

POLITECNICO DI TORINO

MASTER OF SCIENCE IN MECHANICAL ENGINEERING



Master thesis

Fatigue life prediction models for components subjected to manufacturing induced residual stresses

Thesis advisors

Eng. Paolo Baldissera

Prof. Cristiana Delprete

Candidate

William Mosca

A.A. 2017/18

Index

1)	Literature state of the art review	13
1.1	Mechanical fatigue literature review	13
1.1.1	Stress-based fatigue approach	14
1.1.2	Strain-based fatigue approach	20
1.1.3	Fracture mechanics	24
1.1.3.1	Linear elastic fracture mechanics	24
1.1.3.2	Nonlinear fracture mechanics	28
	Crack tip opening displacement	28
	J-Integral	30
	HRR crack tip stresses and strains	32
	Fatigue crack growth (FCG) models	33
	Fatigue crack growth related phenomena	38
1.1.4	Thermomechanical fatigue	42
1.1.4.1	Taira model	42
1.1.4.2	Neu-Sehitoglu model	43
1.1.4.3	Chaboche model	44
1.2	Residual stresses	45
1.2.1	Overview	45
1.2.1.1	Welding residual stress	46
1.2.1.2	Shot-peening residual stresses	47

1.2.1.3	Quenching residual stresses	48
1.2.2	Residual stresses measurement methods	49
1.2.2.1	Non-destructive methods	49
	Ultrasounds velocity method	49
	Photo-elastic method	50
	Magnetic methods	50
	X-ray diffraction	50
1.2.2.2	Destructive methods	52
	Stäblein method	52
	Sachs method	52
1.2.2.3	Semi-destructive methods	53
	Hole drilling method	53
1.3	Fatigue life prediction models including residual stresses	54
1.3.1	The approaches in literature	54
1.3.1.1	Geometry model realization	54
1.3.1.2	Application of the residual stresses on the unnotched model	55
1.3.1.3	Definition of the parameters and the method used to account for the residual stresses in fatigue analysis	59
1.3.1.4	Fatigue crack growth simulation	66
2)	A python program for the optimized residual stress importing into the FEA model	80
2.1	introduction	80
2.1.1	Applications	82
2.1.2	The program	84
2.2	The main file logic	87
2.2.1	The main file	89
2.2.2	The pop-up window for the data insertion	91
2.2.3	extract1	92
2.2.4	run_job	94

2.2.5	extract2	95
2.2.6	statistics	96
2.2.7	Pi_adjustment	97
2.2.8	stepx	98
2.2.9	reload_data	98
2.2.10	report	99
2.2.11	The calculations of "statistics.py" and "report.py"	101
2.3	The program testing	103
2.3.1	First example: the CTS specimen	104
2.3.2	Second example: bent 3D beam	112
2.4	Further enhancements	116
2.4.1	Input by text file version	116
2.4.2	Input on sets version	118
2.4.3	Lean version	123

Symbols

N	Number of cycles
S	Engineering stress
σ	True stress
R	Stress ratio
A	Alternate stress
ϵ_p	Plastic strain
ϵ_a	Alternate strain
E	Elastic modulus
ν	Poisson's ratio
σ'_f	Fatigue strength coefficient
ϵ'_f	Fatigue ductility coefficient
σ_m	Mean stress
Γ	Crack separation energy
γ	Crack separation energy per unit area
a	Crack length
s	Crack thickness
U_0	Plate uncracked potential energy
U	Plate cracked potential energy
K_c	Material fracture toughness
Y	Shape factor
r	Crack tip radius
θ	Crack propagation angle
δ	Crack opening displacement

δ_t	Crack tip opening displacement
a_{eff}	Effective crack length
J	Rice's integral
ΔK	Stress intensity factor (SIF) range
K	SIF
K_{max}	Maximum SIF
K_{min}	Minimum SIF
K_{res}	Residual stress field SIF
K_{crit}	Critical SIF
K_c	SIF at failure
ΔK_{th}	SIF range threshold
K_{op}	Crack opening SIF
ΔK_{eff}	Effective SIF
da/dN	Crack advancement
D	Material damage
N_f	Cycles to failure
b	Weldment length
σ_i	Internal stress
v	velocity
M	Magnetization
H	Magnetic field
μ	Permeability
d	Interplanar distance
β	Integral factor
G	Strain energy release rate
$F_{y,y}$	Nodal reaction force

Tables and figures

Figure	Pag.
1.1. <i>Alternating and mean stress concept [2]</i>	15
1.2.a <i>Experimental S-N curve determination [1]</i>	16
1.2.b <i>Stress-life curves [1]</i>	16
1.3. <i>Wöhler curve of a steel specimen [1]</i>	17
1.4. <i>Goodman curve, mean stress effect depicted on the Wöhler curve [1]</i>	18
1.5. <i>Qualitative Gerber, Goodman and Soderberg curves in comparison [1]</i>	19
1.6. <i>Hysteresis cycle ascribed to the Bauschinger effect during a test [3]</i>	20
1.7. <i>Isotropic hardening [1]</i>	22
1.8. <i>Kinematic hardening [1]</i>	22
1.9. <i>Griffith case study, a cracked plate subjected to tensile stress [4]</i>	25
1.10. <i>Fracture modes relative to KIc, KIIc, KIIIc respectively [1]</i>	26
1.11. <i>Critical SIF versus plate thickness [1]</i>	27
1.12. <i>Crack tip geometry description [1]</i>	28
1.13. <i>Irwin correction [5]</i>	29
1.14. <i>Fatigue crack growth curve [1]</i>	35
1.15. <i>Overload and its effect on the FCG [7]</i>	38
1.16. <i>Change in stiffness due to the crack opening/closing [13]</i>	40

1.17. <i>The crack opening stress intensity factor [13]</i>	41
1.18. <i>Weldments RS [11]</i>	46
1.19. <i>Shot peening treatment [11]</i>	47
1.20. <i>Cylinder specimen quench [11]</i>	48
1.21. <i>X-ray diffractometer [11]</i>	51
1.22. <i>Hole drilling method [12]</i>	54
1.23. <i>Computation scheme [13]</i>	68
1.24. <i>Computation scheme [36]</i>	69
1.25. <i>Computational procedure [30]</i>	73
1.26. <i>Computation scheme [14]</i>	75
1.27. <i>Computation procedure [16]</i>	78
2.1. <i>Comprehensive assembly of the engine 2.2 JTD</i>	83
2.2. <i>Engine 2.2 JTD, basis, pistons and head</i>	84
2.3. <i>Program components</i>	88
2.4. <i>Pop-up window for the inputs</i>	92
2.5. <i>CTS specimen sketch</i>	105
2.6. <i>Boundary conditions and loads</i>	106
2.7. <i>Residual stress after releasing the load (the data are in Pa)</i>	107
2.8. <i>Filling in the input window</i>	108
2.9. <i>Results comparison</i>	111
2.10. <i>Cantilever beam sketch</i>	112
2.11. <i>Boundary conditions</i>	112
2.12. <i>Deformed shape cantilever beam</i>	115
2.13. <i>Comparison between the Von Mises stress given as initial condition and obtained</i>	115

Table	Pag.
2.1. <i>Plastic behaviour of the selected steel</i>	105

Abstract

The residual stresses were proven to play a relevant role in the fatigue life expectations of components and specimens. It is well established that almost every manufacturing process to some extent leads to the residual or internal stress fields formation. The challenge is to properly account for these internal stress fields in the fatigue life prediction.

In this essay a comprehensive study of the mechanical fatigue subject is presented, with particular focus on the residual stresses issue, as their impact on the fatigue life expectations, the manufacturing processes behind their formation, their measurement methods and the most performing way to account them in the fatigue life estimates.

The most challenging issues in terms of the residual stress simulation in fatigue are presented through the previous works found in literature and finally in this thesis my personal contribution was given in the development of a Python program for the optimized insertion of a residual stress field on a finite element model, which will be potentially employed at FCA Automobiles for research purpose.

Introduction

This master thesis focus on the subject of the fatigue life prediction in its classical approaches, known also as stress-life and strain-life based approaches and on the fracture mechanics approach in its variants.

The target of this research was to comprehensively determine, through the works found in literature, the most performing strategies to account for the residual or internal stress fields, inherited by the component after its manufacturing process, in their fatigue life estimates.

In order to gain a deep understanding of this subject a complete study on the residual stress issue was performed, starting from their classification, the most common manufacturing processes behind their formation and the techniques employed in the industrial practise to esteem their field and magnitude, with particular attention to the limits and difficulties commonly encountered.

A relevant point of this work was a collaboration with the engineers of FCA Automobiles, through my academic advisor Eng. Paolo Baldissera, which were conducting a research indeed on the residual stress issue in FEM simulation on a product.

The essay is structured in two parts, in the first chapter a literature state of the art review was made, whereas the second chapter focuses on the Python program developed.

The literature review is subdivided into three sections, starting with a deep study on the mechanical fatigue branches, that are the strain-based, stress-based and fracture mechanics approaches, and some outlines to the thermomechanical fatigue.

A sound background on fracture mechanics was very important for the research purpose, since the bulk of the authors in literature have provided empirical models and approaches starting from the well-known Paris law, which have been adapted to account for the residual stress fields.

In addition, a brief subsection on the thermomechanical fatigue was added, since the product under investigation in FCA Automobiles is interested by both thermal and mechanical stresses, then it was sensible to gain some knowledge on the subject.

The second section of the review, deals totally with the residual stresses. As first some outlines are given on the manufacturing processes inducing these stress fields, that are broadly speaking welding, shot-peening or quenching processes. Subsequently a subsection describes the internal stress measurement methods used in the industrial practise, classified according to their degree of damage caused on the specimen.

In the third and last section of the chapter, all the works found in literature, dealing with the residual stress issue in the mechanical fatigue subject, have been gathered and analysed, in such a way to break down all the presented approaches into few common steps,

and underlining per each step the pros and cons of the different strategies adopted and the technical difficulties and limits.

In the second chapter and part of the proposed work a Python program was developed in order to solve a problem encountered in a research of FCA Automobiles, during the simulation and the importing of the residual stress fields.

The program is thought to allow the importing of the residual stress tensors per each model element, from a text file or a previous analysis, to Abaqus.

An iteration loop, allows the software to solve the technical problem of internal stress equilibration, which will be explained in deep afterwards, based on the first approach developed by O'Dowd et al. [27] in his studies.

The chapter starts with a general description of the program and its potential applications; then follows a complete description of each program subscript, and finally a couple of analysis examples are presented in order to show the potentials of the developed software.

The last subsection describes some further versions and enhancements which are applicable to the original program, since it was developed to have a robust basis and flexible for further potential improvements and add-ons.

Chapter 1

1) Literature state of the art review

In this chapter a literature review was made, comprehensive of all the theoretical background needed to carry out the Master thesis work. As first the main fatigue life prediction approaches found in literature are presented. Afterwards, a second section is dedicated to the residual stresses topic, namely the manufacturing processes to which they can be ascribed and the measurement techniques. Finally, a third section is dedicated to the state of the art of the models used to take into account the potential detrimental or beneficial residual stresses effects on the fatigue life prediction.

1.1 Mechanical fatigue literature review

The approaches known in literature can be subdivided into the classical approaches, which are the stress-based and the strain-based methodologies, and the fracture mechanics methodology. The choice of the method to be employed is related to the application, the stress magnitude, and the number of cycles to failure.

1.1.1 Stress-based fatigue approach

The so called stress-life approach for fatigue analysis in metals has its own origins in the work of Wöhler from about 1850. In this method the stress amplitude σ_a at a suitable location in the specimen is the key parameter which takes the leads in the number of cycles to failure N_f evaluation.

Wöhler called a "safe stress level" the one below which failure does not occur: above this safe stress level, failure will occur within a certain life, measured as number of cycles.

Crack growth is not explicitly accounted for in the stress-life method. Because of this, stress-life methods are often considered crack initiation (or incubation) life estimates [1].

This method involves the experimental determination of $S-N$ or $\sigma-N$ curves (the latter referred to the so called "true stresses"), which are characterized by the following equations:

$$S_a = \frac{\Delta S}{2} = \frac{S_{max} - S_{min}}{2}$$

$$S_m = \frac{S_{max} + S_{min}}{2}$$

$$S_{max} = S_m + S_a$$

$$S_{min} = S_m - S_a$$

In figure 1.1 the concept of alternating and mean stress is furtherly clarified:

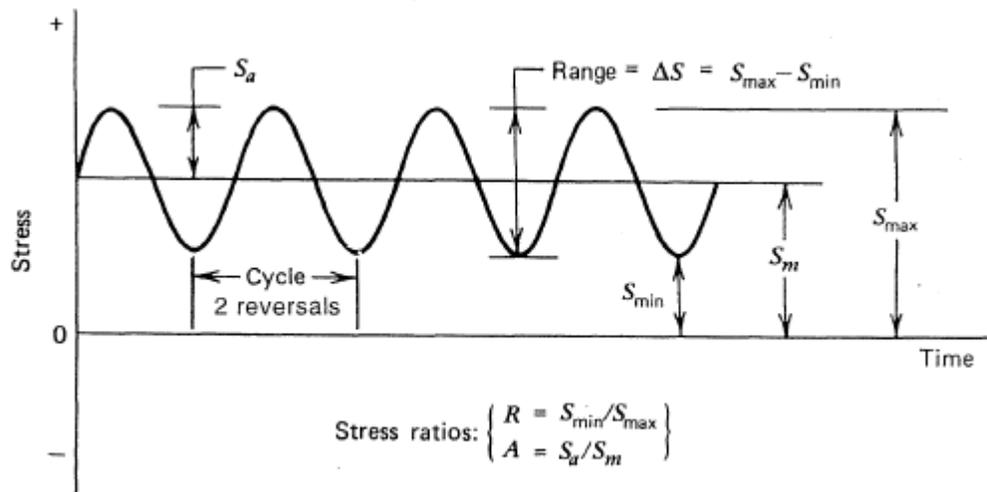


Figure 1.1 "Alternating and mean stress concept [2]".

Where S_{max} and S_{min} are respectively the maximum and minimum stresses and ΔS is the stress range. Other largely employed parameters are the stress ratio R and the alternating stress ratio A , namely:

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

$$A = \frac{S_a}{S_m}$$

The stress ratio R is commonly employed in literature in its two common reference values, that are " $R=0$ ", implying the so called "pulsating tension", and " $R=1$ ", which involves the "fully reversed" condition.

As to the experimental determination of the Wöhler curve, a group of specimens is tested at least at three different stress levels; the data are then used to obtain the life probability curves at each of these stress levels [1].

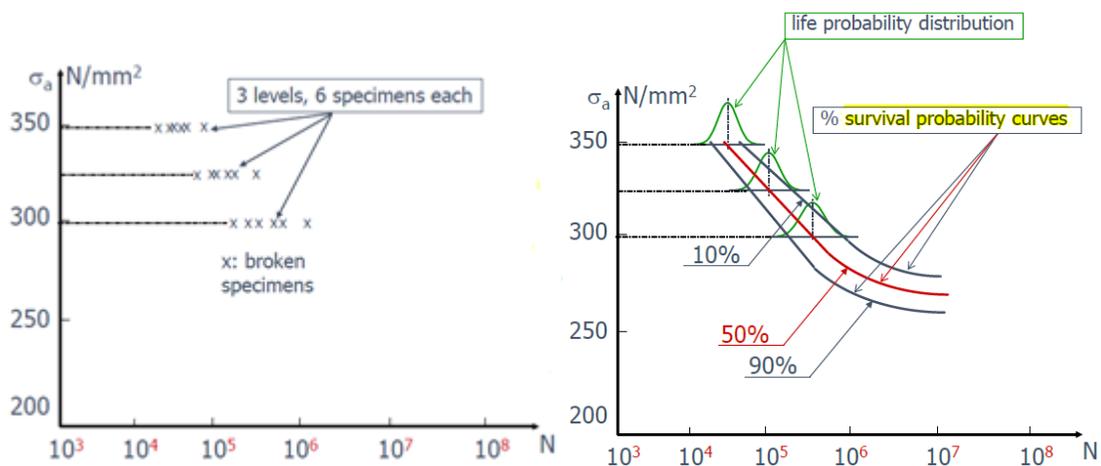


Figure 1.2a: "experimental S-N curve determination", 1.2b: "stress-life curves"[1].

The fatigue limit can be then determined with the "staircase" method, by which a small number of equally spaced stress levels is set around the expected fatigue limit: the first specimen is tested at the highest stress level and if failure occurs, a further test is carried out at a lower stress level.

If failure does not occur at a certain number of cycles, the test is stopped and a further specimen is tested at the upper stress level.

The obtained S-N curve can then be subdivided into three different ranges, according to the number of cycles to failure and the magnitude of the applied alternating stress, as in fig. 1.3:

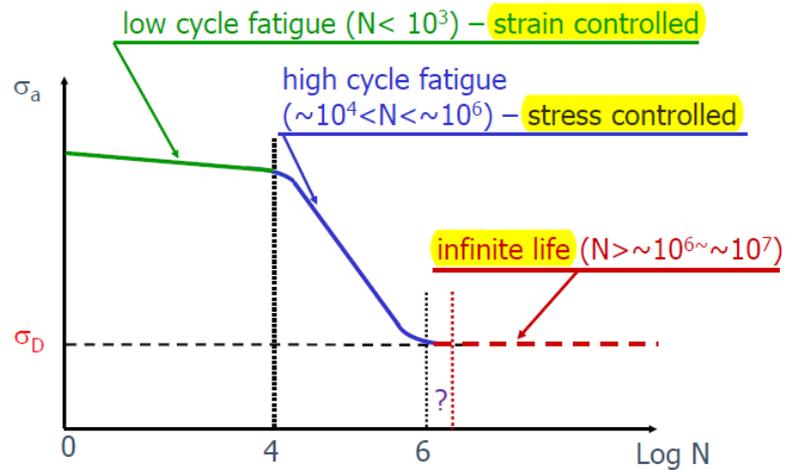


Figure 1.3: "Wöhler's curve of a steel specimen"[1].

Hence, the curve can be divided into a low cycle fatigue range in which the phenomena is said to be strain controlled, and an intermediate range called high cycle fatigue and stress controlled and finally an infinite life range characterized by a stress threshold or fatigue limit σ_D , below which theoretically no failure occurs independently on the number of cycles undergone by the specimen. It is important to specify that many non-ferrous metals and alloys, such as aluminium, magnesium, and copper alloys, do not exhibit well-defined endurance limits; these materials instead display a continuously decreasing S - N response [1]. Hence, in this instance a fatigue limit σ_D must be specified case by case.

In addition, according to the employed guidelines a certain number of corrections to the parameters of these curves, such as the fatigue limit, must be employed. Generally speaking, the aim is to transfer the statistical data obtained when dealing with the standardized test specimens, to the real components, therefore accounting also for the scale

effects, the process influence, the temperature, the surface finishing, and the kind of load applied.

An important factor which must be taken into account is the mean stress influence summed to the alternating stress. A popular tool available is the so called Goodman's curve, which relates the mean stress σ_m to the fatigue limit of the Wöhler's curve; it was proven that broadly speaking a compressive (negative) mean stress distribution over the specimen implies a beneficial effect on the fatigue life, whereas by contrast a tensile (positive) mean stress may compromise the life expectancy of the component. In fig.1.4 this concept is furtherly clarified and Goodman's curve for a steel is reported:

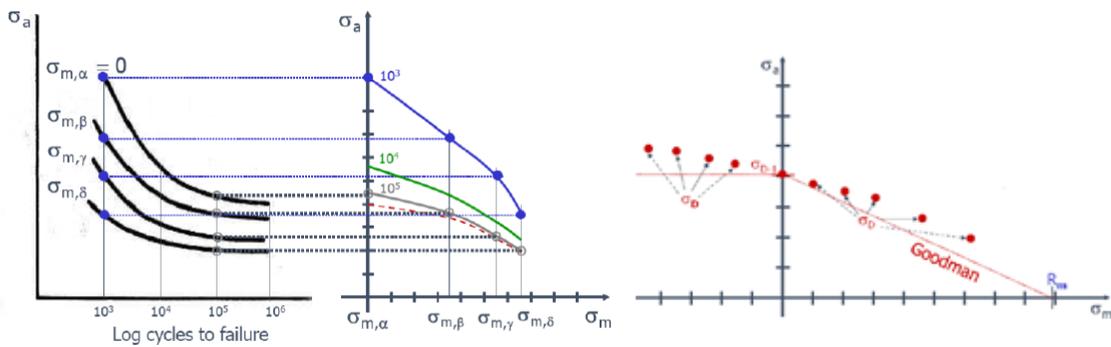


Figure 1.4: "Goodman's curve, mean stress effect depicted on the Wöhler's curve" [1].

However, the Goodman's model is not the only one present in literature, others curves are indeed the Gerber and Soderberg which are compared in figure 1.5:

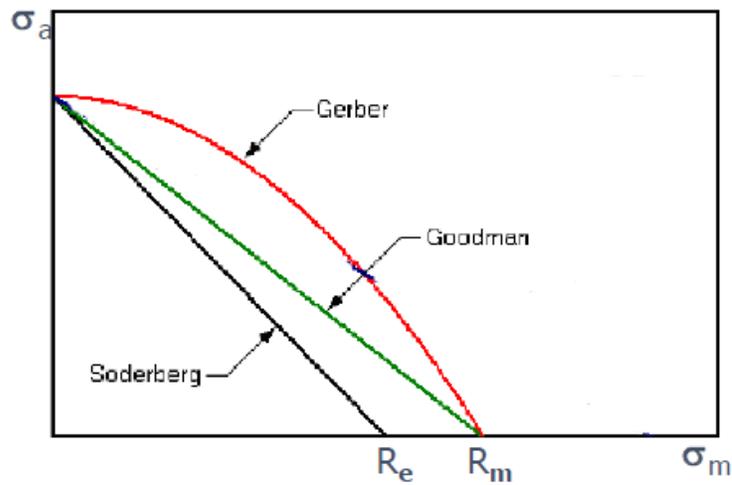


Figure 1.5: "Qualitative Gerber, Goodman and Soderberg's curves in comparison" [1].

These Goodman, Gerber and Soderberg's models are represented by the following equations respectively:

$$\frac{\sigma_D}{\sigma_{D-1}} + \frac{\sigma_m}{R_m} = 1$$

$$\frac{\sigma_D}{\sigma_{D-1}} + \left(\frac{\sigma_m}{R_m}\right)^2 = 1$$

$$\frac{\sigma_D}{\sigma_{D-1}} + \frac{\sigma_m}{R_e} = 1$$

Where σ_{D-1} is the fatigue limit in absence of mean stress, R_m is the material strength and R_e is the yielding strength of the material.

1.1.2 Strain-based fatigue approach

The so called strain-life method concerns the low cycle fatigue range, characterized by higher stresses and lower excitation frequencies. This means that elastic and plastic strains might occur together. Hence, under these conditions a characterization of the material through the applied strain is generally better than a stress-based characterization.

An interesting fact is that the material, during its life-cycle is likely to reach its yielding point, which implies the arising of a hysteretic phenomenon, described by the so called Bauschinger effect [2].

The Bauschinger effect refers to a property of materials where the material's stress/strain characteristics change as a result of the microscopic stress distribution of the material. For example, an increase in tensile yield strength occurs at the expense of compressive yield strength [3].

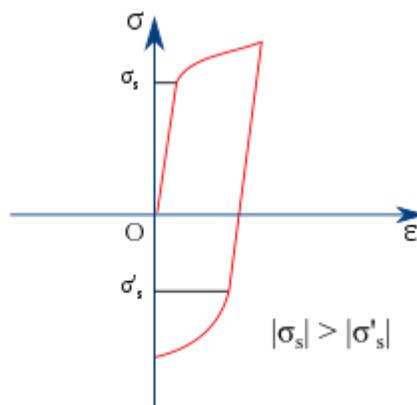


Figure 1.6: "Hysteresis cycle ascribed to the Bauschinger effect during a test [3]"

The Bauschinger effect is normally associated with conditions where the yield strength of a metal decreases when the direction of strain is changed. It is a general phenomenon found in most polycrystalline metals. The basic mechanism for the Bauschinger effect is related to the dislocation structure in the cold worked metal. As deformation occurs, the dislocations will accumulate at barriers and produce dislocation pile-ups and tangles [3].

According to the loading cycle applied to the specimen and the material properties, different hysteresis loop shapes and material responses can be obtained, namely both hardening and softening responses can be observed.

An important issue in the attempt to describe the material behaviour in low cycle fatigue is the estimation of a suitable stress-strain hardening curve, able to describe the relationship amid the cyclic plastic deformation and the true or engineering stress, namely:

$$\sigma = g^{-1}(\epsilon_p)$$

For instance, the descriptions of Ramberg and Osgood, Hollomon and Ludwick, respectively reported:

$$\sigma = \sigma_y + K_y \epsilon_p^{\frac{1}{M_y}}$$

$$\sigma = K \epsilon_p^n$$

$$\sigma = \sigma_y + K \epsilon_p^n$$

Where K_y , M_y , K and n , are material constants. The most common hardening models named in literature are essentially the isotropic and the kinematic hardening: in the first

case it is assumed that the material works in the plastic field both in tension and in compression following a uniform expansion of the flow surface, meanwhile at each cycle the elastic limit increases and the area of the hysteresis loop decreases.

As to the kinematic hardening model, it is instead assumed a translation of the elastic domain and a constant hysteresis loop area at each loading cycle.

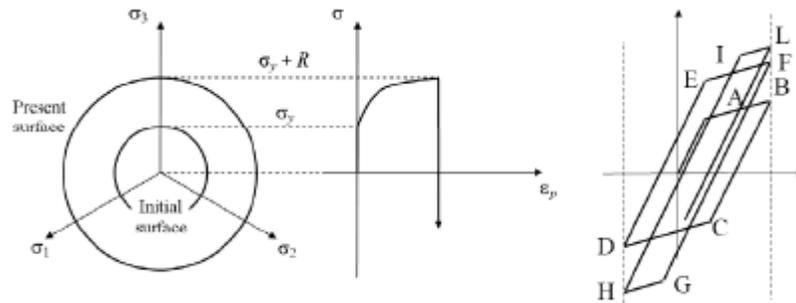


Figure 1.7: "isotropic hardening" [1].

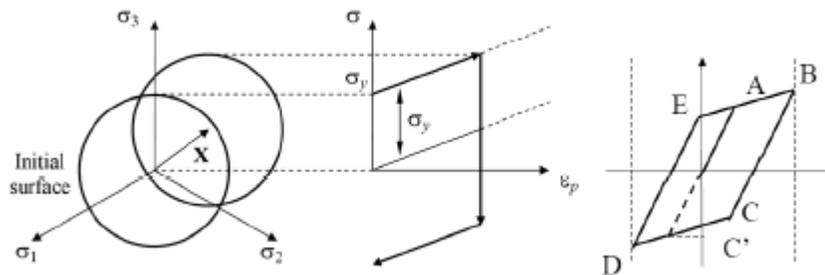


Figure 1.8: "kinematic hardening" [1].

However, it is important to remark that the hardening phenomenon in real materials cannot in general be described by the isotropic or kinematic models. Indeed, it is generally assumed a combined model, closer to the real phenomenon.

Along with the formulation of an assumption to describe the plastic behaviour of the material, it is then necessary to consider a suitable model for the fatigue life estimate.

Several models have been developed in literature, suitable for both pure phenomenological and thermo-mechanical fatigue conditions, the most relevant are herein reported:

$$\text{Basquin-Manson-Coffin} \quad \epsilon_a = \frac{\sigma_f'}{E} (2N_f)^b + \epsilon_f' (2N_f)^c$$

$$\text{Morrow} \quad \epsilon_a = \frac{\sigma_f'}{E} \left(1 - \frac{\sigma_m}{\sigma_f'}\right) (2N_f)^b + \epsilon_f' \left(1 - \frac{\sigma_m}{\sigma_f'}\right)^{\frac{c}{b}} (2N_f)^c$$

$$\text{Walker} \quad \epsilon_a = \frac{\sigma_f'}{E} \left(\frac{1-R}{2}\right)^{1-\gamma} (2N_f)^b + \epsilon_f' \left(\frac{1-R}{2}\right)^{c\frac{1-\gamma}{b}} (2N_f)^c$$

$$\text{Smith-Watson-Topper} \quad \sigma_{max} \epsilon_a = \frac{(\sigma_f')^2}{E} (2N_f)^{2b} + \sigma_f' \epsilon_f' (2N_f)^{b+c}$$

Where b and c are fitting constants to be determined experimentally, σ_f' is the fatigue strength coefficient and ϵ_f' is the fatigue ductility coefficient. The mathematical models of Morrow, Walker and SWT are suitable to take into account also the mean stress effect on the fatigue life expectancy, the first two by incorporating the mean stress σ_m , or the stress ratio R into the equation. The SWT equation is based on strain-life data obtained by various mean stress values; this equation is based on the assumption that for different combinations of strain amplitude, and mean stress, the product $\sigma_{max} \epsilon_a$ remains constant for a given life [2].

1.1.3 Fracture mechanics

In contrast to the classical approaches (stress-life or strain-life), in the fracture mechanics methodology the presence of flaws or defects is taken for granted. Indeed, fracture mechanics is strictly related to the damage tolerant design philosophy, which assumes as well the potential presence of a certain damage or defect in each part of a system, which not necessarily will lead to the component or structure total impairment. The aim is indeed to foresee the potential damage evolution during the component service life, and so to prevent a possible catastrophic failure.

Several theories were developed in literature, which may be more or less suitable for describing the actual phenomenon according to the material physical properties and the loading conditions.

1.1.3.1 *Linear elastic fracture mechanics*

The so called LEFM is particularly suitable for brittle materials characterized by a small plasticization area at crack tip.

In its studies Griffith developed a theory based on an energy approach; the fundamental study case was a plate of thickness s , subjected to a tensile stress σ , in which a crack of length $2a$ is present at the plate centre. It is assumed that a certain amount of energy Γ is required to form the crack and so to divide two equal rectangular surfaces of area $2as$, where the separation energy can be regarded as:

$$\Gamma = 4asy$$

Where γ represents the separation energy per unit area.

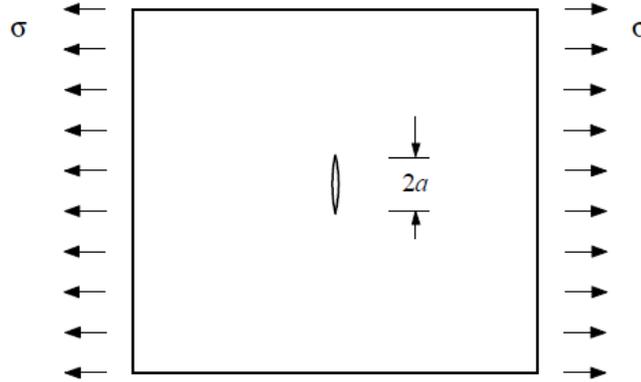


Figure 1.9: "Griffith case study, a cracked plate subjected to tensile stress" [4].

It is then possible to demonstrate that the difference in elastic energy of the cracked and uncracked version of the plate can be computed as:

$$U_0 - U = \frac{\pi\sigma^2 a^2 s}{E}$$

According to the theory, when the crack extends of a length of $2a$ some of its energy is spent for the separation but in turn a certain amount of elastic energy is generated. If the generated elastic energy is of a greater amount respect to the separation energy, then the crack is bound to propagate in an instable manner. The propagation condition can be described by the next equations:

$$\frac{d(U_0 - U)}{da} \geq \frac{d\Gamma}{da}$$

$$\sigma\sqrt{\pi a} \geq \sqrt{2E\gamma}$$

Where the term on the right represents a measure of the crack propagation driving force, which is not only dependent on the applied stress, but the crack length a plays an

important role as well; the term on the left is a sort of measure of the material toughness, also K_c : the higher the toughness, the higher the material resistance to the crack propagation.

The fracture toughness can be detected through a Charpy test, but depends strictly on the fracture mode:

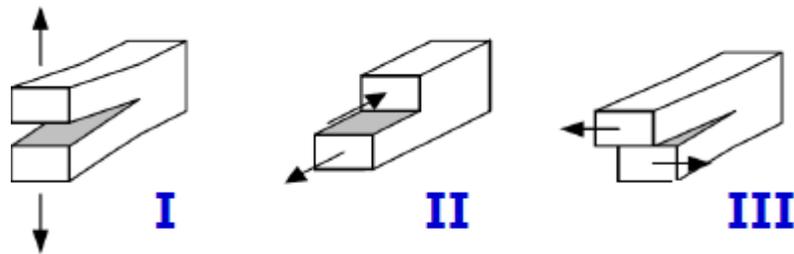


Figure 1.10: "Fracture modes relative to K_{Ic} , K_{IIc} , K_{IIIc} respectively [1]."

The stress field at the crack tip was firstly analytically computed by Westergaard as follows:

$$\sigma_{ij}(r, \theta) = \frac{K_{I,II,III}}{\sqrt{2\pi r}} f_{ij}(\theta)$$

It is interesting the fact that in these solutions the stress field is strictly related to the stress intensity factor K or SIF, which has the dimension of a stress multiplied for the square root of a length, namely:

$$K = Y\sigma\sqrt{a}$$

Where Y is the shape factor and depends on the crack type/shape. In case of plane stress conditions, the equivalent stress can be computed as follows:

$$\sigma_{id} = \frac{K}{\sqrt{2\pi r}}$$

Instead in case of plane strain condition it follows:

$$\sigma_{id} = \frac{K}{\sqrt{2\pi r}}(1 - 2\nu)$$

Therefore, in the latter case the crack tip normal stress is higher and the plasticized area is smaller (more critical situation).

It was then proven that the critical value of the SIF which leads to the instable crack propagation, is inversely proportional to the specimen thickness, and stabilizes at a minimum value regarded as fracture toughness K_{Ic} .

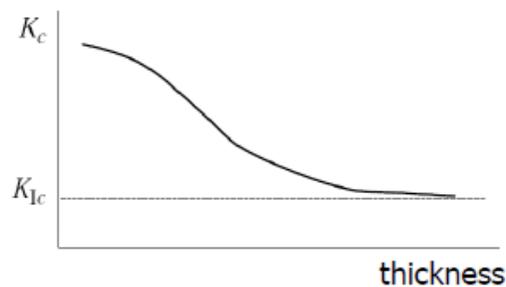


Figure 1.11: "Critical SIF versus plate thickness" [1].

In addition, generally speaking the fracture toughness is inversely proportional to the material strength $R_{p0.2}$ [4].

1.1.3.2 Nonlinear fracture mechanics

It is also called elastoplastic fracture mechanics (EPFM), and it is more suitable when the small scale yielding assumption of LEFM is no longer valid.

Crack tip opening displacement

As described in [5], let consider an xy -coordinate system placed at the centre of the crack, with the crack aligned along x , the crack displacement along the opposite direction u_y can be calculated as follows:

$$u_y = \frac{\sigma\sqrt{\pi a}}{2\mu} \sqrt{\frac{r}{2\pi}} \left[\sin\left(\frac{1}{2}\theta\right) \left\{ k + 1 - 2 \cos^2\left(\frac{1}{2}\theta\right) \right\} \right]$$

This displacement is function of the angle θ and the radius of the plasticized zone r :

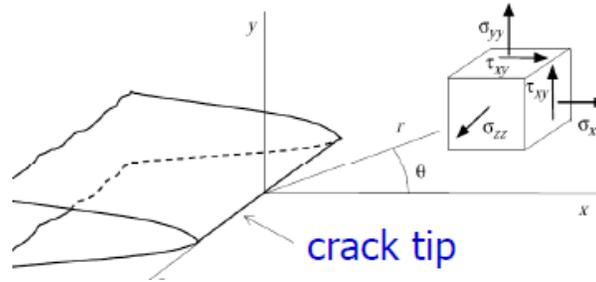


Figure 1.12: "Crack tip geometry description" [1].

The material points displacement at crack tip results for " $\theta=\pi$ " and by taking " $r=a-x$ " it follows:

$$u_y = \frac{(1 + \nu)(k + 1)}{E} \left(\frac{\sigma}{2}\right) \sqrt{2a(a - x)}$$

The crack opening displacement COD or δ is two times this displacements, whereas trivially the crack tip opening displacement CTOD or δ_t is nil.

$$\delta(x) = 2u_y(x) = \frac{(1 + \nu)(k + 1)}{E} \sigma \sqrt{2a(a - x)}$$

$$\delta_t = \delta(x = a) = 0$$

The CTOD can be used in a crack growth criterion, when plasticity at the crack tip is taken into account and the actual crack length is replaced by the effective crack length [5].

According to Irwin the influence of the crack tip plastic zone can be taken into account by using an effective crack length a_{eff} , which is the actual crack length plus the length of the plastic zone in front of the crack tip [5].

$$a_{eff} = a + r_y = a + \frac{1}{2\pi} \left(\frac{K_I}{\sigma_y} \right)^2$$

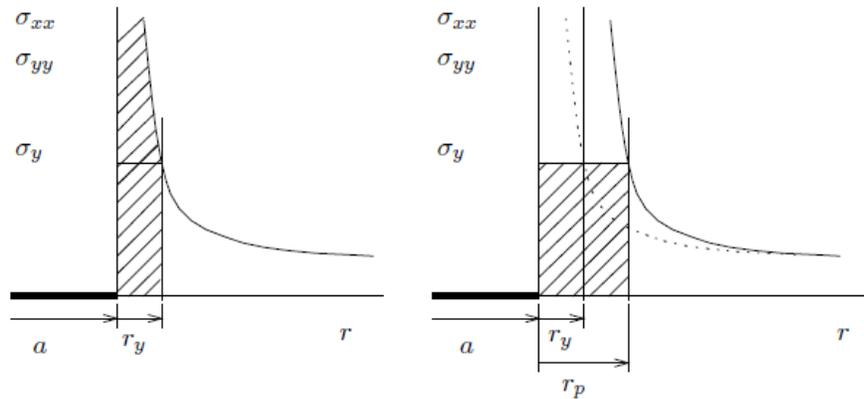


Figure 1.13: "Irwin correction" [5].

According to this correction the CTOD can be calculated as follows for plane stress and plane strain respectively:

$$\delta_t = \frac{4K_I^2}{\pi E \sigma_y} = \frac{4G}{\pi \sigma_y}$$

$$\delta_t = \frac{4(1 - \nu^2)K_I^2}{\sqrt{3}\pi E \sigma_y}$$

Differently from the LEFM theory by which the CTOD can be related to the energy release rate by means of the SIF in EPFM or NLFM the CTOD is a measure for the deformation at the crack tip, which can then be compared to a critical value in a crack growth criterion.

J-Integral

The J integral was firstly introduced by Rice, and basically it is a vector made up of three components in the Cartesian coordinate system. The integration is performed along a trajectory Γ , and for each interested point the specific elastic energy must be calculated from the known stress and strains.

The J integral can be calculated as follows:

$$J_k = \int_{\Gamma} \left(W_{n_k} - \frac{t_i \partial u_i}{\partial x_k} \right) d\Gamma$$

$$W = \int_0^{\epsilon_{pq}} \sigma_{ij} d\epsilon_{ij}$$

When integration path is closed it can be proven that the resulting value is always zero, under the conditions that there are no singularities in the area within the closed path. In addition, if the behaviour is hyper-elastic and homogeneous, the material is not subjected to volume loads and the acceleration is nil [5], along with the assumption of linear strain-displacement, implying small rotations and deformations.

Under these assumptions it follows:

$$J_k = \int_{\Gamma} (W \delta_{jk} - \sigma_{ij} u_{i,k}) n_j d\Gamma$$

Thanks to the assumption of no singularities along the path, the Stokes theorem yields:

$$\int_{\Omega} \left(\left(\frac{dW}{d\epsilon_{mn}} \right) \left(\frac{\partial \epsilon_{mn}}{\partial x_j} \right) \delta_{jk} - \sigma_{ij} u_{i,k} - \sigma_{ij} u_{i,kj} \right) d\Omega$$

Homogeneous hyper-elastic

$$\sigma_{mn} = \frac{\partial W}{\partial \epsilon_{mn}}$$

Linear strain

$$\epsilon_{mn} = \frac{1}{2} (u_{m,n} + u_{n,m})$$

Equilibrium equations

$$\sigma_{ij,j} = 0$$

Finally,

$$\int_{\Omega} \left(\frac{1}{2} \sigma_{mn} (u_{m,nk} + u_{n,mk} - \sigma_{ij} u_{i,kj}) \right) d\Omega = \int_{\Omega} (\sigma_{mn} u_{m,nk} - \sigma_{ij} u_{i,kj}) d\Omega = 0$$

Another important feature of the J integral is the path independency; thanks to this the integration path can be chosen to be a circle around the crack tip centre. It is also possible to use the J integral in LEFM instead of K or G because strictly related, for mode I it follows for plane stress and plane strain respectively:

$$J = \frac{1}{E} K_I^2 \quad J = \frac{1 - \nu^2}{E} K_I^2$$

Moreover, it is related also to the CTOD as follows for plane stress and plane strain respectively:

$$J = \frac{\pi}{4} \sigma_y \delta_t \quad J = \frac{\pi}{4} \sigma_y \delta_t \sqrt{3}$$

In addition, the ASTM guidelines suggests a more precise empirical relationship which accounts also for the specimen size W :

$$J = m \sigma_y \delta_t$$

$$m = -0,111 + \frac{0,817a}{W} + \frac{1,36\sigma_u}{\sigma_y}$$

HRR crack tip stresses and strains

This solution for the crack tip stress and displacement was derived by the researchers Hutchinson, Rice and Rosengren. The Ramberg-Osgood material model was taken into account, namely:

$$\frac{\epsilon}{\epsilon_{y0}} = \frac{\sigma}{\sigma_{y0}} + \alpha \left(\frac{\sigma}{\sigma_{y0}} \right)^n$$

Where n is the hardening exponent, and the subscript 0 is related to the initial conditions.

The HRR solution for the stress and displacement at the crack tip is wholly determined by means of a special parameter β , which is strictly related to the J integral, the hardening exponent n , and a constant I_n which can be determined experimentally. It follows:

$$\sigma_{ij} = \sigma_{y0} \beta r^{-\frac{1}{1+n}} \tilde{\sigma}_{ij}(\theta)$$

$$u_i = \alpha \epsilon_{y0} \beta^n r^{\frac{1}{1+n}} \tilde{u}_i(\theta)$$

$$\beta = \left[\frac{J}{\alpha \sigma_{y0} \epsilon_{y0} l_n} \right]^{\frac{1}{n+1}}$$

Finally, analogously to the energy release rate or the SIF in LEFM, in NLFM the J integral wholly describes the stress and deformation state at crack tip, and must be compared to a critical value J_c which potentially lead to failure. That value can be experimentally derived.

Fatigue crack growth (FCG) models

In this section the most important FCG models found in literature are reported, starting with the well-known Paris law, which is at the basis of all the other developed models. Such models are in the form of power laws, and generally depend upon the SIF, the SIF range, experimentally determined fitting constants and in some cases other parameters related to the mean stress influence or other phenomena.

It is important to remark that the Paris law and its extended versions are valid for the high cycle fatigue regime, where stresses are so low that ΔK characterizes the stress amplitude. For low cycle fatigue this is not the case anymore. Crack growth laws for this high stress regime with large plastic crack tip regions, are still under development. For high values of crack tip stress and consequently a large plastic zone, the Paris law may be used with ΔJ instead of ΔK [5].

Paris law

According to Paris, if the plastic zone in proximity of the crack tip is sufficiently small, the fatigue crack growth can be evaluated essentially by means of the stress intensity factor, generally regarded as K ; the power law proposed by Paris and Erdogan was:

$$\frac{da}{dN} = C(\Delta K)^m$$

Where C and m are fitting constants related to the material properties; this relationship implies that the crack growth ratio da/dN is strictly related to the stress intensity factor range of the loading cycle, which can be regarded as "driving force". As we may notice this law can be regarded as a one parameter driving force law.

We recall that the stress intensity factor range is defined as:

$$\Delta K = K_{max} - K_{min}$$

Where K_{max} and K_{min} are respectively the maximum and the minimum stress intensity factors affecting the specimen during its fatigue cycle.

In fig.1.14 the FCG curve is depicted in its three characteristic regions:

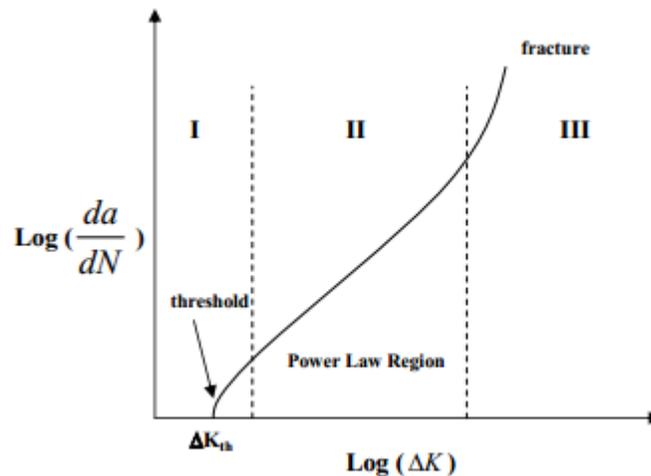


Figure 1.14: "Fatigue crack growth curve"[1].

The curve can be subdivided into three parts, that is a first region in which there is a slow crack growth, a second region described by the Paris, or a Paris-like power law, and a third region characterized by a fast crack growth which yields eventually to a fracture.

Another important parameter in the FCG study is the stress ratio R , defined as:

$$R = \frac{K_{min}}{K_{max}}$$

That is the ratio of the minimum versus the maximum SIF of the loading cycle. Generally speaking, as the stress ratio increases, the FCG ratio increases as well.

The Paris law is not able to take into account its effect, then other more complex versions have been developed, and have been largely used to deal with mean stress, or residual stress fields.

Hereafter some Paris law variants used in literature in this field are reported.

Walker's equation

$$\frac{da}{dN} = C \left[\frac{\Delta K}{(1-R)^{1-m}} \right]^n$$

The Walker's equation was essentially an enhancement of the Paris Equation, in order to account for the stress ratio R effects.

Forman's equation

$$\frac{da}{dN} = \frac{C \Delta K^m}{(1-R)(K_c - K_{max})}$$

The above relationship is suitable to depict the FCG both in regions 2 and 3; actually C and m are fitting constants, whereas K_c is the SIF at fracture and R the stress ratio.

This equation was indeed an improvement to the Walker's equation to account to the third region of FCG in which the data becomes asymptotic to the value of ΔK_c at fracture [89].

Klesnil and Lukas equation

$$\frac{da}{dN} = C(\Delta K^m - \Delta K_{th}^m)$$

In the above equation the importance of the SIF range threshold in the FCG driving force is underlined, that is, the SIF range must overcome a certain value to contribute to the further crack extension.

McEvelly's equation

$$\frac{da}{dN} = C(\Delta K - \Delta K_{th})^2 \left[1 + \frac{\Delta K}{K_{crit} - K_{max}} \right]$$

The McEveley's equation was an attempt to depict the FCG rate versus the SIF range curve throughout its domain, that is from region 1 to region 3. Indeed, generally speaking, these power laws focuses on the second region only.

Erdogan equation

$$\frac{da}{dN} = \frac{C(1 + \beta)^m(\Delta K - \Delta K_{th})^n}{K_{Ic} - (1 + \beta)\Delta K} \quad \text{with } \beta = \frac{K_{max} + K_{min}}{K_{max} - K_{min}}$$

Broek and Schijve

$$\frac{da}{dN} = CK_{max}^2\Delta K$$

Donahue

$$\frac{da}{dN} = C(\Delta K - \Delta K_{th})^m$$

Priddle

$$\frac{da}{dN} = \left(\frac{\Delta K - \Delta K_{th}}{K_{Ic} - K_{max}} \right)^m$$

NASGRO equation

The NASGRO equation was firstly developed by NASA's researchers to be implemented in their homonym crack growth prediction program NASGRO [89].

The equation is here presented:

$$\frac{da}{dN} = \frac{C \left[\left(\frac{1-f}{1-R} \right) \Delta K \right]^n \left(1 - \frac{\Delta K_{th}}{\Delta K} \right)^p}{\left(1 - \frac{K_{max}}{K_{crit}} \right)^q}$$

Where C , p and q are empirically derived. A more detailed explanation of this model can be found in literature [6].

Fatigue crack growth related phenomena

In this section some potentially relevant aspect of the fracture mechanism influencing the fatigue life of a structure are treated.

Overload effects

As stated in [106], "since fatigue-crack growth is driven predominantly by the crack-tip plasticity, and plastic strains are irreversible, changes in the load patterns invariably result in transient effects, which affect FCG rates and fatigue lives". Figure 1.15 better clarifies the overload phenomenon:

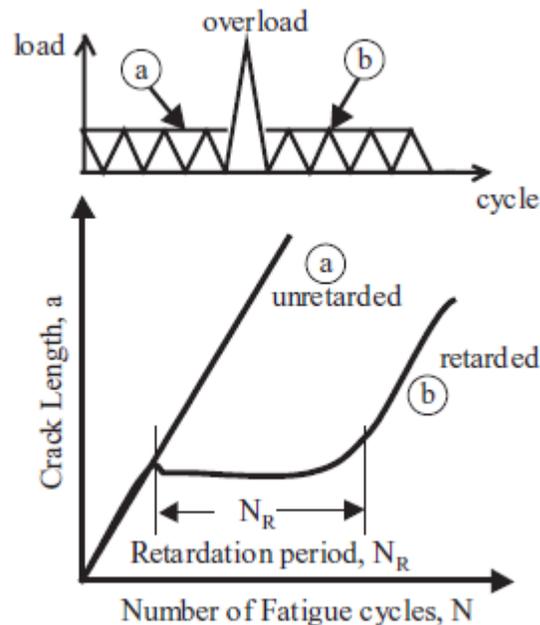


Figure 1.15: "Overload and its effect on the FCG" [7].

In fig.1.15 is depicted a regular, pulsating loading cycle interested by a sudden overload of relatively short duration; as it is observable from the curves below, the overload

causes a tangible retardation of duration N_R which dramatically changes the FCG trend. From the literature it is known that this retardation period increases with the magnitude and the number of overloads [7].

Other important observations made in [7], are that the retardation effect depends on the overload ratio, namely:

$$O.R. = \frac{K_{max}^{overload}}{K_{max}^{fatigue}}$$

Along with ΔK and the stress ratio R . In addition, it was found that the overload can produce a very short initial FCG rate acceleration followed by a much longer and more significant deceleration, like it is visible in the curves of figure 1.15.

It is additionally appreciable that the maximum deceleration of growth rate occurs at a short distance from the overload point; this phenomenon is called delayed retardation and depends upon the overload ration, R and ΔK .

The overload is supposed to eventuate into a larger plasticized area all over the crack tip, indeed according to the theory the retardation continues until the crack has propagated and moved out of such area.

Moreover, the retardation effect depends upon the specimen geometrical properties, such as its thickness, since the plastic zone size, under plane stress and plane strain conditions differ. The effects of the retardation phenomenon are generally larger under plane stress conditions.

Crack closure

The overload-induced retardation is strictly related to the crack closure phenomenon, which can arise due to many reasons, such as the crack tip blunting, crack deflection, branching, plasticity induced closure, roughness induced closure, oxidation induced closure and phase transformation induced closure.

This phenomenon was demonstrated to have an important role in crack growth propagation and was firstly documented by Elber [8]. He proposed an explanation for both the stress ratio R effect and the SIF range threshold ΔK_{th} . He noticed that at low loads, the fatigue specimen behaviour was very close to that of an uncracked body, whereas at higher loads, the trend shifted dramatically [13].

Elber proposed indeed that this change in stiffness was due to crack face contact, or crack closure.

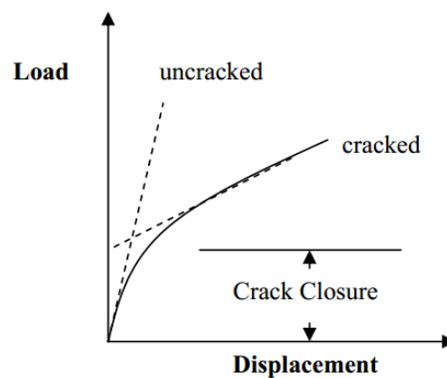


Figure 1.16: "Change in stiffness due to the crack opening/closing" [13].

The picture above depicts the change in behaviour due to the crack closure effect. In addition, he noticed that this phenomenon causes a reduction in the actual SIF range, and proposed a correction to its definition.

Indeed, he proposed the concept of the opening stress intensity factor, generally indicated as K_{op} , namely the minimum SIF required to cause the crack surfaces detachment and the consequent crack growth; any applied load below this threshold value should therefore not be considered in the damage evolution. Consequently, K_{op} can be used to calculate as effective SIF range as:

$$\Delta K_{eff} = K_{max} - K_{op}$$

Which can be employed as effective driving force in a power law, namely:

$$\frac{da}{dN} = C \Delta K_{eff}^m$$

A better understanding of the K_{op} concept can be given by figure 1.17:

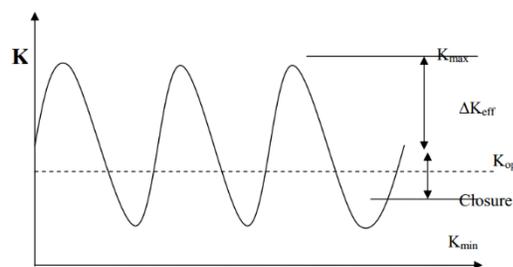


Figure 1.17: "The crack opening stress intensity factor" [13].

As to the plastically induced crack closure, it results from compressible residual stresses developing in the plastic wake. This concept assumes that a plastically transformed

area is formed at the crack tip which leaves a wake of plastically deformed zone along the crack length. This zone has residual compressive stress induced by the elastic and plastic deformation of the material during unloading. During the next cycle, while loading, the crack tip does not open unless the applied load is enough to overcome the residual compressive stress present in the plastic wake zone. Thus the effective stress at the crack tip is lowered [3].

1.1.4 Thermomechanical fatigue

In the previously illustrated models the effect of the temperature was in general not taken into account. In order to consider this effect two issues must be tackled, that are, to find an equivalent constant temperature in such a way to induce the same damage of the thermal loading cycle applied to the specimen, and to consider the potential occurrence of creep or/and oxidation, induced by the high temperature.

1.1.4.1 Taira model

As to the equivalent temperature evaluation it is interesting to resume the Taira's theory; the correlation amid thermal fatigue and isothermal LCF at high temperatures was investigated. In order to account for the temperature variations Taira modified the Manson-Coffin model as follows:

$$\lambda(T)(\Delta\epsilon_p)^n N_f = C$$

Where λ , n and C are material constants to be experimentally determined. Hence, this model assumes that the TMF is a particular case of LCF tested at an equivalent temperature, and giving the same damage of a thermal cycle amid two temperatures [1].

1.1.4.2 Neu-Sehitoglu model

The Neu-Sehitoglu model considers the total damage due to both phenomenological and thermal fatigue as the linear combination of mechanical, creep and oxidation fatigue damages.

$$D_{tot} = D_{fat} + D_{creep} + D_{ox}$$

The above equation corresponds to the Miner's formulation, and according to this model it is assumed that failure occurs when D_{tot} equals 1. The total damage is related to the number of cycle to failure as: $D_{tot} = \frac{1}{N_f}$.

The fatigue damage D_{fat} is related with the mechanical strains and can be evaluated with a Manson-Coffin-like approach. The creep damage can be computed as follows:

$$D_{creep} = \Phi_{creep} \int_0^{t_c} A e^{-\frac{\Delta H}{RT}} \left(\frac{\alpha_1 \sigma + \alpha_2 \sigma_H}{K} \right)^m dt$$

Finally, the oxidation damage contribution is given by:

$$D_{ox} = \frac{\left(\frac{h_{cr} \delta_0}{B \Phi_{ox} k_p^{eff}} \right)^{-\frac{1}{\beta}} \left(2 \Delta \epsilon_{mech}^{1+\frac{2}{\beta}} \right)}{\epsilon_{mech}^{1-\frac{\alpha}{\beta}}}$$

For further details, look up [10].

1.1.4.3 *Chaboche model*

The Chaboche model is a fatigue damage evolution model, based on stresses and on the assumption that the cumulated damage is not linear. Another important aspect is its definition of reciprocal interaction between creep and fatigue damages. This model defines an incremental damage as the sum of mechanical fatigue and creep damage contributions, both depending on the instantaneous value of the total damage:

$$dD = dD_{fat} + dD_{creep}$$

Further details on these models and further models can be found in the TMF overview given in [10].

1.2 Residual stresses

1.2.1 Overview

The residual stresses are internal stresses existing in the specimen in absence of any load applied externally. They are said to be "self-equilibrated", that is, both compressive and tensile residual stresses must exist in the body to achieve a self-balancing condition.

$$\int \sigma dA = 0$$

$$\int dM = 0$$

Above all, the RS can be classified according to their scale of action, namely:

- Type 1: Macroscopic RS, constant in magnitude and scale of action over a relatively large area (several grain diameters).
- Type 2: Microstructural RS, constant over an area of about the size of a grain.
- Type 3: Intergranular RS, they are mainly due to dislocations and reticular defects, and are not constant in magnitude even at the Intergranular scale.

In the structural field the main attention is on the RS of type 1, that is, macroscopic RS. However, in some cases there is interest also for the RS of type 2, especially when dealing with superficial coatings [11].

RSs are implied by the presence of an incompatible strain field produced from the extern, due to a change in the object geometry and shape. As to the origins of the RS, the possibilities are:

- Mechanically induced RS: from milling, turning, extrusion, cold working, etc.
- Thermally induced RS: due to non-uniform heating and cooling processes, such as welding, quenching, tempering and other heat treatments.
- Chemically induced RS: in this case are induced by volume variations due to physical/chemical transformations, such as phase change, reactions, precipitations, coating depositions, etc.

1.2.1.1 *Welding residual stress*

In this case the RS are both due to thermal stresses and plastic strains; the longitudinal RS has a peak in the welding region

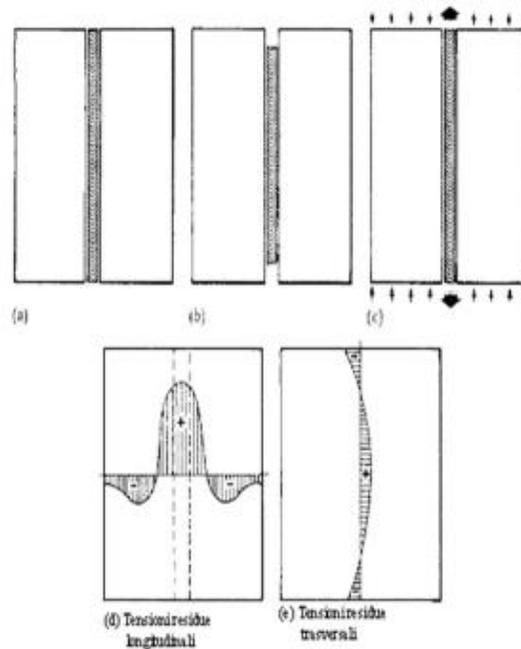


Figure 1.18: "weldments RS [11]"

The combination of thermal and mechanical RS in the HAZ zone (thermally affected zone), might produce premature cracking and rupture.

In the case of fig.1.18 the longitudinal stress can be calculated analytically as follows:

$$\sigma_x(y) = \sigma_m \left(1 - \left(\frac{y}{b}\right)^2\right) e^{-\frac{1}{2}\left(\frac{y}{b}\right)^2}$$

1.2.1.2 Shot-peening residual stresses

This process is performed on components in order to achieve beneficial effects on the fatigue endurance. Indeed, through the indentation of small balls it is possible to provoke a compressive residual stress state on the component surface.

The process actually produces the stretching of the surface layers through the balls indentation, and the plasticization of the sub-superficial layers through the Hertzian pressure.

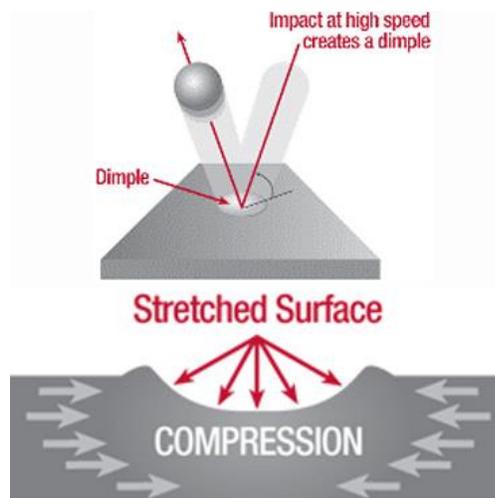


Figure 1.19: "Shot peening treatment" [11].

1.2.1.3 Quenching residual stresses

The RSs arise after quenching due to the temperature gradient occurring during its cooling. Indeed, the external layers of the component are subjected to a more intense and rapid cooling with respect to the inner layers, and to a temperature drop correspond a volume change, which is at the basis on the RSs formation. Namely, the outer layers shrink more rapidly than the others, therefore they will be initially subjected to a tensile stress, while the inner layers to a compressive stress. Afterwards, also the inner part cools down and shrinks causing a compressive stress state in the outer layers and conversely a tensile stress state at the core. Figure 1.20 qualitatively shows the phenomena for a quenched cylinder:

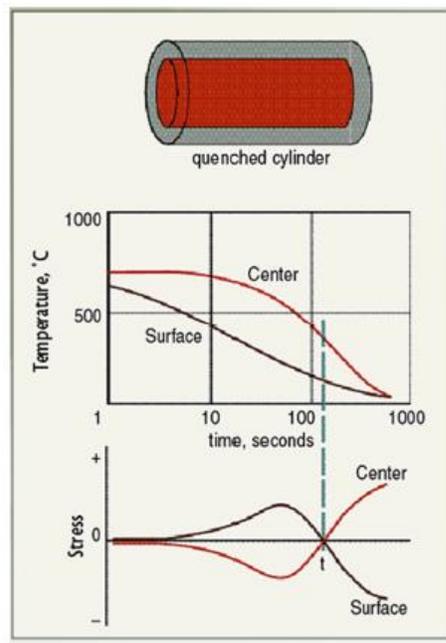


Figure 1.20: "Cylinder specimen quench" [11].

1.2.2 Residual stresses measurement methods

The measurement methods are classified according to the degree of damage they imply on the specimen, namely:

- Non-destructive techniques: X-ray diffraction, ultrasounds, magnetic method, photo-elasticity method.
- Semi-destructive techniques: hole drilling, strain gauges, rosettes.
- Destructive techniques: layer cut, section cut.

1.2.2.1 *Non-destructive methods*

Ultrasounds velocity method

When a body features internal residual stresses the propagation velocity of the sound waves is altered. This is the physical phenomenon exploited in this kind of measurement; let be v the sound velocity across the material subjected to internal stress, v_0 the velocity in their absence, and k a parameter related to the material, the internal stress σ_i can be derived as follows:

$$v = v_0 + k\sigma_i$$

It is preferable to use ultrasounds, because at high frequencies the sound waves better propagate in the material and the wavelengths are measurable [11].

Photo-elastic method

This method can be employed only for some material categories, such as glass. It is commonly used to measure residual stresses in tempered glasses.

Magnetic methods

These techniques exploit the magnetic properties of the materials, the most commonly used is the Barkhausen noise method. It consists in the measure of the magnetic field variations ascribed to the internal stresses.

A constant external magnetic field is applied, and the magnetization of the material is measured: this property is directly proportional to the magnetic permeability whose value is influenced by the RS.

$$M = \mu H$$

Where M is the material magnetization, H is the applied magnetic field and μ is the magnetic permeability.

X-ray diffraction

The diffraction phenomenon occurs when a crystal lattice is investigated by impinging rays with a certain angle θ and assuming that the wavelength is comparable to the inter-planar distance in the crystal lattice.

Since the inter-planar distance d_0 is for each material is known a priori, this technique assumes to measure the actual inter-planar distance of the stressed material d_1 and therefore to estimate the internal stress state thanks to this information.

The diffractometric technique assumes indeed to impinge the material with a beam of monochromatic X-rays with a sufficient energy to interact with the material atoms, which will react in turn emitting a radiation of the same wavelength.

In order to measure the interplanar distance we can exploit of the Bragg equation

$$d = \frac{n\lambda}{2 \sin \theta}$$

After some passages it follows:

$$\Delta d = -d \operatorname{ctg} \theta \Delta \theta$$

And by definition the strains are calculated as:

$$\epsilon = \frac{\Delta d}{d} = -\operatorname{ctg} \theta \Delta \theta$$

Subsequently, the internal stress can be determined by the calculated strains, in figure 1.21

the machine used in this measurements is reported:

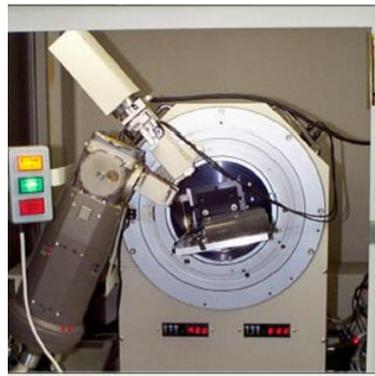


Figure 1.21: "X-ray diffractometer" [11].

1.2.2.2 Destructive methods

Stäblein method

This is a layer removal method and can be used for linear beams and planar plates. By removing some superficial layers, the RSs are relaxed and the cut surfaces tend to deform; the layer removal can be performed mechanically, chemically or electrochemically.

After the layer removal the surfaces curvature is measured, for example with gauges, and by means of analytical formulas is possible to calculate the internal stresses.

Sachs method

It is a layer removal method and is used for cylinder elements characterized by an axial symmetric RS distribution, constant along its longitudinal axis.

Some layers are removed by turning and subsequently the longitudinal and circumferential strains are measured on the component surface. Then, the RSs components can be calculated analytically as follows:

$$\sigma_l = \frac{E}{1 - \nu^2} \left[\frac{(S_l - S)dL}{dS} - L \right]$$

$$\sigma_r = \frac{E}{1 - \nu^2} \frac{(S_l - S)C}{2S}$$

$$\sigma_c = \frac{E}{1 - \nu^2} \left[\frac{(S_l - S)dC}{dS} - \frac{(S_l + S)C}{2S} \right]$$

$$\text{Where } L = \epsilon_l - \nu\epsilon_c \quad C = \epsilon_c + \nu\epsilon_l$$

Where σ_l , σ_r , and σ_c are respectively the longitudinal, radial and circumferential stresses, and S_l and S are the external and internal cylinder section area.

1.2.2.3 Semi-destructive methods

Hole drilling method

The hole drilling strain gauge method is the most widely used for measuring RSs. The measurement procedure involves some basic steps. A three or six strain gauge rosette is installed on the specimen, above the part of interest for the RS measurement.

Afterwards a precision milling guide is attached to the test part and centred over the drilling target on the rosette. Hence, after zero-balancing the gage circuits, a small, shallow hole is drilled and the readings are effected of the relaxed strains, corresponding to the initial RSs.

Finally, using data-reduction relationships, the principal RSs and their angular orientation are calculated from the measured strains [12].

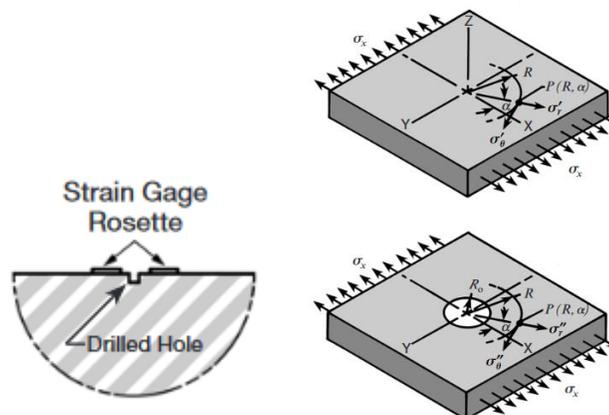


Figure 1.22: "Hole drilling method [12]".

1.3 Fatigue life prediction models including residual stresses

1.3.1 The approaches in literature

The approaches found in literature dealing with fatigue crack growth analysis in presence on residual stresses can be broken down into different steps that must be carried out to fulfil the finite element simulation; these steps are essentially:

- Geometry model realization
- Application of the residual stresses on the unnotched model
- Creation of the initial crack on the model
- Definition of the parameters and the method used to account for the residual stresses in the fatigue analysis
- Fatigue crack growth simulation

Hereafter the most interesting steps will be analysed.

1.3.1.1 *Geometry model realization*

The case studies in literature are generally based on standard geometries, in which often a symmetry condition is used to spare computational effort during the simulations, and the crack is applied on the symmetry line, i.e. SENT specimens [13, 14], welded plates [15, 16, 17], notched CT specimens [18, 19, 20], M(T) specimens [21].

1.3.1.2 *Application of the residual stresses on the unnotched model*

This is the first critical issue, because it was found by many authors that an accurate residual stress field applied to the model is crucial in order to obtain a good correlation between the experimental fatigue crack growth curves and the ones obtained by FEM.

The mechanical processes which the majority of the researches have dealt with, as to the FE modelling of residual stress fields are the shoot-peening treatments [22], welding processes [23, 21, 24, 25], four-points bending tests [26, 13, 27], or quenching processes [19].

In the literature study three different approaches have been identified; An used approach is to simulate the physical process causing the residual stress fields by means of a commercial software for elastic-plastic analysis, such as ABAQUS.

In this case the main issue is to insert into the program the right material constants, to choose an appropriate yield criterion (ex: Von Mises) and a hardening model (isotropic, kinematic or combined) in order to rightly characterize the material behaviour, and furthermore to properly simulate the physical process in terms of applied stress, temperatures and constraints.

The main shortcomings of this approach are the computational effort of doing elastic-plastic analysis, and the difficulty when trying to reproduce accurately the manufacturing process induced plastic deformations and residual stresses [13].

Generally, this method is used for weldments analysis, by means of coupled thermal and mechanical welding simulations [23], or for simulated four-points bending tests by

which the material undergoes an overload causing residual plastic strains and so self-equilibrated residual stress fields [27].

A second possibility is to employ an Abaqus subroutine, in order to insert into the geometrical model a desired residual stress or strain field as initial condition.

As was summarized by Servetti et al. [21] this methodology can be in turn broken down into two different approaches, that are the "displacement input method" and the "stress input method".

In both cases there is the need to precisely know in advance the residual stress state of the component after the manufacturing process. In the displacement input method, the measured strains are inputted into the numerical model as initial conditions, indirectly producing the residual stress field into the model. In the alternative method, the measured residual stresses are directly inputted in the model, also in this case by means of a dedicated subroutine such as "SIGINI". Then, the Abaqus command "UNBALANCED STRESSES" can be used to balance the inputted stress field and so to satisfy the balance condition.

The latter approach is generally preferable because more accurately reproduces the experimentally measured residual stresses into the FE model, and satisfies the virtual work principle. Furthermore, since each node must be constrained, the displacement input method cannot be used to develop a FCG model accounting for the residual stresses redistribution phenomenon, which plays a relevant role [21].

Anyways, O'Dowd [27] pointed out the difficulties in properly employing this approach. The first issue is the non-uniqueness of the set initial condition, since a number

of different loading histories could result in the same residual stress field. Another practical problem in its implementation is that when in a stress-free region a residual stress field is applied the resultant stress distribution after an equilibrium step generally differs from the expected one. In fact, was demonstrated that the residual stress fields during the equilibrium step smoothed out to satisfy the equilibrium condition over the whole body causing a discrepancy between the inputted and the output stress fields [27].

In order to fix this, a proportional integral adjustment can be used, such as the reported equation:

$$\sigma(x)_{inp}^{i+1} = \sigma(x)_{inp}^i + \beta(\sigma(x)_{targ} - \sigma(x)_{out}^i)$$

Very briefly, this iterative method can be implemented to better match the inputted and output stress fields.

The last practical issue, is that generally residual stresses are measured in specimens only at selected positions, therefore a full stress distribution is not often available.

Another approach presented in literature to model the residual stress fields is based on the eigenstrain concept firstly developed by Mura [28] and Ueda [29] and then used by some authors in their simulations such as Roberts [13], Matos [25].

This method is also regarded as inherent strain approach, and was developed for the weldments. Very briefly, consists in a second-order tensor of strain fields that represents the plastic deformation due to manufacturing operations [30].

It is well-known that manufacturing processes induce residual strain fields, which generally do not satisfy the geometric compatibility relations. Consequently, residual

stresses must be present to eliminate the incompatibility, thereby restoring geometric continuity of the component. The description given by Matos et al [25] is here briefly reported:

"The incompatible property is described mathematically by the six strain compatibility equations in terms of the three independent displacement components u_i in Cartesian coordinates it follows:

$$R_{pq} = \epsilon_{pki} \epsilon_{qlj} \epsilon_{ij,kl}^*$$

Where the usual index notation with implied summation is employed with commas denoting differentiation. Where ϵ_{pki} denotes the third order alternating tensor.

When the symmetric tensor R_{pq} vanishes for a given field of eigenstrains ϵ_{ij}^* no residual stresses are required to restore the geometric compatibility.

The total (compatible) strain tensor can be decomposed as:

$$\epsilon_{ij} = \epsilon_{ij}^e + \epsilon_{ij}^*$$

Where ϵ_{ij}^e denotes the elastic strain tensor necessary to restore compatibility created by the eigenstrain tensor ϵ_{ij}^* , and the strains arising from separately applied mechanical loads. Under such conditions, the final linear-elastic stresses are then given by:

$$\sigma_{ij} = D_{ijkl}(\epsilon_{kl} - \epsilon_{kl}^*)$$

In the absence of mechanically applied loads (or restraints), the response of the component to the eigenstrains must generate a residual stress field σ_{ij}^* that satisfies equilibrium:

$$\begin{aligned} \sigma_{ij}^* &= D_{ijkl} \epsilon_{kl}^e \\ \sigma_{ij,j}^* &= 0, \sigma_{ij}^* n_j = 0 \end{aligned}$$

And compatibility":

$$R_{pq} = \epsilon_{pki}\epsilon_{qlj}(\epsilon_{ij}^e + \epsilon_{ij}^*),_{kl} \equiv 0$$

In their works, Roberts [13] and Matos [25], computed the eigenstrains simply as thermal strains, by means of the methodology developed by Hill and Nelson [31]. This approach consists in setting a spatial distribution of anisotropic thermal expansion coefficients, and applying a unit thermal load, such that:

$$a_{ij} = \epsilon_{ij}$$

Anyways, the implementation of this method is not always straight forward [13].

1.3.1.3 *Definition of the parameters and the method used to account for the residual stresses in fatigue analysis*

Generally speaking, the aim is to capture the effect of the initial residual stresses on the fatigue crack growth driving force. According to the chosen empirical law, listed in the background section, this driving force may be proportional to one or more parameters.

The residual stresses affect the stress field at crack tip and have effect on the parameters K_{max} , R , and K_{min} . That is why, many authors preferred to the well-known Paris law, a two parameter driving force like the Walker law.

A common approach is the superposition method, which is suitable in case of small scale yielding condition.

According to this approach it is sufficient to compute a SIF factor only related to the residual stresses, to be then superposed to the SIF factor only related to the applied stress, namely:

$$K^{tot} = K^{app} + K^{res}$$

According to this simple rule it is possible to calculate the total maximum and minimum SIFs, and the effective stress ratio R_{eff} , namely:

$$K_{max}^{tot} = K_{max}^{app} + K^{res}$$

$$K_{min}^{tot} = K_{min}^{app} + K^{res}$$

$$R_{eff} = \frac{K_{min}^{tot}}{K_{max}^{tot}}$$

It is straightforward that with this approach the SIF range is not affected by the residual stresses, namely:

$$\Delta K = K_{max}^{tot} - K_{min}^{tot} = K_{max} - K_{min}$$

Indeed, this method is always coupled with a fatigue crack growth law accounting for K_{max} or R_{eff} instead of the SIF range only.

This approach has been employed in [32,13,33,34,21,35,36,16,17,19,24], and anyway features some shortcomings and difficulties.

The first issue is how to compute the SIF factor due to residual stresses, namely K_{res} ; the possibilities are essentially two, that is, by means of a weight function approach or by FEM. Broadly speaking, the weight functions are suitable when dealing with simple geometries, whereas FEM are necessary for more complex cases.

Assuming that the weight function is known, the K_{res} can be computed as [13]:

$$K_{res} = \int_{\Gamma_c} \sigma_{res}(x)h(x, a)dx$$

Where σ_{res} is the initial residual stress distribution and h is the employed weight function.

As to the use of weight functions for the calculation of K at the crack tip, Fitzpatrick in his analysis stated: "solutions for different geometries have been tabulated by, among others, Rooke and Cartwright, and Tada et al. Even though many solutions are available, derived by various means and levels of complexity, the results are often not applicable to new problems with complex structures and loading conditions" [37].

Roberts [13] in his work tried to compare weight function methods and FEM when dealing with residual stress fields in specimens and concluded that "comparison of this energy method for obtaining K_{res} with that of the weight function method has revealed a very significant discrepancy. The weight function method, while seemingly appropriate for external loading, appears to require at least two inherent assumptions which do not apply to residual stresses. The first of these assumptions is that the residual stress maintains its initial magnitude and distribution throughout the entire crack growth process. The second inherent assumption in the use of weight functions for residual stress is that it is sufficient to consider only that part of the residual stress that was acting on the crack plane up to the current crack tip".

Both of these conditions arising when using weight functions were demonstrated to be erroneous in practise, implying simplistic simulations, far from the experimental evidences.

As to the FEM methods to compute the SIF at crack tip the analysis of Lin and Chang [17] is subsequently reported.

"There are a few approaches for evaluating crack tip stress intensity factor by finite element method, such as the crack tip displacement extrapolation, the modified virtual crack closure technique and the J-integral. In the displacement extrapolation method, the displacement and stress data of the finite element analysis at the vicinity of the crack tip are assumed to obey their asymptotic behaviour and the crack tip parameters are computed. This method is simple but it is not easy to guarantee the accuracy. The modified virtual crack closure technique (also known as virtual crack closure integral method) is another way to compute the energy release rate. In this method, energy that is required to close the crack for one finite element length is calculated by multiplying the nodal reaction force perpendicular to the crack growth path at the crack tip node and the opening displacement at the node immediately behind the crack tip. The advantages of the technique are in its simplicities in computing the energy release rate such that only the nodal reaction force and opening displacement are used in the computation and that using their appropriate vector components, the energy release rate is decomposed into the modes I-III components. The Jintegral approach can also obtain the energy release rate through the domain integral method. The domain integral method utilizes the subroutines in the finite element program to perform integrations based on finite element geometries. Therefore, the domain integral approach is well suited for fracture mechanics analysis using the finite element method.

These methods are widely used practices and easily implemented in commercial finite element packages" [17].

Some of these approaches will be encountered after and furtherly clarified.

It is common then to employ a similar alternative to the approach above mentioned, considering that the superposition of the applied load and the residual stresses may lead to negative results, i.e. compressive stress fields, in case of negative $(K^{tot})_{min}$, this is set to zero to simulate a closed crack [36].

Another largely employed approach in literature is based on the crack closure effect firstly documented by Elber. This method is known as effective stress intensity factor range approach, and it is based on the definition of a modified ΔK_{eff} .

Elber considers that as a crack propagates, crack closure occurs as a result of plastically deformed material left in the path taken by the crack. This material is referred to as the plastic wake. The plastic wake enables the crack to close before minimum load is reached, and Elber reasoned that the stress intensity factor at the crack tip does not change while the crack is closed even when the applied load is changing [36].

Consequently, the target is to compute a crack opening stress S_o or its relative crack opening SIF, namely K_{op} , in the residual stress field. This factor can then be used to compute the effective stress range as:

$$\Delta K_{eff} = K_{max} - K_{op}$$

The crack opening SIF can be calculated by means of experimental methods or numerically; many researchers tried to compute it through FEM as plasticity induced crack

closure. McClung [38] have provided a critical overview of this works. The basic algorithm employed is always the same: a crack tip node is released after each cycle allowing the crack to advance and the resultant plastic wake to incrementally form, and the crack opening is accounted by monitoring the crack faces contact [36].

Anyways only few studies have dealt with fatigue crack closure in presence of residual stresses, like as Beghini and Bertini [33], Lacarac [34], and Choi and Song [39].

More recently Larue and Daniewicz [36] performed a study about the fatigue crack growth from a hole with a pre-existing compressive residual stress using two-dimensional elastic-plastic finite element analyses. They employed both the ΔK_{eff} method and the superposition approach described above, finding that accounting for crack closure effect leads to more precise results.

A requirement for the plasticity-induced crack closure FEA simulation is that the element size must be smaller than the size of the plastic zone in front of the crack tip, otherwise the plasticity zone is not resolvable [30].

However, in their studies, there is no emphasis on the residual stress redistribution due to the fatigue crack growth and the effects of the plastic wake left behind the crack are not accounted.

Recently Garcia et al [30] carried out a study on the fatigue crack growth propagation in a four-points bending beam in presence of residual stresses. The goal of their work was to supply to the lacks of the models found in literature, that is, to develop a model to take

into account indeed the residual stress redistribution, due to the crack advancement, and the plastic wake effects.

Indeed, Garcia employed a different strategy to account for residual stresses, that is an alternate method to the ΔK_{eff} approach. Namely, contrary to its previous definition they used the finite element method to directly compute an effective SIF, or K , of the actual residual stresses during the fatigue crack growth, and not the initial ones. This, with the aim to account for the residual stress redistribution.

Hence, the actual K and R are computed accounting for both the externally applied stresses and the internal residual stresses under evolution; namely:

$$\Delta K = effective(K_{max} - K_{min})$$

$$R = \frac{K_{min-effective}}{K_{max-effective}}$$

And finally the fatigue crack growth rate is computed by integration of a Paris-like power law alike the other approaches.

Actually, the effective K_{max} and K_{min} were directly computed with the displacement correlation method from the finite element nodal displacements behind the crack tip along the crack face, namely:

$$K = \frac{Eu}{4(1-\nu^2)} \sqrt{\frac{2\pi}{r}}$$

Considering a plane-strain condition, and where u is the node displacement and r the distance from the crack tip; E and ν are the material elastic modulus and Poisson's coefficient.

Garcia et al. performed the simulations in three cases, that is, without crack closure effect, with crack closure effect and with both crack closure and plastic wake effects.

In order to account for the crack closure effect in the model a rigid crack closure condition was set; namely an analytical rigid surface was added at the symmetry plane of the model (where the crack was added), and a contact interaction was set between the crack surface and the analytic rigid surface.

Basically, to account for the plastic wake effects, at each load cycle the crack is advanced of one element size by releasing the crack tip nodes. This process is repeated several times for the formation of the plastic wake [30], and at each increment K and R are calculated to be used in the fatigue crack growth simulation.

1.3.1.4 *Fatigue crack growth simulation*

After that the residual stress fields have been determined and employed to properly compute the fracture mechanics corrected parameters, with the approaches before discussed, there is the choice of the empirical power law to employ in order to calculate the fatigue crack growth rate and then the fatigue life until a critical crack size is reached.

The most important power laws have already been reported in the Background section, and some of them have been frequently used in literature when dealing with residual stress fields in specimens.

However, the choice of the FCG law is not independent from the approach to correct the fracture mechanics parameters employed, they are instead strictly related.

For instance, Roberts [13] in his work employed the Forman equation here reported again:

$$\frac{da}{dN} = \frac{C\Delta K^m}{(1-R)(K_{crit} - K_{max})}$$

In combination with a superposition approach, that is the summation of the SIF related to the applied stress and the one related to the internal residual stresses.

He developed a routine in which starting from some input parameters (Forman law fitting constants, initial and final crack size) was possible to implement the Forman equation and evaluate the crack growth rate. Actually after each cycle the crack was advanced releasing the crack tip nodes, and the process reiterated until the predetermined failure crack size was achieved.

Here the scheme from the Roberts work is reported:

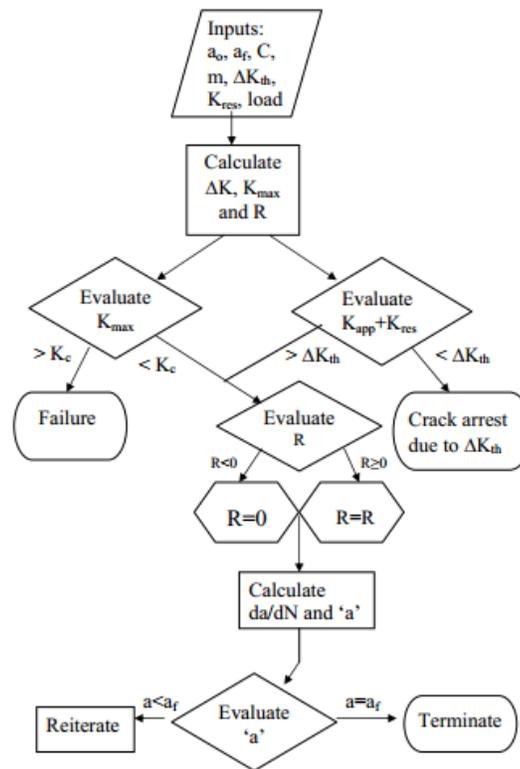


Figure 1.23: "Computation scheme [13]".

As pointed out by Roberts, the choice of the power law is suitable, because it allows to take into account both R and K_{max} corrected via the superposition approach.

As reported before, Larue et al. [36] performed a study employing the ΔK_{eff} and the superposition approaches, in order to make a comparison of the two methodologies.

Firstly, their analyses allow the determination of the crack opening stress as the crack propagates through the residual stress field, from which the effective stress intensity factor range ΔK_{eff} and the fatigue crack growth is predicted.

Furthermore, they pointed out that predictions from the closure-based method are highly dependent on the FCG constitutive relationship used, highlighting the need for experimental methods to reliably measure this correlation.

As to the superposition simulation they simply computed K_{res} by means of a weight function, whereas for the crack closure approach they computed the crack opening stress S_o via FEM simulation.

The equations employed for the FCG prediction were the Liu baseline crack growth curve for " $R = 0.7$ " [99], and the NASGRO equation presented in the Background section. Each curve was used to separately predict crack growth rates from the computed effective stress intensity ranges.

The procedure is here reported for the crack closure approach:

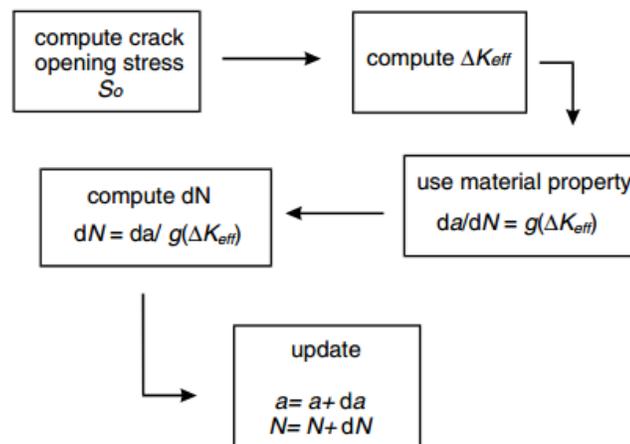


Figure 1.24: "Computation scheme [36]".

Relevant was the contribution of Servetti and Zhang [21] in their studies; they proposed a simple method for predicting fatigue crack growth rate in welded butt joints.

Three different empirical crack growth laws were employed using the material constants that were obtained from the base material coupon tests. Based on the superposition rule of the linear elastic fracture mechanics, welding residual stress effect was accounted for by replacing the nominal stress ratio R in the empirical laws by the effective stress intensity factor ratio R_{eff} . The key part of the analysis process was to calculate the stress intensity factor due to the initial residual stress field and also the stress relaxation and redistribution due to crack growth. The finite element method in conjunction with the modified virtual crack closure technique was used for this analysis. Fatigue crack growth rates were then calculated by the empirical laws and comparisons were made among these predictions. Test samples were M(T) geometry made of aluminium alloy 2024-T351 with a longitudinal weld by the variable polarity plasma arc welding process.

For each crack length, the SIF was calculated at the applied stress level followed by releasing the crack tip node to the next crack length. This process was repeated for the crack length range from the initial to the final crack size. Therefore, residual stress redistribution due to crack extension was modelled [21].

Virtual crack closure technique, or modified virtual crack closure technique is a method to compute the SIF, starting from the strain energy release rate, namely:

$$G = \frac{1}{2t\Delta a} F_{yy,i}(v_j - v_j^*)$$

Then in case of plane stress:

$$K = \sqrt{GE}$$

where $F_{yy,i}$ is the nodal reaction force perpendicular to the crack growth path at the crack tip node I , $v_j - v_j^*$ is the crack opening displacement at node j immediately behind the crack, Δa the crack extension length that equals to the crack tip element size, and t the plate thickness [21].

In case G_{res} and G_{app} are computed with two separate FEM simulations, the total contribution must account also to the mutual work amid internal residual stresses and applied load:

$$G_{tot} = \frac{1}{2t\Delta a} (F_{app} + F_{res})(v_{app} + v_{res})$$

The superposition approach equations are here reported again for convenience:

$$\Delta K_{tot} = (K_{app,max} + K_{res}) - (K_{app,min} + K_{res}) = \Delta K_{app}$$

$$R_{eff} = \frac{K_{app,min} + K_{res}}{K_{app,max} + K_{res}}$$

Finally, these parameters were used in the Walker equation, NASGRO equation and with the Harter-T-method. The first two equations were already reported in the Background section, whereas the Harter-T-method will be herein briefly explained:

Harter-T-method

This method is well-suited to account for the residual stress redistribution which is considered by the change of the stress ratio at each crack step, in that it allows to compute at each of these steps the proper crack growth rate da/dN in accordance to the new stress ratio R .

Then this method is a well-suited alternative to the Walker or NASGRO equations used by other authors.

The Harter T-method employs an interpolation or extrapolation to determine a stress intensity factor for a stress ratio R of interest as a function of the FCG rate [30]. A minimum of two baseline FCG curves (da/dN vs ΔK) are required to apply it. It employs the Walker equation which takes into account the stress ratio R and is here newly reported:

$$\frac{da}{dN} = C \left[\frac{\Delta K}{(1-R)^{1-m}} \right]^n$$

If we consider two FCG curves, for different R and at the same crack growth rate $(da/dN)_i$, it follows:

$$\left(\frac{da}{dN} \right)_i = C \left[\frac{\Delta K_1}{(1-R_1)^{1-m_i}} \right]^n = C \left[\frac{\Delta K_2}{(1-R_2)^{1-m_i}} \right]^n$$

Which yields to:

$$\frac{\Delta K_1}{(1-R_1)^{1-m_i}} = \frac{\Delta K_2}{(1-R_2)^{1-m_i}}$$

As first is needed to solve for the fitting constant relative for such crack growth rate m_i :

$$m_i = 1 + \left[\frac{\log_{10} \left(\frac{\Delta K_1}{\Delta K_2} \right)}{\log_{10} \left(\frac{1-R_2}{1-R_1} \right)} \right] \text{ with } R_1 \text{ and } R_2 \geq 0$$

$$m_i = 1 + \left[\frac{\log_{10} \left(\frac{K_{max1}}{\Delta K_2} \right)}{\log_{10}((1-R_1)(1-R_2))} \right] \text{ with } R_1 < 0 \text{ and } R_2 \geq 0$$

Subsequently it is possible to compute the SIF range for any stress ratio of interest R_{int} , namely:

$$\Delta K_i = \Delta K_1 \left(\frac{1 - R_1}{1 - R_{int}} \right)^{m_i - 1} \quad \text{with } R_1 \text{ and } R_2 \geq 0$$

$$\Delta K_{max,i} = \frac{\Delta K_2 (1 - R_2)^{m_i - 1}}{(1 - R_{int})^{1 - m_i}} \quad \text{with } R_1 < 0 \text{ and } R_2 \geq 0$$

Finally, by means of a linear interpolation in the log-log scale the entire FCG curve can be found for the stress ratio of interest [13].

The Harter-T-method was not only used coupled to the superposition method, but also with the definition of effective ΔK and R given by Garcia [30] in his work.

The algorithm employed by Garcia was analogous to the previously examined ones and is here reported:

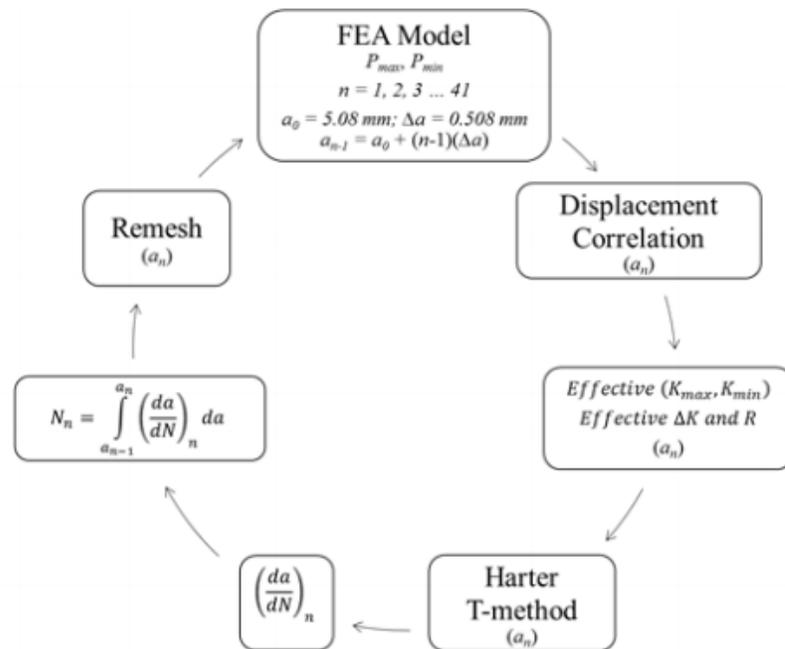


Figure 1.25: "Computational procedure" [30].

Also in this case, the procedure starts from the initial and material data (E, ν, a_0, a_f) and plans to compute the fracture mechanics parameters of interest, in this case via displacement correlation, to be then employed in a FCG empirical law. This process is then iterated till the critical crack size is reached.

Schnubel and Huber [14], presented a study about a numerical approach for predicting the fatigue crack growth in Aluminium CT specimens containing one line of laser heating, which is a process employed to create compressive residual stresses and increase the fatigue life of the specimens.

The proposed methodology exploited the MVCCT technique to calculate the strain energy release rate like it has been done by other authors, in order to extract a total SIF factor K_{tot} accounting for both applied and residual stresses. Then a prediction of the resulting fatigue crack growth rates by an empirical crack growth law was performed.

In addition, they validated their numerical approach with experiments, finding a good agreement.

As ΔK_{tot} and R_{tot} were calculated through the MVCCT techniques they were employed in the Walker equation to predict the FCG.

The steps in the extended procedure adopted are summarized by Schnubel and Huber in the follow-up figure:

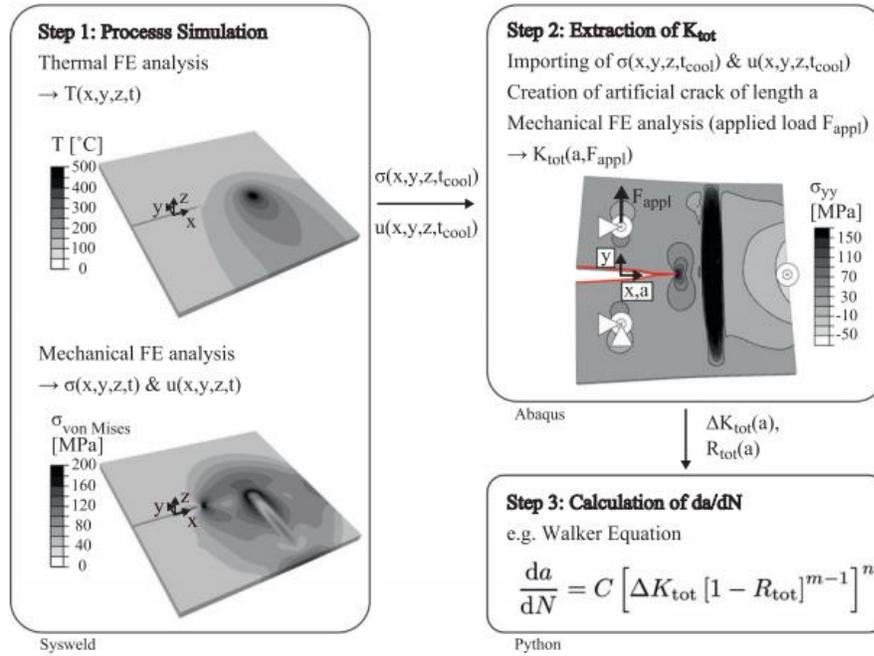


Figure 1.26: “Computation scheme [14]”.

Hence, briefly summarizing, they started with a simulation on the software “Sysweld” to simulate the process determining the thermal residual stresses and subsequently they transferred this data on Abaqus for applying the MVCCT and calculate K_{tot} and R_{tot} . Finally, by means of the well-known Walker equation the FCG rate was computed.

Finally, the fatigue life was computed by integration as follows:

$$\int_0^N dN = \int_{a_0}^a (C[\Delta K_{\text{tot}}[1 - R_{\text{tot}}]^{m-1}]^n)^{-1} da$$

Lin and Zhu [16] presented a study on the change of residual stress distribution during fatigue crack propagation. Their research attempted then to predict the crack

propagation by considering the residual stress fields. An analysis approach for the change in residual stress distribution is then established according to the diffusion theory of cavity.

The authors also pointed out the importance of the proper selection of the effective crack increment used for calculating the crack propagation rate, according to the distribution state of residual stress.

The approach employed by Lin and Zhu can be summarized in few steps; as first the critical crack length was calculated starting from the fracture toughness, as follows:

$$a_c = \frac{K_C^2}{\pi\sigma^2}$$

The next step is to calculate the appropriate crack length increment Δa_i according to the distribution of residual stress. The selection principle of the crack length increment takes place in such a way that there is no obvious mutation of the stress values between two consecutive incremental points. Namely, it is ensured the following condition:

$$\frac{|\sigma_r(x_r) - \sigma_r(x)|}{|\sigma_r(x_r)|} \leq \eta$$

Where the first term is the residual stress of crack tip, the second is the distribution function of residual stress on the direction of crack propagation and η is a set constant.

Then another control on the calculation accuracy is performed on the SIF factors:

$$\frac{\Delta K_i - \Delta K_{i-1}}{\Delta K_i} < \epsilon$$

The final step is to calculate the crack propagation rate of the determined crack increments. The total stress intensity factor and the stress ratio were the parameters used

for the calculation of crack propagation rate, which was computed according to the superposition rule, already explained for other approaches.

Then, these parameters have been employed in the Forman equation as follows:

$$\left(\frac{da}{dN}\right)_i = \frac{C(\overline{\Delta K_{tot,i}})^m}{(1-R)K_C - \overline{\Delta K_{tot,i}}}$$

Where,

$$\overline{\Delta K_{(tot,i)}} = \frac{\Delta K_{tot,i} + \Delta K_{tot,i-1}}{2}$$

And the subscript i refers to the crack increment Δa_i . The procedure stops after a critical crack length is reached and further allows to count the number of cycles to failure.

The interesting fact of this work is the attention deployed to the selection of the crack increment based on the stress distribution, indeed other authors in older researches did not implement a method to continuously update the optimal incremental crack length step.

However, in this work the residual stress redistribution due to fatigue crack growth was not taken into account, overestimating the FCG rate due to the constant tensile residual stresses present in the specimen.

The following picture better clarifies the procedure:

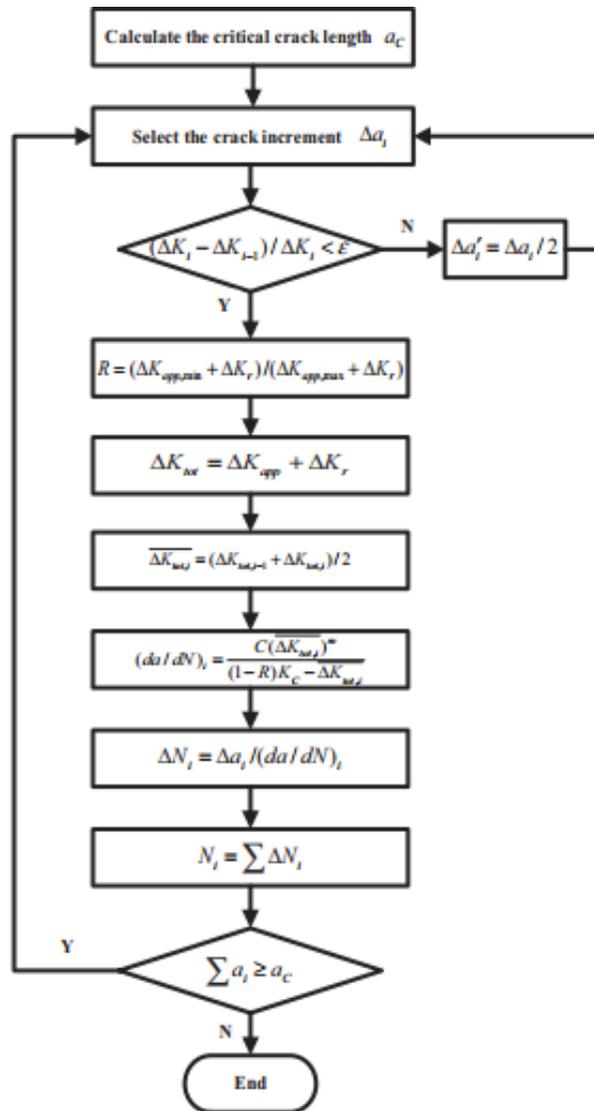


Figure 1.27: "Computation procedure" [16].

In the study of Lee and Chang [17] a three-dimensional thermal-mechanical finite element model was first developed in order to accurately predict the weld-induced residual stresses. Then, the rate of fatigue crack growth in the welds subject to the applied mechanical stress in conjunction with the residual stress was predicted by calculating the

stress intensity factors resulting from the residual stress field using the modified J-integral definition.

The fatigue crack growth rate due only to the applied mechanical stress was also computed for comparison.

In this study, the J-integral technique was adopted to calculate the stress intensity factor resulting from the residual stresses.

After that the SIF was computed as:

$$K = \sqrt{\frac{EJ}{\beta}}$$

In order to account for the residual stress in the FCG the superposition method was used, according to the following formulas newly reported for convenience:

$$K_{eff} = K_{applied} + K_{residual}$$

$$\Delta K_{eff} = (K_{applied}^{max} + K_{residual}) - (K_{applied}^{min} + K_{residual}) = K_{applied}^{max} - K_{applied}^{min} = \Delta K_{applied}$$

$$R_{eff} = \frac{K_{eff}^{min}}{K_{eff}^{max}} = \left(\frac{K_{applied}^{min} + K_{residual}}{K_{applied}^{max} + K_{residual}} \right)$$

As was pointed out before, the superposition approach needs to be coupled with a power law incorporating the corrected stress ratio and/or the maximum SIF, then the following modified Paris-like law was used:

$$\frac{da}{dN} = \frac{C(\Delta K_{eff})^m}{1 - R_{eff}}$$

Chapter 2

2) A python program for the optimized residual stress importing into the FEA model

2.1 Introduction

As was stated by many authors in literature the first critical step in order to perform an analysis in presence of residual stresses is the correct determination of their magnitude and field. The most modern measurement methods employed in research have been described in section 1.2; in addition to the experimental measurement methods, an alternative is the residual stress simulation, which implies to simulate the specific manufacturing process or loading history at the ground of the internal stress development in the specimen.

When using experimental measurements for the RS, the main shortcoming highlighted in the previous sections is due to the fact that such stresses can be estimated only in a limited part of the specimen and until a certain depth, still with some accuracy limits, as stated in [12].

Subsequently, the residual stress data must be inputted into a FEA software for the further calculations, such as fatigue life estimates or fatigue crack simulations.

This method was used in literature firstly by O'Dowd [27], in that case for studying the correct way to simulate the crack growth in a well-established RS field in a 2D simple model.

As was noticed by O'Dowd, when inserting unbalanced stresses into a FEA model the software needs to firstly perform an equilibrium step, in which the inserted stresses are redistributed and smoothed out to achieve an equilibrium condition for each model's element.

Hence, when the inputted stresses are strongly unbalanced, or are anyway inserted only over a small portion of the model geometry (ex: where measured), the resulting RS field after the equilibrium step are likely to be remarkably different from the desired ones.

In order to comply with this, O'Dowd developed a method to iterate over the inputted stress, obtaining to some extent an improvement in the resulting RS distribution.

As mentioned before, the iterative equation employed was:

$$\sigma(x)_{inp}^{i+1} = \sigma(x)_{inp}^i + \beta(\sigma(x)_{targ} - \sigma(x)_{out}^i)$$

Where σ_{inp} is the inputted stress into the FEA program and is updated at every iteration, σ_{out} is the resulting stress after the equilibrium step, σ_{targ} is the desired stress in the region, and finally β is the integral factor.

The employed program/method/subroutine to perform this adjustment was not available in literature, so the second part of this thesis work dealt with the development of my own program.

2.1.1 Applications

The program illustrated in this section was developed for the testing of a product of FCA Automobiles. The research scope of the FCA's engineers was to properly compute the residual stress fields of a car engine, or some of its components, inherited by the casting manufacturing process, and so the cooling phase along with the follow up heat treatments, in order to subsequently employ these data in a fatigue cracking analysis.

The target was indeed to achieve more satisfactory results in the useful-life estimates of the product, and actually the incorporation of the residual stresses into the model brought to more accurate results, in comparison to experimental data obtained by means of test benches.

The initial part of the simulation was performed on a dedicated FEM software, optimized for casting procedures. The correct simulation of this part is necessary to compute a reliable stress field, comparable to real data.

Subsequently the internal stress data are stored by the software in a text file, whereby to each element ID corresponds a stress tensor.

The next steps are performed by means of the Abaqus solver, hence the stress data will be read as input condition for the following simulations.

The product on which the investigation was made is the engine 2.2 JTD N-S 180CV-210CV-240CV, which is depicted in pictures 2.1 and 2.2 with the courtesy of FCA Automobiles.

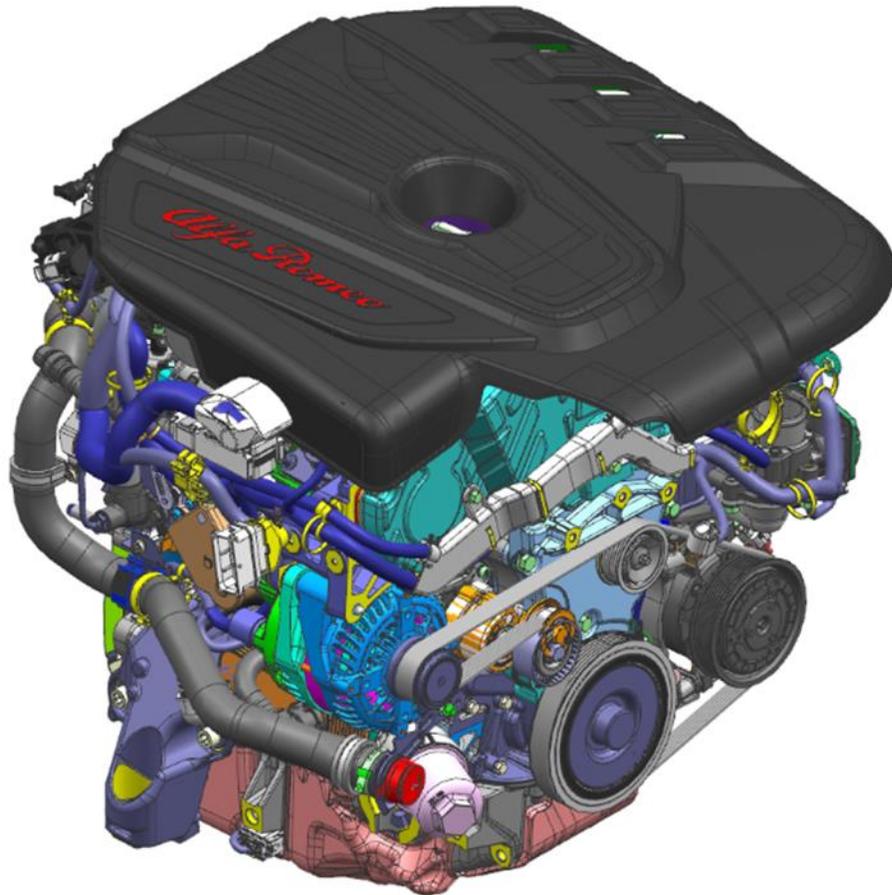


Figure 2.1: "Comprehensive assembly of the engine 2.2 JTD".

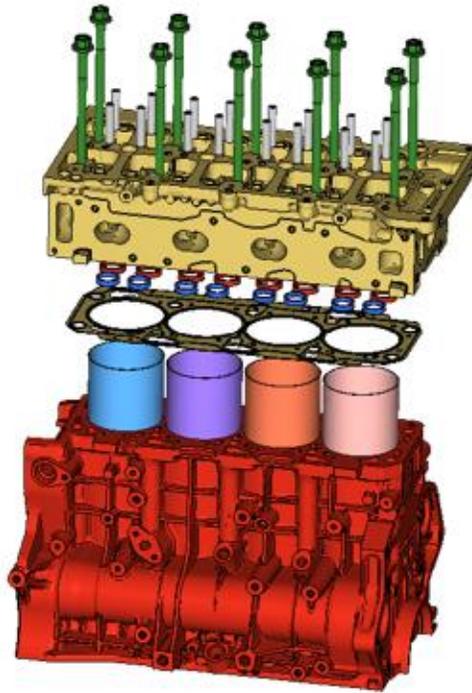


Figure 2.2: "Engine 2.2 JTD, basis, pistons and head".

2.1.2 The program

The software considered for the finite elements analysis was Abaqus, a product of Dassault Systems Simulia Corporation, which is the most used in literature in this research field.

As to the residual stresses, the CAE software offers more possibilities; when an analysis is performed Abaqus generates a set of output files containing the results, among which an output database, with extension ODB which stores all the analysis results information at each element's integration point.

It is then possible, according to the Abaqus user's manual [9] to start a completely new analysis only importing the residual stresses as initial conditions from the ODB file, all over the model or only over a selected set. A second possibility is instead to select from the viewport a set made out of finite elements and to assign a uniform stress tensor manually.

Hence, there are two practical applications which can be covered thanks to these techniques, that are for instance to get the experimental residual stresses measurements over a small area of the specimen, for example by means of a cutting and some strain gauges, to be subsequently inputted into the FE model. Whereas, the other practical application might be to perform a first analysis, with the aim of only computing the residual stresses all over the FE model, to be then inputted as initial condition in a second analysis, for instance for fatigue life estimates. A practical example might be to simulate an iron alloy solidification into its mould to get its initial internal stress field, to be then imported in a second moment in a fatigue or cracking analysis.

However, in both cases attention must be paid to achieve a satisfactory accuracy, and it must be taken into consideration that both the RS stress measurements (via either destructive or non-destructive methods) and the manufacturing process FE simulation may lead to misleading results.

Moreover, another issue affecting these RS importing techniques is the RS smoothening during the equilibrating step, which may be less or more severe according to different cases.

As already mentioned, the Abaqus user's manual recommends, when inputting residual stresses into the model, to create a first static step in order to check the equilibrium condition. In order to then remove the initial out-of-balance forces Abaqus employs the following algorithm:

- The material initial stresses are imported from the ODB file or are either given by direct specification of the user, and will refer to the elements material point
- At each element's material point a further set of stresses are automatically created; these stresses have the same magnitude of the inputted ones, but opposite sign, therefore at this time the internal stress at each integration point is null all over the model, and the equilibrium is so guaranteed.
- Subsequently, during this first step, the artificial stresses created by the software are gradually decreased, until by the end of the step the static equilibrium is finally achieved.

After these steps, the analysis can proceed with any other kind of analysis step or procedure.

However, the accuracy of this technique for some kinds of applications may not be satisfactory, therefore in this thesis a python software was developed with the aim of put forward a method to furtherly improve the residual stresses importing in Abaqus.

Different versions of the software were developed, the more interesting will be afterwards illustrated, along with some simple application examples which was employed to test its effectiveness and potential worthiness in these kinds of analyses.

2.2 The main file logic

The program hereafter illustrated is thought to be run when dealing with an analysis in which the initial condition "initial stress" was specified from an output database file, that was the result of a previous analysis. Hence the model assembly and the mesh must be consistent amid the two analyses. More in detail, to use the initial condition "initial stress", Abaqus requires the ODB file, and the PRT file (extension ".prt") which are both results files.

The main file has to be run from the Abaqus GUI or either from the Abaqus PDE, which are respectively the graphic user interface and the Python development environment provided by the software suite.

In turn, the main file launches the other scripts with assigned other smaller tasks, the scheme of picture 2.1 depicts the program flow and logic in short.

The Python scripts employ many commercial libraries such as:

- time: this library allows to use some functions related to the execution time counting
- numpy: it is often indicated by "np" and it allows to perform several operations on arrays with ease
- math: it allows to use some mathematical functions
- string: general actions on strings
- os: miscellaneous operating system interfaces
- namedtuple: factory function for creating tuple subclasses with named fields
- inputfile: it allows to work with files and extract data



Figure 2.3: “Program components”.

It is then necessary to add some statements to gain access to some Abaqus resources and modules, like as the visualization module, or the ODB file, for instance:

- `from abaqus import *`
- `from abaqusConstants import *`
- `from odbAccess import *`
- `from odbMaterial import *`
- `from odbSection import *`

All these statements and library calls need to be provided only once, and before the execution of their related instructions. Hence, for convenience they were placed at the starting lines of the main file.

All the employed scripts will be hereafter described in detail.

2.2.1 The main file

```
from abaqus import *
from abaqusConstants import *
from odbAccess import *
from odbMaterial import *
from odbSection import *
import time
import numpy as np
import visualization
import fileinput
import os
import sys, getopt, os, string
import math
from collections import namedtuple
```

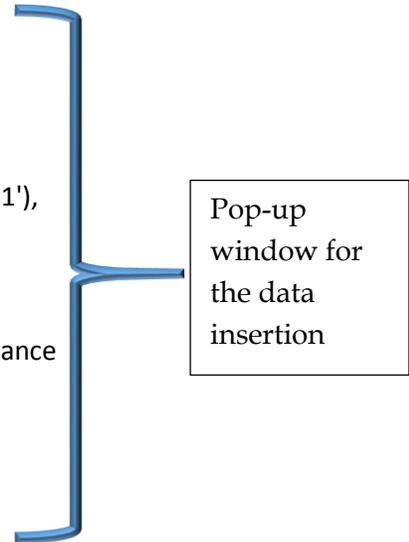


Import Python libraries and
Abaqus modules

```
input_data= getInputs( ('first analysis ODB path:', 'first_analysis.odb'),  
('last step name:', 'releasing'),  
('instance name:', ''), ('second analysis ODB path:', 'second_analysis.odb'),  
('last step name:', 'equilibrium_step'),  
('Model database name:', ''), ('job name:', 'second_analysis'), ('integral factor:', '1'),  
('number of iterations:', '4')),  
'Input required fields' )
```

```
MyStruct = namedtuple("MyStruct", "first_odb_path first_last_step first_instance  
sec_odb_path sec_last_step sec_MDB sec_job beta iterations")
```

```
inp=MyStruct(input_data[0],input_data[1],input_data[2],input_data[3],  
input_data[4],input_data[5],input_data[6],input_data[7],input_data[8])
```



```
start = time.time()
```

Start execution time recording here

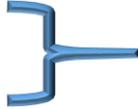
```
r=0  
flag1=0  
x=0  
flag3=0
```

Variables and flags

```
execfile('extract1.py')  
  
for x in xrange(int(inp.iterations)):  
    if flag1>0:  
        execfile('reload_data.py')  
        flag1=1  
  
    execfile('run_job.py')  
    execfile('extract2.py')  
    execfile('statistics.py')  
    execfile('PI_adjustment.py')  
    execfile('stepx.py')
```

Loop execution

```
end = time.time()  
dt=end-start
```



Final time

```
execfile('report.py')
```

2.2.2 The pop-up window for the data insertion

This window was thought to allow even an unexperienced Python programming language user to use this code with ease. With the given code, it is indeed necessary to specify for any analysis some model data from the first analysis from which take the residual stress and from the upcoming analysis as well.

As to the first analysis the program needs the ODB file path, or just its name if it is placed in the same folder of the scripts, the name of the last step, from which take the stress data and the assembly instance name.

With regard to the second analysis are similarly needed its ODB file path, its last analysis step name, its model database name or "MDB", and finally the job name.

Finally, are required two parameters which are only related to the iteration procedure performed on the residual stress importing. These parameters are the integral factor β of the adjustment equation and the number of iterations allowed for the script execution.

The choice for these last two parameters depends to some extent to the kind of model involved, therefore their optimum choice must be found by search and trial, in order to achieve better results in the least number of iterations.

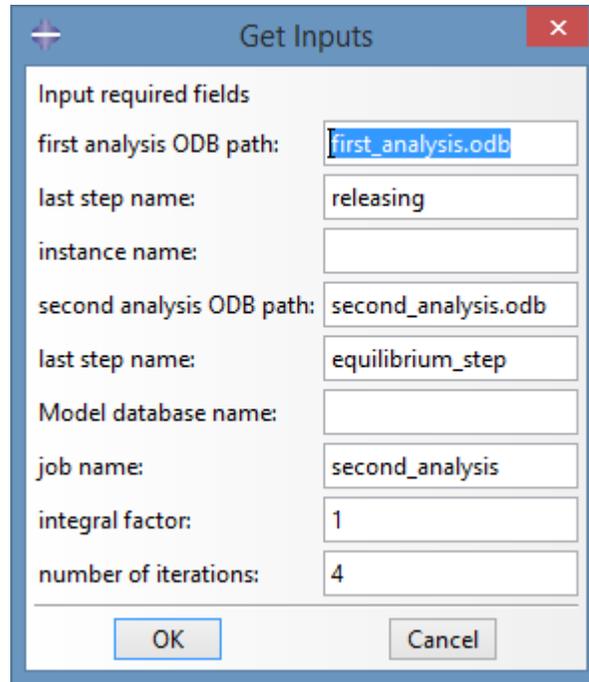


Figure 2.4: "Pop-up window for the inputs".

Some entries are already set by default as hint, but can be modified by the user.

2.2.3 extract1

```
odb = openOdb(path=inp.first_odb_path)
lastFrame=odb.steps[inp.first_last_step].frames[-1]
Stress=lastFrame.fieldOutputs['S']
ex1=np.empty((len(Stress.values),6), dtype=np.float32)
i=0
```

It opens the first analysis ODB file, and gets the last frame of the last step stress field, to be imported in the array "ex1"

```
for S in Stress.values:
    ex1[i][0]=S.data[0]
    ex1[i][1]=S.data[1]
    ex1[i][2]=S.data[2]
    ex1[i][3]=S.data[3]
    ex1[i][4]=S.data[4]
    ex1[i][5]=S.data[5]
```

By means of this loop the stress tensor of each mesh element is imported in "ex1", in such a way that each column holds one stress component per each element

i+=1
targ=ex1



The data held in "ex1" are copied in the array "targ" which will stay fixed in time to be compared with the obtained residual stresses at the end of the equilibrium step. The data held in "ex1" will be instead adjusted at each iteration cycle

```
import os  
myfile=inp.first_odb_path[0:(len(inp.first_odb_path)-4)]+'.lck'  
  
if os.path.isfile(myfile):  
    os.remove(myfile)
```

Every time that an ODB file is manipulated Abaqus creates a lock file, to avoid the file to be read/written concurrently by multiple applications.

It is then necessary to remove this file at each cycle to keep on reading/writing the file

```
del i, myfile, Stress,lastFrame  
odb.close()
```



Temporary variables and arrays are deleted to spare memory space.

Finally, the ODB file is closed

2.2.4 run_job

```
mdb=openMdb(inp.sec_MDB)
myJob=mdb.jobs[inp.sec_job]
myJob.submit()
myJob.waitForCompletion()
mdb.close()
```

It opens the second analysis model database (file with extension ".CAE", which holds the model, the mesh, the loads and the analysis job.

Then it submits the job, in this way creating the second analysis ODB file, which will have the same name given to the analysis job

```
##### delete lck file
import os

myfile=inp.sec_job+'.lck'

if os.path.isfile(myfile):
    os.remove(myfile)
```

It checks the presence of a lock file, and eventually deletes it

2.2.5 extract2

```
odb = openOdb(path=inp.sec_odb_path)
lastFrame=odb.steps[inp.sec_last_step].frames[-1]
Stress=lastFrame.fieldOutputs['S']
ex2=np.empty([len(Stress.values),6], dtype=np.float32)
i=0
```

Analogously to the script "extract1", it initially opens the second analysis ODB file, and gets the stress field output at the last frame of the last step, in order to copy it in the array "ex2" which is analogous to the array "ex1"

```
for S in Stress.values:
    ex2[i][0]=S.data[0]
    ex2[i][1]=S.data[1]
    ex2[i][2]=S.data[2]
    ex2[i][3]=S.data[3]
    ex2[i][4]=S.data[4]
    ex2[i][5]=S.data[5]
    i+=1
```

This loop allows to load the stress tensor in "ex2" per each mesh element

```
import os
myfile=inp.sec_odb_path[0:(len(inp.sec_odb_path)-4)]+'.lck'

if os.path.isfile(myfile):
    os.remove(myfile)
del lastFrame,Stress, i , myfile
odb.close()
```

It checks the presence of a lock file and if so it deletes it

```
del lastFrame,Stress, i , myfile
odb.close()
```

It deletes temporary variables and closes the file

2.2.6 statistics

```
### CALCULATE STATISTICS DATA
```

```
vm1=np.sqrt(0.5*((targ[:,0]-targ[:,1])**2+(targ[:,1]-targ[:,2])**2+
(targ[:,2]-targ[:,0])**2+6*(targ[:,3])**2+targ[:,4]**2+targ[:,5]**2)))
```

```
vm2=np.sqrt(0.5*((ex2[:,0]-ex2[:,1])**2+(ex2[:,1]-ex2[:,2])**2+
(ex2[:,2]-ex2[:,0])**2+6*(ex2[:,3])**2+ex2[:,4]**2+ex2[:,5]**2)))
```

The statistics calculated in this script take into account the Von Mises stress and exploits the data held in "ex2" and "targ", so first of all the Von Mises stresses of these two data sets are computed

```
if flag3==0:
    delta_max=[]
    delta_perc_max=[]
    delta_mean=[]
    delta_perc_mean=[]
    scostd=[]
    scostp=[]
    flag3=1
```

Initialization of the arrays which will hold the statistics, to be carried out at the first loop cycle only

```
delta_max.append(max(abs(vm1-vm2)))
delta_perc_max.append(max(abs(((vm1-vm2)*100/vm1)))

delta_mean.append(np.mean(abs(vm1-vm2)))
delta_perc_mean.append(np.mean(abs(((vm1-vm2)*100/vm1)))
```

Calculation of the maximum difference and mean difference between the inputted and obtained residual stresses both in absolute and percental terms

```
scostd.append(float((abs(vm1-vm2)>1000).sum()*100/len(ex1))
scostd.append(float((abs(vm1-vm2)>10000).sum()*100/len(ex1))
scostd.append(float((abs(vm1-vm2)>100000).sum()*100/len(ex1))
scostd.append(float((abs(vm1-vm2)>1E6).sum()*100/len(ex1))
scostd.append(float((abs(vm1-vm2)>10E6).sum()*100/len(ex1))
scostd.append(float((abs(vm1-vm2)>100E6).sum()*100/len(ex1))
scostd.append(float((abs(vm1-vm2)>200E6).sum()*100/len(ex1))
```

Calculation of the cumulative percentage of absolute differences between the input/output stresses

```
scostp.append(float(((abs(vm1-vm2)*100/vm1)>0).sum()*100/len(ex1))
scostp.append(float(((abs(vm1-vm2)*100/vm1)>15).sum()*100/len(ex1))
scostp.append(float(((abs(vm1-vm2)*100/vm1)>30).sum()*100/len(ex1))
scostp.append(float(((abs(vm1-vm2)*100/vm1)>50).sum()*100/len(ex1))
scostp.append(float(((abs(vm1-vm2)*100/vm1)>75).sum()*100/len(ex1))
scostp.append(float(((abs(vm1-vm2)*100/vm1)>100).sum()*100/len(ex1))
scostp.append(float(((abs(vm1-vm2)*100/vm1)>200).sum()*100/len(ex1))
```

Calculation of the cumulative percentage of the percental differences between the input/output stresses

```
del vm1, vm2
```

} Temporary variables deletion

2.2.7 Pi_adjustment

ADJUSTMENT OF THE INPUTTED STRESSES FROM THE FIRST ANALYSIS

```
beta=float(inp.beta)
```

} Integral factor, inputted by the user at the pop-up window

```
ex1=ex1+beta*(targ-ex2)
```

} It corresponds to the equation:

$$\sigma(x)_{inp}^{i+1} = \sigma(x)_{inp}^i + \beta(\sigma(x)_{targ} - \sigma(x)_{out}^i)$$

```
del ex2, beta
```

} Temporary variables

2.2.8 stepx

```
##### step name writer
```

```
r=r+1
```

```
a=str(r)
```

```
b='step'
```

```
step_name=b+a
```

```
del a, b
```

Simple script whose duty is to create per each loop cycle a new name for the added fictitious step, since they must be different

2.2.9 reload_data

```
### CREATE A FAKE STEP IN THE FIRST ANALYSIS ODB WITH THE RECOMPUTED INPUTTED STRESSES
```

```
odb = openOdb(path=inp.first_odb_path,readOnly=False)
```

```
step1 = odb.Step(name=step_name,description="", domain=TIME, timePeriod=1.0)
```

```
frame1 = step1.Frame(incrementNumber=1,frameValue=0.1, description="")
```

It opens the first analysis ODB file and creates a new fictitious step, progressively named thanks to the script "stepx" at each cycle

```
EL=range(1,len(ex1)+1)
```

```
instance1=odb.rootAssembly.instances[inp.first_instance]
```

```
stress_field = frame1.FieldOutput(name='S',  
description='stresses', type=Tensor_3D_Full)
```

```
stress_field.addData(position=INTEGRATION_POINT,  
instance=instance1, labels=EL, data=ex1)
```

It gets the assembly instance of interest and creates the stress field at the last frame of the last step.

Finally, it charges in the ODB the adjusted stresses held in "ex1"

```
odb.save()  
odb.close()
```

```
del step1,frame1,EL,odb,step_name
```



The ODB file has been modified, hence must be saved and closed.

Temporary variables deleted

2.2.10 report

```
### WRITE STATISTICS TO FILE
```

```
myfile = open('statistics_report.txt', 'w')
```

```
myfile.write(' MAX(MPa) MEAN(KPa) MAX% MEAN%\n')
```

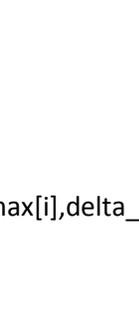


It creates a text file for writing the data calculated by the script "statistics"

```
for i in range(len(delta_max)):
```

```
    myfile.write('%6.3f %6.3f %6.3f  
    %6.3f\n'%(delta_max[i]/1E6,delta_mean[i]/1000,delta_perc_max[i],delta_perc_mean[i]))
```

```
myfile.write('\n\n')
```



This loop writes to the file the maximum and mean stress differences collected at each cycle

```
for i in range(0,len(scostd),7):
```

This other loop writes the percent cumulative difference amid the inputted and output stresses

```
myfile.write('scostamento%%>0: %6.3f%%\t\tdelta_sigma>1kPa:
%6.3f%%\n'%(scostp[i],scostd[i]))
myfile.write('scostamento%%>15: %6.3f%%\t\tdelta_sigma>10kPa:
%6.3f%%\n'%(scostp[i+1],scostd[i+1]))
myfile.write('scostamento%%>30: %6.3f%%\t\tdelta_sigma>100kPa:
%6.3f%%\n'%(scostp[i+2],scostd[i+2]))
myfile.write('scostamento%%>50: %6.3f%%\t\tdelta_sigma>1MPa:
%6.3f%%\n'%(scostp[i+3],scostd[i+3]))
myfile.write('scostamento%%>75: %6.3f%%\t\tdelta_sigma>10MPa:
%6.3f%%\n'%(scostp[i+4],scostd[i+4]))
myfile.write('scostamento%%>100: %6.3f%%\t\tdelta_sigma>100MPa:
%6.3f%%\n'%(scostp[i+5],scostd[i+5]))
myfile.write('scostamento%%>200: %6.3f%%\t\tdelta_sigma>200MPa:
%6.3f%%\n\n\n'%(scostp[i+6],scostd[i+6]))
```

```
myfile.write('\n\n\n')
```

```
myfile.write('Execution time: %8ds %8dmin'%(dt,dt/60))
```



It writes the execution time recording

```
myfile.close()
```

```
del delta_max, delta_perc_max, delta_mean,
delta_perc_mean, scostd, scostp, flag3, dt
```

File closure and temporary variables deletion

2.2.11 The calculations of "statistics.py" and "report.py"

In this section is presented a quick explanation of the statistics calculations performed to assess to program effectiveness according to the set parameters (integral factor β and number of iterations) as well as the model tested.

The stress components relative to each mesh element are held in a field output object in a tuple as "data= (S11, S22, S33, S12, S13, S23)".

Hence, as first is needed to calculate the Von Mises stress as follows:

$$\sigma_{vm} = \sqrt{\left(\frac{1}{2} * ((\sigma_{11} - \sigma_{22})^2 + (\sigma_{11} - \sigma_{33})^2 + (\sigma_{22} - \sigma_{33})^2 + 6 * (\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{13}^2))\right)}$$

Afterwards, all the calculations will be carried out on such stress component, and will be indicated simply as σ for convenience.

The Von Mises stress is calculated for any element in the model and collected in the vectors "vm1" and "vm2", whereby in the first vector are held the data read from the first analysis ODB file, that is actually the target the program tries to get closer to, whilst the second holds the data relative to the ODB file of the second analysis, after the equilibrium step. In the list called "delta_max", at each iteration cycle, the maximum absolute difference between the data of "vm1" and "vm2" is saved, whereas in the vector "delta_perc_max" is saved the maximum percent difference, namely:

$$\Delta\sigma_{max} = \max\left(\text{abs}(\sigma_i^{vm1} - \sigma_i^{vm2})\right)$$

$$\Delta\sigma_{\%max} = \max\left(\text{abs}(\sigma_i^{vm1} - \sigma_i^{vm2}) * \frac{100}{\sigma_i^{vm1}}\right)$$

In addition, the mean value of the absolute and percent differences was calculated and saved in the lists "delta_mean" and "delta_perc_mean" respectively.

$$\Delta\sigma_{mean} = mean\left(abs(\sigma_i^{vm1} - \sigma_i^{vm2})\right)$$

$$\Delta\sigma_{mean} = mean\left(abs(\sigma_i^{vm1} - \sigma_i^{vm2}) * \frac{100}{\sigma_i^{vm1}}\right)$$

Hence, thanks to these data it is already possible to monitor the extent of the improvement in terms of maximum and mean stress field distortion. Moreover, a further control was added to help concentrating on the most critical distortions both in absolute or percent terms.

Namely, in the vectors called "scostd" and "scostp", was made a count of the elements in which the stress difference was greater than a certain amount, in order to have an indication of the amount of severe differences, since in many applications a gap of only few MPa would be irrelevant.

The algorithm employed can be summarized as follows:

$$\sum (abs(\sigma_i^{vm1} - \sigma_i^{vm2}) > x) * \frac{100}{len(vm1)}$$

$$\sum (abs(\sigma_i^{vm1} - \sigma_i^{vm2}) * \frac{100}{vm1} > x) * \frac{100}{len(vm1)}$$

Where "x" stands for a value to be compared, by default set to values from 1 kPa to 200 Mpa and in percent from 0% to 200%. The optimum range should be decided upon the magnitude of the RS.

2.3 The program testing

The Python program described in the previous sections was tested by means of some simple analysis examples. The number of elements of the selected models was set to a very low amount with the aim to reduce to few minutes the analysis time, allowing to make several tests with different parameters combinations and models.

The tests consisted in a first analysis, from which to extract the residual stress data, and a second one, on which the residual stresses were given as initial condition by the software.

The first analysis was composed by a first loading step, whereby a load or displacement sufficient to produce some plastic deformation on the model was applied. Then a second step was created wherein only the boundaries conditions were kept and the deforming loads were instead removed.

In this way at the end of the analysis the plastic residual stresses were the only still on the specimen model.

By running this analysis, the first analysis ODB file was created along with its part file (extension "prt") with all the necessary information for the subsequent analysis.

The second analysis consisted in only a static step, called equilibrium step, whereby only the boundary conditions and the initial condition "initial stress" were set up.

The employed model was the same for both the analyses in terms of geometry, material and mesh.

It is indeed very important that the two meshes are identical for the correct program execution.

Finally, the Python main script was executed from Abaqus which displayed the input parameters windows and automatically carried out all the remaining tasks.

Thanks to the text file "statistics_report.txt" generated by the script "report.py" it is then possible to check the achieved accuracy improvement in the internal stress importing and decide whether is necessary or not to change the number of iterations or the integral factor β .

After ensuring a satisfactory internal stress importing accuracy, any subsequent analysis step can be added to the second analysis after the equilibrium step and so the work submission can proceed normally.

2.3.1 First example: the CTS specimen

The specimen adopted for this test example is a compact tension specimen, whose dimension are reported as meters in the sketch of figure 2.3.

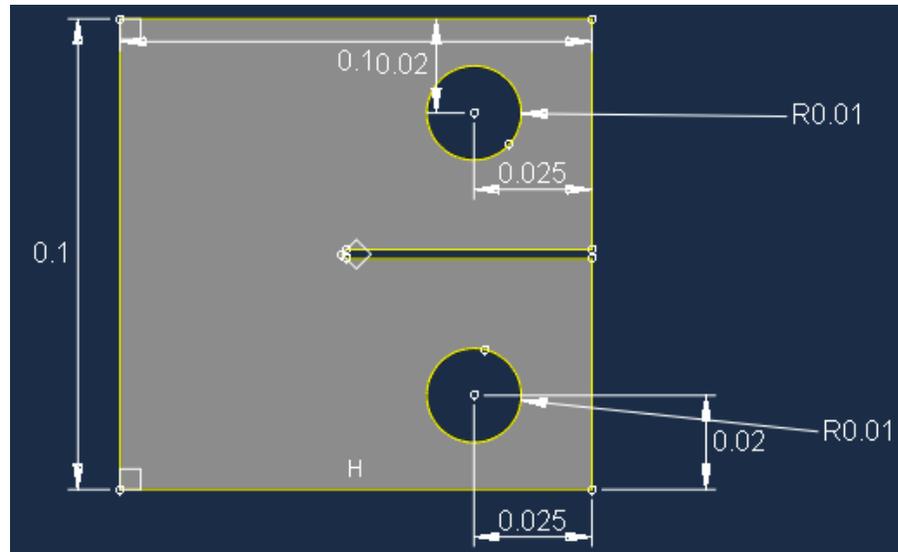


Figure 2.5: "CTS specimen sketch".

Where its depth is of 0,5 cm and the material chosen is a mild steel with elastic modulus $E=270$ GPa and Poisson coefficient $\nu=0,3$. Since also plastic deformations were involved was necessary to define the plastic behaviour of the steel, the chosen values are collected in table 2.1:

	Yield stress (MPa)	Plastic strain
1	200	0
2	246	0,02352
3	294	0,04739
4	374	0,09353
5	437	0,1377
6	480	0,1800

Table 2.1: "Plastic behaviour of the selected steel"

The boundary conditions consist in an encastre on the vertical face of length 10 cm of figure 2.3, and in a Z-direction constraint (orthogonal to the drawing and in the upward direction).

These two conditions are held for the entire analysis time, in contrast with the loads which was given as material displacements and was set inactive in the last step, called "releasing".

The loads consist in displacements imposed to the two holes in upward and downward direction, as clarified by picture 2.4.

The applied displacement is of 1 mm for both the holes, and as suggested by the Abaqus user manual, since plastic deformations are produced the option "Nlgeom" was set to include the effects of large deformations.

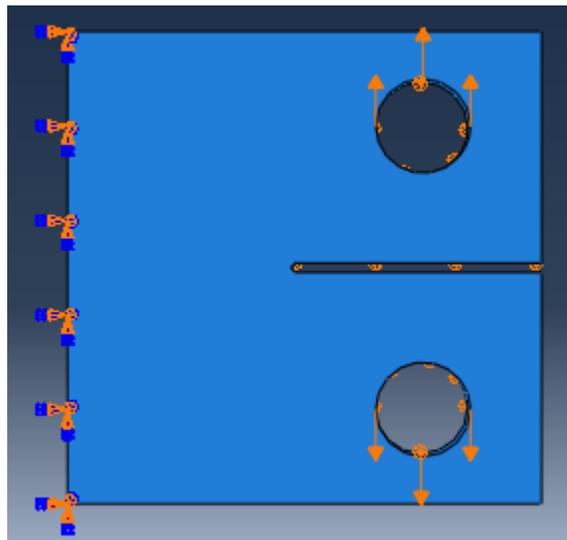


Figure 2.6: "Boundary conditions and loads".

Finally, the mesh was made out of 8481 elements, and 11940 nodes. The chosen kind of element is identified by C3D8R which stands for hexahedral linear elements.

In figure 2.5 are shown the Von Mises stresses over the specimen after releasing the load, therefore consist of residual internal stresses.

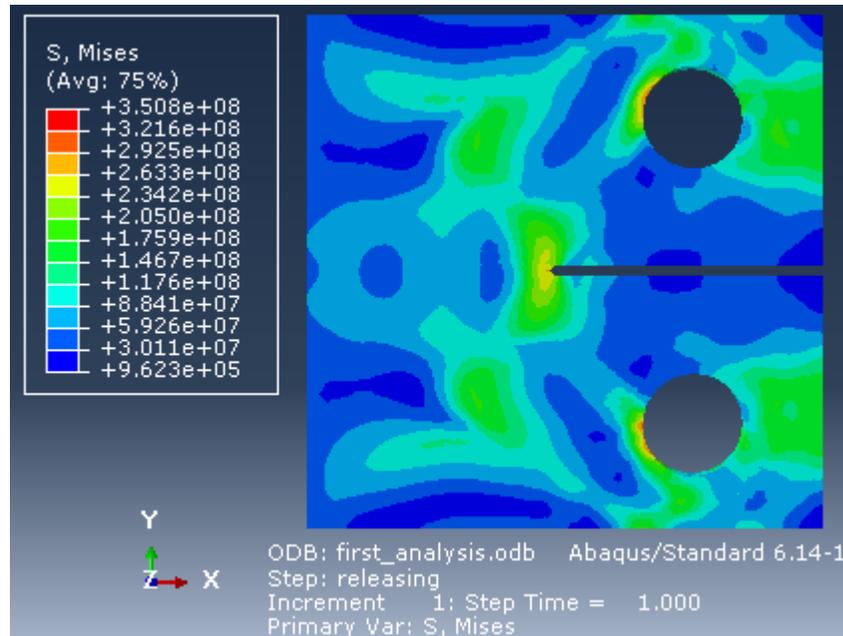
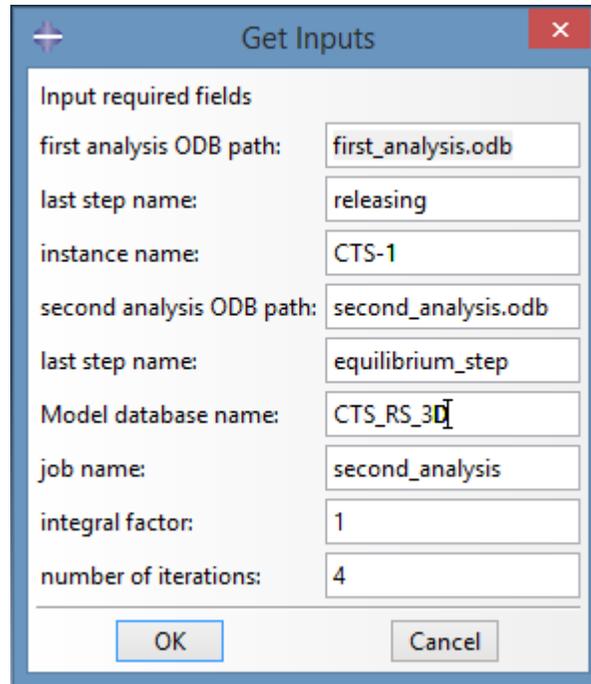


Figure 2.7: "Residual stress after releasing the load (the data are in Pa)".

As the first analysis was completed its ODB and PRT files are available, hence is now possible to start a second analysis in which we want to import the residual stress data as initial condition and then to simulate a further set of loading steps of any kind.

With the aim of uniquely test the program, in the second analysis only one step was created, called "equilibrium step", in order to then compare the internal stresses given as input held in the first ODB file and the ones obtained after the importing procedure and the equilibrium step held in the second ODB file.

In order to test the program, now the file "main.py" is run, and the pop-up window which appears immediately is filled in as follows:



Input required fields	
first analysis ODB path:	first_analysis.odb
last step name:	releasing
instance name:	CTS-1
second analysis ODB path:	second_analysis.odb
last step name:	equilibrium_step
Model database name:	CTS_RS_3D
job name:	second_analysis
integral factor:	1
number of iterations:	4

Figure 2.8: "Filling in the input window".

Hence, were set four iteration steps and the integral factor to $\beta=1$. This means that the Python script executes the job for four times adjusting each time the inputted internal stresses.

The script called "report.py" prints the statistics parameters in the file "statistic report.txt", which is reported hereafter:

MAX(MPa)	MEAN(KPa)	MAX%	MEAN%
149.406	3456.138	94.470	4.500
148.062	3241.253	90.356	4.159
146.742	3097.240	86.163	3.945
145.439	2983.142	82.192	3.777

scostamento%>0: 100.000%	delta_sigma>1kPa: 99.988%
scostamento%>15: 4.622%	delta_sigma>10kPa: 99.811%
scostamento%>30: 1.368%	delta_sigma>100kPa: 97.406%
scostamento%>50: 0.307%	delta_sigma>1MPa: 73.541%
scostamento%>75: 0.083%	delta_sigma>10MPa: 4.964%
scostamento%>100: 0.000%	delta_sigma>100MPa: 0.189%
scostamento%>200: 0.000%	delta_sigma>200MPa: 0.000%

scostamento%>0: 100.000%	delta_sigma>1kPa: 99.965%
scostamento%>15: 3.997%	delta_sigma>10kPa: 99.682%
scostamento%>30: 1.191%	delta_sigma>100kPa: 96.958%
scostamento%>50: 0.153%	delta_sigma>1MPa: 72.562%
scostamento%>75: 0.047%	delta_sigma>10MPa: 4.292%
scostamento%>100: 0.000%	delta_sigma>100MPa: 0.177%
scostamento%>200: 0.000%	delta_sigma>200MPa: 0.000%

scostamento%>0: 100.000%	delta_sigma>1kPa: 99.976%
scostamento%>15: 3.584%	delta_sigma>10kPa: 99.705%
scostamento%>30: 0.931%	delta_sigma>100kPa: 97.382%
scostamento%>50: 0.130%	delta_sigma>1MPa: 72.232%
scostamento%>75: 0.047%	delta_sigma>10MPa: 3.938%
scostamento%>100: 0.000%	delta_sigma>100MPa: 0.153%
scostamento%>200: 0.000%	delta_sigma>200MPa: 0.000%

scostamento%>0: 100.000%	delta_sigma>1kPa: 99.988%
scostamento%>15: 3.231%	delta_sigma>10kPa: 99.776%
scostamento%>30: 0.837%	delta_sigma>100kPa: 97.371%
scostamento%>50: 0.106%	delta_sigma>1MPa: 71.336%
scostamento%>75: 0.035%	delta_sigma>10MPa: 3.726%
scostamento%>100: 0.000%	delta_sigma>100MPa: 0.141%
scostamento%>200: 0.000%	delta_sigma>200MPa: 0.000%

Execution time: 575s 9min

In the first group of data, labelled as "MAX(Mpa) MEAN(Mpa) MAX% MEAN%", each row refers to an iteration, after this first group of data other four sets labelled as "scostamento%" and "delta_sigma" were written; each set of data refers to a different iteration step, indeed four iterations were selected in the initial pop-up window.

Finally, also the execution time is reported both in seconds and in minutes (rounded up).

Hence, after four iteration steps the absolute stress difference dropped from about 3,456 MPa to 2,983 MPa and in percent terms from 4,5% to 3,777% and so forth. In figure 2.7 the Von Mises stress field given as input, the one obtained after the equilibrium step without the intervention of the Python program, and after four steps of iterations are compared.

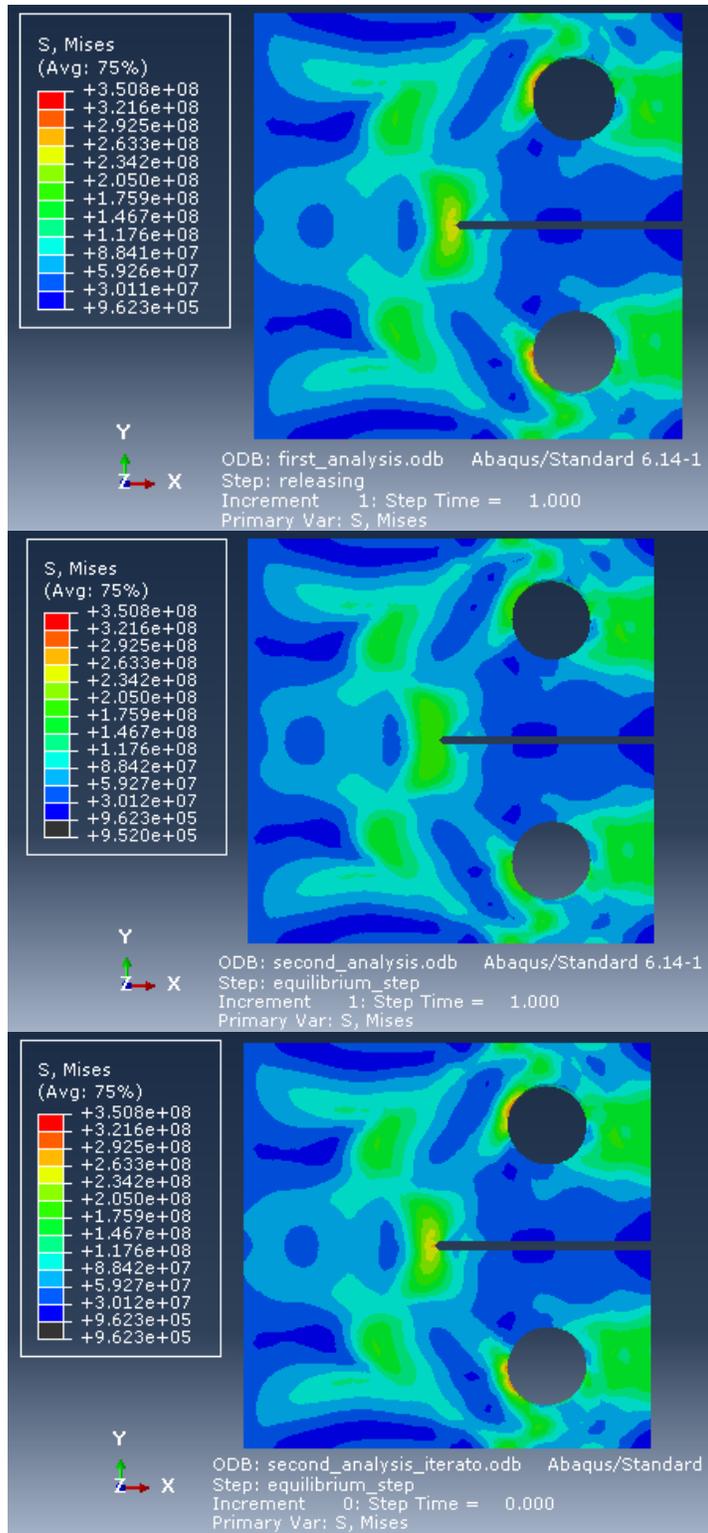


Figure 2.9: "Results comparison".

2.3.2 Second example: bent 3D beam

The second example is a cantilever beam loaded at one of its ends by means of a non-null displacement boundary condition. The material properties assigned to the model are the same used for the compact test specimen of the previous example.

In figure 2.8 are illustrated the sketch dimensions in meters.

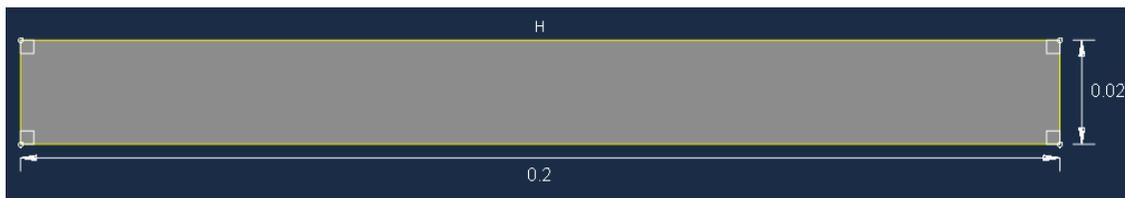


Figure 2.10: "Cantilever beam sketch".

The depth of the beam is of 0,5 cm, and as to the boundary conditions at one end face is applied an encastre condition, and at the other a displacement of X mm in the Z direction, as depicted in figure 2.9:

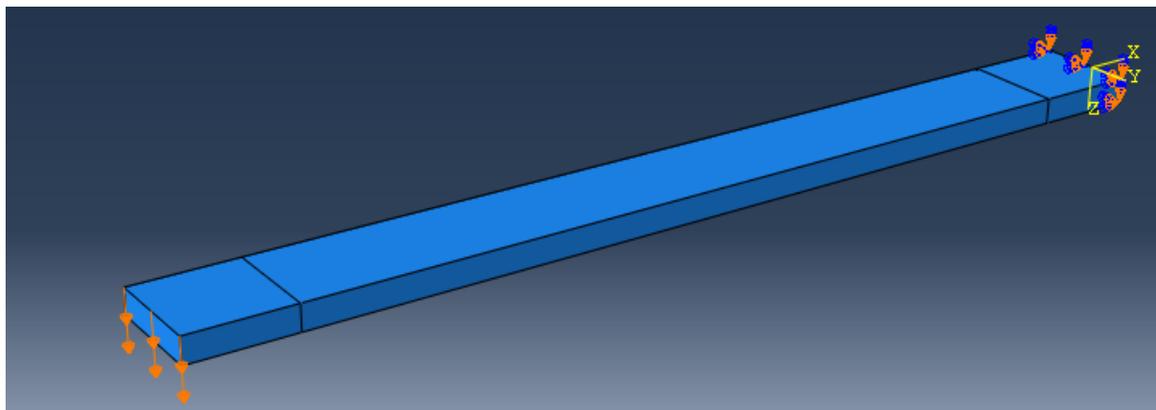


Figure 2.11: "Boundary conditions".

Also in this case, when executing the Python scripts, at the input pop-up window was selected an integral factor $\beta=1$ and four iterations; in the following, the "statistic_report.txt" file is reported to analyse the results:

MAX(MPa)	MEAN(KPa)	MAX%	MEAN%
1.612	48.071	485.391	3.140
1.423	44.286	268.923	2.522
1.279	42.028	275.494	2.372
1.162	40.429	251.292	2.307

scostamento%>0: 100.000%	delta_sigma>1kPa: 59.921%
scostamento%>15: 3.172%	delta_sigma>10kPa: 39.820%
scostamento%>30: 0.989%	delta_sigma>100kPa: 14.581%
scostamento%>50: 0.356%	delta_sigma>1MPa: 0.112%
scostamento%>75: 0.213%	delta_sigma>10MPa: 0.000%
scostamento%>100: 0.127%	delta_sigma>100MPa: 0.000%
scostamento%>200: 0.045%	delta_sigma>200MPa: 0.000%

scostamento%>0: 100.000%	delta_sigma>1kPa: 58.704%
scostamento%>15: 1.921%	delta_sigma>10kPa: 40.015%
scostamento%>30: 0.315%	delta_sigma>100kPa: 13.738%
scostamento%>50: 0.097%	delta_sigma>1MPa: 0.097%
scostamento%>75: 0.049%	delta_sigma>10MPa: 0.000%
scostamento%>100: 0.041%	delta_sigma>100MPa: 0.000%
scostamento%>200: 0.011%	delta_sigma>200MPa: 0.000%

scostamento%>0: 100.000%	delta_sigma>1kPa: 57.966%
scostamento%>15: 1.708%	delta_sigma>10kPa: 40.000%
scostamento%>30: 0.292%	delta_sigma>100kPa: 13.352%
scostamento%>50: 0.090%	delta_sigma>1MPa: 0.082%
scostamento%>75: 0.049%	delta_sigma>10MPa: 0.000%
scostamento%>100: 0.034%	delta_sigma>100MPa: 0.000%
scostamento%>200: 0.015%	delta_sigma>200MPa: 0.000%

scostamento%>0: 100.000%	delta_sigma>1kPa: 57.753%
scostamento%>15: 1.483%	delta_sigma>10kPa: 39.831%
scostamento%>30: 0.225%	delta_sigma>100kPa: 12.861%
scostamento%>50: 0.090%	delta_sigma>1MPa: 0.067%
scostamento%>75: 0.049%	delta_sigma>10MPa: 0.000%
scostamento%>100: 0.026%	delta_sigma>100MPa: 0.000%
scostamento%>200: 0.007%	delta_sigma>200MPa: 0.000%

Execution time: 3211s 53min

Also in this case is noticeable a progressive improvement of the importing procedure at each iteration step both in terms of mean and maximum stress difference.

In this model the mesh features 26700 elements and 32578 nodes, and the kind of elements selected is the C3D8R, which stands for 8-node linear brick, reduced integration elements. In analogy to the first example, also in this case a first analysis was performed in order to produce plastic deformations in the part, which in the last step was released from

the applied load in such a way to keep at the end of the analysis only the residual internal stresses. Subsequently a second analysis was created, which started with the initial condition "input stress" to be read from the previously produced ODB file.

In figure 2.10 the deformed shape model:

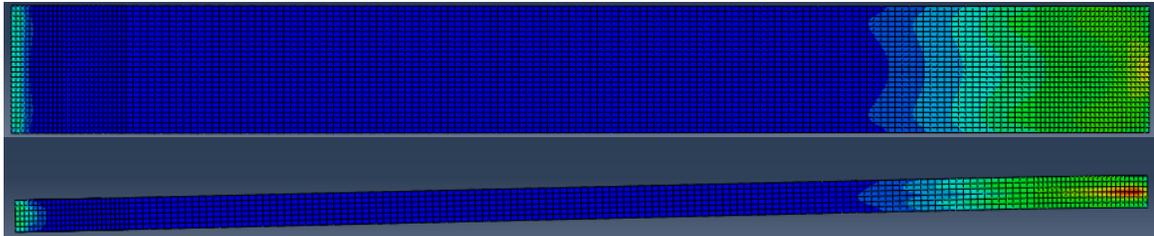


Figure 2.12: "Deformed shape cantilever beam".

In this case is not possible to notice the difference at a first sight (fig.2.11) but is anyway possible to assess the program effectiveness from the report statistics.

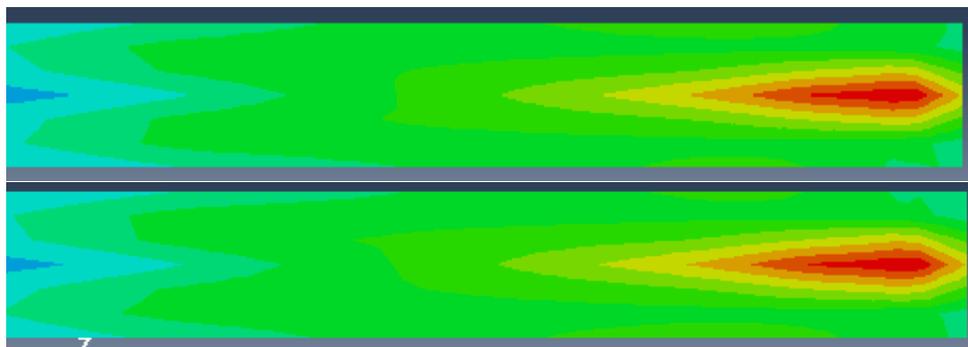


Figure 2.13: "Comparison between the Von Mises stress given as initial condition and obtained after the equilibrium step; is not possible in this case to see the difference".

2.4 Further enhancements

2.4.1 Input by text file version

The developed program is quite flexible and can be easily modified, adding or substituting subprograms. The most straight forward enhancement possible to the presented code is to make it able to read the stress input data from a text file instead of an Abaqus ODB file, making then possible to read data from any other FEM software.

Let suppose to have a text file whereby the stress tensor data per each element are listed in order as:

```
8.285606E+07 -2.837972E+06 -2.051437E+06 9.665228E+06 -1.037617E+07 2.252248E+06
```

Hence, only separated by a tab character "\t", it is then possible to extract such data and to copy them in an array for further calculations:

```
f=open('myfile.txt','r')      #insert input file name
sigma_targ=f.read()
sigma_targ=sigma_targ.split('\t')
sigma_targ.pop()
a=len(sigma_targ)/6+1
sigma_targ = np.reshape(sigma_targ,(a-1,6))
sigma_targ=np.array(sigma_targ, dtype=np.float32)
f.close()
```

It is therefore possible through Python to easily read and write the stress tensors from Abaqus to text files and conversely. For example, in the first phases of the program development the tests were performed on simpler model, then it was possible to manually

compute statistics by means of excel directly. In order to write the data to an excel workbook

the following script was employed:

```
#DATA CONVERTER

import xlrd

import xlwt

from xlutils.copy import copy

import os.path

import xlswriter

data = []

with open("first_analysis_extraction.txt") as f:

    for line in f:

        data.append([word for word in line.split("\t") if word])

wb = xlswriter.Workbook('first_analysis_extraction.xlsx')

sheet = wb.add_worksheet("sheet_1")

iterrange=iter(range(len(data)))

next(iterrange)

for row_index in iterrange:

    for col_index in range(len(data[row_index])):

        v=float(data[row_index][col_index])

        sheet.write_number(row_index, col_index,v)

    for first_col_index in range(len(data[row_index])):

        sheet.write(0,first_col_index,data[0][first_col_index])

wb.close()
```

Another solution investigated was the possibility to read as initial stress conditions the data relative only to a small portion of the model. This is motivated by a potential application that is to measure (estimate) the internal stresses on a real specimen by means of strain gauges such as rosettes gauges. It would be then interesting to be able to insert these measurements on the FEA problem. However, when dealing with mechanical equilibrium this was proven to be not feasible in most cases, since the inputted stresses in a small region are pretty prone to smooth out, all over the model. Even the use of the developed program does not carry satisfactory results.

However, this version of the program will be herein briefly reported in case of future enhancement which may lead to appreciable results.

2.4.2 Input on sets version

This version is thought to allow the user to insert internal stresses as initial conditions which do not derive from a previous FEM analysis but instead from a measurement. Hence, to be used the user must already be provided with the geometric model, the mesh, and the boundary conditions.

By means of the Abaqus GUI (graphic user interface) is possible to select and create an element set. It is then possible to assign a certain stress tensor to all the selected elements in the set as initial condition. Anyways, with this simple manual method was not possible to implement the correction algorithm used in the script "PI_adjustment".

Hence, the code was modified in order to create a fake first analysis for the residual stress evaluation to be inputted in the follow-up analysis.

In order to better clarify this procedure, the modified main file is presented:

```
import time
start = time.time()
r=0
flag1=0
x=0
execfile('fictious_odb.py')
for x in xrange(4):
    if flag1>0:
        execfile('reload_data_extra.py') ###load ex1 in the ODB
    flag1=1
    execfile('run_job.py')
    execfile('extract2_extra.py')
    execfile('PI_adjustment_set.py')
    execfile('stepx_extra.py')
end = time.time()
dt=end-start
```

Hence, as first a new script called "fictious_ODB.py" was created; this script opens the model database and gets the number of mesh elements, the number of selected sets by the user and their elements.

Then it creates a fictious job to be executed, in this way a fictious ODB file and PRT file is created, with the same name assigned to the job. Subsequently the stress filed is added

to this ODB, in which all the stresses are set to zero, except the ones held in the sets selected by the user. In these the stress tensor is assigned directly into the code, but may be inserted by the user in a pop-up window potentially.

In the following the script is reported, accordingly to what was the duty of the script "extract1.py" also in this case the aim is to create the arrays "ex1" and "targ".

```
mdb=openMdb('cantilever_beam_RS.cae')
k=mdb.models['cantilever_beam'].rootAssembly.instances['beam-1'].elements
mesh_elements=len(k)
stress=np.zeros([mesh_elements,6], dtype=np.float32)
nsets=len(mdb.models['cantilever_beam'].rootAssembly.sets)
sets=np.empty([nsets],dtype=tuple)
h= mdb.models['cantilever_beam'].rootAssembly.sets
H=[]
T=[]
for i in range(nsets):
    sets[i]='Set-'+str(i+1)

for i in range(nsets):
    for j in range(len(h[sets[i]].elements)):
        H.append(h[sets[i]].elements[j].label)
        T=str(h[sets[i]].elements[j].type)

H=np.array(H,dtype=int) #array with all the element labels to be imported
```

```
#### SET STRESS VALUE FOR THE SETS
```

```
ST=(0,0,3E5,0,0,0)           #desired stress tensor for the set--300 KPa, 0.3MPa
```

```
stress[H-1]=ST              #array to be inserted into the fictious ODB
```

```
#create the fictious ODB and prt file
```

```
#create a fictious job in the current mdb
```

```
step1 = mdb.models['cantilever_beam'].StaticStep(name='fictious_step',previous='Initial',  
timePeriod=1.0)
```

```
job1=mdb.Job(name='first_analysis',model='cantilever_beam')
```

```
job1.submit()
```

```
job1.waitForCompletion()      #this will create a fictious .prt and ODB file
```

```
#add stress fieldOutput to ODB
```

```
odb = openOdb(path='first_analysis.odb',readOnly=False)
```

```
step1 = odb.Step(name='first_step',description="", domain=TIME, timePeriod=1.0)
```

```
frame1 = step1.Frame(incrementNumber=1,frameValue=0.1, description="")
```

```
EL=range(1,len(stress)+1)
```

```
instance1=odb.rootAssembly.instances['BEAM-1'] #insert instance name
```

```
stress_field = frame1.FieldOutput(name='S', description='stresses', type=TENSOR_3D_FULL)
```

```
stress_field.addData(position=INTEGRATION_POINT, instance=instance1, labels=EL, data=stress)
```

```
#delete fictious step and job
```

```
del mdb.models['cantilever_beam'].steps['fictious_step']
```

```
del mdb.jobs['first_analysis']
```

```
#create equilibrium step and second analysis job

step1 = mdb.models['cantilever_beam'].StaticStep(name='equilibrium_step',previous='Initial',
timePeriod=1.0)

job1=mdb.Job(name='second_analysis',model='cantilever_beam')

#create input stress command

mdb.models['cantilever_beam'].Stress(name='stress_input',distributionType=FROM_FILE,
fileName='first_analysis', step=-1, increment=-1)

ex1=stress
targ=stress
odb.save()
odb.close()
mdb.save()
mdb.close()
```

The other scripts employed in this version are equivalent to the ones presented beforehand, hence they will not be treated in this section.

This program version is then able to allow the user to prescribe any stress tensor to an element or a selected set in the discretized model. In addition, it exploits the proportional adjustment described before to get closer to the aimed result after the equilibrium step.

However, after some trials it was proven that the adjustment algorithm is not effective in this situation, even employing different values for the integral factor β , then a different approach must be developed. Indeed, even after several iterations the desired inputted internal stresses cannot be obtained, when inputted only on small areas or when strongly unbalanced.

2.4.3 Lean version

This version is very close to the original one, and was developed in order to make the iteration procedure leaner, in case of very large models (very high number of elements and nodes), or in case of high number of iterations expected.

In the original version, discussed in section 2.2, the script called "reload_data", had the duty to upload on the ODB file at each iteration step the modified stress tensors. In such a way, after some loops, the size of the ODB file can grow remarkably, especially in case of large FE models.

However, during the testing and development phases this issue was not accounted as negative, since the size of the models was irrelevant, and moreover the data stored at each loop were useful for testing and comparison purposes.

In this new version the script "reload_data.py" was replaced by a new dedicated script called "new_input_ODB".

The script is hereafter reported:

```
### DELETE OLD ODB
if flag4==1:
import os
myfile=inp.first_odb_path
if os.path.isfile(myfile):
    os.remove(myfile)
flag4==1
```

```
### CREATE AN ORIGINAL ODB COPY
```

```
if flag5==0:
```

```
    copyfile(inp.first_odb_path,'temp.odb')
```

```
    flag5==1
```

```
### REPLACE THE COPY TO BE MODIFIED
```

```
if flag2==1:
```

```
    copyfile('temp.odb',inp.first_odb_path)
```

```
    flag2==1
```

```
### MODIFY ODB FILE
```

```
odb = openOdb(path=inp.first_odb_path,readOnly=False)
```

```
step1 = odb.Step(name=step_name,description="", domain=TIME, timePeriod=1.0)
```

```
frame1 = step1.Frame(incrementNumber=1,frameValue=0.1, description="")
```

```
EL=range(1,len(ex1)+1)
```

```
instance1=odb.rootAssembly.instances[inp.first_instance]
```

```
stress_field = frame1.FieldOutput(name='S', description='stresses', type=TENSOR_3D_FULL)
```

```
stress_field.addData(position=INTEGRATION_POINT, instance=instance1, labels=EL, data=ex1)
```

```
odb.save()
```

```
odb.close()
```

```
del step1,frame1,EL,odb,step_name
```

Above all, the program creates a copy of the ODB file with the stresses to be given in input. The copy is saved as "temp.odb", and this operation occurs only at the first loop cycle. Then the original ODB file is modified as it occurred running the script "reload_data", hence the stress tensors modified are uploaded in a newly created fictitious step; finally, the file is saved and closed.

From the second loop cycle on, the program will delete the original ODB file (at the second run) or either the last used modified ODB file (from the third run).

Therefore, at each cycle the program will copy the file saved beforehand as "temp.odb", rename it as it was the original file and proceed normally with its data modification.

In this way it is possible to save on the file only the original stress tensors and their last update, enabling a consistent memory saving. In turn, all the other intermediate data will be lost, apart from the statistics generated in the file "statistics_report.txt".

Hence, depending on the kind of application, that is either testing the program effectiveness or its normal use, the user must decide which one of the two versions to run.

Conclusions

In this master thesis a deep study on the relationship between the residual stresses in manufacturing and the mechanical fatigue was carried out.

The most relevant works in literature were gathered and analysed in order to gain a deep understanding on the fatigue life simulation issue in presence of stress fields. It was found that the majority of the researchers pursued the fracture mechanics approach, since the crack growth rate can be related to a Paris-like power law, able to account to the physical phenomenon interesting the fatigue crack growth in a stress field, such as the crack closure effect (induced by compressive fields) and the internal stress redistribution, which has been the main challenge in the past for the approaches relying on weight function methods for the computation of the stress intensity factors, but in recent times thanks to the work of Garcia et al. and Roberts et al., was demonstrated that by means of the FEM is actually possible to tackle this issue.

In addition, the more relevant works were found to share some few common steps in the simulation procedure, which have been identified in section 1.3 as the model geometry

realization, the application of the residual stress field, the definition of the parameters and the empirical law to be taken into account for the fatigue life estimate.

For each of these steps, the encountered alternatives were presented, along with their difficulties and limits.

The second part of the work dealt with the first of these issues, that is to correctly create a self-equilibrated stress field into a FE model. The target of the Python program was to solve or to fade the problem of the equilibrium step during the analysis. It was indeed encountered that if the inputted field is not perfectly self-equilibrated in the model geometry, it might be difficult to achieve an equilibrium or it may yield to unsatisfactory results in the importing procedure.

More versions of the Python program were developed, the original one is deeply depicted in section 2.2, whereas the others are outlined in section 2.4.

The performance of the program was assessed by means of two simple FE models, which are described in section 2.3. The first example is a 3D compact test specimen or CTS, made out by 8481 elements; in this case after the equilibrium step the average error in the importing procedure was of 4,5% or 3456 Kpa in absolute terms, whereas after 4 iteration steps performed by the developed software it decreased to 3,777% and 2983 Kpa.

The second example was a simple cantilever beam, made out of 26700 elements, and had an initial error in the stress field definition of 3,14% or 48,07 Kpa, whereas after four iteration of the program featured an error of 2,307% or 40,429 Kpa.

Further details about the software performances are presented in the chapter 2, along with the method through which they have been assessed.

The developed program features a robust base, that allows different and new subscripts to be added, removed or modified. Therefore, the software is optimized for potential further enhancements.

In the last section three different versions are outlined; in the base version the stress field is inputted by an Abaqus ODB file, then is assumed that such field was computed through Abaqus. However, a second version allows to read the tensors directly from a text file, thus allowing the user to perform the stress field calculation with any other kind of software. Obviously the software would be modified according to the input text file formatting.

Another version was created to make the program run leaner, in terms of computer resources. Since this program was thought to work with a FCA Automobiles' product model, counting millions of elements, an enhancement of this kind was necessary.

Finally, with the aim of cover a different application, was attempted to develop a version which was supposed to allow the user to input any kind of stress field obtained by an experimental measurement into a small portion of the FE model, but unfortunately in this case the approach was proven to not be effective. Further studies need to be performed in finding an alternative approach to cover this application.

Hence, the next step of this research will potentially be to still improve the approach developed in this work and to create a more comprehensive and versatile tool for the

residual stress fields creation in FE models, in such a way to have an optimal basis for the follow up steps in the determination of a method for the fatigue life assessment under these conditions.

Bibliography

- [1] E. Brusa, slides of the lectures of "Machine design", Politecnico di Torino, DIMEAS, version 2017.
- [2] Ralph I. Stephens, Ali Fatemi, Robert R. Stephens, Henry O. Fuchs, "Metal fatigue in engineering", 2000.
- [3] Available on Wikipedia.
- [4] L. Goglio, "Fondamenti di meccanica strutturale", 2014.
- [5] P.J.G. Schreurs, "Fracture mechanics", lecture notes, course 4A780, 2012.
- [6] James A. Harter, AFGROW users guide and technical manual, 1999.
- [7] Yinan Sun, "The Study of Overload Effects on Fatigue-Crack-Growth Behaviour by Neutron Diffraction", 2006.
- [8] W. Elber, "Fatigue crack closure under cyclic tension", 1970.
- [9] Abaqus User's manual, Dassault Systems, version 2017.
- [10] C. Delprete, C. Rosso, R. Sesana, "Comparison between damage criteria in thermos-mechanical fatigue", International journal of mechanics and control, Vol. 09, 2008.

- [11] R. Nobile, slides of the lectures of "Progettazione assistita e meccanica sperimentale-Tensioni residue".
- [12] Vishay precision group, "Measurement of Residual Stresses by the Hole-Drilling Strain Gage Method", 2010.
- [13] J.L. Roberts, "Residual Stress Effects on Fatigue Life via the Stress Intensity Parameter, K ", 2002.
- [14] D. Schnubel, N. Huber, "Retardation of fatigue crack growth in aircraft aluminium alloys via laser heating- Numerical prediction of fatigue crack growth", 2012.
- [15] K. Miyazaki, M. Mochizuki, S. Kanno, "Analysis of stress intensity factor due to surface crack propagation in residual stress fields caused by welding", 2002.
- [16] Lin Zhu, Min-Ping Jia, "A new approach for the influence of residual stress on fatigue crack propagation", 2017.
- [17] Chin-Hyung Lee, Kyong-Ho Chang, "Finite element computation of fatigue growth rates for mode I cracks subjected to welding residual stresses", 2011.
- [18] L.Y. Chen, G.Z. Wang, J.P. Tan, F.Z. Xuan, S.T. Tu, "Effects of residual stress on creep damage and crack initiation in notched CT specimens of a Cr-Mo-V steel", 2013.
- [19] C.J. Lammi, "The effects of processing residual stresses on the fatigue crack growth behaviour of structural materials", 2009.
- [20] R. Citarella, G. Cricri, "A two-parameter model for crack growth simulation by combined FEM-DBEM approach", 2009.

- [21] G. Servetti, X. Zhang, "Predicting fatigue crack growth rate in a welded butt joint: The role of effective R ratio in accounting for residual stress effect", 2009.
- [22] Jinxiang Liua, Huang Yuanb, Ridong Liaoa, "Prediction of fatigue crack growth and residual stress relaxations in shot-peened material", 2010.
- [23] Z.C. Liu, C. Jiang, B.C. Li, X.G. Wang, "A residual stress dependent multiaxial fatigue life model of welded structures", 2017.
- [24] Ž. Bozic, S. Schmauder, M. Milikota and M. Hummel, "Multiscale fatigue crack growth modelling for welded stiffened panels", 2014.
- [25] C.G. Matos, R.H. Dodds Jr, "Modeling the effects of residual stresses on defects in welds of steel frame connections", 1999.
- [26] C. Garcia, T. Lotz, M. Martinez, A. Artemev, R. Alderliesten, R. Benedictus, "Fatigue crack growth in residual stress fields", 2016.
- [27] Y. Lei, O'Dowd and G.A. Webster, "Fracture mechanics analysis of a crack in a residual stress field", 2000.
- [28] T. Mura, "Micromechanics of defects in solids", 2013.
- [29] Y. Ueda, K. Fukuda, K. Nakacho, S. Endo, "A new measuring method of residual stresses with the aid of finite element method and reliability of estimated values", 1975.
- [30] C.J. Garcia Lopez, "Modelling of Residual Stress Fields and their Effects on Fatigue Crack Growth", 2015.

- [31] M.R. Hill, D.V Nelson, "The inherent strain method for residual stress determination and its application to a long welded joint", 1995.
- [32] V.M.A. Leitaó, M.H. Aliabadi, D.P. Rooke, "Crack growth simulation in the presence of stress fields", 1995.
- [33] M. Beghini, L. Bertini, "Fatigue crack propagation through residual stress fields with closure phenomena", 1990.
- [34] V. Lacarac, D. J. Smith, "Fatigue crack growth from plain and cold expanded holes in aluminium alloys", 2000.
- [35] J. Toribio, J.C. Matos, B. González and J. Escudra, "Influence of Residual Stress Field on the Fatigue Crack Propagation in Prestressing Steel Wires", 2015.
- [36] J.E. LaRue, S.R. Daniewicz, "Predicting the effect of residual stress on fatigue crack growth", 2006.
- [37] M.E. Fitzpatrick and L. Edwards, "Fatigue Crack/Residual Stress Field Interactions and Their Implications for Damage-Tolerant Design", 1997.
- [38] R. C. McClung, "A literature survey on the stability and significance of residual stresses during fatigue", 2006.
- [39] H.-C. Choi and J.-H. Song, "Finite element analysis of closure behaviour of fatigue cracks in residual stress fields", 1995.

Acknowledgements

Above all, I need to thank my family for supporting me in these five years of higher education both economically and emotively.

Then I want to thank my academic advisors Eng. Paolo Baldissera and Prof. Cristiana Delprete for supporting me in the work about my master thesis, along with the engineers of FCA Automobiles Eng. Andrea Vercelli, Eng. Lamonaca Gianni and Eng. Lo Guasto Salvatore.

Finally, I want to thank "my university", intended as all the professors, the management staff and the workers for keep on providing an excellent level of education and services to all the students.