

Scalability of Cohesive Fatigue Analyses Using Explicit Solvers

Frank A. Leone* and Carlos G. Dávila†
NASA Langley Research Center, Hampton, Virginia, 23681

A cohesive fatigue law has been integrated into a constitutive material model compatible with an explicit finite element solver. The cohesive fatigue model response is based on engineering approximations of the endurance limit and the Goodman diagram. This approach can predict stress-life diagrams for crack initiation, the Paris law regime, and transient effects of crack initiation and stable tearing. Simplified cyclic loading is utilized so that the applied load (or displacement) corresponds to the peak load of a fatigue cycle. Loads are held constant during fatigue while damage develops with increasing solution increments. An automatically-calculated ratio of fatigue cycles per solution increment controls the rate of damage growth, ensuring that damage growth is modeled with a sufficient minimum number of increments and damage growth advances to a minimum desired extent within the explicit analysis step time. The compatibility with an explicit finite element solver enables the analysis of structures that are computationally intractable for implicit finite element solvers. Scalability studies are conducted for geometrically nonlinear problems involving fiber-reinforced composite structures that exhibit fatigue damage growth of interacting matrix cracks and delaminations.

I. Introduction

As part of the NASA Advanced Composites Project (ACP), a verification and validation framework was developed and applied to a set of progressive damage analysis (PDA) methods for fiber-reinforced composite materials [1], with the goals of increasing confidence in the computational tools and advancing the state-of-the-art through additional method improvements. One of the focus problems for the ACP was skin-stiffener disbond in a post-buckled stiffened panel, in which networks of interacting matrix cracks and delaminations develop and contribute to failure. A variety of PDA methods were exercised, including both continuum and discrete damage mechanics methods, methods compatible with implicit and explicit finite element (FE) solvers, and methods developed for static and fatigue loading conditions. The CompDam material model [2] was one of the methods selected for evaluation and further development during the ACP. At the onset of the project, the CompDam material model had no existing fatigue prediction capabilities, and the evaluation of the model was limited to the statically loaded validation problems, including seven-point bending (7PB) [3], single-stringer [4], and multi-stringer compression [5] specimens. The expansion of CompDam's capability to include matrix and delamination fatigue crack growth was identified as a technical development need.

The CompDam material model is a continuum damage mechanics (CDM) method that represents matrix cracks as cohesive planes embedded in a volume of unidirectional fiber-reinforced material. The local deformation gradient tensor is scaled by the characteristic dimensions of the element and decomposed into a cohesive displacement jump and a bulk material deformation gradient tensor [6]. The deformation decomposition is performed so as to maintain equilibrium across the inserted cohesive plane. The constitutive response of the bulk material is generally linear-elastic orthotropic, though pre-peak nonlinearity can be activated. The constitutive response of the embedded cohesive plane is a conventional mixed-mode linear-softening traction-separation law. CompDam is implemented as a VUMAT user material subroutine for use with Abaqus/Explicit [7]. The material model can represent either individual fiber-reinforced plies when used with three-dimensional solid elements or interlaminar interfaces when used with cohesive elements. The same cohesive law logic is used for both intralaminar matrix cracks and interlaminar delaminations. Therefore, the integration of fatigue capability into the material model cohesive law would enable the modeling of matrix cracks, delaminations, and their interactions.

A novel cohesive fatigue model developed by Dávila [8] is based on engineering approximations of the endurance limit and the Goodman diagram. The method requires specification of the typical strength and fracture toughness material properties for cohesive traction-separation laws and information concerning the stress-life behavior of the material. The stress-life behavior of the material is characterized by the endurance limit, ϵ , and the number of fatigue

*Research Aerospace Engineer, Durability, Damage Tolerance, and Reliability Branch, Mail Stop 188E

†Senior Research Aerospace Engineer, Durability, Damage Tolerance, and Reliability Branch, Mail Stop 188E, AIAA Member

cycles to the endurance, γ . The endurance limit is modified depending on the current mode mixity and stress ratio of the cyclic loading. The method is capable of predicting the Paris law regime, as well as the transient effects of crack initiation and stable tearing, for a variety of fatigue loading conditions. The model was initially developed for implicit analyses and was implemented as a UMAT user material subroutine for cohesive elements for use with Abaqus/Standard.

In Dávila's cohesive fatigue model, the cohesive damage variable evolves as a function of fatigue cycles, with the rate of damage evolution depending on the current displacement jump. An empirical two-parameter function for incremental fatigue damage as a function of fatigue cycles was proposed in Ref. [8]. Additional definitions for the incremental fatigue damage function are proposed and evaluated in Ref. [9]. The evaluations included test/analysis correlations for double cantilever beam (DCB) and mixed-mode bending fracture characterization specimens and three-point bend (3PB) doubler validation specimens from the ACP. Of the evaluated incremental fatigue damage functions, the four-parameter CF20 function had the most success in matching experimentally observed behaviors while requiring a minimal amount of experimental data.

In this paper, Dávila's CF20 cohesive fatigue law is integrated into the CompDam material model. Analyses that take advantage of the scalability of explicit FE solvers are conducted, including size effect studies on laboratory-scale elements that are computationally intractable with the implicit implementation. The paper is organized as follows. Section II presents the integration of the cohesive fatigue model into an explicit constitutive material model. Section III includes a verification of the proper functioning of the explicit fatigue model in terms of cohesive traction-separation laws and stress-life behavior. Section IV includes a validation of the explicit implementation in which double cantilever beam (DCB) and three-point bend (3PB) doubler specimens are analyzed. Section V presents additional parametric 3PB doubler analyses in which specimen width is increased to vary the computational cost, following the approaches laid-out in Refs. [10, 11]. Discussions follow on the advantages of performing fatigue analyses in an explicit framework regarding computational scalability and the additional high-fidelity and larger-scale analyses that are enabled by the explicit implementation of the cohesive fatigue law.

II. Cohesive Material Model

A. Quasi-Static Loading

The constitutive response of the cohesive law is a linear-softening traction-separation law:

$$\sigma = \begin{cases} K_n(1-d)\delta_n, & \text{if } \delta_n \geq 0 \\ K_n\delta_n, & \text{otherwise} \end{cases} \quad (1)$$

$$\tau = K_s(1-d)\delta_s \quad (2)$$

where σ is the normal traction, τ is the shear traction, the δ terms represent the cohesive displacement jumps, the K terms represent the numerical penalty stiffness, and d is a scalar damage state variable, Fig. 1. The subscripts n and s represent the crack normal and shear directions. The Mode I and Mode II strength and fracture toughness properties are σ_c , τ_c , G_{Ic} , and G_{IIc} , respectively.

Damage initiates when the current cohesive displacement jump, δ , reaches the failure initiation displacement jump, δ_0 . After damage initiation, the damage state variable evolves from zero (representing an undamaged state) to one (representing a fully damaged state). Damage is fully developed when δ reaches the final failure displacement jump δ_f .

The current damage variable, d , which affects the penalty stiffness terms in Eq. 1, is a function of the current displacement jump, δ , and δ_0 and δ_f :

$$d = \frac{\delta_f(\delta - \delta_0)}{\delta(\delta_f - \delta_0)} \quad (3)$$

An alternative damage variable can be expressed in terms of normalized fracture energy dissipation with the form:

$$\bar{d} = \frac{d\delta_0}{(1-d)\delta_f + d\delta_0} \quad (4)$$

which has the benefit of being independent of the numerical penalty stiffness.

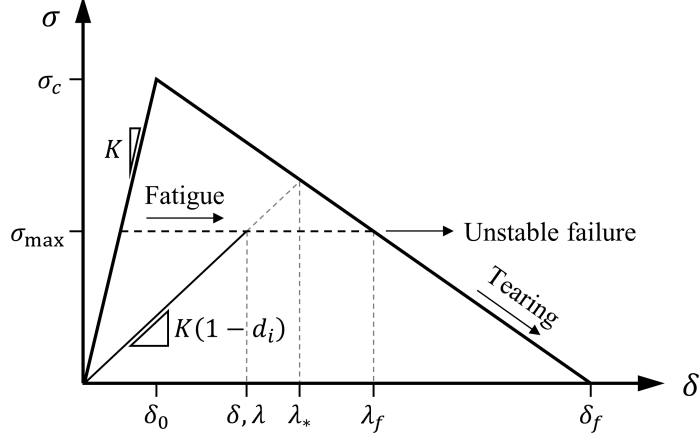


Fig. 1 Cohesive fatigue traction-separation law for constant-traction fatigue growth.

The displacement jump values for damage initiation and final failure are functions of the strength and fracture toughness material properties, the penalty stiffness terms, and the mode mixity, B , following the derivation in Ref. [12]. Mode mixity is defined according to the Mode II energy release rate over the total energy release rate through:

$$B = \begin{cases} \frac{K_s \delta_s^2}{K_s \delta_s^2 + K_n \delta_n^2}, & \text{if } \delta_n > 0 \\ 1, & \text{otherwise} \end{cases} \quad (5)$$

For pure normal and shear loading conditions, the initiation displacement jumps are:

$$\begin{aligned} \delta_{0n} &= \frac{\sigma_c}{K_n} \\ \delta_{0s} &= \frac{\tau_c}{K_s} \end{aligned} \quad (6)$$

and the final failure displacement jumps are:

$$\begin{aligned} \delta_{fn} &= \frac{2G_{Ic}}{\sigma_c} \\ \delta_{fs} &= \frac{2G_{IIc}}{\tau_c} \end{aligned} \quad (7)$$

Under mixed-mode loading conditions where $0 < B < 1$, the cohesive response is defined according to the Benzeggagh-Kenane (BK) law [13]. The mixed-mode initiation displacement jump δ_0 is:

$$\delta_0 = \sqrt{\left(\delta_{0n}^2 + \left(\delta_{0s}^2 \frac{K_s}{K_n} - \delta_{0n}^2 \right) B^{\eta_{BK}} \right) \left(1 + B \left(\frac{K_n}{K_s} - 1 \right) \right)} \quad (8)$$

and the mixed-mode final failure displacement jump δ_f is:

$$\delta_f = \frac{1}{\delta_0} \left(\delta_{0n} \delta_{fn} + \left(\delta_{0s} \delta_{fs} \frac{K_s}{K_n} - \delta_{0n} \delta_{fn} \right) B^{\eta_{BK}} \right) \left(1 + B \left(\frac{K_n}{K_s} - 1 \right) \right) \quad (9)$$

where η_{BK} is the exponent of the BK fracture criterion.

B. Fatigue Loading

The CF20 incremental fatigue damage function requires information regarding the endurance limit ϵ and the fatigue cycles to endurance γ . ϵ is the endurance limit for fully reversed ($R = -1$) axial loading. Shear loading representative of Mode II and mixed-mode loading conditions can be accounted for in the endurance limit through:

$$\epsilon_{\text{mixed}} = \epsilon(1 - 0.42B), \quad (10)$$

Table 1 CF20 cohesive fatigue model inputs

Parameter	Description	Value
γ	Number of fatigue cycles to reach endurance	1×10^7
ϵ	Endurance limit	0.2
η	Brittleness	0.95
p	Curve-fitting parameter to match Paris Law results	β

and different stress ratios, R , can be accounted for with the correction:

$$E = \frac{2\epsilon_{\text{mixed}}}{\epsilon_{\text{mixed}} + 1 + R(\epsilon_{\text{mixed}} - 1)}, \quad (11)$$

where E is referred to as the relative endurance limit.

The CF20 incremental fatigue damage function is written in terms of normalized energy dissipation and has the assumed shape:

$$\Delta \bar{d}_{\text{fat}} = \frac{1}{\gamma} \frac{(1 - \bar{d}_{i-1})^{\beta-p}}{E^{\beta}(p+1)} \left(\frac{\lambda}{\lambda_*} \right)^{\beta} \Delta N_i \quad (12)$$

where the subscript i represents the current solution increment and ΔN_i is the number of fatigue cycles represented by the current solution increment. λ is the peak displacement jump within a fatigue cycle, and λ_* is the displacement jump at which static damage progression would occur given the current mode mixity and prior damage state, as shown in Fig. 1. The ratio $\frac{\lambda}{\lambda_*}$ is referred to as the relative displacement jump, and can be expressed as:

$$\frac{\lambda}{\lambda_*} = \frac{\delta_i}{(1 - \bar{d}_{i-1})\delta_0 + \bar{d}_{i-1}\delta_f} \quad (13)$$

β is a function of E and a “brittleness” input parameter η :

$$\beta = \frac{-7\eta}{\log E} \quad (14)$$

p is an additional parameter from the fatigue damage accumulation law of Allegri and Wisnom [14] that affects the predicted prefactor of the Paris law. In total, the incremental fatigue damage function in Eq. 12 has four input parameters, which are listed in Table 1 along with their values used in the analyses presented in this paper.

The current damage state is the sum of the previous damage state and current incremental fatigue damage:

$$\bar{d}_i = \bar{d}_{i-1} + \Delta \bar{d}_{\text{fat}} \quad (15)$$

Converting from the normalized energy dissipation form to the cohesive damage state variable form for compatibility with the rest of the material model logic yields:

$$d_i = \frac{\delta_f \bar{d}_i}{\delta_0(1 - \bar{d}_i) + \delta_f \bar{d}_i} \quad (16)$$

If damage evolution occurs due to both static and fatigue loading, the maximum of the current static and current fatigue damage states is used.

C. Representation of Fatigue Cycles

The cohesive fatigue model simulates the accumulation of fatigue damage via simplified cyclic loading (SCL) in which either maximum applied loads or displacements are held constant while fatigue cycles pass with step time. A fatigue analysis with SCL requires at least two analysis steps. In the first step(s), the structure is quasi-statically loaded to the peak of a fatigue cycle. Then, in the following fatigue analysis step, the applied loads or displacements are held constant while fatigue damage accumulates.

In the implicit implementation of the cohesive fatigue model, the analysis pseudo-time represents fatigue cycles [8]. In an implicit analysis, the time increment, Δt , automatically increases or decreases as necessary to reach converged

Table 2 User-defined solution parameters controlling the evolution of ΔN_i

Parameter	Description	Default Value
$(\Delta N_i)_{\text{initial}}$	fatigue cycles per increment, initial	1×10^{-4}
$(\Delta N_i)_{\text{max}}$	fatigue cycles per increment, max.	1×10^5
$(\Delta N_i)_{\text{min}}$	fatigue cycles per increment, min.	1×10^{-5}
$(\Delta N_i)_{\text{mod}}$	percent change modifier of ΔN_i	1×10^{-1}
$(\Delta \tilde{d}_{\text{fat}})_{\text{max}}$	incremental energy dissipation, max.	1×10^{-4}
$(\Delta \tilde{d}_{\text{fat}})_{\text{min}}$	incremental energy dissipation, min.	5×10^{-6}

solutions while the fatigue damage accumulates. Smaller time increments can be used when damage growth is rapid and convergence is more difficult to achieve, and larger time increments can be used when damage growth is more gradual and steady. During an explicit analysis step, the time increment is capped at the minimum stable time increment, Δt_{stable} . The total number of solution increments is also limited by the eventual accumulation of significant numerical error. These two limits on the time incrementation must be considered while designing the representation of fatigue cycles in the explicit analysis steps. Three options were considered with regard to how to represent the fatigue cycles in an explicit analysis step, starting with the approach adopted in the implicit implementation:

1) **Constant fatigue cycles per delta time**, $\Delta N / \Delta t = c_1$

This is the approach used in the implicit implementation, and would be simple to implement. Negative factors associated with this approach include the possibility that Δt_{stable} can change during the fatigue analysis step due to either fatigue damage accumulation or other nonlinear effects. Because the explicit solver cannot raise the time increment above Δt_{stable} , the solution cannot advance more quickly when damage growth is slow and inefficiently spends many solution increments modeling periods of slow damage growth. In addition, the selection of the constant value c_1 requires a good estimate of the fatigue life before performing the analysis.

2) **Constant fatigue cycles per solution increment**, $\Delta N_i = c_2$

Making fatigue cycles depend on the solution increment, Δi , rather than Δt allows fatigue damage to accumulate independent of Δt_{stable} . However, a linear distribution of solution increments over the fatigue cycle history is maintained. A constant ratio of fatigue cycles per solution increment requires a compromise between using too few increments during periods of rapid damage growth and not reaching the fatigue life before numerical errors accumulate to unacceptable levels. Again, a good estimate of the fatigue life would be needed to select a value for c_2 that may produce acceptable results.

3) **Variable fatigue cycles per solution increment**, $\Delta N_i = f(\cdot)$

Allowing the ratio of fatigue cycles per solution increment to change during the analysis addresses the concerns of the prior two options: it avoids coarse representation of rapid changes in damage states; fewer increments are wasted modeling periods of slow damage growth; the model can adapt to the current damage accumulation rate; and *a priori* knowledge of the fatigue life is not required. A negative factor with this option is the relative complexity in implementation, requiring that a procedure, $f(\cdot)$, is developed that allows ΔN_i to change as needed.

In order to allow for computationally efficient fatigue analyses without *a priori* knowledge of the fatigue life, a numerical procedure was developed to define ΔN_i as a function of $\Delta \tilde{d}_{\text{fat}}$ (Eq. 12). Several user-defined parameters are used to control the evolution of ΔN_i during the analysis, each of which is listed and defined in Table 2 and illustrated in Fig. 2. The parameter $(\Delta N_i)_{\text{initial}}$ defines the initial ratio of fatigue cycles per solution increment. At each material point and during every solution increment, $\Delta \tilde{d}_{\text{fat}}$ is calculated. At the end of each solution increment, if the maximum value of $\Delta \tilde{d}_{\text{fat}}$ for all material points is outside the bounds set by $(\Delta \tilde{d}_{\text{fat}})_{\text{min}}$ and $(\Delta \tilde{d}_{\text{fat}})_{\text{max}}$, ΔN_i is adjusted for the next solution increment. If fatigue damage is progressing too slowly with respect to solution increments, ΔN_i is multiplied by $(1 + (\Delta N_i)_{\text{mod}})$, up to a maximum value of $(\Delta N_i)_{\text{max}}$. If, instead, fatigue damage is progressing too rapidly, ΔN_i is divided by $(1 + (\Delta N_i)_{\text{mod}})$, down to a minimum value of $(\Delta N_i)_{\text{min}}$.

At the beginning of a fatigue analysis step, ΔN_i typically increases after every increment until either damage is progressing by $(\Delta \tilde{d}_{\text{fat}})_{\text{min}}$ somewhere in the model or $(\Delta N_i)_{\text{max}}$ is reached. During this period (shaded green in Fig. 2), ΔN and N are functions of i and the solution parameters with the forms:

$$\Delta N(i) = (\Delta N_i)_{\text{initial}} [1 + (\Delta N_i)_{\text{mod}}]^{i-1} \quad (17)$$

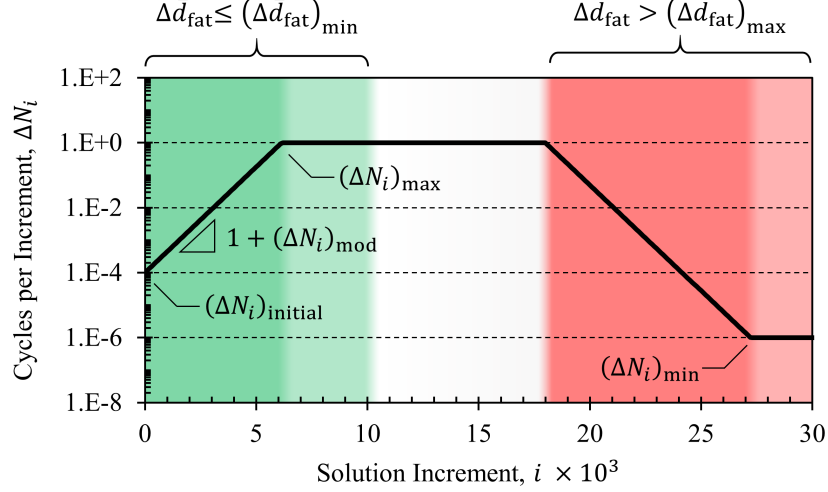


Fig. 2 Solution parameters controlling the evolution of ΔN_i .

$$N(i) = \frac{(\Delta N_i)_{\text{initial}}}{(\Delta N_i)_{\text{mod}}} ([1 + (\Delta N_i)_{\text{mod}}]^i - 1) \quad (18)$$

These functions for $\Delta N(i)$ and $N(i)$ can be used to select solution parameters that will, for example, achieve a certain ratio of fatigue cycles per solution increment after a certain number of solution increments.

III. Verification

A series of single-element FE analyses are used to verify the performance of the material model for pure Mode I and pure Mode II loading conditions. The objective of these verification analyses is to ensure that the shapes of constitutive traction-separation laws and the stress-life plots replicate their designed, intended behaviors. Eight-node COH3D8 cohesive elements with in-plane dimensions of 0.1 mm in the x and y directions and zero initial thickness in the z direction are used in the simulations. The nominal IM7/8552 interlaminar properties from Table 3 are used. Two analysis steps are used in each analysis. The first analysis step includes the quasi-static application of equal loads to the four $+z$ -surface nodes of the cohesive element, in the $+z$ direction for axial loading and in the $+x$ direction for shear loading. The displacements of $+z$ nodes are constrained in the unloaded directions, and the $-z$ nodes are fixed in all three directions. During the second analysis step, load is held constant while fatigue damage accumulates. The verification analyses are performed using the Abaqus/Explicit 2019 solver with double precision. Geometric nonlinearity is enabled in all analysis steps. Mass scaling is applied so as to maintain $\Delta t_{\text{stable}} \geq 1 \times 10^{-7}$ seconds at all increments.

Traction-displacement histories for six analyses are shown in Fig. 3, including static loading, fatigue loading with $\sigma_{\text{max}}/\sigma_c = 0.5$, and fatigue loading with $\sigma_{\text{max}}/\sigma_c = 0.8$ for both normal and shear loading conditions. The model traction-displacement histories match the designed behavior (Fig. 1) of linearly loading up to the applied traction level, maintaining constant traction while damage evolves with increasing fatigue cycles, and then unloading when the corresponding static unloading curve is reached. When the static unloading curve of the cohesive law is reached in these explicit models with constant applied load, the material can no longer sustain the applied load and fails abruptly. The applied load remains constant during failure, and dynamic reaction forces develop that counteract the applied load such that the cohesive traction in the material model can still follow the constitutive unloading curve, albeit coarsely, as shown in Fig. 3.

Single-element, constant-maximum-stress analyses are performed for four different values of R (-1 , 0 , 0.1 , and 0.5) to evaluate the ability of the models to reproduce the stress-life diagrams which correspond to the assumptions made in deriving Eq. 12. The expected fatigue life N_f can be derived by integrating Eq. 12 with respect to the damage variable over the full range of fatigue damage accumulation, which yields:

$$N_f = \gamma E^\beta \frac{\sigma_{\text{max}}}{\sigma_c}^{-\beta} \left(1 - \frac{\sigma_{\text{max}}}{\sigma_c}^{p+1} \right) + 1 \quad (19)$$

Table 3 IM7/8552 interface material properties for use with a single (nominal) layer and with two superposed (initiation + bridge) layers of cohesive elements [8]

Property	Units	Nominal	Initiation	Bridge
σ_c	MPa	80.1	80.0	0.8
τ_c	MPa	97.6	99.0	1.0
G_{Ic}	N-mm/mm ²	0.240	0.214	0.140
G_{IIc}	N-mm/mm ²	0.739	0.59	0.193
η_{BK}	-	2.1	2.1	3.0
K_n	MPa/mm	200,000	200,000	2,000

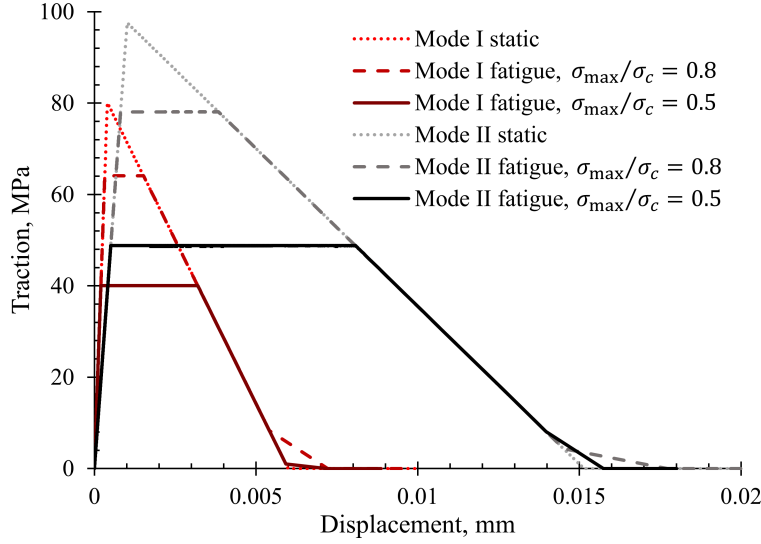


Fig. 3 Traction-displacement histories of single-element verification analyses.

Analyses are performed for each R value with several values of the fatigue strength $\sigma_{\max}/\sigma_c$, starting at 0.99 and decreasing until the predicted fatigue life matches or exceeds the endurance limit of 1×10^7 cycles. The resulting strength-life plots are shown in Fig. 4. The analysis results (black and white open symbols) match very well the behavior described by the evaluated fatigue life equations (black solid and dashed lines).

IV. Validation

A. Doubler Cantilever Beam

Analyses of double cantilever beam (DCB) specimens subjected to cyclic fatigue loading under displacement control are used to validate the concept of the cohesive fatigue model and its implementation within an explicit FE material model. The geometry and conditions of this example problem correspond to the DCB configuration presented in Refs. [8, 9]. The models are composed of unidirectional IM7/8552 carbon/epoxy tape, with pertinent intralaminar material properties listed in Table 4. An initial crack length, a_0 , of 50.8 mm is assumed, and the loading arms have a thickness, t_{arm} , equal to 2.25 mm. Global and crack tip views of the model are shown in Fig. 5. Twelve layers of CPE4R reduced-integration plane strain elements represent each loading arm. A plane strain width, w , of 25.4 mm is used. The interface is modeled with two superposed layers of two-dimensional, four-node COH2D4 cohesive elements. The two superposed layers of cohesive elements are used to represent an experimentally observed R-curve response in which noticeable fiber bridging occurs. The two superposed layers of cohesive elements use the “Initiation” and “Bridge” interlaminar material properties listed in Table 3. Based on the prior DCB fatigue analyses in Refs. [8, 9], the CF20 cohesive fatigue model input values listed in Table 1 are used.

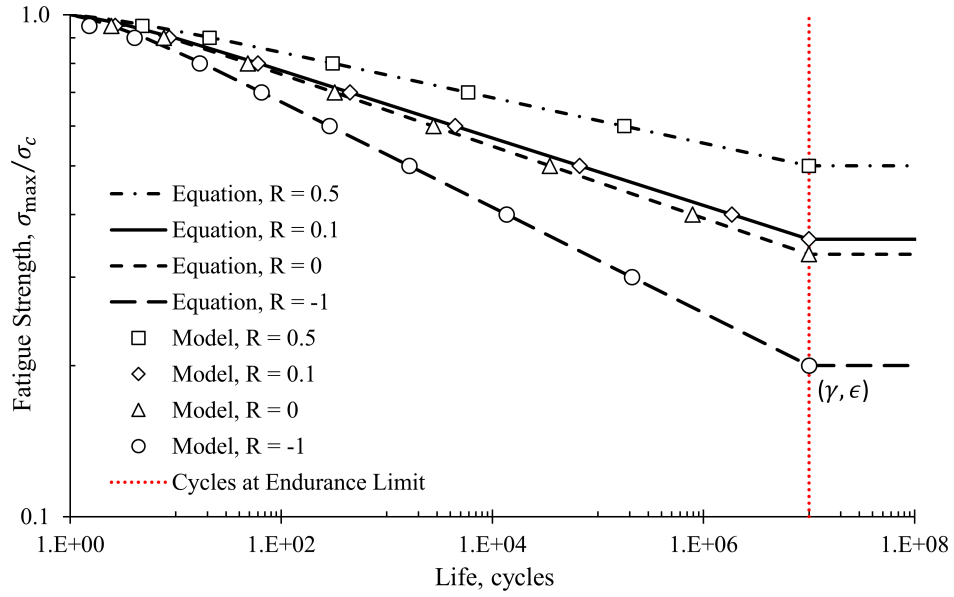


Fig. 4 Strength-life plots from single-element verification analyses for Mode I loading, compared to the fatigue life equations on which the code is based.

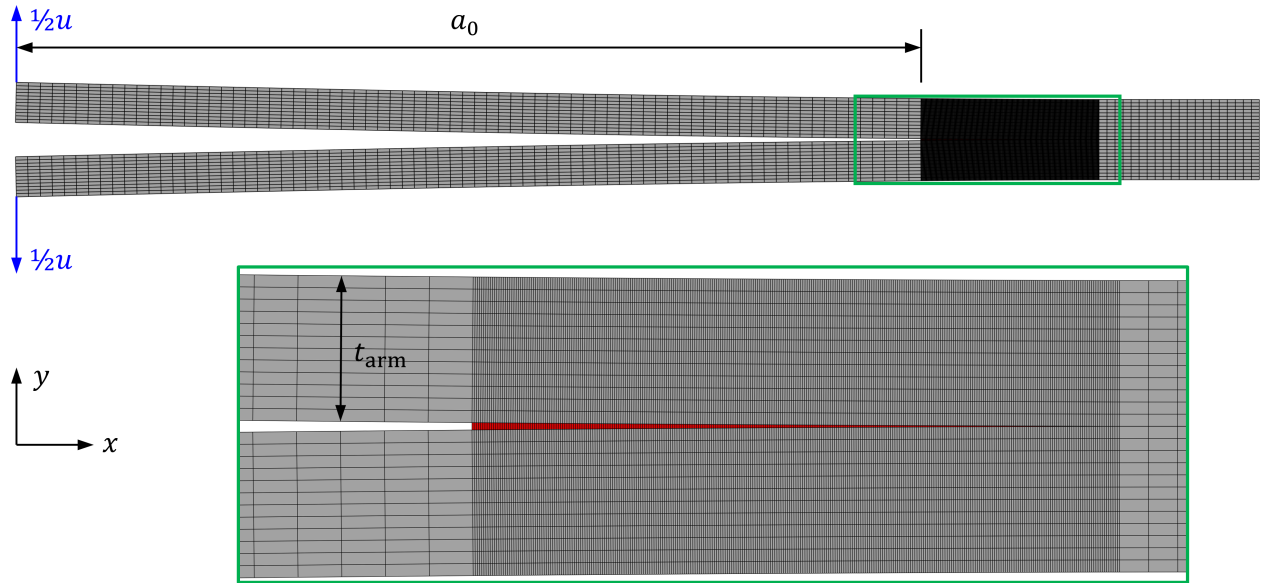


Fig. 5 DCB finite element model.

Table 4 IM7/8552 unidirectional tape material properties

Property	Units	Value
E_1	MPa	146,671
E_2	MPa	8703
E_3	MPa	8703
ν_{12}	-	0.32
ν_{13}	-	0.32
ν_{23}	-	0.45
G_{12}	MPa	5164
G_{13}	MPa	5164
G_{23}	MPa	3001

In the experimental work of Murri [15], four different values of maximum applied displacements, u , were used: 1.48, 1.70, 1.92, and 2.25 mm. The stress ratio was $R = 0.1$ for all tests. The load and fatigue cycle histories allow for the calculation of the specimen compliance, crack length, a , and energy release rate, $G_{I,\max}$, following the approach of Reeder et al. [16]. Implicit and explicit fatigue analyses for each of these four applied displacement values were conducted. The predicted rates of crack propagation, da/dN , for each of the maximum applied displacements are shown in Fig. 6 for the explicit analyses, the experiments, and the implicit analyses. The explicit analysis results (solid lines) match the implicit results (dashed lines) extremely well for all four values of applied displacement, indicating that the two implementations produce equivalent results and that any dynamic effects in the explicit analyses have been sufficiently minimized. The analysis results also possess the same good correlation with the experimental data for each applied displacement value previously demonstrated in Refs. [8, 9].

B. Three-Point Bend Doubler

Analyses of 3PB specimens with a central doubler are performed to validate the implementation of the cohesive fatigue law in the explicit material model. The 3PB doubler specimens have more complicated loading conditions than the DCB specimens, and have mode mixity conditions that vary across the damage front and change as damage develops. A cross-sectional view of a deformed mesh of a 3PB doubler FE analysis is shown in Fig. 7. The specimen skin (light blue) is composed of IM7/8552 tape. The skin is modeled with three layers of SC8R reduced-integration continuum shell elements, has a stacking sequence of $[+45/-45/0/90/+45/-45]_S$, and a thickness t_{skin} of 2.20 mm. The doubler (green) is composed of IM7/8552 plain-weave fabric (Table 5) and has a length L_{doubler} of 101.6 mm. The doubler is modeled with three layers of SC8R continuum shell elements through the thickness. The stacking sequence of the doubler is $[+45/-45/0/+45_2/0/-45/+45_2/-45/0/+45]$, and it has a thickness, t_{doubler} , of 2.38 mm. The width, w , of the test specimens is 25.4 mm. To reduce the computational cost of the models and allow for comparisons between the predictions of the implicit and explicit implementations, the model width is reduced to $1/8$ scale in the initial validation analyses.

The interface between the doubler and the skin panel is modeled with two superposed layers of three-dimensional, eight-node COH3D8 cohesive elements to represent an experimentally observed R-curve response [17]. The two superposed layers of cohesive elements use the “Initiation” and “Bridge” interlaminar material properties listed in Table 6. In order to accurately model the fracture process zone that develops in the cohesive interface, the mesh is highly refined for 16.25 mm from either end of the doubler. The x -direction (parallel to the specimen length) element size in this refined region is 0.05 mm. Across the width of the specimen, an element size of 0.15 mm is used. Cohesive element deletion is enabled for fully damaged elements. The CF20 cohesive fatigue model inputs listed in Table 1 are used in each 3PB analysis.

Fatigue tests were conducted for the 3PB doubler specimens under load control with a stress ratio $R = 0.1$ and three different peak fatigue cycle load values: 214 N, 167 N, and 155 N. The loads applied in the models are proportionally scaled by the width scale factor. Two cylindrical, 50.8-mm-diameter outer rollers and a single, central 12.7-mm-diameter roller are used to load the specimens. The spacing of the two outer rollers, L_{loaders} , is 203.2 mm, which is also the length of the modeled specimen. All three rollers are modeled as rigid analytical surfaces. Friction is modeled between the three rollers and the specimen surfaces with a coefficient of friction equal to 0.3. In the experimental results, crack

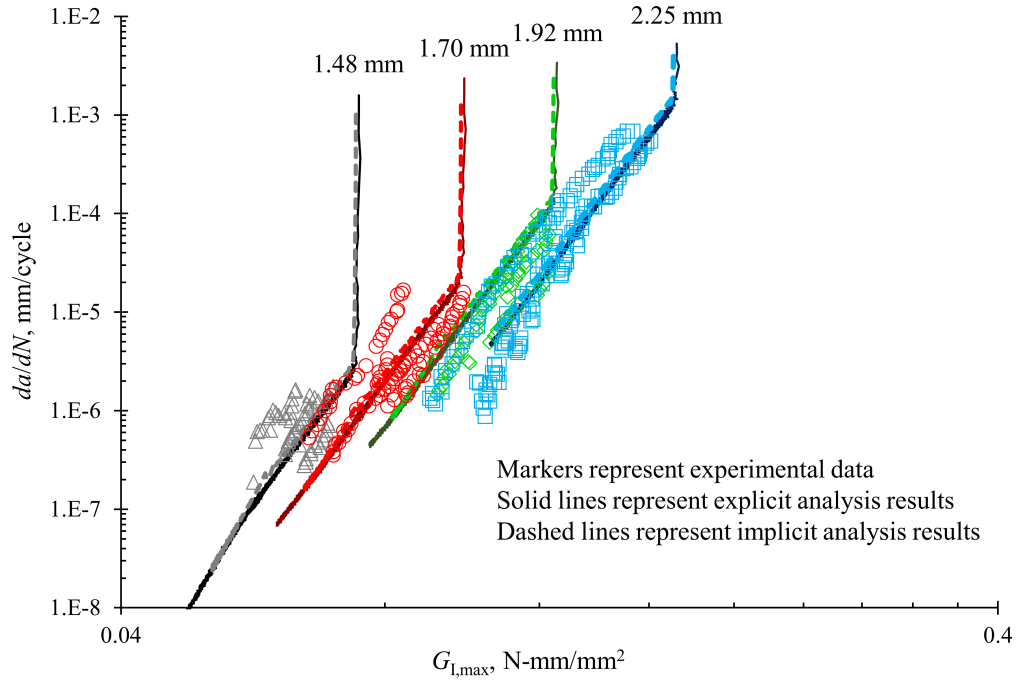


Fig. 6 da/dN versus $G_{I,max}$ for DCB explicit analyses, experiments, and implicit analyses for different peak applied displacements.

growth rates on either side of the doubler are not equal. To break the symmetry of the analyses and promote different crack growth rates on the left and right sides of the model (as pictured in Fig. 7), the x position of the center roller is shifted by -1.016 mm, i.e., one element length in this region of the model. This translation of the center roller causes crack growth rates on the left side of the model to be slightly dominant, which approximates the experimentally observed behavior. Additional boundary conditions at the model center ($x = L_{loaders}/2$) include constraining the x -direction displacement of the nodes on the bottom surface of the skin panel and the y -direction displacement of a single node on the $-y$ surface to avoid unintended rigid body displacements before the applied loads are sufficiently high to hold the specimen in place.

Two analysis steps are used in the 3PB doubler analyses. In the first analysis step, load is applied quasi-statically up to the peak load state of a fatigue cycle. Load is applied using a smooth step amplitude to minimize the accelerations applied to the model. Mass scaling is applied at every solution increment on an element-by-element basis so as to maintain a minimum stable time increment of 1×10^{-7} seconds over the step time of 0.1 seconds. The second analysis step holds the loading constant while fatigue damage evolves. The explicit analyses terminate when the delamination length as measured from either end of the doubler reaches 16.25 mm, i.e., the length of the refined mesh region.

Non-default values are used for the solution parameters that control the evolution of ΔN_i in the 3PB analyses. $(\Delta N_i)_{initial}$ and $(\Delta N_i)_{min}$ are lowered to 1×10^{-6} to reduce dynamic effects due to damage growth at the beginning of the fatigue step. The minimum damage increment is increased to 8×10^{-5} to enable crack growth to the desired length of 16.25 mm before significant numerical errors accumulate. Different values of $(\Delta N_i)_{max}$ are used for the three applied loads: 2.9×10^{-3} , 6.0×10^{-2} , and 1.5×10^{-1} for the 214 N, 167 N, and 155 N cases, respectively. $(\Delta N_i)_{mod}$ is set using Eq. 17 so that at least 150,000 increments pass before $(\Delta N_i)_{max}$ is reached, yielding 7.34×10^{-5} for 167 N and 7.95×10^{-5} for 155 N. A more conservative value of $(\Delta N_i)_{mod}$ (2.50×10^{-5}) is used for the 214 N case due to the high load reached during the static step.

Experimental and analysis results are compared in terms of delamination length, Δa , versus fatigue cycles, N , in Fig. 8. Only the explicit result curves are plotted, as the implicit and explicit analysis results are indistinguishable in terms of their predicted Δa versus N curves. Analysis results are plotted for crack growth from the left (solid black lines) and right (dashed black lines) sides of the doubler. The crack growth trends of the experiments match the analysis predictions for all three applied loads. For the 214 N prediction, approximately 1 mm of crack growth develops on either side of the doubler due to the static loading alone. The fatigue damage growth from this initial damage state

Table 5 IM7/8552 plain-weave fabric material properties

Property	Units	Value
E_1	MPa	66,000
E_2	MPa	65,800
ν_{12}	-	0.052
G_{12}	MPa	5100
G_{13}	MPa	3400
G_{23}	MPa	3400

Table 6 Tape/fabric interlaminar material properties for the 3PB analyses

Property	Units	Initiation	Bridge
σ_c	MPa	105.0	4.0
τ_c	MPa	90.0	12.0
G_{Ic}	N-mm/mm ²	0.35	0.60
G_{IIc}	N-mm/mm ²	1.0	3.5
η_{BK}	-	2.07	3.0
K_n	MPa/mm	500,000	19,048

aligns well with the experimental measurements. The experimental damage onset for the 167 N case occurs suddenly, jumping to approximately 7 mm of crack growth while under fatigue cycling. The predicted damage onset occurs notably earlier, but, again, the predicted smooth response aligns well with the available experimental data points. Good test/analysis agreement is also observed for the fatigue cycles at which damage growth becomes unstable, though a slight under-prediction of the average experimental result occurs for 214 N and a slight over-prediction of the one 155 N result is observed.

V. Scalability

Computational scalability studies are here presented using the 3PB doubler fatigue FE model presented in the previous section. To vary the computational cost of the models, the width of the model is incrementally increased while maintaining all other material inputs and model parameters, e.g., element size across the width, solution increment time. Model width scale factors of $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, $\frac{3}{4}$, 1.0, and 1.5 are considered. The 214 N peak fatigue load case is used.

All analyses performed as part of the scalability study were conducted on the NASA Langley K4 cluster. Each compute node of the K4 cluster is equipped with two 2.4 GHz, 20-core Intel Skylake CPUs. Explicit analyses are conducted with either 40 or 80 cores. The maximum of 80 cores for the explicit analyses was governed by Abaqus token availability, not an observed or expected limit to the scalability of the considered problems. Previous scalability studies by Bergan et al. [10, 11] and a previous practical application of Abaqus/Explicit to predict damage initiation growth in stiffened composite structures [4] found that efficient computational scaling can be expected up to at least 512 cores. Implicit analyses are here conducted with either 8 or 16 cores. The maximum of 16 cores is based on experience of the authors on the scalability of Abaqus/Standard. For each combination of solver and core count (Implicit 8, Implicit 16, Explicit 40, and Explicit 80), fatigue analyses were initially conducted with the smallest model width scale factor. Run times and other results were logged and the next largest model width scale factor was evaluated. This sequence was repeated with each combination of solver and core count until the anticipated model run time using the next largest scale factor was expected to exceed three days.

A summary of the model size in degrees of freedom (DoF), solution increments required to complete the fatigue analysis step, total run time, and the predicted fatigue life for 16.25 mm of crack growth are listed in Table 7 for the explicit analyses. The static analysis steps require an additional 1,042,084 solution increments, not included in the values reported in Table 7. For each pair of 40- and 80-core analyses, a speed-up factor is calculated by dividing the 40-core run time by the corresponding 80-core run time. Ideally, this value would be 2.0 for a doubling of the cores. Instead, the speed-up increases with increasing model size, from 1.14 at a model width factor of $\frac{1}{8}$ to 1.79 at a model

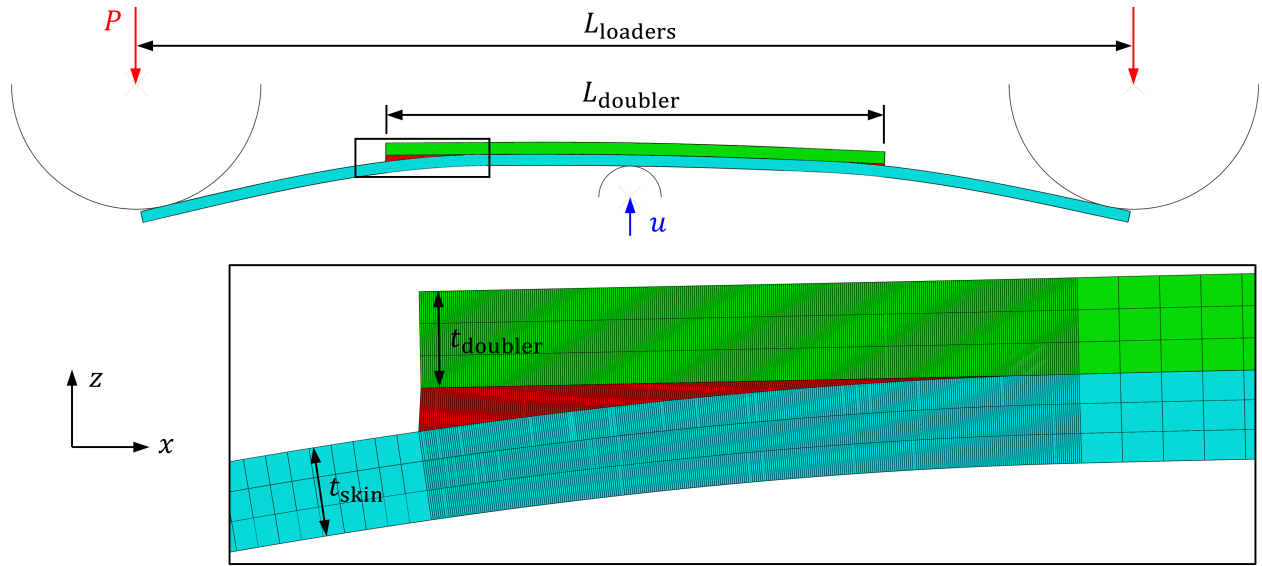


Fig. 7 3PB doubler FE model, shown in its deformed state at the end of the fatigue analysis step.

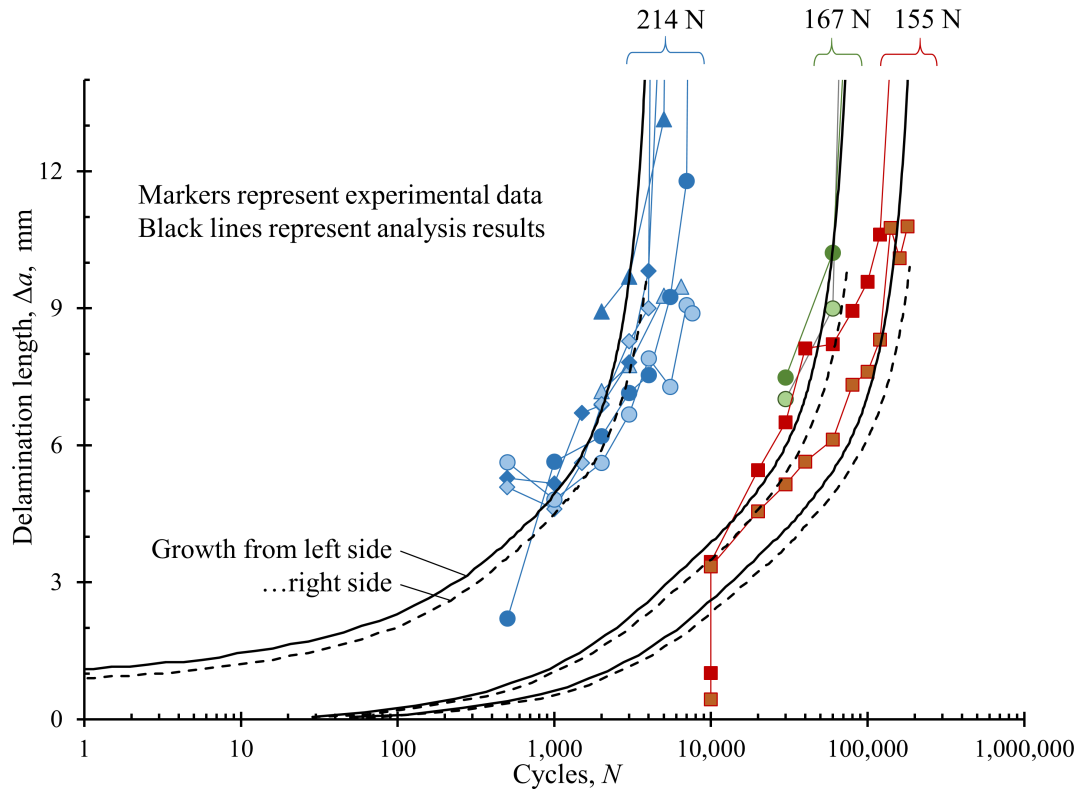


Fig. 8 Crack extension Δa versus fatigue cycles N for the 3PB doubler tests and analyses at three different fatigue cycle peak loads: 214 N (blue, 3 specimens), 167 N (green, 1 specimen), and 155 N (red, 1 specimen). Only explicit results are plotted, as the analyses yielded equal results.

Table 7 Explicit fatigue analysis run times and results

Width factor	DoF	40 cores		80 cores		Speed-up	N_f
		Increments	Time, sec.	Increments	Time, sec.		
0.125	417,666	1,608,836	26,913	1,609,639	23,581	1.14	3,856
0.25	816,330	1,763,029	49,024	1,762,639	30,368	1.61	4,304
0.5	1,613,658	1,960,447	98,977	1,959,956	56,399	1.75	4,876
0.75	2,410,986	2,216,791	143,400	2,222,271	85,184	1.68	5,620
1.0	3,227,298	2,539,925	222,571	2,532,231	124,177	1.79	6,557
1.5	4,821,954	-	-	3,102,806	205,140	-	8,189

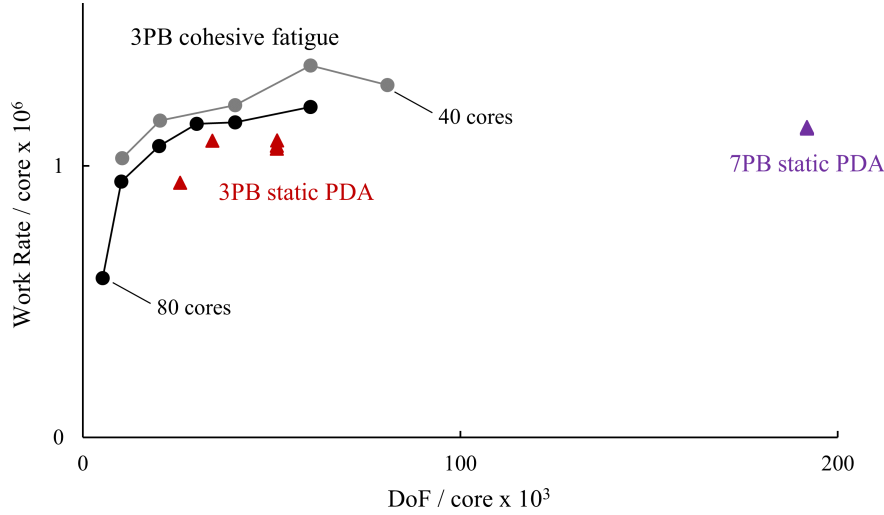


Fig. 9 Work rate per core versus DoF per core for 3PB doubler cohesive fatigue models (black and gray), 3PB static PDA models [3, 11] (red) and 7PB static PDA models [3] (purple). Work rate is defined as total solution increments multiplied by the degrees of freedom, divided by clock time.

width factor of 1.0. The improvements in speed-up factor are nonlinear, with the greatest improvements occurring when increasing from the smallest model sizes. A constant value of speed-up factor is approached as the model sizes become increasingly larger. Here, work rate is defined as the total solution increments multiplied by model size in DoF, divided by the model run time. Plotting work rate per core as a function of DoF per core in Fig. 9 alongside work rates calculated from previous static PDA studies [3, 11] suggests that a minimum model size per core exists, around 50×10^3 DoF per core, above which these models run most efficiently.

Model run time and speed-up results for the 8-core and 16-core implicit analyses are listed in Table 8. The largest implicit analysis that could be completed in less than three days with 16 cores has a model width scale factor of $\frac{3}{4}$. As a result, speed-up factors can only be calculated for model width scale factors of $\frac{1}{8}$, $\frac{1}{4}$, and $\frac{1}{2}$, yielding a maximum speed-up of 1.37 (compared to an ideal of 2.0) for the largest pair of analyses. The assumption that the implicit analyses would scale poorly above 16 cores was spot-checked by running the $\frac{1}{4}$ width factor model with 32 cores, which yielded a speed-up factor of only 1.05 with respect to the 16-core run time despite doubling the cores.

Model run times for the explicit and implicit analyses are plotted together as a function of model width in Fig. 10. Due to the many differences between the implicit and explicit solvers, it is not expected that similar run times should be achieved with similar core counts. The implicit and explicit models happen to take a similar amount of time (6-8 hours) to complete for a width factor of $\frac{1}{8}$. As model sizes increase, however, the explicit implementation is capable of more efficiently utilizing higher core counts to solve the largest of the considered problems. For larger, more complex damage analyses, use of the implicit implementation of the cohesive fatigue model may become computationally intractable.

For relatively simple problems such as the 3PB doubler specimen, using a simplification such as a reduced width scale factor or reducing a problem from three dimensions to two is a generally safe approach and typically introduces

Table 8 Implicit fatigue analysis run times and results

Width factor	DoF	8 cores	16 cores	Speed-up
		Time, sec.	Time, sec.	
0.125	442,380	29,088	23,248	1.25
0.25	865,551	72,217	54,701	1.32
0.5	1,711,893	185,418	135,315	1.37
0.75	2,558,235	-	214,913	-

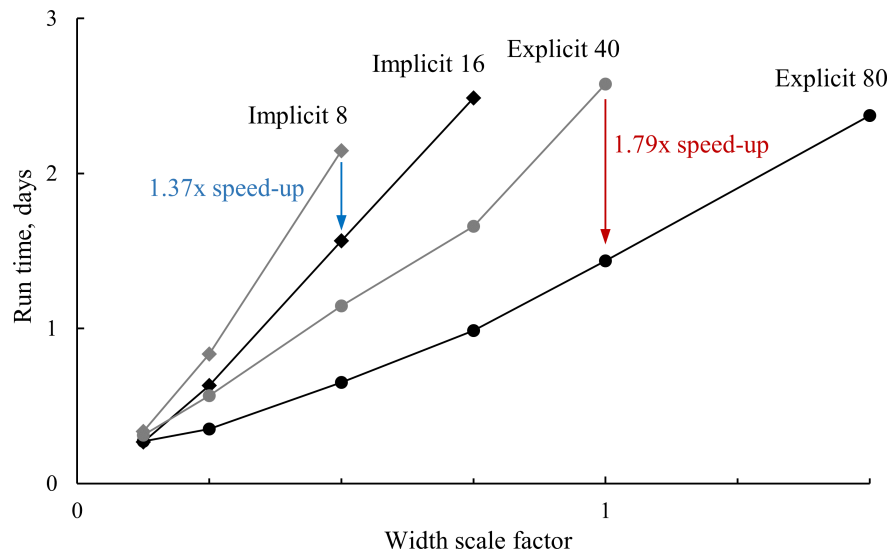


Fig. 10 Run times of 3PB doubler fatigue analyses.

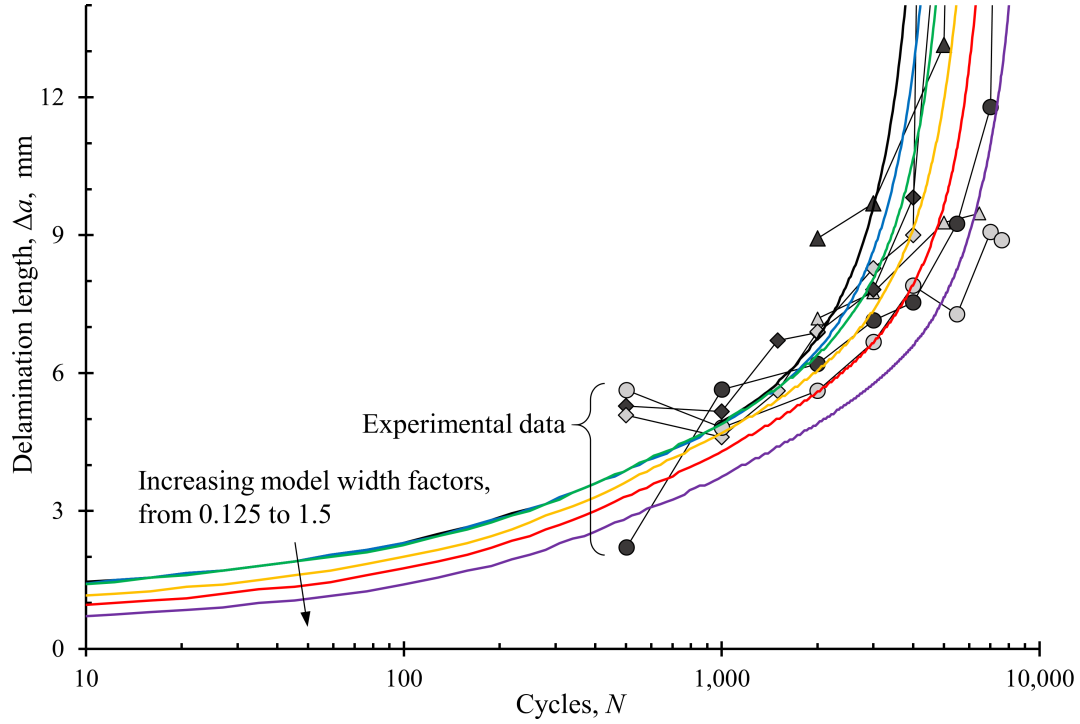


Fig. 11 Crack extension Δa versus fatigue cycles N for the 3PB doubler tests conducted with a peak load of 214 N and analyses performed with increasing model widths.

an acceptable amount of error. The improved computational scalability of the explicit cohesive fatigue law allows for the examination of errors that result from artificially reducing the 3PB doubler model width with respect to the test specimen dimensions. Examination of the predicted fatigue life N_f in Table 7 reveals that the $\frac{1}{8}$ -scale model predicts a life 41% shorter than the full-scale model prediction. The Δa versus N curves for the 214 N 3PB doubler analyses with all evaluated model width scale factors are plotted in Fig. 11. The larger model widths increase the amount of anticlastic bending in the models and affect the mode mix of the tractions acting on the interface between the doubler and the skin panel. The analysis results with width factors $\frac{1}{8}$, $\frac{1}{4}$, and $\frac{1}{2}$ are initially equivalent, but begin to deviate after reaching 1,000 to 2,000 fatigue cycles, after which the wider models are predicted to have longer fatigue lives. Models with larger width factors, i.e., greater than 0.5, have shorter initial crack lengths at the beginning of the fatigue analysis step, leading to altogether different damage evolution histories and slower overall damage progression.

VI. Closing Remarks

An explicit implementation of a novel cohesive fatigue damage material model is herein demonstrated to be able to make equivalent predictions to an implicit implementation at scales including single elements, material characterization coupons, and simple structural details. When models are built that sufficiently minimize any dynamic effects and maintain quasi-static loading conditions throughout the specimen loading history, implicit and explicit analysis predictions agree in terms of load histories, damage initiation, damage morphology, and remaining life predictions. The explicit implementation of the cohesive fatigue law exhibits the same favorable computational scalability of explicit CDM methods, enabling the analysis of structural problems which would be otherwise intractable for an implicit analysis method. Also enabled by this development is the ability to increase the fidelity of fatigue damage analysis predictions, by, for example, modeling mesoscale matrix cracks and their interactions with delaminations by utilizing the presented cohesive fatigue model in existing explicit PDA material models.

References

- [1] Wanthal, S., Schaefer, J., Justusson, B., Hyder, I., Engelstad, S., and Rose, C., “Verification and Validation Process for Progressive Damage and Failure Analysis Methods in the NASA Advanced Composites Consortium,” *32nd American Society for Composites Technical Conference*, West Lafayette, IN, 2017.
- [2] Leone, F. A., Bergan, A. C., and Dávila, C. G., “CompDam - Deformation Gradient Decomposition (DGD), v2.5.0,” 2019. https://github.com/nasa/CompDam_DGD.
- [3] Leone, F. A., Song, K., Johnston, W., Rose, C. A., Jackson, W. C., and Dávila, C. G., “Test/Analysis Correlation of Damage States in Post-buckled Stiffened Validation Building Block Specimens,” *34th American Society for Composites Technical Conference*, Atlanta, GA, 2019. <https://doi.org/10.12783/asc34/31277>.
- [4] Leone, F. A., Song, K., Rose, C. A., and Jackson, W. C., “Progressive Damage Analysis of Post-Buckled Stiffened Panels under Static Compressive Loading,” *AIAA SciTech Forum*, Orlando, FL, 2020. <https://doi.org/10.2514/6.2020-1479>.
- [5] Action, J., and Leone, F. A., “Progressive Damage Failure Analysis of a Multi-Stringer Post-Buckled Panel,” *AIAA SciTech Forum*, Orlando, FL, 2020. <https://doi.org/10.2514/6.2020-1481>.
- [6] Leone, F. A., “Deformation Gradient Tensor Decomposition for Representing Matrix Cracks in Fiber-Reinforced Materials,” *Composites Part A: Applied Science and Manufacturing*, Vol. 76, 2015, pp. 334–341. <https://doi.org/10.1016/j.compositesa.2015.06.014>.
- [7] ABAQUS, *Abaqus 6.19 Online Documentation*, Dassault Systèmes, Providence, RI, 2019.
- [8] Dávila, C. G., “From S-N to the Paris Law with a New Mixed-Mode Cohesive Fatigue Model,” NASA/TP-2018-219838, NASA Langley Research Center, Hampton, VA, Dec. 2018.
- [9] Dávila, C. G., Rose, C. A., Jackson, W. C., and Johnston, W., “Evaluation of Fatigue Damage Accumulation Functions for Delamination Initiation and Propagation,” NASA/TP-2020-220584, NASA Langley Research Center, Hampton, VA, Apr. 2020.
- [10] Bergan, A. C., “Computational Performance of Progressive Damage Analysis of Composite Laminates using Abaqus/Explicit with 16 to 512 CPU Cores,” NASA/TM-2019-220251, NASA Langley Research Center, Hampton, VA, Feb. 2019.
- [11] Bergan, A. C., Leone, F. A., Boyd, S., Hyder, I., and Schaefer, J. D., “Computational Considerations in Scaling the Application of Progressive Damage Analysis Methods from Coupons to Sub-Components,” *34th American Society for Composites Technical Conference*, Atlanta, GA, 2019. <https://doi.org/10.12783/asc34/31278>.
- [12] González, E. V., Maimí, P., Turon, A., Camanho, P. P., and Renart, J., “Simulation of Delamination by Means of Cohesive Elements Using an Explicit Finite Element Code,” *Computers, Materials & Continua*, Vol. 9, No. 1, 2009, pp. 52–92. <https://doi.org/10.3970/cmc.2009.009.051>.
- [13] Benzeggagh, M. L., and Kenane, M., “Measurement of Mixed-Mode Delamination Fracture Toughness of Unidirectional Glass/Epoxy Composites with Mixed-Mode Bending Apparatus,” *Composites Science and Technology*, Vol. 56, 1996, pp. 439–449. [https://doi.org/10.1016/0266-3538\(96\)00005-X](https://doi.org/10.1016/0266-3538(96)00005-X).
- [14] Allegri, G., and Wisnom, M. R., “A Non-linear Damage Evolution Model for Mode II Fatigue Delamination Onset and Growth,” *International Journal of Fatigue*, Vol. 43, 2012, pp. 226–234. <https://doi.org/10.1016/j.ijfatigue.2012.03.016>.
- [15] Murri, G. B., “Evaluation of Delamination Onset and Growth Characterization Methods under Mode I Fatigue Loading,” NASA/TP-2013-217966, NASA Langley Research Center, Hampton, VA, Feb. 2013.
- [16] Reeder, J. R., Demarco, K., and Whitley, K. S., “The Use of Doubler Reinforcement in Delamination Toughness Testing,” *Composites: Part A*, Vol. 35, 2004, pp. 1337–1344. <https://doi.org/10.1016/j.compositesa.2004.02.021>.
- [17] Dávila, C. G., Leone, F. A., Song, K., Ratcliffe, J. G., and Rose, C. A., “Material Characterization for the Analysis of Skin/Stiffener Separation,” *32nd American Society for Composites Technical Conference*, West Lafayette, IN, 2017.