

Submitted: 2023-05-16 | Revised: 2023-06-03 | Accepted: 2023-06-07

Keywords: helical spring, spring design, spring stiffness, load-deflection curve, spring geometry

Krzysztof MICHALCZYK [0000-0002-1024-5947]*,
Mariusz WARZECHA [0000-0002-7417-1561]*, Robert BARAN [0000-0002-0711-230X]*

A NEW METHOD FOR GENERATING VIRTUAL MODELS OF NONLINEAR HELICAL SPRINGS BASED ON A RIGOROUS MATHEMATICAL MODEL

Abstract

This paper presents a new method for generating nonlinear helical spring geometries based on a rigorous mathematical formulation. The model was developed for two scenarios for modifying a spring with a stepped helix angle: for a fixed helix angle of the active coils and for a fixed overall height of the spring. It allows the development of compression spring geometries with non-linear load-deflection curves, while maintaining predetermined values of selected geometrical parameters, such as the number of passive and active coils and the total height or helix angle of the linear segment of the active coils. Based on the proposed models, Python scripts were developed that can be implemented in any CAD software offering scripting capabilities or equipped with Application Programming Interfaces. Examples of scripts that use the developed model to generate the geometry of selected springs are presented. FEM analyses of quasi-static compression tests carried out for these spring models showed that springs with a wide range of variation in static load-deflection curves, including progressive springs with a high degree of nonlinearity in characteristics, can be obtained using the proposed tools. The obtained load-deflection curves can be described with a high degree of accuracy by power function. The proposed method can find applications in both machine design and spring manufacturing.

1. INTRODUCTION

The requirements for compression coil springs used in industry are increasing, and a wide range of applications forces the search for modifications of their classic geometry. Spring geometry influences its static and dynamic properties. These properties can be analysed by analytical models, such as the classic study (Wittrick, 1966), or using computer methods (Ding & Selig, 2004; Taktak et al., 2008; Pöllänen & Martikka, 2010; Zhuo et al., 2022). The analytical and numerical methods can also be combined, as shown in (Gobbi & Mastinu,

* AGH University of Science and Technology, Faculty of Mechanical Engineering and Robotics, Department of Machine Design and Maintenance, Poland, kmichal@agh.edu.pl, mwarzech@agh.edu.pl, rbaran@agh.edu.pl

2001) for optimisation of composite material helical spring. The most accurate results are obtained using models that treat the spring as a spatially curved bar (Michalczyk, 2015). However, such an approach is usually implemented under the assumption that the spring is linear and has a homogeneous geometry along its entire length. One of the few papers that considers the effect of changing the geometry of the spring in the area of the end coils is the paper (Liu & Kim, 2009). In this work, the authors proposed a model in which the effect of the end coils on the spring axial stiffness is modelled using modified boundary conditions. Modelling springs with non-linear characteristics, i.e. with variable pitch and/or diameter, is particularly problematic. These springs have an advantage over linear springs due to their better performance in reducing the dynamic response at high frequencies. In the work (Zhao et al., 2023), the FE model based on Timoshenko beam theory was used to model the work of these springs and was compared with the model based on solid elements and experimental research.

Commercial CAD software is used in work on analyzing the properties of springs with relatively simple geometries (Gu et al., 2020; Rahul & Rameshkumar, 2021; Arshad et al., 2022), since it has limited capabilities for shaping the geometry of helical springs and usually allows only the control of selected parameters within an arbitrarily imposed spring scheme, most often in accordance with EN 13906-1:2013 standard. Other studies that used CAD software with such limitations include (Chandravanshi & Mukhopadhyay, 2017; Gzal et al., 2017; Nazir et al., 2020; Sahu et al., 2022). Commercial spreadsheet-based engineering software can also be used for the design and analysis of such springs (Schorcht et al., 1998; Liberman, 2006; Meissner & Schorcht, 2007). Computer methods based on genetic algorithms (Cimolai et al., 2022; Bai et al., 2021; Warzecha et al., 2022) are used to create complex spring shapes by optimising for specific criteria.

The results of the computer simulation of the coil spring properties require validation with experimental data. This process demands high agreement between the virtual and manufactured spring geometry. However, the inaccuracy of the manufacturing process is often the source of differences between them. The differences are mainly in the helix angle, inner and outer diameter, height, wire diameter and uniformity of the outer diameter along the entire length of the spring. In a publication (Gu et al., 2020), it was shown that the differences between the geometry of a spring designed in a commercial program and that of made in the factory can be so large that it makes validation of the numerical model difficult. A potential solution to this problem may involve application of 3D scanning technology, which allows virtual models from real objects. However, in the case of coil springs with bent end coils, this method has difficulties. The scanner cannot read the geometry of the surfaces that separate adjacent wires, forcing complicated processing of the obtained model.

The indicated problems show the need for better spring geometry creation algorithms. Such algorithms would be more flexible, allowing for creation of coil spring geometry with variable parameters described in a strict mathematical manner. To achieve this goal, mathematical relations are necessary to define the spatially curved axis of the spring wire. Such relations should give the coordinates of points laying on the curved axis as a function of the angular coordinate. Such an approach makes it possible, among other things, to build a spring with a progressive pitch, varying over the entire height of the spring. Previous studies relied heavily on the limited tools available in commercial CAD software for spring geometry creation (Chandravanshi & Mukhopadhyay, 2017; Gzal et al., 2017; Nazir et al., 2020; Sahu et al., 2022). Because of the limited design features, creation of intricate helical

spring geometry was based on simple CAD features such as extrude, sweep, revolve, and spline. Thus, the exact control of the spring geometry was either not possible or time-consuming. Therefore, the novel approach proposed in this article will support future studies. Implementation of mentioned algorithms would allow creation of spring models with more complex geometry, which will be a significant improvement for research conducted using FEM, commonly applied in the study of coil springs.

The purpose of this work is to develop an effective method for generating models of helical springs with arbitrary geometric parameters, within certain limits. Its application is demonstrated through examples. The proposed method uses the Python language and can be implemented in both open source and commercial software. The article begins with an introduction, then the mathematical formulation of the model is described, the program's algorithm is presented, and finally, the practical application of the method is presented for numerical testing.

2. MATHEMATICAL MODEL

The geometry of helical spring can be described using the parametric equations of the helix, which allow a transition from cylindrical coordinates to Cartesian coordinates (Yang et al., 2014):

$$\begin{cases} x = R\cos\varphi \\ y = R\sin\varphi \\ z = \frac{P(\varphi)}{2\pi}\varphi \end{cases} \quad (1)$$

where: R – radial coordinate, equal to half of the mean spring diameter,
 φ – azimuth, the angle between the position vector at the beginning of the helix and the position vector at the specific point on the helix, measured in the plane perpendicular to the axis of the helix,
 $P = 2\pi R \tan \delta$ – pitch of helix,
 $\delta = \delta(\varphi)$ – helix angle, which represents the angle between the axis of the wire and a line perpendicular to the axis of the spring, measured at the mean spring diameter,
 x, y, z – Cartesian coordinates.

In the case of cylindrical springs with a constant helix angle, the pitch value P in Eq. (1) is invariant, however, under real-world conditions helical compression springs usually have sections where this parameter changes gradually. Fig. 1a shows a section of a theoretical spring with a step-changing helix angle, and Fig. 1b shows a spring having a section with a continuously increasing helix angle. To accurately determine the height of the spring and its number of active coils, it is necessary to strictly formulate the geometry of the progressive pitch section. The most advantageous from the point of view of the fatigue strength of the spring is to use a constant transition radius between sections with different pitch angles of the helical line. The simplest example of this is shown in Fig. 2. This figure shows the development of the helical line of a spring with squared end coils, with transition sections of constant curvature in the circumferential direction, defined by radii ρ_1 and ρ_2 . The functions describing the course of these sections are denoted by z_{ρ_1} and z_{ρ_2} respectively.

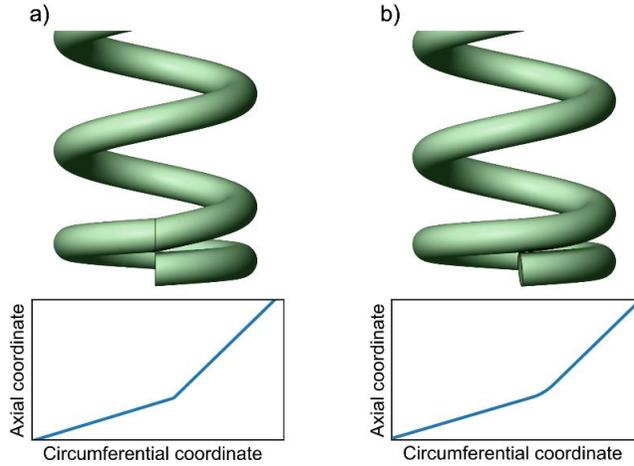


Fig. 1. Spring model with stepwise variable helix angle a), spring model with continuously variable helix angle b) with corresponding plots, showing helix angle change

In compression springs, the end coils i.e. the first and the last coils are adjacent to each other, providing stable support and reducing the maximum contact stresses. Under this assumption, the values of the helix angles δ_1 and δ_3 can be expressed by the relation:

$$\delta_1 = \delta_3 = \text{atan} \frac{r}{\pi R} \quad (2)$$

where r denotes half of the wire diameter. In general, the angles δ_1 and δ_3 can be arbitrarily chosen acute angles. The segments visible in Fig. 2 between points O_1 and A_1 and between points A_2 and O_2 represent end coils of numbers n_1 and n_3 , while the lines between points A_1 and A_2 correspond to active coils of number n_2 . The figures n_1 , n_2 and n_3 represent the number of corresponding coils. In general, these numbers can take arbitrary non-negative values. The relationships between the numbers of coils and the corresponding azimuth φ can be written in the following form:

$$\varphi_{A1} = 2\pi n_1; \quad \varphi_{A2} = 2\pi n_2 + \varphi_{A1}; \quad \varphi_{O2} = 2\pi n_3 + \varphi_{A2} \quad (3)$$

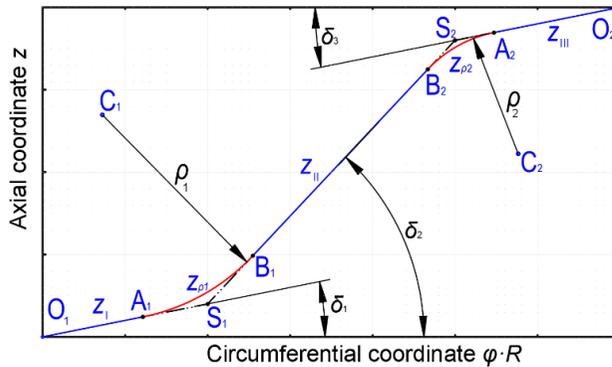


Fig. 2. Development of the wire centreline of a spring with squared end coils, with transition sections of constant curvature

In this paper, two cases of significant practical interest will be presented. In the first case, a model enabling the generation of a spring geometry with a fixed helix angle δ_2 will be presented, while the height of the spring will depend on the radii ρ_1 and ρ_2 . In the second case, the developed model will allow the generation of a spring geometry with a fixed overall height H and a helix angle δ_2 depending on the radii ρ_1 and ρ_2 .

In order to obtain a generalised calculation model and simplify the notation, a new relative variable \bar{z} was introduced:

$$\bar{z} = \frac{z}{R} \quad (4)$$

Similarly, the functions describing successive sections of the wire centreline development, expressed in relative coordinates φ and \bar{z} are denoted successively \bar{z}_I , \bar{z}_{ρ_1} , \bar{z}_{II} , \bar{z}_{ρ_2} and \bar{z}_{III} . In the same way, the relative radii of $\bar{\rho}_1 = \frac{\rho_1}{R}$ and $\bar{\rho}_2 = \frac{\rho_2}{R}$ are labelled.

2.2. Computational model of a spring geometry with a fixed helix angle of the active coils

Function \bar{z}_I describing the first linear segment of spring wire centreline in relative coordinates, has the following form:

$$\bar{z}_I = \varphi \tan \delta_1 \quad (5)$$

The slope of the second linear function \bar{z}_{II} is equal to $\tan \delta_2$, whilst its offset can be calculated using the consistency condition between z_I and z_{II} at point S_1 :

$$\bar{z}_I(\varphi = \varphi_{S_1}) = \bar{z}_{II}(\varphi = \varphi_{S_1}) \quad (6)$$

The azimuth of point S_1 depends on the azimuth value of point A_1 , the value of the relative radius $\bar{\rho}_1$ and the difference in angles δ_2 and δ_1 :

$$\varphi_{S_1} = \varphi_{A_1} + \bar{\rho}_1 \tan \frac{\delta_2 - \delta_1}{2} \cdot \cos \delta_1 \quad (7)$$

The angular coordinate of point S_1 can be calculated from Eq. (5):

$$\bar{z}_{S_1} = \varphi_{S_1} \tan \delta_1$$

Azimuth of point A_1 is expressed by Eq.(3). Using Eq. (6) and (7) we find the function \bar{z}_{II} :

$$\bar{z}_{II} = \varphi \tan \delta_2 + \left(\varphi_{A_1} + \bar{\rho}_1 \tan \frac{\delta_2 - \delta_1}{2} \cdot \cos \delta_1 \right) (\tan \delta_1 - \tan \delta_2) \quad (8)$$

On the basis of the well-known geometrical relationships, the function \bar{z}_{ρ_1} can be determined to describe the curve between points A_1 and B_1 :

$$\bar{z}_{\rho 1} = \bar{z}_{C1} - \bar{\rho}_1 \sqrt{1 - \left(\frac{\varphi - \varphi_{C1}}{\bar{\rho}_1}\right)^2} \quad (9)$$

where:

$$\bar{z}_{C1} = \varphi_{A1} \tan \delta_1 + \bar{\rho}_1 \cos \delta_1 \quad (10)$$

$$\varphi_{C1} = \varphi_{A1} - \bar{\rho}_1 \sin \delta_1 \quad (11)$$

The slope of the third linear function \bar{z}_{III} equals $\tan(\delta_3)$, and its offset can be calculated in a similar way to the offset in function \bar{z}_{II} . The azimuth of the point A_2 is given by Eq.(3), and points A_2 , S_2 and C_2 constitute the vertices of a right-angle triangle. Using the above, the azimuth of point S_2 can be determined and thus, with the help of elementary transformations, the function \bar{z}_{III} and the function $\bar{z}_{\rho 2}$ can be found:

$$\varphi_{S2} = \varphi_{A2} - \bar{\rho}_2 \tan \frac{\delta_2 - \delta_3}{2} \cdot \cos \delta_3 \quad (12)$$

$$\begin{aligned} \bar{z}_{III} &= (\varphi - \varphi_{S2}) \tan \delta_3 + \varphi_{S2} \tan \delta_2 + \\ &+ \left(\varphi_{A1} + \bar{\rho}_1 \tan \frac{\delta_2 - \delta_1}{2} \cdot \cos \delta_1 \right) (\tan \delta_1 - \tan \delta_2) \end{aligned} \quad (13)$$

$$\bar{z}_{\rho 2} = \bar{z}_{C2} + \bar{\rho}_2 \sqrt{1 - \left(\frac{\varphi_{C2} - \varphi}{\bar{\rho}_2}\right)^2} \quad (14)$$

where:

$$\bar{z}_{C2} = \bar{z}_{A2} - \bar{\rho}_2 \cos \delta_3 \quad (15)$$

$$\varphi_{C2} = \varphi_{A2} + \bar{\rho}_2 \sin \delta_3 \quad (16)$$

The relative axial coordinate of point A_2 can be determined by substituting φ_{A2} in Eq. (13) in place of φ . Total height of the spring corresponds to the axial coordinate of point O_2 :

$$H = R \cdot \bar{z}_{III}(\varphi = \varphi_{O2}) \quad (17)$$

Increasing the values of $\bar{\rho}_1$ i $\bar{\rho}_2$ results in a shortening of the length of the rectilinear segment described by the function \bar{z}_{II} , which in the limit case will be reduced to zero. The following condition should be satisfied:

$$\varphi_{B1} - \varphi_{A1} + \varphi_{A2} - \varphi_{B2} \leq 2\pi n_2 \quad (18)$$

Using the relations shown in Fig. 2, the inequality (18) can be written in the following form:

$$\bar{\rho}_1 \tan \frac{\delta_2 - \delta_1}{2} \cdot (\cos \delta_1 + \cos \delta_2) + \bar{\rho}_2 \tan \frac{\delta_2 - \delta_3}{2} \cdot (\cos \delta_2 + \cos \delta_3) \leq 2\pi n_2 \quad (19)$$

Assuming that $\delta_3 = \delta_1$, we obtain:

$$(\bar{\rho}_1 + \bar{\rho}_2)_{max} = \frac{2\pi n_2}{(\cos \delta_1 + \cos \delta_2) \tan \frac{\delta_2 - \delta_3}{2}} \quad (20)$$

2.3. Computation model of a geometry with a fixed total height of the spring

The functions \bar{z}_I and $\bar{z}_{\rho 1}$ describing the centreline of the wire on the first passive coil segment and the first transition segment with constant curvature have the form given by equations (5), (9), (10) and (11).

In the case of a fixed total height H of the spring, with fixed numbers of coils n_1 , n_2 and n_3 , the position of point O_2 is also known. Its axial coordinate in the assumed relative coordinate system equals:

$$\bar{z}_{O2} = \frac{H}{R} \quad (21)$$

whereas its azimuth φ_{O2} is given by Eqs. (3). Thus, the function \bar{z}_{III} describing the centreline of the wire between the points A_2 and O_2 will take a form:

$$\bar{z}_{III} = \bar{z}_{O2} - (\varphi_{O2} - \varphi) \tan \delta_3 \quad (22)$$

A second transition segment with a fixed relative radius of curvature $\bar{\rho}_2$ can be determined from Eq. (14), with the relative axial coordinate of point C_2 being determined from Eq. (19) by substituting $\varphi = \varphi_{C2}$. Under this assumption, the equation describing the $\bar{z}_{\rho 2}$ function can be written in the form:

$$\bar{z}_{\rho 2} = \bar{z}_{O2} + \bar{\rho}_2 \sqrt{1 - \left(\frac{\varphi_{C2} - \varphi}{\bar{\rho}_2} \right)^2} - (\varphi_{C2} - \varphi) \tan \delta_3 \quad (23)$$

where φ_{C2} is given by Eq. (16).

The angle δ_2 can be determined on the grounds of the geometric relationships between the characteristic points of the analysed system, which is shown in Fig. 3.

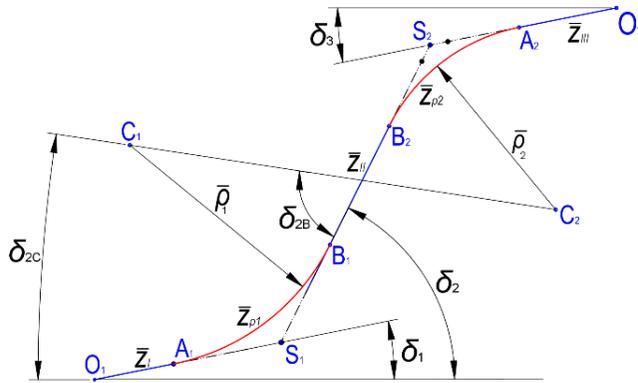


Fig. 3. Diagram of analysed geometric system

Let us denote the angle between the line passing through points C_1 and C_2 and the line connecting points B_1 and B_2 as δ_{2B} and the angle between the line passing through points C_1 and C_2 and the horizontal line as δ_{2C} . Based on the geometric relationships shown in Fig. 3, the equation describing the distance L_C between points C_1 and C_2 can be formulated:

$$L_C = \sqrt{(\varphi_{C2} - \varphi_{C1})^2 + (\bar{z}_{C1} - \bar{z}_{C2})^2} \quad (24)$$

And the equation describing the distance L_B between points B_1 and B_2 :

$$L_B = \sqrt{L_C^2 - (\bar{\rho}_1 + \bar{\rho}_2)^2} \quad (25)$$

The angles δ_{2B} and δ_{2C} amount to respectively:

$$\delta_{2B} = \tan^{-1} \frac{\bar{\rho}_1 + \bar{\rho}_2}{L_B} \quad (26)$$

$$\delta_{2C} = \tan^{-1} \frac{\bar{z}_{C1} - \bar{z}_{C2}}{\varphi_{C2} - \varphi_{C1}} \quad (27)$$

Using the designations as in Fig. 3, the angle δ_2 can be expressed as a difference:

$$\delta_2 = \delta_{2B} - \delta_{2C} \quad (28)$$

Already knowing the value of the angle δ_2 , it is possible to write the \bar{z}_{II} equation describing the centreline of the wire on the segment between points A_2 and O_2 :

$$\bar{z}_{II} = (\varphi - \varphi_{B1}) \tan \delta_2 + \bar{z}_{B1}$$

where:

$$\varphi_{B1} = \varphi_{C1} + \bar{\rho}_1 \sin \delta_2 \quad (29)$$

$$\bar{z}_{B1} = \bar{z}_{C1} - \bar{\rho}_1 \cos \delta_2 \quad (30)$$

Coordinates φ_{C1} and \bar{z}_{C1} are given by Eqs. (10) and (11).

The maximum possible value of the sum of $\bar{\rho}_1$ and $\bar{\rho}_2$ can be determined from Eq. (25) by substituting $L_B = 0$, however, this equation in case of arbitrary values of angles δ_1 and δ_3 can only be solved with respect to sum of $\bar{\rho}_1$ and $\bar{\rho}_2$ numerically.

Under the assumption, that $\delta_3 = \delta_1$, in the case where $L_B = 0$, the distance between the points C_1 and C_2 is constant, regardless of the ratio between the relative radii $\bar{\rho}_1$ and $\bar{\rho}_2$. The same applies to the value of the angle δ_{2C} . Using the above conditions, an equation enabling the maximum value of the sum of $\bar{\rho}_1$ and $\bar{\rho}_2$ to be determined can be written:

$$(\bar{\rho}_1 + \bar{\rho}_2)_{max} = \frac{\sqrt{(\bar{z}_{A2} - \bar{z}_{A1})^2 + (\varphi_{A2} - \varphi_{A1})^2}}{2 \sin(\delta_{2I} - \delta_1)} \quad (31)$$

where δ_{2I} denotes the helix angle of the active coils of the spring with initial geometry, i.e. when $\bar{\rho}_1 = \bar{\rho}_2 = 0$. Using Eq. (3), Eq. (31) after transformations will take the following form:

$$(\bar{\rho}_1 + \bar{\rho}_2)_{max} = \frac{\pi n_2 \sqrt{\tan^2 \delta_{2I} + 1}}{\sin(\delta_{2I} - \delta_1)} \quad (32)$$

3. IMPLEMENTATION OF THE CALCULATION ALGORITHM

The mathematical relations given in Section 2 define the centerline of the spring wire. These relations can be applied to obtain the spring geometry with the constant helix angle or the constant spring height. Nevertheless, such process may be time-consuming and require scripting skills, which creates a difficulty for a user. To mitigate this problem, a Python script was created. The general idea behind this script utilises the fact that most CAD software offers scripting capabilities or Application Programming Interfaces (APIs), which allow an automation of a geometry creation process. This section describes significant features of the Python script and gives a reference for its usage.

There are many CAD programmes that can be used to generate the spring geometry. Most of them offer scripting languages, which can be used to create macros, or APIs allowing for creation of user-defined functions. Both features can be used to automate geometry creation. Because the scripting languages or APIs are usually not compatible between different CAD software, there was no possibility to create a universal solution. Consequently, the Python script currently can generate an input file for two selected programmes, SpaceClaim and Gmsh (Geuzaine & Remacle, 2009), but it was conceptualised to be easily extendable for other CAD software. This was achieved by modularity of the source code, in which separate functions were utilised for each task. The selected programs were chosen because of the possibility to generate finite element mesh based on 3D geometry, licence availability and their grounded position in scientific community.

The script source code was divided into three parts: user-defined parameters, algorithm implementation, and CAD software interface. The user-defined parameters were intended as an interface for the user. They contain variables which can be assigned by the user to select the algorithm (either constant helix angle or constant spring height), used CAD software (currently SpaceClaim and Gmsh are supported) and define needed spring geometry parameters. Each variable was documented directly in the source code comments. The algorithm implementation part consists of two functions: `gen_central_points_angle_control()` (implements equations given in Section 2.2) and `gen_central_points_height_control()` (implements equations given in Section 2.3). Both functions return a NumPy array containing the coordinates of points defining the spring helix. This array is then taken as an input by the third part functions, which build the CAD interface. These functions generate files, which can be directly used in the CAD programme to generate the spring geometry. Currently two interfaces are available (SpaceClaim and Gmsh) but this part can be easily extended by adding functions specific to other CAD software.

The structure and workflow of the Python script is presented in Fig. 4. First, the user defines the required parameters defined as variables. Some of them are specific to the chosen algorithm. Then the script is run with Python interpreter. It automatically selects the specified algorithm and generates the CAD input file in the given folder. This file can then be used in the CAD software to create spring geometry. Next section presents exemplary use cases based on SpaceClaim. The script source code can be found on GitHub [<https://github.com/warzechm/Coil-Spring-Geometry-Generation.git>]

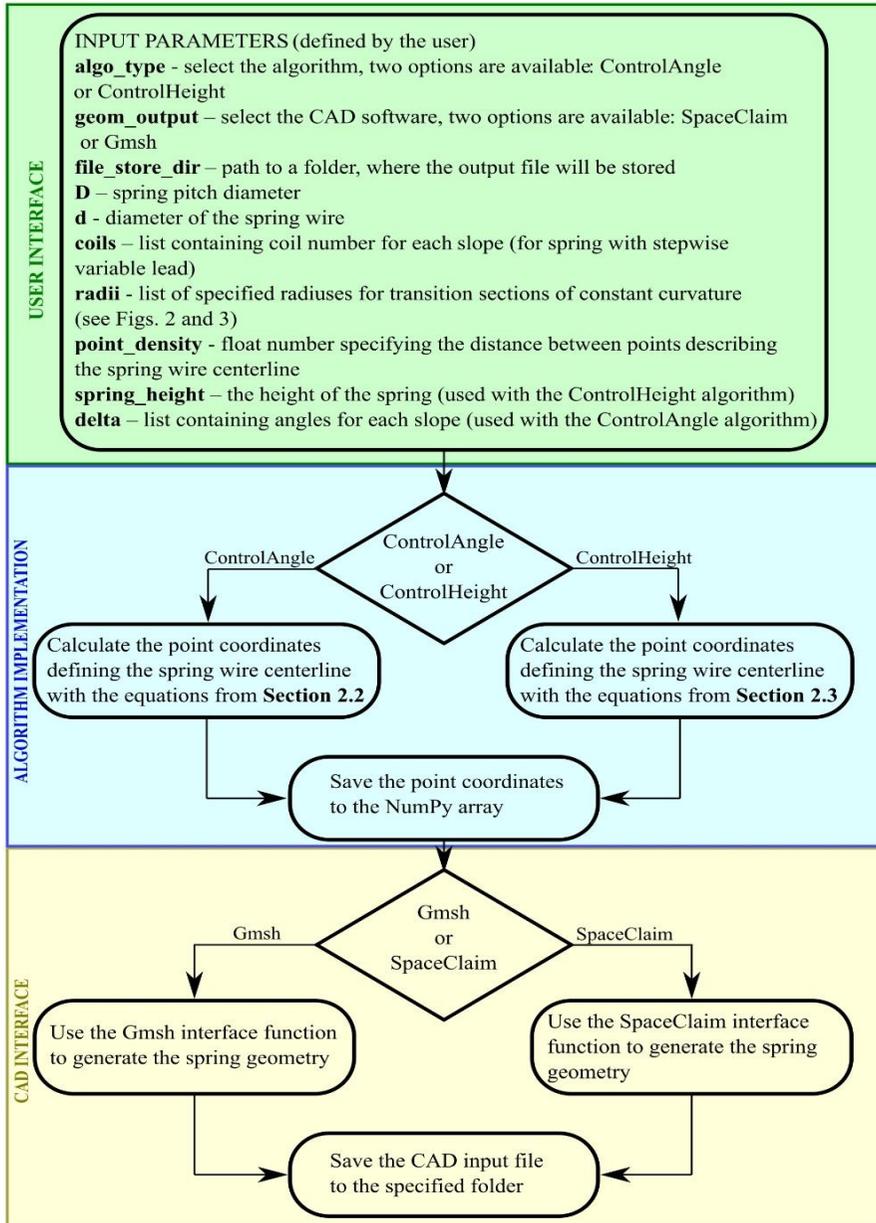


Fig. 4. The diagram of the Python script implementation

4. PRACTICAL APPLICATION OF THE WRITTEN SCRIPT

The mathematical models presented in Chapter 2 and their implementation in the Python language environment make it possible to generate models of compression springs with strictly defined geometries and effectively analyze the impact of the modifications introduced on the properties of these springs. The curvature of the helical line of the

transition section has a significant impact on the properties of the spring. By changing the curvature value of the radii ρ_1 and ρ_2 , the nonlinearity of the characteristics of the designed spring can be controlled to a large extent.

Demonstration of the operation of algorithms generating models of springs and the impact of geometry modifications on their static characteristics was performed for five springs using the SpaceClaim and Ansys Workbench software. The geometric parameters of the prepared models and the models of the springs are shown in Fig. 5. The springs have one passive coil on each side, ground to an end thickness equal to 0.25 of the wire diameter.

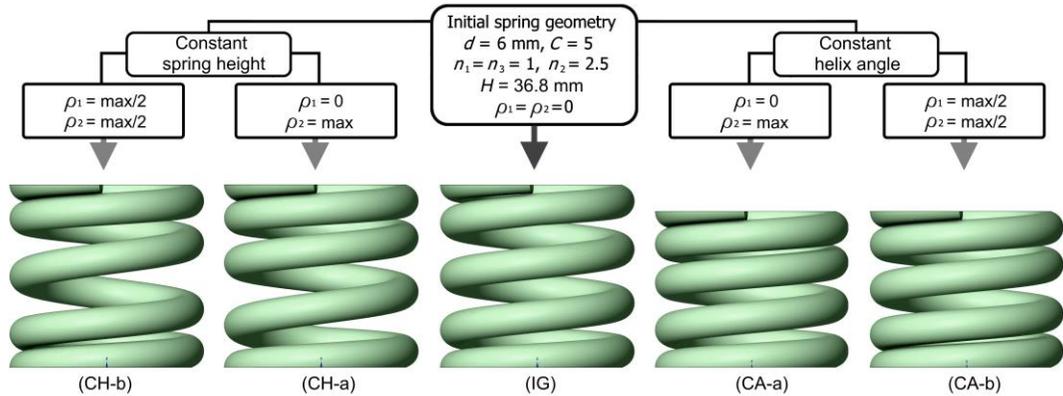


Fig 5. Geometrical parameters of modelled springs, models and their designations

In this figure, under the springs, the accepted designations used in the following part of the work are also given in parentheses. Initial spring (IG) with rounding radii $\rho_1 = \rho_2 = 0$, was modified using the constant height (CH) algorithm and the constant helix angle (CA) algorithm, changing the value of these radii. In both cases, a spring was generated in which the radius ρ_2 covered all active coils (designation -a) while the radius $\rho_1 = 0$ and a spring in which both the radii ρ_1 and ρ_2 had the same value and the roundings also covered all active coils (designation -b).

The completed models were transferred to the Ansys Workbench environment to simulate quasi-static compression. On the basis of the analysis, the load-deflection curves of the 5 springs were determined. To represent real support conditions, supporting plates with a diameter of 45 mm and a thickness of 5 mm were modelled. Frictional contact between the individual halves of the spring coils and between the ground surfaces and the plates was set. Steel was given as the material of the spring and supporting plates. The coefficient of friction between the spring and plates was assumed to be equal to 0.18 in each case. The same value of the coefficient of friction was assumed in the contact pairs between the surfaces of the coils. The material properties of the supports and springs were assumed to be as follows: Young's modulus $E = 206,000 \text{ MPa}$, Kirchhoff's modulus $G = 79,500 \text{ MPa}$. Mesh parameters selection was carried out on the basis of initial simulations of axial compression of the IG spring. Initial discretization of the model was carried out using tetrahedral, 10-node elements with an edge length of 2 mm for the entire model (47159 nodes, 28534 elements). The axial stiffness of the spring was calculated equal to 159.66 N/mm in the force range from 100 N to 1000 N. Then the edge length of the elements was reduced to 1.5 mm (105466 nodes, 67112 elements) and an axial stiffness of 159.62 N/mm was obtained. Due to the negligible

difference in results (less than 0.05%), a further reduction in element size was regarded as unnecessary. For all five spring models, the average skewness mesh parameter was 0.24, which is a satisfactory value (Fatchurrohman & Chia, 2017) and the average aspect ratio was 1.95, which indicates that the obtained mesh is of high quality. Compression analyses were performed including effects associated with large deformations. In the analyses carried out, an axial load increasing in 30 steps from 0 to 3000 N was applied to the top surface of the upper plate. In addition, all rotational DOFs of this surface were fixed as well as translational DOFs in transverse directions. The lower support was restrained. The boundary conditions and meshed model of the spring CH-a are shown in Fig. 6a. Fig. 6b shows the displacement contour plot for this spring under a load of 3000 N, and Fig. 6c shows the axial load-deflection curves obtained for each of the five springs.

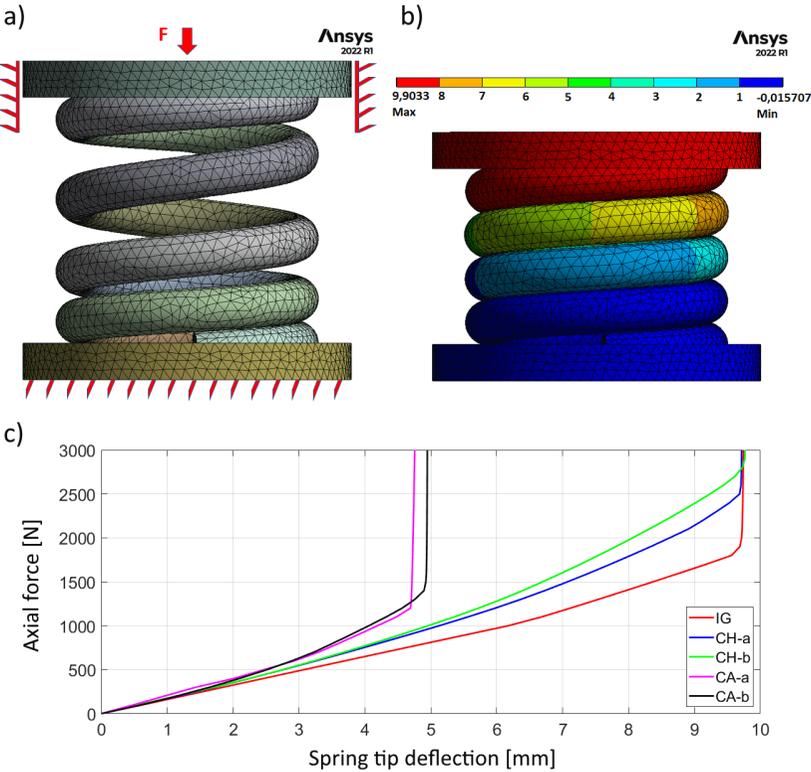


Fig. 6. CH-a spring model with illustration of analysis conditions a), example of displacement contour plot b), axial load-deflection curves of all 5 spring models obtained from FEM analyses c)

The load-deflection curves presented in Fig. 6c show significant variation. It can be seen that the axial stiffnesses of the CH-b and CA-b springs show the greatest non-linearity. The IG spring model shows an approximately linear load-deflection curve with a segmentally varying slope. The change in the slope of the curve occurs around 60% of the maximum deflection and is related to the elimination of the initial clearance between the passive coil and half of the first active coil on both sides of the spring. In order to further determine the impact of the modifications made on the spring load-deflection curves, the fitting operation

was performed for each of them. The approximation was carried out for a deflection ranging from zero to about 95% of the maximum compression, i.e. solid compression. Curve fitting was carried out using the power function of the form:

$$F(u) = a \cdot u^b \quad (33)$$

where F is the compression axial force and u is the spring tip deflection. The obtained values of coefficients a and b and determination coefficients R -square for each approximation are presented in Table 1 and Fig. 7 shows an example of a graphical representation of the fitting results for spring IG (Fig. 7a) and spring CH-b (Fig. 7b).

Tab. 1. Parameters of approximating functions for each of the 5 characteristics

Spring:	CH-b	CH-a	IG	CA-a	CA-b
a	107.2	121.8	132.1	171.3	134.9
b	1.404	1.298	1.139	1.219	1.435
R -square	0.9968	0.9969	0.9967	0.9945	0.9973

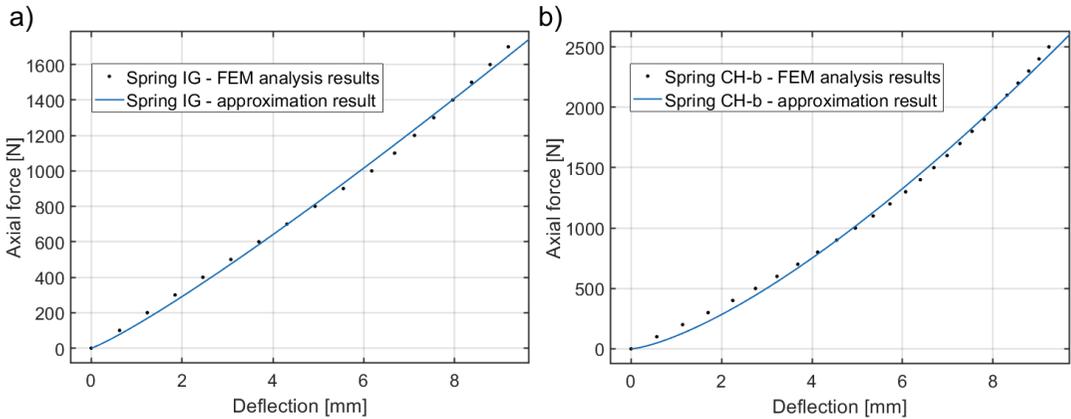


Fig. 7. Graphical representation of the curve fitting results for IG spring model a), and for CH-b spring model b) with illustration of analysis conditions a), example of displacement contour plot b), axial load-deflection curves of all 5 spring models obtained from FEM analyses c)

Analyzing the results shown in Table 1, it can be seen that the approximated load-deflection curves of springs with one-sided rounding show less nonlinearity than those of springs with both sides rounding. It can also be observed that the load-deflection curve of the CA-a spring shows the highest value of the slope coefficient a among all tested spring models. At the same time the load-deflection curve of this spring model is characterized by a relatively low value of the power exponent b . This example demonstrates the wide possibilities of modifying the load-deflection curve of a compression spring using the proposed geometry generation method in order to obtain the desired course of axial stiffness variation with changing load.

5. CONCLUSION

The mathematical models presented in this paper, together with their implementation in the numerical environment, make it possible to create helical spring geometries with both linear and progressive characteristics. The assumption of a fixed value for the radius of curvature of nonlinear sections makes it possible to generate spring geometries with a favorable distribution of stresses along the length of the spring wire. Thanks to the use of a strict model describing the spring geometry, it is possible to precisely match it to the requirements determined by the geometry of the spring installation in the machine. Numerical analyses have shown that the proposed model allows, under given boundary conditions, significant modifications of the load-deflection curve both in terms of changing the original angle of inclination of this curve and its nonlinearity. For a deflection of approximately 95% of the maximum value, the linearised stiffness of the spring with initial geometry (IG) reached 271 N/mm. This indicates that, in this case, the modification of the geometry resulted in a stiffness increase of approximately 46%. When comparing the stiffness of the IG spring with that of the CH-a spring, an increase in stiffness of approximately 33% was obtained. Analogous stiffness values were obtained for springs CA-b – 273 N/mm and CA-a – 246 N/mm. From the results presented in Table 1 it can also be concluded that the power function approximates the obtained load-deflection curves with a high degree of accuracy and the determination coefficient exceeded the value of 0.99 in each case.

The scripts developed in the Python environment can be implemented in any CAD software equipped with Application Programming Interfaces (APIs). The spring models generated with them can be characterized by a highly progressive load-deflection curve. Springs with such axial stiffness are desirable for many applications, especially in systems subjected to dynamic loads, since this type of load-deflection curve contributes to limiting vibration amplitudes under peri-resonant conditions. This applies both to vibrations of mass-spring systems and longitudinal vibrations of the spring itself. The paper also shows that the use of a power function to approximate force-deflection curves for springs with geometries generated by the proposed methods gives satisfactory results. In the worst case studied, the coefficient of determination was 0.9945.

Author Contributions

Concept of the paper, development of mathematical models, editing work, partially numerical analysis, KM; development and implementation of the numerical algorithm, manuscript co-writing and editing, MW; preparation of the numerical model, carrying out simulations, partially introduction, RB. All authors have read and agreed to the published version of the manuscript.

Funding

This project was financed by the AGH University of Science and Technology under research program No. 16.16.130.942

Conflicts of Interest

The authors declare no conflict of interest

REFERENCES

- Arshad, A., Nazir, A., & Jeng, J.-Y. (2022). Design and performance evaluation of multi-helical springs fabricated by Multi Jet Fusion additive manufacturing technology. *International Journal of Advanced Manufacturing Technology*, 118, 195-206. <https://doi.org/10.1007/s00170-021-07756-2>
- Bai, J.-B., Liu, T.-W., Wang, Z.-Z., Lin, Q.-H., Cong, Q., Wang, Y.-F., Ran, J.-N., Li, D., & Bu, G.-Y. (2021). Determining the best practice – Optimal designs of composite helical structures using Genetic Algorithms. *Composite Structures*, 268, 113982. <https://doi.org/10.1016/j.compstruct.2021.113982>
- Chandravanshi, M.L., & Mukhopadhyay, A.K. (2017). Analysis of variations in vibration behavior of vibratory feeder due to change in stiffness of helical springs using FEM and EMA methods. *Braz. Soc. Mech. Sci. Eng* (vol. 39, pp. 3343–3362). Springer. <https://doi.org/10.1007/s40430-017-0767-z>
- Cimolai, G., Dayyani, I., & Qin, Q. (2022). Multi-objective shape optimization of large strain 3D helical structures for mechanical metamaterials. *Materials & Design*, 215, 110444. <https://doi.org/10.1016/j.matdes.2022.110444>
- Ding, X., & Selig J.-M. (2004). On the compliance of coiled springs. *International Journal of Mechanical Sciences*, 46(5), 703-727. <https://doi.org/10.1016/j.ijmecsci.2004.05.009>
- Fatchurrohman, N., & Chia, S.-T. (2017). Performance of hybrid nano-micro reinforced mg metal matrix composites brake calliper: simulation approach. *Materials Science and Engineering*, 257, 012060. <https://doi:10.1088/1757-899X/257/1/012060>
- Geuzaine, C., Remacle, J.-F. (2009). Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities: THE GMSH PAPER, *International Journal for Numerical Methods in Engineering*, 79(11), 1309–1331. <https://doi.org/10.1002/nme.2579>
- Gobbi, M., & Mastinu, G. (2001). On the optimal design of composite material tubular helical springs. *Meccanica*, 36, 525-553. <https://doi.org/10.1023/A:1015640909013>
- Gu, Z., Hou, X., Keating, E., & Ye, J. (2020). Non-linear finite element model for dynamic analysis of high-speed valve train and coil collisions. *International Journal of Mechanical Sciences*, 173, 105476. <https://doi.org/10.1016/j.ijmecsci.2020.105476>
- Gzal M., Groper, M., & Gendelman, O. (2017) Analytical, experimental and finite element analysis of elliptical cross-section helical spring with small helix angle under static load. *International Journal of Mechanical Sciences*, 130, 476-486. <https://doi.org/10.1016/j.ijmecsci.2017.06.025>
- Liberman, K. (2006). *Optimierung von Schraubendruckfedern*. Technische Akademie Esslingen
- Liu, H., & Kim, D. (2009). Effects of end Coils on the Natural Frequency of Automotive Engine Valve Springs. *International Journal of Automotive Technology*, 10(4), 413–420. <https://doi.org/10.1007/s12239-009-0047-8>
- Meissner M., & Schorcht H.-J. (2007). *Metallfedern - Grundlagen, Werkstoffe, Berechnung, Gestaltung und Rechnereinsatz*. Springer.
- Michalczyk, K. (2015). Analysis of lateral vibrations of the axially loaded helical spring. *Journal of Theoretical and Applied Mechanics*, 53(3), 745–755. <https://doi.org/10.15632/jtam-pl.53.3.745>
- Nazir, A., Ali, M., Hsieh, CH., & Jeng J.W. (2020). Investigation of stiffness and energy absorption of variable dimension helical springs fabricated using multijet fusion technology. *The International Journal of Advanced Manufacturing Technology*, (vol. 110, pp. 2591–2602). Springer. <https://doi.org/10.1007/s00170-020-06061-8>
- Pöllänen, I., & Martikka, H. (2010). Optimal re-design of helical springs using fuzzy design and FEM. *Advances in Engineering Software*, 41(3), 410-414. <https://doi.org/10.1016/j.advengsoft.2009.03.010>
- Rahul, M.S., & Rameshkumar, K. (2021). Multi-objective optimization and numerical modelling of helical coil spring for automotive application. *Materialstoday: Proceedings*, 46(10), 4847–4853. <https://doi.org/10.1016/j.matpr.2020.10.324>
- Sahu, D. K., Dandena, J., Mahapatra, T. R., & Mishra. D. (2022). Design and Characterization of Progressive Coil Spring for Suspension Systems. *Journal of The Institution of Engineers (India): Series C*, 103, 705–715. <https://doi.org/10.1007/s40032-022-00817-9>
- Schorcht, H.-J., Kletzin, U., Micke, D., Wauro, F. (1998). Entwicklung eines modularen, wissensbasierten CAD/FEM-Systems zur integrierten Gestaltung und Berechnung von Federn und Federanordnungen. In Pahl G. (Ed.) *Professor Dr.-Ing. E.h. Dr.-Ing. Wolfgang Beitz zum Gedenken Sein Wirken und Schaffen*, (pp. 543-557). Springer-Verlag Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-41164-3>

- Taktak, M., Dammak, F., Abid, S., & Haddar, M. (2008). A finite element for dynamic analysis of a cylindrical isotropic helical spring. *Journal of Mechanics, Materials and Structures*, 3(4), 641–658. <https://doi.org/10.2140/jomms.2008.3.641>
- Warzecha, M., Michalczyk, K., & Machniewicz, T. (2022). A novel slotted cylinder spring geometry with an improved energy storing capacity. *Arabian Journal for Science and Engineering*, 47, 15539–15549. <https://doi.org/10.1007/s13369-022-06692-x>
- Wittrick W.-H. (1966). On elastic wave propagation in helical springs. *International Journal of Mechanical Sciences*, 8(1), 25-47. [https://doi.org/10.1016/0020-7403\(66\)90061-0](https://doi.org/10.1016/0020-7403(66)90061-0)
- Yang, C.-J., Zhang, W.H., Ren, G.X., & Liu, X.-Y. (2014). Modeling and dynamics analysis of helical spring under compression using a curved beam element with consideration on contact between its coils. *Meccanica*, (vol. 49, pp. 907–917). Springer. <https://doi.org/10.1007/s11012-013-9837-1>
- Zhao, J., Gu, Z., Yang, Q., Shao, J., & Hou, X. (2023). Dynamic Finite Element Model Based on Timoshenko Beam Theory for Simulating High-Speed Nonlinear Helical Springs. *Sensors*, 23(7), 3737. <https://doi.org/10.3390/s23073737>
- Zhuo, Y., Qi, Z., Zhang, J., & Wang, G. (2022). A geometrically nonlinear spring element for structural analysis of helical springs. *Archive of Applied Mechanics*, 92, 1789–1821. <https://doi.org/10.1007/s00419-022-02147-9>