FINITE ELEMENT METHOD PARAMETERIZATION OF DEEP DRAWING PROCESS USING A CAD-CAE APPLICATION

Alin Pop[1], Ioan Radu[2] and Florin Blaga[3]
[1] University of Oradea, alinpop@yahoo.com
[2] University of Oradea, iradu@uoradea.ro
[3] University of Oradea, fblaga@uoradea.ro

ABSTRACT:Finite element analysis is a time consuming process and cost. Optimization of the process and accuracy measurement is one of the most common objectives in our days. In this paper described an application CAD-CAE which aims to solve the problem of geometry regeneration where obtained measurements through the finite element analysis are not satisfying, and also aims to makes the process of defining boundary conditions and deep drawing forces implementation easier..
KEYWORDS: deep drawing, finite element analysis, abaqus

1. INTRODUCTION

By using the programming interface of the software it can be done the following:

- the automation of repetitive task. For example it can be created algorithms for generate libraries with standard materials or for running certain finite element analysis;
- parametric studies. Using programming interface in Abaqus, the geometry of punch and die can be modify by a simple increment of variable which defines the dimensions and the result can be analyze.
- changes in patterns database;
- accessing data output of finite element analysis.

Such a software can be used to generate geometry models of deep drawing components system. The user interface is an extension of the object-oriented language Python software.[1]

The design and manufacturing process of plastic deformation is supported with CAD tools. Besides the parts and tools design process, in the last 15 years it has been used very frequently tools that allow stimulation of plastic deformation process.

For a correct and efficient simulation of a deep drawing process is recommended to use a model in which the links between the design validation process is continuous.[2]

The objective of virtual planning is to reduce the response time and to increase the accuracy of results. In the first steps of the process is recommended to use the standards. These standards enable product development in a certain pattern.

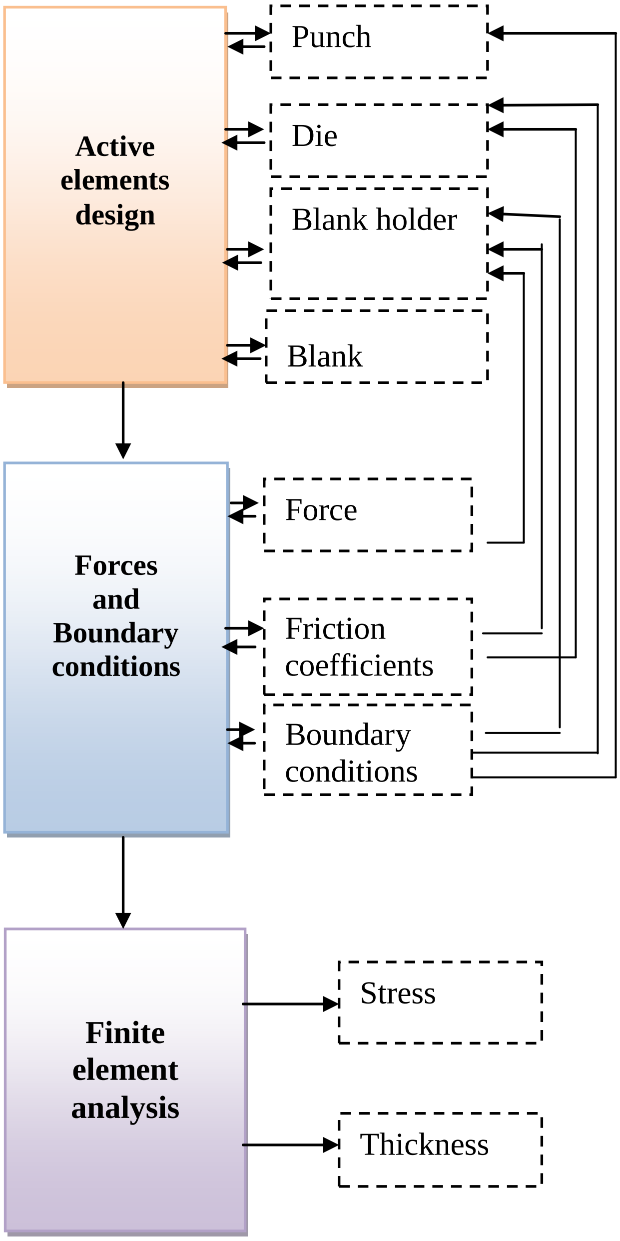

Figure 1. Structure of CAD-CAE application

A design and efficient manufacturing process consists of three main components:

- geometry and technology process standardization;
- fast design geometry instruments proposed;
- efficient CAE tools.

The first component refers to the geometric details which are repeating and which can be defined through a parameters combination. In this way is very easy to build the geometric model whatever parameters it been chose.

The second component is the CAD-CAE application solve the problem of defining conditions in which the deep drawing process take place.

Starting from the number of steps necessary to obtain the deep drawing part, it can be established the force and the boundary conditions which act on the system components.

The surfaces that are in contact and the friction forces have to be defined at this level. The blank holder force and the boundary conditions of deep drawing components will be defined as well.[3]

Finite element analyse is only one step in the CAD-CAE application process and is a very important one. The objective of the simulation process is to predict the defects of blank deformation

The CAD-CAE application consists of several modules divided as follows:

- module for active elements geometry and blank generation
- module for material property definition;
- module for setup the deep drawing boundary conditions and forces.

2. ACTIVE ELEMENTS GEOMETRIC MODULE

The CAD module allows the user to generate the punch, die, blank holder, blank geometry. The CAD module has two chapters, one in which it's defined the user interface and other where the geometric model is generated.[4]

2.1 Punch modelling

The punch is generated by defining values for the three parameters: diameter, length, and radius (Figure 2).

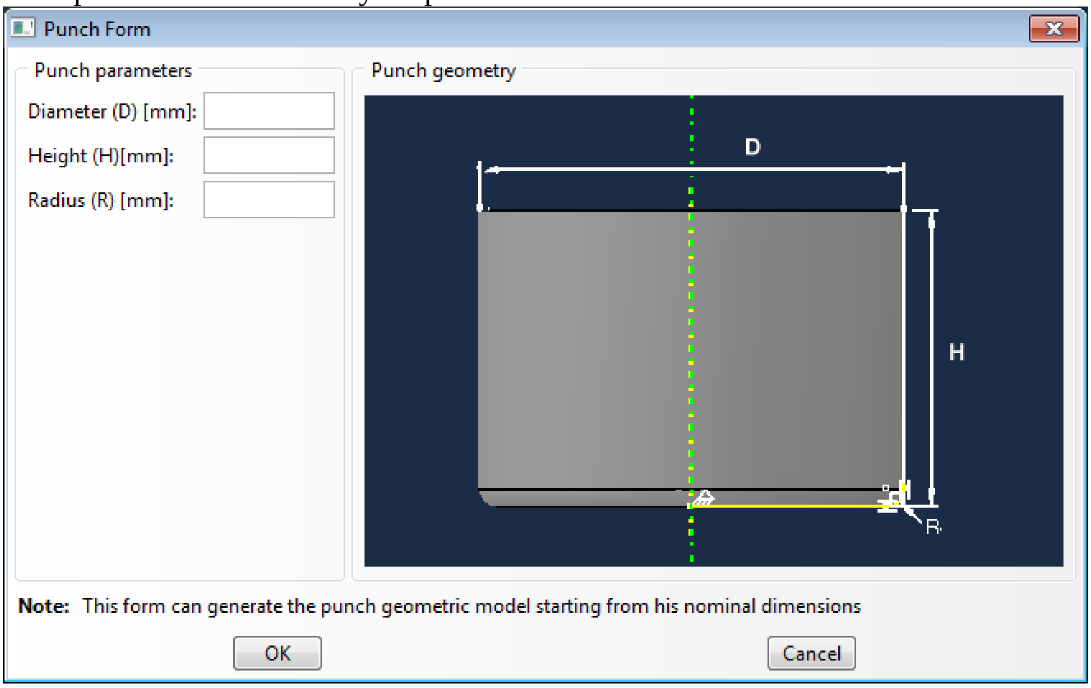

Figure 2. Punch geometric parameters setup

Geometric modules are made using the Python programming language. The user interface for generate the punch geometry consist in the development of a class that contains input parameters variables and buttons definition.

AFXTextField(va, 12, 'Diameter (D) [mm]:', form.widthKw, 0)

AFXTextField(va, 12, 'Height (H)[mm]:', form.heightKw, 0)

AFXTextField(va, 12, 'Radius (R) [mm]:', form.radiusKw, 0)

AFXDataDialog.__init__(self, form, 'Punch form', self.OK|self.CANCEL)

2.2 Die modelling

For the die geometry generation it's necessary to specify the input values for the height, inside diameter, outside diameter, ad die radius. It was made a graphical user interface (Figure 3) for entering the geometric dimensions.

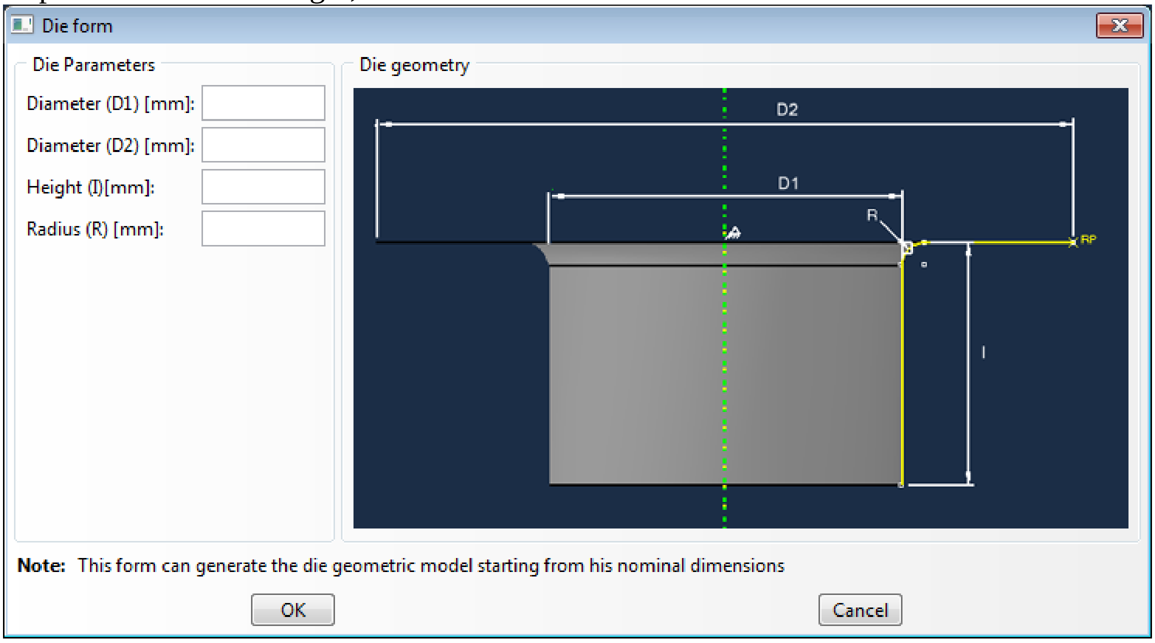

Figure 3. Die geometric parameters setup

For the interface implementation it was created a class, where the geometric input die parameters has to be added. Also is presented a basic sketch of the die geometry.

AFXTextField(va, 12, 'Diameter (D1) [mm]:', form.widthKw_small, 0)

AFXTextField(va, 12, 'Diameter (D2) [mm]:', form.widthKw_big, 0)

AFXTextField(va, 12, 'Height (I)[mm]:', form.heightKw, 0)

AFXTextField(va, 12, 'Radius (R) [mm]:', form.radiusKw, 0)

AFXDataDialog.__init__(self, form, 'Die form', self.OK|self.CANCEL)

2.3 Blank Modelling

The geometry of the blank has only one entry dimension, his diameter. The graphical user interface is shown in Figure

:

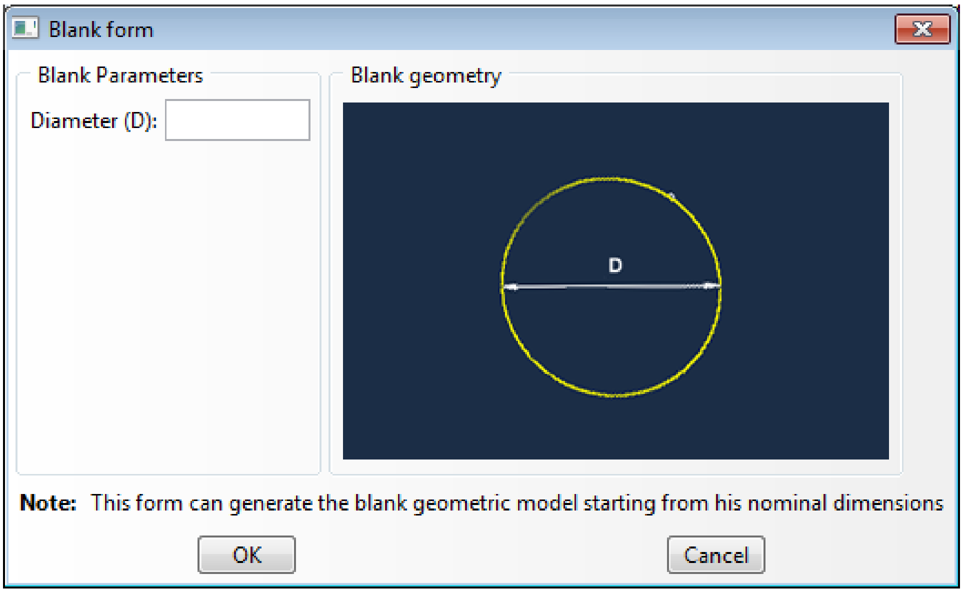

Figure 4. Blank geometric parameters setup

The class used in Python programming language for blank geometry generation is;

class semifabricatDB(AFXDataDialog):

def __init__(self, form):

AFXDataDialog.__init__(self, form, 'Blank form', self.OK|self.CANCEL)

hf = FXHorizontalFrame(self, LAYOUT_FILL_X, 0,0,0,0, 0,0,0,0)

gb = FXGroupBox(hf, 'Blank parameters', LAYOUT_FILL_Y|FRAME_GROOVE)

va = AFXVerticalAligner(gb)

AFXTextField(va, 12, 'Diametru (D):', form.widthKw, 0) *gb = FXGroupBox(hf, Blank geometry, LAYOUT_FILL_Y|FRAME_GROOVE)*

icon=afxCreateIcon(os.path.join(thisDir, 'semifabricat.png'))

FXLabel(gb, '', icon)

2.4 Blank holder modelling

The last module in the CAD application project is generating the geometry blank holder. The input parameters are inside diameter and outside diameters. The user interface is illustrated in **figure 5**

.

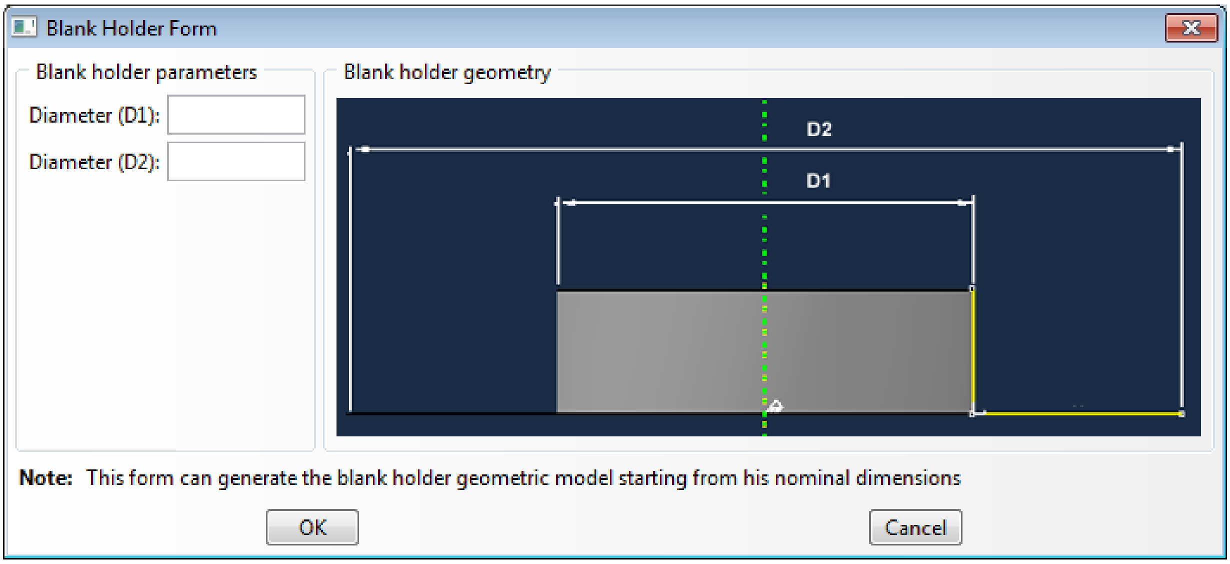

Figure 5. Blank holder geometric parameters setup

The source code for defining the inner diameter and the outer diameter is:

AFXTextField(va, 12, 'Diameter (D1):', form.widthKw_small, 0)

AFXTextField(va, 12, 'Diameter (D2):', form.widthKw_big, 0)

3. MATERIAL BEHAVIORS SETUP MODULE

Defining the properties of the material involves specifying multiple parameters related to its mechanical properties. The graphical interface (Figure 6) will ask the user to input the following parameters: material density, elastic modulus, Poisson's ratio, and points that define the stress-strain relationship.

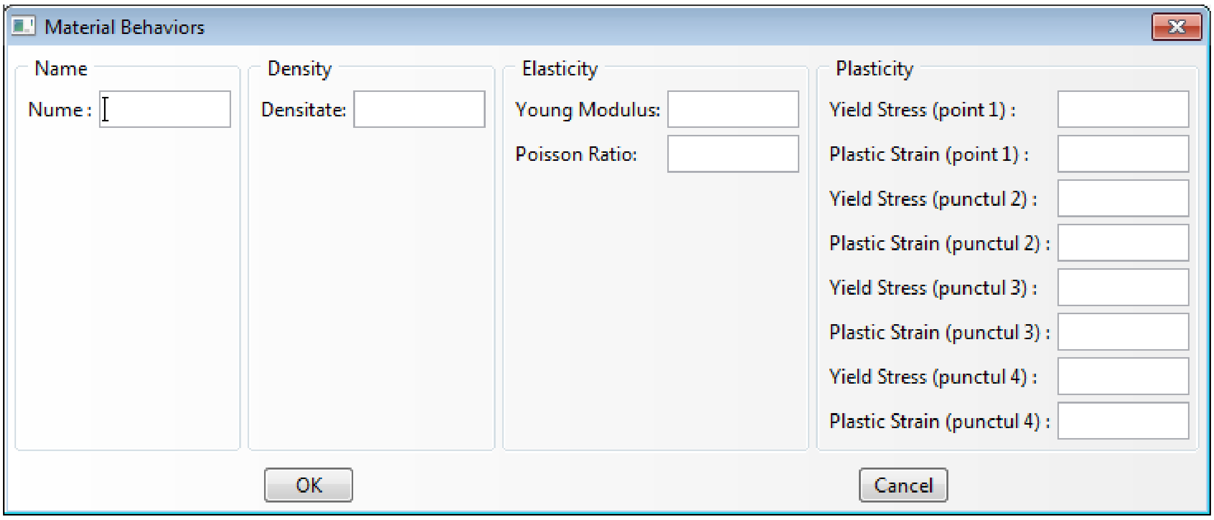

Figure 6. Material parameters setup

For designing this form it has been defined a class. Source code is showed in the following lines:

class loadForm(AFXForm):

def __init__(self, owner):

AFXForm.__init__(self, owner)

self.cmd = AFXGuiCommand(self, 'load', 'action')

self.coefpas = AFXFloatKeyword(self.cmd, 'coefpas', TRUE)

self.coefprs=AFXFloatKeyword(self.cmd,'coefprs',TRUE) *self.forcer = AFXFloatKeyword(self.cmd, 'forcer', TRUE)*

def getFirstDialog(self):

return loadDB(self)

Beside the geometrical parameters that influence the deep drawing process simulation quality, there is another category of parameters which include the following: coefficient of friction, blank holder force, punch speed.

All these parameters must be entered in the preprocessing. With the help define how created in this application is done in an extremely simple

.

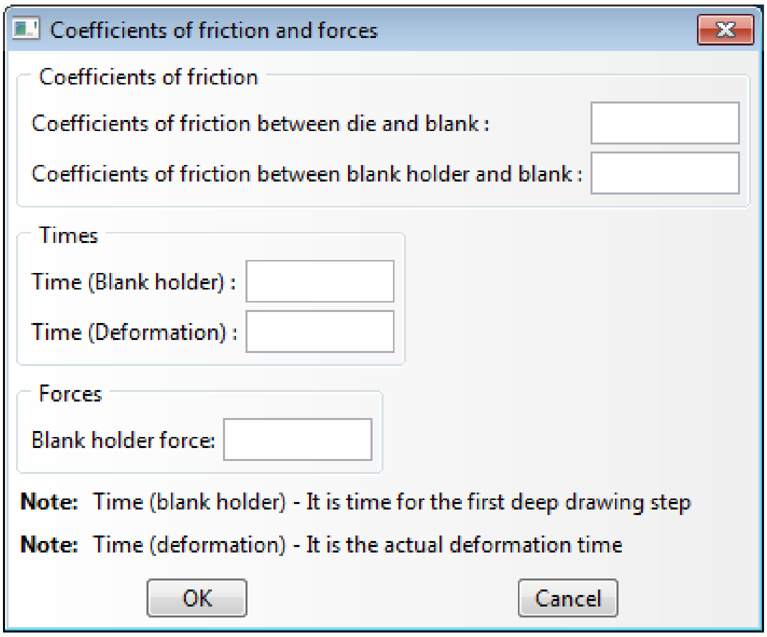

Figure 7. Friction coefficient and the blank holder force setup

The user interface for the boundary conditions is showed in figure 7. For establish the friction coefficient, and the blank holder force. it's defined a class in Python programming language.

Using the CAD- CAE application it was generated the finite element analysis for different versions of input parameters.

The geometric parameters and the friction coefficient can be modify easy through the application. It was analyse the thickness variation , and stress

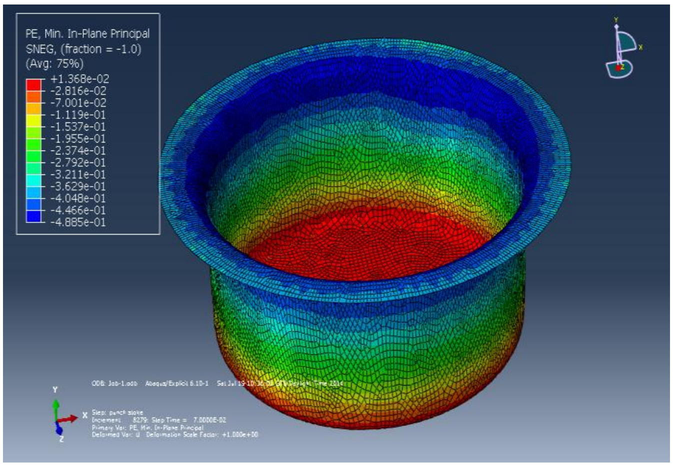

Figure 8. Finite element analysis

4. CONCLUSIONS

Python language programming used in combination with finite element methods can bring significant reduction of necessary time for modelling and simulation process. The deep drawing optimization process can be done only by varying input deep drawing parameters. all these simulations can be made in a cycle that is repeated until the desired result.

5. REFERENCES

1. http://www.tu-chemnitz.de/projekt/abq_hilfe/docs/v6.12/books/gsa/default.htm
2. Ayari, T. Lazghab , E. Bayraktar - Parametric Finite Element Analysis of square cup deep drawing – computational material science and surface engineering – vol 1, 2009
3. Grebenisan Gavril, Radu I - Finite element analysis of a forming process, USING Static Structural (ANSYS) and ansys LS-DYNA, Annual Session of Scientific Papers “IMT Oradea 2010”, Oradea
4. GAO E., LI H., KOU H., CHANG , LI J., and ZHOU L. - Finite element simulation on the deep drawing oftitanium thin-walled surface part - Rare Metals (2010) 29: 108-113