An efficient computational strategy of cycle-jumps dedicated to fatigue of composite structures

Orianne Sally, Frédéric Laurin, Cédric Julien, Rodrigue Desmorat, Florent Bouillon

An efficient computational strategy of cycle-jumps dedicated to fatigue of composite structures

O. Sally[a,b,c], F. Laurin[a], C. Julien[a], R. Desmorat[b], F. Bouillon[c]

[a] *DMAS, ONERA, Université Paris Saclay, F-92322 Châtillon, France*
[b] *LMT (ENS Paris-Saclay, CNRS, Université Paris Saclay), 61 avenue du Président Wilson, F-94230 Cachan, France*
[c] *Safran Ceramics, a technology platform of Safran Tech, 105 avenue Marcel Dassault, F-33700 Mérignac, France*

Abstract

A method to accelerate numerical simulations of composite structures subjected to complex fatigue loading is investigated. A kinetic (rate) damage model predicts the damage evolution of composites during fatigue loadings, even for spectral loadings. A non-linear cycle-jump approach designed specifically for rate damage models is proposed, it allows to reduce drastically the computational time of fatigue Finite Element simulations. The simulations using the proposed cycle-jump method are compared to reference calculations where all cycles are conducted and also benchmarked against a classical cycle-jump method. This approach accounts for the skipped cycles, even under multiaxial loadings, and leads to large computational cost reductions.

Keywords: Composite materials, Fatigue life, Damage model, Cycle-jump, Finite Element analysis

Email address: orianne.sally@onera.fr (O. Sally)

1. Introduction

Due to their specific properties, composites are currently widely used in aeronautics for structural applications. Composite materials are much less sensitive to fatigue loadings than metallic materials, but their mechanical properties do decrease during repetitive loadings. During service fatigue loadings, their moduli can decrease due to damage, such as matrix cracking, fibre/matrix debonding or fibre failure. Therefore, the determination of the lifetime of composite structural parts has become a major issue for design offices, especially for the design of large rotating parts. Determining the fatigue life of composite structures may turn out to be quite costly compared to most metallic parts, due to the differences between their damage scenarios.

Metals are slightly affected by mechanical damage (visible at the macroscopic scale) during most of their lifetime, but fail shortly after the first crack initiation. Constitutive models are aimed at reaching a stabilized cycle, due to plasticity, from which the fatigue life of the structure is determined by post-processing. On the contrary, in composite materials, damage occurs from the beginning of loading and these materials degrade progressively throughout their lifetime. Due to load transfers and damage propagation, there is no stabilized cycle. Consequently, it is necessary to simulate the whole cyclic loading. For this purpose, a material model describing the evolution of material properties during loading is required. In addition, computational strategies need to be considered in order to be able to simulate high cycle fatigue loadings within time durations, which are acceptable for a design office.

Different numerical approaches have been proposed in the past decades to obtain a predictive model of fatigue lifetime and simulate the evolution of the damage mechanisms. Many of them are based on continuum damage models, describing the progressive degradation through damage variables. Two main families can be identified. On the one hand, cyclic models propose evolution laws for damage increment per cycle $\frac{\mathrm{d}d}{\mathrm{d}N} = ...$, which have been recently applied to composite materials [1, 2, 3, 4, 5]. On the other hand, kinetic (rate) models consider a time evolution of damage $\dot{d} = ...$ [6, 7, 8]. In the present work, we focus on a kinetic model, the continuous evolution of damage enabling to naturally consider spectral loadings, representative of real industrial applications. Moreover, it can be easily implemented in a Finite Element (FE) code through a standard thermodynamics framework for constitutive equations [9].

However, to determine the lifetime of mechanical components, major computational issues remain. The number of cycles to failure can be as high as tens of millions or more and it is practically not feasible to perform a cycle-by-cycle simulation (let alone a simulation with several time steps per cycle) of the whole structure. To reduce computational costs, several strategies are possible and have been proposed in the literature.

Numerical methods of model reduction such as the Proper Orthogonal Decomposition (POD) [10] method, which consists in giving a reduced approximation of the FE solution, could be used. Another alternative is the combination of the LATIN method (Large Time Increments) [11], which is an iterative pro-

cedure accounting for the whole loading process in a single (large) time increment, with the Proper Generalized Decomposition (PGD) [12]. Indeed, these two methods have recently been coupled for cyclic damage simulations [13, 14]. However, these methods perform well under the condition that all mechanisms are activated during the training (first) cycles, which is not necessarily the case for composites (activation of fibre degradation for instance). In addition, these methods remain too intrusive for any simple implementation in a commercial FE solver.

A different method consists in using a cycle-jump strategy. Such a method is quite easy to implement into a FE code, and reduces computational costs by skipping full blocks of intermediate cycles after some «training cycles », *i.e.* when the structure properties evolve slightly. To account for the unsimulated cycles, called jumped cycles, the internal variables are extrapolated. For metallic materials, the variables of interest are, firstly, the plastic strain and, secondly, the damage variable; for composites, they are usually the damage variables. This approach is, however, limited to periodic cyclic loadings. This technique was first developped in the late 80's by Lesne and Savalle [15] and Lemaitre and Doghri [16], shortly afterwards. It enables to quickly achieve a stabilized cycle and/or to determine the crack initiation conditions in viscoplastic simulations performed on metallic materials. These authors proposed extrapolation methods based on a first or a second order Taylor development. The jump length may vary throughout the simulation (*i.e.* may be different from a jump to another) and the authors provided different methods to calculate its optimal value. This method was implemented into the FE code Zset by Sai [17]. Kiewel [18] developed a cycle-jump method during which each internal variable at each integration point of the FE model is extrapolated, based on a polynomial or a spline function. Van Paepegem [1] was the first to apply a cycle-jump method to composite structure simulations. The extrapolation method was based on an explicit Euler integration formula (equivalent to a first order Taylor development). The jump lengths are also adaptive. Locally, it is calculated using the Euler's formula with a given maximal allowed damage increment (as in [16]). Van Paepegem [1] proposed a new method to determine the optimized jump length at the structure scale, using the cumulative statistical distribution of local jump lengths. A similar cycle-jump strategy was also applied to simulate the propagation of delamination in laminated composites under high-cycle fatigue loadings [19]. Fish and Yu [20] also used an explicit Euler method to extrapolate the fatigue damage variable in a multi-scale model dedicated to brittle composites, where the jump length is controlled through a modified Euler integrator. Cojocaru [21] proposed a method of cycle-jumps dedicated to evolving structural properties. His method is based on a trend line, established during FE analysis for training cycles. This trend is used to extrapolate blocks of cycles. This cycle-jump method works as a post-processing block where the extrapolated state is used as initial state for additional FE simulations.

Due to the extrapolation, all these methods induce an error in the simulation, regardless of the jump length, that may be prohibitive for lifetime predictions. In addition, they use a cyclic form of damage, $\frac{\mathrm{d}d}{\mathrm{d}N} = ...$, even when the model

evolution law is initially time dependant. However, there is, to our knowledge, no method directly relying on a kinetic model, $\dot{d} = ...$, allowing real consideration of spectral loadings.

Therefore, this article deals with a new method of non-linear cycle-jumps, which is based on a kinetic model and applied to composite structures. This method is applied to cyclic fatigue, the cycles of which may be either min-max or sine cycles, and to the so-called quasi-random fatigue, which consist of random cycles repeated over time. The considered cycles may be proportional or not. Moreover, such a kinetic model enables the simulation of a numerical reference, where all the cycles are computed. The efficiency and the interest of the proposed approach is then assessed against this reference. Its interest is highlighted by comparisons with a classical method, that uses a first order extrapolation such as that developped in [1].

The kinetic damage model, on which the proposed cycle-jump method is based, is presented in section 2. Its particularity consists in its kinetic damage evolution law, which is able to automatically switch between static (monotonic) and fatigue cases, without user interaction. The cycle-jump extrapolation method and its implementation in a commercial FE code are described in section 3. Finally, the method is applied to different test cases of increasing complexity, from cyclic loadings on an integration point to an open-hole plate subjected to complex spectral loadings.

2. A kinetic (rate) damage model

2.1. General principle

The model is expressed in the Continuum Damage Mechanics framework and is defined at the macroscopic scale in order to consider large composite structures. It is a unified damage model, able to predict the strength and the fatigue lifetime of composite materials under static and complex fatigue loadings.

The formulation of the present model is based on previous works performed at Onera and LMT for composites with polymer or ceramic matrix [5, 8, 22, 23, 24]. However, the model is written as a function of the total strain for numerical reasons, in order to simplify its implementation in a FE code. It could be used for any composite material which damage mechanisms are oriented by the microstructure. It is assumed that the observed non-linearities are only due to the two following diffuse damage mechanisms, a relevant assumption for ceramic composite materials at room temperature, which are represented by 2 in-plane scalar damage variables: (i) matrix cracking in the warp direction noted d_1 and (ii) matrix cracking in the weft direction noted d_2, which are induced by combined in-plane normal and shear loadings. The composite material being considered as thin with high through-the-thickness strengths and subjected only to high cycle fatigue (low loading level); delamination is not considered in the present study.

The present approach is thermodynamically consistent and thus the macroscopic behaviour, expressed in Eq.(1), derives directly from the Helmholtz free energy density,

$$\boldsymbol{\sigma} = \mathbb{C}^{\text{eff}} : \boldsymbol{\varepsilon} - \mathbb{C}^{0} : \boldsymbol{\varepsilon}^{r} \tag{1}$$

$$\text{with} \quad \mathbb{S}^{\text{eff}} = \left(\mathbb{C}^{\text{eff}}\right)^{-1} = \mathbb{S}^{0} + \sum_{k=1}^{2} d_k \mathbb{H}_k \tag{2}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\mathbb{C}^{\text{eff}}$ the effective elastic stiffness tensor taking into account the effects of the two different damage mechanisms, $\mathbb{C}^{0}$ the initial elastic stiffness tensor, $\boldsymbol{\varepsilon}$ the total strain tensor and $\boldsymbol{\varepsilon}^{r}$ the residual strain (due to damage). In the present approach, the effects of damage mechanisms on the macroscopic behaviour are applied through the increase of the initial elastic compliance, $\mathbb{S}^{0}$, with an additional term, $\sum_k d_k \mathbb{H}_k$, which depends on the damage variables, d_k, and the corresponding effect tensors, $\mathbb{H}_k$, describing the effect of an open crack on the different components of the effective stiffness. The subscript k refers to the considered damage mechanism ($k = 1$ for the damage mechanism in the warp direction and $k = 2$ for the damage mechanism in the weft direction).

2.2. Damage variables

2.2.1. Damage equivalent strain

The driving forces associated with the damage variables are formulated in a so-called non-standard thermodynamic framework [5, 25, 23]. They are expressed, using Voigt notation, as scalar equivalent strains (Eq.(3)), as in [8].

$$\begin{cases} \varepsilon_{eq,1} = \sqrt{(\varepsilon_1^{d_1^+})^2 + e_{15}(\varepsilon_5^{d_1^+})^2 + e_{16}(\varepsilon_6^{d_1^+})^2} \\ \varepsilon_{eq,2} = \sqrt{(\varepsilon_2^{d_2^+})^2 + e_{24}(\varepsilon_4^{d_2^+})^2 + e_{26}(\varepsilon_6^{d_2^+})^2} \end{cases} \tag{3}$$

They depend on (i) the material parameters, e_{ij}, associated with the damage thresholds for pure tensile or shear loadings and (ii) on the positive strain tensors, $\boldsymbol{\varepsilon}^{d_k^+}$, related to each damage mechanism. The material parameters (e_{15}, e_{16}, e_{24}, e_{26}) are defined as functions of the in-plane tensile strains at the onset of damage in the material directions (ε_1^{0s},ε_2^{0s}) and of the in-plane and out-of-plane shear damage thresholds (ε_6^{0s},ε_5^{0s},ε_4^{0s}) for static loadings. These threshold strains can be determined using quasi-static tensile tests at 0^o, 90^o and 45^o, and ILSS tests for the out-of-plane shear thresholds. These material parameters are given by the relation $e_{ij} = 4\left(\frac{\varepsilon_i^{0s}}{\varepsilon_j^{0s}}\right)^2 - \frac{1}{4}$ with i=(1, 2) and j=(4, 5, 6). It allows predicting accurately the onset of damage for pure tensile, pure shear, and also for combined tensile/shear or compressive/shear loadings.

The positive strain tensors $\boldsymbol{\varepsilon}^{d_k^+}$ correspond to the positive part, as proposed initially by [26] and used later in [5], of the total strain tensor $\boldsymbol{\varepsilon}^{d_k}$ where all the components are zeros except those inducing damage associated with the

variable d_k, as reported in Eq. (3) and in AppendixA. The use of the positive strain tensors is an elegant way to capture the reinforcement of the apparent onset of damage for combined compressive and shear loading.

2.2.2. Evolving mean equivalent strain to take into account the mean stress effect

To take into account the mean stress effect in the present model, an evolving mean equivalent strain, $\breve{\varepsilon}_{eq}$, is introduced in the kinetic damage evolution law, $\dot{d}_k = \ldots$, presented in the next paragraph. It is calculated using a kinetic mean (frequency independent where time, t, is the current time), based on the past history of loading and described in [7, 8],

$$\breve{\varepsilon}_{eq}(t) = \frac{1}{\varepsilon_{eq}^{ac}} \int_0^t \varepsilon_{eq} \, |d\varepsilon_{eq}^{ac}| \quad \text{with} \quad \varepsilon_{eq}^{ac}(t) = \int_0^t |d\varepsilon_{eq}| \tag{4}$$

The proposed formalism of the kinetic mean has been validated through comparisons on basic cyclic loadings (*i.e.* using constant sine or triangular cycles) with the arithmetic mean value which can be easily determined in such tests, as reported in Eq. (5).

$$\breve{\varepsilon}_{eq}(t) \xrightarrow[\text{cyclic loading}]{t\to\infty} \int_{1 \text{ cycle}} \varepsilon_{eq} \mathrm{d}t = \frac{1}{2}\left(\varepsilon_{eq}^{min} + \varepsilon_{eq}^{max}\right) \tag{5}$$

2.2.3. Damage evolution law

Previous works on Polymer Matrix Composites (PMCs) [4, 27] have shown that damage induced by static or by fatigue loadings present the same effects on the behaviour and the same damage saturation level, but have different evolution laws (the fatigue kinetics being much slower than the static one). Therefore, a unique thermodynamics (state) damage variable and a unique damage evolution law have been proposed for each damage mechanism, k, that merge the contributions from both types of loadings.

Following Angrand [8], the evolution laws for the damage variables, d_k, represent a key point in the present approach. They are formulated as kinetic (rate) damage laws (Eq.(6)), that describe the time evolution of damage and are expressed in a rate form: $\dot{d}_k = ...$

For both mechanisms, $k = 1$ or $k = 2$:

$$\begin{aligned}\dot{d}_k =& (d_{\infty,k} - d_k) \left\langle \frac{\varepsilon_{eq,k}^{max} - \varepsilon_{eq,k}^{0s}}{S_k^s} \right\rangle_+^{s_k^s} \left\langle \frac{\mathrm{d}\varepsilon_{eq,k}^{\max}}{\mathrm{d}t} \right\rangle_+ \\ &+ (d_{\infty,k} - d_k)^{\gamma_k} \left\langle \frac{\varepsilon_{eq,k} - m_k \breve{\varepsilon}_{eq,k} - \varepsilon_{eq,k}^{0f}}{S_k^f} \right\rangle_+^{s_k^f} \left(\left\langle \frac{\mathrm{d}\varepsilon_{eq,k}}{\mathrm{d}t} \right\rangle_+ - \frac{\mathrm{d}\varepsilon_{eq,k}^{\max}}{\mathrm{d}t} \right)\end{aligned} \tag{6}$$

where $< . >_+$ are the classical Macauley brackets, $d_{\infty,k}$ corresponds to the saturation level of damage, d_k, (S_k^s, s_k^s) and (m_k,S_k^f,s_k^f,γ_k) are material parameters, which allow for calibrating respectively static and fatigue damage evolution laws.

The parameters ($\varepsilon_{eq,k}^{0s}$, $\varepsilon_{eq,k}^{0f}$) stand for the static and fatigue damage thresholds. The fatigue threshold, $\varepsilon_{eq,k}^{0f}$, which is lower than the static one, corresponds to the fatigue limit of the material. The identification procedure of the damage evolution law is presented in [28] and necessitates the analysis of tests performed in the warp and weft directions: two quasi-static tensile tests, two cyclic fatigue tests with different maximal stress levels but with the same fixed stress ratio and two Locati tests with two different stress ratios.

The static and fatigue contributions to damage cannot be activated simultaneously. Indeed, the evolution of $\varepsilon_{eq,k}^{\max} = \sup_{t<\tau}(\varepsilon_{eq,k}(\tau))$, which is the maximum value of the equivalent strain, $\varepsilon_{eq,k}$, over the whole loading history, allows to automatically switch between the static and fatigue evolution. For static loadings, $\varepsilon_{eq,k}^{\max}$ increases in a similar manner as the current equivalent strain, $\varepsilon_{eq,k}$; therefore, only the first term of Eq.(6) is activated. Conversely, for fatigue loadings, $\varepsilon_{eq,k}^{\max}$ remains constant and only the second term of Eq.(6) is responsible for the damage evolution. Moreover, the use of the equivalent strain is convenient for multiaxial loadings. It should be noted that the damage is only allowed to increase (between 0 and $d_{\infty,k}$) during the loading phases, in order to ensure the second principle of thermodynamics.

This kinetic formulation enables naturally the model to simulate any kind of loading: monotonic, cyclic or spectral [7, 8]. It decomposes time into intervals of either static or fatigue loading, a property that will be useful to implement efficiently cycle jumps. Consequently, the damage evolution law can be analytically integrated with the assumption that, during a time increment, the mean equivalent strain does not evolve. The static (resp. fatigue) damage increment can then be calculated over one cycle.

2.3. Effect tensors $\mathbb{H}_k$

Concerning the effect of the different damage mechanisms, in the warp direction for instance ($k = 1$), the effect tensor, $\mathbb{H}_1$, associated with the damage variable, d_1, is defined as a function of a deactivation index, η_1, and the effect tensors, $\mathbb{H}_1^+$ and $\mathbb{H}_1^-$. These two tensors describe the effects of an open crack or a closed crack on the effective stiffness. The tensors $\mathbb{H}_1$, $\mathbb{H}_1^+$ and $\mathbb{H}_1^-$ are given in Eq.(7), in Voigt notation. The parameters (h_{11}^+, h_{66}^+) are material coefficients, which must be identified. These two material parameters can be determined using a micromechanical approach [29] which necessitates only the knowledge

of the elastic properties of the material.

$$\mathbb{H}_1 = \eta_1 \mathbb{H}_1^+ + (1-\eta_1)\mathbb{H}_1^- \tag{7}$$

$$\text{with } \eta_1 = \begin{cases} 1 \text{ if } \sigma_{11}^n \geq 0,\ \boldsymbol{\sigma}^n = \mathbb{C}^{eff} : \boldsymbol{\varepsilon} \\ 0 \text{ instead} \end{cases} \quad \text{and}$$

$$\mathbb{H}_1^+ = \begin{pmatrix} h_{11}^+ S_{11}^0 & & & & & & \\ & 0 & & & & (0) & \\ & & 0 & & & & \\ & & & 0 & & & \\ & & & & 0 & & \\ & (0) & & & & h_{66}^+ S_{66}^0 & \end{pmatrix}, \mathbb{H}_1^- = \begin{pmatrix} 0 & & & & & & \\ & 0 & & & & (0) & \\ & & 0 & & & & \\ & & & 0 & & & \\ & & & & 0 & & \\ & & & & & h_{66}^+ S_{66}^0 & \end{pmatrix}$$

2.4. Residual strains

The specific strain tensor, $\boldsymbol{\varepsilon}^r$, accounts for the residual strains after unloading that are due to the evolution of the different types of damage. Its growth rate is directly related to the damage rate variables, $\dot{d}_k$, as shown in Eq.(8),

$$\frac{\mathrm{d}\boldsymbol{\varepsilon^r}}{\mathrm{d}t} = \sum_{k=1}^{3} \left[\chi_k e^{r_k d_k} \dot{d}_k \mathbb{R}_k \boldsymbol{e_k^*} \right] \tag{8}$$

$$\text{with} \quad \boldsymbol{e}_k^* = \frac{\boldsymbol{\varepsilon}}{|\max_i \varepsilon_{ki}|}, \quad i = [1,3] \text{ and } \mathbb{R}_k = \mathbb{H}_k : \mathbb{C}^0$$

where χ_k and r_k are material parameters, that can be identified using quasi-static tensile tests with regular unloading/loading until failure in the warp and weft directions, as detailed in [28]. The direction, $\boldsymbol{e}_k^*$, for each mechanism k, is considered constant over the time increment. This assumption is exact for a uniaxial loading. This formulation complies with the need of reducing computational cost at each increment, as it enables to compute analytically the residual strains, as proposed in [8].

This type of model has already been validated through comparisons with experimental results on PCMs [7, 30]. In the present article, the model is applied to a 2D woven oxide/oxide composite with a 50/50 warp/weft ratio. Therefore, the material parameters associated with the warp direction are considered equal to those associated with the weft direction. The oxide/oxide composite material considered in this study has been developed by Safran Group and the material parameters, used in section 4, are confidential. Nevertheless, other similar 2D woven oxide/oxide composite materials can be found in the literature and have been subjected both to static [24, 31] and fatigue [32, 33] loadings.

3. Modelling of cycle-jumps dedicated to damage models for composites

The cycle-jump procedure, first developped by [15, 16], enables to speed up cyclic fatigue simulations while maintaining their predictive capacities. For

metals, the variable of interest that pilots the non linearity of the material, is the plastic strain whereas for composites, damage is the key quantity.

The general approach for cycle-jumps, also applied in this work, is illustrated in Fig. 1, where N stands for the number of cycles, each of period T. The simulation can be decomposed as follows:

- Monotonic loading ramp-up,
- Conduct training cycles: a set of cycles are fully computed with the model to establish the trend of the damage variable evolution,
- Jump (in dotted line): extrapolation of the internal variables, based on the damage increments calculated during the previous cycle, at the end of the unsimulated cycles.
- Control cycles (in bold): a set of cycles fully computed before jumping again to update the damage increment variables with accurate values not disturbed by the previous jump.

These last two steps are repeated until the end of the simulation.

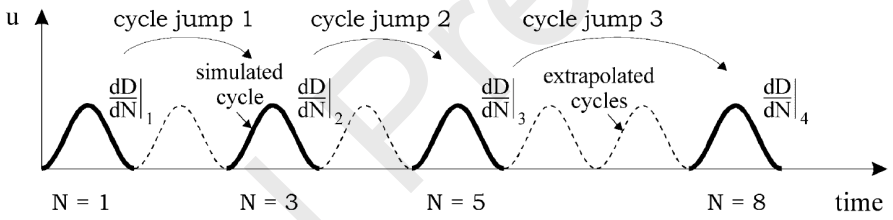


Figure 1: Illustration of the cycle-jump principle [1] (where D accounts for the damage variable, d_k, in the present work).

We propose here a non-linear extrapolation method, which relies closely on the model damage law (Eq. (6)).

The damage value integration during a jump of $\overline{\Delta N}$ cycles is detailed in section 3.1. Under some hypotheses, which will be described next, the extrapolation of the damage value is quasi-exact.

It should be emphasized that the proposed cycle-jump procedure applies to complex loadings, with the proper definition of a cycle. In fact, periodicity in all the variables is not required but only in the applied loading, formulated either as applied displacements or applied loads at some boundaries of a structure. One does not make use of the notion of stabilized cycle. Moreover, the cycles can be complex, *i.e.* multiaxial and quasi-random, as in [34]. By quasi-random, we mean transient loading, but repeated. The only required hypothesis is that damage does not evolve much (compared to a saturation value, d_∞, or to a critical value, d_c) over a cycle.

The subscript $k = 1, 2$ is omitted in this section and we set $d_N = d(t_N)$ and $d_{N+\overline{\Delta N}} = d(t_{N+\overline{\Delta N}})$ as the damage values before and after the jump, respectively.

3.1. Non-linear extrapolation scheme

The present method is based on the damage law property according to which static and fatigue damage contributions cannot evolve at the same time. Therefore, the damage evolution law (Eq.(6)) can be rewritten as follows, considering that $\mathcal{M}$ represents the static (monotonic) contribution to damage and $\mathcal{F}$ the fatigue contribution.

$$\dot{d} = (d_\infty - d)\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}t} + (d_\infty - d)^\gamma \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}t} \tag{9}$$

$$\text{with} \quad \begin{aligned} \frac{\mathrm{d}\mathcal{M}}{\mathrm{d}t} &= \left\langle \frac{\varepsilon_{eq}^{max} - \varepsilon_{eq}^{0s}}{S^s} \right\rangle_+^{s^s} \frac{\mathrm{d}\varepsilon_{eq}^{\max}}{\mathrm{d}t} \\ \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}t} &= \left\langle \frac{\varepsilon_{eq} - m\breve{\varepsilon}_{eq} - \varepsilon^{0f}}{S^f} \right\rangle_+^{s^f} \left\langle \frac{\mathrm{d}\varepsilon_{eq}}{\mathrm{d}t} \right\rangle_+ \end{aligned} \tag{10}$$

Let us remind, as expressed in Eq. (11), that when the fatigue part is activated, there is no static damage, which justifies the modification of the fatigue damage evolution law in Eq. (10):

$$\begin{cases} \dot{\varepsilon}_{eq}^{max} \neq 0 \Rightarrow \frac{\mathrm{d}\mathcal{M}}{\mathrm{d}t} > 0 \Rightarrow \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}t} = 0 \\ \dot{\varepsilon}_{eq}^{max} = 0 \Rightarrow \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}t} > 0 \Rightarrow \frac{\mathrm{d}\mathcal{M}}{\mathrm{d}t} = 0 \end{cases} \tag{11}$$

The same idea of contribution separation is applied during the jumps. Indeed, after some training cycles, the damage growth per cycle is very low or constant (assumption often relevant in fatigue [35]). Moreover, in cyclic loadings, the equivalent strain mean, $\breve{\varepsilon}_{eq}$, defined by Eq.(4), has reached or is close to the classical mean value, $\bar{\varepsilon}_{eq}$ (Eq.(5)).

When its value is assumed constant during a cycle, this allows for a closed form integration of $\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}t}$ over a cycle (please refer to AppendixB for analytical calculations of $\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}$ and $\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}$ for sine/triangular periodic loadings, of period T). In the general case, however, the calculation over a cycle of the quantities $\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}$ and $\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}$ is performed numerically, time step by time step over the cycle preceding the jump (between times t_{N-1} and t_N), with no simplifying assumption. These quantities represent the increase in both static and fatigue parts of damage per cycle.

The damage increment per cycle (Eq.(12)) is therefore obtained by integration of Eq.(9) over one cycle,

$$\frac{\mathrm{d}d}{\mathrm{d}N} \approx (d_\infty - d)\,\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} + (d_\infty - d)^\gamma \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N} \tag{12}$$

In the numerical scheme, the quantities $\mathcal{M}$ and $\mathcal{F}$ are calculated incrementally, summing the contributions of every computed time step, as internal variables of time integration of standard subroutines. Their evolutions per cycle, $\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}$ and $\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}$, calculated during the last cycle before the jump, are then considered constant during the whole $\overline{\Delta N}$ cycles of the jump. This is the main hypothesis allowing for analytical calculation of the damage variable, by resolution of the following simple equation,

$$\overline{\Delta N} = \int_{N}^{N+\overline{\Delta N}} \mathrm{d}N = \int_{d^N}^{d_{N+\overline{\Delta N}}} \frac{\mathrm{d}d}{(d_\infty - d)\,\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} + (d_\infty - d)^\gamma\,\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}} \tag{13}$$

As the dependency on the damage itself is kept, note that $\overline{\Delta N}$ is in fact nonlinear in d during the cycle-jump.

Two cases are distinguished:

- If the simulation is strain-controlled, static damage cannot be generated ($\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} = 0$ and $\varepsilon_{eq,N}^{max} = \varepsilon_{eq,N-1}^{max}$). Moreover, the assumption that $\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}$ is constant during the jump is ensured. Indeed, the mean equivalent strain converges very quickly towards the mean value of a cycle and remains so as long as the equivalent strain is cyclic. In this case, the extrapolation is quasi-exact, as it will be illustrated in section 4.1.
- On the contrary, for stress-controlled fatigue, both the maximal equivalent strain and the mean equivalent strain increase, due to damage, even though they are considered constant during the jump. The fatigue damage increment, $\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}$, is then not constant over the jump and our approximation induces an error directly related to the size of the jump.

Details of the integration of Eq.(13) are presented in AppendixC. Eq.(14) and Eq.(15) give the two separate closed-form solutions, which provide the damage value after the cycle-jump, depending on whether or not new static damage is generated.

If $\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} = 0$,

$$d_{N+\overline{\Delta N}} = d_\infty - \left[-\overline{\Delta N}(1-\gamma)\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N} + (d_\infty - d_N)^{1-\gamma}\right]^{\frac{1}{1-\gamma}} \tag{14}$$

If $\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} > 0$,

$$d^{N+\overline{\Delta N}} = d_\infty - \left[\exp\left(-\overline{\Delta N}(1-\gamma)\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}\right)\left((d_\infty - d^N)^{1-\gamma} + \frac{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}}{\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}}\right) - \frac{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}}{\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}}\right]^{\frac{1}{1-\gamma}} \tag{15}$$

Due to the closed-form integration of the extrapolated damage state, convergence is achieved with a single increment. Let us remark that, in both cases,

damage can only increase: $d_{N+\overline{\Delta N}} \geq d_N$. Moreover, the damage value cannot exceed the saturation value: $d_{N+\overline{\Delta N}} \leq d_\infty$. The latter statement ensures the convergence of the simulation.

In all the applications described in section 4, the jump lengths, $\overline{\Delta N}$, are prescribed and constant throughout the simulations. Indeed, the aim of this paper is to evaluate the efficiency of the proposed strategy and to compare it with another cycle-jump method, for equivalent computational costs, *i.e.* for the same number of simulated cycles and, consequently, identical jump lengths. However, several authors [1, 15, 16] pilot the jump length by its damage increment, Δd, to prevent jeopardizing the accuracy of the cycle-jumps. The present method can also achieve such a jump length control: Eq.(14) and (15) can be inverted to obtain $\overline{\Delta N}$, for a given user-prescribed value of damage increment: $\Delta d = d_{N+\overline{\Delta N}} - d_N$.

3.2. Implementation of the strategy into a Finite Element code

This section aims at explaining the key points of the implementation of the non-linear cycle-jump procedure in a FE code. Let us recall that the present kinetic damage model is formulated as standard (thermodynamics) rate form constitutive equations. They are first implemented as a behaviour law subroutine, using the kinetic constitutive equations presented in section 2. The model is thus compatible with a multithreading computational strategy.

The proposed cycle-jump procedure is sufficiently general to be relevant either for a prescribed constant cycle-jump length, $\overline{\Delta N}$, (in terms of number of cycles) or for an updated cycle-jump length, calculated from a prescribed damage increment, Δd. Originality of the method, it is directly implemented inside the behaviour law subroutine and does not need to create any additional file with the Gauss points informations.

The flow chart reported in Fig. 2 presents the different steps of the integration of the behaviour law at local scale, including the cycle-jump procedure. The damage cycle-jump procedure can be decomposed into two steps, which can be illustrated through a simple uniaxial cyclic loading as reported in Fig. 3.

On the one hand, during the training and the control cycles, a program (the behaviour law subroutine) integrates the constitutive equations, including damage. Its inputs are the usual inputs of standard rate form constitutive equations (material properties, time, strain tensor, strain increment and state variables at the previous converged increment). At the end of each cycle and once global equilibrium is satisfied, the next increment jump length, $\overline{\Delta N}$, may be updated. Indeed, this jump length can either be a fixed user input or, better, can be derived from the past state variables, as detailed in AppendixC. In any case, its value is applied over the whole structure. It takes information from the Gauss point with the highest damage evolution during the control cycles, more specifically of its updated jump length, calculated locally in the behaviour law subroutine (dashed green box in Fig. 2). Moreover, during this step, the increments per cycle $\frac{\mathrm{d}\mathcal{M}_k}{\mathrm{d}N}$ and $\frac{\mathrm{d}\mathcal{F}_k}{\mathrm{d}N}$ (defined by Eq. (16), for each degradation mechanism k) are estimated. The outputs are therefore the

updated state variables, the stress tensor and the consistent tangent matrix, which are standard finite element outputs, but also the updated jump length $\overline{\Delta N}$, and $\frac{\mathrm{d}\mathcal{M}_k}{\mathrm{d}N}$ and $\frac{\mathrm{d}\mathcal{F}_k}{\mathrm{d}N}$, which are mandatory for the cycle-jump procedure described in the second step.

On the other hand, during the cycle jump, the behaviour law subroutine also integrates the constitutive equations, including damage, but the damage evolution law is estimated using the non linear extrapolation scheme detailed in section 3.1. Its inputs consist first of usual inputs of standard rate form constitutive equations, second of the —possibly complex— loading period, T, the jump size $\overline{\Delta N}$ and, novelty, $\frac{\mathrm{d}\mathcal{M}_k}{\mathrm{d}N}$ and $\frac{\mathrm{d}\mathcal{F}_k}{\mathrm{d}N}$ (which are assumed constant during the cycle jump). The outputs are the updated state variables, the stress tensor and the consistent tangent matrix. Note that the tangent matrix is given in a closed form.

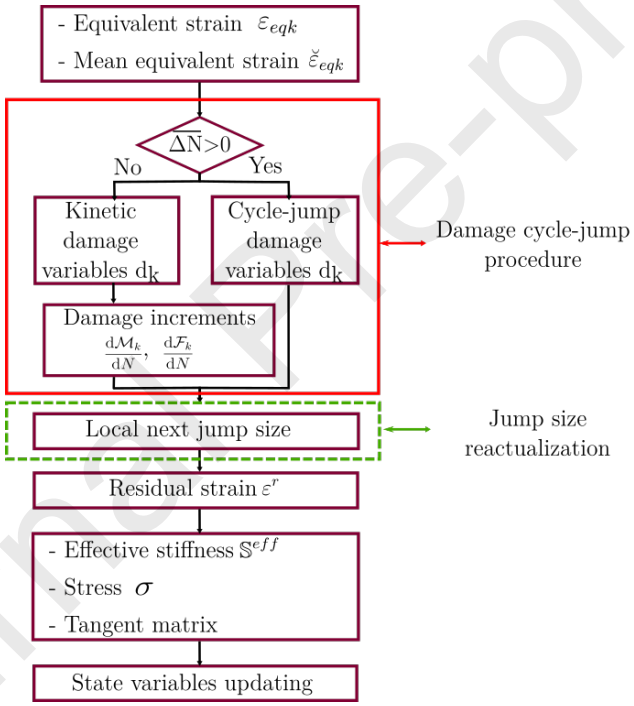


Figure 2: Numerical algorithm used to integrate the constitutive equations (including damage), either by time-step by time-step or by cycle-jump procedure.

The damage extrapolation is performed at the local scale in the damage cycle-jump procedure (continuous red box in Fig.2). The analysis automatically switches between a kinetic resolution or a cycle-jump, depending on the value of the jump length, $\overline{\Delta N}$, which is an input data for the local program integrating the constitutive equations.

- When $\overline{\Delta N} = 0$, the simulation uses kinetic (rate) equations and damage

is calculated by Eq.(6).

- Otherwise, if $\overline{\Delta N} > 0$, then a cycle-jump is conducted. Damage is calculated using either Eq.(14) or Eq.(15). The increments of static and fatigue damage, $\frac{\mathrm{d}\mathcal{M}_k}{\mathrm{d}N}$ and $\frac{\mathrm{d}\mathcal{F}_k}{\mathrm{d}N}$, for each mechanism, k, are calculated over the cycle prior to the jump, characterized by two time indicators defining its beginning and its end, as

$$\frac{\mathrm{d}\mathcal{M}_k}{\mathrm{d}N} = \mathcal{M}(t_N) - \mathcal{M}(t_{N-1}), \qquad \frac{\mathrm{d}\mathcal{F}_k}{\mathrm{d}N} = \mathcal{F}(t_N) - \mathcal{F}(t_{N-1}) \tag{16}$$

  where t_N (resp. t_{N-1}) is the time at the end (resp. the beginning) of cycle N, $\Delta\mathcal{M}(t_N)$, $\Delta\mathcal{F}(t_N)$ being numerically updated at each time step, at each Gauss point. The damage variables, $\mathcal{M}_k$ (resp. $\mathcal{F}_k$), and its increment over a cycle, $\frac{\mathrm{d}\mathcal{M}_k}{\mathrm{d}N}$ (resp. $\frac{\mathrm{d}\mathcal{F}_k}{\mathrm{d}N}$), are stored at each time step.

The extrapolation during the jump is conducted on the damage variables. However, since the residual strain and the effective stiffness depend solely on the damage variables (see Eq.(8) and Eq.(2)), they are automatically updated. The stress is then calculated using Eq.(1). The FE solver, in which the input data is the strain tensor, iterates until equilibrium is reached.

To sum up, the stored variables useful from pass to pass are for each integration point: (i) the equivalent strains, $\varepsilon_{eq,k}$, (ii) the mean equivalent strains, $\breve{\varepsilon}_{eq,k}$, (iii) the damage variables, d_k, and (iv) the residual strains, $\boldsymbol{\varepsilon^r}$. The additional state variables required by the cycle-jump procedure are static and fatigue damage variables ($\mathcal{M}_k$ and $\mathcal{F}_k$) and static and fatigue damage increments ($\frac{\mathrm{d}\mathcal{M}_k}{\mathrm{d}N}$ and $\frac{\mathrm{d}\mathcal{F}_k}{\mathrm{d}N}$).

Particular attention has been paid to the numerical scheme, in order to minimize the computational costs. Data exchange between programs is performed through global variables. In particular, the jump size, $\overline{\Delta N}$, and the time indicators are frequently updated and must be accessible from all considered programs of the FE code.

The FE model is decomposed into several steps, according to the different loading phases presented in section 3. Fig. 3 is a schematic representation of the applied loading. It is illustrated on a triangular loading, but a more complex loading could be considered, without any change in the implementation. In this paper, the number of control cycles is taken equal to 2, the first cycle is performed to take into account the load transfers within the structure and the second to actualize the damage increments.

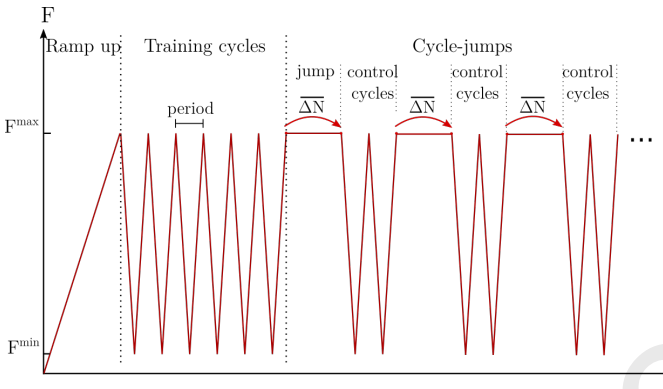


Figure 3: Example of prescribed loading for simulations with cycle-jumps.

Thus, the implementation of the cycle-jump procedure is non-intrusive, as everything is programmed at the local scale, and generalisable for several jump length calculation methods. There is, moreover, no need to restart the analysis. The post-processing is standard.

4. Applications

An advantage of using a kinetic model consists in being able to simulate continuously (time-step by time-step in a FE code) every cycle. Consequently, a simulation computing all the cycles is exact from a numerical point of view. It can be considered as a reference against which simulations with computational strategies can be compared. The proposed method of cycle-jumps is applied to test cases of increasing complexity.

4.1. Application to simple cyclic loadings at the material scale

The simulations with cycle-jumps are compared to the reference simulation, as illustrated in Fig. 4.

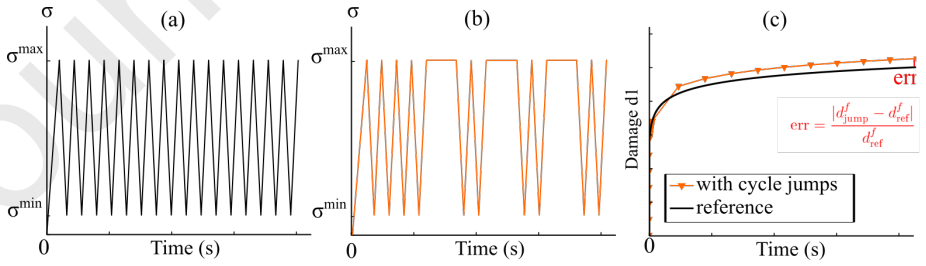


Figure 4: Schematic description of the comparison between (a) a numerical reference and (b) a simulation with jumps. The relative error due to jumps is reported in (c).

The relative error is calculated through Eq.(17), using the final value of damage, because it affects failure mechanisms and seems critical for the fatigue

lifetime of structures.

$$\text{err} = \frac{|d^f_{\text{jump}} - d^f_{\text{ref}}|}{d^f_{\text{ref}}} \tag{17}$$

where d^f_{ref} and d^f_{jump} correspond, respectively, to the final values of damage in the reference simulation and in a simulation with cycle-jumps. All comparisons could be made with the classical Root Mean Square Error (RMSE) indicator, with unchanged conclusions.

This comparison is conducted on simulations with the proposed cycle-jump method as well as with a classical cycle-jump method, which extrapolation scheme is based on a first order integration (Eq.(18)), issued from a Taylor expansion or an explicit Euler's formula [1, 16].

$$d_{N+\overline{\Delta N}} = d_N + \frac{\mathrm{d}d}{\mathrm{d}N}\overline{\Delta N} \tag{18}$$

$$\text{with} \qquad \frac{\mathrm{d}d}{\mathrm{d}N} = \int_{t_{N-1}}^{t_N} \frac{\mathrm{d}d}{\mathrm{d}t} dt = d_N - d_{N-1}$$

The first application case is an integration point subjected to simple cyclic loadings (tension-tension fatigue) at 1 Hz. The loading is either strain or stress-controlled. For each configuration, the loading has been chosen to reach the same final value of damage ($d \equiv 0.2$).The load ratio is fixed at $R_{\sigma/\varepsilon} = 0.05$. All simulations are run with Matlab on the same desktop computer with 1 Central Processing Unit (CPU), for comparison of the computational costs. In each case, the reference is a simulation where all the cycles are simulated without jump ($\overline{\Delta N} = 0$).

To facilitate the comparison of the methods, the simulations are performed with a prescribed and constant jump length, $\overline{\Delta N}$. The cycle-jumps start after the 100^{th} cycle and each jump is followed by two control cycles to update the static and fatigue damage increments.

4.1.1. Imposed strain cyclic loading

The method is first of all applied to strain-controlled cyclic loadings. In that case, the model being written in total strain, there is no creation of static damage during fatigue. Fig. 5a and 5b show that an accurate value of damage can be reached, even with a single jump, for two cyclic loadings with two different maximum strains. That corroborates the statement reported in section 3, according to which the extrapolation is quasi-exact under strain-controlled fatigue. To be more quantitative, the relation between jump length and error for both extrapolation methods is also illustrated (Fig. 5c and 5d). The aberrant results, obtained with the first order extrapolation, are banned from the graphs. Indeed, nothing prevents the final damage value, $d_{N+\overline{\Delta N}}$, to exceed the saturation level, d_∞, in Eq.(18), for large jump lengths, $\overline{\Delta N}$.

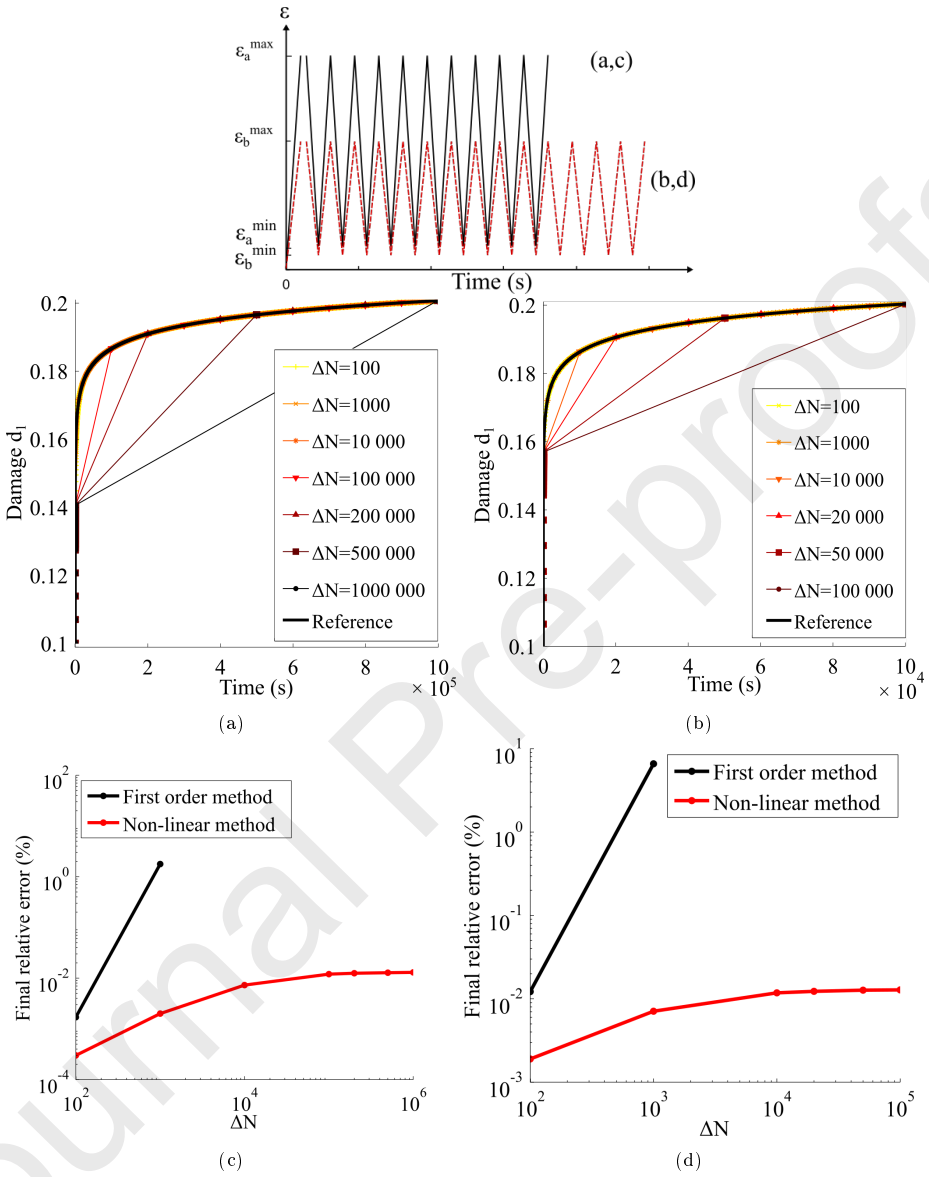
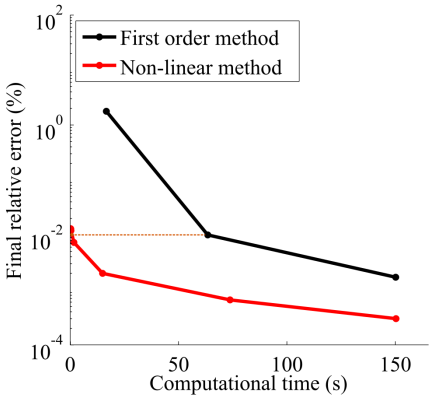


Figure 5: Cycle-jump procedure applied to pure fatigue loading at several maximum applied strains (a,c) 0.78×10^{-3} (10^6 cycles), (b,d) 0.88×10^{-3} (10^5 cycles). (a) and (b): damage evolution of the reference and of the cycle-jump simulations. (c) and (d): evolution of the relative error as a function of the prescribed jump length. The red dots correspond to simulations with non-linear cycle jumps and the black dots to the same simulations with the first order extrapolation, banning those presenting aberrant results.

As the errors remain within the scope of numerical noise with the proposed method, simulations are both more precise and a lot faster than the classical

first order method (see Fig. 6).



err (%)	1st order ΔN	1st order t(s)	Non-linear ΔN	Non-linear t(s)	Time gain(%)
0.01	220	63.46	9×10^{5}	0.25	253

Figure 6: Efficiency of the proposed cycle-jump method as a function of computational time for the 0.88×10^{-3} strain-controlled loading case (Fig.5b).

4.1.2. Imposed stress cyclic loading

The same procedure is then applied to stress-controlled cyclic loadings (see Fig. 7). The conclusions are similar to the imposed strain case: the simulations with cycle-jumps (i) overtake the reference simulation (in black), (ii) converge towards the reference with decreasing jump lengths and (iii) even with large blocks of skipped cycles, the final damage value does not exceed the saturation level. To ensure satisfactory results, the user has, in this case, to choose an appropriate value of jump length or to prescribe varying jump lengths throughout the simulation.

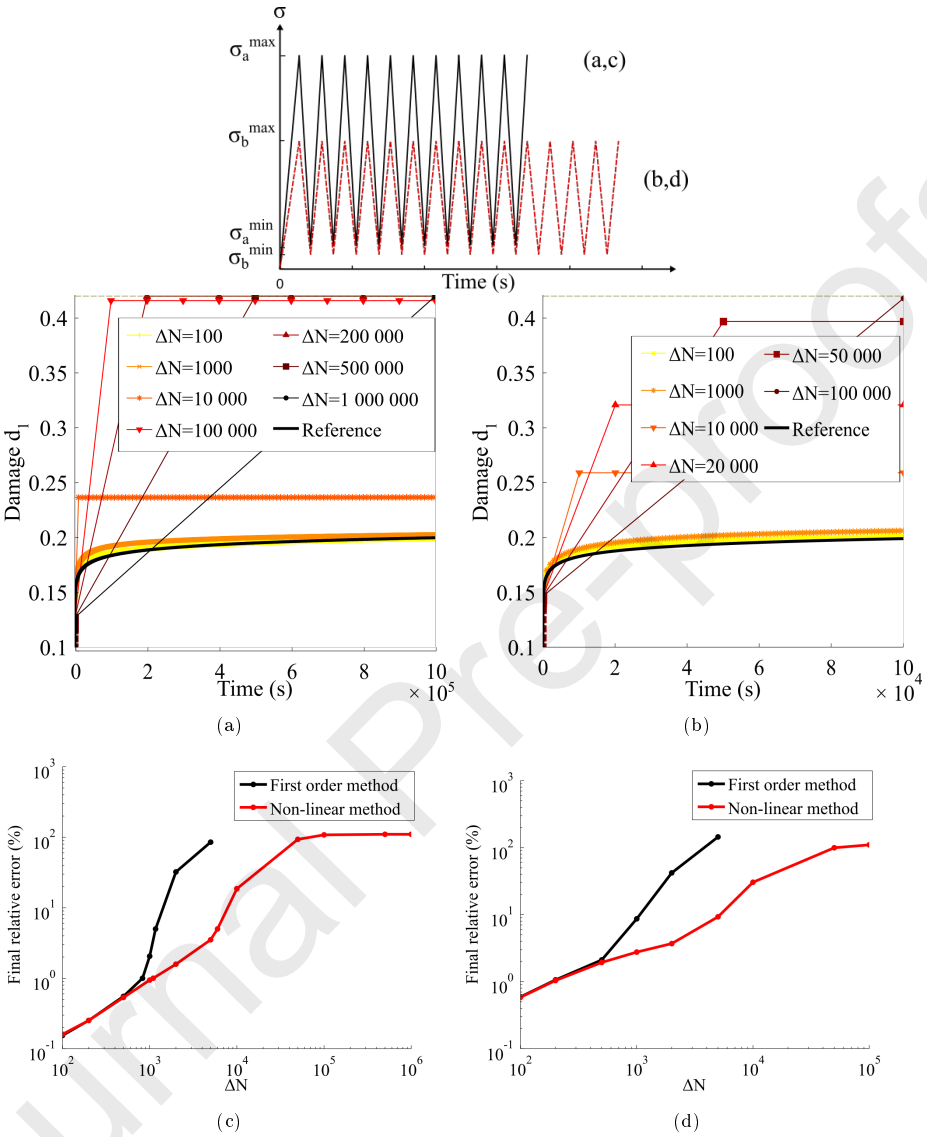
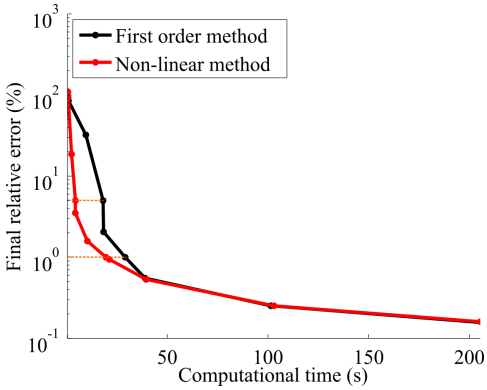


Figure 7: Cycle-jump procedure applied to fatigue loading at several maximum applied stresses: (a,c) 80 MPa (10^6 cycles), (b,d) 90 MPa (10^5 cycles). (a) and (b): damage evolution of the reference and of the cycle-jump simulations. (c) and (d): evolution of the relative error as a function of the prescribed jump length. The red dots correspond to simulations with non-linear cycle-jumps and the black dots to the same simulations with the first order extrapolation, banning those presenting aberrant results.

All the simulations with cycle-jumps induce an error related to the jump length (see Fig. 7c and 7d), even with a non-linear extrapolation because the assumption of constant damage increments, $\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}$ and $\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}$, is not ensured.

Fig. 8 shows the trade-off between computational time and accuracy for

both extrapolation methods. The benefit of the proposed method is slight for very short jump lengths but is still interesting for large jump lengths. For a given allowable error, the relevant jump length is larger with non-linear cycle-jumps and simulations run faster. It is illustrated in Fig. 8 for 1% and 5% errors. Conversely, if the computational time is set, the proposed method is more precise for large jump lengths.



err (%)	1st order ΔN	1st order t(s)	Non-linear ΔN	Non-linear t(s)	Time gain(%)
1	830	29.0	1080	21.0	27
5	1175	18.5	6000	4.4	76

Figure 8: Efficiency of the proposed cycle-jump method as a function of computational time for the 90 MPa stress-controlled loading case (Fig.7b).

For simple periodic fatigue loadings, the proposed cycle-jump procedure can describe the non-linearities of the damage curves quite accurately and reduces computational cost efficiently. Moreover, convergence is achieved for any loading and any prescribed number of cycles, which is not necessarily the case with the classical first order method.

4.2. Application to an open-hole plate under simple periodic loadings

In this section, an open-hole plate subjected to tensile loading is considered. It is representative of a more complex structure because load transfers and stress concentrations must be taken into account. It is constituted of a single composite ply, oriented at 45°, in order to highlight the fact that damage is oriented by the microstructure. Moreover, this orientation activates all damage mechanisms at the same time in the whole structure, thus constituting a discriminant test case. The boundary conditions consist in clamping the plate on one side and subjecting it to an effort on the opposite side, as illustrated in Fig. 9:

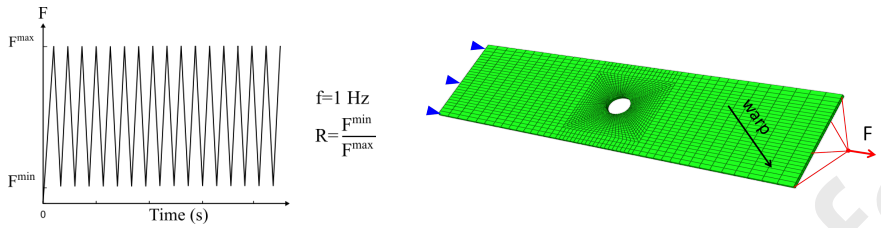


Figure 9: Finite element model of an open-hole plate used as a simple structure. The dimensions of the plate are 100 mm × 30 mm × 1 mm with a 2.18 mm in diameter central hole. The mesh is constituted with one layer consisting of 2500 hexahedral elements and 5232 nodes. The total amount of degrees of freedom reaches 15696.

From now on, the non-linear extrapolation is used for cycle-jumps, since its performance was demonstrated on simpler examples in section 4.1. The performance of the cycle-jump procedure is assessed in terms of precision as well as computational time gain with respect to a reference cycle-by-cycle simulation. The relative error in the damage variable, d_k, calculated for each integration point, i, is given by:

$$\mathrm{err}(\%) = \frac{|d^f_{k,\mathrm{jump}} - d^f_{k,\mathrm{ref}}|}{\max_i d^f_{k,\mathrm{ref}} - \min_i d^f_{k,\mathrm{ref}}} \times 100 \tag{19}$$

The open-hole plate is subjected to simple cyclic loadings (with triangular waves). The load is applied through a concentrated force and is representative of current experimental tests. The load evolves between 0N and 900N ($R_F = 0$) during 1021 cycles at 1Hz.

The proposed non-linear cycle-jump procedure is applied to this test case. The jumps, 48 cycles in length ($\overline{\Delta N} = 48$), start after 20 training cycles and are followed by 2 control cycles. Fig. 10 presents the damage field, d_1, at the final time of the reference (left) and at the final time of the cycle-jump simulation (right). The damage scaling is the same for the two simulations. The relative error, which map is also represented in Fig. 10, does not exceed 6.1% in the damage variable, d_1. The maximal error is located at the location of maximum damage, *i.e.* at the stress concentration points.

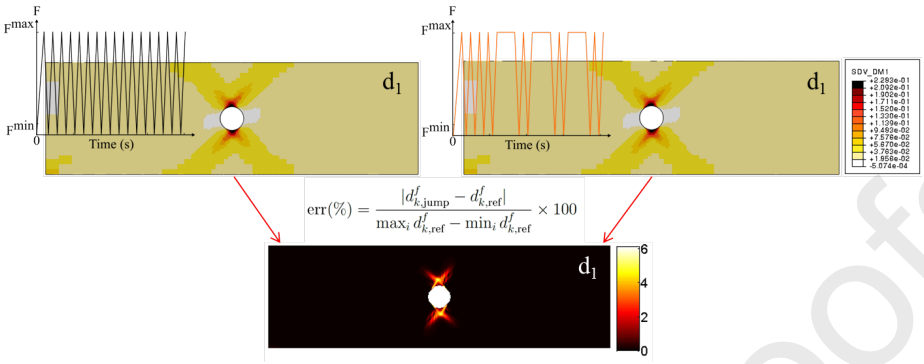


Figure 10: Comparison of the damage fields, d_1, after 1021 cycles and the associated error field.

Concerning the simulation cost, the reference lasts 12618 seconds (about 3.5 hours) with 4 CPUs but only 641 seconds for the simulation with jumps of 50 cycles. Once the training cycles are run, the control cycles account for most of the remaining computational time (88% for the simulation with 50-cycle long jumps). The gain of time is expected to increase with increasing jump lengths and it corroborates with the obtained results. Table 1 provides the maximal relative error among integration points and time gain with respect to the reference simulation, for simulations with three different jump lengths.

$\overline{\Delta N}$	err. max. on d_1	speed-up factor
100	10.67%	34
50	6.07 %	20
20	3.36 %	10

Table 1: Error and time gain in simulation with cycle-jumps with respect to the reference simulation.

The more cycles are skipped, the more time gain is obtained. Therefore, the benefit of the procedure should be more important for simulation with higher cycle numbers. These conclusions have been confirmed by performing fatigue simulations on open-hole plates subjected to 10 000 cycles. The reference lasts 178771 seconds (about 49 hours) while only 6169 seconds for a simulation with 50 cycles long jumps, meaning a speed up by a factor of 29.

This method of non-linear cycle-jumps is consequently able to accelerate fatigue simulations with an acceptable precision. It is, in addition, able to predict the behaviour of composite structures under high cycle fatigue loadings.

4.3. Application to quasi-random loadings

The methodology is finally applied to more complex periodic loadings, that are denoted as quasi-random loadings. It consists in a random loading sequence, called unit pattern, which is representative of a flight loading history. To predict

fatigue lifetime of composite structures, thus considering several flights, this unit pattern is repeated over time. An example of such a loading is illustrated in Fig. 11.

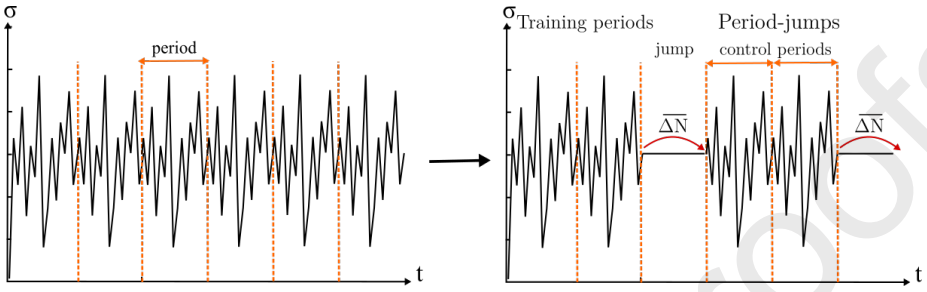


Figure 11: Example of quasi-random loading, with period-jumps of constant length, $\overline{\Delta N}$.

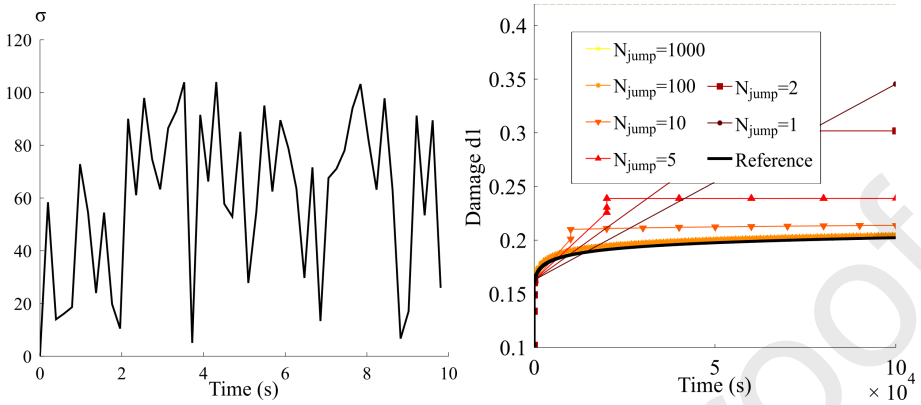
The advantage of using a kinetic model is its capacity to consider random loadings. As the signal is repeated, the cycle-jump procedure can be applied. The model should then be able to compute the damage evolution during quasi-random loadings.

First, an integration point (at the local scale) and, then, an open-hole plate, both subjected to tensile fatigue loadings, are considered.

4.3.1. Local scale

As previously (section 4.1), it was demonstrated that the non-linear cycle-jump procedure is quasi-exact for strain-controlled quasi-random loading, as long as the cycle-jumps start after at least 2 periods (see the black curve in Fig. 13).

We focus now on a stress-controlled loading. Its unit pattern, represented in Fig. 12a, has a period $T = 10$s and is repeated 10^4 times. The period-jumps start after 13 periods and are followed by two control periods. Fig. 12b presents the results of the period-jump simulations and Fig. 13 quantifies the relative error with respect to the reference simulation, where all the periods are simulated, for both strain and stress-controlled loadings.



(a) (b)

Figure 12: (a) Considered stress unit loading pattern, (b) Damage evolution during period-jump simulation.

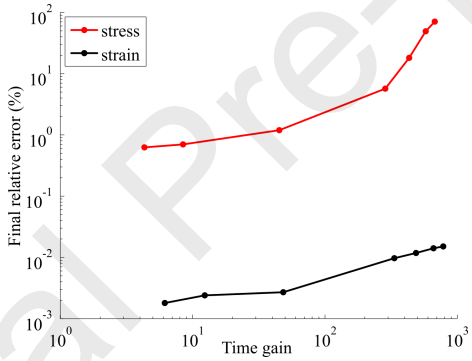


Figure 13: Relative error in the period-jump simulations versus the computational time gain with respect to the reference simulations for both strain and stress-controlled loadings.

This study demonstrates that the proposed cycle-jump method is applicable to all kind of periodic fatigue loadings.

4.3.2. Structural scale

A quasi-random load is applied to an open-hole plate, with the boundary conditions presented in section 4.2. The load unit sequence of 20 points is reported in Fig. 14a and repeated 103 times.

The reference simulation lasts 9015 seconds (about 2.5 hours) using 4 CPUs. The cycle-jump procedure is then applied to the same test case. The jumps start after the 4^{th} period, because the damage variable evolution is already slowing, and are followed by 2 control periods. The jump length, constant on the whole simulation, is set at 9 periods. Fig.14b presents the error maps of the damage fields. The simulation lasts 870 seconds using 4 CPUs. Most of

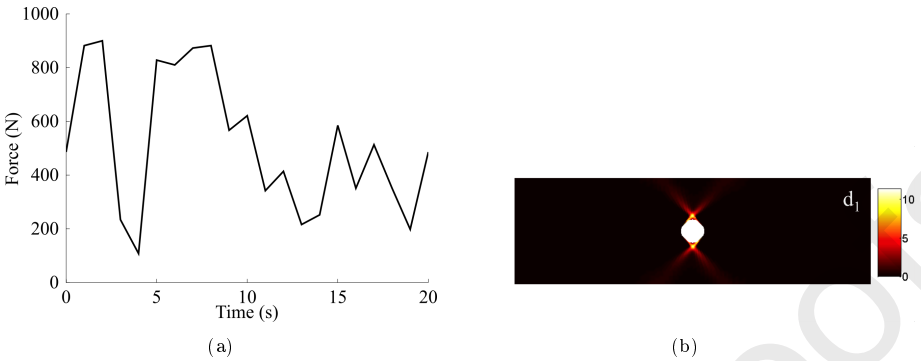


Figure 14: Application of the period-jump procedure to an academical structural test case. (a) Unit loading sequence. (b) Relative error map (%) of damage variable, d_1, between the reference simulation and a period-jump simulation after 103 periods.

the computational time is devoted to the training and control cycles (each unit sequence lasts approximately 35 seconds). Thus the time gain of about a factor of 40 for 100 cycles, will be higher for larger jump lengths and longer time loadings.

The kinetic model enables us to treat random loadings and this example illustrates the possibility of applying the proposed cycle-jump procedure to accelerate fatigue simulations of composite structures subjected to complex quasi-random loadings.

5. Conclusion and perspectives

This paper presents a method to reduce the simulation time of structures subjected to fatigue loadings, either simple periodic (triangular or sine waves) or more complex ones (quasi-random).

It combines a cycle-jump technique with finite element simulations (and from a numerical point of view within a commercial FE code). The key point of the method is its extrapolation scheme. It has been designed specifically for kinetic damage models, enabling a non-linear extrapolation of the damage variables. The non-linearities in the composite behaviour are accurately described.

The accuracy and performance of the cycle-jump procedure is assessed by comparison with reference simulations where all cycles are explicitly computed. However, no prior result of such simulations are required before performing cycle-jump simulations.

The method remains close to the cycle-by-cycle simulations even for complex loadings, while reducing markedly the computational costs. Illustrated with constant jump lengths, it has been implemented into a FE code for constant or variable jump lengths.

This extrapolation scheme provides promising satisfactory results in fatigue, both at the integration point scale and at the structural scale. This method can

consequently give reliable results for high cycle fatigue loadings, while drastically reducing the computational costs.

6. Acknowledgements

The collaboration with SAFRAN Ceramics is gratefully acknowledged. This work was supported under PRC MECACOMP, a French research project co-funded by DGAC and SAFRAN Group, piloted by SAFRAN Group and involving SAFRAN Group, ONERA and CNRS. The authors would like to express their sincere gratitude to Dr. R. Valle, for valuable and helpful discussions.

7. References

[1] W. Van Paepegem, J. Degrieck, P. De Baets, Finite element approach for modelling fatigue damage in fibre-reinforced composite materials, Composites Part B: Engineering 32 (7) (2001) 575–588.

[2] W. Van Paepegem, Development and finite element implementation of a damage model for fatigue of fibre-reinforced polymers, Doctorate thesis, Ghent University, Belgium (2002).

[3] C. Hochard, Y. Thollon, A generalized damage model for woven ply laminates under static and fatigue loading conditions, International Journal of Fatigue 32 (2010) 158–165.

[4] N. Revest, Comportement en fatigue de pièces épaisses en matériaux composites, Doctorate thesis, Ecole Nationale Supérieure des Mines de Paris, France (2011).
URL `https://pastel.archives-ouvertes.fr/pastel-00677076/document`

[5] C. Rakotoarisoa, Prévision de la durée de vie en fatigue des composites à matrice organique tissés interlock, Doctorate thesis, University of Technology of Compiègne, France (2014).

[6] L. Gornet, H. Ijaz, A high-cyclic elastic fatigue damage model for carbon fibre epoxy matrix laminates with different mode mixtures, Composites Part B: Engineering 42 (2011) 1173–1180.

[7] R. Desmorat, L. Angrand, P. Gaborit, M. Kaminski, C. Rakotoarisoa, On the introduction of a mean stress in kinetic damage evolution laws for fatigue, International Journal of Fatigue 77 (2015) 141–153.

[8] L. Angrand, Modèle d'endommagement incrémental en temps pour la prévision de la durée de vie des composites tissés 3D en fatigue cyclique et en fatigue aléatoire, Doctorate thesis, University of Paris-Saclay, France (2016).
URL `https://tel.archives-ouvertes.fr/tel-01300513/document`

[9] J. Lemaitre, J.-L. Chaboche, Mécanique des matériaux solides, Dunod, Paris (english translation 'Mechanics of solid materials', Cambridge University Press, 1990), 1985.

[10] D. Ryckelynck, D. Missoum Benziane, S. Cartel, J. Besson, A robust adaptive model reduction method for damage simulations, Computational Materials Science 50 (2011) 1597–1605.

[11] J.-Y. Cognard, P. Ladeveze, A large time increment approach for cyclic viscoplasticity, International Journal of Plasticity 9 (1993) 141–157.

[12] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on Proper Generalized Decomposition, Archives of Computational Methods in Engineering 18 (2011) 395–404.

[13] M. Bhattacharyya, A model reduction approach in space and time for fatigue damage simulation, Doctorate thesis, University of Paris-Saclay, France (2018).
URL https://tel.archives-ouvertes.fr/tel-01808371/document

[14] M. Bhattacharyya, A. Fau, R. Desmorat, S. Alameddin, D. Néron, P. Ladevèze, U. Nackenhorst, A kinetic two-scale damage model for high-cycle fatigue simulation using multi-temporal Latin framework, European Journal of Mechanics - A/Solids 77 (2019) 103808.

[15] P.-M. Lesne, S. Savalle, An efficient cycles jump technique for viscoplastic structure calculations involving large number of cycles, in: 2nd International Conference on "Computational Plasticity" Models Software and Applications, Barcelona, Spain, 1989.

[16] J. Lemaitre, I. Doghri, Damage 90: a post processor for crack initiation, Computer Methods in Applied Mechanics and Engineering 115 (1994) 197–232.

[17] K. Sai, Modèles à grand nombre de variables internes et méthodes numériques associées, Doctorate thesis, Ecole Nationale Supérieure des Mines de Paris, France (1993).
URL https://pastel.archives-ouvertes.fr/pastel-00001728/document

[18] H. Kiewel, J. Aktaa, D. Munz, Application of an extrapolation method in thermocyclic failure analysis, Computer Methods in Applied Mechanics and Engineering 182 (2000) 55–71.

[19] J. Llobet, P. Maimí, A. Turon, B. L. V. Bak, E. Lindgaard, Y. Essa, F. Martin de la Escalera, Progressive damage modelling of notched carbon/epoxy laminates under tensile fatigue loadings, in: 18th Europoean Conference on Composite Materials, Athens, Greece, 2018.

[20] J. Fish, Q. Yu, Computational mechanics of fatigue and life predictions for composite materials and structures, Computer Methods in Applied Mechanics and Engineering 191 (2002) 4827–4849.

[21] D. Cojocaru, A. Karlsson, A simple numerical method of cycle jumps for cyclically loaded structures, International Journal of Fatigue 28 (2006) 1677–1689.

[22] L. Marcin, Modélisation du comportement, de l'endommagement et de la rupture de matériaux composites à renforts tissés pour le dimensionnement robuste de structures, Doctorate thesis, University of Bordeaux I, France (2010).
URL https://tel.archives-ouvertes.fr/tel-00481601/document

[23] E. Hemon, Modèles multi-niveaux de prévision des durées de vie en fatigue des structures composites à matrice céramique pour usage en turbomachines aéronautiques, Doctorate thesis, University of Bordeaux I, France (2013).
URL https://tel.archives-ouvertes.fr/tel-01020093/document

[24] C. Ben Ramdane, Etude et modélisation du comportement mécanique de CMC oxyde/oxyde, Doctorate thesis, University of Bordeaux, France (2014).
URL https://hal.archives-ouvertes.fr/tel-01113148/document

[25] C. Rakotoarisoa, F. Laurin, M. Hirsekorn, J.-F. Maire, L. Olivier, Development of a fatigue model for 3D woven polymer matrix composites based on a damage model, in: 15$^{\text{th}}$ European Conference on Composite Materials, Venice, Italy, 2012.

[26] J. Ju, On energy-based coupled elastoplastic damage theories: constitutives modeling and computational aspects, International Journal of Solids and Structures 25 (1989) 803–833.

[27] N. Revest, A. Thionnet, J. Renard, L. Boulay, P. Castaing, Fatigue behaviour of composite structures, in: 16$^{\text{ième}}$ Journées Nationales des Composites, Toulouse, France, 2009.

[28] O. Sally, C. Julien, F. Laurin, R. Desmorat, F. Bouillon, Fatigue lifetime modeling of oxide/oxide composites, Procedia Engineering 213 (2018) 797–803.

[29] D. Perreux, D. Oytana, Continuum damage mechanics for microcraked composites, Composites Engineering 3 (1) (1993) 115–122.

[30] F. Laurin, M. Kaminski, L. Angrand, R. Desmorat, Proposition of a unified model to predict strength and fatigue lifetime of 3D woven composite structures with polymer matrix, in: 18$^{\text{th}}$ European Conference on Composite Materials, Athens, Greece, 2018.

[31] C. Ben Ramdane, A. Julian-Jankowiak, R. Valle, Y. Renollet, M. Palier, E. Martin, P. Diss, Microstructure and mechanical behaviour of a Nextel^{TM}610/alumina weak matrix composite subjected to tensile and compressive loadings, Journal of the European Ceramic Society 37 (2017) 2919–2932.

[32] M. Ruggles-Wrenn, J. Braun, Effects of steam environment on creep behavior of Nextel^{TM}720/alumina ceramic composite at elevated temperature, Materials Science and Engineering A. 497 (2008) 101–110.

[33] L. P. Zawada, R. S. Hay, S. S. Lee, J. Staehler, Characterization and high-temperature mechanical behavior of an oxide/oxide composite, Journal of the American Ceramic Society 86 (6) (2003) 981–990.

[34] R. Desmorat, A. Kane, M. Seyedi, J. Sermage, Two scale damage model and related numerical issues for thermo-mechanical high cycle fatigue, European Journal of Mechanics - A/Solids, 26 (2007) 909–935.

[35] J. Lemaitre, A. Benallal, D. Marquis, Lifetime prediction of structures in anisothermal viscoplasticity coupled to damage, Nuclear Engineering and Design 133 (1992) 345–360.

AppendixA. Equivalent positive strain tensors

The methodology adapted by Rakotoarisoa [5] from [26] is reported in the next equations, in Voigt notation, for the calculation of $\varepsilon^{d_1^+}$ in the warp direction:

$$\lambda_{d_1^+} = \frac{1}{2}\left(\varepsilon_1 + \sqrt{\varepsilon_1^2 + \varepsilon_5^2 + \varepsilon_6^2}\right) \tag{A.1}$$

$$\Omega_{d_1^+} = (\lambda_{d_1^+})^2 + \frac{1}{4}\left(\varepsilon_5^2 + \varepsilon_6^2\right) \tag{A.2}$$

$$\begin{cases} \varepsilon_1^{d_1^+} = \frac{(\lambda_{d_1^+})^3}{\Omega_{d_1^+}} \\ \varepsilon_5^{d_1^+} = \frac{\varepsilon_5(\lambda_{d_1^+})^2}{\Omega_{d_1^+}} \\ \varepsilon_6^{d_1^+} = \frac{\varepsilon_6(\lambda_{d_1^+})^2}{\Omega_{d_1^+}} \end{cases} \tag{A.3}$$

The other tensor, in the weft direction, is obtained through permutation of indexes.

AppendixB. Static (monotonic) and fatigue contribution for sine/triangular loadings

For sine/triangular loadings, the monotonic part is

$$\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} = \int_{t_{N-1}}^{t_N} \frac{\mathrm{d}\mathcal{M}}{\mathrm{d}t} dt = \int_{\varepsilon_{eq,N-1}^{max}}^{\varepsilon_{eq,N}^{max}} \left\langle \frac{\varepsilon_{eq}^{max} - \varepsilon^{0s}}{S^s} \right\rangle_+^{s^s} \mathrm{d}\varepsilon_{eq}^{max} \tag{B.1}$$
$$= \frac{S^s}{s^s+1} \left[\left\langle \frac{\varepsilon_{eq,N}^{max} - \varepsilon^{0s}}{S^s} \right\rangle_+^{s^s+1} - \left\langle \frac{\varepsilon_{eq,N-1}^{max} - \varepsilon^{0s}}{S^s} \right\rangle_+^{s^s+1} \right]$$

and the cyclic part is

$$\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N} = \int_{t_{N-1}}^{t_N} \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}t} dt = \int_{\varepsilon_{eq,N-1}^{min}}^{\varepsilon_{eq,N-1}^{max}} \left\langle \frac{\varepsilon_{eq} - m\bar{\varepsilon}_{eq} - \varepsilon^{0f}}{S^f} \right\rangle_+^{s^f} \tag{B.2}$$
$$= \frac{S^f}{s^f+1} \left[\left\langle \frac{\varepsilon_{eq,N-1}^{max} - m\bar{\varepsilon}_{eq} - \varepsilon^{0f}}{S^f} \right\rangle_+^{s^f+1} - \left\langle \frac{\varepsilon_{eq,N-1}^{min} - m\bar{\varepsilon}_{eq} - \varepsilon^{0f}}{S^f} \right\rangle_+^{s^f+1} \right]$$

assuming that the evolving mean strain, $\breve{\varepsilon}_{eq}$, has reached standard (here constant) value, $\bar{\varepsilon}_{eq}$.

AppendixC. Damage integration over a cycle-jump

The damage law, written as a cycle-function (Eq.(12) = Eq.(C.1)), comes from the integration of Eq.(6) over a cycle, with the assumption that the damage value, d, is quasi-constant during a cycle, and is given by:

$$\frac{\mathrm{d}d}{\mathrm{d}N} \approx (d_\infty - d) \frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} + (d_\infty - d)^\gamma \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N} \quad \text{with } \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N} > 0,\ \frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} \geq 0 \tag{C.1}$$

By integration over a jump, the damage value after the jump, $d_{N+\overline{\Delta N}}$, is provided by Eq.(13):

$$\int_{d^N}^{d_{N+\overline{\Delta N}}} \frac{\mathrm{d}d}{(d_\infty - d)\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} + (d_\infty - d)^\gamma \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}} = \int_N^{N+\overline{\Delta N}} \mathrm{d}N \tag{C.2}$$

Let us consider two cases:

- $\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} = 0$ (Pure fatigue)
  After training cycles, the mean equivalent strain does not evolve any more and $\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}$ is constant. The hypothesis of static and fatigue damage increment being constant during the jump is satisfied. It follows that:

$$\overline{\Delta N} = -\frac{1}{(1-\gamma)\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}} \int_{d^N}^{d_{N+\overline{\Delta N}}} -\frac{\mathrm{d}d(1-\gamma)}{(d_\infty - d)^\gamma} \tag{C.3}$$

The integral of a power function can be identified, and thus:

$$\overline{\Delta N} = \frac{1}{(1-\gamma)\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}}\left[\left(d_\infty - d^N\right)^{1-\gamma} - \left(d_\infty - d^{N+\overline{\Delta N}}\right)^{1-\gamma}\right] \tag{C.4}$$

$$d^{N+\overline{\Delta N}} = d_\infty - \left[\left(d_\infty - d^N\right)^{1-\gamma} - \overline{\Delta N}(1-\gamma)\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}\right]^{\frac{1}{1-\gamma}} \tag{C.5}$$

- $\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} > 0$

  Eq.(C.2) cannot be simplified. It is assumed that both $\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}$ and $\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}$ are constant. This assumption is nevertheless not satisfied because of the evolution of the mean equivalent strain due to static damage. It follows that:

$$\begin{aligned}\overline{\Delta N} =& \int_{d^N}^{d_{N+\overline{\Delta N}}} \frac{(d_\infty - d)^{-\gamma}\,\mathrm{d}d}{(d_\infty - d)^{1-\gamma}\,\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N} + \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}} \\ =& \frac{1}{(1-\gamma)\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}} \int_{d^N}^{d_{N+\overline{\Delta N}}} \frac{(1-\gamma)\,(d_\infty - d)^{-\gamma}\,\mathrm{d}d}{(d_\infty - d)^{1-\gamma} + \frac{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}}{\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}}}\end{aligned} \tag{C.6}$$

  The derivative of a logarithm function can be identified in the integral; consequently, for material parameter $\gamma \neq 1$:

$$\overline{\Delta N} = -\frac{1}{(1-\gamma)\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}} \ln\left(\frac{(d_\infty - d^{N+\overline{\Delta N}})^{1-\gamma} + \frac{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}}{\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}}}{(d_\infty - d^N)^{1-\gamma} + \frac{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}}{\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}}}\right) \tag{C.7}$$

$$\begin{aligned}d^{N+\overline{\Delta N}} = d_\infty - \Bigg[&\exp\left(-\overline{\Delta N}(1-\gamma)\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}\right) \\ &\left((d_\infty - d^N)^{1-\gamma} + \frac{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}}{\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}}\right) - \frac{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}N}}{\frac{\mathrm{d}\mathcal{M}}{\mathrm{d}N}}\Bigg]^{\frac{1}{1-\gamma}}\end{aligned} \tag{C.8}$$

Highlights

- Method reducing the simulation time of structures subjected to fatigue loadings.
- Method designed for kinetic models, enabling a non-linear extrapolation of variables.
- The accuracy and performance of the cycle-jump procedure is assessed.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: