NATIONAL TECHNICAL UNIVERSITY OF
ATHENS
SCHOOL OF MECHANICAL ENGINEERING

MACHINE DESIGN LABORATORY

DIPLOMA THESIS

# Development of a spur gear tooth flank optimization method for minimizing surface fatigue

Panagiotis Koronaios

Supervisor:
Vasileios Spitas
Associate Professor NTUA

Athens, July 2021

# Abstract

Gears are used under demanding operating conditions which lead to a continuous change of the tooth surface. There are various wear and fatigue mechanisms through gear's life circle that can lead them to failure. Increasing operation time might have as a result the development of 'pitting' which is appearances of fatigue on the surface of the tooth flank. The development of pitting is possible to ensue growing micro-cracks in the tooth surface causing intense stresses. According to Hertz's theory, which is presented thoroughly in this thesis, these stresses are directly related to the equivalent curvature of the bodies in contact. As a result of this analysis, methods to reduce pitting are proposed including redesign of the gear pair geometry. However, there are some geometrical criteria and constraints that must be met according to theory of gearing such as the law of gearing or pitch compatibility. Furthermore, constrains is predominant part of the present analysis. The optimization of gear profiles is attempted with genetic algorithms, Fmincon method (gradient based method) and with steepest descent method (deterministic optimization) -in MATLAB/SIMULINK environment-for both closed contact path and open contact path. The profile of pinion or rack gear is modeled with 4th degree B-Splines and the optimization algorithm determines the optimum positions of the control points in order to minimize the equivalent curvature of the conjugate flanks at each point. This analysis leads to the comparison of the three optimization methods given the results provided and their computational cost. An optimum solution corresponding to a typical one-stage speed reducer was reached and the performance of the optimized gear pair was found to surpass the performance of involute and sine-rack of similar geometry in terms of both root strength and pitting resistance. Finally, a finite element analysis completes this thesis comparing the best optimization result with the involute gears.

# Περίληψη

Οι οδοντωτοί τροχοί χρησιμοποιούνται υπό απαιτητικές συνθήκες που συχνά οδηγούν στην καταστροφή τους λόγω των μηχανισμών φθοράς στους οποίους υπόκεινται. Υπάρχουν διάφοροι μηχανισμοί φθοράς και κόπωσης και ένας από τους πλέον σημαντικούς είναι η ανάπτυξη εκκοιλάνσεων επί της επιφάνειας της κατατομής. Αυτό το φαινόμενο συχνά προκαλεί μικρό-ρωγμές στην επιφάνεια του οδόντα προκαλώντας έντονες καταπονήσεις και τάσεις. Σύμφωνα με τη θεωρία του Hertz που παρουσιάζεται διεξοδικά, αυτές οι τάσεις σχετίζονται άμεσα με την ισοδύναμη καμπυλότητα των σωμάτων που έρχονται σε επαφή. Ως αποτέλεσμα αυτής της ανάλυσης, προτείνονται μέθοδοι για τη μείωση της ισοδύναμης καμπυλότητας και συνεπώς της καταπόνησης των κατατομών, συμπεριλαμβανομένου του επανασχεδιασμού της γεωμετρίας των οδοντοτών τροχών. Ωστόσο, υπάρχουν ορισμένα γεωμετρικά κριτήρια και περιορισμοί που πρέπει να πληρούνται σύμφωνα με τη θεωρία των οδοντώσεων, όπως μεταξύ άλλων ο βασικός νόμος οδοντώσεων (ΒΝΟ) ή η συμβατότητα του βήματος. Επιπλέον, οι περιορισμοί είναι βασική πτυχή της παρούσας εργασίας. Η βελτιστοποίηση των κατατομών επιχειρείται με γενετικούς αλγόριθμους, με τον αλγόριθμο Fmincon (gradient based optimization) και με την μέθοδο της απότομης καθόδου (steepest descent) (ντετερμινιστική βελτιστοποίηση) στο περιβάλλον της MATLAB/SIMULINK τόσο για κλειστή τροχιά επαφών όσο και για ανοικτή τροχιά επαφών. Μοντελοποιείται είτε το πινιόν είτε ο κανόνας, ανάλογα τι εξυπηρετεί καλύτερα στην κάθε περίπτωση, με B-Splines 4ου βαθμού, ενώ, ο αλγόριθμος δέχεται ως είσοδο τη θέση των σημείων ελέγχου και καθορίζει την ισοδύναμη καμπύλη καμπυλότητας σε ολόκληρη την κατατομή σύμφωνα με τη θεωρία του Hertz. Αυτή η ανάλυση οδηγεί στη σύγκριση των τριών μεθόδων βελτιστοποίησης δεδομένων των αποτελεσμάτων που παρέχονται και του υπολογιστικού κόστους τους. Επιτεύχθηκε μια βέλτιστη λύση που αντιστοιχεί σε ένα ζεύγος οδοντωτών τροχών που η ισοδύναμη καμπυλότητα είναι σημαντικά βελτιωμένη σε σχέση με τους οδοντωτούς τροχούς εξειλιγμένης και ημιτονοειδούς κανόνα όσον αφορά τόσο την αντοχή όσο και στην εμφάνιση εκκοιλάνσεων. Τέλος, τα αποτελέσματα επιβεβαιώνονται με τη μέθοδο των πεπερασμένων στοιχείων και συγκρίνεται η βέλτιστη λύση με αυτή τα αντίστοιχα γρανάζια εξειλιγμένης.

# Acknowledgments

Throughout the writing of this dissertation, I have received a great deal of support and assistance.

I would first like to thank my supervisor, Professor Vasileios Spitas, whose expertise was invaluable in formulating the research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I would also like to thank all my tutors for their valuable guidance throughout my studies. You provided me with the tools that I needed to choose the right direction and successfully complete my dissertation.

Additionally, I would like to express gratitude to researcher PhD candidate Christos Kalligeros, for his treasured and dedicated support which was really influential in shaping my experiment methods and critiquing my results.

Moreover, I would like to thank my parents Amalia and Theologos for their wise counsel and sympathetic ear and my sister Maria for her encouragement and support. You are always there for me. Furthermore, I could not have completed this dissertation without the support of my friends who provided stimulating discussions as well as happy distractions to rest my mind outside of my research.

Finally, I want to thank God for blessing me to have my undergraduate studies completed and have all these supportive people by my side.

## Contents

## Figures

## Tables

# 1. Introduction

Gears are one of the most common machine components used in modern industry. Although their use has been known since antiquity, their widespread use was achieved after the improvement of their construction machines, which were able to produce wheels of high precision and profile quality and deliver shapes suitable for more complex forms of transmission. Their ability to allow stable transmission and the transfer of high power at high efficiency has made their application valuable or even necessary in a wide range of applications and has allowed their incorporation in machines and mechanisms of great importance, in critical positions. For this reason, their production today is massive and their quality is of great interest to the industry.

The effort to model the kinematics and strength of gears dates back to the late 18th and early 19th century with the discovery of Euler-developed involute and the modeling of gear kinematics by Relaux respectively. Although in the beginning the predominant gears were those with cycloidal teeth, they were quickly displaced by the involute gears, and especially the 20° involute gears which combines sufficient overlap between meshing gears and increased load capacity.

The first attempt to compute the stresses in gears belongs to Lewis, who formulated the classic method of estimating the bending stresses in a gear tooth (Lewis's equation). The first to compile pre-existing theories and formulations of computation concerning involute gears was Buckingham in his book "Analytical Mechanics of Gears" [11]. Later, many researchers such as Timoshenko [12], Kelley [13], Nieman [14], Small [15], Wellauer [16], Heywood [17] have worked on the to the development of stresses in gears while others such as Dolan and Broghamer [18] calculated with the method of photoelasticity an empirical formula to calculate the stress coefficient.

Around the middle of the 20th century F. Livtin [19,20] began to develop the "Theory of Gearing" according to which it was possible to calculate any gear tooth pair in mesh. At the same time, various methodologies and theories for calculating the geometry of gears were developed either by individual researchers (Baxter, Colbourne, Dudley, Henriot, Merrit, Salamoun, Stipelman, Wildhaber) or by manufacturers (Gleason, Illinois Tool Works) [21-23]. Given the geometry of the gears in the stress analysis with the Finite Element Method, many researchers were engaged in the optimization of the gears while the computational optimization methods were being developed. This optimization focuses mainly on how displacements of gears are distributed (geometrical optimization) in order to design gears that for given loads they will experience less fatigue while there are studies focusing on new materials that will contribute to a better stress distribution (Hoffman, Townsend [24]) or new construction methods (Daniewicz [25]). Finally, methods that suggest the use of parametric curves such as polynomials are also sporadically implemented (Tsai [26]).

Additionally, H. Hertz with his two papers in 1881 and 1882 [32], has introduced his theory for the stress distribution generated between two elastic bodies in contact at a single point or line upon being pressed together with a force. Hertz's theory, remains the foundation for the analysis of most contact problems. This theory is described in detail in the present thesis because the optimization methods that are being developed are based on the principles of his theory.

Optimization methods for gear transmissions are classified mainly into two categories. Firstly, methods that provide low computational cost by using a combination of analytical and empirical formulations for the determination of stress calculations and as a result they are less reliable to give an optimum solution (Mabie [27], Rogers [28]). Secondly, methods that use more accurate methods for the estimation of stresses but with an extensive computational cost. Finally, standards such as AGMA [29], DIN [30] or ISO [31] have given design and computational instructions that are based on theoretical results of the prementioned methods and observations from industrial implementations.

In modern days, gears are very often used in power transmissions. They mesh at high rotational speeds and carry high loads. Under these unfavorable conditions, gears fatigue and possibly fail, with their failure impeding the functionality of the mechanical construction in which they are included. Failure can occur either statically, due to high load, or due to dynamic phenomena, such as wear and fatigue. Static failure is the result of either poor design (resizing) or misuse (operation in worse conditions than those provided by the designer). So, neglecting the case of static failure, which can be avoided with proper design and use, we examine the wear and fatigue of gears. These two phenomena will certainly occur during the meshing of two gears, as they are in contact withstanding dynamic loads.

Even if the appearance of wear and fatigue is a given, it is not necessary that the functionality of a gear will be affected. With an alternative design the consequences of the above for a certain system may be insignificant during the life of the construction. So, by examining the mechanisms of wear and fatigue we can reduce their negative effects and achieve more reliable constructions, cheaper, with longer life.

Some other benefits that can derive from the reduction of pitting (appearances of surface fatigue) in gears are the following. Firstly, the reduction of the mean equivalent curvature of the flanks leads to the reduction of mean surface pressure in gears and to its normalization. This means that now we are able to also use materials other than steel for gear manufacturing. For instance, ceramic materials that have low resistance in surface pressure could be an option while they present better behavior in high temperatures and in intense compression loads. Furthermore, the development of high surface pressure in gears can also create problems in their lubrication since it can cause a squeeze flow of the oil and change its viscosity with the development of high temperatures. So, we are forced to use more sophisticated lubricants which are also more expensive. However, if the surface pressure is lowered then we are given the option to use cheaper lubricants and even plain water in some applications like when we have ceramic gears or when the application is marine. [7]

The present thesis is divided in 7 main chapters. The first topic to cover is Hertz’s theory of contact and surface fatigue as a failure mechanism. In this chapter possible ways to reduce pitting in a gear transmission are proposed. The next chapter discusses some of the most important equations according to which the optimization is achieved. In the 4th chapter the optimization process that was suggested is being discussed; The use of B-Spline curves, kinematics and modeling of gear meshing, the constraints that should be taken into consideration and the optimization methods that are more efficient for this problem are some of the topics that are discussed in this chapter. Moreover, in the 5th and 6th chapter the closed contact path analysis and the open contact path analysis and the respective results are presented thoroughly. In chapter 7 a finite analysis method confirms the optimization results comparing them with the involute gears. Finally, conclusions are presented in order to assess the efficiency of the present modeling.

# 2. Bodies in Contact

## 2.1 Hertz theory

Let two bodies be in contact as shown in the figure below and let them be deformed under the load, creating a contact zone. [4]

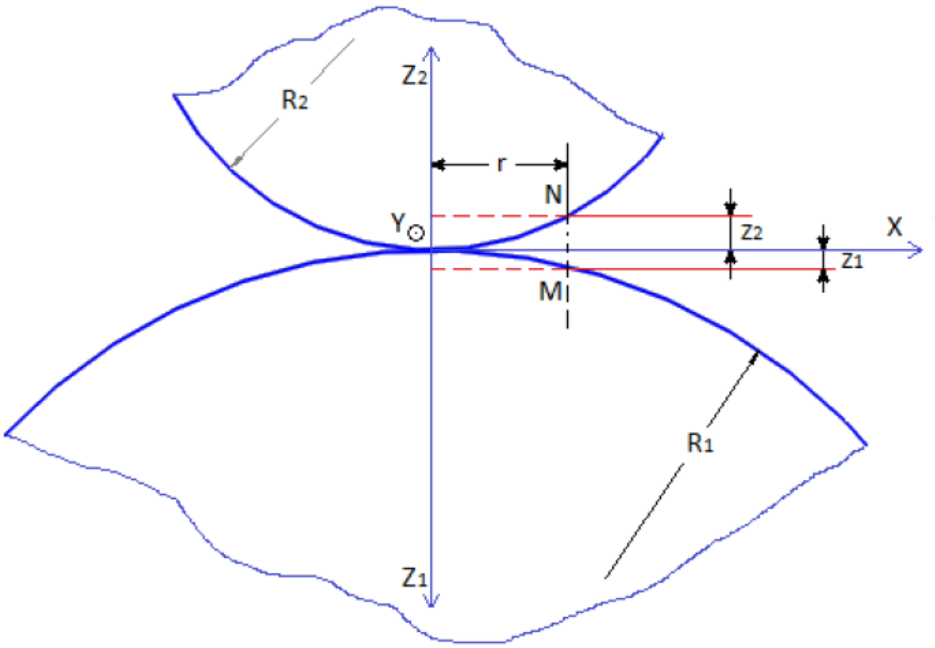


**Figure 2.1.1. Two bodies in contact on the plane** [10]

Ignoring higher order terms, the surfaces of bodies in the point of contact can be expressed as

$$z_1 = A_1x^2 + A_2xy + A_3y^2 \tag{2.1.1}$$

$$z_2 = B_1x^2 + B_2xy + B_3y^2 \tag{2.1.2}$$

The distance between two points M and N of the two bodies (figure 2.1.1) can be written as:

$$z_1 + z_2 = (A_1 + B_1)x^2 + (A_2 + B_2)xy + (A_3 + B_3)y^2 \tag{2.1.3}$$

By choosing an appropriate coordinate system we can delete the product term xy whenever equation (2.1.3) can be written as:

$$z_1 + z_2 = Ax^2 + By^2 \tag{2.1.4}$$

The coordinate system to which equation (2.1.4) applies is the system of main curvatures. Let $R_1$, $R'_1$, be the major radii of curvature of body 1 and $R_2$, $R'_2$, the main radii of curvature of body 2. $R_1$, $R'_1$, belong to planes perpendicular to each other as well as $R_2$, $R'_2$, respectively. Let $\psi$ be the angle formed between the planes containing $R_1$ and $R_2$. Then the constants of equation (2.1.4) arise as the solution of the system :

$$A + B = \frac{1}{2}\left(\frac{1}{R_1} + \frac{1}{R_1'} + \frac{1}{R_2} + \frac{1}{R_2'}\right) \tag{2.1.5}$$

$$B - A = \frac{1}{2}\left[\left(\frac{1}{R_1} - \frac{1}{R_1'}\right)^2 + \left(\frac{1}{R_2} - \frac{1}{R_2'}\right)^2 + 2\left(\frac{1}{R_1} - \frac{1}{R_1'}\right)\left(\frac{1}{R_2} - \frac{1}{R_2'}\right)cos\psi\right]^{\frac{1}{2}} \tag{2.1.6}$$

Suppose that the coordinate system of the figure 2.1.1 is the coordinate system of the main curvatures with the angle $\psi = 0$ the angle between the planes containing $R_1, R_2$. As the two bodies are deformed locally, around the point of contact, under the influence of compressive load, the points $M, N$ are displaced by $w_2$ and $w_1$ respectively in the directions $Z_2$ and $Z_1$ respectively, as shown in the figure 2.1.1. If the distance between $M, N$ was originally $d$, then:

$$w_1 + w_2 + z_1 + z_2 = d \tag{2.1.7}$$

Combining equations (2.1.4) and (2.1.7) we have:

$$w_1 + w_2 = d - Ax^2 - By^2 \tag{2.1.8}$$

Assuming that bodies 1 and 2 are semi-continental (this is especially true for low-yield materials, such as gears, as the contact deformation zone is very small) we can we get the following expression:

$$w_1 + w_2 = \left(\frac{1 - v_1^2}{\pi E_1} + \frac{1 - v_2^2}{\pi E_2}\right)\iint \frac{qdA}{r} \tag{2.1.9}$$

Where $qdA$ is the infinite load exerted on an infinite surface $dA$ at a distance $r$ from the center of the contact zone (theoretical point of contact). The integration extends over the entire contact area (surface). Let $E_1, E_2$ be the measure of elasticity of body material 1 and 2 respectively and $v_1, v_2$ let be the Poisson ratios.

$$k_1 = \frac{1 - v_1^2}{\pi E_1} \tag{2.1.10}$$

$$k_2 = \frac{1 - v_2^2}{\pi E_2} \tag{2.1.11}$$

Combining equations (2.1.8), (2.1.9), (2.1.10) and (2.1.11) we have:

$$(k_1 + k_2)\iint \frac{qdA}{r} = d - Ax^2 - By^2 \tag{2.1.12}$$

Based on equation (2.1.12) the task is to find stress distribution q that verifies it. Hertz showed that a stress distribution forming a half-ellipsoid satisfies equation (2.1.12). Hertz's case has also been proven experimentally. Obviously if the stresses follow such a distribution, the contact zone has an elliptical shape. Figure 2.1.2 shows the contact zone as well as the stresses distribution (view of the 2 dimension distribution)

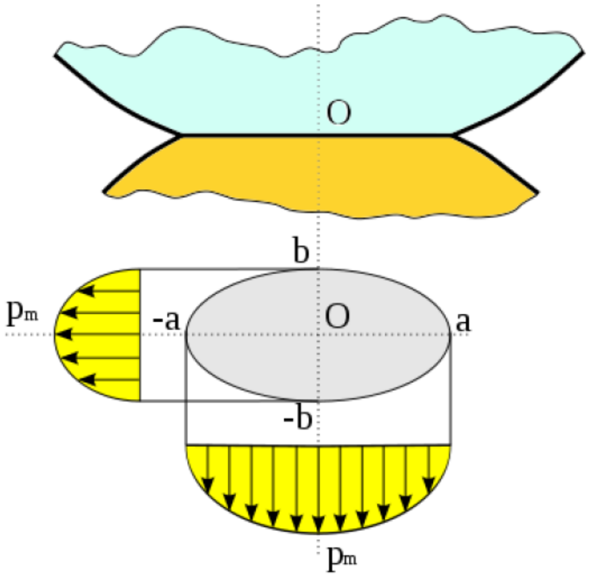


**Figure 2.1.2. Deformation zone and distribution of compressive stresses in it according to Hertz**

If the load exerted between the surfaces is *P* then the maximum pressure develops at the center of the ellipse of contact and is:

$$q_o = \frac{2}{3}\frac{P}{\pi ab} \qquad (2.1.13)$$

Where α, b let be the axes of the ellipse of contact. These are calculated by the following formulas:

$$\alpha = m\sqrt[3]{\frac{3}{4}\frac{\pi P(k_1 + k_2)}{A + B}} \qquad (2.1.14)$$

$$b = n\sqrt[3]{\frac{3}{4}\frac{\pi P(k_1 + k_2)}{A + B}} \qquad (2.1.15)$$

Where m, n coefficients depend on the ratio and their value is obtained from arrays that arrays that have emerged experimentally. An example is the one proposed by H. L. Whittemore and S. N. Petrenko [33]:

| θ | 30° | 35° | 40° | 45° | 50° | 55° | 60° | 65° | 70° | 75° | 80° | 85° | 90° |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| m | 2.731 | 2.397 | 2.137 | 1.926 | 1.754 | 1.611 | 1.486 | 1.378 | 1.284 | 1.202 | 1.128 | 1.061 | 1.000 |
| n | 0.493 | 0.530 | 0.567 | 0.604 | 0.641 | 0.678 | 0.717 | 0.759 | 0.802 | 0.846 | 0.893 | 0.944 | 1.000 |

Where:

$$\theta = \frac{B - A}{A + B} \tag{2.1.16}$$

From equations (2.1.13), (2.1.14), (2.1.15) the factors that determine the magnitude of the growing surface pressure are derived. These are:

1) The load $P$. Reduction of the load leads to lower surface pressure.
2) The material of the gears ($k_{1,2}$ depend on $E_{1,2}$ and $v_{1,2}$ ,equations (2.1.10) and (2.1.11)). The more compliant the material, the lower the growing pressure (for large magnitudes of elasticity E the deformation around the contact point is limited, hence the dimensions of the contact zone)
3) Curvature of tooth profile at the point of contact. The term $A + B$ (referred to as equivalent curvature) depends on the curvature of the teeth, equation (2.1.5). Increasing the radius of curvature of the tooth (i.e. decreasing the curvature $k = 1 / R$) leads to a decrease in surface pressure.

The above refers to the general case of two bodies in contact. Applying the results to spur gears results in some simplifications:

1) The radii of curvature $R'_1$ and $R'_2$ tend to infinity as the teeth have no curvature in the direction of the width. So, the equivalent curvature is given by the following formula:

$$A + B = \frac{1}{2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \tag{2.1.17}$$

2) For the angle ψ formed between the planes containing $R_1, R_2$ , $\psi = 0$
3) The theoretical contact is made on a line and not a point, so the contact zone is no longer elliptical in shape but a rectangle of length h (the width of the gear) and height b, where:

$$b = \sqrt{\frac{4P'(k_1 + k_2)R_1R_2}{R_1 + R_2}} \tag{2.1.18}$$

$$P' = \frac{P}{h} \tag{2.1.19}$$

4) The stress distribution is elliptical but has fixed profile along the contact line.
5) The maximum magnitude of surface pressure is:

$$q_o = \frac{2P'}{\pi b} \tag{2.1.20}$$

## 2.2 Surface fatigue - Pitting

The requirements of the machines in terms of efficiency, durability and power density are steadily increasing. This situation is intensified especially for gears due to climate change and therefore originated legal regulations that lead to the need of e.g., effective car gearboxes or wind turbine transmission. To favor the development of these advanced future tools is essential to understand the mechanical and tribological mechanisms in detailed contact of the gears. The analysis and understanding of these mechanisms will allow the design of optimized gear flanks with maximum power density and increased safety and durability. Tools such as the fatigue simulation, are necessary for this kind of future engineering. The models are based on detailed understanding the contact of the gear teeth and the prevailing conditions.

When meshing profiles transmit loads, a small pressure band appears at the point of contact of the meshing teeth. In this area the loads are intense and at the same time the surface on which they are exerted is very small. In a pair of gears the contact theoretically takes place in a line and the stresses are infinite. Of course, the teeth are deformed locally, so a small contact area appears.

As it results from the above, the stresses that appear in the contact zone, are very high but they are compressing, so at a first glance they do not seem dangerous. However, due to the Poisson effect, shear stresses appear from the surface of the tooth inwards. These shear stresses take a maximum value at a certain depth in the center of the contact zone.

Referring to a specific point of the working profile, the resulting shear stresses are periodic since each tooth engages with a given frequency and in addition each "point" of the working side comes into contact with the meshing tooth. Therefore, due to shear fatigue, small cracks appear at shallow depths below the tooth surface, which then are gradually developing. These appearances of fatigue are known as pitting.

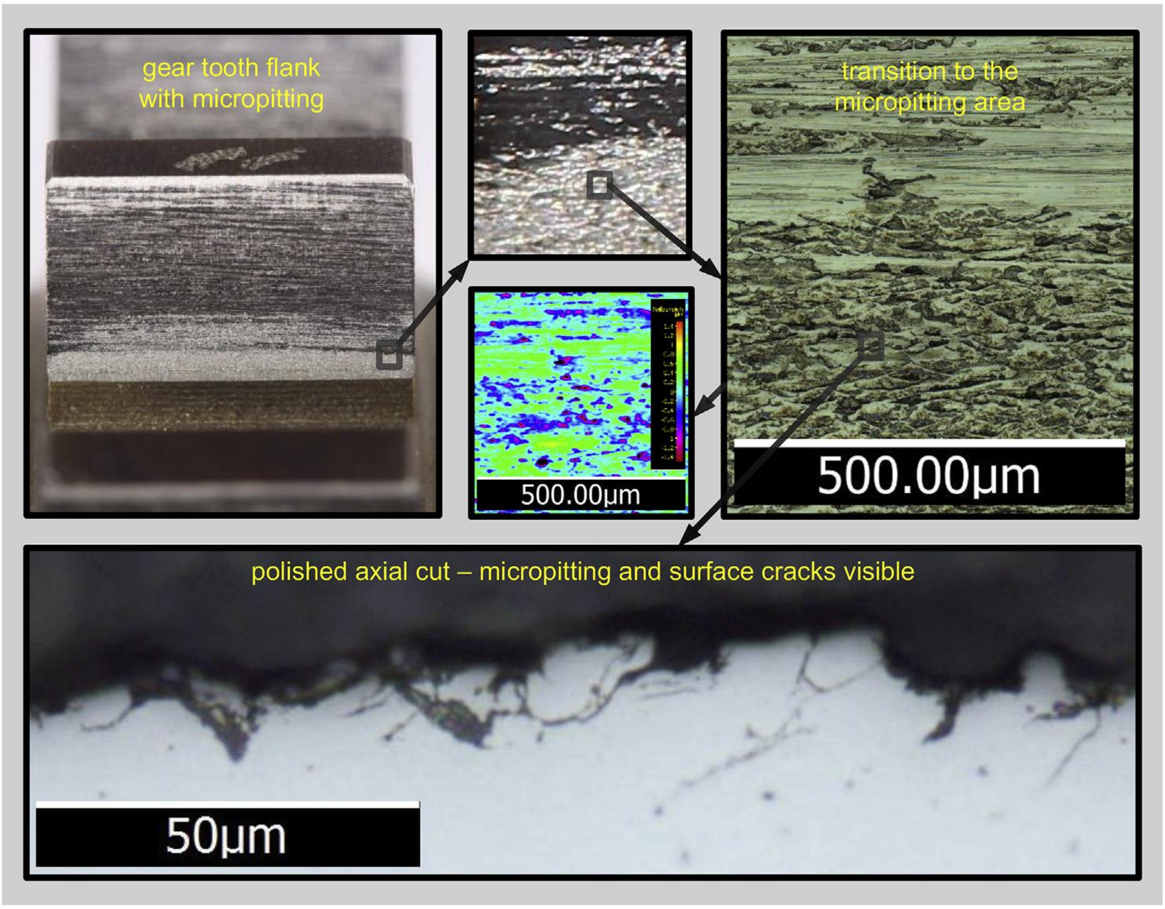


**Figure 2.2.1. Gear tooth with micropitting area – micropitting in detailed pictures** [6]

The loads at the point of contact are compressive and during compression the stress concentration phenomenon is absent. Thus, the cracks do not have a significant effect on the functionality of the teeth while they are below the surface. However, as they gradually develop, under certain conditions they come to the surface when the following phenomenon is observed. Lubricant enters the crack and is trapped below the tooth surface. Along with the zone above, the "lubricated" crack enters into meshing, the strong compressive stresses that develop cause high pressures to develop inside the crack due to the minimal compressibility of the lubricant. These pressures extend the cracks to the surface of the tooth. The end result is the detachment of thin and small sheets of material from the working profile.[6]

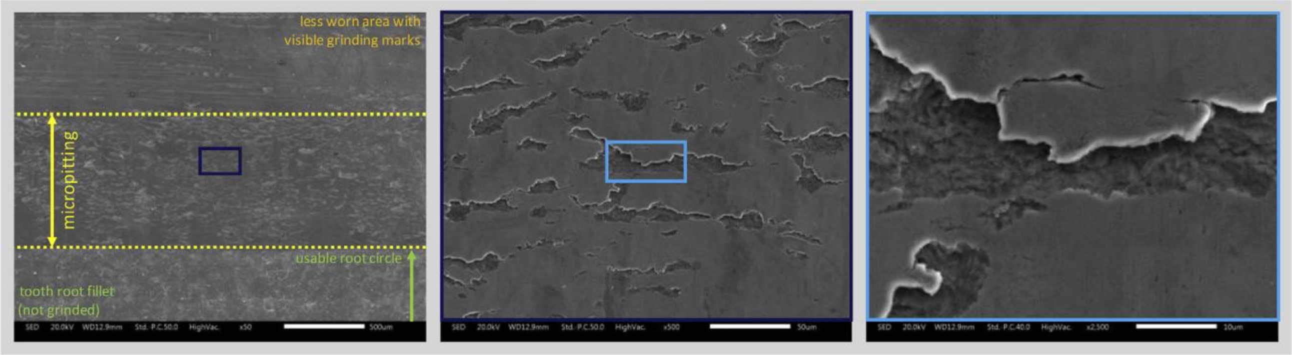

less worn area with visible grinding marks
micropitting
usable root circle
tooth root fillet (not grinded)

**Figure 2.2.2. Scanning electron microscope pictures of a micropitting area** [6]

During this phenomenon the working side loses its original geometry and surface quality. At the same time the increased roughness favors the wear mechanisms mentioned above so gears are soon driven to failure. Failure may not necessarily mean fracture but generally operation that is over specifications (increased oscillations, uneven power transmission etc.).

To avoid the pitting effect, we would ideally like to avoid (practically reduce) the appearance of micro-cracks, their development and their "climbing" to the surface. This could be achieved in two ways in general.

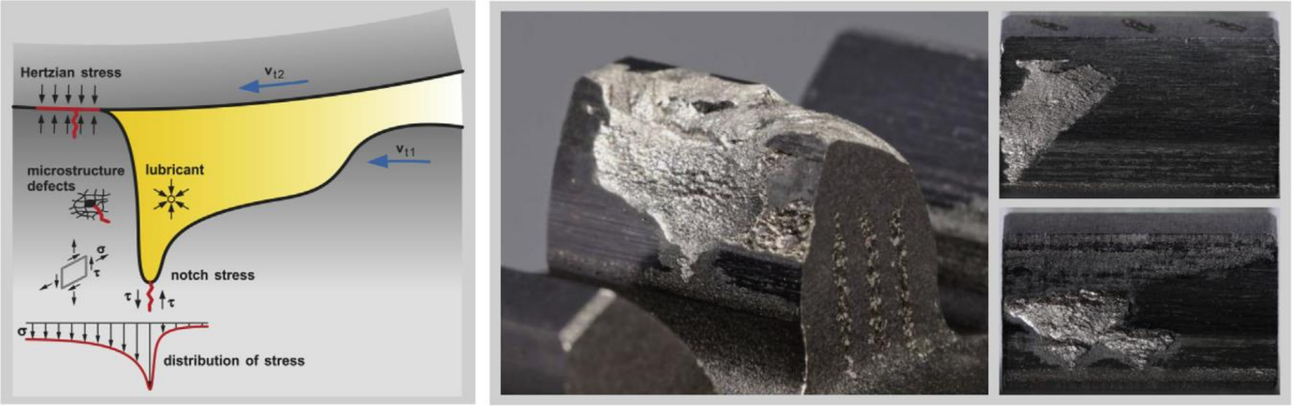


**Figure 2.2.3. Micropitting area transition and test gears with pitting** [6]

The cause of the appearance of cracks and their development is, as mentioned above, the shear fatigue that occurs just below the surface due to the pressure in the contact zone of the gears. Therefore, the factors that determine the pressure in the contact zone and consequently the shear caused (maximum shear depth and magnitude) must be determined. The theory of contact of deformable bodies (for small deformations) has been developed by Hertz and has been experimentally verified. The causes that lead to cracks in the surface must also be identified.

### 2.2.1 Rise of cracks on the surface

It has been reported above that surface pressure causes secondary shear at a shallow depth below the contact zone. This depth z, depends on the ratio a / b (half-axis ratio of the lack of contact) and is of the same order of magnitude as a, b.

At this depth, cracks appear first and then develop. Their rise to the surface can be caused by various factors such as material defects or persistent stresses on the tooth surface. But there is a main mechanism that causes them to rise and this is based on the change of surface pressure from the engagement of the teeth.

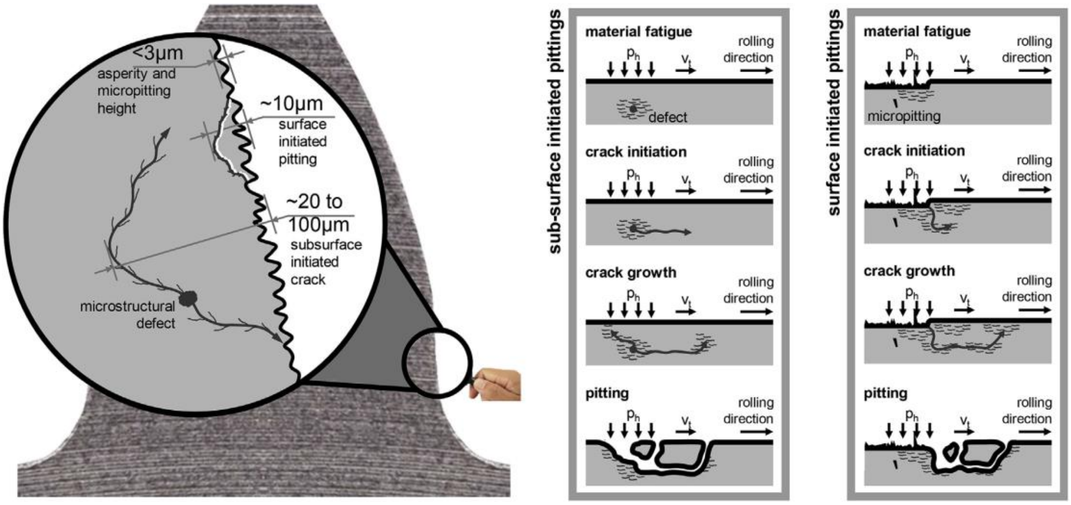

<3μm
asperity and micropitting height
~10μm
surface initiated pitting
~20 to 100μm
subsurface initiated crack
microstructural defect
sub-surface initiated pittings
material fatigue
rolling direction
$p_h$
$v_t$
defect
crack initiation
crack growth
pitting
surface initiated pittings
micropitting

**Figure 2.2.4. Surface and subsurface crack initiation mechanisms and dimensions** [6]

The distribution of stresses obtained above refers to firm bodies. The working sides of the gears, however, slide among themselves. With the superimposition of the friction the distribution is not symmetrical with respect to the center of the contact zone (elliptical or cylindrical distribution). This results in the development of torque to the center of the contact zone. This torque "pushes" the cracks towards the surface. Similarly, cracks are driven to the surface when the developing stresses differ significantly between adjacent positions during the course of the engagement of the teeth.

### 2.2.2 Ways to reduce surface fatigue

Based on the previous analysis, we conclude that the pitting effect could be reduced in the following ways (for a specific gear material):

1) Reduction of the applied load P between the working flanks. For a given transmitted torque the load per tooth can be reduced by increasing the degree of overlap, i.e., the number of teeth at all times in engagement.
2) Reduction of the equivalent curvature A + B. In this way a larger surface area of the contact zone is achieved, so that the growing surface pressure is lower.
3) Maintain a constant surface pressure during tooth engagement to limit the rise of cracks on the surface. This is achieved when the product P (A + B) is kept constant throughout the contact trajectory.

Suppose we want to reduce the pitting effect to a degree that conveys a certain torque. The ways of limiting the pitting that we ended up above lead us to the search for a suitable geometry of the gears. This is for the following reasons:

1) For a given transmitted torque, the developing load P depends on the degree of overlap, a size determined by the geometry of the meshing profiles.
2) Equivalent curvature is purified on the basis of equation (2.1.17) from the curvatures of the meshing profiles. These are also determined by the geometry of the profiles.

All the above analysis shows that the criterion that must be met by a gear wheel to minimize the pitting effect is stable and minimum $P\ (A+B)$ throughout the contact trajectory.

## 2.3 Aims of optimization

The basic aim of this optimization is to produce a pair of gears with a constant equivalent curvature curve. However, there are a lot of parameters that have to be designated such as the optimization variables, the constraints and others that will be thoroughly discussed in the next chapters. This optimization focuses mainly in minimizing the contact pressure and as a result the surface fatigue. The bending stresses are important as well as the contact pressure however they have not been taken into consideration in the optimization process. The final solution though is compared with the involute gears in both contact pressure and bending stresses.

The objective function could become more complicated aiming to optimize both contact pressure and bending stresses. To be specific a constant bending stresses curve would be much better than a non-stable curve.

# 3. Theory of Gears

Taking into consideration the known curves used for the construction of gears (involute, epicyclic, orthocyclic, other special curves that have been proposed such as S-gears) none meet the above criterion. Of course, the curve we are looking for must, in addition to satisfying the above condition, be able to be used in the construction of gears

In summary we are looking for a pair of collaborating curves that:

1) They satisfy the basic law of gearing
2) They have pitch compatibility
3) They allow continuous meshing (depends on the overlap ratio)
4) Satisfy the condition $P\ (A\ +\ B)\ =\ min, c$ where c is constant.

## 3.1 Law of gearing

Suppose two meshing profiles such as those in figure 3.1.1. At a given time they touch at point E. The two profiles rotate with respect to points $O_1$ and $O_2$ with angular velocities $\omega_1$ and $\omega_2$ respectively. The linear velocities at point E due to the rotation of profiles 1 and 2 are plotted as $v_1$ and $v_2$ respectively. The TT is denoted by the common tangent of the profiles at the point of meshing E and the NN is the common perpendicular to the TT. [1],[10]

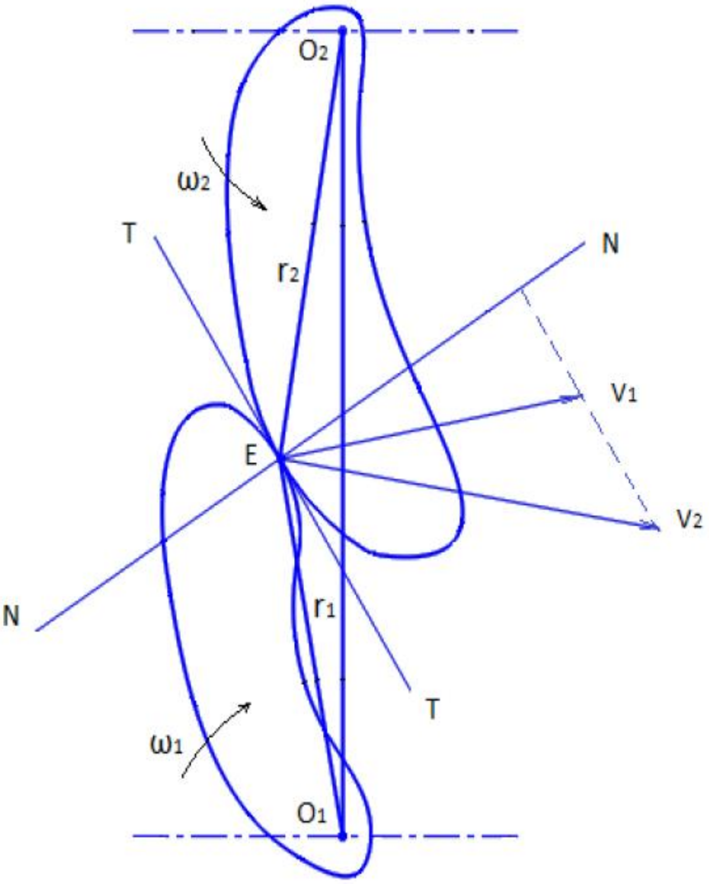


**Figure 3.1.1. Kinematic analysis two random meshing profiles** [10]

If at the given time the projections of $v_1$ and $v_2$ on NN are not equal then either one profile would tend to penetrate the other ($v_1 >\ v_2$) or the profiles would tend to move away from each other ($v_1$ <$v_2$). Both of these conditions are undesirable as they lead to uneven power transfer and discontinuous transmission. Therefore, the projections of velocities $v_1$ and $v_2$ on the common perpendicular NN must be equal at all times. This condition consists the law of gearing.

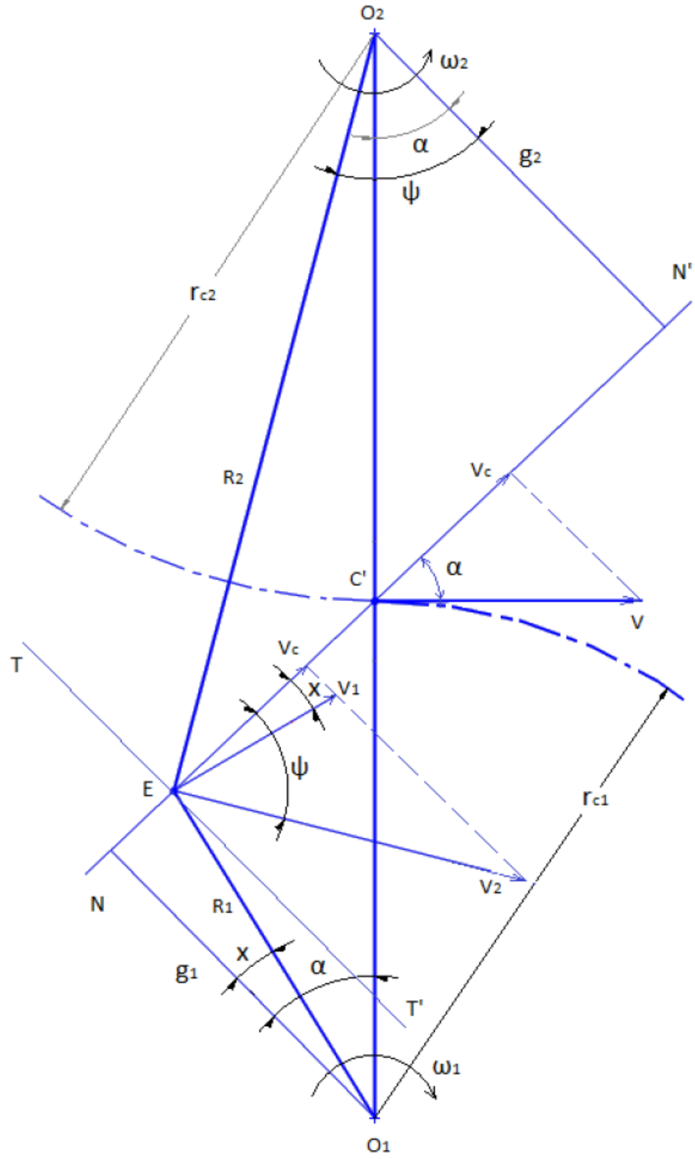


**Figure 3.1.2. Detailed illustration of the velocities of two meshing profiles in the plane and their analysis on the common perpendicular and tangent to the point of meshing** [10]

Using 3.1.2, the law of gearing can be formulated in a format that is particularly easy to use when designing profiles. From the previous we have:

$$V_1 cosx = V_2 cos\psi = V_c \tag{3.1.1}$$

In addition to the triangles $O_1EN$ and $O_2EN'$ we get the equations:

$$cosx = \frac{g_1}{R_1} \tag{3.1.2}$$

$$sin\psi = \frac{g_2}{R_2} \tag{3.1.3}$$

And from the kinematics we get the equations:

$$V_1 = \omega_1 R_1 \tag{3.1.4}$$

$$V_2 = \omega_2 R_2 \tag{3.1.5}$$

Combining the equations (3.1.1) - (3.1.5) we have:

$$\frac{\omega_1}{\omega_2} = \frac{O_2C'}{O_1C'} \qquad 3.1.6$$

If C is the rolling point on the center $O_1O_2$ of the profiles of figure 3.1.2, then due to the definition of the gear ratio:

$$\frac{\omega_1}{\omega_2} = \frac{O_2C}{O_1C} \qquad 3.1.7$$

From equations (3.1.6) and (3.1.7) we conclude that if points C and C' are not identical then the transmission ratio $\frac{\omega_1}{\omega_2}$ changes during meshing. So, the law of gearing can be formulated as follows: The smooth and continuous transfer of power and transmission of motion between two meshing profiles is ensured when the common verticality of the profiles at any point of contact passes through the rolling point. The pitch point is defined as the point of contact of the gear wheel rolling cycles.

## 3.2 Pitch compatibility

Two meshing profiles that satisfy the law of gearing have a given transmission relationship defined as:

$$i_{12} = \frac{\omega_1}{\omega_2} \tag{3.2.1}$$

In order to be able to make gears from the specific profiles, there must be integers Z1 and Z2 that correspond to the number of teeth of the two meshing gears such that:

$$i_{12} = \frac{\omega_1}{\omega_2} = \frac{Z_2}{Z_1} \tag{3.2.2}$$

## 3.3 Continuous meshing

The law of gearing and the pitch compatibility are necessary conditions but not sufficient for the operation of a stage of meshing gears. The last condition is the continuous and smooth succession of the meshing teeth. This is ensured when at least one pair of teeth is in meshing at any given time. The average number of teeth in meshing is given by the overlap ratio with the step. So, the last necessary condition is:

$$\varepsilon \geq 1 \tag{3.3.1}$$

Particularly when the working profile is involute the overlap ratio is defined as follows. The figure 3.3.1 shows two gears in meshing. The contact of path ΚΛ is observe where meshing starts at point K and end at point Λ.

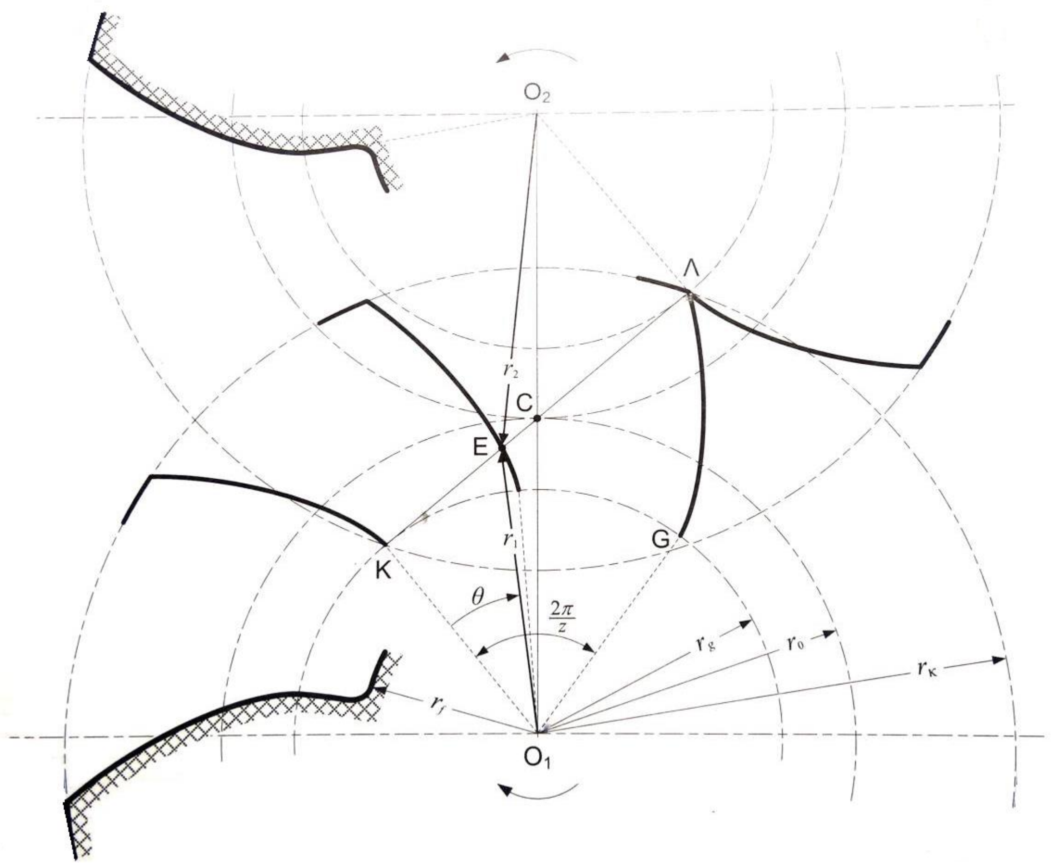


**Figure 3.3.1. Involute gears in meshing** [8]

From the definition of overlap ratio, we have the following equation:

$$\varepsilon = \frac{\mathrm{K}\Lambda}{t_0 \cos(a_0)} = \frac{K\Lambda}{t_g} \tag{3.3.2}$$

where $a_0$ is the pressure angle, $t_g$ is the pitch at base circle and $t_0$ is the pitch at pitch circle.

## 3.4 Flank Geometry Design, simplifications and assumptions

Before proceeding to the description of how to solve the optimization problem, we represent how to find the tooth profile, the meshing profile and the path of contact if the geometry of the respective rack gear is known. Both of these processes will be used in the optimization cycle

If $Z$ is the number of the first gear, $Z2$ the number of the meshing gear and $m$ is the module for these two gears we have the Pitch Circle Diameter $D_0$

$$D_0 = Zm \tag{3.4.1}$$

and,

$$r_0 = \frac{D_0}{2} \tag{3.4.2}$$

If $y = F(x)$ is the function that defines the geometry of gear rack and $(x_1, y_1)$ the coordinates of gear flank. These coordinates will be defined by B-Spline polynomials that we are going to analyze in the next chapter. The control points that will describe these polynomials will be our optimization variables. When the gear rotates by the center of base circle of the gear. Let $\theta$ be the rotation angle of the gear and $K$ the respective displacement of the rack, these two are connected by the following equation:

$$K = \theta\, r_0 \tag{3.4.3}$$

As it results from the law of gearing, the displacement of the rack is given by:

$$K = -\left(y\frac{dy}{dx} + x\right) \tag{3.4.4}$$

The coordinates $(x_1, y_1)$ of the gear flank are given by:

$$x_1 = (x + K)cos\theta - (y + r_0)sin\theta \tag{3.4.5}$$

$$y_1 = (x + K)sin\theta + (y + r_0)cos\theta - r_0 \tag{3.4.6}$$

The Contact path coordinates $(x_{CL}, y_{CL})$ are given by:

$$x_{CL} = x + K \tag{3.4.7}$$

$$y_{CL} = y \tag{3.4.8}$$

The second gear (external gear) that is in meshing with the first gear has the following coordinates:

$$x_2 = x_{CL}cos\theta_2 - (y_{CL} - r_{02})sin\theta_2 \tag{3.4.9}$$

$$y_2 = x_{CL}sin\theta_2 + (y_{CL} + r_{02})cos\theta_2 + r_{02} \tag{3.4.10}$$

Where,

$$r_{02}\theta_2 = -r_0\theta \tag{3.4.11}$$

Now that we have the gear coordinates, we can compute the curvature for the two gear flanks by the following equation:

$$\frac{1}{R_i} = \frac{\frac{d^2y_i}{dx_i^2}}{\left(1 + \left(\frac{dy_i}{dx_i}\right)^2\right)^{\frac{3}{2}}} \tag{3.4.12}$$

Where $i$ could be either 1 for the pinion or 2 for the wheel. Thus, the equivalent curvature is:

$$\frac{1}{R} = \frac{1}{R_1} - \frac{1}{R_2} \tag{3.4.13}$$

According to Hertz's theory of elastic contact, the most critical factor which controls the initialization and the development of pitting is the curvature of the gear tooth profiles. So, the optimization function might be:

$$G = mean\left(\frac{1}{R}\right) \tag{3.4.14}$$

Or alternatively it might be:

$$G = w_1\, mean\left(\frac{1}{R}\right) + w_2\, std\left(\frac{1}{R}\right) + w_3 \frac{1}{2}\int \left(\frac{1}{R}\right)^2 dx + w_4\, max\left(\left|\frac{1}{R}\right|\right) \tag{3.4.15}$$

Where mean is the mean function for the set of points of the gear flanks that are in meshing and std is the standard deviation function and $w_1, w_2, w_3$ and $w_4$ are weights.

It is crucial to decide the design variables of our optimization problem. These will be the control points of the pinion flank or the rack gear that will be modeled with 4rd degree B-Splines Polynomials.

## 3.5 Gear Tooth flank Involutization

The problem of identifying the rack gear flank coordinates that produces a particular gear flank is crucial given the fact that the rack gear can be used as cutting tool in order to produce gears. Thus, one scenario could be to optimize the rack gear, as it was fists attempted. In this case, the Control Points would model the rack gear's profile. However, in the present thesis it was preferred to model the gear profile so that the conditions and constraints are imposed effortlessly. The theory of involutization is an analytical way to find the coordinates of the path of contact given the coordinates of any gear profile. [3]

In a two-dimensional gear tooth profile (not necessarily involute), where the working tooth profile $y_G = G(x)$ and the rolling circle of the gear $r_0$ are given (3.5.1). The problem of conjugate tooth geometry is to determine the path of contact $y_P = P(x)$, the tooth profile of the generating rack $y_R = R(x)$. and the tooth profile of the conjugate gear $y_W = W(x)$. Let us also consider a random point $G(x_G, y_G)$ on the working gear tooth profile $y_G = G(x)$ and the $Oxy$ Cartesian coordinate system, where O is the center of rotation of the gear and $Oy$ coincides with the tooth centerline. At point $G$ the tooth profile is approximated with an involute segment with corresponding pressure angle equal to $a_{0G}$, such that the local involute has the same tangent with the actual profile at that point. The normal $(\varepsilon_1)$ to the profile at point $G$ has inclination equal to $-\left|\frac{1}{dG/dx}\right|_{(x_G, y_G)}$ and equation:

$$y_1 = -\left.\frac{1}{dG/dx}\right|_{(x_G, y_G)} + B \tag{3.5.1}$$

where B is a constant.

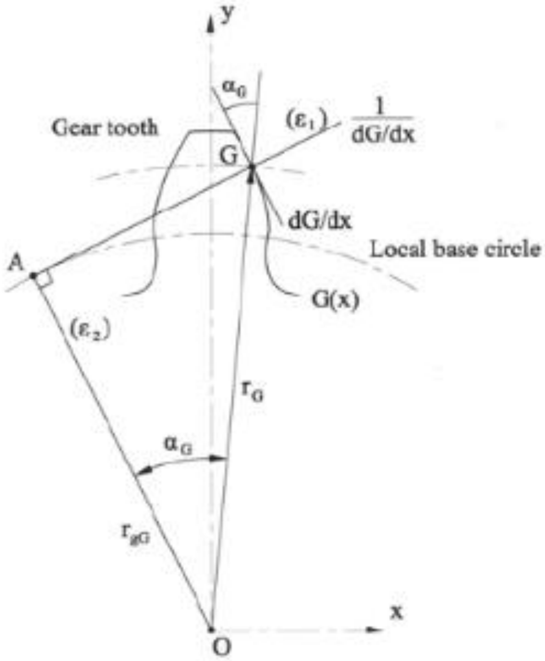


**Figure 3.5.1. The concept of local involute** [3]

Point $G(x_G, y_G)$ belongs to $(\varepsilon_1)$ and therefore should verify equation (3.5.1):

$$y_G = B - \frac{x_G}{{dG}/{dx}}\bigg|_{(x_G,y_G)} \quad \text{and} \quad B = G(x_G) + \frac{x_G}{{dG}/{dx}\big|_{(x_G,y_G)}}$$

By substituting B equation (3.5.1) becomes:

$$y = G(x_G) + \frac{x_G - x}{{dG}/{dx}\big|_{(x_G,y_G)}} \tag{3.5.2}$$

From the center $O$ of the gear, line $(\varepsilon_2)$ normal to $(\varepsilon_1)$ is drawn, so that it intersects with it at point $A(x_A, y_A)$. Since $(\varepsilon_2)$ is parallel to the tangent to the profile at point $G$, its inclination is ${dG}/{dx}\big|_{(x_G,y_G)}$ and its formula:

$$y = x\frac{dG}{dx}\bigg|_{(x_G,y_G)} + D \tag{3.5.3}$$

where D is a constant.
Since $(\varepsilon_2)$ passes through the center $(x = 0, y = 0)$ we have the following equation:

$$y = x\frac{dG}{dx}\bigg|_{(x_G,y_G)} \tag{3.5.4}$$

while point $A(x_A, y_A)$ must verify both (3.5.2) and (3.5.4):

$$x_A \left.\frac{dG}{dx}\right|_{(x_G, y_G)} = G(x_G) + \frac{x_G}{\left.{}^{dG}\!/_{dx}\right|_{(x_G, y_G)}} - \frac{x_A}{\left.{}^{dG}\!/_{dx}\right|_{(x_G, y_G)}}$$

or

$$x_A = \frac{G(x_G) + \frac{x_G}{\left.{}^{dG}\!/_{dx}\right|_{(x_G, y_G)}}}{\left.\frac{dG}{dx}\right|_{(x_G, y_G)} + \frac{1}{\left.{}^{dG}\!/_{dx}\right|_{(x_G, y_G)}}} \tag{3.5.5}$$

and from equation (3.5.4) we have:

$$y_A = x_A \left.\frac{dG}{dx}\right|_{(x_G, y_G)} \tag{3.5.4}$$

The radii of the local base circle $r_{gG}$ at point $G$ is:

$$r_{gG} = \sqrt{x_A^2 + y_A^2} = x_A \sqrt{1 + \left(\left.\frac{dG}{dx}\right|_{(x_G, y_G)}\right)^2}$$

or

$$r_{gG} = \frac{x_G + G(x_G)\left.{}^{dG}\!/_{dx}\right|_{(x_G, y_G)}}{\sqrt{1 + \left(\left.\frac{dG}{dx}\right|_{(x_G, y_G)}\right)^2}} \tag{3.5.6}$$

The local pressure angle to the local involute at $G$ is:

$$a_{oG} = \cos^{-1}\frac{r_{gG}}{r_o} \tag{3.5.7}$$

Equations (3.5.6) and (3.5.7) are independent of the chosen coordinate system. Thus, these equations remain the same even if the coordinate system changes. In the following procedure the path of contact coordinates is being calculated:

Let $y_G = G(x_G)$ be the tooth profile and $(O, x_G, y_G)$ the coordinate system (Figure 3.5.2). The local involute segment corresponds to a straight segment on the path of contact. Let $a_{oG}$ be the inclination of point P of the path of contact which lies on the intersection of the circle $(O, r_G)$ and line CP.

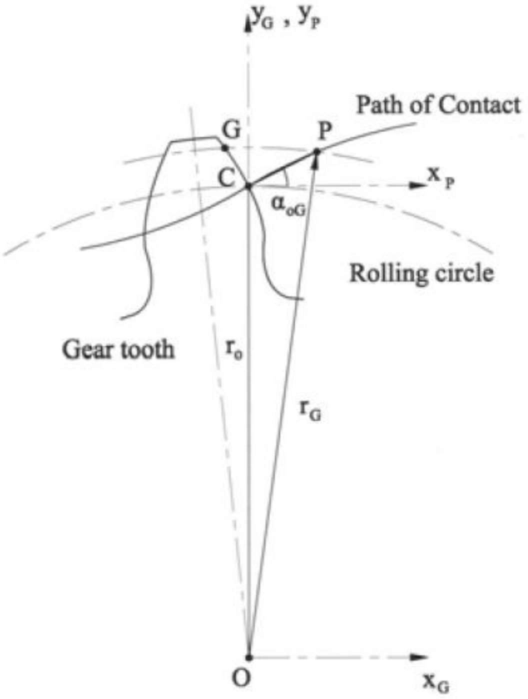


**Figure 3.5.2. Path of contact and local pressure angle** [3]

Point P satisfies of course the equation of the circle:

$$x^2 + y^2 = r_G^2 \tag{3.5.8}$$

and the equation of the line CP:

$$y = x \tan a_{oG} + r_o \tag{3.5.9}$$

From the above equations we have:

$$x^2 + (x \tan a_{oG} + r_o)^2 = r_G^2$$

and

$$x\ (1 + \tan^2 a_{oG}) + x(2r_o \tan^2 a_{oG}) + (r_o^2 + r_G^2) = 0$$

therefore,

$$x_P = \frac{-2r_o \tan a_{oG} + 2\sqrt{(1 + \tan^2 a_{oG})r_G^2 - r_o^2}}{2(1 + \tan^2 a_{oG})} =$$

$$= \sqrt{\frac{r_G^2}{(1 + \tan^2 a_{oG})} - \frac{r_o^2}{(1 + \tan^2 a_{oG})^2}} - \frac{r_o \tan a_{oG}}{1 + \tan^2 a_{oG}}$$

Therefore,

$$x_P = \sqrt{r_G^2 \cos^2 a_{oG} - (r_o \cos a_{oG})^2 \cos^2 a_{oG}} - r_o \cos a_{oG} sin\, a_{oG}$$

$$x_P = r_{gG} \left[ \sqrt{\left(\frac{r_G}{r_o}\right)^2 - \left(\frac{r_{gG}}{r_o}\right)^2} \; - \sqrt{1 - \left(\frac{r_{gG}}{r_o}\right)^2} \right] \tag{3.5.10}$$

and

$$y_P = x_P \tan a_{oG} + r_o = r_{gG} \sqrt{\left(\frac{r_o}{r_{gG}}\right)^2 - 1} \left[ \sqrt{\left(\frac{r_G}{r_o}\right)^2 - \left(\frac{r_{gG}}{r_o}\right)^2} \; - \sqrt{1 - \left(\frac{r_{gG}}{r_o}\right)^2} \right] \tag{3.5.11}$$

Hence the radii $r_{gG}$ and $r_G$ are given by the following equations:

$$r_{gG} = \frac{\left| x_G + G(x_G) {}^{dG}\!/_{dx} \big|_{(x_G, y_G)} \right|}{\sqrt{1 + \left( \frac{dG}{dx}\Big|_{(x_G, y_G)} \right)^2}} \text{ and } r_G^2 = x_G^2 + G^2(x_G) \tag{3.5.12}$$

The coordinates of the conjugate rack profile $y_R = R(x)$ and the tooth profile of the conjugate gear $y_W = W(x)$ can be calculated with known equations some of which are in the previous chapter.

# 4. Optimization process

## 4.1 Optimization Terminology

In order to apply an optimization method to a problem like this, it must be preceded by the determination of the following three basic features [2]:

i. *Objective function* that mathematically describes the problem and what needs to be achieved via optimization.
ii. *Optimization parameters* that describe how the system adapts to achieve the goals set. These are usually identified by the number of variables of an objective function and the result of an optimization method is to find the vector of the optimization parameters, which satisfies the objective function in the required way.
iii. *Constraints* that regulate the variables of the problem and lead to optimization through requirements that must be met.

## 4.2 Classification of optimization algorithms

One of the most popular classifications of optimization methods is based on whether derivatives of the objective function are used for the optimization process. Thus, there are zero order's, first order's and second order's optimization methods. The zero order's optimization methods do not use information concerning derivatives, the first order's optimization methods, use information concerning the first derivative and the second order's optimization methods, require the computation of the first and the second derivative [2].

Zero order's optimization methods are classified into deterministic methods and stohastic methods depending on how they choose the potential solution.

Deterministic methods are based on the gradual shifting of an initial solution to the optimal one through stepwise corrections. Unfortunately, any changes to the cost function or modeling of the problem require a redefinition of the optimization model. Although deterministic methods are particularly fast, they run the risk of not converging to the optimum but being trapped in local extrema, depending on the choice of initial solution.

Stochastic methods, on the other hand, are based on the almost random search for new solutions that are better than the existing one, which ultimately leads to the best. Their main feature is that they are not easily trapped in local extrema and are therefore preferred for solving convoluted optimization problems. Their only drawback is that they often have a high computational cost compared to deterministic methods.

Apart from this categorization other criteria on the basis of which we separate the algorithms are:

1. Optimization through the trial-and-error process and optimization through mathematical transfer function. There are cases where we do not know enough about how the input parameters of a system affect its output. In contrast, theorists prefer that optimization which is based on revealing the correlation between input and output and then modeling it.

2. If there is only one parameter then the optimization is characterized as one-dimensional. A problem with two or more parameters requires the use of multidimensional optimization. As the parameter algorithm increases, so does the difficulty of solving the corresponding problem. Several multidimensional optimization methods rely on reducing the problem to a series of one-dimensional optimizations.

3. Dynamic optimization means that output is a function of time, as opposed to static optimization which is time independent. For example, the shortest distance is not always the least time consuming. Thus, with the addition of the time parameter the optimization problem becomes more complicated.

4. Optimization methods are also categorized based on the continuous or discrete nature of the parameters. Discrete parameters can only take certain values, while continuous parameters can take an infinite number of values. In a discrete optimization problem, the optimal solution is one of the possible combinations of discrete sizes.

5. Parameters usually have constraints. Unrestricted optimization means that the parameters can take any value. Constrained optimization means that the optimal solution is sought by simultaneously imposing certain parameters-related conditions. In some cases, problems with constraints can be expressed in problems without constraints. Most can be equality or inequality constraints, linear or non-linear constraints. When the optimization problem is characterized by linear constraints then it is characterized as linear optimization problem. Otherwise, it is characterized as non-linear optimization problem.

### 4.2.1 Steepest Descent Method

One of the most popular optimization methods is Steepest Descent method. Although this is an unconstrained optimization method its results are always judged to ensure that the desired constraints are met. It was first pioneered by Cauchy concerning solving of linear equations system. According to this method the direction resulting the minimization of

$$\nabla \mathbf{f}^T \mathbf{s} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} s_i \tag{4.2.1}$$

where **s** is the unit vector at n-dimensional space:

$$\mathbf{s}^T.\mathbf{s} = 1 \tag{4.2.2}$$

and it is equal to:

$$\mathbf{s} = \frac{\nabla \mathbf{f}}{||\nabla \mathbf{f}||} \tag{4.2.2}$$

The minimization of function the function f is achieved through the following formulation:

$$\mathbf{x}_{\mathrm{k+1}} \ = \mathbf{x}_{\mathrm{k}} + \mathrm{a}\mathbf{s} \tag{4.2.3}$$

The variable a is selected in a way that f is minimized in the direction of **s**. For a second order's function

$$\mathrm{f}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{b}^T\mathbf{x} + \mathrm{c} \tag{4.2.4}$$

It is proved that there is the following formulation for the size of the step a:

$$a^* = \frac{\left(\mathbf{x}_{\mathrm{k}}^{T} + \mathbf{b}^{\mathrm{T}}\right)\mathbf{s}}{\left(\mathbf{s}^{\mathrm{T}}\mathbf{Q}\mathbf{s}\right)} \tag{4.2.5}$$

where the symmetry of the Hessian matrix **Q** has been used. In case f is not in a square formation – square function- or **Q** is not in an analytical formation, a can be determined through one-dimension techniques of search.

It is clear that the convergence of this method depends on f and in case f diverges from square formation - square function - steepest decent results in a vacillation or zig-zag in the parametric space. [2]

### 4.2.2 Genetic Algorithm

Genetic Algorithms are one category of Evolutionary Algorithms which are based on Darwin's theory of evolution. According to Darwin's theory the natural evolution of population depends on sorting (skillful and fast animals have more chances to survive), reproduction (species reproduce by transferring genes) and mutation (throughout centuries all species undergo mutations that change their features).

Similar to the biological analogue, Genetic Algorithms are based on the existence of populations that are potential solutions of the problem. Population varies from one generation to another while it undergoes (a) selection that depends whether it is suitable, (b) crossover in order to transfer its features to the next generations and (c) mutation in order to cover the whole space of potential solutions.

In Genetic Algorithms, exploitation is achieved through the parent selection mechanism while exploration is achieved through crossover and mutation. [5],[9].

## 4.3 Mathematical formulation of B-Splines by De Boor

### 4.3.1 General definitions

De Boor's formulation is widely used in modern software today and the optimization problems presented in the present thesis are using this formulation. [34]

Based on the interpolation points (breakpoints) and depending on the choice of continuity ($C^{p-\lambda}$) we create the knot vector consisting of the two ends of the span, with multiplicity ($p + 1$) in each, and the ($n - 1$) intermediate points of the vector each with multiplicity λ.

Thus, for the interpolation points $\xi_0, \xi_1, \ldots, \xi_n$, the corresponding knot vector is:

$$\mathbf{U} = \left\{\underbrace{\xi_0, \ldots, \xi_0}_{p+1}, \underbrace{\xi_1, \ldots, \xi_1}_{\lambda}, \ldots, \underbrace{\xi_{n-1}, \ldots, \xi_{n-1}}_{\lambda}, \underbrace{\xi_n, \ldots, \xi_n}_{p+1}\right\} \tag{4.3.1}$$

and it consists of

$$m = 2(p + 1) + \lambda(n - 1) \tag{4.3.2}$$

elements.

Therefore if $0 \leq x \leq L$ the interpolation points acquire secondary importance since the new view consists in the parameterization of the geometric curve by the introduction of Control Points, $P_i\left(i = 0, \ldots, n_p\right)$ whose number is equal to the number of polynomial coefficients that we will analyze.

Thus, the position of the corresponding point of the intervening function is given by the equation

$$x(\xi) = \sum_{i=0}^{n_p} B_{i,n_p}(\xi) x_{P_i}, 0 \leq \xi \leq 1 \tag{4.3.3}$$

Where $B_{i,n_p}(\xi)$ are the B-Spline base functions. That is, the interpolation of the coordinates of the Control Points gives exactly the coordinate of that point.

On the other hand, if the values of a random function, let $u(\xi)$ be this function, are considered at the $(n_p + 1)$ control points, the equation (4.3.3) cannot be implemented without determining certain coefficients $a_i, i = 0, \dots, n_p$, according to which this function is interpolated with sufficient accuracy.

### 4.3.2 Base functions

There are many ways to define B-Splines base functions and prove their important properties. At this point we use the retrospective formula as it is the most efficient for programming.

Let $U = \{\xi_0, \dots, \xi_m\}$ be a non-descending sequence of real numbers with $\xi_i \leq \xi_{i+1}, i = 0, \dots, m-1$. $\xi_i$ let be the knots and U let be the knot vector. The i-th base function of B-Spline's depth p, let be $N_{i,p}(\xi)$ is described as:

$$N_{i,0}(\xi) = \begin{cases} 1 & if\ \xi_i \leq \xi < \xi_{i+1} \\ & 0 \end{cases}$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{4.3.4}$$

It should be noted that:

i. $N_{i,0}(\xi)$ is a step function that equals to zero everywhere except $\xi_i \leq \xi < \xi_{i+1}$.

ii. If $p > 0$ then $N_{i,p}(\xi)$ is a linear combination of two base functions order's $p - 1$.

iii. The computation of a set of bases requires the determination of the knot vector, U, and the value of p.

iv. The equation (4.3.4) includes the quotient 0/0 that is defined equal to zero.

v. The functions $N_{i,p}(\xi)$ are partial polynomials that are defined for all the real values but they are of interest only in $[\xi_0, \xi_m]$.

vi. The semi-open space $[\xi_i, \xi_{i+1})$ is called $i^{th}$ knot span and it can have zero length given the fact that the knots are not always discrete values.

vii. When the knot spam is of the following formulation $\mathbf{U} = \left\{ \underbrace{0, \dots, 0}_{p+1\ terms}\ ,\ \underbrace{1, \dots, 1}_{p+1\ terms} \right\}$ then concludes to Berstein Polynomials of p degree in Bezier Curves.

viii. $N_{i,p}(\xi) = 0$ when u is out of $[\xi_i, \xi_{i+p+1})$ according to local support property.

ix. In a given knot span $[\xi_i, \xi_{i+p+1})$ a maximum of $p + 1$ base functions $N_{i,p}$ are not equal to zero and particularly $N_{j-p,p} \dots N_{j,p}$

x. $N_{i,p}(\xi) \geq 0$ for every i, p and u (non negative).

xi. It is proved that $\sum_{i-p}^{i} N_{j,p}(u) = 1 \quad \forall\ \xi \in [\xi_i, \xi_{i+1})$ for a random knot span (Partition of unity)

xii. All the derivatives of $N_{i,p}(\xi)$ appear in the interior of a knot span in which they are polynomials. In a particular knot, $N_{i,p}(\xi)$ can be differentiated $p - k$ times where k is the multiplicity of the knot.

xiii. Apart from the case where $p = 0$, $N_{i,p}(\xi)$ receives exactly a maximum value.

### 4.3.3 Derivatives of B-Splines

It is proved that the first derivative of B-Spline curve can be evaluated as:

$$N'_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{4.3.5}$$

Let $N^{(k)}{}_{i,p}(\xi)$ be the k-order's derivative of $N_{i,p}(\xi)$; After repeated differentiations we conclude to the following general formulation:

$$N^{(k)}{}_{i,p}(\xi) = p(\frac{N^{(k-1)}{}_{i,p-1}}{\xi_{i+p} - \xi_i} - \frac{N^{(k-1)}{}_{i+1,p-1}}{\xi_{i+p+1} - \xi_{i+1}}) \tag{4.3.6}$$

An alternative way to compute the k-order's derivative as function of $N_{i,p-k}, \dots, N_{i+k,p-k}$ is the following:

$$N^{(k)}{}_{i,p} = \frac{p!}{(p-k)!} \sum_{j=0}^{k} a_{k,j} N_{i+j,p-k} \tag{4.3.7}$$

where,

$$a_{0,0} = 1$$

$$a_{k,0} = \frac{a_{k-1,0}}{\xi_{i+p-k+1} - \xi_i}$$

$$a_{k,j} = \frac{a_{k-1,j} - a_{k-1,j-1}}{\xi_{i+p-k+1} - \xi_{i+j}}, j = 1, \dots, k-1$$

$$a_{k,k} = \frac{-a_{k-1,k-1}}{\xi_{i+p+1} - \xi_{i+k}}$$

## 4.4 Flowchart

The flowchart that describes the optimization methodology of the present thesis is the one presented in figure 4.4.1. It consists of two equivalent options which are the following. It is possible to either start by modelling the pinion's flank or the pinion's rack gear flank. Our choice depends on whether the contact path is closed or open. In closed contact path the rack gear is modeled while in open contact path the pinion's flank is modeled. Both choices are equivalent; However, in closed contact path it is easier to determine the position of the control points for the rack gear.

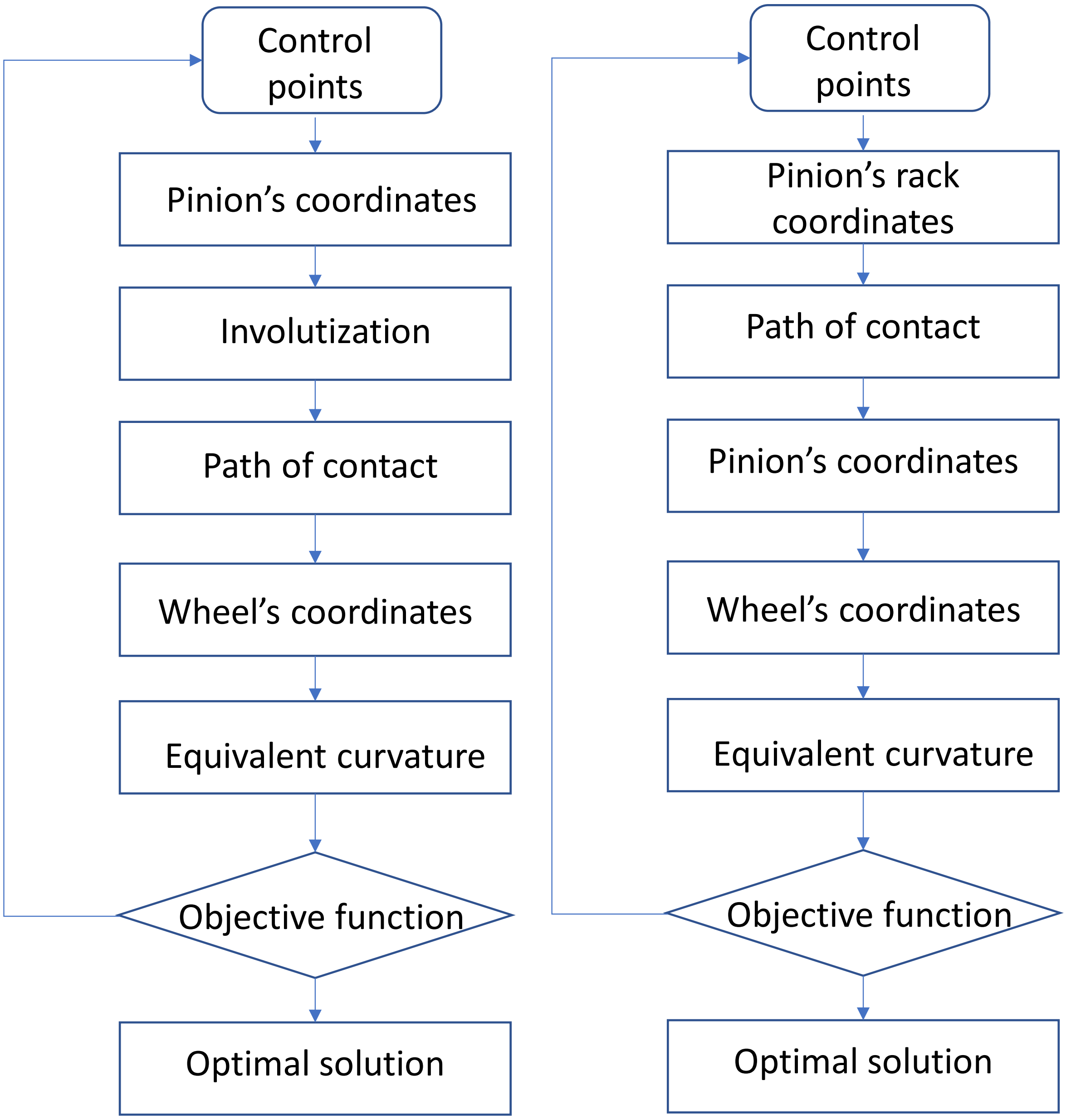


**Figure 4.4.1. Flowchart of the optimization process**

# 5. Optimizing gear flanks in closed contact path

## 5.1 General and constraints

Point of contact is defined as any point where the two working profiles intersect while the set of these points is defined as path of contact. The path of contact could be either closed which is discussed in the present chapter or open which is discussed in chapter 6. One example of closed contact path is the contact path of sinusoidal gears.

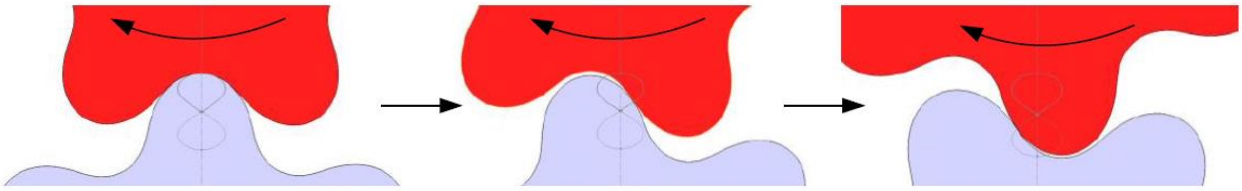


**Figure 5.1.1. Closed path of contact** [8]

However, this is a theoretical model since the meshing profiles not always mesh in closed contact path shown in the figure 5.1.1. It is true that they might mesh in points that do not belong to this closed contact path. The following figure shows how such a case could be:

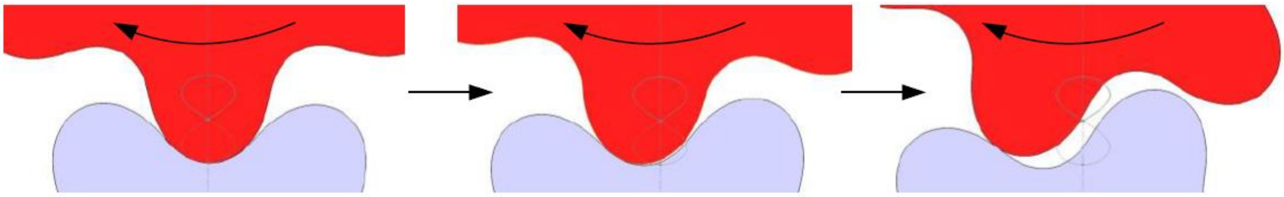


**Figure 5.1.2. Meshing out of the contact path [8]**

In the present thesis, we assume that the meshing gears always mesh in the defined closed contact path which means that the working profiles have no other contact points apart from it.

One of the most important steps in the optimization procedure is to define the constraints of the problem and the coordinate system. At first glance, it might seem easier to figure out the parameters of the problem when the tooth is in position 1 (Figure 5.1.3). A closer examination, though, reveals that this is not the best choice. Let $y = G(x)$ be the profile of the gear shown in position 1 of the following figure. That does not necessarily mean that every element of set $x$ corresponds to only one element of set $y = G(x)$ and the most common example that proves this is the involute profile.

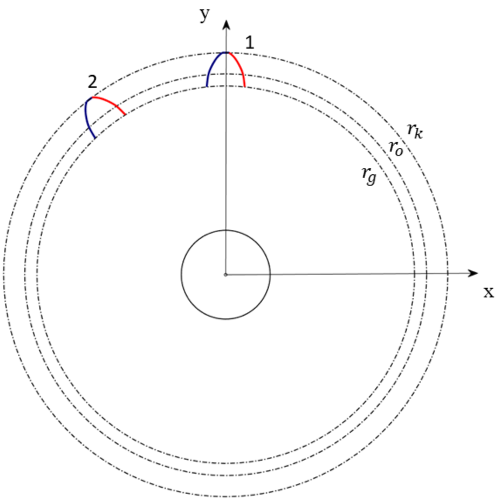


**Figure 5.1.3 Potential coordinate systems**

However, rotating the coordinate system by $-n\frac{2\pi}{\#tooth}$ rad (n=2, #tooth=19 in our case study) we observe that every element of set $x$ corresponds to only one element of set $y = G(x)$ with always ascending $x$ coordinate of working profile and always descending $y$ coordinate of working profile and also these are two of the restrictions imposed. In case these limitations are neglected it is possible that the pinion profile will be as presented in the next figure, a set of not ascending $x$ and not descending $y$:

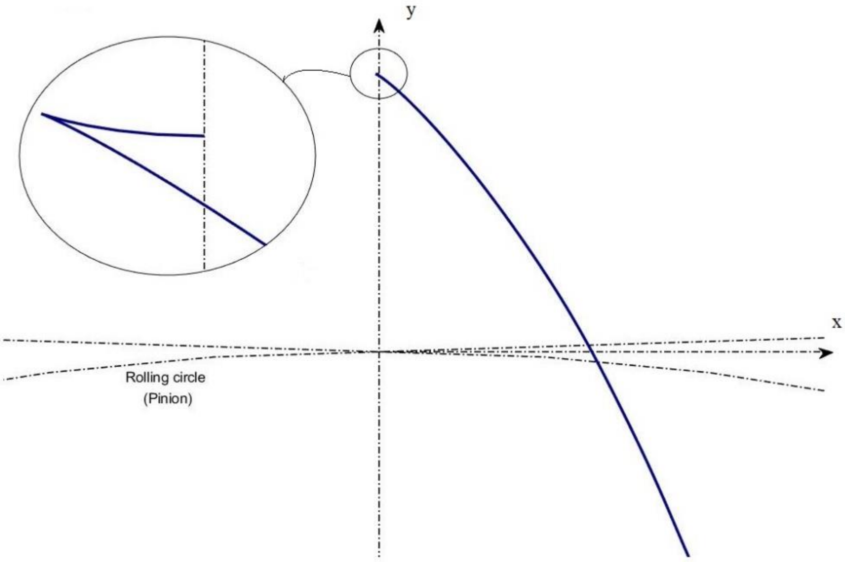


**Figure 5.1.4 Undesirable result when constraints are neglected**

Another constraint to be considered is the overlap ratio which has to be greater than 1. In closed contact path overlap ratio equals to 1 but in case of open contact path this factor should be taken into consideration.

One of the most common results, no matter if the problem is being solved with deterministic or genetic algorithms, is abrupt changes in equivalent curvature curve. In the figure below it is shown an undesirable steep change in the curve which corresponds in an abrupt change in stress. This is the main reason to cause pitting. Thus, such results in our attempt are immediately rejected.

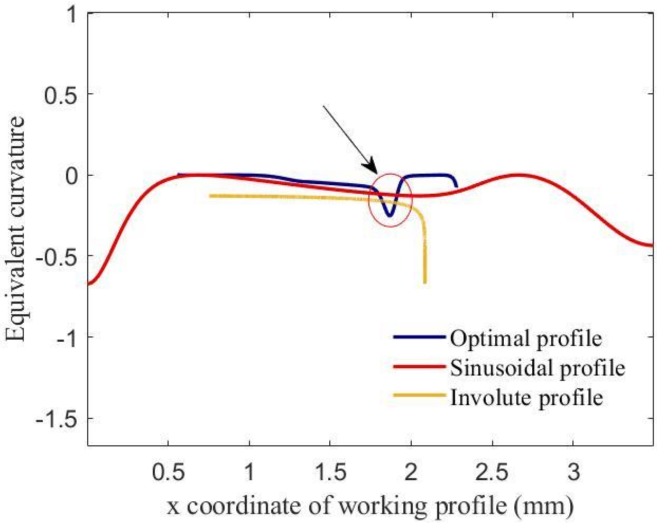


**Figure 5.1.5 Undesirable steep change in equivalent curvature curve**

Although it has this steep change in the curve, it has a better average equivalent curvature in comparison with sinusoidal and involute profile. This feature is a problem faced in both open contact path and closed contact path cases. However, there are ways to avoid such steep changes. This could be achieved with two kinds of terms in the objective function. The first is:

$$std\left(\frac{1}{R}\right) \tag{5.1.1}$$

and the second is:

$$\max\left(\left|\frac{1}{R}\right|\right) \tag{5.1.2}$$

Both these two terms are very efficient to confront abrupt changes in the equivalent curvature curve. Of course, sometimes the presence of weights is necessary to reach a better result.

## 5.2 Steepest Descent Results

Steepest descent algorithm of the present thesis consists an original code written for the purposes of this optimization problem. In contrast to genetic algorithm optimization of MATLAB-SIMULINK where the restrictions are imposed through MATLAB's optimization toolbox, the constraints are part of the original code. Nevertheless, there are several ways to do this such as to include them in the objective function. For example, in some of the following cases one of the objective function's terms is a weight multiplied by the inverse of overlap ratio (in open contact path cases) so as to exclude results with overlap ratio less than 1.

One of the advantages, among others, of modeling the rack gear profile is that the first two and the last two control points are in the same height. As a result, in an optimization problem of closed contact path with five control points results in a number of four objective variables $\{X_{CP2}, X_{CP3}, Y_{CP3}, X_{CP4}\}$ while the others are defined as far as gears parameters are designated (Figure 5.2.1).

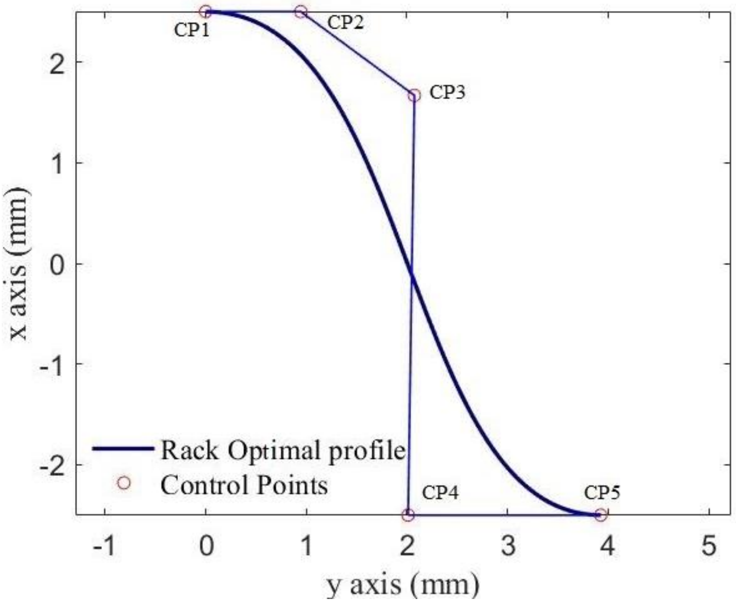


**Figure 5.2.1 Rack gear profile modeled with B-Spines**

| *Optimization parameters* | |
|---|---|
| Modeled with Control Points | Pinion's rack gear |
| Number of Control Points | 5 |
| Number of variables | 4 |
| Number of iterations | 100 |
| Tolerance -Golden Section- | $1e^{-8}$ |
| Number of searching steps | 100 |
| Objective function | $\max\left(abs\left(\frac{1}{R}\right)\right) + mean\left(abs\left(\frac{1}{R}\right)\right) + std\left(\frac{1}{R}\right)$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| *B-Spline parameters* | |
| Polynomial's degree | 4 |

**Table 5.2.1 Case 1 parameters**

In the following figure we have the path of contact of both sinusoidal and the optimal solution of steepest descent's algorithm. It is true that they have at least 3 points that these two contact paths intersect which in our case are $(0,0), (0,2.5)$ and $(0,-2.5)$ . The last two are directly related with module's value that is equal to 2.5.

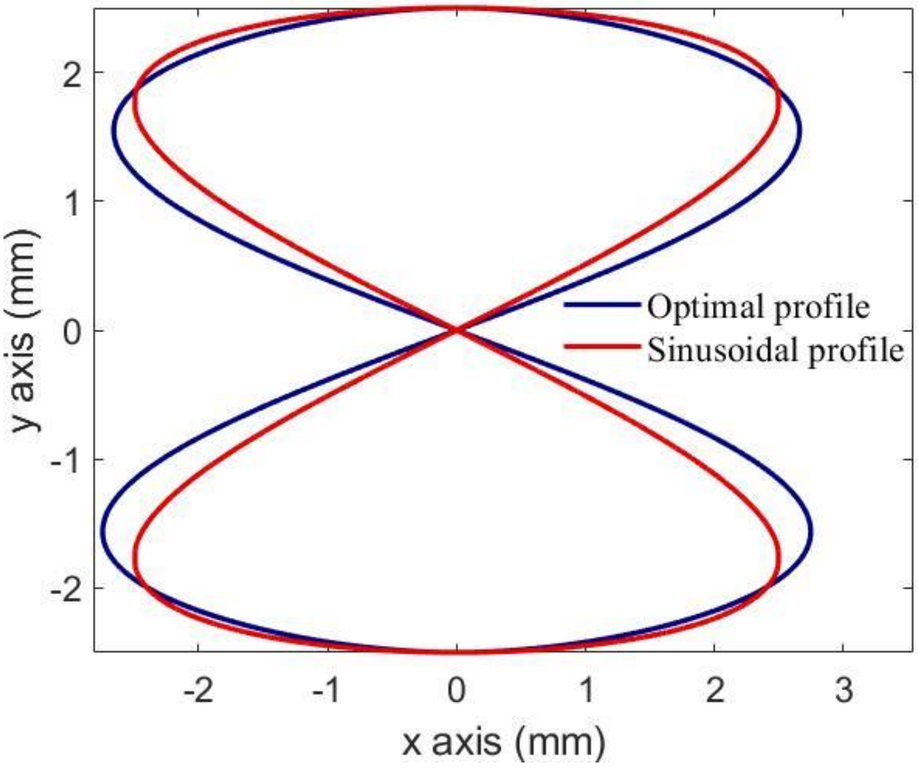


**Figure 5.2.2 Path of contact - case 1**

The working profiles that correspond to these paths of contact are shown in figure 5.2.3 They both have common start and end which was expected.

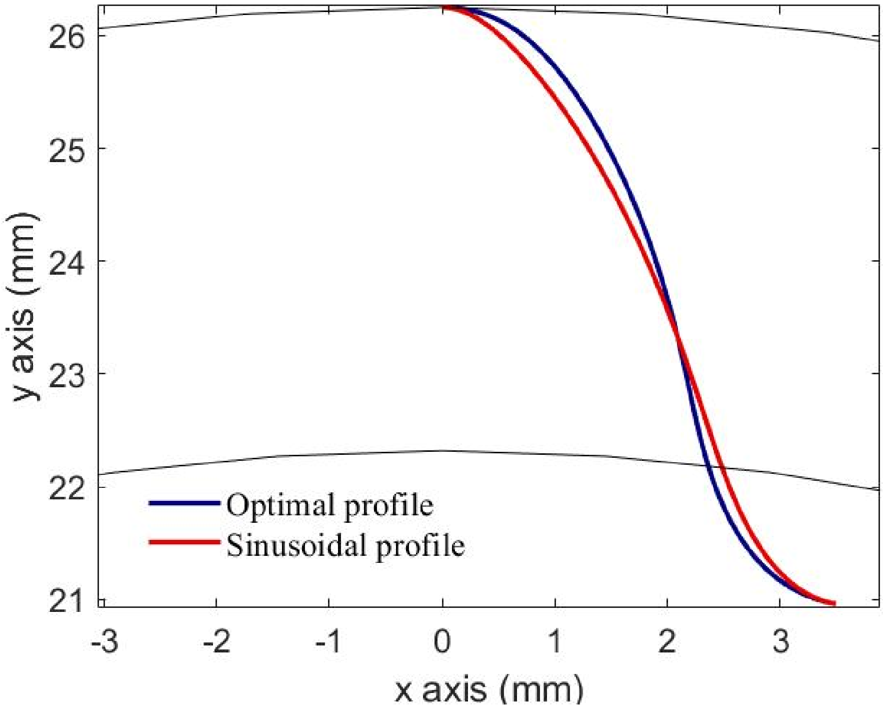


**Figure 5.2.3 Pinion working profiles - case 1**

Finally, the figure 5.2.4 shows our optimization aim. The equivalent curvature should be as stable as possible so as to avoid abrupt changes in stresses of the working profile. Although the results of optimal's profile in figure 5.2.4 are better that sinusoidal's profile, they are not satisfying enough due to the fact that the optimal curve is still wavy.

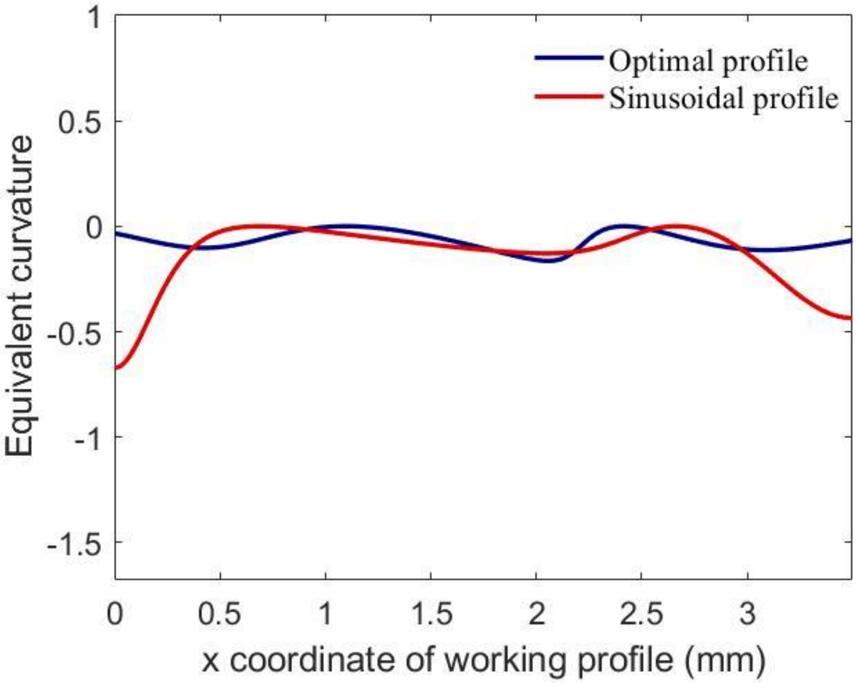


**Figure 5.2.4 Equivalent curvature - case 1**

The following table shows the some of the main factors that were used in order to optimize and assess the results of the problem. First, the average (mean) should be as smaller as possible because is directly related with the stress magnitude according to Hertz. Furthermore, the maximum of the equivalent curve indicates where the maximum stress could take place in the working profile; it was a very efficient term to use in the objective function because in association with the standard deviation of the equivalent curvature the results of the optimization become more stable.

| *Optimization Results (absolute values)* | | | |
|---|---|---|---|
| | Sinusoidal profile | Optimal profile | Decrease % |
| mean of equivalent curvature | 0.1333 | 0.0708 | 94.69 |
| max of equivalent curvature | 0.6719 | 0.1644 | 75.53 |
| std of equivalent curvature | 0.1542 | 0.0490 | 68.21 |

**Table 5.2.2 Case 1 results**

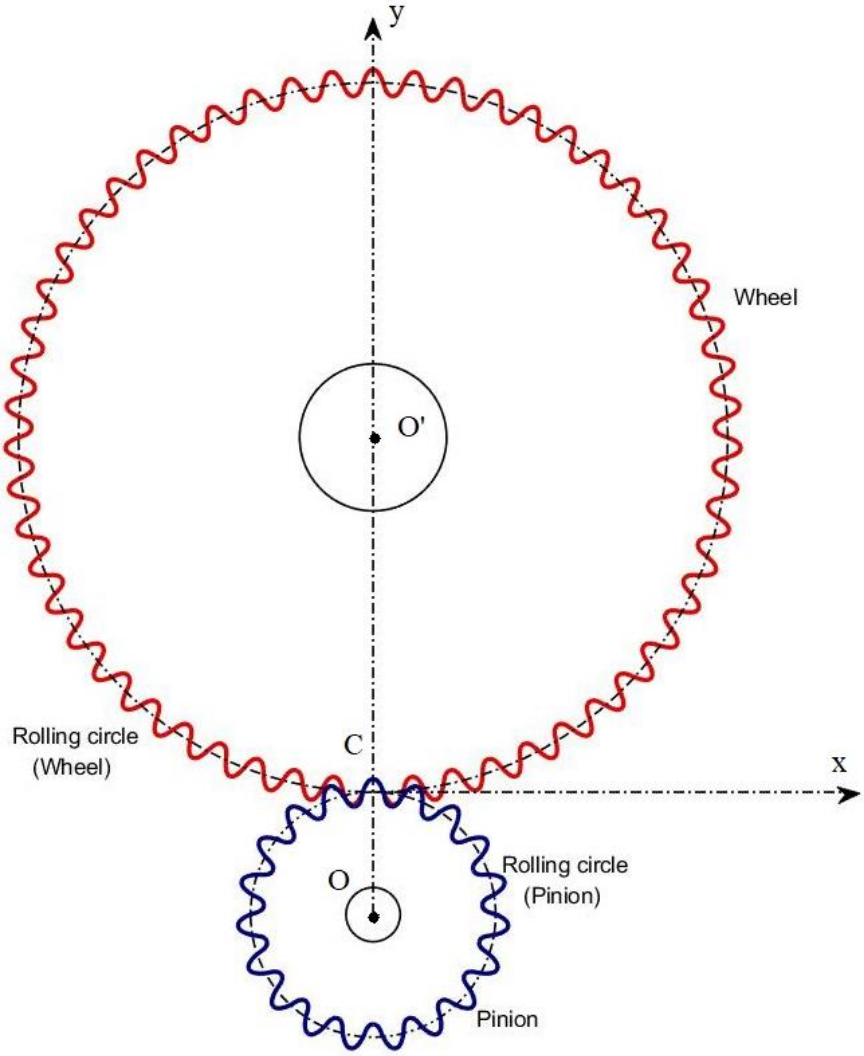


**Figure 5.2.5 Generating pinion and wheel - case 1**

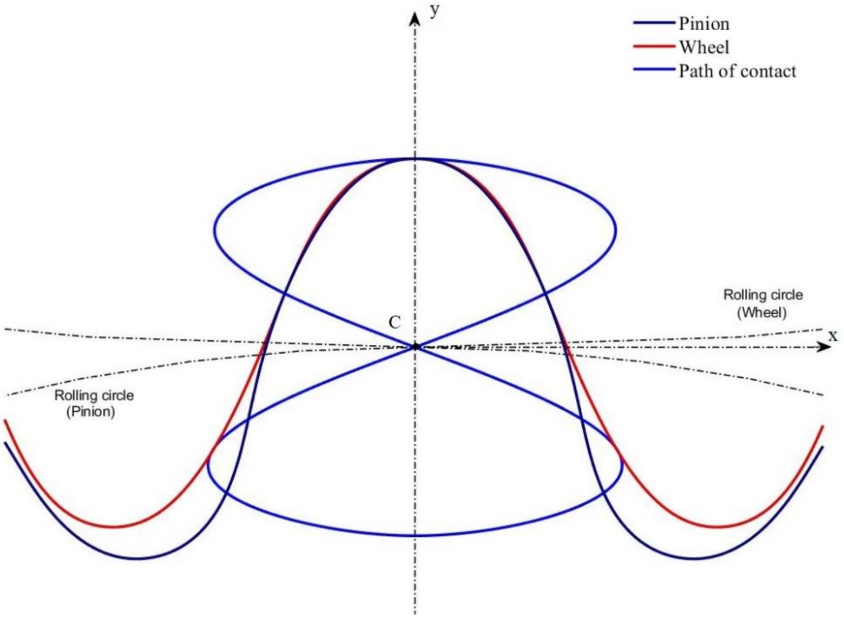


**Figure 5.2.6 Pinion and wheel in meshing - case 1**

As it was previously mentioned the equivalent curvature curve should be much more stable than the one in figure 5.2.4 Thus the same problem is being solved with genetic algorithm in chapter 5.3. Although the equivalent curvature curve is not as expected, it can be considered improved for closed contact path cases. The number of control points is another important factor to take into account and for this reason the following result is presented.

| *Optimization parameters* | |
|---|---|
| Modeled with Control Points | Pinion's rack gear |
| Number of Control Points | 8 |
| Number of variables | 10 |
| Number of iterations | 100 |
| Tolerance -Golden Section- | $1e^{-8}$ |
| Number of searching steps | 100 |
| Objective function | $\max\left(abs\left(\frac{1}{R}\right)\right) + mean\left(abs\left(\frac{1}{R}\right)\right) + std\left(\frac{1}{R}\right)$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| *B-Spline parameters* | |
| Polynomial's degree | 4 |

**Table 5.2.3 Case 2 parameters**

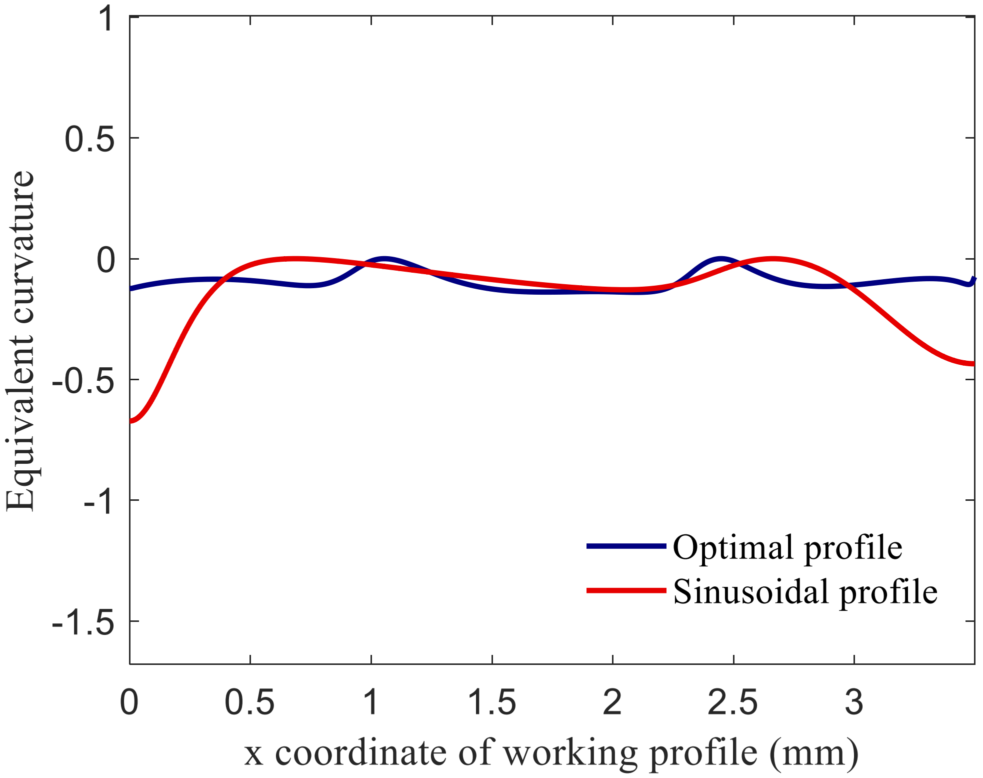


**Figure 5.2.7 Equivalent curvature - case 2**

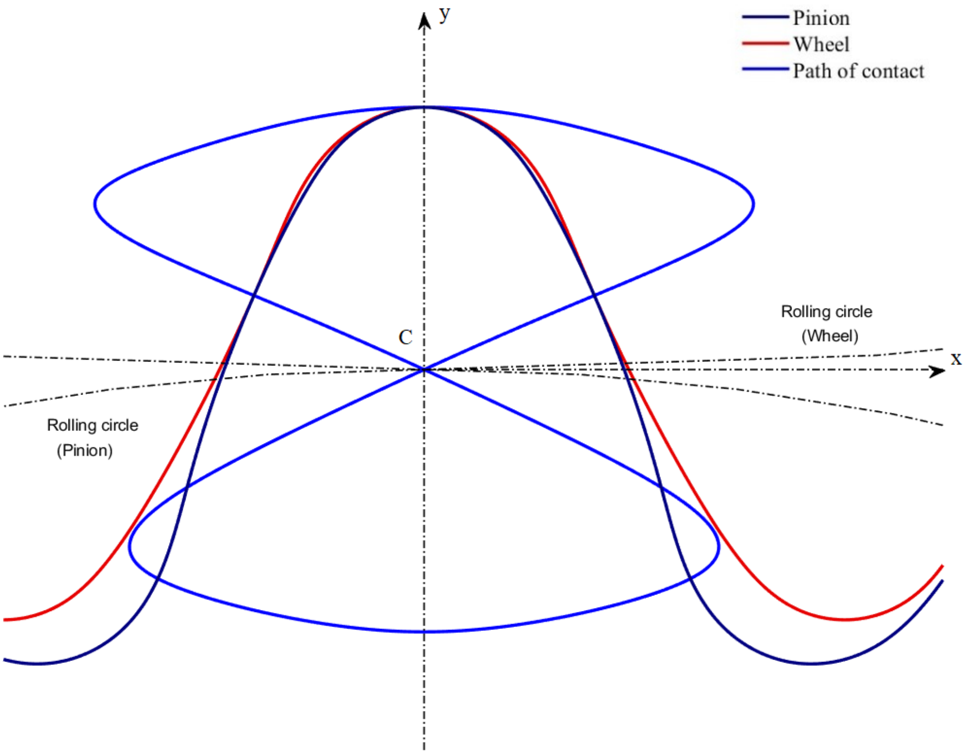


**Figure 5.2.8 Pinion and wheel in meshing - case 2**

| *Optimization Results (absolute values)* | | | |
|---|---|---|---|
| | Sinusoidal profile | Optimal profile | Decrease % |
| mean of equivalent curvature | 0.1333 | 0.0899 | 93.25 |
| max of equivalent curvature | 0.6719 | 0.1392 | 79.27 |
| std of equivalent curvature | 0.1542 | 0.0410 | 73.42 |

**Table 5.2.4 Case 2 results**

This result shows that the increased number of control points cannot assure the desirable improvement. Although the curve is not as equal as expected we observe reduction in the values in table 5.2.4 which shows that an improvement has been achieved. Between the first two cases the first is by far more stable than the second one.

## 5.3 Genetic Algorithm Results

Genetic algorithms are a credible way to reduce the equivalent curvature. However, every run of a genetic algorithm code could last from 5 to 45 minutes in contrast with steepest descent that needed less than 1 minute until convergence. It is true that the objective function is critical to reach satisfying results. In order to find the best objective function for the genetic algorithm often steepest descent was used to assess the objective function’s credibility.

| *Optimization parameters* | |
|---|---|
| Modeled with Control Points | Pinion’s rack gear |
| Number of Control Points | 7 |
| Number of variables | 9 |
| Population Size | 200 |
| Generation Size | 900 |
| Crossover | 0.4 |
| Objective function | $\max\left(abs\left(\frac{1}{R}\right)\right) + mean\left(abs\left(\frac{1}{R}\right)\right) + std\left(\frac{1}{R}\right)$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| *B-Spline parameters* | |
| Polynomial’s degree | 4 |

**Table 5.3.1 Case 3 parameters**

The path of contact shown in figure 5.3.1 still remains an eight-shape path which always intersects the pitch point $(0,0)$ .

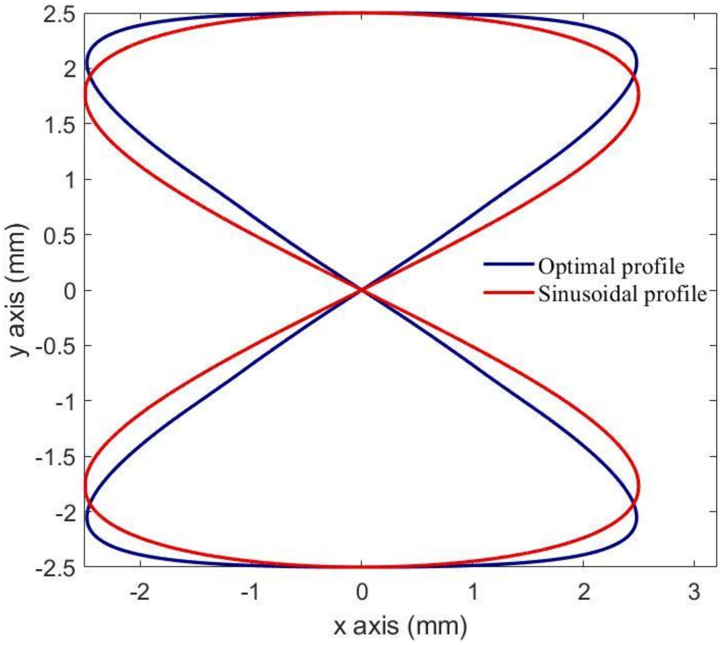


**Figure 5.3.1 Path of contact - case 3**

The pinion's working profile of the genetic algorithm's result is shown in figure 5.3.2. Again, as it was expected both curves have common start and end.

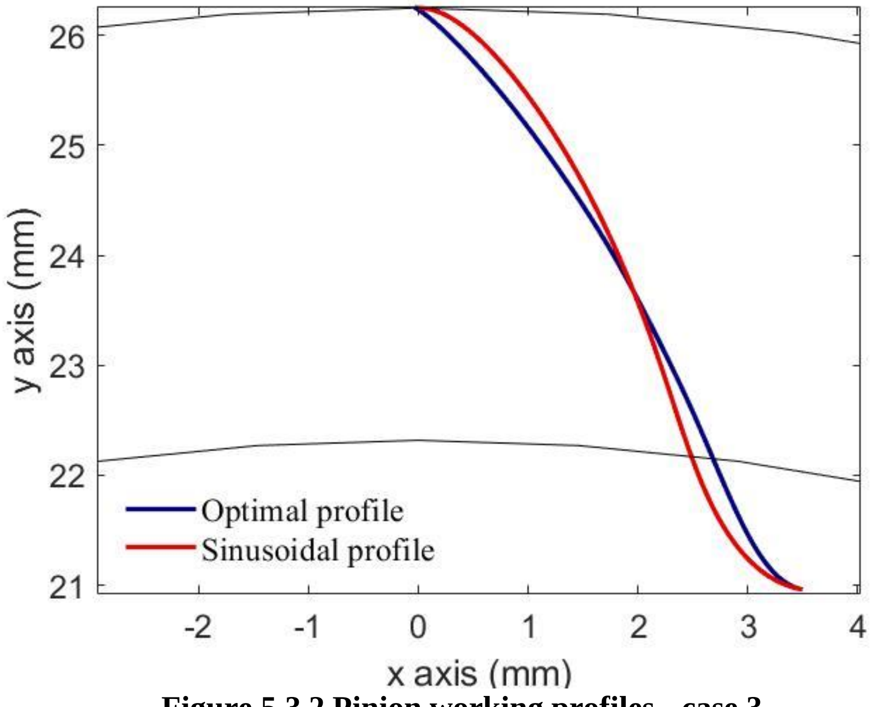


**Figure 5.3.2 Pinion working profiles - case 3**

The results shown in figure 5.3.3 are improved in comparison with the steepest descent's results previously shown. The equivalent curvature is not as wavy as presented before while it is much better than sinusoidal's profile.

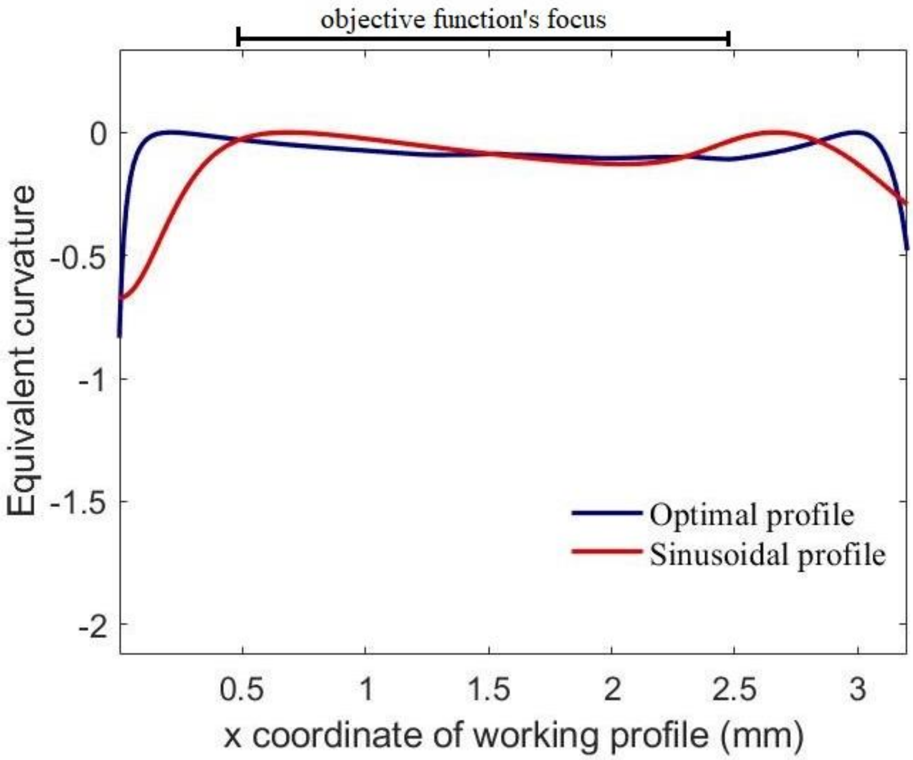


**Figure 5.3.3 Equivalent curvature - case 3**

The truth is that this was not the first result that the genetic algorithm converged. The whole process was very time-demanding given the fact that alternative objective functions were used. Apart from this, a certain objective function does not necessarily correspond only to a particular result. For example, an objective function might aim to optimize the whole array of equivalent curvature or just a part of it. In the figure above it is shown which part of the curve was optimized in the present result. It is worth to mention that even a small change in this area might have totally unexpected results.

| *Optimization Results (absolute values)* | | | |
|---|---|---|---|
| | Sinusoidal profile | Optimal profile | Decrease % |
| mean of equivalent curvature | 0.1333 | 0.0915 | 93.14 |
| max of equivalent curvature | 0.6719 | 0.1245 | 81.48 |
| std of equivalent curvature | 0.1542 | 0.0222 | 85.58 |

**Table 5.3.2 Case 3 results**

The factors of this optimization result are shown in the table above and are much better than the previous shown given the percentage decreases.

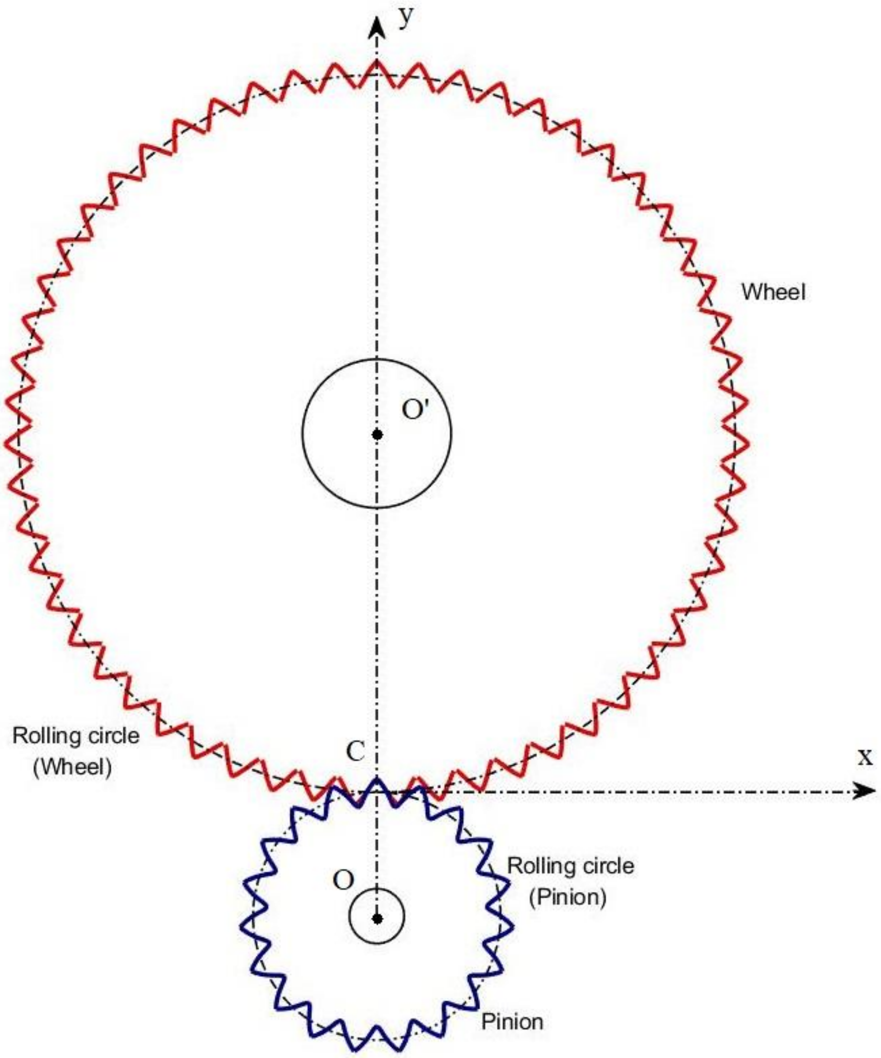


**Figure 5.3.4 Generating pinion and wheel - case 3**

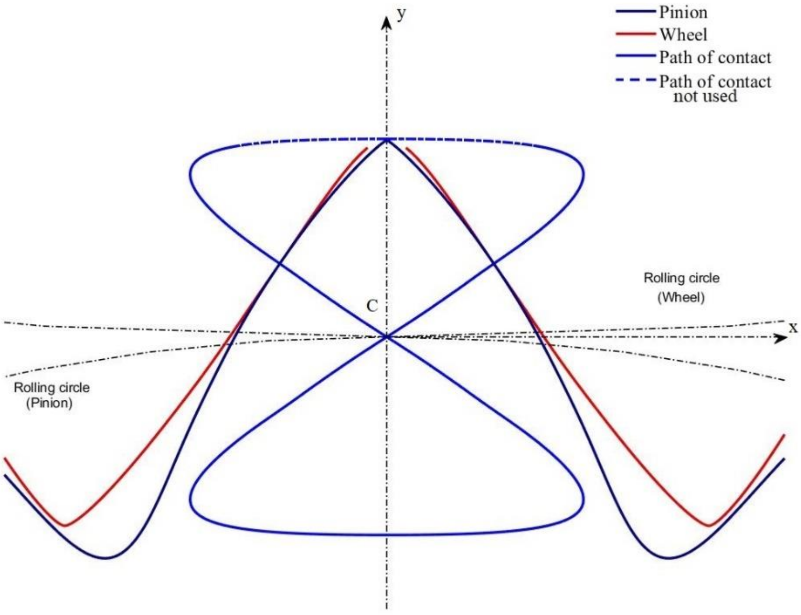


**Figure 5.3.5 Pinion and wheel in meshing - case 3**

Although the result above shows an improved equivalent curve in comparison with other genetic algorithm results for closed contact path it is characterized by the problem shown in figure 5.1.4. As a result, the it is only possible to mesh with the wheel only if a part of the meshing flank is used. For this reason, in the figure 5.3.5 part of the path of contact is intermittent. It is the part of contact that corresponds to the problematic part of the pinion's flank. Another genetic algorithm result, without this kind of problem is shown below.

| | |
|---|---|
| *Optimization parameters* | |
| Modeled with Control Points | Pinion's rack gear |
| Number of Control Points | 5 |
| Number of variables | 5 |
| Population Size | 300 |
| Generation Size | 900 |
| Crossover | 0.4 |
| Objective function | $\max\left(abs\left(\frac{1}{R}\right)\right)+mean\left(abs\left(\frac{1}{R}\right)\right)+std\left(\frac{1}{R}\right)$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| *B-Spline parameters* | |
| Polynomial's degree | 4 |

**Table 5.3.3 Case 4 parameters**

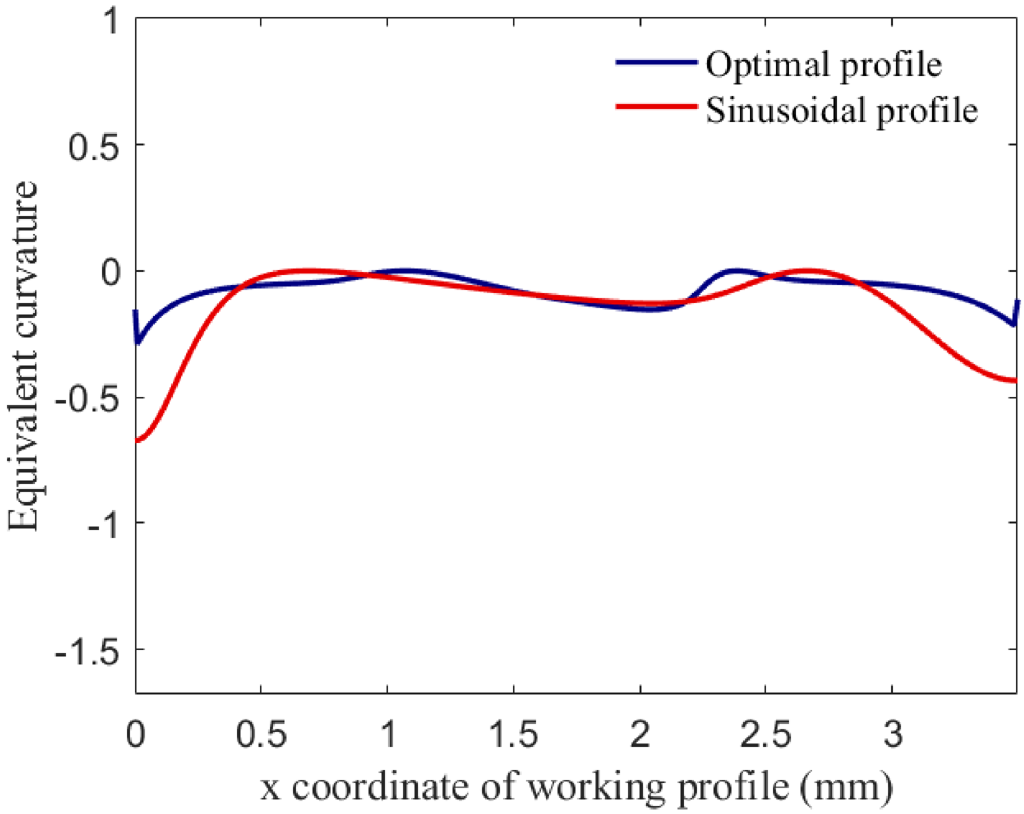


**Figure 5.3.6 Equivalent curvature - case 4**

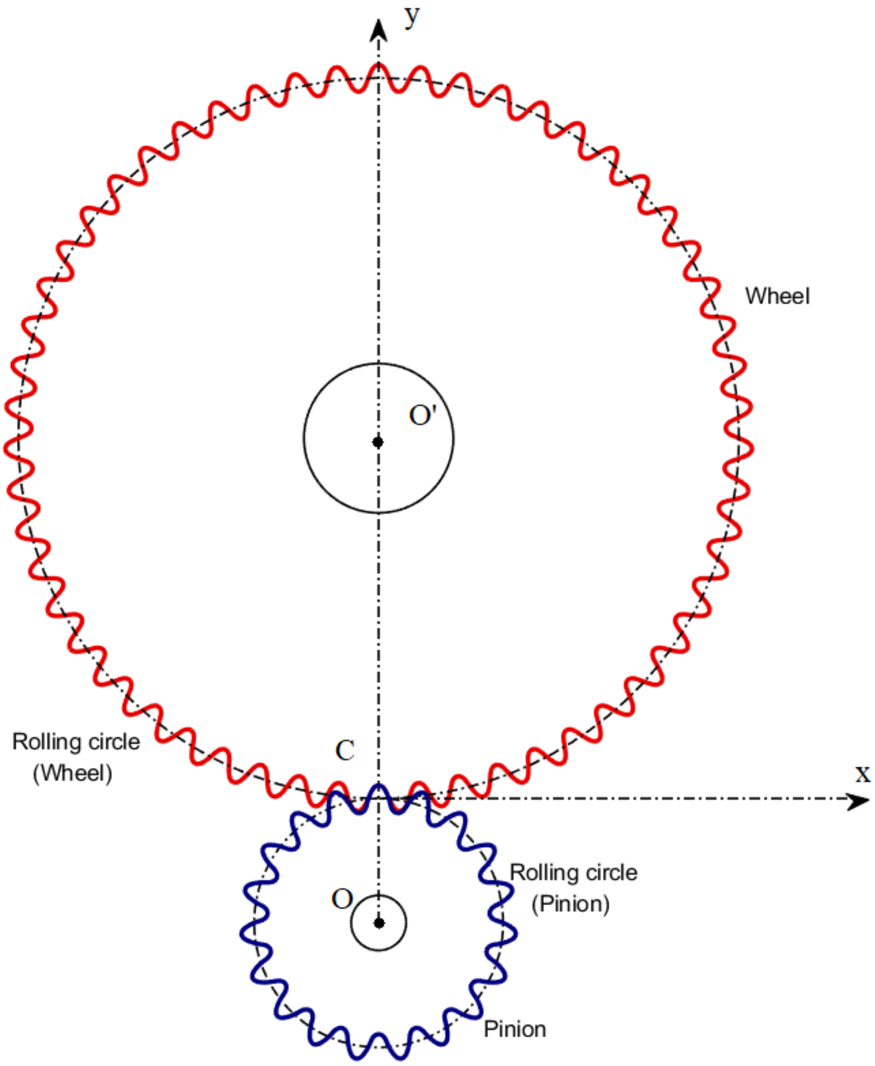


**Figure 5.3.7 Generating pinion and wheel - case 4**

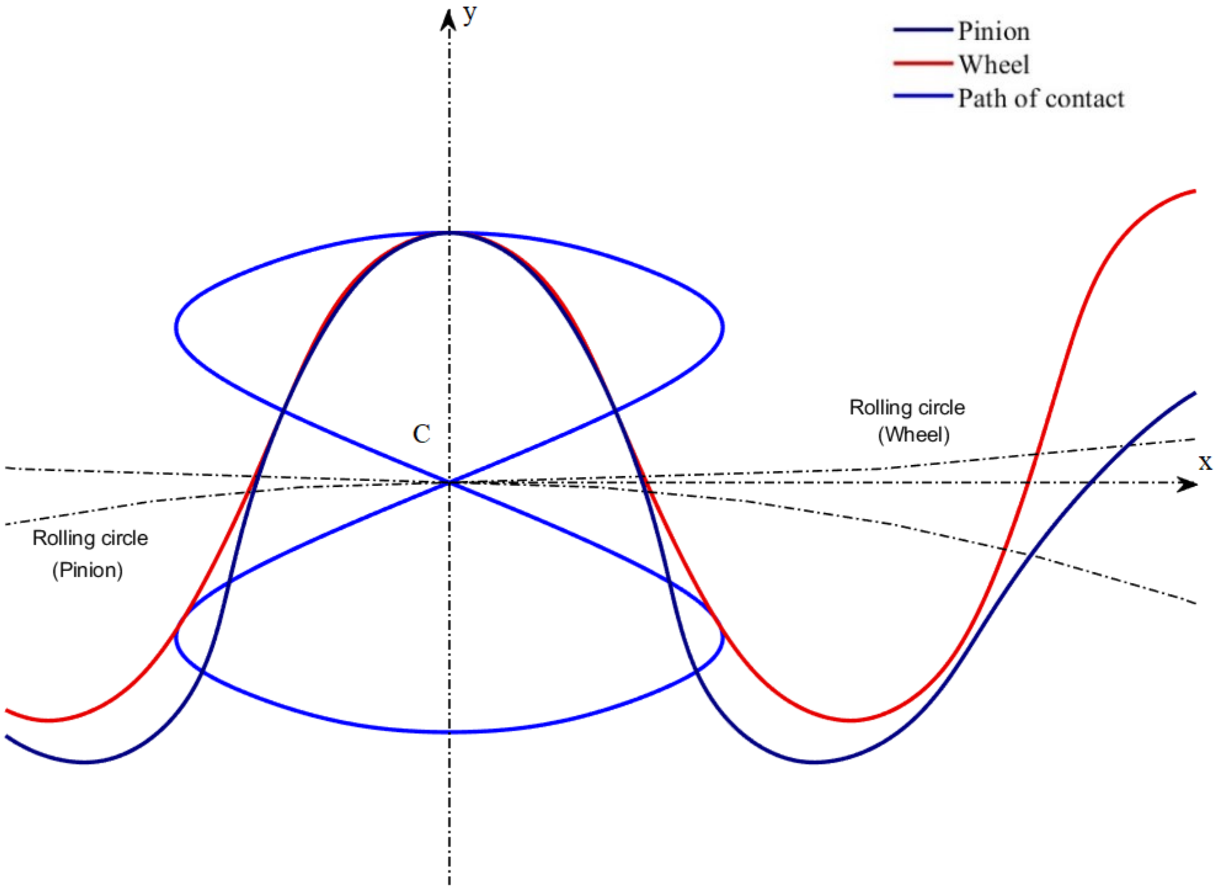


**Figure 5.3.8 Pinion and wheel in meshing - case 4**

| *Optimization Results (absolute values)* | | | |
|---|---|---|---|
| | Sinusoidal profile | Optimal profile | Decrease % |
| mean of equivalent curvature | 0.1333 | 0.0718 | 94.61 |
| max of equivalent curvature | 0.6719 | 0.1601 | 76.17 |
| std of equivalent curvature | 0.1542 | 0.0499 | 67.64 |

**Table 5.3.4 Case 4 results**

In this case we have a completely accurate pair of gears of closed contact path as shown in the figure 5.3.8. The table 5.3.4 shows the reduction of this case which is not as high as case 3 but still consists an interesting result with an average reduction of equivalent curvature around 95%, reduction 76% int the maximum value and reduction around 68% in the standard deviation.

# 6. Optimizing gear flanks in open contact path

## 6.1 General and constraints

In open contact path cases the control points model the pinion's profile instead of the rack gear's profile.

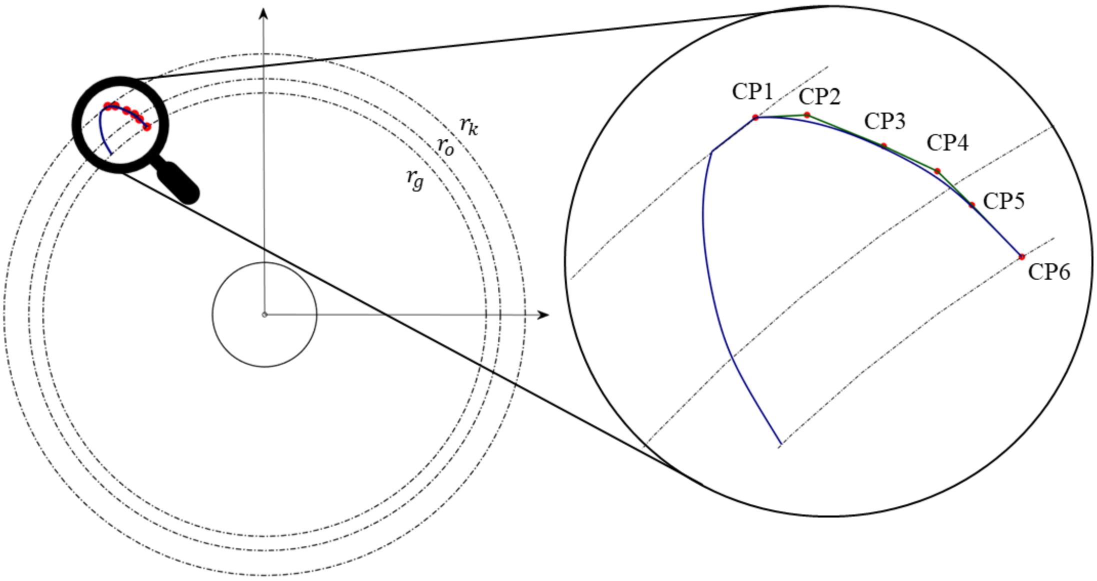


**Figure 6.1.1 Pinion's flank profile modeled with B-Spines**

In an optimization problem of open contact path with six control points results in a number of height objective variables $\{X_{CP1}, X_{CP2}, Y_{CP2}, X_{CP3}, Y_{CP3}, X_{CP4}, Y_{CP4}, X_{CP5}, Y_{CP5,} X_{CP6}\}$ while the others are defined as far as gears parameters are designated (figure 6.1.1). $Y_{CP1}$and $Y_{CP6}$ are not free dofs because they must satisfy the following equations:

$$\begin{gathered} X_{CP1}^2 + Y_{CP1}^2 = r_k^2 \\ X_{CP6}^2 + Y_{CP6}^2 = r_g^2 \end{gathered} \tag{5.3.1}$$

In open contact path cases we have to take into consideration the overlap ratio which must be greater than 1. Otherwise, we might end up in results that cannot be implemented in practice. For this reason, one of the objective's function terms is related to the overlap ratio.

Furthermore, the coordinates of the first and last control point have limitations too. When the pinion and wheel are generated, the flanks must not overlap. In order to understand how crucial is this factor the figure 6.1.3 presents the case where no constraints have been set for $x$ coordinate of the first control point. We observe that the $x$ coordinate of the first control point is negative and this is undesirable while it has a minimum and a maximum positive value related with $S_k$. The same condition must be reach for the last control point which has a minimum and a maximum value related with $S_g$. It is true that these restrictions refer to the flank when it is in position 1 figure 5.1.3. As a result, in order to impose these restrictions a rotation from position 2 to position 1 (figure 5.1.3) is needed in the optimization algorithm.

Notwithstanding these limitations an investigation was attempted so as to conceive the relation between the control points position and the equivalent curvature curve.

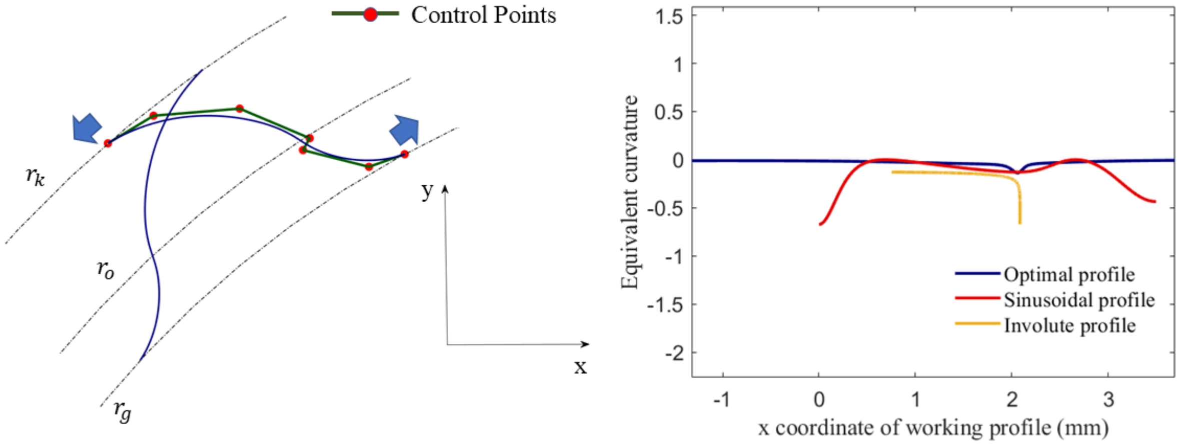


**Figure 6.1.2 Pinion's flank profile modeled with B-Spines with absence of overlap ratio from the objective function**

The equivalent curvature of this result is very close to zero and is quite stable which both are desirable. The main disadvantage of this result is that the profiles of the same tooth overlap and that the overlap ratio is lower than 1.

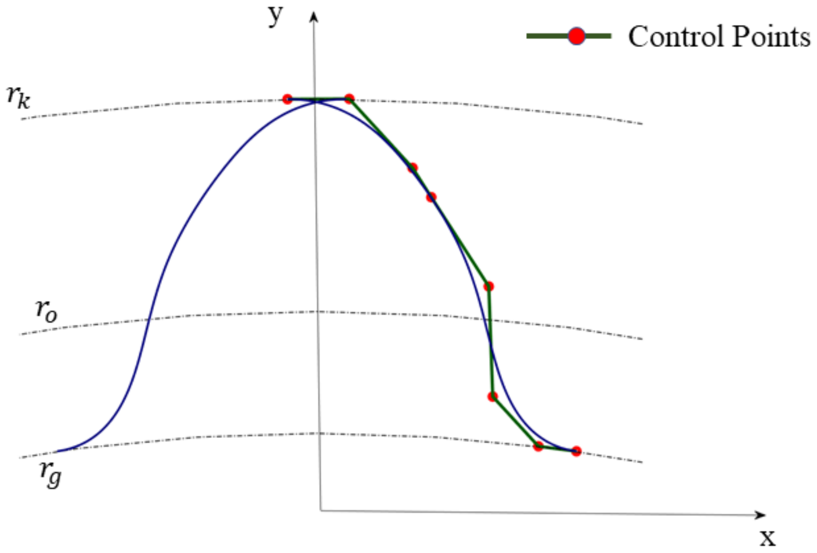


**Figure 6.1.3 Undesirable pinion's flank**

It is obvious that the result shown in figure 6.1.2 cannot be accepted. A closer examination though reveals that the first and the last control points have the tendency to move as the arrows shows in figure. This could encourage running the algorithm when only the first control point is designated, only the second control point is designated or both are designated. The results show that the best choice between the previous case studies is when all five (or more) control points are free to move. One of these results is presented in the next chapter.

## 6.2 Steepest Descent Results

Optimization algorithms of this kind need an initialization array so as to start the loop. In closed contact path the initialization usually was the sinusoidal profile while in open contact path the initialization was the involute profile.

The present result is of the most encouraging and satisfying among others because with only five control points and less than one minute to run the code the working profile is highly improved.

| *Optimization parameters* | |
|---|---|
| Modeled with Control Points | Pinion |
| Number of Control Points | 5 |
| Number of variables | 8 |
| Number of iterations | 100 |
| Tolerance (Golden Section) | $1e^{-8}$ |
| Number of searching steps | 100 |
| Objective function | $2.5\int\left(\frac{1}{R}\right)^2 dx + std\left(\frac{1}{R}\right) + 0.1\frac{1}{overlap\ ratio}$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| overlap | 1.04 |
| *B-Spline parameters* | |
| Polynomial's degree | 4 |

**Table 6.2.1 Case 5 parameters**

The figure below presents the contact paths of sinusoidal profile, involute profile and optimal profile. As it known, the involute contact path is a straight line with inclination $\tan(a_0)$ where $a_0$is the pressure angle. The optimal contact path is an open contact path which is not a straight line. All of them, though, intersect at the pitch point $(0,0)$.

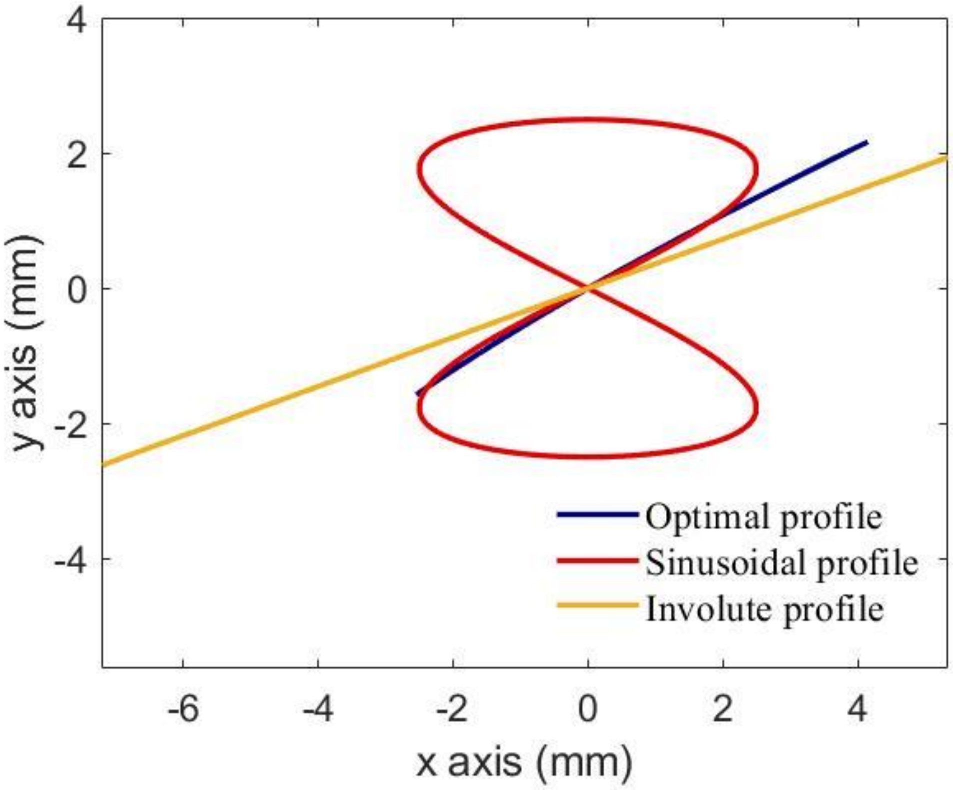


**Figure 6.2.1 Path of contact - case 5**

The working profile of this result is shown in figure 6.2.2 and it is compared with involute and sinusoidal profile.

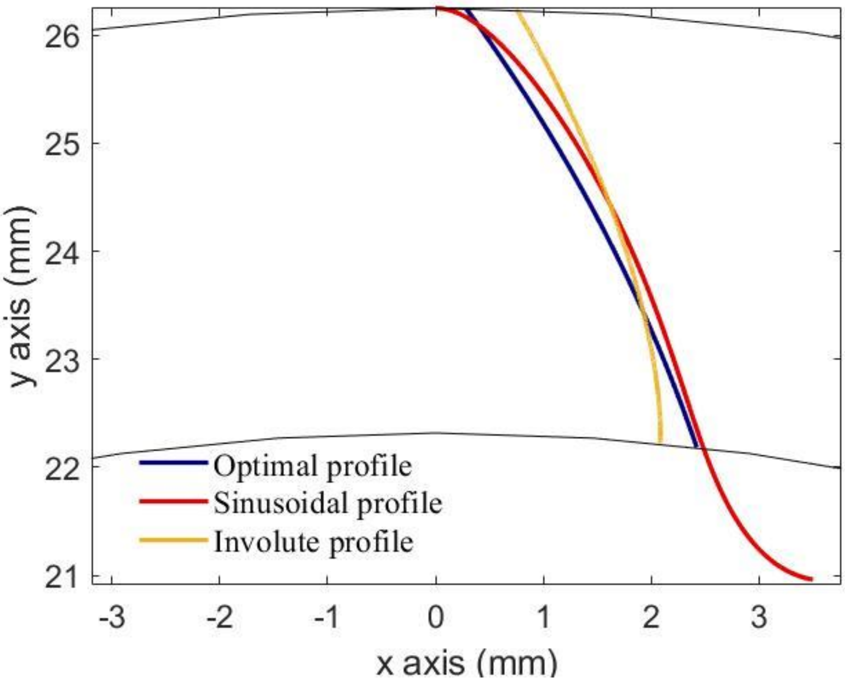


**Figure 6.2.2 Pinion working profiles - case 5**

The equivalent curvature of the present case study is one of the most stable curves among other results. It is permanently above involutes equivalent curvature and by far mor stable than sinusoidal's profile. This is obvious from the factors that are used to assess the curves in the optimization results table.

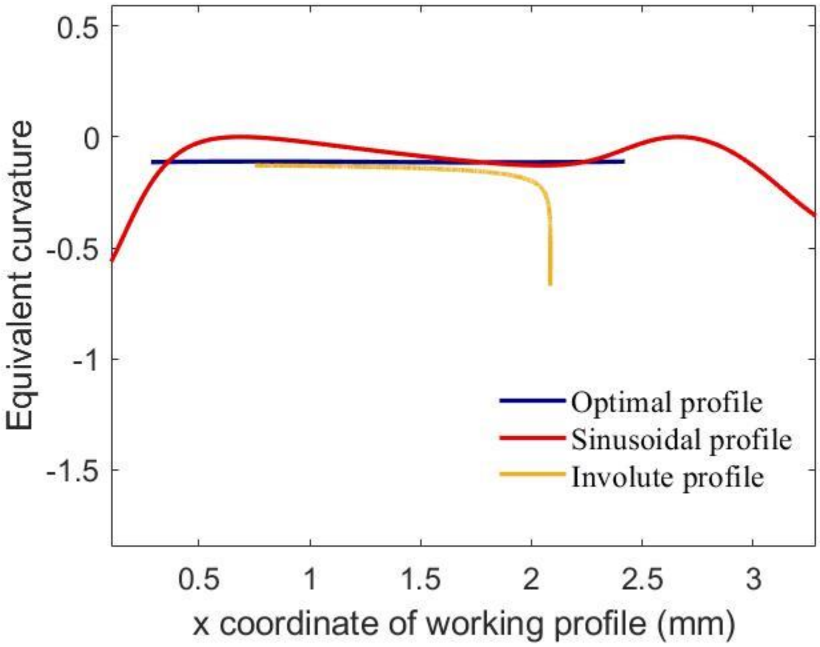


**Figure 6.2.3 Equivalent curvature - case 5**

| *Optimization Results (absolute values)* | | | | | |
|---|---|---|---|---|---|
| | Sinusoidal profile | Involute profile | Optimal profile | Decrease % (rel. to Sin) | Decrease % (rel. to Inv) |
| mean of equivalent curvature | 0.1333 | 0.1783 | 0.1130 | 91.52 | 36.62 |
| max of equivalent curvature | 0.6719 | 0.6708 | 0.1148 | 82.92 | 82.89 |
| std of equivalent curvature | 0.1542 | 0.0762 | 0.0014 | 99.10 | 98.18 |

**Table 6.2.2 Case 5 results**

The standard deviation of equivalent curvature is decreased 98% in comparison with involute which is generally much more stable curve than sinusoidal. This means that the optimal profile of this case study will have much less abrupt changes in stress magnitudes in surface fatigue.

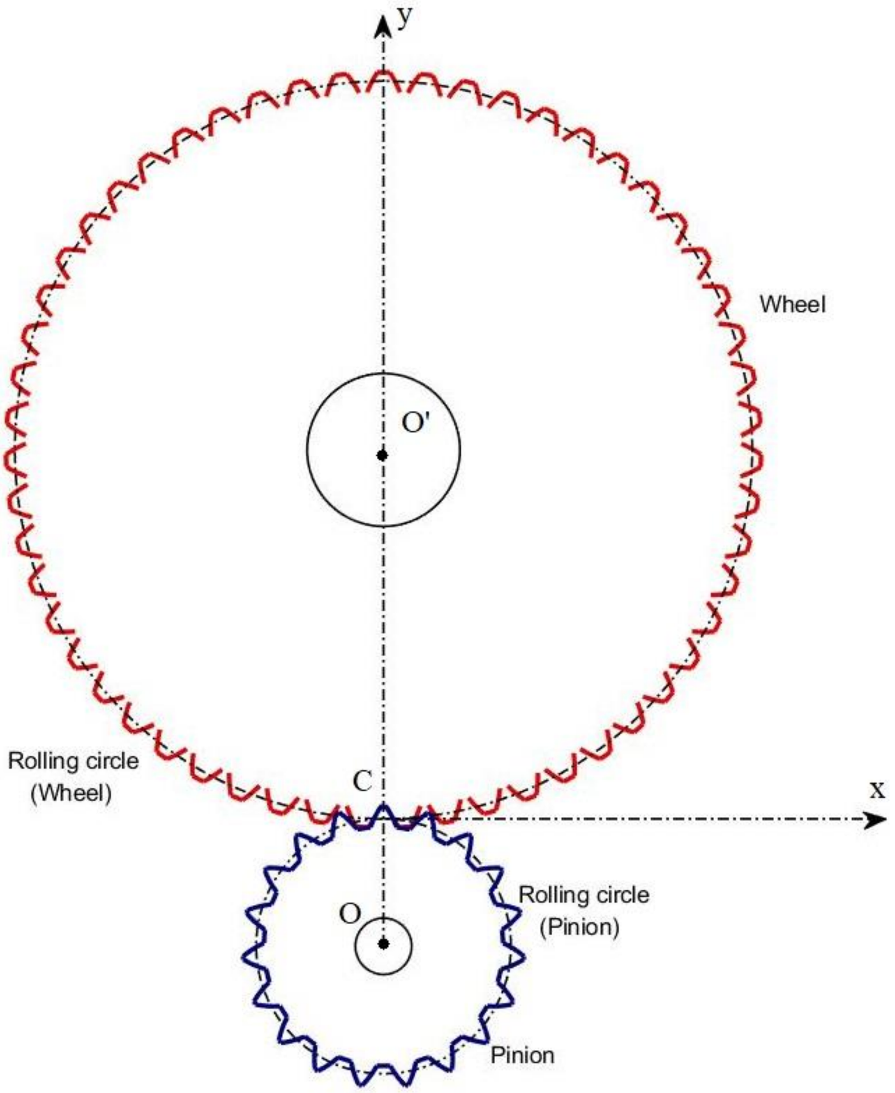


**Figure 6.2.4 Generating pinion and wheel - case 5**

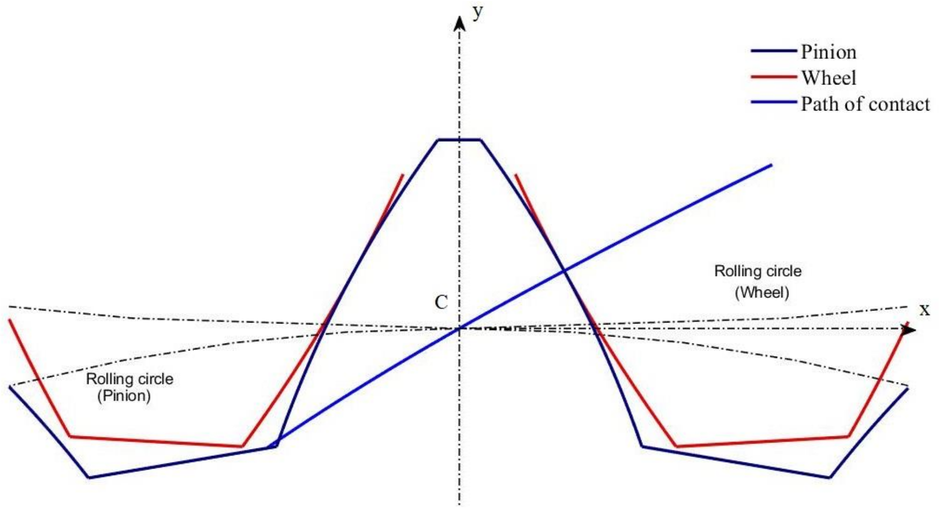


**Figure 6.2.5 Pinion and wheel in meshing - case 5**

One of the disadvantages of this result might be the low overlap ratio which is equal to 1.04. However, the fact that the equivalent curvature curve is almost a straight line encourages us to use the same objective function but with different weights in its terms. Particularly the following parameters correspond to another interesting result.

| *Optimization parameters* | |
|---|---|
| Modeled with Control Points | Pinion |
| Number of Control Points | 5 |
| Number of variables | 8 |
| Number of iterations | 100 |
| Tolerance (Golden Section) | $1e^{-8}$ |
| Number of searching steps | 100 |
| Objective function | $0.5\int\left(\frac{1}{R}\right)^{2}dx+0.5\,std\left(\frac{1}{R}\right)+0.1\frac{1}{overlap\ ratio}$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| overlap | 1.45 |
| *B-Spline parameters* | |
| Polynomial’s degree | 4 |

**Table 6.2.3 Case 6 parameters**

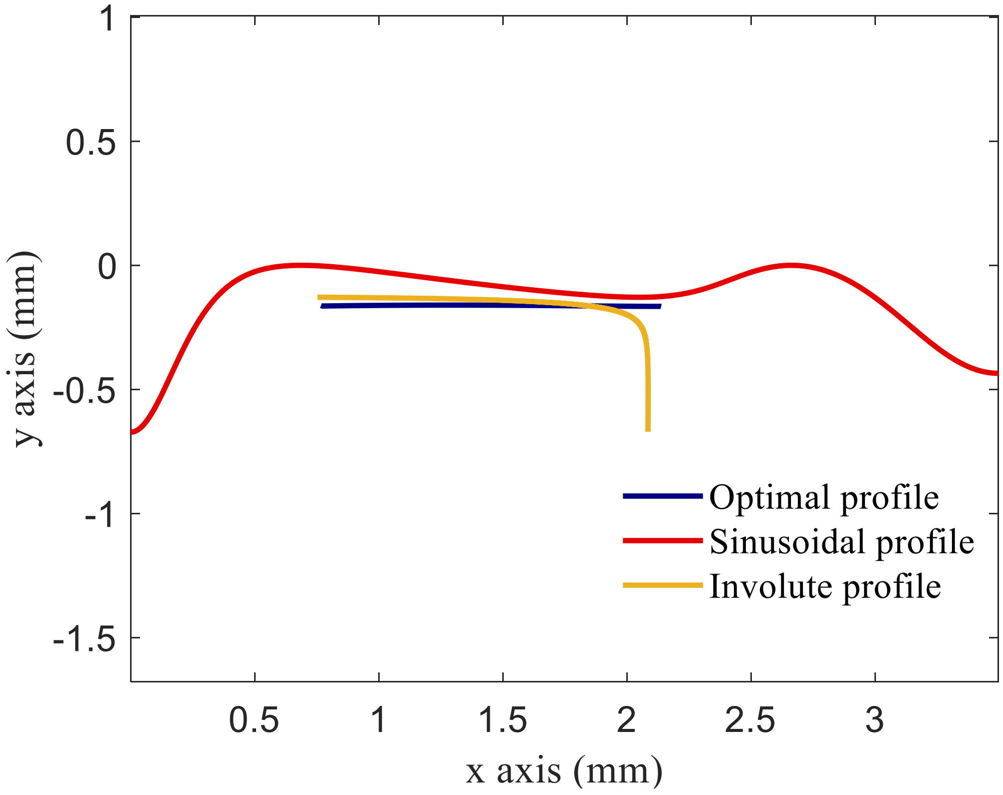


**Figure 6.2.6 Equivalent curvature - case 6**

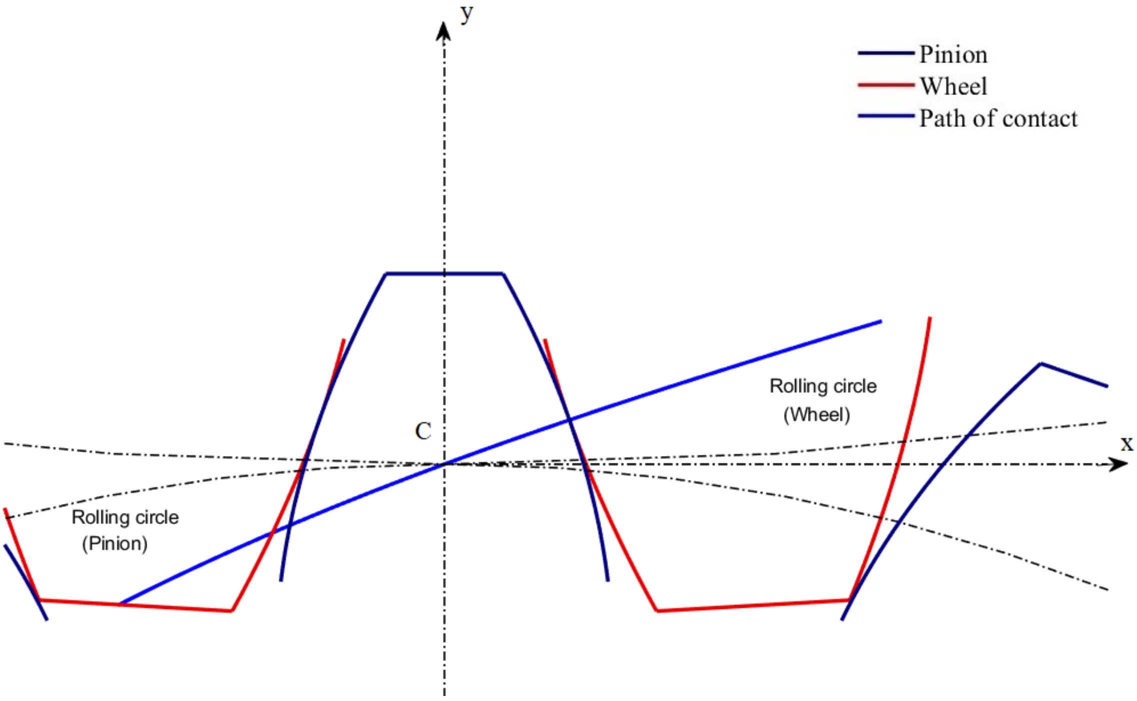


**Figure 6.2.7 Pinion and wheel in meshing - case 6**

| *Optimization Results (absolute values)* | | | | | |
|---|---|---|---|---|---|
| | Sinusoidal profile | Involute profile | Optimal profile | Decrease % (rel. to Sin) | Decrease % (rel. to Inv) |
| mean of equivalent curvature | 0.1333 | 0.1783 | 0.1627 | 87.80 | 8.73 |

| | | | | | |
|---|---|---|---|---|---|
| max of equivalent curvature | 0.6719 | 0.6708 | 0.1655 | 75.38 | 75.33 |
| std of equivalent curvature | 0.1542 | 0.0762 | 0.0018 | 98.84 | 97.64 |

**Table 6.2.4 Case 6 results**

By adding one more control point we observe that the equivalent curvature curve is not as straight as the previous results. To be specific we apply the following parameters and the result is shown below.

| *Optimization parameters* | |
|---|---|
| Modeled with Control Points | Pinion |
| Number of Control Points | 6 |
| Number of variables | 8 |
| Number of iterations | 100 |
| Tolerance (Golden Section) | $1e^{-8}$ |
| Number of searching steps | 100 |
| Objective function | $2.5\int\left(\frac{1}{R}\right)^2 dx + 1.4\, std\left(\frac{1}{R}\right) + 0.1\frac{1}{overlap\ ratio}$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| overlap | 1.22 |
| *B-Spline parameters* | |
| Polynomial’s degree | 4 |

**Table 6.2.5 Case 7 parameters**

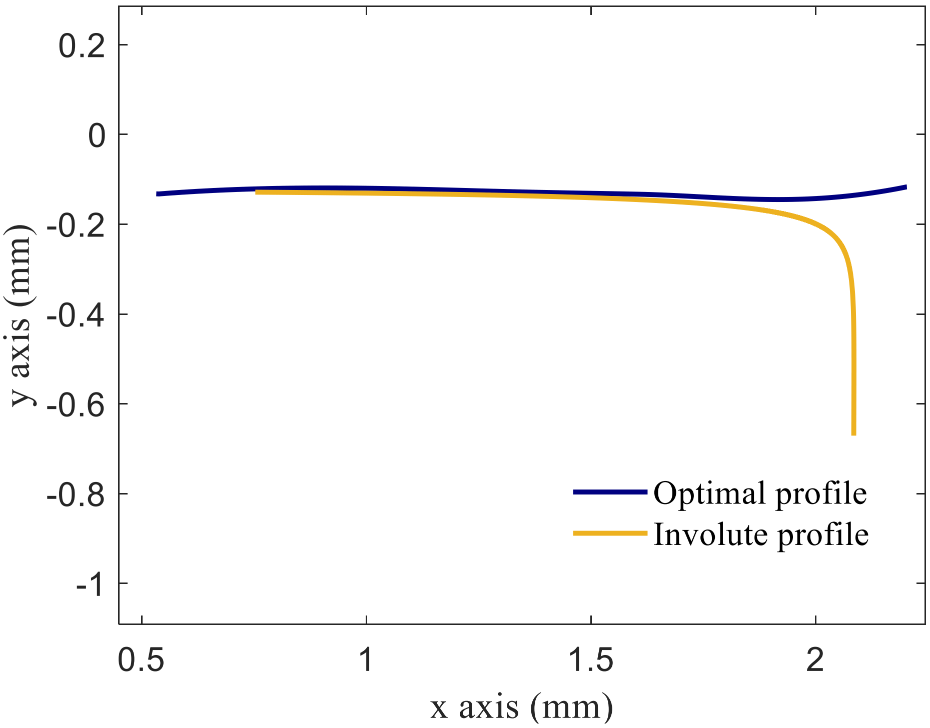


**Figure 6.2.8 Equivalent curvature - case 7**

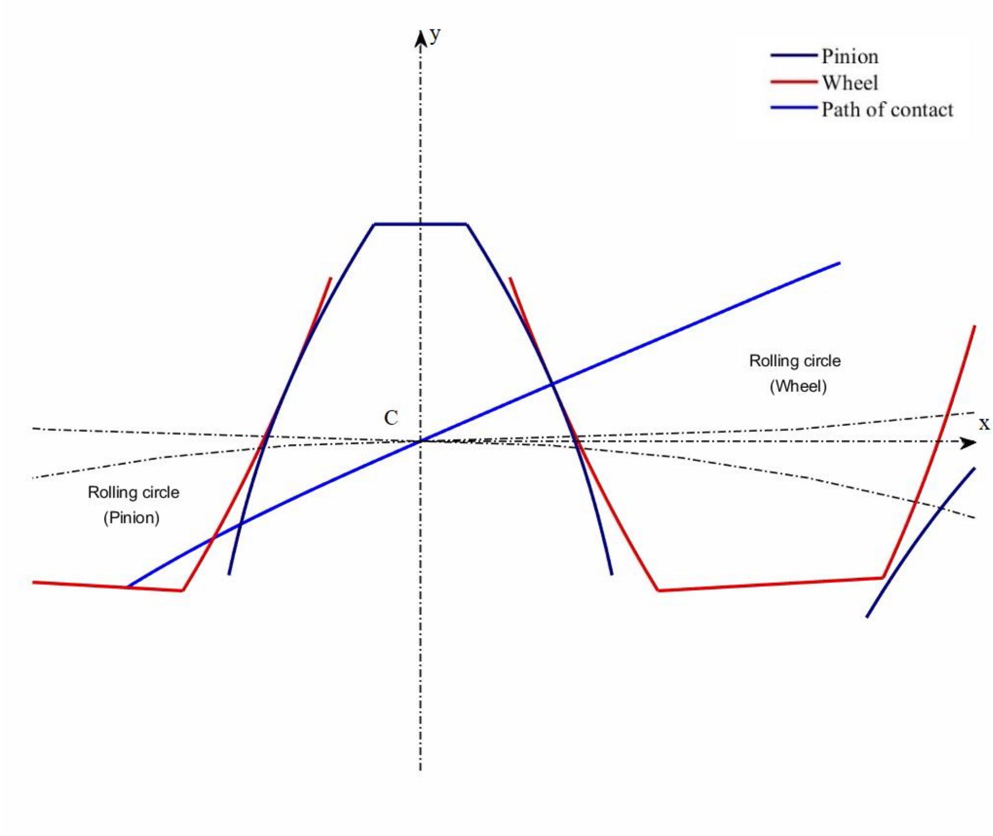


**Figure 6.2.9 Pinion and wheel in meshing - case 7**

| *Optimization Results (absolute values)* | | | | | |
|---|---|---|---|---|---|
| | Sinusoidal profile | Involute profile | Optimal profile | Decrease % (rel. to Sin) | Decrease % (rel. to Inv) |
| mean of equivalent curvature | 0.1333 | 0.1783 | 0.1312 | 90.15 | 26.38 |
| max of equivalent curvature | 0.6719 | 0.6708 | 0.1449 | 78.43 | 78.39 |
| std of equivalent curvature | 0.1542 | 0.0762 | 0.0085 | 94.50 | 88.87 |

**Table 6.2.6 Case 7 results**

Cases 5, 6 and 7 consist the best results in the present thesis given the fact that they have the highest percentage reduction in standard deviation in comparison with the others. The following chapter present results from genetic algorithm in MATLAB/SIMULINK.

## 6.3 Genetic Algorithm Results

In this case study the rack gear is modeled with 6 control points. The number of control points is of course another important factor to take into account. As it is shown in the following table the generation size is equal to 1900 in order to get an accepted result. The more the objective variables are the more factors such as population or generation size should become greater rendering the problem a very difficult task that could probably take hours to be solved. The parameters below are chosen so as to have an acceptable result converged in less than one hour.

| *Optimization Results (absolute values)* | |
|---|---|
| Modeled with Control Points | Pinion's rack gear |
| Number of Control Points | 6 |
| Number of variables | 8 |
| Population Size | 400 |
| Generation Size | 1900 |
| Crossover | 0.8 |
| Objective function | $\max\left(abs\left(\frac{1}{R}\right)\right)+mean\left(abs\left(\frac{1}{R}\right)\right)+std\left(\frac{1}{R}\right)$ |
| additional constraints | $\max\left(abs\left(\frac{1}{R}\right)\right)\leq 0.2$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| overlap | 1.14 |
| *B-Spline parameters* | |
| Polynomial's degree | 4 |

**Table 6.3.1 Case 8 parameters**

The first and the last control point of this case study are designated so the four control points are free to move in both axes. As a result, we have the following contact path which reminds in some of its first part the sinusoidal contact path and that is because the two control points that are defined (first and last) are in the same positions as the closed path case.

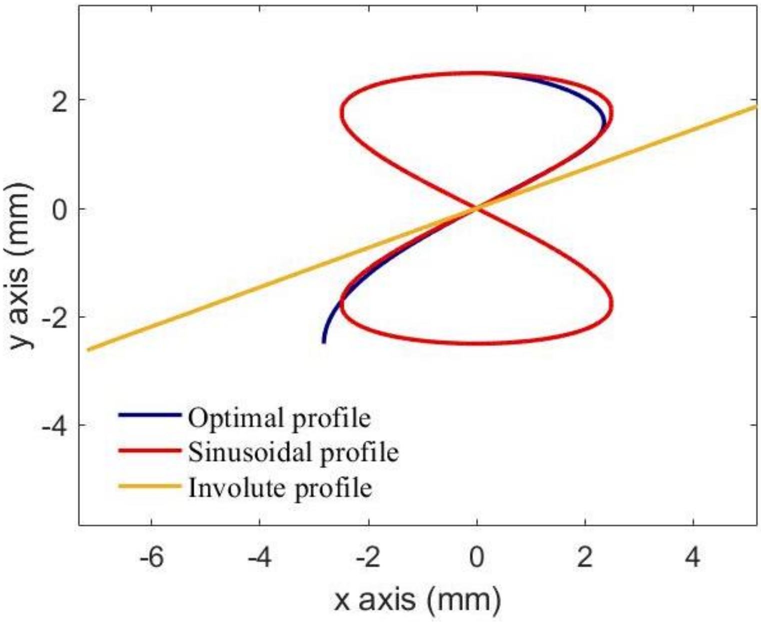


**Figure 6.3.1 Theoretical path of contact - case 8**

The working profiles of the present case study are shown in the figure below.

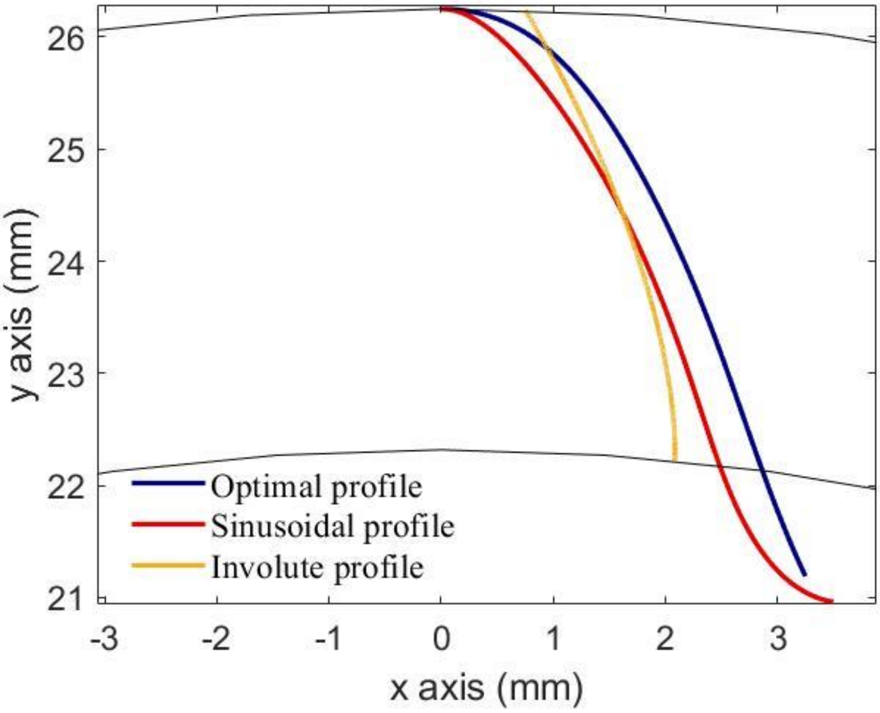


**Figure 6.3.2 Pinion working profiles - case 8**

The equivalent curvature is improved in comparison with the involute and sinusoidal profile but it is not better than the steepest descent's result. It is much above the involute curve however is not as stable as desired.

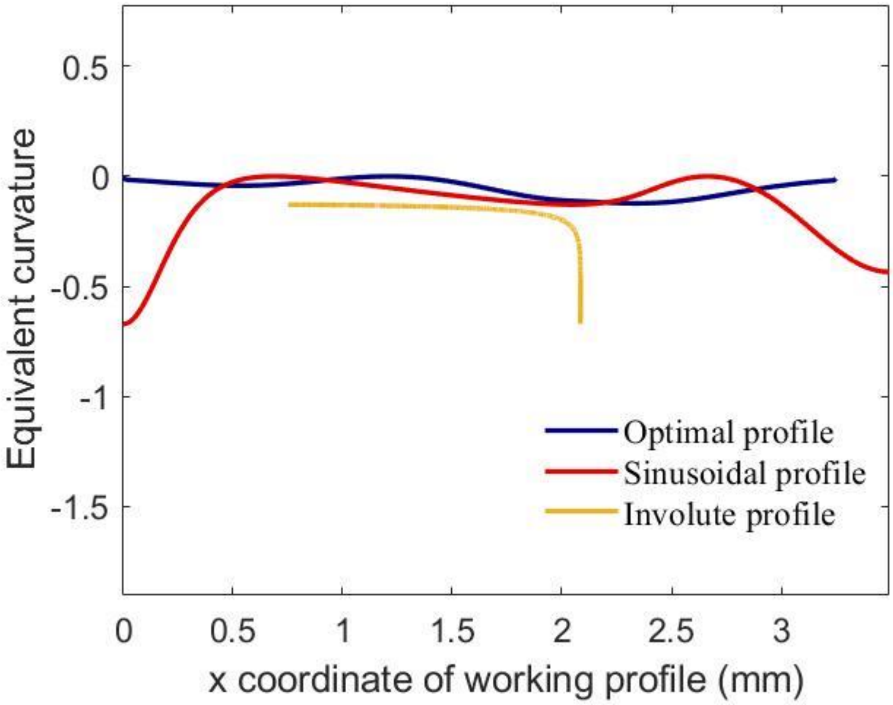


**Figure 6.3.3 Equivalent curvature - case 8**

| *Optimization Results (absolute values)* | | | | | |
|---|---|---|---|---|---|
| | Sinusoidal profile | Involute profile | Optimal profile | Decrease % (rel. to Sin) | Decrease % (rel. to Inv) |
| mean of equivalent curvature | 0.1333 | 0.1783 | 0.0649 | 95.13 | 63.58 |
| max of equivalent curvature | 0.6719 | 0.6708 | 0.1228 | 81.73 | 81.70 |
| std of equivalent curvature | 0.1542 | 0.0762 | 0.0416 | 73.05 | 45.44 |

**Table 6.3.2 Case 8 results**

In this case, a decrease around 64% is observed in the average of equivalent curvature, a decrease around 82% in the maximum value of the curve and a 45% decrease in standard deviation. While this is not the best solution among the other cases, it consists an important improvement in comparison with involute and sinusoidal profile.

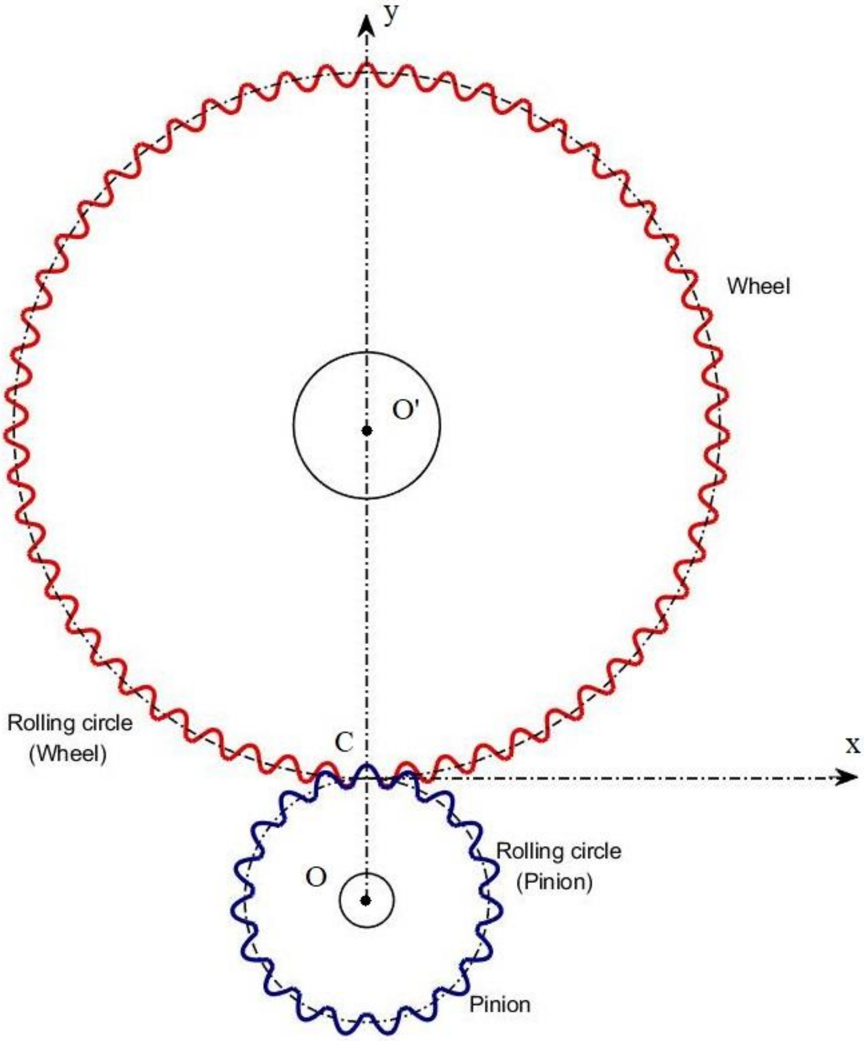


**Figure 6.3.4 Generating pinion and wheel - case 8**

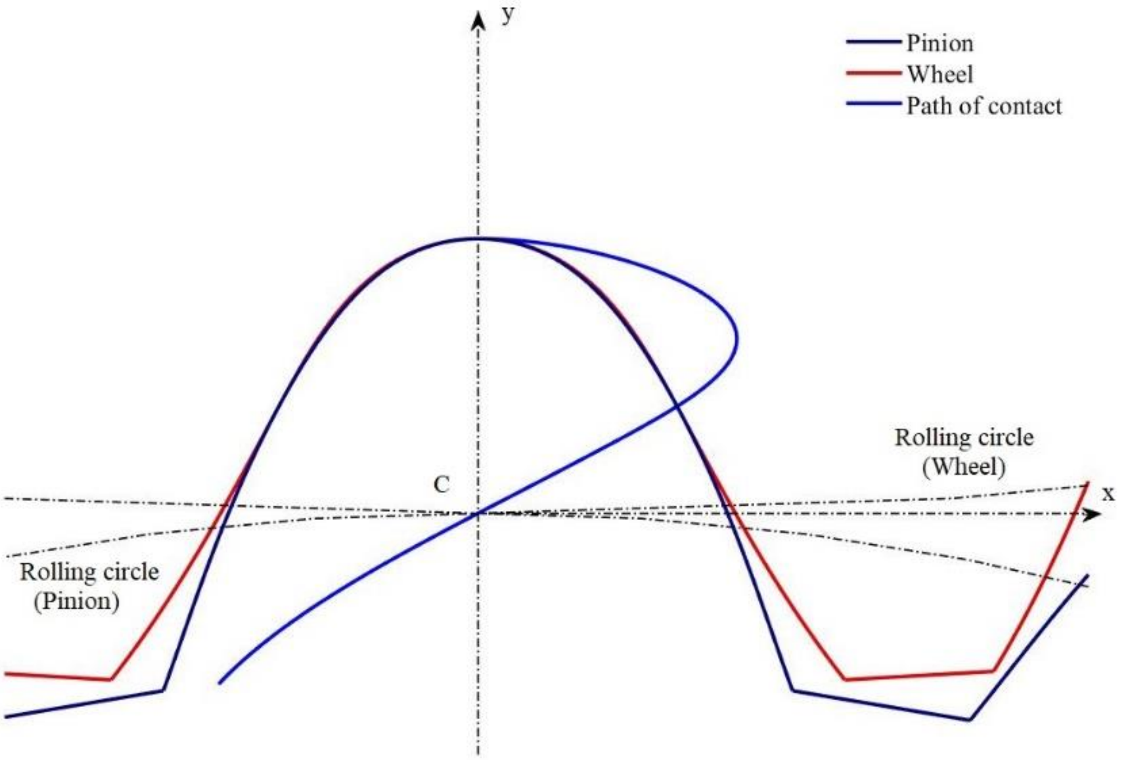


**Figure 6.3.5 Pinion and wheel in meshing - case 8**

Another effort is attempted by increasing the number of the control points and the generation size in order to ensure that these factors do not set limits to the optimization results. The outcome is not very different from what was presented above. However, two more cases are presented with the respective figures:

| *Optimization Results (absolute values)* | |
|---|---|
| Modeled with Control Points | Pinion's rack gear |
| Number of Control Points | 7 |
| Number of variables | 10 |
| Population Size | 400 |
| Generation Size | 2300 |
| Crossover | 0.8 |
| Objective function | $\max\left(abs\left(\frac{1}{R}\right)\right) + mean\left(abs\left(\frac{1}{R}\right)\right) + std\left(\frac{1}{R}\right)$ |
| additional constraints | $\max\left(abs\left(\frac{1}{R}\right)\right) \leq 0.2$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| overlap | 1.17 |
| *B-Spline parameters* | |
| Polynomial's degree | 4 |

**Table 6.3.3 Case 9 parameters**

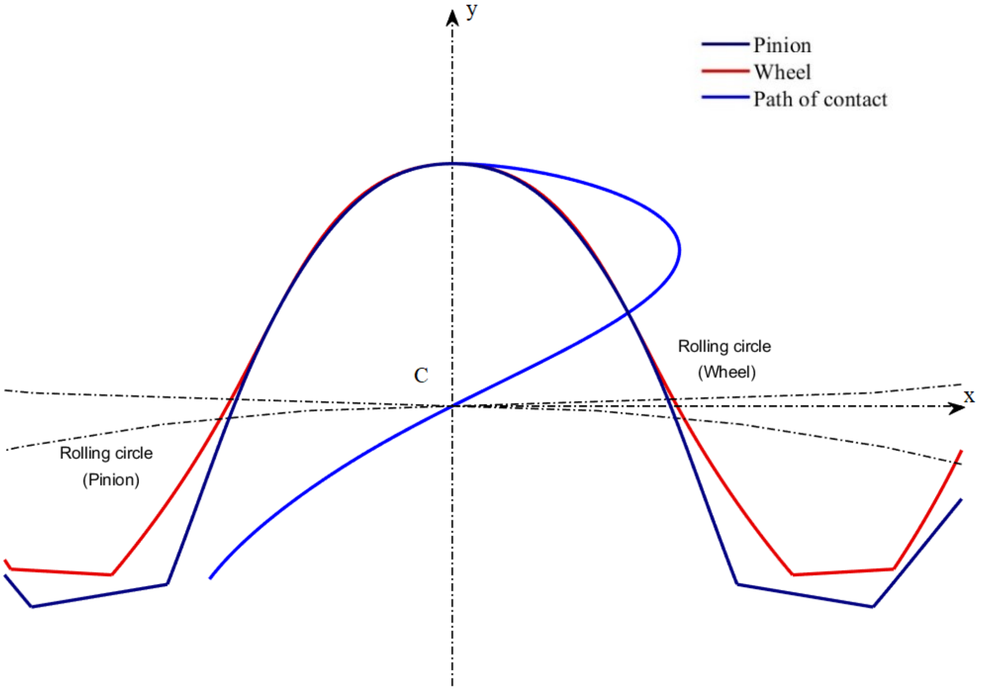


**Figure 6.3.6 Pinion and wheel in meshing - case 9**

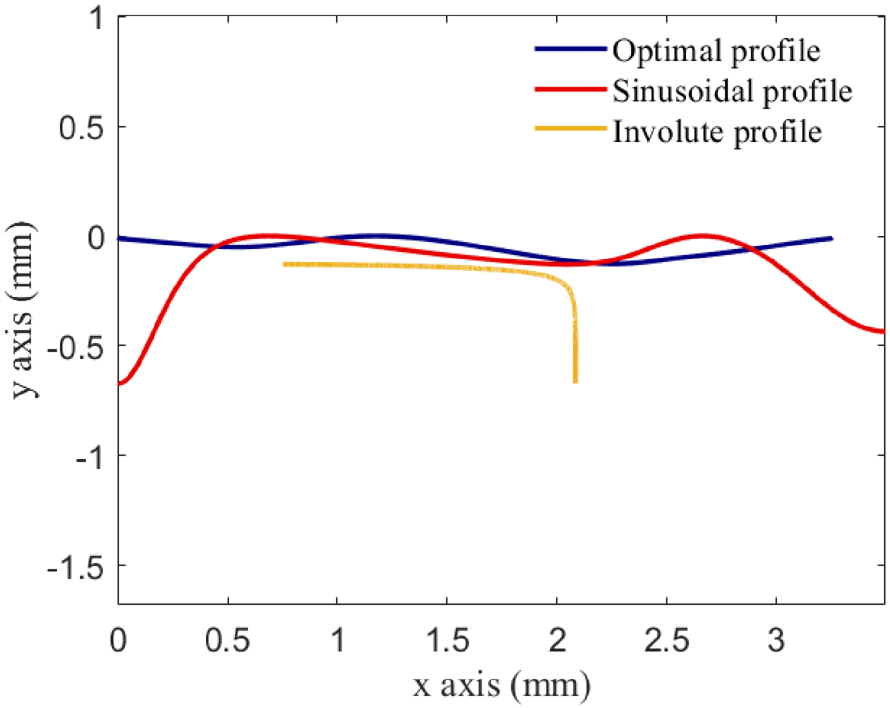


**Figure 6.3.7 Equivalent curvature - case 9**

| *Optimization Results (absolute values)* | | | | | |
|---|---|---|---|---|---|
| | Sinusoidal profile | Involute profile | Optimal profile | Decrease % (rel. to Sin) | Decrease % (rel. to Inv) |
| mean of equivalent curvature | 0.1333 | 0.1783 | 0.0625 | 95.31 | 64.93 |
| max of equivalent curvature | 0.6719 | 0.6708 | 0.1263 | 81.19 | 81.16 |
| std of equivalent curvature | 0.1542 | 0.0762 | 0.0396 | 74.30 | 47.97 |

**Table 6.3.4 Case 9 results**

In this case the population size is twice the one in the previous cases. The number of control points remain the same.

| *Optimization Results (absolute values)* | |
|---|---|
| Modeled with Control Points | Pinion’s rack gear |
| Number of Control Points | 7 |
| Number of variables | 10 |
| Population Size | 800 |
| Generation Size | 2300 |
| Crossover | 0.8 |
| Objective function | $\max\left(abs\left(\frac{1}{R}\right)\right) + mean\left(abs\left(\frac{1}{R}\right)\right) + std\left(\frac{1}{R}\right)$ |
| additional constraints | $\max\left(abs\left(\frac{1}{R}\right)\right) \leq 0.2$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| overlap | 1.19 |
| *B-Spline parameters* | |
| Polynomial’s degree | 4 |

**Table 6.3.5 Case 10 parameters**

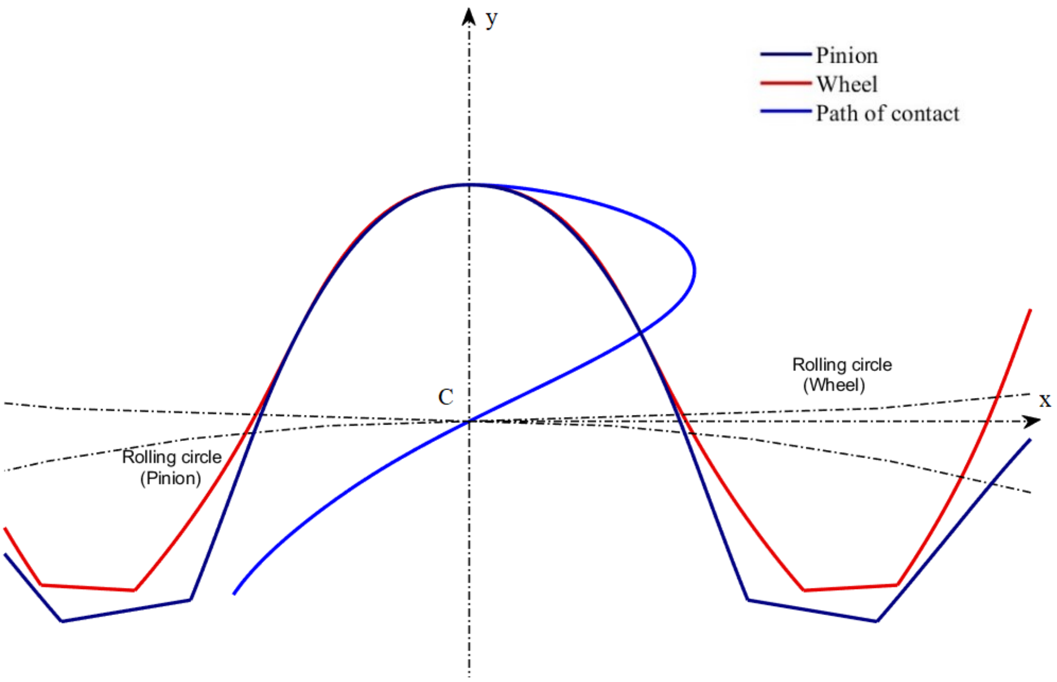


**Figure 6.3.8 Pinion and wheel in meshing - case 10**

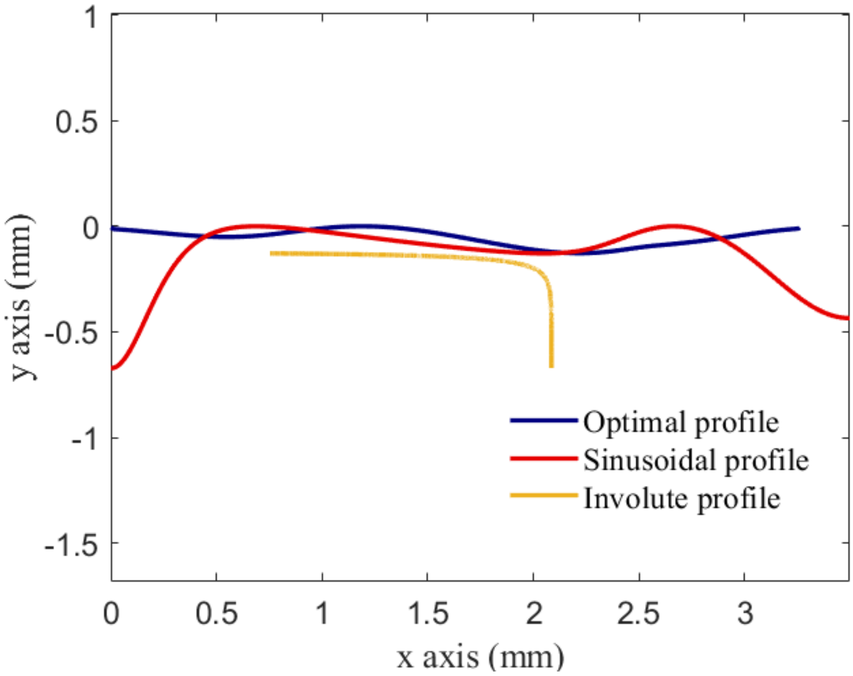


**Figure 6.3.9 Equivalent curvature - case 10**

| *Optimization Results (absolute values)* | | | | | |
|---|---|---|---|---|---|
| | Sinusoidal profile | Involute profile | Optimal profile | Decrease % (rel. to Sin) | Decrease % (rel. to Inv) |
| mean of equivalent curvature | 0.1333 | 0.1783 | 0.0617 | 95.37 | 65.39 |
| max of equivalent curvature | 0.6719 | 0.6708 | 0.1277 | 80.99 | 80.96 |
| std of equivalent curvature | 0.1542 | 0.0762 | 0.0395 | 74.39 | 48.17 |

**Table 6.3.6 Case 10 results**

It is clear that cases 8,9 and 10 have the same behavior and this can be shown from the corresponding tables. In all these three cases an important improvement is observed. Particularly in all of them the equivalent curve is above involute's equivalent curve. However, they cannot be considered as the best results because the reduction in standard deviation is smaller in comparison with other cases.

## 6.4 Fmincon Algorithm

Apart from the above results an attempt was made with fmincon algorithm in MATLAB/SIMULINK. `Fmincon` is a gradient-based method that is designed to work on problems where the objective and constraint functions are both continuous and have continuous first derivatives. It is true that one of the main problems in this optimization method is that the solution converges very close to the initial vector and as a result the algorithm is not searching enough for satisfying solutions. However, the following results prove that this algorithm can converge in remarkable solutions.

| *Optimization Results (absolute values)* | |
|---|---|
| Modeled with Control Points | Pinion's flank |
| Number of Control Points | 6 |
| Number of variables | 10 |
| Max iterations | 1000 |
| Max function evaluations | 3000 |
| X Tolerance | 1e-6 |
| Function tolerance | 1e-6 |
| Constraint tolerance | 1e-6 |
| Crossover | 0.8 |
| Objective function | $2\int\left(\frac{1}{R}\right)^2 dx + max\left(\left\|\frac{1}{R}\right\|\right) + 0.5\ std\left(\frac{1}{R}\right) + 0.12\frac{1}{overlap\ ratio}$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| overlap | 1.0227 |
| *B-Spline parameters* | |
| Polynomial's degree | 4 |

**Table 6.4.1 Case 11 parameters**

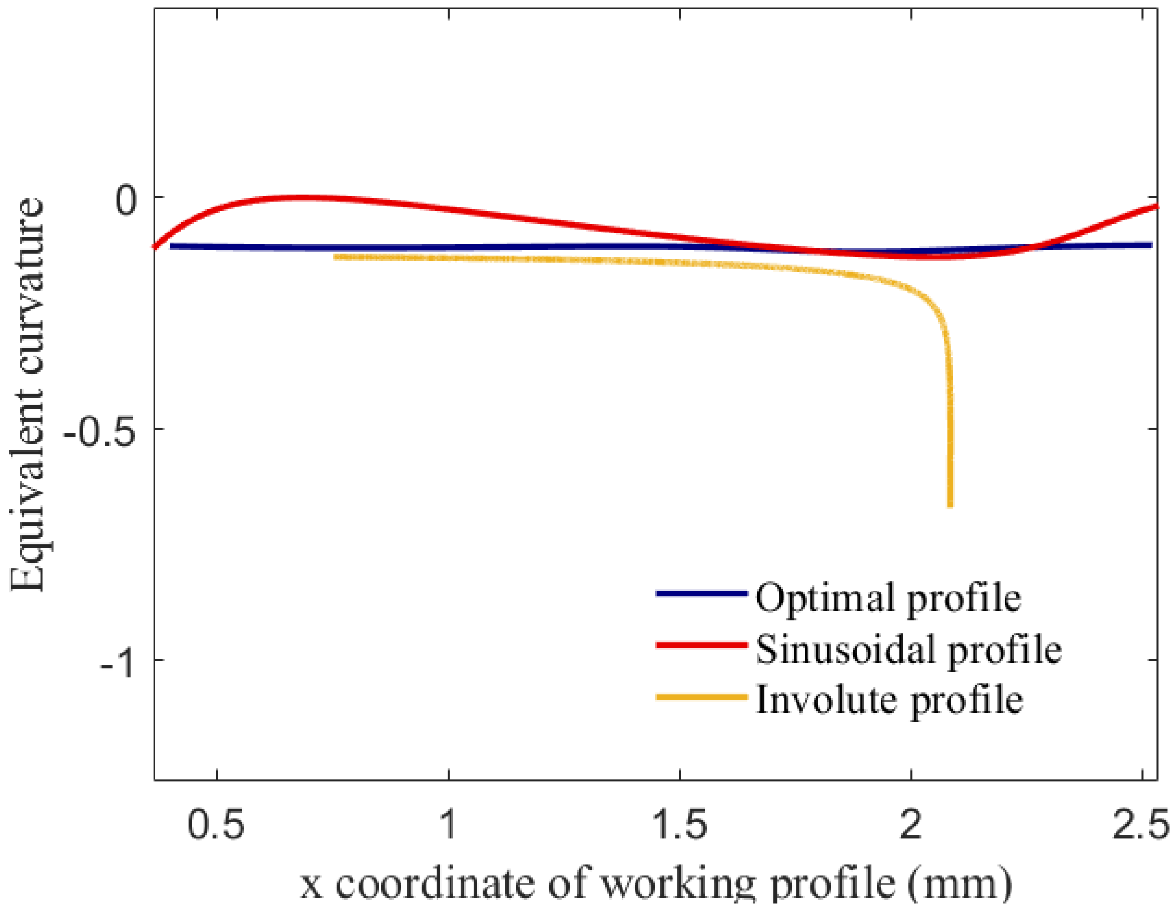


**Figure 6.4.1 Equivalent curvature - case 11**

| *Optimization Results (absolute values)* | | | | | |
|---|---|---|---|---|---|
| | Sinusoidal profile | Involute profile | Optimal profile | Decrease % (rel. to Sin) | Decrease % (rel. to Inv) |
| mean of equivalent curvature | 0.1333 | 0.1783 | 0.1093 | 91.79 | 38.67 |
| max of equivalent curvature | 0.6719 | 0.6708 | 0.1174 | 82.52 | 82.49 |
| std of equivalent curvature | 0.1542 | 0.0762 | 0.0042 | 97.29 | 94.53 |

**Table 6.4.2 Case 11 results**

| *Optimization Results (absolute values)* | |
|---|---|
| Modeled with Control Points | Pinion’s flank |
| Number of Control Points | 6 |
| Number of variables | 10 |
| Max iterations | 1000 |
| Max function evaluations | 3000 |
| X Tolerance | 1e-6 |
| Function tolerance | 1e-6 |
| Constraint tolerance | 1e-6 |
| Crossover | 0.8 |

| Objective function | $1.2\int\left(\frac{1}{R}\right)^2 dx + 0.5\, std\left(\frac{1}{R}\right) + 0.14\frac{1}{overlap\ ratio}$ |
|---|---|
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| overlap | 1.49 |
| *B-Spline parameters* | |
| Polynomial's degree | 4 |

**Table 6.4.3 Case 12 parameters**

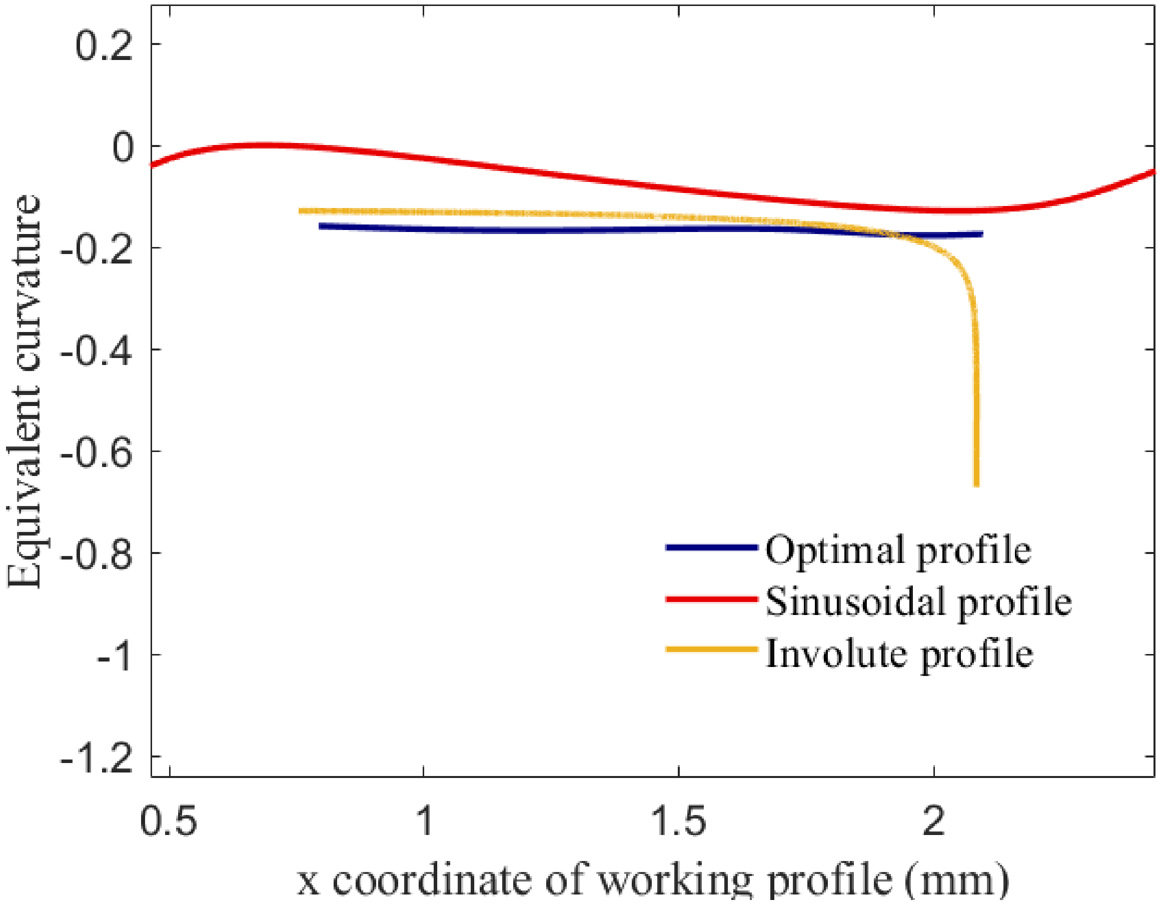


**Figure 6.4.2 Equivalent curvature - case 12**

| *Optimization Results (absolute values)* | | | | | |
|---|---|---|---|---|---|
| | Sinusoidal profile | Involute profile | Optimal profile | Decrease % (rel. to Sin) | Decrease % (rel. to Inv) |
| mean of equivalent curvature | 0.1333 | 0.1783 | 0.1678 | 87.41 | 5.88 |
| max of equivalent curvature | 0.6719 | 0.6708 | 0.1719 | 74.41 | 74.37 |
| std of equivalent curvature | 0.1542 | 0.0762 | 0.0027 | 98.22 | 96.40 |

**Table 6.4.4 Case 12 results**

| | |
|---|---|
| *Optimization Results (absolute values)* | |
| Modeled with Control Points | Pinion’s flank |
| Number of Control Points | 6 |
| Number of variables | 10 |
| Max iterations | 1000 |
| Max function evaluations | 3000 |
| X Tolerance | 1e-6 |
| Function tolerance | 1e-6 |
| Constraint tolerance | 1e-6 |
| Crossover | 0.8 |
| Objective function | $2\int\left(\frac{1}{R}\right)^{2} dx + 2\max\left\lvert\frac{1}{R}\right\rvert + 0.5\, std\left(\frac{1}{R}\right) + 0.3\,\frac{1}{overlap\ ratio}$ |
| *Gear parameters* | |
| Gear (Z1) | 19 |
| Conjugate Gear (Z2) | 55 |
| module | 2.5 |
| overlap | 1.1378 |
| *B-Spline parameters* | |
| Polynomial’s degree | 4 |

**Table 6.4.5 Case 13 parameters**

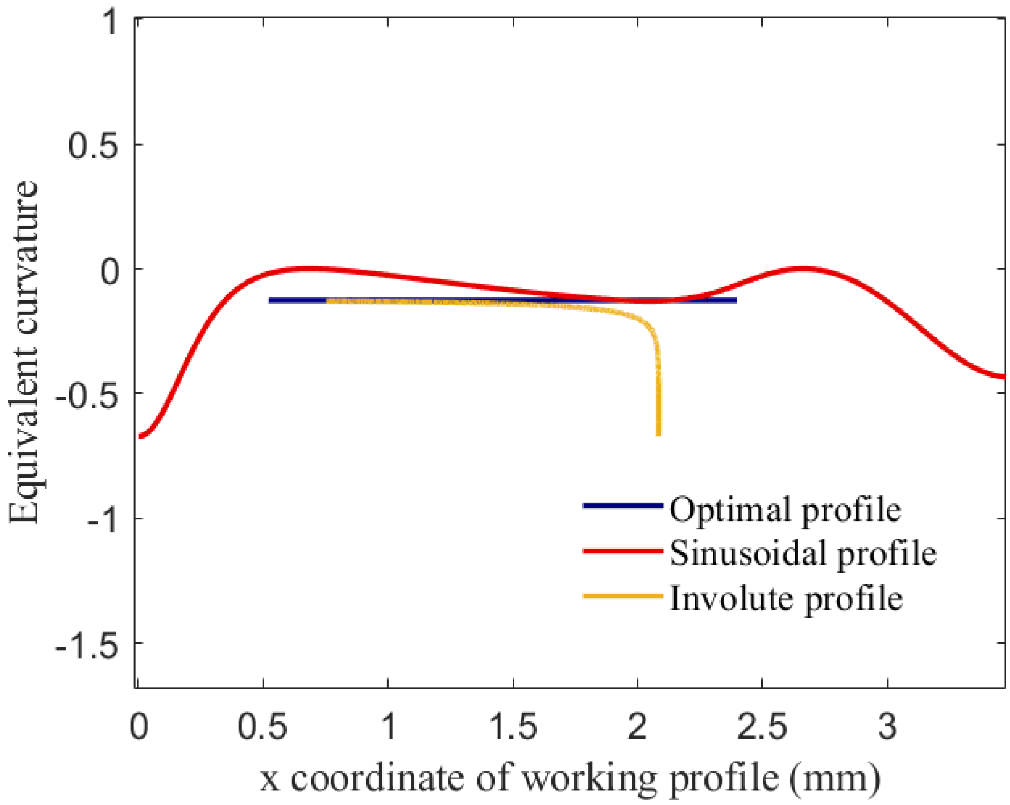


**Figure 6.4.3 Equivalent curvature - case 13**

| *Optimization Results (absolute values)* | | | | | |
|---|---|---|---|---|---|
| | Sinusoidal profile | Involute profile | Optimal profile | Decrease % (rel. to Sin) | Decrease % (rel. to Inv) |
| mean of equivalent curvature | 0.1333 | 0.1783 | 0.1254 | 90.59 | 29.64 |
| max of equivalent curvature | 0.6719 | 0.6708 | 0.1255 | 81.31 | 81.28 |
| std of equivalent curvature | 0.1542 | 0.0762 | 0.0001 | 99.95 | 99.91 |

**Table 6.4.6 Case 13 results**

The above results are very similar to the ones of the steepest descent algorithm. Although the algorithm is not capable to reduce the value of the equivalent curvature to zero, it assures that the equivalent curvature curve is stable. That is an indication that the optimization function is characterized by robustness given the fact that with two different methods we had the same results converged.

# 7. Contact analysis with Finite Element Model

## 7.1 Case study description

For the case study in which we were going to test the optimized gears (case 5) versus the involute gears, we selected the following pair of gears. The pinion gear has 19 teeth and the gear in meshing has 55 teeth. The module of the gears is 2.5 and their width is 20mm. Moreover, we assume the following parameters for the calculation of the moment applied in the pinion gear:

| | |
|---|---|
| $Z_1$ | 19 |
| $Z_2$ | 55 |
| m | 2.5 |

Where $Z_1$ is the teeth of the pinion gear, $Z_2$ is the teeth of the gear in meshing, and m is the module of gears.

The material used for the simulation is steel with the following properties:

| | |
|---|---|
| E(MPa) | $2\times10^5$ |
| v | 0.3 |

## 7.2 Cad models

Apart from the Involute gears whose cad models were imported to SOLIDWORKS –for the creation of the assembly- from the software KISSSOFT, all the others were created with the following procedure. Firstly, from a MATLAB code a .txt file with the coordinates of the tooth profile for each gear was created. Afterwards, the file was inserted in Design Modeler of Ansys Workbench where the 2D-surface of the entire gear was designed and then the assembly of the gear-set was completed. The CAD models for each case of gears are shown in the following pictures.

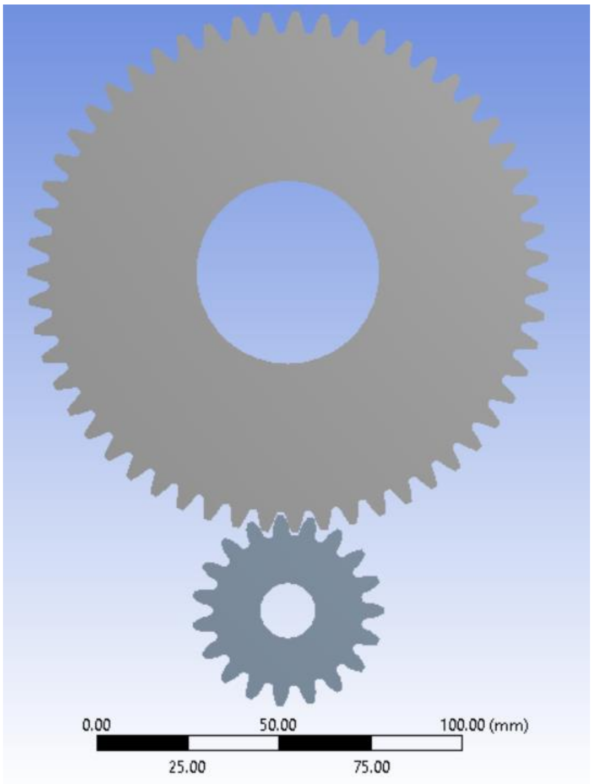


**Figure 7.2.1. CAD model of involute gears**

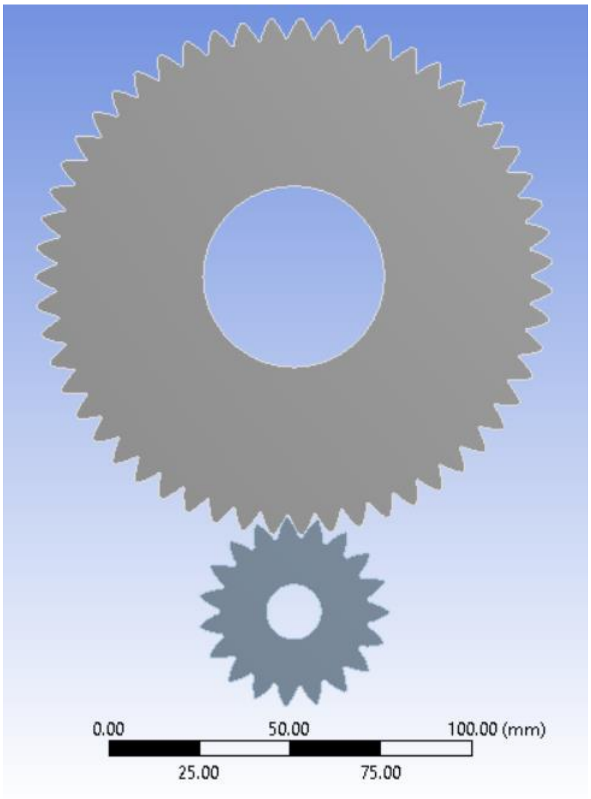


**Figure 7.2.2. CAD model of optimal gears**

## 7.3 Ansys model

The analysis was performed in the Static Structural section of ANSYS with the surface model of the assembly imported from SOLIDWORKS. The steps for the setup of the analysis where the following.

## 7.4 Meshing

For the meshing of the gears the Hertzian contact formulas were used to calculate the theoretical area of contact. The contact area calculates was then divided by using approximately 100 elements and the mesh was creates using the option of Edge Sizing and Refinement. The mesh created is shown in the following image for the case of involute gears (it is similar for the other cases).

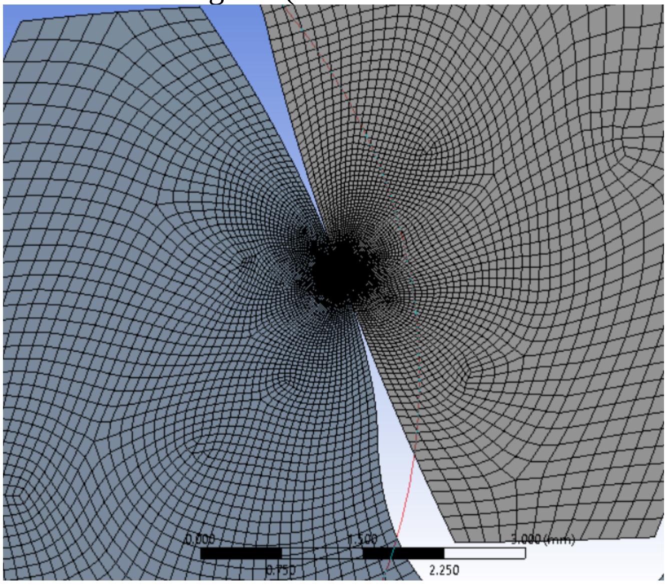
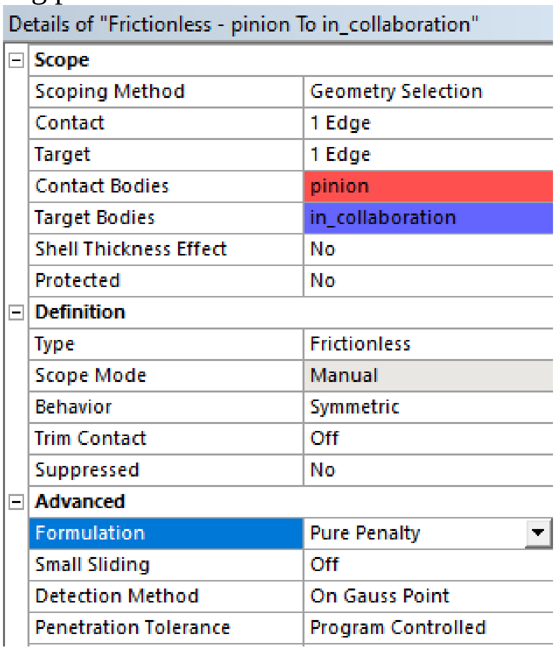


**Figure 7.4.1. Mesh of involute gears.**

## 7.5 Contact

Since a comparison of the results from ANSYS with the theoretical formulas derived for a Hertzian contact was necessary for the validity of the model, the contact between the teeth of the gears was set to Frictionless and the following parameters where used:



Details of "Frictionless - pinion To in_collaboration"

| | |
|---|---|
| ⊟ **Scope** | |
| Scoping Method | Geometry Selection |
| Contact | 1 Edge |
| Target | 1 Edge |
| Contact Bodies | pinion |
| Target Bodies | in_collaboration |
| Shell Thickness Effect | No |
| Protected | No |
| ⊟ **Definition** | |
| Type | Frictionless |
| Scope Mode | Manual |
| Behavior | Symmetric |
| Trim Contact | Off |
| Suppressed | No |
| ⊟ **Advanced** | |
| Formulation | Pure Penalty |
| Small Sliding | Off |
| Detection Method | On Gauss Point |
| Penetration Tolerance | Program Controlled |

**Figure 7.5.1. Definition of contact between gears.**

## 7.6 Boundary conditions

For the definition of the boundary conditions the option of Remote Point was used. Two Remote Points were defined; one at the center of each gear.

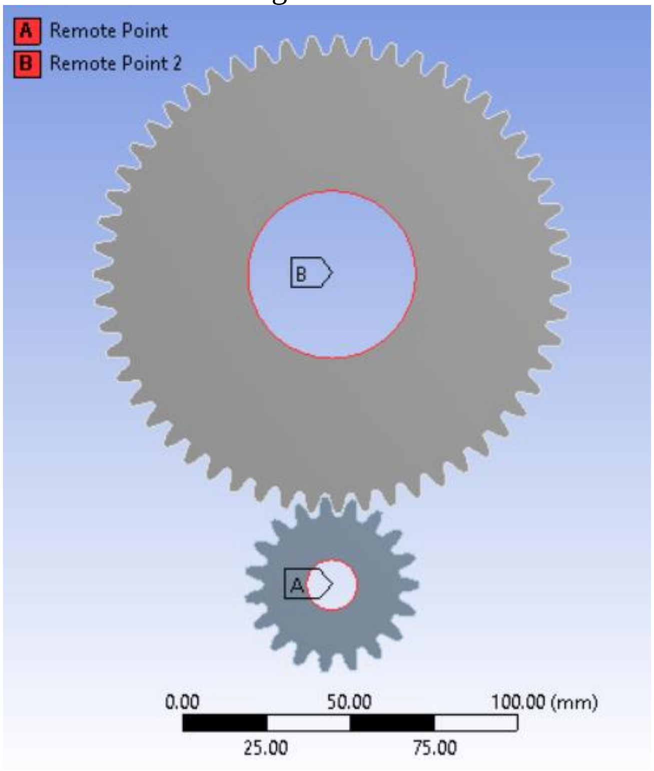


**Figure 7.6.1 Application of Remote Points in gears**

Moreover, the option of Remote Displacement was used. Two Remote Displacement were defined; one at each Remote Point. For the pinion gear all the degrees of freedom were set to zero except for the rotation around Z-axis and for the gear in collaboration all the degrees of freedom were set to zero.

## 7.7 Load

The load that was applied was a torque of $5000\ Nmm$ magnitude. Particularly it was applied in the remote point located in the center of the pinion gear as shown below.

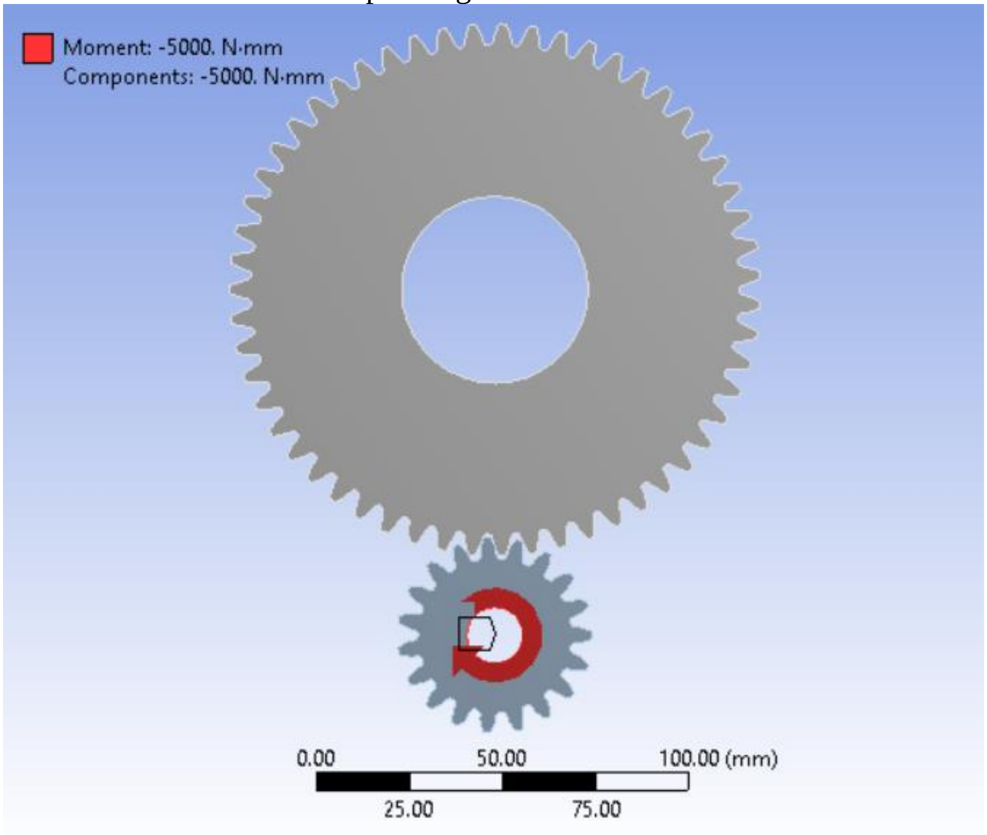


**Figure 7.7.1. Application of Moment in pinion gear**

## 7.8 Results

The Von-Mises stress and surface pressure are presented for all the cases in the position where the gears have contact in their pitch circle.

### 7.8.1 Equivalent Von-Mises Stress

For the calculation of Von-Mises stress the option Equivalent Von-Mises stress was inserted from the Solution Tab. The results are shown in the following figures.

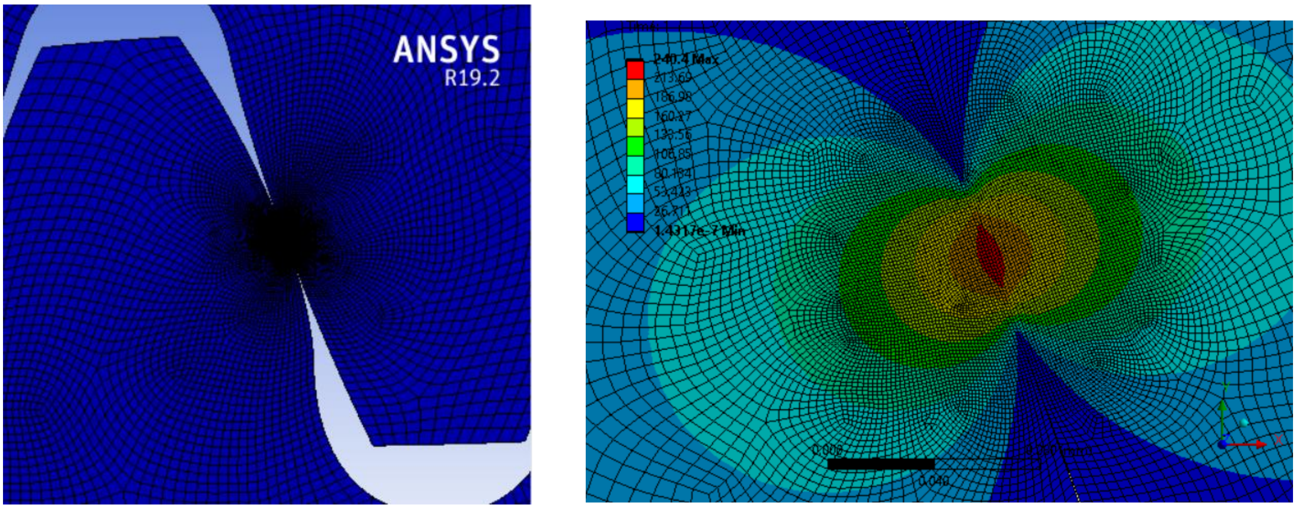


**Figure 7.8.1. Von-Mises stress - Involute gears, Maximum stress is 240.4 MPa**

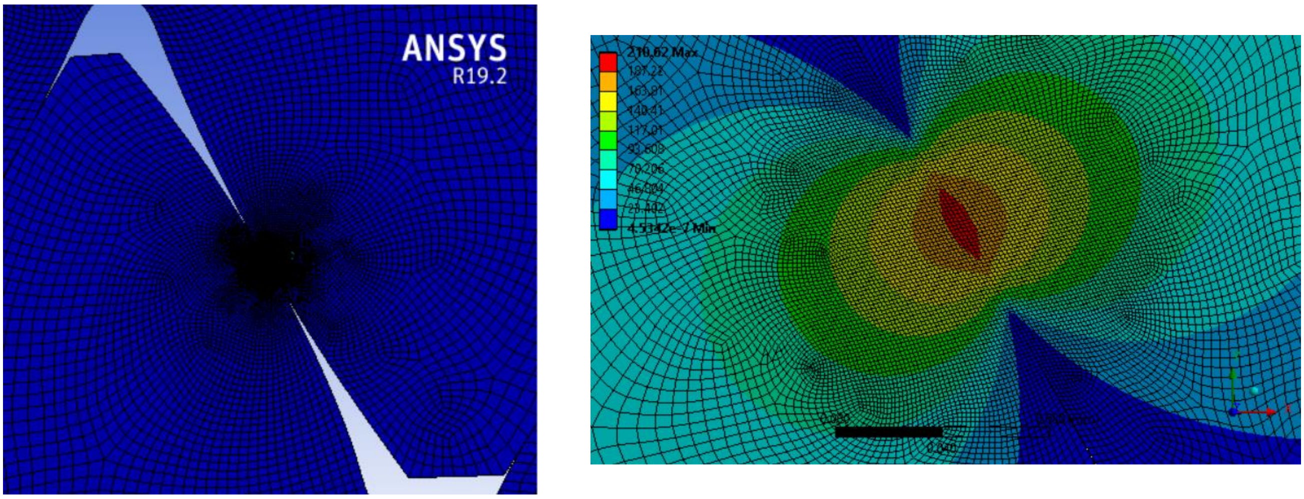


**Figure 7.8.2. Von-Mises stress - Optimal gears, Maximum stress is 210.62 MPa**

### 7.8.2 Surface pressure

For the calculation of the surface pressure the option Pressure of the Contact Tool was enabled. The results are shown in the following figures.

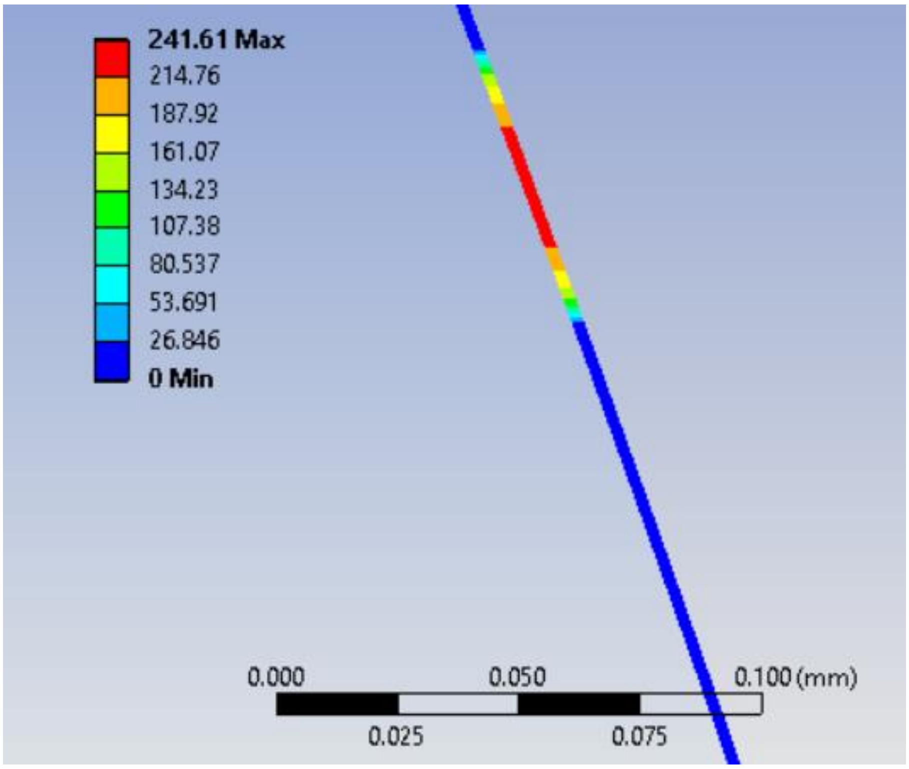


Figure 7.8.3. Surface pressure - Involute gears, Maximum stress is 241.61 MPa

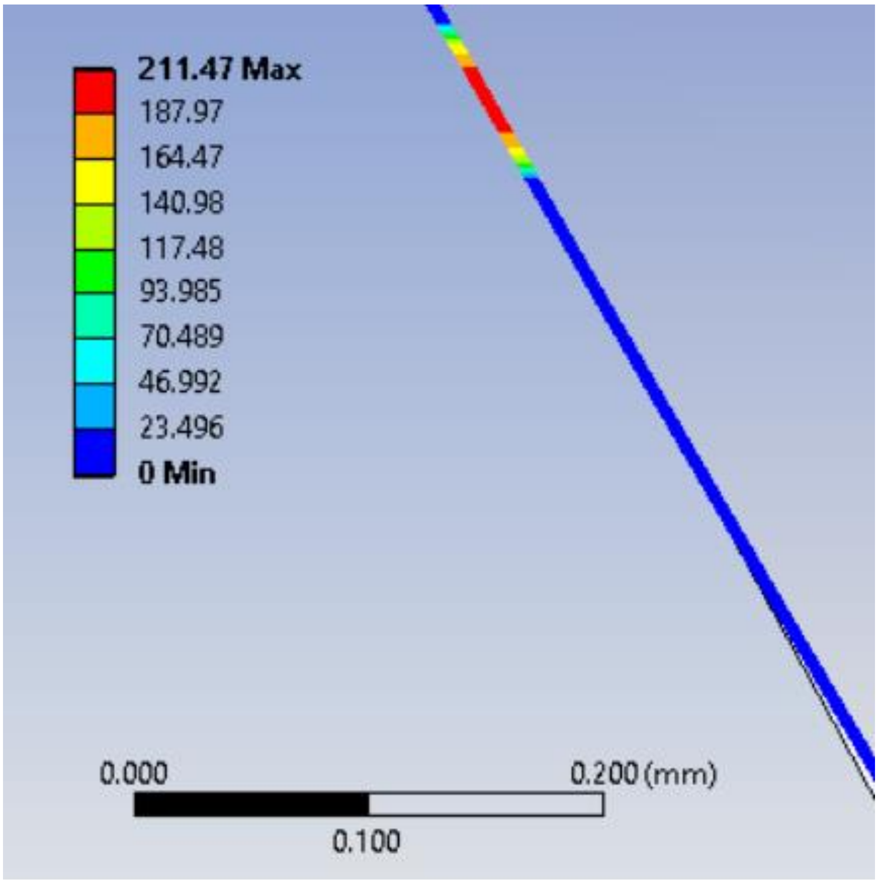


Figure 7.8.4. Surface pressure - Optimal gears, Maximum stress is 211.47 MPa

## 7.9 Verification of Finite Element Model

To verify that the finite element model yields proper results, the surface pressure of the gears was calculated for the above point of contact with the theoretical Hertz formula for the involute gear and the results are presented below:

| Hertz surface pressure(MPa) | Finite Element Model surface pressure (MPa) |
|---|---|
| 254.13 | 241.61 |

We can observe that the values of surface pressure are very close. In fact, the error between the theoretical and the numerical value is -5.1%. Furthermore, we observe that the optimal gears have lower stresses than the involute gears. This is another verification of the functionality of the model, since in this point of contact the equivalent curvature of involute gears is lower than the one in the optimal gears. Therefore, we can verify that the model works properly.

In addition, after running the finite element model in different points throughout the flanks of the gears the following data were gathered in order have a more complete comparison between them.

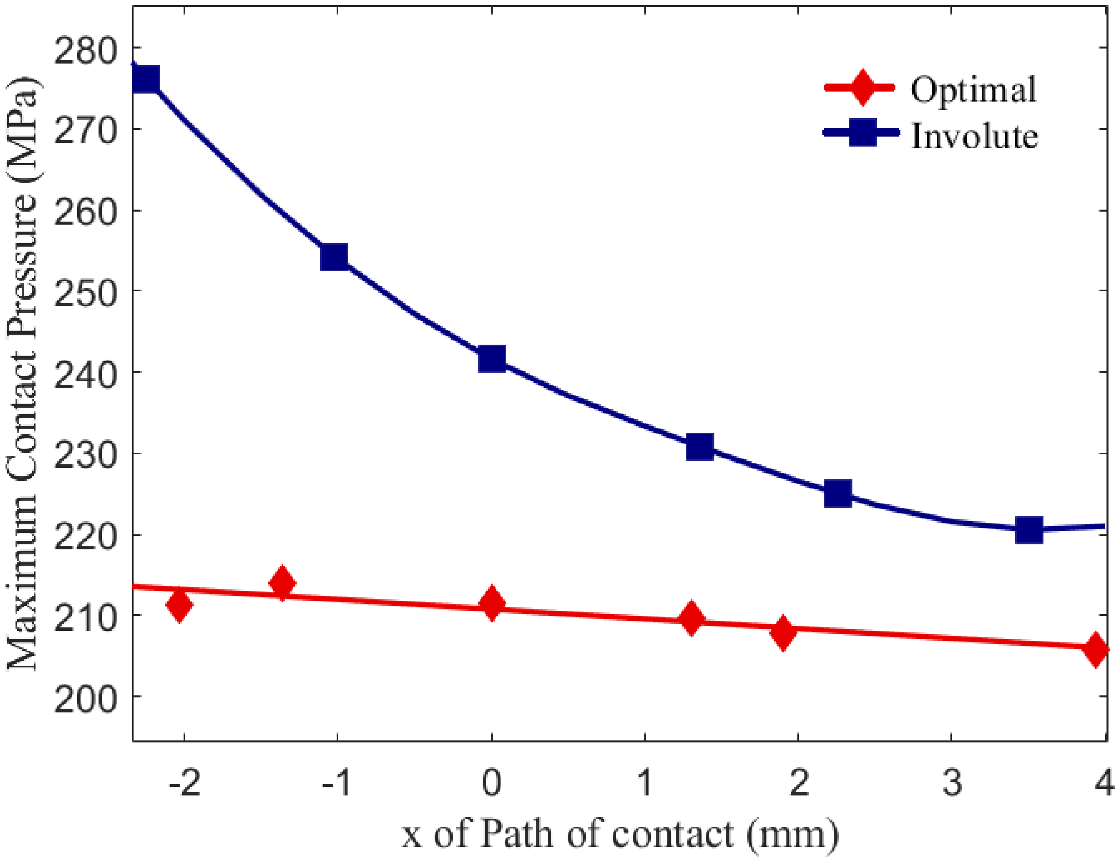


**Figure 7.9.1. Variation of surface pressure in different points of contact of the gears.**

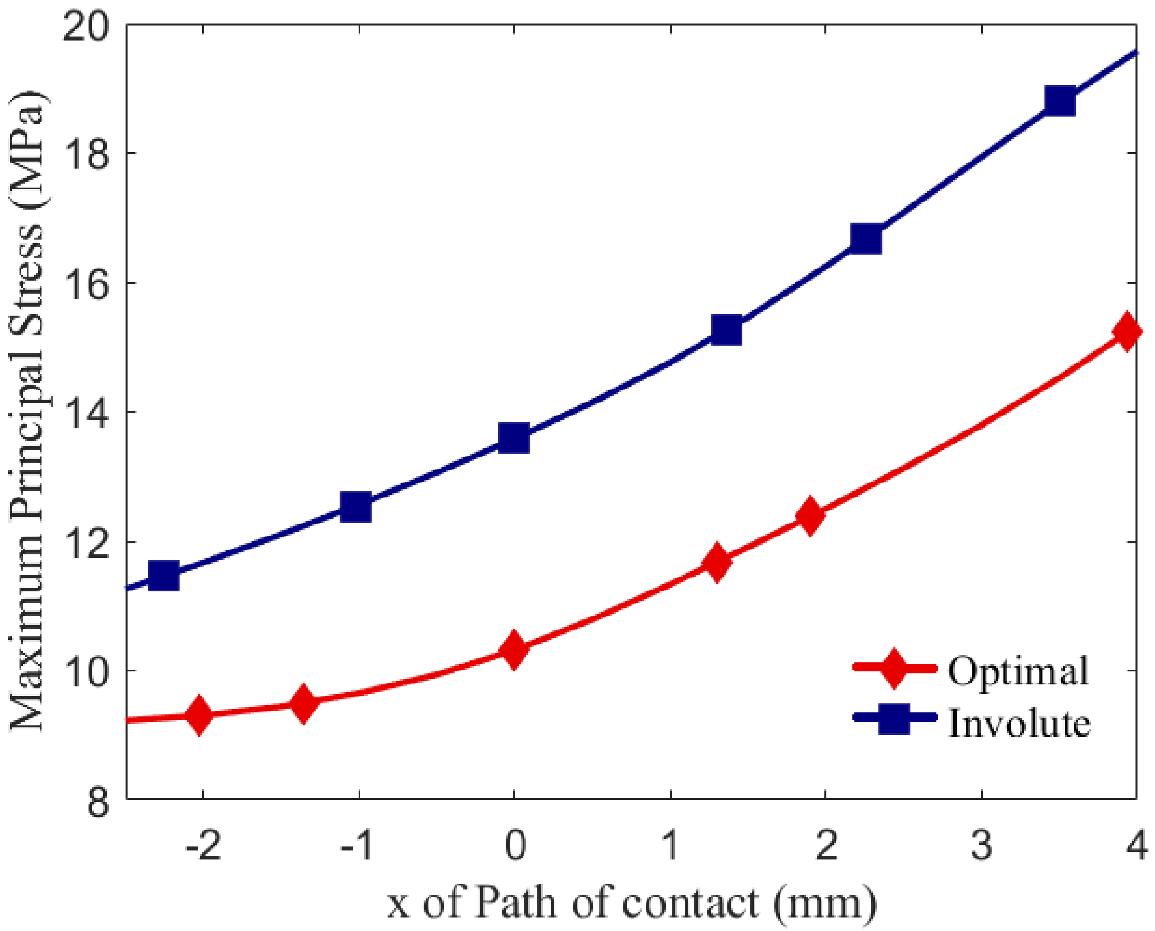


**Figure 7.9.2. Variation of bending stress in different points of contact of the gears.**

The following table contains the total results:

| Z1=19, Z2=55, m=2.5 | Involute gears | Optimal gears - case 5 |
|---|---|---|
| Mean of equivalent curvature | 0.1783 | 0.1130 |
| Mean surface pressure (MPa) | 241.4067 | 210.02 |
| Mean bending stress (MPa) | 14.7187 | 11.40 |

**Table 7.9.1 Case 5 Finite Element c**

Finally, in figure 7.9.3 the generated pinion and wheel is presented while the figure 7.9.4 shows a better view of the gears in meshing.

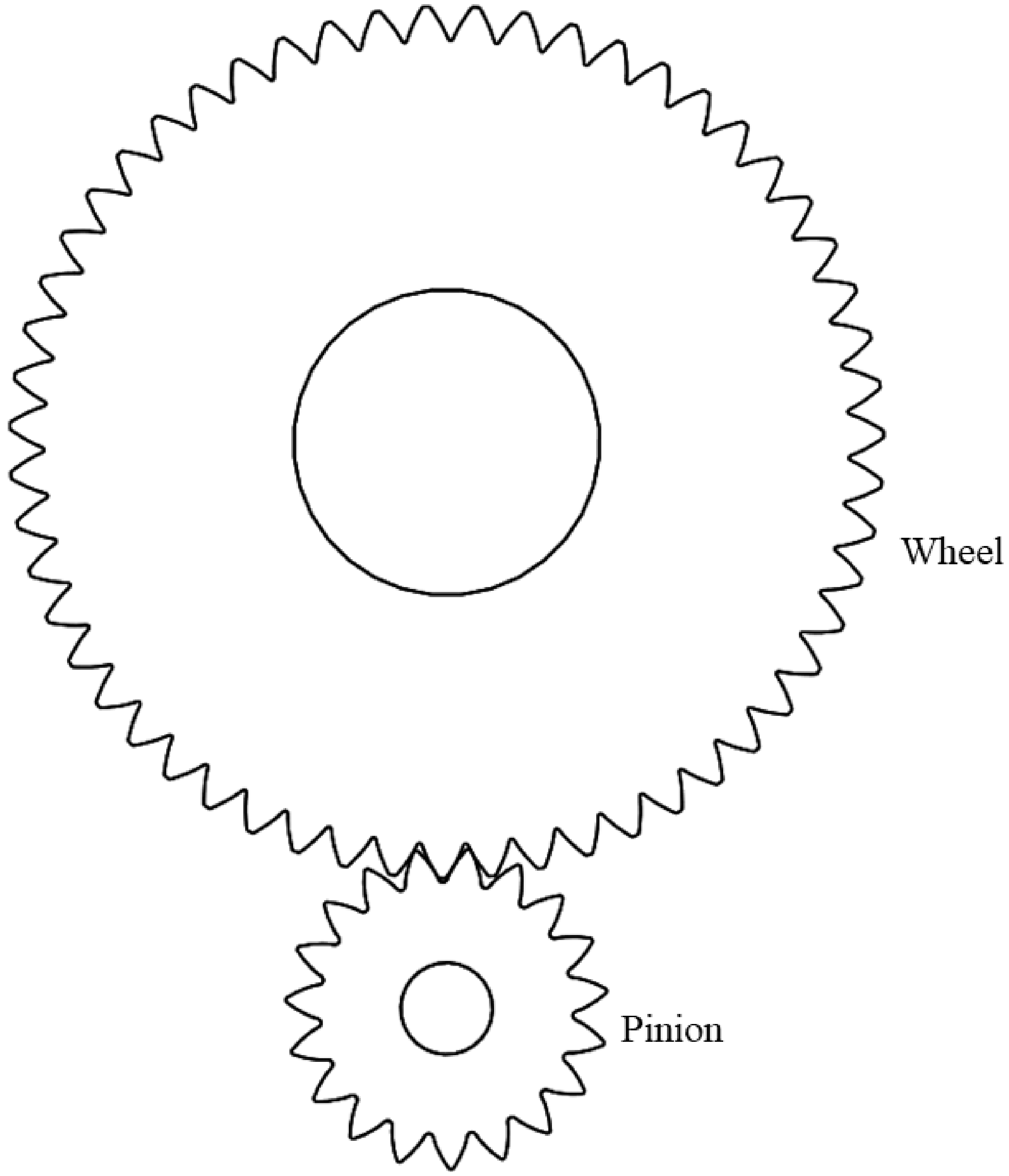


**Figure 7.9.3. Generating pinion and wheel - case 5**

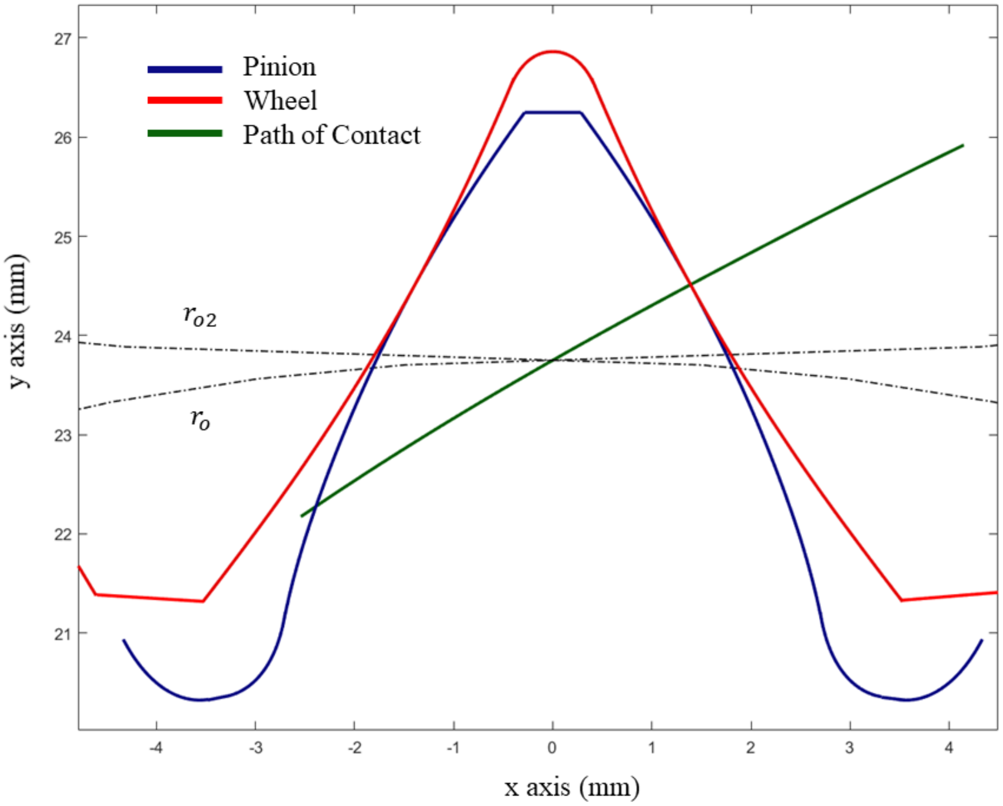


**Figure 7.9.4. Pinion and wheel in meshing - case 5**

# 8. Conclusions

Taking into consideration all these results we observe that the steepest descent's solution in open contact path shows better behavior than the rest. The Fmincon algorithm came up with solutions very close to those that Steepest Descent came up with given the fact that both methods generated gears with constant equivalent curvature curves. The equivalent curvature of case 5 is permanently above involute's equivalent curvature and by far mor stable than sinusoidal's profile. This is confirmed by the respective factors that were suggested in order to assess the results (average curvature, standard deviation and maximum of equivalent curvature). As a result, we expect an improved stress behavior in comparison with involute gears while the optimal solution is likable to resist surface fatigue in comparison with sinusoidal gears. However, there are improvements to be done such as attempts to "push" the equivalent curvature curve towards zero. That would ensure lower stresses magnitudes when gears mesh.

Steepest Descent and Fmincon algorithm are more sensitive in abrupt changes in the equivalent curvature curve than genetic algorithms and this can be seen in the results. The main problem in genetic algorithms is that the solutions have undesirable peaks that will be responsible for pitting. Another problem faced in this optimization attempt is when the algorithm sticks to the initialization vector which means that the algorithm has not searched enough for a desirable solution.

One of the main parts of this modeling is the objective function. It is true that the same method can provide a totally different result only with subtle changes in the objective function. There are terms in the objective function useful to avoid solutions that do not satisfy the geometrical criteria and constraints that must be met according to theory of gearing such as the law of gearing or pitch compatibility.

The results presented prove that there are ways to reduce surface fatigue while the bending stresses in the root could also be reduced. The standard deviation of the equivalent curvature curve is reduced up to 98.2% relatively to the involute gears and this is depicted in the finite elements confirmation.

# 9. References

[1] Costopoulos T. N. (1991), Gears and Gearboxes, B’ Edition, Simeon’s Publications, Athens (in Greek).

[2] Provatidis C. G. (2017), Optimization & CAD: Finite Elements, Isogeometric Elements, Boundary Elements, TZIOLA’S Publications, Athens (in Greek).

[3] Spitas V. A., Costopoulos T. N., Spitas C. A., (2002) A quick and efficient algorithm for the calculation of gear profiles based on flank involutization, Athens.

[4] D. H. Cooper (1968), Tables of hertzian contact-stress coefficients

[5] Saraswat M., Sharma A. K., (2013) Genetic Algorithm for optimization using MATLAB, International Journal of Advanced Research in Computer Science, Volume 4, No. 3.

[6] Max Weibring, Leonard Gondecki, Peter Tenberge (2018), Simulation of fatigue failure on tooth flanks in consideration of pitting initiation and growth, Bochum.

[7] Spitas V. A., (2001), Modelling and design optimal gears using analytical, computational and experimental methods (Doctoral Thesis), Athens (in Greek).

[8] Peroukidis K., (2021), Analysis, design and simulation of active gear pump flow ripple damping systems (Diploma Thesis), Athens (in Greek).

[9] gamultiobj Algorithm, MathWorks Documentation, Available in:
https://www.mathworks.com/help/gads/gamultiobj-algorithm.html

[10] Kostakis F., (2013), Optimal design of gears of closed contact path in order to even distribute stresses (Diploma Thesis), Athens (in Greek).

[11] Buckingham, E., Analytical Mechanics of Gears, Dover Publications Inc., New York, 1988

[12] Timoshenko, S., and Baud, R., V., “Strength of Gear Teeth”, Mechanical Engineering, Vol. 48, p. 1105, 1926

[13] Kelley, B., W., and Pedersen R., “The Beam Strength of modern Gear Tooth Design”, Trans. SAE, Vol. 66, 1950.

[14] Nieman G., Machine Elements – Design and Calculation in Mechanical Engineering. Volume II GEARS, Springer Verlag, 1978 (English Edition)

[15] Small, N., C., “Bending of a Cantilever Plate Supported from an Elastic Half Space”, ASME J. Appl. Mech., Vol. 28, no. 3, September 1961.

[16] Wellauer, E., J., and Seireg, A., “Bending Strength of Gear Teeth by Cantilever-plate Theory”, ASME J. Eng. Ind., Vol. 82, August 1960.

[17] Heywood, R., B., Designing against Fatigue of Metals, Reinhold Publishing Corporation, 1962.

[18] Dolan, T., J., and Broghamer, E., L., “A Photo-elastic Study of Stresses in Gear Tooth Fillets”,

Univ. III. Eng. Exp. Sta. Bull 335, 1942.

[19] Litvin, F. L., 1989, “Theory of Gearing”, NASA Reference Publication 1212, AVSCOM Technical Report, 88-C-035

[20] Litvin, F., L., Theory of Gearing, 2nd ed., Nauka, Moscow, 1968

[21] Illinois Tool Works, "Helicon Gears", Illinois Tool Works, Chicago, 1959

[22] Illinois Tool Works, "Planoid Gears", Illinois Tool Works, Chicago, 1958

[23] Illinois Tool Works, "Spiroid Gears", Illinois Tool Works, Chicago, 1959

[24] Townsend D., P., and Bambeger E., N., "Surface Fatigue Life of Carburized and Hardened M50NiL and AISI 9310 Spur Gears and Rolling Contact Test Bars", Journal of Propulsion and Power, Vol. 7, No.1, 1991.

[25] Daniewicz S., R., and Moore D., H., "Increasing the bending Fatigue Resistance of Spur Gear Teeth Using a Presetting Process", International Journal of Fatigue, Vol. 20, No.7, pp. 537-542, 1998.

[26] Tsai M., and Tsai Y., "Design of High Contact Ratio Spur Gears Using Quadratic Parametric Tooth Profiles", Mechanism and Machine Theory, Vol. 33, No.5, pp. 551- 564, 1998.

[27] Mabie H., H., Rogers c., A., and Reinholtz c., F., "Design of Nonstandard Spur Gears Cut by a Hob", Mechanism and Machine Theory, Vol. 25, No.6, pp. 635-644, 1990.

[28] Rogers c., A., Mabie H., H., and Reinholtz c., F., "Design of Spur Gears Generated with Pinion Cutters", Mechanism and Machine Theory, Vol. 25, No.6, pp. 623-634, 1990.

[29] AGMA 210 l-C95, "Fundamental rating factors and calculation methods for involute spur and helical gear. (Metric version)", American Gear Manufacturers Association, 1995.

[30] DIN 3990 - "Calculation of load capacity of cylindrical gears", Deutsches Institut fur Normung E. v., 1987.

[31] ISO 6336-3:1996 "Calculation of the load capacity of spur and helical gears-Part 3: Calculation of tooth bending strength".

[32] Heinrich Hertz, Miscellaneous Papers (Trans, by D. E. Jones and G. A. Schott) Macmillan and Co., Ltd., London, 1896, pp. 146-183. (Contains English translations of the papers from J. Reine and Angew. Math. (Crelle) 9£, 156-171, 1881, and Verhand. Vereins Befb'rder. Gewerbefleisscs, Nov. 1882.)

[33] H.L. Whittemore and S.N. Petrenko, Natl. Bur. Std. Tech. Paper 201,1921 in : S.P. Timoshenko and J.N. Goodier, (Eds.) Theory of Elasticity Third Edition, McGRAW-HILL INTERNATIONAL EDITIONS, New York, 1970

[34] Provatidis C. G. (2017), Finite Elements in Construction Analysis, 2$^{nd}$ Edition, TZIOLA’S Publications, Athens (in Greek).

# Appendix

The scripts concern the open contact path.

## Steepest Descent

### Starting_Script.m

```
clear all;clc;
format short
% To paron programma ekkinei ton algorithmo ths steepest descent
n=5;     % Control Points
Z=19;    % pinion
m=2.5;   % module
tooth=2; % posa dontia aristera tha ginei h modelopoihsh tou flank
profile
functname = 'fitness';
dvar0     =LeastmS(Z,m,n,tooth);

niter     = 100;
tol       = 1e-8;
lowbound  = 0;
intvl     = 1;
ntrials   = 100;

[x,Y]=SD_Project(functname, dvar0, niter, tol, lowbound, intvl,
ntrials);
```

### LeastmS.m

```
%% LEAST MIN SQUARES
function b = LeastmS(Z,m,n,tooth)
%%
% Sto paron programma eisagontai oi suntetagmenes involute line 11
% strefontai aristera kata duo dontia (kata voulhsh auto) wste na
einai panta
% sunarthsh oi suntetagmenes tou flank profile
% kai meta montelopoiountai me B-Splines gia na dwthoun ws arxikh
timh ston
% algoruthmo veltistopoihshs
%%
% Z= dontia pinion
% m= module
% n= numel of Control Points
% tooth = # twn dontiwn aristera pou tha ginoun oi upologismoi
%% Input Data SIN - (x,y) coordinates ths Troxias Epafwn  P : Data
to interpollate P(:,1)=x;P(:,2)=y;
[xf1,yf1]=Involute_Coordinates(Z,m);
do=Z*m;ro=do/2;       % [pcd] Pitch Circle Diameter
phi=tooth*2*pi/Z;
x9=xf1*cos(phi)-(yf1+ro)*sin(phi);
y9=xf1*sin(phi)+(yf1+ro)*cos(phi);
P(:,1)=x9';P(:,2)=y9';
div=size(P);
%% Input of B-Spline parameters
```

```
% n : Number of CP , proekupsan apo sxesh n+k+1-numel(T)  =  0
% k : deg of B-Spline , elaxisto 3
k=4;
% T : knot vector , numel(T)=n+k+1 , pollaplothta : 1 h parapanw
analoga
T=[zeros(1,k+1) 1:1:n-k-1 (n-k)*ones(1,k+1)];
% Basis Functions of B-Splines
t=T(1):(T(end)-T(1))/(div(1)-1):T(end);
%
N=zeros(k+n,k+1,numel(t));
for it=1:numel(t)
  for i=1:numel(T)-1
    if t(it)>=T(i) && t(it) < T(i+1)
     N(i,1,it)=1;
    else
     N(i,1,it)=0;
    end
  end
  a=1;
  for j=2:k+1
    for i=1:numel(T)-1-a
      if  (N(i,j-1,it)==0 && N(i+1,j-1,it)==0)  % zero kai oi
duo oroi
        N(i,j,it)=0;
        if i<=n
          dN(i,it)=0;
        end
      elseif N(i,j-1,it)==0                     % zero 1os oros
        N(i,j,it)=((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-
1,it);
        if i<=n
          dN(i,it)=- (k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
        end
      elseif N(i+1,j-1,it)==0                   % zero 2os oros
        N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-1,it);
        if i<=n
          dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it);
        end
      else
        N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-
1,it)+((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
        if i<=n
          dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it) -
(k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
        end
      end
      if (i==n && j==k+1 && t(it)==T(end))
         N(:,:,it)=round(N(:,:,it-1),0);
         dN(:,it)=round(dN(:,it-1),0);
      end
    end
```

```
       a=a+1;
    end
% Nk(:,it)=N(1:n,k+1,it); % basis functions
end
C0=zeros(numel(t),2);

%% Least mean Squares
B_sp_k=zeros(n,numel(t));
for i=1:n
B_sp_k(i,:)=N(i,k+1,:);
end
CP(1,:)=inv(B_sp_k*transpose(B_sp_k))*B_sp_k*P(:,1);  % Euresh
Control Points
CP(2,:)=inv(B_sp_k*transpose(B_sp_k))*B_sp_k*P(:,2);
%%
CP(1,1)=P(1,1); CP(1,end)=P(end,1);    % To prwto kai to teleutaio
CP
CP(2,1)=P(1,2); CP(2,end)=P(end,2);    % ta orizw me vash ta shmeia
paremvolhs
b=[CP(1,:),CP(2,2:end-1)]; % prin thn allagh -tmhmata
end
```

Involute_Coordinates.m

```
function [xf1,yf1] = Involute_Coordinates(Z,m)
%%
% Sto paron programma briskoume tis suntetagmenes tou profile ths
involute
% gia pinion me Z dontia kai module m
%%
a0=deg2rad(20);       % pressure angle
div=1e4;              % #of nodes
%% Gear parameters
dk=(Z+2)*m;rk=dk/2;   %[od] Outside diameter Cost pp 48
do=Z*m;ro=do/2;       %[pcd] Pitch Circle Diameter
dg=do*cos(a0);        %[bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg=dg/2;              %[rbase]
So=(0.475*pi)*do/Z;   %(0.475-0.5) Tooth thickness at pcd  Cost pp
46 (7)
inva0 = tan(a0) - a0; %Involute function
%% Involute generation
for i = 1:div
r(i) = rk-((rk-rg)/(div-1))*(i-1); % radius from rk to rg
a(i) = acos(dg / (2 * r(i)));          % angle a of each involute
point
inva(i) = tan(a(i)) - a(i);            % involute function fi for
each involute point
S(i) = So * (r(i) / ro) + 2 * (inva0 - inva(i)) * r(i); % Tooth
thickness at each point Cost pp 46 (10)
ak=acos(rg/rk);                     % Cost pp 46
xp(i)= r(i) * sin(inva(i));         % change from polar coordinates
```

```
yp(i)= r(i) * cos(inva(i));         % to cartesian coordinates
end
Sg=S(div);                        % paxos at base circle
%% Rotation of involute to its symmetrical initial position
for i=0:Z-1
el=(2*pi*rg-Z*Sg)/Z;
phig = i*((Sg+el)/rg);
rot = [ cos(phig) -sin(phig);
        sin(phig) cos(phig)];
inv_rot = rot * [xp; yp];
xp_rot = inv_rot(1,:);
yp_rot = inv_rot(2,:);
%%
phi=deg2rad(360*Sg/(2*pi*rg))/2;
xp_rot1= cos(phi)*xp_rot -sin(phi)*yp_rot;
yp_rot1= sin(phi)*xp_rot+ cos(phi)*yp_rot;
if i==0
    xf1=-xp_rot1;
    yf1=yp_rot1-ro;
end
end
```

UpperBound.m [2]

```
function ReturnValue = UpperBound(functname,x,s,a0,da,ns)
if (ns~=0)
    ntrials = ns;
else
    ntrials =10;
end
if (da~=0)
    das = da;
else
    das =1;
end
for i=1:ntrials
    j=0;
    dela = j*das;
    a00 = a0+dela;
    dx0 = a00*s;
    x0 = x + dx0;
    f0 = feval(functname,x0);
    j=j+1;
    dela = j*das;
    a01 = a0+dela;
    dx1=a01*s;
    x1=x+dx1;
    f1 = feval(functname,x1);
    f1s=f1;
    if f1<f0
        for j=2:ntrials
            a01=a0+j*das;
            dx1=a01*s;
            x1=x+dx1;
            f1 = feval(functname,x1);
            f1s=min(f1s,f1);
            if f1>f1s
                ReturnValue = [a01 f1 x1];
                return;
            end
        end
%         disp('Cannot increase function in ntrials')
        ReturnValue = [a01 f1 x1];
        return;
    else
        das=0.5*das;
    end
end
ReturnValue = [a0 f0 x0];
end
```

SD_Project.m [2]

```
function [designvar,Y] = SD_Project(functname, dvar0, niter, tol,
lowbound, intvl, ntrials)
e1=1.0e-08;
nvar = length(dvar0);
xs(1,:) = dvar0;
x = dvar0;
fs(1)= feval(functname,x);
as(1)=0;
s = -(GradFun(functname,x));
convg(1)=s*s';
for i=1:niter-1
    output = GoldenSection(functname, tol, x, s, lowbound, intvl,
ntrials);
    as(i+1) = output(1);
    fs(i+1) = output(2);
      for k=1:nvar
      xs(i+1,k) = output(2+k);
      x(k) = output(2+k);
      end
    s = -(GradFun(functname,x));
    convg(i+1)=s*s';
           if (i<niter-1)
           age1=feval(functname,x);
           age=(i/(niter-1))*100;
           name='%';
           fprintf('%.2f%s    Loading...     F    equals    %d
cycle %d.\n',age,name,age1,i)
           else
           age=100;
           name='%';
           fprintf('%d%s      Completed min  F    equals    %d
cycle %d.\n',age,name,age1,i)
           end
      if (convg(i+1)<=e1) break; end;
end

len=length(as);
designvar = xs(length(as),:);
Y=fs(len);
disp('***** Application of Steepest Descent Method *****')
disp('Optimized design vector')
disp(designvar)
disp('Optimized Objective Function Value')
disp(fs(len))
end
```

GradFun.m [2]

```
function Return = GradFun(functname,x)
hstep=1.0e-3;
n = length(x);
f = feval(functname,x);
for i=1:n
    xs = x;
    xs(i) = xs(i) + hstep;
    gradx(i) = (feval(functname,xs)-f)/hstep;
end
Return = gradx;
end
```

GoldenSection.m [2]

```
function ReturnValue = GoldenSection(functname, tol, x, s,
lowbound, intvl, ntrials)
upval = UpperBound(functname, x, s, lowbound, intvl, ntrials);
au = upval(1);
fau = upval(2);

if (au<=1.0e-06)
    aL = lowbound; xL=x+aL*s;
    faL = feval(functname,x);
    ReturnValue = [aL faL x];
    return
end
if (tol == 0) tol = 1.0e-04;
end
eps1 = tol/(au - lowbound);
tau = 0.38197;
nmax = round(-2.078*log(eps1));
aL = lowbound;
 xL = x+aL*s;
  faL = feval(functname,xL);
a1 = (1-tau)*aL +tau*au;
```

```
 x1 = x + a1*s;
  fa1 = feval(functname,x1);
a2 = tau*aL +(1-tau)*au;
 x2 = x + a2*s;
  fa2 = feval(functname,x2);
avec = [aL a1 a2 au;faL fa1 fa2 fau];
for i=1:nmax
    if fa1>=fa2
        aL=a1;
        faL=fa1;
        a1=a2;
        fa1=fa2;
        a2 = tau*aL + (1-tau)*au;
        x2 = x+a2*s;
        fa2 = feval(functname,x2);
        au=au;
        fau=fau;
        avec = [aL a1 a2 au;faL fa1 fa2 fau];
    else
        au=a2;
        fau=fa2;
        a2=a1;
        fa2=fa1;
        a1=(1-tau)*aL+tau*au;
        x1=x+a1*s;
        fa1 = feval(functname,x1);
        aL=aL;
        faL=faL;
        avec = [aL a1 a2 au;faL fa1 fa2 fau];
    end
end
avec = [aL a1 a2 au;faL fa1 fa2 fau];
ReturnValue=[a1 fa1 x1];
end
```

EvaluationScript.m

```
clear all;clc;close all;
format short
tic
x=[-15.8981  -14.4656  -13.5060  -12.6621  -11.7163   20.4769
20.0263   19.6103];% potential result
n=5;
Z=19;Z2=55;        % number of teeth
m=2.5;                     % module
tooth=2; % Posa dodia aristera apo th symetria 8a einai to donti
%
div=500;%
Rtar=-(1.e-18)*zeros(1,div); % Target
%
do=Z*m;ro=do/2;            % [pcd] Pitch Circle Diamete
do2=Z2*m;ro2=do2/2;        % [pcd] Pitch Circle Diamete
```

```
dk=(Z+2)*m;rk=dk/2;        % [od] Outside diameter Cost pp 48
dk2=(Z2+2)*m;rk2=dk2/2;    % [od] Outside diameter Cost pp 48

a0=deg2rad(20);        % pressure angle
dg=do*cos(a0);         % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg=dg/2;               % [rbase]%

dg2=do2*cos(a0);       % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg2=dg2/2;             % [rbase]%
%
CP=[x(1:n)
sqrt((rk)^2-x(1)^2)    x(n+1:end)     sqrt((rg)^2-x(n)^2)];%
phi=-tooth*2*pi/Z;
CP0(1,:)=CP(1,:)*cos(phi)-CP(2,:)*sin(phi);
CP0(2,:)=CP(1,:)*sin(phi)+CP(2,:)*cos(phi);
clearvars CP
CP=CP0;
% n : Number of CP , proekupsan apo sxesh n+k+1-numel(T)  =  0
n=numel(CP)/2;
% k : deg of B-Spline , elaxisto 3
k=4;
% T : knot vector , numel(T)=n+k+1 , pollaplothta : 1 h parapanw
analoga
T=[zeros(1,k+1) 1:1:n-k-1 (n-k)*ones(1,k+1)];
% Basis Functions of B-Splines
t=T(1):(T(end)-T(1))/(div-1):T(end);
%
N=zeros(k+n,k+1,numel(t));
for it=1:numel(t)
    for i=1:numel(T)-1
       if t(it)>=T(i) && t(it) < T(i+1)
         N(i,1,it)=1;
       else
         N(i,1,it)=0;
       end
    end
    a=1;
    for j=2:k+1
       for i=1:numel(T)-1-a
          if  (N(i,j-1,it)==0 && N(i+1,j-1,it)==0)  % zero kai oi
duo oroi
              N(i,j,it)=0;
              if i<=n
                 dN(i,it)=0;
              end
          elseif N(i,j-1,it)==0                     % zero 1os oros
              N(i,j,it)=((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-
1,it);
              if i<=n
```

```
                 dN(i,it)=- (k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          elseif N(i+1,j-1,it)==0                % zero 2os oros
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it);
              end
          else
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-
1,it)+((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it) -
(k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          end
          if (i==n && j==k+1 && t(it)==T(end))
                N(:,:,it)=round(N(:,:,it-1),0);
                dN(:,it)=round(dN(:,it-1),0);
          end
       end
       a=a+1;
    end
% Nk(:,it)=N(1:n,k+1,it); % basis functions
end
for it=1:numel(t)
       for i=1:n
          a10= 1/(T(i+k)-  T(i));
          a11=-1/(T(i+k+1)-T(i+1));
          a20=a10/(T(i+k-1)-T(i));
          a21=(a11-a10)/(T(i+k)-T(i+1));
          a22=-a11/(T(i+k+1)-T(i+2));

          if     N(i,end-2,it)==0 && N(i+1,end-2,it)~=0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(                  a21*N(i+1,end-
2,it)+a22*N(i+2,end-2,it));

          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+
a22*N(i+2,end-2,it));

          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)~=0 &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+a21*N(i+1,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(
a22*N(i+2,end-2,it));
```

```
          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)~=0     &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(                 +a21*N(i+1,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)==0     &&
N(i+2,end-2,it)==0
          dN2(i,it)=0;
          else
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+a21*N(i+1,end-
2,it)+a22*N(i+2,end-2,it));
          end
       end

end
dN2(:,it)=flip(dN2(:,1));

for it=1:numel(t)
x1(it) =N(1:n,k+1,it)'*CP(1,:)';
y1(it) =N(1:n,k+1,it)'*CP(2,:)';
dx1(it)=dN(:,it)'*CP(1,:)';
dy1(it)=dN(:,it)'*CP(2,:)';
d2x1(it)=dN2(:,it)'*CP(1,:)';
d2y1(it)=dN2(:,it)'*CP(2,:)';
end
dydx1=dy1./dx1; % 1st Derivative of B-Splines

f=dy1;  % = dy/dt
g=dx1;  % = dx/dt

d2ydx1=((d2y1.*g-d2x1.*f)./(g.^2))./g; % 2nd Derivative of B-
Splines

rgG=abs(x1+y1.*dydx1)./(sqrt(1+dydx1.^2)); %1.50 dr V.Spita
rG=sqrt(x1.^2+y1.^2);
% Contact Line Coordinates
term1=((rG/ro).^2-(rgG/ro).^2);
term2=(1-(rgG/ro).^2);
xp=rgG.*(sqrt(term1)-sqrt(term2)); %1.48 dr V.Spita / Contact Line
term3=((ro./rgG).^2-1);
yp=sqrt(term3).*xp;                %1.49 dr V.Spita / Contact Line
%
for i=1:numel(xp)
A=[x1(i)       y1(i)
```

```
   y1(i)      -x1(i)];  % exw vgalei ta ro apo ton tupo 30 & 31 pp
41
rot(:,i)=inv(A)*[xp(i);yp(i)+ro];
end
thita=atan(rot(2,:)./rot(1,:));
K=thita*ro;
% rack
xr=xp-K;
yr=yp;
v=rot(2,:).^2+rot(1,:).^2; % sin^2 + cos^2 = 1
t0=K./ro;                            % Cost pp 39 (19)
% (xp,yp) coordinates ths Troxias Epafwm sta (x1_exw,y1_exw)
Sunergazomenos troxos coordinates
t2=-t0*ro/ro2;
x1_exw=xp.*cos(t2)-(yp-ro2).*sin(t2);              % Cost pp 40
(27)
y1_exw=xp.*sin(t2)+(yp-ro2).*cos(t2)+ro2;          % Cost pp 40
(28)
%
S1=gradient(y1_exw)./gradient(x1_exw);
sa=(S1(2)-S1(3))/(x1_exw(2)-x1_exw(3));
sb=S1(2)-sa*x1_exw(2);
S1(1)=sa*x1_exw(1)+sb;
sa=(S1(end-2)-S1(end-1))/(x1_exw(end-2)-x1_exw(end-1));
sb=S1(end-1)-sa*x1_exw(end-1);
S1(end)=sa*x1_exw(end)+sb;
S2=gradient(S1)    ./gradient(x1_exw);
sa=(S2(2)-S2(3))/(x1_exw(2)-x1_exw(3));
sb=S2(2)-sa*x1_exw(2);
S2(1)=sa*x1_exw(1)+sb;
sa=(S2(end-2)-S2(end-1))/(x1_exw(end-2)-x1_exw(end-1));
sb=S2(end-1)-sa*x1_exw(end-1);
S2(end)=sa*x1_exw(end)+sb;
%
R1=d2ydx1./((1+dydx1.^2).^(3/2)); % tupos (54) kostakis
R2=S2./((1+S1.^2).^(3/2));% tupos (54) kostakis
R=R1-R2;
dR=gradient(R)./gradient(x1);
sa=(dR(2)-dR(3))/(x1(2)-x1(3));
sb=dR(2)-sa*x1(2);
dR(1)=sa*x1(1)+sb;
sa=(dR(end-2)-dR(end-1))/(x1(end-2)-x1(end-1));
sb=dR(end-1)-sa*x1(end-1);
dR(end)=sa*x1(end)+sb;
fun=0.5*(R-Rtar).^2;
fun1=0.5*(dR-Rtar).^2;

fun=0.5*(R-Rtar).^2;
Y0=trapz(x1,fun);
kapa=abs(max(d2ydx1));
%
```

```
AB=0;
for i=1:div-1
AB=AB+sqrt((xp(i+1)-xp(i))^2+(yp(i+1)-yp(i))^2);
end
overlap=AB/(pi*dg/Z)
flag=0;
for i=1:numel(y1)-1
    if y1(i+1)>y1(i)
    flag=flag+1;
    end
end
flag1=0;
for i=1:numel(x1)-1
    if x1(i+1)<x1(i)
    flag1=flag1+1;
    end
end
flag2=0;
for i=1:numel(y1_exw)-1
    if y1_exw(i+1)>y1_exw(i)
    flag2=flag2+1;
    end
end
flag3=0;
for i=1:numel(x1_exw)-1
    if x1_exw(i+1)<x1_exw(i)
    flag3=flag3+1;
    end
end
C=[flag
   flag1 % ? mallon oxi
flag2
flag3
overlap];
hold on;axis equal;
plot(xp,yp+ro,'Color','blue','LineWidth',2)
c=[0;ro2];

for i=0:Z2-1
phi=i*2*pi/Z2;
Tr=[cos(phi) -sin(phi);
    sin(phi)  cos(phi)];
d=[x1_exw;y1_exw]-c;
xd=Tr*d+c;
plot(xd(1,:),xd(2,:)+ro,'Color',[0.9 0 0],'LineWidth',2)

end
for i=0:Z2-1
phi=i*2*pi/Z2;
Tr=[cos(phi) -sin(phi);
    sin(phi)  cos(phi)];
```

```
d=[-x1_exw;y1_exw]-c;
xd=Tr*d+c;
plot(xd(1,:),xd(2,:)+ro,'Color',[0.9 0 0],'LineWidth',2)
if i==1
    xh=xd(:,end);
end
end

dseg1=[x1(1) -x1(1);
       y1(1)  y1(1)];
for i=0:Z-1
phi=i*2*pi/Z;
xfc=dseg1(1,:)*cos(phi)-dseg1(2,:)*sin(phi);
yfc=dseg1(1,:)*sin(phi)+dseg1(2,:)*cos(phi);
plot(xfc,yfc,'Color',[0 0 0.5],'LineWidth',2)
end

circle(0,0,rk);
circle(0,0,rg);
circle(0,ro+ro2,rg2);
circle(0,ro+ro2,rk2);
rf=ro-1.25*m;
circle(0,0,ro);
circle(0,0,rf);
circle(0,ro+ro2,ro2);
circle(0,0,rk/5);
circle(0,ro+ro2,rk2/5);

title('ST-D,Open contact path,overlap=1.04')

xt = [-10 -6]-60;
yt = [-10 -16]+45;
str = {'Rolling circle','(Wheel)'};
text(xt,yt,str)
xt = [-10 -6]+35;
yt = [-10 -16]+20;
str = {'Rolling circle','(Pinion)'};
text(xt,yt,str)
xt = [35 -15]+35;
yt = [100 -40]+20;
str = {'Wheel','Pinion'};
text(xt,yt,str)
for i=0:Z-1
phi=i*2*pi/Z;
xc=x1*cos(phi)-(y1)*sin(phi);
yc=x1*sin(phi)+(y1)*cos(phi);
plot(xc,yc,'Color',[0 0 0.5],'LineWidth',2)
end
for i=0:Z-1
phi=i*2*pi/Z;
xc=-x1*cos(phi)-(y1)*sin(phi);
```

```
yc=-x1*sin(phi)+(y1)*cos(phi);
plot(xc,yc,'Color',[0 0 0.5],'LineWidth',2)
end

%% TROXOEIDES PINION
% SECTION 1
ar=(yr(end)-yr(end-1))/(xr(end)-xr(end-1));
br=yr(end)-ar*xr(end);
xf=(-1.25*m-br)/ar;
xr_trox=xr(end):0.01:xf;
yr_trox=ar*xr_trox+br;
%
hold on
x_=xr_trox;    % gia na emfanistei to 1o oxtari
y_=yr_trox;
dydx_=gradient(y_)./gradient(x_);
%
K_=-(y_.*dydx_+x_);                     % Cost pp 39 (22)
t_=K_./ro;                            % Cost pp 39 (19)
% Apo (x,y) coordinates tou Kanona sta (x1,y1) coordinates tou gear
flank
x1_=(x_+K_).*cos(t_)-(y_+ro).*sin(t_);    % Cost pp 39 (23)
y1_=(x_+K_).*sin(t_)+(y_+ro).*cos(t_)-ro; % Cost pp 39 (24)
% Apo (x,y) coordinates tou Kanona sta (x_te,y_te) ths Troxias
Epafwn
x_te_=x_+K_;                                         % Cost pp 40
(20)
y_te_=y_;                                           % Cost pp 40
(20)
% (x_te,y_te) coordinates ths Troxias Epafwm sta (x1_exw,y1_exw)
Sunergazomenos troxos coordinates
ro2=Z2*m/2;
t2_=-t_*ro/ro2;
x1_exw_=x_te_.*cos(t2_)-(y_te_-ro2).*sin(t2_);
y1_exw_=x_te_.*sin(t2_)+(y_te_-ro2).*cos(t2_)+ro2;

atrox=ar:0.01:0;
K__=-(yr_trox(end).*atrox+xr_trox(end));                     % Cost
pp 39 (22)
t__=K__./ro;                            % Cost pp 39 (19)
% Apo (x,y) coordinates tou Kanona sta (x1,y1) coordinates tou gear
flank
x1__=(xr_trox(end)+K__).*cos(t__)-(yr_trox(end)+ro).*sin(t__);    %
Cost pp 39 (23)
y1__=(xr_trox(end)+K__).*sin(t__)+(yr_trox(end)+ro).*cos(t__)-ro; %
Cost pp 39 (24)
hold on;axis equal;

% SECTION 2
ar_=(yr(1)-yr(2))/(xr(1)-xr(2));
br_=yr(1)-ar_*xr(1);
```

```
x_start=(1*m-br_)/ar_;
xr_trox_=x_start:0.01:xr(1);
yr_trox_=ar_*xr_trox_+br_;

Xrack=[xr_trox_ xr xr_trox];
Yrack=[yr_trox_ yr yr_trox];

rc=0.09*m;aj=608;

at=rad2deg(atan((Yrack(end)-Yrack(end-1))/(Xrack(end)-Xrack(end-
1))))+90;

xo=(cosd(at)+tand(at)*sind(at)-tand(at))*rc+Xrack(end);
yo=Yrack(end)+rc;

x_radi=Xrack(aj):0.001:xo;
y_radi=-sqrt(rc^2-(x_radi-xo).^2)+yo;

dydx_radi=gradient(y_radi)./gradient(x_radi);
%
K_radi=-(y_radi.*dydx_radi+x_radi);                     % Cost pp
39 (22)
t_radi=K_radi./ro;                                % Cost pp 39 (19)
% Apo (x,y) coordinates tou Kanona sta (x1,y1) coordinates tou gear
flank
x1_radi=(x_radi+K_radi).*cos(t_radi)-(y_radi+ro).*sin(t_radi);    %
Cost pp 39 (23)
y1_radi=(x_radi+K_radi).*sin(t_radi)+(y_radi+ro).*cos(t_radi)-ro; %
Cost pp 39 (24)
plot(x1_,y1_+ro,'Color',[0 0 0.5],'LineWidth',2)
plot(-x1_,y1_+ro,'Color',[0 0 0.5],'LineWidth',2)
plot(x1_radi,y1_radi+ro,'Color',[0 0 0.5],'LineWidth',2)
plot(-x1_radi,y1_radi+ro,'Color',[0 0 0.5],'LineWidth',2)

p=Z2*m*pi/2;
XrackPI=Xrack*cos(pi)-Yrack*sin(pi) +p;XrackPI=flip(XrackPI);
YrackPI=Xrack*sin(pi)+Yrack*cos(pi) ;YrackPI=flip(YrackPI);

%% TROXOEIDES WHEEL
arpi=(YrackPI(end)-YrackPI(end-1))/(XrackPI(end)-XrackPI(end-1));
brpi=YrackPI(end)-arpi*XrackPI(end);
xfpi=(-1.25*m-brpi)/arpi;
xr_troxPI=XrackPI(end):0.01:xfpi;
yr_troxPI=arpi*xr_troxPI+brpi;

y_=[YrackPI  yr_troxPI];
x_=[XrackPI  xr_troxPI];
aj2=642;

at2=rad2deg(atan((y_(end)-y_(end-1))/(x_(end)-x_(end-1))))+90;
```

```
xo2=(cosd(at2)+tand(at2)*sind(at2)-tand(at2))*rc+x_(end);
yo2=y_(end)+rc;

x_radi2=x_(aj2):0.001:xo2;
y_radi2=-sqrt(rc^2-(x_radi2-xo2).^2)+yo2;

dydx_radi=gradient(y_radi2)./gradient(x_radi2);
%
K_radi2=-(y_radi2.*dydx_radi+x_radi2);                     % Cost
pp 39 (22)
t_radi2=K_radi2./ro2;                                 % Cost pp 39 (19)
% Apo (x,y) coordinates tou Kanona sta (x1,y1) coordinates tou gear
flank
x1_radi2=(x_radi2+K_radi2).*cos(t_radi2)-
(y_radi2+ro2).*sin(t_radi2);    % Cost pp 39 (23)
y1_radi2=(x_radi2+K_radi2).*sin(t_radi2)+(y_radi2+ro2).*cos(t_radi2
)-ro2; % Cost pp 39 (24)

plot(x1_radi2,y1_radi2+ro2+ro+ro2,'Color',[0.9 0 0],'LineWidth',2)
% <----- thmhma B / Troxoeides
plot(-x1_radi2,y1_radi2+ro2+ro+ro2,'Color',[0.9 0 0],'LineWidth',2)

dydx_=gradient(y_)./gradient(x_);
%
K_=-(y_.*dydx_+x_);                             % Cost pp 39 (22)
t_=K_./ro2;                                    % Cost pp 39 (19)
% Apo (x,y) coordinates tou Kanona sta (x1,y1) coordinates tou gear
flank
x1_=(x_+K_).*cos(t_)-(y_+ro2).*sin(t_);     % Cost pp 39 (23)
y1_=(x_+K_).*sin(t_)+(y_+ro2).*cos(t_)-ro2; % Cost pp 39 (24)

plot(x1_(45:end),y1_(45:end)+ro2+ro+ro2,'Color',[0.9 0
0],'LineWidth',2)
plot(-x1_(45:end),y1_(45:end)+ro2+ro+ro2,'Color',[0.9 0
0],'LineWidth',2)

rf2=ro2-1.25*m;
box on
% set(gca,'visible','off')
hold off
figure2 = figure;
hold on;
plot(x1,R,'Color',[0 0 0.5],'LineWidth',2)

function h = circle(x,y,r)
hold on
th = 0:pi/50:2*pi;
xunit = r * cos(th) + x;
yunit = r * sin(th) + y;
h = plot(xunit, yunit,'-.','Color','k','LineWidth',1);
```

```
end
```

### SinRackGear.m

This script is to compare the equivalent curvature ($1/R$) of optimal gears with the Sin-Rack Gears and it is independent of the optimization method chosen.

```
clear all;clc
format long
Z=19;Z2=55;m=2.5;              % number of teeth  % module

h=2*m;             %height imitonoeidous kanona (mm)
div=1e5;              %#of nodes
% Gear parameters
dk=(Z+2)*m;rk=dk/2;   %[od] Outside diameter Cost pp 48
df=(Z-2.5)*m;         %[rd] Root diameter    Cost pp 48
do=Z*m;ro=do/2;       %[pcd] Pitch Circle Diamete
So=(0.475*pi)*do/Z;   %(0.475-0.5) Tooth thickness at pcd  Cost pp
46 (7)
rc=ro;                %kuklos kulhshs
% (x,y) coordinates tou Kanona Cost pp 38-39
% x=0:(Z*m*pi)/(div-1):Z*m*pi;
x=0:(1*m*pi/2)/(div-1):1*m*pi/2;     % gia na emfanistei to 1o
oxtari
y=(h/2)*cos(Z*x/rc);
dydx=-(h/2)*(Z/rc)*sin((Z/rc)*x);
%
K=-(y.*dydx+x);                     % Cost pp 39 (22)
t=K./ro;                            % Cost pp 39 (19)
% Apo (x,y) coordinates tou Kanona sta (x1,y1) coordinates tou gear
flank
x1=(x+K).*cos(t)-(y+ro).*sin(t);     % Cost pp 39 (23)
y1=(x+K).*sin(t)+(y+ro).*cos(t)-ro; % Cost pp 39 (24)
% Apo (x,y) coordinates tou Kanona sta (x_te,y_te) ths Troxias
Epafwn
x_te=x+K;                                         % Cost pp 40 (20)
y_te=y;                                           % Cost pp 40 (20)
% (x_te,y_te) coordinates ths Troxias Epafwm sta (x1_exw,y1_exw)
Sunergazomenos troxos coordinates
ro2=Z2*m/2;
t2=-t*ro/ro2;
x1_exw=x_te.*cos(t2)-(y_te-ro2).*sin(t2);
y1_exw=x_te.*sin(t2)+(y_te-ro2).*cos(t2)+ro2;
% Aktina kabilothtas - Gear 1
dydx_1=gradient(y1)./gradient(x1);                   %
derivative(x1,y1);
d2ydx_1=gradient(dydx_1)./gradient(x1);     %
derivative(x1,dydx_1);
R1=d2ydx_1./((1+dydx_1.^2).^(3/2)); % tupos (54) kostakis
% Aktina kabilothtas - Gear 2
dydx1_exw=gradient(y1_exw)./gradient(x1_exw);
d2ydx1_exw=gradient(dydx1_exw)./gradient(x1_exw);
R2=d2ydx1_exw./((1+dydx1_exw.^2).^(3/2));% tupos (54) kostakis
```

```
%
R=R1-R2;
```

### InvoluteGear.m

This script is to compare the equivalent curvature ($1/R$) of optimal gears with the Involute Gears and it is independent of the optimization method chosen.

```
clear all
close all;clc;
%% input data
Z=19;Z2=55;           % number of teeth
a0=deg2rad(20);       % pressure angle
m=2.5;                  % module
div=1e5;              % #of nodes
do=Z*m;ro=do/2;           % [pcd] Pitch Circle Diamete
do2=Z2*m;ro2=do2/2;       % [pcd] Pitch Circle Diamete
dk=(Z+2)*m;rk=dk/2;       % [od] Outside diameter Cost pp 48
dk2=(Z2+2)*m;rk2=dk2/2;   % [od] Outside diameter Cost pp 48

a0=deg2rad(20);       % pressure angle
dg=do*cos(a0);        % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg=dg/2;              % [rbase]%

dg2=do2*cos(a0);      % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg2=dg2/2;            % [rbase]%
%% Gear parameters
So=(0.475*pi)*do/Z;   % (0.475-0.5) Tooth thickness at pcd  Cost pp
46 (7)
inva0 = tan(a0) - a0; % Involute function
xo=0;%-rg*sin(a0);
% (x_te,y_te) coordinates ths Troxias Epafwm
x_te=linspace(-rg*sin(a0),rg*sin(a0),div); % Sunolikh troxia epafwn
y_te=tan(a0)*x_te;           % dierxetai apo to (0,0) Fine!
% (x,y) coordinates tou Kanona Cost pp 38-39
x=(-(tan(a0))^2)*x_te+(1+(tan(a0))^2)*xo;  % Cost pp 36
y=y_te;                                    % Cost pp 36
dydx=(-1/tan(a0));
%%
K=-(y.*dydx+x);                     % Cost pp 39 (22)
t=K./ro;                            % Cost pp 39 (19)
% Apo (x,y) coordinates tou Kanona sta (x1,y1) coordinates tou gear
flank
x1=(x+K).*cos(t)-(y+ro).*sin(t);    % Cost pp 39 (23)
y1=(x+K).*sin(t)+(y+ro).*cos(t)-ro; % Cost pp 39 (24)
phig = 0;
x1e = cos(phig)*x1 - sin(phig)*(y1+ro);
y1e = sin(phig)*x1 + cos(phig)*(y1+ro);
x1=x1e;y1=y1e;

% (x_te,y_te) coordinates ths Troxias Epafwm sta (x1_exw,y1_exw)
Sunergazomenos troxos coordinates
```

```
ro2=Z2*m/2;
t2=-t*ro/ro2;
x1_exw=x_te.*cos(t2)-(y_te-ro2).*sin(t2);
y1_exw=x_te.*sin(t2)+(y_te-ro2).*cos(t2)+ro2;
% Aktina kabilothtas - Gear 1
dydx_1=gradient(y1)./gradient(x1);
d2ydx_1=gradient(dydx_1)./gradient(x1);
R1=d2ydx_1./((1+dydx_1.^2).^(3/2)); % tupos (54) kostakis
% Aktina kabilothtas - Gear 2
dydx1_exw=gradient(y1_exw)./gradient(x1_exw);
d2ydx1_exw=gradient(dydx1_exw)./gradient(x1_exw);
R2=d2ydx1_exw./((1+dydx1_exw.^2).^(3/2));% tupos (54) kostakis
%
R=R1-R2;
axis equal;hold on;grid on;
plot(x1,y1)
plot(x1(9000:end),R(9000:end)+ro)
%
```

## Genetic Algorithm

### GA_start_code.m

```
clc;clear all;close all;
tic
numelCP=7;
InitialPopulationMatrix_Data=LeastmS(numelCP);
PopulationSize_Data=300;    % population size
CrossoverFraction_Data=0.8; % crossover fraction
MaxGenerations_Data=100*(4*numelCP-5);
%
m=2.5;
nvars=2*(numelCP-2); % number of variables
lb=[        zeros(1,nvars/2)    -m*ones(1,nvars/2)];
ub=[(m*pi/2)*ones(1,nvars/2)     m*ones(1,nvars/2)];

Aineq=zeros(nvars,nvars);
bineq=zeros(nvars,1);
%
[x,fval,exitflag,output,population,score] =
fitness_ga_code(nvars,...
Aineq,bineq,lb,ub,InitialPopulationMatrix_Data,PopulationSize_Data,
CrossoverFraction_Data,MaxGenerations_Data);
toc
```

### fitness_ga_code.m

```
function [x,fval,exitflag,output,population,score] =
fitness_ga_code(nvars,Aineq,bineq,lb,ub,InitialPopulationMatrix_Dat
a,PopulationSize_Data,CrossoverFraction_Data,MaxGenerations_Data)
% This is an auto generated MATLAB file from Optimization Tool.

% Start with the default options
options = optimoptions('gamultiobj');
% Modify options setting
% options = optimoptions(options,'InitialPopulationRange',
InitialPopulationRange_Data);
options = optimoptions(options,'InitialPopulationMatrix',
InitialPopulationMatrix_Data);
options = optimoptions(options,'PopulationSize',
PopulationSize_Data);
options = optimoptions(options,'CrossoverFraction',
CrossoverFraction_Data);
options = optimoptions(options,'MaxGenerations',
MaxGenerations_Data);
% options = optimoptions(options,'CrossoverFcn', {
@crossoverintermediate [] });
% options = optimoptions(options,'MutationFcn', @YourFcnNameHere);
options = optimoptions(options,'Display', 'off');
[x,fval,exitflag,output,population,score] = ...%
gamultiobj(@fitness,nvars,Aineq,bineq,[],[],lb,ub,@nonlinear_constr
aints,options);
%
gamultiobj(@fitness_function,nvars,Aineq,bineq,[],[],lb,ub,options)
;
```

LeastmS.m

```
%% LEAST MIN SQUARES
% eisagontai ta theorytika (x,y) tou rack kai pianoume thn katatomh
me
% elaxista tetragwnaZ
function b=LeastmS(n)
% Kost pp 44 (23) means Kostopoulos selida 44 sxesh 23
%% Input Data SIN - (x,y) coordinates ths Troxias Epafwn  P : Data
to interpollate P(:,1)=x;P(:,2)=y;
Z=19;Z2=55;m=2.5;                  % number of teeth  % module
h=2*m;                 %height imitonoeidous kanona (mm)
div=1e5;                  %#of nodes
% Gear parameters
dk=(Z+2)*m;               %[od] Outside diameter Cost pp 48
do=Z*m;ro=do/2;           %[pcd] Pitch Circle Diamete
rc=ro;                    %kuklos kulhshs
x=0:(1*m*pi/2)/(div-1):1*m*pi/2;     % gia na emfanistei to 1o
oxtari
y=(h/2)*cos(Z*x/rc);
P(:,1)=x';P(:,2)=y';
div=size(P);
%% Input of B-Spline parameters
% n : Number of CP , proekupsan apo sxesh n+k+1-numel(T)  =  0
% k : deg of B-Spline , elaxisto 3
k=4;
% T : knot vector , numel(T)=n+k+1 , pollaplothta : 1 h parapanw
analoga
T=[zeros(1,k+1) 1:1:n-k-1 (n-k)*ones(1,k+1)];
% Basis Functions of B-Splines
t=T(1):(T(end)-T(1))/(div(1)-1):T(end);
%
```

```
N=zeros(k+n,k+1,numel(t));
for it=1:numel(t)
    for i=1:numel(T)-1
       if t(it)>=T(i)  && t(it) < T(i+1)
         N(i,1,it)=1;
       else
         N(i,1,it)=0;
       end
    end
    a=1;
    for j=2:k+1
       for i=1:numel(T)-1-a
          if  (N(i,j-1,it)==0 && N(i+1,j-1,it)==0)  % zero kai oi
duo oroi
              N(i,j,it)=0;
              if i<=n
                 dN(i,it)=0;
              end
          elseif N(i,j-1,it)==0                  % zero 1os oros
              N(i,j,it)=((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-
1,it);
              if i<=n
                 dN(i,it)=- (k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          elseif N(i+1,j-1,it)==0                % zero 2os oros
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it);
              end
          else
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-
1,it)+((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it) -
(k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          end
          if (i==n && j==k+1 && t(it)==T(end))
                N(:,:,it)=round(N(:,:,it-1),0);
                dN(:,it)=round(dN(:,it-1),0);
          end
       end
       a=a+1;
    end
% Nk(:,it)=N(1:n,k+1,it); % basis functions
end
%% Least mean Squares
B_sp_k=zeros(n,numel(t));
for i=1:n
B_sp_k(i,:)=N(i,k+1,:);
end
```

```
CP(1,:)=inv(B_sp_k*transpose(B_sp_k))*B_sp_k*P(:,1);  % Euresh
Control Points
CP(2,:)=inv(B_sp_k*transpose(B_sp_k))*B_sp_k*P(:,2);
%%
CP(1,1)=P(1,1); CP(1,end)=P(end,1);    % To prwto kai to teleutaio
CP
CP(2,1)=P(1,2); CP(2,end)=P(end,2);    % ta orizw me vash ta shmeia
paremvolhs
% CP(2,2)=CP(2,1);
b=[CP(1,2:end-1),CP(2,2:end-1)];
```

GradFun.m

```
function Return = GradFun(functname,x)
hstep=1.0e-3;
n = length(x);
f = feval(functname,x);
for i=1:n
    xs = x;
    xs(i) = xs(i) + hstep;
    gradx(i) = (feval(functname,xs)-f)/hstep;
end
Return = gradx;
end
```

fitness.m

```
function Y = fitness(x)
Z=19;Z2=55;m=2.5;              % number of teeth  % module
dk=(Z+2)*m;rk=dk/2;   % [od] Outside diameter Cost pp 48
a0=deg2rad(20);       % pressure angle
do=Z*m;ro=do/2;       % [pcd] Pitch Circle Diameter
dg=do*cos(a0);        % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg=dg/2;              % [rbase]
div=1e3;%
Rtar=-(1.e-3)*ones(div,1); % Target
% PX=x(1:numel(x)/2+1);
% PY=[sqrt(rk^2-x(1)^2)-ro, x(numel(x)/2+2:end) ,sqrt(rg^2-
x(numel(x)/2+2)^2)-ro];
CP=[0   x(1:numel(x)/2)      m*pi/2
    m   x(numel(x)/2+1:end)     -m];

% Input data 1 - Gear parameters
% n : Number of CP , proekupsan apo sxesh n+k+1-numel(T)   =   0
n=numel(CP)/2;
% k : deg of B-Spline , elaxisto 3
k=4;
% T : knot vector , numel(T)=n+k+1 , pollaplothta : 1 h parapanw
analoga
T=[zeros(1,k+1) 1:1:n-k-1 (n-k)*ones(1,k+1)];
% Basis Functions of B-Splines
t=T(1):(T(end)-T(1))/(div-1):T(end);
```

```
%
N=zeros(k+n,k+1,numel(t));
for it=1:numel(t)
    for i=1:numel(T)-1
       if t(it)>=T(i) && t(it) < T(i+1)
         N(i,1,it)=1;
       else
         N(i,1,it)=0;
       end
    end
    a=1;
    for j=2:k+1
       for i=1:numel(T)-1-a
          if  (N(i,j-1,it)==0 && N(i+1,j-1,it)==0)  % zero kai oi
duo oroi
              N(i,j,it)=0;
              if i<=n
                 dN(i,it)=0;
              end
          elseif N(i,j-1,it)==0                  % zero 1os oros
              N(i,j,it)=((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-
1,it);
              if i<=n
                 dN(i,it)=-  (k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          elseif N(i+1,j-1,it)==0                % zero 2os oros
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it);
              end
          else
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-
1,it)+((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it) -
(k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          end
          if (i==n && j==k+1 && t(it)==T(end))
                N(:,:,it)=round(N(:,:,it-1),0);
                dN(:,it)=round(dN(:,it-1),0);
          end
       end
       a=a+1;
    end
% Nk(:,it)=N(1:n,k+1,it); % basis functions
end
%
C0=zeros(numel(t),2);
for it=1:numel(t)
C0(it,1) =N(1:n,k+1,it)'*CP(1,:)';
```

```
C0(it,2) =N(1:n,k+1,it)'*CP(2,:)';
dC0(it,1)=dN(:,it)'*CP(1,:)';
dC0(it,2)=dN(:,it)'*CP(2,:)';
end
dydx=dC0(:,2)./dC0(:,1); % 1st Derivative of B-Splines
interpolation
% Metatopish Kanona
K=-(C0(:,2).*dydx+C0(:,1));           % Cost pp 39 (22)
t0=K./ro;                             % Cost pp 39 (19)
% Apo (x,y) coordinates tou Kanona sta (x1,y1) coordinates tou gear
flank
x1=(C0(:,1)+K).*cos(t0)-(C0(:,2)+ro).*sin(t0);    % Cost pp 39 (23)
y1=(C0(:,1)+K).*sin(t0)+(C0(:,2)+ro).*cos(t0)-ro; % Cost pp 39 (24)
% y1=y1+ro;
%
x_te=C0(:,1)+K;                                         % Cost pp
40 (20)
y_te=C0(:,2);
% (x_te,y_te) coordinates ths Troxias Epafwm sta (x1_exw,y1_exw)
Sunergazomenos troxos coordinates
ro2=Z2*m/2;
t2=-t0*ro/ro2;
x1_exw=x_te.*cos(t2)-(y_te-ro2).*sin(t2);              % Cost pp 40
(27)
y1_exw=x_te.*sin(t2)+(y_te-ro2).*cos(t2)+ro2;          % Cost pp 40
(28)
% Aktina kabilothtas - Gear 1
dydx_1=gradient(y1)./gradient(x1);                   %
derivative(x1,y1);
d2ydx_1=gradient(dydx_1)./gradient(x1);      %
derivative(x1,dydx_1);
R1=d2ydx_1./((1+dydx_1.^2).^(3/2)); % tupos (54) kostakis
% Aktina kabilothtas - Gear 2
dydx1_exw0=gradient(y1_exw)./gradient(x1_exw);             %
derivative(x1_exw,y1_exw);
d2ydx1_exw0=gradient(dydx1_exw0)./gradient(x1_exw);%
derivative(x1_exw,dydx1_exw);
R2=d2ydx1_exw0./((1+dydx1_exw0.^2).^(3/2));% tupos (54) kostakis
% R=gradient(R1-R2)./gradient(x1);%270
R=R1-R2;
fun=0.5*(R-Rtar).^2;
sum=[0 0];
for i=2:2:div-1
sum(1)=sum(1)+4*fun(i);
end
for i=3:2:div-2
sum(2)=sum(2)+2*fun(i);
end
Y=(fun(1)+fun(div)+sum(1)+sum(2))/(3*(div-1));
```

nonlinear_constraints.m

```
function [C,Ceq] = nonlinear_constraints(x)
Z=19;Z2=55;m=2.5;                 % number of teeth  % module
dk=(Z+2)*m;rk=dk/2;    % [od] Outside diameter Cost pp 48
a0=deg2rad(20);        % pressure angle
do=Z*m;ro=do/2;        % [pcd] Pitch Circle Diameter
dg=do*cos(a0);         % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg=dg/2;               % [rbase]
div=1e3;%
Rtar=-(1.e-3)*ones(div,1); % Target
% PX=x(1:numel(x)/2+1);
% PY=[sqrt(rk^2-x(1)^2)-ro, x(numel(x)/2+2:end) ,sqrt(rg^2-
x(numel(x)/2+2)^2)-ro];
% CP=[0   x(1:numel(x)/2)      m*pi/2
%     m   x(numel(x)/2+1:end)      -m];
CP=[1.156000000000000   x(1:numel(x)/2)      2.87
    1.948416103110457   x(numel(x)/2+1:end)      -
2.760760193834766];
% Input data 1 - Gear parameters
% n : Number of CP , proekupsan apo sxesh n+k+1-numel(T)  =  0
n=numel(CP)/2;
% k : deg of B-Spline , elaxisto 3
k=4;
% T : knot vector , numel(T)=n+k+1 , pollaplothta : 1 h parapanw
analoga
T=[zeros(1,k+1) 1:1:n-k-1 (n-k)*ones(1,k+1)];
% Basis Functions of B-Splines
t=T(1):(T(end)-T(1))/(div-1):T(end);
%
N=zeros(k+n,k+1,numel(t));
for it=1:numel(t)
    for i=1:numel(T)-1
       if t(it)>=T(i) && t(it) < T(i+1)
         N(i,1,it)=1;
       else
         N(i,1,it)=0;
       end
    end
    a=1;
    for j=2:k+1
       for i=1:numel(T)-1-a
          if  (N(i,j-1,it)==0 && N(i+1,j-1,it)==0)  % zero kai oi
duo oroi
              N(i,j,it)=0;
              if i<=n
                 dN(i,it)=0;
              end
          elseif N(i,j-1,it)==0                    % zero los oros
              N(i,j,it)=((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-
1,it);
```

```
              if i<=n
                 dN(i,it)=- (k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          elseif N(i+1,j-1,it)==0                      % zero 2os oros
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it);
              end
          else
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-
1,it)+((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it) -
(k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          end
          if (i==n && j==k+1 && t(it)==T(end))
                N(:,:,it)=round(N(:,:,it-1),0);
                dN(:,it)=round(dN(:,it-1),0);
          end
       end
       a=a+1;
    end
% Nk(:,it)=N(1:n,k+1,it); % basis functions
end
%
C0=zeros(numel(t),2);
for it=1:numel(t)
C0(it,1) =N(1:n,k+1,it)'*CP(1,:)';
C0(it,2) =N(1:n,k+1,it)'*CP(2,:)';
dC0(it,1)=dN(:,it)'*CP(1,:)';
dC0(it,2)=dN(:,it)'*CP(2,:)';
end
dydx=dC0(:,2)./dC0(:,1); % 1st Derivative of B-Splines
interpolation
% Metatopish Kanona
K=-(C0(:,2).*dydx+C0(:,1));          % Cost pp 39 (22)
t0=K./ro;                            % Cost pp 39 (19)
% Apo (x,y) coordinates tou Kanona sta (x1,y1) coordinates tou gear
flank
x1=(C0(:,1)+K).*cos(t0)-(C0(:,2)+ro).*sin(t0);    % Cost pp 39 (23)
y1=(C0(:,1)+K).*sin(t0)+(C0(:,2)+ro).*cos(t0)-ro; % Cost pp 39 (24)
y1=y1+ro;
%
x_te=C0(:,1)+K;                                         % Cost pp
40 (20)
y_te=C0(:,2);
% (x_te,y_te) coordinates ths Troxias Epafwm sta (x1_exw,y1_exw)
Sunergazomenos troxos coordinates
ro2=Z2*m/2;
t2=-t0*ro/ro2;
```

```
x1_exw=x_te.*cos(t2)-(y_te-ro2).*sin(t2);               % Cost pp 40
(27)
y1_exw=x_te.*sin(t2)+(y_te-ro2).*cos(t2)+ro2;           % Cost pp 40
(28)
% Aktina kabilothtas - Gear 1
dydx_1=gradient(y1)./gradient(x1);                     %
derivative(x1,y1);
d2ydx_1=gradient(dydx_1)./gradient(x1);     %
derivative(x1,dydx_1);
R1=d2ydx_1./((1+dydx_1.^2).^(3/2)); % tupos (54) kostakis
% Aktina kabilothtas - Gear 2
dydx1_exw0=gradient(y1_exw)./gradient(x1_exw);             %
derivative(x1_exw,y1_exw);
d2ydx1_exw0=gradient(dydx1_exw0)./gradient(x1_exw);%
derivative(x1_exw,dydx1_exw);
R2=d2ydx1_exw0./((1+dydx1_exw0.^2).^(3/2));% tupos (54) kostakis
% R=gradient(R1-R2)./gradient(x1);%270
R=R1-R2;
AB=0;
for i=1:numel(x_te)-1
AB=AB+sqrt((x_te(i+1)-x_te(i))^2+(y_te(i+1)-y_te(i))^2);
end
overlap=AB/(pi*dg/Z);
flag=0;
for i=1:numel(y1)-1
    if y1(i+1)>y1(i) || x1(i+1)<x1(i)
    flag=flag+1;
    end
end
C=[flag;
... %    (0.751890311618654-x1(1));
    max(abs(R))-0.2
   -(overlap-1)];
Ceq =[];%abs(R1-R2)-0.05;
```

EvaluationScript.m

```
clc;clear all;close all;
format short
x=[0.6139    1.4593    2.1734    2.9364    3.5574    2.4978
1.9106    0.0138   -1.2411   -2.1062];

Z=19;Z2=55;m=2.5;      % number of teeth  % module
dk=(Z+2)*m;rk=dk/2;    % [od] Outside diameter Cost pp 48
dk2=(Z2+2)*m;rk2=dk2/2;    % [od] Outside diameter Cost pp 48
a0=deg2rad(20);        % pressure angle
do=Z*m;ro=do/2;        % [pcd] Pitch Circle Diameter
dg=do*cos(a0);         % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg=dg/2;               % [rbase]
div=1e3;%
```

```
Rtar=-(1.e-3)*ones(div,1); % Target

CP=[0   x(1:numel(x)/2)      m*pi/2
    m   x(numel(x)/2+1:end)     -m];
% Input data 1 - Gear parameters
% n : Number of CP , proekupsan apo sxesh n+k+1-numel(T)  =  0
n=numel(CP)/2;
% k : deg of B-Spline , elaxisto 3
k=4;
% T : knot vector , numel(T)=n+k+1 , pollaplothta : 1 h parapanw
analoga
T=[zeros(1,k+1) 1:1:n-k-1 (n-k)*ones(1,k+1)];
% Basis Functions of B-Splines
t=T(1):(T(end)-T(1))/(div-1):T(end);
%
N=zeros(k+n,k+1,numel(t));
for it=1:numel(t)
    for i=1:numel(T)-1
       if t(it)>=T(i) && t(it) < T(i+1)
         N(i,1,it)=1;
       else
         N(i,1,it)=0;
       end
    end
    a=1;
    for j=2:k+1
       for i=1:numel(T)-1-a
          if  (N(i,j-1,it)==0 && N(i+1,j-1,it)==0)  % zero kai oi
duo oroi
              N(i,j,it)=0;
              if i<=n
                 dN(i,it)=0;
              end
          elseif N(i,j-1,it)==0                  % zero 1os oros
              N(i,j,it)=((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-
1,it);
              if i<=n
                 dN(i,it)=-  (k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          elseif N(i+1,j-1,it)==0                % zero 2os oros
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it);
              end
          else
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-
1,it)+((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it) -
(k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          end
          if (i==n && j==k+1 && t(it)==T(end))
                N(:,:,it)=round(N(:,:,it-1),0);
                dN(:,it)=round(dN(:,it-1),0);
          end
       end
       a=a+1;
    end
% Nk(:,it)=N(1:n,k+1,it); % basis functions
end
%
C0=zeros(numel(t),2);
for it=1:numel(t)
C0(it,1) =N(1:n,k+1,it)'*CP(1,:)';
C0(it,2) =N(1:n,k+1,it)'*CP(2,:)';
dC0(it,1)=dN(:,it)'*CP(1,:)';
dC0(it,2)=dN(:,it)'*CP(2,:)';
end
dydx=dC0(:,2)./dC0(:,1); % 1st Derivative of B-Splines
interpolation
% Metatopish Kanona
K=-(C0(:,2).*dydx+C0(:,1));          % Cost pp 39 (22)
t0=K./ro;                            % Cost pp 39 (19)
% Apo (x,y) coordinates tou Kanona sta (x1,y1) coordinates tou gear
flank
x1=(C0(:,1)+K).*cos(t0)-(C0(:,2)+ro).*sin(t0);     % Cost pp 39 (23)
y1=(C0(:,1)+K).*sin(t0)+(C0(:,2)+ro).*cos(t0)-ro; % Cost pp 39 (24)
% y1=y1+ro;
%
x_te=C0(:,1)+K;                                          % Cost pp
40 (20)
y_te=C0(:,2);
% (x_te,y_te) coordinates ths Troxias Epafwm sta (x1_exw,y1_exw)
Sunergazomenos troxos coordinates
ro2=Z2*m/2;
t2=-t0*ro/ro2;
x1_exw=x_te.*cos(t2)-(y_te-ro2).*sin(t2);              % Cost pp 40
(27)
y1_exw=x_te.*sin(t2)+(y_te-ro2).*cos(t2)+ro2;          % Cost pp 40
(28)
% Aktina kabilothtas - Gear 1
dydx_1=gradient(y1)./gradient(x1);                   %
derivative(x1,y1);
d2ydx_1=gradient(dydx_1)./gradient(x1);     %
derivative(x1,dydx_1);
R1=d2ydx_1./((1+dydx_1.^2).^(3/2)); % tupos (54) kostakis
% Aktina kabilothtas - Gear 2
dydx1_exw0=gradient(y1_exw)./gradient(x1_exw);             %
derivative(x1_exw,y1_exw);
d2ydx1_exw0=gradient(dydx1_exw0)./gradient(x1_exw);%
derivative(x1_exw,dydx1_exw);
```

```
R2=d2ydx1_exw0./((1+dydx1_exw0.^2).^(3/2));% tupos (54) kostakis
R=R1-R2;%270
% dR=GradFun(Rcurva,x);
fun=0.5*(R-Rtar).^2;
sum=[0 0];
for i=2:2:div-1
sum(1)=sum(1)+4*fun(i);
end
for i=3:2:div-2
sum(2)=sum(2)+2*fun(i);
end
Y=(fun(1)+fun(div)+sum(1)+sum(2))/(3*(div-1));
AB=0;
for i=1:865%div-1
AB=AB+sqrt((x_te(i+1)-x_te(i))^2+(y_te(i+1)-y_te(i))^2);
end
overlap=AB/(pi*dg/Z)
flag=0;
for i=1:numel(y1)-1
    if y1(i+1)>y1(i)
    flag=flag+1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
mhden=x1.^2+y1.^2-rg.^2;
% d0=865;
d0=865;
circle(0,0,rg);
circle(0,0,rk);

x1g=x1(1:d0);y1g=y1(1:d0);
x1g_e=x1_exw(1:d0)';y1g_e=y1_exw(1:d0)';

clearvars x1 y1 x1_exw y1_exw
x1=x1g;y1=y1g+ro;x1_exw=x1g_e;y1_exw=y1g_e;
hold on;axis equal;

plot(x_te(1:d0),y_te(1:d0)+ro,'Color','blue','LineWidth',2)
c=[0;ro2];
for i=0:Z2-1
phi=i*2*pi/Z2;
Tr=[cos(phi) -sin(phi);
    sin(phi)  cos(phi)];
d=[x1_exw;y1_exw]-c;
xd=Tr*d+c;
plot(xd(1,:),xd(2,:)+ro,'Color',[0.9 0 0],'LineWidth',2)
end
for i=0:Z2-1
phi=i*2*pi/Z2;
Tr=[cos(phi) -sin(phi);
```

```
    sin(phi)  cos(phi)];
d=[-x1_exw;y1_exw]-c;
xd=Tr*d+c;
plot(xd(1,:),xd(2,:)+ro,'Color',[0.9 0 0],'LineWidth',2)
if i==1
    xh=xd(:,end);
end
end
% plot(x1(1),y1(1),'o')
% plot(-x1(1),y1(1),'o')

dseg1=[x1(1) -x1(1);
       y1(1)  y1(1)];
% plot(dseg1(1,:),dseg1(2,:),'Color',[0 0 0.5],'LineWidth',1)
for i=0:Z-1
phi=i*2*pi/Z;
xfc=dseg1(1,:)*cos(phi)-dseg1(2,:)*sin(phi);
yfc=dseg1(1,:)*sin(phi)+dseg1(2,:)*cos(phi);
% plot(xfc,yfc,'Color',[0 0 0.5],'LineWidth',2)
end
dseg=[x1_exw(end)    xh(1);
      y1_exw(end)+ro xh(2)+ro];
% plot(dseg(1,:),dseg(2,:),'Color',[0.9 0 0],'LineWidth',1)
for i=0:Z2-1
phi=i*2*pi/Z2;
Tr=[cos(phi) -sin(phi);
    sin(phi)  cos(phi)];
d=[dseg(1,:);dseg(2,:)-ro]-c;
xdseg=Tr*d+c;
% plot(xdseg(1,:),xdseg(2,:)+ro,'Color',[0.9 0 0],'LineWidth',2)
end
circle(0,0,ro);
do2=Z2*m;ro2=do2/2;       % [pcd] Pitch Circle Diamete

dg2=do2*cos(a0);      % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg2=dg2/2;            % [rbase]%

% plot(x1,R,'Color',[0 0 0.5],'LineWidth',2)
circle(0,ro+ro2,ro2);
circle(0,ro+ro2,rg2);
circle(0,ro+ro2,rk2);

circle(0,0,rk/5);
circle(0,ro+ro2,rk2/5);
plot([0  0],[0 ro+ro2+80],'-.','Color','k','LineWidth',1)
plot([0  ro+ro2],[ro ro],'-.','Color','k','LineWidth',1)

xt = [-10 -6]-60;
yt = [-10 -16]+45;
str = {'Rolling circle','(Wheel)'};
```

```
text(xt,yt,str)
xt = [-10 -6]+35;
yt = [-10 -16]+20;
str = {'Rolling circle','(Pinion)'};
text(xt,yt,str)
xt = [35 -15]+35;
yt = [100 -40]+20;
str = {'Wheel','Pinion'};
text(xt,yt,str)
% Gear 1
for i=0:Z-1
phi=i*2*pi/Z;
xc=x1*cos(phi)-(y1)*sin(phi);
yc=x1*sin(phi)+(y1)*cos(phi);
plot(xc,yc,'Color',[0 0 0.5],'LineWidth',2)
end
for i=0:Z-1
phi=i*2*pi/Z;
xc=-x1*cos(phi)-(y1)*sin(phi);
yc=-x1*sin(phi)+(y1)*cos(phi);
plot(xc,yc,'Color',[0 0 0.5],'LineWidth',2)
if i==Z-1
    xh2=[xc(end) yc(end)];
end
end
dseg2=[x1(end) xh2(1);y1(end) xh2(2)];

for i=0:Z-1
phi=i*2*pi/Z;
xcd=dseg2(1,:)*cos(phi)-dseg2(2,:)*sin(phi);
ycd=dseg2(1,:)*sin(phi)+dseg2(2,:)*cos(phi);
% plot(xcd,ycd,'Color',[0 0 0.5],'LineWidth',2)
end
title('GA,Open contact path,overlap=1.18')

% set(gca,'visible','off')
f2 = figure;
hold on
plot(x1(1:d0),R(1:d0),'Color',[0 0 0.5],'LineWidth',2)
function h = circle(x,y,r)
hold on
th = 0:pi/50:2*pi;
xunit = r * cos(th) + x;
yunit = r * sin(th) + y;
h = plot(xunit, yunit,'-.','Color','k','LineWidth',1);
% hold off
end
```

## Fmincon Algorithm

### StartingScript.m

```
clear all;clc;close all;
```

```
Z=19;Z2=55;m=2.5;          % number of teeth % module
%
n=6;
tooth=2;
x0 = LeastmS(Z,m,n,tooth);
lb=x0-4;
ub=x0+4;

[x,fval,exitflag,output,lambda,grad,hessian] =
fminconcode(x0,lb,ub);
```

nonlinear_constraints.m

```
function [C,Ceq] = nonlinear_constraints(x)
Z=19;Z2=55;m=2.5;          % number of teeth % module
%
div=1e3;%
Rtar=-(1.e-18)*zeros(1,div); % Target
%
do=Z*m;ro=do/2;            % [pcd] Pitch Circle Diamete
do2=Z2*m;ro2=do2/2;        % [pcd] Pitch Circle Diamete
dk=(Z+2)*m;rk=dk/2;        % [od] Outside diameter Cost pp 48
dk2=(Z2+2)*m;rk2=dk2/2;    % [od] Outside diameter Cost pp 48

a0=deg2rad(20);        % pressure angle
dg=do*cos(a0);         % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg=dg/2;               % [rbase]%

dg2=do2*cos(a0);       % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg2=dg2/2;             % [rbase]%
%
n=(numel(x)+2)/2;
CP=[x(1:n)
sqrt((rk)^2-x(1)^2)    x(n+1:end)     sqrt((rg)^2-x(n)^2)];%
%
% n : Number of CP , proekupsan apo sxesh n+k+1-numel(T)  =  0
% n=numel(CP)/2;
% k : deg of B-Spline , elaxisto 3
k=4;
% T : knot vector , numel(T)=n+k+1 , pollaplothta : 1 h parapanw
analoga
T=[zeros(1,k+1) 1:1:n-k-1 (n-k)*ones(1,k+1)];
% Basis Functions of B-Splines
t=T(1):(T(end)-T(1))/(div-1):T(end);
%
N=zeros(k+n,k+1,numel(t));
for it=1:numel(t)
    for i=1:numel(T)-1
       if t(it)>=T(i) && t(it) < T(i+1)
         N(i,1,it)=1;
    else
     N(i,1,it)=0;
    end
  end
  a=1;
  for j=2:k+1
    for i=1:numel(T)-1-a
      if  (N(i,j-1,it)==0 && N(i+1,j-1,it)==0)  % zero kai oi
duo oroi
        N(i,j,it)=0;
        if i<=n
          dN(i,it)=0;
        end
      elseif N(i,j-1,it)==0                  % zero 1os oros
        N(i,j,it)=((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-
1,it);
        if i<=n
          dN(i,it)=-  (k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
        end
      elseif N(i+1,j-1,it)==0                % zero 2os oros
        N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-1,it);
        if i<=n
          dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it);
        end
      else
        N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-
1,it)+((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
        if i<=n
          dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it) -
(k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
        end
      end
      if (i==n && j==k+1 && t(it)==T(end))
         N(:,:,it)=round(N(:,:,it-1),0);
         dN(:,it)=round(dN(:,it-1),0);
      end
    end
    a=a+1;
  end
% Nk(:,it)=N(1:n,k+1,it); % basis functions
end
for it=1:numel(t)
    for i=1:n
      a10= 1/(T(i+k)-  T(i));
      a11=-1/(T(i+k+1)-T(i+1));
      a20=a10/(T(i+k-1)-T(i));
      a21=(a11-a10)/(T(i+k)-T(i+1));
      a22=-a11/(T(i+k+1)-T(i+2));

      if     N(i,end-2,it)==0 && N(i+1,end-2,it)~=0 &&
N(i+2,end-2,it)~=0
```

```
          dN2(i,it)=(k-1)*k*(                  a21*N(i+1,end-
2,it)+a22*N(i+2,end-2,it));

          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+
a22*N(i+2,end-2,it));

          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)~=0 &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+a21*N(i+1,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(
a22*N(i+2,end-2,it));

          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)~=0     &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(                 +a21*N(i+1,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)==0     &&
N(i+2,end-2,it)==0
          dN2(i,it)=0;
          else
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+a21*N(i+1,end-
2,it)+a22*N(i+2,end-2,it));
          end
       end

end
dN2(:,it)=flip(dN2(:,1));

for it=1:numel(t)
x1(it) =N(1:n,k+1,it)'*CP(1,:)';
y1(it) =N(1:n,k+1,it)'*CP(2,:)';
dx1(it)=dN(:,it)'*CP(1,:)';
dy1(it)=dN(:,it)'*CP(2,:)';
d2x1(it)=dN2(:,it)'*CP(1,:)';
d2y1(it)=dN2(:,it)'*CP(2,:)';
end
dydx1=dy1./dx1; % 1st Derivative of B-Splines

f=dy1;  % = dy/dt
```

```
g=dx1;  % = dx/dt

d2ydx1=((d2y1.*g-d2x1.*f)./(g.^2))./g; % 2nd Derivative of B-
Splines

rgG=abs(x1+y1.*dydx1)./(sqrt(1+dydx1.^2)); %1.50 dr V.Spita
rG=sqrt(x1.^2+y1.^2);
% Contact Line Coordinates
term1=((rG/ro).^2-(rgG/ro).^2);
term2=(1-(rgG/ro).^2);
xp=rgG.*(sqrt(term1)-sqrt(term2)); %1.48 dr V.Spita / Contact Line
term3=((ro./rgG).^2-1);
yp=sqrt(term3).*xp;                %1.49 dr V.Spita / Contact Line
%
for i=1:numel(xp)
A=[x1(i)       y1(i)
   y1(i)      -x1(i)];  % exw klasei ta ro apo ton tupo 30 & 31 pp
41
rot(:,i)=inv(A)*[xp(i);yp(i)+ro];
end
thita=atan(rot(2,:)./rot(1,:));
K=thita*ro;
% rack
xr=xp-K;
yr=yp;
v=rot(2,:).^2+rot(1,:).^2; % sin^2 + cos^2 = 1
% dydx=gradient(yr)./ gradient(xr); % 1st Derivative of B-Splines
interpolation
% Metatopish Kanona || Allos tropos upologismou tou K ||
confirmation
% Ks=-(yr.*dydx+xr);                         % Cost pp 39 (22)
t0=K./ro;                                 % Cost pp 39 (19)
% (xp,yp) coordinates ths Troxias Epafwm sta (x1_exw,y1_exw)
Sunergazomenos troxos coordinates
t2=-t0*ro/ro2;
x1_exw=xp.*cos(t2)-(yp-ro2).*sin(t2);               % Cost pp 40
(27)
y1_exw=xp.*sin(t2)+(yp-ro2).*cos(t2)+ro2;           % Cost pp 40
(28)
% [S1,S2]=bsplines(x1_exw,y1_exw);
S1=gradient(y1_exw)./gradient(x1_exw);
sa=(S1(2)-S1(3))/(x1_exw(2)-x1_exw(3));
sb=S1(2)-sa*x1_exw(2);
S1(1)=sa*x1_exw(1)+sb;
sa=(S1(end-2)-S1(end-1))/(x1_exw(end-2)-x1_exw(end-1));
sb=S1(end-1)-sa*x1_exw(end-1);
S1(end)=sa*x1_exw(end)+sb;
S2=gradient(S1)     ./gradient(x1_exw);
sa=(S2(2)-S2(3))/(x1_exw(2)-x1_exw(3));
sb=S2(2)-sa*x1_exw(2);
S2(1)=sa*x1_exw(1)+sb;
```

```
sa=(S2(end-2)-S2(end-1))/(x1_exw(end-2)-x1_exw(end-1));
sb=S2(end-1)-sa*x1_exw(end-1);
S2(end)=sa*x1_exw(end)+sb;
%
R1=d2ydx1./((1+dydx1.^2).^(3/2)); % tupos (54) kostakis
R2=S2./((1+S1.^2).^(3/2));% tupos (54) kostakis
% if abs(R2(end)-R2(end-1))>10*abs(R2(end-1)-R2(end-2)) %<---
diorthwnei teleutaio shmeio
%     R2(end)=R2(end-1);
% end
R=R1-R2;
dR=gradient(R)./gradient(x1);
sa=(dR(2)-dR(3))/(x1(2)-x1(3));
sb=dR(2)-sa*x1(2);
dR(1)=sa*x1(1)+sb;
sa=(dR(end-2)-dR(end-1))/(x1(end-2)-x1(end-1));
sb=dR(end-1)-sa*x1(end-1);
dR(end)=sa*x1(end)+sb;
AB=0;
for i=1:div-1
AB=AB+sqrt((xp(i+1)-xp(i))^2+(yp(i+1)-yp(i))^2);
end
overlap=AB/(pi*dg/Z);
% if imag(overlap)~=0
%     overlap=0;
% end
flag=0;
for i=1:numel(y1)-1
    if y1(i+1)>y1(i)
    flag=flag+1;
    end
end
flag1=0;
for i=1:numel(x1)-1
    if x1(i+1)<x1(i)
    flag1=flag1+1;
    end
end
flag2=0;
for i=1:numel(y1_exw)-1
    if y1_exw(i+1)>y1_exw(i)
    flag2=flag2+1;
    end
end
flag3=0;
for i=1:numel(x1_exw)-1
    if x1_exw(i+1)<x1_exw(i)
    flag3=flag3+1;
    end
end
C=[flag
flag1
flag2
flag3
... kapa-15
... abs(mean(R))-0.05
...% -(overlap-1.1)
];
Ceq =[];
```

### LeastmS.m

```
function b = LeastmS(Z,m,n,tooth)
% Z= dontia pinion
% m= module
% n= numel of Control Points
% tooth = # twn dontiwn aristera pou tha ginoun oi upologismoi
%% LEAST MIN SQUARES
% eisagontai ta theorytika (x,y) tou rack kai pianoume thn katatomh
me
% elaxista tetragwnaZ
% Kost pp 44 (23) means Kostopoulos selida 44 sxesh 23
%% Input Data SIN - (x,y) coordinates ths Troxias Epafwn  P : Data
to interpollate P(:,1)=x;P(:,2)=y;
[xf1,yf1]=Involute_Coordinates(Z,m);
%
do=Z*m;ro=do/2;       % [pcd] Pitch Circle Diameter
phi=tooth*2*pi/Z;
x9=xf1*cos(phi)-(yf1+ro)*sin(phi);
y9=xf1*sin(phi)+(yf1+ro)*cos(phi);
% x9=x1d*cos(phi)-(y1d)*sin(phi);
% y9=x1d*sin(phi)+(y1d)*cos(phi);
P(:,1)=x9';P(:,2)=y9';
% P(:,1)=x1_exw';P(:,2)=y1_exw';
div=size(P);
%% Input data 1 - Gear parameters
% a0=deg2rad(20);       % pressure angle
%
% dk=(Z+2)*m;rk=dk/2;   % [od] Outside diameter Kost pp 48
% dg=do*cos(a0);        % [bcd] Base Circle Diameter Kost pp,
rg=ro*cos(ao)
% rg=dg/2;              % [rbase]
%% Input of B-Spline parameters
% n : Number of CP , proekupsan apo sxesh n+k+1-numel(T)  =  0
% k : deg of B-Spline , elaxisto 3
k=4;
% T : knot vector , numel(T)=n+k+1 , pollaplothta : 1 h parapanw
analoga
T=[zeros(1,k+1) 1:1:n-k-1 (n-k)*ones(1,k+1)];
% Basis Functions of B-Splines
t=T(1):(T(end)-T(1))/(div(1)-1):T(end);
%
N=zeros(k+n,k+1,numel(t));
```

```
for it=1:numel(t)
    for i=1:numel(T)-1
       if t(it)>=T(i) && t(it) < T(i+1)
         N(i,1,it)=1;
       else
         N(i,1,it)=0;
       end
    end
    a=1;
    for j=2:k+1
       for i=1:numel(T)-1-a
          if  (N(i,j-1,it)==0 && N(i+1,j-1,it)==0)  % zero kai oi
duo oroi
              N(i,j,it)=0;
              if i<=n
                 dN(i,it)=0;
              end
          elseif N(i,j-1,it)==0                   % zero 1os oros
              N(i,j,it)=((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-
1,it);
              if i<=n
                 dN(i,it)=-  (k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          elseif N(i+1,j-1,it)==0                 % zero 2os oros
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it);
              end
          else
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-
1,it)+((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it) -
(k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          end
          if (i==n && j==k+1 && t(it)==T(end))
                N(:,:,it)=round(N(:,:,it-1),0);
                dN(:,it)=round(dN(:,it-1),0);
          end
       end
       a=a+1;
    end
% Nk(:,it)=N(1:n,k+1,it); % basis functions
end
C0=zeros(numel(t),2);

%% Least mean Squares
B_sp_k=zeros(n,numel(t));
for i=1:n
B_sp_k(i,:)=N(i,k+1,:);
```

```
end
CP(1,:)=inv(B_sp_k*transpose(B_sp_k))*B_sp_k*P(:,1);  % Euresh
Control Points
CP(2,:)=inv(B_sp_k*transpose(B_sp_k))*B_sp_k*P(:,2);
%%
CP(1,1)=P(1,1); CP(1,end)=P(end,1);    % To prwto kai to teleutaio
CP
CP(2,1)=P(1,2); CP(2,end)=P(end,2);    % ta orizw me vash ta shmeia
paremvolhs
b=[CP(1,:),CP(2,2:end-1)]; % prin thn allagh -tmhmata
end
```

Involute_Coordinates.m

```
function [xf1,yf1] = Involute_Coordinates(Z,m)
a0=deg2rad(20);       % pressure angle
div=1e4;              % #of nodes
%% Gear parameters
dk=(Z+2)*m;rk=dk/2;   %[od] Outside diameter Cost pp 48
% df=(Z-2.5)*m;         %[rd] Root diameter Cost pp sel 48
do=Z*m;ro=do/2;       %[pcd] Pitch Circle Diameter
dg=do*cos(a0);        %[bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg=dg/2;              %[rbase]
So=(0.475*pi)*do/Z;   %(0.475-0.5) Tooth thickness at pcd  Cost pp
46 (7)
inva0 = tan(a0) - a0; %Involute function
%% Involute generation
for i = 1:div
r(i) = rk-((rk-rg)/(div-1))*(i-1); % radius from rk to rg
a(i) = acos(dg / (2 * r(i)));         % angle a of each involute
point
inva(i) = tan(a(i)) - a(i);           % involute function fi for
each involute point
S(i) = So * (r(i) / ro) + 2 * (inva0 - inva(i)) * r(i); % Tooth
thickness at each point Cost pp 46 (10)
ak=acos(rg/rk);                    % Cost pp 46
xp(i)= r(i) * sin(inva(i));        % change from polar coordinates
yp(i)= r(i) * cos(inva(i));        % to cartesian coordinates
end
%%
% Sk = S(1);                      % paxos odontos kefalhs
Sg=S(div);                      % paxos at base circle
%% Rotation of involute to its symmetrical initial position
for i=0:Z-1
el=(2*pi*rg-Z*Sg)/Z;
phig = i*((Sg+el)/rg);
rot = [ cos(phig) -sin(phig);
        sin(phig) cos(phig)];
inv_rot = rot * [xp; yp];
xp_rot = inv_rot(1,:);
yp_rot = inv_rot(2,:);
```

```
%%
phi=deg2rad(360*Sg/(2*pi*rg))/2;
xp_rot1= cos(phi)*xp_rot -sin(phi)*yp_rot;
yp_rot1= sin(phi)*xp_rot+ cos(phi)*yp_rot;
if i==0
    xf1=-xp_rot1;
    yf1=yp_rot1-ro;
end
end
```

fminconcode.m

```
function [x,fval,exitflag,output,lambda,grad,hessian] =
fminconcode(x0,lb,ub)
%% This is an auto generated MATLAB file from Optimization Tool.

%% Start with the default options
options = optimoptions('fmincon');
%% Modify options setting
options = optimoptions(options,'Display', 'off');
[x,fval,exitflag,output,lambda,grad,hessian] = ...
fmincon(@fitness,x0,[],[],[],[],lb,ub,@nonlinear_constraints,option
s);
```

fitness.m

```
function Y = fitness(x)
%
Z=19;Z2=55;m=2.5;         % number of teeth % module
%
div=1e3;%
Rtar=-(1.e-18)*zeros(1,div); % Target
%
do=Z*m;ro=do/2;           % [pcd] Pitch Circle Diamete
do2=Z2*m;ro2=do2/2;       % [pcd] Pitch Circle Diamete
dk=(Z+2)*m;rk=dk/2;       % [od] Outside diameter Cost pp 48
dk2=(Z2+2)*m;rk2=dk2/2;   % [od] Outside diameter Cost pp 48

a0=deg2rad(20);       % pressure angle
dg=do*cos(a0);        % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg=dg/2;              % [rbase]%

dg2=do2*cos(a0);      % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg2=dg2/2;            % [rbase]%
%
n=(numel(x)+2)/2;
CP=[x(1:n)
sqrt((rk)^2-x(1)^2)    x(n+1:end)     sqrt((rg)^2-x(n)^2)];%
%
% n : Number of CP , proekupsan apo sxesh n+k+1-numel(T)   =  0
% n=numel(CP)/2;
% k : deg of B-Spline , elaxisto 3
k=4;
% T : knot vector , numel(T)=n+k+1 , pollaplothta : 1 h parapanw
analoga
T=[zeros(1,k+1) 1:1:n-k-1 (n-k)*ones(1,k+1)];
% Basis Functions of B-Splines
t=T(1):(T(end)-T(1))/(div-1):T(end);
%
N=zeros(k+n,k+1,numel(t));
for it=1:numel(t)
    for i=1:numel(T)-1
       if t(it)>=T(i) && t(it) < T(i+1)
         N(i,1,it)=1;
       else
         N(i,1,it)=0;
       end
    end
    a=1;
    for j=2:k+1
       for i=1:numel(T)-1-a
          if   (N(i,j-1,it)==0 && N(i+1,j-1,it)==0)  % zero kai oi
duo oroi
              N(i,j,it)=0;
              if i<=n
                 dN(i,it)=0;
              end
          elseif N(i,j-1,it)==0                  % zero 1os oros
              N(i,j,it)=((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-
1,it);
              if i<=n
                 dN(i,it)=-  (k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          elseif N(i+1,j-1,it)==0                % zero 2os oros
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it);
              end
          else
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-
1,it)+((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it) -
(k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          end
          if (i==n && j==k+1 && t(it)==T(end))
                N(:,:,it)=round(N(:,:,it-1),0);
                dN(:,it)=round(dN(:,it-1),0);
          end
       end
       a=a+1;
    end
% Nk(:,it)=N(1:n,k+1,it); % basis functions
end
for it=1:numel(t)
       for i=1:n
          a10= 1/(T(i+k)-  T(i));
          a11=-1/(T(i+k+1)-T(i+1));
          a20=a10/(T(i+k-1)-T(i));
          a21=(a11-a10)/(T(i+k)-T(i+1));
          a22=-a11/(T(i+k+1)-T(i+2));

          if     N(i,end-2,it)==0 && N(i+1,end-2,it)~=0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(                  a21*N(i+1,end-
2,it)+a22*N(i+2,end-2,it));

          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+
a22*N(i+2,end-2,it));

          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)~=0 &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+a21*N(i+1,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(
a22*N(i+2,end-2,it));

          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)~=0     &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(                 +a21*N(i+1,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)==0     &&
N(i+2,end-2,it)==0
          dN2(i,it)=0;
          else
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+a21*N(i+1,end-
2,it)+a22*N(i+2,end-2,it));
          end
       end

end
```

```
dN2(:,it)=flip(dN2(:,1));

for it=1:numel(t)
x1(it) =N(1:n,k+1,it)'*CP(1,:)';
y1(it) =N(1:n,k+1,it)'*CP(2,:)';
dx1(it)=dN(:,it)'*CP(1,:)';
dy1(it)=dN(:,it)'*CP(2,:)';
d2x1(it)=dN2(:,it)'*CP(1,:)';
d2y1(it)=dN2(:,it)'*CP(2,:)';
end
dydx1=dy1./dx1; % 1st Derivative of B-Splines

f=dy1;  % = dy/dt
g=dx1;  % = dx/dt

d2ydx1=((d2y1.*g-d2x1.*f)./(g.^2))./g; % 2nd Derivative of B-
Splines

rgG=abs(x1+y1.*dydx1)./(sqrt(1+dydx1.^2)); %1.50 dr V.Spita
rG=sqrt(x1.^2+y1.^2);
% Contact Line Coordinates
term1=((rG/ro).^2-(rgG/ro).^2);
term2=(1-(rgG/ro).^2);
xp=rgG.*(sqrt(term1)-sqrt(term2)); %1.48 dr V.Spita / Contact Line
term3=((ro./rgG).^2-1);
yp=sqrt(term3).*xp;               %1.49 dr V.Spita / Contact Line
%
for i=1:numel(xp)
A=[x1(i)       y1(i)
   y1(i)      -x1(i)];  % exw klasei ta ro apo ton tupo 30 & 31 pp
41
rot(:,i)=inv(A)*[xp(i);yp(i)+ro];
end
thita=atan(rot(2,:)./rot(1,:));
K=thita*ro;
% rack
xr=xp-K;
yr=yp;
v=rot(2,:).^2+rot(1,:).^2; % sin^2 + cos^2 = 1
% dydx=gradient(yr)./ gradient(xr); % 1st Derivative of B-Splines
interpolation
% Metatopish Kanona || Allos tropos upologismou tou K ||
confirmation
% Ks=-(yr.*dydx+xr);                    % Cost pp 39 (22)
t0=K./ro;                            % Cost pp 39 (19)
% (xp,yp) coordinates ths Troxias Epafwm sta (x1_exw,y1_exw)
Sunergazomenos troxos coordinates
t2=-t0*ro/ro2;
x1_exw=xp.*cos(t2)-(yp-ro2).*sin(t2);              % Cost pp 40
(27)
```

```
y1_exw=xp.*sin(t2)+(yp-ro2).*cos(t2)+ro2;          % Cost pp 40
(28)
% [S1,S2]=bsplines(x1_exw,y1_exw);
S1=gradient(y1_exw)./gradient(x1_exw);
sa=(S1(2)-S1(3))/(x1_exw(2)-x1_exw(3));
sb=S1(2)-sa*x1_exw(2);
S1(1)=sa*x1_exw(1)+sb;
sa=(S1(end-2)-S1(end-1))/(x1_exw(end-2)-x1_exw(end-1));
sb=S1(end-1)-sa*x1_exw(end-1);
S1(end)=sa*x1_exw(end)+sb;
S2=gradient(S1)    ./gradient(x1_exw);
sa=(S2(2)-S2(3))/(x1_exw(2)-x1_exw(3));
sb=S2(2)-sa*x1_exw(2);
S2(1)=sa*x1_exw(1)+sb;
sa=(S2(end-2)-S2(end-1))/(x1_exw(end-2)-x1_exw(end-1));
sb=S2(end-1)-sa*x1_exw(end-1);
S2(end)=sa*x1_exw(end)+sb;
%
R1=d2ydx1./((1+dydx1.^2).^(3/2)); % tupos (54) kostakis
R2=S2./((1+S1.^2).^(3/2));% tupos (54) kostakis
% if abs(R2(end)-R2(end-1))>10*abs(R2(end-1)-R2(end-2)) %<---
diorthwnei teleutaio shmeio
%     R2(end)=R2(end-1);
% end
R=R1-R2;
dR=gradient(R)./gradient(x1);
sa=(dR(2)-dR(3))/(x1(2)-x1(3));
sb=dR(2)-sa*x1(2);
dR(1)=sa*x1(1)+sb;
sa=(dR(end-2)-dR(end-1))/(x1(end-2)-x1(end-1));
sb=dR(end-1)-sa*x1(end-1);
dR(end)=sa*x1(end)+sb;
AB=0;
for i=1:div-1
AB=AB+sqrt((xp(i+1)-xp(i))^2+(yp(i+1)-yp(i))^2);
end
overlap=AB/(pi*dg/Z);
fun=0.5*(R-Rtar).^2;
fun1=0.5*(dR-Rtar).^2;
% Y=15*trapz(x1,fun)+0.0.15*abs(mean(R))+0.10*std(R);
% Y=+;0.5*trapz(x1,fun)
Y=1*trapz(x1,fun1)+1*mean(abs(R))+1*(std(R));%+abs(CP(1,1)+15);%;%a
bs(mean(R))+0.50.05*max(abs(R))+2*+
end
```

### EvaluationScript.m

```
clear all;clc;close all;
format short
tic
```

```
x=[ -15.7080  -15.0546  -13.9610  -13.1249  -12.2167  -11.7355
20.8106   20.3594   19.9236   19.3419];
Z=19;Z2=55;m=2.5;          % number of teeth % module
n=6;
tooth=2; % Posa dodia aristera apo th symetria 8a einai to donti

%
div=1e3;%
Rtar=-(1.e-18)*zeros(1,div); % Target
%
do=Z*m;ro=do/2;            % [pcd] Pitch Circle Diamete
do2=Z2*m;ro2=do2/2;        % [pcd] Pitch Circle Diamete
dk=(Z+2)*m;rk=dk/2;        % [od] Outside diameter Cost pp 48
dk2=(Z2+2)*m;rk2=dk2/2;    % [od] Outside diameter Cost pp 48

a0=deg2rad(20);        % pressure angle
dg=do*cos(a0);         % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg=dg/2;               % [rbase]%

dg2=do2*cos(a0);       % [bcd] Base Circle Diameter Cost pp,
rg=ro*cos(ao)
rg2=dg2/2;             % [rbase]%
%
CP=[x(1:n)
sqrt((rk)^2-x(1)^2)    x(n+1:end)     sqrt((rg)^2-x(n)^2)];%
phi=-tooth*2*pi/Z;
CP0(1,:)=CP(1,:)*cos(phi)-CP(2,:)*sin(phi);
CP0(2,:)=CP(1,:)*sin(phi)+CP(2,:)*cos(phi);
CK=CP;
clearvars CP
CP=CP0;
% n : Number of CP , proekupsan apo sxesh n+k+1-numel(T)  =  0
n=numel(CP)/2;
% k : deg of B-Spline , elaxisto 3
k=4;
% T : knot vector , numel(T)=n+k+1 , pollaplothta : 1 h parapanw
analoga
T=[zeros(1,k+1) 1:1:n-k-1 (n-k)*ones(1,k+1)];
% Basis Functions of B-Splines
t=T(1):(T(end)-T(1))/(div-1):T(end);
%
N=zeros(k+n,k+1,numel(t));
for it=1:numel(t)
    for i=1:numel(T)-1
       if t(it)>=T(i) && t(it) < T(i+1)
         N(i,1,it)=1;
       else
         N(i,1,it)=0;
       end
    end
    a=1;
    for j=2:k+1
       for i=1:numel(T)-1-a
          if  (N(i,j-1,it)==0 && N(i+1,j-1,it)==0)  % zero kai oi
duo oroi
              N(i,j,it)=0;
              if i<=n
                 dN(i,it)=0;
              end
          elseif N(i,j-1,it)==0                  % zero 1os oros
              N(i,j,it)=((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-
1,it);
              if i<=n
                 dN(i,it)=- (k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          elseif N(i+1,j-1,it)==0                % zero 2os oros
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it);
              end
          else
              N(i,j,it)=((t(it)-T(i))/(T(i+j-1)-T(i)))*N(i,j-
1,it)+((T(i+j)-t(it))/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              if i<=n
                 dN(i,it)=(k/(T(i+j-1)-T(i)))*N(i,j-1,it) -
(k/(T(i+j)-T(i+1)))*N(i+1,j-1,it);
              end
          end
          if (i==n && j==k+1 && t(it)==T(end))
                N(:,:,it)=round(N(:,:,it-1),0);
                dN(:,it)=round(dN(:,it-1),0);
          end
       end
       a=a+1;
    end
% Nk(:,it)=N(1:n,k+1,it); % basis functions
end
for it=1:numel(t)
       for i=1:n
          a10= 1/(T(i+k)-  T(i));
          a11=-1/(T(i+k+1)-T(i+1));
          a20=a10/(T(i+k-1)-T(i));
          a21=(a11-a10)/(T(i+k)-T(i+1));
          a22=-a11/(T(i+k+1)-T(i+2));

          if     N(i,end-2,it)==0 && N(i+1,end-2,it)~=0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(                  a21*N(i+1,end-
2,it)+a22*N(i+2,end-2,it));
```

```
          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+
a22*N(i+2,end-2,it));

          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)~=0 &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+a21*N(i+1,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)~=0
          dN2(i,it)=(k-1)*k*(
a22*N(i+2,end-2,it));

          elseif N(i,end-2,it)~=0 && N(i+1,end-2,it)==0 &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)~=0     &&
N(i+2,end-2,it)==0
          dN2(i,it)=(k-1)*k*(                 +a21*N(i+1,end-2,it))
;

          elseif N(i,end-2,it)==0 && N(i+1,end-2,it)==0     &&
N(i+2,end-2,it)==0
          dN2(i,it)=0;
          else
          dN2(i,it)=(k-1)*k*(a20*N(i,end-2,it)+a21*N(i+1,end-
2,it)+a22*N(i+2,end-2,it));
          end
       end

end
dN2(:,it)=flip(dN2(:,1));

for it=1:numel(t)
x1(it) =N(1:n,k+1,it)'*CP(1,:)';
y1(it) =N(1:n,k+1,it)'*CP(2,:)';
dx1(it)=dN(:,it)'*CP(1,:)';
dy1(it)=dN(:,it)'*CP(2,:)';
d2x1(it)=dN2(:,it)'*CP(1,:)';
d2y1(it)=dN2(:,it)'*CP(2,:)';
end
dydx1=dy1./dx1; % 1st Derivative of B-Splines

f=dy1;  % = dy/dt
g=dx1;  % = dx/dt
```

```
d2ydx1=((d2y1.*g-d2x1.*f)./(g.^2))./g; % 2nd Derivative of B-
Splines

rgG=abs(x1+y1.*dydx1)./(sqrt(1+dydx1.^2)); %1.50 dr V.Spita
rG=sqrt(x1.^2+y1.^2);
% Contact Line Coordinates
term1=((rG/ro).^2-(rgG/ro).^2);
term2=(1-(rgG/ro).^2);
xp=rgG.*(sqrt(term1)-sqrt(term2)); %1.48 dr V.Spita / Contact Line
term3=((ro./rgG).^2-1);
yp=sqrt(term3).*xp;                %1.49 dr V.Spita / Contact Line
%
for i=1:numel(xp)
A=[x1(i)       y1(i)
   y1(i)      -x1(i)];  % exw vgalei ta ro apo ton tupo 30 & 31 pp
41
rot(:,i)=inv(A)*[xp(i);yp(i)+ro];
end
thita=atan(rot(2,:)./rot(1,:));
K=thita*ro;
% rack
xr=xp-K;
yr=yp;
v=rot(2,:).^2+rot(1,:).^2; % sin^2 + cos^2 = 1
% Metatopish Kanona || Allos tropos upologismou tou K ||
confirmation
% dydx=gradient(yr)./ gradient(xr); % 1st Derivative of B-Splines
interpolation
% Ks=-(yr.*dydx+xr);                    % Cost pp 39 (22)
t0=K./ro;                            % Cost pp 39 (19)
% (xp,yp) coordinates ths Troxias Epafwm sta (x1_exw,y1_exw)
Sunergazomenos troxos coordinates
t2=-t0*ro/ro2;
x1_exw=xp.*cos(t2)-(yp-ro2).*sin(t2);              % Cost pp 40
(27)
y1_exw=xp.*sin(t2)+(yp-ro2).*cos(t2)+ro2;          % Cost pp 40
(28)
%
S1=gradient(y1_exw)./gradient(x1_exw);
sa=(S1(2)-S1(3))/(x1_exw(2)-x1_exw(3));
sb=S1(2)-sa*x1_exw(2);
S1(1)=sa*x1_exw(1)+sb;
sa=(S1(end-2)-S1(end-1))/(x1_exw(end-2)-x1_exw(end-1));
sb=S1(end-1)-sa*x1_exw(end-1);
S1(end)=sa*x1_exw(end)+sb;
S2=gradient(S1)    ./gradient(x1_exw);
sa=(S2(2)-S2(3))/(x1_exw(2)-x1_exw(3));
sb=S2(2)-sa*x1_exw(2);
S2(1)=sa*x1_exw(1)+sb;
sa=(S2(end-2)-S2(end-1))/(x1_exw(end-2)-x1_exw(end-1));
sb=S2(end-1)-sa*x1_exw(end-1);
```

```
S2(end)=sa*x1_exw(end)+sb;
%
R1=d2ydx1./((1+dydx1.^2).^(3/2)); % tupos (54) kostakis
R2=S2./((1+S1.^2).^(3/2));% tupos (54) kostakis
% if abs(R2(end)-R2(end-1))>10*abs(R2(end-1)-R2(end-2)) %<---
diorthwnei teleutaio shmeio
%     R2(end)=R2(end-1);
% end
R=R1-R2;
dR=gradient(R)./gradient(x1);
sa=(dR(2)-dR(3))/(x1(2)-x1(3));
sb=dR(2)-sa*x1(2);
dR(1)=sa*x1(1)+sb;
sa=(dR(end-2)-dR(end-1))/(x1(end-2)-x1(end-1));
sb=dR(end-1)-sa*x1(end-1);
dR(end)=sa*x1(end)+sb;
fun=0.5*(R-Rtar).^2;
fun1=0.5*(dR-Rtar).^2;
fun=0.5*(R-Rtar).^2;
Y0=trapz(x1,fun);
kapa=abs(max(d2ydx1));
%
AB=0;
for i=1:div-1
AB=AB+sqrt((xp(i+1)-xp(i))^2+(yp(i+1)-yp(i))^2);
end
overlap=AB/(pi*dg/Z)
flag=0;
for i=1:numel(y1)-1
    if y1(i+1)>y1(i)
    flag=flag+1;
    end
end
flag1=0;
for i=1:numel(x1)-1
    if x1(i+1)<x1(i)
    flag1=flag1+1;
    end
end
flag2=0;
for i=1:numel(y1_exw)-1
    if y1_exw(i+1)>y1_exw(i)
    flag2=flag2+1;
    end
end
flag3=0;
for i=1:numel(x1_exw)-1
    if x1_exw(i+1)<x1_exw(i)
    flag3=flag3+1;
    end
end
```

```
C=[flag
   flag1 % ? mallon oxi
flag2
flag3
overlap];
load('R_sin.mat')
load('R_inv.mat')
hold on;axis equal;
% plot(x1,R,'Color',[0 0 0.5],'LineWidth',2)
% plot(xsin,Rsin,'Color',[0.9 0 0],'LineWidth',2)
% plot(xinv,Rinv,'LineWidth',2)
plot(0,0,'o')
%
% plot(xp,yp+ro,'Color','blue','LineWidth',2)
c=[0;ro2];
plot(xp,yp+ro)
% plot(x1,y1)
% plot(x1_exw,y1_exw+ro,'red')
for i=0:Z2-1
phi=i*2*pi/Z2;
Tr=[cos(phi) -sin(phi);
    sin(phi)  cos(phi)];
d=[x1_exw;y1_exw]-c;
xd=Tr*d+c;
plot(xd(1,:),xd(2,:)+ro,'Color',[0.9 0 0],'LineWidth',2)

end
for i=0:Z2-1
phi=i*2*pi/Z2;
Tr=[cos(phi) -sin(phi);
    sin(phi)  cos(phi)];
d=[-x1_exw;y1_exw]-c;
xd=Tr*d+c;
plot(xd(1,:),xd(2,:)+ro,'Color',[0.9 0 0],'LineWidth',2)
if i==1
    xh=xd(:,end);
end
end
% plot(x1(1),y1(1),'o')
% plot(-x1(1),y1(1),'o')

dseg1=[x1(1) -x1(1);
       y1(1)  y1(1)];
% plot(dseg1(1,:),dseg1(2,:),'Color',[0 0 0.5],'LineWidth',1)
plot(CK(1, :), CK(2, :), 'ro','LineWidth',0.5);
plot(CK(1, :), CK(2, :), 'b','LineWidth',0.5);

for i=0:Z-1
phi=i*2*pi/Z;
xfc=dseg1(1,:)*cos(phi)-dseg1(2,:)*sin(phi);
yfc=dseg1(1,:)*sin(phi)+dseg1(2,:)*cos(phi);
```

```
if i==2 || i==0
plot(xfc,yfc,'Color',[0 0 0.5],'LineWidth',2)
end
end
dseg=[x1_exw(end)    xh(1);
      y1_exw(end)+ro xh(2)+ro];
% plot(dseg(1,:),dseg(2,:),'Color',[0.9 0 0],'LineWidth',1)
for i=0:Z2-1
phi=i*2*pi/Z2;
Tr=[cos(phi) -sin(phi);
    sin(phi)  cos(phi)];
d=[dseg(1,:);dseg(2,:)-ro]-c;
xdseg=Tr*d+c;
plot(xdseg(1,:),xdseg(2,:)+ro,'Color',[0.9 0 0],'LineWidth',2)
end

circle(0,0,rk);
circle(0,0,rg);
% circle(0,ro+ro2,rg2);
% circle(0,ro+ro2,rk2);

circle(0,0,ro);
circle(0,ro+ro2,ro2);

circle2(0,0,rk/5);
% circle2(0,ro+ro2,rk2/5);
plot([0  0],[0 ro+ro2+80],'-.','Color','k','LineWidth',1)
plot([0  ro+ro2],[ro ro],'-.','Color','k','LineWidth',1)
% title('ST-D,Open contact path')%,overlap=1.04')
% plot(0,0,'o','Color','k','LineWidth',1)
plot(0,ro+ro2,'o','Color','k','LineWidth',1)
plot(0,ro,'o','Color','k','LineWidth',1)
xt = [-10 -6]-60;
yt = [-10 -16]+45;
str = {'Rolling circle','(Wheel)'};
text(xt,yt,str)
xt = [-10 -6]+35;
yt = [-10 -16]+20;
str = {'Rolling circle','(Pinion)'};
text(xt,yt,str)
xt = [35 -15]+35;
yt = [100 -40]+20;
str = {'Wheel','Pinion'};
text(xt,yt,str)

% circle(0,ro+ro2,rg2+1.6);
for i=0:Z-1
phi=i*2*pi/Z;
xc=x1*cos(phi)-(y1)*sin(phi);
yc=x1*sin(phi)+(y1)*cos(phi);
% if i==2 || i==0
```

```
plot(xc,yc,'Color',[0 0 0.5],'LineWidth',2)
% end
end
for i=0:Z-1
phi=i*2*pi/Z;
xc=-x1*cos(phi)-(y1)*sin(phi);
yc=-x1*sin(phi)+(y1)*cos(phi);
% if i==2 || i==0
plot(xc,yc,'Color',[0 0 0.5],'LineWidth',2)
% end
if i==Z-1
    xh2=[xc(end) yc(end)];
end
end


dseg2=[x1(end) xh2(1);y1(end) xh2(2)];

for i=0:Z-1
phi=i*2*pi/Z;
xcd=dseg2(1,:)*cos(phi)-dseg2(2,:)*sin(phi);
ycd=dseg2(1,:)*sin(phi)+dseg2(2,:)*cos(phi);
% plot(xcd,ycd,'Color',[0 0 0.5],'LineWidth',2)
end

% circle(0,ro+ro2,ro2)
% for i=0:Z2-1
% phi=i*2*pi/Z2;
% x1c=x1_exw*cos(phi)-(y1_exw+ro2)*sin(phi);
% y1c=x1_exw*sin(phi)+(y1_exw+ro2)*cos(phi);
% plot(x1c,y1c+ro+ro2,'Color',[0.9 0 0],'LineWidth',2)
% end
% for i=0:Z2-1
% phi=i*2*pi/Z2;
% x1c=-x1_exw*cos(phi)-(y1_exw+ro2)*sin(phi);
% y1c=-x1_exw*sin(phi)+(y1_exw+ro2)*cos(phi);
% plot(x1c,y1c+ro+ro2,'Color',[0.9 0 0],'LineWidth',2)
% end
set(gca,'visible','off')

% hold off
% f2=figure;
% hold on;axis equal;
% plot(x1,R,'Color',[0 0 0.5],'LineWidth',2)
% plot(xsin,Rsin,'Color',[0.9 0 0],'LineWidth',2)
% plot(xinv,Rinv,'LineWidth',2)
% p1 = [0 0];                         % First Point
% p2 = [0 rk+5];                         % Second Point
% dp = p2-p1;                         % Difference
% figure
% quiver(p1(1),p1(2),dp(1),dp(2),0,LineSpec,'filled')
% grid
% axis([0  10    0  10])
% text(p1(1),p1(2), sprintf('(%.0f,%.0f)',p1))
% text(p2(1),p2(2), sprintf('(%.0f,%.0f)',p2))
% p1 = [0 0];                         % First Point
% p2 = [rk+5 0];                         % Second Point
% dp = p2-p1;                         % Difference
% % figure
% quiver(p1(1),p1(2),dp(1),dp(2),0,LineSpec,'filled')
% grid
% axis([0  10    0  10])
% text(p1(1),p1(2), sprintf('(%.0f,%.0f)',p1))
% text(p2(1),p2(2), sprintf('(%.0f,%.0f)',p2))
function h = circle(x,y,r)
hold on
th = 0:pi/50:2*pi;
xunit = r * cos(th) + x;
yunit = r * sin(th) + y;
h = plot(xunit, yunit,'-.','Color','k','LineWidth',1);
% hold off
end
function h = circle2(x,y,r)
hold on
th = 0:pi/50:2*pi;
xunit = r * cos(th) + x;
yunit = r * sin(th) + y;
h = plot(xunit, yunit,'Color','k','LineWidth',1);
% hold off
end
```