2 - t_1) \right]$$ or, for constant β and $m \neq 2$ only, can use the closed-form solution: $$a_2 = \left[a_1^{\left(1-\frac{m}{2}\right)} + C\,(1-\tfrac{m}{2})\left(\sigma_{ref}\beta\sqrt{\pi}\right)^m (t_2 - t_1) \right]^{\left(\frac{1}{1-\frac{m}{2}}\right)}$$ otherwise use the numerical solution IntegrateGeneral() function with $q = 0$

Appendix C: Matlab IntegrateGeneralCGM() function

Input	Input description	
INTEGRATEGENERALCGM(		
H	General model coefficient, H	Required
p	General model exponent, p	Required
q	General model exponent, q	Required
RefStress	Gross-section reference stress, $\sigma_{\text{gross-reference}}$	Required
ai	Initial (start) crack size, $a_{initial}$	Required

fType	Final (stop) type:
	    "a" for crack size OR
	    "K" for stress intensity factor OR
	    "t" for time difference
		Required

fValue	Final (stop value): a_{final} OR K_{final} OR Δt	Required

Output_Type	Return value type:
	    "a" for crack size OR
	    "K" for stress intensity factor OR
	    "t" for time difference
		Required

Beta	Beta-solution array: $[a_{B1},\ \beta_{B1};\ a_{B2},\ \beta_{B2};\ ...\ ]$	Optional (Defaults to β = 1 for all crack sizes)
NetStressRatio	Stress-ratio solution array: $[a_{S1},\ SR_{S1};\ a_{S2},\ SR_{S2};\ ...\ ]$	Optional (Defaults to a ratio of 1 for all crack sizes)
)		

```
function [tf, af, Kf] = IntegrateGeneralCGM(H, p, q, RefStress, ai, fType, fValue, Beta, NetStressRatio)
% function to integrate the DSTO General crack growth rate model
%
% Copyright © 2006 by
% Defence Science and Technology Organisation
% 506 Lorimer St
% Fishermans Bend VIC 3207
% Australia
%
%  General crack growth rate model:
%    da/dt = H.(K^p).(NetStresst^q)
%
%  Other model types can be derived from the General equation as follows:
%
%   Paris          Let p = the typical Paris exponent, set q = 0,
%                  and let H = the typical Paris coefficient
%
%   Frost & Dugdale Let p = 2 to force a log-linear crack-size relationship,
%                  let q = 1 to force a stress-cube relationship,
%                  set Beta = 1/sqrt(pi),
%                  and let H = the typical Frost & Dugdale coefficient
%
%
%
% INPUT PARAMETERS:
% (H, p, q, RefStress, ai, fType, fValue, Beta, NetStressRatio)
%
%   H          General CGM rate coefficient constant
%   p          General CGM rate exponent constant on K
```

```
%  q              General CGM rate exponent constant on NetStress
%  RefStress      Reference stress for the variable amplitude spectrum
%  ai             Initial crack size
%
%  fType          Final criteria to end the integration
%                    fType = 'a' final crack size
%                    fType = 't' time difference (positive or negative)
%                    fType = 'K' final reference stress intensity factor
%  fValue         Magnitude of the final criteria
%
%  Beta             Beta table - OPTIONAL
%                   Array of [Crack size, Beta]
%                   where Beta is the geometry and load/constraint factor
%                           used to calculate the stress intensity factor
%                   Defaults to Beta = 1 for all crack sizes
%
%  NetStressRatio   Net-section stress ratio table - OPTIONAL
%                   Array of [Crack size, NetStressRatio]
%                   where NetStressRatio = Net-section stress with a crack
%                           divided by Net-section stress with no crack
%                   Defaults to NetStressRatio = 1 for all crack sizes
%
% OUTPUT PARAMETERS:
% [tf, af, Kf]
%  tf             Final time difference
%  af             Final crack size
%  Kf             Final reference stress intensity factor
%
%


% ----------- Check for optional arguments --------------
if nargin == 7
    % Beta was not parsed
    % Assume value of 1
    warning('CGM:VariableNotParsed:AssumeBetaIs1',...
        'Beta was not parsed to the function.\n%s', ...
        'A default value of 1 for all crack sizes will be used')
    Beta = [1,1;2,1]
end
if nargin == 7 || nargin == 8
    % NetStressRatio was not parsed
    % Assume value of 1
    warning('CGM:VariableNotParsed:AssumeNetStressRatioIs1', ...
        'NetStressRatio variable was not parsed to the function.\n%s', ...
        'A default value of 1 for all crack sizes will be used')
    NetStressRatio = [1,1;2,1]
end

% ------------- Perform Input Checks -----------------------------------
PerformInputChecks(H, p, q, RefStress, ai, fType, fValue, Beta, NetStressRatio) ;
% Expand 1-row arrays to 2 rows to allow extrapolations
if length(Beta(:,1))==1
    Beta = [Beta;2.*Beta(1) Beta(2)] ;
end
if length(NetStressRatio(:,1))==1
    NetStressRatio = [NetStressRatio;2.*NetStressRatio(1) NetStressRatio(2)] ;
end

% ------------- Synchronise Beta and NetStressRatio ---------------------
% --------------------- into a single matrix ---------------------------
a = (unique([Beta(:,1);NetStressRatio(:,1)])) ;
ABS = [a,...
        interp1(Beta(:,1),Beta(:,2),a,'linear','extrap'), ...
        interp1(NetStressRatio(:,1),NetStressRatio(:,2),a,'linear','extrap')];
% ABS
% Check that the extrapolated values are non-zero and positive
if ~(min(min(ABS > (ABS .* 0))'))
    ABS
    'Crack Size, Beta, NetStressRatio'
    error('Extrapolated values for the Beta and NetStressRatio distributions must be positive.')
end

% ------ Determine the final crack size --------
% ------ and stress intensity factor -----------

% Determine the start index
icount = length(ABS(:,1)) ;
ABSi = [1:1:icount]' ;
istart = floor(interp1(ABS(:,1),ABSi,ai,'linear','extrap')) ;
istart = max(min(istart,icount-1),1) ;

SIF_handle = @SIF ;
options = optimset ;

if fType == 'a'
    af = fValue ;

elseif fType == 'K'
    Kf = fValue ;

    % Determine the initial K
    i = istart ;
    Bg = (ABS(i+1,2)-ABS(i,2))/(ABS(i+1,1)-ABS(i,1)) ;
    Bc = ABS(i,2)- Bg.*ABS(i,1) ;
    CheckPositiveExtrapolations(ai,ai,Bc,Bg,1,0);
    Ki = SIF(ai,RefStress,Bc,Bg,0) ;

    % Determine if crack is growing forwards or backwards
```

```
% Assume if Kf > Ki then grows forward, which is usually the case.
% However K can decrease with increasing crack size if the Beta distribution
% is reducing significantly.
% An error message will occur if Kf cannot be found.

% Determine the final crack size

if Ki == Kf
    Kfound = true ;
    af = ai ;
else
    Kfound = false ;
    a1 = ai ;
    a2 = ai ;
    if Kf > Ki
        GrowForward = true ;
        warning('CGM:Kincreasing:GrowForward', ...
        'Kf > Ki therefore assuming forward crack growth')
      else
        GrowForward = false ;
        warning('CGM:Kdecreasing:GrowBackward', ...
        'Kf < Ki therefore assuming backward crack growth')
    end
end

while ~Kfound

    % Check if Kf is in the current segment
    if GrowForward
        if i == icount
            % Calculate Beta constants for the current segment
                Bg = (ABS(i,2)-ABS(i-1,2))/(ABS(i,1)-ABS(i-1,1)) ;
                    Bc = ABS(i-1,2)- Bg.*ABS(i-1,1) ;
            % Check if Beta is negative at a very large crack size
            % if so then reduce a2 to within positive limits
            a2 = 1e+10 ;
            if Bc+a2*Bg <= 1e-10
                a2 = (1e-10 - Bc)/Bg ;
            end
        else
            % Calculate Beta constants for the current segment
                Bg = (ABS(i+1,2)-ABS(i,2))/(ABS(i+1,1)-ABS(i,1)) ;
                    Bc = ABS(i,2)- Bg.*ABS(i,1) ;
            a2 = ABS(i+1,1) ;
        end
        % Calculate the stress intensity factor
        K2 = SIF(a2,RefStress,Bc,Bg,0) ;
        if K2 == Kf
            Kfound = true ;
            af = a2 ;
        else
            if K2 > Kf
                % Kf is in this segment, can now calculate exact value of af
                % There exists an exact solution to this, but the equation is so
                % long that the fzero function was used instead. If calculation speed
                % becomes an issue, then consider replacing this.
                af = fzero(SIF_handle,[a1 a2],options,RefStress,Bc,Bg,-Kf) ;
                Kfound = true ;
            else
                if i == icount
                    warning('CGM:IntegrationIncomplete:CouldNotFindKf', ...
                        'INTEGRATION  INCOMPLETE:  Could  not  find  a  crack  size  for  Kf  assuming  forward  crack
growth.\n%s%s\n%s\n%s%s%f\n%s%e%s\n%s%f', ...
                        'Either the crack size became too large, ', ...
                        'or the limit of the positive range of Beta was reached.', ...
                        'Stopped at:', 'K = ', K2, ', a = ', a2, ' (limit 1e+10)', 'Beta = ', Bc+a2*Bg)
                    af = a2 ;
                    Kf = K2 ;
                    Kfound = true ;
                else
                    a1 = a2 ;
                    i = i + 1 ;
                end
            end
        end
    else
        if i == 0 % First segment
            % Calculate Beta constants for the current segment
                Bg = (ABS(i+2,2)-ABS(i+1,2))/(ABS(i+2,1)-ABS(i+1,1)) ;
                    Bc = ABS(i+1,2)- Bg.*ABS(i+1,1) ;
            % Check if Beta is negative at a very large crack size
            % if so then reduce a1 to within positive limits
            a1 = 1e-20 ;
            if Bc+a1*Bg <= 1e-10
                a1 = (1e-10 - Bc)/Bg ;
            end
        else % Not first segment
            % Calculate Beta constants for the current segment
                Bg = (ABS(i+1,2)-ABS(i,2))/(ABS(i+1,1)-ABS(i,1)) ;
                    Bc = ABS(i,2)- Bg.*ABS(i,1) ;
            a1 = ABS(i,1) ;
        end
        % Calculate the stress intensity factor
        K1 = SIF(a1,RefStress,Bc,Bg,0) ;

        if K1 == Kf
            Kfound = true ;
            af = a1 ;
```

```
                else
                    if K1 < Kf
                        % Kf is in this segment, can now calculate exact value of af
                        % There exists an exact solution to this, but the equation is so
                        % long that the fzero function was used instead. If calculation speed
                        % becomes an issue, then consider replacing this.
                        af = fzero(SIF_handle,[a1 a2],options,RefStress,Bc,Bg,-Kf) ;
                        Kfound = true ;
                    else
                        if i == 0
                            warning('CGM:IntegrationIncomplete:CouldNotFindKf', ...
                                'INTEGRATION  INCOMPLETE:  Could  not  find  a  crack  size  for  Kf  assuming  backward  crack
growth.\n%s%s\n%s\n%s%f\n%s%e%s\n%s%f', ...
                                'Either the crack size became too small, ', ...
                                'or the limit of the positive range of Beta was reached.', ...
                                'Stopped at:', 'K = ', K1, ', a = ', a1, ' (limit 1e-20)', 'Beta = ', Bc+a1*Bg)
                            af = a1 ;
                            Kf = K1 ;
                            Kfound = true ;
                        else
                            a2 = a1 ;
                            i = i - 1 ;
                        end
                    end
                end
            end
        end
    end
end

if fType == 'a' | fType == 'K'
    % Determine the final index
    ifinish = floor(interp1(ABS(:,1),ABSi,af,'linear','extrap')) ;
    ifinish = max(min(ifinish,icount-1),1) ;
end

if fType == 'a'
    % Determine the final stress intensity factor
    i = ifinish ;
    Bg = (ABS(i+1,2)-ABS(i,2))/(ABS(i+1,1)-ABS(i,1)) ;
    Bc = ABS(i,2)- Bg.*ABS(i,1) ;
    CheckPositiveExtrapolations(af,af,Bc,Bg,0,1) ;
    Kf = SIF(af,RefStress,Bc,Bg,0) ;
end


% -------- Perform time integration ----------
if fType == 'a' | fType == 'K'
    % Create a function handle to the derivative
    dtda_handle = @Generaldtda ;

    tf = 0 ;

    if ai == af
        StopIntegration = true ;
    else
        StopIntegration = false ;
        a1 = ai ;
        i = istart ;
          % Determine if growing forwards or backwards
          if af > ai
            GrowForward = true ;
          else
            GrowForward = false ;
          end
    end

    while ~StopIntegration
        % Check if af is in the current segment
        if i == ifinish
            a2 = af;
            StopIntegration = true;
        else
            if GrowForward
                a2 = ABS(i+1,1) ;
            else
                a2 = ABS(i,1) ;
            end
        end
        if ~(a1 == a2)
                Bg = (ABS(i+1,2)-ABS(i,2))/(ABS(i+1,1)-ABS(i,1)) ;
                Bc = ABS(i,2)- Bg.*ABS(i,1) ;
                Sg = (ABS(i+1,3)-ABS(i,3))/(ABS(i+1,1)-ABS(i,1)) ;
                Sc = ABS(i,3)- Sg.*ABS(i,1) ;
            % Check for positive extrapolations of Beta and NetStressRatio
            CheckPositiveExtrapolations(a1,a2,Bc,Bg,Sc,Sg);
            % Integrate time from a1 to a2
            tf = IntegrateGeneraldtda(a2,a1,dtda_handle,H,p,q,RefStress,Bc,Bg,Sc,Sg,tf) ;
            % [a1,a2,Bc,Bg,Sc,Sg]
        end
        if GrowForward
            i = i + 1 ;
        else
            i = i - 1 ;
        end
        a1 = a2 ;
    end
end
```

```
% -------- Perform size integration ----------
if fType == 't'
    tf = fValue ;

     % Create a function handle to the derivative
    dtda_handle = @Generaldtda ;
     integrate_handle = @IntegrateGeneraldtda ;

    if tf == 0
        StopIntegration = true ;
        af = ai ;
    else
        StopIntegration = false ;
        i = istart ;
        a1 = ai ;
        a2 = ai ;
        t1 = 0 ;
        t2 = 0 ;
        if tf > 0
            GrowForward = true ;
          else
            GrowForward = false ;
          end
    end

    while ~StopIntegration

        if GrowForward
            if i == icount % Last segment
                % Calculate Beta and NetStressRatio constants
                    Bg = (ABS(i,2)-ABS(i-1,2))/(ABS(i,1)-ABS(i-1,1)) ;
                            Bc = ABS(i-1,2)- Bg.*ABS(i-1,1) ;
                            Sg = (ABS(i,3)-ABS(i-1,3))/(ABS(i,1)-ABS(i-1,1)) ;
                            Sc = ABS(i-1,3)- Sg.*ABS(i-1,1) ;
                % Check if Beta is negative at a very large crack size
                % if so then reduce a2 to within positive limits
                a2 = 1e+10 ;
                if Bc+a2*Bg <= 1e-10
                    a2 = (1e-10 - Bc)/Bg;
                end
                % Check if NetStressRatio is negative
                % if so then reduce a2 to within positive limits
                if Sc+a2*Sg <= 1e-10
                    a2 = (1e-10 - Sc)/Sg;
                end
            else % Not last segment
                % Calculate Beta and NetStressRatio constants
                    Bg = (ABS(i+1,2)-ABS(i,2))/(ABS(i+1,1)-ABS(i,1)) ;
                            Bc = ABS(i,2)- Bg.*ABS(i,1) ;
                            Sg = (ABS(i+1,3)-ABS(i,3))/(ABS(i+1,1)-ABS(i,1)) ;
                            Sc = ABS(i,3)- Sg.*ABS(i,1) ;
                a2 = ABS(i+1,1) ;
            end
            % Integrate time from a1 to a2
            t2 = IntegrateGeneraldtda(a2,a1,dtda_handle,H,p,q,RefStress,Bc,Bg,Sc,Sg,t1) ;
            if t2 == tf
                StopIntegration = true ;
                af = a2;
            else
                % Check if tf is in the current segment
                if t2 > tf
                    % tf is in this segment, can now calculate exact value of af
                    % There exists an exact solution to this, but the solution
                    % contains a hypergeometric term that Matlab does not support,
                    % hence the fzero function was used instead. If calculation speed
                    % becomes an issue, then consider revisiting this.
                    af = fzero(integrate_handle,[a1 a2],options,a1,dtda_handle, ...
                            H,p,q,RefStress,Bc,Bg,Sc,Sg,t1-tf) ;
                    StopIntegration = true ;
                else
                    % check if this is the last segment, and increment to the next if OK
                    if i == (icount)
                        warning('CGM:IntegrationIncomplete:CrackTooLarge', ...
                            'INTEGRATION INCOMPLETE: Final crack size too large or\n%s\n%s\n%s%f\n%s%e%s\n%s%f\n%s%f', ...
                            'limit of positive range of Beta or NetStressRatio was reached', ...
                            'Stopped at:', 't = ', t2, ', a = ', a2, ' (limit 1e+10)', ...
                            'Beta = ', Bc+a2*Bg, 'NetStressRatio = ', Sc+a2*Sg)
                        af = a2 ;
                        tf = t2 ;
                        StopIntegration = true ;
                    else
                        t1 = t2 ;
                        a1 = a2 ;
                        i = i + 1 ;
                    end
                end
            end
        else
            if i == 0 % First segment
                % Calculate Beta and NetStressRatio constants
                    Bg = (ABS(i+2,2)-ABS(i+1,2))/(ABS(i+2,1)-ABS(i+1,1)) ;
                            Bc = ABS(i+1,2)- Bg.*ABS(i+1,1) ;
                            Sg = (ABS(i+2,3)-ABS(i+1,3))/(ABS(i+2,1)-ABS(i+1,1)) ;
                            Sc = ABS(i+1,3)- Sg.*ABS(i+1,1) ;
                % Check if Beta is negative at a very small crack size
                % if so then increase a1 to within positive limits
                a1 = 1e-20 ;
                if Bc+a1*Bg <= 1e-10
```

```
                    a1 = (1e-10 - Bc)/Bg;
                end
                % Check if NetStressRatio is negative
                % if so then increase a1 to within positive limits
                if Sc+a1*Sg <= 1e-10
                    a1 = (1e-10 - Sc)/Sg;
                end
            else % Not first segment
                % Calculate Beta and NetStressRatio constants
                    Bg = (ABS(i+1,2)-ABS(i,2))/(ABS(i+1,1)-ABS(i,1)) ;
                        Bc = ABS(i,2)- Bg.*ABS(i,1) ;
                        Sg = (ABS(i+1,3)-ABS(i,3))/(ABS(i+1,1)-ABS(i,1)) ;
                        Sc = ABS(i,3)- Sg.*ABS(i,1) ;
                a1 = ABS(i,1) ;
            end
            % Integrate time from a2 to a1
            t1 = IntegrateGeneraldtda(a1,a2,dtda_handle,H,p,q,RefStress,Bc,Bg,Sc,Sg,t2) ;
            if t1 == tf
                StopIntegration = true ;
                af = a1;
            else
                if t1 < tf
                    % tf is in this segment, can now calculate exact value of af
                    % There exists an exact solution to this, but the solution
                    % contains a hypergeometric term that Matlab does not support,
                    % hence the fzero function was used instead. If calculation speed
                    % becomes an issue, then consider revisiting this.
                    af = fzero(integrate_handle,[a2 a1],options,a2,dtda_handle, ...
                            H,p,q,RefStress,Bc,Bg,Sc,Sg,t2-tf) ;
                    StopIntegration = true ;
                else
                    if i == 0
                        warning('CGM:IntegrationIncomplete:CrackTooSmall', ...
                            'INTEGRATION INCOMPLETE: Final crack size too small or\n%s\n%s\n%s%f\n%s%e%s\n%s%f\n%s%f', ...
                            'limit of positive range of beta or NetStressRatio was reached', ...
                            'Stopped at:', 't = ', t1, 'a = ', a1, ' (limit 1e-20)', ...
                            'Beta = ', Bc+a1*Bg, 'NetStressRatio = ', Sc+a1*Sg)
                        tf = t1 ;
                        af = a1 ;
                        StopIntegration = true ;
                    else
                        t2 = t1 ;
                        a2 = a1 ;
                        i = i - 1 ;
                    end
                end
            end
        end
    end
    % Calculate the final stress intensity factor
    Kf = SIF(af,RefStress,Bc,Bg,0) ;
end


%----------------------------------------------------------------
%---------------------------SIF----------------------------------
%----------------------------------------------------------------

function K = SIF(x,RefStress,Bc,Bg,Koffset)
% calculates the Stress Intensity Factor
%
% x          Crack size
% RefStress  Reference stress
% Bc         Beta intercept at x=0 (OPTIONAL, default = 1)
% Bg         Beta gradient (OPTIONAL, default = 0)
% Koffset    Stress intensity factor offset (OPTIONAL, default = 0)

if nargin == 2
    Koffset=0 ;
    Bc = 1 ;
    Bg = 0 ;
end
if nargin == 3
    Koffset=0 ;
    Bg = 0 ;
end
if nargin == 4
    Koffset=0 ;
end
K = Koffset + RefStress.*(Bc+Bg.*x).*sqrt(pi.*x) ;
return

%----------------------------------------------------------------
%-----------------------Generaldtda------------------------------
%----------------------------------------------------------------

function dtda = Generaldtda(x,H,p,q,RefStress,Bc,Bg,Sc,Sg)
% dt/da crack growth rate equation
dtda = 1./(H.*((RefStress.*(Bc+Bg.*x).*sqrt(pi.*x)).^p).*(RefStress.*(Sc+Sg.*x)).^q) ;
return

%----------------------------------------------------------------
%-------------------IntegrateGeneraldtda-------------------------
%----------------------------------------------------------------

function t2 = IntegrateGeneraldtda(x2,x1,dtda_handle,H,p,q,RefStress,Bc,Bg,Sc,Sg,t1)
% Integral of the dt/da crack growth rate equation
t2 = t1 + quadl(dtda_handle,x1,x2,1.0e-6,0,H,p,q,RefStress,Bc,Bg,Sc,Sg) ;
```

```
return

%----------------------------------------------------------------
%-----------------CheckPositiveExtrapolations---------------------
%----------------------------------------------------------------

function CheckPositiveExtrapolations(a1,a2,Bc,Bg,Sc,Sg);
% Check that Beta and NetStressRatio extrapolations are positive
if Bc+a1*Bg <= 0
    error('%s%f','Non-positive Beta value occurs at crack size = ', a1)
end
if Bc+a2*Bg <= 0
    error('%s%f','Non-positive Beta value occurs at crack size = ', a2)
end
if Sc+a1*Sg <= 0
    error('%s%f','Non-positive NetStressRatio value occurs at crack size = ', a1)
end
if Sc+a2*Sg <= 0
    error('%s%f','Non-positive NetStressRatio value occurs at crack size = ', a2)
end
return

%----------------------------------------------------------------
%---------------------PerformInputChecks-------------------------
%----------------------------------------------------------------

function PerformInputChecks(H, p, q, RefStress, ai, fType, fValue, Beta, NetStressRatio)
% Perform input checks. Abort and display error message if there is a problem
if (ai <= 0)
    ai
    error('ai must be greater than 0')
end
if (RefStress <= 0)
    RefStress
    error('RefStress must be greater than 0')
end
if (H <= 0)
    H
    error('H must be greater than 0')
end
if (p <= 0)
    p
    error('p must be greater than 0')
end
if ~((strcmp(fType,'a')) || (strcmp(fType,'t')) || (strcmp(fType,'K')))
    error('fType not specified correctly.\n\n%s\n%s\n%s\n%s', ...
        'fType must = ',...
        '''a'' for crack size, or', ...
        '''t'' for time, or', ...
        '''K'' for stress intensity factor.')
end
if (fValue <= 0) & ((fType == 'a') | (fType == 'K'))
    fValue
    error('fValue must be greater than 0')
end
% ------------- Beta Checks ---------------
% Check that there are two (2) columns
if ~(size(Beta,2)==2)
    Beta
    error('Beta array must contain two columns.\n%s\n%s', ...
        '1st column must contain crack size data',...
        '2nd column must contain beta factor data')
end
% Check that all numbers are numeric
if ~(isnumeric(Beta))
    Beta
    error('Beta array must contain all numeric values.'),
end
% Check that all numbers are non-zero and positive
if ~(min(min(Beta > (Beta .* 0))'))
    Beta
    error('Beta array must contain all positive values.')
end
% Check that the crack size data is sorted in ascending order
if ~(min(min(Beta==sortrows(Beta,1))'))
    Beta
    error('Beta crack size data must be sorted in ascending order.')
end
% Check that all crack size data are unique
if ~(length(Beta(:,1))==length(unique(Beta(:,1))))
    Beta
    error('All Beta crack size data must be unique.')
end

% ------------- NetStressRatio checks ---------------
% Check that there are two (2) columns
if ~(size(NetStressRatio,2)==2)
    NetStressRatio
    error('NetStressRatio array must contain two columns.\n%s\n%s', ...
        '1st column must contain crack size data',...
        '2nd column must contain net/gross section stress ratio factor data')
end
% Check that all numbers are numeric
if ~(isnumeric(NetStressRatio))
    NetStressRatio
    error('NetStressRatio array must contain all numeric values.'),
end
% Check that all numbers are non-zero and positive
```

```
if ~(min(min(NetStressRatio > (NetStressRatio .* 0))')))
    NetStressRatio
    error('NetStressRatio array must contain all positive values.')
end
% Check that the crack size data is sorted in ascending order
if ~(min(min(NetStressRatio==sortrows(NetStressRatio,1))')))
    NetStressRatio
    error('NetStressRatio crack size data must be sorted in ascending order.')
end
% Check that all crack size data are unique
if ~(length(NetStressRatio(:,1)) == length(unique(NetStressRatio(:,1))))
    NetStressRatio
    error('All NetStressRatio crack size data must be unique.')
end
return
```

Appendix D: Excel IntegrateGeneral() function

Input	Input description	
INTEGRATEGENERAL(		
Ref_Stress_Gross	Gross-section reference stress, $\sigma_{gross\text{-}reference}$	Required
GeneralCoeff_H	General model coefficient, H	Required
GeneralExp_p	General model exponent, p	Required
GeneralExp_q	General model exponent, q	Required
Initial_a	Initial (start) crack size, $a_{initial}$	Required

Final_Type	Final (stop) type:
	    "a" for crack size OR
	    "K" for stress intensity factor OR
	    "t" for time difference
		Required

Final_Value	Final (stop value): a_{final} OR K_{final} OR Δt	Required

Output_Type	Return value type:
	    "a" for crack size OR
	    "K" for stress intensity factor OR
	    "t" for time difference
		Required

BetaTble_a_Rng	Array of beta-solution crack size values: a_{B1}, a_{B2}, ...	Optional (Defaults to $\beta = 1$ for all crack sizes)
BetaTble_B_Rng	Array of beta-solution beta values: β_{B1}, β_{B2}, ...	
StressTble_a_Rng	Array of stress-ratio crack size values: a_{S1}, a_{S2}, ...	Optional (Defaults to a ratio of 1 for all crack sizes)
StressTble_R_Rng	Array of stress-ratio values: SR_{S1}, SR_{S2}, ...	
)		

```
'
' Copyright © 2010 by
' Defence Science and Technology Organisation
' 506 Lorimer St
' Fishermans Bend VIC 3207
' Australia
'
'
' Function for the integration of the
' GENERAL CRACK GROWTH RATE MODEL
' da/dt = H.(RefStress.beta.sqrt(Pi.a))^p.RefNetStress^q
' by M. McDonald, Air Vehicles Division.
'
' LIMITS
' Maximum number of integration cycles = 10000000
' Maximum crack size = 1E+100
' Minimum crack size = 1E-100
'
'
Function INTEGRATEGENERAL( _
Ref_Stress_Gross As Double, _
GeneralCoeff_H As Double, _
GeneralExp_p As Double, _
GeneralExp_q As Variant, _
Initial_a As Double, _
Final_Type As String, _
Final_Value As Double, _
Output_Type As String, _
Optional BetaTble_a_Rng As Variant, _
Optional BetaTble_B_Rng As Variant, _
```

```
    Optional StressTble_a_Rng As Variant, _
    Optional StressTble_R_Rng As Variant _
    )
'
' Output_Type: "a" = crack size, "t" = time, "K" = stress intensity factor
'
    Dim error_msg As String
    Dim rCell As Range
'
    Dim a As Double
    Dim t As Double
    Dim K As Double
    Dim da As Double
    Dim da2 As Double
    Dim dt As Double
    Dim dK As Double
    Dim Final_a As Double
    Dim Final_t As Double
    Dim Final_K As Double, Initial_K As Double
    Dim Pi As Double
    Dim Ref_Stress_Net As Double
'
    Dim beta As Double
    Dim beta_a() As Double
    Dim beta_B() As Double
    Dim beta_slope() As Double
'
    Dim StressRatio As Double
    Dim StressRatio_a() As Double
    Dim StressRatio_R() As Double
    Dim StressRatio_slope() As Double
'
    Dim an As Integer, a_count As Integer
    Dim bn As Integer, b_count As Integer, beta_count As Integer
    Dim R_count As Integer, StressRatio_count As Integer
    Dim i As Integer, j As Integer
    Dim split As Integer
'
    Dim Integration_Cycle As Long
    Dim Maximum_Integration_Cycles As Long
'
    Dim bNoError As Boolean
    Dim bGrow_Forward As Boolean
    Dim bInitial_K_IncreasingForward As Boolean
    Dim bContinue_to_Grow As Boolean
    Dim bContinue As Boolean
    Dim bDebug As Boolean

    On Error GoTo ErrorHandler

    split = 1000
    Maximum_Integration_Cycles = 10000000
    bNoError = True
    bDebug = False
    bContinue = False

    '-------------------------------------------------
    ' ONLY continue if following input conditions are met

    If (Initial_a <= 0) Then
        bNoError = False
        error_msg = "Initial crack length must be greater than 0"
    End If
    If (Ref_Stress_Gross <= 0) Then
        bNoError = False
        error_msg = "Reference stress must be greater than 0"
    End If
    If (GeneralCoeff_H <= 0) Then
        bNoError = False
        error_msg = "B constant must be greater than 0"
    End If
    If (GeneralExp_p <= 0) Then
        bNoError = False
        error_msg = "p constant must be greater than 0"
    End If
    If Not ((Final_Type = "a") Or (Final_Type = "t") Or (Final_Type = "K")) Then
        bNoError = False
        error_msg = "Final_Type not specified correctly (""a"" = crack size, ""t"" = time, ""K"" = stress intensity factor)"
    End If
    If Not ((Output_Type = "a") Or (Output_Type = "t") Or (Output_Type = "K")) Then
        bNoError = False
        error_msg = "Output_Type not specified correctly (""a"" = crack size, ""t"" = time, ""K"" = stress intensity factor)"
    End If
    If (Final_Type = "a") Or (Final_Type = "K") Then
        If Final_Value <= 0 Then
                bNoError = False
                error_msg = "Final_Value must be greater than 0"
        End If
    End If
'
    '-------------------------------------------------------------------
    ' Get BETA FACTOR table values and perform checks
'
    beta_count = 0
    a_count = 0
    b_count = 0
'
    If IsMissing(BetaTble_a_Rng) Or IsMissing(BetaTble_B_Rng) Then
```

```
        '
        ' At least one argument is missing, therefore assume DEFAULT VALUES
        ' Redim arguments to a single array
        ' and assign the net stress ratio to unity
        '
        ReDim beta_a(1)
        ReDim beta_B(1)
        ReDim beta_slope(1)
        beta_a(1) = 1#
        beta_B(1) = 1#
        beta_slope(1) = 0#
        beta_count = 1
        a_count = 1
        b_count = 1
Else
        '
        ' If the beta factor arguments are numeric
        '
        If TypeName(BetaTble_a_Rng) = "Double" Or TypeName(BetaTble_B_Rng) = "Double" Then
            '
            ' Then only one data pair can exist, resize arrays
            '
            ReDim beta_a(1)
            ReDim beta_B(1)
            ReDim beta_slope(1)
            beta_count = 1

            If TypeName(BetaTble_a_Rng) = "Double" Then
                ' Check positive
                If BetaTble_a_Rng > 0 Then
                    beta_a(1) = BetaTble_a_Rng
                    a_count = 1
                Else
                    bNoError = False
                    error_msg = "The Beta crack size must be positive"
                    a_count = 0
                End If
            End If
            If TypeName(BetaTble_B_Rng) = "Double" Then
                ' Check positive
                If BetaTble_B_Rng > 0 Then
                    beta_B(1) = BetaTble_B_Rng
                    b_count = 1
                Else
                    bNoError = False
                    error_msg = "The Beta factor value must be positive"
                    b_count = 0
                End If
            End If
        End If
        '
        ' If the crack size table argument points to a range of cells
        '
        If TypeName(BetaTble_a_Rng) = "Range" Then
            '
            ' If only one data point can exist, then read in only one value,
            ' else read in all cell values
            '
            ' Redim the array if beta_count has NOT been set to 1
            If beta_count = 0 Then
                ReDim beta_a(BetaTble_a_Rng.Count)
            End If
            '
            ' Read in and check the crack size values
            ' Stop when a non-numeric, negative, or non-ascending value is encountered
            '
            bContinue = True
            a_count = 0
            For Each rCell In BetaTble_a_Rng
                ' Continue reading in values while they pass all checks
                If bContinue Then
                    ' Check numeric
                    If IsNumeric(rCell.Value) And Not IsEmpty(rCell.Value) Then
                        ' Check positive
                        If rCell.Value > 0 Then
                            ' Check ascending order
                            If (a_count >= 1) Then
                                If rCell.Value > beta_a(a_count) Then
                                    ' Cell value passed all checks, OK to save
                                    a_count = a_count + 1
                                    beta_a(a_count) = rCell.Value
                                Else
                                    ' Not ascending
                                    bNoError = False
                                    error_msg = "All Beta crack size cell values must be in ascending order"
                                    bContinue = False
                                End If
                            Else
                                ' 1st cell
                                ' Cell value passed all checks, OK to save
                                a_count = a_count + 1
                                beta_a(a_count) = rCell.Value
                            End If
                        Else
                            ' Not positive
                            bNoError = False
                            error_msg = "All Beta crack size cell values must be positive"
                            bContinue = False
```

```
                    End If
                Else
                    ' Not numeric
                    ' Do nothing, assume blank was found
                    ' Allow blanks to occur in the range
                    ' Do not abort, however discontinue reading in more cell values
                    bContinue = False
                End If
                ' Discontinue reading in more cell values if only one data pair can exist
                If beta_count = 1 Then
                    bContinue = False
                End If
            End If
        Next
    End If
    '
    ' If the Beta table argument points to a range of cells
    '
    If TypeName(BetaTble_B_Rng) = "Range" Then
        '
        ' If only one data point can exist, then read in only one value,
        ' else read in all cell values
        '
        ' Redim the array if beta_count has NOT been set to 1
        If beta_count = 0 Then
            ReDim beta_B(BetaTble_B_Rng.Count)
        End If
        '
        ' Read in and Beta values
        ' Stop when a non-numeric, or negative value is encountered
        '
        bContinue = True
        b_count = 0
        For Each rCell In BetaTble_B_Rng
            ' Continue reading in values while they pass all checks
            If bContinue Then
                ' Check numeric
                If IsNumeric(rCell.Value) And Not IsEmpty(rCell.Value) Then
                    ' Check positive
                    If rCell.Value > 0 Then
                        ' Cell value passed all checks, OK to save
                        b_count = b_count + 1
                        beta_B(b_count) = rCell.Value
                    Else
                        ' Not positive
                        bNoError = False
                        error_msg = "All Beta factor cell values must be positive"
                        bContinue = False
                    End If
                Else
                    ' Not numeric
                    ' Do nothing, assume blank was found
                    ' Allow blanks to occur in the range
                    ' Do not abort, however discontinue reading in more cell values
                    bContinue = False
                End If
                ' Discontinue reading in more cell values if only one data pair can exist
                If beta_count = 1 Then
                    bContinue = False
                End If
            End If
        Next
    End If
    '
    ' Set beta_count based on available data that has passed all checks
    '
    If bNoError Then
        If a_count = 0 Then
            bNoError = False
            error_msg = "No acceptable Beta crack size data available"
            StressRatio_count = 0
        End If
        If b_count = 0 Then
            bNoError = False
            error_msg = "No acceptable Beta factor data available"
            StressRatio_count = 0
        End If
    End If
    '
    If bNoError Then
        If beta_count = 1 Then
            ' Only one data pair can exist
        Else
            If a_count <= b_count Then
                beta_count = a_count
            Else
                beta_count = b_count
            End If
        End If
        '
        ' Pre-calculate the slopes of the piecewise-linear Beta function
        '
        If beta_count >= 1 Then
            '
            ReDim beta_slope(beta_count)
            '
            If beta_count = 1 Then
                beta_slope(1) = 0
```

```
                    beta_slope(0) = 0
                Else
                    For i = 1 To (beta_count - 1)
                        beta_slope(i) = (beta_B(i + 1) - beta_B(i)) / (beta_a(i + 1) - beta_a(i))
                    Next
                    ' Use linear extrapolation beyond the Beta table range
                    beta_slope(0) = beta_slope(1)
                    beta_slope(beta_count) = beta_slope(beta_count - 1)
                End If
            End If
        End If
End If
'
'----------------------------------------------------------------
' Get NET-SECTION STRESS RATIO table values and perform checks
'
StressRatio_count = 0
a_count = 0
R_count = 0
'
If IsMissing(StressTble_a_Rng) Or IsMissing(StressTble_R_Rng) Then
    '
    ' At least one argument is missing, therefore assume DEFAULT VALUES
    ' Redim arguments to a single array
    ' and assign the net stress ratio to unity
    '
    ReDim StressRatio_a(1)
    ReDim StressRatio_R(1)
    ReDim StressRatio_slope(1)
    StressRatio_a(1) = 1#
    StressRatio_R(1) = 1#
    StressRatio_slope(1) = 0#
    StressRatio_count = 1
    a_count = 1
    R_count = 1
Else
    '
    ' If the net-section stress arguments are numeric
    '
    If TypeName(StressTble_a_Rng) = "Double" Or TypeName(StressTble_R_Rng) = "Double" Then
        '
        ' Then only one data pair can exist, resize arrays
        '
        ReDim StressRatio_a(1)
        ReDim StressRatio_R(1)
        ReDim StressRatio_slope(1)
        StressRatio_count = 1

        If TypeName(StressTble_a_Rng) = "Double" Then
            ' Check positive
            If StressTble_a_Rng > 0 Then
                StressRatio_a(1) = StressTble_a_Rng
                a_count = 1
            Else
                bNoError = False
                error_msg = "The StressRatio crack size must be positive"
                a_count = 0
            End If
        End If
        If TypeName(StressTble_R_Rng) = "Double" Then
            ' Check positive
            If StressTble_R_Rng > 0 Then
                StressRatio_R(1) = StressTble_R_Rng
                R_count = 1
            Else
                bNoError = False
                error_msg = "The StressRatio factor must be positive"
                R_count = 0
            End If
        End If
    End If
    '
    ' If the crack size table argument points to a range of cells
    '
    If TypeName(StressTble_a_Rng) = "Range" Then
        '
        ' If only one data point can exist, then read in only one value,
        ' else read in all cell values
        '
        ' Redim the array if StressRatio_count has NOT been set to 1
        If StressRatio_count = 0 Then
            ReDim StressRatio_a(StressTble_a_Rng.Count)
        End If
        '
        ' Read in and check the crack size values
        ' Stop when a non-numeric, negative, or non-ascending value is encountered
        '
        bContinue = True
        a_count = 0
        For Each rCell In StressTble_a_Rng
            ' Continue reading in values while they pass all checks
            If bContinue Then
                ' Check numeric
                If IsNumeric(rCell.Value) And Not IsEmpty(rCell.Value) Then
                    ' Check zero or positive
                    If rCell.Value >= 0 Then
                        ' Check ascending order
                        If (a_count >= 1) Then
```

```
                            If rCell.Value > StressRatio_a(a_count) Then
                                ' Cell value passed all checks, OK to save
                                a_count = a_count + 1
                                StressRatio_a(a_count) = rCell.Value
                            Else
                                ' Not ascending
                                bNoError = False
                                error_msg = "All StressRatio crack size cell values must be in ascending order"
                                bContinue = False
                            End If
                        Else
                            ' 1st cell
                            ' Cell value passed all checks, OK to save
                            a_count = a_count + 1
                            StressRatio_a(a_count) = rCell.Value
                        End If
                    Else
                        ' Not positive
                        bNoError = False
                        error_msg = "All StressRatio crack size cell values must be positive"
                        bContinue = False
                    End If
                Else
                    ' Not numeric
                    ' Do nothing, assume blank was found
                    ' Allow blanks to occur in the range
                    ' Do not abort, however discontinue reading in more cell values
                    bContinue = False
                End If
                ' Discontinue reading in more cell values if only one data pair can exist
                If StressRatio_count = 1 Then
                    bContinue = False
                End If
            End If
        Next
    End If
'
' If the StressRatio table argument points to a range of cells
'
If TypeName(StressTble_R_Rng) = "Range" Then
    '
    ' If only one data point can exist, then read in only one value,
    ' else read in all cell values
    '
    ' Redim the array if StressRatio_count has NOT been set to 1
    If StressRatio_count = 0 Then
        ReDim StressRatio_R(StressTble_R_Rng.Count)
    End If
    '
    ' Read in and StressRatio values
    ' Stop when a non-numeric, or negative value is encountered
    '
    bContinue = True
    R_count = 0
    For Each rCell In StressTble_R_Rng
        ' Continue reading in values while they pass all checks
        If bContinue Then
            ' Check numeric
            If IsNumeric(rCell.Value) And Not IsEmpty(rCell.Value) Then
                ' Check positive
                If rCell.Value > 0 Then
                    ' Cell value passed all checks, OK to save
                    R_count = R_count + 1
                    StressRatio_R(R_count) = rCell.Value
                Else
                    ' Not positive
                    bNoError = False
                    error_msg = "All StressRatio factor cell values must be positive"
                    bContinue = False
                End If
            Else
                ' Not numeric
                ' Do nothing, assume blank was found
                ' Allow blanks to occur in the range
                ' Do not abort, however discontinue reading in more cell values
                bContinue = False
            End If
            ' Discontinue reading in more cell values if only one data pair can exist
            If StressRatio_count = 1 Then
                bContinue = False
            End If
        End If
    Next
End If
'
' Set StressRatio_count based on available data that has passed all checks
'
If bNoError Then
    If a_count = 0 Then
        bNoError = False
        error_msg = "No acceptable StressRatio crack size data available"
        StressRatio_count = 0
    End If
    If R_count = 0 Then
        bNoError = False
        error_msg = "No acceptable StressRatio factor data available"
        StressRatio_count = 0
    End If
```

```
            End If
        '
        If bNoError Then
            If StressRatio_count = 1 Then
                ' Only one data pair can exist
            Else
                If a_count <= R_count Then
                    StressRatio_count = a_count
                Else
                    StressRatio_count = R_count
                End If
            End If
            '
            ' Pre-calculate the slopes of the piecewise-linear StressRatio function
            '
            If StressRatio_count >= 1 Then
                '
                ReDim StressRatio_slope(StressRatio_count)
                '
                If StressRatio_count = 1 Then
                    StressRatio_slope(1) = 0
                    StressRatio_slope(0) = 0
                Else
                    For i = 1 To (StressRatio_count - 1)
                        StressRatio_slope(i) = (StressRatio_R(i + 1) - StressRatio_R(i)) / (StressRatio_a(i + 1) - StressRatio_a(i))
                    Next
                    ' Use linear extrapolation beyond the StressRatio table range
                    StressRatio_slope(0) = StressRatio_slope(1)
                    StressRatio_slope(StressRatio_count) = StressRatio_slope(StressRatio_count - 1)
                End If
            End If
        End If
    End If
End If
'
' Assign the final stop criterion and determine if the integration
' is to be performed forwards or backwards
'
If bNoError Then
    If Final_Type = "a" Then
        Final_a = Final_Value
        If Final_a > Initial_a Then
            bGrow_Forward = True
        Else
            bGrow_Forward = False
        End If
    ElseIf Final_Type = "t" Then
        Final_t = Final_Value
        If Final_t >= 0 Then
            bGrow_Forward = True
        Else
            bGrow_Forward = False
        End If
    ElseIf Final_Type = "K" Then

        Final_K = Final_Value

        ' Calculate K for the initial crack size
        a = Initial_a

        ' Find the beta table index number for the current crack size
        bn = 1
        For i = 1 To beta_count
            If a >= beta_a(i) Then
                bn = i
            End If
        Next

        beta = beta_B(bn) + beta_slope(bn) * (a - beta_a(bn))

        ' Check that the beta value is positive
        If beta <= 0 Then
            bNoError = False
            error_msg = "(1) Negative or zero beta (=" & beta & ") occurred at a = " & a & ". Integration aborted."
        Else
            Pi = 4 * Atn(1)
            Initial_K = Ref_Stress_Gross * beta * ((Pi * a) ^ 0.5)
            '
            ' Determine if K is increasing with a, then grow forwards or backwards
            ' depending on the final K relative to the initial K.
            '
            da = a / CDbl(split)
            K = Ref_Stress_Gross * beta * ((Pi * (a + da)) ^ 0.5)
            If (K > Initial_K) Then
                bInitial_K_IncreasingForward = True
                If Final_K >= K Then
                    bGrow_Forward = True
                Else
                    bGrow_Forward = False
                End If
            Else
                bInitial_K_IncreasingForward = False
                If Final_K >= K Then
                    bGrow_Forward = False
                Else
                    bGrow_Forward = True
                End If
            End If
        End If
    End If
```

```
        End If
    End If

    If bNoError Then

        ' Initialise variables

        bContinue_to_Grow = True
        Pi = 4 * Atn(1)
        t = 0#
        a = Initial_a
        Integration_Cycle = 0

        If bGrow_Forward Then
            da = a / CDbl(split)
        Else
            da = -a / CDbl(split)
        End If

        ' Find the beta table index number for the current crack size
        bn = 1
        For i = 1 To beta_count
            If a >= beta_a(i) Then
                bn = i
            End If
        Next

        ' Find the StressRatio table index number for the current crack size
        an = 1
        For i = 1 To StressRatio_count
            If a >= StressRatio_a(i) Then
                an = i
            End If
        Next

        ' Calculate the current K and dK
        beta = beta_B(bn) + beta_slope(bn) * (a - beta_a(bn))
        If beta <= 0 Then
            bNoError = False
            bContinue_to_Grow = False
            error_msg = "(2) Negative or zero beta value occurred at a = " & a & ". Integration aborted."
        Else
            K = Ref_Stress_Gross * beta * ((Pi * a) ^ 0.5)
            beta = beta_B(bn) + beta_slope(bn) * (a + da - beta_a(bn))
            dK = (Ref_Stress_Gross * beta * ((Pi * (a + da)) ^ 0.5)) - K
        End If

        ' Continue to grow crack while final criteria is not exceeded and no errors occur
        Do While (bContinue_to_Grow And bNoError)

            ' Check if maximum integration cycles has been reached
            ' if so then abort
            Integration_Cycle = Integration_Cycle + 1
            If Integration_Cycle >= Maximum_Integration_Cycles Then
                bContinue_to_Grow = False
                bNoError = False
                error_msg = "The maximum number of integration cycles has been reached. Integration aborted."
            End If

            ' Check if final criteria has been reached, if so then exit
            If Final_Type = "a" Then
                If bGrow_Forward Then
                    If a >= Final_a Then
                        bContinue_to_Grow = False
                    End If
                Else
                    If a <= Final_a Then
                        bContinue_to_Grow = False
                    End If
                End If
            ElseIf Final_Type = "t" Then
                If bGrow_Forward Then
                    If t >= Final_t Then
                        bContinue_to_Grow = False
                    End If
                Else
                    If t <= Final_t Then
                        bContinue_to_Grow = False
                    End If
                End If
            ElseIf Final_Type = "K" Then
                If Final_K > Initial_K Then
                    If dK > 0 Then
                        If K >= Final_K Then
                            bContinue_to_Grow = False
                        End If
                    Else
                        ' K is decreasing or has flattened out,
                        ' indicating Final_K may not be reached.
                        ' Continue integration incase K begins to
                        ' increase again, and only abort if
                        ' K drops below the Initial_K.
                        If K <= Initial_K Then
                            bContinue_to_Grow = False
                            bNoError = False
                            error_msg = "K has become less than the Initial_K. Suspect Final_K will not be reached. Integration
aborted."
                        End If
```

```
                End If
            ElseIf Final_K < Initial_K Then
                If dK < 0 Then
                    If K <= Final_K Then
                        bContinue_to_Grow = False
                    End If
                Else
                    ' K is increasing or has flattened out,
                    ' indicating Final_K may not be reached.
                    ' Continue integration incase K begins to
                    ' decrease again, and only abort if
                    ' K increases above the Initial_K.
                    If K >= Initial_K Then
                        bContinue_to_Grow = False
                        bNoError = False
                        error_msg = "K (= " & K & ") has become greater than the Initial_K (= " & Initial_K & "). Suspect Final_K
will not be reached. Integration aborted."
                    End If
                End If
            Else
                ' Final_K = Initial_K
                bContinue_to_Grow = False
            End If
        Else
            '
            ' Debug trap no. 1
            '
            ' Code shouldn't be able to get here due to input checks above.
            ' Trap to prevent integrating all the way to the maximum cycles.
            bContinue_to_Grow = False
            bNoError = False
            error_msg = "Error in function code input checks. Debug trap no. 1"
        End If

        ' Integrate next increment

        If bContinue_to_Grow And bNoError Then

            ' Set delta_a as a fraction of the current crack size
            da = a / CDbl(split)
            If Not bGrow_Forward Then
                da = -da
            End If

            ' If a+da crosses the next beta(a) then adjust da to hit beta_a perfectly
            If bGrow_Forward Then
                If bn < beta_count Then
                    If a < beta_a(bn + 1) And a + da > beta_a(bn + 1) Then
                        da = beta_a(bn + 1) - a
                    End If
                End If
            Else
                If bn >= 1 Then
                    If a > beta_a(bn) And a + da < beta_a(bn) Then
                        da = beta_a(bn) - a
                    End If
                End If
            End If

            ' If a+da > Final_a then adjust da to hit Final_a perfectly
            If Final_Type = "a" Then
                If (bGrow_Forward And a + da > Final_a) Or ((Not bGrow_Forward) And a + da < Final_a) Then
                    da = Final_a - a
                    bContinue_to_Grow = False
                End If
            End If

            ' If K+dK > Final_K then adjust da to hit Final_K perfectly
            If Final_Type = "K" Then
                beta = beta_B(bn) + beta_slope(bn) * (a - beta_a(bn))
                K = Ref_Stress_Gross * beta * (Pi * a) ^ 0.5
                beta = beta_B(bn) + beta_slope(bn) * (a + da - beta_a(bn))
                dK = (Ref_Stress_Gross * beta * (Pi * (a + da)) ^ 0.5) - K
                If (Final_K > Initial_K And K + dK > Final_K) Or _
                   (Final_K < Initial_K And K + dK < Final_K) Then
                    ' Search for da to hit Final_K
                    ' Use Bi-section method with 10000 iterations
                    ' Note: The Newton-Raphson method was trialed and found to be too sensitive
                    da2 = da / 2#
                    For i = 1 To 10000
                        ' Assume beta constant
                        beta = beta_B(bn) + beta_slope(bn) * (a + da - beta_a(bn))
                        K = Ref_Stress_Gross * beta * (Pi * (a + da)) ^ 0.5
                        If (Final_K > Initial_K And K > Final_K) Or _
                           (Final_K < Initial_K And K < Final_K) Then
                            da = da - da2
                            da2 = da2 / 2#
                        Else
                            da = da + da2
                            da2 = da2 / 2#
                        End If
                    Next
                    bContinue_to_Grow = False
                End If
            End If


            ' Calculate average beta (assume constant over da)
```

```
            beta = beta_B(bn) + beta_slope(bn) * (a + da / 2# - beta_a(bn))

            ' Calculate average Net reference stress (assume constant over da)
            StressRatio = StressRatio_R(an) + StressRatio_slope(an) * (a + da / 2# - StressRatio_a(an))

            ' Check that the beta and StressRatio values are positive and da is nonzero
            If beta <= 0 Or StressRatio <= 0 Or da = 0 Then
                bNoError = False
                bContinue_to_Grow = False
                If beta <= 0 Then
                    error_msg = "(3) Negative or zero beta value occurred at a = " & a & ". Integration aborted."
                ElseIf StressRatio <= 0 Then
                    error_msg = "(3) Negative or zero StressRatio value occurred at a = " & a & ". Integration aborted."
                Else
                    error_msg = "da is zero at a = " & a & ". Integration aborted."
                End If
            Else

            Ref_Stress_Net = Ref_Stress_Gross * StressRatio

                ' Calculate dt (use different integrals depending if p = 2 or not)
                If GeneralExp_p = 2 Then
                    dt = Log((a + da) / a) / (GeneralCoeff_H * Pi * ((beta * Ref_Stress_Gross) ^ 2#) * (Ref_Stress_Net ^
GeneralExp_q))
                Else
                    dt = (2# / (GeneralCoeff_H * (2# - GeneralExp_p) * (Ref_Stress_Net ^ GeneralExp_q))) * ((Ref_Stress_Gross *
beta * (Pi ^ 0.5)) ^ (-GeneralExp_p)) * _
                            (((a + da) ^ (1# - GeneralExp_p / 2#)) - ((a) ^ (1# - GeneralExp_p / 2#)))
                End If

                ' If t+dt > Final_T then adjust da to hit Final_t perfectly
                If Final_Type = "t" Then
                    If (bGrow_Forward And t + dt > Final_t) Or ((Not bGrow_Forward) And t + dt < Final_t) Then
                        ' Iterate 10000 times to keep beta and StressRatio up-to-date
                        dt = Final_t - t
                        For i = 1 To 10000
                            ' Assume beta and StressRatio constant
                            beta = beta_B(bn) + beta_slope(bn) * (a + da / 2# - beta_a(bn))
                            StressRatio = StressRatio_R(an) + StressRatio_slope(an) * (a + da / 2# - StressRatio_a(an))
                            Ref_Stress_Net = Ref_Stress_Gross * StressRatio
                            If GeneralExp_p = 2 Then
                                da = a * Exp(GeneralCoeff_H * Pi * ((Ref_Stress_Gross * beta) ^ 2#) * (Ref_Stress_Net ^
GeneralExp_q) * dt) - a
                            Else
                                da = (a ^ (1# - GeneralExp_p / 2#) + dt * GeneralCoeff_H * (1# - GeneralExp_p / 2#) * _
                                    ((Ref_Stress_Gross * beta * (Pi ^ 0.5)) ^ GeneralExp_p) * (Ref_Stress_Net ^ GeneralExp_q)) ^ (1# /
(1# - GeneralExp_p / 2#)) - a
                            End If
                        Next
                        bContinue_to_Grow = False
                    End If
                End If

                ' Increment time
                t = t + dt

                ' Increment crack size, but do error check first
                a = a + da

                ' Check to see if the crack size is extremely large or small
                ' If it is then stop integrating
                If a > 1E+100 Then
                    a = 1E+100
                    bContinue_to_Grow = False
                ElseIf a < 1E-100 Then
                    a = 1E-100
                    bContinue_to_Grow = False
                End If

                ' Check if the beta table index needs to be increased or
                ' decreased for the next iteration.
                If bGrow_Forward Then
                    If bn < beta_count Then
                        If a >= beta_a(bn + 1) Then
                            bn = bn + 1
                        End If
                    End If
                Else
                    If bn > 1 Then
                        If a < beta_a(bn) Then
                            bn = bn - 1
                        End If
                    End If
                End If

                ' Check if the StressRatio table index needs to be increased or
                ' decreased for the next iteration.
                If bGrow_Forward Then
                    If an < StressRatio_count Then
                        If a >= StressRatio_a(an + 1) Then
                            an = an + 1
                        End If
                    End If
                Else
                    If an > 1 Then
                        If a < StressRatio_a(an) Then
                            an = an - 1
                        End If
                    End If
```

```
                End If
            End If

            ' Calculate the current stress intensity factor
            beta = beta_B(bn) + beta_slope(bn) * (a - beta_a(bn))
            K = Ref_Stress_Gross * beta * ((Pi * a) ^ 0.5)
        End If
    End If
    Loop
End If


'Output result or error message
If bNoError Then
    ' Return output
    If Output_Type = "a" Then
        Output_Value = a
    ElseIf Output_Type = "t" Then
        Output_Value = t
    ElseIf Output_Type = "K" Then
        Output_Value = K
    Else
        Output_Value = "Error: Output type not specified correctly"
    End If
    INTEGRATEGENERAL = Output_Value
Else
    ' Return error message
    INTEGRATEGENERAL = "Input error: " & error_msg
End If

Exit Function
ErrorHandler:
INTEGRATEGENERAL = "Program execution error: " & Error()

End Function
```

Appendix E: Scaling methods for supplementary adjustment of EBA model parameters

The following methods are relevant for an exponent-type CGR model in the form:

$$CGR = C \times (similitude\ parameter)^E$$

Where *CGR* is the crack growth rate, *C* is the coefficient parameter and *E* is the exponent parameter. This exponent-type model produces a linear relationship between the logarithm of the CGR and the logarithm of the similitude parameter.

E.1 Method 1: Scaling for a *constant offset* exponent relationship between CGR and the similitude parameter

A constant exponent ratio relationship is given when the logarithm of the CGRs for two conditions (*e.g.,* load spectra) displays a constant offset over a range of the logarithm of the similitude parameter such as shown in Figure E.1. This scaling relationship is given by:

$$E2 = E1 \qquad\qquad (E.1.a)$$

where *E1* and *E2* are the exponent values for condition 1 and 2 respectively, and

$$C2 = C1 \times \text{scale\_factor} \qquad\qquad (E.1.b)$$

where *C1* and *C2* are the coefficient values for condition 1 and 2 respectively.

It is postulated that the scale factor may be consistent between methods. That is, the scale factor derived from one method (*e.g.,* test) may be the same as that derived from another (*e.g.,* analysis). This gives a prediction of the parameters for the unknown condition 2 based on scaling the coefficient parameter of the known condition 1, using a scale factor derived from an analysis as follows:

from (E.1.a) $$E_{Condition2} = E_{Condition1} \qquad\qquad (E.1.c)$$

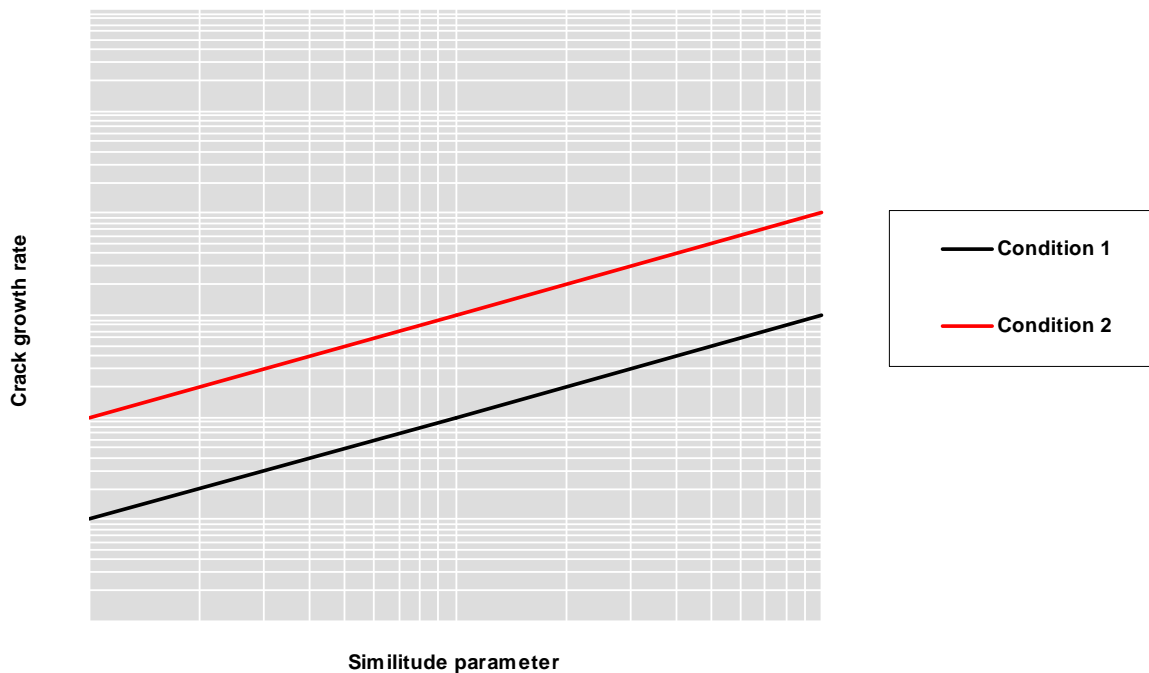from (E.1.b) $$C_{Condition2} = C_{Condition1} \times \left(\frac{C_{Condition2}}{C_{Condition1}} \right)_{analysis} \qquad\qquad (E.1.d)$$

*Figure E.1: Schematic of a **constant offset** logarithmic ratio relationship between the CGR and the similitude parameter.*

E.2 Method 2: Scaling for an *linear offset* exponent relationship between CGR and the similitude parameter

A linear exponent ratio relationship is given when the logarithm of the CGRs for two conditions (*e.g.,* load spectra) displays a linearly varying offset over a range of the logarithm of the similitude parameter values such as shown in Figure E.2. This scaling relationship is given by:

$$E2 = E1 + \textit{offset\_factor} \qquad \text{(E.2.a)}$$

where *E1* and *E2* are the exponent values for condition 1 and 2 respectively, and

$$C2 = C1 \times \textit{scale\_factor} \qquad \text{(E.2.b)}$$

where *C1* and *C2* are the coefficient values for condition 1 and 2 respectively.

It is postulated that the offset and scale factors may be consistent between methods. That is, the offset and scale factors derived from one method (*e.g.,* test) may be the same as that derived from another (*e.g.,* analysis). This gives a prediction of the parameters for the unknown condition 2 based on offsetting the exponent, and scaling the coefficient parameters of the known condition 1, using a factors derived from an analysis as follows (for derivations see [7, 38]) :

from (E.2.a) $\qquad E_{Condition2} = E_{Condition1} + \left(E_{Condition2} - E_{Condition1} \right)_{analysis} \qquad \text{(E.2.c)}$

from (E.2.b) $\qquad C_{Condition2} = C_{Condition1} \times \left(\dfrac{C_{Condition2}}{C_{Condition1}} \right)_{analysis} \qquad \text{(E.2.d)}$
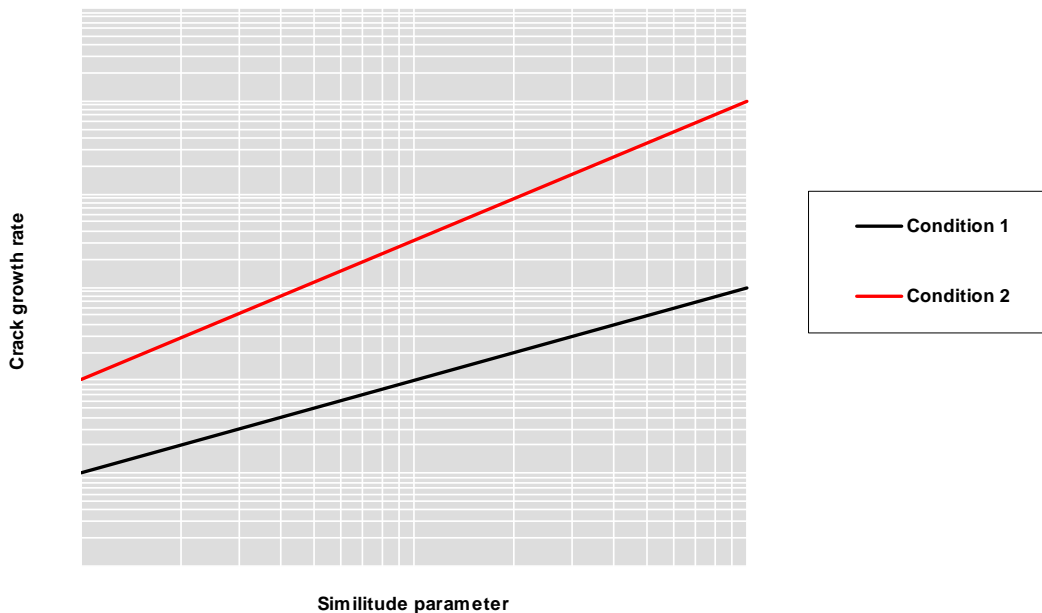

*Figure E.2: Schematic of a **linear offset** logarithmic ratio relationship between the CGR and the similitude parameter.*

Page classification: UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION **DOCUMENT CONTROL DATA**	1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

| 2. TITLE

Guide on the Effective Block Approach for the Fatigue Life Assessment of Metallic Structures | 3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

Document   (U)
Title   (U)
Abstract   (U) |

| 4. AUTHOR(S)

Marcus McDonald | 5. CORPORATE AUTHOR

DSTO Defence Science and Technology Organisation
506 Lorimer St
Fishermans Bend Victoria 3207 Australia |

6a. DSTO NUMBER DSTO-TR-2850	6b. AR NUMBER AR-015-630	6c. TYPE OF REPORT Technical Report	7. DOCUMENT DATE January 2013

8. FILE NUMBER 2010/1064444/1	9. TASK NUMBER 07/048	10. TASK SPONSOR DGTA	11. NO. OF PAGES 78	12. NO. OF REFERENCES 60

DSTO Publications Repository http://dspace.dsto.defence.gov.au/dspace/	14. RELEASE AUTHORITY Chief, Air Vehicles Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS	Yes

18. DSTO RESEARCH LIBRARY THESAURUS

Fatigue Tests; Fatigue Loads; Cracking (Fracturing); Crack Propagation; Fatigue Lifing

19. ABSTRACT
Simpler empirical characterisation of the mechanics of metal fatigue crack growth can offer an alternative to physics-based models, where the latter can be challenged by complex material behaviour of commercial metals when loaded by variable amplitude spectra. Such characterisation is anticipated to be more appropriate for cases where the loading consists of repeating blocks of spectra or other scenarios that result in near steady-state crack growth rates. This report examines the many types of characteristic block approaches available, and recommends a particular type that is considered suitable for fatigue life assessments, herein titled the Effective Block Approach, or EBA. Guidance is given on methods and implementation; and several case studies are reviewed where the EBA has been used to assess the life of RAAF airframes. In addition to computational efficiency, significant accuracy gains were achieved when the approach was underpinned by representative coupon tests. This capability is anticipated to give the RAAF more options to better assess and manage the safety, availability and cost of ownership of its air vehicles.

Page classification: UNCLASSIFIED