Master Thesis

Developing a Python-based fatigue tool using the Critical Plane Method

carried out for the purpose of obtaining the degree of Master of Science
submitted at TU Wien, Faculty of Mechanical and Industrial Engineering
by

Miha Gulic, BSc
Mat.Nr.: 01635070
Maribor, Slovenia

under the supervision of

Assoc. Prof. Dipl.-Ing. Dr.techn. Heinz Pettermann
Institute of Lightweight Design and Structural Biomechanics
TU Wien

Vienna, 13/01/2023

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume. If text passages from sources are used literally, they are marked as such.

I confirm that this work is original and has not been submitted elsewhere for any examination, nor is it currently under consideration for a thesis elsewhere.

I acknowledge that the submitted work will be checked electronically-technically using suitable and state-of-the-art means (plagiarism detection software). On the one hand, this ensures that the submitted work was prepared according to the high-quality standards within the applicable rules to ensure good scientific practice "Code of Conduct" at the TU Wien. On the other hand, a comparison with other student theses avoids violations of my personal copyright.

Vienna, 09/15/2022 Signature

Acknowledgements

I would like to express my sincerest gratitude to all those who have supported me throughout the course of my master thesis. I would like to acknowledge the support and guidance of my mentor Assoc. Prof. Dipl.-Ing. Dr.techn. Heinz Pettermann, who has been instrumental in the development of my thesis.

I am also grateful to my classmates, colleagues, and friends who have shared their ideas and experiences with me and made this journey a truly enriching one.

I would also like to acknowledge the support of my family, who have always believed in me and encouraged me to pursue my passions. Their encouragement and support have been my greatest source of strength throughout this process.

Finally, I would like to thank the authors and researchers whose works have served as the foundation for this thesis. Their contributions have been invaluable, and I am deeply appreciative of their efforts.

Thank you all for your support, encouragement, and guidance.

Contents

Abstract

The present thesis presents a problem of fatigue failure in suspension springs of reciprocating compressors. Due to random displacements that equate to a non-constant, non-proportional loading profile, an advanced fatigue evaluation method is necessary. Based on available literature the critcial plane method (CPM) is chosen for this purpose, being especially well-suited for a software implementation. In conjunction with a finite element method (FEM) stress analysis it forms a good basis for local stress based fatigue evaluation methods.

An object-oriented fatigue tool is conceptualized and subsequently implemented in Python. It is systematically tested and validated on cases of increasing complexity. Its class- and object-based structure enables future extensions of functionality.

Finally, real-life use cases of a compressor crankshaft and compressor spring are presented to showcase the use of the tool for a simplified and a full fatigue evaluation process.

Keywords: fatigue, critcial plane method (CPM), suspension spring

Kurzfassung

Diese Arbeit stellt ein Problem des Dauerfestigkeitsversagens von Federn in Kolbenkompressoren vor. Aufgrund von zufälligen Bewegungen des Kompressors, die einem nicht-konstanten, nicht-proportionalem Lastprofil entsprechen, werden fortgeschrittene Methoden der Dauerfestigkeitsanalyse benötigt. Basierend auf verfügbarer Literatur wird das critcial plane method (CPM) ausgewählt, die sich gut für eine Softwareimplementirung eignet. In Kombination mit einer finite element method (FEM) Spannungsanalyse stellt dies eine gute Basis für eine Dauerfestigkeitsanalyse dar, die auf der lokalen Spannung basiert. Ein objektorientiertes Tool für Dauerfestigkeitsanalyse ist konzeptualisiert und in Python implementiert. Das systematische Testen und Validierung erfolgt mittels Test-Beispielen steigender Komplexität. Ihre klassen- und objektbasierte Struktur ermöglicht zukünftige Erweiterungen der Funktionalität. Schlussendlich werden zwei realitätsnahe Anwendungsbeispiele, eine Kompressor-Kurbelwelle und eine Feder, vorgerechnet, um sowohl eine vereinfachte, als auch eine vollständige Dauerfestigkeitsanalyse zu demonstrieren.

Stichwörter: Dauerfestigkeit, critcial plane method (CPM), Feder

List of Figures

List of Tables

List of Abbreviations

CPM critcial plane method

FE finite element

FEM finite element method

GUI graphical user interface

HCF high cycle fatigue

LCF low cycle fatigue

SWT Smith-Watson-Topper

1 Introduction

Since the industrial revolution over 200 years ago compressors have been an integral part of the industrial process, their main purpose being the transport of air or other gases. Unlike fans or blowers they operate in higher differential pressure ranges. From the thermodynamic aspect compressors convert mechanical energy into gas energy and there are two different approaches to this [1].

1. By positive displacement of the gas into a smaller volume. Flow is directly proportional to the speed of the compressor, but the pressure differential is determined by the system characteristics.

2. By dynamic action imparting velocity to the gas. The velocity is then converted into pressure via diffusors.

The present thesis focuses on positive displacement compressors of the reciprocating type (see Figure 1), where forces acting on the system are much more dynamic as the compression process itself is not continuous. The forces and moments resulting from moving rigid bodies are only partially balanced through the use of counterweights, while the remaining forces are transferred to the foundation through the mounts. Generally, helical suspension springs in conjunction with steel or rubber holders are used to facilitate structural stiffness and minimise the transfer of vibrations to the base, with a balance of the two being set by the spring stiffness [4]. During regular operation the main load contributor for the springs is static compression, while small lateral deflections due to unbalanced forces play only a minor role. However, in the instances of starting and

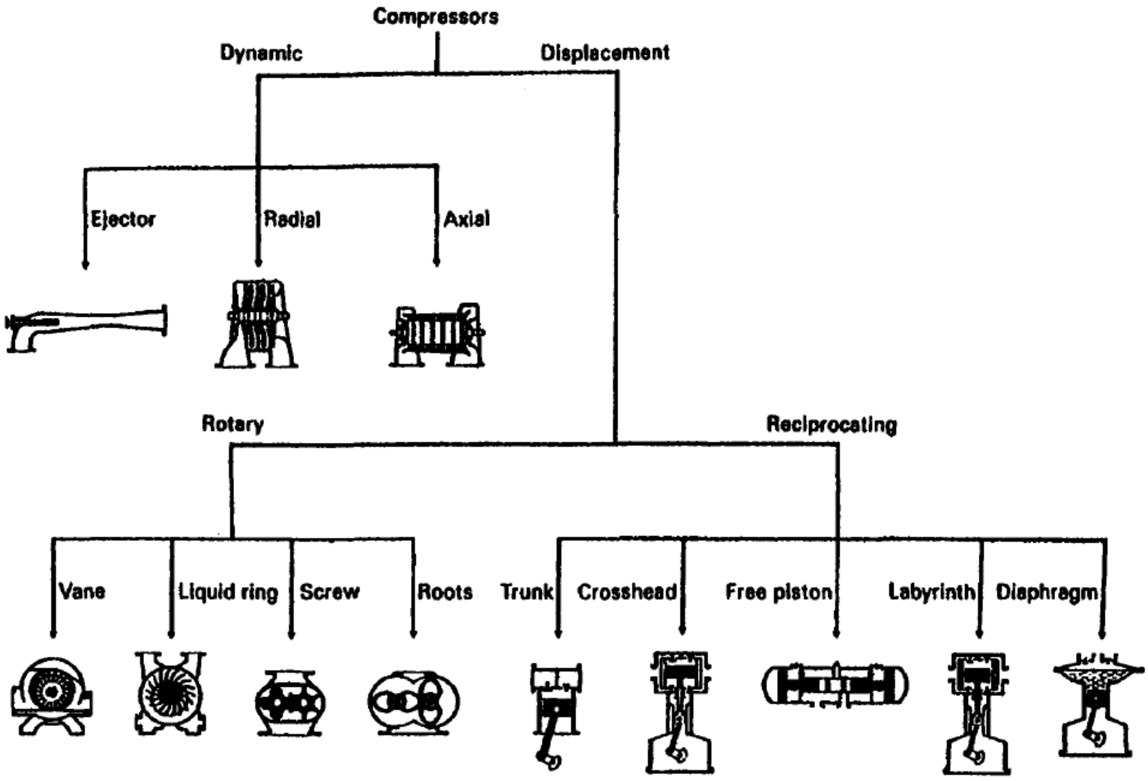


Figure 1: Principal compressor types [1]

stopping or the transport of the whole compressor, the lateral deflections are no longer negligible and the springs must counteract them, thereby preventing the pump unit from hitting the shell. The spring holders affect the deformation of the springs under high lateral deflections and their damaging effects are discussed in [4].

Compressors undergo accelerated reliability testing to assure that reliability goals are met. This is especially critical for hermetically sealed compressors, which are expected to reach the end of their operating life without maintenance. The reliability of suspension springs in terms of fatigue is tested by repeating a large number of starting and stopping cycles as described in more detail in [5]. With the number of starting and stopping cycles ranging from 300,000 to 500,000 the testing is prolonged and expensive, requiring a good analytical or numerical approach for estimating fatigue life prior to experimental testing. An analytical approach is introduced and investigated in [5], showing promising results at the time of writing, but has now been surpassed by the steady development of advanced numerical methods. Today simulations are a vital part of investigating and estimating fatigue life of components, having the potential to decrease the necessity of experimental tests.

The present thesis aims to develop a Python-based fatigue tool, utilising the CPM, for fatigue evaluation of compressor components. The emphasis lies on suspension spring, but should be universally applicable to any component under cyclic loading, such as the crankshaft or conrod.

2 Theoretical background

The following chapter gives a thorough introduction into the theoretical background of the methods and theories applied in this thesis. The chapter is divided into four sections. Section 2.1 covers the basics of continuum mechanics necessary for understanding the topic. In Section 2.2 an introduction into the fatigue of materials is followed by a more detailed description of the methods required for handling complex multi-axial load histories. In Section 2.3 the CPM is introduced as a concept, as well as an algorithm well-suited to object-oriented programming languages.

2.1 Basics of continuum mechanics

Stress and strain tensors, traction vectors and tensor/vector operations are given next.

2.1.1 Vector

A vector is a rank one tensor most commonly used to describe positions or movements within a two- or three-dimensional space, such as a cartesian coordinate system. Vector rotation is often needed to transform between different coordinate systems of the same type.

Rotation A vector is rotated by constructing a rotation matrix $\boldsymbol{R}$ and performing a matrix multiplication with the original vector.

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x} \tag{1}$$

Rodrigues' rotation matrix An efficient way of constructing the rotation matrix in thee-dimensional space when given an axis and angle of rotation is using the Rodrigues' formula. We let $\boldsymbol{K}$ denote the cross-product matrix

$$\boldsymbol{K} = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_z & 0 \end{pmatrix}, \tag{2}$$

where $\boldsymbol{a} = (a_x, a_y, a_z)$ is the unit vector defining the axis of rotation. The rotation matrix $\boldsymbol{R}$ is constructed as

$$\boldsymbol{R} = \boldsymbol{I} + (\sin\theta)\boldsymbol{K} + (1 - \cos\theta)\boldsymbol{K}^2, \tag{3}$$

where $\boldsymbol{I}$ is the identity matrix and θ is the angle of rotation.

2.1.2 Stress tensor

The stress tensor is a rank two tensor that describes the stress state at a specific location and is always given with respect to a coordinate system. A generic stress tensor has the following form,

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{\mathrm{xx}} & \tau_{\mathrm{xy}} & \tau_{\mathrm{zx}} \\ \tau_{\mathrm{xy}} & \sigma_{\mathrm{yy}} & \tau_{\mathrm{yz}} \\ \tau_{\mathrm{zx}} & \tau_{\mathrm{yz}} & \sigma_{\mathrm{zz}} \end{pmatrix} \quad , \tag{4}$$

and is symmetric.

2.1.3 Traction vector

A traction vector is simply the internal force vector on a cross-section divided by that cross-section's area. It represent the internal equilibrium in the body and is dependant on the choice of the cross-section. In the most general case it can be computed from the stress tensor as

$$\vec{T} = \boldsymbol{\sigma} \cdot \vec{n} \tag{5}$$

$$= \begin{pmatrix} \sigma_{\mathrm{xx}} & \tau_{\mathrm{xy}} & \tau_{\mathrm{zx}} \\ \tau_{\mathrm{xy}} & \sigma_{\mathrm{yy}} & \tau_{\mathrm{yz}} \\ \tau_{\mathrm{zx}} & \tau_{\mathrm{yz}} & \sigma_{\mathrm{zz}} \end{pmatrix} \cdot \begin{pmatrix} \mathrm{n_x} \\ \mathrm{n_y} \\ \mathrm{n_z} \end{pmatrix} = \begin{pmatrix} \sigma_{\mathrm{N}} \\ \tau_{\mathrm{y}} \\ \tau_{\mathrm{z}} \end{pmatrix} \quad , \tag{6}$$

where $\vec{n}$ is the unit normal of the selected cross section. σ_{N} the normal component of the traction vector, whereas the shear components are represented by τ_{y} and τ_{z}.

2.2 Fatigue of materials

Design of components to withstand static loads has been well understood for a long time, however the damaging effects of low-level cyclic loads were relatively unknown until the second half of 19th century. The field was pioneered by August Wöhler, following an investigation of an 1842 train crash due to fatigue failure of an axle. It was not until after the introduction of stress-life curves, also known as Wöhler-curves, that cyclic stress was better understood and fatigue could be predicted in a more consistent manner.

Subsequent investigations into the topic revealed two distinct regimes of fatigue damage, i.e. low cycle fatigue (LCF) and high cycle fatigue (HCF). In LCF macroscopic plastic deformations are present and the usual design life is in the order of 10^4 cycles. In HCF there are no macroscopic plastic deformations present and the design life up to 10^9 cycles is usual. This thesis aims to develop a tool for long-term reliability in the HCF regime, therefore any fatigue mentioned from this point onward is assumed to be HCF.

2.2.1 Fatigue-life methods

The three major fatigue-life methods used in design, the stress-life method, the strain-life method and the linear-elastic fracture mechanics method attempt to predict the life in number of cycles to failure (N). In this section all three methods are briefly introduced, however, only the stress-based approach is discussed in more detail.

The stress-life method, based only on stress levels, is considered to be the least accurate approach, especially in the LCF regime. It is, however, the traditional approach with ample supporting data and the advantage of a straightforward implementation for a wide array of applications and a high degree of validity for the HCF regime [6].

The strain-life method is considered to be the best approach yet to explain the nature of fatigue failure, involving a more detailed analysis of the local plastic deformations. In areas of stress concentrations the elastic limit is exceeded and plastic strain occurs. If an LCF fatigue failure is to occur there must exist cyclic plastic strains. Depending on the material the elastic limit may change when subjected to cyclic loading. The approach is well-suited for the LCF regime and is becoming increasingly popular with the rapid improvements in the use of finite element analyses. At the moment, however, little supporting data is available in literature. A more detailed explanation of the strain-life approach is given in [6].

The linear-elastic fracture mechanics method assumes that a crack is already present and studies crack growth, rather that its initiation. It attempts to predict crack growth with respect to stress intensity and is most well-suited to large structure with periodic inspection programs [6].

Stress-life method

It is common engineering practice to evaluate fatigue based on stress values, whether it be a uniaxial or a multiaxial stress state. Stress-based fatigue includes two distinct approaches: the nominal stress approach and the local stress approach.

The nominal stress approach is historically the most relevant and forms the base of fatigue evaluation in many areas. The aim is to represent the periodic loading as a time-varying profile of the nominal stress, which can then be described as a load collective, a rainflow matrix or a series of maximum and minimum values [7]. The advantage of this approach is the simplicity of determining the nominal stress based on a nominal cross-section. The choice of the nominal cross-section is in most cases made based on conventions and is to be taken from literature for most standard situations. Acknowledging the fact that actual stress often deviates from the nominal stress due to part geometry, correction factors are introduced to account for that. Local stress concentrations due to notches can also lead to multiaxial stress states, necessitating the use of equivalent stress theories [7]. This approach is generally suitable for long and flat structures.

The local stress approach has gained in relevance since the introduction of numerical methods into everyday engineering practice and the trend towards lightweight structures. As opposed to the nominal stress concept that requires a clearly defined nominal cross-section, the local stress concept is based on local stresses in the structure. This makes the approach more suitable for bulkier components with a substantial influence of geometry [7]. At the cost of higher computation cost it is the more versatile of the two approaches. The underlying numerical analysis is subject to a separate set of guidelines (briefly introduced in [7]) that aim to ensure realistic results.

The present thesis focuses on the implementation of the CPM, which inherently follows the local stress approach.

2.2.2 Loading characterisation

Unlike static stress, which is analysed for a single stress state, fatigue damage occurs when the stress state at a certain point changes with time. This calls for a general classification of the loading profile, so as to enable an informed choice of the applied fatigue calculation method [8]. A load cycle is sufficiently described by the minimum stress (σ_{min}) and the maximum stress (σ_{max}), but is typically described via the mean stress (σ_m) and the alternating stress amplitude (σ_a). The load ratio is given by

$$R = \frac{\sigma_{min}}{\sigma_{max}} \quad . \tag{7}$$

Generally, the loading can be classified as one of the following four classes:

- Constant amplitude, proportional loading is the most trivial case. The load has a constant maximum value and only one set of FEM results, along with the loading ratio, is needed to determine the alternating stress amplitude and the mean value. The loading is proportional as the principal axes do not change over time. The complete loading history can be described by only two parameters, the mean stress(σ_m) and alternating stress amplitude (σ_a), eliminating the need for cycle counting or damage accumulation theories [8].

- Constant amplitude, non-proportional loading retains the constant amplitude, but the principal axes are free to change between two or more load cases. It is characterized by two or more principal stress axes fluctuating in time. Due to the constant amplitude cycle counting may be omitted. Since the loading is non-proportional the critical location in terms of fatigue may not be clearly identifiable when analyzing the load cases separately. This type of fatigue loading can describe an alternation between two or more distinct load cases, superimposed static and cyclic loads or cases where the loading is proportional, but the results are not. This may be the case in analyses with material and/or geometric non-linearity, as well as contact problems [8].

- Non-constant amplitude, proportional loading requires only one load case that is scaled. The transient characteristic is given by a time-varying load ratio. Since the loading is proportional the critical location in terms of fatigue is easily identifiable when analyzing the base result set, however it is not clear, which loading produces the most fatigue damage. This necessitates the use of cycle counting techniques and subsequently also damage accumulation theories.

- Non-constant amplitude, non-proportional loading is the most general loading case, that considers two or more load cases that have no relation to one another. It is characterized by two or more principal stress axes fluctuating in time. Not only is the critical location in terms of fatigue not identifiable by looking at the base results separately, but also the combination of loads that cause the most fatigue damage is unknown [8]. More advanced critical plane methods or multi-axial cycle counting techniques are required to handle this type of loading [2]

The present thesis aims to develop a fatigue tool that is to be as universally applicable as possible. Consequently, the handling of non-constant amplitude, non-proportional loading cases is a requirement.

2.2.3 Stress-life curves

Stress-life curves (S-N curves), also known as Wöhler curves in german-speaking areas, relate stress to life of a material or a component. The fatigue life of a component refers to the fatigue initiation life defined as the number of cycles (N) or load reversals ($2N$) to a specific crack initiation length under cyclic stress controlled tests, whereas one cycle consists of two reversals [2].

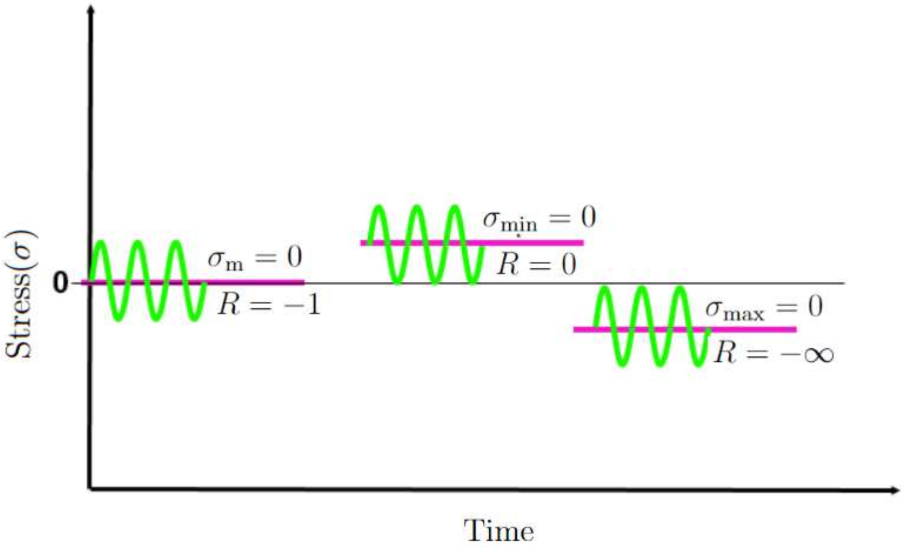

Stress(σ)
0
$\sigma_{\mathrm{m}} = 0$
$R = -1$
$\sigma_{\mathrm{min}} = 0$
$R = 0$
$\sigma_{\mathrm{max}} = 0$
$R = -\infty$
Time

Figure 2: Load cycles with different maximum and minimum stress and the corresponding load ratio

Load cycle The stress profile within a cycle can be described in two ways, either by stress amplitude σ_a and mean stress (σ_m) or by maximum (σ_{max}) and minimum stress (σ_{min}). Since σ_a, being derived from the maximum and minimum stress, is the primary factor affecting the fatigue life (N), it is often chosen as the controlled parameter in fatigue testing. Therefore, S-N curves are generally given as an $N(\sigma_a)$ relation [2].

Generating S-N curves S-N curves are generated by fatigue testing material specimens or real components at various load/stress levels and depending on the test subject there can be either material or component S-N curves. For this type of fatigue testing, the mean stress is held at a constant and is commonly equal to zero [2].

The constant amplitude S-N curve is usually plotted in the log-log space, where it appears as a linear curve. Generally, it is constructed as a piecewise-straight curve, consisting of two distinct linear regimes. A typical S-N curve for a material sample made of steel, as schematically illustrated in Figure 3, consists of a constant downward-inclined slope for the HCF regime and one horizontal asymptote for the fatigue limit [2]. Additional influence factors may be applied based on the heat treatment, surface treatment and geometry. The S-N curve is appropriately modified, leading to either lower or higher fatigue limits.

Fatigue limit The fatigue limit can be defined as the fully reversed stress amplitude σ_a at which the fatigue initiation life becomes infinite or when fatigue initiation does not

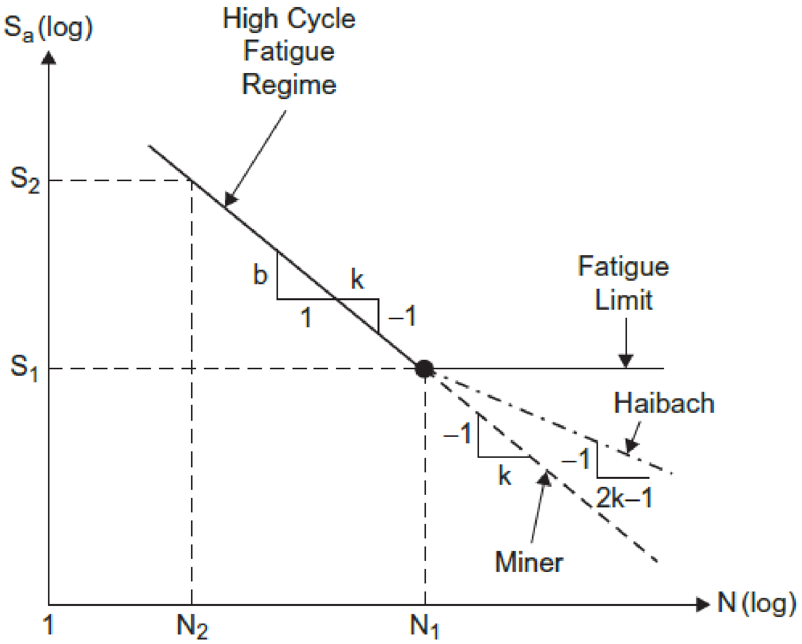


Figure 3: Schematic constant amplitude S-N curve for a component made of steels [2]

occur. From a physical perspective it can be seen as a microcrack nucleating from within a grain of material and growing to the size of the order of the grain width. Upon reaching the grain boundary, the boundary will either inhibit the crack's growth or if the crack growth driving force is too high, the crack will propagate over the boundary and may lead to failure of the component. The minimum stress amplitude required for the microcrack to overcome the grain boundary and propagate further is referred to as the fatigue limit [2].

Endurance limit Not all materials exhibit a fatigue limit under which fatigue failure does not occur. This is the case for aluminium alloys and austenitic steels. As the fatigue limit does not exist, the fatigue testing must be terminated at a specified high number of cycles, most often this is set at 10^6 cycles. The stress amplitude, which does not lead to fatigue failure after 10^6 is referred to as the endurance limit. It is common to use the endurance limit to characterize all materials in terms of fatigue, also ones that do exhibit a fatigue limit [2].

2.2.4 Mean stress effects

The total fatigue damage of a material specimen or a component is primarily determined by the stress amplitude σ_a and is secondarily influenced by the mean stress σ_m. Especially in the HCF regime the mean stress has a significant effect and should not be ignored. From a physical aspect, mean normal stresses are responsible for the opening and closing of microcracks, thereby either accelerating or inhibiting the rate of crack propagation, respectively. The LCF regime is predominantly characterized by the large local plastic deformations, where the mean stress has little or no effect on total fatigue damage [2]. The mean normal stress effect can be represented by the mean normal stress

$$\sigma_m = \frac{\sigma_{max} - \sigma_{min}}{2} \quad , \tag{8}$$

or by the stress ratio given in Eq. (7).

The mean stress effects can be considered by either using a suitable S-N curve or by applying a mean stress correction. The wide-scale availability of standard S-N curves at zero mean stress makes the use of mean stress correction formulas the more rational approach. The correction is used to transform a stress cycle with an arbitrary mean stress to an equivalent stress cycle with zero mean stress, enabling the use of standard S-N curves. Of the numerous models developed only the ones implemented in this thesis are presented here: Goodman [9] and Smith-Watson-Topper (SWT) [10].

For the fully reversed stress amplitude in the case of moderate mean stress values the mean stress corrections are described by the following equations:

- Goodman

$$\sigma_a' = \frac{\sigma_a}{1 - \dfrac{\sigma_m}{\mathrm{R_m}}} \tag{9}$$

$$\mathrm{R_m} \text{...material ultimate strength} \tag{10}$$

- SWT

$$\sigma_a' = \sqrt{\sigma_{max}\sigma_a} = \sqrt{(\sigma_a + \sigma_m)\,\sigma_a} \tag{11}$$

An extensive study on the validity and accuracy of the above models and more has been conducted in [11] for R ratio ranges from -2 to 0.45. It was concluded that the Goodman's model is highly inaccurate and should not be used, whereas the SWT model offers good results for steels, aluminum alloys and titanium alloys, making it a reasonable choice for general use. Note that due to the nature of Equation (11) the SWT model is not defined for maximum normal stress less than or equal to zero, meaning $\sigma_{max} \leq 0$ [2]. Essentially, it predicts that a fatigue crack will not initiate if the maximum normal stress in a cycle is a compressing one.

2.2.5 Rainflow cycle counting

The damage to a material specimen or a component can be easily calculated for a cyclic loading with a constant amplitude using the S-N curve. However, in a real-life, variable loading scenario the number of cycles and their respective amplitude is not easily determined, which necessitates the use of cycle counting methods. Rainflow cycle counting is a method to determine the number of fatigue cycles present in a load-time history, where each cycle is associated with a closed stress-strain hysteresis loop [2]. Essentially, cycle counting reduces the stress time history to the minimal amount of data required to preserve the damage information.

Uniaxial load-time history For a uniaxial load-time history the cycle counting is comparatively simple and consists of directly applying the counting algorithm to the existing load or stress profile at a certain location in the material [2]. Since the introduction of the first widely accepted rainflow counting technique by [12] many more efficient algorithms have been introduced, such as the three-point and the four-point cycle counting techniques.

Multiaxial load-time history In the case of a multiaxial load time history there are two generally accepted approaches to reducing the stress tensor to scalar quantities. In the first approach, the CPM, the material volume is segregated into candidate planes, commonly referred to as the cutting planes. On each candidate plane the equivalent stress-time history is subjected to cycle counting and the accumulated fatigue damage is calculated. The plane that accumulates the most damage is deemed to be the critical plane [2]. In the second approach, the equivalent stress or strain amplitude method, the complicated equivalent loading history is subjected to a multiaxial cycle counting technique. The underlying assumption is that the fatigue damage can be reliably evaluated based on the complicated equivalent loading history [2]. The advantages and shortcomings of both approaches are discussed in a thorough manner in [13].

A concise description and comparison of different cycle counting techniques is given in [2].

2.2.6 Linear damage theory

For a component subjected to variable amplitude loading over time (see Section 2.2.2), a rainflow cycle counting technique as presented in Section 2.2.5 is typically used to convert the complicated time-varying stress history into a series of constant amplitude stress events, described by their mean stress, stress amplitude and number of cycles (n_i) [2]. The fatigue life (N_i) corresponding to the number of cycles to failure at a specific stress event (i) can be estimated from the component S-N curve. The fatigue damage is calculated accordingly as the cycle ratio

$$\mathrm{D}_i = \frac{\mathrm{n}_i}{\mathrm{N}_i}. \tag{12}$$

A linear damage theory must be adopted to accumulate the damage. Typically the Palmgren-Miner linear damage rule is used, which assumes that fatigue failure occurs when the sum of the cycle ratios at each constant amplitude stress event reaches a critical damage value ($\mathrm{D}_{\mathrm{crit}}$) [2]. Failure is predicted when

$$\sum \mathrm{D}_i = \sum \frac{\mathrm{n}_i}{\mathrm{N}_i} \geq \mathrm{D}_{\mathrm{crit}}, \tag{13}$$

although the the critical damage value differs depending on material and source [7].

The S-N curve is supposed to be used for estimating fatigue life for constant stress events and fails to consider the effects of periodic overloads that may lower the original fatigue limit of the component. Therefore, subsequent loads below the original fatigue limit may cause damage [2]. This critique resulted in a multitude of approaches being developed, aimed at considering the damage of stress events below the original fatigue limit. These are described in a thorough manner in [7], summarized in [2] and briefly presented here. The terminology is adopted from [7] and used consistently as it differs in other sources. The attempt to consider the negative effects of periodic overloads on the fatigue limit resulted in three generally accepted rules.

The elementary Miner rule does not consider the negative effect of periodic overloads on the original fatigue limit and prescribes no damage to constant stress events below the fatigue limit visualised in Fig. 3.

The original Miner rule considers the constant stress events below the fatigue limit to be equally, but proportionally less damaging as the events above it. Graphically this can be seen as an extension of the downward-inclined section of the S-N curve by a straight line with the same slope as seen in Fig. 3 [2]. This approach has been shown to be conservative and not in good agreement with experimental results [7].

The modified Miner rule, also referred to as the Miner-Haibach rule in [2], considers the constant stress events below the fatigue limit to be damaging, but less so than the stress events above it. This can be seen as an extension of the downward-inclined section of the S-N curve by a straight line with a flatter slope as seen in Fig. 3. This approach has been shown to be in good agreement with experimental results [7].

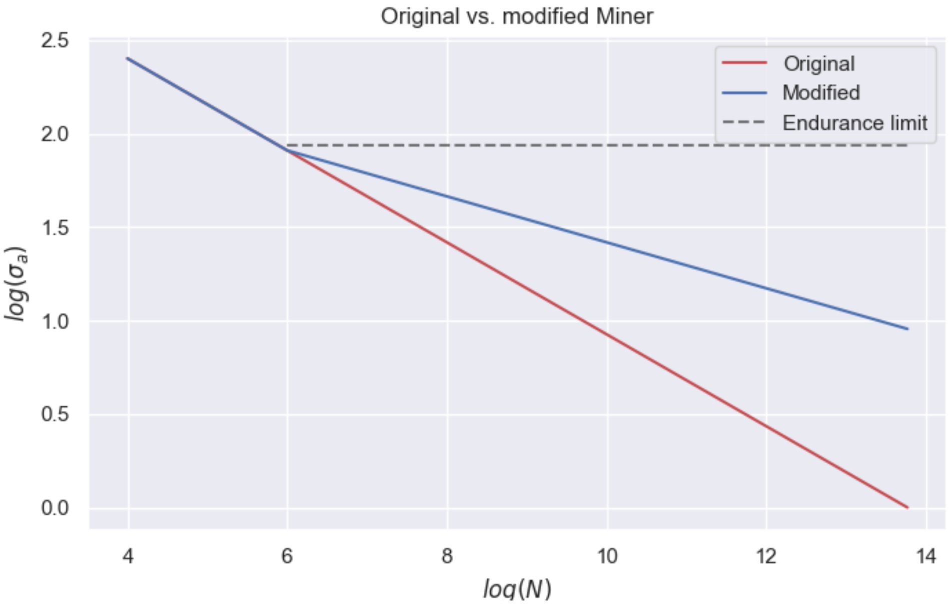


Figure 4: The original and modified Miner rules applied to a showcase stress-life curve

Figure 4 depicts the original and modified Miner rules applied to the stress-life curve of a steel material. In both cases the curve is extended below the endurance limit, however, the slope is what differentiates the two linear damage rules.

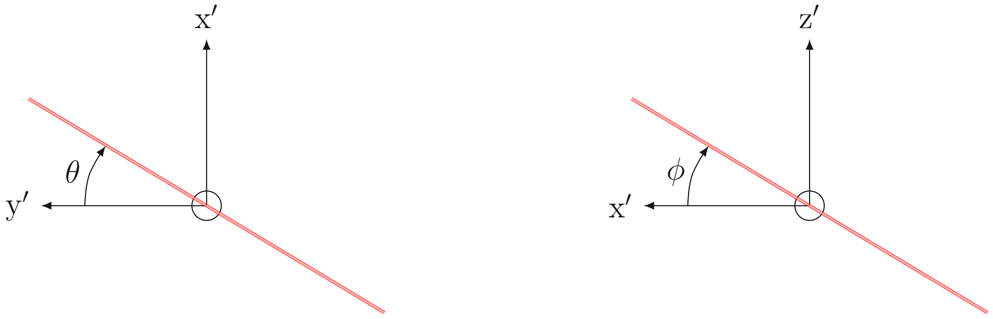
2.3 Critical Plane Method

The CPM is a general term referring to the analysis of stresses or strains experienced by a particular plane in a material, as well the identification of the plane, which is likely to accumulate the most fatigue damage. Due to its universality, flexibility and accuracy it is widely used in engineering [2]. The approach was first proposed by W. N. Findley in 1959 in [14], based on the physical observations that cracks initiate and grow on preferred planes [15]. Modern CPM procedures can be traced back to M. W. Brown and K. J. Miller and their publication [16]. Since then, a number of researchers have proposed modifications or extensions of the method.

When structures are subjected to cyclic multiaxial loads it is necessary to use multiaxial fatigue criteria. When, additionally to being multiaxial, the loading is also nonproportional (see Sec. 2.2.2), it is necessary to use the proper multiaxial fatigue criterion. The main drawback of these criteria is that they are generally nonconservative for nonproprtional loadings [15]. It is known that nonproportional loadings result in rotating the maximum principal and shear planes at a crack initiation location, leading to a change in potential crack orientation. A final crack will eventually initiate on the plane where the fatigue damage parameter representing the crack nucleation and growth is maximized. Therefore, CPM can be summarized as a searching technique with the aim to identify the location and orientation of the highest damage plane [2].

2.3.1 Candidate planes

A subdivision of the material into candidate planes forms the basis of any CPM procedure. The spatial discretization is typically adopted from the underlying FEM analysis. For the most general case of a triaxial stress state all possible plane orientation in one location are to be considered [3].



(a) Rotation about z′

(b) Rotation about y′

Figure 5: The definition of candidate planes in one location

By varying the angles ϕ and θ (see Fig. 5) all planes can be identified [3]. Note that the present thesis follows the convention of identifying z′ as the normal of the element face.

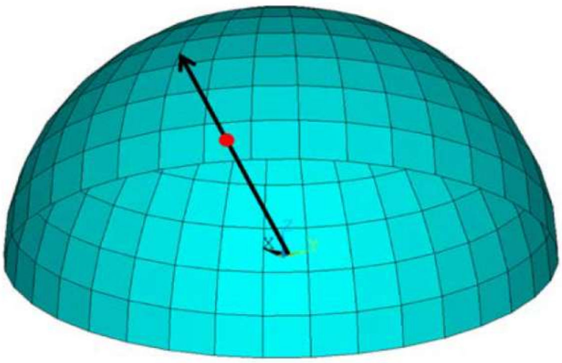


Figure 6: All planes at a point can be represented by a surface of a sphere [3]

Figure 6 shows all planes at a point. The total number of planes is determined by the angle increment chosen. A vector pointing from the point, chosen as the sphere center, to the sphere surface is essentially the normal vector of one candidate plane [3].

Plane stress state simplification

Fatigue cracks typically initiate at a surface of a component, therefore a biaxial stress state simplification of the CPM is often employed in practice [2]. By setting either $\phi = 90°$ or $\theta = 0°$ it is possible to reduce the set of combinations of ϕ and θ by several times, leading to a decrease in computational effort.

Setting $\phi = 90°$ reduces the set of all candidate planes to only planes that lie perpendicular to the component surface. Typically these planes accommodate the initiation and growth of surface cracks, which tend to be shallow and have a small aspect ratio [2].

2.3.2 Stress components

At each location, either at a node or at the element face, a local coordinate system is defined.

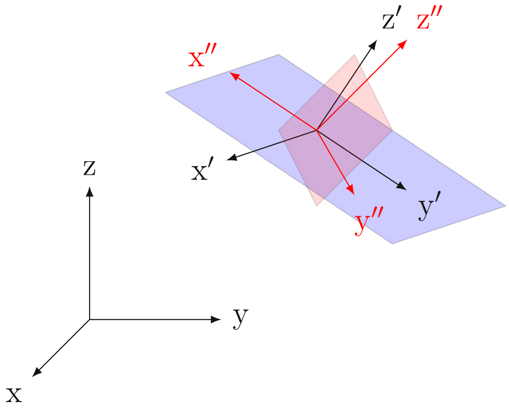


Figure 7: The global coordinate system, the element local coordinate system placed at the center of an element face (blue) and the candidate plane (red) local coordinate system

The candidate planes are defined through their respective cartesian coordinate system within the element local coordinate system. The approach presented in [2] transforms the stress tensor into the local coordinate system. The present thesis, however, rather defines the local coordinate system and all the subsequent candidate plane normals within the global coordinate system to avoid stress tensor transformation. This procedure is based on the projection of the original stress tensor ($\boldsymbol{\sigma}$) onto the candidate plane to get the traction vector ($\vec{T}$). The candidate plane is fully defined by its normal vector ($\vec{\boldsymbol{n}}_{\mathrm{CP}}$) and the present thesis follows the convention of equating $\vec{\boldsymbol{n}}_{\mathrm{CP}} = \vec{\boldsymbol{x}}''$.

The components of the traction vector are computed as

$$\sigma_{\mathrm{N}} = \boldsymbol{T} \cdot \vec{\mathbf{e}}_{\mathrm{x}''} \quad , \tag{14}$$

$$\tau_{\mathrm{y}''} = \boldsymbol{T} \cdot \vec{\mathbf{e}}_{\mathrm{y}''} \quad , \tag{15}$$

$$\text{and} \quad \tau_{\mathrm{z}''} = \boldsymbol{T} \cdot \vec{\mathbf{e}}_{\mathrm{z}''} . \tag{16}$$

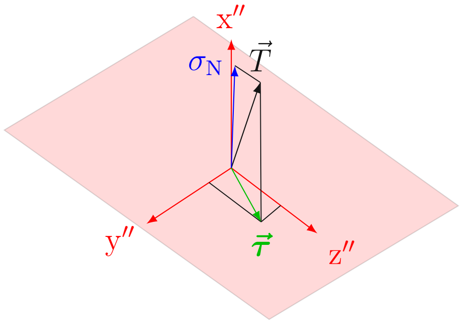


Figure 8: The traction vector

To compute the normal stress (σ_{N}) and the two shear stresses ($\tau_{\mathrm{z}''}, \tau_{\mathrm{y}''}$) in the plane a definition of not only the candidate plane normal ($\vec{\boldsymbol{n}}_{\mathrm{CP}}$) is needed, but also of the entire candidate plane coordinate system ($\mathrm{x}'', \mathrm{y}'', \mathrm{z}''$).

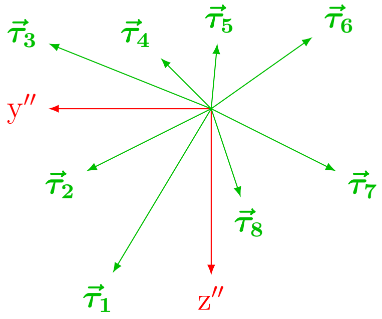


Figure 9: The shear vector at different points in time (adapted from [3])

In the most general loading case the normal component will change its magnitude and sign, but not the orientation - it always lies on the x''-axis. The shear stress vector $\vec{\tau}$, however, remains a fully vectorial quantity with changing magnitude and direction as seen in Fig. 9 [3]. Under proportional loading the direction does not change with time and the definition of stress amplitudes is straightforward. In the case of nonproportional loading the direction changes with time, leading to more complex formulation of the fatigue damage criteria.

2.3.3 Fatigue damage criteria

The fatigue damage criterion, as originally proposed by [14], can be expressed as a function of shear stress and normal stress on a plane. The dominant stress component used in a fatigue damage criterion depends on the fatigue damage mechanism. The fatigue initiation life of ductile materials is typically dominated by crack initiation and growth on maximum shear stress planes, therefore shear stress and normal stress are primary and secondary damage parameters, respectively. Contrary to ductile material, the crack growth of brittle materials is typically controlled by crack growth along maximum tensile planes, therefore normal stress and shear stress are primary and secondary damage parameters, respectively [2].

The vectorial nature of shear stress acting in each candidate plane makes determining the shear stress amplitude a nontrivial problem. Several approaches developed are summarized in [17], with perhaps the most straightforward method presented by [3]. Due to their complexity these topics are deemed to be beyond the scope of this project. The problem is largely disregarded in commercial implementation of CPM, such as presented by [18]:

Normal stress in critical plane is typically used for very brittle materials [18]. There is no need to consider the change in direction. The fatigue damage criterion is given by:

$$\sigma_{\mathrm{ev}} = \sigma_{\mathrm{N}} \tag{17}$$

Total shear stress in critical plane is typically used for very ductile materials [18]. The fatigue damage criterion is given by:

$$\sigma_{\mathrm{ev}} = \mathrm{sgn}(\tau_{\mathrm{x}}) \cdot \left(\frac{\sigma_{\mathrm{w,zd}}}{\tau_{\mathrm{w,t}}}\right) \cdot \sqrt{\tau_{\mathrm{x}}^2 + \tau_{\mathrm{y}}^2} \tag{18}$$

The change in shear stress vector direction is partially considered by the change in sign of one shear stress component, which is also needed for the subsequent cycle counting. However, in the case of nonproportional loading this can result in non-physical discontinuities in the stress-time profile [18].

Equivalent stress in critical plane is typically used for materials that are neither very brittle nor very ductile [18]. The fatigue damage criterion is given by

$$\sigma_{\mathrm{ev}} = \mathrm{sgn}(\sigma_{\mathrm{N}}) \cdot \sqrt{\sigma_{\mathrm{N}}^2 + \left(\frac{\sigma_{\mathrm{w,zd}}}{\tau_{\mathrm{w,t}}}\right)^2 \cdot \left(\tau_{\mathrm{x}}^2 + \tau_{\mathrm{y}}^2\right)} \tag{19}$$

where $\sigma_{\mathrm{w,zd}}$ and $\tau_{\mathrm{w,t}}$ is the material fatigue strength for a load ratio $R = -1$ for tension and torsion, respectively. The sign is required for the subsequent cycle counting and is adopted from the normal stress. However, when dealing with large pre-stressing, this can lead to non-physical discontinuities and subsequently to unrealistically large stress amplitudes [18].

Scaled normal stress in critical plane , also referred to as the Gaier and Dannbauer criterion, shows good agreement with experimental data for a wide range of materials [2] and solves the problem of signs for some other criterions [18]. The procedure can be summarized as [2]:

- Calculation of principal normal stresses $\sigma_1 > \sigma_2 > \sigma_3$ from the stress tensor at every point in time.
- Calculation of the ratio of minimum to maximum principal normal stress at all times:

$$V = \frac{\sigma_3}{\sigma_1} \qquad \text{for} \quad |\sigma_1| > |\sigma_3| \tag{20}$$

$$V = \frac{\sigma_1}{\sigma_3} \qquad \text{for} \quad |\sigma_3| > |\sigma_1| \tag{21}$$

V is a value between -1 and $+1$ and has the following physical interpretation:

$$V = -1 : \qquad \text{dominant shear load} \tag{22}$$

$$V = 0 : \qquad \text{dominant tensile/compressive load} \tag{23}$$

$$V = +1 : \qquad \text{hydrostatic stress state} \tag{24}$$

- The stress tensor at all times is now scaled as a function of V. The scaling factor is defined as

$$f = 1 + (1 - k)V \tag{25}$$

$$\text{with} \quad k = \frac{\sigma_\mathrm{w}}{\tau_\mathrm{w}} \tag{26}$$

and the stress tensor scaled as

$$\boldsymbol{\sigma}_\mathrm{scaled} = f \cdot \boldsymbol{\sigma}. \tag{27}$$

The material constant k is $k = 2$ for ductile and $k = 1$ for brittle materials. For a pure tension/compression the stress remains unaltered. For a dominant shear load the stress tensor is scaled up by f to model the damaging effects of shear. The hydrostatic stress is scaled down, which is in good agreement with the distortion energy criterion. Brittle materials with $k = 1$ result in $f = 1$, which is compliant with the normal stress hypothesis [2].

2.3.4 Algorithm

In the most general case for a single point in space and time (a single time step) the critical plane method can be expressed as a sequential algorithm. Due to multiple nested iteration loops it is of advantage to clearly define each iterator and the respective maximum number of iterations.

- A single time step is denoted by t_i with $i = 0, 1, ..., \mathrm{n_t}$
- A single element is denoted by E_j with $j = 0, 1, ..., \mathrm{n_E}$
- A single candidate plane is denoted by P_k with $k = 0, 1, ..., \mathrm{n_P}$

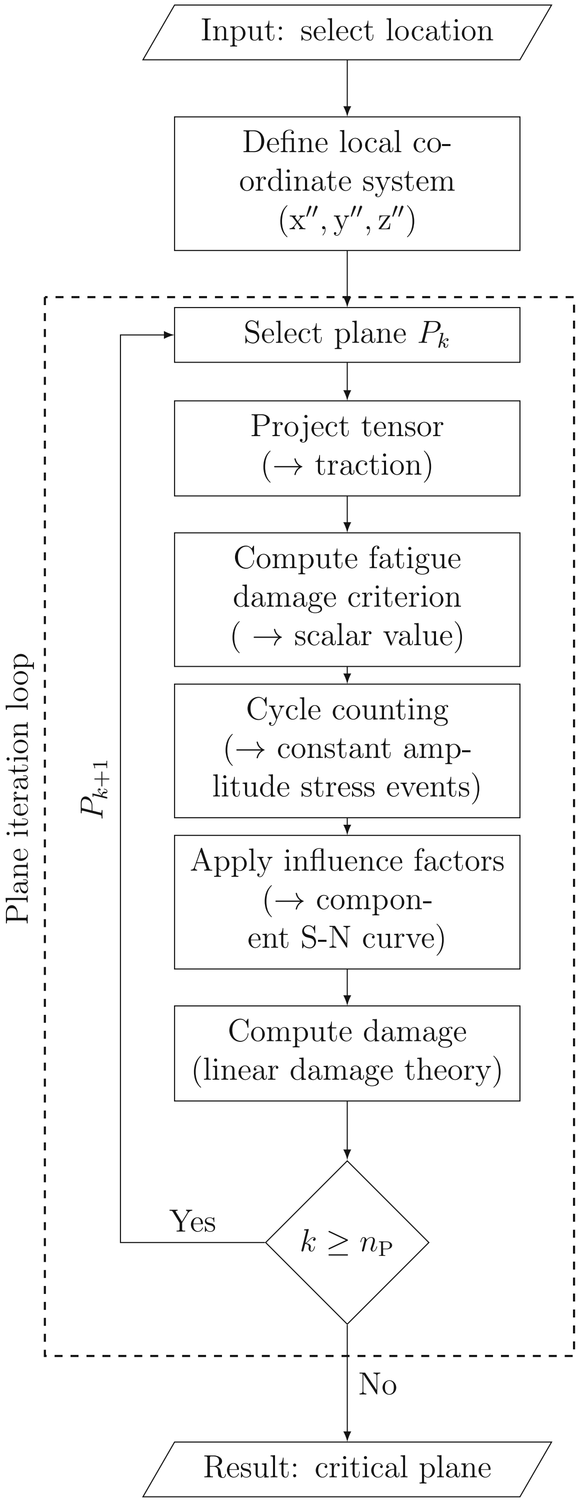

Input: select location
Define local coordinate system (x″, y″, z″)
Select plane P_k
Project tensor (→ traction)
Compute fatigue damage criterion (→ scalar value)
Cycle counting (→ constant amplitude stress events)
Apply influence factors (→ component S-N curve)
Compute damage (linear damage theory)
$k \geq n_\mathrm{P}$
Yes
No
P_{k+1}
Plane iteration loop
Result: critical plane

Figure 10: The schematic general case algorithm of CPM for one point in space and time

Fig. 10 shows the steps taken to determine the critical plane for a single location. The input data and different options at each block of code are omitted to make the representation clear.

3 Research objective and requirements

The following chapter contains the research objective and the corresponding research questions that define the scope of the project. These are based on an analysis of the literature review.

3.1 Research objective

The research objective is defined as:

To develop a Python-based fatigue tool, utilizing the CPM

The developed tool is to be as universally applicable as possible, suitable for fatigue analysis of most components of a reciprocating compressor. The critical plane method is especially suitable for numerical implementation, due to the iterative nature of the algorithm. The main advantage lies in the required discretization that is inherited through the mesh of the underlying finite element (FE) model. This ensures a high degree of geometrical insensitivity and makes CPM suitable for any geometry. CPM being a local stress concept (see Section 2.2) also eliminates the need for special component S-N curves, using the more widely available material S-N curve instead. This offers a distinct advantage early in the design phase, where actual components or prototypes may not even be available.

Some components, e.g. the crankshaft, experience highly nonproportional loading due to body rotation. This translates to large displacements of the geometry, meaning the tool must be able to consider geometric nonlinearities due to large displacements and/or deformations. This is facilitated by considering a discretized transient profile, on which the CPM is applied. CPM enables the user to choose the discretization level, both spatially and temporally, the equivalent stress theory, the mean stress correction theory, the cycle counting method and the linear damage rule. The potential for almost unlimited expansion and implementation of new functionalities is what makes CPM so appealing.

The Python programming language is chosen primarily due to its open-source nature, immense knowledge base and scalability. The interface between the Python-based CPM tool and the FEM software performing the underlying stress analysis is very important. The possibility of use with multiple FEM commercial packages requires the interface to operate on a relatively low level, typically the element level. Therefore, its transparency, ease of use and flexibility must be prioritized when creating a robust and future-proof interface. Most commercial FE packages use some sort of Python-based scripting for automation and user subroutines, translating into a more user-friendly experience.

3.2 Requirements

The previous section offers a broader description of the problem at hand, including the motivation and resulting benefits. Before the initiation of a product development process these must be translated into a list of requirements from an engineering standpoint:

- Interface: The input must consist of nodal stress and deformation results.
- Large deformation: Deformation is considered in the node location.
- Transient: Input of multiple timesteps to cover transient stress results.
- Universality: Use of either a triangular or quadrilateral face mesh.

4 Methodology

This chapter describes the methodology followed in this research project. This consists of the following subsections:

- Creating an FE model for stress analysis
- Results formatting and export
- Processing of text files
- Python implementation of the CPM
- Testing and validation
- Real-life use case

The section describing the creation of an FE model serves as a showcase, demonstrating the template to which the underlying stress analysis must adhere to. The FEM software used for structural analyses throughout the present work is Ansys Mechanical, part of the Ansys Workbench 2022 R2 commercial package.

4.1 FE model

The setup of the underlying finite element stress analysis follows the general guidelines for stress analyses. Few restrictions are imposed on the model, mainly limited to the mesh. Due to the working principle of the function defining the element local coordinate system, a single element cannot have multiple outer faces, i.e. an element edge cannot at the same time be a geometry edge. As a direct result of this requirement such elements must be excluded from the calculation. The exclusion is done on the basis of the number of surface nodes belonging to an individual element, which necessitates the specification of the element type by the user. In practice the surface mesh in the region of interest must be either triangular or quadrilateral.

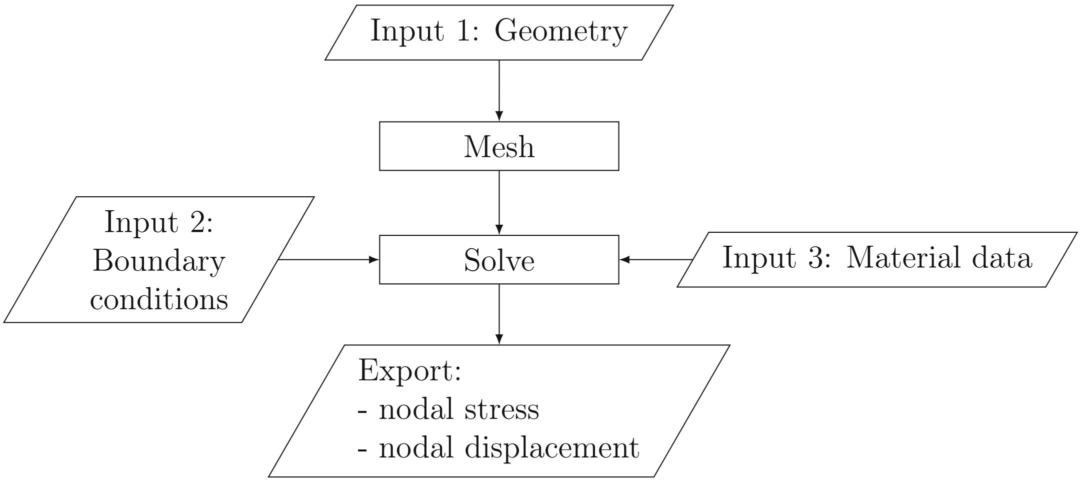


Figure 11: The general process of configuring the underlying FE model for stress analysis

As seen inf Fig. 11 the aim of the stress analysis is to provide nodal stress and displacement data for export. The region of interest is defined as a face selection ("Named selection" in Ansys) and only those data are exported. In the case of transient analyses a time increment for data export is specified.

4.2 Pre-processing of text files

In this section the pre-processing of the exported text files is described. Note that while the exact procedure is software-specific, the final formatting of the text files should be standard.

Problem: The text files containing stress data do not contain nodal displacements

Goal: Combine stress and displacement results in one file, to enable handling of geometrical non-linearities and transient analyses.

The pre-processor is implemented as a Python function within the main code, but can be used as a standalone tool as well. It represents the interface between the FEM software package and the Python-based tool and can be customised or modified for use with a different FEM package.

4.3 Python implementation

This section describes the workflow of CPM as it is to be implemented in the present thesis.

4.3.1 Data structure and type

Python is an object-oriented programming language, meaning that data is classified as objects and stored as such. Many object classes are pre-defined in Python, such as lists and a multitude of number formats. It is up to the developer to define custom classes, suited to storing data and performing actions needed for the task at hand. The developed fatigue tool defines the following object classes:

- `Model` is the master class, acting as the container of all loaded and computed data. The only method is the action of loading the text files and creating instances of lower tier objects. It facilitates the potential functionality of creating multiple meshes within the same model.

- `Mesh` also acts as the container of all loaded and computed data. It is initiated when loading the input text files. In addition, it also contains methods that perform actions on all elements. The comparison of individual element accumulated damages and mesh-level visualisation is included.

- `Element` is the single most important object class and is initiated when loading the input text files. The entire algorithm of the CPM is implemented as its method. Element-level visualisation is included.

- `Node` is the most elementary object class, containing most of the loaded data. Since there are no operations being performed on the node level it contains no custom methods.

- `Material` is the object defining and manipulating material properties. It contains the name, ultimate strength $\mathrm{R_m}$, fatigue strengths σ_w and τ_w, the stress-life curve, the critical accumulated damage $\mathrm{D_{crit}}$ and the number of loading sequences leading to fatigue failue $\mathrm{N_{inf}}$. When the loading sequence consists of only one load cycle then $\mathrm{N_{inf}}$ is equal to N. It contains a method that manipulates the stress-life curve according to the linear damage rule specified by the user. Materials can be stored for repeated use.

- `Parameters` is an object defining the parameters to be used for the entire fatigue evaluation, including CPM. It contains user-specified information about the mesh type, the fatigue damage parameter, the stress state assumptions, the means stress correction, the angle increment for the CPM and a potential stress multiplier factor to be applied. This object is saved whenever a solve is performed, so as to enable the user to repeat an evaluation with the same settings.

The choice of data types in any implementation is largely up to the user. The present code utilises a combination of lists (`list`) and arrays (`array`) to store the loaded and computed values. It is important to note that any data, even a simple number of type `int`, is an object in itself. In Python a `list` can contain an instance of any other object class, such as another `list` or a user-defined object, such as a `Node`.

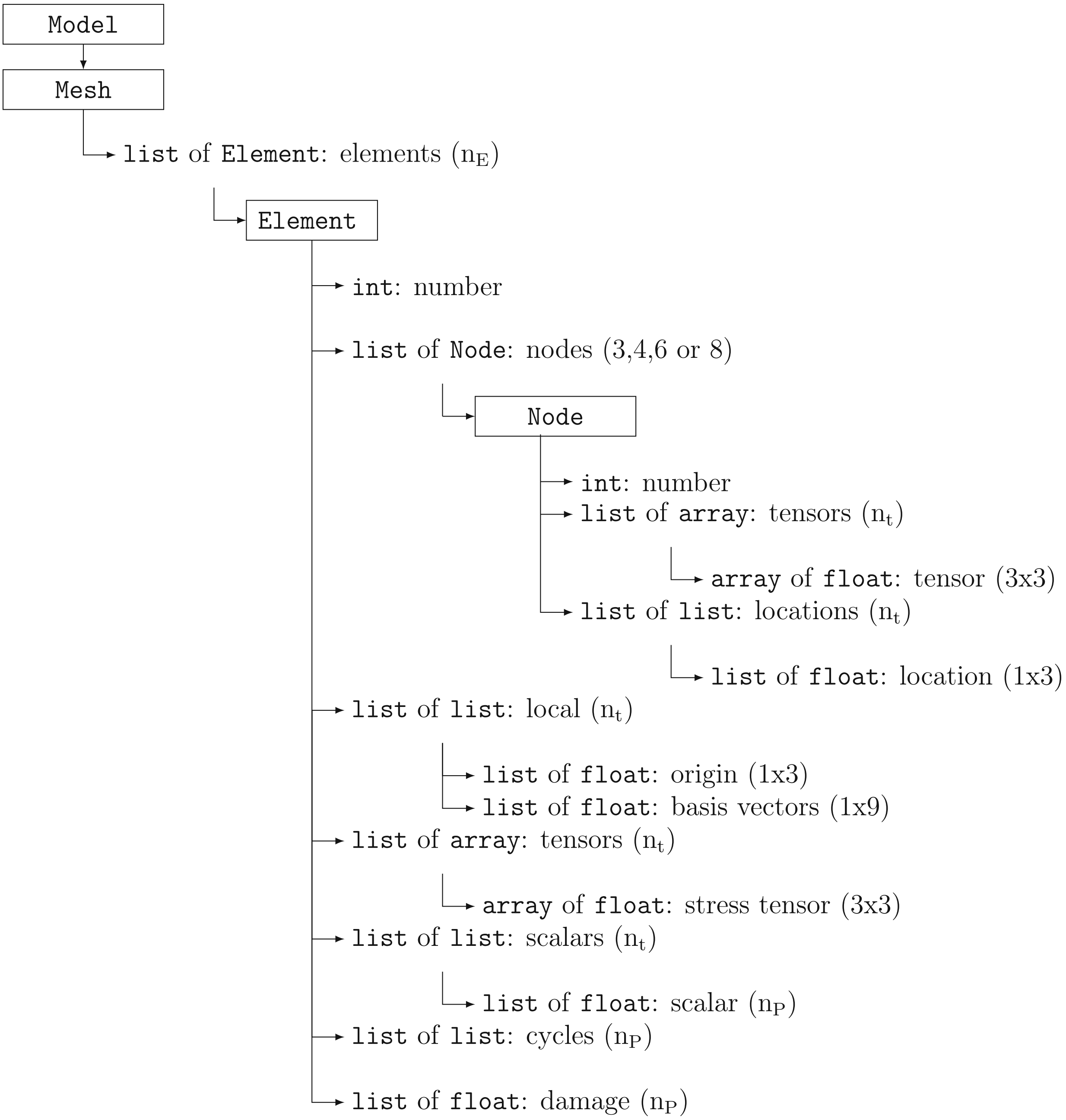

Model
Mesh
list of Element: elements (n_E)
Element
int: number
list of Node: nodes (3,4,6 or 8)
Node
int: number
list of array: tensors (n_t)
array of float: tensor (3x3)
list of list: locations (n_t)
list of float: location (1x3)
list of list: local (n_t)
list of float: origin (1x3)
list of float: basis vectors (1x9)
list of array: tensors (n_t)
array of float: stress tensor (3x3)
list of list: scalars (n_t)
list of float: scalar (n_P)
list of list: cycles (n_P)
list of float: damage (n_P)

Figure 12: Data structure of loaded and computed values, including the object type and size

Figure 12 visualises the data structure, meaning the classes and their attributes. When loading the input text files, `Node` objects are created first and their attributes are populated entirely by the loaded data, which is not modified later on. `Element` objects are initiated with an element number and a single `Node` (additional nodes are added during the loading process). All other attributes are initiated as empty lists and populated by the object methods in subsequent steps.

4.3.2 Code workflow

With defined object classes and their attributes, the concept of implementing the CPM as a Python-based tool becomes much more clear.

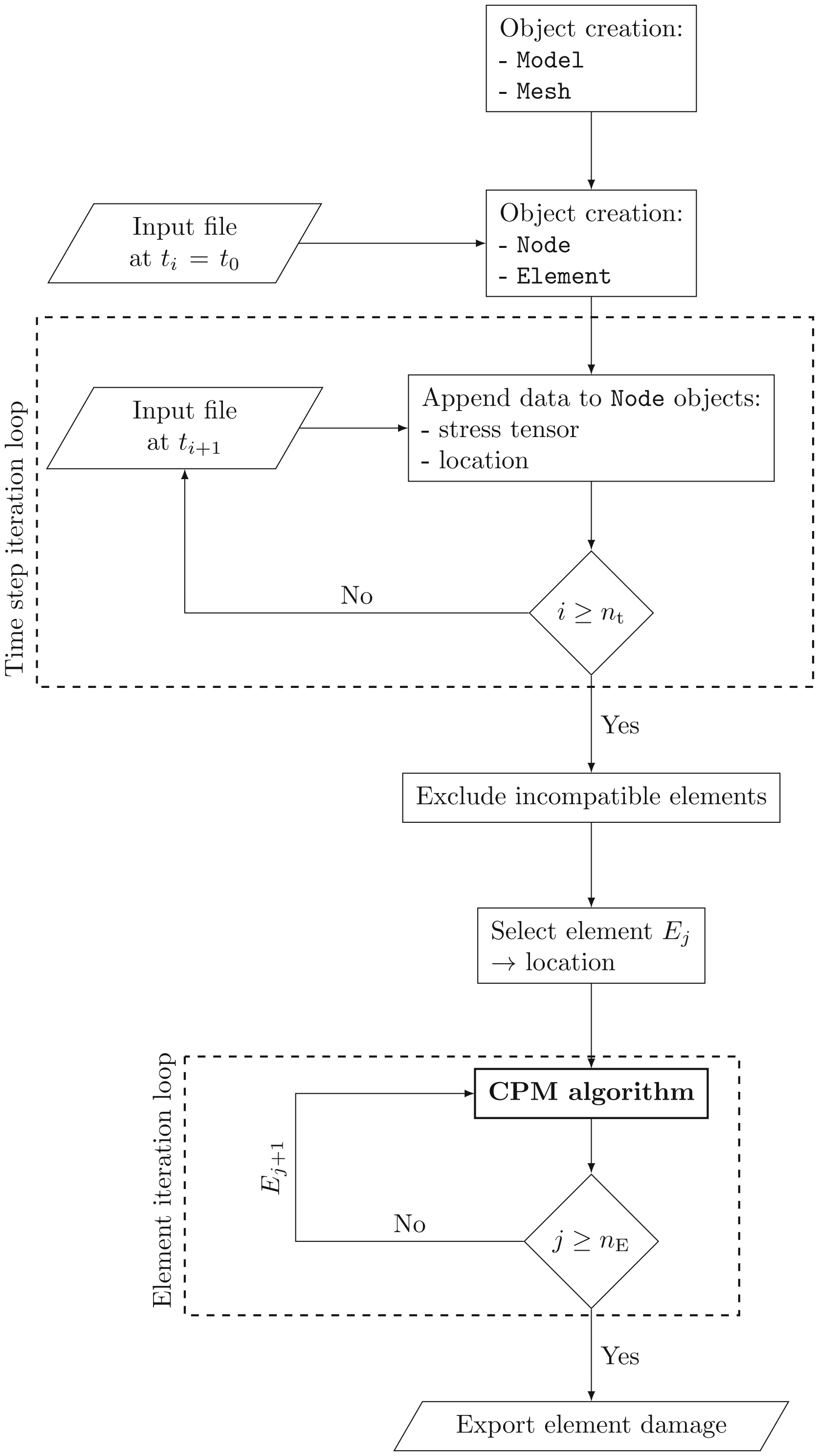


Figure 13: Schematic flowchart of the code

Fig. 13 shows the working principle of the implemented CPM on an object level.

4.3.3 Element local coordinate system

The definition of the element local coordinate system (x', y', z') is a prerequisite for all subsequent steps in the implementation of the critical plane method. Naturally, the coordinate system is defined in the finite element pre-processor, but its export is not universal to all commercial software packages. To retain the universality of the tool only initial nodal locations and displacements are exported from the stress analysis and the element local coordinate system is redefined in the fatigue tool itself by a user function.

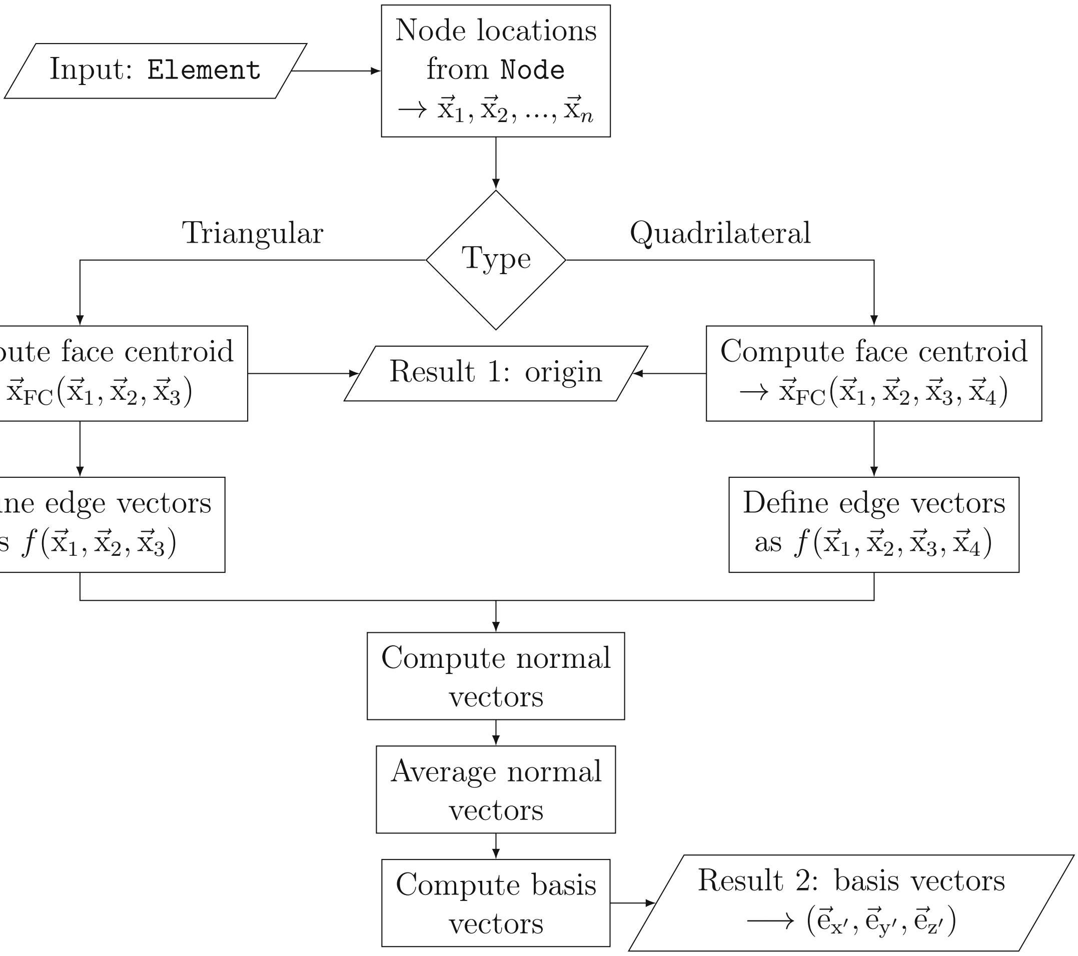

Input: Element
Node locations from Node $\rightarrow \vec{x}_1, \vec{x}_2, ..., \vec{x}_n$
Type
Triangular
Quadrilateral
Compute face centroid $\rightarrow \vec{x}_{FC}(\vec{x}_1, \vec{x}_2, \vec{x}_3)$
Result 1: origin
Compute face centroid $\rightarrow \vec{x}_{FC}(\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4)$
Define edge vectors as $f(\vec{x}_1, \vec{x}_2, \vec{x}_3)$
Define edge vectors as $f(\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4)$
Compute normal vectors
Average normal vectors
Compute basis vectors
Result 2: basis vectors $\longrightarrow (\vec{e}_{x'}, \vec{e}_{y'}, \vec{e}_{z'})$

Figure 14: Schematic workflow of defining the element local coordinate system (x', y', z') within the global coordinate system (x, y, z) for one point in time

The method presented in Fig. 14 makes use of only the position of either four or three face corner nodes for quadrilateral or triangular faces, respectively. This results in an element local coordinate system, whose orientation is arbitrary with respect to the global coordinate system. The origin is defined by the face center, the $\vec{e}_{z'}$ basis vector represents the face normal, while $\vec{e}_{x'}$ is defined as a normalized vector from the origin to an arbitrary corner node. The $\vec{e}_{y'}$ is constructed as a normalized vector orthogonal to both $\vec{e}_{x'}$ and $\vec{e}_{z'}$. This procedure is repeated for all points in time.

Note that while the (x', y', z')-system is redefined at each point in time to account for large deformations, the critical plane method is only valid as long as the (x', y', z')-system is not skewed in the process. This leads to the inherent limitation that the implementation is valid only for small element strains, making it especially useful for metals in the elastic region. Nor further excursion is done into researching the effects this may have on the results and what a realistic threshold is.

After the selection of an `Element` object E_j the locations of its surface nodes are extracted from the `Node` objects contained within. In total three or four corner nodes are considered for triangular or quadrilateral element faces, respectively. The element face centroid is computed as

$$\vec{\mathrm{x}}_{\mathrm{FC}} = \frac{1}{3}(\vec{\mathrm{x}}_1 + \vec{\mathrm{x}}_2 + \vec{\mathrm{x}}_3) \qquad \text{for a triangular face} \tag{28a}$$

$$\vec{\mathrm{x}}_{\mathrm{FC}} = \frac{1}{4}(\vec{\mathrm{x}}_1 + \vec{\mathrm{x}}_2 + \vec{\mathrm{x}}_3 + \vec{\mathrm{x}}_4) \qquad \text{for a quadrilateral face} \tag{28b}$$

where $\vec{\mathrm{x}}_1, \vec{\mathrm{x}}_2, ..., \vec{\mathrm{x}}_n$ are the vectors denoting node locations in the global coordinate system and $\vec{\mathrm{x}}_{\mathrm{FC}}$ represents the origin of the element local coordinate system.

In the following step edge vectors connecting the corner nodes are defined for triangular element faces as

$$\vec{\mathrm{v}}_1 = \vec{\mathrm{x}}_2 - \vec{\mathrm{x}}_1 \tag{29a}$$

$$\vec{\mathrm{v}}_2 = \vec{\mathrm{x}}_3 - \vec{\mathrm{x}}_1 \tag{29b}$$

$$\vec{\mathrm{v}}_3 = \vec{\mathrm{x}}_1 - \vec{\mathrm{x}}_2 \tag{29c}$$

$$\vec{\mathrm{v}}_4 = \vec{\mathrm{x}}_2 - \vec{\mathrm{x}}_3 \tag{29d}$$

and for quadrilateral faces as

$$\vec{\mathrm{v}}_1 = \vec{\mathrm{x}}_2 - \vec{\mathrm{x}}_1 \tag{30a}$$

$$\vec{\mathrm{v}}_2 = \vec{\mathrm{x}}_3 - \vec{\mathrm{x}}_1 \tag{30b}$$

$$\vec{\mathrm{v}}_3 = \vec{\mathrm{x}}_2 - \vec{\mathrm{x}}_4 \tag{30c}$$

$$\vec{\mathrm{v}}_4 = \vec{\mathrm{x}}_3 - \vec{\mathrm{x}}_4. \tag{30d}$$

All subsequent steps are universal to both mesh types. In the following step two normalized vectors are constructed from a cross product of a pair of edge vectors as

$$\vec{\mathrm{n}}_1 = \frac{\vec{\mathrm{v}}_1 \times \vec{\mathrm{v}}_2}{||\vec{\mathrm{v}}_1 \times \vec{\mathrm{v}}_2||} \tag{31a}$$

$$\vec{\mathrm{n}}_2 = \frac{\vec{\mathrm{v}}_3 \times \vec{\mathrm{v}}_4}{||\vec{\mathrm{v}}_3 \times \vec{\mathrm{v}}_4||} \tag{31b}$$

and the average of the two is taken as the final normal vector

$$\vec{\mathrm{n}} = \frac{\frac{1}{2}(\vec{\mathrm{n}}_1 + \vec{\mathrm{n}}_2)}{||\frac{1}{2}(\vec{\mathrm{n}}_1 + \vec{\mathrm{n}}_2)||}. \tag{32}$$

The normalized basis vectors of the element local coordinate system are computed as

$$\vec{\mathrm{e}}_{\mathrm{x}'} = \frac{\vec{\mathrm{x}}_1 - \vec{\mathrm{x}}_{\mathrm{FC}}}{||\vec{\mathrm{x}}_1 - \vec{\mathrm{x}}_{\mathrm{FC}}||}, \quad \vec{\mathrm{e}}_{\mathrm{y}'} = \frac{\vec{\mathrm{e}}_{\mathrm{x}} \times \vec{\mathrm{n}}}{||\vec{\mathrm{e}}_{\mathrm{x}} \times \vec{\mathrm{n}}||} \quad \text{and} \quad \vec{\mathrm{e}}_{\mathrm{z}'} = \vec{\mathrm{n}}. \tag{33a}$$

This process is repeated for each time step, ensuring that the element local coordinate system is not skewed by the deformation of the structure and all base axes remain orthogonal.

4.3.4 Computing the fatigue damage parameter

The essential core of any implementation of CPM is the projection of the stress tensor onto all candidate planes and the computation of the fatigue damage parameter. The candidate planes are defined with respect to the element local coordinate system (x', y', z') and can be stored permanently or as a temporary variable within one iteration. The chosen approach depends primarily on the needs of post-processing, where access to individual plane definitions may be required. A temporary definition of candidate planes is chosen for the present thesis. For computing the traction normal component only the plane normal is required, for the shear components of the traction, however, a full set of base axes is needed. To avoid repeated work we use the readily available element local coordinate system (x', y', z') and treat it as it were a candidate plane local coordinate system (x'', y'', z'') for the first time step, which we the rotate accordingly.

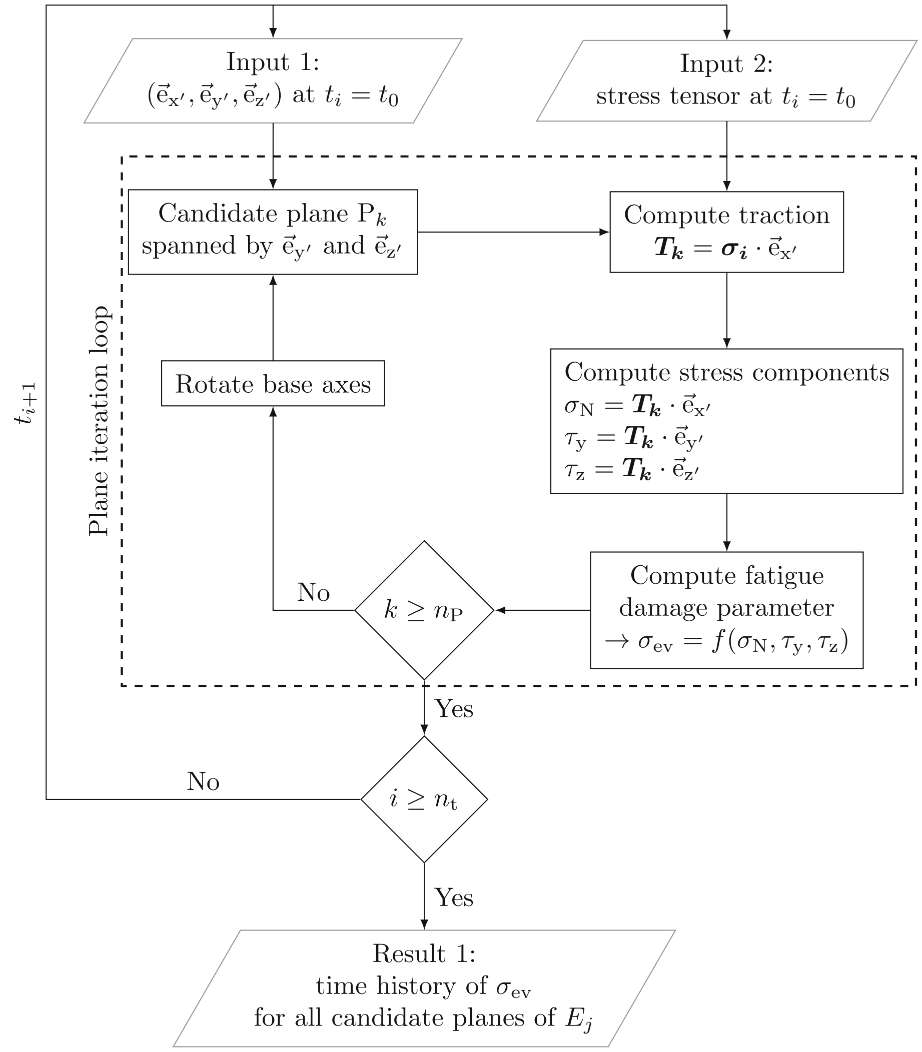

Input 1:
$(\vec{e}_{x'}, \vec{e}_{y'}, \vec{e}_{z'})$ at $t_i = t_0$
Input 2:
stress tensor at $t_i = t_0$
Candidate plane P_k
spanned by $\vec{e}_{y'}$ and $\vec{e}_{z'}$
Compute traction
$\boldsymbol{T_k} = \boldsymbol{\sigma_i} \cdot \vec{e}_{x'}$
Rotate base axes
Compute stress components
$\sigma_N = \boldsymbol{T_k} \cdot \vec{e}_{x'}$
$\tau_y = \boldsymbol{T_k} \cdot \vec{e}_{y'}$
$\tau_z = \boldsymbol{T_k} \cdot \vec{e}_{z'}$
Plane iteration loop
t_{i+1}
No
$k \geq n_P$
Compute fatigue
damage parameter
$\rightarrow \sigma_{ev} = f(\sigma_N, \tau_y, \tau_z)$
Yes
No
$i \geq n_t$
Yes
Result 1:
time history of σ_{ev}
for all candidate planes of E_j

Figure 15: Working principle of the implemented function computing the time history of the fatigue damage parameter $(\sigma_{ev}(t))$ for all candidate planes of one element E_j

The working principle of the candidate plane iteration loop is schematically presented in Fig. 15. After computing the fatigue damage parameter for one plane the (x'', y'', z'')-system is rotated around the (x', y', z')-system base axes, depending on whether the stress state is assumed to be two- or three-dimensional.

4.3.5 Computing the damage

Once the time history of the fatigue damage parameter has been computed for all candidate planes of all elements, the damage can be computed.

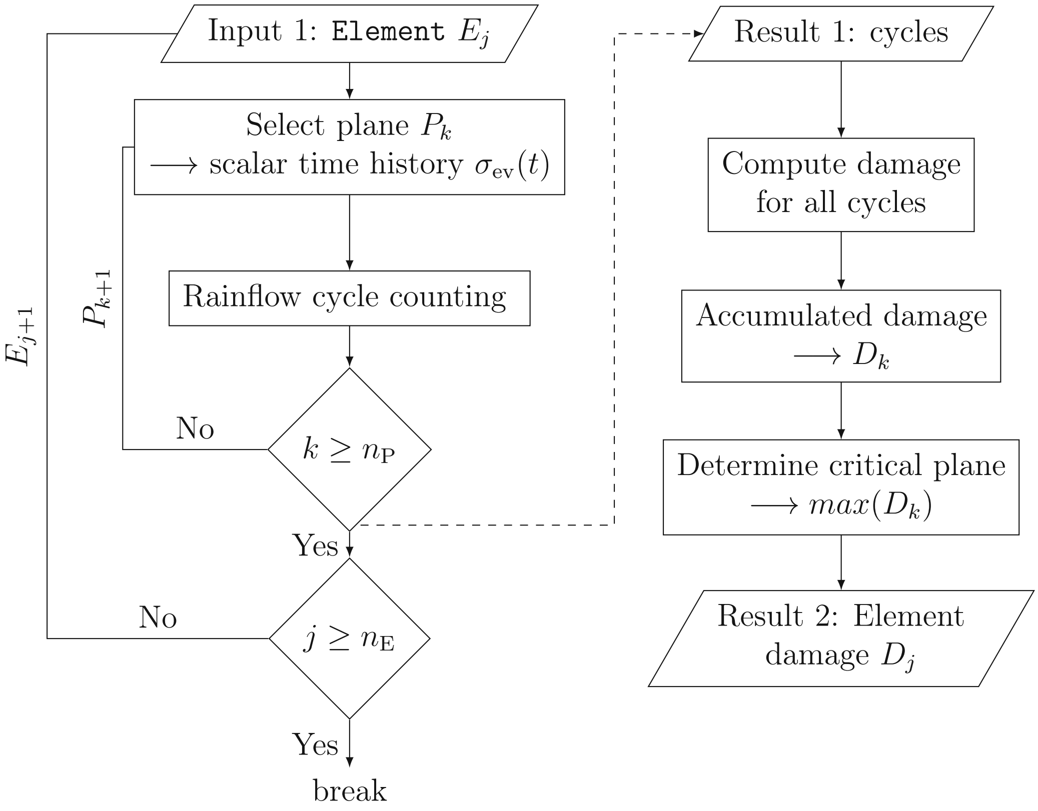

Input 1: Element E_j
Select plane P_k
$\longrightarrow$ scalar time history $\sigma_{\mathrm{ev}}(t)$
Rainflow cycle counting
$k \geq n_{\mathrm{P}}$
No
Yes
P_{k+1}
E_{j+1}
$j \geq n_{\mathrm{E}}$
No
Yes
break
Result 1: cycles
Compute damage for all cycles
Accumulated damage $\longrightarrow D_k$
Determine critical plane $\longrightarrow max(D_k)$
Result 2: Element damage D_j

Figure 16: Schematic workflow of the process of cycle counting for all elements

Fig. 16 shows the basic concept and the loops needed to effectively compute the damage on an element level. Element E$_j$ serves as the input and represents the start of the element iteration loop (marked by the index j). Within the element the first candidate plane (P$_k$) is chosen and the time history of the fatigue damage parameter ($\sigma_{\mathrm{ev}}(t)$) for this plane is extracted from the attribute of the element. This action marks the start of the plane iteration loop (marked by the index j) nested within the element iteration loop. A rainflow cycle counting technique (see Sec. 2.2.5) is applied to the time history of the fatigue damage. Once all candidate planes within E$_j$ have been treated and a list of cycles have been stored as an attribute of the element E$_j$, the algorithm proceeds into the next iteration of the element loop with E$_{j+1}$.

Once all cycles of all candidate planes of all elements have been determined the implementation proceeds with assigning damage to each cycle, subsequently evaluating fatigue damage on a candidate plane level, element level and mesh level, in that order. Note that Fig. 16 shows the cycle counting and damage computation as parallel processes with the aim of being as concise as possible.

Each cycle is treated as a stress event, for which a mean stress correction must be performed. The mean stress correction (see Sec. 2.2.4) is specified by the user before the start of the evaluation. Interpolating the stress-life curve, the damage corresponding to the individual cycle is computed. The sum of all damages of the individual cycles corresponds to the accumulated plane damage (D$_k$), which forms the basis of finding the critical plane of an element. The plane that accumulates the most damage is the critical one and is representative of damage on an element level (D_j). This value is not saved as an attribute of an `Element` object, but is rather exported as a text file once all elements have been processed. Note that the choice of the fatigue damage parameter (see Sec. 2.3.3) directly influences which candidate plane is the critical one.

4.4 Post-processing

All methods presented in Sec.4.3 are aimed at computing the damage on an element level. The damage after one loading sequence is the primary result available after the fatigue evaluation. Typically, the user requires results that are more easily comparable or offer a conclusion directly. To this effects certain post-processing function are implemented, which are called when exporting the results as text files.

Element damage D_j

The element damage D_j is a primary result, representing the highest accumulated damage within one element (see Fig. 16). Each candidate plane accumulates a damage D_k, whereas D_j is assigned the maximum of all D_k values.

Element life N_j

The element life N_j in number of cycles or loading sequences is a result derived from the element damage, considering the critical damage for fatigue failure of the material $\mathrm{D}_{\mathrm{crit}}$ and is computed as

$$\mathrm{N}_j = \frac{\mathrm{D}_{\mathrm{crit}}}{\mathrm{D}_j}. \tag{34}$$

Element safety factor S_j

The element safety factor S_j is subsequently derived from the element life N_j, considering the number of cycles or loading sequences defined as infinite life $\mathrm{N}_{\mathrm{inf}}$ and is computed as

$$\mathrm{S}_j = \frac{\mathrm{N}_j}{\mathrm{N}_{\mathrm{inf}}}. \tag{35}$$

4.5 Testing and validation

The CPM implementation described in detail in Sec. 4.3 is based on the author's interpretation of CPM and subsequently developed concept of implementation. As any product development, the process of developing the fatigue tool is prone to error. With the aim of detecting the errors in concept and/or implementation, a thorough testing and validation procedure is set up. Based on simple FE models it aims to isolate the functionalities to be tested as best as possible. It is assumed that loading the input files and initiating objects is correct and strictly speaking not a part of the CPM implementation. Therefore, the testing and validation procedure is meant to detect errors in object methods. We refer to Sec. 3.2 for the list of functionalities to be tested and summarize them as:

- The definition of the element local coordinate system and its re-definition during subsequent time steps and large deformations of the structure. This is to be tested for both the triangular and the quadrilateral face mesh.
- The definition of candidate plane normal vectors for a tri-axial stress state.
- The correctness of primary stress values arising from the projection of the stress tensor onto the candidate plane.
- The ability to correctly capture and represent a transient loading scenario.
- The correctness of the rainflow cycle counting technique for both single-step and multi-step static analyses, as well as for fully transient analyses.
- The correctness of the mean stress correction, linear damage rule application, stress-life curve interpolation and final damage computation.

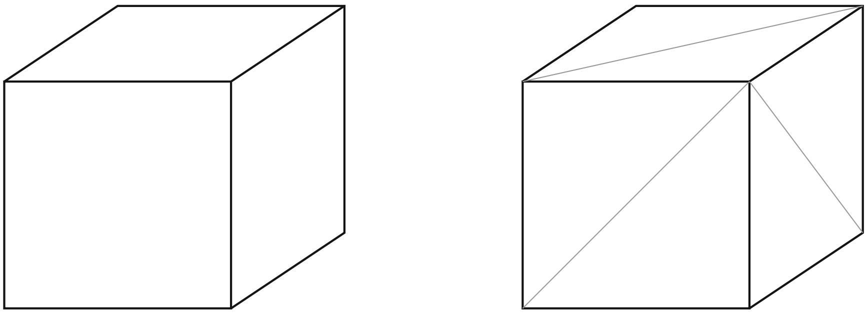
Additionally to testing the primary functionality of the tool, some auxiliary functions require validation as well. These are:

- The generation of synthetic stress-life curves from limited data.
- The interpolation of the stress-life curve.
- The application of the linear damage theory.
- The exclusion of incompatible elements from the computation.

The above functionalities may not fall into the originally defined scope of the thesis, but are of paramount importance for correct fatigue evaluation and are just as error-prone in the implementation as any other part.

4.5.1 Simple cube

A single element represents the most basic FE model possible. Seemingly having little to do with real-life use cases, it enables the developer to thoroughly test functionalities on the element level and is by far the most vital part of the testing and validation procedure. In terms of the implementation it can be understood as an examination of the `Element` object methods.



(a) One hexahedron element

(b) Four tetrahedron elements

Figure 17: Geometry and mesh of the simple cube

It is modelled as a 1 mm × 1 mm × 1 mm cube that can be meshed with either four tetrahedra or one hexahedron element, of either linear or quadratic order. The following loading scenarios are defined:

- **Load case 1:** Transient large shear deformation
- **Load case 2:** Single-step uniaxial compression
- **Load case 3:** Multi-step uniaxial compression and tension
- **Load case 4:** Multi-step uniaxial tension and hydrostatic compression

The following subsections define multiple loading scenarios in detail, explain the motivation and list the aims. The material used is structural steel as defined in Ansys Mechanical 2022.

Load case 1: Transient large shear deformation

The presented load case introduces a displacement to the single hexahedron element in a way that causes a simple shear deformation.

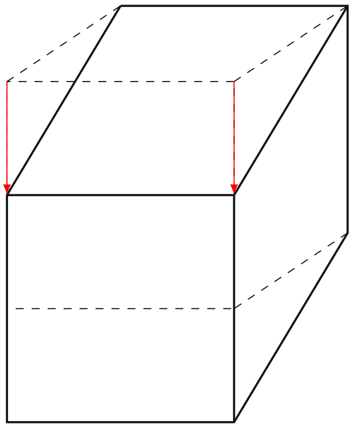


Figure 18: Large shear deformation of a single hexahedron element

Resulting in a macroscopic deformation as seen in Fig. 18, it facilitates the validation of the element local coordinate system and its re-definition in subsequent time steps. Due to unrealistically high element strain (see Sec. 4.3.3) it is not suitable for stress evaluation or comparison with reference stress results. The correctness of the results is evaluated based on visual inspection.

Keeping the geometrical dimensions, but using tetrahedron elements instead of hexahedrons, results in a cube consisting of four elements. Analogous to the hexahedron element such a model is used for validating the definition of the element local coordinate system on triangular faces (see Sec. 4.3.3).

Load case 2: Single-step uniaxial compression

The presented load case introduce a pressure onto one face of the cube. The opposite face is supported in a way to counteract the force, but allow for free deformation of the cube as a result of the compression and the Poisson effect. This results in a uniform stress distribution within the element. Note, that it is vital for the stress to be uniformly distributed within the element, otherwise a comparison of surface values from CPM and element values from the FE package is not valid. The analysis is repeated with different pressure values to test multiple functionalities of the code.

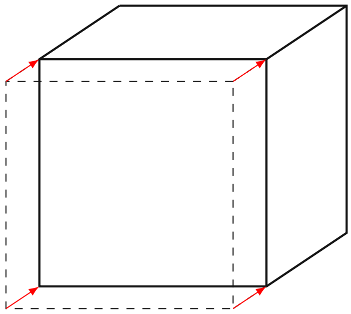


Figure 19: Uniaxial compression of a single hexahedron element

Owing to the acceptably small element strain, the deformation is barely visible. However, the force applied is within the bounds of reason and enables a validation of stress values. A relatively low number of candidate planes (8 in total) is defined, under the assumption of a plane stress state, making a direct value-to-value comparison feasible.

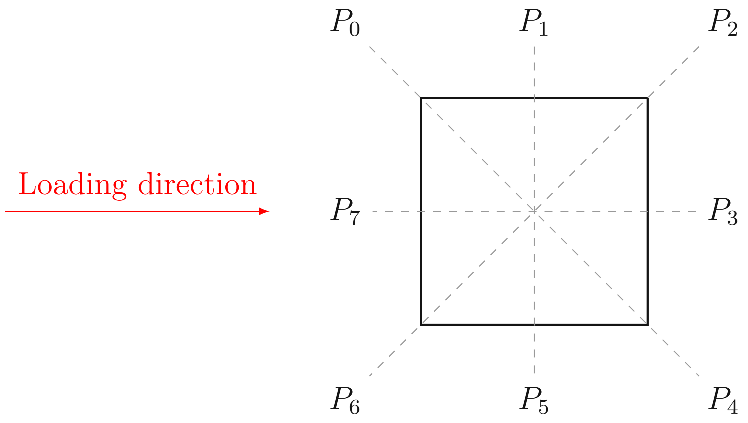


Figure 20: Eight candidate planes defined for the single hexahedron element

It is clear from Fig. 20 that candidate planes are either aligned with the loading direction, lie perpendicular to it or at an angle of 45° to it. It is visible from Fig. 20 that half the planes are duplicates with inverted normal vectors, which is the case with the plane stress state assumption. However, this is no longer valid in when considering all the planes for a triaxial stress state. For the sake of universality the planes are also not omitted in the plane stress state.

In the case of uniaxial compressing stress state these planes are used to compare stress values computed via CPM to values computed on an element basis by the FE software. Normal stress in the loading direction is compared for the aligned (P_3 and P_7) and perpendicular planes (P_1 and P_5), while the maximum shear stress is compared for the 45° candidate planes (P_0, P_2, P_4 and P_6). In a deformed state shown in Fig. 19 the planes become slightly misaligned, which can be remedied by skipping the pre-processing of input text files, where the deformation is added to node locations. Note that no such direct comparison is not possible for a triangular face mesh as the planes typically do not coincide with the 45° planes.

The load case also serves the purpose of validating the cycle counting to a certain extent, the Goodman mean stress correction and damage calculation. The fatigue damage parameter time history is qualitatively assessed. The computed damage values are comparable to calculations done by hand or to commercial FE fatigue evaluation packages, such as the fatigue tool in Ansys Mechanical 2022 R1.

Load case 3: Multi-step uniaxial compression and tension

In terms of boundary conditions set-up of the FE model the presented load case is analogous to single-step uniaxial compression (see Fig. 19). However, instead of a single load step, multiple ones are defined, with changing magnitude and sign.

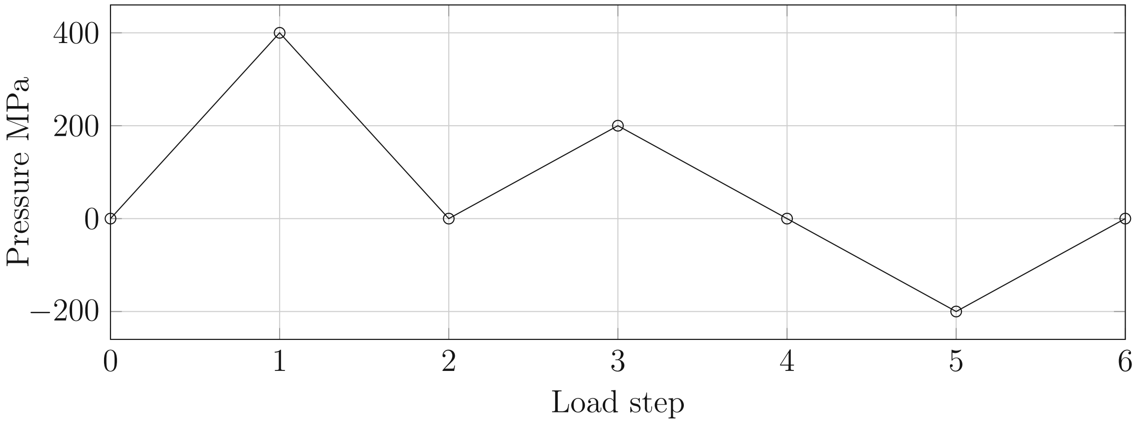
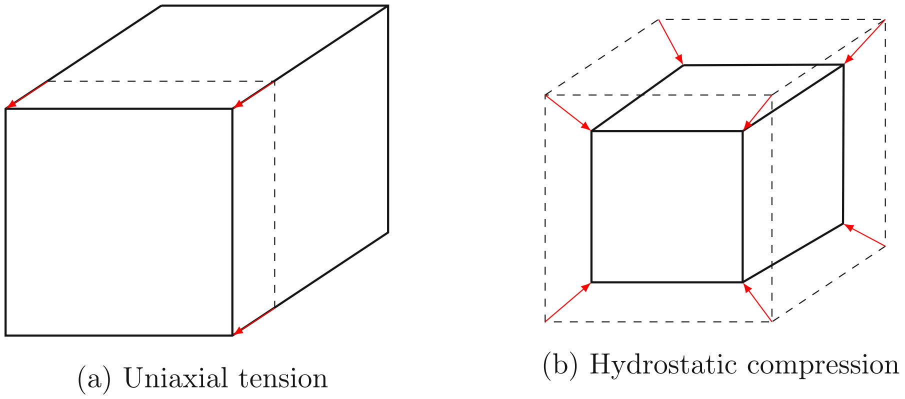


Figure 21: Pressure loading profile

In Fig. 21 the pressure loading profile is shown, with respect to the load step number. It aims to test the multi-step analysis functionality of the developed tool, with an emphasis on fatigue parameter time history. A change in sign of the pressure aims to validate the implementation of equations given in Sec. 2.3.3.

Load case 4: Multi-step uniaxial tension and hydrostatic compression

The following load case aims to validate the computed stress values without the plane stress assumption. As a control case a uniaxial tension load is applied analogously to load case 2.


(a) Uniaxial tension

(b) Hydrostatic compression

Figure 22: Loads applied to single hexahedron element

In the following step the tension is unloaded and a uniform pressure is applied to all faces of the cube, resulting in a hydrostatic stress state depicted in Fig. 22b. Just as depicted in Fig. 20, an angle increment of 45° is used for defining candidate planes, both for θ and ϕ (see Sec. 2.3).

The 45° angle increment results in a candidate plane configuration depicted in Fig. 23. The hydrostatic stress state is marked by the absence of shear stresses, therefore, the only

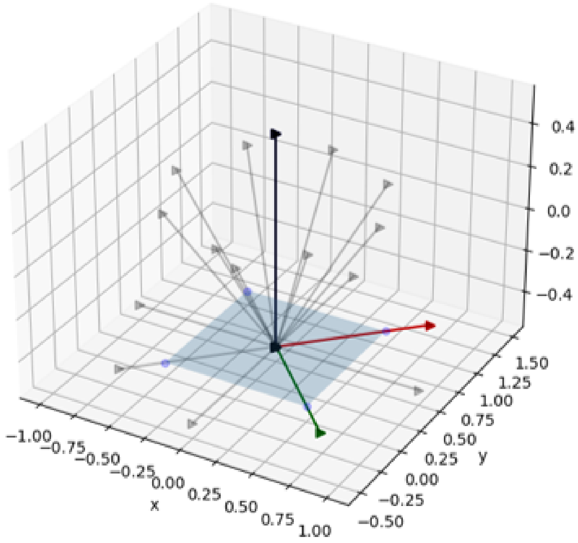


Figure 23: Candidate plane configuration for a 3D stress state

directly comparable stress value is the normal stress acting in each candidate plane. For uniaxial tension one expects a reduction of normal stress with increasing ϕ, while it should remain constant for all candidate planes in a pure hydrostatic stress.

4.5.2 Cantilever beam with a square cross-section

The simple cube test case is vital for validating the code functionality, but may not reveal all errors. As only a single element is present it omits the element iteration loop. To this effect additional test cases are set up on a cantilever beam model for further and thorough testing. The following load cases are defined:

- **Load case 2:** Single-step uniaxial tension
- **Load case 1:** Single-step uniaxial bending
- **Load case 2:** Single-step uniaxial torsion

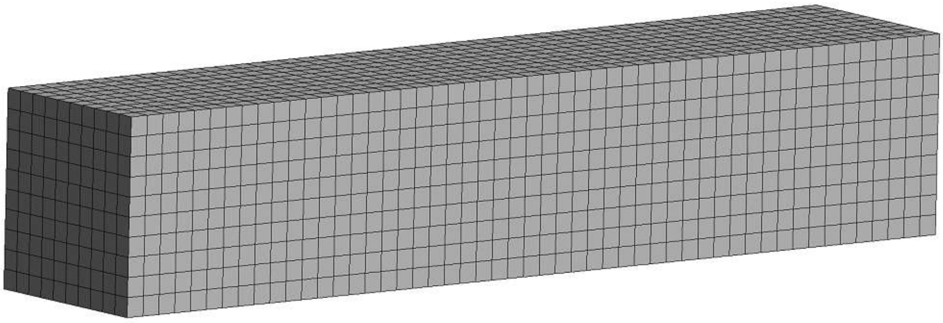


Figure 24: Cantilever beam with a square cross-section meshed using hexahedron elements

As shown in Fig. 24, the cantilever beam with a square cross-section is meshed using a relatively fine hexahedron mesh. A larger number of candidate planes is used, corresponding to a finer angle increment, while keeping the tool setting for plane stress assumption. A value-to-value comparison is not the aim of this test case, but rather a qualitative comparison with results obtained by the FE software. Maximum shear stress and maximum normal stress are most easily comparable. The comparison is repeated using a tetrahedron mesh.

An additional goal is to test the procedure of excluding incompatible elements from the computation. When evaluating all four side faces of the geometry, the elements having two outer faces must be excluded.

4.5.3 Cantilever beam with a circular cross-section

The following test case is similar to the cantilever beam, but now with a circular cross-section, and is meant as an additional case for qualitative assessment. The following load cases are defined:

- **Load case 2:** Single-step uniaxial tension
- **Load case 1:** Single-step uniaxial bending
- **Load case 2:** Single-step uniaxial torsion

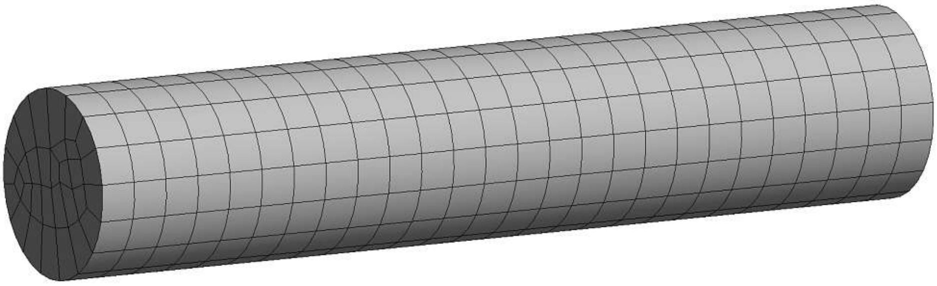


Figure 25: Cantilever beam with a circular cross-section meshed using hexahedron elements

As seen in Fig. 25 the mantle of the circular shaft does not contain any geometrical edges, therefore no exclusion of incompatible elements is expected. Up to this point the tetrahedron mesh functionality is assumed to have been thoroughly tested and this configuration is omitted for the circular shaft. However, to explore the effects of element shape on the computation, the geometry is meshed using hexahedron elements with a high aspect ratio.

4.6 Real-life applications

The testing and validation procedure described in Sec. 4.5 is aimed at progressively testing and validating the implementation of the CPM, as well as any auxiliary functions needed. It offers a good basis for evaluating the fatigue tool, but fails to demonstrate the use on real-life examples an engineer might encounter in his/her day-to-day tasks. In Sec. 1 the fatigue problem is presented for the case of suspension springs.

Two distinct use-cases are set up to demonstrate the intended use of the tool. A fatigue evaluation of a compressor crankshaft is undertaken as a simplified, single-step approach, serving as a preliminary fatigue evaluation. Secondly, a more involved transient model of a suspension spring is evaluated for fatigue. The following sections describe the models, setup and CPM configuration in more detail.

4.6.1 Compressor crankshaft

A rotating crankshaft is a classic example of a component under cyclic loading, being transmitted over the connecting rod from the cylinder. The loading is very predictable and periodic with one rotation of the crankshaft. A traditional, nominal stress approach is often chosen for fatigue evaluation, due to easily definable cycles. However, traditional approaches typically require form correction factors, often forcing the engineer to create un-notched models as well. This is time consuming and redundant in the case of a local stress approach, such as CPM.

For a basic fatigue evaluation a single-step, static structural analysis is set up, representing the time and position of critical load. Typically this occurs in the vicinity of the top dead center (TDP).

FE Model

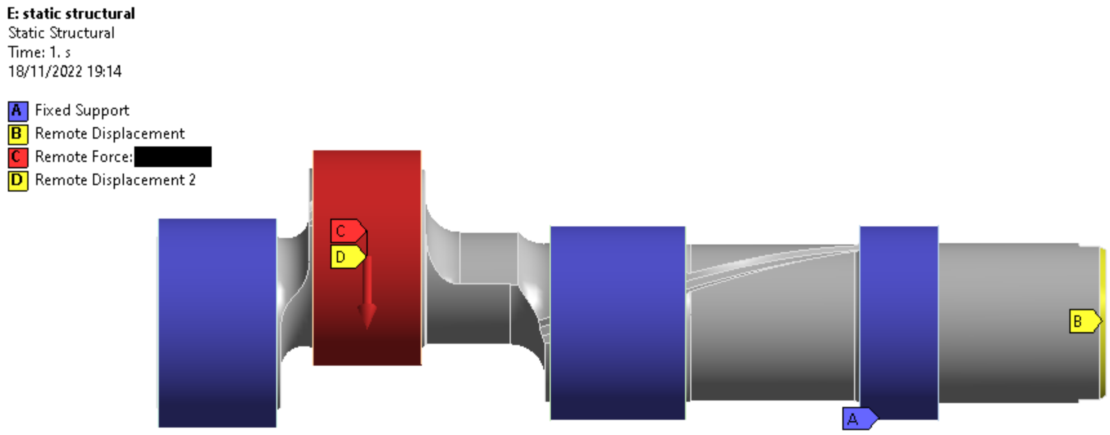


Figure 26: Boundary conditions applied on the crankshaft

The crankshaft is supported by three bearings, simplified as hollow cylinders in the model. The piston force is applied through the fourth sliding bearing as depicted in Fig. 26. Rigid body rotation is prevented by fixing the bottom of the crankshaft. A tetrahedron mesh is used. A total of one file is exported for a single-step static analysis.

Material

The crankshaft is made of EN-GJS-700-2 cast iron material.

Table 1: EN-GJS-700-2 mechanical properties

Young's modulus E	176 GPa
Poisson's ratio ν	0.28
Ultimate strength R_m	700 MPa
Endurance limit σ_w	280 MPa
Endurance limit τ_w	630 MPa
Critical damage D_crit	1.0

Table 1 summarizes the material data used in the FE model and for the subsequent fatigue evaluation. The stress-life curve is synthesized, using the endurance limit σ_w and the slope of the curve $\mathrm{k} = -5$, taken from [2]. The linear damage rule, as it is implemented in the code, directly modifies the existing stress-life curve data and extends it below the endurance limit. The modified Miner rule is chosen (see Sec. 2.2.6).

CPM settings

It is assumed that fatigue failure initiates at the surface of the crankshaft, making the plane stress assumption valid.

Table 2: CPM settings

Mesh type	tri
Fatigue damage criterion	scaled normal stress
Mean stress correction	SWT
Number of planes	72
N_inf	10^6

Table 2 summarizes the settings specific to CPM, used for the fatigue evaluation. The scaled normal stress in critical plane fatigue damage criterion is chosen (see Sec. 2.3.3), as suggested by [18]. An angle increment of 5°, as suggested by [2], results in a total of 72 candidate planes per element. The present analysis considers one loading and unloading of the crankshaft, corresponding to one crankshaft rotation during actual operation. Consequently, one loading sequence is equal to one loading cycle, making the choice of N_inf trivial.

4.6.2 Compressor suspension spring

Suspension springs are the most common component to experience fatigue failure in a reciprocating compressor. The starting and stopping of the compressor is especially damaging, due to large lateral deflections (see Sec. 1). The movement in one starting or stopping sequence is very random and can only be adequately described by measurement data. Therefore, a transient simulation is configured, using actual measurement data for transversal deflection of the top spring holder. As the spring holders affect the spring deformation they are included in the analysis and contact between the coils and holders is defined.

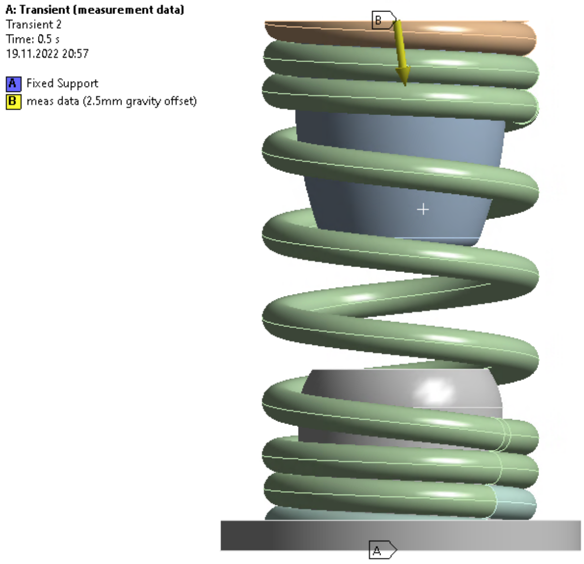

A: Transient (measurement data)
Transient 2
Time: 0.5 s
19.11.2022 20:57
A Fixed Support
B meas data (2.5mm gravity offset)
B
A

Figure 27: Displacement boundary conditions applied on the suspension spring

Figure 27 shows the model setup for the transient analysis. The bottom of the spring is fixed via the bottom holder, which is in turned fixed to the ground. A transient displacement profile is applied to the base of the top holder, effectively moving the top of the spring with it.

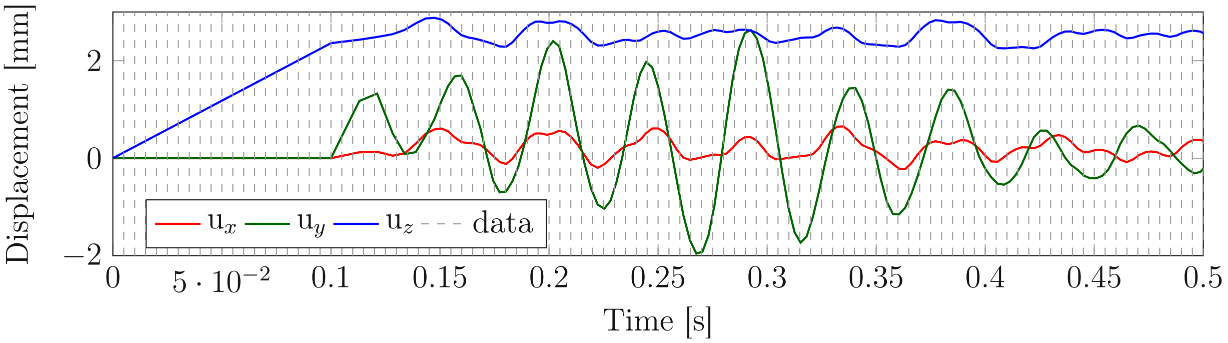


Figure 28: The displacement profile of the top spring holder based on measurement data

Figure 28 shows the displacement profile applied to the top of the holder. The measurement data describes the movement in one stopping sequence, it does not, however, contain the static compression due to gravity. To this effect a ramp is added as the initial load step, gradually compressing the spring according to the weight of the pump unit.

Material

The suspension spring is made of shot-peened steel EN-10270-2-VDSiCr music wire.

Table 3: EN-10270-2-VDSiCr mechanical properties

Young's modulus E	210 GPa
Poisson's ratio ν	0.3
Ultimate strength R_m	2100 MPa
Endurance limit σ_w	945 MPa
Ratio $\sigma_\mathrm{w}/\tau_\mathrm{w}$	0.923
Critical damage D_crit	1.0

The mechanical properties are summarized in Tab. 3. The endurance limit is initially not known and is computed from the ultimate strength as

$$\sigma_\mathrm{w} = 0.45 \cdot R_\mathrm{m}, \tag{36}$$

where 0.45 is a material-specific constant suggested by [2]. The endurance limit τ_w is not known, but its absolute value is not relevant, as only the ratio $\sigma_\mathrm{w}/\tau_\mathrm{w}$ ever appears in the fatigue damage parameter equations (see Sec. 2.3.3). As the musical wire is a steel the ration is assumed to be similar to the one of steel as defined by Ansys Mechanical for steel.

CPM settings

The examination of failed suspension springs has revealed that all fatigue cracks initiate on the surface of active coils, close to the transition to inactive coils. There is not actual contact taking place and it is therefore concluded that a plane stress assumption is valid in this case. Based on the transient displacement profile enough data points are exported to cover capture all maxima and minima (see Fig. 28).

Table 4: CPM settings

Mesh type	quad
Fatigue damage criterion	total shear stress
Mean stress correction	SWT
Number of planes	20
N_inf	$500 \cdot 10^3$

Table 4 summarizes the CPM setting used in the present fatigue evaluation. The dominant stress component in helical springs is shear stress, therefore the total shear stress in critical plane fatigue damage parameter is chosen (see Eq. 18). Due to the amount of computational resources required for a fatigue evaluation of a fully transient loading sequence, the angle increment for candidate planes is increased to 18°, resulting in a total of 20 candidate planes per element. One stopping sequence is no longer equivalent to a single loading cycle, as understood in traditional fatigue of materials. Therefore the choice of N_inf is no longer trivial and must represent some physical limit. In this case the limit is given by the reliability testing procedure, where $300,000$ to $500,000$ starting and stopping sequences are defined as the passing criterion.

5 Results

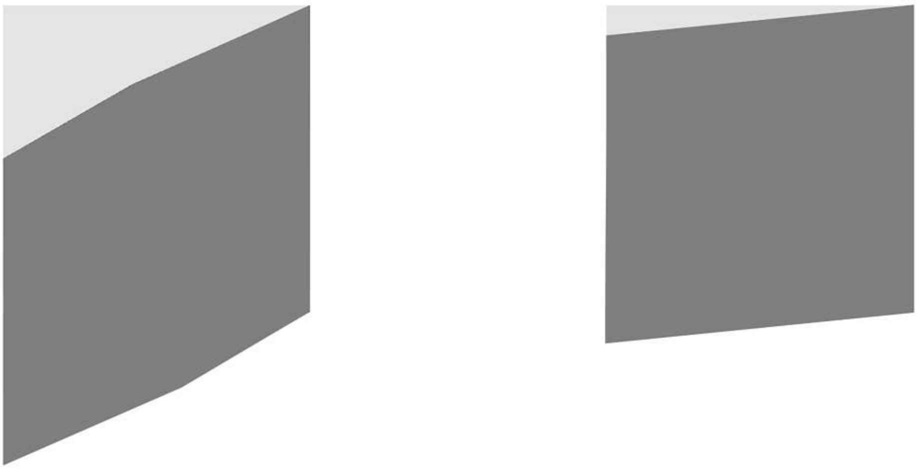
This chapter describes the results of the testing and validation procedure. Some cases are described in more detail as others, so as not to overstep the scope of the present thesis.

5.1 Simple cube

The simple cube results are described in most detail as they are of vital importance for validation. The comparison is very straightforward and results can in some cases be compared to analytical solutions.

Load case 1: Transient large shear deformation

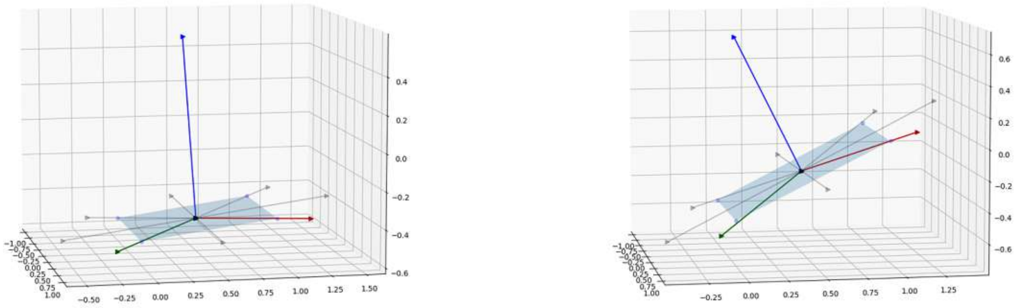
The deformation of the cube is exaggerated, which makes a qualitative assessment of the element local coordinate system possible.



(a) $t = 0.2\,\mathrm{s}$

(b) $t = 1\,\mathrm{s}$

Figure 29: Total deformation of the cube under shear at different time steps



(a) $t = 0.2\,\mathrm{s}$

(b) $t = 1\,\mathrm{s}$

Figure 30: The element local coordinate system at different time steps

Figure 29 shows the total shear deformation of the cube at different time steps. Figure 30 shows the corresponding element local coordinate system as defined by the fatigue tool. The coordinate system is redefined at each time step, essentially moving with the element face. The test has validated the correct import of transient data, the correct definition and re-definition of the local coordinate system and the correct definition of candidate planes.

Load case 2: Single-step uniaxial compression

The aim of this test case is to compare the values of maximum shear to analytical solution as well as to the results of the FE analysis. In the first analysis an arbitrary pressure of 400 MPa is applied.

Table 5: Stress values in a deformed cube

Plane	$\theta[°]$	τ_{max}
P_0	0	199.999 MPa
P_1	45	0.520 MPa
P_2	90	199.999 MPa
P_3	135	0.520 MPa
P_4	180	199.999 MPa
P_5	225	0.520 MPa
P_6	270	199.999 MPa
P_7	315	0.520 MPa

Table 5 shows the magnitude of the shear stress vector in each candidate plane, as defined in Fig. 20. The values for planes lying at 45° to the loading direction correspond almost perfectly to the analytical solution of 200 MPa. The slight deviation can be traced back to the re-definition of the element local coordinate system on the deformed element face. The planes in question therefore no longer lie at exactly 45° to the loading direction.

Table 6: Stress values in an undeformed cube

Plane	$\theta[°]$	τ_{max}
P_0	0	200.000 MPa
P_1	45	1.706×10^{-14} MPa
P_2	90	200.000 MPa
P_3	135	1.706×10^{-14} MPa
P_4	180	200.000 MPa
P_5	225	1.706×10^{-14} MPa
P_6	270	200.000 MPa
P_7	315	1.706×10^{-14} MPa

To this effect the analysis is repeated, this time skipping the pre-processing of text files. Essentially, the cube remains undeformed while under stress. This is not physical, but serves as a showcase. Table 6 again shows the magnitude of the shear stress vector in each candidate plane. The values correspond perfectly to the analytical solution for planes lying at 45° to the loading direction, while other values reflect precision limits and can be seen as zero. The test case confirms the correctness of stress values computed by the fatigue tool.

In the second analysis a pressure of 882 MPa is applied, which correspond to a data point of the stress-life curve. Therefore, no interpolation of the stress-life curve is needed. The aim of this test case is to simulate a full loading cycle, compute the resulting damage and apply a mean stress correction..

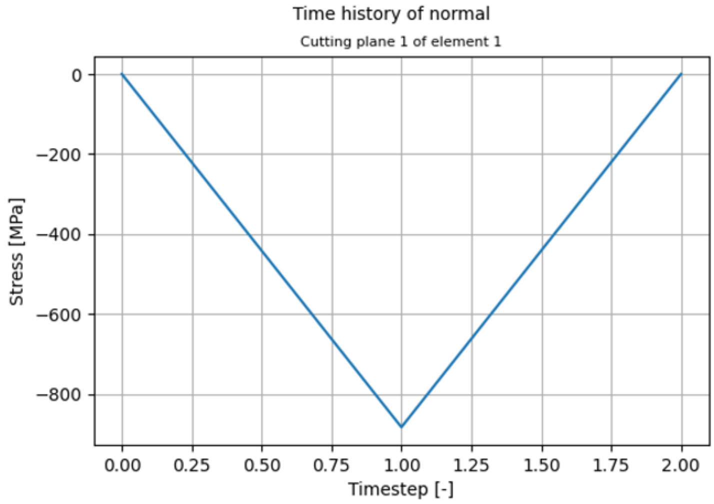


Figure 31: Time history of the fatigue damage parameter $\sigma_{\text{ev}}(t)$ for the candidate plane P_1

Figure 31 shows the time history of the chosen fatigue damage parameter (in this case the normal stress). The profile reflects a loading and unloading of the cube, corresponding to a full loading cycle. The fatigue damage after one cycle corresponds well to values computed by the fatigue tool available in Ansys Mechanical. The values show good correlation without mean stress correction, as well as with Goodman correction (see Eq. 9).

The analysis is repeated for three different linear damage theories presented in Sec. 2.2.6 and assessed qualitatively. The original and modified Miner linear damage rules assign damage to low stress events, or analogously to planes that may not experience maximum normal stress.

The analysis is repeated for a third time with a pressure value of 703 MPa, which requires an interpolation of the stress-life curve to compute the resulting damage. The implementation of the log-log interpolation is shown to be correct and corresponds well to results of the Ansys fatigue tool.

Load case 3: Multi-step uniaxial compression and tension

The aim of this test case is to validate the multi-step functionality, correctness of the sign and of the cycle counting algorithm.

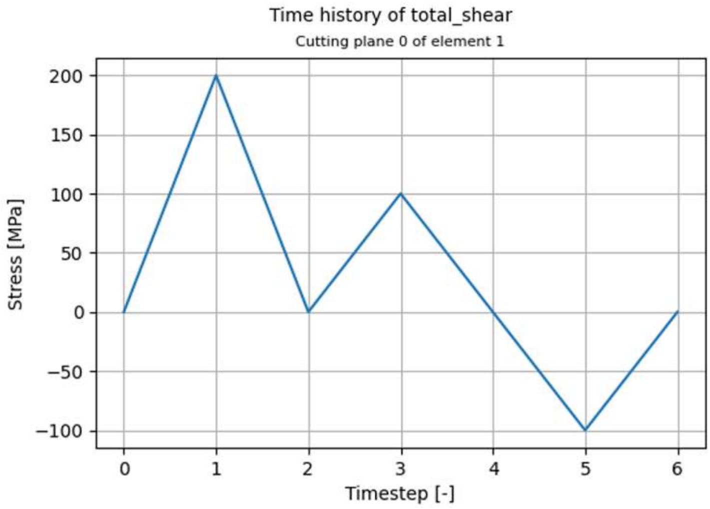


Figure 32: Time history of the fatigue damage parameter $\sigma_{\text{ev}}(t)$ for the candidate plane P_0

Figure 32 shows the resulting time history of shear stress vector magnitude for candidate plane P_0, lying at 45° to the loading direction. The profile corresponds to the loading profile depicted in Fig. 21, whereas the absolute values are multiplied by a factor of $\frac{1}{2}$, which can be explained easily by studying the analytical solution. The change of sign is also considered by the equation used for computing the fatigue damage parameter, indicating that the damaging effects of a change of direction of the shear vector are properly considered.

Table 7: Cycles extracted from the profile in Fig. 32

Range [MPa]	Mean [MPa]	Duration [cycles]
200	100	0.5
100	50	1
300	50	0.5
100	-50	0.5

Table 7 list the loading cycles extracted from the time history of the shear vector magnitude depicted in Fig. 32. The extracted cycles are correct. As an additional validation step the analysis was redone with an example loading profile seen in [2] and the extracted cycles were compared to literature data, validating the correctness of the rainflow cycle counting technique.

Load case 4: Multi-step uniaxial tension and hydrostatic compression

This aim of this test case is to validate the 3D stress state functionality, by using a relatively simple candidate plane configuration (see Fig. 23). The absolute values are comparable to analytic solutions, however a qualitative assessment shall suffice for such a simple case.

Table 8: Normal stress in candidate planes
for both load steps

$\phi[°]$	$\theta[°]$	Normal stress [Mpa]	
		Hydrostatic compression	Uniaxial tension
0	0	-100	100
	45	-100	200
	90	-100	100
	135	-100	≈ 0
	180	-100	100
	225	-100	200
	270	-100	100
	315	-100	≈ 0
45	0	-100	50
	45	-100	100
	90	-100	50
	135	-100	≈ 0
	180	-100	50
	225	-100	100
	270	-100	50
	315	-100	≈ 0
90	0	-100	≈ 0
	45	-100	≈ 0
	90	-100	≈ 0
	135	-100	≈ 0
	180	-100	≈ 0
	225	-100	≈ 0
	270	-100	≈ 0
	315	-100	≈ 0

Table 8 summarizes the normal stress values in all 24 candidate planes. The plane configuration can be seen as three sets of 8 planes, each corresponding to one value of the angle ϕ. In the case of hydrostatic compression all planes experience exactly the same normal stress, corresponding to the uniform pressure applied to all faces of the cube. This serves as basic validation of the 3D stress state functionality. It is further validated with the uniaxial tension load step, where a gradual decrease of all values is seen with increasing angle ϕ. All planes with $\phi = 90°$ experience zero normal stress under uniaxial tension. This is correct as all these planes coincide and lie perpendicular to the loading direction, resulting in zero normal stress.

5.2 Cantilever beam with a square cross-section

This test case proves to be very important for testing the exclusion of incompatible elements before beginning the computation. The values of maximum shear stress, normal stress and von Mises equivalent stress are compared and found to be in excellent agreement with results of Ansys Mechanical, with the usual deviation in the range of a few percent. More deviation is seen due to averaging of stress values over all nodes of the element to provide a single element stress value, however, that is only visible in elements with larger stress gradients within. The engineer should consider avoiding large stress gradient in elements as part of the general guideline in creating FE models. It validates the correctness of stress values for components with more than one element, be it either tetrahedron or hexahedron elements.

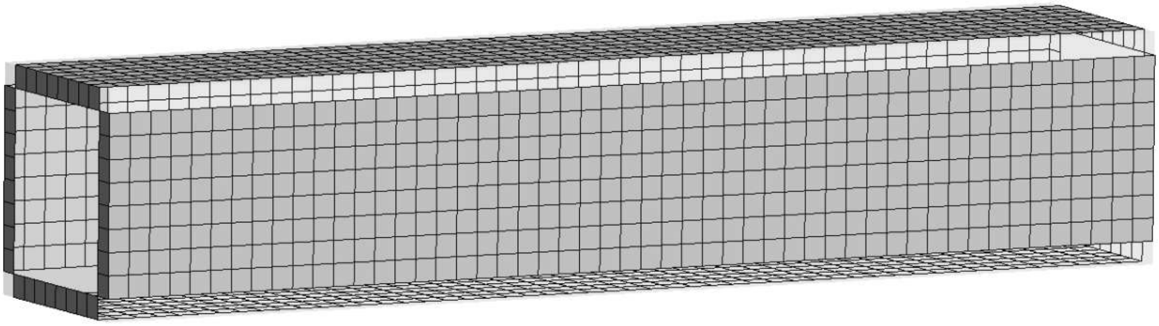


Figure 33: Outer faces of the hexahedron mesh after the exclusion of incompatible elements

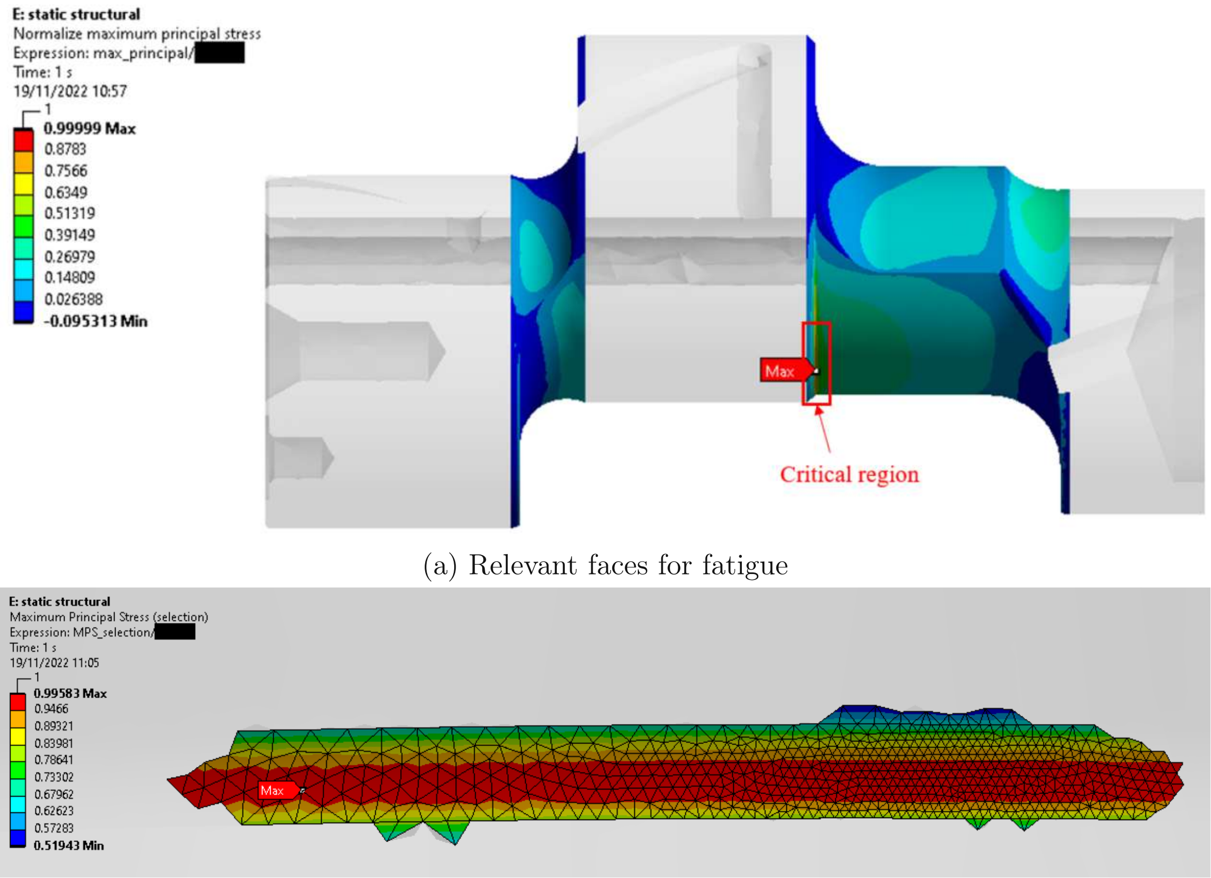
Figure 33 shows the mesh of the cantilever beam after the exclusion of incompatible elements. The elements in all four corners of the square cross-section must be excluded from the evaluation, due to having two outer faces. Not excluding them would result in a single `Element` object having too many nodes assigned, which would inevitably cause a failure during the process of creating element local coordinate systems. When fatigue results at these locations are required, the user is prompted to export two separate results sets for the two adjacent faces, thereby avoiding the problem all-together.

5.3 Cantilever beam with a circular cross-section

The geometry used for this test case has no geometrical edges on the mantle, therefore no elements need to be excluded from the computation. All defined load cases have additionally validated the correctness of stress values computed by the fatigue tool. An additional study, utilising hexahedron elements with a high aspect ratio, revealed no known limitation of the tool in this regard.

5.4 Compressor crankshaft

The aim of this test case is to demonstrate the functionality of the developed tool on a more involved model and assess the result of the fatigue evaluation. First a stress analysis is performed in Ansys Mechanical 2022, the critical location is assumed, based on stress values and previous experience, and nodal stress data is exported for the selected region.



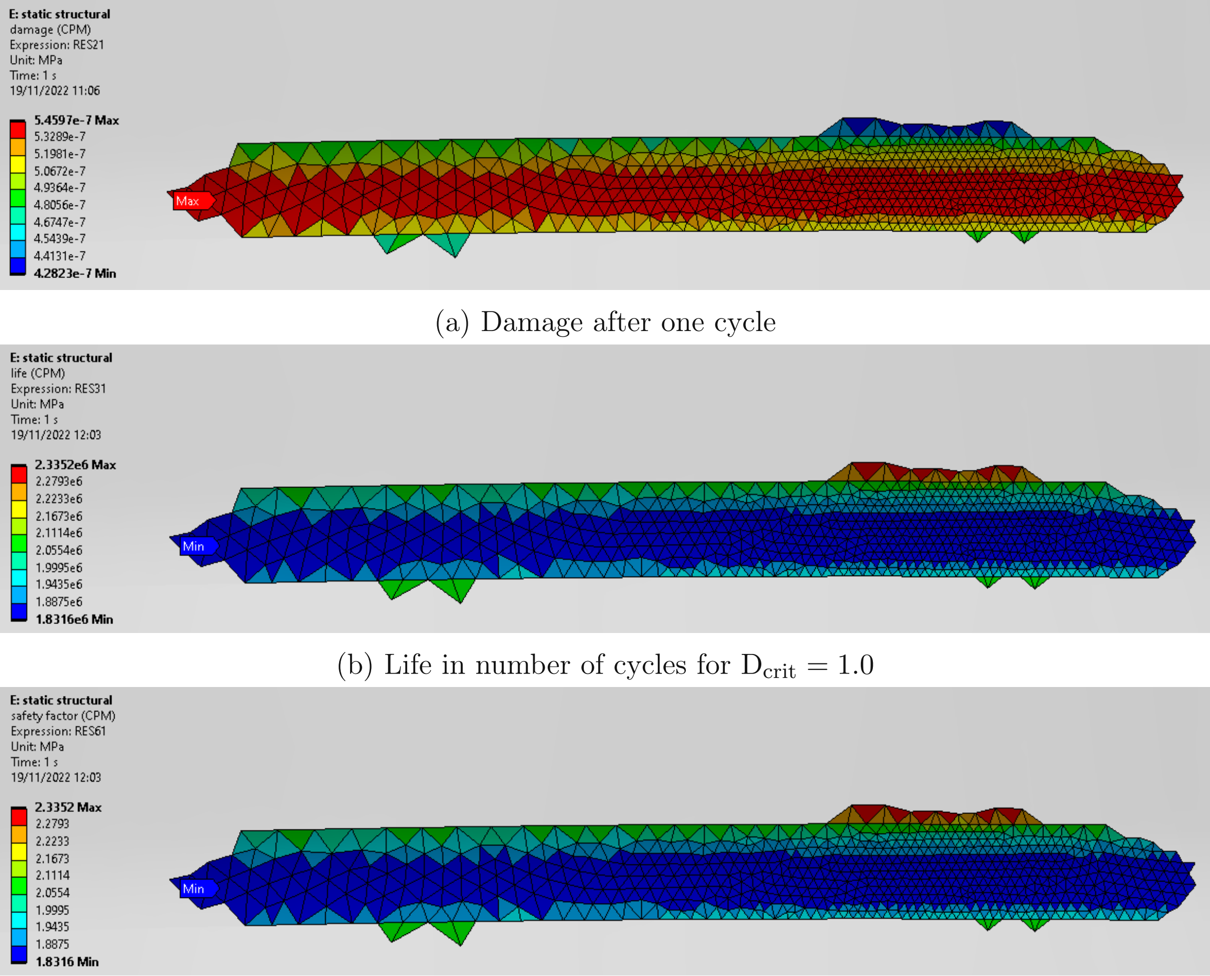
(a) Relevant faces for fatigue

(b) The region around the critical location to be evaluated for fatigue

Figure 34: Normalized maximum principal stress shown as post-processing result in Ansys Mechanical 2022

The post-processing result of the stress analysis are presented in Fig. 34. Due to the cast iron material of the crankshaft the critical location is chosen based on the values of maximum principal stress. Figure 34a shows the contour of maximum principal stress on all the faces relevant for fatigue evaluation, while Fig. 34b shows only an element face selection containing the critical location. The values are normalized so as to protect intellectual property. The nodal stress data is exported only for the selection shown in Fig. 34b.

After the export of the nodal stress data from Ansys Mechanical 2022, fatigue evaluation using the developed fatigue tool can be performed. For a better user experience the results are formatted and exported as a text file, which can be used to visualize the result in Ansys, using a plug-in provided by Ansys.



(a) Damage after one cycle

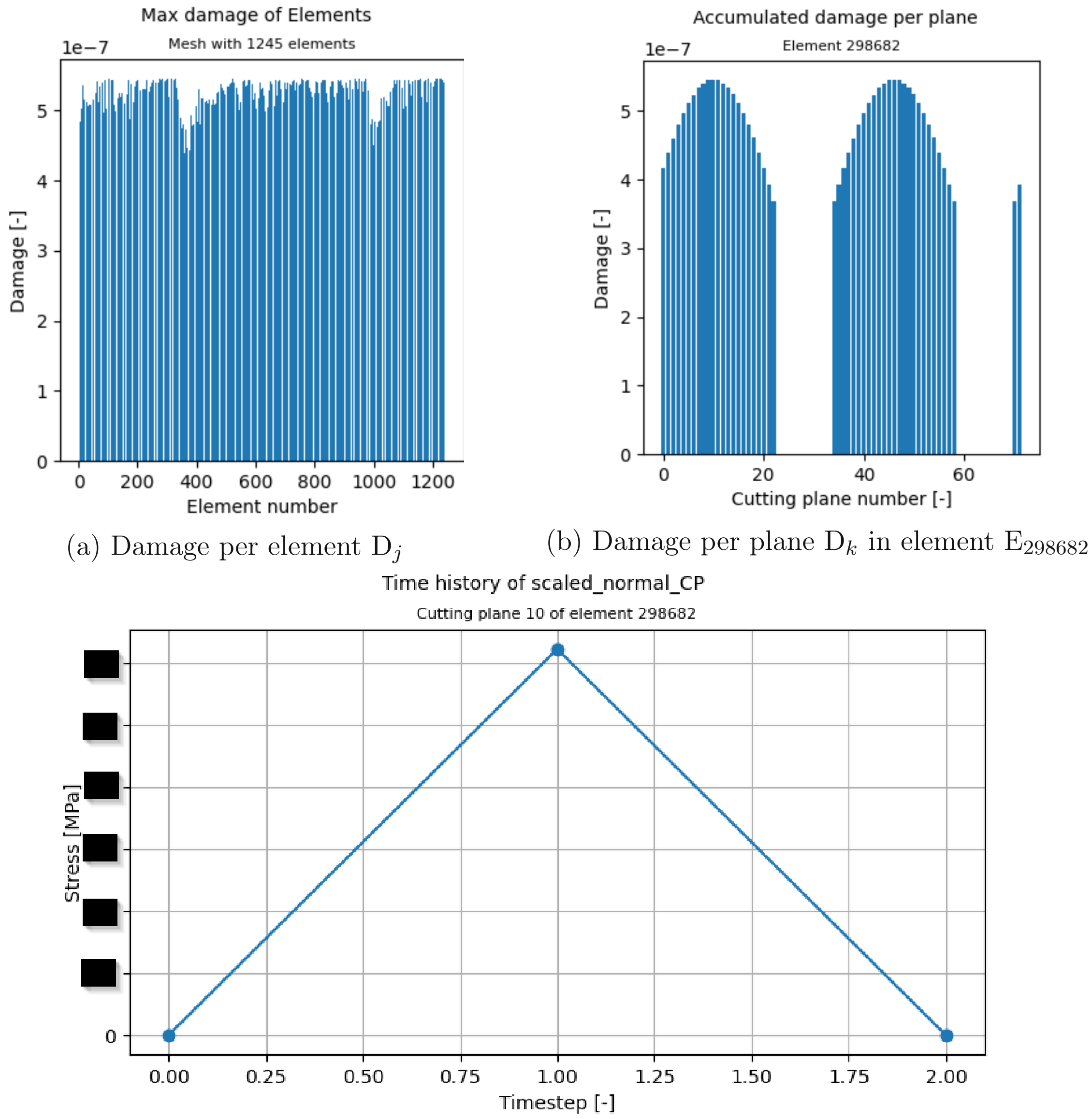
(b) Life in number of cycles for $D_{crit} = 1.0$

(c) Safety factor for a design life of $N_{inf} = 10^6$ cycles

Figure 35: Results of the fatigue evaluation

Figure 35 shows a summary of standard result available after the completion of the fatigue evaluation. The damage after one cycle depicted in Fig. 35a is the primary result, while others are derived results (see Sec. 4.4). The result of most interest to the user is typically the safety factor visualised in the form of a contour plot in Fig. 35c. This preliminary fatigue analysis concludes the crankshaft is safe against fatigue failure, with a minimum safety factor of 1.836. This result is compared to a fatigue evaluation method based on the nominal stress approach. The nominal approach method yielded a minimum safety factor of 3.09, which is significantly higher that the fatigue tool. Primarily the deviation can be traced back to the method used and secondarily to material data, where surface treatment was not considered properly. No further research was done into this topic, as material characterisation is beyond the scope of this topic and is case-specific. However, it can be concluded the difference is small enough to be bridged by correctly adjusting material data for surface treatment.

In order to demonstrate the use of the developed fatigue tool, the output is studied in more detail. The graphical user interface displays the element number of the critical element in terms of fatigue and the corresponding critical plane number.



(a) Damage per element D_j

(b) Damage per plane D_k in element E_{298682}

(c) Time history of the scaled normal stress in plane P_{10} of element E_{298682}

Figure 36: Advanced results of the fatigue evaluation

The first step in closer examination of fatigue results is Fig. 36a, showing a distribution of accumulated damage over all elements in the model. Note that only elements defined in the element face selection shown in Fig. 34b are considered. The element with the highest accumulated damage D_j is chosen for closer examination, in this case E_{298682}. Figure 36b shows the candidate plane damage D_k within the element E_{298682}. From all candidate plane the critical plane P_{10} is chosen for further examination. Figure 36c shows the time history of the chosen fatigue damage parameter for plane P_{10}. In this preliminary fatigue evaluation the crankshaft is statically loaded and unloaded, resulting in this profile.

5.5 Compressor suspension spring

The aim of this test case is to demonstrate the most advanced use of the fatigue tool and the initial motivator for development. A fully transient, random loading sequence is considered, consisting of many varying cycles. First a detailed stress analysis is performed in Ansys Mechanical 2022.

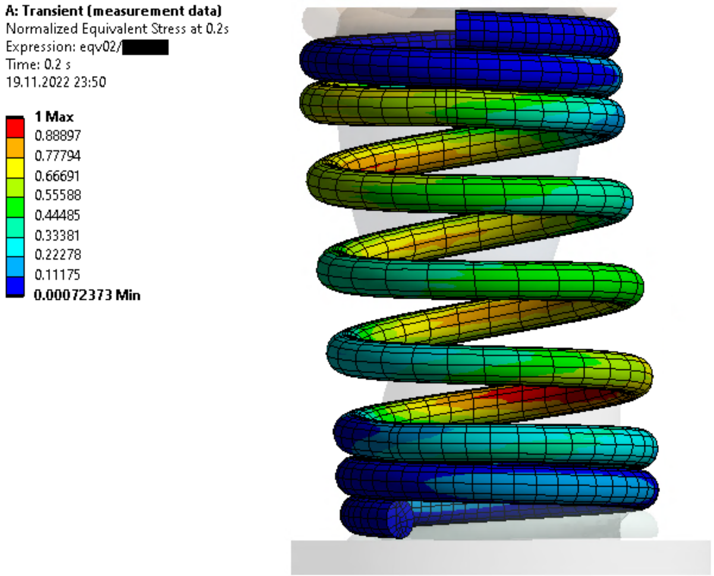


Figure 37: Normalized Equivalent stress displayed on the element faces selected for fatigue evaluation

Figure 37 shows the normalized von Mises equivalent stress at an arbitrary point in time. The elements displayed also correspond to the selection of element faces for which nodal stress data is exported. The data is exported from Ansys using a Python script, where the number of time steps is defined, corresponding to the number of text files exported. For the present analysis with a duration of 0.5 s 50 text files are exported.

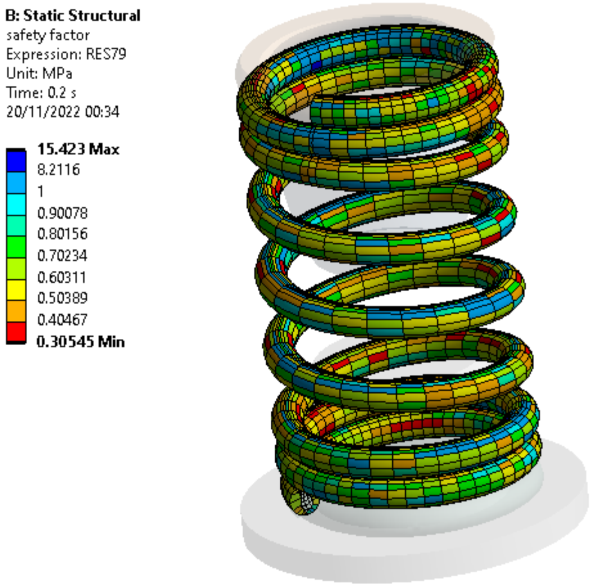
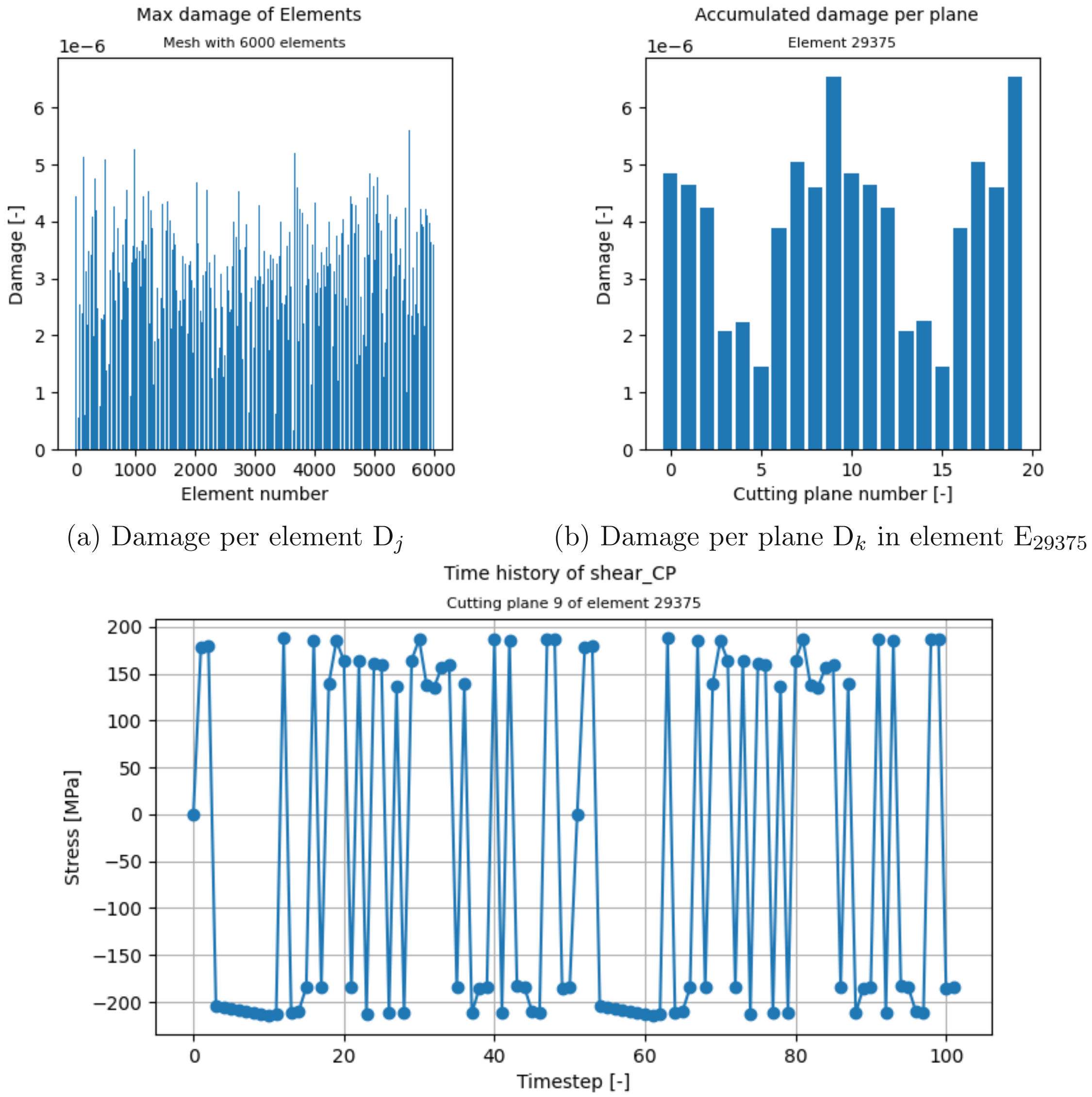


Figure 38: Safety factor displayed on the element faces selected for fatigue evaluation

Figure 38 visualizes the safety factor as a result of the fatigue evaluation. The present fatigue evaluation shows a fatigue failure of the spring before the passing criterion of 500, 000 starting or stopping sequences. Further examination of material properties and the consideration of shot-peening in the synthetic stress-life curve is needed for a more accurate evaluation, however this is beyond the scope of this thesis.



(a) Damage per element D_j

(b) Damage per plane D_k in element E_{29375}

(c) Time history of the scaled normal stress in plane P_9 of element E_{29375}

Figure 39: Advanced results of the fatigue evaluation

Figure 39a shows a relatively high fluctuation of element damage D_j, in comparison to the compressor crankshaft test case (see Fig. 36a). This can be attributed to the larger number of elements considered for the fatigue evaluation and an inherent difference in stress for elements on the inside or on the outside of the coils (see Fig. 37). The fatigue tool concluded that E_{29375} is critical, therefore we examine the individual plane damage D_k shown in Fig. 39b. All planes accumulate some damage, which can be attributed to multiple changes of principal axes, caused by non-proportional, random loading (see Sec. 2.2.2). For demonstration purposes the time history of the scaled normal stress in the critical plane P_9 is shown in Fig. 39c. The time history is random and the extraction of cycles is a complicated procedure, only feasible with the use of numerical algorithms.

It is up to the user to assess whether enough sampling points were used to adequately describe the time history. In Fig. 28 the sampling rate is overlaid over the displacement profile, leading to conclusion that that a high enough sampling rate is used.

This test case shows the degree of complexity involved in properly configuring a fully transient fatigue evaluation with random loading. The user must have sound existing knowledge, best coming from prior experience, of fatigue failures in similar components.

5.6 Discussion

The testing and validation procedure described in detail in Sec. 4.5 has proven to be vital in discovering and fixing unexpected bugs in the implementation. The mistakes ranged from false concepts, overlooked limitations and Python programming language specifics. The procedure has seen many iterations until all issues were fixed, being either minor or capital mistakes.

The simple cube example proves to be the most important and illustrative for basic functionalities of the code. The function defining the element local coordinate system is conceptualized on this test case, first for a hexahedron mesh and later on for a tetrahedron one. The function defining the temporarily stored candidate plane normal vectors is tested for both a plane stress and a 3D stress state on the same cube. The very simple candidate plane configuration on a square element face enables a straightforward comparison with analytical solutions, forming the basis for stress value validation. Additionally, it is used for the validation of damage calculation and the implemented mean stress correction. As the stress values are proven to be correct, more elaborate loading profiles are applied to test the rainflow cycle counting algorithm.

Of course not all implementations that work with a single element can be scaled for use with thousands of elements over multiple time steps. To this effect progressively more elaborate test cases were defined. The cantilever beam test cases served the purpose of properly configuring the exclusion of incompatible elements and comparing the computed stress values with the reference values available in the stress analysis performed in Ansys Mechanical 2022. The values were found to be in excellent agreement, despite the inherent deviation resulting from averaging the nodal stress values to produce a single element value. This inevitably relaxes some stress gradient, especially in critical locations. It is up to the engineer to refine the mesh where needed to avoid loss of critical information through averaging. These test cases proved the scalability of the implementation, thereby paving the path for even more elaborate test cases described in Sec. 4.6. The aim of the compressor crankshaft and suspension spring cases was not to validate stress or damage values, but rather to examine the user experience when using the developed fatigue tool. There is no reference solution for direct comparison, therefore the results were assessed qualitatively. The crankshaft example shows a relatively simple fatigue evaluation, where a simplified, single step, static stress analysis is used as the basis. Based on the stress distribution (the relevant equivalent stress is chosen by the user) a critical location is determined and nodal stress results are exported for only a limited element face selection. Multiple operating conditions of the compressor could be examined, without much rework.

The suspension spring example shows the initially intended use for the developed tool. The suspension spring is subjected to completely random loading, measured during the stopping of the compressor. A detailed transient stress analysis gives little insight into what the critical location in term of fatigue might be. In a simplified fatigue evaluation one would consider the largest deflection of the displacement profile shown in Fig. 28, thereby neglecting the many lower stress cycles present in one stopping sequence. The developed tool considers the full stress history in each plane of each element to determine the critical location in terms of fatigue. The effort for a detailed fatigue evaluation is transferred from the user to the software. Although being fundamentally different in terms of the fatigue evaluation concept, the crankshaft and suspension spring examples require roughly the same amount of set-up time. This shows a definitive advantage over traditional approaches.

6 Conclusion

The present thesis thoroughly describes and documents the development process of a software tool, aimed at providing a platform for accurate fatigue evaluation of different components subjected to cyclic loading. As most development project, it is motivated by an existing problem and the subsequent need for a reliable and efficient solution. Based on the problem description the relevant literature on fatigue of materials is studied. A suitable approach is found in the form of CPM. The method is first studied from a purely theoretical viewpoint and later on the idea of a Python implementation is conceptualized. The concept is presented in detail with flowcharts and written descriptions. The reader is encouraged to study the implementation and understand the working principle of individual functions as well as their inherent limitations.

After a thorough study the proposed concept is implemented in Python, providing the user with a graphical user interface (GUI). An extensive testing and validation procedure is set up to examine the code for bugs and mistakes in the concept or implementation. The procedure is progressive, first testing the implementation on simplified, unphysical examples, providing valuable insight into the most primitive of functionalities before moving on to more elaborate test cases. The procedure is repeated until all known issues are resolved. With a tested and validated code two real-life use-case are presented, demonstrating some of the intended application of the developed tool.

The initial goal is achieved and a usable tool is developed, however, the author acknowledges the almost unlimited room for improvement and expansion. Hopefully, frequent use of the tool will provided the developers with much needed feedback on bugs, wanted features and overall user experience.

6.1 Outlook and improvements

The process of developing a software tool is lengthy and comprised of many steps. Naturally, with the experience gained during the course of this thesis some potential for improvement is recognised. The following sections groups these potential improvements into the relevant categories.

6.1.1 GUI

The GUI available to the user now is a result of an incremental creation process, with no clear idea of user experience. Above all, in some cases there is no distinction between output meant for developers/troubleshooting and the one actually relevant to the engineer. Currently a material library is defined externally and materials can only be added to it by hard-coding them into the source files. This is not trivial for the inexperienced user and a material creator module is surely a better solution. It should enable the user to create custom materials and save them for future use, without leaving the GUI. Some functionality for saving models with loaded and computed data is implemented, but there is large potential to improve on user friendliness in this area.

6.1.2 Materials

CPM assumes a stress-life curve is available for the material to be evaluated, which is often not the case, especially in simplified or preliminary fatigue evaluations. To this effect an auxiliary module for synthesizing stress-life curves is created. It is created due to lacking material data and proven to be useful. A further development of this module shows high potential for expanding the functionality of the tool. It can be included in the GUI as a part of a larger material creation/storage module. Adding functionality to consider surface treatment and other influence factors would greatly improve the accuracy and enable calibration with experimental results.

6.1.3 Fatigue

The present tool contains only a few of the mean stress corrections methods, linear damages theories and fatigue damage parameters available in literature. Due to the modular structure of the code additional options can be added at any time and tested with the same testing and validation procedure.

6.1.4 Python implementation

An object-oriented programming language was chosen for development. Acknowledging the comparatively inefficient computation, Python was chosen due to abundant community knowledge and user friendliness. Additionally, many engineers already have prior experience with Python, making a further development of the tool far more likely and tailored to specific engineering needs. There is unlimited potential in optimisation of the code, through use of more computationally and memory efficient objects and methods. A further examination into the data structure is advisable to find even more potential for improvement. One of the more time-consuming steps in using the fatigue tool is the loading of input text files, where parsing is done line-by-line and is most likely very inefficient.

6.1.5 Project

The author has made continuous effort to keep the organisation of the code and all auxiliary files at a respectable level. Additionally to being an academic work, the present thesis also serves as a manual and reference for future users. A rather limited batch of examples was created and is available. The project makes use of version control via Git, enabling users to clone the repository, implement additional functions and commit the changes as a separate branch, pending approval. Included in the repository are also the examples.

References

[1] Heinz P. Bloch and John J. Hoefner. *Reciprocating compressors: operation & maintenance.* Gulf Publishing Company, Houston, TX, 1996.

[2] Yung-Li Lee, Mark E. Barkey, and Hong-Tae Kang. *Metal Fatigue Analysis Handbook: practical problem-Solving Techniques for Computer-Aided Engineering.* Butterworth-Heinemann, Waltham, MA, 2012.

[3] Kumarswamy Karpanan. Critical plane search method for biaxial and multiaxial fatigue analysis. *Proceeding of the ASME 2016 Pressure Vessels Piping Conference*, 2016.

[4] Rinaldo Puff, Marcos Giovani Dropa de Bortoli, and Raul Bosco Jr. Fatigue analysis of helical suspension springs for reciprocating compressors. *International Compressor Engineering Conference*, (Paper 1989), 2010.

[5] M. A. Zaccone. Failure analysis of helical suspension springs under compressor start/stop conditions. *ASM International*, 1(3):51–62, 2001.

[6] Richard G. Budynas and J. Keith Nisbett. *Shigley's mechanical Engineering Design.* McGraw-Hill Education, New York, 10 edition, 2015.

[7] E. Haibach. *Betriebsfestigkeit.* Springer-Verlag, Berlin, Germany, 2006.

[8] Raymond Browell and Al Hancq. *Calculating and Displaying Fatigue Results.* ANSYS, Inc., 2006.

[9] J. Goodman. *Mechanics applied to engineering.* Green & Company, London, 1914.

[10] K. N. Smith, P. Watson, and T. H. Topper. A stress-strain function for the fatigue of metals. *Journal of Materials*, 5, 1970.

[11] N. E. Dowling, C. A. Calhoun, and A. Arcari. Mean stress effects in stress-life fatigue and the walker equation. *Fatigue and Fracture of Engineering Materials and Structures*, 2009.

[12] M Matsuishi and T. Endo. Fatigue of metals subjected to varying stress. Fukuoka, Japan, 1968. Japan Society of Mechanical Engineers.

[13] D. Taylor. *The Theory of Critical Distances.* Elsevier Science, London, 2007.

[14] W. N. Findley. A theory for the effect of mean stress on fatigue of metals under combined torsion and axial load or bending. *Journal of Engineering for Industry*, pages 301–305, 1959.

[15] Jinsoo Park and Drew Nelson. Evaluation of an energy-based approach and a critical plane approach for predicting constant amplitude multiaxial fatigue life. *International Journal of Fatigue*, 22(1):23–39, 2000.

[16] M. W. Brown and K. J. Miller. A theory for fatigue failure under multiaxial stress-strain conditions. *Proceedings of the Institution of Mechanical Engineers*, 187(1):745–755, 1973.

[17] Dariusz Skibicki. *Phenomena and Computational Models of Non-Proportional Fatigue of Materials*, volume 18. 05 2014.

[18] *FEMFAT Max user manual.* MAGNA, 2015.