

HU ISSN 1785-6892 in print
HU ISSN 2064-7522 online

DESIGN OF MACHINES AND STRUCTURES

A Publication of the University of Miskolc

Volume 7, Number 1 (2017)



Miskolc University Press
2018

EDITORIAL BOARD

- Á. DÖBRÖCZÖNI
Editor in Chief
Institute of Machine and Product Design
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
machda@uni-miskolc.hu
- Á. TAKÁCS
Assistant Editor
Institute of Machine and Product Design
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
takacs.agnes@uni-miskolc.hu
- R. CERMAK
Department of Machine Design
University of West Bohemia
Univerzitni 8, 30614 Plzen, Czech Republic
rcermak@kks.zcu.cz
- B. M. SHCHOKIN
Consultant at Magna International Toronto
borys.shchokin@sympatico.ca
- W. EICHLSEDER
Institut für Allgemeinen Maschinenbau
Montanuniversität Leoben,
Franz-Josef Str. 18, 8700 Leoben, Österreich
wilfrid.eichlseder@notes.unileoben.ac.at
- S. VAJNA
Institut für Maschinenkonstruktion,
Otto von Guericke Universität Magdeburg,
Universität Platz 2, 39106 Magdeburg, Deutschland
vajna@mb.uni-magdeburg.de
- P. HORÁK
Department of Machine and Product Design
Budapest University of Technology and Economics
horak.peter@gt3.bme.hu
H-1111 Budapest, Műegyetem rkp. 9.
MG. ép. I. em. 5.
- K. JÁRMAI
Institute of Materials Handling and Logistics
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
altjar@uni-miskolc.hu
- L. KAMONDI
Institute of Machine and Product Design
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
machkl@uni-miskolc.hu
- GY. PATKÓ
Department of Machine Tools
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
patko@uni-miskolc.hu
- J. PÉTER
Institute of Machine and Product Design
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
machpj@uni-miskolc.hu

CONTENTS

<i>Csiszár, L. Richárd:</i> Safety increase with material removal	5
<i>Florian, Pavel–Cermak, Roman:</i> Solving multibody dynamics problems using Python.....	15
<i>Jónás, Szabolcs–Tisza, Miklós:</i> Use of surface response methodology (RSM) in clinching process	23
<i>Kratochvil, Martin:</i> Mesh influence on contact stress results of teeth spur gears	32

SAFETY INCREASE WITH MATERIAL REMOVAL

CSISZÁR L. RICHÁRD

Budapest University Of Technology and Economics,
Department of Machine and Product Design

Abstract: This study is about an optimization of an axial tensioned plate with a central single circular hole. The objective is to increase the safety with remove material from the plate. The study contains the method of the optimization, topology and shape optimization.

Keywords: *optimization, topology optimization, shape optimization*

1. INTRODUCTION

The era of computer simulation the machine design process getting shorter. The simulation replaces the expensive and time-consuming physical tests. The structural optimization was only privileged by the researches but with commercial software the product designers can use these tools easily.

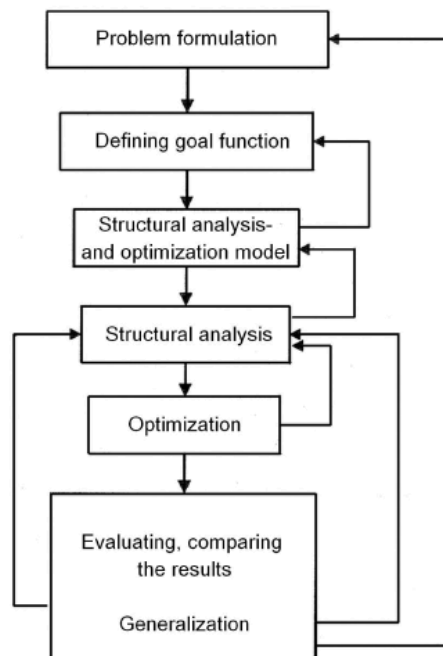


Figure 1. Design process with optimization step [3]

2. TASK PRE-PROCESS

2.1. Define the problem [2]

The task is to find a geometry which gives the highest safety, the geometry seen below.

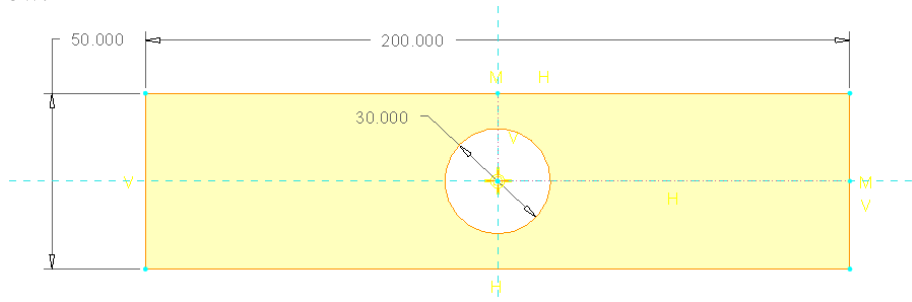


Figure 2. Original geometry

The material of the plate is S355 EN 10025:2004 structural steel. Around the hole for bolt must be a non-removable material 20 mm outside the hole. The tension force is 20 kN. The safety is equal to the maximum von Mises stress divided by the yield strength.

2.2. Data and information collecting

S355 EN 10025:2004 has the following properties:

Table 1. Material properties

Yield strength	355 MPa
Young modulus	210 GPa
Poisson ratio	0.3
Density	7850 kg/m ³

The pre calculations for the structure can see below. This is a stress concentration area example [1].

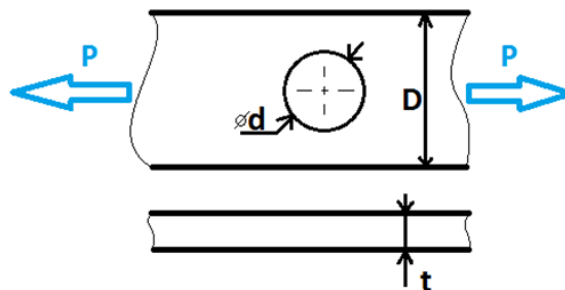


Figure 3. Parameters of the calculation

$$\text{Structural stress calculation} \quad \sigma_{nom} = \frac{P}{[t \cdot (D - d)]} = 200 \text{ MPa} \quad (1)$$

$$\text{Define } K_t \text{ factor} \quad 0 \leq d/D \leq 1 \quad (2)$$

$$K_t = 3.000 - 3.140 \left(\frac{d}{D}\right) + 3.667 \left(\frac{d}{D}\right)^2 - 1.527 \left(\frac{d}{D}\right)^3 = 2.11 \quad (3)$$

$$\text{The maximum principal stress} \quad \sigma_{max} = K_t \sigma_{nom} = 422 \text{ MPa} \quad (4)$$

It means that the plate with a hole has the maximum tension stress of 422 MPa. It is higher than the yield strength so the safety is lower than acceptable.

2.3. Definition of variables

The design variables can be seen in *Figure 4*, which is the green area. The elements around the constraints and elements close to the hole must be a non-design space.

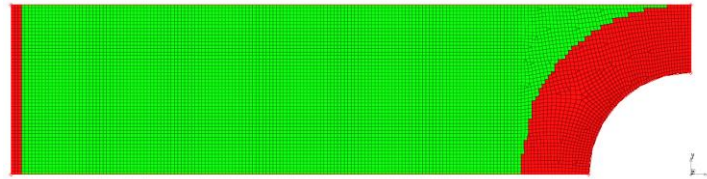


Figure 4. Design variables in quarter model

2.4. Objective of the optimization

The objective is the mass reduction, because the task is to remove material and reduce stress too. The algorithm search lighter structure, just the case of mass minimizing objective.

2.5. Constraints of the optimization

The constraint is the yield strength, so none of the elements can reach the 355 MPa von Mises stress.

3. OPTIMIZATION PROCESS

3.1. FE model

The first step is to build up the FE model of the structure. It is an obligatory task, because of the comparison with the analytical calculation. This is the input of the optimization, so it must be eligible. The analysis and the optimization is making by HyperWorks®. The geometry, the loads and the constraints are symmetrical, so

symmetrical simplification can be used. The model is calculated in 2D plane stress so 2D elements can be used. The constraints can be seen on Figure 5. $dX = 0$ at the lower edge of the model and $dY = 0$ at the right side of the structure. The loads applied on the left side of the structure, the half of the structures has half of the loads and it is 10 kN.

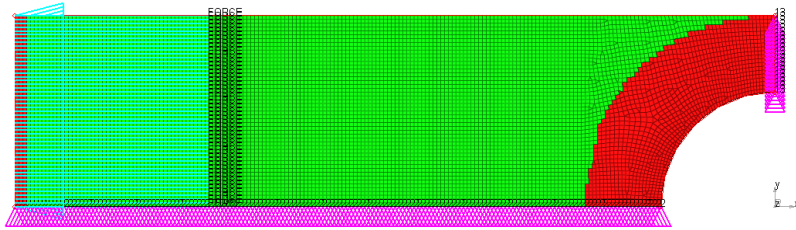


Figure 5. FEA model

The result of the analysis can be seen on Figure 6. The differences between the analytical (Equation 1–4) and computational results are 3% so it is acceptable.

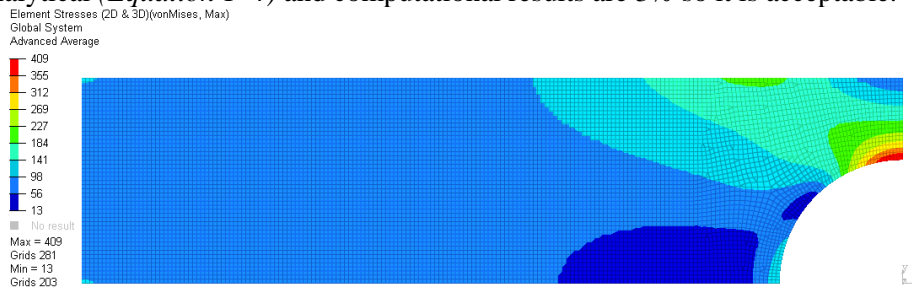


Figure 6. Result of the FEA

3.2. Parameters of the Optimization

The parameters in the first iteration are the task defined parameters. The variables are topology variables; that mean that the algorithm can change the relative modulus of the elements from almost zero to the real value of the rigidity. The total volume fraction response function has applied. The non-design space elements also have a response and it is the static stress response. The stress in the design space can be constrained with maximum von Mises stress as a parameter of the design variable, not as a response function. The non-design space is constrained as well, 355 MPa is the upper boundary. The objective is to minimize the volume fraction.

3.3. Results of the optimization

The process of optimization the algorithm reduces the volume and pay attention to the responses, stress responses in this case. These steps can be seen on the Figure 7. In the graph shows that in the second step the constraint is satisfied, so the stresses

are lower than 355 MPa after the third iteration, the volume is dropping monotonous. The 19th iteration the constraint is penetrated so the next iteration the volume begins to rise but just for a few iterations. The convergence criteria reached after the 40 iteration so the iteration process stopped.

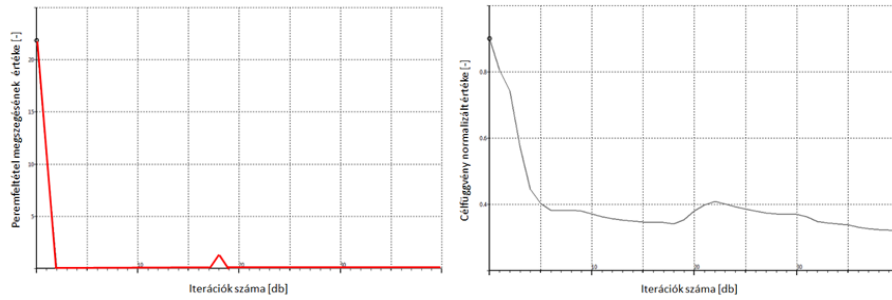


Figure 7. Iteration steps

The rough result can be seen on *Figure 8*. The explanation of the result is the following. The picture a) shows the density plots from 0 to 1 value. The red area shows the element with E0 young modulus so the original one. The blue areas have much lower modulus and density values almost zero. The real values in the reality can be zero or E0 nothing between, so that is why the structure must be interpreted. The b) picture shows the stress plot of the rough result.

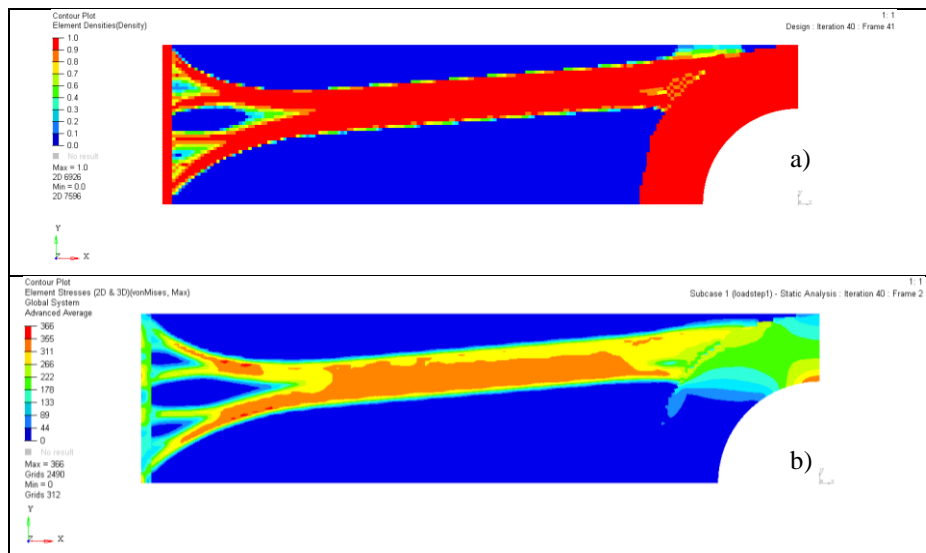


Figure 8. Result of optimization

4. INTERPRETATION AND VERIFICATION

4.1. Automatic smooth

OSSMOOTH® is a build in program, where the results can be interpreted automatic. The first step is to delete the elements which are lower, than the input parameter, here 0.5 is used. The next step is that the algorithm smoothed the surfaces of this rough area and then it could re-mesh the smoothed surface it is on the *Figure 9*. The most useful feature is that the constraints and the loads are reapplied by the program.

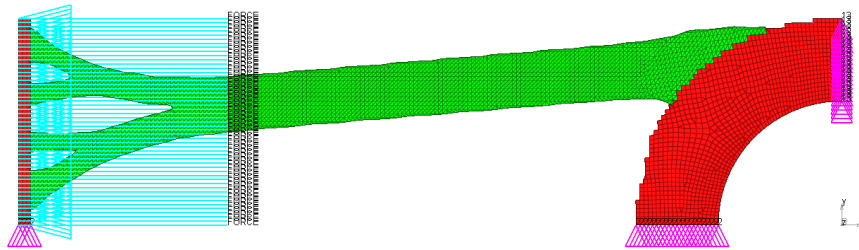


Figure 9. Smoothed geometry

4.2. Manual smooth

The interpreted geometry can be created in a CAD program. The process is easy with 2D structure. Take the picture of the rough optimization result, paste it into a plane and draw it. In 3D the rough surface of the result can be exported to *.stl file format. The 2D interpretation can be seen on *Figure 10*.

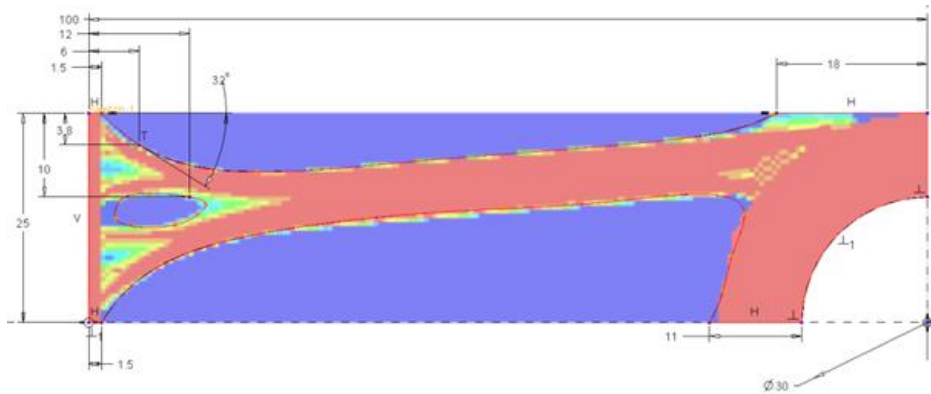


Figure 10. Manual interpretation

4.3. Usability of the result

The validation of the interpreted geometry is the next step. The *Figure 11 a)* part shows the non-interpreted result, this is feasible, but if the almost zero rigidity elements are disappeared the geometry is unfeasible. The *b)* shows the automatic

interpreted geometry, *c*) shows the manual interpreted geometry, and none of them is feasible. The result needs a small amount of modification.

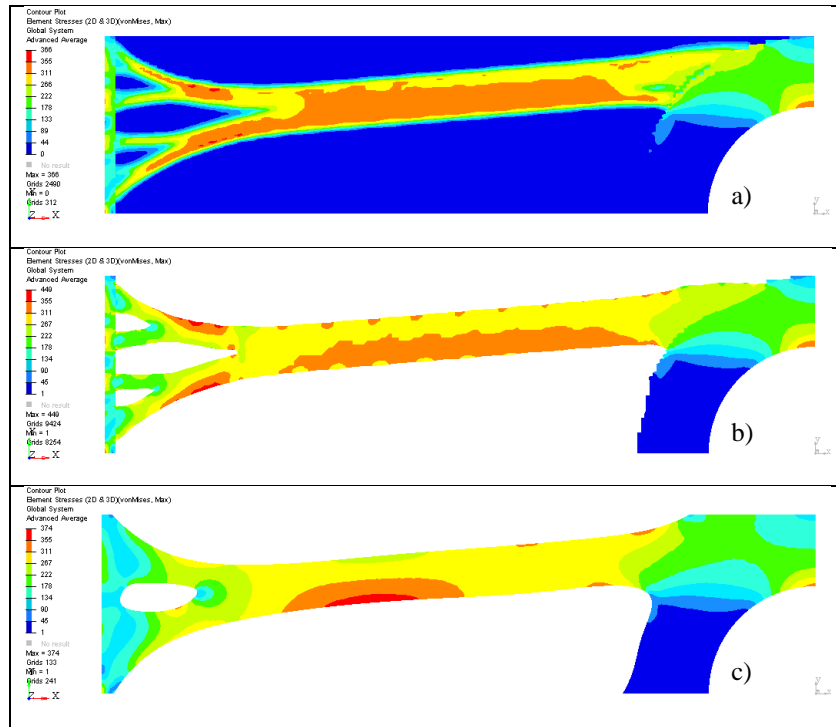


Figure 11. Validation of the result

4.4. Re-optimizing the result

The shape optimization is a tool for re-optimize the almost fine structure. The method can modify the nodes coordinates *Figure 12*, and with that local stress concentrations can be removed. The loads and constraints are the same, as in the FE model.

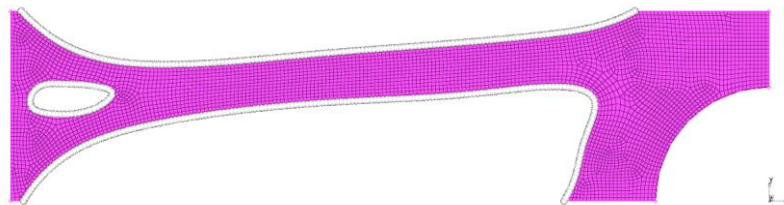


Figure 12. Design variables of the shape optimization

The task is to reduce stress on hot spots. The objective was the volume reduction but now it is reduced, so here the goal is to find the feasible geometry. The software can

use minmax objective for stress response, it means that the previous maximum stress should be reduced for the next step. The shape optimization should be limited by the geometry because it can be larger than the original geometry. The node coordinates are limited Y+ directions in the upper side and Y- directions in the lower side. The results can be seen on *Figure 13*. The *a)* show the starting geometry; in the *b)* the final iteration stress plot can be seen. The *c)* shows the shape change magnitude in mm.

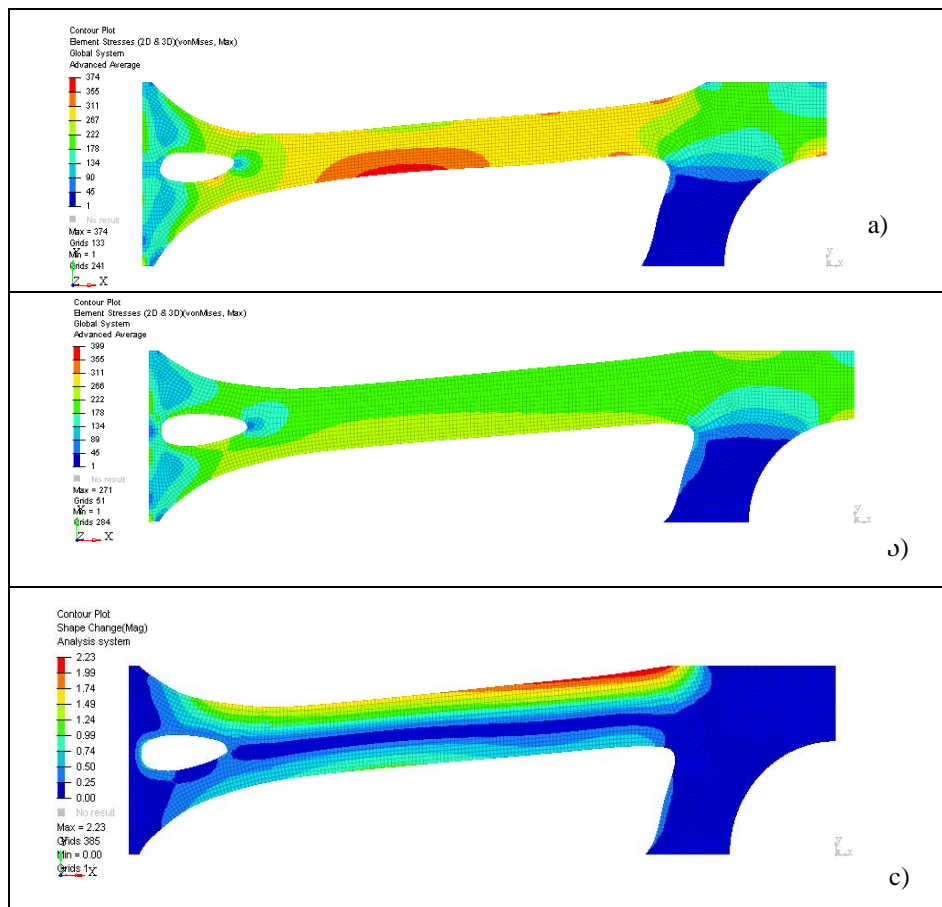


Figure 13. Minmax stress objective

If the case is to reach the minimum mass than the parameters of the optimization is different. The objective is the same as in the topology optimization, this is volume minimizing, the stresses constrained for the whole structure. The result of this optimization can be seen on *Figure 14* a) shows the starting geometry, b) shows the final iteration, and the c) shows the change of the shape.

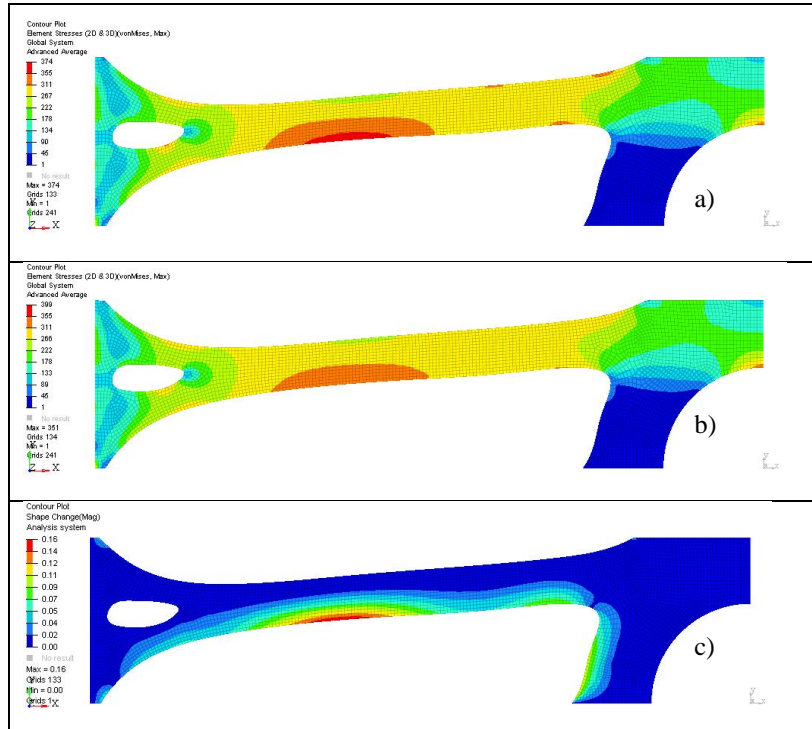


Figure 14. Minimizing volume

5. SUMMARY

The refine process the first task objective has to be interpreted, for define the sub task objectives. The final geometry can be difficult to manufacture, or can cause more problems for the non-constrained areas. The problems can be predicted with the detailed task intractable. The real difficulty is to formulate these constraints. The time that the optimization process needs the 50% is to refine the input data and not the calculations. The software knowledge is a base of the optimization, because it is can-not solve every problem. The final geometry can be seen on the *Figure 15* ones from the built-in ones. Further task is to develop the subassembly-kit and the tip-list.

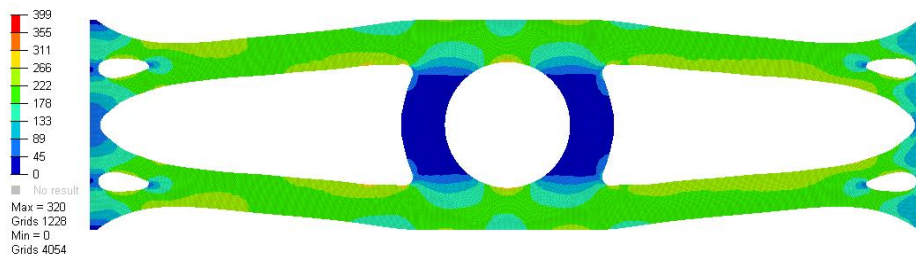


Figure 15. Final geometry stress plot

6. ACKNOWLEDGEMENT

The recent study and publication was realized within the Knorr-Bremse Scholarship Program supported by the Knorr-Bremse Rail Systems Budapest.

7. REFERENCES

- [1] PILKEY, W. D. (2005): *Formulas for Stress, Strain, and Structural Matrices*. 2nd Edition, John Wiley & Sons.
- [2] ARORA, J. (1989): *Introduction to Optimum Design*. McGraw-Hill.
- [3] ERDŐSNÉ SÉLLEY Csilla–GYURECZ György–JANIK József–KÖRTÉLYESI Gábor: *Mérnöki optimalizáció*.

SOLVING MULTIBODY DYNAMICS PROBLEMS USING PYTHON

PAVEL FLORIAN–ROMAN ČERMÁK

University of West Bohemia, Department of Machine Design

Univerzitni 8, 306 14 Pilsen

pflorian@kks.zcu.cz

rcermak@kks.zcu.cz

Abstract: The aim of this paper is inform the reader with possibilities of deriving equations of motion for arbitrarily complex multibody systems using Python programming language. Sympy is a library of functions for Python and a full computer algebra system (CAS) which has among others a built-in feature that allows assembling a multibody system and derive corresponding equations of motion.

Keywords: *multibody dynamics, Python, Sympy, EOM*

1. INTRODUCTION

Solving multibody dynamics problems is possible by hand. There are methods such as Lagrange equations that allow that. However, this approach is only viable for simple systems with limited number of degrees of freedom, as hand-derivation of EOMs is a tedious and error-prone process. Sympy provides an alternative that is under BSD license which means it is free for both academic and commercial use. The initial phase consisting of setting up the environment and gathering required libraries is made simple thanks to Python scientific distributions such as Anaconda that involve all the necessary features. In this paper it is shown how one can obtain equations of motion describing dynamics of a simple four bar mechanism as shown in *Figure 1*.

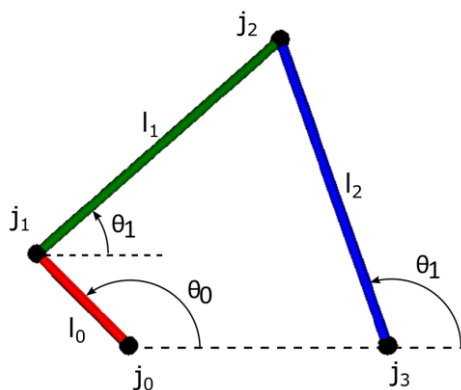


Figure 1. Four bar mechanism

2. GENERAL PROCEDURE WHEN USING SYMPY

When using Sympy it is good practice to follow a certain path. In this case we will start with defining kinematics of the mechanism, followed by inertia, forces, EOM generation and simulation.

2.1. Kinematics

The first step is to decide how many generalized coordinates will be used to describe the mechanism. The four bar mechanism as described here has only one degree of freedom and can be unambiguously described by one angular coordinate. However in this example three generalized coordinates corresponding to each link will be used. Python indexing starts with 0 therefore the same approach will be used when naming variables.

The first three lines in the block of code bellow import all the necessary functions. Time-dependant variables are created using *dynamicsymbols()* command, whereas for time-independent constants *symbols()* is used.

```

from __future__ import print_function, division
from sympy.physics.mechanics import *
from sympy.physics.vector import time_derivative
from sympy import symbols
from numpy import array, linspace, rad2deg, deg2rad
from scipy.integrate import odeint
from pydy.codegen.ode_function_generators import
generate_ode_function

# number of links in the mechanism
n = 3
# generalized speeds and coordinates
theta = dynamicsymbols('theta:{}'.format(n))
theta_d = dynamicsymbols('theta:{}'.format(n), 1)
omega = dynamicsymbols('omega:{}'.format(n))
omega_d = dynamicsymbols('omega:{}'.format(n), 1)
# the extra symbol thanks to n+1 stands for the distance
# between the grounding joints
length_bars = symbols('L_B:{}'.format(n+1))
g = symbols('g') # gravity

```

Figure 2. Definition of variables

In the second step reference frames are defined. An inertial frame is introduced and then three other frames, one for each link, are oriented with respect to the inertial frame. The rotations are given by *theta* angles and their direction is the z-axis of the inertial frame. In a similar fashion angular velocities of the reference frames are introduced. There is more than one way to define rotation between reference frames. For non-planar systems it is easier to use type *Body* since less lines of code are needed for the definition as shown in the block of code for *bar2_frame*.


```

# reference frames
inertial_frame = ReferenceFrame('I')
bar0_frame = inertial_frame.orientnew('B0', 'Axis', (theta[0],
                                                    inertial_frame.z))
bar1_frame = inertial_frame.orientnew('B1', 'Axis', (theta[1],
                                                    inertial_frame.z))
bar2_frame = inertial_frame.orientnew('B2', 'Body', [0, 0, theta[2]],
                                      'XYZ')

# angular velocities
bar0_frame.set_ang_vel(inertial_frame, omega[0]*inertial_frame.z)
bar1_frame.set_ang_vel(inertial_frame, omega[1]*inertial_frame.z)
bar2_frame.set_ang_vel(inertial_frame, omega[2]*inertial_frame.z)

```

Figure 3. Orientation and velocity

Once orientation and velocity of frames is set important points of the system such as joints and centres of mass can be introduced. X-axis of each frame defines link orientation and centres of mass are assumed to be in the middle of each link for simplicity.

```

# joints
joint0 = Point('j0')
joint1 = joint0.locatenew('j1', length_bars[0] * bar0_frame.x)
joint2 = joint1.locatenew('j2', length_bars[1] * bar1_frame.x)
joint3 = joint2.locatenew('j3', -length_bars[2] * bar2_frame.x)

# cm positions
bar0_cm = joint0.locatenew('b0cm', length_bars[0]/2 * bar0_frame.x)
bar1_cm = joint1.locatenew('b1cm', length_bars[1]/2 * bar1_frame.x)
bar2_cm = joint3.locatenew('b2cm', length_bars[2]/2 * bar2_frame.x)

```

Figure 4. Joint and centre of mass locations

Just like angular velocities were defined separately linear velocities of the points need to be defined in a similar way. Velocity of the grounding joints $j0$ and $j3$ is set to 0 as they are stationary.

```

# velocity of the grounding joints is 0
joint0.set_vel(inertial_frame, 0)
joint3.set_vel(inertial_frame, 0)

# velocity if the remaining joints
joint1.v2pt_theory(joint0, inertial_frame, bar0_frame)
joint2.v2pt_theory(joint1, inertial_frame, bar1_frame)

# velocity if centres of mass
bar0_cm.v2pt_theory(joint0, inertial_frame, bar0_frame)
bar1_cm.v2pt_theory(joint1, inertial_frame, bar1_frame)
bar2_cm.v2pt_theory(joint2, inertial_frame, bar2_frame)

```

Figure 5. Velocities of points

When defining kinematics of the mechanism the last step is to introduce configuration and velocity constraints. As mentioned above the mechanism has only one degree of freedom, but three angular coordinates were used to define it. In this case we need to implement two configuration constraints, one for each excessive coordinate, and two velocity constraints in order to obtain the correct solution. The configuration constraints say that the joint $j3$ is located at distance of $length_bars[3]$ from joint $j0$ on the x-axis of the inertial frame. There are two constraints, one for each x and y direction. The velocity constraints simply mean that these two points are not moving with respect to each other. The following block of code concludes the definition of kinematics.

```
# configuration constraint
zero = joint3.pos_from(joint0) + length_bars[3] * inertial_frame.x
f_c = [zero & inertial_frame.x, zero & inertial_frame.y]
# velocity constraint
dzero = time_derivative(zero, inertial_frame)
f_v = [dzero & inertial_frame.x, dzero & inertial_frame.y]
```

Figure 6. Constraints

2.2. Mass and inertia

This particular mechanism is only a planar problem therefore the definition of inertia is fairly simple. In this case we only introduce rotational inertia related to the xy plane of the inertial frame. In Sympy inertia is implemented in terms of dyadic. Inertia is defined in a frame that is stationary with respect to a given body. The information about mass and inertia is then coupled in a *RigidBody* object.

```
# mass and inertia
mass_bars = symbols('m_B:{}'.format(n))
inertia_bars = symbols('I_B:{}'.format(n))
# inertia(frame, ix=0, iyy=0, izz, ixy=0, iyz=0, izx=0)
bar0_indyad = inertia(bar0_frame, 0, 0, inertia_bars[0])
bar1_indyad = inertia(bar1_frame, 0, 0, inertia_bars[1])
bar2_indyad = inertia(bar2_frame, 0, 0, inertia_bars[2])
bar0_inertia = (bar0_indyad, bar0_cm)
bar1_inertia = (bar1_indyad, bar1_cm)
bar2_inertia = (bar2_indyad, bar2_cm)
# bodies
bar0 = RigidBody('Bar 0', bar0_cm, bar0_frame, mass_bars[0],
                bar0_inertia)
bar1 = RigidBody('Bar 1', bar1_cm, bar0_frame, mass_bars[1],
                bar1_inertia)
bar2 = RigidBody('Bar 1', bar2_cm, bar0_frame, mass_bars[2],
                bar2_inertia)
bodies = [bar0, bar1, bar2]
```

Figure 7. Mass and inertia

2.3. Forces

In this case the only applied forces are the ones due to gravity. In order to introduce for example driving torque an extra tuple needs to be created and added to the *loads* list.

```
# forces
bar0_force = (bar0_cm, -mass_bars[0] * g * inertial_frame.y)
bar1_force = (bar1_cm, -mass_bars[1] * g * inertial_frame.y)
bar2_force = (bar2_cm, -mass_bars[2] * g * inertial_frame.y)
loads = [bar0_force, bar1_force, bar2_force]
```

Figure 8. Loads

2.4. EOM generation

Once kinematics, inertia and forces are specified the following step is generation of equations of motion. There are two methods available is Sympy. In this paper Kane's method is used [2]; however it is also possible to utilize Lagrange's method. Firstly we need to introduce kinematic differential equations that bind variables for coordinates with speeds. After that Kane's method is initialized. This method has many arguments as shown below. Lastly Fr and Fr^* need to be calculated as described in [2].

```
# kinematic differential equations
KDE = [theta_d[0] - omega[0], theta_d[1] - omega[1],
       theta_d[2] - omega[2]]
# Kanes Method
kane = KanesMethod(inertial_frame, q_ind=[theta[0]],
                  u_ind=[omega[0]],
                  q_dependent=[theta[1], theta[2]],
                  u_dependent=[omega[1], omega[2]],
                  configuration_constraints=f_c,
                  velocity_constraints=f_v,
                  kd_eqs=KDE)
fr, frstar = kane.kanes_equations(bodies, loads)
```

Figure 9. Kane's method

After running the code above mass matrix of the system can be finally shown running *kane.mas_matrix_full* command. Unfortunately despite the mechanism being fairly simple the output matrix is too large to be included in this paper. In this case the shape of the mass matrix is 6 by 6 since we used three coordinates to describe the mechanism and it is already converted to a set of first-order differential equations.

2.5. Simulation

In order to proceed with integration of equations of motion we need generate the right hand side. Then numerical integration methods can be used to solve such equation:

$$\mathbf{g} = \mathbf{M}^{-1}(\mathbf{x}, t)\mathbf{f}(\mathbf{x}, t), \quad (1)$$

where $\mathbf{M}^{-1}(\mathbf{x}, t)$ is the inverted mass matrix and $\mathbf{f}(\mathbf{x}, t)$ stands for the forcing vector. Firstly we need to group all constants into a list and pass it into right hand side generator function. Then vector of initial conditions needs to be created. After we define time step and length of simulation the integration itself might begin.

```
# list of constants
constants = [g,
             mass_bars[0],
             mass_bars[1],
             mass_bars[2],
             length_bars[0],
             length_bars[1],
             length_bars[2],
             length_bars[3],
             inertia_bars[0],
             inertia_bars[1],
             inertia_bars[2]]
kdd = kane.kindiffdict()
mass_matrix = kane.mass_matrix_full.subs(kdd)
forcing_vector = kane.forcing_full.subs(kdd)
right_hand_side = generate_ode_function(forcing_vector,
                                       theta, omega, constants, mass_matrix=mass_matrix)
# list of numerical values
numerical_constants = [9.81,
                      2.0,
                      5.0,
                      4.0,
                      2.0,
                      5.0,
                      5.0,
                      4.0,
                      1.0,
                      1.0,
                      1.0]
# initial conditions
x0 = array([deg2rad(135), deg2rad(41.3340), deg2rad(109.3884), 0.0,
           0.0, 0.0])
# timeframe
frames_per_sec = 100
final_time = 20.0
t = linspace(0.0, final_time, final_time * frames_per_sec)
# integration
y = odeint(right_hand_side, x0, t,
           args=(dict(zip(constants, numerical_constants)),))
```

Figure 10. Simulation

3. RESULTS

In *Figure 11* results for angular coordinates are shown. The plots were created using *matplotlib* library but the code itself to do that is not included here. There are many examples of usage available on the internet.

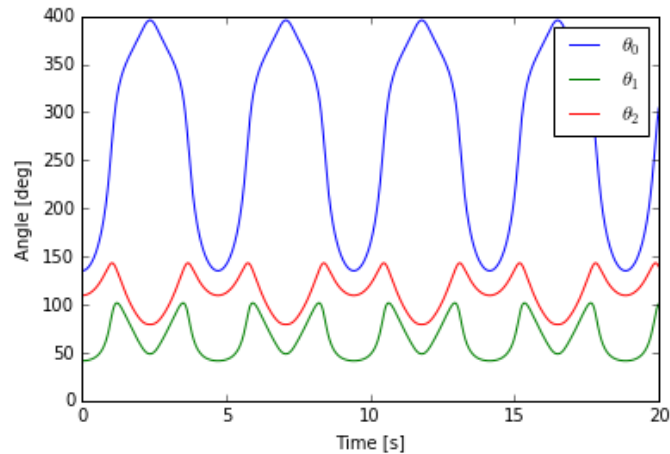


Figure 11. Results – angular coordinates

From the plot above the reader can see that angle of the first link θ_0 oscillates between 135 and 395 degrees. It means that it never completes a full loop. Angular velocities can be plotted in a similar way as shown in *Figure 12*.

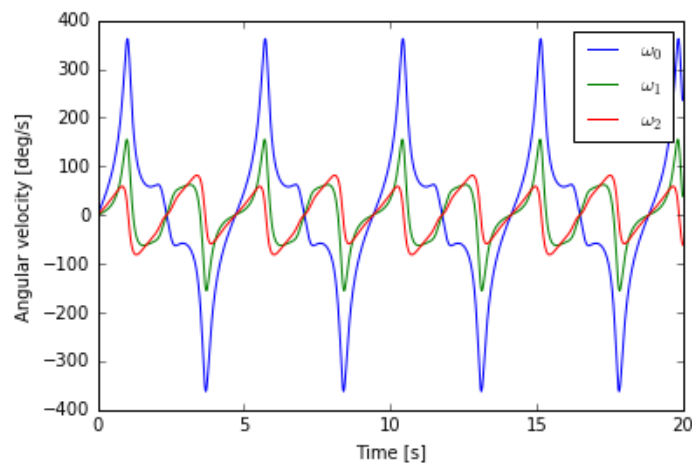


Figure 12. Results – angular velocities

Thanks to another Python library PyDy it is possible to create 3D animations based on simulation results easily. However the code to do that is not included in this paper.

4. CONCLUSION

This paper showed an automated method that allows assembling an arbitrary multibody system and extracting its equations of motion. Such functionality is possible thanks to SymPy which is a free library for Python. This library is included in Anaconda distribution. It is actually possible to copy the whole code shown in this paper into a Python editor to obtain the same results. The usage of SymPy is rather straightforward when dealing with open loop mechanisms and when having the same number of coordinates as there are degrees of freedom. Closed loop mechanisms require introducing constraint equations that need to be correctly defined. The user must have a good understanding of the problem.

5. ACKNOWLEDGEMENT

The research work shown here was made possible by SGS-2016-012.

6. REFERENCES

- [1] MEURER, A.–SMITH, C. P.–PAPROCKI, M.–ČERTÍK, O.–KIRPICHEV, S. B.–ROCKLIN, M.–KUMAR, A.–IVANOV, S.–MOORE, J. K.–SINGH, S.–RATHNAYAKE, T.–VIG, S.–GRANGER, B. E.–MULLER, R. P.–BONAZZI, F.–GUPTA, H.–VATS, S.–JOHANSSON, F.–PEDREGOSA, F.–CURRY, M. J.–TERREL, A. R.–ROUČKA, Š.–SABOO, A.–FERNANDO, I.–KULAL, S.–CIMRMAN, R.–SCOPATZ, A.: SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3:e103, <https://doi.org/10.7717/peerj-cs.103>, 2017
- [2] KANE, T. R.–LEVINSON, D. A.: *Dynamics: Theory and Applications*. McGraw Hill, New York, NY, 1985.
- [3] GEDE, G.–PETERSON, D. L.–NANJANGUD, A. S.–MOORE, J. K.–HUBBARD, M.: Constrained multibody dynamics with Python: From symbolic equation generation to publication. *Proceedings of ASME 2013 IDETC/CIE 2013*, Portland, USA, DOI 10.1115/DETC2013-13470
- [4] *Sympy documentation*. <https://docs.sympy.org>.

USE OF SURFACE RESPONSE METHODOLOGY (RSM) IN CLINCHING PROCESS

SZABOLCS JÓNÁS–MIKLÓS TISZA

Knorr-Bremse Rail Systems Hungary, University of Miskolc
1238, Budapest, Helsinki str. 105, 3515, Miskolc-Egyetemváros
szabolcs.jonas@knorr-bremse.com

Abstract: This study deals with the RSM by clinch joints especially the clinching of high strength steel type DP600. The goal is to find a relationship between the thickness of the clinched joints and the undercut or interlock (C-value). The joining procedure was made by a round TOX tool. The DP600 steels are dual phase (ferritic-martensitic) types (AHSS – advanced high strength steels). For the C-values non-linear finite element simulations were performed in the ANSYS system.

Keywords: *clinch joint, RSM, FEA*

1. INTRODUCTION

The response surface methodology (RSM) is a collection of statistical and mathematical techniques useful for different types of processes (improving, optimizing, and developing). The RSM also has important applications in the design of new products. The most extensive applications of RSM are in the industrial world where several parameters potentially affect the results. The characteristic of the results is called the response. It is typically measured on a continuous scale. Most real world applications of RSM will involve more than one response. The parameters (or input variables) are sometimes called independent variables. The graphical representation of the fitted surface to the results has led to the term response surface methodology.

These joints are used mostly in automotive, computer and aircraft industries, but for instance according to the standards not allowed to use in food industry. This article is the first results of this research programme. The goal of the research is to determine the optimal parameters of clinched joints of high strength steels and aluminium. Furthermore to find a solution for fatigue evaluation and to replace some welded joints to clinched joints because of the cost of clinched joints are lower. The most difficult goal is to use the lowest number of tests and use the articles and other available material and test data to determine the relevant parameters. The clinch joints are quite new types of joints, the first patent was accepted in 1989. This joint can be done between 2–3 thin sheet plates. The material of the plates can be ferrous or non-ferrous, so this joint can realize dissimilar joints without any added material (weld material or glue). The joint made by metal plastic forming by a special tool.

After the patent the increasing industrial needs of this type of joints led the researchers to analyse the joint much more deeply. Several studies carried out the geometry optimization of the clinching tool to get better joints by different optimization methods. Other studies were carried out on the so-called hybrid joints. These joints have an adhesive layer between the sheets. These joints have higher strength but need much more time because the adhesive layer's drying is a time-consuming process [1], [2].

In this study the interlock was analysed. The analyses carry out on the field of the standard sheet plate sizes. FE simulations were performed in ANSYS to determine the different interlocks. From the result the size of the interlock will easily calculate and it is a basis for further analysis (e.g. prediction of pull out strength).

2. RMS MATHEMATICAL BACKGROUND

In general, a process involving a response y , that depends on the input parameters $\xi_1, \xi_2, \dots, \xi_n$. The relationship can be written as it follows:

$$y = f(\xi_1, \xi_2, \dots, \xi_n) + \varepsilon \quad (1)$$

where the response function is f and the term of ε is represent other sources of variability not accounted for in f (e.g. measurement error, etc. assume that its mean is zero):

$$E(y) \equiv \eta = E[f(\xi_1, \xi_2, \dots, \xi_n)] + E[\varepsilon] = f(\xi_1, \xi_2, \dots, \xi_n). \quad (2)$$

In much work it is a convenient to transform these variable to the so-called coded variables x_1, x_2, \dots, x_n . The coded variables are usually dimensionless with mean zero and standard deviation. It can be written as

$$\eta = f(x_1, x_2, \dots, x_n). \quad (3)$$

The f is unknown there an approximation is needed. When the small zone of the independent variable space is appropriate a low-order polynomial (first or second order is used). For the case of two independent variables the first-order model in terms of the coded variables is

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2. \quad (4)$$

If there is an interaction between the parameters, use the main effects model which can be written as

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2. \quad (5)$$

Sometimes the second-order model is needed to define the response function. The following equation will describe that with interaction between the variables

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_1 x_1^2 + \beta_2 x_2^2 + \beta_{12} x_1 x_2. \quad (6)$$

The second-order model has the ability for the easy estimation of the β values and flexible [9].

3. CLINCHING RESULTS

In this study DP600 type of advanced high strength steel was tested. The clinching tool was set up in an MTS servo-hydraulic testing machine. The maximum permitted load on the tool is 50kN. The set up can be seen in *Figure 1*. The specimens were pre-drilled for this application. Two holes were drilled which centralized the specimens on the one hand and prevented them moving on the other hand.

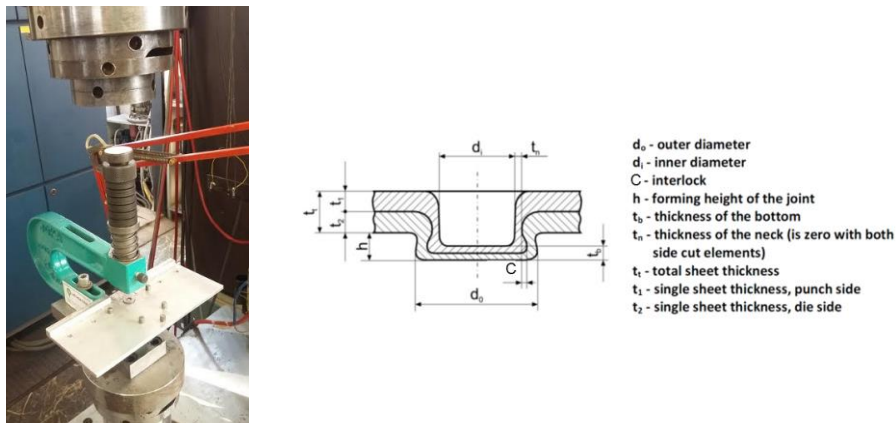


Figure 1. Clinching tool (TOX) and the relevant geometrical parameters of the joint [8]

After the measurement a microscopic investigation was done. The grinded section can be seen in *Figure 2*. The extended grains can be observed due to the forming procedure. The undercut can be also seen on the section. This curve between the formed sheets will be used to compare different simulated cases.

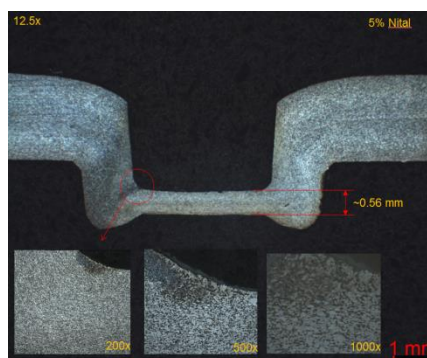


Figure 2. Microscopic investigation – cross section of the specimen after clinching

The FE simulation model was built in ANSYS WB 17.2 [3]. A 2D axisymmetric model presented below (*Figure 4*). The tools were taking account as linear elastic materials. The sheets have multilinear isotropic hardening material model. The anisotropic behaviour of the sheets was neglected. The material model can be seen in *Figure 3*. The material law is fitted to the measured flow curve. The contact zones and the sheets also have a high mesh density. The boundary conditions applied to the model can be seen in *Figure 4*. The tools prescribed in all degrees of freedom on their sides and the punching tool is displacement controlled. It has got a -2.8 mm displacement in the vertical direction till the end of the simulation. Between the parts Augmented Lagrange contact behaviour was applied. The frictional coefficient between the sheets is 0.2 and for other contacts 0.01 values were taken into consideration. The contact stiffness behaviour was updated in each iteration.

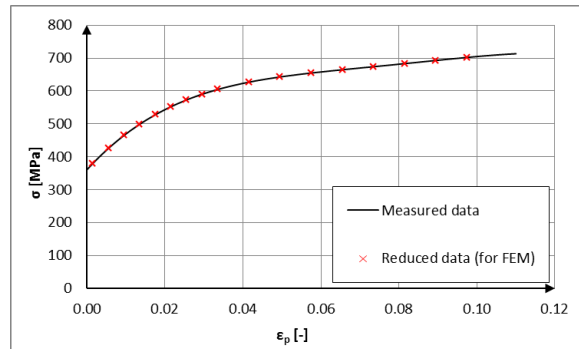


Figure 3. DP600 true stress-true plastic strain curve

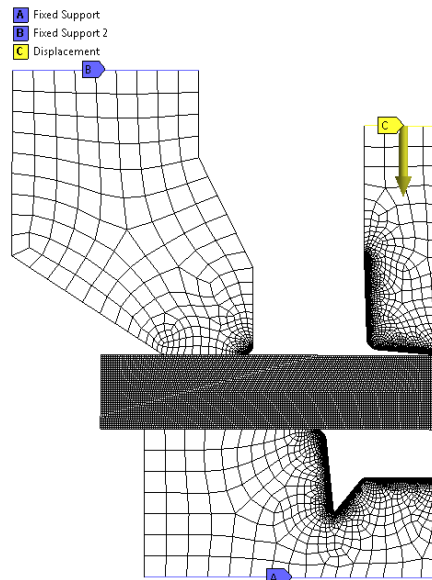


Figure 4. Boundary conditions of the model

4. RESULTS

The results of the simulations can be seen in *Figure 8*. The figure shows the equivalent plastic strain distribution of the sheets. The most affected zones are the neck region of the upper sheet where the punching tool is contacted with it and between the two sheets. *Figure 5* shows the measured and simulated sections of the sheets. The results show good agreement; we can say that the FE model is valid. With this model further simulations can be made to analyse the behaviour of the sheets with different thickness.

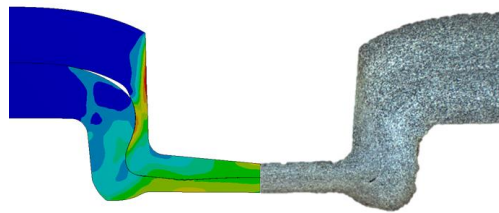


Figure 5. Comparison of the measured and simulated joints

4.1. Application of the RSM in clinching process

The variable filed was the standard deviation of the sheet thicknesses for cold rolled steel sheets (± 0.07 mm). The interlock was the unknown parameter. The “C” value is the difference (see *Figure 6*) of the maximum and the minimum value of the interlock.

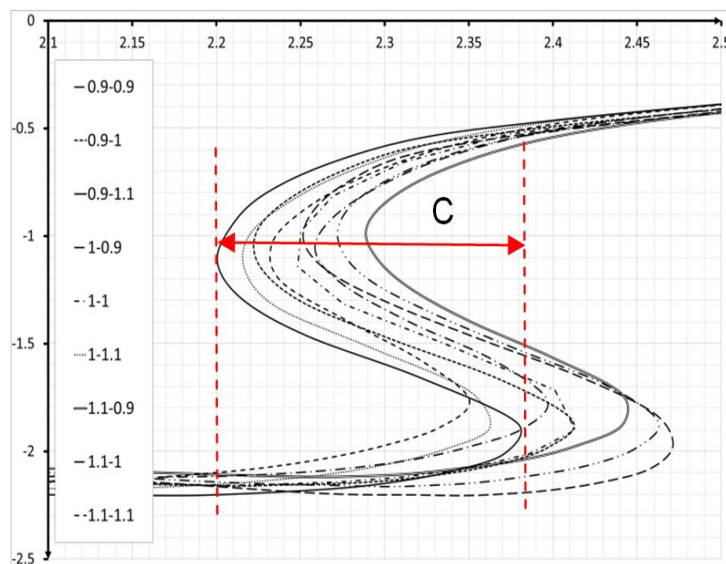


Figure 6. Simulated interlocks

From the simulation the following C values derived (*Table 1*).

Table 1. Variables

$t_{\text{upper}} (X_1)$	$t_{\text{lower}} (X_2)$	C
0.9	0.9	0.22
0.9	1	0.19
0.9	1.1	0.18
1	0.9	0.2
1	1	0.17
1	1.1	0.15
1.1	0.9	0.16
1.1	1	0.14
1.1	1.1	0.12

The f surface was fitted to the calculated values. The *Figure 7* shows the data points and the fitted surface. The vertical axle of it is the values of the C parameters; the other two axles are the thicknesses.

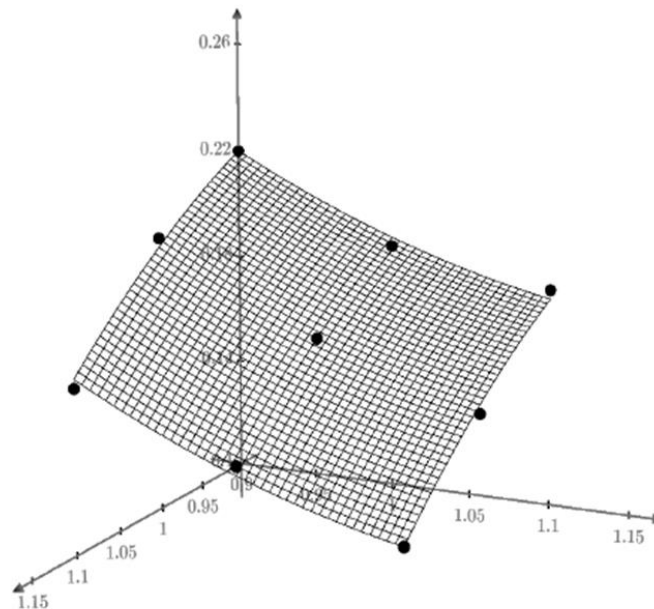


Figure 7. Fitted response surface with design points

A better view of the surface can be seen on the *Figure 8*. The colours show the effect of the parameters.

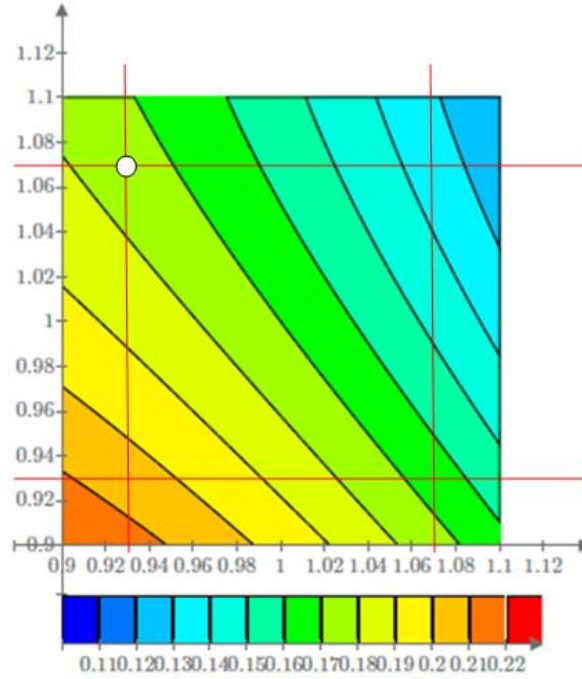


Figure 8. Contour plot of the response surface (red squared area is the true size of the acceptable thicknesses)

The fitted surface's equation is a second order polynomial which can be written as

$$C_{prediction} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2, \quad (7)$$

where the x_1 is the thickness of the lower sheet, x_2 is the thickness of the upper sheet. The mixed member (β_3 – see *Table 3*) can be neglected due its low value. The values of the parameters:

Table 2. Variables

β_0	β_1	β_2	β_3	β_4	β_5
0.67	0.717	-1.217	$-3.013 \cdot 10^{-15}$	-0.5	0.5

In absolute value the β_2 is almost the double of the β_1 . From the *Eq. (7)* the predicted C value for $X_1 = 0.93$ mm and $X_2 = 1.07$ mm is $C_{prediction} = 0.175$ mm (white point in *Figure 8*). The simulated value for C is $C_{Ansys} = 0.178$ mm. This can be seen in *Figure 9*. In opposite case the ($X_1 = 1.07$ mm and $X_2 = 0.93$ mm) the predicted value is $C_{prediction} = 0.165$ mm.

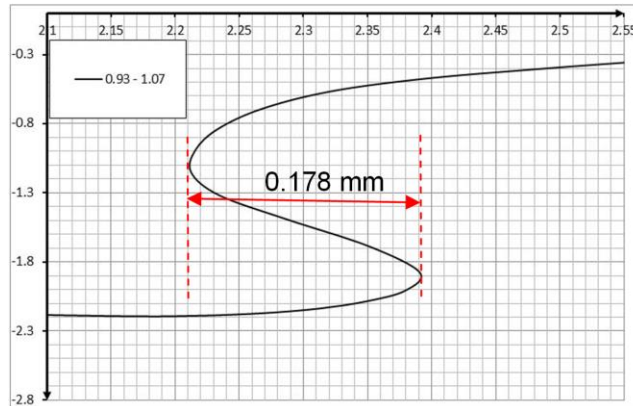


Figure 9. Undercut line of the re-checking dimensions

5. SUMMARY AND CONCLUSION

The FE calculations and the prediction model have been developed for thicknesses in case of DP600 type of steel. It can be observed that the thickness of the upper sheet has a greater effect on the C-value, which leads a higher undercut in the result and a higher pull out strength and finally a better joint.

The results show a very good agreement between the predicted and the calculated values. The deviance is ~2% which is acceptable. The RSM method is useful to determine unknown variables and to find relationships between distinct parameters. Further analyses are needed to find the relationships and the affecters for a better joint design.

6. REFERENCES

- [1] T. SADOWSKI–T. BALAWENDER–P. GOLEWSKI: *Technological aspects of manufacturing and numerical modelling of clinch-adhesive joints*. Springer, 2015.
- [2] L. KASCÁK–E. SPISÁK: Clinching as a non-standard method for joining materials of dissimilar properties. *Zeszyty naukowe politechniki rzeszowskiej*, Nr. 284, *Mechanika*, z. 84 2012.
- [3] ANSYS WB 17.2 User's guide.
- [4] P. Z. KOVÁCS–M. TISZA: Investigation and Finite element modelling of technological parameters of clinched joints. *Miskolci Egyetemi Közlemények*, Miskolc, X. kötet, 2015.
- [5] M. TISZA–G. GÁL–A. KISS–Z. P. KOVÁCS–ZS. LUKÁCS: Alakítható nagyszilárdságú anyagok klincs kötése. *Multidiszciplináris tudományok*, 4. kötet, 2014, 1. sz., 49–58.

- [6] P. Z. KOVÁCS–M. TISZA: Klincs kötés technológiai paramétereinek vizsgálata, végelelemes modellezése. *Anyagmérnöki Tudományok*, 39/1, 2016, 7–18.
- [7] Chan-Joo LEE, Jae Young KIM, Sang-Kon LEE, Dae-Cheol KO, Byung-Min KIM: Parametric study on mechanical clinching process for joining aluminum alloy and high-strength steel sheets. *Journal of Mechanical Science and Technology*, 24, 2010, 123–126.
- [8] W. G. DROSSEL–M. ISRAEL–T. FALK: *Robustness evaluation and tool optimization on forming applications*. 9th Weimar Optimization and Stochastic Days, 2012.
- [9] R. H. MYERS–D. C. MONTGOMERY–C. M. ANDERSON-COOK: *Response surface methodology – Process and Product Optimization Using Designed Experiments*. 3rd edition, 2009, Wiley.

MESH INFLUENCE ON CONTACT STRESS RESULTS OF TEETH SPUR GEARS

MARTIN KRATOCHVIL

University of West Bohemia, Department of Machine Design
Univerzitni 2732/8, 301 00 Pilsen, Czech Republic
kratochv@kks.zcu.cz

Abstract: Meshing is very important part of finite element analysis, because it determines parameters of numerical solution of a physics phenomenon. There are many options as type of element, its size, shape or number of nodes. Every physical phenomenon needs different discretization corresponding to equation or set of equations. This paper deals with the influence of mesh settings on results of contact stress of spur gear teeth.

Keywords: *spur gears, contact stress, mesh influence, Ansys*

1. INTRODUCTION

A geometry of gear teeth was created in NX11 CAD system and exported to .stp file. Then the geometry was imported to Ansys Design Modeller. All calculations were done only on one pair of mating teeth (not a complete geometry of a gear). Contact force, geometry and constrains are the same in all cases.

Gear tooth parameters:

- Spur gear, tooth profile according CSN 01 4607 [1]
- Module 6 mm
- Width 20 mm
- Number of teeth 20 (gear 1) and 31 (gear 2)

Material of both gears is Structural Steel from Ansys Engineering Data library. Contact type is Frictionless with Stabilization Damping Factor Value 1.

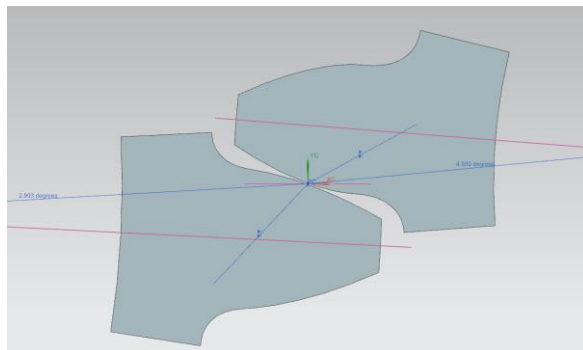


Figure 1. Geometry of teeth (NX11 CAD system)

Meshes with element size lower than 0.5 mm were created on a divided tooth geometry (Model 2). The divided tooth geometry was created in SpaceClaim (a part of Ansys bundle). The aim of divided geometry is to mesh a part of tooth geometry by smaller element size. Smaller element size is used only in small space around contact line. This procedure was chosen to shorten the calculation time. The geometry is preserved, because the nodes of edge elements mate. Mesh connection is set by a function *Form New Part*. A middle geometry is generated as multizone mesh type.

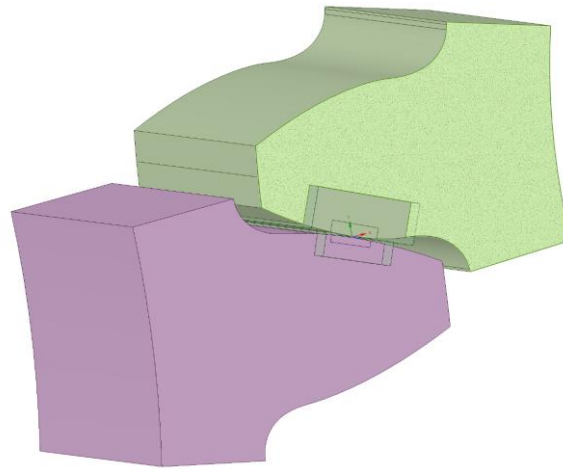


Figure 2. Divided Model 2 geometry in SpaceClaim

Path orientation, boundary conditions and mesh connection are evident from the figures bellow.

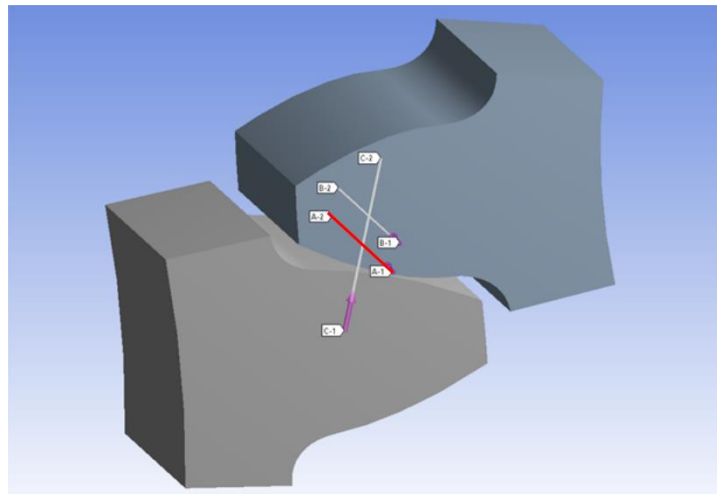


Figure 3. Path-1 orientation (marked by red line)

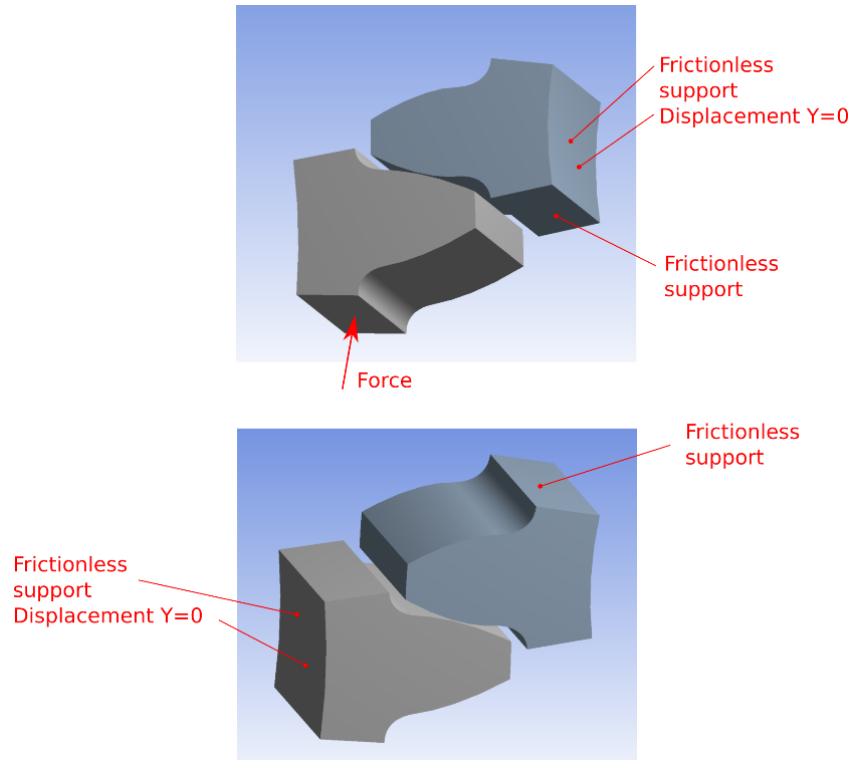


Figure 4. FEM constrains

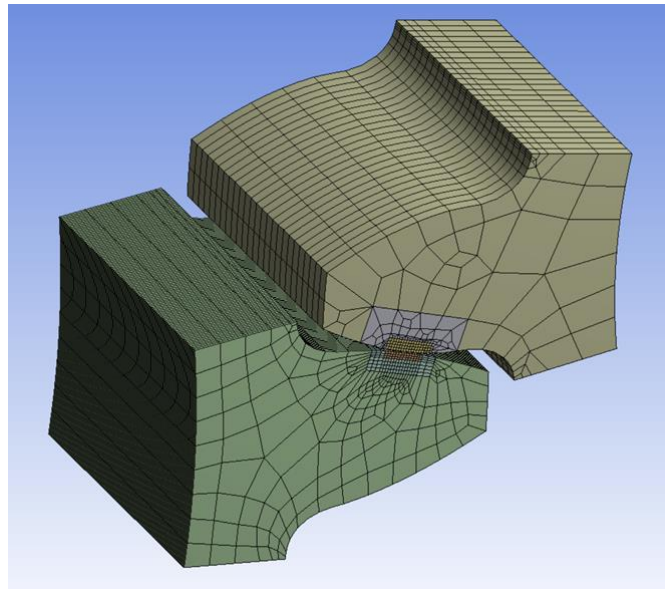


Figure 5. Example of mesh connection of divided volumes (Model 2)

2. EXAMPLES

This part of the paper describes several examples with different mesh settings and its parameters. These options are described in tables. If midside node option isn't specified, then midside nodes are used (kept option).

2.1. Hexahedron vs Tetrahedron, element size estimation (Example 1)

The first calculation was made to determine an optimal size of finite element. Both calculations have been made on simple model with different size of element. The table below describes element type and its size.

*Table 1
Example 1 settings*

Result	Type of model	Element shape	Element size
tetra 2 mm	Model 1	Tetra	2 mm
tetra 0.5 mm	Model 1	Tetra	0.5 mm
hex 2 mm	Model 1	Hex	2 mm
hex 0.5 mm	Model 1	Hex	0.5 mm

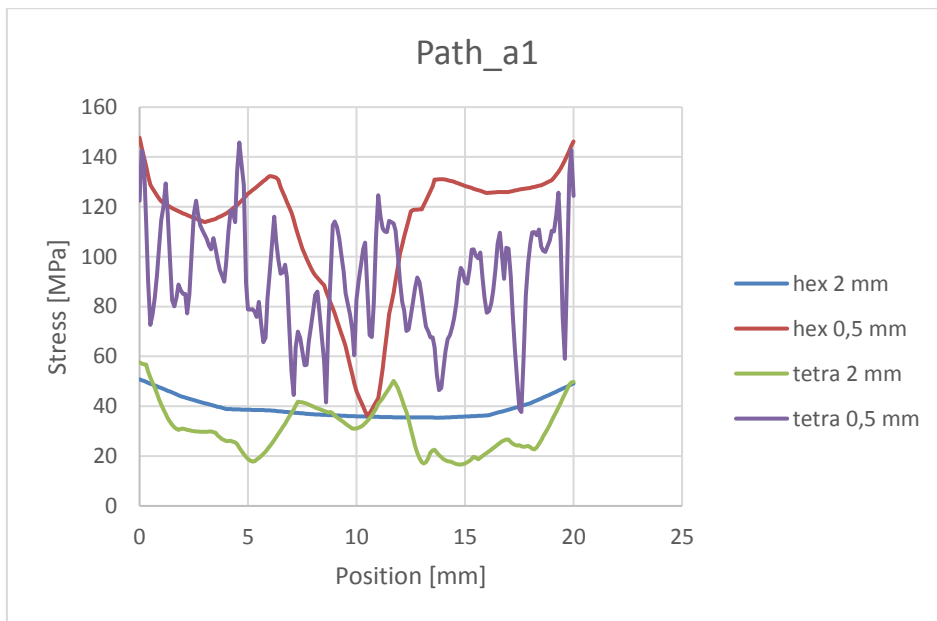


Figure 6. Stress result of Example 1

The graph shows contact stress result on the contact line. It is clear that both element sizes provide improper results. Stress value of 2 mm elements is probably low. It is assumed that smaller elements provide more accurate result. In this case, smaller 0.5 mm tetrahedron elements have higher stress, but large range of values. Values calculated on hexahedron elements are relatively smooth but there is a big

gap in the middle of tooth width. After evaluating of these results, it is necessary to reduce the size of the finite elements around the contact line.

2.2. Hexahedron vs Tetrahedron, suitable element size (Example 2)

The aim of second calculation is to compare 0.5 mm and 0.2 mm size of finite element. Lower element size increases number of total finite element, so it is necessary to optimize the mesh. The mesh optimization is based on the Model 2, which consists of three divided volumes. Volumes in contact have 0.2 mm element size.

Table 2
Example 2 settings

Result	Type of model	Element shape	Element size
tetra 0.5 mm	Model 1	Tetra	0.5 mm
hex 0.5 mm	Model 1	Hex	0.5 mm
tetra 0.2 mm	Model 2	Tetra	0.2 mm
hex 0.2 mm	Model 2	Hex	0.2 mm

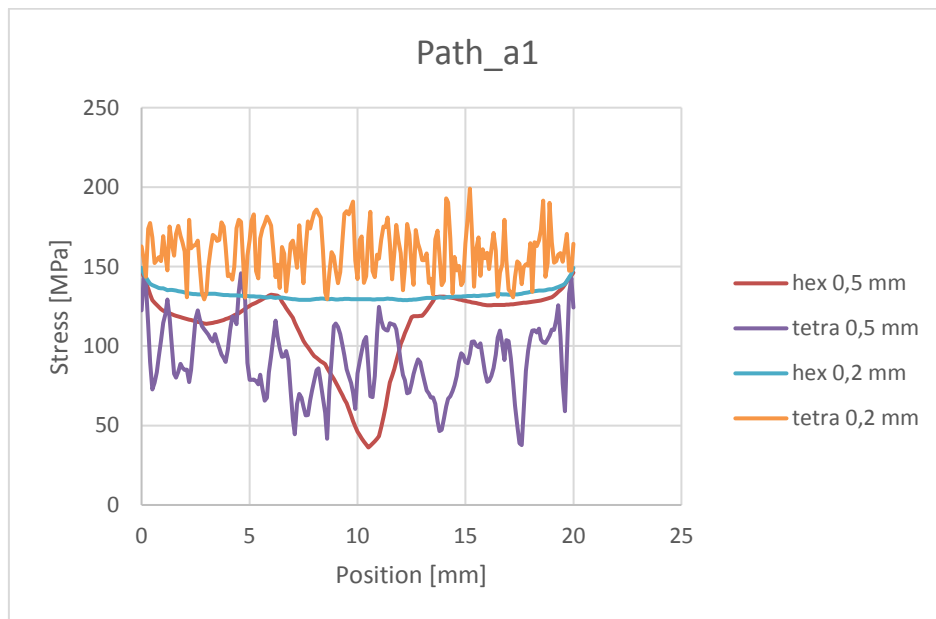


Figure 7. Stress result of Example 2

The element size of 0.2 mm seems to provide better result than 0.5 mm or 2 mm. The difference between higher and lower stress of result of tetrahedron elements is smaller than 0.5 mm element size. In the case of 0.2 mm hexahedron element, the gap mentioned in previous analysis disappeared. The stress calculated on hexahedron element is evenly distributed, which corresponds to a static load and a precisely made tooth (nominal dimensions without tolerances). Tetrahedron elements have

still relative high range of values, but average contact stress as a result from tetrahedron mesh could be calculated.

2.3. Influence of midsize nodes (Example 3)

This calculation explains the influence of midside nodes. Midside nodes have certain importance in some cases, where use of linear elements requires a much larger number of elements to get appropriate result. There is the same element size to demonstrate the difference.

Table 3
Example 3 settings

Result	Midsized	Type of model	Element shape	Element size
tetra 0.2 mm dropped	Dropped	Model 2	Tetra	0.2 mm
hex 0.2 mm dropped	Dropped	Model 2	Hex	0.2 mm
tetra 0.2 mm	Kept	Model 2	Tetra	0.2 mm
hex 0.2 mm	Kept	Model 2	Hex	0.2 mm

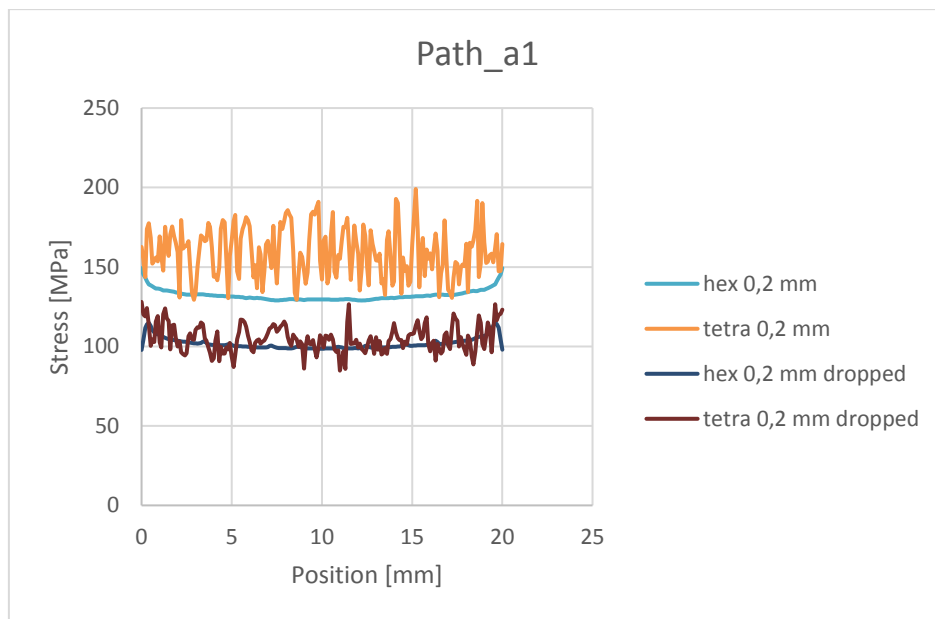


Figure 8. Stress result of Example 3

Stress calculated on both linear hexahedron and linear tetrahedron elements is lower than values calculated on elements with midside nodes. Average values of contact stress on linear elements are almost identical. But the values of contact stress are relatively low, smaller element size of linear elements should be better, but lower size increases number of elements and calculation time. For this reason, the use of linear elements is not appropriate.

2.4. Inflation meshing (Example 4)

Inflation is a method of meshing based on creation of layers, a layer thickness obviously graduates with growing distance from an influence area. Ansys can create inflation based mesh that consists of linear elements (dropped). For comparison, a mesh of hexahedron elements with 0.08 mm (Model 2) size was created.

Table 4
Example 4 settings

Result	Midsized	Type of model	Element shape	Element size
hex 0.2 mm	Kept	Model 2	Hex	0.2 mm
Inflation	Dropped	Model 1	Hex	0.2 mm
Hex 0.08 mm	Kept	Model 2	Hex	0.08 mm

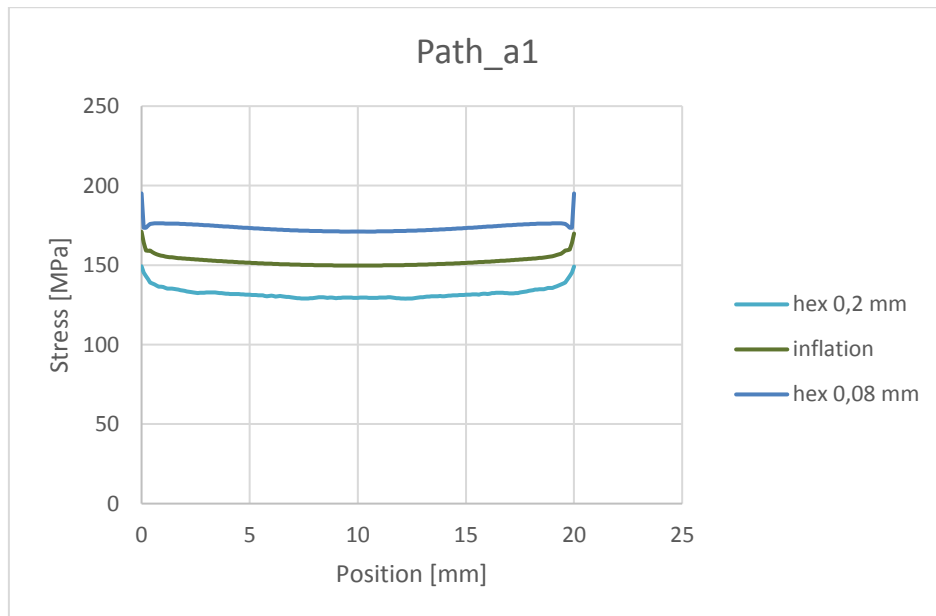


Figure 9. Stress result of Example 4

Lowering the size of element to 0.08 mm increases contact stress. This result is probably more accurate than the mesh with 0.2 mm element size. But theoretically, 0.08 mm size mesh has approximately 15 times more elements than 0.2 mm. In this case, divided volume (Model 2) decreases total number of elements because 0.08 mm size elements are only close to the contact line (2,2 million nodes, 520 000 elements, elapsed time 47 000 seconds). Calculation of inflated mesh was significantly faster (elapsed time 5930 s, 730 005 nodes, 172 500 elements). The fastest was 0.2 mm hexahedron mesh (elapsed time 1174 s, 244 273 nodes, 64 520 elements). Elapsed time of all calculations can be different on another computer. In-

flated mesh seems to be perspective in the case of teeth contact, but further verification is needed.

3. Summary

Results of examples previously described suggest, that too large elements cannot provide proper values of contact stress. Smaller elements with size 0.5 mm lead to more appropriate values, but the optimum is probably 0.2 mm element size or lower. These sizes provide higher stress values and lower range of maximal and minimal contact stress. Calculation with linear elements of the same element size (as dropped) is faster than quadratic elements, but the contact stress is too low. The results suggest that use of linear elements in the case of inflated mesh could make sense, because there is higher stress probably caused by higher number of layers.

4. Acknowledgement

The research work presented in the paper was supported by a project SGS-2016-012.

5. References

- [1] CSN 01 4607. *Gears with involute profile: Basic Profile*. Prague, Office for standardization and measurement, 1978 (in Czech).

REVIEWING COMMITTEE

- Á. DÖBRÖCZÖNI
Institute of Machine and Product Design
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
machda@uni-miskolc.hu
- M. GERGELY
Acceleration Bt.
mihaly_gergely@freemail.hu
- K. JÁRMAI
Institute of Materials Handling and Logistics
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
altjar@uni-miskolc.hu
- I. KERÉKES
Institute of Mechanics
University of Miskolc,
H-3515 Miskolc-Egyetemváros, Hungary
mechker@uni-miskolc.hu
- F. J. SZABÓ
Institute of Machine and Product Design
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
machszf@uni-miskolc.hu
- A. SZILÁGYI
Department of Machine Tools
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
szilagy.attila@uni-miskolc.hu
- J. PÉTER
Institute of Machine and Product Design
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
machpj@uni-miskolc.hu

Secretariat of the Vice-Rector for Research and International Relations,
University of Miskolc,
Responsible for the Publication: Prof. dr. Tamás Kékesi
Published by the Miskolc University Press under leadership of Attila Szendi
Responsible for duplication: Erzsébet Pásztor
Editor: Dr. Ágnes Takács
Technical editor: Csilla Gramantik
Corrector: Zoltán Juhász
Number of copies printed: 50
Put the Press: April 2018
Number of permission: TNRT–2018– 180 –ME
HU ISSN 1785-6892 in print
HU ISSN 2064-7522 online