

# POLITECNICO DI TORINO

---

Dipartimento di Ingegneria Meccanica ed Aerospaziale

Corso di Laurea Magistrale  
in Ingegneria Aerospaziale



*Tesi di Laurea Magistrale*

Development of calculation tool  
to study multiaxial fatigue behaviour  
based on FEA evaluation.

*Relatore:*

PROF. ING.  
MARCO GHERLONE

*Referente Aziendale:*

ING.  
DAVIDE CUFFIANI

*Laureando:*

ANDREA  
CASAGRANDE

---

Anno Accademico 2019-2020



# Contents

<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>1 Mechanical Fatigue</b>	<b>15</b>
1.1 General overview . . . . .	15
1.2 Fatigue origins . . . . .	15
1.3 Single-axial stress state . . . . .	18
1.4 Multiaxial stress state . . . . .	20
<b>2 Fatigue life evaluation</b>	<b>23</b>
2.1 Introduction . . . . .	23
2.2 Uniaxial stresses . . . . .	23
2.2.1 Palmgren-Miner rule . . . . .	23
2.2.2 Peak counting method . . . . .	25
2.3 Multi axial stresses . . . . .	27
2.3.1 Principal stresses approach . . . . .	29
2.3.2 Critical plane approach . . . . .	31
2.3.3 Mesoscopic scale approach . . . . .	34
<b>3 Solver development</b>	<b>37</b>
3.1 Introduction . . . . .	37
3.2 Application . . . . .	37
3.3 Matake's theory . . . . .	39
3.3.1 Discretization . . . . .	39
3.4 Solution . . . . .	40
3.5 Mesoscopic Approach . . . . .	44
3.5.1 Discretization . . . . .	44
3.5.2 Solution . . . . .	44
3.6 Principal stress theory . . . . .	48
3.6.1 Own principal directions . . . . .	48
3.6.2 Dynamic tensor principal direction . . . . .	50
3.6.3 Goodman correction . . . . .	51
3.6.4 Calculus time-analysis . . . . .	52
3.7 Analytical development . . . . .	54
3.7.1 Remarks . . . . .	56

<b>4</b>	<b>Graphical interface development</b>	<b>59</b>
4.1	Introduction . . . . .	59
4.2	Problem first approach . . . . .	59
4.3	Sensitivity matrices . . . . .	60
4.3.1	The graphical interface . . . . .	62
4.3.2	Pre-processing activities . . . . .	62
4.3.3	Calculus . . . . .	65
4.4	Discrete step solution . . . . .	66
4.4.1	Stress tensor . . . . .	66
4.4.2	The graphical interface . . . . .	67
4.5	Damage tolerance analysis . . . . .	68
4.5.1	Basic theory . . . . .	68
4.5.2	Numerical implementation . . . . .	71
<b>5</b>	<b>Results</b>	<b>75</b>
5.1	Introduction . . . . .	75
5.2	Loads system . . . . .	75
5.3	Sensitivity matrices analysis . . . . .	77
5.4	Stress tensor analysis . . . . .	82
5.5	Comparison with other theories . . . . .	89
5.5.1	Matake's theory results . . . . .	89
5.5.2	Mesoscopic approach results . . . . .	92
5.6	Damage tolerance results . . . . .	93
5.7	Fatigue curves and experimental comparison . . . . .	98
5.8	Time management . . . . .	100
<b>6</b>	<b>Conclusions and final remarks</b>	<b>103</b>
<b>7</b>	<b>Conclusioni</b>	<b>107</b>
<b>A</b>	<b>Matake's theory solver</b>	<b>109</b>
<b>B</b>	<b>Mesoscopic approach solver</b>	<b>113</b>
<b>C</b>	<b>Maximum principal stress solver</b>	<b>115</b>
	<b>Bibliography</b>	<b>119</b>



# List of Figures

1	Stock image of the Leonardo helicopter AW169 [1]. . . . .	7
2	Main rotor rendering of the AW169 helicopter [2]. . . . .	8
3	AW169 main rotor detail [2]. . . . .	9
4	Immagine di repertorio dell'elicottero AW169 [1]. . . . .	11
5	Rendering del rotore principale dell'elicottero AW169 [2]. . . . .	12
6	Dettaglio del rotore dell'AW 169 [2]. . . . .	13
1.1	Ideal schematic diagram of a fatigue load cycle [3]. . . . .	16
1.2	Cross-sections examples with different failures due to several type of loads [3]. . .	17
1.3	Uniaxial load/stress cycle example [3]. . . . .	19
1.4	Flight-test loads [2]. . . . .	21
2.1	Double uniaxial cycle and ideal S-N curve [4]. . . . .	24
2.2	Rain-flow method graph [3]. . . . .	25
2.3	“Range-pair” method scheme [3]. . . . .	26
2.4	“Race track” method scheme [3]. . . . .	26
2.5	Spectrum partition based on “level-crossing” method [3]. . . . .	27
2.6	Graphical scheme of “peak counting” method [3]. . . . .	27
2.7	Elementary plane inside the reference control volume. . . . .	28
2.8	Generic path of $C$ on an elementary plane, with ambiguous evaluation. [5] . . . .	29
2.9	Root of Resolved Shear Stress components [5]. . . . .	30
2.10	Description of a critical plane ad local reference frame. [6] . . . . .	32
2.11	Relation between macroscopic and mesoscopic scales. [7] . . . . .	34
3.1	Stress path on the reference plane $\Delta$ . . . . .	40
3.2	Cauchy stress tensor. . . . .	49
3.3	Goodman-Haigh diagram, with Goodman correction. . . . .	51
3.4	Goodman-Haigh diagram with negative mean stresses and with three corrections. .	52
4.1	Preliminary section flowchart of the GUI. . . . .	62
4.2	Discretization step flowchart. . . . .	63
4.3	Load time history import step flowchart. . . . .	64
4.4	Stresses time history step flowchart. . . . .	65
4.5	Reduced analysis import stage flowchart. . . . .	67
4.6	Reference modes for fracture mechanics. [3] . . . . .	68
4.7	Thickness effect on experimental test to evaluate $Kc$ . [3] . . . . .	70
4.8	Growth curve described by Paris law. [3] . . . . .	70
4.9	Examples of remarkable cracks position. . . . .	71

4.10	Damage control stage flowchart. . . . .	72
5.1	Time history of the loads used in the validation test. . . . .	76
5.2	Sine functions used for the analysis. . . . .	76
5.3	Goodman diagram for sensitivity matrices analysis. . . . .	79
5.4	Critical sets of nodes. . . . .	79
5.5	Critical area located at the pitch horn of the pitch link. . . . .	80
5.6	Critical area located at the edge of the hole, the hole is the seat of the pitch link pin. . . . .	80
5.7	Critical nodes. . . . .	81
5.8	Pitch link pin and pitch lever horn model. . . . .	82
5.9	Critical condition identification with both approaches. . . . .	83
5.10	Comparison between GUIs on the critical area on the pitch link horn. . . . .	85
5.11	Comparison between GUIs on the critical area of the hole which is the slot of the connection pin between the lever and the link. . . . .	85
5.12	Printed Goodman diagram with two steps analysis. . . . .	87
5.13	Stress value of node number 1018016. . . . .	88
5.14	First side with Mataka's results. . . . .	89
5.15	Hole region with Mataka's results. . . . .	90
5.16	Second side with Mataka's results. . . . .	91
5.17	Zoom over the fractured area. . . . .	91
5.18	First side with Mesosocpic's results . . . . .	92
5.19	Second side with Mesosocpic's results . . . . .	93
5.20	Own direction rotation (Method 1) with a surface crack of $0.25\text{ mm}$ . . . . .	95
5.21	Own direction rotation (Method 1) with a surface crack of $0.38\text{ mm}$ . . . . .	95
5.22	Dynamic direction rotation (Method 2) with a corner crack of $0.25\text{ mm}$ . . . . .	96
5.23	Dynamic direction rotation (Method 2) with a corner crack of $0.38\text{ mm}$ . . . . .	96
5.24	Static stress of critical nodes evaluated with "No Growth" control. . . . .	98
5.25	Dynamic stress of critical nodes evaluated with "No Growth" control. . . . .	98
5.26	Wöhler diagram with reference material fatigue curve: experimental and numerical results. . . . .	99
5.27	Picture of the are where the failure occurred. . . . .	100
5.28	Bi-logarithmic graph to compare results of sensitivity matrices solver. . . . .	101
5.29	Bi-logarithmic graph to compare results of reduced step solver. . . . .	102

# List of Tables

3.1	Time comparison between MathCad and MATLAB codes with equal solving structure. . . . .	38
3.2	Global evaluation time for 100000 nodes with the first MATLAB code. . . . .	38
3.3	Matake's theory code improvements. . . . .	42
3.4	Matake's theory final code improvements. . . . .	43
3.5	Mesoscopic's theory code improvements. . . . .	47
3.6	Different Goodman correction approaches. . . . .	52
3.7	Calculus time evolution with principal stresses approach. . . . .	53
3.8	Standard discretization parameters adopted. . . . .	54
4.1	FEM model loads used to evaluate sensitivity matrices. . . . .	60
4.2	Table with size and location combinations for each material. . . . .	71
5.1	Useful sine loads data. . . . .	77
5.2	Reference nodes results with: 35 time steps, sensitivity matrices and first principal stress method. . . . .	77
5.3	Reference nodes results with: 35 time steps, sensitivity matrices and second principal stress method. . . . .	78
5.4	FEM loads used to generate the stresses for the reduced analysis. . . . .	83
5.5	Stress difference between load approaches. . . . .	84
5.6	Numerical results with two-step analysis and own direction rotation. . . . .	86
5.7	Numerical results with two-step analysis and dynamic direction rotation. . . . .	86
5.8	Percentage error between interfaces due to own rotation of tensor. . . . .	87
5.9	Percentage error between interfaces due to dynamic rotation of tensor. . . . .	88
5.10	Matake's theory global results in the reference nodes. . . . .	92
5.11	Mesoscopic's theory global results in the reference nodes. . . . .	93
5.12	Reference material main properties. . . . .	94
5.13	Comparison between experimental and numerical results . . . . .	99
5.14	Parallel environment starting time. . . . .	100
5.15	Tool evaluation time with sensitivity matrices solver. . . . .	101
5.16	Tool evaluation time with reduced step solver. . . . .	101

---

***"Helicopters don't fly, they vibrate  
so badly the ground rejects them."***

---

*Tom Clancy*  
TomClancyQuotes

***"Keep calm and carry on"***

---

*Wiston Churchill*  
British government

***A Mario e Alvisè  
che con l'infalibile lungimiranza dei nonni  
mi hanno accompagnato nei primi passi  
del percorso che oggi culmina.***



# Acknowledgments

There are many people I want to address a special thanks.

Rewinding my memories back, my first thought has to be directed to my family. At home they supported me during the first discouragement period, they encouraged me to carry on. My family has always believed in me and in this goal. Many thanks to mother, father and sister.

A huge thanks you to my company manager **Davide Cuffiani**. Since the first day of my stage in his office, I have felt myself as a small but useful part of the extended world called helicopter. His support and his explanations were enlightening for my technical problems. He played a key role in my master thesis project and I want to renew a thank you.

A thanks to my academic mentor, Professor **Marco Gherlone**. He has always supported my project and the entire thesis since my first proposal, without any hesitation.

Furthermore, I would not have got along without **Fabrizio Ghirardini**'s stinging cheerfulness. I shared with him the work station, the coffee breaks after lunch and more than one story about helicopter world. Thank you for endlessly arousing my curiosity and thank you for pushing me towards the development of my approach and my job over and over again.

In the end, *ultimi ma non per importanza*, my friends, an unlimited source of debate but above all of laughs and amusement. They made this course of studies less gruelling then it really would have been. A special dedication to Andrea Corazza, Giovanni Bortolin first and Elisa Zancanaro later, you are far away from the easy definition of roommate, you closed the distance between me and what I call home. Additionally, Mattia, Paolo, Antonio, Gianluca, Jennifer, Anna, Alessandro, Manuel, Matteo e Davide, I met you as a group the night after my placement exam as a seal of the course which ends today.

I need to renew my gratitude to Elisa and Andrea due to the priceless help they gave me with the writing of this thesis in the language from the other side of the Channel.

Finally, a special thanks to everyone supported me making a faraway milestone to become an imminent success.





# Ringraziamenti

Sono molte le persone alle quali voglio rivolgere un ringraziamento speciale.

Riavvolgendo i ricordi di questo percorso, il primo pensiero deve essere necessariamente a chi, dentro le mura di casa, mi ha supportato nei primi momenti di sconforto e delusione, incoraggiandomi a proseguire per arrivare alla meta; hanno sempre creduto che questo traguardo sarebbe stato tagliato, grazie a Mamma, Papà e Martina.

Un enorme ringraziamento al mio responsabile aziendale **Davide Cuffiani** che, fin dal primo giorno della mia presenza in azienda, mi ha fatto sentire piccola ma importante parte di quel vastissimo mondo che è l'elicottero. Il suo aiuto e le sue spiegazioni nelle mie difficoltà tecniche sono state sempre illuminanti nell'evoluzione del progetto che ha portato a questa tesi.

Un ringraziamento al mio relatore, il Professor **Marco Gherlone**, che senza indugiare ha sempre supportato la mia attività di tesi e tirocinio fin dalla mia prima proposta.

Inoltre, la pungente ilarità di **Fabrizio Ghirardini** è un ricordo del quale non farò mai a meno; con lui ho condiviso la postazione di lavoro, i caffè dopo pranzo e più di qualche aneddoto sul mondo dell'elicottero. Grazie per avermi continuamente solleticato la curiosità e spronato a continui miglioramenti al mio approccio e al mio lavoro.

Infine, *last but not least*, gli amici, quella fonte inesauribile di confronto, ma soprattutto di risate e divertimento, necessari a rendere il percorso decisamente meno impervio di quello che sarebbe stato realmente. Una dedica particolare va a Andrea Corazza, Giovanni Bortolin prima ed Elisa Zancanaro poi che, ben lontani da essere dei semplici coinquilini, sono riusciti a ridurre le distanze, facendomi sentire in ogni momento a casa; poi Mattia, Paolo, Antonio, Gianluca, Jennifer, Anna, Alessandro, Manuel, Matteo e Davide, conosciuti in gruppo la sera del test d'ingresso come suggello del nuovo ciclo e diventati ben presto parte integrante delle risate di tutti i giorni.

Per la stesura di questa tesi devo rinnovare il mio ringraziamento ad Elisa ed Andrea che mi hanno supportato nella scrittura nella lingua dei cugini d'oltremania.

Un ringraziamento speciale a chi, anche in modo apparentemente marginale, mi ha supportato facendo sì che un traguardo lontano diventasse successo imminente.



# Abstract

The aim of this thesis is the analysis of different fatigue behaviour theories for mechanical parts; our goal is to find which one better describes the nucleation of cracks in order to estimate components life. The work is placed in *safe-life* fatigue life evaluation, indeed you consider the end of the component life when a crack occurs, well know as “time to crack enucleation”. Conversely, a crack growth equation is not the aim of this work, you do not need to correctly predict the crack growth rate, nevertheless a flaw growth analysis will be studied. The *damage tolerance* control is referred to the *No-Growth* curve, you compare the acting and the limit threshold stresses in order to understand if a crack propagates or not. If it grows the part is considered as critical. Because of extreme complexity of the components that work inside the main rotor of an helicopter, the research has been restricted to the metal components only. I will neglect all the phenomenology of composite materials. Moreover, due to the high number of parts that cooperate inside an articulated rotor, in the first part of the study I abstract the solution from the real parts and I create a free-standing solver. The code will be able to solve any geometry when the correct input is provided.



Figure 1: Stock image of the Leonardo helicopter AW169 [1].

The input of the whole analysis is another detail of our work, indeed, I want to reuse the main results of Finite Element Analysis (FEA) as input for the code. FEA approach entails the discretization of the main body into many smaller elements, if compared with the main body. These smaller elements are defined, in 3D space, by boundary points called nodes. From the FEA code the analyst extracts the stress tensor calculated in these particular locations of the body. As we will see in the following chapters, from these data we are able to extract different

---

information: a priori the origin and the type of stress, a posteriori the critical areas of the body in order to predict failure locations.

We might consider the different nature of the outputs which FEA modelling is able to give us; in fact I will develop two different approaches linked with these different FEA outputs. The same matrix is a FEM code output and at the same time it is an input for the tool. The difference is the point of view, when you are talking about FEM the matrix is an output, on the other hand it is an input. These FEM-outputs are referred to different matrices:

1. **Sensitivity matrices:** a tensor which stores the stress sensitivity inside its components. The sensitivity describes the relationship between applied load and resultant stress in the nodes. Evaluating these matrices for each type of load and linearly scaling them with the applied load, we are able to calculate the real stress in the nodes of the body. If the applied load is a time history thanks to these matrices you can obtain a stress time history for each node applying the following formula in every time step.

$$\sigma_{real} = \sigma_{sens} \cdot \left| \frac{P_{real}}{P_{sens}} \right| \quad (1)$$

2. **Stress tensor:** in this case a static analysis is performed within the FEA code and the analyst extracts from the results the real stress tensor for all nodes. These stresses are related to the loads set in the FEA code, hence a scaling is not necessary and you can directly solve the fatigue problem. In order to obtain a stress time history you need to launch FEM solver more than once. If you want  $n$  steps you have to solve  $n$  times the FEM model with  $n$  load cases. To obtain the same stress time history the whole calculus is more demanding and you waste too much time.

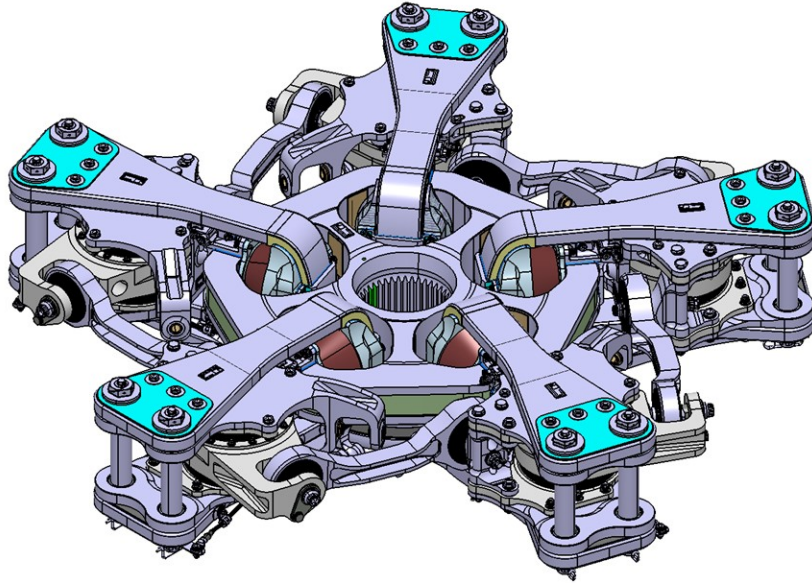


Figure 2: Main rotor rendering of the AW169 helicopter [2].

Once either the first or the second input is known, the code analyses the fatigue behaviour in order to identify the areas with the highest stresses which we consider critical. The first challenge is to write a code which precisely solve the problem in a reasonable time. Time is our first enemy

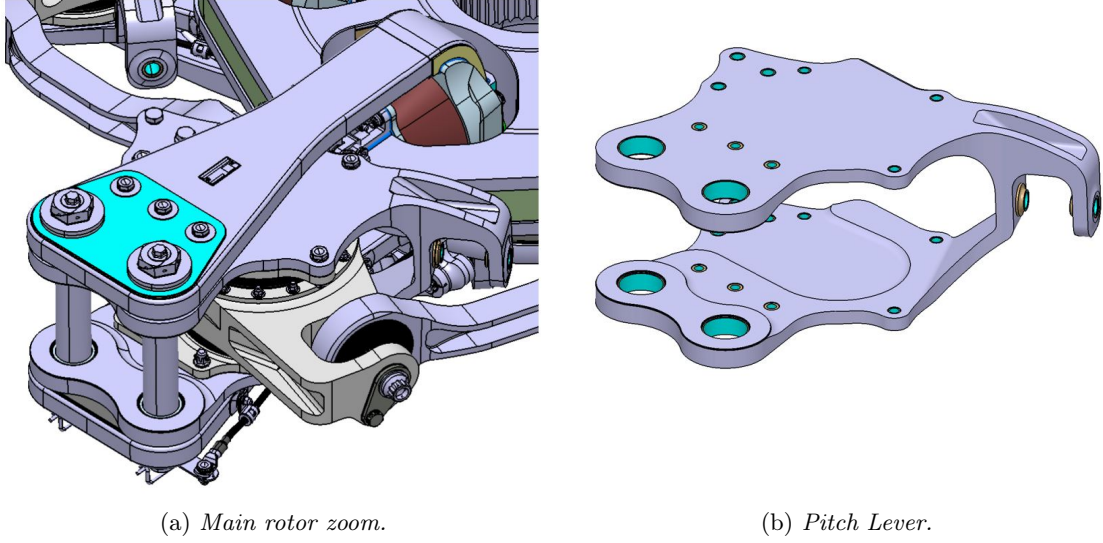


Figure 3: AW169 main rotor detail [2].

because the great sophistication of modern FEM models brings the global number of nodes to around  $10^6$ . We reduce this value to  $10^5$  if we consider only the skin of the body. The number is not negligible and the development of codes will impose massive steps to overcome problems linked with solution time.

The need of the technical office is to have a easy-to-use code, therefore I created a Graphical User Interface which allows setting all the input parameters. It contains all the functions to solve the fatigue problem and to carry out the post processing phase.

The development of resolving functions and the GUI program has been achieved using MATLAB<sup>®</sup> language. Thanks to this program, independent solvers have been developed, one for each criterion; after I have created the functions and finally I put together all the parts inside the GUI. In order to have a clear workflow two separated GUIs were created. The first one is referred to the sensitivity matrices input, the other to the stress tensor.

In the end, we turned to the industrial world and real items of helicopter rotors. We want to find a test part which may be able to provide information about the features of both programs. The aim is to validate not only the calculus made by GUIs but also the entire chain design, composed by FEA model, new calculation tool and laboratory experimental test.

To accomplish this we take as benchmark the AW169 built by Leonardo<sup>®</sup>. This helicopter belongs to medium weight class with its maximum take off weight of 4500 *Kg*. It was born to fill up the market segment between the smaller AW109 and the larger AW139. [1]

Particularly, the main rotor is the most important part of the whole helicopter. We can observe a similar representation in Fig. 2; from the main image we are able to identify our benchmark component, called Pitch Lever and shown in Fig. 3 (b).



# Sommario

Lo scopo principale del lavoro di tesi è stato l'analisi di diverse teorie, nell'ambito del fenomeno della fatica meccanica, per ricercare quale meglio si adattasse al fine di determinare la vita utile del componente meccanico in questione. Il lavoro si colloca nella ricerca della vita utile in termini *safe-life* ossia consideriamo finita la vita utile del componente nell'istante in cui la cricca enuclea. Non risulta di particolare interesse per il nostro primissimo scopo la ricerca di una legge di accrescimento del difetto che modelli correttamente la propagazione del difetto. Il controllo di tipo *damage tolerance* viene limitato alla comprensione dei nodi critici con una dimensione di difetto predefinita. Se il componente ha dai punti che superano il limite imposto è da considerarsi critico. Alla luce dell'estrema complessità dei componenti che costituiscono il rotore principale di un elicottero ci si è limitati allo studio delle componenti realizzate in lega metallica, tralasciando tutta la fenomenologia delle componenti in composito. In virtù dell'elevato numero di componenti metalliche che un rotore articolato presenta si è preferito impostare lo studio astruendo dal singolo componente, bensì creando un solutore indipendente dal componente reale, capace quindi di adattarsi a tutti i componenti metallici noto un input preciso.



Figura 4: Immagine di repertorio dell'elicottero AW169 [1].

L'input di tutta l'analisi è una seconda peculiarità della tesi svolta, infatti, come da prassi nel mondo industriale su ciascun componente vengono svolte diverse analisi mediante la teoria agli Elementi Finiti (FEM). Tale approccio progettuale prevede la discretizzazione in elementi più piccoli del componente finale, tali elementi sono essenzialmente delimitati da punti fisici appartenenti al modello 3D del pezzo, detti nodi. Alla fine di una analisi è dunque possibile avere lo stato di sollecitazione in ciascun nodo del modello, entrando a conoscenza dello stato



tensionale di sollecitazione in quel punto. Come verrà spiegato nei capitoli successivi da tale stato è possibile ricavare delle informazioni: a priori sul tipo di sollecitazione subita dal corpo e a posteriori sulla criticità o meno di certe zone in termini di vita a fatica.

La diversa natura degli input calcolabili tramite la teoria degli elementi finiti porterà a sviluppare due filoni di analisi legati proprio alla diversa genesi dell'input stesso. I due approcci di analisi si basano sul calcolo di due differenti matrici:

1. **Matrici di sensibilità:** ossia un tensore che immagazzina nelle proprie componenti la "sensibilità" del componente a quella precisa componente di carico agente. Valutando queste matrici per ogni componente di carico è possibile, in un secondo momento, riscalarne linearmente tale risultato con il carico reale al fine di trovare la tensione che il carico genera sul componente stesso. SE applicassimo tale approccio ad una time history di carico saremmo dunque capaci di ottenere una time history di stress per ciascun nodo applicando per ogni istante la seguente formulazione:

$$\sigma_{real} = \sigma_{sens} \cdot \left| \frac{P_{real}}{P_{sens}} \right| \quad (2)$$

2. **Tensore di stress:** Si procede con una analisi statica importando direttamente nel solutore agli elementi finiti la storia di carico da analizzare e si ottengono in output le tensioni, già legate al carico imposto e direttamente manipolabili con i criteri di fatica. Con tale approccio possiamo ottenere una time history di stress solamente analizzando più volte il modello FEM e in ciascuna analisi modificando i carichi per ottenere un istante temporale differente. Per equiparare la time history precedente l'operazione risulta molto più onerosa.

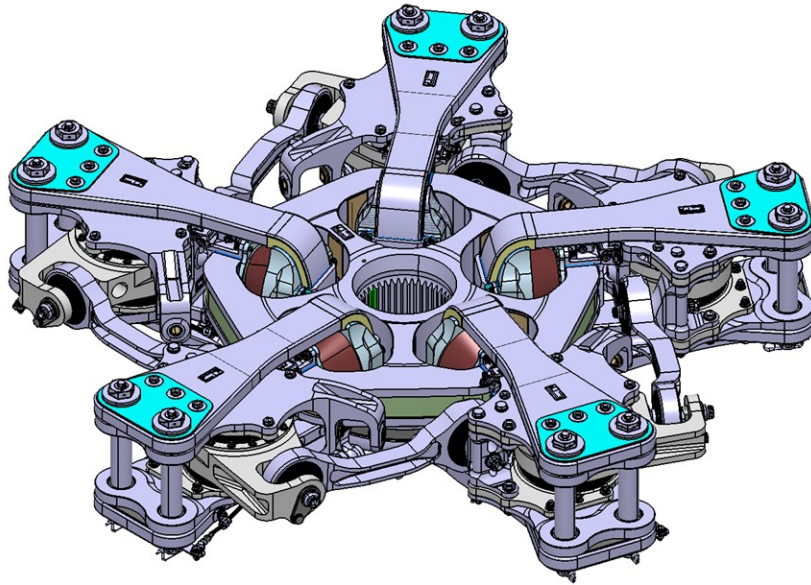


Figura 5: Rendering del rotore principale dell'elicottero AW169 [2].

L'obiettivo è quindi, noto l'uno o l'altro input, di analizzare la criticità o meno del nodo in questione o della zona di corpo maggiormente critica. La prima sfida è quella di ottenere una soluzione accurata in tempi ragionevoli, in quanto l'estrema raffinatezza dei modelli FEM moderni porta il numero di nodi globale a valori dell'ordine di  $10^6$ . Tale valore può essere



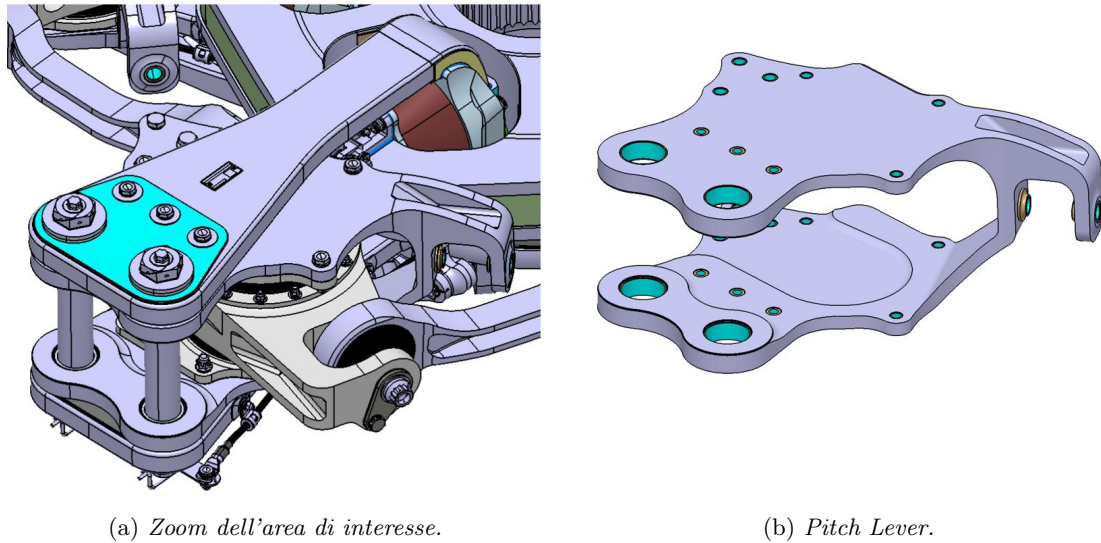


Figura 6: Dettaglio del rotore dell'AW 169 [2].

ridotto di un ordine se si riduce lo studio alla sola pelle del componente; il numero di valutazioni rimane comunque non trascurabile e imporrà degli step di notevole miglioria al codice risolutivo in termini di velocità di calcolo.

Infine, vista la necessità di avere uno strumento fruibile da parte dell'ufficio, presso il quale questo lavoro è stato sviluppato, è stata da me creata una interfaccia grafica che congloba tutti i solutori ottimizzati e permette di settare tutti i parametri necessari alla riuscita del calcolo, sfornando diversi output utili a comprendere le zone maggiormente critiche per la vita a fatica.

L'implementazione dei solutori e dell'interfaccia grafica sono state possibili grazie al linguaggio MATLAB<sup>®</sup>, tramite il quale, in una prima fase sono stati sviluppati dei solutori indipendenti per i differenti criteri, poi sono stati create delle function indipendenti, infine è stato tutto accorpato in una interfaccia grafica, in gergo GUI, prodotto finale del lavoro svolto. Per migliorare il workflow legato all'utilizzo finale le due casistiche sono state separate alla nascita e le interfacce si sviluppano su due binari paralleli ma distinti.

Come step finale, ci si è rivolti al mondo industriale e alla componentistica concreta del rotore dell'elicottero, andando a ricercare dei componenti test con i quali comprendere le peculiarità di funzionamento del codice e valutare le sue performance. Con lo scopo di validare i 3 anelli fondamentali nell'analisi: Modello FEM, tool di calcolo, prova sperimentale.

Per fare ciò ci si è riferiti ad un elicottero della gamma di Leonardo<sup>®</sup>, l'AW169 velivolo di classe media dal peso di circa 4500 Kg, pensato per andare ad colmare il segmento di mercato compreso tra il più piccolo AW109 e il più grande AW139. [1]

Nella fattispecie il rotore dell'elicottero è la componente di maggior rilevanza, ne possiamo apprezzare una rappresentazione verisimile in Fig. 5, dalla quale possiamo identificare la componente di riferimento per il nostro studio, palesata in Fig. 6 (b).



# Chapter 1

## Mechanical Fatigue

### 1.1 General overview

In the following chapter we will discuss a general overview of mechanical fatigue behaviour. The aim is to understand which are the main features of this mechanical phenomenon, which factors influence it and which ones are negligible. Lots of books belong to the bibliography from which we can take a cue and examine in depth the fatigue behaviour. We mention [3] and [4] as two remarkable examples to enhance your knowledge of this branch of engineering.

### 1.2 Fatigue origins

Mechanical fatigue could be defined as follow: the gradual decay of mechanical performances of a part owing to stress cycle repetition. The stress cycle is defined by a huge amount of repetitions with lower stress limit if compared with material yielding limit. As first step the load time history is described as a sine function, which gives us informations about the maximum value of load and the frequency of repetition occurred in the working cycle. We are able to show a shred of a load time history, as shown in Fig. 1.1, where the sine amplitude remains constant in time as the other parameters.

The most relevant parameters are:

- $S_a$ : alternate stress, which represents half of the whole amplitude of the sine function; in the following chapters it will be called dynamic stress.
- $S_m$ : mean stress. As the name suggests, it describes the central value of the function; we will call it static value.
- $S_{max}$ : maximum stress.
- $S_{min}$ : minimum stress.

Fig. 1.1 does not display a particular index called *stress ratio*, described with  $R$  and defined by the following formula:  $R = S_{min}/S_{max}$ . The relevance of this parameter is strongly linked with the multiplicity of data we are able to extract. Briefly summarizing we can extract informations of  $R$  with the following information:

- $R > 1$ : The work cycle is completely compressive, the area is absolutely safe if related with fatigue crack propagation due to negative stress.

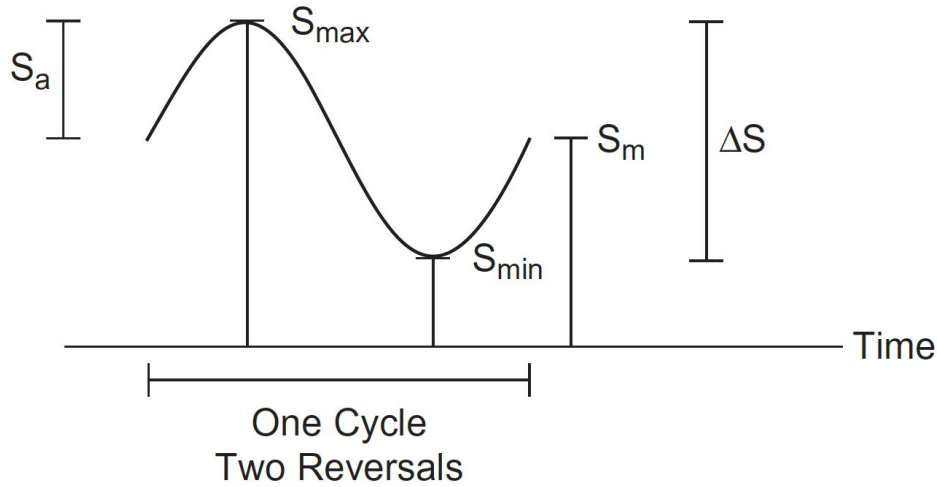


Figure 1.1: Ideal schematic diagram of a fatigue load cycle [3].

- $R = 1$ : Static load condition, nearby 1 the theory commits an error, maximum and minimum value are the same.
- $R \in [0, 1]$ : positive cycle loads, worst circumstance for the part.
- $R = 0$ : positive cycle with minimum value set to zero.
- $R < 0$ : work cycle with a percentage of negative stress, therefore compressive. It helps the fatigue life.
- $R = -1$ : Case with the sine mean value equal to zero. Moreover, it is considered as a benchmark because is the easiest to be recreate in laboratory. Additionally, positive and negative ranges are the same.

When the load cycle is achieved we only need to wait the huge amount of repetitions show their damaging effect over mechanical behaviour, until the structural collapse of the whole part. As introduced in [3], we are able to understand the nature of the load analysing the broken area of the specimen as well. The Fig. 1.2 clarifies this concept with some real examples.

The picture 1.2 merges and intersects three powerful effects for fatigue life of mechanical parts: load cycle mean value, notch factor and type of applied load.

Spontaneously, you can understand that a different load causes a different fatigue behaviour. If we refer only to bending loads, the phenomenology is similar among cases. Let's try to better understand the images: first on thing, the hatching area describes the brittle fracture, contrary the white area shows the growth of the flaw. The arrows reveal the growth directions of cracks through the cross section. You can see the crack nucleation, that is the crack birth, on the skin of the body and it propagates according to critical direction defined by stresses acting over there. Moreover, we detect the presence or not of notch comparing brittle fracture areas. We give also information about loads with beach lines: they show the higher propagation speed of notched specimen than smooth ones. There is a macroscopic difference between low and high mean value. When the mean value reaches the yielding limit there will be needed few cycles to exhibit a failure; on the other hand a mean value far from yielding limit shows a higher cycle fatigue behaviour. We need to remind that a brittle failure occurs when the residual area is no

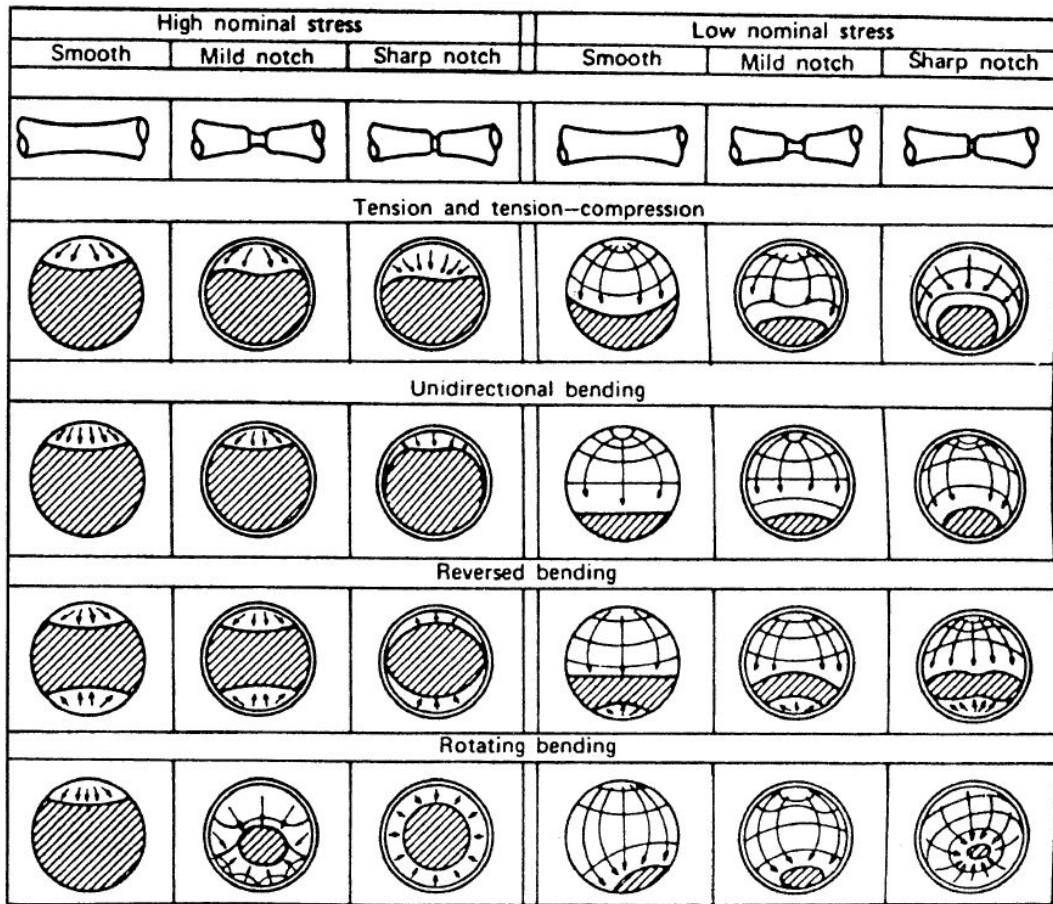


Figure 1.2: Cross-sections examples with different failures due to several type of loads [3].

more able to sustain the applied load. The residual area is the region of the cross section not deteriorated by cracks.

The elements describe in Fig. 1.2 are not the only taking place in fatigue phenomenon; in the following list the most valuable players in fatigue life estimation are described:

- **Material type:** in fatigue life estimation there is the *fatigue limit* concept. Thanks to this idea we want to describe the stress limit under which fatigue does not show its effects, neither nucleation or propagation. It follows that component life will be infinite, obviously talking about fatigue. The steel alloys and titanium alloys show this fatigue limit, yet aluminium alloys not.
- **Grain size:** crystal grain's shape and size have a basic relevance to slow down crack propagation. Small size of grain is the best choice for low temperature applications.
- **Thermal treatment:** is strongly linked with the previous point. Quench, tempering and ageing are three crucial steps to obtain the requested grain size.
- **Geometrical discontinuity:** is referred with notch factors. They describe the sensitivity referred to foot radius, holes, hub shoulder, etc... We can consider as a discontinuity also a welded area. Welding process modifies the internal structure of metal, and in some cases the external geometry is modified as well.
- **Skin roughness:** the less the skin roughness is, the more component life will be longer. With smooth part's skin we discourage crack enucleation.
- **Skin's residual stresses:** If we are able to let a residual compressive stress state all over the body's skin, we will have longer-life components.
- **Corrosion:** the combined effect of corrosion and fatigue is to be taken into account in order to correctly estimate fatigue life. Inside corrosion environment we should insert: moisture interaction, oxidation, brackish air and pollution in general. Above all the fretting is the most demanding, it refers to wear and corrosion due to asperities of contact surface and relative motion between parts.
- **Scale effect:** Analysing bending and torsional fatigue loads, even combined, the region with maximum stresses is larger in bigger specimens then smaller ones. This straightforward observation is validated by experimental tests, but is not a general law; in fact with tension-compression loads there is not scale relationship.

### 1.3 Single-axial stress state

Uniaxial state is the first approach to fatigue problem in mechanical parts. Its development and application are widely popular due to easy evaluation but, above all, to availability of experimental tests. Applied load peculiarity is related to the single application direction, in order to create only one stress component inside the body. This feature simplifies experimental tests. As you can understand, stress is created with an external load varying with sine function. The sine is created to generate stresses you want to study, such as tension-compression, bending or torsion. In Fig. 1.3 you can see a standard representation of an uniaxial fatigue cycle, described

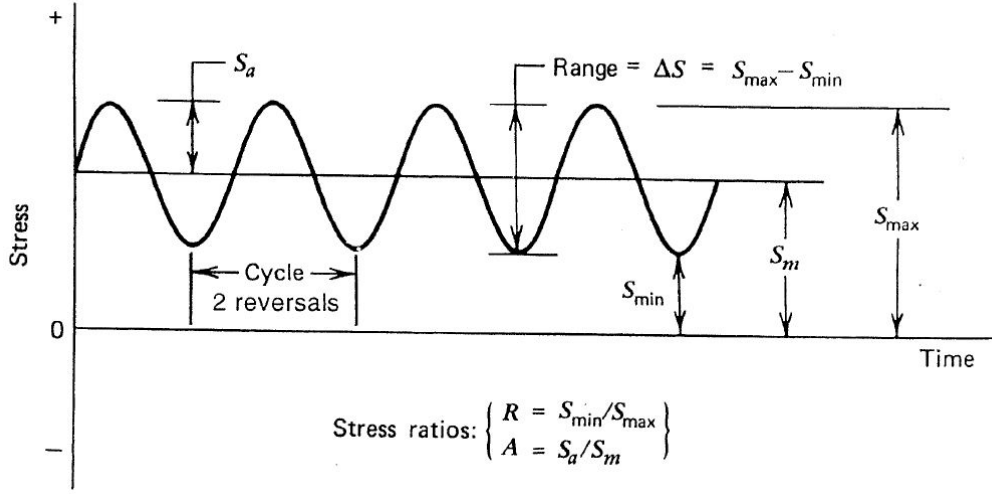


Figure 1.3: Uniaxial load/stress cycle example [3].

by many parameters defined by the following relationships:

$$\begin{aligned}
 S_a &= \frac{\Delta S}{2} = \frac{S_{\max} - S_{\min}}{2} \\
 S_m &= \frac{S_{\max} + S_{\min}}{2} \\
 S_{\max} &= S_m + S_a \\
 S_{\min} &= S_m - S_a
 \end{aligned} \tag{1.1}$$

In addition, the ratios are typical parameters, particularly  $R$ , the *Stress Ratio*, defined as follow:

$$R = \frac{S_{\min}}{S_{\max}} \tag{1.2}$$

$R$  takes different values but the useful range is from  $-2$  to  $1$ : in this region we identify certain fatigue cycles.  $R = -1$  identifies fully reversal-cycle: in these cycle maximum value is as larger as minimum one, obviously with opposite sign. In this case the tractive area is equal to the compressive one.  $R = 0$  describes origin cycle where the minimum value is equal to zero and the stress is always above that value. This second situation is more demanding than the first one, because the stress never assumes negative values. If we think about tension-compression, with  $R = 0$  we never have compression phase helpful to delay crack enucleation. In engineering technique this cycle is frequently used to describe the most challenging condition, moreover it is more damaging if compared with real work-cycle. It is called *Start-Stop* cycle, or *SS*, in order to clarify the relation with starting and finishing points, both set to zero value. This cycle is well know as *GAG* (*Ground Air Ground*) cycle in fixed-wing aircraft.

Restricting the study to only one load cycle is exceedingly reducing. If we imagine the variability of loads which act over mechanical components; indeed is possible to compare different sinusoidal loads. We are talking about cumulative fatigue, the phenomenon described with Palmgren-Miner rule. In the section 2.2.1 we will explain this analysis method.

## 1.4 Multiaxial stress state

As the name suggests, we are going to analyse a stress state where the loads act in more than one direction at the same time. Multiaxial stress state is surely closer to the real behaviour than uniaxial one where the main way is only one. Before going on, let us correctly define what we mean when we are talking about multiaxial stress state.

We identify a multiaxial stress state whenever, inside the stress tensor, there is more than one principal stress component different from zero acting at the same time. The most important feature, I want to remark, is the following: a multiaxial stress state could be created by an uniaxial external load. If we think about notched specimens or real parts geometry, over there there is a triaxial stress state, even though the specimen is loaded with tractive load only. Hence, it is the geometric discontinuity one of the reasons which create 3D stress state.

This case lost its importance because we will impose loads in different directions, in order to fill all the stress components inside the stress tensor.

In this first analysis of multiaxial stress behaviour, we want to clarify which attitudes are the most important. First, we define one stress as a function of time for each point of the body. The tensor is a time dependent function of six different components, its shape is:

$$\sigma(t) = \begin{bmatrix} \sigma_{xx}(t) & \tau_{xy}(t) & \tau_{xz}(t) \\ \tau_{yx}(t) & \sigma_{yy}(t) & \tau_{yz}(t) \\ \tau_{zx}(t) & \tau_{zy}(t) & \sigma_{zz}(t) \end{bmatrix} \quad (1.3)$$

where every single component could be time dependent or not. At least one component have to keep time dependency.

Depending on how the matrix is filled up we can separate different situations easier to analyse. The solution is less complicated because one or more components of the stress matrix are equal to zero.

- **Proportional stresses:** case described by lots of sine functions each one with the same frequency and the same initial phase. The feature allows to align the peaks of the sine functions and to clearly identify the critical instances immediately. Furthermore, we deduce that the principal directions are steady. According to *Mohr* theory we can calculate the rotation necessary to find principal directions in the  $\Pi_{X,Y}$  plane, as the following formula shows:

$$\tan(2\phi) = \frac{2\tau_{xy}}{\sigma_{xx} - \sigma_{yy}} \quad (1.4)$$

you will find the other directions permuting the stress indices coherently. Now, substituting the sine equations with same frequency and initial phase:

$$\tan(2\phi) = \frac{2A \cdot \sin(\omega t + \phi_0)}{B \cdot \sin(\omega t + \phi_0) - C \cdot \sin(\omega t + \phi_0)} = \frac{2A}{B - C} = \text{cost} \quad (1.5)$$

Another advantage is linked with the stress path that the load create over the reference plane. Plotting this path over a cartesian graph with  $\sigma_{xx}$  as abscissa and  $\tau_{xy}$  as ordinate, we discover a path equal to a segment. The feature will be discussed in the following chapter, where we take into account all the differences between fatigue criteria.

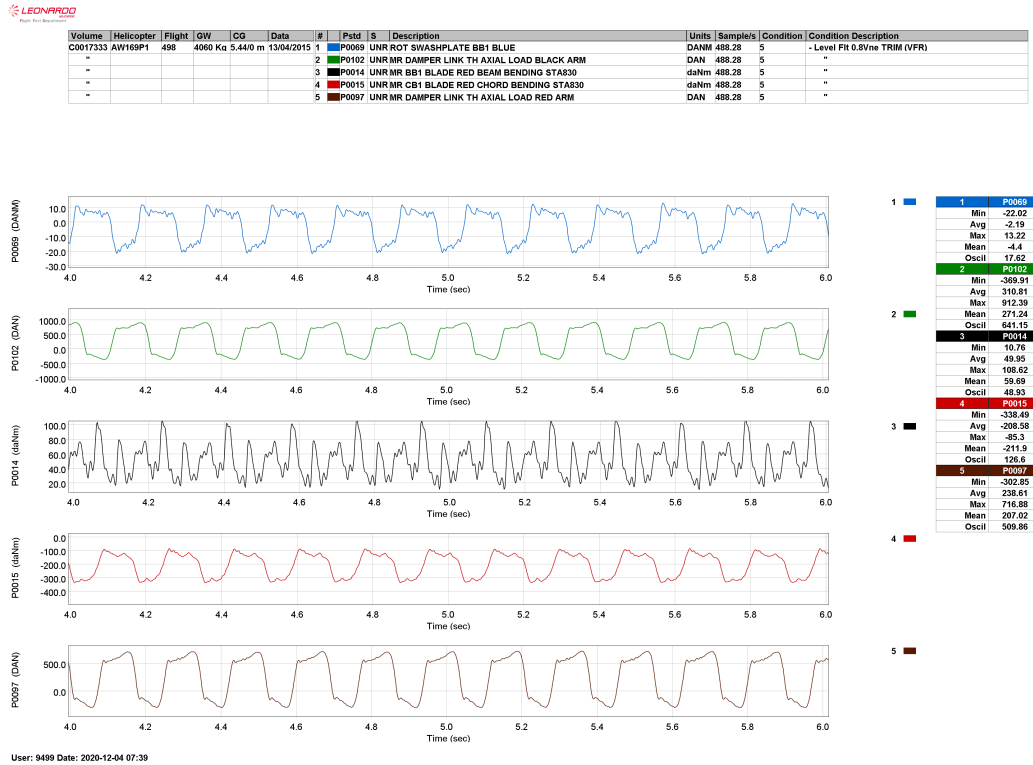
- **Not proportional stresses:** what we have told you before is no more correct, however load path is no more a segment but is a well known geometrical figure: an ellipse. The



critical condition will be linked with the major axis of the ellipse but it will not be recognizable rapidly and uniquely. In this case we maintain the same principal directions; in fact principal stress changes its modulus but not its direction.

- **Random stresses:** closest case to the real world where stresses are not describable with pure sine functions but with discrete values which we individually analyse. The stress path does not assume a canonical shape and it will be describe by a random path. This is the case we will debate in this thesis. In the following chapters I will explain which are the joint cause that create random stress path.

In the end, just a clarification must be done. Momentarily, I have talked of loads and stress with an incorrect ambivalence. In the following sections will be clear when we will talk about the first one or the other. The first one, measured in  $[N]$  or  $[kg]$ , will be described with a sine function because we want to maintain the strong relationship between model and experimental test. The second one, expressed in  $[MPa]$ , is affected by two factors: load time history and geometrical shape of the body. Each one varies with its own law described by a different sine. In example, we assert that a constant cross section specimen made by homogeneous isotropic material is free from notch effects and it shows sinusoidal stresses produced by sinusoidal loads. On the other hand, if we consider varying cross section through the specimen, or notch effect, it is possible to see a random stress time history generated by a sinusoidal load. It is possible to create non-sinusoidal stresses from sinusoidal loads. To better clarify the final aim of the thesis I report a flight-test report with the random loads in Fig. 1.4. The code will be able to analyse





## Chapter 2

# Fatigue life evaluation

### 2.1 Introduction

In the following chapter, fatigue criteria will be shown, in order to discuss fatigue life evaluation. We will present statements linked with uniaxial and multiaxial stress tensor, furthermore you will search which kind of information these criteria give to the designer. The ultimate purpose is to highlight, from criteria described in literature, the most valuable for our research. I carry out the maximum information with the lowest computational effort.

### 2.2 Uniaxial stresses

Uniaxial stress tensor allows a quicker solution than multiaxial one, because the critical direction is previously known and it is unique. Consequently, thanks to experimental curves, called Wöhler curves, we are able to relate stress and life cycles, thus we can evaluate the number of cycles the component bears before collapsing.

#### 2.2.1 Palmgren-Miner rule

The approach is known as cumulative damage method. First, we define what damage means: damage for each cycle with failure dynamic stress at  $N_f$  cycles is  $D = 1/N_f$  where  $N_f$  is the number of failure cycle with the reference dynamic stress. As you can understand, the previous formula describes one cycle damage, yet we want the damage for  $n$  cycles. We can write:  $nD = n/N_f$ .

The presented formula is referred to only one stress but we want to discuss multiple stresses as well. As multiple stress we interpret a stress time history created by more than one sine, each one acting in a different time for a different time slice. Let's assume that the stress time history is described by two blocks of stress, each one with mean value equal to zero, but with different dynamic stresses. Each one of them create different damage on the part. Now, two questions should be asked: how can I correctly calculate the damage of different stresses? Which rule should I use to connect both cycles? The answer to the first question is "Wöhler curve": Fig. 2.1(b) allows to immediately evaluate failure cycles knowing alternated stress only, in notation  $S_a$ . Wöhler curve is a bi-logarithm chart with  $S_a$  as ordinate and  $N_f$  as abscissa. If you know the number of cycles the part is subjected to in its life, described with  $n$ , you will be able to calculate the percentage of damage which that particular cycle produces in your component. The second

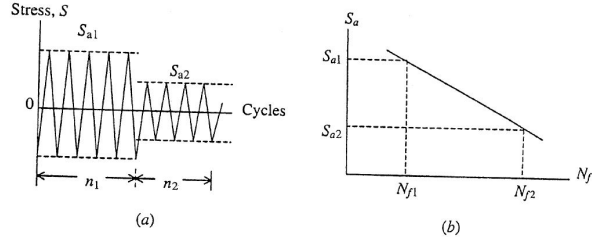


Figure 2.1: Double uniaxial cycle and ideal S-N curve [4].

question regards the real innovation introduced by Palmgren-Miner rule, they said:

$$\sum_i \frac{n_i}{N_{f,i}} = \frac{n_1}{N_{f,1}} + \frac{n_2}{N_{f,2}} + \dots = 1 \quad (2.1)$$

verifies component failure.

When the internal cumulative damage, the specimen could tolerate, reaches value equal to 1, or 100% as well, a failure occurs.

This analysis method has lots of drawbacks. For example, time history of load could not be ordered inside the formula. If I apply an increasing load time history, with a rising or decreasing of dynamic stress value; Palmgren-Miner rule gives me the same result: the part works for the same slice of time or for the same number of cycles. The formula describes a linear relation among different damages and due to linearity the order loss its importance. As some experiments have demonstrated, the stress cycles order is one of the most powerful key factors to modify components life. Indeed, a decreasing load time history extends parts life. Unfortunately, there is no possibility to take into account this factor with this rule. Another crucial rule's handicap is related to the residual stresses. There is a relationship between the sign of the last one cycle of stress, residual stresses and components life. Experimental tests can state the life is related with the last one cycle of stress. If the last one cycle reaches a positive peak, the resulting stresses will be compressive; thereby the following step will include a compressive starting stress. This detail has positive implications for fatigue life behaviour. Conversely, a negative peak will cause an opposite situation, with a drop of fatigue cycles [3]. The residual stress must be considered in the grain-size scale not in the macroscopic scale. High number of cycle fatigue works with stresses lower than the the yielding stress of the material, the global behaviour is linear and elastic but in a single grain a residual stress could be different from zero.

I believe it fitting to specify the real constraint which is hidden in the previous lines: I am talking about mean value of stimulating stress. This value is setted to  $S_m = 0 \text{ MPa}$  in order to better relate experimental problems and S-N curves. What I want to highlight is: if the cycle does not respect the value  $R = -1$ , the result is affected by an error. Although some theories have been developed we are trying to compare a fully reversal cycle with an equivalent one.

$S - N$  curve could be written as follow:

$$S_{Nf} = \sigma_f (2N_f)^b \quad (2.2)$$

where  $\sigma_f$  and  $b$  are two constant values of the method and they are linked with material properties;  $S_{Nf}$  is the stress with  $R = -1$ . As I told you before, if you do not have a cycle which respects stress ratio's constraint you will apply a correction. There are several possibilities, but

the well known and used are:

$$\text{Modified Goodman : } \frac{S_a}{S_f} + \frac{S_m}{S_u} = 1 \quad (2.3a)$$

$$\text{Gerber : } \frac{S_a}{S_f} + \left( \frac{S_m}{S_u} \right)^2 = 1 \quad (2.3b)$$

$$\text{Morrow : } \frac{S_a}{S_f} + \frac{S_m}{\sigma_f} = 1 \quad (2.3c)$$

When you select one of those formulae you know all the parameters except for  $S_f$ .  $S_a$  and  $S_m$  are data from stress cycle and  $S_u$  is a known material property. Thanks to Eq. (2.2), it is possible to evaluate  $N_f$ . This value represents the number of cycles which shows a fatal failure, on condition that you have applied only one segment of load time history.

### 2.2.2 Peak counting method

A general overview shows a crucial turning point related is the comparison of different failure occurrence with linear damage hypothesis. In fact, when you want to create a load spectrum from real data you are not immediately able to identify the worst condition. It is sometimes difficult to identify the cycle which envelop the real stress spectrum. Methods to find out criticality or not of a cycle are well known as *peak counting method*.

#### 1. Rain flow counting method.

The scheme in Fig. 2.2(a) reveals how it is possible to perform peak counting. The

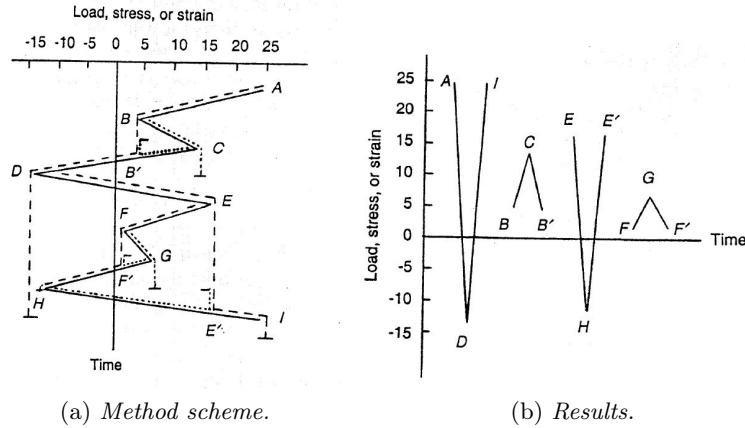


Figure 2.2: Rain-flow method graph [3].

procedure follows the path the water should be made if the stress time history would be compared with a roof. Starting from A the maximum path is A-B-D. From D you arrive to the end of the cycle with D-E-I path. After that you erase the “wet” area and you focus on the remaining part. Repeating these steps you are able to identify three others sub-cycles: E-H-E', B-C-B' and F-G-F'. Pay attention because the first hat-trick ends in E', it does not finish at the end of the starting spectrum because the area E'-I is already wet from the very first cycle. Rewriting on a chart the acting stresses and the hat-trick previously identified we obtain Fig. 2.2(b). From the graph is an easy task to define the most demanding cycle hidden into the starting spectrum.

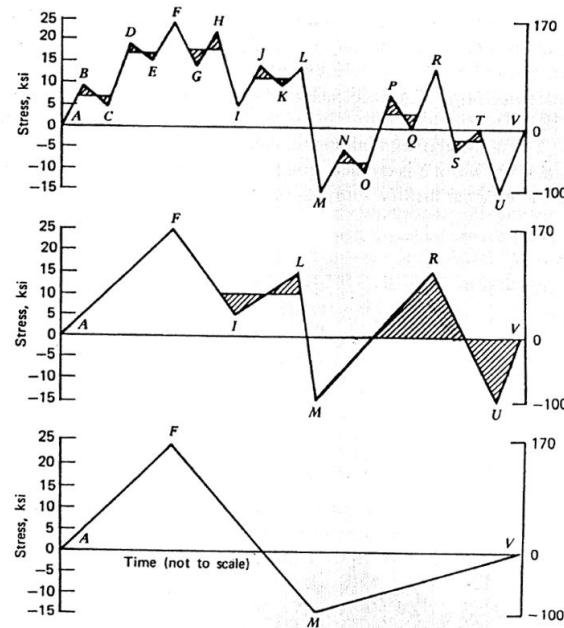


Figure 2.3: “Range-pair” method scheme [3].

2. **Range-Pair counting method.** The aim of this approach is to erase, step by step, the smallest delta stress. You need to iterate the process many times until you can find a basic variation. On the final graph you must see only one peak and one dip into the whole time range. As shown in Fig. 2.3 the more peak and valley you have, the more you need to iterate the calculus. The very last result is to define a more demanding cycle which cover all the origin spectrum.
3. **Racetrack counting method.** The method accurately follows its name. A band is

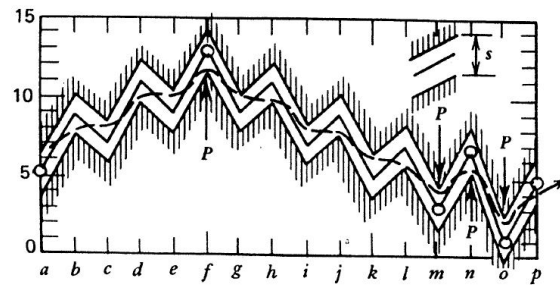


Figure 2.4: “Race track” method scheme [3].

drawn around load spectrum, it create a sort of race track where the borders reflects load spectrum itself. In Fig. 2.4 the contoured peaks are shown with a empty dot, these point main feature is the U-turn of the trend of the path, from ascending to descending and vice versa. A quickly implementation is the real power of the method, we can easily set the track's width in the code. In this case is useful to remind you that the 90% of damage is produced by 10% of whole spectrum; hence it is especially convenient isolating the critical section of the spectrum.

4. **Level crossing method.** First, you need to decide a threshold limit related to design



Figure 2.5: Spectrum partition based on “level-crossing” method [3].

parameters. Second, you slice the graph in some bands, the delta stress of a band could be set to an arbitrary value providing that the value is the same for ascending and descending stresses. Now you can start with the analysis in a separate manner, ascending and descending stresses are studied in the same way but in different steps. You count a peak when an ascending load passes throughout a positive stress’ band; with the same approach you count a peak when a descending load passes across a negative stress’ band. The Fig. 2.5 clarifies the described approach.

5. **Peak counting method.** Peak counting method is the easiest and quickest to apply. You

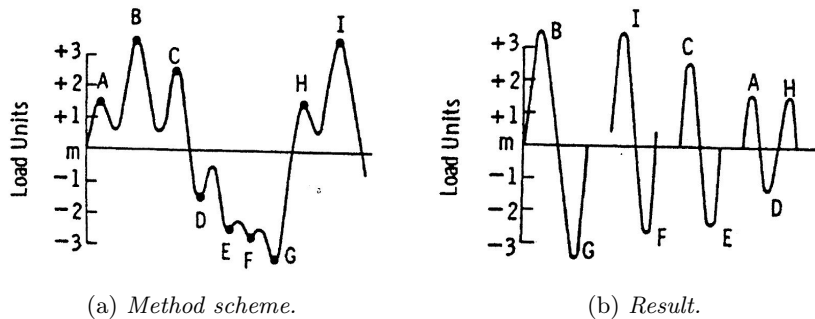


Figure 2.6: Graphical scheme of “peak counting” method [3].

take the spectrum and you identify absolute maximum value and absolute minimum value. The identified points represent the max and min values of the most critical cycle hidden into the spectrum. When the first cycle is known you delete the points from the starting graph and you iterate the procedure, you find a new couple of points which describes the second worst cycle and so on until the end of whole spectrum of loads. The Fig. 2.6(b) is the final result of the implemented method.

## 2.3 Multi axial stresses

As already said in the previous chapter multi axial stress state is the normal routine in engineering design, indeed it should be the only one approach to proceed to reach a high-fidelity simulation of reality. However uniaxial state simplifies a lot the calculus and in some situations its approximation

is acceptable. Henceforth, these terms will not be negligible. The dissertation reported in [3] introduces, as first step, the significant difference between proportional loads and and not ones. This first obstacle introduces the problem of the phase difference between sine function. This increasing complexity is not enough yet. The explanation is to find in the nature of loads, and inside complex geometry of parts witch works inside an helicopter's rotor. You will not be sure that a stress tensor's component is equal to zero in the whole period considered in the analysis. The tensor is described in Eq. (1.3).

Before joining code's development it is appropriate to describe basis theories and criteria in order to give a wide overview of fatigue behaviour. Most relevant criteria, described in the following sections, will be translated and implemented in the final code.

The mostly used level, talking about fatigue criteria, is the *elementary volume* one. This volume is defined as the volume which contains a such high number of crystal grains sufficient to consider chemical and mechanical properties as homogeneous. In this volume we are able to identify the central point which we will relate with the FEM node. This relation is one key-point of this work because form FEM node I will be able to extract stress components. We assert the stress tensor  $\sigma$  act with all its component on the central point of the reference volume. In order to understand tensor criticism we proceed as shown by Fig. 2.7, that is: The Eq. (2.4) describes

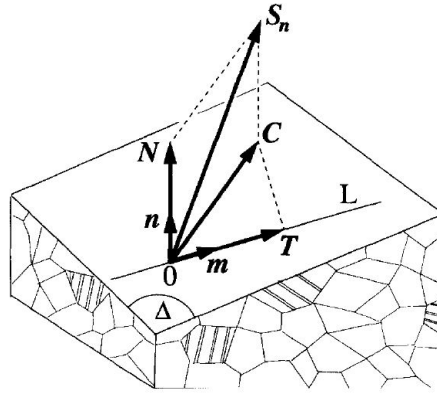


Figure 2.7: Elementary plane inside the reference control volume.

the projection on the plane identified by  $\Delta$ . Particularly,  $\bar{S}_n$  is the projection of the whole tensor referred to the plane. Now, on the plane, the stress could be broken up into its orthogonal and parallel components.

$$\bar{S}_n = \bar{\sigma} \cdot \bar{n} \quad (2.4)$$

The orthogonal component is not ambiguous, its modulus is:

$$N = \bar{n} \cdot \bar{\sigma}(t) \cdot \bar{n} \quad (2.5)$$

moreover, considering  $N$  as a time dependent variable, you can already focus on periodicity of  $N$  itself. The following formulas are a direct consequence of the work proposed in paper [5].

$$N_a = \frac{1}{2} [\max(\bar{n} \cdot \bar{\sigma}(t) \cdot \bar{n}) - \min(\bar{n} \cdot \bar{\sigma}(t) \cdot \bar{n})] \quad (2.6)$$

$$N_m = \frac{1}{2} [\max(\bar{n} \cdot \bar{\sigma}(t) \cdot \bar{n}) + \min(\bar{n} \cdot \bar{\sigma}(t) \cdot \bar{n})] \quad (2.7)$$

The component lie on  $\Delta$  plane could be evaluated as a difference between  $\bar{S}_n$  e  $\bar{N}$ , additionally this component is identified in Fig. 2.7 by the quantity called  $C$ . "C" evaluations is onerous



because  $C$  vector vary whether in modulus or in direction, however it stay grounded to  $\Delta$  plane. As you can learn in [7] and [8] papers a singular situation occurs; in practice the calculus steps might not be unique, you must follow some cleverness tips to avoid doubtful results. Fig. 2.8 shows a clear image of the ambiguity we were talking before. We assumes the main chord of this curve as the dynamic stress multiplied by two and the centre is the static value. This indefiniteness result is referred to mean stress; there are two different maximum chords with the same length, but the centre carried out from them is significantly different. Reminding that the centre is joined with the mean stress of the fatigue stress cycle, we are over-or-under-estimating stresses with a hidden and prejudicial effect on our calculus.

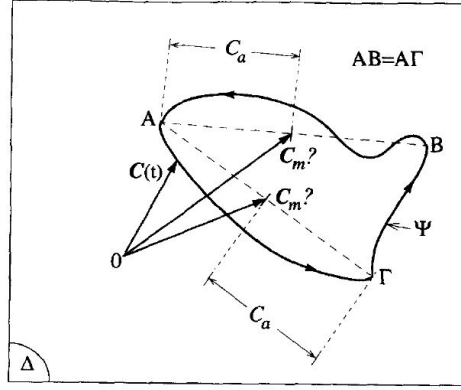


Figure 2.8: Generic path of  $C$  on an elementary plane, with ambiguous evaluation. [5]

Basically, the tactic we will follow in the code is a “brutal force” approach. Considering the discrete nature of data the code will have as input, it will be easier to calculate all the possibilities in order to find the most troublesome which we will call critical.

Before entering specific criteria, the Fig. 2.7 shows a third significant yardstick called  $T$ . This variable lies on  $\Delta$  plane and it is measured as projection of  $C$  over versor  $\bar{m}$  main orientation. This variable main formulation is:

$$T(t) = \bar{m} \cdot \bar{C}(t) \quad (2.8)$$

called *Resolved Shear Stress*, it is a scalar, time dependant function. You can evaluate it with the same approach presented for  $N$ . You split  $C$  on its own main directions as shown in Fig. 2.9. For sake of completeness we report the analytical formulas which solve the problem displayed in Fig. 2.9:

$$T_a = \frac{1}{2} [\max(\bar{m} \cdot \bar{C}(t)) - \min(\bar{m} \cdot \bar{C}(t))] \quad (2.9)$$

$$T_m = \frac{1}{2} [\max(\bar{m} \cdot \bar{C}(t)) + \min(\bar{m} \cdot \bar{C}(t))] \quad (2.10)$$

### 2.3.1 Principal stresses approach

These theories have been developed from the beginning of fatigue life study. The key point is linked with the principal directions of the stress tensor which have to be identified in order to arrive at a useful result.

At every point in a stressed body there are at least three planes, called *principal planes*, with normal vectors  $n$ , called *principal directions*, where the corresponding stress vector is perpendicular to the plane, i.e., parallel or in the same direction as the normal vector  $n$ , and where there

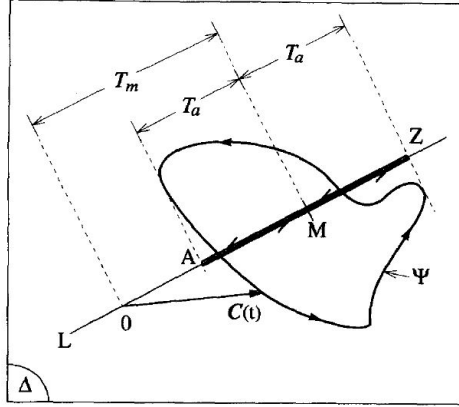


Figure 2.9: Root of Resolved Shear Stress components [5].

are no normal shear stresses. The three stresses normal to these principal planes are called principal stresses. The assessment of plane orientations could be performed with Mohr's theory [9]. This rotation assumes a principal role in principal stress methodology because the final aim of the theory is to rely a three-axis-stress tensor with an uniaxial equivalent one. In literature we can find different approaches to perform this equivalence, we select the most valuable:

1. **Tresca's or maximum shear stress theory.** This theory asserts a complete stress state is equal to a uniaxial one when the shear stresses  $\tau_{1,2}$  are maximized. For the purpose of finding this tension we need to prior know the principal directions. We realize the rotation necessary and we obviously find zero shear stress. Tresca's theory says that accomplishing a  $45^\circ$  rotation around the third principal axes,  $\tau$  components born and this these shear stress are maximum as well. Another feature: these stresses act in the plane not orthogonal to the rotation axes. The rotations of  $\pm 45^\circ$  fulfil maximum and minimum value of tangential stresses. In the end, a third rotation around first principal direction is needed, this rotation allow  $\tau_{13}$  evaluation. At this time we know three values which create a hat-trick over that we evaluate the absolute maximum.

Mohr's circle helps our analytical evaluation. You can geometrically highlight that maximum  $\tau$  is the vertical radius of the circumference and the horizontal diameter is the difference between the principal stresses. It is easy to understand that the relation between principal stresses and maximum shear is the same as the relation between radius and diameter of a circumference. Called  $\sigma_1, \sigma_2, \sigma_3$  the three principal stresses the maximum shear will be research in:  $|\sigma_1 - \sigma_2|/2, |\sigma_1 - \sigma_3|/2$  or  $|\sigma_2 - \sigma_3|/2$ . Evaluated  $\tau_{max}$  we reuse Mohr's circle to relate  $\tau_{max}$  and the uniaxial tractive equivalent stress. According with the theory we find:  $\sigma_{eq} = 2\tau_{max}$ .

2. **Von Mises or maximum distortion energy theory** this criterion searches a relationship between a multiaxial complete stress state and a uniaxial one. The parameter Von Mises wants to match the distortion energy, not only a stress component but a quantity linked with the whole energy. I need to precise the difference between distortion and deformation energies. The first one talks about the deviatoric part of the stress tensor, on the contrary deformation talks about whole tensor components. When you analyse distortion energy you neglect hydrostatic part of the tensor. The procedure by which you can find the solution is fully described in [9]; in sake of simplicity we summarize it with main steps:

- (a) Bring an arbitrary 1D stress.
- (b) Evaluate hydrostatic term.
- (c) Evaluate hydrostatic work.
- (d) Calculate global elastic deformation work.
- (e) Subtract the terms:  $L_{distr} = L_{def} - L_{hydr}$ .
- (f) Do again procedure (points from b to e) with a 3D state.
- (g) Even out the results of distortion energy for 1D and 3D states
- (h) Reshape formula.

You obtain:

$$\sigma_{id} = \frac{1}{\sqrt{2}} \sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2} \quad (2.11)$$

The Von Mises' formula gives a ratio respect to the Tresca's one which is equal to:  $\sigma_{id}/\tau_{cr} = \sqrt{3}$ .

The notation with 1, 2, 3 subscripts refers to principal stress components.

These theories are used with static stresses in order to evaluate structural collapse, they could be adjusted to fatigue behaviour with correct attention. If we consider only stress state with proportional stress time history, these theories are able to figure our problem out in a precise manner. The real limitation is attached with the same phase all the stresses must follow.

Every single tensor component is the result of a sine function described by static and dynamic values. You identify principal directions, which are fixed in space according with Eq. 1.5, and you can directly evaluate static and dynamic values of equivalent weary cycle. Formulas useful for our purpose are:

$$\sigma_{id,m} = \frac{1}{\sqrt{2}} \sqrt{(\sigma_{1,m} - \sigma_{2,m})^2 + (\sigma_{2,m} - \sigma_{3,m})^2 + (\sigma_{3,m} - \sigma_{1,m})^2} \quad (2.12)$$

$$\sigma_{id,a} = \frac{1}{\sqrt{2}} \sqrt{(\sigma_{1,a} - \sigma_{2,a})^2 + (\sigma_{2,a} - \sigma_{3,a})^2 + (\sigma_{3,a} - \sigma_{1,a})^2} \quad (2.13)$$

Both criteria want to evaluate the maximum shear stress on the deviatoric plane. Tresca's and Von Mises' approaches remark the fatigue genesis is linked with the shear which creates the local tangential slipping of the crystal planes. The phenomenon is born of distortion energy than the shear stresses are the useful only.

### 2.3.2 Critical plane approach

All the critical plane theories fetch normal stress and tangential one, both acting on the critical plane. There are several theories we count in this macro category. These approaches have been used to analyse non-proportional multiaxial stress states, in order to achieve critical location for cracks enucleation and growth. [4]

First, we refereed our explanation on Fig. 2.10 because we want to identify geometrical breakdown we will follow in the analytical demonstration and in the code as well. Global reference frame is centred in the leading point of the study, which will be the FEM node. Obviously, the point is the only one position in space where we can access stress information. The primary angles  $\varphi$  and  $\theta$  allow unmistakable definition of plane called  $\Delta$ . Over this plane we made stress

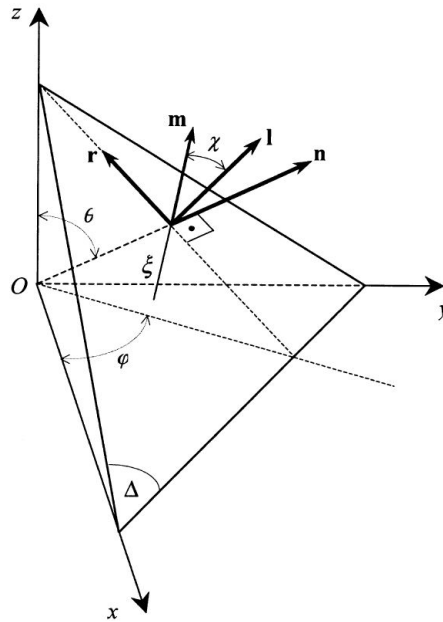


Figure 2.10: Description of a critical plane ad local reference frame. [6]

tensor projection with a view to obtain the same scheme reported in Fig. 2.7. The local frame empower us to correctly write rotation matrices and tensor projections.

Plane  $\Delta$  is a generic one, it is not the critical plane yet. The criterion defines the critical condition, only if the plane verify the condition it will call critical; consequently, method steps are always two:

1. Identify the critical plane thanks to stresses projection and according with the chosen theory. Only in this step the code needs to identify the plane according with the reference theory.
2. Control whether the criteria is verified on the plane or not. Now, you take into account the material properties which are evaluated by previous experimental tests. Thanks to the criterion and the acceptable material stresses you perform the criterion evaluation.

Solved these points we can summarize the solution. We effort, if the criterion is not verified, the area, referred to that particular plane, will not be an enucleation area. Clearly, the most relevant information are linked with the directions of crack birth and initial propagation; these directions are defined by the critical plane ones.

In the following chapter I will describe the implementation strategy I followed inside the codes to numerically solve the problem. I will explain interaction between my solver and FEM models explaining problems and solutions adopted. Now, you will find analytical explanation of main theories.

## Findley's theory

The theory is based on delimit linear combination of  $C$  and  $N_{max}$  acting on the critical plane. Previously, you need to recognize the critical plane defined as the plane which maximize the same

method equation. The plane descibed in spherical coordinates,  $(\varphi^*, \theta^*)$ , is:

$$(\varphi^*, \theta^*) = \max [C_a(\varphi, \theta) + \kappa N_{max}(\varphi, \theta)] \quad (2.14)$$

once identified the plane we apply the criterion as follow:

$$C_a(\varphi^*, \theta^*) + \kappa N_{max}(\varphi^*, \theta^*) < \lambda \quad (2.15)$$

The principle is to give the same weight both plane and criterion main equation. Due to this idea there is an over extimation of mean torsional relation, consequently the critical areas are affected by an error. More precisely there is not a correct relation between fully reversed torsional fatigue limit and mean stress of demanding cycle. The relationship between averaged shear stresses and torsional fatigue limit is opposite to experimental evidence. This statement adds with the experimental confirm shown by [6] and [7] leads us to remove the criterion form our study.

### Matake's theory

T. Matake's theory [10] considers a linear combination constraining of  $C$  and  $N_{max}$ , but with a substantial difference form Findley's approach. The critical plane is defined as the plane with the maximum value of the alternated stress component of  $C$ . We remind that  $C$  is the projection of stress tensor over  $\Delta$ -plane, acting on analysis point. In analytical words we can write:

$$(\varphi^*, \theta^*) = \max [C_a(\varphi, \theta)] \quad (2.16)$$

once identified the plane we apply the criterion which is the same of Eq. (2.15), that is:

$$C_a(\varphi^*, \theta^*) + \kappa N_{max}(\varphi^*, \theta^*) < \lambda \quad (2.17)$$

A remarkable clarification should be done. Eq. (2.15) and Eq. (2.17) are the same but the formulas hidden inside  $\kappa$  and  $\lambda$  are different. Talking about Matake's definition, you calculate these coefficients as related with pure torsion ad pure bending behaviour with fully reversal cycle, i.e. ( $R = -1$ ). With fully reversal torsion you find:

$$C_a(\varphi^*, \theta^*) = t_{-1} \quad (2.18)$$

$$N_{max}(\varphi^*, \theta^*) = 0 \quad (2.19)$$

hence:

$$\lambda = t_{-1}$$

At the same time with fully reversal bending, you do:

$$C_a(\varphi^*, \theta^*) = \frac{f_{-1}}{2} \quad (2.20)$$

$$N_{max}(\varphi^*, \theta^*) = \frac{f_{-1}}{2} \quad (2.21)$$

whence:

$$\kappa = \frac{2t_{-1}}{f_{-1}} - 1$$

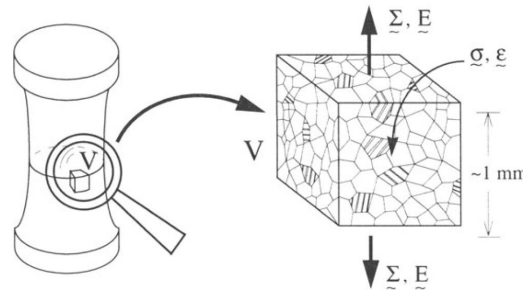


Figure 2.11: Relation between macroscopic and mesoscopic scales. [7]

### 2.3.3 Mesoscopic scale approach

With this approach we analyse high cycle fatigue, where plastic behaviour is negligible to find out a precise life estimation, we suppose the macroscopic material is linear and elastic. In the previous sections we have talked about different approaches revealing the distinction between uniaxial and multiaxial fatigue investigation. Mesoscopic approach move the focus to the scale of the problem, as the title suggest.

In engineering sphere the most useful scale is the continuum scale, essentially a macroscopic approach. The main example of this scale is related with loads, stresses and strains. All of this act over the whole body. The scale is largely used because there are an high number of solved theories useful to sort structural problems out. Sometimes there is an approximation hidden in the usage of this approach, i.e. strain gauges. The collected data are affected by an approximation which born from the area of the gauge, yet they are attached as punctual ones. In our analysis, with non-proportional loads we commit errors when we analyse huge difference of phase between applied loads.

Diametrically on the other hand, there is the discrete scale approach. It is used in metallurgy where you examine problems related with molecular structures, dislocation plane and micro-cracks. This level of detail complicates the theories used to achieve the result. We never use this scale in this work.

At the end, we reach an intermediate position, the mesoscopic scale. The first aim of this approach is to collect benefices from both continuum and discrete scales. Particularly, you assumes strains acting on crystal grain. The grain is considered as the middle point between continuum and discrete. However, is not possible to identify material properties directly form crystal grain. We need to base the method on some hypothesis, the main one is: the grain is considered as the reference unit of out study, it is the smallest quantity in which we are able to identify homogeneous material properties. Thanks to this statement we have just relate mesoscopic world with the macroscopic one. The scale, we were talking before, you can see the Fig. 2.11.

The detailed steps is written in [11], but in sake of simplicity for the following sections I will follow steps presented in [7].

You consider the volume  $V$  made up several crystal grains, each one with its properties whether mechanical or physical. You must also give attention to slip-plane, that is linked with predominant direction of chrystal grains inside the volume  $V$ . This step has to be done because if the body endure an external load the global behaviour is mostly linear and elastic but locally it might shows micro plasticization or micro plastic-slips. The relationship between slip-ways

and macro mechanical properties is the focus of the theory. The relationship between plastic-deformation growth is proportional to a macroscopic entity, called  $T_a$  and shown in Fig. 2.9. This parallelism is legal if we consider infinite life behaviour, in other words you have to verify a mean stress far from macroscopic yielding limit of the material. This linkage is readable as  $T_a \approx \Delta\gamma^p$ , it declares the criterion to verify fatigue limit is a sort of threshold limit of  $T_a$ . Moreover, there is a limit of plastic-slip and of plasticity growth inside crystal grain described by  $\Delta\gamma^p$ .

Understood the previous paragraph we need to take into account the engineering practice. Our goal is to avoid crack enucleation and propagation with a size similar to volume  $V$ . What you want to really avoid is the coalescence of micro cracks each one in a different grain but all in the same area, inside volume  $V$ . The method perform this task restricting an averaged quantity. This value is strongly related with  $T_a$  and it is evaluated as a volumetric mean integration.

In sake of linear explanation I follow the scheme reported in Fig. 2.10. The angles, the frames and the planes will be called in the same way of the critical plane approach. Reminding Eq. (2.9), and taking advantage from previous steps we can obtain the formula I will write in the code in few passages. The averaged value born from our observations is:

$$\sqrt{\langle T_a^2 \rangle} = \sqrt{5} \sqrt{\frac{1}{8\pi^2} \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi} \int_{\chi=0}^{2\pi} T_a(\varphi, \theta, \chi)^2 \sin \theta d\chi d\theta d\varphi} \quad (2.22)$$

$\chi$  angle allow to scan all the possibilities on the plane, in order to find out the worst one. It make the volumetric average critical. In addition,  $\varphi$  and  $\theta$  allow an unmistakable identification of the plane inside the control volume  $V$ . The approach is in perfect similarity with the critical plane approach. The integral-average of quantity well know as *Resolved Shear Stress* is evaluated as “root mean square”. It is the major parameter we will control to avoid crack enucleation. You understand the datum is the threshold limit to impose in our stress life to avoid crack enucleation. However  $T_a$  does not describe all the enucleation behaviour, moreover we need to consider the effect of the static stress in order to correctly understand the attitude. The relation with the static stress is linked with a averaged integral of normal stress, assessable as follow:

$$\langle N \rangle = \frac{1}{4\pi} \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi} N(\varphi, \theta) \sin \theta d\theta d\varphi \quad (2.23)$$

this integral mean is in strongly relationship with hydrostatic stress, thus the theory confuse with each other. The notation has changed in order to avoid misunderstanding, and we identify the new parameter as  $\sigma_H$ . Thanks to integral mean you can overcome the elementary volume concept and write a criterion which assumes the following aspect:

$$\sqrt{\langle T_a^2 \rangle} + \alpha \sigma_{H,max} < \beta \quad (2.24)$$

where the hydrostatic stress is maximized for safety reasons and to study the most demanding situation.  $\langle N \rangle$  is a time dependent function which reaches the maximum value which is similar to  $\sigma_{H,max}$ . The coefficients  $\alpha$  and  $\beta$  are taken from material properties. The calculus process of aforementioned coefficients is excessive long and useless. In sake of simplicity I avoid the passages to better explain the results we could obtain.

Baseline idea is to evaluate the coefficients in the best situation ever. They describe material properties which does not change with different stress cycles. With the easiest cycle you can simplify the analytical formulation and you set up an experimental test with the aim to find the parameters. You stress the body with pure bending and pure torsion separately. You have an

equation with untied  $\sigma_a$  and  $\tau_a$  components: the first talks about bending the second discuss about torsion.

$$\sqrt{\frac{\sigma_a^2}{3} + \tau_a^2} + \alpha \frac{\sigma_a + \sigma_m}{3} \leq \beta \quad (2.25)$$

Applying fully reversal torsion, i.e.  $t_a = t_{-1}$  and fully reversal bending, i.e.  $\sigma_a = f_{-1}$  and  $\sigma_m = 0$ , we obtain:

$$\alpha = \left( \frac{3t_{-1}}{f_{-1}} \right) - \sqrt{3} \quad e \quad \beta = t_{-1}$$

where  $t_{-1}$  and  $f_{-1}$  constitute torsional fatigue limit and bending fatigue limit respectively.



## Chapter 3

# Solver development

### 3.1 Introduction

In this chapter I will explain the numerical implementation and the solver development which are the core of the project. The Chapter 4 will explain different types of input and the managing of work flow inside the whole tool. As first step I will bring the essential formulas which are the core of the methods I want to implement inside the code. Second, I will explain numerical problems of previous version of codes in order to improve performances. At the end, you will see how dimensions of engineering problems play a primary role in the definition of global calculus time. This massive amount of data forces me to develop of codes which have been improved in order to reduce computational cost. I will show you the time improvement of every step. Final version of the code will be shown in the Appendix.

### 3.2 Application

First part of implementation consists in translating from MathCad<sup>®</sup> language some multiaxial fatigue criteria. During the translation phase I perform two different tasks: understanding of loop-strategy inside the code and optimization ways accessible to improve time performances. Last one goal is to guarantee a solution in a reasonable slice of time compared with the massive number of nodes which we can find in the skin of a structural part described in a FEA model.

The criteria, or the solver strategies, chosen to achieve the highest quantity of useful information are the following:

- **Matake's theory:** I remind this theory is numbered among “critical plane methods”. The approach has the aim of evaluating which is the 3D space orientation of the plane which shows critical behaviour. The criticality of a plane is defined by the method according with dynamic shear stresses acting on the point. When the critical plane has been identified you evaluate method's equivalent stress.
- **Mesoscopic's approach:** this method goal is the evaluation of the damaging equivalent stress but the process to follow to reach the solution is slightly different from the previous one. The process is based on the volumetric mean over control volume  $V$ .
- **Principal stresses approach:** This approach, developed in the Rotors D&D department of Leonardo Helicopters <sup>©</sup> [12], aims at evaluating principal directions with a rotation of

the stress tensor performed with the diagonalization of the tensor itself. A focus on the method will be explained in the dedicated section.

At the beginning, first translation assesses the main difference between the programming languages. Obviously, I am talking about time difference, nothing about syntax will be shown in this work. During introductory part is important to be sure the MATLAB<sup>®</sup> language performs a real earning of time. Furthermore, I validate the translation, in fact you want to reduce time but you do not want to downgrade the numerical precision of the calculus. You find out that with the same numerical structure solution the time falls down from minutes to seconds per node, as shown in Tab. 3.1. This elementary result allowed me to continue with the work knowing that

Time comparison MathCad-MATLAB		
	MathCad	MATLAB
Matake	2 min 15 s	45 s
Mesoscopic	3 min 48 s	75 s
Principal stresses	21 s	0.51 s

Table 3.1: Time comparison between MathCad and MATLAB codes with equal solving structure.

the remodelling of the solver structure and its subsequent optimization would lead to further time improvements. These times are a good omen because the difference is clear without any modification of the structure of loops and local variables. For better understanding which might be the size of an engineering problem I set the number of iterations at 100000. The number is equal to the number of nodes in FEA model.

Considering the surface nature of fatigue, the evaluation could be performed only over the skin nodes. Obviously, the less the nodes are, the faster the code is. In Tab. 3.2 I show the evaluation time for 100000 nodes. Global time is extremely far from our idea of reasonable, they

Global time with MATLAB		
	1 evaluation	100000 evaluations
Matake	45 s	~ 52 d
Mesoscopic	75 s	~ 86 d
Principal stresses	0.51 s	~ 14 h

Table 3.2: Global evaluation time for 100000 nodes with the first MATLAB code.

do not have any worth in our study, but they set the starting point of my development.

Let's focus on the singleton criterion. In order to achieve a linear reading of the work, I describe optimization procedure of each criterion in a separate section. Thanks to this you can follow the complete development of a criterion without pauses. Inside the section I will try to explain all the problems I manage during this work.

### 3.3 Matake's theory

As you can see in the Chapter 2.3.2, the final expression of this theory is:

$$C_a(\varphi^*, \theta^*) + \kappa N_{max}(\varphi^*, \theta^*) < \lambda \quad (3.1)$$

with  $\kappa$  and  $\lambda$  the coefficients described in the Eq. (2.19).

At the beginning of the solution, the analyst knows the stress tensor in a precise point in space, the tensor is evaluated in some time steps. The last one sentence poses a question over the time discretization which you should do to reach the best solution. In order to easily compare my codes with MathCad ones I set the same discretization both in time and space. Let's delve into this problem.

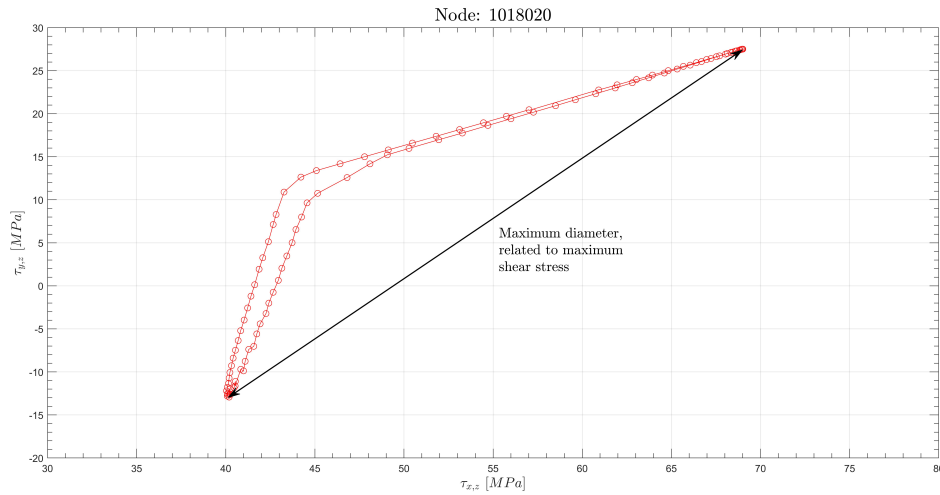
#### 3.3.1 Discretization

MathCad inputs data are 26 time steps into a 1.5 sec time slice and 51 steps in the space discretization. Consider that the time slice is referred to reference frequency similar to 1 Hz.

Focusing on time steps, the number of steps has to be connected with application-load-frequency. Every single component of load has its own sine function described by: mean value, amplitude, initial phase and own frequency. If the value of frequency does not overcome 4 Hz, 25 steps are enough to correctly capture the behaviour of the sine. Indeed, in this first stage I maintain the same values. Moreover, you can talk about the specific weight of time slices in relation with global calculus time. The time-control cycle is the deepest in the solving structure, therefore its effect over global time is dependent from space discretization. If you modify the time discretization from 26 to 27, you will not increase by 1 the global iterations, yet you will increase of  $51 * 51 = 2601$  steps. Both fifty-one represent the discretization of  $\varphi$  and  $\theta$  respectively. This idea could be flipped when you reduce the number of steps. All these numerical problems will be clarified in the following section.

Watching out for space discretization we remark the previous concept: you maintain the same number of steps to numerically validate the translation after we modify it to understand the importance of the discretization on the solution. To deeply understand the specific weight of space steps you need to focus on critical plane, sure enough the plane it will be one of the discretized ones. Between one plane and the following there is nothing for the code and if the first step is far from the second, you are probably committing an error. The error is linked with the "blank space" let among planes by a sparse discretization. On the other hand a dense discretization soars the global time. Hence, you need find a compromise between time and precision.

Talking about Fig. 3.1 it is useful to understand the stress path on the reference plane  $\Delta$ . On the plane the evaluated steps create a path with a random form, the code numerically evaluates all the couples and it extracts the maximum length shown with the double arrow in the figure. The distance is related with the maximum shear stress amplitude, moreover the path is a closed curve because the imposed stress is a pure sine. If the input is not a pure sine the path will be open but the code evaluates the curve with the discrete points and this makes no difference regard implementation structure. Comparing the figure with the theory described in the Chapter 2.3.2 the double arrow represents the diameter of the maximum circumscribed circle which described the most demanding cycle.


 Figure 3.1: Stress path on the reference plane  $\Delta$ .

### 3.4 Solution

Understood discretization problems you pass to the real solution phase. The critical plane maximizes a certain stress quantity, as shown in Eq. (2.16), then you evaluate all the couple of  $\varphi$  and  $\theta$ . The couples describe all the planes inside out space discretization, after you insert time dependency. The code follows the steps described in the list:

- **Step 1:** I identify the “active plane” where the code evaluates the tensor rotation.
- **Step 2:** You rotate stress tensor in order to project its components in the directions defined by the plane. The process fulfils rotation matrix evaluation and the product  $R^{-1} \cdot S \cdot R$ .
- **Step 3:** I evaluate maximum shear stress amplitude on the plane following the time history.
- **Step 4:** I apply the formula and I identify the critical plane.
- **Step 5:** I evaluate the equivalent stress on the plane.
- **Step 6:** Then I repeat the same procedure on the following plane.

Inside the steps there is a 4-nested-loop structure, composed by *for* loops. The first and the second, the external ones, allow the code to access the coordinates of the physical plane. Moreover, the internal ones enable the stress evaluation with an easy subtraction. The time cycles identify the time couples in the time discretization.

The crucial point of the analysis is the stress tensor rotation, which is performed with backslash command. The operation takes a slice of time equal to the 94% of whole time necessary to achieve the task. Later the time cycles evaluate the maximum dynamic stress and after these cycles the code extracts the global maximum value. This value is not the overall maximum value, but is the maximum for the reference plane; in fact I will find a maximum for each couple of  $\varphi$  and  $\theta$ . The code extracts the overall maximum only at the end of all cycles.

The maximum position is identified with matrix's indices which identify the plane in the discretization vectors. Now, you apply the formulation to calculate the equivalent stress in the same plane.

The code shows some benefits but lots of drawbacks. The most useful advantage is the extremely easy way used to write the code. I analysed the code, then I focused on key points, but I did not solve manually and I decided to implement the same structure in the machine code. This approach gives readability to the code but it gives back wasteful calculus. The main wasteful points are:

- Tensor rotation is a not time dependant function, it depends on the angles only. Its evaluation could be done faster and faster.
- Matrices are symmetrical, it has no sense to fill them completely.
- The evaluated data are superabundant if compared with the needed ones. I wast time evaluating unnecessary values.

### First improvement

First variation is the reduction of the evaluation number of the rotated tensor. Thanks to *if-else-end* construct I delete the multiple and repetitive tensor rotation evaluation.

### Second improvement

I take an advantage of symmetry of matrices which came from inside properties of stress tensor. I reduce the iterations number forcing the second loop indices. In the new code the inner loop is not *for r = 1 : end* but it is *for r = q : end*; with *q* the index of time in the outer loop. Thanks to this play the evaluated part of the matrix is more or less an half. To give you an idea: if you have 30 time steps, that is  $900 = 30 * 30$  time iterations, the new code will fill 465 positions only. The Fig. 3.1 is a clear method to understand this passage. You identify the couple and you evaluate the distance only once. In addition, I neglect the main diagonal because it is the result of subtraction between time step *i* and itself; the results are identically equal to zero.

In spite of iterations reduction plan, global calculus time is similar to 0.5 seconds. This time is a nice step if referred with the previous code. It is still far away from the application with real problems, indeed, if you consider a problem with a real magnitude, around 100000 iterations, you will find:

$$TotalTime = \frac{SingleNodeTime \cdot NumberOfNode}{3600} = \frac{0.5 \cdot 100000}{3600} \simeq 13h 50m \quad (3.2)$$

This results is to compare with the Tab. 3.2 where the overall time was fifty-two days.

### Third improvement

Thanks to global evaluation time and implementation structure I understand to leave the core of the code, that is the matrix reversal hidden in MATLAB command *backslash*. I suppose the approach improves the time saving of my code and I start writing all the analytical steps to find out the final result. The aim of this massive amount of maths passages is to write the precise formula of the rotated stress components. I calculate all the components but in the code I implement the useful ones only.

In sake of transparency, all the analytical steps are developed in the Chaper 3.7. Over there you can find the passages to escape *backslash* command. The implementation of the new formulas in the code created the third version of the code. The newest version brings about extreme time saving of the whole code. Tab. 3.3 shows the improvements:

Solver improvement				
	Starting time	First step	Second Step	Third step
Single iteration	45 s	20 s	0.5 s	0.15 s
1e+5 iterations	28 d	9 d	14 h	4 h 30 m

Table 3.3: Matake's theory code improvements.

The time advancement is remarkable and it justifies the strategy change, summarizing we have:

1. I delete the *backslash* command, MATALB does not switch any matrix.
2. I exploit single stress components, thereby it is not necessary to store the whole tensor in a 3X3 structure, but you could store them how you prefer. Honestly, the aim is to save space wherever it is possible. All of this performs an advantage you will find in the Chapter 4.
3. You can evaluate the useful components only for Matake's method:  $S_{r,n}$ ,  $S_{l,n}$ ,  $S_{n,n}$ .
4. Save usage of RAM. You think about a 3X3 tensor, each one needs  $3 * 3 * nNode = 900000$  positions, each one requires 64 bit, that is 6.86 MB of RAM. Storing six components only, you can reduce the used RAM to 4.57 MB, with a space earning of 33%. You are not worried about few MB of RAM if compared with the power of modern computers, but the whole optimization passes through this point as well.

I contextualize the overall time with the future usage of the code. In the preliminary design of the part you do not wait 4-hour analysis to see the critical areas. This time is a good result but it is not enough. Additionally, it is necessary to find something else to improve time saving and to improve solver speed.

### Final improvement

This evolutive step takes place from the idea that the FEM nodes are independent, from the point of view of the the code. You well know the strong relationship between FEM element and its nodes in the whole structure. When you extract from FEA code the stress tensor, this feature is lost. The node becomes a stand-alone entity. This statement causes two advantages: first, the code can solve the node in an absolutely random order; second, maybe it is possible to solve more than one node at the same time.

To set up this idea are useful the main features of modern CPUs: they are built with a multicore architecture, furthermore MATLAB toolbox plays a vital role. In order to reduce overall time I decide to use two different MATLAB toolboxes: GPU computing toolbox<sup>TM</sup> and the Parallel Computing Toolbox<sup>TM</sup>. Briefly, you can think these toolboxes use the parallel calculus architecture to speed the solver. The first uses GPU, the second multicore structure of CPUs.

Talking about GPUs, well known as *Graphics Processing Unit*, my previous implementation is unusable, let's try to understand why. The GPUs are built to process parallel tasks, with the aim of work demanding jobs off in the fastest way. MATLAB allows the possibility to pass the vectors to the GPU, after the GPU itself elaborates the code, in the end it returns the results to MATLAB. The three steps could be performed only in case of GPUs MATLAB functions, the list

of MATLAB functions which work on the GPU is not the real problem because is extended. The first block is linked with the word “pass”. It is necessary a not negligible slice of time to pass input vectors, which are allocated in the RAM, to the GPU for the computing. This time is negligible if the computing is completely done in the CPU. Moreover, there is the return time necessary to copy the results from GPU to RAM. The second block is related with the implemented structure. The code has lots of input vectors and matrices, i.e. stress tensors, space and time discretization vectors and so forth. Additionally, the results are composed by more than one value to return to the CPU. The sum of the transfer time is a disadvantage and the GPU is not able to outdo with its speed. Due to this evidence I decide to leave this strategy to focus on the other.

The most accessible way to speed the code is the Parallel Computing Toolbox<sup>TM</sup> which allows to solve at the same time more nodes iterations. The code will identify the number of cores available on the machine and it will use it for the calculus.

Thanks to this toolbox the overall time for 100000 nodes is 40 *mins*. This time is extremely correlated with CPU’s power and CPU’s activities. Obviously, a 4-core machine is less performing than an 8-core one. If I utilize all the power of an 8-core laptop the time is the aforementioned. This time is reducible if you use the code on a 12-core work station.

Tab. 3.4 shows all the development of the code.

Complete solver improvement					
	Starting time	First step	Second Step	Third step	Final step
Single iteration	45 s	20 s	0.5 s	0.08 s	0.018 s
1e+5 iterations	28 d	9 d	14 h	2.2 h	35 m

Table 3.4: Mataka’s theory final code improvements.

## 3.5 Mesoscopic Approach

Mesoscopic scale approach, or mesoscopic approach only, provides a numerical integration which averages stresses over the control volume. To be clear I report the main formula of the criterion written in the Eq. (2.24):

$$\sqrt{\langle T_a^2 \rangle} + \alpha \sigma_{H,max} < \beta$$

Let's proceed in the same manner of Matake's theory.

### 3.5.1 Discretization

Time discretization is equal to the one implemented before. The main code reflects MathCad first implementation and Matake's discretization. This step is necessary in order to comfortably compare the results, especially overall time through improvements.

The most challenging problem of the method is the introduction of a new variable which is a new space discretization parameter: angle  $\chi$ . By referring to the Fig. 2.10 you understand the angle is not useful to identify the reference plane,  $\Delta$ . The angle  $\chi$  is helpful to scan the stress path which lies on the reference plane. As it is shown in the Fig. 2.9, the stress components, defined by Eq. (3.31) and (3.32), create a path on the plane  $\Delta$  when they are added. The stress path is called  $\Psi$  and thank to  $\chi$  we explore the entire stress path with the intent to find the most critical region.

Thanks to the averaged nature of the criterion, you want to evaluate the maximum value of  $T_a$  on each plane of our discretization. Henceforth, you average all the results from the different planes. For our purpose is not necessary to know the real path  $\Psi$ , yet you need to know the stress components which describe the discrete steps of the path. Translating this idea we need only the stresses components as  $Sr_{1,3}$  e  $Sr_{2,3}$ . Now, you project both components with  $\sin \chi$  e  $\cos \chi$  respectively, to obtain the point. In sake of clarity the notation  $Sr$  means "rotated stress tensor", the subscript 1, 3 and 2, 3 talks about of the components; consequently  $Sr_{1,3}$  is equal to  $\tau_{x,z}$  in the reference plane  $\Delta$  and  $Sr_{2,3}$  follows the same rule.

### 3.5.2 Solution

My first approach to the numerical problem is the following:

- **Step 1:** You identify the reference plane.
- **Step 2:** You set the value of  $\chi$ .
- **Step 3:** The code rotates the stress tensor in order to project its members in previous defined direction. The process is performed with the matrices product  $R^{-1} \cdot S \cdot R$ .
- **Step 4:** You evaluate the T part for every time step relating it with the angles.
- **Step 5:** On the plane, you extract the amplitude.
- **Step 6:** You repeat the steps from 1 to 5 until the end of the reference planes.
- **Step 7:** Numerical integration and average of method quantity.

If you write the mentioned steps, you will bump into some numerical problems. I want to remark the importance of the starting point of the code. Starting from a basic structure I am able to explain the improvements step by step.

The crucial points you will need to overcome are:



- Take an advantage of rotation matrix independence of  $\chi$  angle, in order to reduce global iterations.
- Symmetry of matrices.
- Reduction to the minimum the information evaluated. The code has to calculate the requested information only.
- Vectorization of code, deleting a cycle, particularly  $\chi$  cycle.

First improvement is to erase the inversion of rotation matrix. This change is performed in a different manner if compared with Mataka's theory. You prefer to separate the rotation matrix by writing a new formulation. The matrices coming from the latest approach describe the versors necessary for the rotation and they avoid inverse matrix calculation.

Let's step back and focus on the first code which needs three matrices and three angles:

$$R(\varphi) = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$P(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (3.4)$$

$$C(\chi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \chi & -\sin \chi \\ 0 & \sin \chi & \cos \chi \end{bmatrix} \quad (3.5)$$

The matrices structure needs three angles, and three loops to control them as well. Obviously, the matrix inversion is a dead-end way for our analysis. But it is correct to show the first implementation:

$$Sr(\varphi, \theta, \chi, t) = (R(\varphi) \cdot P(\theta) \cdot C(\chi))^{-1} \cdot S(t) \cdot (R(\varphi) \cdot P(\theta) \cdot C(\chi)) \quad (3.6)$$

This structure is a huge waste of computational time. The code is the same previous criterion but with a cycle for  $\chi$  variable. The time has no way to reduce itself.

### First improvement

The first step is to avoid matrix inversion in favour of versors. You will obtain:

$$Nor(\varphi, \theta) = \begin{bmatrix} -\cos \varphi \sin \theta \\ -\sin \varphi \sin \theta \\ \cos \theta \end{bmatrix} \quad (3.7)$$

which is the versor normal to  $\Delta$ ;

$$Lver(\varphi) = \begin{bmatrix} -\sin \varphi \\ -\cos \varphi \\ 0 \end{bmatrix} \quad (3.8)$$

which is the L direction in the local reference;

$$R_{ver}(\varphi, \theta) = \begin{bmatrix} \cos \varphi \cos \theta \\ \sin \varphi \cos \theta \\ \sin \theta \end{bmatrix} \quad (3.9)$$

which is R direction.

Thanks to these versors the three useful components are:

$$\begin{aligned} Sr_{1,3} &= S(t) \cdot Nor \cdot R_{ver} \\ Sr_{2,3} &= S(t) \cdot Nor \cdot L_{ver} \\ Sr_{3,3} &= S(t) \cdot Nor \cdot Nor \end{aligned} \quad (3.10)$$

The mathematical passages to find the formulations shown in Eq. (3.10) have been done by myself; but I prefer to neglect the passages because there are not useful information.

The most important result, I want to highlight, is the time response of the code. Time decreases from 75 *sec* per node to 35 *sec* per node. The improvement is not infinitesimal but the overall time for 100000 nodes is far from businesslike time.

### Second improvement

The analysis of the implemented code reveals the huge difference in the number of cycle if compared with Matake's code. Now, mesoscopic code works with one more variable  $\chi$ , that means you need one more cycle to control it. If I was able to avoid the inner space cycle, I would pair the overall time of the code with Matake's code.

Thanks to this improvement I manage time and the single iteration time plummets to 0.8 *sec* per node. This datum is closer to our target but not so good.

### Third improvement

With this third step you completely shake the code up. You leave the tensor rotation matrices and you trust in previous implementation success. I decide to use single stress components in this code as well. The mathematical steps, described in the Chapter 3.7, are absolutely correct for our purpose and those passages allow to directly find the result.

Now, the averaged normal stress is substitute for hydrostatic stress, as shown in the Eq. (2.23), the formula is the clearest you can write:

$$\sigma_H = \frac{S_{1,1} + S_{2,2} + S_{3,3}}{3} \quad (3.11)$$

On the other hand, the two stresses components which act on the reference plane will be calculated with Eq. (3.31) e (3.32). Furthermore, the components will be projected with  $\chi$  angle to find the path at the reference time step.

Actual code is slightly different. The distinction is not with Matake's code but with the order of the parameters. Before the code called  $\varphi$ ,  $\theta$ ,  $\chi$  at the end  $t$ . Now the order is changed, first the plane with  $\varphi$  and  $\theta$ , second the time, finally the angle  $\chi$ . The angle  $\chi$  is not organized with a loop but it is already vectorized. What I mean is that all the evaluations of  $\chi$  are performed at the same time. Thanks to this reshaping of code we obtain the same structure of the previous version but calculated with  $n = 26$  time steps and not with  $n \cdot o = 26 \cdot 51 = 1326$  global iterations;

where  $o$  are  $\chi$  discretization steps. After those loops you extract the real result of the method:  $T_a$ .

Let's focus on  $T_a$  structure. The resolved shear stress is a function of three variables,  $T_a = T_a(\varphi, \theta, \chi)$ , indeed it is described in the code with a 3-dimension matrix. You must fulfil all the structure with 132651 elements, if you consider our discretization composed by 51 step for each space vector. It is not possible to avoid complete matrix filling, the process could not be skipped because you need to integrate the function in the whole volume. It is not possible to compare the  $i - th$  step until you know the all the others.

The current core function is the following:

$$Stress_\chi(\varphi, \theta, t) = Sr_{1,3} \cdot \sin \chi + Sr_{2,3} \cdot \cos \chi \quad (3.12)$$

where  $\chi$  is the angle which scans the plane and the stress components are the same evaluated in Chapter 3.7.

### Final improvement

I take an advantage from parallel evaluation of nodes and from ParallelComputingToolbox, in order to solve more than one node at the same time. This step concludes the development for the mesoscopic approach and it allows to summarize the time improvements in the Tab. 3.5

Complete solver improvement					
	Starting time	First step	Second Step	Third step	Final step
Single iteration	75 s	35 s	0.8 s	0.1 s	0.021 s
1e+5 iterations	86 d	28 d	22 h	2.7 h	36 m

Table 3.5: Mesoscopic's theory code improvements.

### 3.6 Principal stress theory

In this section I will follow a different scheme to describe code improvement, because the development has been performed in few passages. As you can see in the Tab. 3.1 the single node time is already comparable with other solution times. Matake and mesocopic's solvers are as quick as principal stress code but they have already been developed, contrary, principal stress code is the first version.

Making a comparison between first and second code version there is an adjustment of post-process section of code, in order to reduce to the minimum the evaluations made by the code itself. A remarkable difference is the double criteria integrated inside the solver. There are two different approaches solved for each node, in the following sections I will explain the features of the data you can extract from the criteria.

#### 3.6.1 Own principal directions

The first step is the evaluation of the static stress tensor and the dynamic one. The code isolates two time steps in the time discretization and it determines the typical tensors as follow:

$$S_{stat} = \frac{S(t_i) + S(t_j)}{2} \quad (3.13)$$

$$S_{dyn} = \frac{S(t_i) - S(t_j)}{2} \quad (3.14)$$

where stress tensor  $S$  is a matrix belonging to  $\mathbb{R}^{3 \times 3}$ , it comes from the stresses acting on the FEM node. In this segment the code works in a wrong way with respect of the previous criteria. Furthermore, as input the function reads six stress components as a row and it recreates the complete matrix. You lose some time to perform the operation, you can write the formulation:

$$\begin{bmatrix} \sigma_{1,1} & \sigma_{2,2} & \sigma_{3,3} & \sigma_{2,1} & \sigma_{2,3} & \sigma_{3,1} \end{bmatrix} \implies S(t) = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \sigma_{1,3} \\ \sigma_{2,1} & \sigma_{2,2} & \sigma_{2,3} \\ \sigma_{3,1} & \sigma_{3,2} & \sigma_{3,3} \end{bmatrix} \quad (3.15)$$

You must arrange the data in the matrix because the rotation strategy is completely different. It is the newest strategy which imposes the matrix sorting out. In this particular circumstances stress tensor rotation is not performed thanks to a pre-existing plane. The rotation is performed according with "Mohr's circle" theory. The final intent is to evaluating principal directions and principal stresses acting on that point owing to applied global stress tensor. The Eq. (1.4) might estimate those directions but it is not our case. With the mentioned relation you are able to identify a versor, to relate it with a rotation matrix, in the end you apply rotation matrix to tensor  $S$  to find out the principal directions. Obviously, following the proposed steps you fall again into the previous code's numerical problems.

The principal directions settlement suggests travelling another way. Let's remind the definition: *a principal direction is defined as the way where the global applied stresses do not create shear stress on the orthogonal plane, yet it has a component in the main direction only*. When we have a global stress tensor structure, as it is shown in the Fig. 3.2, we extract an equal formulation only with the three main components. You want to describe stresses as:

$$S_{princ} = \begin{bmatrix} \sigma p_{1,1} & 0 & 0 \\ 0 & \sigma p_{2,2} & 0 \\ 0 & 0 & \sigma p_{3,3} \end{bmatrix} \quad (3.16)$$

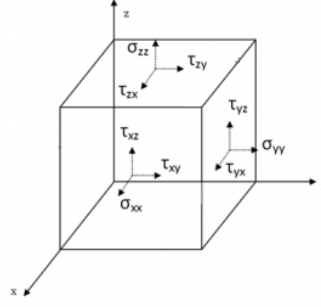


Figure 3.2: Cauchy stress tensor.

Dialoguing with algebraical rules you understand that the way to find the principal stresses are linked with matrix diagonalization. To find out the principal stresses and directions you need to evaluate eigenvalues and eigenvectors respectively.

You proceed reminding Eq. (3.13) and (3.14), after that you define two time steps, called  $t1$  and  $t2$ . Finally, we have the formulation of the static and dynamic tensors:

$$S_{stat} = \begin{bmatrix} \frac{\sigma_{xx}(t2) + \sigma_{xx}(t1)}{2} & \frac{\sigma_{xy}(t2) + \sigma_{xy}(t1)}{2} & \frac{\sigma_{xz}(t2) + \sigma_{xz}(t1)}{2} \\ \frac{\sigma_{yx}(t2) + \sigma_{yx}(t1)}{2} & \frac{\sigma_{yy}(t2) + \sigma_{yy}(t1)}{2} & \frac{\sigma_{yz}(t2) + \sigma_{yz}(t1)}{2} \\ \frac{\sigma_{zx}(t2) + \sigma_{zx}(t1)}{2} & \frac{\sigma_{zy}(t2) + \sigma_{zy}(t1)}{2} & \frac{\sigma_{zz}(t2) + \sigma_{zz}(t1)}{2} \end{bmatrix} \quad (3.17)$$

while alternate, or dynamic, stress tensor is:

$$S_{dyn} = \begin{bmatrix} \frac{\sigma_{xx}(t2) - \sigma_{xx}(t1)}{2} & \frac{\sigma_{xy}(t2) - \sigma_{xy}(t1)}{2} & \frac{\sigma_{xz}(t2) - \sigma_{xz}(t1)}{2} \\ \frac{\sigma_{yx}(t2) - \sigma_{yx}(t1)}{2} & \frac{\sigma_{yy}(t2) - \sigma_{yy}(t1)}{2} & \frac{\sigma_{yz}(t2) - \sigma_{yz}(t1)}{2} \\ \frac{\sigma_{zx}(t2) - \sigma_{zx}(t1)}{2} & \frac{\sigma_{zy}(t2) - \sigma_{zy}(t1)}{2} & \frac{\sigma_{zz}(t2) - \sigma_{zz}(t1)}{2} \end{bmatrix} \quad (3.18)$$

Evaluating the diagonal matrix, thanks to eigenvalues, you obtain two matrices, each one rotated in relation with their own main directions.

Our goal is to find the worst condition ever, obviously between the computed eigenvalues. I defined as worst condition the couple of stresses, one static and one dynamic, which is composed by the maximum values of each triplet. Regarding dynamic part the code extracts the maximum modulus as follow.

$$\sigma_{dyn} = \max[|\sigma_{1,dyn}|; |\sigma_{2,dyn}|; |\sigma_{3,dyn}|] \quad (3.19)$$

It is well known that the amplitude is always a positive value. On the other hand, the static part extraction is performed considering sign:

$$\sigma_{stat} = \max[\sigma_{1,stat}; \sigma_{2,stat}; \sigma_{3,stat}] \quad (3.20)$$

### Main features

The mentioned formulas have some positive features but also some drawbacks. First, the dynamic stress is completely incoherent with the static one. The extractions are fully independent. After

tensor creation they proceed into totally different ways, there is not mutual influence. You can explain this choice thanks to two key points: safety margin and genesis of fatigue behaviour.

Regarding safety, you assert evaluating different maximum values we are searching the worst working condition of dynamic stress tensor and of static one. Thereby the sum of the worst situations creates the weariest cycle ever, if we consider all the combination with the six eigenvalues. Evaluating the most demanding working cycle we are overestimating the stresses. If you overestimated the stresses, the code might provide a shorter and shorter life of the part. Thanks to this approach you have a greater safety margin. The drawback of this technique is the safety margin itself. Sometimes, you uncover a heavy cycle, which is too much far from the real stress situation. The underestimation of part life is too much high and you could not accept this margin.

Talking about second point, I am sure you will find the answer in Chapter 5. Over there we will see that in the critical areas the methods are in perfect match. Fatigue is a phenomenon strictly related to the skin of the body, if we are thinking about crack enucleation. Moreover, the stress intensity factor reaches the highest values near holes, sharp edges or narrow radii of connection. All this areas are located in boundary part of the body, you can assert in this location the geometry of the body constrains principal directions. Thanks to this statements we understand why this approach and the following provide same values for critical areas.

### 3.6.2 Dynamic tensor principal direction

High cycle fatigue behaviour is connected with stress amplitude. The dynamic stress is the real director of the part behaviour. You can correctly think that it defines the most stressed principal direction.

You start from the same step of the previous approach. The code evaluates static and dynamic tensors, each one between time steps  $t1$  and  $t2$ .

You continue with the diagonalization of the dynamic tensor, it must be the same of the evaluated before one. Now, the code extracts the eigenvectors and the eigenvalues. As described before, the eigenvalues are the three scalar values which describe the dynamic principal stresses. Conversely, the eigenvectors provide the rotation matrix which is applied to the static tensor. The eigenvectors create the *directional cosines matrix* and thanks to this matrix the code is able to rotate the static tensor. The rotation brings the static tensor, which is described in global reference frame, and it projects it over dynamic tensor principal directions. The directional cosines matrix is the following:

$$R = \begin{bmatrix} \cos(xx_1) & \cos(xy_1) & \cos(xz_1) \\ \cos(xx_2) & \cos(xy_2) & \cos(xz_2) \\ \cos(xx_3) & \cos(xy_3) & \cos(xz_3) \end{bmatrix} \quad (3.21)$$

At this point of the calculus you identify the maximum dynamic stress. The identified direction supplies the rotation of the static part. The code creates  $R$  and it perform the rotation:

$$S_{stat,Xdyn} = R^T \cdot S_{stat} \cdot R \quad (3.22)$$

where it is taken an advantage from matrix property, orthogonality property for completeness.

At this point you know two stress components, both static and dynamic, which act in the same direction, the dynamic one.

In sake of clarity, the static tensor might be not diagonal. The code does not know static principal directions a priori, but its is not a problem. The real aim of this technique is the

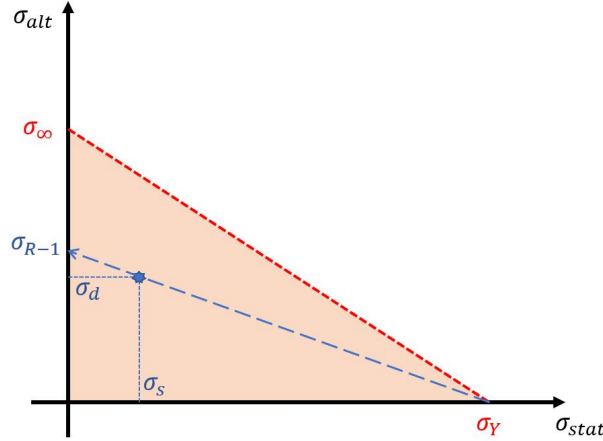


Figure 3.3: Goodman-Haigh diagram, with Goodman correction.

compliance of the directions between dynamic and static stress tensors. Obviously, it is the dynamic one which directs the behaviour, consequently it is the main character of our study.

### 3.6.3 Goodman correction

In the previous description I talked about two time steps, but if you want to consider a time history, composed by more than two steps, you need a rule to relate different steps, in order to understand the worst one. In this evaluation is really helpful the *Goodman-Haigh* diagram, described in Fig. 3.3.

The procedure is necessary because the material characterization test is performed with a fixed Stress Ratio. The selected value is equal to  $R = -1$ , that means zero static stress. In order to correctly relate numerical data with experimental ones we need to translate the numerical data bringing R value to  $-1$ . Goodman says: it is enough to trace on the diagram the line which passes through the numerical point and the yielding point. The crossing value between line and ordinate axis is Goodman corrected dynamic value. This is the value I will compare with the experimental datum. Numerically, it is not necessary to physically draw the line, you write the equation, rearrange it and perform the evaluation as follow:

$$\begin{aligned} \frac{\sigma_{R-1}}{\sigma_Y} &= \frac{\sigma_{alt}}{\sigma_Y - \sigma_{stat}} \\ \sigma_{R-1} &= \frac{\sigma_Y \sigma_{alt}}{\sigma_Y - \sigma_{stat}} \end{aligned} \quad (3.23)$$

Dealing with the same node, the code estimates the Goodman correction for each couple of time steps. After, it finds the worst value out, which will be the worst condition on reference node.

There is a blind spot which needs more attention. I am talking about nodes with negative static stress. If a fatigue cycle has a negative mean value, you will find a longer life if compared with the same cycle with  $R = -1$ . This fact is explained by the higher and higher percentage of compressive cycle, percentage which grows with greater negative mean stress. To solve this problem you can use one of these three options:

1. You flip the Fig 3.3 in the second quarter and you correct the equation of the correction according to the new graph. Thanks to this method we find that two nodes with the same

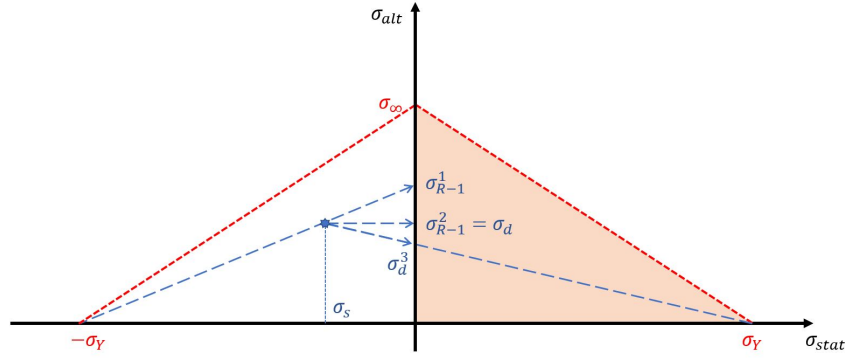


Figure 3.4: Goodman-Haigh diagram with negative mean stresses and with three corrections.

dynamic stress but with opposite static stress reaches the same Goodman correction. This idea is totally wrong if compared with what I just said before. You loose the beneficial effect of the compressive state.

2. You consider open Goodman-Haigh diagram, that means neglect negative static stress. Indeed, you do not correct the dynamic stress.
3. You correct the point with the same relation used in positive stress case. In this case you are trying to estimate beneficial effect of compressive stress.

The three approaches provoke three different corrections of  $\sigma_{R-1}$ , and the Fig. 3.4 is an example.

As you can see from the chart, the first approach produces an increasing of the dynamic data, the third a decreasing, the second is neutral. Considering the transparency of the second method I decide to use an open Goodman-Haigh diagram to find the correction out. My choice is questionable because I am not considering compression state, the aim of this choice is the sake of safety.

A numerical example is written in Tab. 3.6:

Goodman correction $\sigma_Y = 400MPa$			
	Method 1	Method 2	Method 3
$\sigma_s + \sigma_d [MPa]$	$-100 + 250$	$-100 + 250$	$-100 + 250$
$\sigma_{R-1} [MPa]$	330	250	200

Table 3.6: Different Goodman correction approaches.

### 3.6.4 Calculus time-analysis

As described in the starting lines of the reference section, the development of the code has been performed with less steps than other criteria. Eigenvalues and eigenvectors evaluation is performed by MATLAB with an efficient algorithm. In the first period of the development, I



made some comparisons with existing codes in order to validate the results of my solving function. Moreover I modified the structure to delete useless steps in order to reach the best performance.

Principal directions approach time			
	First step	Development	Parallel computing
Single iteration	0.51 s	0.2	0.007
1e5 iterations	14 h	5 h	12 min

Table 3.7: Calculus time evolution with principal stresses approach.

As you can see in Tab. 3.7 the best step is performed by ParallelComputinToolbox<sup>TM</sup>, due to MATLAB internal optimization. This short time of calculus allows me to implement both approaches in the same function. Thanks to this you can immediately relate the result of the methods in the output graphs.

### 3.7 Analytical development

In this section you need to remind the Fig. 2.10 in order to better understand the meaning of the angles. In the Tab. 3.8 I describe useful parameters:

Discretization			
$\varphi \in [0, 2\pi]$	$\theta \in [0, \pi]$	$\chi \in [0, 2\pi]$	$t \in [0, 1]$
50	50	50	30

Table 3.8: Standard discretization parameters adopted.

In order to rotate the tensor in  $O.lnr$  directions, I use a classic rotation matrix. The matrix is a 2-variable function; the variables are  $\varphi$  and  $\theta$ :

$$RP = \begin{bmatrix} \cos \theta \cos \varphi & -\sin \varphi & -\sin \theta \cos \varphi \\ \cos \theta \sin \varphi & \cos \varphi & -\sin \theta \sin \varphi \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (3.24)$$

To find the analytical components out you need to rotate the starting matrix using the following expression:

$$RP^{-1} \cdot S \cdot RP \quad (3.25)$$

First, you need to know the inverse matrix of the rotation one, you can follow the definition:

$$RP^{-1} = \frac{1}{|RP|} \cdot RP_{gg}^T \quad (3.26)$$

the determinant of  $RP$  will be:

$$\begin{aligned} |RP| &= \cos \theta^2 \cos \varphi^2 + \sin \theta^2 \sin \varphi^2 + \sin \theta^2 \cos \varphi^2 + \cos \theta^2 \sin \varphi^2 \\ &= \cos \theta^2 (\cos \varphi^2 + \sin \varphi^2) + \sin \theta^2 (\cos \varphi^2 \sin \varphi^2) \\ &= 1 \end{aligned} \quad (3.27)$$

The adjugate matrix evaluation, necessary to perform inversion, comes from definition itself. I describe it with my own words as follow: it is the product between  $-1^{i+j}$  and the determinant of the minor of row  $i$  and column  $j$ . Applying the definition to the first element you will find:

$$RP_{gg}^{1,1} = -1^{1+1} \cdot \begin{vmatrix} \cos \varphi & -\sin \theta \sin \varphi \\ 0 & \cos \theta \end{vmatrix} = \cos \varphi \cos \theta \quad (3.28)$$

Now, I perform the calculus for all the elements of the matrix to write  $RP_{gg} = RP$ , that means  $RP^{-1} = RP^T$ . This statement perfectly matches what you know about rotation matrices and it allows to say that RT is orthogonal.

Before the calculation of the first product shown in Eq. (3.25), I need to know which are our target components. I want to calculate the only stress components which Mataka's and mesoscopic theories require. Fortunately, the terms are the same for both theories, even if they will use in a different way inside the code. The components are:  $Sr_{1,3}$ ,  $Sr_{2,3}$  e  $Sr_{3,3}$ , that represent the last one column of whole rotated tensor.

The last one column is performed with the product between intermediate matrix and the last one column of rotation matrix. Let's call the intermediate matrix as  $RPS$  and start with the evaluation.

$$\begin{aligned}
 RP^{-1} \cdot S &= RP^T \cdot S = \\
 &= \begin{bmatrix} \cos \theta \cos \varphi & \cos \theta \sin \varphi & \sin \theta \\ -\sin \varphi & \cos \varphi & 0 \\ -\sin \theta \cos \varphi & -\sin \theta \sin \varphi & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} \quad (3.29)
 \end{aligned}$$

where:

- $RPS_{1,1} = S_{1,1} \cos \theta \cos \varphi + S_{2,1} \cos \theta \sin \varphi + RPS_{3,1} \sin \theta$
- $RPS_{1,2} = S_{1,2} \cos \theta \cos \varphi + S_{2,2} \cos \theta \sin \varphi + RPS_{3,2} \sin \theta$
- $RPS_{1,3} = S_{1,3} \cos \theta \cos \varphi + S_{2,3} \cos \theta \sin \varphi + RPS_{3,3} \sin \theta$
- $RPS_{2,1} = -S_{1,1} \sin \varphi + S_{2,1} \cos \varphi$
- $RPS_{2,2} = -S_{1,2} \sin \varphi + S_{2,2} \cos \varphi$
- $RPS_{2,3} = -S_{1,3} \sin \varphi + S_{2,3} \cos \varphi$
- $RPS_{3,1} = -S_{1,1} \sin \theta \cos \varphi - S_{2,1} \sin \theta \sin \varphi + S_{3,1} \cos \theta$
- $RPS_{3,2} = -S_{1,2} \sin \theta \cos \varphi - S_{2,2} \sin \theta \sin \varphi + S_{3,2} \cos \theta$
- $RPS_{3,3} = -S_{1,3} \sin \theta \cos \varphi - S_{2,3} \sin \theta \sin \varphi + S_{3,3} \cos \theta$

Now, you identify necessary and sufficient components of the method:

$$RPS \cdot RP = \begin{bmatrix} RPS_{1,1} & RPS_{1,2} & RPS_{1,3} \\ RPS_{2,1} & RPS_{2,2} & RPS_{2,3} \\ RPS_{3,1} & RPS_{3,2} & RPS_{3,3} \end{bmatrix} \begin{bmatrix} RP_{1,1} & RP_{1,2} & RP_{1,3} \\ RP_{2,1} & RP_{2,2} & RP_{2,3} \\ RP_{3,1} & RP_{3,2} & RP_{3,3} \end{bmatrix} \quad (3.30)$$

and you extract final formulation:

•

$$\begin{aligned}
 Sr_{1,3} &= -S_{1,1} \cos \theta \sin \theta \cos^2 \varphi + \\
 &\quad -2S_{2,1} \cos \theta \sin \theta \cos \varphi \sin \varphi + \\
 &\quad -S_{2,2} \cos \theta \sin \theta \sin^2 \varphi + \\
 &\quad -S_{3,1} (\sin^2 \theta - \cos^2 \theta) \cos \varphi + \\
 &\quad -S_{3,2} (\sin^2 \theta - \cos^2 \theta) \sin \varphi + \\
 &\quad +S_{3,3} \cos \theta \sin \theta \quad (3.31)
 \end{aligned}$$

•

$$\begin{aligned}
 Sr_{2,3} &= (S_{1,1} - S_{2,2}) \sin \theta \cos \varphi \sin \varphi + \\
 &\quad +S_{2,1} \sin \theta (\sin^2 \varphi - \cos^2 \varphi) + \\
 &\quad -S_{3,1} \cos \theta \sin \varphi \\
 &\quad -S_{3,2} \cos \theta \cos \varphi \quad (3.32)
 \end{aligned}$$

•

$$\begin{aligned}
 Sr_{3,3} = & -S_{1,1} \sin^2 \theta \cos^2 \varphi + \\
 & + 2S_{2,1} \sin^2 \theta \cos \varphi \sin \varphi + \\
 & + S_{2,2} \sin^2 \theta \sin^2 \varphi + \\
 & - 2S_{3,1} \cos \theta \sin \theta \cos \varphi + \\
 & - 2S_{3,2} \cos \theta \sin \theta \sin \varphi + \\
 & + S_{3,3} \cos^2 \theta
 \end{aligned} \tag{3.33}$$

You want to guarantee the perfect parallelism with  $N, C, T$ , components, which are described in Chapter 2.3. The component  $Sr_{3,3}$  is referred to the versor  $n, n$  which corresponds to  $z, z$ , that is the situation with  $(\varphi, \theta) = (0, 0)$ . Indeed, I am talking about stress component orthogonal to the reference plane, which will be the hydro statical part of the implemented criteria. The remaining components are the stresses which act on the reference plane. Particularly,  $Sr_{1,3} = \tau_{x,z}$  that is  $\tau_{r,n}$  in *O.rln* reference system; and  $Sr_{2,3} = \tau_{y,z}$  becomes  $\tau_{l,n}$ . The sum of the aforementioned plane stresses creates the path described in the Fig. 2.8.

### 3.7.1 Remarks

These expressions allow a huge time saving owing to two reasons:

1. Evaluation of necessary and sufficient components only. You reduce the assessed data by 66% passing from 9 to 3.
2. MATLAB *Back slash* does not perform a matrix inversion. It analyses the type of matrices passed as input and it factorizes in order to solve the associated linear system. This process is burdensome and, whenever is possible, you have to avoid it.

In order to obtain the fastest solution I analyse the number of elementary operations done by the code to evaluate the stress components. I remodel the formulas to reduce the number of basic operations, I obtain a time saving of  $0.6\text{ms}$  for each analysed node. This time is close to be negligible even if I compare it with 100000 nodes, but the cost of this development is free from loss time than I decide to remodel the formulations with trigonometrical rules.

The earning deals with the number of operations and evaluations the code must do to find the result. The idea is the following:

$$\cos^2 \theta - \sin^2 \theta = \cos \theta \cos \theta - \sin \theta \sin \theta = \cos 2\theta$$

The first term is the slowest one due to power operation, in fact you have: two function evaluations to pull out the value of  $\theta$  from the vector which stores all the values and five arithmetical operations. The operations are: trigonometrical evaluation, sum and product. Second term has a higher number of operations than the first one, but the overall time is lower because there are basic operations. The power calculus is slower than the product. The last one step is the fastest, it is composed by basic arithmetical operation and two function extractions. You can move the objection related to the creation of the vector  $2\theta$  which erases time saving. In my codes the vector is created once and the formula is evaluated thousand and thousand times for each node. The preliminary part of each function performs vectors calculation, in the preprocess

the code prepares space discretization vectors and time steps before starting the real evaluation. Preprocess time is negligible if compared with global time of calculus.

Another advantage generated from this implementation is RAM saving. With the intention to evaluate the rotated tensor with the old formulation, i.e.  $RP^{-1} \cdot S \cdot RP$ , you need to write the tensor as a  $3 \times 3$  matrix, wasting space because the matrix is symmetrical. Moreover, MATLAB does not store the matrices as we can see on the graphic interface, but it stores as a column. If you want to extract position 1,2 of the matrix, MATLAB translates the double indices in a unique one in order to correctly access the RAM and to satisfy your request. At this time of the development, I know the analytical formulas and it is useless to save the whole matrix, I store only the six different components, in a column vector, in order to maximize the time management of the code. The Eq. (3.34) shows what I was talking before.

$$\begin{bmatrix} Sr_{1,1} & Sr_{1,2} & Sr_{1,3} \\ Sr_{2,1} & Sr_{2,2} & Sr_{2,3} \\ Sr_{3,1} & Sr_{3,2} & Sr_{3,3} \end{bmatrix} \implies [Sr_{1,1} \quad Sr_{1,2} \quad Sr_{2,2} \quad Sr_{1,3} \quad Sr_{2,3} \quad S_{3,3}] \quad (3.34)$$

In the chapter dedicate to Graphical User Development I will describe the possibility of sorting the vector in order to let the user more flexibility in the input structure.



## Chapter 4

# Graphical interface development

### 4.1 Introduction

As described in the Abstract I need to take into account two different types of input. These inputs will be translated into two different way in order to better perform the calculus. Indeed I create two different Graphical User Interface, each one works on a specific input type. The main solver, inside both GUIs is the same but the pre process stage which creates the correct structure for the solver is very different. In this Chapter we will see the differences. However, as shown in the Chapter 3, it is possible to develop a unique solver which will be the same for both GUIs.

### 4.2 Problem first approach

The inputs to deal with come from Finite Element Theory models of the part. The widespread use of this analysis brings engineers to have a large amount of data referred to the component you want to study. These information are variegated, you can think about all the different analysis you can perform with a FEM software. For our purpose the useful solution is the static one. Inertial effects are negligible if compared with other effects. You can assert the load frequency is far from critical frequency of the part. Thanks to this statement you assert the static solution is a correct approximation of dynamic solution. Previous lines find a support in the strategy used to relate a random load time history with the load to apply in FEM model. Taking as reference the Chapter 2.2.2, the department developed a own methodology which mixes some of the mentioned Peak-Counting methods. The baseline is the safety, you could not tolerate an error in this preliminary phase of the analysis. The most useful cycles are:

1. **Start-Stop:** Cycle with  $R \sim 0$  with maximum value equal to the absolute maximum stress.
2. **High Frequency:** Variable cycle linked with experimental data and test flight data.

I argue that the most demanding cycle is the low-frequency cycle, this fact confirms what I said before about dynamic analysis. Moreover, these statements are correct for both inputs:

- **Sensitivity matrices**
- **Stress tensor**

Let's analyse both cases in order to better understand what does it mean choosing one or the other.

### 4.3 Sensitivity matrices

This matrices are displayed with the same shape and dimension equal to a stress tensor. They are built with 9 components with the symmetry property which is the main feature of all stress tensors. If the loads time history is composed by different components you need to weight each other the components. This is the genesis and the reason of the importance of these matrices. With the word “different” I mean with different phase, frequency and stresses. Each sine function, used to describe the load’s variation in time, is composed by mean and alternate values, moreover by frequency and initial phase. As we can see in the Fig. 5.2, you understand how difficult is to understand which is the correct critical condition of the whole load spectrum. It is fruitful to have a matrix which weight the importance of the single load component compared with the others.

How can I evaluate these matrices?

1. You model the part with FEM code, or the assembly you want to study.
2. You assume a static load for each one load component.
3. You run the analysis to figure the matrices out.

The previous steps are extremely simplified if you want to evaluate the matrices, but for our aim is sufficient because the matrix evaluation is beyond the scope of this thesis. The most important aspect is the correct application of these matrices, in fact the first data you want to analyse from FEM code is the load used in the run. Thanks to Tab. 4.1 you can see the used loads in the different runs.

Loads on FEM model						
	Centrifugal	Beam	Chord	PLink	Damperin	Damperout
1	190000	0	0	0	0	0
2	190000	10000	0	0	0	0
3	190000	-10000	0	0	0	0
4	190000	0	10000	0	0	0
5	190000	0	-10000	0	0	0
6	190000	0	0	10000	0	0
7	190000	0	0	-10000	0	0
8	190000	0	0	0	10000	0
9	190000	0	0	0	-10000	0
10	190000	0	0	0	0	10000
11	190000	0	0	0	0	-10000

Table 4.1: FEM model loads used to evaluate sensitivity matrices.

The table allows more reasoning:

- **Static load:** is the normalization load used in order to create matrices with FEM, it must be used as scale factor to reach the correct stress state inside the structure.
- **Linear solving:** is not always the correct way to find the best solution. Indeed, on our part there are six load components and the solver needs to evaluate their importance a



priori. Among components, you identify the centrifugal one as the most powerful. This has a key role because of its magnitude, the magnitude of the global load over the part is linked with the centrifugal component. This fact forbid the independent evaluation of the matrices. In each run we load the body with centrifugal and one of the other loads, in the end a run with the only centrifugal load is made. To figure the correct matrices out you subtract the centrifugal sensitivity stress matrix from the others, in order to maintain the own load.

In sake of clarity, if you evaluate each load by itself, without considering centrifugal force, you will commit a huge error due to linear application of loads. It is necessary to proceed as described to avoid error accumulation.

- **Linear application:** is linked with the formulae described in the Abstract. The formula is the following:  $\sigma_{\mathfrak{R}} = \sigma_{sens} \cdot F_{\mathfrak{R}}/F_{sens}$ . I assert that in order to guarantee the linear application of matrices you must evaluate them as described before. Additionally, the reference loads have to be of the same magnitude of the real loads.

The matrices, I will use in the test case, are already calculated respecting previous suggests and I need only to import and scale them.

Another complication comes from the sign of the applied load, indeed when the load passes from positive value to a negative one the code must change the reference sensitivity matrix. This idea is linked with the asymmetrical behaviour which a component shows when it is stressed with positive or negative load. Thinking about thrust variation of the main rotor, quick increase of trust are permitted, conversely the decrease is slower than increase. Due to the previous lines, you have to evaluate two matrices for each load type, one for positive load and the other for negative one. There is an exception to this rule, the centrifugal component is always greater than zero, its sensitivity matrices it is only one.

At the beginning of the code, it does not know how many matrices you will import. Moreover the code does not know if the centrifugal component is the only one with the single matrix structure. Owing to this unknown data I am not able to preallocate the variables need to store all the matrices information. I have developed a strategy which works with the matrix names which is able to understand if a matrix is a single or double. This feature implement two powerful aspects:

1. The code is able to access the correct matrix with the name in order to associate correct load time history, and the stress time history is evaluated associating load name and sensitivity matrix name; it is not performed associating matrices or vectors.
2. The number of matrices importable is not a constrain of the method.

Before continuing I want to briefly describe the load in Tab 4.1:

- **Centrifugal:** most prevalent load in term of magnitude, it grows from rotation of the main rotor, the positive sign is traction.
- **Beam:** acts orthogonally to the rotor plane, is linked with flapping mode. Positive sign is upward.
- **Chord:** acts in the plane of the rotor, it is born due to lead-lag motion. The positive direction is which goes from the leading edge to the trailing edge of blade.

- **Pitch Link:** is an upward component, it is related with the pilot control who modifies the blade pitch.
- **Damperin:** is a damping component which comes from the following blade.
- **Damperout:** is a damping component which is born form the previous blade.

#### 4.3.1 The graphical interface

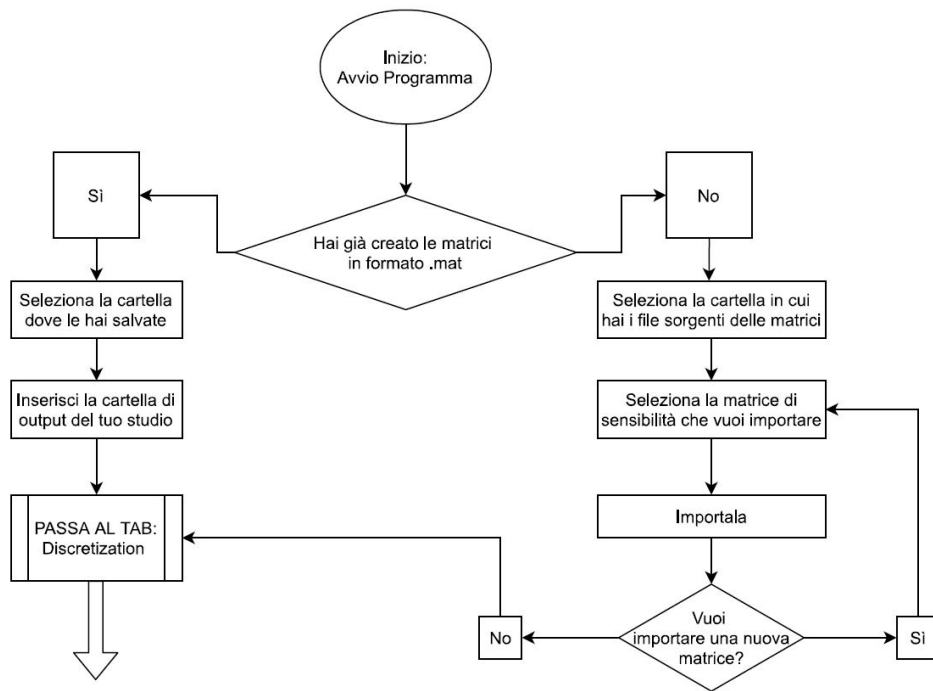


Figure 4.1: Preliminary section flowchart of the GUI.

The Graphical User Interface I completely develop in MATLAB language is split in different working steps. These steps are:

1. **Pre-process:** is the first stage and it is divided into subsections: import data, setting of useful parameters (i.e. discretization parameters), import of load time history and stress time history evaluations.
2. **Calculus:** is the stage where you actively solve the implemented criteria.
3. **Post-process:** is the stage where you analyse the results. You can see the damage tolerance evaluation dealing with a material database. Additionally you can create a binary file attachable into FEM software, in order to visualize the results as a chart over your model.

#### 4.3.2 Pre-processing activities

During pre-process stage I have divided the task into 4 different “tabs”, you need to carry four steps out before continuing. These steps are the following:

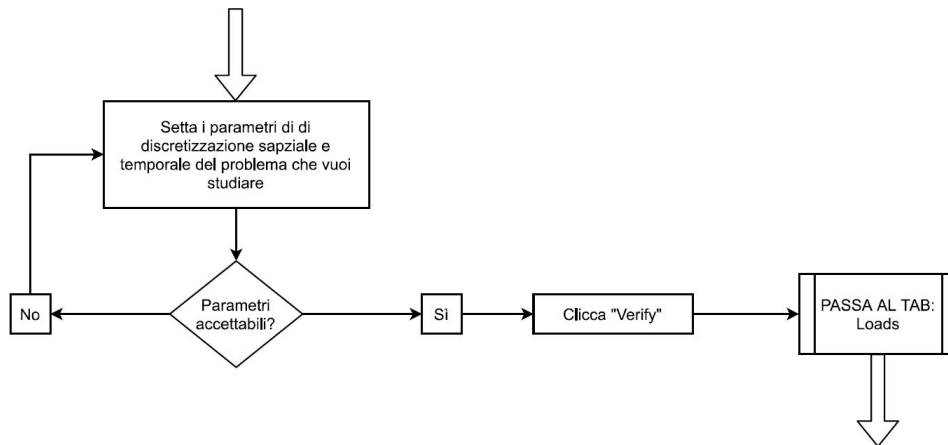


Figure 4.2: Discretization step flowchart.

1. **Import:** stage where the user selects and imports the sensitivity matrices. If there is a directory with already imported matrices the user selects that folder and skips the import section. Fig. 4.1 gives you a scheme of the required passages.
2. **Discretization:** useful parameters setting stage. Fig. 4.2 shows the process.
3. **Loads:** the user gets loads and saves them. There are different ways to perform this step as described in Fig. 4.3.
4. **Stress:** The code evaluates the stress time history thank to sensitivity matrices and load time history. The stage is described in Fig. 4.4.

As you can see in Fig. 4.1, the first step sets the working directory, both in input and output, in order to let the code loading and writing the input and output files. The key idea is related with the matrices which are the most important actors in this analysis. The idea is: one set of matrices correspond to one working folder. Inside of this main folder there is the input directory and the result directories. This structure is born because with one set of matrices you study lots of cases. You can vary discretization, loads, materials; the possibility are practically countless. The main folder is a global box where you can find the matrices and all the analysis you made before as well. On the other hand, if you want to import new matrices, the code creates the new file in the “input” subfolder and you work on the loop described in the flowchart. The scheme of the loop is basic and user-friendly, it hides lots of controls the code makes avoiding errors in the input files. The code controls the imported matrix sizes and values. The implemented checks are: uniqueness of node-identification numbers, sorting of rows in order to avoid import problems linked with the order of the origin files, check of the stress components. All of this steps are hidden to the user and they print an error with useful diagnostic messages. For example, if there are not unique nodes, due to an export error from the FEM code, the GUI detects the glitch, it shows the message and it automatically prints a file with useful information to identify the doubled lines.

Most of this steps are completely useless if you relate them with the results quality. In my opinion they are significant to validate the final result and to be sure that the final values are correctly evaluated.

The discretization stage, shown in Fig. 4.2 is a quick phase. You set spacial parameters and time steps. The specific weight of these few numbers is significant, indeed the time steps

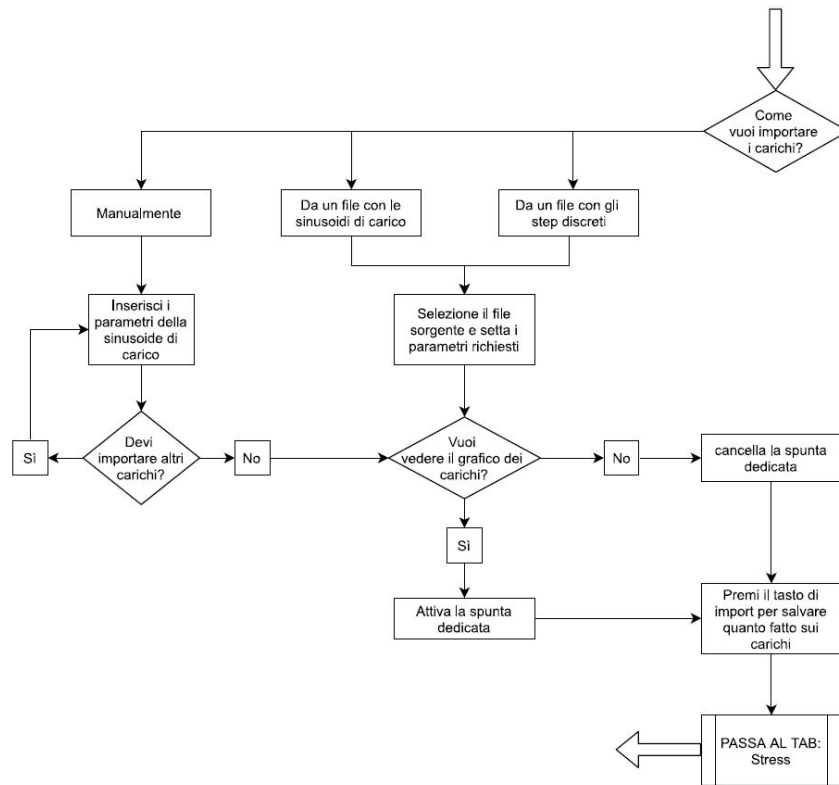


Figure 4.3: Load time history import step flowchart.

have to take into account because the loads samples is defined with this parameter. Moreover, the loads samples is strongly related with loads frequency. If you have a high frequency load (e.g. 10 Hz) and you set few time steps, you will not be able to correctly describe the sine motion. The accuracy of the entire calculus process is already compromised in the input stage. Furthermore, space discretization affects the overall calculus time of the method which uses that kind of discretization, i.e. Matake's and mesoscopic theories.

Moving on loads import tab, you have 3 ways to achieve this task:

- **Manual:** you insert the data of the sine function in the dedicated panel and you save them. The new load appears in the underlying table.
- **From file:** you save the data into an *.xlsx* file and you import the file with all the data in one shot.
- **From flight load:** thanks to an external file you import discrete points which describes the sine with steps, not with an analytical function. The loads may come from flying test, experimental data or previous calculus.

Whatever way you decide to travel to achieve the import the code creates the same output structure. It creates a vector as long as the time discretization parameter imposes, in order to guarantee the maximum control to the user. In the end of the selection of the load there is the import button which definitely store the data inside the GUI and, if you want, it plot the graph with the loads you have imported before. Last one stage is the creation of the real stresses. In this step the sensitivity matrices meet the load time history. The stress time history is created in

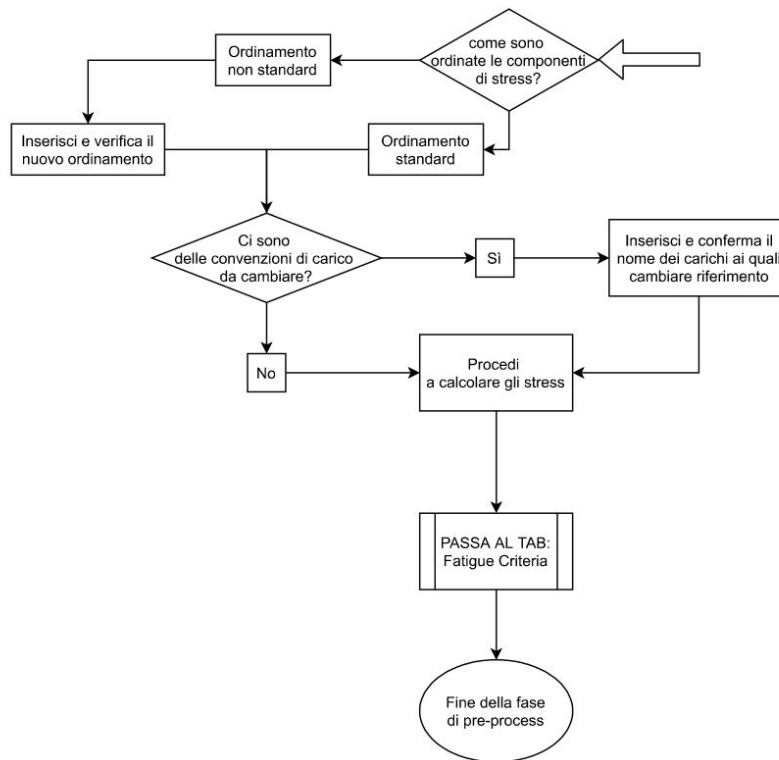


Figure 4.4: Stresses time history step flowchart.

few seconds of calculus and in an absolutely independent way from the user. The code is able to access the correct matrix thanks to load name and to load sign. In this tab are implemented two different panels which are described with the two questions shown in the flowchart in Fig. 4.4. The first question asks the stress components order of the imported files. This box is useful because when you export the stress with a FEM program it saves the matrices with its own idea; the idea is different for each FEM solver. Thanks to this panel you can insert the order of the imported files whatever is your FEM code. Obviously, there is default order which is the mainly used. The second request is referred to the sign convention. There are some cases, i.e. experimental tests, where the laboratory sign convention is opposite to model one. Maybe you want to change the sign of one ore more load components. Thanks to this panel you can write the name of the load you want to analyse with opposite sign. In the test case I studied the Pitch Link component falls into this case. You insert the real load of the Pitch Link in the load import stage, after you write the Pitch Link name inside the panel and the code flips the sine in the coherent way of your input.

### 4.3.3 Calculus

The calculus section asks you to set few parameters, particularly:

- Fatigue criteria you want to analyse.
- Which nodes you want to solve: you can solve all nodes or a restricted group. The restricted group is useful to local detailed analyses.

- Loop strategy: you want to use a classical for loop or you prefer to take an advantage from parallel evaluation.
- The output you want to save: numerical and graphical output are allowed.

After this phase the code activates the solve button, when it is pushed a window is opened and you need to insert material properties before continuing. The request is method dependent, indeed the code asks you:

- **Matake:** taking in account the Eq. (3.1), you need fully reversal bending stress limit and fully reversal torsion stress limit.
- **Mesoscopic:** taking in account the Eq. (2.24), you need fully reversal bending stress limit and fully reversal torsion stress limit.
- **Principal stresses:** the code uses the Goodman-Haigh diagram, it needs:  $\sigma_Y$  of yielding,  $\sigma_\infty$  of fully reversal fatigue stress limit and reduction factor  $Ks$ .

after that the code starts the evaluation of the chosen criterion on the selected nodes. In the end of the analysis it prints the required results.

All the post process section will be described in the Chapter 4.5 because it is equal between GUIs.

## 4.4 Discrete step solution

The name which describes this GUI is a bit deceptive respect to the previous interface. Inside all the solvers there is a discretization stage, as I described before you can set your own parameters for the analysis, you decide how many steps you want inside your time slice and the code split it according with your selection. In this GUI this is the main feature you will loose. In this particular case the stress state is the input of the problem it is no more evaluated inside the GUI. Different genesis of the input create the degrees of separation between GUIs.

### 4.4.1 Stress tensor

Stress tensor is the fastest way to obtain the necessary information useful for fatigue criteria evaluation. It is faster than sensitivity matrices approach, because the pre-process step is promptly complete. Indeed, the section linked with FEM discretization and loads are let to the analysts which models the part with the FEM code. At the end of the FEM analysis you extract the stress tensor evaluated in the element nodes. The FEM code manipulates the data with interpolation strategies in order to find the node stress, after that the structure of the exported file is similar to the sensitivity matrices. However, it contains a totally different type of data.

The load spectrum is totally hidden in this GUI, it is inside FEM model; indeed the import of this files is faster and simpler than sensitivity matrices. Obviously, the relative uncomplicated import phase involves into a absolutely stiffness of the solution. A load modification implies a FEM manipulation, a new solution of the entire FEM component. On the other hand with sensitivity approach you change the sine function in the dedicated tab then the code re-calculate the stress with a lost of time which is similar to zero if compared with FEM solving time.

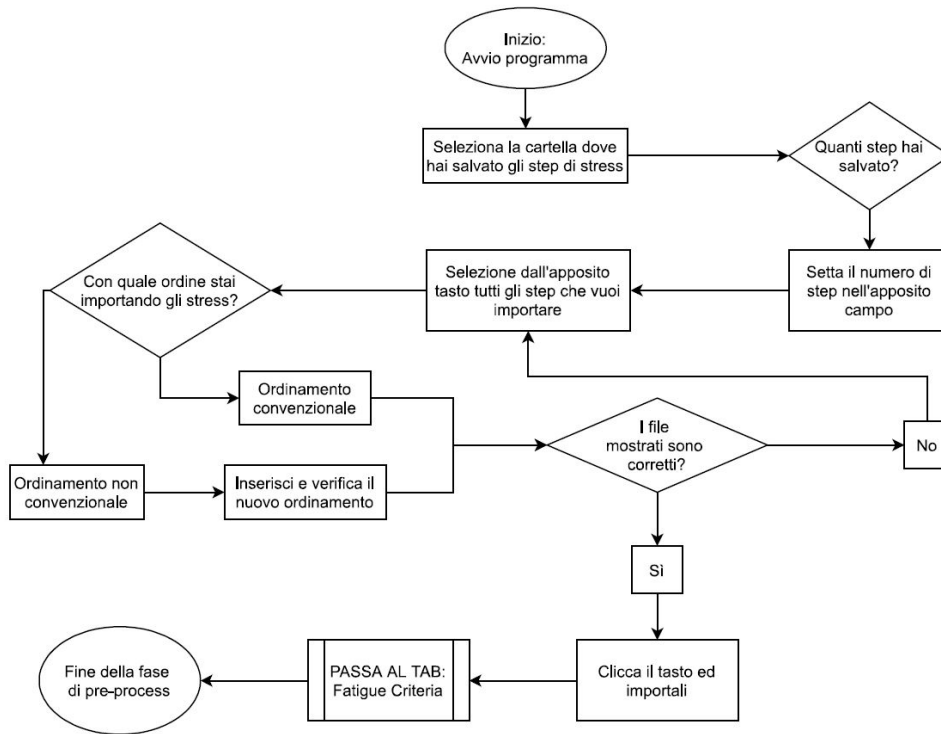


Figure 4.5: Reduced analysis import stage flowchart.

#### 4.4.2 The graphical interface

The interface pre-process stage is described in the Fig. 4.5 and it follows the same steps previously defined:

1. **Pre-process:** input data.
2. **Calculus:** setting of useful parameters.
3. **Post-process:** results analysis and output writing.

As I said before in this GUI you import the stress tensor calculated with the FEM modelling of the part. The number of the tensors you want to import is absolutely arbitrary, it is related with loads you set on your FEM code. For example, if in your FEM model you have a time history of loads described with maximum and minimum values of the sine functions, you will create two time steps and you will import two stress tensors inside the GUI. An important remark is related to the type of the analysis which are static and independent. Static evaluation of the stress condition is the first approximation made in the analysis, because the nature of the load fluctuation is obviously dynamic. In order to avoid rough errors the analyst must check that the load frequency is far from own frequency of the part. For example, considering a real load, the sine function has a frequency similar to  $0.5Hz$ , the datum is adequately far from the resonance point. Coming back to the number of steps you can decide to divide the sine function in more than two steps. You create  $n$  steps as many as the sine steps and you launch FEM solution with all the selected steps. You extract a stress tensor by each solution and you import in the GUI at the same time. The GUI reads the multiple steps as time steps and it creates a stress

time history according to the imported files. In this case is the correct modelling which ensure a smooth and high quality solution.

When you have all the steps from the FEM code you import all of them in the GUI, at the same time. If you are loading two steps the order is useless due to symmetry. However, if you are loading three or more steps the order is necessary for the correct solution. Owing to the file name the code is able to correctly sort the imported files and it prints on the underlying table the imported order. Over there the user checks if the order is correct or not. Moreover the code implements the verification stage useful to validate inputs. The validation stage is hidden to the user and only in case of an error the code shows the problem to the user.

Now, you accede to the solution tab where the parameters are similar to the previously described. The concept is the same and the logical idea is in perfect parallelism with sensitivity solver. The only one difference is the spacial discretization, in this GUI there is a panel where you need to insert the parameters for the methods which requires that variables. However, the time discretization is useless because it is enforced by the number of steps loaded before. After the selection of the criterion, the discretization tab becomes editable if the solver needs those variables, on the other hand if is useless it becomes uneditable.

Before starting with the solution other steps are the same of the sensitivity approach: selection the nodes you want to solve, choice of the loop strategy and output types. The first step of the analysis is the material properties editing, the code asks you the material properties for the analysis. The question is performed with an interface window as described for the other GUI.

With this stage the analysis is completed, henceforth the GUIs follow the same strategy and in sake of clarity and brevity I describe the process for both GUIs at the same time.

## 4.5 Damage tolerance analysis

In this section I will describe the main features of project philosophy called *damage tolerance*. Before we proceed with numerical implementation it is necessary to briefly describe which are the key-point of the theory. Thanks to the introduction you will better understand the implementation strategy and the future development of the code. The theory is well described in lots of books, I recommend [3], [4] and [13] for enhance the argument knowledge and for a complete explanation.

### 4.5.1 Basic theory

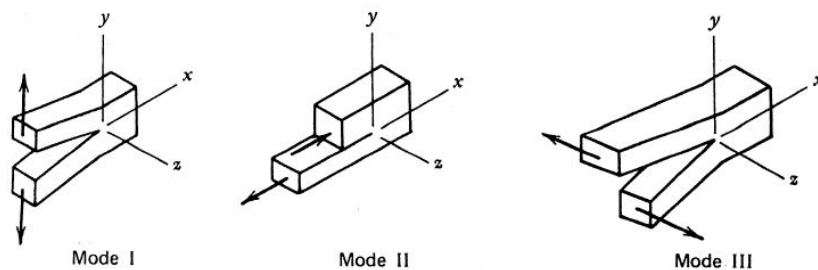


Figure 4.6: Reference modes for fracture mechanics. [3]

The first-ever idea of this project philosophy is the existence of the flaw. Other theories or other philosophy of projection try to avoid cracks enucleation, conversely in here the crack is



modelled and studied during its life. Hence, you try to describe the growth of the flaw during operative life of the components. Thanks to this statement, you assert that fracture mechanic and damage tolerance could not work without each other. Fracture mechanics is a branch of structural fatigue analysis which studies crack propagation inside of the metal body. Extreme complexity, linked with microscopic nature of the phenomenon, impose restrictive hypothesis described in LEFM theory: *Linear Elastic Fracture Mechanics* theory.

The load method plays a key role in the behaviour of crack growth, in the literature you can find three approaches:

1. **Modo1**: traction pressure motion which open and close the flaw.
2. **Modo2**: plane shear load.
3. **Modo3**: transversal shear load, referred to torsional load.

Fig. 4.6 shows the motions described in the list.

The procedure to estimate life needs one more parameter called *Stress Intensity Factor*, in notation  $K$ . This is the most important parameter and it assumes importance in the identification of the critical condition, moreover it is linked with the load method.  $K$  is evaluated as follow:

$$K = S\sqrt{\pi a} \quad (4.1)$$

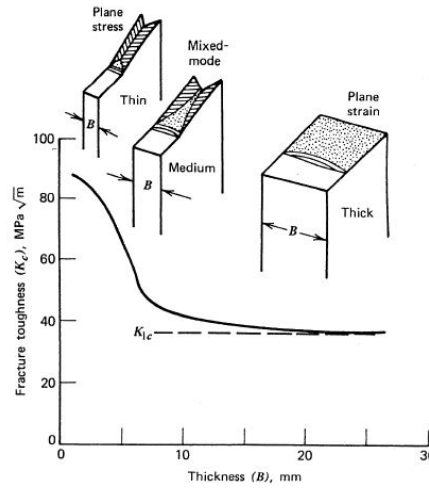
where  $S$  is the tractive stress and  $a$  is the half-length of the flaw. Its unit of measurement in the I.S. is  $MPa\sqrt{m}$ .

When this parameter reaches the threshold limit the part verifies a failure. The evaluation of this parameter can be performed analytically or experimentally. Analytical formulation is useful to clarify the main causes of the failure. Hence:

$$K_c = S_c\sqrt{\pi a_c}f\left(\frac{a_c}{w}\right) \quad (4.2)$$

each term with  $c$  subscript is a variable which might become critical. First, we can talk about load, or better stress; if the stress raises with unmodified flaw dimension, the part might verifies a failure. Dually, if the flaw dimension raises with the same stress, the failure occurs as well.  $f$  function assumes a particular role in this phenomenology because is linked with the geometrical shape of the flaw. With regard to experimental tests the thickness of the specimen is the first reason of mistakes. Indeed, critical factor evaluation has a necessary condition, which is: the specimen thickness must be large enough to allow flaw propagation towards a plain strain condition, this is assured by huge thickness. As shown in the Fig. 4.7 when the thickness is too tiny the flaw propagates into a state that is similar to a plain stress state, due to the negligible stress over thickness. In the same Figure you can see how the datum became independent form load with higher and higher thickness and it becomes useable.

Made these previews I conclude talking about propagation laws, because the most important think is to know how the flaw grows in the mechanical part. It is practical to divide the crack life into different stages to study each one with the best approach. There are several experimental theories which try to correctly describe the crack growth, they use lots of date of the material and of the load. The highest complexity, introduced by these theories, gives us better result notwithstanding more complexity to understand and implement the theory. It this initial stage that level of complexity is not required. Taking as reference aforementioned books I will use *Paris laws* as first step to understand the approach.


 Figure 4.7: Thickness effect on experimental test to evaluate  $Kc$ . [3]

From an absolute general point of view, you need a law which correlates crack size, useful life and stress intensity factor, Paris performs this task as follows:

$$\frac{da}{dN} = f(\Delta s, a) = f(\Delta K) \implies \text{Paris} : \frac{da}{dN} = C \Delta K^n \quad (4.3)$$

Thanks to this equation you are able to describe speed of propagation with stress intensity factor and two constant values:  $C$  and  $n$  which are material properties.

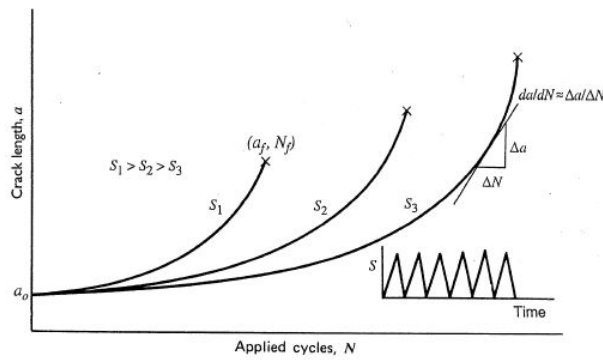


Figure 4.8: Growth curve described by Paris law. [3]

Due to this graph it is possible, thanks to numerical integration, to evaluate the crack size after a fixed number of cycles. When the part reaches the critical size we arrive at the failure of the component. Flaw critical point is linked with stress intensity factor critical value.

The procedure described before talks about crack propagation, henceforth component life, which are the most valuable data. However, it does not give me any information about first part of flaw life, the enucleation part. Indeed, the test I will implement in the GUIs poses a different question. **Knowing the acting stress over the node and the size of the flaw, will the flaw grow over those conditions?**

To give an answer to this question it is necessary to introduce a new one curve, well known as *No growth curve*. This curve is described in the  $R - \sigma_{dyn}$  plane and it reveals which nodes

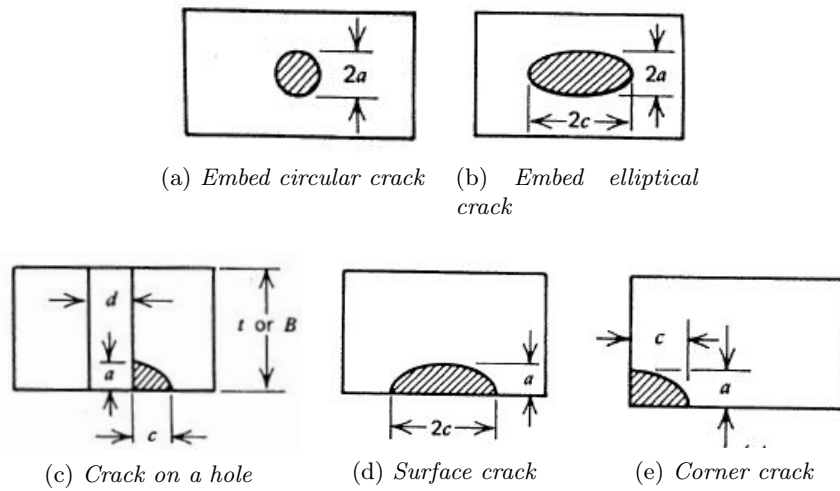


Figure 4.9: Examples of remarkable cracks position.

causes a crack growth. The nodes identification and the subsequent localization over the 3D model allows to identify the most critical areas of the whole part. On these areas there will be planned more maintenance controls than non critical locations.

#### 4.5.2 Numerical implementation

The numerical implementation of damage tolerance control, is based on critical or threshold curve for the reference material. As I described in the previous section, the size and the location of the flaw are peculiar data of crack propagation. The Tab. 4.2 shows the combinations studied in the current version of the GUIs. The images related to the table are shown in the Fig. 4.9, they

Mix of crack sizes and locations		
	$a = 0.25 \text{ mm}$	$a = 0.38 \text{ mm}$
circular flaw	✗	✗
elliptical flaw	✗	✗
flaw on a hole	✗	✗
surface flaw	✓	✓
corner flaw	✓	✓

Table 4.2: Table with size and location combinations for each material.

reveal the remarkable positions of the flaws. The selected flaws came from a line of reasoning linked with fatigue behaviour. The internal embed flaws are useless for our study because the code analyses the skin of the body. However, these flaws are useful in manufacturing damage analysis. The flaws on the hole are not studied because the holes are problematic areas. Their modelling is a challenging task and, as you will see in the Chapter 5, the results are not reliable for the test case. Finally, I decide to model cases (d) and (e) of the following images.

Talking about a database of mostly used metals, I decided to implement the structure with the highest number of degrees of freedom, in order to let the highest choice to the user as possible. The analyst can select a material from the current database or he can create a new material with

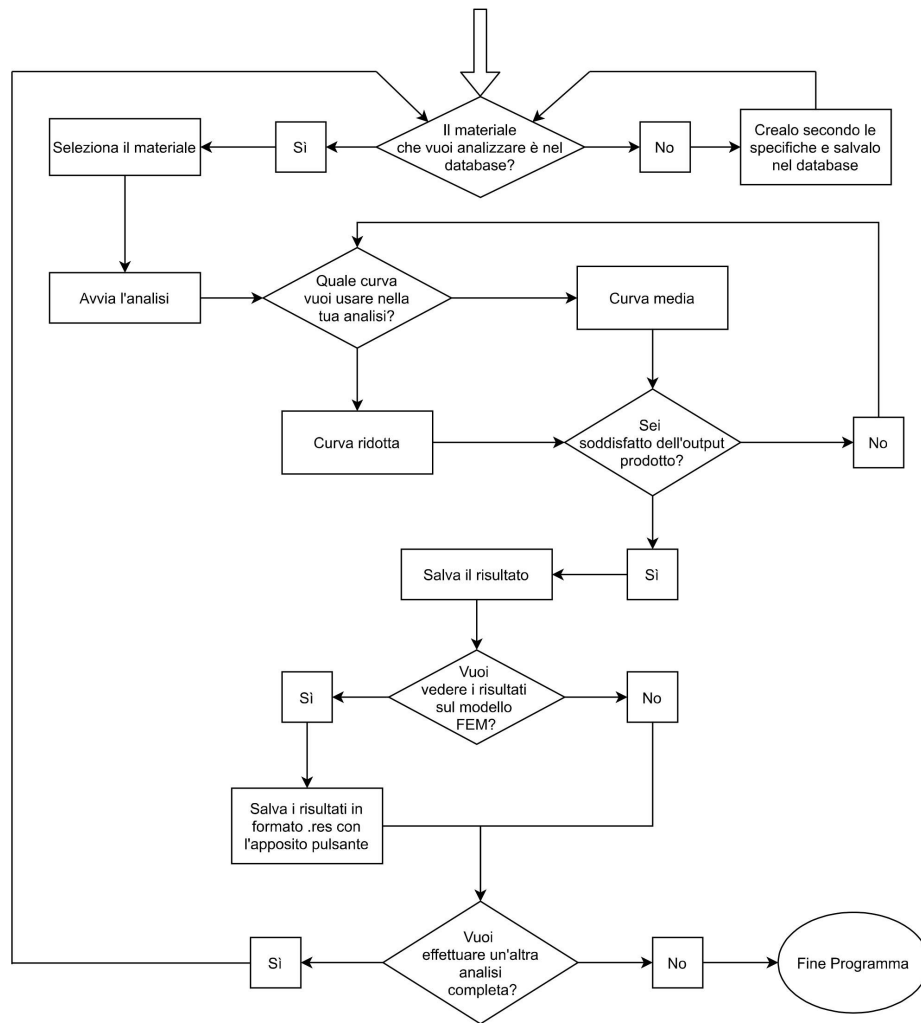


Figure 4.10: Damage control stage flowchart.

a new material. Moreover, the code asks the user which kind of curve you want to use. There are two types of curves: *mean curve* and *reduced curve*. The most relevant difference between curves is the *Scatter factor* which is equal to  $Ks = 1.15$ . It reduces the mean curve in order to be more restrictive in the damage test.

The selected material belonging to three different alloys:

- **Aluminium alloys:** I take into account the alloys belonging to 7xxx class, that means zinc (Zn) as the first alloy. These non-ferrous alloys are built with thermal and mechanical treatments in order to raise material properties. Most relevant are: quench and artificial ageing as regards thermal treatments, shop-peening and cold-working concerning mechanical ones.
- **Titanium alloys:** are based on  $\alpha + \beta$  alloy well known as Ti6Al4V, with possibility of shop-peening.
- **Steel:** are rarely used due to their high density. They are employed with fasteners, bolts

and screws because you obtain high performances into a small components. This performance is allowed by extended thermal and chemical superficial hardening.

Defined the list of the useful materials, I access the internal database of fatigue in order to create the files for the code from the whole database. I create some files for each material. All of this are accessible from the GUIs for an immediate analysis.

The reference scheme for a correct job in this section is shown in the Fig. 4.10. The flowchart describes the steps and the questions the code asks you to perform the analysis. The most relevant feature is the outer loop, indeed with one numerical solution you are able to analyse all the materials.

If you stay in this tab you can analyse different materials, flaws and curves. The baseline idea is to allow an evaluation, the storage of data into graph or into numerical files and the repetition of the previous steps changing a parameter. Instantly, you are able to relate different flaws or different materials. Moreover, you can create a binary file which is attachable over the FEM model and it shows the exact locations of the critical nodes in the component.



# Chapter 5

## Results

### 5.1 Introduction

In the following chapter I will describe the most relevant results which come from the calculation tool I developed. I will explain different points: the analysis results, the comparison and the validation between GUIs and FEM, in the end the experimental data. I will specify the different approaches you will follow to correctly use the tools because the results are slightly different, however I will demonstrate that from the same input you extract the same output.

The validation between the tool and the FEM model it will be shown not only with graphs but with tables as well. If it is possible, I will evaluate percentage errors. I will explain damage tolerance analysis made on the component and a relation between experimental fatigue curves and numerical data.

During the first part of the chapter the results refer to the “principal stress methodology” because it is the only one which gives us useful information to FEM correlation. In the last one part of the chapter I will consider the other criteria implemented in the GUIs and I will make a comparison in order to understand which are the most relevant fruit of those criteria and if the results of the theories are confirmed by each other.

### 5.2 Loads system

The first step towards results is linked with load time history. The load time history I select for the analysis is unique. It will be the same for all the following runs with the GUIs. In order to guarantee an absolute availability of the numerical results I decide to analyse the same loads implemented in the FEM model and used in the experimental test. Due to this choice the results are easily comparable with the other evaluated in different manners. As you can see in Fig. 5.1 the most significant load component is the centrifugal one which assumes a constant value for our purpose. This chart clarifies the previously described idea of the centrifugal component specific weight.

With the aim of modulating the analysis I select one period of the different sine functions. I cut off the time period after 2.05 *sec* in order to guarantee a complete oscillation of the the functions. Additionally one other advantage of the reduction of the time period is the code distributes the time steps into two seconds and not five ones. This procedure is correct because the loads are defined as 1 *per rev*, that is their period is referred to the blade complete motion around the mast. The load which acts one time per rev shows its effects ones in a complete blade revolution. The phase difference, which plays a central role, does not invalidate this procedure

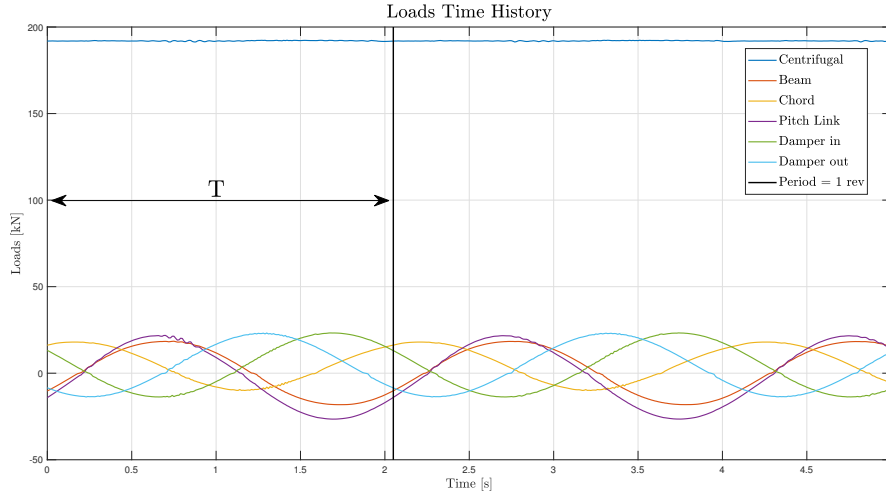


Figure 5.1: Time history of the loads used in the validation test.

because the frequency of the sines is the same. Moreover, I implement the time normalization; the final time is set to one and the whole time slice is divided linearly. All of this allows a easier comparison between different load time histories.

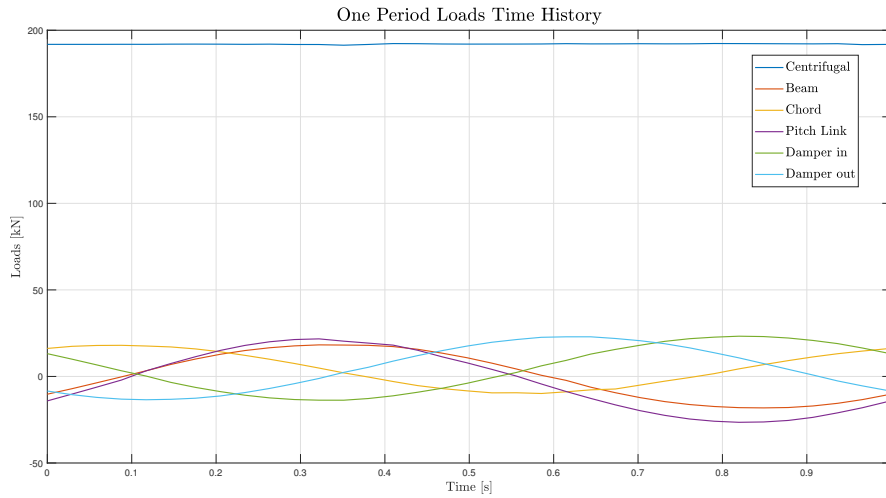


Figure 5.2: Sine functions used for the analysis.

Another useful modification of the curves is the time steps reduction. As you can see the sines are “contaminated” by high frequency loads which modifies the smooth trend of the pure sine function. This phenomenon is common if you think about flight data which are affected by a huge amount of vibrations and which are sampled lots of times per second. Without using a interpolant sine function to extract the fundamental sine from loads I decide to describe the function with wider mesh, the result is shown in Fig. 5.2.

In order to ensure a better read of the graph, I report into the Tab. 5.1 the most relevant values for each sine:



Features of load sine functions					
	$F_{MAX} [kN]$	$F_{Min} [kN]$	$F_{stat} [kN]$	$F_{dyn} [kN]$	$F_{ase} [deg]$
Centrifugal	192	192	192	0	ND
Beam	18.5	-18.1	0.2	18.3	-32.4
Chord	18	-10	4	14	44.5
Pitch Link	21.8	-26.5	-2.3	24.1	-36.8
Damper in	23.2	-13.7	4.8	18.4	136.4
Damper out	23.2	-13.7	4.8	18.4	-121.3

Table 5.1: Useful sine loads data.

The first remark I do is related with phase difference in the load spectrum. If you identified which is the most critical load step, you would have not succeed owing to loads variety.

When the analysis force us to identify the worst point on load spectrum an error occurs. I will show the problems and the errors which came from this choice. The choice have to be performed with experimental information and experience know-how in order to correctly identify the real worst load condition.

### 5.3 Sensitivity matrices analysis

Thanks to the loads described by the sines in Fig. 5.2, I can directly solve the problem if I have the sensitivity matrices.

The extremely favourable side of this GUI is the absolute freedom to create a time discretization in the interval  $[0, 1]$ . With the mesh the code is able to identify the values of the sines and it evaluates the criteria into these points. The most critical stage has not be performed by the analyst as pre-process step, instead it is performed by the solver on its own. When you define the time mesh the solver automatically evaluates all the possibilities and it returns the critical one, regardless of where the critical step is on the time discretization grid.

Results with own principal stress method - Method 1				
Node ID	Static $\sigma_s [MPa]$	Dynamic $\sigma_d [MPa]$	Stress Ratio	Goodman correction
1037485	9.188e+00	3.090e+02	-9.423e-01	3.155e+02
1037497	-9.937e+00	3.119e+02	-1.066e+00	3.119e+02
1148524	8.355e+01	1.982e+02	-4.070e-01	2.434e+02
5030520	7.077e+00	3.125e+02	-9.557e-01	3.175e+02
5030521	-2.463e+01	3.476e+02	-1.153e+00	3.476e+02
5030532	-6.098e+00	3.179e+02	-1.039e+00	3.179e+02
5032567	9.298e+01	1.949e+02	-3.540e-01	2.456e+02
5032808	8.339e+01	1.925e+02	-3.955e-01	2.363e+02
5069106	9.346e+01	1.958e+02	-3.538e-01	2.471e+02
5069124	1.034e+02	1.944e+02	-3.057e-01	2.525e+02

Table 5.2: Reference nodes results with: 35 time steps, sensitivity matrices and first principal stress method.

Solution quality is linked with the point of view radical changing. When the analyst has to

decide the critical condition basing its evaluations on the only loads, unavoidably he commits an error. The error could be enclosed into small areas maybe not critical, but it will be. On the other hand, this solver evaluates the critical time step couple thanks to sensitivity matrices information and it obtains the result comparing the stresses of the whole spectrum which acts in the reference node.

I fix some critical nodes and I show the results with the different solvers. Due to the high number of nodes, precisely this model has 86247 (eighty six thousand two hundred forty seven) skin nodes, I show the results of the most relevant ones, but the conclusions will be described with the usage of the graph as well.

Results with dynamic principal stress method - Method 2				
Node ID	Static $\sigma_s$ [MPa]	Dynamic $\sigma_d$ [MPa]	Stress Ratio	Goodman correction
1037485	-1.237e+02	3.090e+02	-2.336e+00	3.090e+02
1037497	-2.041e+02	3.119e+02	-4.788e+00	3.119e+02
1148524	8.348e+01	1.982e+02	-4.074e-01	2.434e+02
5030520	-1.391e+02	3.125e+02	-2.604e+00	3.125e+02
5030521	-2.121e+02	3.476e+02	-4.129e+00	3.476e+02
5030532	-2.027e+02	3.179e+02	-4.520e+00	3.179e+02
5032567	9.279e+01	1.949e+02	-3.549e-01	2.455e+02
5032808	8.321e+01	1.925e+02	-3.964e-01	2.362e+02
5069106	9.327e+01	1.958e+02	-3.547e-01	2.470e+02
5069124	1.033e+02	1.944e+02	-3.062e-01	2.524e+02

Table 5.3: Reference nodes results with: 35 time steps, sensitivity matrices and second principal stress method.

The results are not directly comparable. The most appropriate comparison should be done between these tables and the tables I will describe in the Chapter 5.4. Those tables will show the results evaluated with step solution GUI.

The Goodman-Haigh diagram which came from the previous tables, Tab. 5.2 and 5.3, is described in the Fig. 5.3. What you can immediately see is referred to red markers, they are moved towards the negative semi-axis of abscissa. This trend is absolutely correct and it is related with the different solver strategy. Reminding what is described in Chapter 3.6.1, the first method (own principal directions) considers the maximum static and the maximum dynamic separately, it evaluates the principal direction as independent. This approach mixes different directions and it obtains a mean stress which is generally greater than zero. The second method (dynamic principal directions) considers the dynamic directions to project the static tensor. It does not matter the maximum static, the used component has the same direction of the dynamic principal stress, whatever it is. The detailed explanation of the theory is presented in the Chapter 3.6.2. Second approach is the most suitable.

Goodman correction mitigates this benefit. I suggest to check Fig. GHneg to better understand what I mean. Looking at the modulus of the static stress revealed by the analysis, I asset this method might be questioned. A future development moves in this way. The user should be able to select the correction strategy he prefers.

The research of the most critical areas is focused into two groups of nodes, I identify them in Fig. 5.4.

Previous tales collect critical nodes data which are highlighted in the ellipses shown in Fig. 5.4.

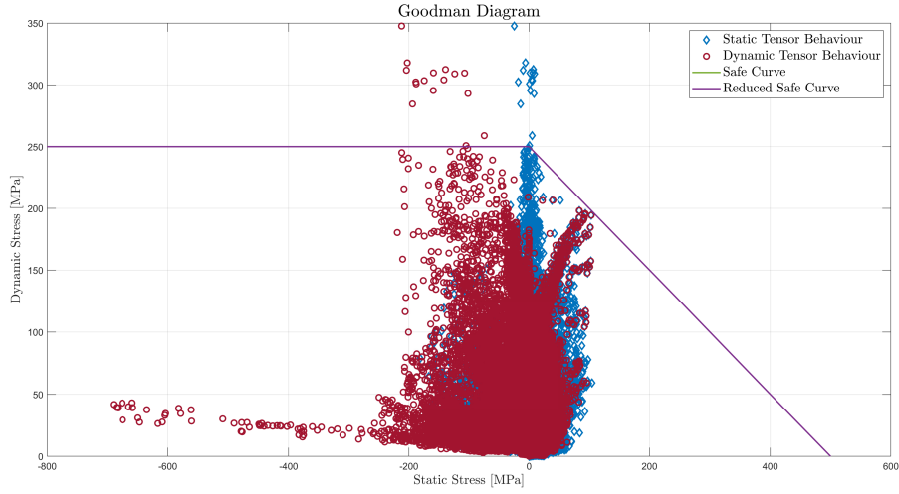


Figure 5.3: Goodman diagram for sensitivity matrices analysis.

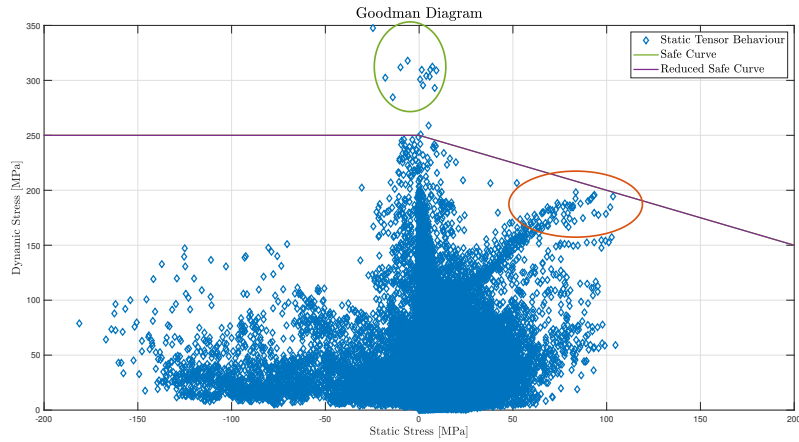


Figure 5.4: Critical sets of nodes.

Green ellipse picks up the nodes with the highest dynamic value, they are out of Goodman border and they should be the most critical for structural collapse. Bordeaux ellipse nodes are inside of Goodman line but they show the worst mix of static and dynamic stress. Ten of both groups of nodes are described into the tables.

In order to understand which are the most stressed areas of the body I need to plot the results on the FEM model to reveal the areas identified by the ellipses.

The images come from the parallelism between the solutions of the FEM model on the left, and the results of my tool on the right. FEM model is the same used to validate the experimental test. Talking about multi-coloured sectors the difference is negligible. These images are referred to the own tensor rotation (Method 1). The comparison is made with only this method because it is the only one implemented in the FEM model.

Analysing the Fig. 5.5 and looking at the connection radius between the horn and the lever, you can see a region of red colour quite extended, with a medium-stressed area as border. In the Fig. 5.6 the edge of the hole is the most critical area. Performing the comparison between

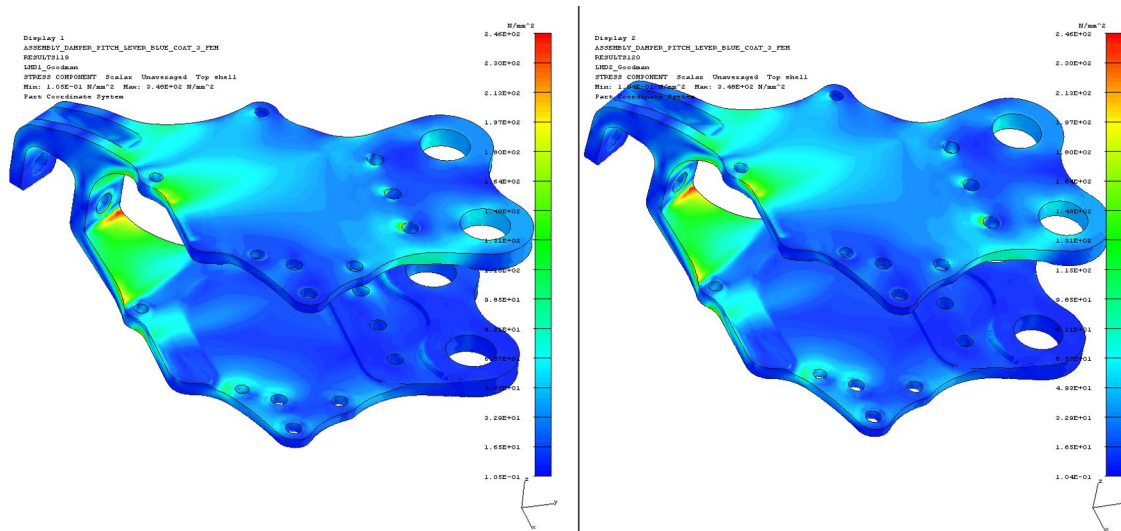


Figure 5.5: Critical area located at the pitch horn of the pitch link.

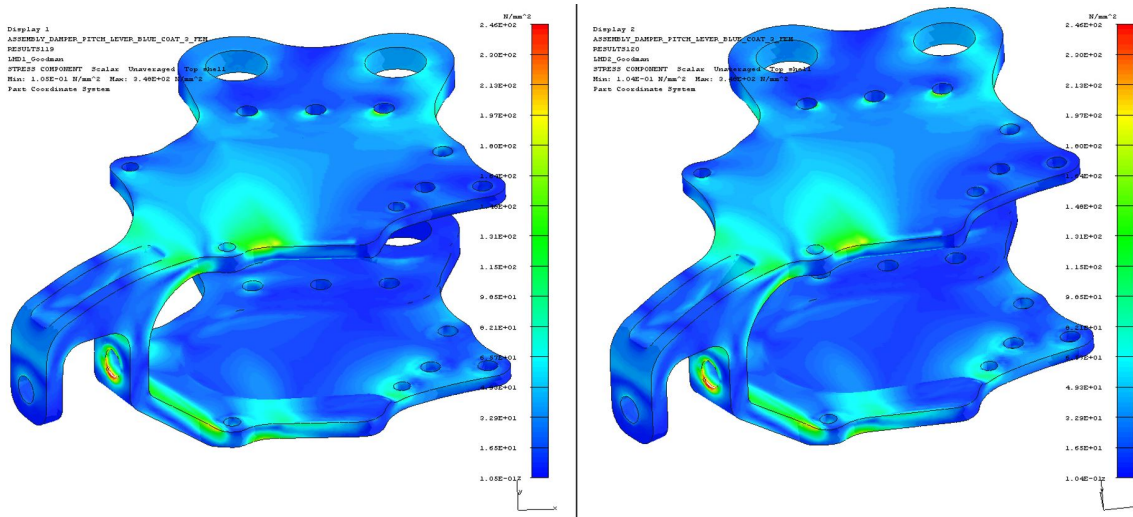


Figure 5.6: Critical area located at the edge of the hole, the hole is the seat of the pitch link pin.

the image and the ellipses I am able to identify green ellipse nodes in the hole and red ellipse nodes in the connection radius of the pitch horn.

To understand which is the most demanding area it has been vital the explanation of FEM modelling technique. In the FEM model which gives us the results for the test are modelled lots of components. This parts interact as normal parts do in an helicopter rotor. In addition to the lever, our reference part, are studied all the connections which create the working group. Some components are passive ones, they endure the load and they fulfil their tasks. Other parts are used to define the loads which stress the whole structure, this is the case of the pitch link which is connected to the lever thanks to a pin. The pin's slot is the critical hole described into the Fig. 5.6. The interaction between the lever and the pin is performed modelling all the contact behaviour. Consequently, the lever shows a peak of stress on the edge of the hole which is related to the load application and it is a more serious condition than the real one. It is correct to assert

the load is the highest in magnitude but this load it will not be the most decisive for the fatigue behaviour owing to the genesis of the load itself. Due to this explanation the most critical area to the fatigue behaviour is the radius between horn and link described in Fig. 5.5.

Let's get back to the Tab. 5.2 and 5.3 in order to better understand the relationship among principal stress methods.

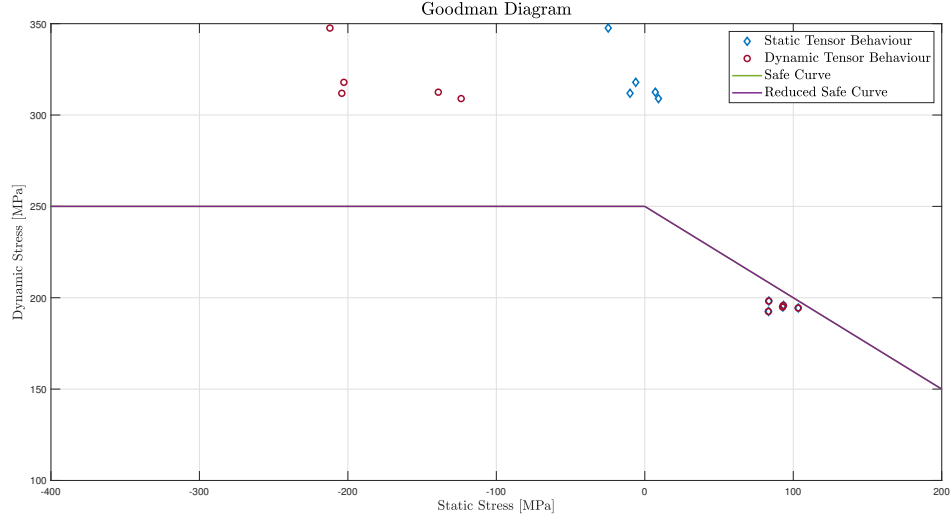


Figure 5.7: Critical nodes.

Thanks to Fig. 5.7 you understand the nature of the solution methods. Investigating green ellipse nodes, which have the highest dynamic stress, the first method says that these nodes are the most important and most stressed because they have the maximum dynamic stress. But this stress is associated with a static stress which is around zero. On the other hand with the second approach the static stress is far from zero, it is evaluated in the dynamic direction and due to the hole behaviour it is correct to find a negative value. Thanks to this situation we can explain why I chose to implement the open-Goodman diagram, the correction helps me to avoid this false-critical points. This huge variance between methods is a “wake-up call” for the real criticism of nodes.

Moving our focus on bordeaux ellipse you can see a perfect coincidence between methods, both in static and dynamic stresses. This numerical behaviour motives why the plot from FEM model and from first approach are perfectly the same, as shown in Fig. 5.5. The critical directions evaluated with the first approach are the same of the second one. This behaviour comes from the geometrical shape as well. The area is located into an edge of the structure which restricts the possible directions related to the acting load. Henceforth the most demanding equivalent weary cycle is related with dynamic principal directions.

Goodman correction allows the numerical understanding of critical level of nodes, moreover it evaluates the weight of the static stress which is a not negligible actor in critical node evaluation. Considering the most relevant nodes, rows three, nine and ten of results tables, the worst node is the tenth which has a dynamic stress slightly lower than the others. I am talking about 4 *MPa* difference, but due to the static value this node is the most problematic.

It is possible to identify the critical condition based on Goodman correction if nodes respect some points:

- The nodes show a equal mix of static and dynamic stresses. The node with a dynamic stress which is a tiny percentage of the static stress are not useful for fatigue study.
- Both approaches, own directions and dynamic ones, converge into a single critical direction. Cohesion of results is assured.
- The nodes with negative mean stress but with higher and higher dynamic stress which shows positive stress for most of cycle. Cycles with high percentage of negative stress are useless for fatigue analysis.

## 5.4 Stress tensor analysis

This solution could be called “reduced step analysis” because you start with some well know steps and you can solve the time history in the only step you know. In order to numerically correlate the GUI with the results already shown I have to use this second interface in a not completely correct manner. The correct usage of this interface is to extract from the FEM model the stress tensor and solve the problem with that data. Now, I create the stress tensor with the same load time history. In this study I will use only two time steps.

Obviously, when you are designing a new part the solution of FEM model is needed. The only data extract from the complete FEM analysis are comparable with the previous tool results only. Confirmed the perfect parallelism among results, shown in the above section, it is right to use the previous result to make the comparisons.

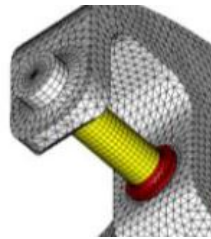


Figure 5.8: Pitch link pin and pitch lever horn model.

As first step I need to be coherent with upper stages, in order to correctly compare the results. The first step is to identify the the critical load condition among sine functions. I remind you the previous section to understand the problem linked with this choice. I will describe, with graphs and tables, what it means to base the study on loads and not on stresses. The usage of loads will cause an error. To perform this task you have some ways, each one is linked with the information you have:

- **Cumulative procedure:**

1. You sum all the load components each others, in order to find a cumulative sine function.
2. You search the maximum and the minimum values of the cumulative sine.
3. You extract the time steps referred to both max and min of the sine.
4. With the time steps you find the values of every single component on load in those time steps.

5. Solve the FEM model with the static loads referred to max and min and I extract the stress tensors for both analysis.
  6. You enter the GUI with the previous results and you solve fatigue problems.
- **Most dangerous load:** is based on your own knowledge of the part. Do you know which is the most stressed area? Do you know which are the gravest loads? If you were able to answer this questions, you might use this approach. Evidently, holes, tiny connection radius and rude geometrical modification are popular location for crack enucleation. Actually, the real problem is to find where the load is related with a critical location. It is a mix of load and location which create the damage. Understood which is the most relevant load which acts in the worst location you can select that sine to identify the worst time steps. Max and min values will define the reference time steps, you will use to extract data from the other sines.

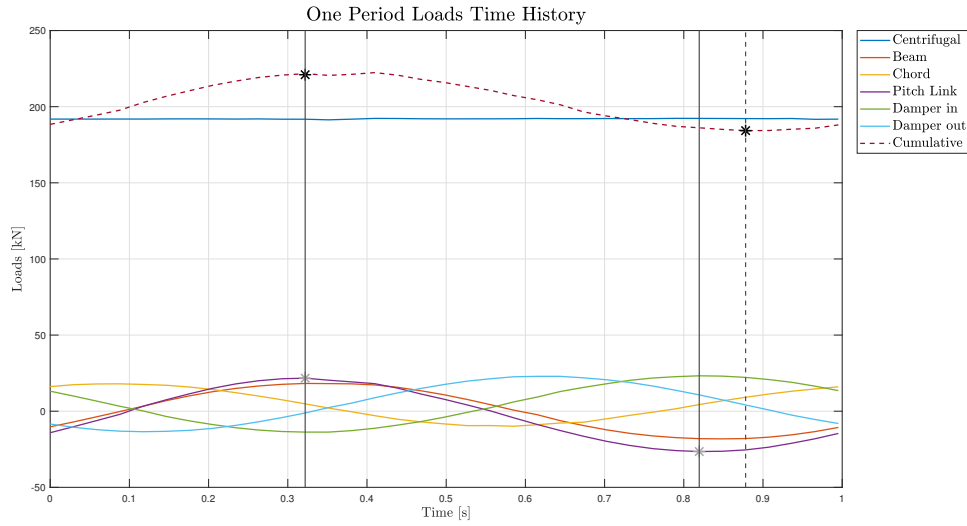


Figure 5.9: Critical condition identification with both approaches.

As you can see in the Fig. 5.9 the “maximum peak” is the same for both approaches, conversely the minimum value reveals a not negligible difference. The reference load is called Pitch Link, that is the load which is born from the cyclic lever command which acts on the yellow pin shown in Fig. 5.8. At this time of the design of the component is an obvious choice, but it is not always easy to do if you are into a preliminary design stage. I report into Tab. 5.4 the loads used in the FEM model to extract the useful stresses. I renew that the choice is fruit of the previous know-how of the component behaviour.

	Maximum and minimum loads					
	Centrifugal	Beam	Chord	Pitch Link	Damper in	Damper out
$F_{Max} [kN]$	192	18.5	5	21.8	-13.8	-1
$F_{Min} [kN]$	192	-18.1	5	-26.5	23.3	10.2

Table 5.4: FEM loads used to generate the stresses for the reduced analysis.

Created the stresses with the FEM solver we can start with the GUI evaluation. Before starting with the GUI I like to briefly compare the stresses, in sake of simplicity I limit the comparison of the main diagonal of the stress tensor. These components are enough to reveal the consistency of approaches.

Difference between two approaches of load evaluation.				
	ID FEM	$\sigma_{XX}$ [MPa]	$\sigma_{YY}$ [MPa]	$\sigma_{ZZ}$ [MPa]
Method 1	1037485	-122.4141	-119.5461	-401.4010
	1037497	-116.7292	-144.2651	-449.2793
	1148524	-35.4741	-33.4474	-47.4380
	5030520	-123.9765	-117.8460	-427.5681
	5030521	-221.6771	-276.8426	-334.3904
	5030532	-128.7070	-137.1287	-474.4068
	5032567	-30.2515	-24.2110	-48.0092
	5032808	-36.2221	-26.9093	-51.4968
	5069106	-35.4892	-21.5996	-62.1718
	5069124	-26.4271	-17.8231	-60.8332
	ID FEM	$\sigma_{XX}$ [MPa]	$\sigma_{YY}$ [MPa]	$\sigma_{ZZ}$ [MPa]
Method 2	1037485	-120.5851	-117.9710	-395.7981
	1037497	-115.0162	-142.1475	-443.0224
	1148524	-35.2010	-33.3201	-47.2840
	5030520	-122.1801	-116.2720	-421.6133
	5030521	-218.9271	-273.0229	-329.9414
	5030532	-126.8256	-135.1294	-467.7741
	5032567	-30.0651	-24.1807	-48.0507
	5032808	-35.8819	-26.9620	-51.2267
	5069106	-35.0960	-21.6695	-61.9289
	5069124	-26.1555	-17.9682	-60.7781

Table 5.5: Stress difference between load approaches.

Thanks to the Tab. 5.5, you understand the difference is not negligible but is not thus high to compromise the final result. Final results of the GUI are shown in the following plots:

In the comparison performed by Fig. 5.10 and 5.11 you can cross between GUI's results; the scale of the colour bar are preventively set to the maximum dynamic value in order to better clarify the most relevant areas. I assert the approaches converge to the same result, they fit each other the results, thus they are usable to make a comparison. The critical areas are captured by the reduced solution as well as the sensitivity one, there is a minimum difference in data because of the load problems. However, the clarity of the presented approach ensures the goodness of the evaluation of the load critical condition.

In order to better understand where the numerical difference are, it is necessary to analyse the numerical results saved from each GUI extracting the reference nodes.

Most of nodes reveal a negligible difference which means a correct computational process of GUIs and a correct estimation of critical loads. However, some nodes shows a difference that is not ascribable to the numerical process, thanks to these nodes we understand where is the basic difference between approaches and GUIs. Thanks to sensitivity matrices you can divide



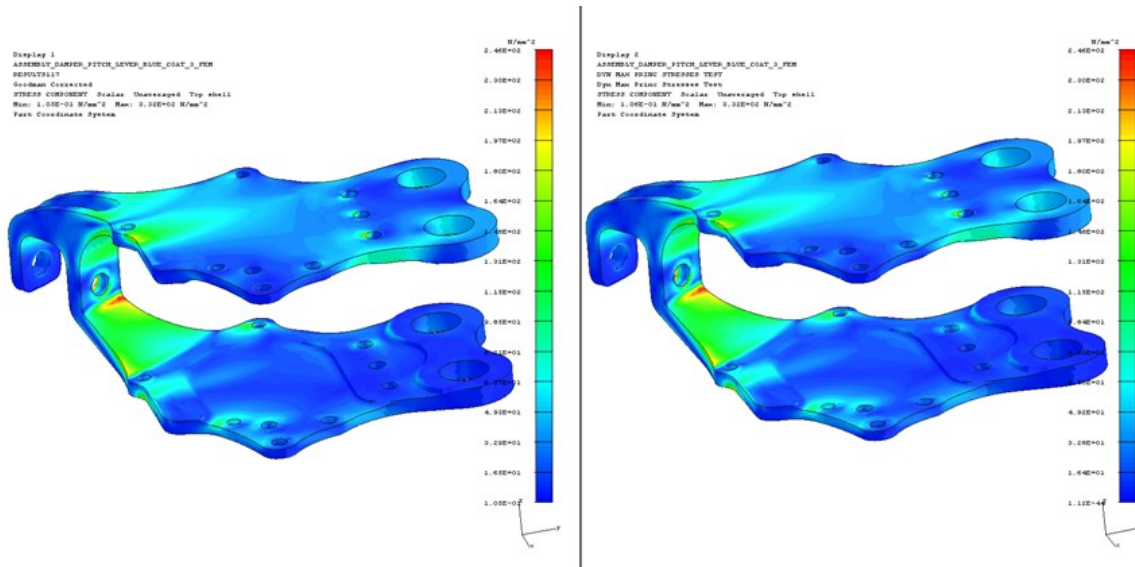


Figure 5.10: Comparison between GUIs on the critical area on the pitch link horn.

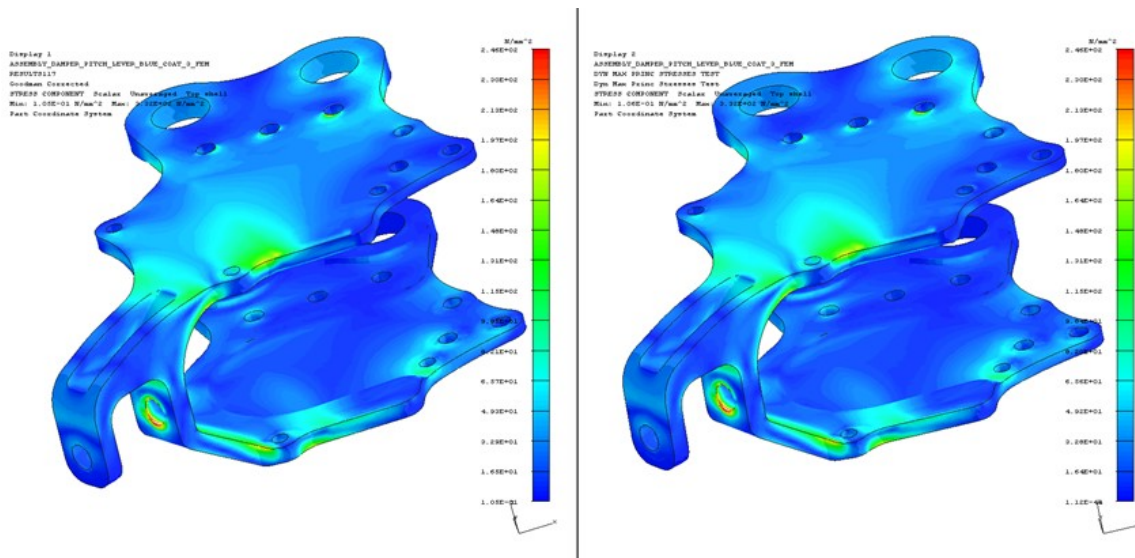


Figure 5.11: Comparison between GUIs on the critical area of the hole which is the slot of the connection pin between the lever and the link.

your time slice into lot of time steps, you take and advantage form this feature of the first procedure to be more precise with the solution. You are accurate because the code evaluates all the possible couple of critical steps and it finds the most demanding wherever it is. This couples are proportional to the squared of the time steps, the code finds the most demanding only when it knows all the possibilities. If you fix two time steps the only couple is between the imported steps, there is no way out. Studying some nodes you can run into errors related with a singular behaviour: the peak of load does not necessary correspond to the peak of stress. This information is the most relevant one hidden into sensitivity matrices.

Now you can numerically discover what I described in the above lines.

Results with reduced interface and own rotation - Method 1				
Node ID	Static $\sigma_s$ [MPa]	Dynamic $\sigma_d$ [MPa]	Stress Ratio	Goodman correction
1037485	8.030e+00	3.032e+02	-9.484e-01	3.082e+02
1037497	-1.029e+01	3.058e+02	-1.070e+00	3.058e+02
1148524	8.089e+01	1.947e+02	-4.130e-01	2.323e+02
5030520	5.964e+00	3.066e+02	-9.618e-01	3.103e+02
5030521	-2.534e+01	3.407e+02	-1.161e+00	3.407e+02
5030532	-6.666e+00	3.117e+02	-1.044e+00	3.117e+02
5032567	9.018e+01	1.915e+02	-3.598e-01	2.337e+02
5032808	8.078e+01	1.891e+02	-4.014e-01	2.255e+02
5069106	9.067e+01	1.923e+02	-3.592e-01	2.349e+02
5069124	1.004e+02	1.911e+02	-3.108e-01	2.391e+02

Table 5.6: Numerical results with two-step analysis and own direction rotation.

Results with reduced interface and dynamic rotation - Method 2				
Node ID	Static $\sigma_s$ [MPa]	Dynamic $\sigma_d$ [MPa]	Stress Ratio	Goodman correction
1037485	-1.244e+02	3.032e+02	-2.391e+00	3.032e+02
1037497	-2.032e+02	3.058e+02	-4.960e+00	3.058e+02
1148524	8.081e+01	1.947e+02	-4.134e-01	2.323e+02
5030520	-1.395e+02	3.066e+02	-2.668e+00	3.066e+02
5030521	-2.118e+02	3.407e+02	-4.286e+00	3.407e+02
5030532	-2.019e+02	3.117e+02	-4.674e+00	3.117e+02
5032567	8.999e+01	1.915e+02	-3.607e-01	2.336e+02
5032808	8.061e+01	1.891e+02	-4.022e-01	2.254e+02
5069106	9.049e+01	1.923e+02	-3.601e-01	2.348e+02
5069124	1.003e+02	1.911e+02	-3.113e-01	2.390e+02

Table 5.7: Numerical results with two-step analysis and dynamic direction rotation.

These tables have to be compared with the twins shown in the previous section: Tab. 5.2 and Tab. 5.3. In order to support the comparison I report the Goodman diagram printed by the interface:

With the aim of evaluating the error between GUIs I set the sensitivity technique as reference and I evaluate the percentage difference following the formula:

$$\varepsilon\% = \frac{Var_{sens} - Var_{steps}}{Var_{sens}} \cdot 100 \quad (5.1)$$

I obtain the following tables:

Due to the results shown in the tables, I assert the two tools generates an error which is into a reasonable range, it never goes up to 5% and in the most of cases it is lower then 1%.

The motivation I can describe are the following:

1. **Numerical:** I am not talking about the solver precision, both solvers have the same precision during the calculus, in fact if I force a numerical input equal for both GUIs, I find the same numerical output results, with all the significant digits equal to each other. With the term equal I mean a difference between values which is lower then machine  $\epsilon$ ,  $\sim 10^{-16}$

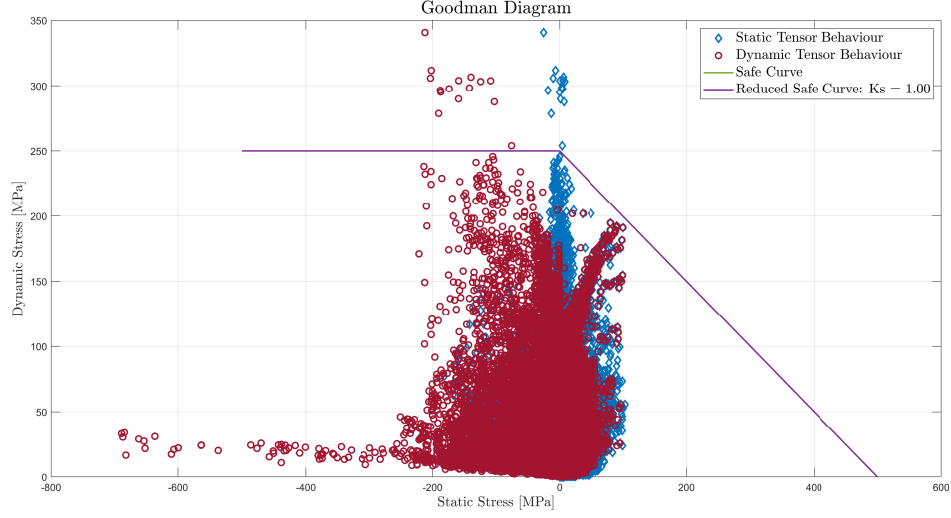


Figure 5.12: Printed Goodman diagram with two steps analysis.

Percentage error with first method				
Node ID	$\varepsilon_{\sigma_s}$	$\varepsilon_{\sigma_d}$	$\varepsilon_R$	Goodman error
1037485	1.260e+00	1.877e+00	-6.474e-01	2.314e+00
1037497	-3.552e+00	1.956e+00	-3.752e-01	1.956e+00
1148524	3.184e+00	1.766e+00	-1.474e+00	4.560e+00
5030520	1.573e+01	1.888e+00	-6.383e-01	2.268e+00
5030521	-2.883e+00	1.985e+00	-6.938e-01	1.985e+00
5030532	-9.315e+00	1.950e+00	-4.812e-01	1.950e+00
5032567	3.011e+00	1.744e+00	-1.638e+00	4.845e+00
5032808	3.130e+00	1.766e+00	-1.492e+00	4.570e+00
5069106	2.985e+00	1.788e+00	-1.526e+00	4.937e+00
5069124	2.901e+00	1.698e+00	-1.668e+00	5.307e+00

Table 5.8: Percentage error between interfaces due to own rotation of tensor.

in MATLAB double precision. With numerical error I think about I/O phase error. Both in load definition and FEM import of data the code truncates the whole decimal digits to a precise number. This behaviour creates a incorrect decimal digits due to truncation.

2. **Load identification:** is the same previously described and linked with critical load identification. About this point I found a glaring example of the problem linked with the load choice. Thanks to the Fig. 5.9 you have already understood how you can treat the loads. I consider the node 1018016, it is not a critical one but it reveals a strange behaviour:

In Fig. 5.13 you can notice the strange path of the stress component  $\sigma_{xx}$  which has a non-sinusoidal trend in time. I remind the load is a perfect sine function. This is a perfect example of the aforementioned problems. The selected time steps, which comes from loads maximum and minimum, are not the critical steps of this particular node. In term of stress the difference is not the highest, consequently the evaluated equivalent stress is not the

Percentage error with second method				
Node ID	$\varepsilon_{\sigma_s}$	$\varepsilon_{\sigma_d}$	$\varepsilon_R$	Goodman error
1037485	-5.659e-01	1.877e+00	-2.354e+00	1.877e+00
1037497	4.410e-01	1.956e+00	-3.592e+00	1.956e+00
1148524	3.198e+00	1.766e+00	-1.473e+00	4.560e+00
5030520	-2.876e-01	1.888e+00	-2.458e+00	1.888e+00
5030521	1.414e-01	1.985e+00	-3.802e+00	1.985e+00
5030532	3.947e-01	1.950e+00	-3.407e+00	1.950e+00
5032567	3.018e+00	1.744e+00	-1.634e+00	4.847e+00
5032808	3.125e+00	1.766e+00	-1.463e+00	4.572e+00
5069106	2.981e+00	1.788e+00	-1.522e+00	4.939e+00
5069124	2.904e+00	1.698e+00	-1.666e+00	5.309e+00

Table 5.9: Percentage error between interfaces due to dynamic rotation of tensor.

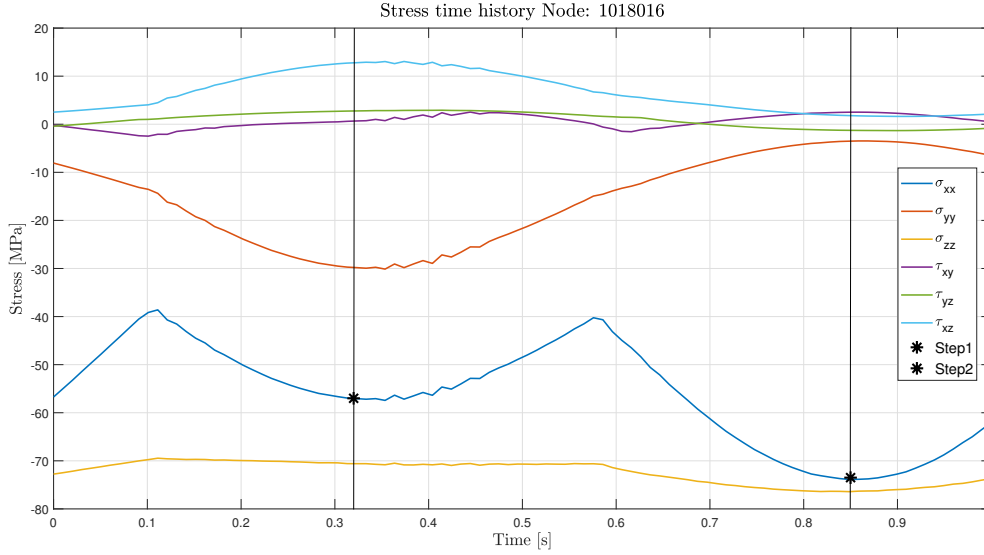


Figure 5.13: Stress value of node number 1018016.

critical and it is far from the correct one. This chart explain the not-correlation between load and stress. The not-correlation is the first source of error.

You know that the abnormal behaviour of the stress is described in the sensitivity matrices. Fortunately, this behaviour is limited to the few nodes if compared with the analysed ones. In the critical nodes the GUIs give the same results and this strange behaviour is not dominant.

Sensitivity matrices were born with the aim of defeating this problem because the code evaluates the critical condition among all possibilities. On the other hand step solution is limited to the known and imported information.

## 5.5 Comparison with other theories

The above sections talk about principal stress results, the reason is to search into two motivations:

1. Fast evaluation time.
2. The code creates the useful information to make the parallelism with FEM model in order to validate the calculus.

Now, it is necessary to deeply understand which kind of data you obtain from the other implemented criteria. After that I will make the comparison with principal stress results.

The other criteria inserted in the actual version of the tools have the purpose to identify the most dangerous areas regard to nucleation behaviour. The criteria are express as an inequality which has to be verified in the locations around the central point of the analysis, the node. I highlight the completely different approach of these criteria if compared with the previous. Here the code searches the locations where a crack can easily nucleate, before we were comparing the 3D state with an equivalent uniaxial one.

From the beginning I assert all the criteria give us complementary information if compared with principal stress methods. The results are not exchangeable because they talk about different part of the flaw-life cycle. We can find nodes with different data from both approaches: some show critical behaviour concerning crack propagation conversely the crack nucleation is not critical. Some other nodes show a critical data of nucleation and a demanding cycle relative to propagation. Obviously the second group of nodes is the most dangerous for fatigue behaviour. It is the combination between nucleation and propagation which describes the level of risk associated with the node.

### 5.5.1 Mataka's theory results

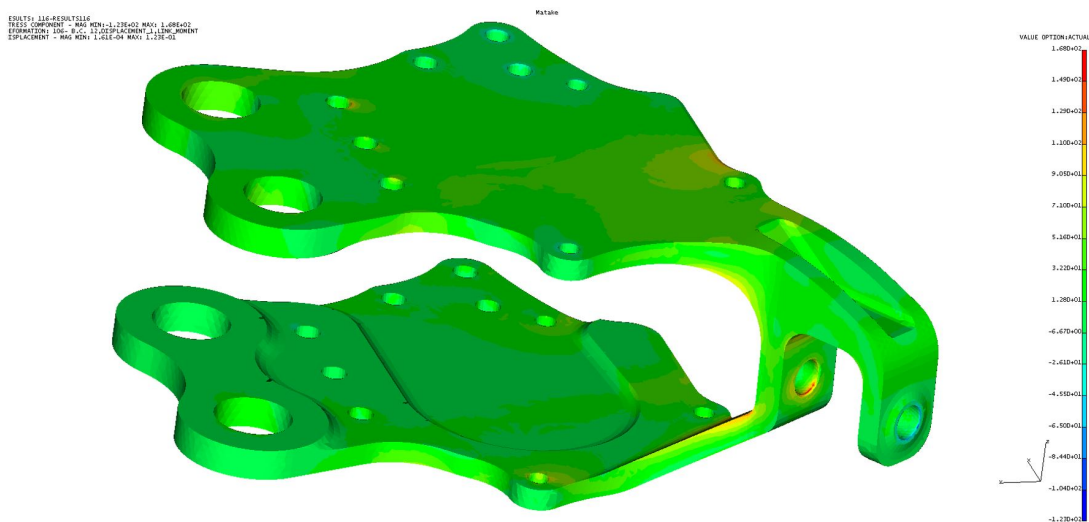
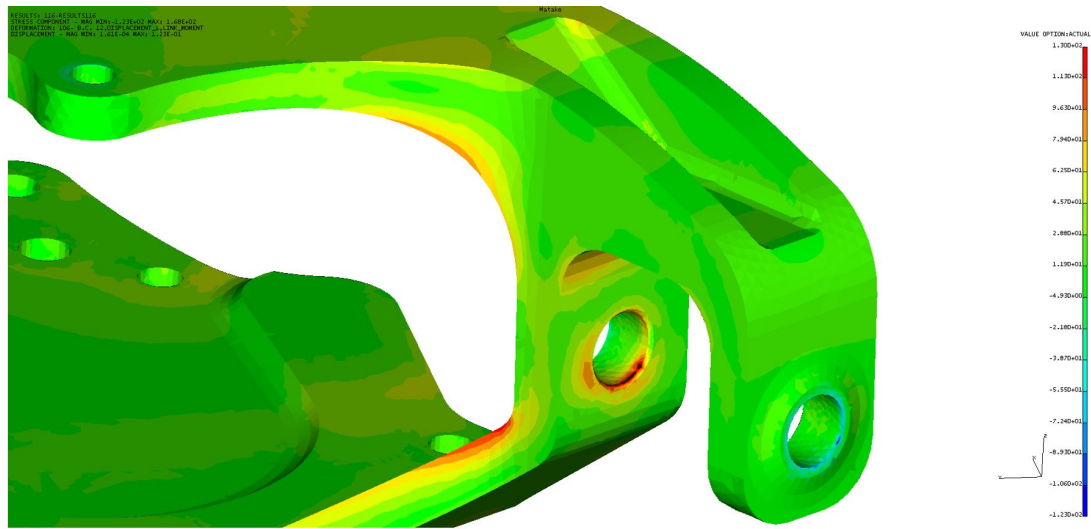


Figure 5.14: First side with Mataka's results.

Analysing the Fig. 5.14 you can clearly appreciate the part is not so much critical if you are talking about crack enucleation. As you could previously suppose the both flanges, upper and lower, do not show particular nucleation behaviour. You realize the hole area, where the



First side with Matake's results

Figure 5.15: Hole region with Matake's results.

connection between link and lever takes place, is one of the riskier area. Thanks to Fig. 5.15 we better examine the location. Talking about the hole, the edge is printed with black colour because that values are out-of-scale due to aforementioned problems. Furthermore, you can see a quite extended red and orange region which is quite significant for our study. The notch factor plays a key role in the amplification of stresses all over the hole region. There are more factors which play around the hole: first, the load is the most dangerous for fatigue cycle; second, the direction of application which is perfectly orthogonal to the lying plane of the lever, finally the notch which amplifies all the problems. Moving our point of view to the external area you can see a coloured map with colder colours. Thanks to the legend we relate the colours with compressive equivalent stress. This area shows a dual behaviour if compared with the other hole. This results are congruent with the phenomenology modelled in the FEM analysis.

Another results from the figure is the edge of the pitch horn link, the area tilted by  $45^\circ$ . In order to better understand that location I show another side, the reference plot are described in Fig. 5.16 and its relative zoom in Fig. 5.17.

Globally you can see how the flanges are not stressed even by this side of the body, but the most relevant area is the linking arm, the area tilted by  $45^\circ$ . The zoom is useful to better highlight the local behaviour. With the aim of comparing the results you take as reference the Fig. 5.5 which has a totally different scale but it shows a similar map. The area 2 is correctly defined even with this theory, that area it will be the most demanding at all because it superimposes to a critical propagation cycle a critical nucleation stress. Talking about areas 1 and 3 they show a nucleation stress similar to the 2 area but the propagation cycle presented before is not so hard and that nodes are critical in relation to nucleation but not to propagation. The nucleation criticism is related with notch factor, indeed these areas are located on the edges and near holes.

Numerically analysing the data shown by the graph I create the Tab. 5.10 which shows the results for the previously selected nodes, the ten nodes already described with the principal stress method. The nodes, taken as benchmark, are included between the thirty most stressed nodes of whole structure. This statement validate Matake's theory in the research of the critical areas.



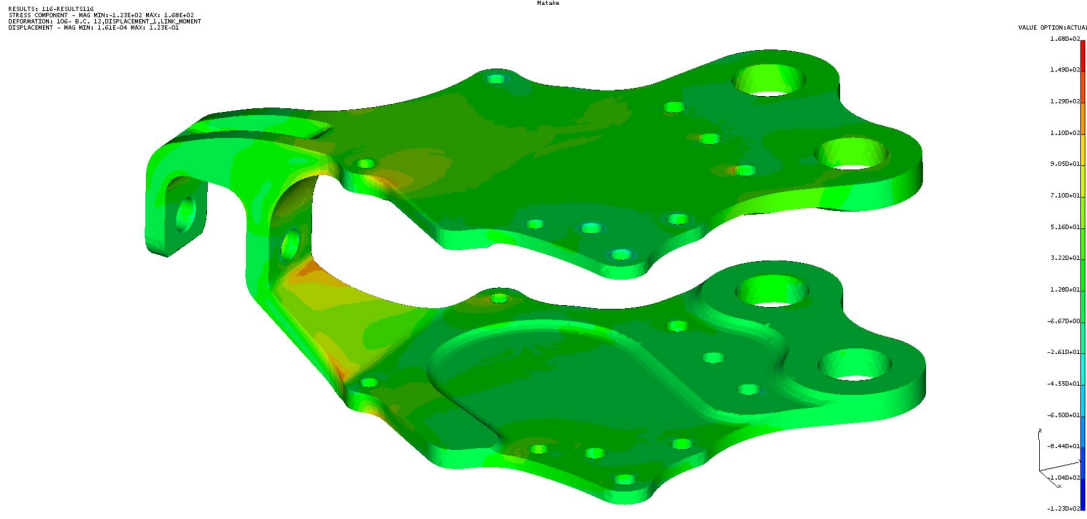


Figure 5.16: Second side with Mataka's results.

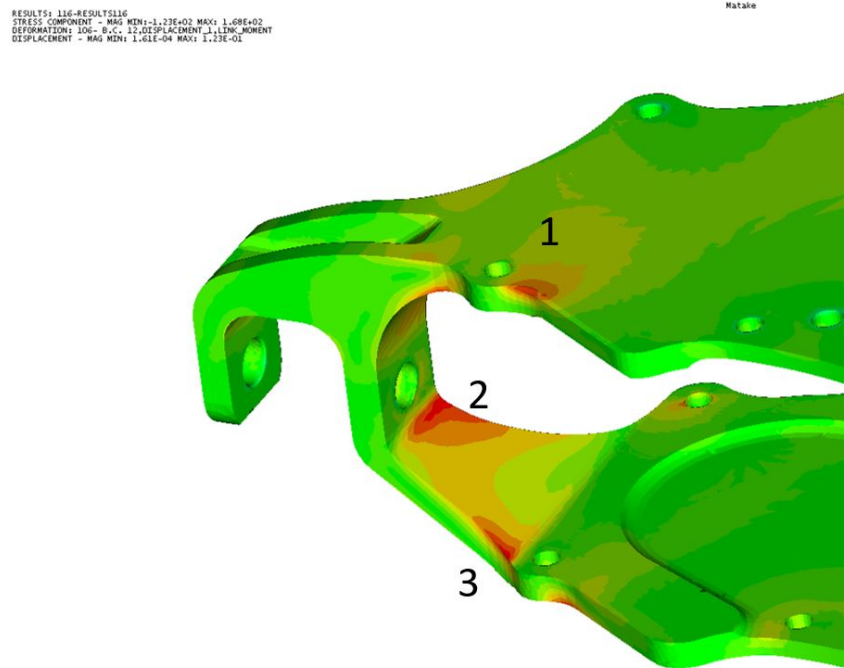


Figure 5.17: Zoom over the fractured area.

The table shows both critical angles,  $\varphi$  and  $\theta$ , which define the most troublesome plane following the scheme proposed by the Fig. 2.10. The very last column describes the margin of safety evaluated as follow:

$$MS = \frac{\sigma_{ADMISSIBLE}}{\sigma_{ACTING}} - 1 \quad (5.2)$$

As you can see the margin is below zero, it has a negative value. With this data the method informs us that those nodes are critical because the inequality is not respected.

Results with Mataka's theory					
Row	Node ID	Equivalent stress $\sigma_M$ [MPa]	$\varphi_{CR}$ [deg]	$\theta_{CR}$ [deg]	Safety margin
5	1037485	1.574e+02	242.40	58.78	-5.235e-01
12	1037497	1.458e+02	183.70	62.45	-4.857e-01
25	1148524	1.276e+02	286.50	132.20	-4.123e-01
3	5030520	1.593e+02	58.78	33.06	-5.292e-01
1	5030521	1.682e+02	242.40	150.60	-5.540e-01
17	5030532	1.396e+02	51.43	36.73	-4.628e-01
28	5032567	1.265e+02	139.60	165.30	-4.071e-01
33	5032808	1.231e+02	279.20	176.30	-3.908e-01
31	5069106	1.241e+02	257.10	7.35	-3.958e-01
29	5069124	1.250e+02	205.70	11.02	-4.000e-01

Table 5.10: Mataka's theory global results in the reference nodes.

### 5.5.2 Mesoscopic approach results

Taking an advantage from this theory you obtain a range of results which is similar to Mataka's ones. The most important parallelism between these two theories is linked with the interpretation you give to the data: the most stressed node is which shows an easier crack nucleation behaviour than the others. The calculus strategy is completely different, in this approach the code performs a numerical integration among planes without searching the critical one. The result is averaged and it takes into account all the planes used to the space discretization in the reference volume.

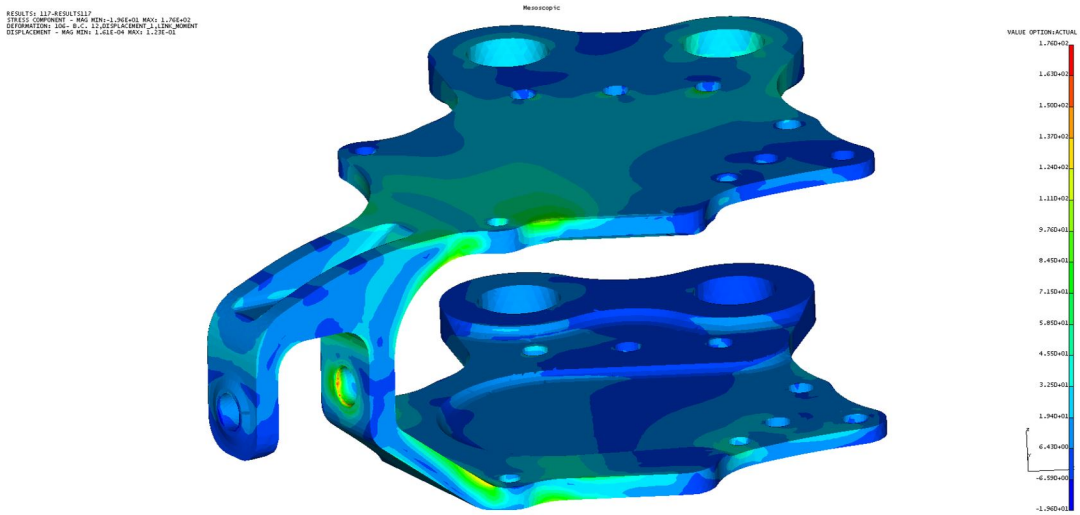


Figure 5.18: First side with Mesoscopic's results

Due to this pictures you start the examination of the results. If you make a parallelism with Mataka's theory you will discover that stressed areas are even in the flanges. Making the comparisons with the result extracted from FEM code I assert the colour map is similar between methods, this is a good point for the purpose of validate this method.

For sake of simplicity I do not repeat the reasoning made in Mataka's section because the



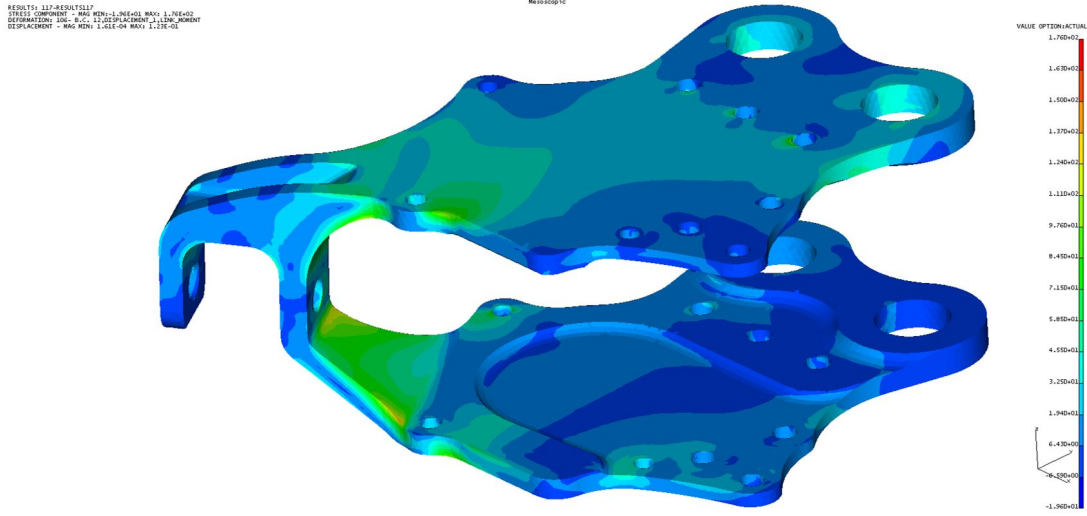


Figure 5.19: Second side with Mesoscopic's results

Results with mesoscopic approach					
Row	Node ID	Equivalent stress $\sigma_M [MPa]$	Resolved shear stresses $\sigma_{RSS} [MPa]$	Hydrostatic stress $\sigma_I [MPa]$	Safety margin
8	1037485	1.504e+02	1.434e+02	1.035e+02	-5.013e-01
9	1037497	1.494e+02	1.446e+02	6.955e+01	-4.978e-01
34	1148524	1.305e+02	1.240e+02	9.550e+01	-4.251e-01
10	5030520	1.490e+02	1.422e+02	9.943e+01	-4.966e-01
1	5030521	1.757e+02	1.694e+02	9.181e+01	-5.731e-01
14	5030532	1.446e+02	1.391e+02	8.042e+01	-4.813e-01
36	5032567	1.293e+02	1.227e+02	9.819e+01	-4.202e-01
41	5032808	1.259e+02	1.193e+02	9.662e+01	-4.044e-01
46	5069106	1.243e+02	1.169e+02	1.084e+02	-3.965e-01
49	5069124	1.217e+02	1.138e+02	1.162e+02	-3.836e-01

Table 5.11: Mesoscopic's theory global results in the reference nodes.

main features are the same. The key points of this analysis are similar.

## 5.6 Damage tolerance results

I completed the overview of the implemented criteria, now the solver development moves towards a post process stage. In this section the code gives to the user some other information in order to better understand the behaviour of the part under the imposed loads.

In aerospace engineering *Non-destructive tests* are a common practice in every day design and maintenance. Their first aim is to identify the presence of cracks or flaws on the part or inside of it, in order to substitute a broken component. Embedded cracks or flaws are not the central point of my study. Most of the time these cracks come from manufacturing process not from fatigue cycles. Talking about surface cracks their identification is vital for a correct maintenance of the helicopter or of each aircraft.

The design philosophy called “*damage tolerance*” could not work without crack propagation theories and laws. Indeed if you think that a micro-crack exists since the beginning of component life you need to know how this crack will grow across the body during operative life. This is fundamental to correctly quantify useful life of the part before the cracks becomes critical. The most relevant task assigned to the code is a post process which compares a reference curve with all the nodes. The procedure is well described in the Chapter 4.5. The curves are described in  $R - \sigma_{dyn}$  plane and they are called *No-growth curves* because they distinguish which nodes propagates flaws and which not.

In order to results standardization I chose two sizes and two locations for each material, the choice has been suggested by the the internal department know-how. In the Tab. 4.2 I report the most relevant informations of the reference material used for the manufacturing of the pitch lever.

Material data	
Material	Al 7475 (Zn 5.2-6.2; Mg 1.9-2.6; Cu 1.2-1.9; Cr 0.18-0.25)
Heat treatment	T7351 (Quench and artificial ageing)
Mechanical treatment	Shot peening
Flaw's shape	Elliptical on the surface or elliptical on the edge
Flaw's sizes	$a = 0.25$ mm o $a = 0.38$ mm
Aspect ratio	$a/c = 1$

Table 5.12: Reference material main properties.

Experimental tests are a basic step in this analysis in order to correctly understand the behaviour of the specimen with different types and with different sizes of cracks. Moreover, different load cases create more variability that should be studied.

Before entering the results paragraph, I have to precise that this post process stage is accessible if the previous analysis is performed with principal stress method. This is a necessary limit because the code must relate the imported graph with an equivalent uniaxial time history which is evaluated with only one method. In order to perform the equivalence between curve and numerical data I translate each node in  $R$  and dynamic stress and I plot in the same graph of the no-growth curve.

The presented images reveal a huge amount of critical node for a part which is in advanced design stage. The explanation of this not common graph is to find in the load spectrum applied to the part. Indeed to be consistent with previous steps I use the same load for all the graph. In other words, I studied the damage tolerance with fatigue test loads which are more demanding than flight loads. The experimental test is born to validate the part and the loads must envelope all the possible situation which the components must endure in its operative life. On the other hand, crack life evaluation is performed with flight test data or with a load spectrum not corrected with the same safety coefficients of the fatigue test.

All the figures show some green dots at  $R = -2$  and  $R = -3$  values. These values are brought as reference value from the useless cycle. They are set by the code automatically. Let's focus on the nodes which are filtered by the code:

- $R \sim 1$ : we are in the area where the experimental curve is numerically created. The code interpolates the experimental data and it extends the curve up to  $\sigma_{dyn} = 0$ . This particular behaviour comes from  $R$  value, the cycles with this stress ratio have a mean stress close to the yielding limit of material and the alternate stress is a small percentage of the mean.

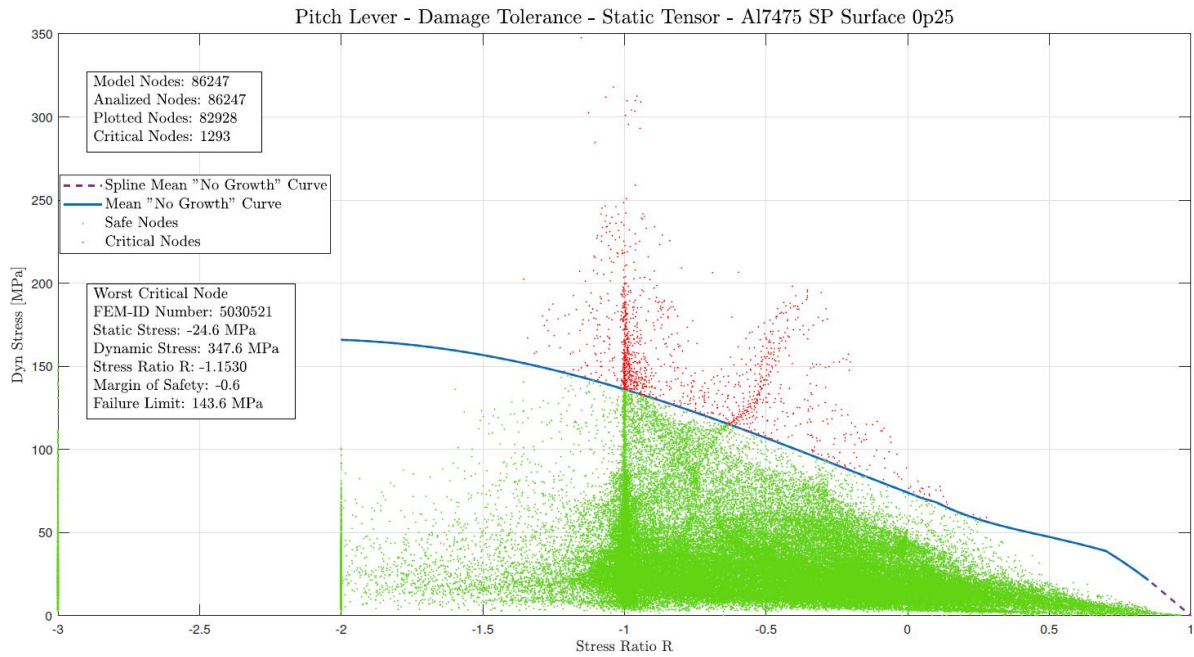


Figure 5.20: Own direction rotation (Method 1) with a surface crack of 0.25 mm.

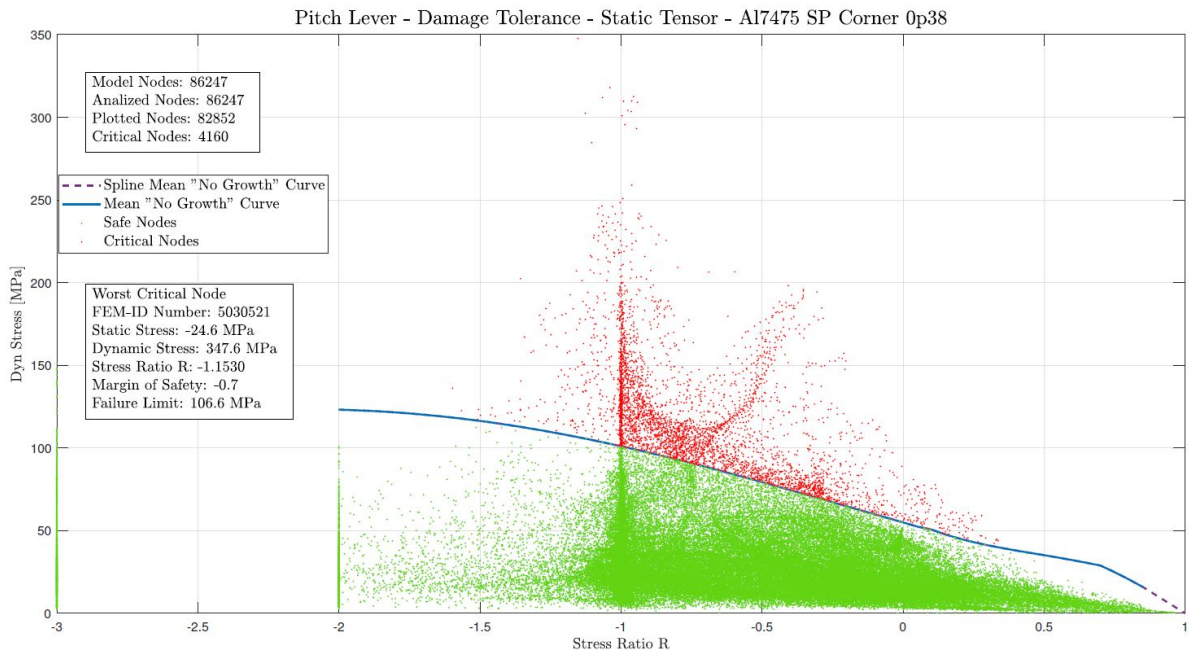


Figure 5.21: Own direction rotation (Method 1) with a surface crack of 0.38 mm.

We are in oligo-cyclic fatigue which degenerate into static load when the stress ration is equal to 1. In this case the code creates the spline curve but all the nodes with stress ratio greater then 0.98 are considered not critical. With few algebraical passages it is possible to demonstrate that the threshold limit set to  $R = 0.98$  correspond to a dynamic stress lower than 1% of the static value. Both conditions give you the same results.

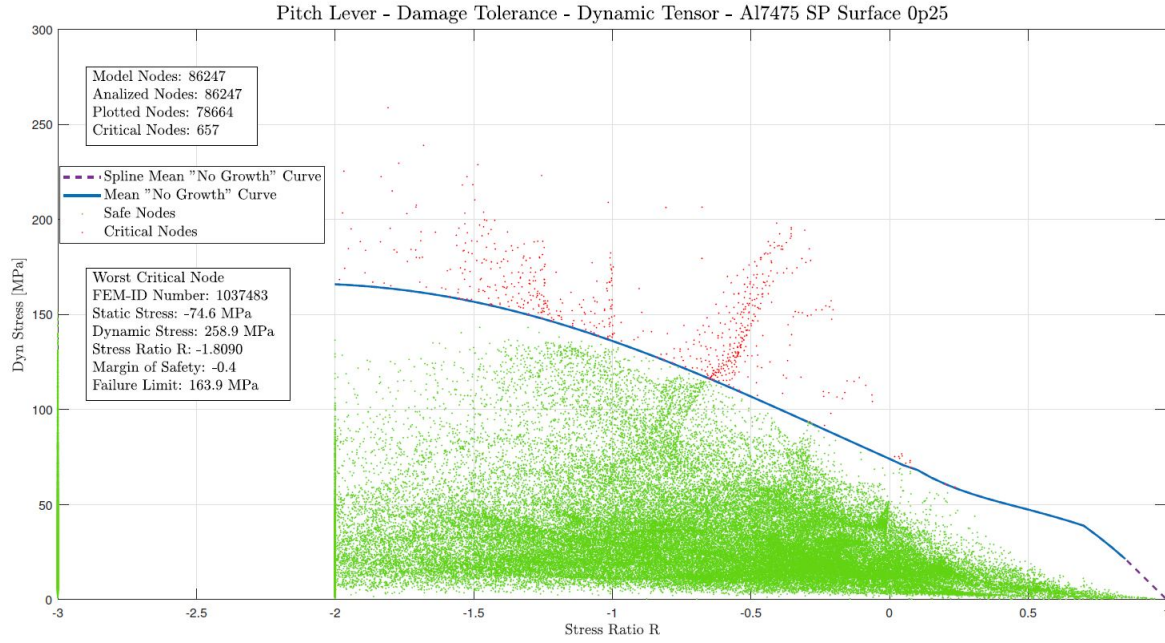


Figure 5.22: Dynamic direction rotation (Method 2) with a corner crack of 0.25 mm.



Figure 5.23: Dynamic direction rotation (Method 2) with a corner crack of 0.38 mm.

- $R > 1$ : in this situation we have a similar approach to the previous if the value is close to one. If you have  $R = 2$  the situation is quite different. You can think about:  $R = \sigma_{min}/\sigma_{max} = 2$ , then  $\sigma_{min} = 2 \cdot \sigma_{max}$ , but this relation produce a correct result with both stress lower then zero. A weary cycle with stress lower then zero in each second has no possibility to show crack propagation, you need positive stress to force the opening of crack

apex. In the graph all the nodes with a stress ratio greater than one are not critical and their stress ratios are modified and set to  $R = -2$ .

- $R < -2$ : now we have a slightly different cycle which has a percentage of compressive stress and a percentage of positive one.  $R = -1$  is the fully reversal cycle with equal percentage of positive and negative stress, if you decrease below  $R = -2$  you will consider the cycle as not critical, that value of the stress ratio told us that the 66.7% of the cycle is compressive. All the nodes with a stress ratio lower then  $R = -2$  are not critical and their value is set to  $R = -3$ .

Discussing about charts with own tensor rotation, Fig. 5.20 and 5.21, you can fast identify the most serious nodes linked with the behaviour of the pin-slot-hole, moreover there is the FEM ID number of the most critical node which is the same of the tables. This is a crucial point because the identification of the worst node is automatically implemented in the GUI and thanks to that data I am sure of a correct implementation. In the dynamic tensor rotation the graph are slightly more complicated and the parallelism does not came out immediately. Thanks to Tab. 5.3 and the chart you can perform a comparison. There is not a connection between the critical nodes, but in the table the stress ratio values are very deep, around minus two, others are even deeper. These values have the largest dynamic stress but due to the static one the stress ratio is into a not-critical area. The code filters these value according with the aforementioned ranges. In sake of exactitude they are less critical than others.

The nodes with a value similar to  $R \sim -0.3$  and a dynamic stress up to  $200\text{ MPa}$  are surely more interesting because the weary cycle is absolutely more demanding because the stress is close to the absolute maximum but the stress ratio makes the situation worse. The real check the user could be perform owing to this graph is not linked with the single node but with a “group” of them. It is mandatory to identify if there is an area with lots of nodes which show a critical behaviour, in fact some nodes with absolute stress value far from the absolute maximum, might become a problem due to the amount of critical nodes and their combined action. I assert the most valuable result is the identification of the most critical areas not the absolute maximum value of stress. It is important to identify the areas with the highest percentage of fatigue cycle with positive stress, which means that the range of the stress ratio is:  $R \in [-1; 0]$ .

To perform this task it is a crucial point the possibility to extract the critical nodes only, translate them into a binary format and plotting over the FEM model. Saved the necessary informations, I generate the following images thanks to the FEM model. I remind you that the saving procedure is automatically implemented in the tools. In Fig. 5.24 and 5.25 you observe the last one result of the analysis, the representation of the critical nodes on the model.

The first figure shows the static stress in a well known location. You can see that, also with this approach, the result is in agreement with the previous steps. The pitch horn is a critical area under every aspect of the analysis. During the operative life of the component the maintenance will be scheduled in order to control the presence or not of the damage. Both images reveal the correct approach presented in the chapter: the dynamic stress is the most demanding value but the static one sets the stress to a level which is always positive, this create the worst condition ever.

Concluding this section I assert that the most suitable result is the availability of the entire process, the speed and fluency of the whole work-flow which add value to the simply graph. Comparisons among materials, saving of data and communication with FEM are the essential feature of this section. Last but not least there are the numerical and graphic results.

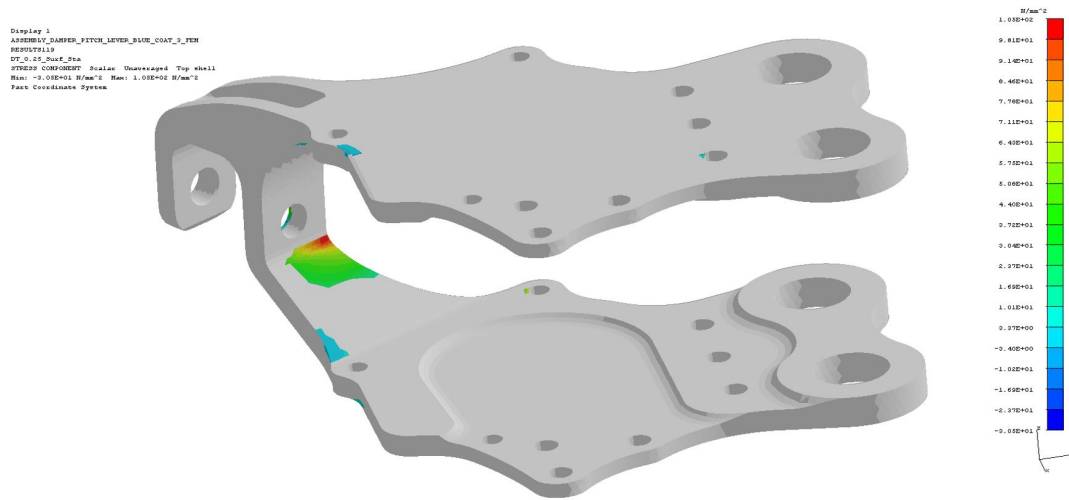


Figure 5.24: Static stress of critical nodes evaluated with “No Growth” control.

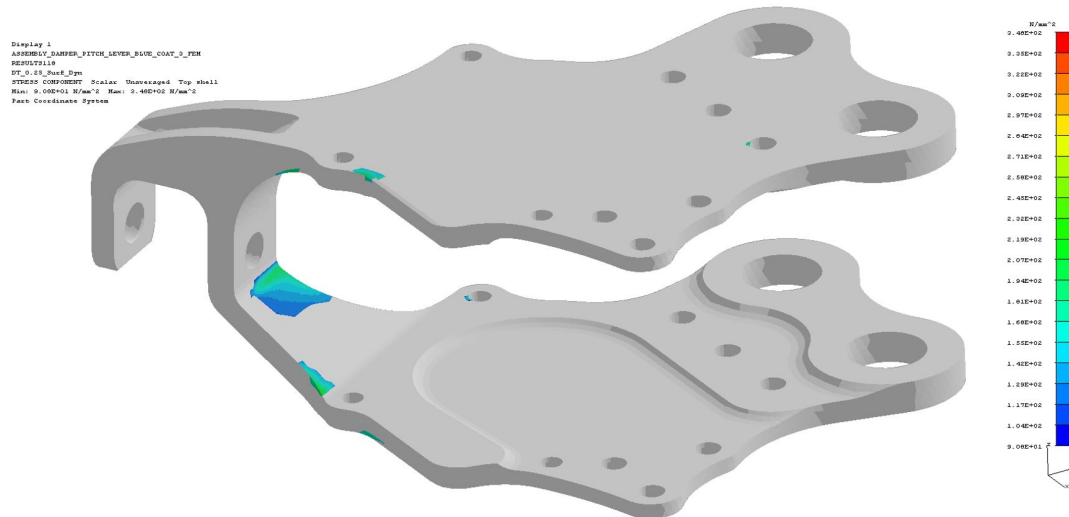


Figure 5.25: Dynamic stress of critical nodes evaluated with “No Growth” control.

## 5.7 Fatigue curves and experimental comparison

Another notable passage is the comparison between numerical and experimental data. The comparison is performed with fatigue curve of the material, well know as Wöhler curve, the experimental failure and the numerical information.

Analysing the graph shown in Fig. 5.26 you can immediately understand the power of the shot-peening procedure in order to raise the fatigue performances of the part. You can examine the difference between the numerical result performed with the tool and the experimental data. In sake of clarity the numerical results are reported in the Tab. 5.13. If I relate the absolute results the ratio is approximately three. This not negligible difference is a safety data because the numerical life is shorter then the experimental one. Thereby with the only numerical solution I am under-estimating part’s life. This is not correct because if you estimated the life in a wrong way, you might replace a part which is still working good. This is an economical problem because



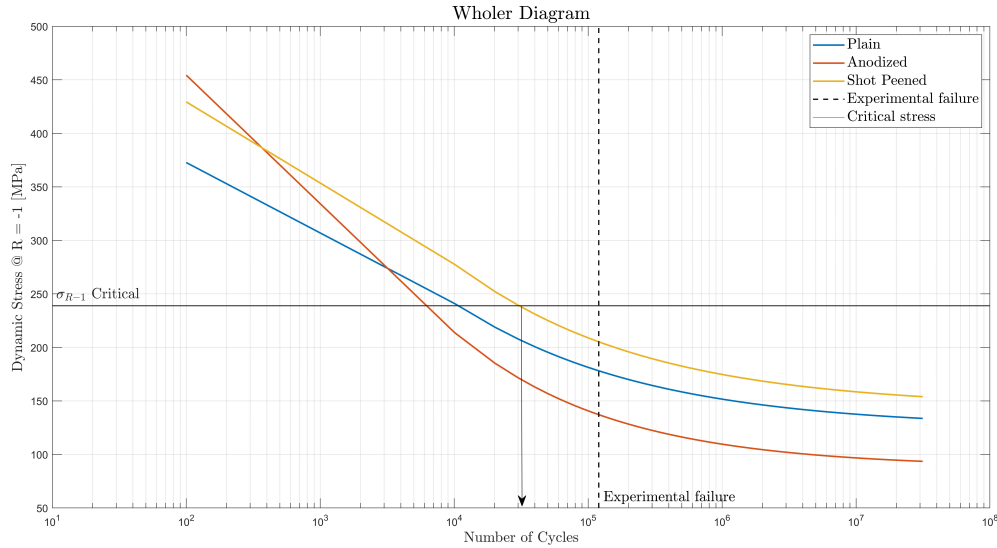


Figure 5.26: Wöhler diagram with reference material fatigue curve: experimental and numerical results.

you are wasting a serviceable part. This type of study is beyond the scope of the thesis and I can only confirm that the numerical analysis gives you a conservative results if compared with the experimental test.

Experimental comparison	
Failure estimation with Wöhler curve	Experimental test failure
38000 cycles	120000 cycles

Table 5.13: Comparison between experimental and numerical results

I try to research an explanation to this difference but it is a complex task. I'm sure about accuracy and precision of numerical calculus, the FEM model is extremely detailed as well. In my own opinion the motivation about this gap is due to fatigue behaviour inherent variability. There are several factors which play a central role in the extension of component life. During fatigue test there were some stops owing to guarantee checks and routine inspections of the parts. There is a strong relationship between the last one cycle and the residual stress in the component, moreover the raising of the dynamic stress could modify the life of the component. This and the variability which is hidden in the experimental practice are the main actors in the gap creation.

In the end I want to highlight the failure location with a real image of the component, in the Fig. 5.27 you can see the crack which causes the failure of the pitch lever. The crack nucleates and propagates the region previously defined.



Figure 5.27: Picture of the are where the failure occurred.

## 5.8 Time management

As described in the Chapter 3, the whole evaluation can be performed with both loop strategy: sequential loop or parallel loop. This second one scheme take an advantage from MultiCore structure of the modern CPUs ad it solves more than one node at the same time.

The time study which is described in the dedicated chapter is a useful guide to understand the global time of the solution, now at the end of tool development there is a basic question: where is the turning point between one strategy and the other? This is not a pointless question because the number of node to analyse is not the only factor which I consider. If you talk about the only solution section the parallel evaluation strategy should be the fastest in every situation. This statements is not completely correct due to transfer time, parallelization time and staring time. Let's try to better understand what I mean. Owing to the possibility of the stand-alone version to work in different computers the number of cores is not a fixed value. The code detects the maximum number of cores and it uses all of them if required. This stage requires a amount of time that is not negligible:

Time to activate parallel evaluation.	
Average on 10 starts:	12.78 sec

Table 5.14: Parallel environment starting time.

From Tab. 5.15 and with the relative graph in Fig. 5.28, you understand that with few hundreds of nodes the parallel loop strategy is convenient. In the graph parallel computing curve is already corrected with the activation time described into Tab. 5.14. The crossover between curves has that position due to the study requires lots of evaluations for each node,



Sensitivity matrices solver time.		
Nodes	Sequential evaluation [s]	Parallel evaluation [s]
1	0.054	2.060
10	0.298	0.490
100	2.684	1.08
1000	26.856	7.191
10000	258.735	74.041
100000	2721.138	646.661

Table 5.15: Tool evaluation time with sensitivity matrices solver.

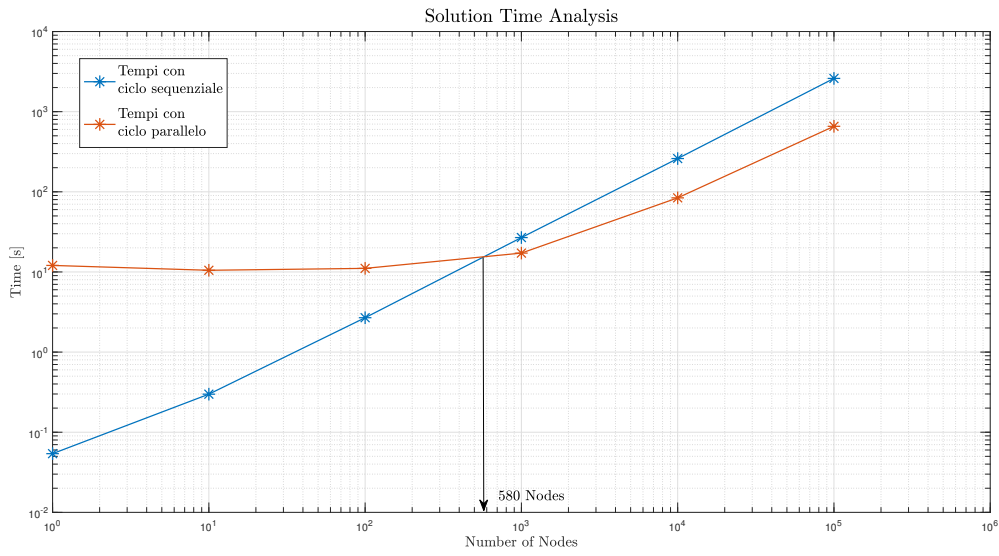


Figure 5.28: Bi-logarithmic graph to compare results of sensitivity matrices solver.

thirty-five in the examined case. With the parallel toolbox the code solves thirty-five time step on each core and the time saved balances out the time wasted for the starting procedure.

reduced step solver time		
Nodes	Sequential evaluation [s]	Parallel evaluation [s]
1	0.0174	1.4500
10	0.0196	0.176
100	0.0580	0.251
1000	0.072	0.371
10000	0.612	0.698
100000	6.020	1.769

Table 5.16: Tool evaluation time with reduced step solver.

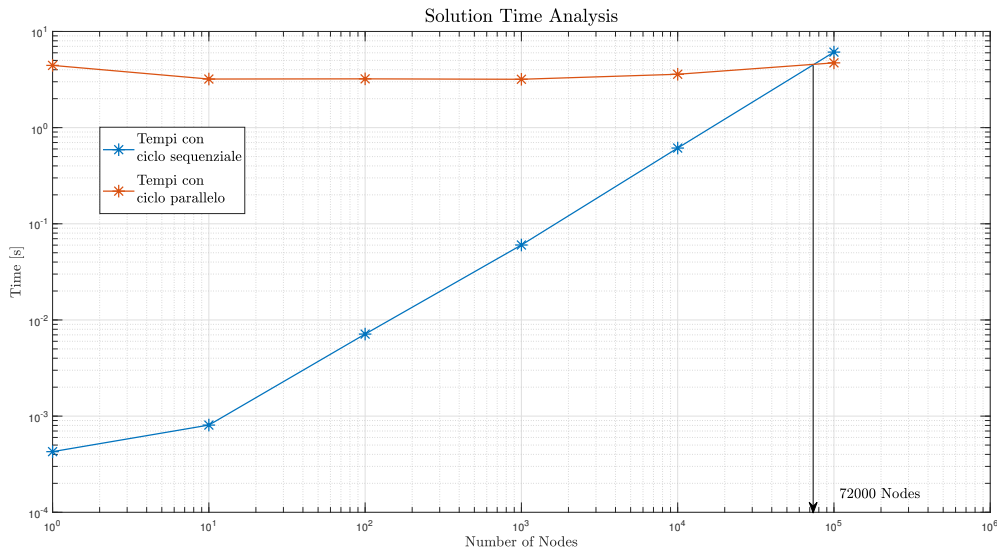


Figure 5.29: Bi-logarithmic graph to compare results of reduced step solver.

The Fig. 5.29 is already corrected with starting procedure time. Thanks to the graph and the Tab. 5.16 you can see the parallel evaluation is useful only in problems with more than hundred thousand nodes. Only when you are analysing the whole model it is useful to activate parallel loop strategy. The time steps which are parallelized by the code are only two and you need more and more nodes to pay the starting time out.

In order to use this tool I need to share it with other computers. There are two ways to do this: stand-alone version or MATLAB<sup>®</sup> app. In this second case you need the licence and the application to correctly use the GUIs but the main code maintain the parallel environment open when you close the GUI, thanks to this trick you wait the starting time only once. Otherwise, is you work with stand-alone version every time you open the GUI you need to re-create the parallel environment from the beginning. The most used version is the stand-alone and the time problem is to take into account as described.

## Chapter 6

# Conclusions and final remarks

This thesis work passed through many development phases. In the very first one, the attention was posed on implementing all the solvers in MATLAB scripting language with the final aim to achieve a better time-to-solution without affecting the precision of the computation. At a second instance, the main purpose of the activity switched the attention to closing the gap with real practical applications, therefore analysing many different cases in terms of input required for a wide range of problems, always keeping in mind the requirements for both the GUI interfaces, with all the modelling capabilities, and the FEM analyses as well: it has been crucial the understanding of which approach would have been better to follow to get specific sets of information one might be interested in drilling-out from the same component.

Hence, all the data structures suitable for the application have been identified depending on their contribution to the different computations one may want to perform employing the tool described herein, in other words: the stress tensor (evaluated at the nodes) and the sensitivity matrices for the reference model.

The analysis performed with the aid of the sensitivity matrices allows the user to shed light on the interaction between the various loads acting on the component being studied, pointing out the sensitivity that each node has with respect to the load itself. The final goal of this approach is to find and evaluate the most critical working conditions for the component once the load spectrum has been defined. In fact, once the load the user wants to use is available, one can directly get precious results in terms of critical stress threshold to estimate the fatigue-life of the component itself: what the user only needs to do, is simply to modify the inputs in the same session as the one he is currently working on, thus leading him to quickly identify all the different existing cases and weight the effects on the fatigue-life behaviour, creating a series of useful datasets for both graphical and numerical further comparisons.

To guarantee the designer a wider variety of information, 3 different analyses paths have been pursued, each one approaching the fatigue-life problem in a different way. The various theories considered investigated nucleation phenomena and defects growth, to get a better understanding on which are the most critical nodes in the model and how much critical they are.

Moreover, considered the typical dimension of the problems derived from the modern FEM modelling techniques, one should find of practical interest or, even more, necessary, to approach computation strategies capable of delivering the highest performances in terms of speed and time needed to complete the required analysis. The main goal is to speed-up as much as possible the computational operations needed to complete the whole investigation. Hence, parallel computing showed to be the right path to follow to sensibly improve the solver global performances, however, one should not neglect the huge potential related to modern GPUs (Graphical Processing Unit),

which can provide and guarantee sky-high improvements in the next future.

To ensure quick results, a reduced solution based on stress-steps instead of the whole analysis has been adopted. These load steps are already available from the FEM model and this led to a drastic reduction of the time-to-solution. This operation, unfortunately, inherently brings with it the introduction of an error related to the very small availability of known working conditions and, consequently, procures the impossibility of identifying with full confidence the critical instant. It has been shown that the two GUI interfaces, starting from the same numerical input, give the same numerical output, however, such verification results from a mere mathematical check during the implementation phase and the two interfaces generate an impalpable discrepancy due to the unfeasibility of an a-priori and uniquely determined identification of a critical condition for all the nodes taken into consideration.

Under the light of the results achieved so far with the resource depicted and extensively treated in this thesis work, one can prove the full integration of the computational tool with both the FEM model and the probatory experimental test. Although the latest showed a discrepancy with the numerically computed data by the tool, we can confidently say the tool offers a valid and sound computational alternative to deeply understand the criticalities of a metallic component being part of a helicopter rotor assembly. On top of that, the two approaches studied herein allowed to make evident how the phase-shift between the different load curves led to have some peculiar stress combinations that cannot be spotted nor quantified in advance solely given the load history. The two interfaces I have created during this thesis, especially under the light of the aforementioned considerations, must be used in their own specific applicative field to avoid any conflict from arising. A FEM evaluations-based analysis, quick in terms of time needed to get the solution and quite slim/slender in the pre- and post-processing phases, originate under the hood of the preliminary analyses, when it is of crucial importance to understand whether the initial design is good enough and compliant with the project requirements or not. Knowing the sensitivity matrices finds its own place certainly at a later time, downstream the preliminary analysis, for sure in a more advanced design phase and its usage unlocks the potential for investigating finer details, like the interaction of different loads and the effects of the loads phase-shifts on the operating-life of the real component.

Besides the numerically-related and implementation issues, the biggest challenge of this project was related to achieve the required usability and flexibility targets set at day one that the codes were asked to possess in order to be applied to any sort of analysis regardless the dimension of the manufacture the user wants to investigate. Generating an ad-hoc graphical user interface and thinking from scratch on how to implement what resulted in the final tool, which I am very proud of, have always been a never-ending source of new challenges and personal enrichment in the field of MATLAB programming techniques. Phases like the input data manipulation, processing and elaboration, making the GUI able to effectively communicate with other software and platforms exchanging information via other data formats are just few ones of the steps I had to climb up. All the work conducted span around two main factors, which are crucial in this specific application: time-to-solution analysis and computational precision.

Studying the fatigue phenomenology cannot be unrelated to a wide testing campaign for material characterization, to evaluate its fatigue limits and the results of the various mechanical operations it could undergo during the production process in terms of enhancement/reduction of its fatigue resistance and stress field given the same load being applied. Moreover, once the component has been assembled, the final in-field testing and evaluation phase must always strengthen and corroborate the numerical results: in fact, the FEM modelling, although it may be very fine and accurate, especially when paired with a very detailed fatigue simulation of the

---

component, might be unable to foresee typical effects that the component will show during the real service on the helicopter. So, it will always be essential to couple the experimental testing with the simulation activities, to identify in advance as many critical areas as possible before the component gets approved and used on the helicopter.

Lastly, the next steps of this work will be about two aspects:

1. Further improvements and slimming down the procedures related to using the interfaces: this first one is mainly due to specific technical requirements of the design office, namely the feasibility of loading the analyses results in one single click with no need to re-run the whole computation to quickly post-process them. So, it would be highly beneficial to find a way to store them in a repository, potentially exploiting the most recently added MATLAB features related to cloud sharing (e.g., GitHub or similar platforms).
2. The implementation of new functionalities linked to the evaluation of the components fatigue-life usage limits. With no doubts, giving the code the potential to know in advance the Wöhler curves may lead to a direct implementation of the Miner-Palmgren criterion, hence giving a close relation to the self-life asserts; or, alternatively, implementing some laws describing the defects growing behaviour may enable the tool to foresee the fatigue-life of the part being investigated thanks to the help of damage-tolerance criteria.



## Capitolo 7

# Conclusioni

Il lavoro svolto si è articolato in diverse fasi di sviluppo. Nella primissima fase ci si è concentrati sull'implementazione dei solutori in ambiente MATLAB<sup>®</sup> con la finalità di migliorarne i tempi di calcolo senza intaccare la precisione del risultato. Come secondo step ci si è avvicinati alla pratica andando ad analizzare le diverse casistiche in termini di input del problema, vista l'interfaccia con le modellazioni e le analisi agli elementi finiti gli input sono giunti proprio da tale approccio. Sono state identificate le strutture dati che possono essere calcolate e che risultano utili al nostro studio, identificate nel tensore di stress valutato ai nodi oppure le matrici di sensibilità calcolabile per il modello di riferimento.

L'analisi svolta con l'ausilio delle matrici di sensibilità permette di studiare l'interazione dei diversi carichi con la struttura alla luce della sensibilità che ciascun nodo ha rispetto al carico stesso. Lo scopo di tale studio è la valutazione di quali siano le condizioni critiche una volta fissato lo spettro di carico sul componente. Infatti, noto il carico che si vuole analizzare è possibile in modo diretto ottenere un risultato in termini di stress critico per la vita a fatica, modificando gli input nella medesima sessione di lavoro si possono identificare diverse casistiche e valutarne gli effetti in termini di vita a fatica, creando una serie di dati utili al raffronto sia grafico che numerico.

Al fine di garantire una maggior varietà di informazioni al progettista sono state perseguite 3 strade di analisi, ciascuna legata a un diverso approccio al problema della fatica. Le diverse teorie indagano fenomeni di nucleazione e di accrescimento del difetto, per comprendere quali siano i nodi critici del modello e in che misura siano critici.

Considerata la dimensione tipica dei problemi derivanti dalle moderne modellazioni agli elementi finiti risulta di interesse pratico, anzi necessario, indagare strategie di calcolo sempre più rapide che permettano di accelerare l'esecuzione globale dell'intera analisi. Il calcolo parallelo è la strada che è stata scelta per migliorare le prestazioni globali del solutore; ma anche le potenzialità delle GPU (Graphics Processing Unit) moderne possono fornire grandi prospettive di miglioramento futuro.

Al fine di garantire soluzioni rapide si è indagata una soluzione ridotta basata su step di stress e non sull'analisi completa, tali step di carico già noti dal modello FEM permettono una soluzione con tempi di calcolo che si riducono in modo drastico. Tale operazione porta con sé l'introduzione di un errore legato alla risicata presenza di condizioni note e conseguentemente all'impossibilità di identificare con assoluta correttezza l'istante critico. Si è dimostrato che le due interfacce partendo dallo stesso input numerico generano lo stesso risultato numerico, ma tale verifica risulta un mero controllo in fase di implementazione e le due interfacce generano una

discrepanza essenzialmente legata all'impossibilità di identificare a priori una condizione critica univoca per tutti i nodi.

Alla luce dei risultati ottenuti con il tool sviluppato possiamo affermare la completa integrazione del tool di calcolo in relazione al modello FEM e alla prova sperimentale. Sebbene quest'ultima abbia presentato una discrepanza con il dato numerico possiamo considerare il tool una valida alternativa di calcolo per comprendere le criticità di un componente metallico.

Inoltre i due approcci studiati hanno permesso di palesare come lo sfasamento tra le diverse curve di carico porti con sé delle peculiarità di stress sul componente non quantificabili a priori con la sola storia di carico. Le due interfacce che ho creato nel corso della tesi, proprio alla luce delle considerazioni fatte, devono essere utilizzate nel rispettivo campo di applicazione, senza conflittualità. Un'analisi basata sulle valutazioni FEM, rapida nel calcolo e snella nelle fasi di pre e post processo nasce nell'ambito dell'analisi preliminare in cui è necessario capire se le scelte progettuali sono state prese nella direzione corretta. La conoscenza delle matrici di sensibilità si colloca sicuramente in una fase più avanzata della progettazione e il suo utilizzo permette di indagare dettagli più fini, come appunto l'interazione tra diversi carichi e l'effetto che gli sfasamenti hanno nella vita operativa del componente.

Oltre alle problematiche numeriche ed implementative la grande sfida del progetto è legata alla fruibilità, nonché alla flessibilità che i codici devono avere per poter essere applicati a qualsiasi problema in modo indipendente dalle dimensioni che lo stesso può avere. Generare una interfaccia grafica dedicata e pensare ex novo una strategia numerica per garantire ciò è stato fonte inesauribile di nuove sfide e di arricchimento personale nell'ambito di programmazione in ambiente MATLAB<sup>®</sup>. Fasi come la manipolazione dati in ingresso, l'elaborazione degli stessi, la comunicazione con altre interfacce e con altri formati dati, sono solo alcuni dei passaggi affrontati. Tutto il lavoro si è sempre incernierato attorno al controllo dei due fattori determinanti in tale campo: l'analisi dei tempi e la precisione del calcolo ottenuto.

Lo studio della fenomenologia della fatica non può prescindere da una campagna di prove sperimentali per la caratterizzazione del materiale, al fine di valutare i limiti di fatica e l'azione delle diverse lavorazioni che il componente può subire. Inoltre, a componente assemblato la prova sperimentale dovrà sempre corroborare i risultati numerici in quanto la modellazione per quanto accurata, sommata anche l'analisi a fatica di dettaglio potrebbero non prevedere effetti tipici che al vita operativa reale del componente paleserà in corso d'opera. Risulterà quindi sempre imprescindibile la prova sperimentale sebbene una analisi di dettaglio permetta l'identificazione preliminare delle criticità presenti nel componente.

Gli sviluppi futuri del lavoro svolto devono riguardare due necessariamente due aspetti:

1. Lo snellimento ulteriore delle procedure di utilizzo dell'interfaccia legate principalmente alle esigenze tecniche dell'ufficio, per esempio la possibilità di caricare un file risultati per studiarne il post processo senza dover svolgere nuovamente l'analisi. Darebbe grande beneficio trovare una metodologia implementativa per il salvataggio su cloud delle informazioni sfruttando a pieno le nuove potenzialità offerte da MATLAB come per esempio GitHub.
2. L'implementazione di nuove funzionalità legate alla valutazione della vita utile del componente. Sicuramente la conoscenza da parte del codice delle curve di Wohler può permettere l'implementazione diretta dei criteri di Palmgren-Miner fornendo un legame con i concetti *safe-life*; oppure l'inserimento di leggi che descrivono la crescita del difetto possono permettere la previsione della vita con l'ausilio dei concetti della filosofia progettuale *damage tolerance*.



## Appendix A

### Matake's theory solver

In Appendix A the MATLAB<sup>®</sup> function I implemented during my thesis project at the *Rotors D&D* department of Leonardo S.P.A. division Helicopters is reported. The function reported is the latest version generated and the one currently implemented in the actual GUI. Previous versions are described in Cap. 3.

---

```
1 function [ MATAKE , theta_cr , phi_cr ]...
2     = Matake_GUI_Tesi(S, k_M, l, m, n, phi, theta)
3 % ===== %
4 %             MATAKE'S THEORY SOLVER %
5 %             Andrea Casagrande %
6 %             Dynamic Systems – Rotors D&D – Leonardo S.P.A. %
7 %             Academic Year 2019/2020 %
8 % ===== %
9 %             evaluate critical condition according to Matake's theory
10
11 %% PRE-PROCESS
12 % 3D space discretization
13 theta2 = 2 .* theta;
14 phi2    = 2 .* phi ;
15
16 cp      = cos(phi) ;
17 sp      = sin(phi) ;
18 ct      = cos(theta) ;
19 st      = sin(theta) ;
20 c2p     = cos(phi2) ;
21 s2p     = sin(phi2) ;
22 c2t     = cos(theta2) ;
23 s2t     = sin(theta2) ;
24
25 %% PROCESS
26 % matrices preallocation
27 S_rot   = zeros(3,n) ;
28 sta     = zeros(1,m) ;
29 dyn     = zeros(1,m) ;
30
```

---

```

31 for i = 1 : l
32     for j = 1 : m
33         for q = 1 : n
34             S_rot(1,q) = ( - S(1,1,q) * ( 1 + c2p(i) ) / 2 ...
35                 - S(1,4,q) * s2p(i) ...
36                 - S(1,2,q) * ( 1 - c2p(i) ) / 2 ...
37                 + S(1,3,q) ) * s2t(j) / 2 ...
38                 +( S(1,6,q) * cp(i) + S(1,5,q) * sp(i) ) * c2t(j);
39
40             S_rot(2,q) = ( S(1,1,q) - S(1,2,q) ) ...
41                 * st(j) * s2p(i) / 2 ...
42                 - S(1,4,q) * st(j) * c2p(i) ...
43                 + S(1,5,q) * ct(j) * cp(i) ...
44                 - S(1,6,q) * ct(j) * sp(i);
45
46             S_rot(3,q) = ( S(1,1,q) + ( S(1,2,q) - S(1,1,q) ) ...
47                 * ( 1 - c2p(i) ) / 2 ) * ( 1 - c2t(j) ) / 2 ...
48                 + S(1,4,q) * ( 1 - c2t(j) ) / 2 * s2p(i) ...
49                 - ( S(1,6,q) * cp(i) ...
50                 + S(1,5,q) * sp(i) ) * s2t(j) ...
51                 + S(1,3,q) * ( 1 + c2t(j) ) / 2;
52         end
53         % Evaluate all the combination between rotated
54         % tensor components, in order to identify the critical one
55         aM2 = ( ...
56             ( ...
57                 ( repmat(S_rot(1,:),n,1) - ...
58                     repmat(S_rot(1,:),n,1)' ) .^2 ...
59                 + ( repmat(S_rot(2,:),n,1) - ...
60                     repmat(S_rot(2,:),n,1)' ) .^2 ) ...
61                 .^ 0.5 ) ...
62                 ./ 2;
63
64         % find the critical position
65         sta(i,j) = max( S_rot( 3, : ) );
66         dyn(i,j) = max( aM2( : ) );
67     end
68 end
69
70 % search between all the planes the worst condition
71
72 % rows which contains max value for each column of "dyn"
73 [max_dyn_r, IT_phi_max] = max( dyn, [], 1);
74
75 % between maximum values, I search the absolute one
76 [max_dyn, IT_theta_max] = max( max_dyn_r );
77

```

---

```

78 IT_phi_max = IT_phi_max( IT_theta_max );
79
80 % max hydrostatic stress in the critical plane.
81 % NB: we don't want the absolute max in 3D space, but the
82 % max value on the critical plane only.
83 max_sta = sta( IT_phi_max, IT_theta_max );
84
85 % Mataka's equivalent stress
86 MATAKE = max_dyn + k_M * max_sta;
87
88 % translate results in deg
89 theta_cr    = theta( IT_theta_max ) * 180 / pi;
90 phi_cr      = phi( IT_phi_max )      * 180 / pi;
91 end

```

---



## Appendix B

# Mesosopic approach solver

In Appendix B the MATLAB<sup>®</sup> function I implemented during my thesis project at the *Rotors D&D* department of Leonardo S.P.A. division Helicopters is reported. The function reported is the latest version generated and the one officially implemented in the actual GUI. Previous versions are described in Cap. 3.

---

```
1 function [MESOSCOPIC, max_hy, RSS]...
2     = Mesoscopic_GUI_Tesi ( S, alfa, l, m, o, n, phi, theta, chi )
3 % ===== %
4 %           MESOSCOPIC SCALE APPROACH SOLVER %
5 %           Andrea Casagrande %
6 %           Dynamic Systems – Rotors D&D – Leonardo S.P.A. %
7 %           Academic Year 2019/2020 %
8 % ===== %
9 %     evaluate critical condition according to Mesoscopic approach
10
11 %% PRE-PROCESS
12 % 3D space discretization
13 theta2 = 2.*theta;
14 phi2    = 2.*phi;
15
16 cp = cos(phi);
17 sp = sin(phi);
18 ct = cos(theta);
19 st = sin(theta);
20 cx = cos(chi);
21 sx = sin(chi);
22
23 c2p = cos(phi2);
24 s2p = sin(phi2);
25 c2t = cos(theta2);
26 s2t = sin(theta2);
27
28
29 %% PROCESS
30 % matrices preallocation
```

```
31 s          = zeros (l,m,n);
32 dyn        = zeros (l,m,o);
33 a_chi      = zeros (o,n);
34
35 for i = 1 : l
36     for j = 1 : m
37         for r = 1 : n
38             a_chi(:,r) = ( ( S(1,1,r) - S(1,2,r) ) * st(j) * s2p(i) / 2 ...
39                             - S(1,4,r) * st(j) * c2p(i) ...
40                             + S(1,5,r) * ct(j) * cp(i) ...
41                             - S(1,6,r) * ct(j) * sp(i) ) * cx ...
42             + ( ( - S(1,1,r) * ( 1 + c2p(i) ) / 2 ...
43                 - S(1,4,r) * s2p(i) ...
44                 - S(1,2,r) * ( 1 - c2p(i) ) / 2 ...
45                 + S(1,3,r) ) * s2t(j) / 2 ...
46                 +( S(1,6,r) * cp(i) + S(1,5,r) * sp(i) ) ...
47                 * c2t(j) ) * sx;
48         end
49     dyn(i,j,:) = ( max(a_chi,[],2)' - min(a_chi,[],2)' ) / 2;
50 end
51 end
52
53 % Evaluation on hydrostatic component of stress
54 hydro          = ( S(1,1,:) + S(1,2,:) + S(1,3,:) ) / 3;
55 max_hy         = max( hydro );
56
57 % Numerical inegration validated with Mescoscopic approach
58 K_RSS = ( pi ^ 3 / ( (m-1) * (l-1)/2 * (o-1)/2 ) ) / 8 / pi / pi;
59 RSS    = ( ( sum( ...
60             sum( ...
61             sum( dyn.^2 .* ( ones(l,m,o) .* sin(theta) ) ) ...
62                                     ) ...
63                                     ) ...
64             * K_RSS ) ^ 0.5 ) * sqrt(5);
65
66 % Mesoscopic equivalent stress
67 MESOSCOPIC = RSS + alfa * max_hy;
68
69 end
```

---

## Appendix C

# Maximum principal stress solver

In Appendix C the MATLAB<sup>®</sup> function I implemented during my thesis project at the department *Rotors D&D* of Leonardo S.P.A. division Helicopters is reported. The function reported is the latest version generated and the one currently implemented in the actual GUI. Previous versions are described in Cap. 3.

---

```
1 function [ EIG1, EIG11, EIG111, EIG2, EIG22, EIG222 ] ...
2     = LHD_GUI_Tesi( S, n, Fy)
3 % ===== %
4 %             PRINCIPAL STRESS SOLVER %
5 %             Andrea Casagrande %
6 %             Dynamic Systems – Rotors D&D – Leonardo S.P.A. %
7 %             Academic Year 2019/2020 %
8 % ===== %
9 %             The code takes advantage of eigenvalues evaluation
10 %             in order to find principal stresses acting on nodes.
11
12 %% NAMES
13 % 1 symbolize static tensor rotation
14 % 2 symbolize dynamic tensor rotation
15
16 % SINGLE DIGIT : dynamic stress
17 % DOUBLE DIGIT : static stress
18 % TRIPLE DIGIT : Goodman correction
19
20 % create the tensor from the row
21 S_mat = zeros(3,3,n);
22 S_mat(1,1,:) = S(1,1,:);
23 S_mat(1,2,:) = S(1,4,:);
24 S_mat(1,3,:) = S(1,6,:);
25 S_mat(2,1,:) = S(1,4,:);
26 S_mat(2,2,:) = S(1,2,:);
27 S_mat(2,3,:) = S(1,5,:);
28 S_mat(3,1,:) = S(1,6,:);
29 S_mat(3,2,:) = S(1,5,:);
30 S_mat(3,3,:) = S(1,3,:);
```

```
31
32 %% FIRST METHOD (METHOD 1)
33 % Tensor rotation referred to its own principal directions
34
35 a1      = zeros (n,n,2);
36 GM1     = zeros (n,n);
37 for q = 1 : n
38     for r = 1 : n
39         v1      = abs ( eig ( (S_mat(:, :, q) - S_mat(:, :, r) )/2 ) );
40         s1      = eig ( ( S_mat(:, :, q) + S_mat(:, :, r) )/2 );
41         a1(q,r,:) = [max(v1) max(s1)];
42         GM1(q,r)  = Fy * a1(q,r,1) / ( Fy - a1(q,r,2) );
43     end
44 end
45
46 [GM1max_riga, IT1_riga] = max(GM1,[],1);
47
48 % j1 specify the column which identify the max value of GM
49 [~, j1] = max(GM1max_riga);
50
51 % i1 identify the row with the max value of GM
52 i1 = IT1_riga(j1);
53
54 EIG1     = a1(i1,j1,1);
55 EIG11    = a1(i1,j1,2);
56 EIG111   = GM1(i1,j1);
57
58 %% FIRST METHOD (METHOD 1)
59 % Tensor rotation referred to dynamic tensor principal directions
60
61 a2      = zeros (n,n,2);
62 GM2     = zeros (n,n);
63 for q = 1 : n
64     for r = 1 : n
65
66         [d2,v2] = eig ( ( S_mat(:, :, q) - S_mat(:, :, r) ) / 2 );
67         v2 = abs( diag( v2 ) );
68
69         for p = 1 : 3
70             if max(v2) == abs( v2( p ) )
71                 a2(q,r,1) = max( v2 );
72                 vec2 = d2' * ( (S_mat(:, :, q) + S_mat(:, :, r) )/2 ) * d2;
73                 a2(q,r,2) = vec2(p,p);
74             end
75         end
76         GM2(q,r) = Fy * a2(q,r,1) / ( Fy - a2(q,r,2) );
77     end
end
```



---

```
78 end
79
80 [GM2max_riga, IT2_riga] = max(GM2,[],1);
81
82 % j2 specify the column which identify the max value of GM
83 [~, j2] = max(GM2max_riga);
84
85 % i identify the row with the max value of GM
86 i2 = IT2_riga(j2);
87
88 EIG2      = a2(i2,j2,1);
89 EIG22     = a2(i2,j2,2);
90 EIG222    = GM2(i2,j2);
91 end
```

---



# Bibliography

- [1] LeonardoSPA, *AW 169: Next Generation Versatility*. <https://www.leonardocompany.com/it/products/aw-169?f=/air/helicopters/commercial>, 2020.
- [2] D. S. R. D&D, *Main rotor rendering*. Property of Leonardo S.P.A.©.
- [3] R. Stephens, A. Fatemi, R. Stephens, and H. Fuchs, *Metal Fatigue in Engineering*. Wiley-Interscience.
- [4] Y. Lee, M. E. Barkey, and H. Kang, *Metal fatigue analysis handbook*. Butterworth-Heinemann.
- [5] I. Papadopoulos, *A high-cycle fatigue criterion applied in biaxial and triaxial out-of-phase stress conditions*.
- [6] I. Papadopoulos, *Long life fatigue under multiaxial loading*.
- [7] I. Papadopoulos, P. Davoli, C. Gorla, M. Filippini, and A. Bernasconi, *A comparative study of multiaxial high-cycle fatigue criteria for metals*.
- [8] B. Li, L. Reis, and M. de Freitas, *Comparative study of multiaxial fatigue damage models for ductile structural steels and brittle materials*.
- [9] A. Strozzi, *Costruzione di macchine*. Pitagora Editrice.
- [10] T. Mataka, *An explanation on Fatigue Limit under combined stress*.
- [11] K. D. Van and I. Papadopoulos, *High-cycle metal fatigue - from theory to applications*. Springer-Verlag.
- [12] D. Cuffiani, *Multiaxial high-cycle fatigue criteria for metallic materials*.
- [13] D. Broek, *Elementary engineering fracture mechanics*. Martnus Nijhoff Publisher.
- [14] V. Anes, L. Reis, B. Li, M. Freitas, and C. Sonsino, *Minimum Circumscribed Ellipse (MCE) and Stress Scale Factor (SSF) criteria for multiaxial fatigue life assessment*.