Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Numerical investigation through Finite Elements of Neuber's and Molski-Glinka's rules for calculation of elastoplastic stress and strain in notched elements

Autor: Fernando Pascual Goce

Tutor: Alfredo de Jesús Navarro Robles

Dpto. Ingeniería Mecánica y Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Numerical investigation through Finite Elements of Neuber's and Molski-Glinka's rules for calculation of elastoplastic stress and strain in notched elements

Autor:
Fernando Pascual Goce

Tutor:
Alfredo de Jesús Navarro Robles
Catedrático

Dpto. Ingeniería Mecánica y Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Trabajo Fin de Grado: Numerical investigation through Finite Elements of Neuber's and Molski-Glinka's rules for calculation of elastoplastic stress and strain in notched elements

Autor: Fernando Pascual Goce
Tutor: Alfredo de Jesús Navarro Robles

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Abstract

An investigation on elastoplastic stress and strain produced at notched pieces with different geometries and material properties using Finite Elements and numerical computation software. Application of Neuber's and Molski-Glinka's rules directed to obtaining real stress and strain at notches from data obtained with Finite Elements models. This project sets the basis for future research on the same topic, aiming to use a similar process to obtain results in a wider variety of geometries and materials.

Contents

Notation

σ	Real stress
σ_y	Yield stress
σ_u	Ultimate tensile strength
σ_e^*	Pseudo-stress
σ_n	Nominal stress
σ_{max}	Maximum stress
ε	Real strain
ε_e^*	Pseudo-strain
ε_n	Nominal strain
ε_e	Elastic strain
ε_p	Plastic strain
E	Young's modulus
H	Hardening coefficient
n	Hardening exponent
K_σ	Stress concentration factors
P	Load

1 Stress concentration

First, it is necessary to study the facts, to multiply the number of observations, and then later to search for formulas that connect them so as thus to discern the particular laws governing a certain class of phenomena. In general, it is not until after these particular laws have been established that one can expect to discover and articulate the more general laws that complete theories by bringing a multitude of apparently very diverse phenomena together under a single governing principle.

AUGUSTIN-LOUIS CAUCHY, 1857

Stress is a concept used in continuum mechanics to express the internal forces that are produced inside a solid body by external loads and interactions. Force lines is a graphical method used to visualize these forces by picturing how they "travel" inside the material using imaginary surfaces. This method helps us to see where and why stress concentrations happen and it is an easy way to understand its consequences.

1.1 Stress concentration factors

If all elements were to have a perfect constant section, stress would "travel" at a regular, constant and unidirectional flow. However, hardly ever do machine pieces or structural elements have a constant section throughout its entire length, the presence of holes, notches, shoulders or sudden alterations in geometry changes this simple distribution and make force lines to come very near each other at certain points Figure 1.1.

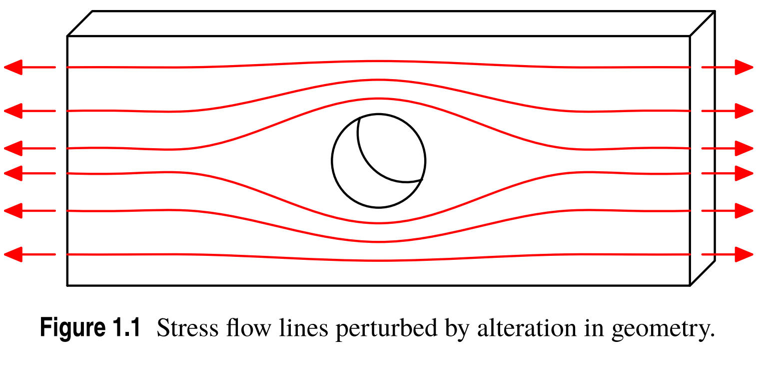

Figure 1.1 Stress flow lines perturbed by alteration in geometry.

This proximity between stress flow lines means that a peak of stress is appearing at a single point. This stress accumulation is often much bigger than the applied load, so we define a *stress concentration factor* as how many times this stress accumulation at a certain point is bigger than the load we are applying.

$$K_\sigma = \frac{\sigma_{max}}{\sigma_n}$$

Where σ_{max} is the maximum stress produced by the perturbation of geometry and σ_{nom} is the nominal stress produced by the load being applied.

1.1.1 Example

If we apply a uniform load, P, to a thin metal sheet with a hole in the center, there will be a stress concentration at the boundaries of this circular hole as this is the point where flow lines are nearest to each other (Figure 1.1). This will result in a stress distribution like the one shown in Figure 1.2.

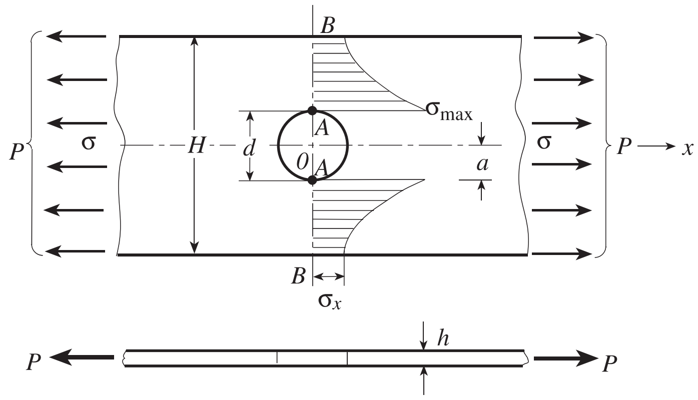

Figure 1.2 Stress distribution in a thin metal sheet with a hole in the center. Image from *Peterson's Stress Concentration Factors, Third Edition.* [1].

Where,

$$\sigma_{nom} = \frac{P}{Hh}$$

and

$$\sigma_{max} = K_\sigma \sigma_n$$

From experience, we know that stress concentration factors from circular holes are in the order of $K_\sigma \approx 3$, so in this case, stress produced at the boundaries of the hole are three times larger than the applied nominal stress. This example shows how important it is to take into consideration stress concentration factors when designing machine elements, as this can be dangerous, especially because their effect is difficult to estimate.

1.2 Traditional calculation of stress concentration factors

Traditionally, stress concentration factors have been estimated by charts included in handbooks. These charts display estimations of K_σ depending on parameters like the radius of curvature of a hole, the penetration of a notch or the width of a notch. As an example, in Figure 1.3 we can see one of these charts. It represents the K_σ produced in an elliptical notch depending on the relation between its radius of curvature and the width of its minimum section when an axial load is applied in a line at the center of this minimum section.

There is a wide variety of charts, depending on the type of load (uniaxial, combined...), the loads applied (axial, torque, momentum...) or changes in geometry (holes, notches, cracks...). All these charts are the result of series of experiments carried out in laboratories in different situations. This means that they are a relatively good representation of reality but this does not mean that all pieces that match the parameters of a certain chart are going to perform the same way as these charts show. This is because the results depicted in these charts are generally averages from several experiments carried out in the same conditions in different pieces that differ from each other due to slight natural differences in their material compositions.

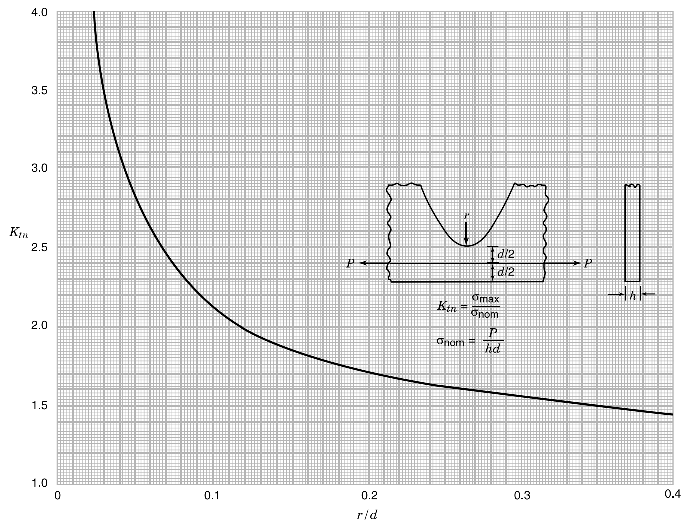

Figure 1.3 Chart representing K_σ produced in an elliptical notch. Image from *Peterson's Stress Concentration Factors, Third Edition.* [1].

1.3 Calculation of stress concentration factors through Finite Elements

The development in modern software of Finite Elements (FE) offers an alternative way to estimate the effects of loads applied in very complex geometries with relative ease. This makes the need for charts not that crucial in design, as we can nowadays create a digital model of the geometry we are interested to study and perform an analysis in any commercial software, obtaining a numerical solution according to theoretical equations from strength of materials' theory. This method has the advantage that we can estimate a K_σ for the exact geometry we are interested in and we do not depend on finding geometries that appear in handbooks and match with our requirements. However, it also has the same disadvantage it was pointed out before for the traditional methods, namely, materials present imperfections that may not make them perform as the idealized models from FE software.

To achieve the objectives of this project, it is necessary to create Finite Elements models that fairly reproduce real phenomena in order to obtain reliable results without the need for laboratory tests, which are often tedious and time consuming. For this purpose, the software ANSYS will be used from now on.

2 Modeling in ANSYS

There is a wide variety of commercial FEM software that offer different possibilities for engineers to calculate stresses and strains for different geometries subjected to different loads. In this project, I will be using the software ANSYS to create models of notched pieces and provide reliable results in order to establish a way to estimate stresses at these notches whose parameters can be easily changed to obtain an infinite number of combinations.

I will be modeling geometries that already appear in handbooks and articles, whose data come from laboratory experiments. The purpose of choosing geometries that already appear in handbooks is to simulate the conditions under which these experimental results were obtained and compare them to the results we get from ANSYS. Both results should be similar, otherwise the numerical model would not be representing the experiment fairly. If this can be achieved, one can be certain that their numerical models are reliable and so, we can change parameters to obtain reliable results without the need of costly laboratory experiments.

The chosen geometries will be the ones shown in Figure 2.1 and Figure 2.2 , which appear in *Peterson's Stress Concentration Factors, Third Edition* [1].

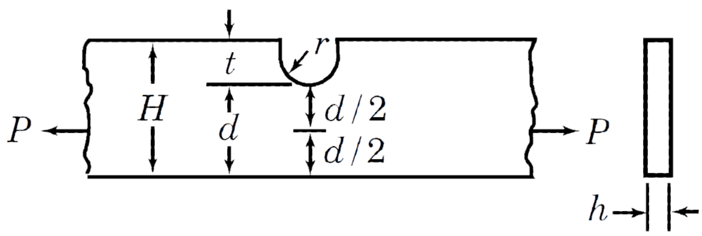

Figure 2.1 Single U-Shape notched plate. Original image taken from *Peterson's Stress Concentration Factors, Third Edition* [1].

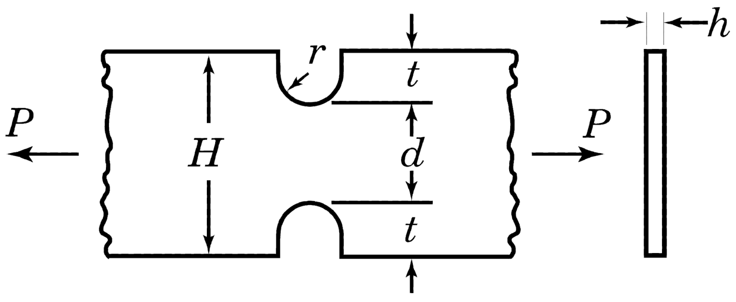

Figure 2.2 Opposite U-Shape notched plate. Original image taken from *Peterson's Stress Concentration Factors, Third Edition* [1].

2.1 Single U-shaped notched plate

For this first model, I have chosen a plate presenting a single U-shaped notch. In the experiment, a constant load was applied along a line at the center of the minimum section all across its depth as shown in Figure 2.3.

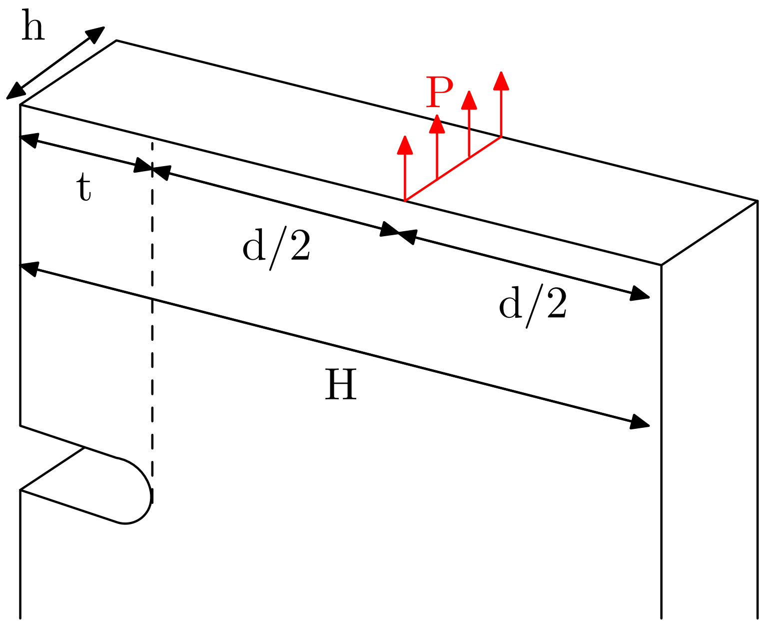

Figure 2.3 Load distribution along a line at the center of the minimum section.

To apply these conditions and be able to obtain results from ANSYS it is necessary to perform the following steps. These steps will show how I created the code in ANSYS to perform a Finite Elements pure elastic solid body analysis. Variations in this code and other types of analysis will be shown in following sections.

2.1.1 Parameters

Parameters can be defined in ANSYS by means of string variables, making it easy to relate every following instruction to the program to these parameters and so on, allowing us to easily change the whole model by changing these parameters. An example of this is shown in Code 2.1.

We can also create loops in order to change one parameter maintaining the rest constant. This is done with the command `*DO`, as it is shown in Code 2.2. In this case I am changing the notch's radius (R) with the values obtained in parameter DIV. Then I am saving all previously saved parameters with command `PARSAV` and erasing all previously defined geometries with command `CLEAR` so we can create a new one defined with the values contained in the next position of vector DIV. Command `PARRES` is used to reestablish parameters saved with `PARSAV`. We also must define where the loop finishes, this is done with command `*ENDDO`.

2.1.2 Material properties

Material properties are a key factor in all analysis in Finite Elements. They will provide the software with vital information about the behavior of the element being studied. Here we can set conditions of plane stress, elasto-plastic behavior and much more. In this case, I am just considering an isotropic, pure elastic material behaving as a solid body, so ANSYS only needs the material's Young's modulus, Poisson's coefficient and the type of element we want to use for our model, in this case `SOLID186`, which is a higher order 3-D 20-node solid element that exhibits quadratic displacement behavior [2]. To define material properties, we need to enter in the *Preprocessor* section, defined as `/PREP7`, as shown in Code 2.3

If we want to set conditions for Plane Stress, we can use element `PLANE182` and use the different KEYOPTIONS to define, in this case, the width of the plate, as shown in Code 2.5. It is also possible to set conditions for Plane Strain just by changing the KEYOPTIONS of the previous code and replace the with the ones shown in Code **??**.

2.1.3 Creating a solid body with SOLID186

The solid body in a numerical model is the one in charge of defining the geometry intended to study and it is also defined in the *Preprocessor* section. Considering the geometry's symmetry, it is possible just to define the upper half of it and then applying symmetry boundary conditions Figure 2.4, avoiding the unnecessary use of nodes and calculation time. At first, I tried to create the whole plate geometry in a sole volume, as it was the easiest way to define solid geometries, but it turned out that it was impossible to apply the boundary conditions needed in this experiment, because there was no way to apply loads on the specific line where they should be. Instead, I opted for creating an area using lines and deliberately cutting the line that defined the upper part of the bar right at the point at the center of the minimum section, as is shown at Figure 2.5. To create the notch, in this case a U-shaped notch, I created a square area and a circular area and subtracted this from the initial rectangular bar area, obtaining in this way the final desired area, shown in Figure 2.5.

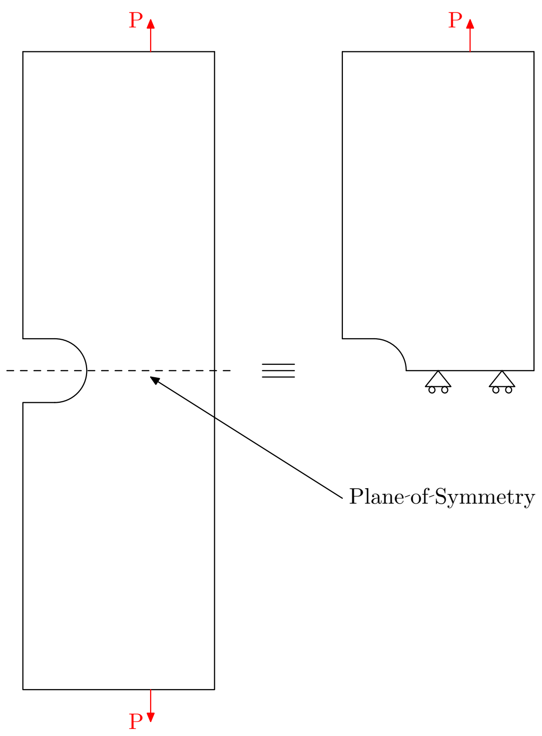

Figure 2.4 Applying symmetry.

After defining this area, I extruded it in both directions in order to obtain the desired depth (half of this depth in each direction) and then glued both volumes into one. The code used for this definition of geometry is shown in Code 2.6

2.1.4 Creating a plane stress plate with PLANE182

The way to define our geometry with `PLANE182` is similar to the one with `SOLID186` but in this case we only have to define an area as we have previously defined the desired plate's width. This also means that all loads applied in a keypoint of our area will be displayed across the whole width of the plate, so we only need to apply our load in the keypoint defined at the center of the minimum section. The code used for this definition of geometry is shown in Code 2.7

2.1.5 Meshing, loads and constraints

Later, still in the *Preprocessor* section, I meshed the whole volume or area, where a line of nodes was created across the geometry's width at the place where lines were cut before, allowing us to apply loads at these nodes in our solid body model, Figure 2.6, simulating the conditions of the experiment being reproduced.

The mesh was refined at certain lines and surfaces, like in the entire notch surface and moreover in its middle line (peak), Figure 2.7, and the line where the loads are applied, to make load better distributed all

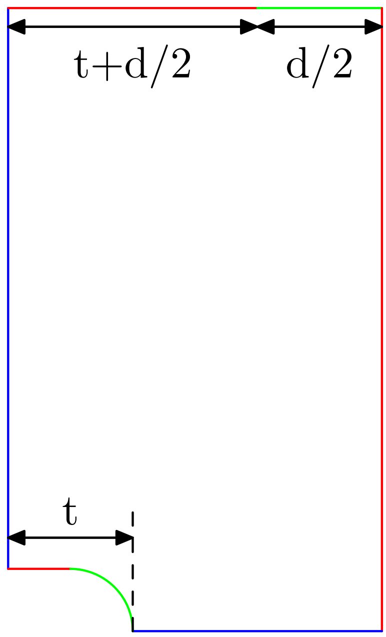

Figure 2.5 Lines that define the area of the desired geometry.

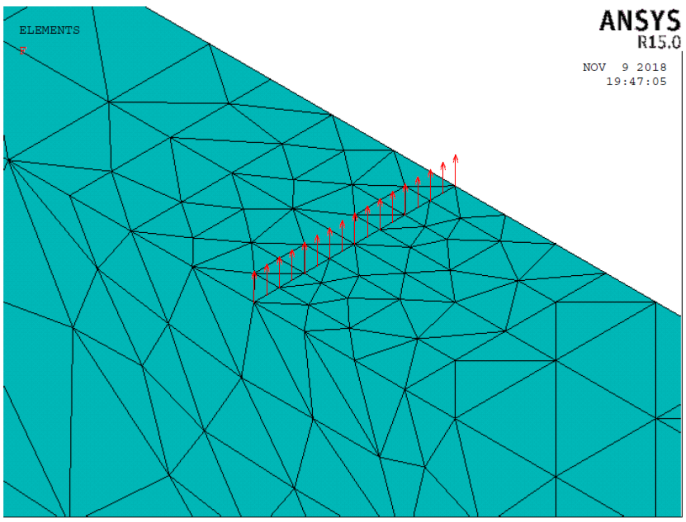

Figure 2.6 Loads applied at nodes situated on the line at the center of the minimum section.

along the depth. It is important to properly define the areas where we are interested in obtaining precise results.

As I have decided to consider the geometry's symmetry, symmetry boundary conditions have been applied all over the minimum section. Also, constraints to the movement in direction X have been applied at the corner of the minimum section with the unnotched face and constraints to the movement in direction Z on a line that passes along the minimum section at the center of its depth, to restrict rigid solid movement, as shown in Figure 2.15.

The code that includes meshing and the application of loads and constraints in the solid body model is shown at Code 2.8. Code in the plane stress model is similar, but it is not required that the movement in the Z direction be restricted.

There is another possibility in order to properly refine the mesh. It is a better but more costly way to do it. This way is to refine the entire volume or area with a fine mesh, using the *smart mesh* option to adapt it to the geometry that we are meshing. This way, we will avoid the appearances of warnings due to the shape of elements situated in the most refined points. To do this, we must use the code shown in Code 2.9 for the solid

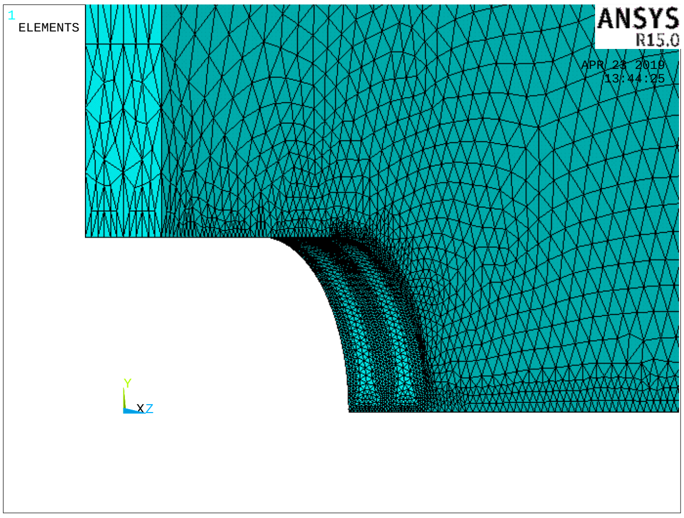

Figure 2.7 Mesh refined at notch.

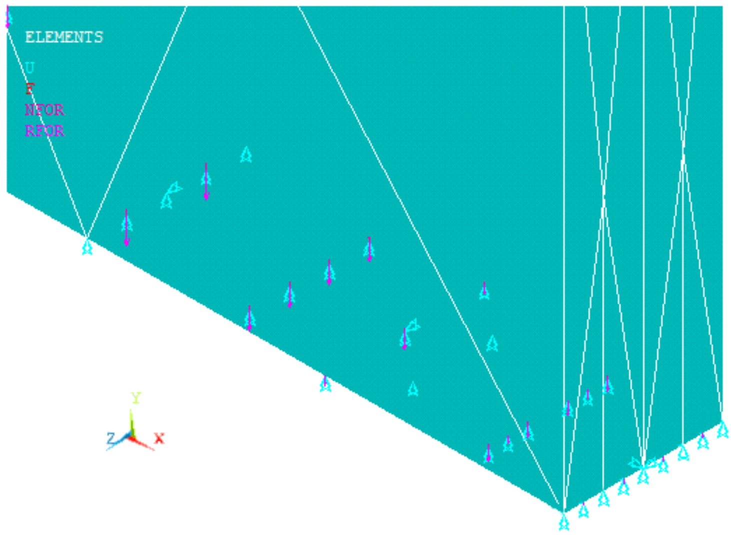

Figure 2.8 Constraints at the corner of the minimum section with the unnotched face.

model and Code 2.10 for the plane stress model.

Another possible change in this code is if we wanted to place a distributed load instead of a load at the center of the middle of the minimum section. Codes 2.11 and 2.12 show how to do this in solid and plane stress models respectively.

2.1.6 Solution

To generate results, it is needed that the software compiles Finite Elements equations for the input it has been given. For this, we need to enter in the *Solution* processor and run the analysis. In this case, a static structural solution is specified. This is done according to Code 2.13.

2.1.7 Obtaining results

When the solution is done, it is possible to plot different results, such as stress, strain, total deformation, etc. It is also possible to create graphs with certain data for certain points we are interested in. The aim of this model is to find K_σ, so what we need to calculate is σ_{max}. The maximum stress is going to be produced at the bottom of the notch, as it is there where the maximum stress concentration is going to be. I have represented stress distribution along the notch's root using a *path* from one edge to the other, representing stress in the Y direction (Figure 2.9) and along the center of the minimum section (Figure 2.10).

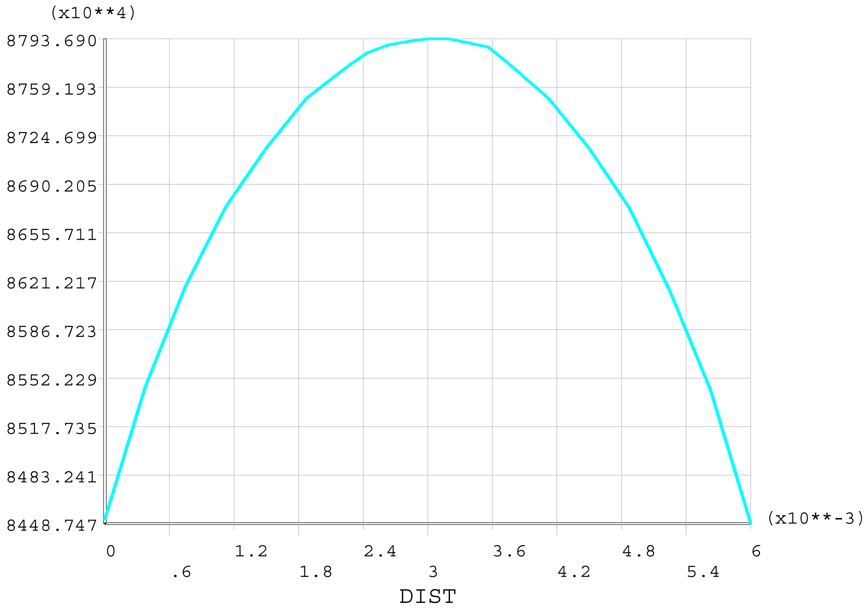

Figure 2.9 Stress distribution along the notch's peak.

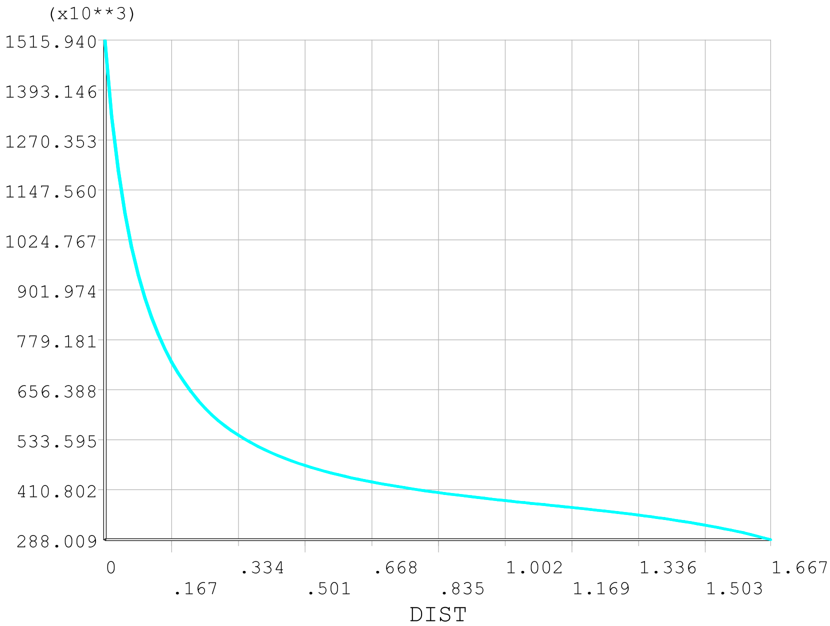

Figure 2.10 Stress distribution along the center of the minimum section.

The codes to create these paths are shown in Code 2.14 and Code 2.15. These codes are defined by the values of different keypoints, so it is important to have them well defined.

2.1.8 Results output

To write results in an output file we can use command `*VWRITE`. If a loop is used, we need to create a *macro* and so we can create an array of results that we can use later in other software like EXCEL or MATLAB. In this case I have created the following *macro*, Code 2.16. There are two *macros* in this code, the first is for writing the first result and the second is to write the following results down in a column structure.

However, before writing anything we need to define what do we want to write in this output file. To do this we use the command `*GET`. As we are interested in obtaining the maximum stress at the notch's root, I decided to get the maximum stress in the Y direction of the last path I plotted and then use the *macro* previously defined. This is done according to Code 2.17.

2.1.9 Codes

Code 2.1 Parameters in ANSYS.

```
!PARAMETERS
H=1 !WIDTH
B=1.5 !HEIGHT
D=0.05 !DEPTH
T=0.3*H !DEPTH OF TOTAL NOTCH
Eyoung=202375000 !YOUNG MODULUS
Poisson=0.33 !POISSON COEFFICIENT
POI=40000  !LOAD
```

Code 2.2 Loop in ANSYS.

```
DIV=1.1,2.1,3.1,4.1,5.1,6.1,7.1,8.1,9.1

*DO,I,1,9

PARSAV,ALL,
FINISH
/CLEAR
PARRES
```

Code 2.3 Material properties for solid body model.

```
/PREP7
ET,1,SOLID186

! MATERIAL
MP,EX,1,Eyoung
MP,EY,1,Eyoung
MP,EZ,1,Eyoung
MP,PRXY,1,Poisson
MP,PRXZ,1,Poisson
MP,PRYZ,1,Poisson
```

Code 2.4 Material properties for plane stress model.

```
/PREP7
ET,1,PLANE182

KEYOPT,1,1,0
KEYOPT,1,3,3
KEYOPT,1,6,0

R,1,D,

! MATERIAL
MP,EX,1,Eyoung
```

```
MP,EY,1,Eyoung
MP,EZ,1,Eyoung
MP,PRXY,1,Poisson
MP,PRXZ,1,Poisson
MP,PRYZ,1,Poisson
```

Code 2.5 Material properties for plane stress model.

```
/PREP7
ET,1,PLANE182

KEYOPT,1,1,0
KEYOPT,1,3,2
KEYOPT,1,6,0
```

Code 2.6 Geometry with SOLID186.

```
!KEYPOINTS
K,1,0,0,0
K,2,H,0,0
K,3,0,B/2,0
K,4,H,B/2,0
K,6,R+(H-R)/2,B/2,0
K,7,0,R,0
K,8,T-R,0,0
K,9,T-R,R,0

!RECTANGLE
LSTR,8,2
LSTR,2,4
LSTR,4,6
LSTR,6,3
LSTR,3,7
LSTR,7,9
LSTR,9,8

FLST,2,7,4
FITEM,2,7
FITEM,2,6
FITEM,2,5
FITEM,2,4
FITEM,2,3
FITEM,2,2
FITEM,2,1
AL,P51X

!NOTCH CIRCLE
CIRCLE, 8, R,,,,

FLST,2,4,4
FITEM,2,8
FITEM,2,9
FITEM,2,10
```

```
FITEM,2,11
AL,P51X

ASBA,      1,      2

!EXTRUSION
VOFFST,3,D/2, ,
VOFFST,3,-D/2, ,

FLST,2,2,6,ORDE,2
FITEM,2,1
FITEM,2,-2
VGLUE,P51X
```

Code 2.7 Geometry with PLANE182.

```
!KEYPOINTS
K,1,0,0,0
K,2,H,0,0
K,3,0,B/2,0
K,4,H,B/2,0
K,6,R+(H-R)/2,B/2,0
K,7,0,R,0
K,8,T-R,0,0
K,9,T-R,R,0

!RECTANGLE
LSTR,8,2
LSTR,2,4
LSTR,4,6
LSTR,6,3
LSTR,3,7
LSTR,7,9
LSTR,9,8

FLST,2,7,4
FITEM,2,7
FITEM,2,6
FITEM,2,5
FITEM,2,4
FITEM,2,3
FITEM,2,2
FITEM,2,1
AL,P51X

!CIRCULAR NOTCH
CIRCLE, 8, R,,,,

FLST,2,4,4
FITEM,2,8
FITEM,2,9
FITEM,2,10
FITEM,2,11
AL,P51X
```

```
ASBA,      1,      2
```

Code 2.8 Meshing, loads and constraints in solid model.

```
!MESH
MSHKEY,0
MSHAPE,1,3d
FLST,5,2,6,ORDE,2
FITEM,5,1
FITEM,5,-2
CM,_Y,VOLU
VSEL, , , ,P51X
CM,_Y1,VOLU
CHKMSH,'VOLU'
CMSEL,S,_Y

VMESH,_Y1

!REFINE
FLST,5,10,5,ORDE,10
FITEM,5,2
FITEM,5,5
FITEM,5,-6
FITEM,5,8
FITEM,5,-9
FITEM,5,11
FITEM,5,13
FITEM,5,-14
FITEM,5,16
FITEM,5,-17
CM,_Y,AREA
ASEL, , , ,P51X
CM,_Y1,AREA
CMSEL,S,_Y
CMDELE,_Y

AREFINE,_Y1, , ,1,0,1,1
CMDELE,_Y1

FLST,5,8,4,ORDE,6
FITEM,5,1
FITEM,5,12
FITEM,5,-15
FITEM,5,22
FITEM,5,28
FITEM,5,-29
CM,_Y,LINE
LSEL, , , ,P51X
CM,_Y1,LINE
CMSEL,S,_Y
CMDELE,_Y

LREFINE,_Y1, , ,1,1,1,1
CMDELE,_Y1
```

```
!LOADS
FLST,5,2,4,ORDE,2
FITEM,5,17
FITEM,5,31
LSEL,S, , ,P51X
NSLL,S,1
FLST,2,9,1,ORDE,7
FITEM,2,32
FITEM,2,2036
FITEM,2,4026
FITEM,2,-4028
FITEM,2,6267
FITEM,2,8257
FITEM,2,-8259

/GO
F,P51X,FY,POI

ALLSEL,ALL

!CONSTRAIN
FLST,2,2,5,ORDE,2
FITEM,2,2
FITEM,2,11
DA,P51X,SYMM

FLST,2,1,4,ORDE,1
FITEM,2,12
/GO
DL,P51X, ,UZ,0
FLST,2,2,4,ORDE,2
FITEM,2,16
FITEM,2,30
/GO
DL,P51X, ,UX,0
```

Code 2.9 Refining for complete volume.

```
FALTA CODIGO
```

Code 2.10 Refining for complete area.

```
FLST,5,1,5,ORDE,1
FITEM,5,3
CM,_Y,AREA
ASEL, , , ,P51X
CM,_Y1,AREA
CMSEL,S,_Y
CMDELE,_Y

AREFINE,_Y1, , ,3,0,1,1
CMDELE,_Y1
```

Code 2.11 Distributed load in solid model.

```
FLST,2,4,5,ORDE,4
FITEM,2,5
FITEM,2,-6
FITEM,2,13
FITEM,2,-14
/GO

SFA,P51X,1,PRES,PRE
```

Code 2.12 Distributed load in plane stress model.

```
FLST,2,2,4,ORDE,2
FITEM,2,3
FITEM,2,-4
/GO

SFL,P51X,PRES,PRE,
```

Code 2.13 Solve.

```
/Sol

ANTYPE,0
Solve
```

Code 2.14 Code to create a path along the notch's peak.

```
/POST1
path,alongZ,2,,1000
ppath,1,,kx(16),ky(16),kz(16)
ppath,2,,kx(8),ky(8),kz(8)
pdef,s_yy,s,y
plpath,s_yy
```

Code 2.15 Code to create a path along the center of the minimum section.

```
/POST1
path,patron,2,,100
ppath,1,,kx(5),ky(5),kz(5)
ppath,2,,kx(2),ky(2),kz(2)
pdef,s_yy,s,y
plpath,s_yy
```

Code 2.16 Code to create a macro to write array of results.

```
*CREATE,'MACRO_ESCRIT_1',' ',' '
 *CFOPEN,DATOS,dat
 *VWRITE,S_yy_max
 ((SP)1E14.6,' ')
 *CFCLOS
*END

*CREATE,'MACRO_ESCRIT_2',' ',' '
 *CFOPEN,DATOS,dat,,APPEND
 *VWRITE,S_yy_max
 ((SP)1E14.6,' ')
 *CFCLOS
*END
```

Code 2.17 Code to get the maximum stress in the Y direction of the last path plotted and use *macro*.

```
*GET,S_yy_max,PATH,0,MAX,s_yy

*IF,I,EQ,1,THEN
 *USE,MACRO_ESCRIT_1
*ELSE
 *USE,MACRO_ESCRIT_2
*ENDIF
```

2.2 Opposite U-shape notched plate

This second geometry is similar to the first one but, this time, the plate has two opposite U-shape notches. Due to this geometry, we can benefit from its symmetry and just create one quarter of the plate, setting symmetry boundary conditions in both cut faces and applying half of the original load, as it is shown in Figure 2.11.

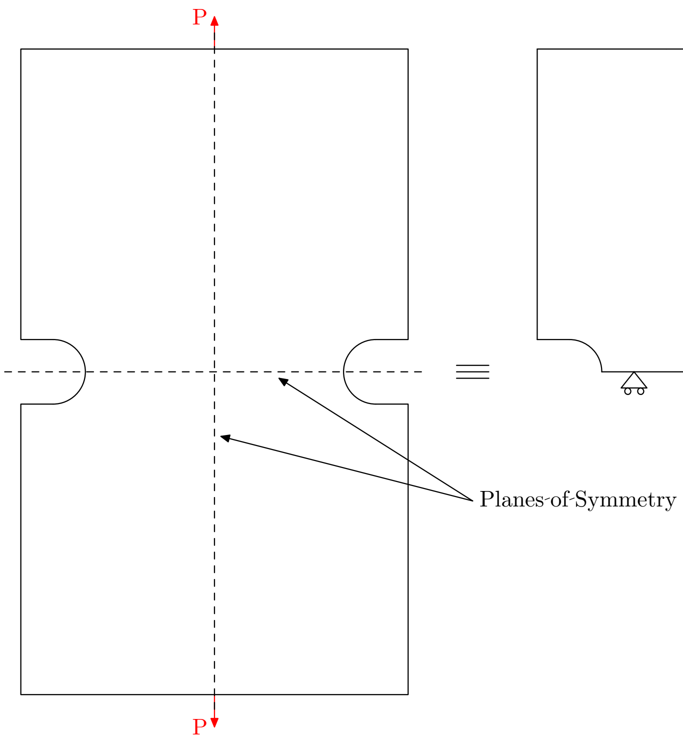

Figure 2.11 Applying symmetry to double symmetrical geometry.

Therefore, to change the model for this geometry we only need to change the base parameter H and the constraints to apply symmetry boundary conditions in the middle face and erase the restriction applied in the X direction displacement. The complete code for this model as a solid body is shown in Code 2.18 and as a plane stress model in Code 2.19.

2.2.1 Codes

Code 2.18 Complete code for solid body opposite U-shape notched plate.

```
/CLEAR

*CREATE,'MACRO_ESCRIT_1',' ',' '
 *CFOPEN,DATOS,dat
 *VWRITE
 (S_yy_max)
 *VWRITE,S_yy_max
 ((SP)1E14.6,' ')
 *CFCLOS
*END

*CREATE,'MACRO_ESCRIT_2',' ',' '
 *CFOPEN,DATOS,dat,,APPEND
 *VWRITE
```

```
 ('S_yy_max')
 *VWRITE,S_yy_max
 ((SP)1E14.6,' ')
 *CFCLOS
*END

!PARAMETERS
H=0.127 !WIDTH
H=H/2
B=0.3048 !HEIGHT
D=0.01 !DEPTH
T=0.3*H !DEPTH OF NOTCH
Eyoung=202375000
Poisson=0.33
POI=40000  !LOAD

DIV=1.1,2.1,3.1,4.1,5.1,6.1,7.1,8.1,9.1
*DO,I,1,9

PARSAV,ALL,
FINISH
/CLEAR
PARRES

R=T/DIV(I) !RADIO

/PREP7
ET,1,SOLID186

! MATERIAL
MP,EX,1,Eyoung
MP,EY,1,Eyoung
MP,EZ,1,Eyoung
MP,PRXY,1,Poisson
MP,PRXZ,1,Poisson
MP,PRYZ,1,Poisson

!KEYPOINTS
K,1,0,0,0
K,2,H,0,0
K,3,0,B/2,0
K,4,H,B/2,0
K,6,R+(H-R)/2,B/2,0
K,7,0,R,0
K,8,T-R,0,0
K,9,T-R,R,0

!RECTANGLE
LSTR,8,2
LSTR,2,4
LSTR,4,6
LSTR,6,3
LSTR,3,7
LSTR,7,9
LSTR,9,8

FLST,2,7,4
```

```
FITEM,2,7
FITEM,2,6
FITEM,2,5
FITEM,2,4
FITEM,2,3
FITEM,2,2
FITEM,2,1
AL,P51X

!CIRCULAR NOTCH
CIRCLE, 8, R,,,,

FLST,2,4,4
FITEM,2,8
FITEM,2,9
FITEM,2,10
FITEM,2,11
AL,P51X

ASBA,      1,      2

!EXTRUSION
VOFFST,3,D/2, ,
VOFFST,3,-D/2, ,

FLST,2,2,6,ORDE,2
FITEM,2,1
FITEM,2,-2
VGLUE,P51X

!MESH
MSHKEY,0
MSHAPE,1,3d
FLST,5,2,6,ORDE,2
FITEM,5,1
FITEM,5,-2
CM,_Y,VOLU
VSEL, , , ,P51X
CM,_Y1,VOLU
CHKMSH,'VOLU'
CMSEL,S,_Y

VMESH,_Y1

!REFINE
FLST,5,2,5,ORDE,2
FITEM,5,9
FITEM,5,17
CM,_Y,AREA
ASEL, , , ,P51X
CM,_Y1,AREA
CMSEL,S,_Y
CMDELE,_Y

AREFINE,_Y1, , ,1,0,1,1
CMDELE,_Y1
```

```
FLST,5,2,4,ORDE,2
FITEM,5,17
FITEM,5,31
CM,_Y,LINE
LSEL, , , ,P51X
CM,_Y1,LINE
CMSEL,S,_Y
CMDELE,_Y

LREFINE,_Y1, , ,1,1,1,1
CMDELE,_Y1

FLST,5,1,3,ORDE,1
FITEM,5,5
CM,_Y,KP
KSEL, , , ,P51X
CM,_Y1,KP
CMSEL,S,_Y
CMDELE,_Y

KREFINE,_Y1, , ,1,1,1,1
CMDELE,_Y1

!LOADS
FLST,5,2,4,ORDE,2
FITEM,5,17
FITEM,5,31
LSEL,S, , ,P51X
NSLL,S,1
FLST,2,9,1,ORDE,7
FITEM,2,32
FITEM,2,2036
FITEM,2,4026
FITEM,2,-4028
FITEM,2,6267
FITEM,2,8257
FITEM,2,-8259

/GO
F,P51X,FY,POI

ALLSEL,ALL

FLST,2,2,5,ORDE,2
FITEM,2,2
FITEM,2,11
DA,P51X,SYMM

FLST,2,1,4,ORDE,1
FITEM,2,12

/GO
DL,P51X, ,UZ,0

FLST,2,2,5,ORDE,2
FITEM,2,4
FITEM,2,12
```

```
DA,P51X,SYMM

FINISH

/Sol
ANTYPE,0
Solve
FINISH

/POST1
path,patron,2,,100
ppath,1,,kx(5),ky(5),kz(5)
ppath,2,,kx(5)+R,ky(5),kz(5)
pdef,s_yy,s,y
plpath,s_yy

path,alongZ,2,,1000
ppath,1,,kx(16),ky(16),kz(16)
ppath,2,,kx(8),ky(8),kz(8)
pdef,s_yy,s,y
plpath,s_yy

*GET,S_yy_max,PATH,0,MAX,s_yy

*IF,I,EQ,1,THEN
 *USE,MACRO_ESCRIT_1
*ELSE
 *USE,MACRO_ESCRIT_2
*ENDIF

*ENDDO
FINISH
```

Code 2.19 Complete code for plane stress opposite U-shape notched plate.

```
/CLEAR

*CREATE,'MACRO_ESCRIT_1',' ',' '
 *CFOPEN,DATOS_BARATTA,dat
 *VWRITE,S_yy_max
 ((SP)1E14.6,' ')
 *CFCLOS
*END

*CREATE,'MACRO_ESCRIT_2',' ',' '
 *CFOPEN,DATOS_BARATTA,dat,,APPEND
 *VWRITE,S_yy_max
 ((SP)1E14.6,' ')
 *CFCLOS
*END

!PARAMETERS
H=2 !WIDTH
H=H/2
B=5 !HEIGHT
```

```
D=0.05 !DEPTH
T=0.3*H !DEPTH OF NOTCH
Eyoung=202375000
Poisson=0.33
POI=10000000  !LOAD
PRE=-POI/(H*D)

DIV=1.1,2.1,3.1,4.1,5.1,6.1,7.1,8.1,9.1

*DO,I,1,9

PARSAV,ALL,
FINISH
/CLEAR
PARRES

R=T/DIV(I) !RADIUS

/PREP7
ET,1,PLANE182

KEYOPT,1,1,0
KEYOPT,1,3,3
KEYOPT,1,6,0

R,1,D,

! MATERIAL
MP,EX,1,Eyoung
MP,EY,1,Eyoung
MP,EZ,1,Eyoung
MP,PRXY,1,Poisson
MP,PRXZ,1,Poisson
MP,PRYZ,1,Poisson

!KEYPOINTS
K,1,0,0,0
K,2,H,0,0
K,3,0,B/2,0
K,4,H,B/2,0
K,6,R+(H-R)/2,B/2,0
K,7,0,R,0
K,8,T-R,0,0
K,9,T-R,R,0

!RECTANGLE
LSTR,8,2
LSTR,2,4
LSTR,4,6
LSTR,6,3
LSTR,3,7
LSTR,7,9
LSTR,9,8

FLST,2,7,4
FITEM,2,7
FITEM,2,6
```

```
FITEM,2,5
FITEM,2,4
FITEM,2,3
FITEM,2,2
FITEM,2,1
AL,P51X

!CIRCULAR NOTCH
CIRCLE, 8, R,,,,

FLST,2,4,4
FITEM,2,8
FITEM,2,9
FITEM,2,10
FITEM,2,11
AL,P51X

ASBA,      1,       2

!MESH
MSHKEY,0
CM,_Y,AREA
ASEL, , , ,      3
CM,_Y1,AREA
CHKMSH,'AREA'
CMSEL,S,_Y

AMESH,_Y1

!REFINE
FLST,5,1,5,ORDE,1
FITEM,5,3
CM,_Y,AREA
ASEL, , , ,P51X
CM,_Y1,AREA
CMSEL,S,_Y
CMDELE,_Y

AREFINE,_Y1, , ,3,0,1,1
CMDELE,_Y1

!CONSTRAINTS
FLST,2,2,4,ORDE,2
FITEM,2,2
FITEM,2,12
DL,P51X, ,SYMM

!LOADS
!FLST,2,1,3,ORDE,1
!FITEM,2,4

!/GO
!FK,P51X,FY,POI

FLST,2,2,4,ORDE,2
FITEM,2,3
FITEM,2,-4
```

```
/GO

SFL,P51X,PRES,PRE,

FINISH

/SOL
SOLVE
FINISH

/POST1
path,patron,2,,100
ppath,1,,kx(5),ky(5),kz(5)
ppath,2,,kx(2),ky(5),kz(5)
pdef,s_yy,s,y
plpath,s_yy

*GET,S_yy_max,PATH,0,MAX,s_yy

*IF,I,EQ,1,THEN
 *USE,MACRO_ESCRIT_1
*ELSE
 *USE,MACRO_ESCRIT_2
*ENDIF

*ENDDO

FINISH
```

2.3 Single and opposite semicircular notched plate

This last model is obtained by setting $t/r = 1$. This means that the notch's shape is one perfect semicircle. The interest for this particularity is that we can find an analytical solutions for this geometry such as those presented by Maunsell [7] and Ling [6], which will be discussed later. In relation with the modeling in ANSYS, the only difference with the U-shape notch codes is that there is no need to create a little rectangle before the circular part of the notch. This makes the code to create the initial unextruded area as shown in Code 2.20, leaving the rest as it was shown in the previous codes.

Code 2.20 Code to create the initial unextruded area for the semicircular notch case.

```
!KEYPOINTS
K,1,0,0,0
K,2,H,0,0
K,3,0,B/2,0
K,4,H,B/2,0
K,6,R+(H-R)/2,B/2,0
K,7,0,0,0

!RECTANGLE
LSTR,7,2
LSTR,2,4
LSTR,4,6
LSTR,6,3
LSTR,3,7

FLST,2,5,4
FITEM,2,5
FITEM,2,4
FITEM,2,3
FITEM,2,2
FITEM,2,1
AL,P51X

!CIRCULAR NOTCH
CIRCLE, 7, R,,,,

FLST,2,4,4
FITEM,2,7
FITEM,2,6
FITEM,2,8
FITEM,2,9
AL,P51X

ASBA,      1,      2
```

2.4 Modeling an elasto-plastic material

The way to model an elasto-plastic material in ANSYS is to create a $\sigma - \varepsilon$ curve that defines the material behavior, see Ramberg-Osgood relationship (Section 4.1). PLANE182 and SOLID186 allow us to create this curve by using a nonlinear, inelastic and rate independent material model, which can be defined with *keyoptions*. We only need to provide a maximum of 20 points and the software automatically interpolates values between adjacent points. I have defined the values of this curve in MATLAB and then picked an equally spaced sample of 20 points from this curve, then I introduced the values in ANSYS. In relation with the code, we only need to change the Material properties (Section 2.1.2) like it is shown in Code 2.21, leaving the rest as it was. The same can be applied to SOLID186. It is very important that the starting slope of the curve we introduced agrees with the Young's modulus we provided.

Code 2.21 Code to introduce elasto-plastic material properties.

```
ET,1,PLANE182

KEYOPT,1,1,0
KEYOPT,1,3,3
KEYOPT,1,6,0

R,1,D,

MPTEMP,,,,,,,,
MPTEMP,1,0
MPDATA,EX,1,,Eyoung
MPDATA,PRXY,1,,Poisson
TB,KINH,1,1,20,0
TBTEMP,0
TBPT,,    0.000049413319990 , 10000000 ,
TBPT,,    0.000307467294401 , 62105263.16 ,
TBPT,,    0.000574851941833 , 114210526.3 ,
TBPT,,    0.000884167341589 , 166315789.5 ,
TBPT,,    0.001306155930431 , 218421052.6 ,
TBPT,,    0.001962113561667 , 270526315.8 ,
TBPT,,    0.003035259750046 , 322631578.9 ,
TBPT,,    0.004781396972906 , 374736842.1 ,
TBPT,,    0.007539035785502 , 426842105.3 ,
TBPT,,    0.011739095766390 , 478947368.4 ,
TBPT,,    0.017914257038170 , 531052631.6 ,
TBPT,,    0.026708016232101 , 583157894.7 ,
TBPT,,    0.038883487418425 , 635263157.9 ,
TBPT,,    0.055331979496578 , 687368421.1 ,
TBPT,,    0.077081375162851 , 739473684.2 ,
TBPT,,    0.105304331911138 , 791578947.4 ,
TBPT,,    0.141326322017036 , 843684210.5 ,
TBPT,,    0.186633525757482 , 895789473.7 ,
TBPT,,    0.242880589999978 , 947894736.8 ,
TBPT,,    0.311898262604064 , 1000000000 ,
```

2.5 Aspects to consider

2.5.1 Plate's width

In solid models, the plate's width seems to be a problem when it is very thin. If the complete volume refine alternative is used (see Section 2.1.5), there seems to be no or very few warnings but when the results are plotted, they do not show a "perfect" parabola as one would expect, but rather a somewhat warped curve instead, as shown in Figure 2.12. In plane stress models, this problem does not appear, allowing us to apply a width as thin as we want.

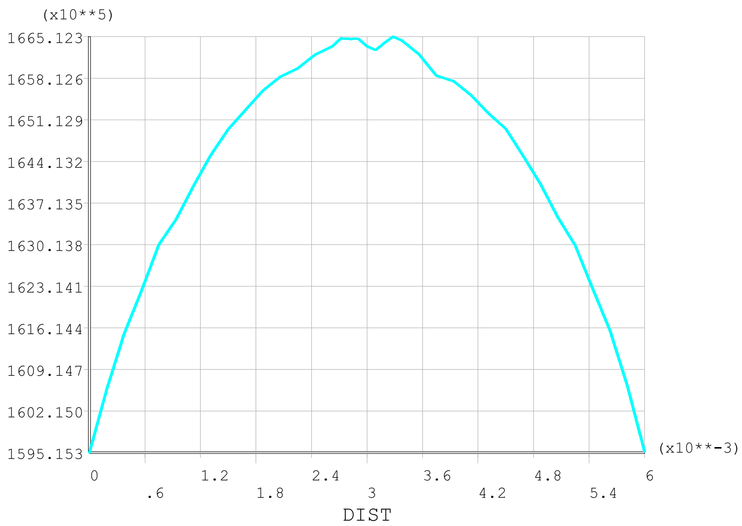

Figure 2.12 Stress distributions along the notch's peak with a "thin" width.

2.5.2 Plate's height

In both solid and plane stress models, the plate's height seems to be crucial in order to obtain reliable results. If we apply a "short" height, stress distributions along the minimum section are distorted due to the effect of the area of application of the loads, Figure 2.13. This effect is more noticeable when when the distribution along the middle line of the minimum section is applied.

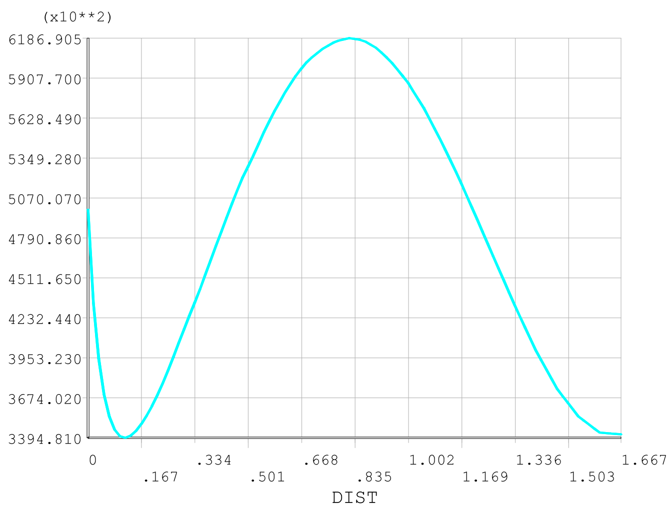

Figure 2.13 Stress distributions along the minimum section with a "short" height.

2.5.3 High stresses produced in loads applied directly to nodes

To apply loads in both solid and plane stress models as they were defined in Peterson's book [1], all along the line at the center of the minimum section, it was necessary to force the software to create a line of nodes at this point, see Sections 2.1.3 and 2.1.4. For some reason, it is impossible to apply a pressure load at this line so, to recreate the effects due to this kind of load, I selected all the nodes contained in this line and applied the same load to them. This causes a very high stress produced at this nodes, Figure 2.14, due to the forces being applied in a point, not an area. When we look at points sufficiently far from the points of application, loads are well distributed all along the model but this peak in stress will produce problems of convergence when an elasto-plastic model is used.

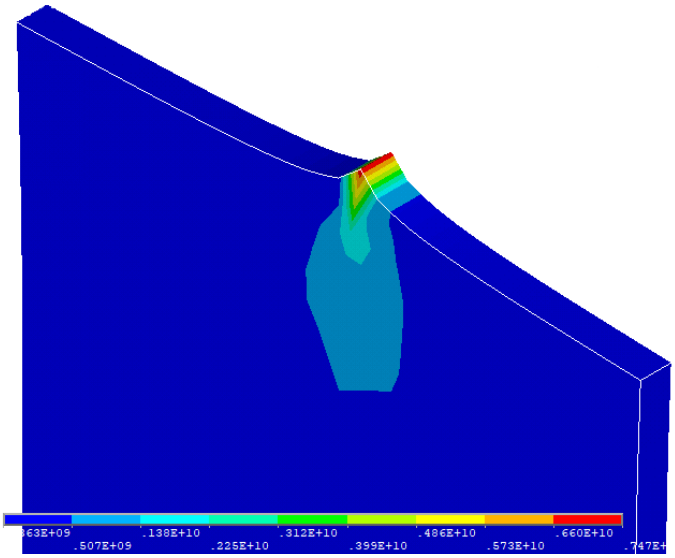

Figure 2.14 High stress produced in the line of application of loads.

2.5.4 Free face boundary conditions

In single, both U-shape and semicircular models, we need to apply some boundary conditions to the free face, the one opposite to the notched face. In particular, to avoid rigid solid movement, we need to constrain movement in direction X, Y and Z. For Y and Z there is no particular problem because, due to symmetry, they are easy to apply, but for direction X we need to find a proper way to constrain it. The easiest way is to impede movement of one node of the free face. I have analyzed two situations, one with a whole line of the free face constrained Figure 2.15, and other with just one node constrained Figure 2.16. There is not much difference between the results obtained from both models as the plate's width is very thin, but it is one aspect to consider when designing different geometries.

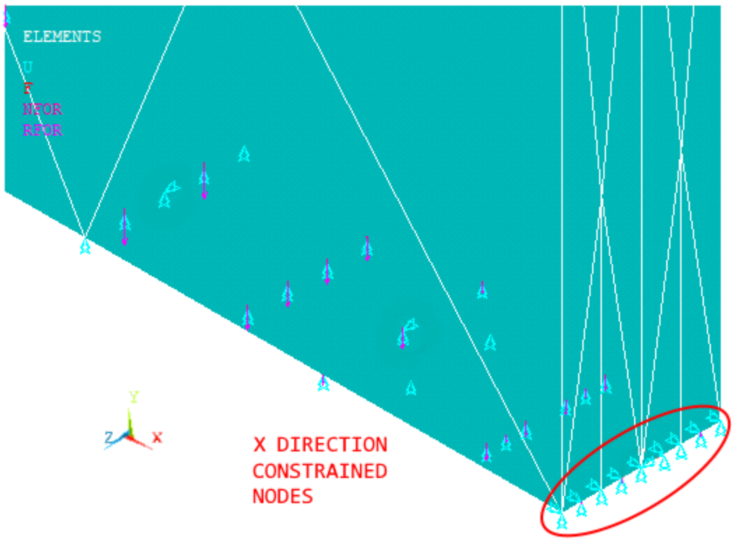

Figure 2.15 Constraints at the corner of the minimum section with the unnotched face. Whole line.

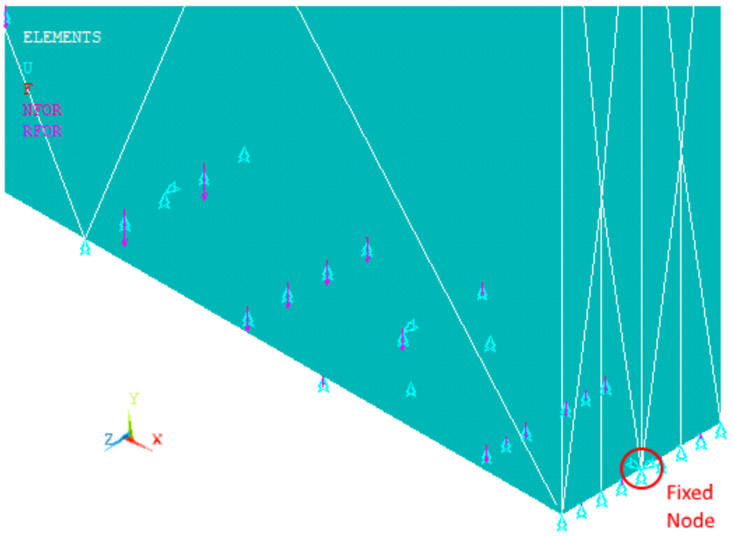

Figure 2.16 Constraints at the corner of the minimum section with the unnotched face. Just one node.

3 Stress concentration factors comparison

3.1 Stress concentration factors in opposite U-shaped notched plates

Once the models are defined, it is time to obtain results and compare them to the ones that appear in handbooks. For this, it is highly important to understand and correctly apply the boundary conditions that are specified in the handbook's model. For this first example, I will be gathering results to compare them with the following chart (Figure 3.1). This chart gives K_σ in terms of the plate's ratio $\frac{r}{d}$, where r is the radius of notch and d is the width of the minimum section. I performed several analysis with different models presenting different $\frac{r}{d}$ ratios. What I did is to fix the total depth of the U-shape notch and then reduce the radius several times, saving these results in a data file and then calculated their K_σ using MATLAB, as showed in Code 3.1. After introducing data from ANSYS and running the code, the results are shown in Figure 3.4.

If we compare the data given by Peterson [1] with the results we have obtained from our models (Figure 3.4), we can see that they are very similar, this proves our model is correct. We can also compare our results to other results given by other authors like Baratta & Neal [3] or Heywood [4] (Figure 3.5) whose expressions to calculate stress concentration factors in this kind of notch also appear in *Peterson's Stress Concentration Factors, Third Edition* [1].

In this comparison (Figure 3.5) we can see that all K_σ seem to have a similar values for the same ratio r/d, the mean error between our ANSYS model and Baratta & Neal's expression being 3%. However, in order to obtain even more accurate results, I have traced back Baratta & Neal's expression to their original article. In this article [3], they show how they created their empirical expression from comparison between experiments and analytical solutions for a semi-infinite plate presenting opposite U-shape notches. They created a chart comparing their results and other author's work, Figure 3.2. I have done the same, comparing my results with Baratta & Neal [3] and Inglis [5] expressions for different models, reducing the total depth of the U-shaped notch from 30% to 10% and 5% of the total base of the plate, approaching to a semi-infinite plate, but this time applying a distributed load at the upper face of the plate, as the article does (Figure 3.6).

3.2 Stress concentration factors in single U-shaped notched plates

For this section, I decided to compare the data obtained from ANSYS with the data given by Baratta & Neal [3] and the equation given by Inglis [5], Figure 3.2. In their article, they use a uniformly distributed load in a semi-infinite plate so I have created models with different notch depths, varying from 30 % to 5 % of the plate's width, and have applied a distributed load at the boundary faces. The results are shown in Figure 3.7. We can see that K_σ at the notch with the biggest depth is higher than the rest, which are very similar to each other and to Baratta & Neal [3] and Inglis [5] data. This can be caused by the effect of boundary conditions applied at the opposite face to the notch. This notch's depth being at 30 % of the total width of the plate makes the effect of the notch affect this face because it is not 3 times the depth of the notch away from it, the recommended distance to avoid this effect. The other two curves do adapt fairly to the data from the articles, which make these models perform as desired.

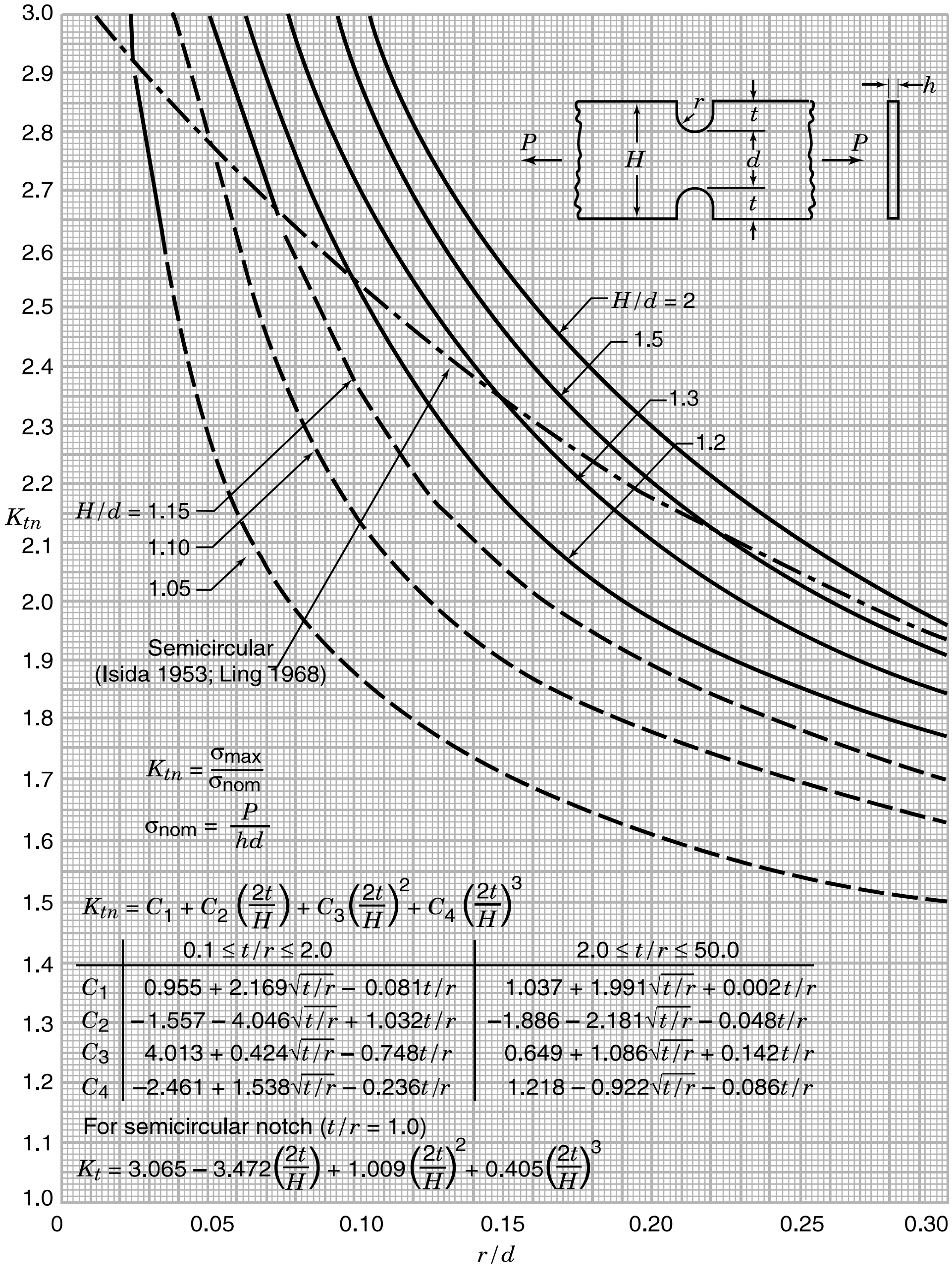

Figure 3.1 Stress concentration factors for a flat tension plate with opposite U-shape notches. Image from *Peterson's Stress Concentration Factors, Third Edition.* [1].

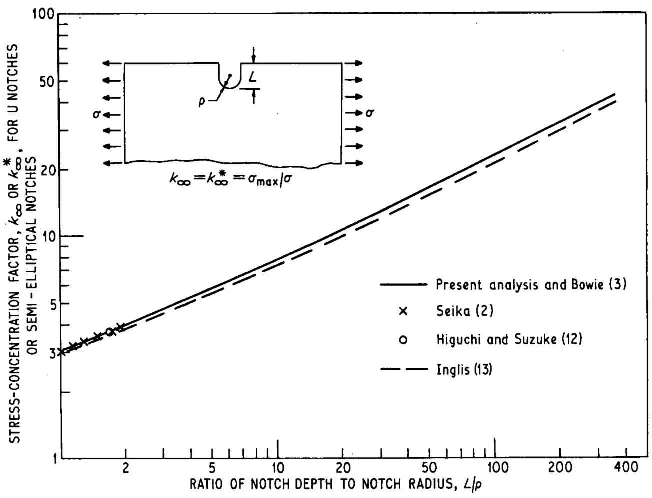

Figure 3.2 Chart from Baratta & Neal article [3].

3.3 Stress concentration factors in opposite semicircular notched plates

This case is a particularization of an opposite U-shaped notched plate, when $t/r = 1$, but it presents a main advantage, it has a semi-analytical solution. Ling [6] and Maunsell [7] deduced formulas from an analytical study of this geometry, making some necessary interpretations and simplifications. In this case, due to the range of the parameters studied in this project, I decided to compare the data obtained from the ANSYS models with Ling's expression [6], Figure 3.3, and with the chart provided by Peterson [1], Figure 3.1. These results are shown in Figure 3.8.

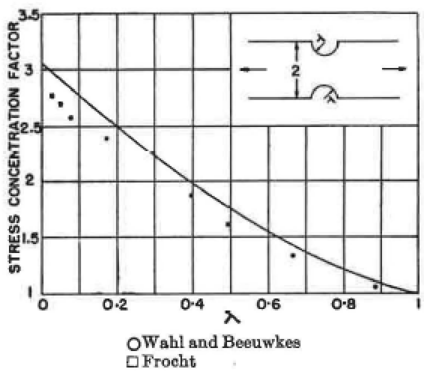

Figure 3.3 Chart from Ling's article [6].

I did the same analysis for a load applied at the center line of the minimum section, the one in Figure 3.3, and applying a uniform load. This change in the way of loading affects drastically the stress concentration produced at the notch. The reason for this is the momentum created by the uniform load at the peak of the notch due to the diminution of section, this makes K_σ for a uniform load to be greater than for the load applied at the center of the minimum section, as shown in Figure 3.9.

3.4 Stress concentration factors in single semicircular notched plates

This case is similar to the previous one, but with only a single semicircular notch. For this geometry I have compared the results obtained from the models with the data given by Peterson [1], Figure 3.10, and I have also made a comparison with different ways of loading, as I did for the previous case, Figure 3.11

3.5 Issues with Pilkie's interpolation expressions

[1] When comparing the data obtained from the FE models with Peterson's results [1], I have found that there is a big discrepancy between the information that Peterson gives in his original handbooks [8], plotted in charts, and the interpolation expressions that Pilkie [1] offers for these plots that appear in Figure 3.1. These discrepancies appear, at least, for K_σ at single and opposite U-shaped notches, and can be seen in Figure 3.12 and Figure 3.13 respectively. The error, in %, produced in reference to Peterson's data can be seen in the tables 3.1 and 3.2:

Table 3.1 Error table for Pilkie's interpolation of opposite U-shaped notches.

	$H/d = 1.05$	$H/d = 1.10$	$H/d = 1.15$	$H/d = 1.20$	$H/d = 1.30$	$H/d = 1.50$	$H/d = 2.00$
$r/d = 0.10$	0.5305	1.1664	4.3877	6.7344	7.5541	8.6806	10.1544
$r/d = 0.15$	1.0883	0.5512	2.7686	4.2633	7.6440	7.7483	8.6763
$r/d = 0.20$	1.5757	1.7244	2.5698	3.5332	6.1570	7.3548	9.2938
$r/d = 0.25$	1.8402	2.1503	2.7186	3.7571	5.3193	7.0790	9.1205
$r/d = 0.30$	2.8176	2.8786	2.8965	4.3797	5.9895	6.9514	7.6625
Average	1.5704	1.6942	3.0682	4.5336	6.5328	7.5628	8.9815

Table 3.2 Error table for Pilkie's interpolation of single U-shaped notches.

	$H/d = 2.00$	$H/d = 1.50$	$H/d = 1.20$	$H/d = 1.10$
$r/d = 0.10$	8.2826	8.9642	3.2888	6.3841
$r/d = 0.15$	8.6869	9.5690	3.0743	4.5167
$r/d = 0.20$	8.3483	9.5271	1.8798	2.4996
$r/d = 0.25$	8.7008	10.0792	0.7766	0.9261
$r/d = 0.30$	8.7940	9.3465	1.0150	0.8803
Average	8.5625	9.4972	2.0069	3.0414

If we compare the expressions for the case of semicircular notch ($r/d = 1$), Pilkie's interpolation [1] seems to be accurate for the case of single semicircular notch, Figure 3.14, and somewhat better for the case of opposite semicircular notch, Figure 3.15.

Due to these mistaken interpolating expressions, I decided to create my own using MATLAB. By picking points from Peterson's charts [8] and interpolating them, I created an expression for each one of these curves. So, for each relation of H/d I offer an expression depending on the r/d ratio.

Expression for Single Semicircular:

$$K_\sigma = 3.0743 - 7.2303\frac{r}{d} + 4.8645(\frac{r}{d})^2 + 14.8185(\frac{r}{d})^3$$

Expression for Opposite Semicircular:

$$K_\sigma = 3.0571 - 5.8969\frac{r}{d} + 8.4157(\frac{r}{d})^2 - 3.8922(\frac{r}{d})^3$$

3.6 Stress concentration factors in elements with different thickness

An interesting topic that can be studied with these models is how do Neuber's and Moslki-Glinka's rules adapt to elements with different thickness. I have created solid, elasto-plastic models with the exact same

[1] The codes used to create the comparisons and expressions that appear in this section appear in Section 3.8.

Table 3.3 Interpolation expressions for Opposite U-shaped notches.

H/d	Interpolation expression
1.05	$K_\sigma = 2.4780 - 8.1476\frac{r}{d} + 24.2857(\frac{r}{d})^2 - 26.6667(\frac{r}{d})^3$
1.10	$K_\sigma = 3.1600 - 14.6190\frac{r}{d} + 51.7143(\frac{r}{d})^2 - 66.6667(\frac{r}{d})^3$
1.15	$K_\sigma = 3.6620 - 18.3714\frac{r}{d} + 63.4286(\frac{r}{d})^2 - 80.0000(\frac{r}{d})^3$
1.20	$K_\sigma = 4.0820 - 21.9381\frac{r}{d} + 75.4286(\frac{r}{d})^2 - 93.3333(\frac{r}{d})^3$
1.30	$K_\sigma = 3.9120 - 15.4452\frac{r}{d} + 38.5714(\frac{r}{d})^2 - 33.3333(\frac{r}{d})^3$
1.50	$K_\sigma = 4.6120 - 23.8381\frac{r}{d} + 77.4286(\frac{r}{d})^2 - 93.3333(\frac{r}{d})^3$
2.00	$K_\sigma = 5.1960 - 30.0810\frac{r}{d} + 104.2857(\frac{r}{d})^2 - 133.3333(\frac{r}{d})^3$

Table 3.4 Interpolation expressions for single U-shaped notches.

H/d	Interpolation expression
1.10	$K_\sigma = 3.6800 - 17.4333\frac{r}{d} + 56.0000(\frac{r}{d})^2 - 66.6667(\frac{r}{d})^3$
1.20	$K_\sigma = 3.7420 - 17.0786\frac{r}{d} + 52.5714(\frac{r}{d})^2 - 60.0000(\frac{r}{d})^3$
1.50	$K_\sigma = 3.6900 - 15.7262\frac{r}{d} + 44.8571(\frac{r}{d})^2 - 46.6667(\frac{r}{d})^3$
2.00	$K_\sigma = 3.6900 - 15.7262\frac{r}{d} + 44.8571(\frac{r}{d})^2 - 46.6667(\frac{r}{d})^3$

parameters but different thickness, varying for every model the parameter t/r from 1 to 9, and compared the values obtained from them with the results taken from plane stress and plane strain analysis. Theoretically, the thinner the plate is, the closer these values must get to plane stress results. On the contrary, the thicker it is, the closer they must be to plane strain results. The values obtained appear in Figure 3.16 and they show little variation when plates are very thick, for example, the curves from the models with ratios H/h equal to 1 and 2 are very close to the values obtained for the Plane Strain model, this means that plates with a ratio H/h close to or lower to 2 can be consider as to be in Plane Strain. The application of Neuber's and Moslki-Glinka's rules are shown in Section 4.5.

3.7 Charts

All charts commented above are included in this section.

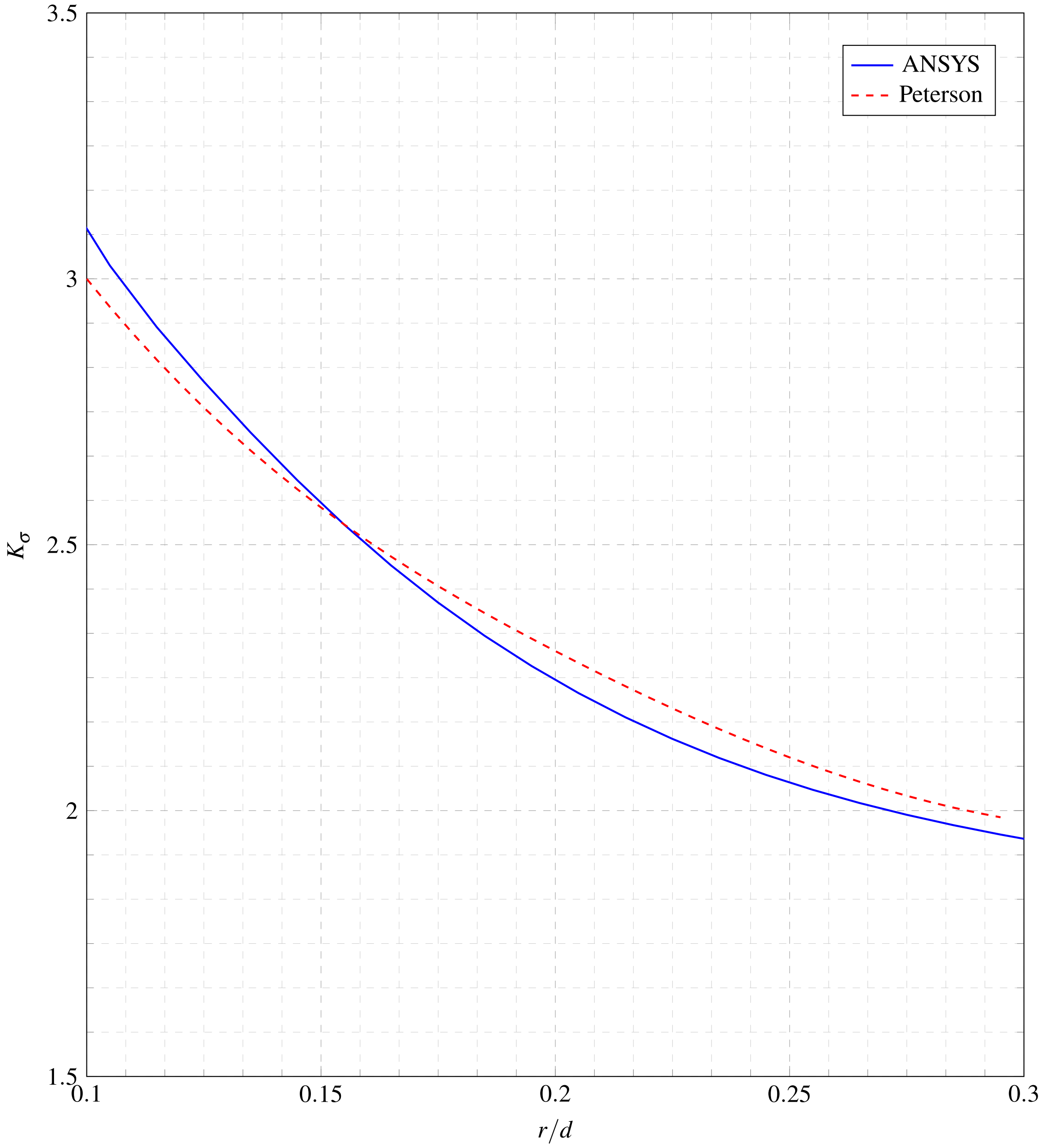

Figure 3.4 Comparison between K_σ at an U-shape notch in one side of a flat tension plate taken from different methods.

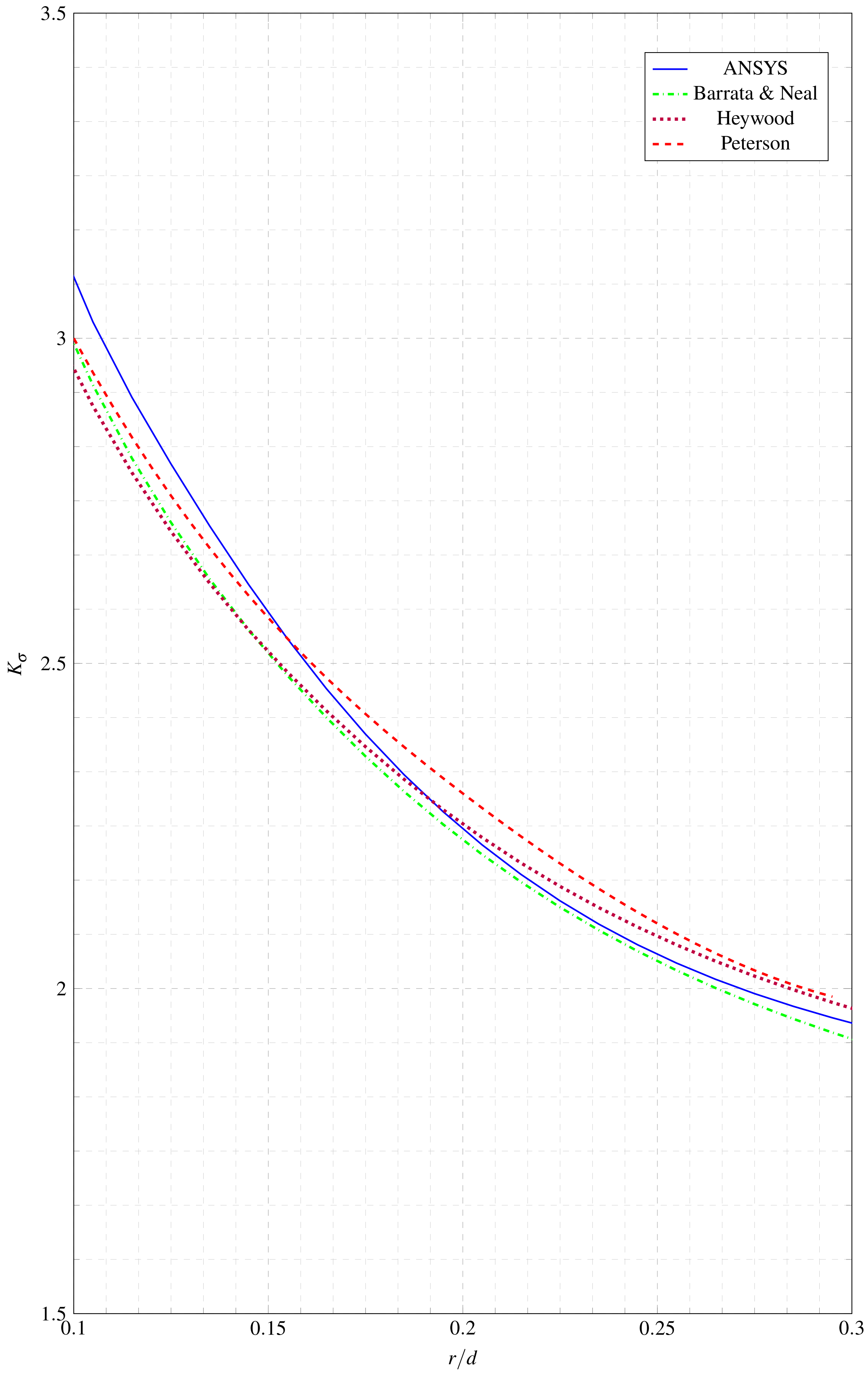

Figure 3.5 Comparison between K_σ at an U-shape notch in one side of a flat tension plate taken from different methods.

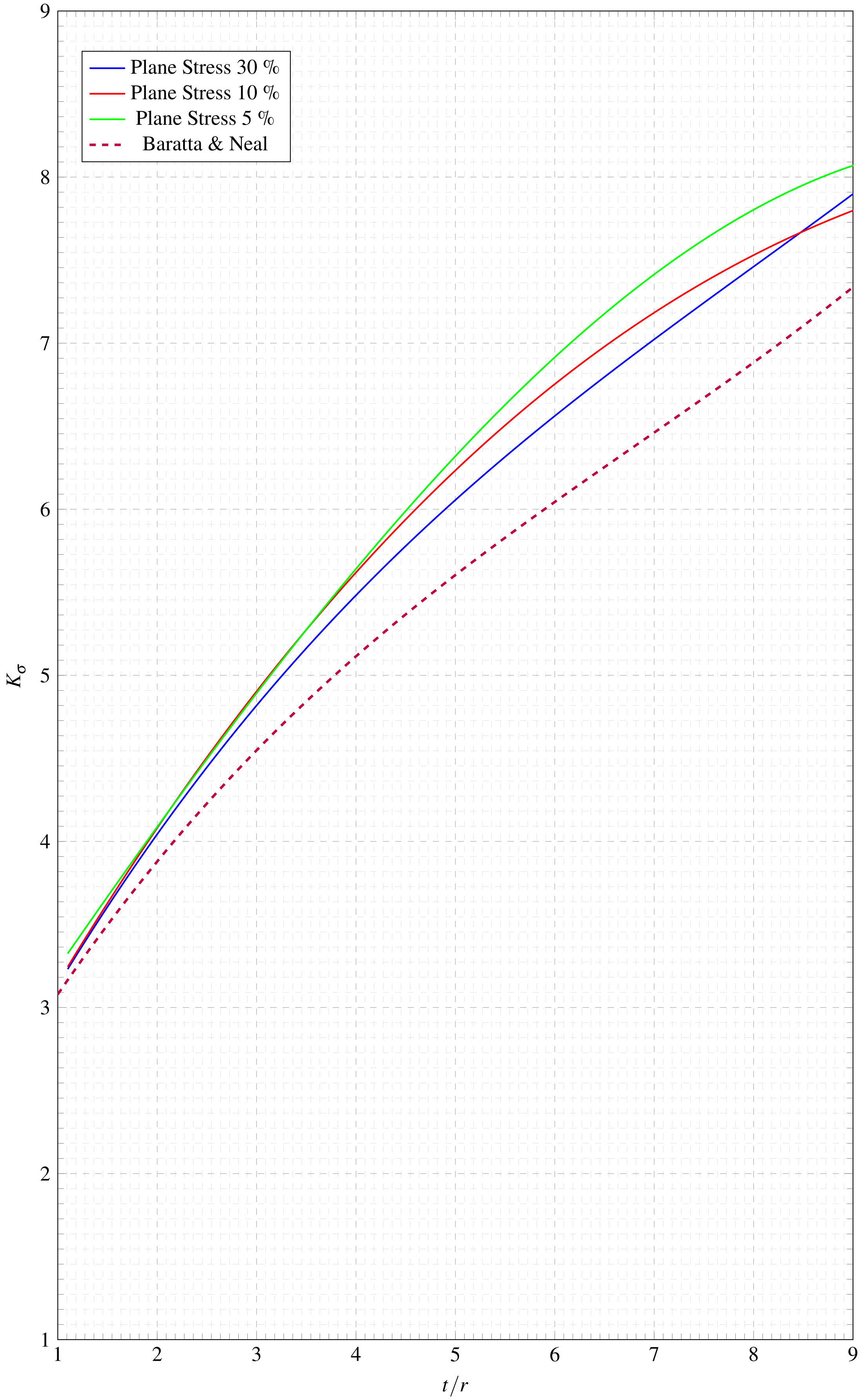

Figure 3.6 Comparison between K_σ in an opposite U-shaped notched plate.

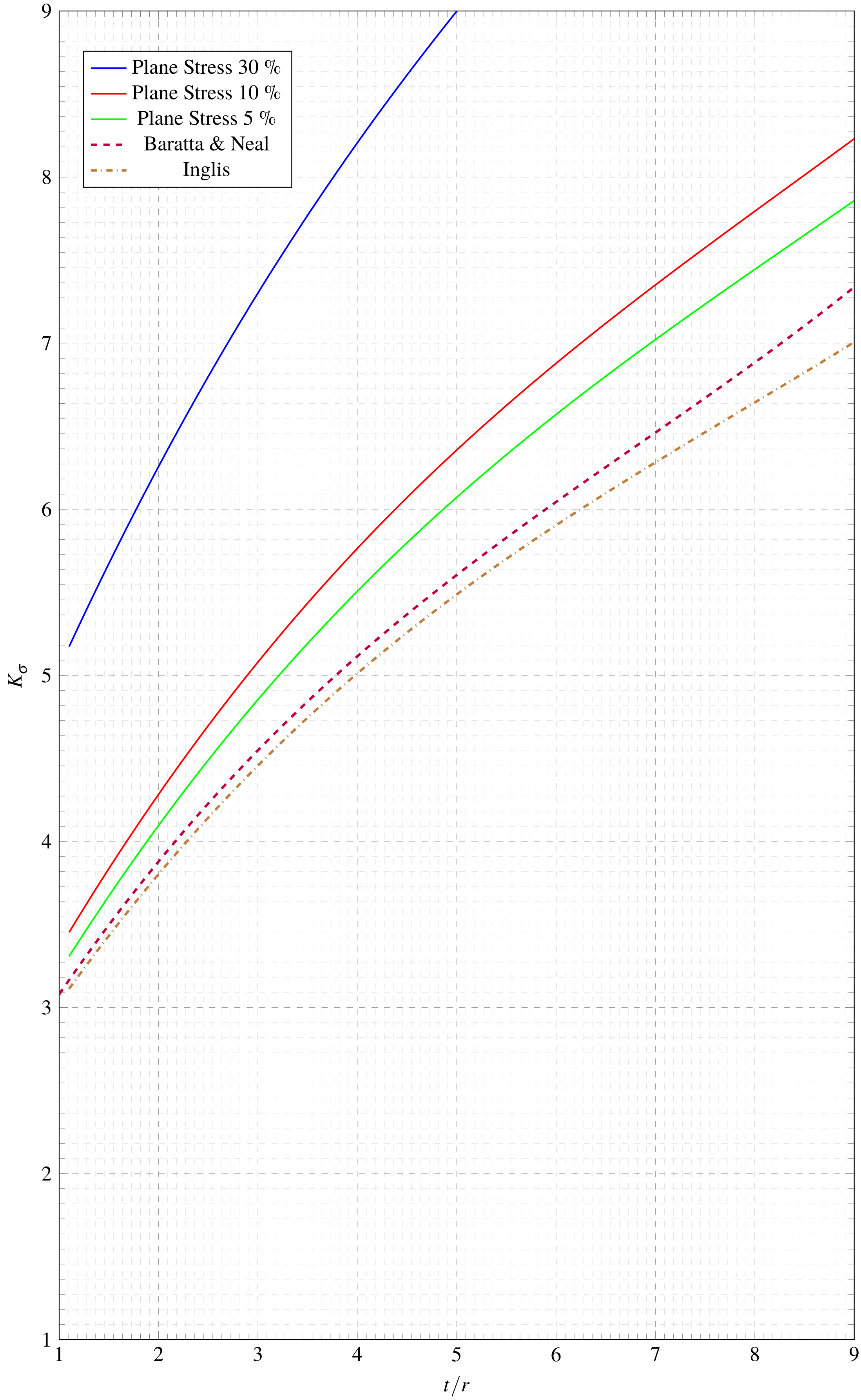

Figure 3.7 Comparison between K_σ in a single U-shaped notched plate.

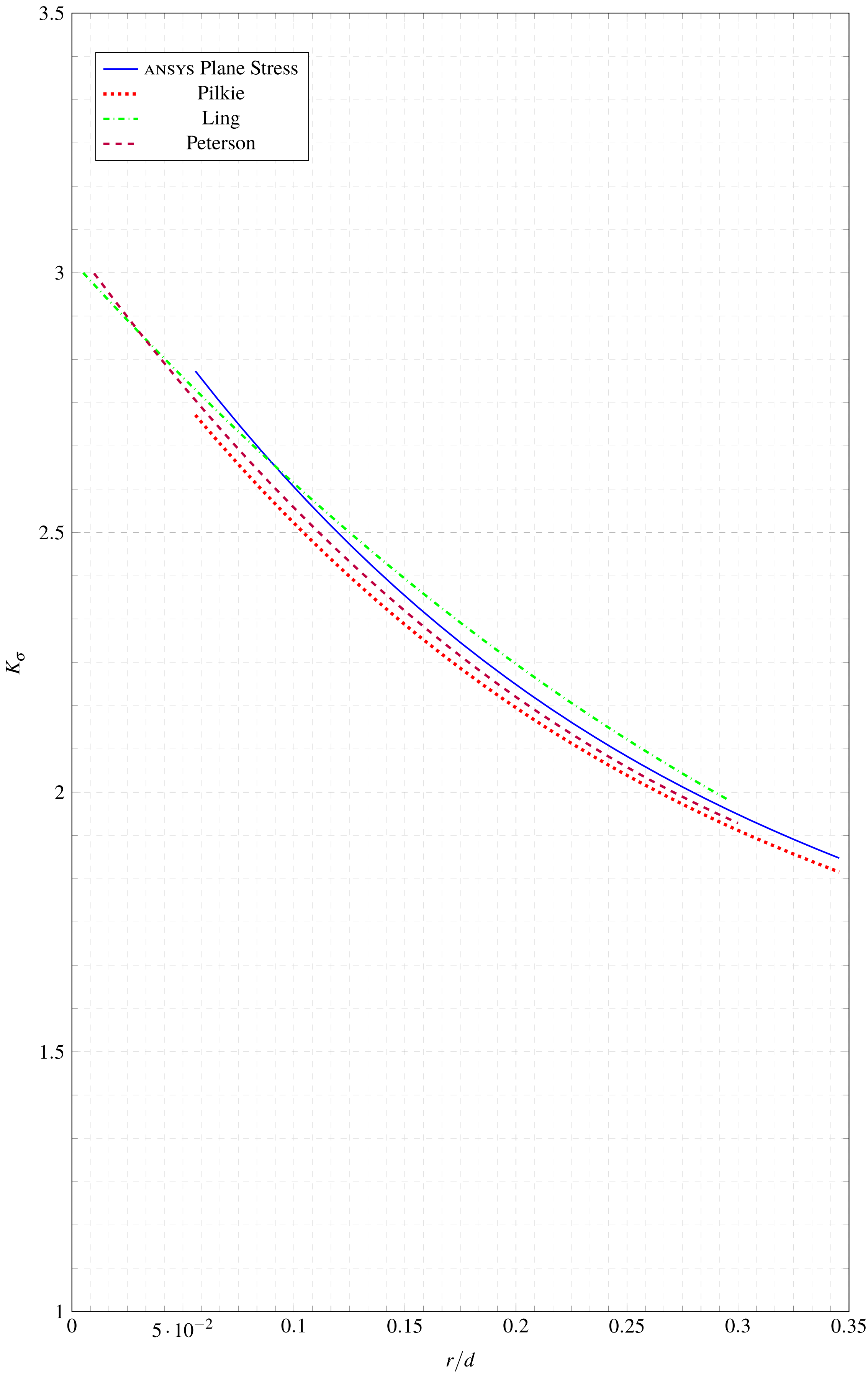

Figure 3.8 Comparison between K_σ in an opposite semicircular notched plate.

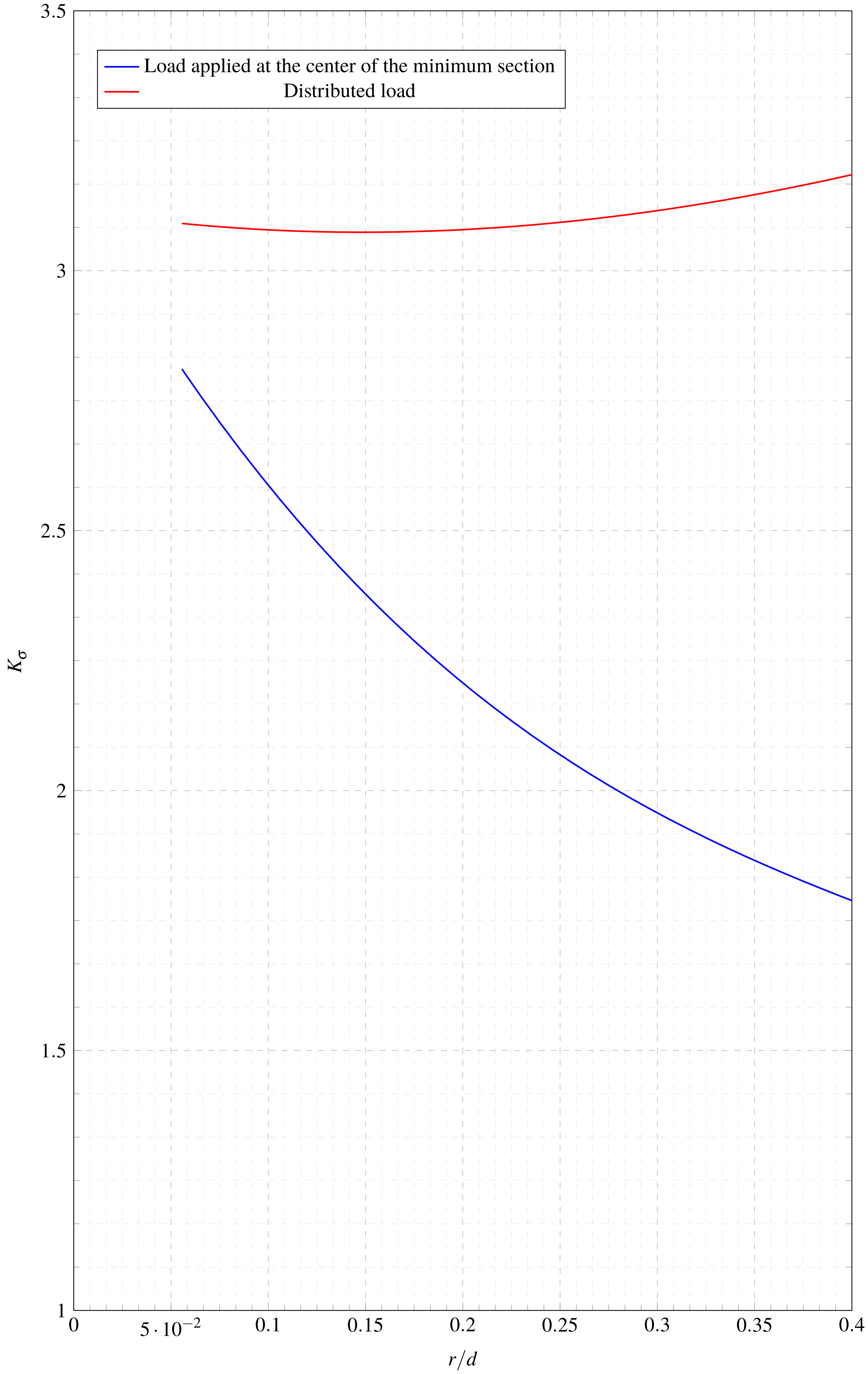

Figure 3.9 Comparison between K_σ in an opposite semicircular notched plate with a uniform load and a load applied at the center of the minimum section.

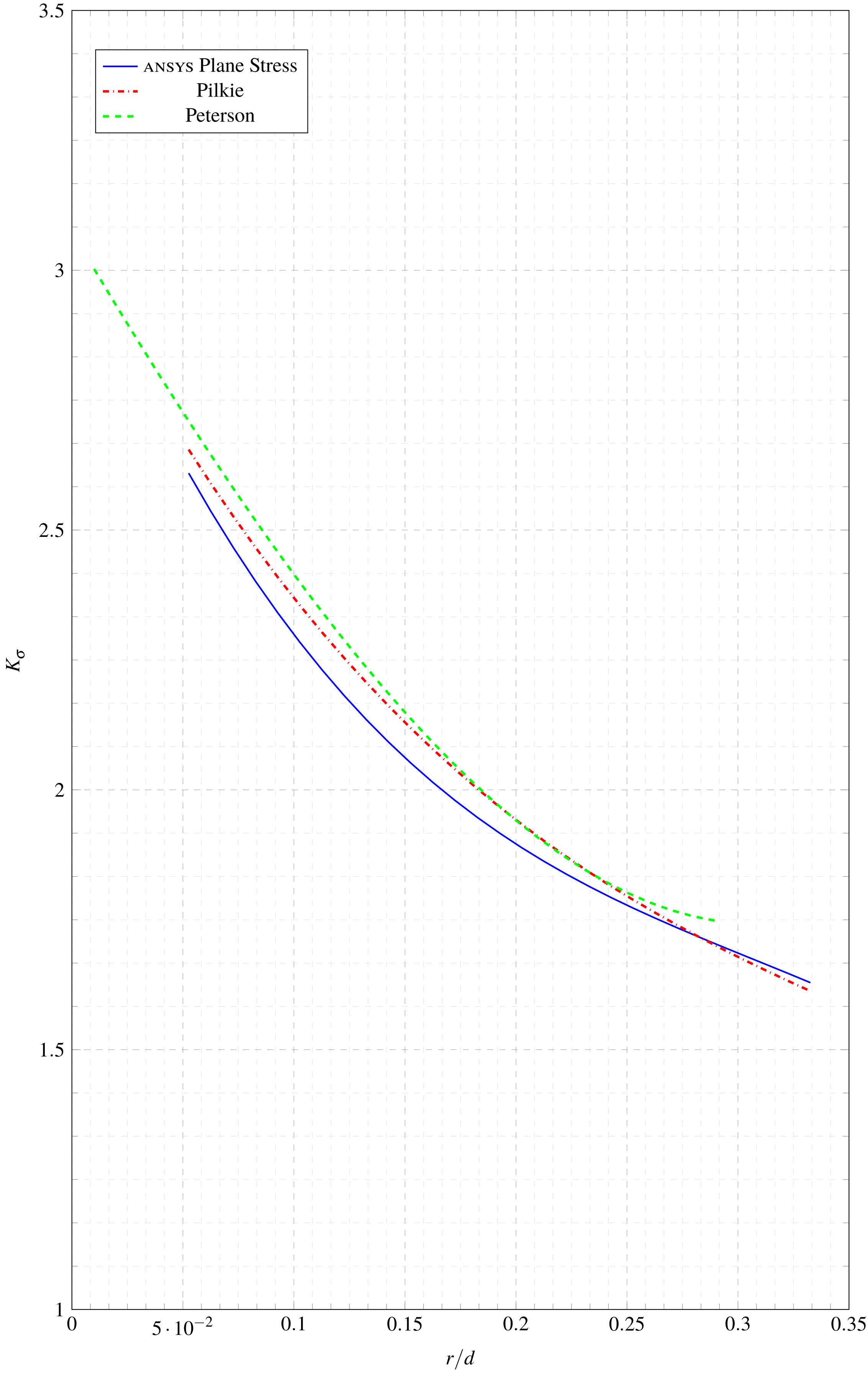

Figure 3.10 Comparison between K_σ in a single semicircular notched plate.

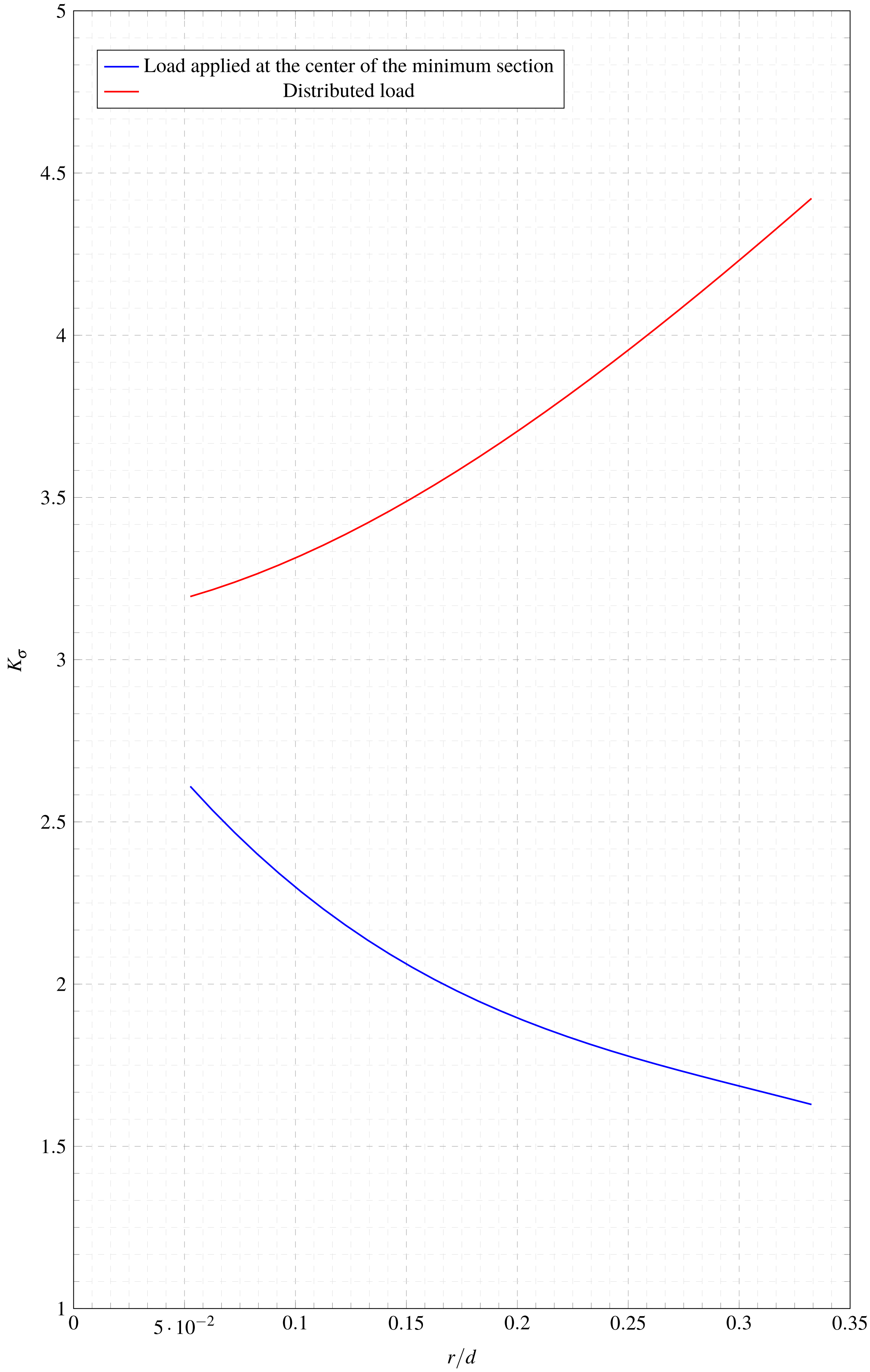

Figure 3.11 Comparison between K_σ in a single semicircular notched plate with a uniform load and a load applied at the center of the minimum section.

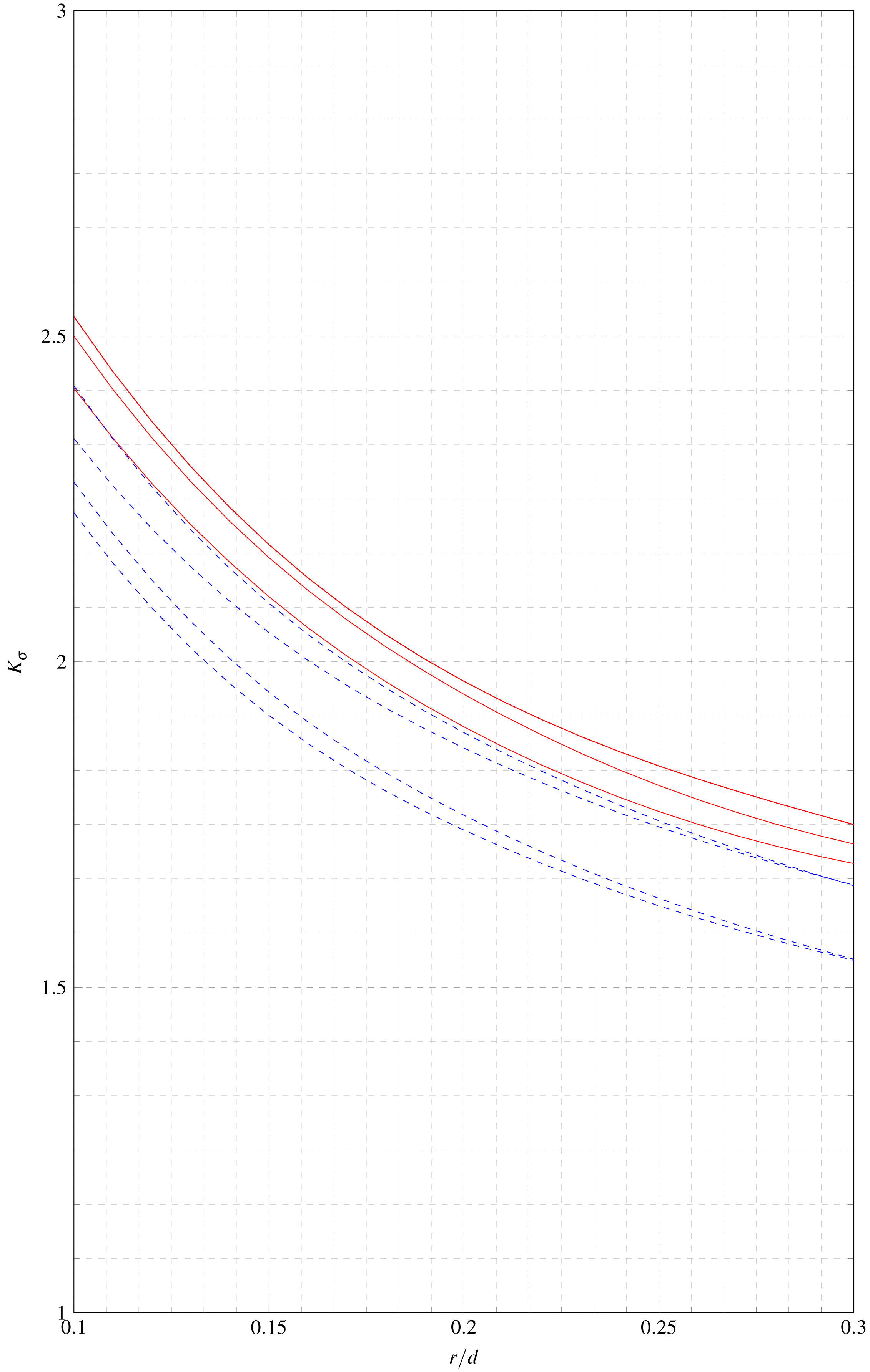

Figure 3.12 Comparison between K_σ at a single U-shape notch obtained by Peterson (in red) and values from Pilkie's interpolation expression (in blue).

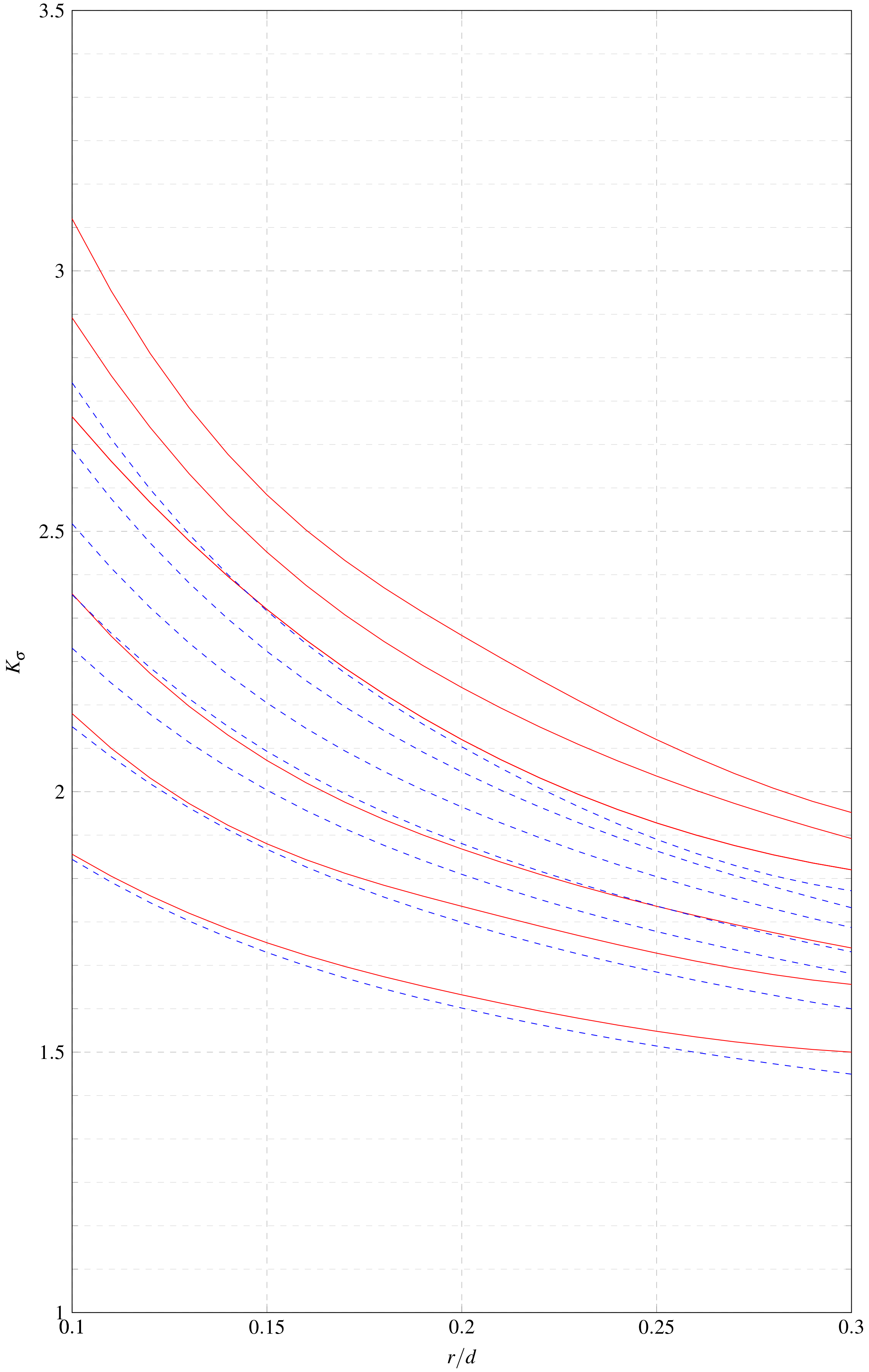

Figure 3.13 Comparison between K_σ at a opposite U-shape notch obtained by Peterson (in red) and values from Pilkie's interpolation expression (in blue).

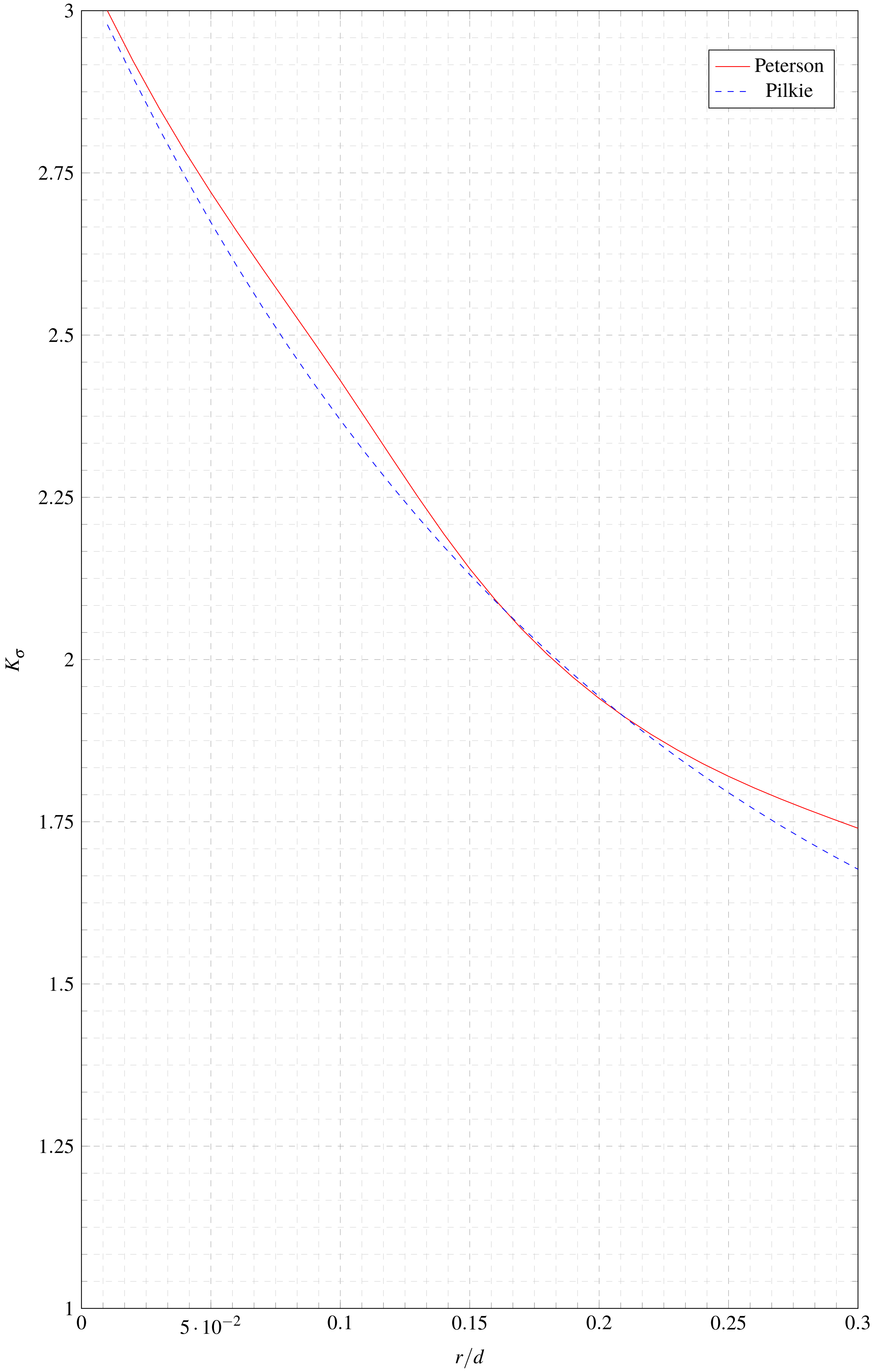

Figure 3.14 Comparison between K_σ at a single semicircular notch obtained by Peterson (in red) and values from Pilkie's interpolation expression (in blue).

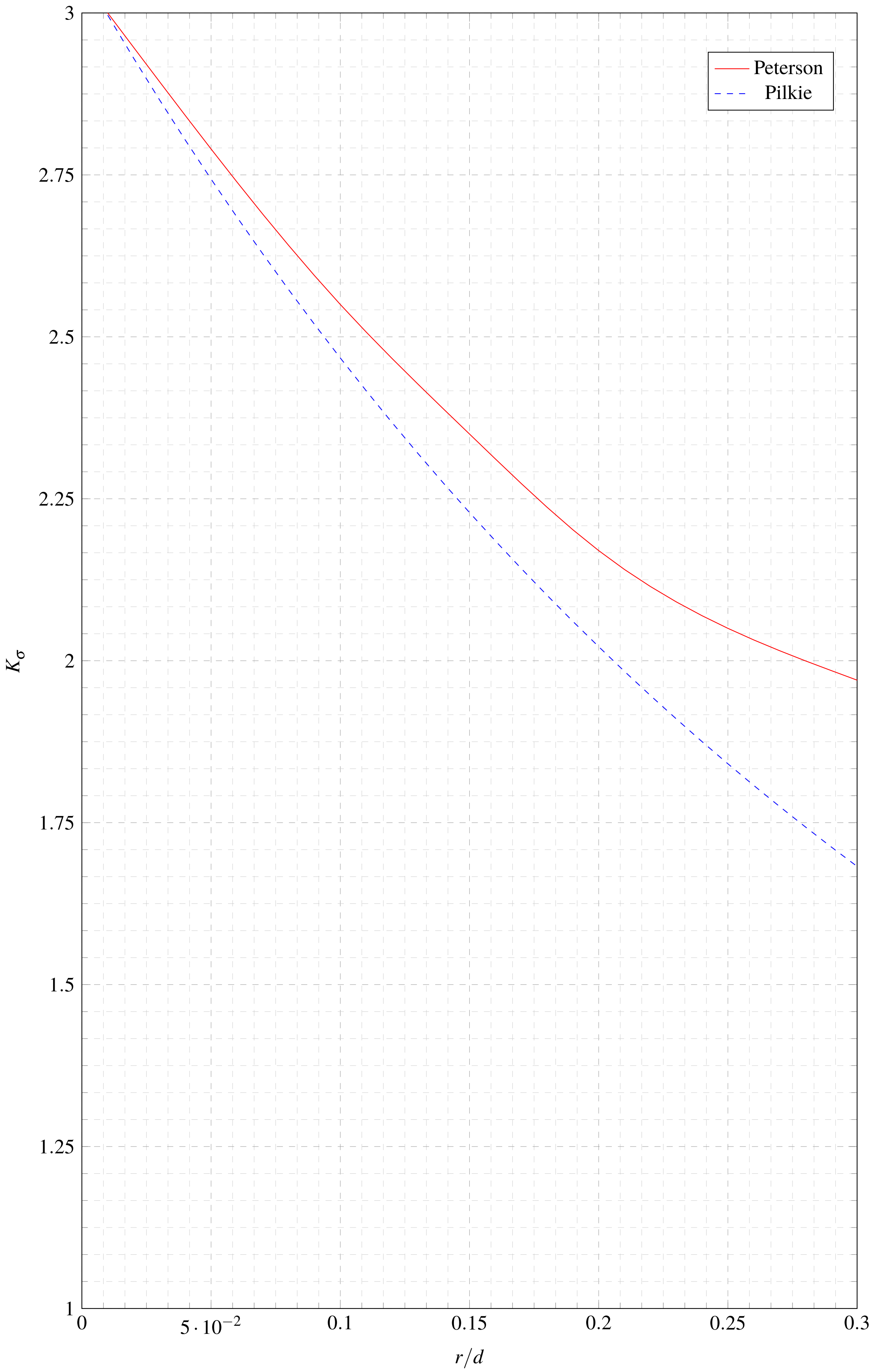

Figure 3.15 Comparison between K_σ at a opposite semicircular notch obtained by Peterson (in red) and values from Pilkie's interpolation expression (in blue).

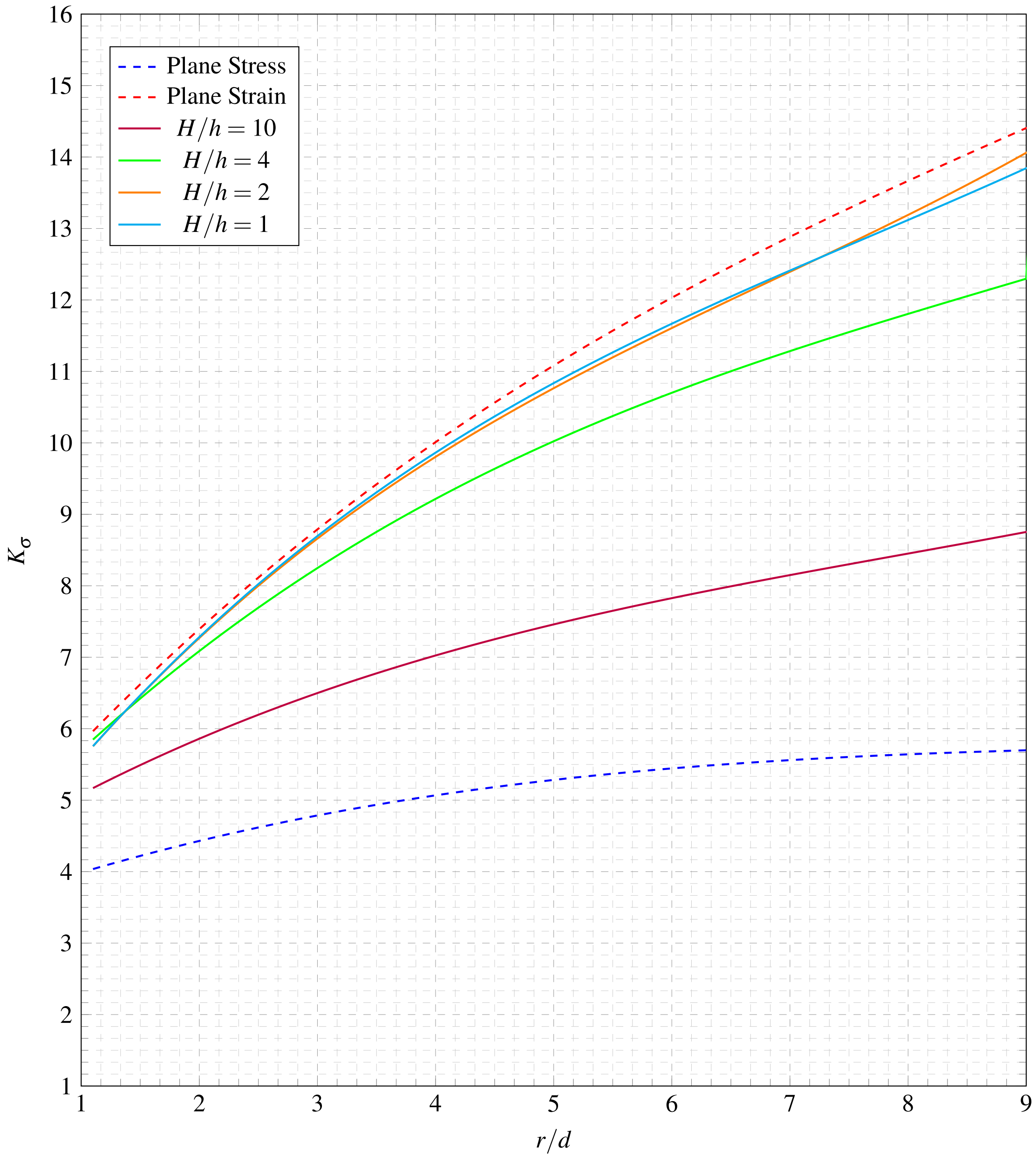

Figure 3.16 Comparison between K_σ at a U-shaped notch in one side of a flat tension plate with different thickness.

3.8 Codes

Code 3.1 MATLAB code to calculate stress concentration factor in different models.

```
E=202375e6;
n=0.211;
K=1283e6;

H=0.127;   % WIDTH
h=0.00254; % DEPTH
P=4e4; % LOAD

B=0.3048;
T=0.1*H;

hr=[1.1 2.1 3.1 4.1];
R=T./hr;

d=zeros([1 length(R)]);
ratio=zeros([1 length(R)]);
Pn=zeros([1 length(R)]);

for i=1:length(R)
   d(i)=H-T;
   ratio(i)=R(i)/d(i);
   Pn(i)=P/(h*d(i));
end


%% PETERSONS RESULTS
Ktn=zeros([1 length(R)]);
for i=1:length(R)
    if 0.5<=T/R(i)<2
        C1=0.907+2.125*sqrt(T/R(i))+0.023*T/R(i);
        C2=0.710-11.289*sqrt(T/R(i))+1.708*T/R(i);
        C3=-0.672+18.754*sqrt(T/R(i))-4.046*T/R(i);
        C4=0.175-9.759*sqrt(T/R(i))+2.365*T/R(i);
    end

    if 2<=T/R(i)<20
        C1=0.953+2.136*sqrt(T/R(i))-0.005*T/R(i);
        C2=-3.255-6.281*sqrt(T/R(i))+0.068*T/R(i);
        C3=8.203+6.893*sqrt(T/R(i))+0.064*T/R(i);
        C4=-4.851-2.793*sqrt(T/R(i))-0.128*T/R(i);
    end

Ktn(i)=C1+C2*(R(i)/H)+C3*(R(i)/H)^2+C4*(R(i)/H)^3;
end

%% RESULTS FROM ANSYS
Kte=zeros([1 length(R)]);
S=[3244.185e5 4762.184e5 5844.532e5 6725.240e5];
for i=1:length(R)
   Kte(i)=S(i)/Pn(i);
end
```

```
%% PLOT
figure(1)
plot(ratio,Ktn,ratio,Kte)
legend('Kt Petersons','Kt ANSYS')
title('Stress concentration factor at U-shape notch')
xlabel('r/d')
ylabel('Kt')

figure(2)
KCB=[2.4 2.9 3.4 3.7];
plot(hr,S./(P/((H-T)*h)), hr, KCB)
legend('ANSYS', 'C&B aprox.')
title('Cole & Brown')
xlabel('h/r')
ylabel('fm/P/(D-h)t')
```

Code 3.2 MATLAB code to compare Pilkie's expression for opposite U-shape notch with Peterson's chart.

```
rd=[0.1 0.15 0.2 0.25 0.3];
Hd=[1.05 1.10 1.15 1.2 1.3 1.5 2];

H=1;
h=0.0254;
B=0.3048;

ratioSP=rd(1):0.01:rd(length(rd));

Ktn=zeros([length(rd) length(Hd)]);
t=zeros([1 length(Hd)]);
d=zeros([1 length(Hd)]);
r=zeros([length(rd) length(Hd)]);

for j=1:length(Hd)
    d(j)=H/Hd(j);
    t(j)=H-d(j);
    for i=1:length(rd)
        r(i,j)=d(j)*rd(i);
        if 0.1<=(t(j)/r(i,j))<2
            C1=0.955+2.169*sqrt(t(j)/r(i,j))-0.081*t(j)/r(i,j);
            C2=-1.557-4.046*sqrt(t(j)/r(i,j))+1.032*t(j)/r(i,j);
            C3=4.013+0.424*sqrt(t(j)/r(i,j))-0.748*t(j)/r(i,j);
            C4=-2.461+1.538*sqrt(t(j)/r(i,j))-0.236*t(j)/r(i,j);
        end

        if 2<=(t(j)/r(i,j))<50
            C1=1.037+1.991*sqrt(t(j)/r(i,j))+0.002*t(j)/r(i,j);
            C2=-1.886-2.181*sqrt(t(j)/r(i,j))-0.048*t(j)/r(i,j);
            C3=0.649+1.086*sqrt(t(j)/r(i,j))+0.142*t(j)/r(i,j);
            C4=1.218-0.922*sqrt(t(j)/r(i,j))-0.086*t(j)/r(i,j);
        end

    Ktn(i,j)=C1+C2*(2*t(j)/H)+C3*(2*t(j)/H)^2+C4*(2*t(j)/H)^3;
    end
```

```
end

KTN=zeros([length(ratioSP) length(Hd)]);
for j=1:length(Hd)
    KTN(:,j)=spline(rd,Ktn(:,j),ratioSP);
end

Peter(7,:)=[3.1 2.58 2.3 2.1 1.98];
Peter(6,:)=[2.9 2.46 2.2 2.02 1.91];
Peter(5,:)=[2.71 2.35 2.1 1.94 1.85];
Peter(4,:)=[2.55 2.17 1.96 1.85 1.77];
Peter(3,:)=[2.37 2.05 1.89 1.78 1.7];
Peter(2,:)=[2.15 1.9 1.76 1.69 1.64];
Peter(1,:)=[1.87 1.71 1.61 1.54 1.5];

KTP=zeros([length(ratioSP) length(Hd)]);
for j=1:length(Hd)
    KTP(:,j)=spline(rd,Peter(j,:),ratioSP);
end

Muestra_Opposite=[KTN KTP];
Table_1 = table(ratioSP',Muestra_Opposite(:,1),Muestra_Opposite(:,2),
    Muestra_Opposite(:,3),Muestra_Opposite(:,4),Muestra_Opposite(:,5),
    Muestra_Opposite(:,6),Muestra_Opposite(:,7),Muestra_Opposite(:,8),
    Muestra_Opposite(:,9),Muestra_Opposite(:,10),Muestra_Opposite(:,12),
    Muestra_Opposite(:,12),Muestra_Opposite(:,13),Muestra_Opposite(:,14));

for j=1:7
       plot(ratioSP,KTN(:,j),'b--');
       hold on
       plot(ratioSP,KTP(:,j),'r');
       hold on
end
xlabel('r/d')
ylabel('Ktn')
hold off
```

Code 3.3 MATLAB code to compare Pilkie's expression for single U-shape notch with Peterson's chart.

```
H=1;
h=0.0254;
B=0.3048;

rd=[0.1 0.15 0.2 0.25 0.3];
Hds=[2 1.5 1.2 1.1];

ts=zeros([1 length(Hd)]);
ds=zeros([1 length(Hd)]);
rs=zeros([length(rd) length(Hd)]);
Ktns=zeros([1 length(rd)]);
for j=1:length(Hds)
    ds(j)=H/Hds(j);
    ts(j)=H-ds(j);
    for i=1:length(rd)
```

```
        rs(i,j)=ds(j)*rd(i);
        if 0.5<=(ts(j)/rs(i,j))<2
            C1=0.907+2.125*sqrt(ts(j)/rs(i,j))+0.023*ts(j)/rs(i,j);
            C2=0.710-11.289*sqrt(ts(j)/rs(i,j))+1.708*ts(j)/rs(i,j);
            C3=-0.672+18.754*sqrt(ts(j)/rs(i,j))-4.046*ts(j)/rs(i,j);
            C4=0.175-9.759*sqrt(ts(j)/rs(i,j))+2.365*ts(j)/rs(i,j);
        end

        if 2<=(t(j)/r(i,j))<20
            C1=0.953+2.136*sqrt(ts(j)/rs(i,j))-0.005*ts(j)/rs(i,j);
            C2=-3.255-6.281*sqrt(ts(j)/rs(i,j))+0.068*ts(j)/rs(i,j);
            C3=8.203+6.893*sqrt(ts(j)/rs(i,j))+0.064*ts(j)/rs(i,j);
            C4=-4.851-2.793*sqrt(ts(j)/rs(i,j))-0.128*ts(j)/rs(i,j);
        end

    Ktns(i,j)=C1+C2*(ts(j)/H)+C3*(ts(j)/H)^2+C4*(ts(j)/H)^3;
    end
end

KTNS=zeros([length(ratioSP) length(Hds)]);
for j=1:length(Hds)
    KTNS(:,j)=spline(rd,Ktns(:,j),ratioSP);
end


PeterS(4,:)=[2.53 2.18 1.97 1.84 1.75];
PeterS(3,:)=[2.53 2.18 1.97 1.84 1.75];
PeterS(2,:)=[2.50 2.16 1.95 1.81 1.72];
PeterS(1,:)=[2.42 2.10 1.90 1.77 1.69];

KTPS=zeros([length(ratioSP) length(Hds)]);
for j=1:length(Hds)
    KTPS(:,j)=spline(rd,PeterS(j,:),ratioSP);
end

Muestra_Single=[KTNS KTPS];
Table_2 = table(ratioSP',Muestra_Single(:,1),Muestra_Single(:,2),Muestra_Single
    (:,3),Muestra_Single(:,4),Muestra_Single(:,5),Muestra_Single(:,6),
    Muestra_Single(:,7),Muestra_Single(:,8));

figure(2)
for j=1:4
      plot(ratioSP,KTNS(:,j),'b--');
      hold on
      plot(ratioSP,KTPS(:,j),'r');
      hold on
end
xlabel('r/d')
ylabel('Ktn')
```

Code 3.4 MATLAB code to compare Pilkie's expression for single semicircular notch with Peterson's chart.

```
H=1;
h=0.0254;
```

```
B=0.3048;

rdcs=[0.01 0.05 0.1 0.15 0.2 0.25 0.3];
ratioSPS=rdcs(1):0.01:rdcs(length(rdcs));

Ktpcs=[3 2.72 2.43 2.14 1.94 1.82 1.74];
KTPCS=spline(rdcs,Ktpcs,ratioSPS);

dcs=zeros([1 length(rdcs)]);
tcs=zeros([1 length(rdcs)]);
Ktcs=zeros([1 length(rdcs)]);
for i=1:length(rdcs)
    dcs(i)=H/(1+rdcs(i));
    tcs(i)=rdcs(i)*dcs(i);
    Ktcs(i)=3.065-8.87*(tcs(i)/H)+14.036*(tcs(i)/H)^2-7.219*(tcs(i)/H)^3;
end

KTCS=spline(rdcs,Ktcs,ratioSPS);

Table_3=table(ratioSPS', KTPCS', KTCS');

figure(3)
plot(ratioSPS', KTPCS',ratioSPS', KTCS')
xlabel('r/d')
ylabel('Ktn')
legend('Peterson','Pilkie')
```

Code 3.5 MATLAB code to compare Pilkie's expression for opposite semicircular notch with Peterson's chart.

```
H=1;
h=0.0254;
B=0.3048;

rdco=[0.01 0.05 0.1 0.15 0.2 0.25 0.3];
ratioSPO=rdco(1):0.01:rdco(length(rdco));

Ktpco=[3 2.79 2.55 2.35 2.17 2.05 1.97];
KTPCO=spline(rdco,Ktpco,ratioSPO);

dco=zeros([1 length(rdco)]);
tco=zeros([1 length(rdco)]);
Ktco=zeros([1 length(rdco)]);
for i=1:length(rdco)
    dco(i)=H/(1+rdco(i));
    tco(i)=rdco(i)*dco(i);
    Ktco(i)=3.065-3.472*(2*tco(i)/H)+1.009*(2*tco(i)/H)^2+0.405*(tco(i)/H)^3;
end

KTCO=spline(rdco,Ktco,ratioSPO);

Table_4=table(ratioSPO', KTPCO', KTCO');

figure(4)
```

```
plot(ratioSPO', KTPCO',ratioSPO', KTCO')
xlabel('r/d')
ylabel('Ktn')
legend('Peterson','Pilkie')
```

4 Neuber and Molski-Glinka's rules

4.1 Ramberg-Osgood relationship

[1] In 1943, Ramberg and Osgood proposed a relationship between strain and strain for both, elastic and plastic zones, that could be summed in the same expression. Strain in the elastic zone is directly related with the Young's modulus of the material, whilst, in the plastic zone, it presents an exponential expression that depends on two parameters, H and n, that are estimated for every material.

$$\varepsilon_e = \frac{\sigma}{E}$$

$$\varepsilon_p = (\frac{\sigma}{H})^{\frac{1}{n}}$$

So, if we sum both of these strains, we get the Ramberg-Osgood expression

$$\varepsilon = \varepsilon_e + \varepsilon_p$$

$$\varepsilon = \frac{\sigma}{E} + (\frac{\sigma}{H})^{\frac{1}{n}}$$

If we give values to the three parameters, E, H and n, and plot the expression, we obtain a single and smooth curve, Figure 4.1, which is not discontinuous at the yield point.

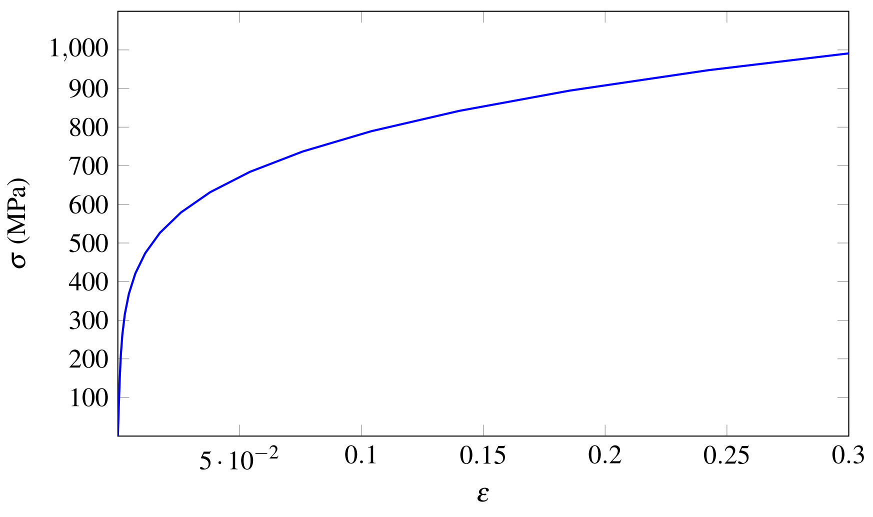

Figure 4.1 Ramberg-Osgood expression curve.

[1] The expressions showed in this section were taken from *Mechanical Behavior of Materials. Engineering Methods for Deformation, Fracture and Fatigue. Fourth Ed.* [9]

4.2 Neuber's rule

[2] Neuber's rule [11] is a mathematical relationship that relates nominal stress and strain with local stress and strain. The rule says that the product of nominal stresses and strains is proportional to the product of the local elastic plastic stresses and strains.

$$K_\sigma \cdot \sigma_n \cdot K_\sigma \cdot \varepsilon_n = \sigma \cdot \varepsilon$$

Some modifications can be made to this expression which allow to write it only in terms of nominal stress. We can assume the behavior of strain as pure elastic even though the real strain is above the yield point. Doing this, we obtain the following hyperbola expression

$$\sigma\varepsilon = \frac{(K_\sigma \sigma_n)^2}{E}$$

This relationship allow us to obtain the real local stress and strain from the nominal stress applied to the element and the K_σ produced by the notch. However, we need another expression to obtain both stress and strain, for this we use the Ramberg-Osgood expression that relates these two variables. If we obtain the point where these two curves intersect, Figure 4.2, we will obtain the real local stress and strain. Therefore, the equation system we have to solve is the following:

$$\sigma\varepsilon = \frac{(K_\sigma \sigma_n)^2}{E}$$

$$\varepsilon = \frac{\sigma}{E} + (\frac{\sigma}{H})^{\frac{1}{n}}$$

Where E, H and n are material properties.

It can be proved that this intersection is unconditionally convergent, and so, we can apply a simple algorithm to solve this system. I'll be using the algorithm published by Navarro [12], implemented in a MATLAB code (Code 4.1).

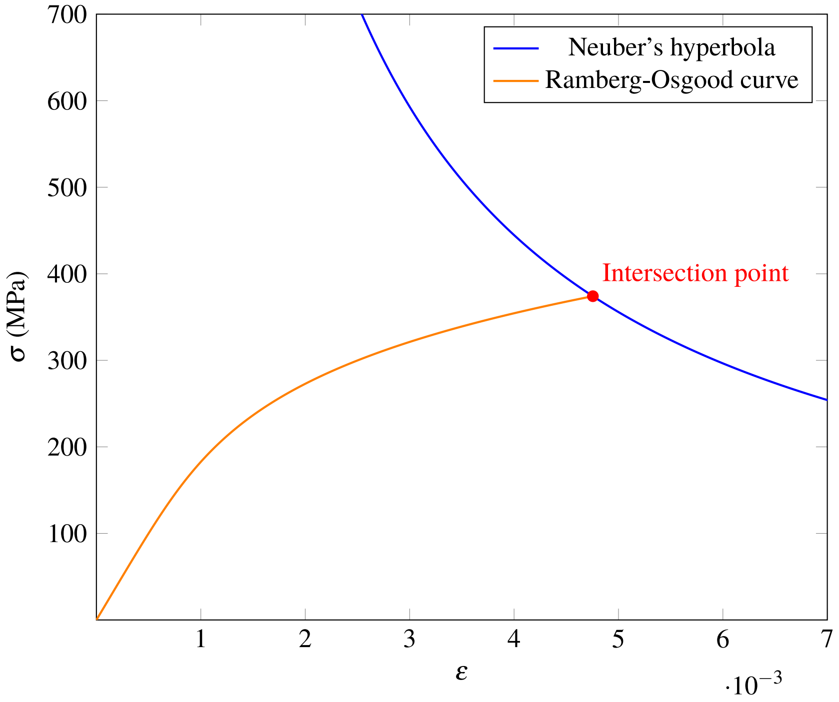

Figure 4.2 Ramberg-Osgood curve intersection with Neuber's hyperbola.

[2] Some of the information given in this section was taken from `https://www.efatigue.com/` [10]

4.3 Molski-Glinka's Rule

In 1980, Molski and Glinka [13] published a similar relationship between stress and strain, but this time it was an energy-based method which also holds when localized plastic yielding occurred at the notch root. They claimed that their method predicts local strain more accurately than Neuber's method. Molski-Glinka's method always show lower values of real stress than the ones obtained by Neuber's method. Molski-Glinka's expression can be arranged as the following [13].

$$\frac{(K_\sigma \sigma_n)^2}{2E} = \frac{\sigma^2}{2E} + \frac{\sigma}{n+1}(\frac{\sigma}{H})^{\frac{1}{n}}$$

Just as with Neuber's method, the equations in this case can also be solved with a simple algorithm very similar to that used for Neuber's method [12] (Code 4.2).

4.4 Application of Neuber's and Molski-Glinka's rules to Finite Elements models

The aim of this project is to obtain real stress and strain by applying these methods to FE models. Having already created and proved that the FE models work properly, the application of these methods is quite simple. For this I will be using again the software MATLAB. In Codes 4.1 and 4.2 appear two functions, *NeuberSe* and *MolskiGlinkaSe*, whose inputs are the material parameters E, H and n and a *pseudo-stress*, which is defined as $\sigma_e^* = K_\sigma \sigma_n$ and whose outputs are the real local stress and strain after having applied the previously mentioned unconditionally convergent algorithms.

To use these functions, we only need to know the material parameters, the value of the applied load and the K_σ produced at the notch. The material parameters used in these project are shown in 4.1, but they can be easily changed in both ANSYS and MATLAB codes. Stress concentration factors have been calculated in previous sections and the applied load has always been 10 000 000 N. Due to some problems of convergence in elasto-plastic models, that appeared when using loads applied to the center line at the middle of the minimum section, I decided to only use distributed loads in this section.

Table 4.1 Material parameters.

Young's modulus (E)	202 375 MPa
Poisson's coefficient (α)	0.3
Hardening coefficient (H)	1 283 MPa
Hardening exponent (n)	0.211

I have made charts comparing K_σ produced at the notch in different situations, a pure elastic model, an elasto-plastic model and the results obtained from Neuber's and Molski-Glinka's algorithms taking as input the stress produced in the pure elastic model. As it was expected, pure elastic results are much greater than the elasto-plastic ones when the stress produced is beyond the yield point. If stress is so low that is under the yield point, both results are equal. As for Neuber's and Molski-Glinka's results, they are also always lower than the elasto-plastic K_σ and similar in value, Molski-Glinka being slightly lower than Neuber. Charts are presented in the following section and codes to apply algorithms and create these charts appear at section "Codes". An Index of these charts and codes is shown in Table 4.2

4.5 Application of Neuber's and Moslki-Glinka's rules to elements with different thickness

Just as they have been applied in the previous sections, Neuber's and Moslki-Glinka's rules have been applied to the models mentioned in Section 3.6. The results are shown in Figure 4.11 and Figure 4.12. In this chart we can see the difference between Neuber's and Molski-Glinka's results and the values obtained from elasto-plastic models. They show an almost straight line at a 45 degree angle, which means that values obtained from the mentioned methods compared to the ones obtained from elasto-plastic models are very

Table 4.2 Chart index.

	Chart	Code
Opposite U-shape notched plate $t/H = 0.15$	Figure 4.3	Code 4.3
Opposite U-shape notched plate $t/H = 0.05$	Figure 4.4	Code 4.3
Opposite U-shape notched plate $t/H = 0.025$	Figure 4.5	Code 4.3
Single U-shape notched plate $t/H = 0.3$	Figure 4.6	Code 4.4
Single U-shape notched plate $t/H = 0.1$	Figure 4.7	Code 4.4
Single U-shape notched plate $t/H = 0.05$	Figure 4.8	Code 4.4
Opposite Semicircular notched plate	Figure 4.9	Code 4.5
Single semicircular notched plate	Figure 4.10	Code 4.6

similar. This is what we were looking for, the rules proposed by Neuber and Moslki-Glinka adapt fairly well, providing an easy way to calculate real stresses and strains avoiding the need of time-consuming numerical or experimental analysis.

The same comparison can be made but this time comparing K_ε. The comparison is made by dividing the nominal strain, taken from the Ramberg-Osgood relationship using σ_n as input, and the output strain from Neuber's and Molski-Glinka's method algorithms. These charts are shown in Figure 4.13 and Figure 4.14, the results being similar to the ones obtained for K_σ.

4.6 Charts

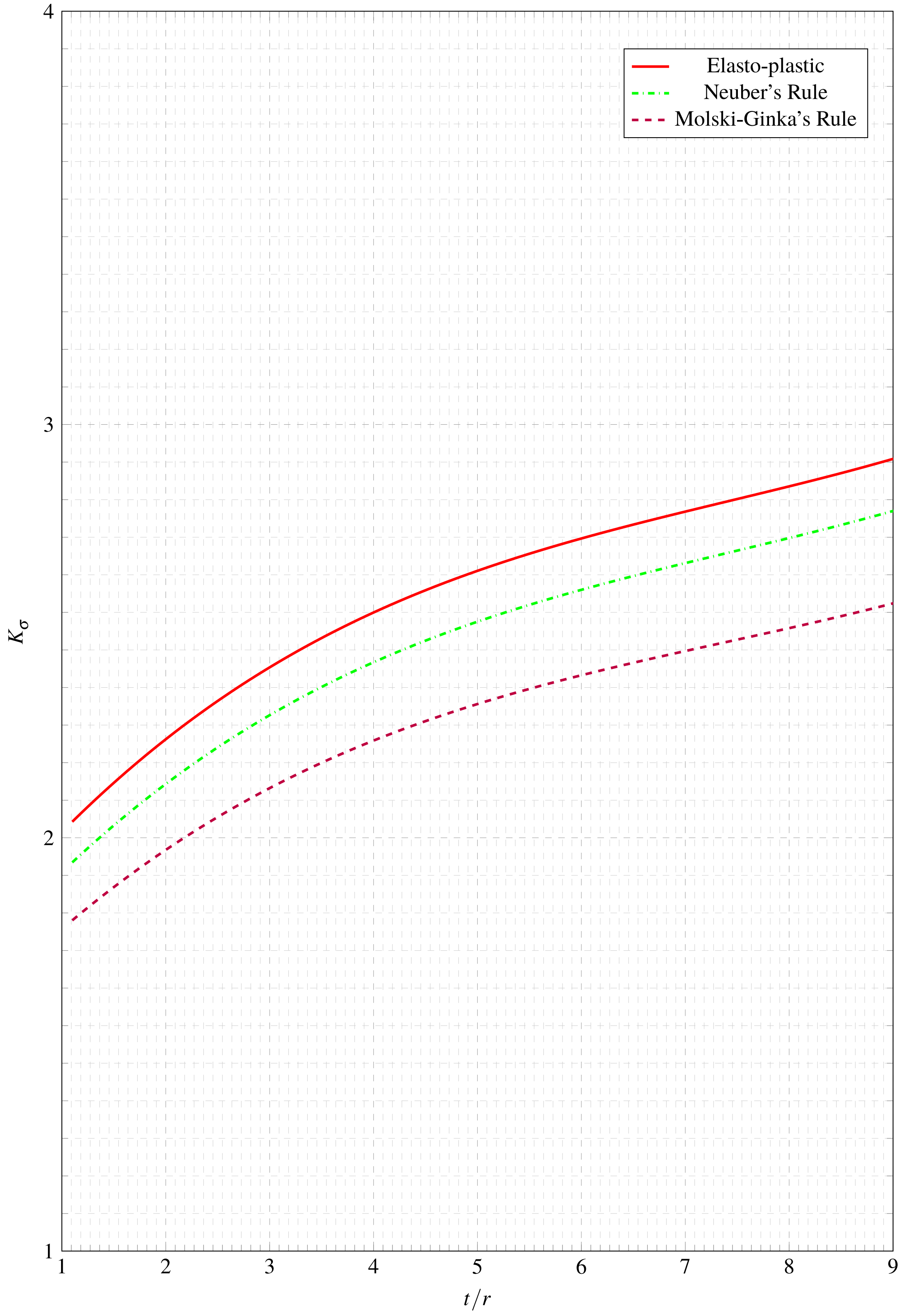

Figure 4.3 Comparison between K_σ from elastic and elasto-plastic models and values from Neuber's and Molski-Glinka's methods in opposite U-shaped notched plates with $t/H = 0.15$.

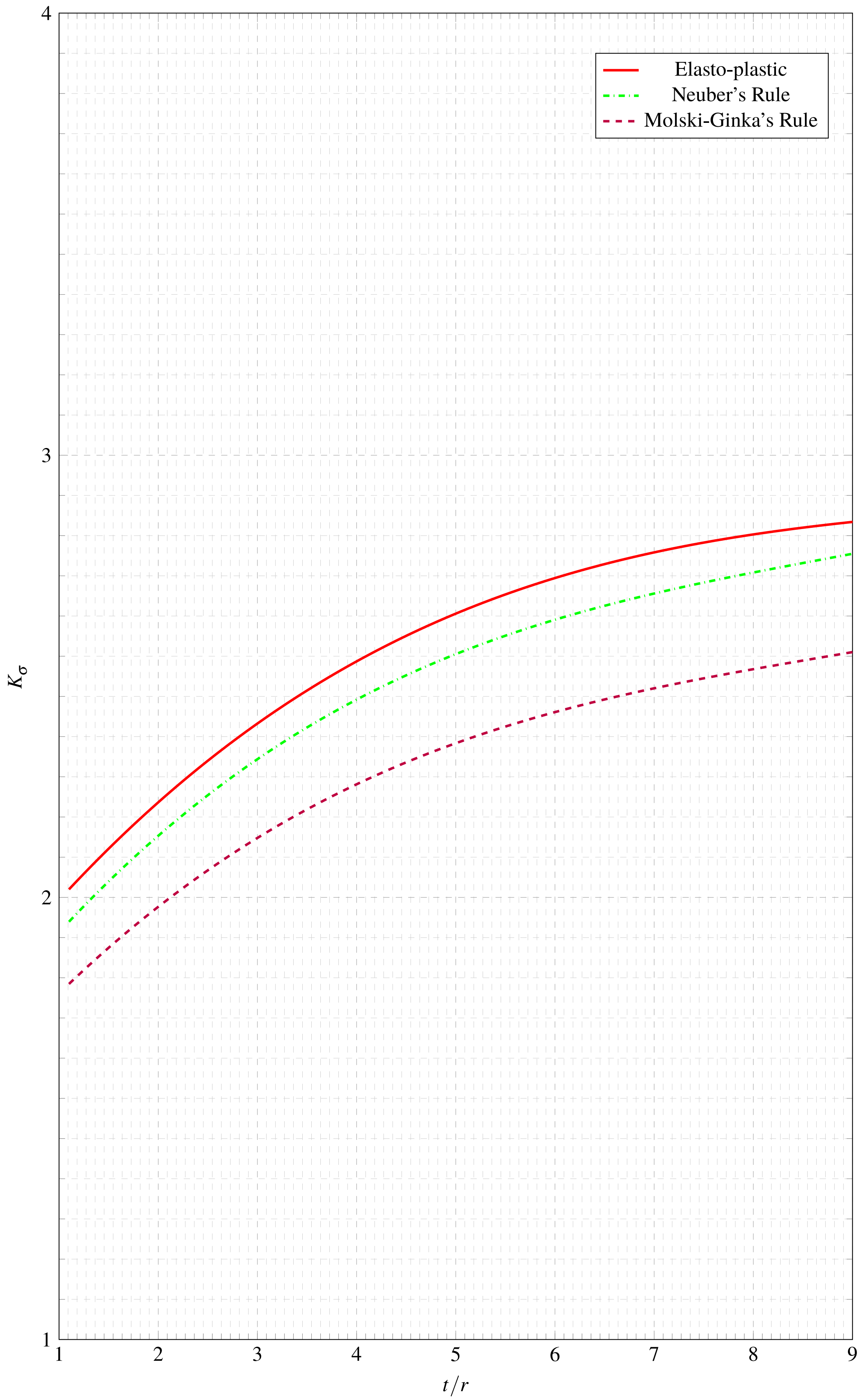

Figure 4.4 Comparison between K_σ from elastic and elasto-plastic models and values from Neuber's and Molski-Glinka's methods in opposite U-shaped notched plates with $t/H = 0.05$.

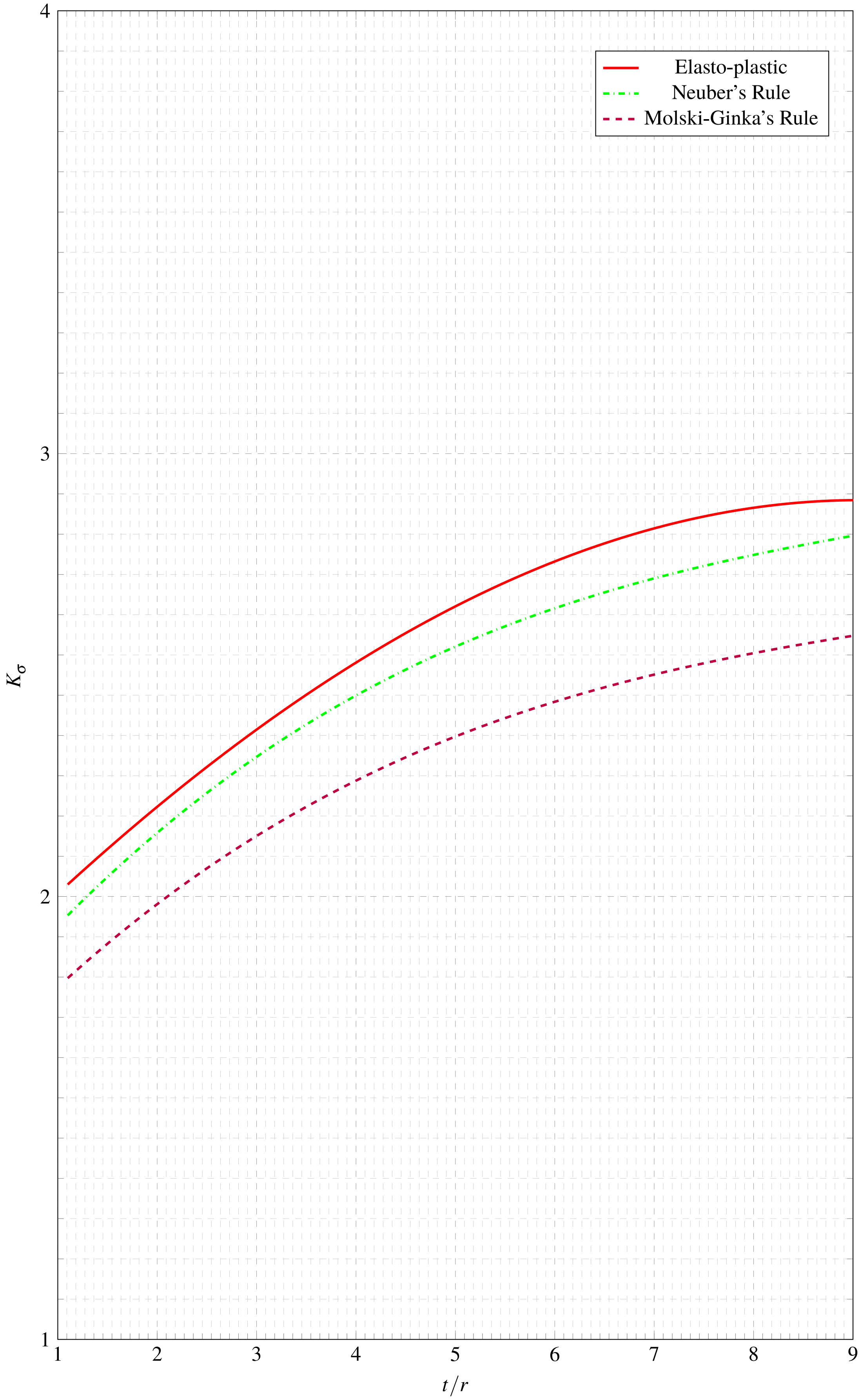

Figure 4.5 Comparison between K_σ from elastic and elasto-plastic models and values from Neuber's and Molski-Glinka's methods in opposite U-shaped notched plates with $t/H = 0.025$.

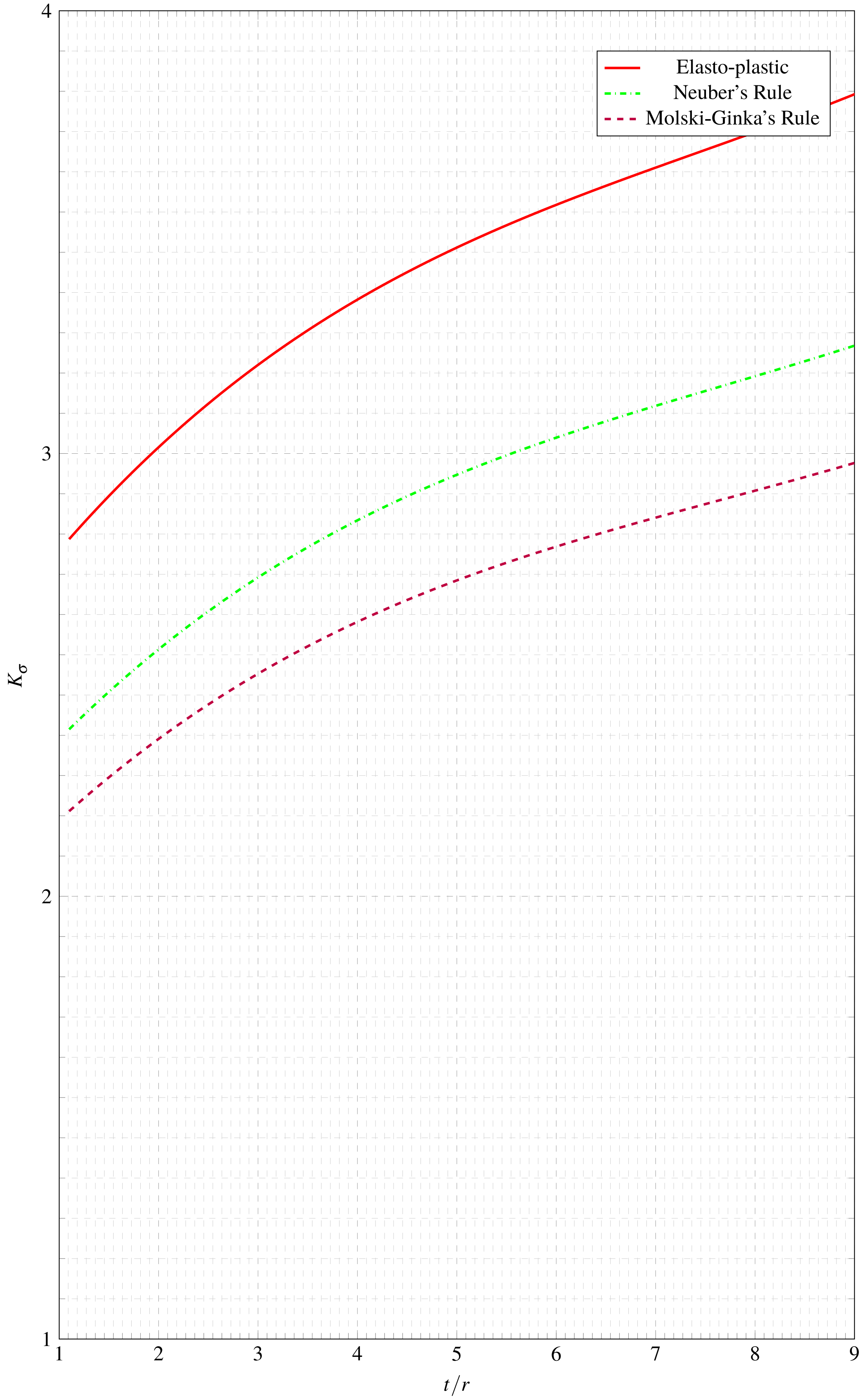

Figure 4.6 Comparison between K_σ from elastic and elasto-plastic models and values from Neuber's and Molski-Glinka's methods in single U-shaped notched plates with $t/H = 0.3$.

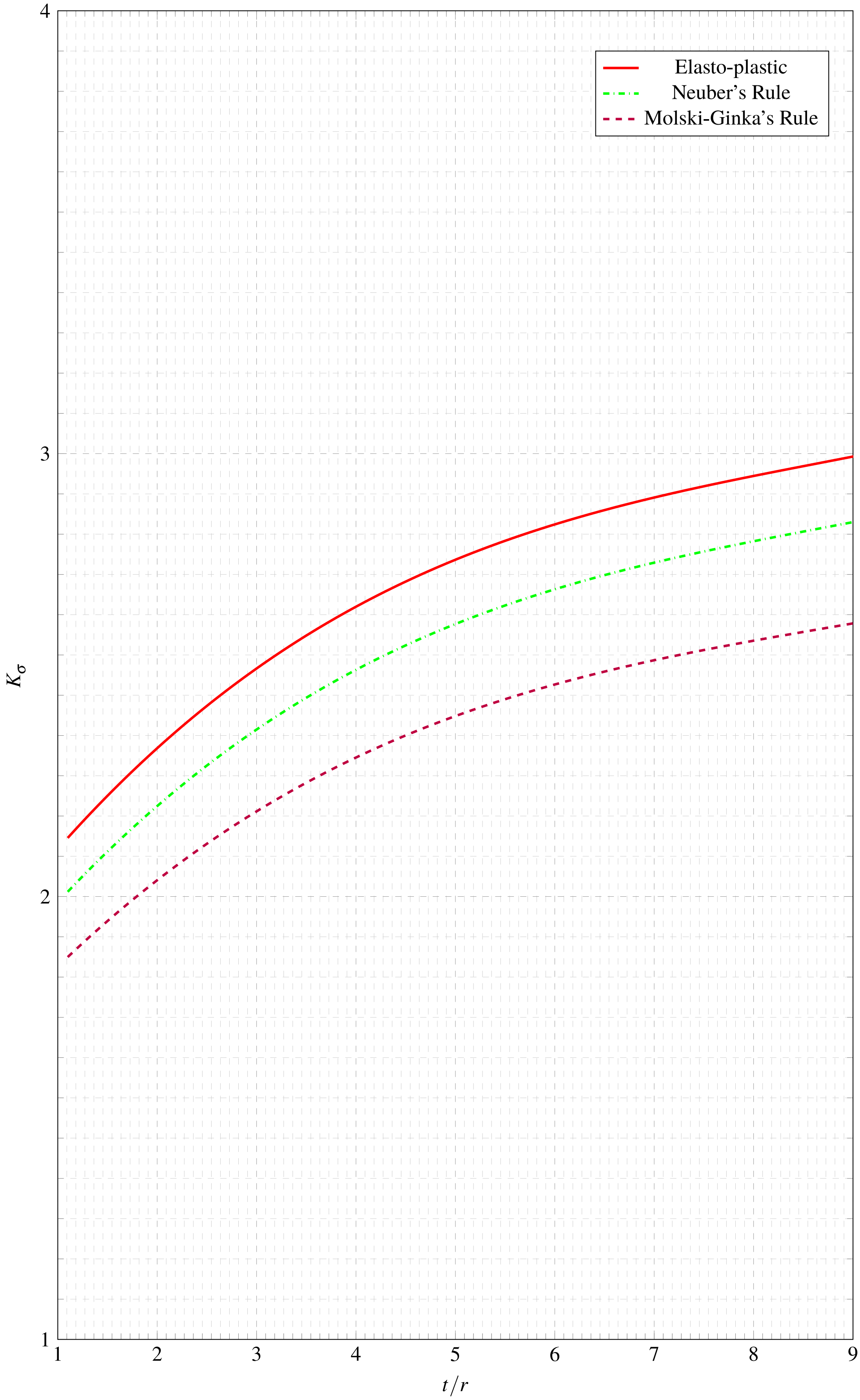

Figure 4.7 Comparison between K_σ from elastic and elasto-plastic models and values from Neuber's and Molski-Glinka's methods in single U-shaped notched plates with $t/H = 0.1$.

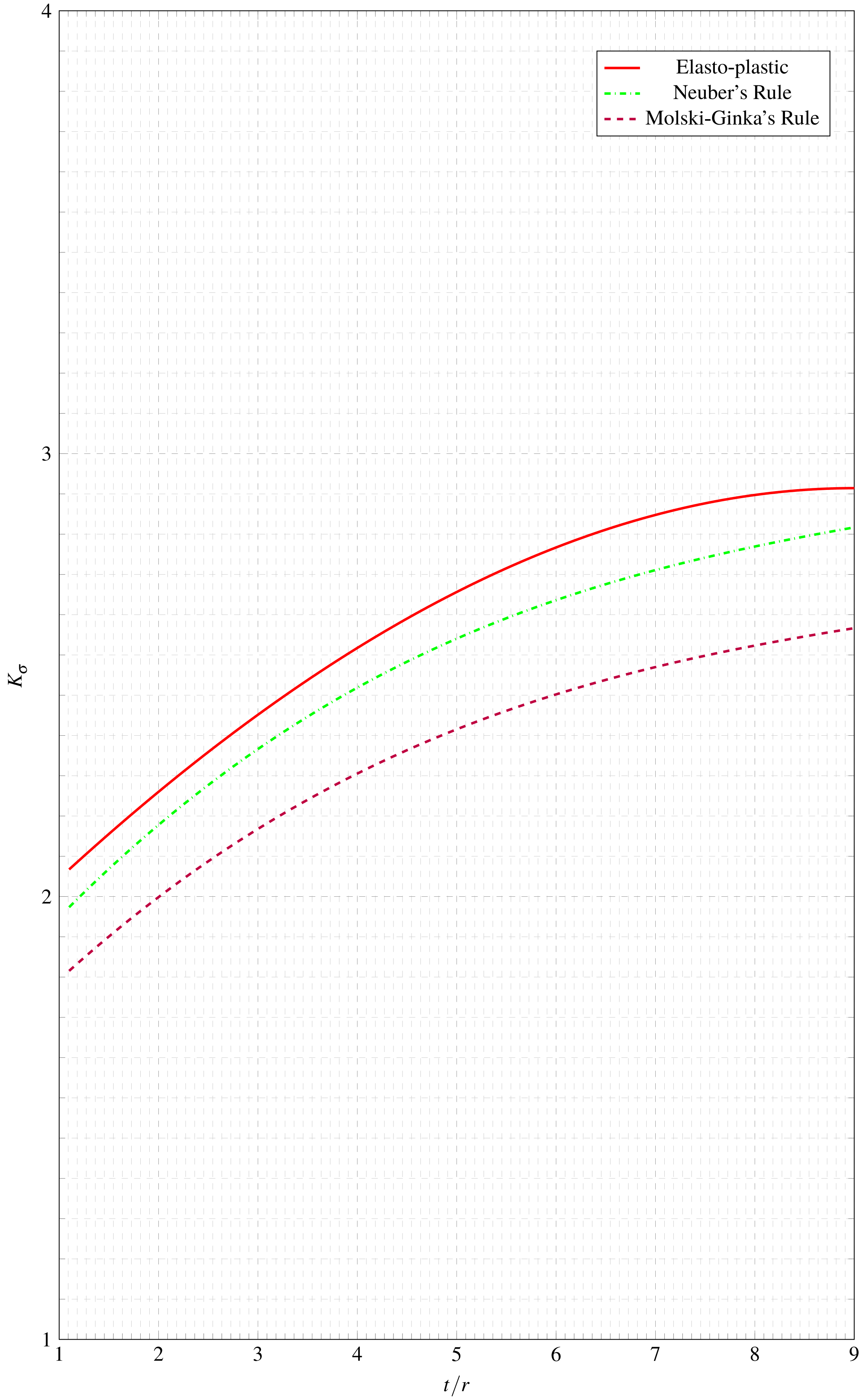

Figure 4.8 Comparison between K_σ from elastic and elasto-plastic models and values from Neuber's and Molski-Glinka's methods in single U-shaped notched plates with $t/H = 0.05$.

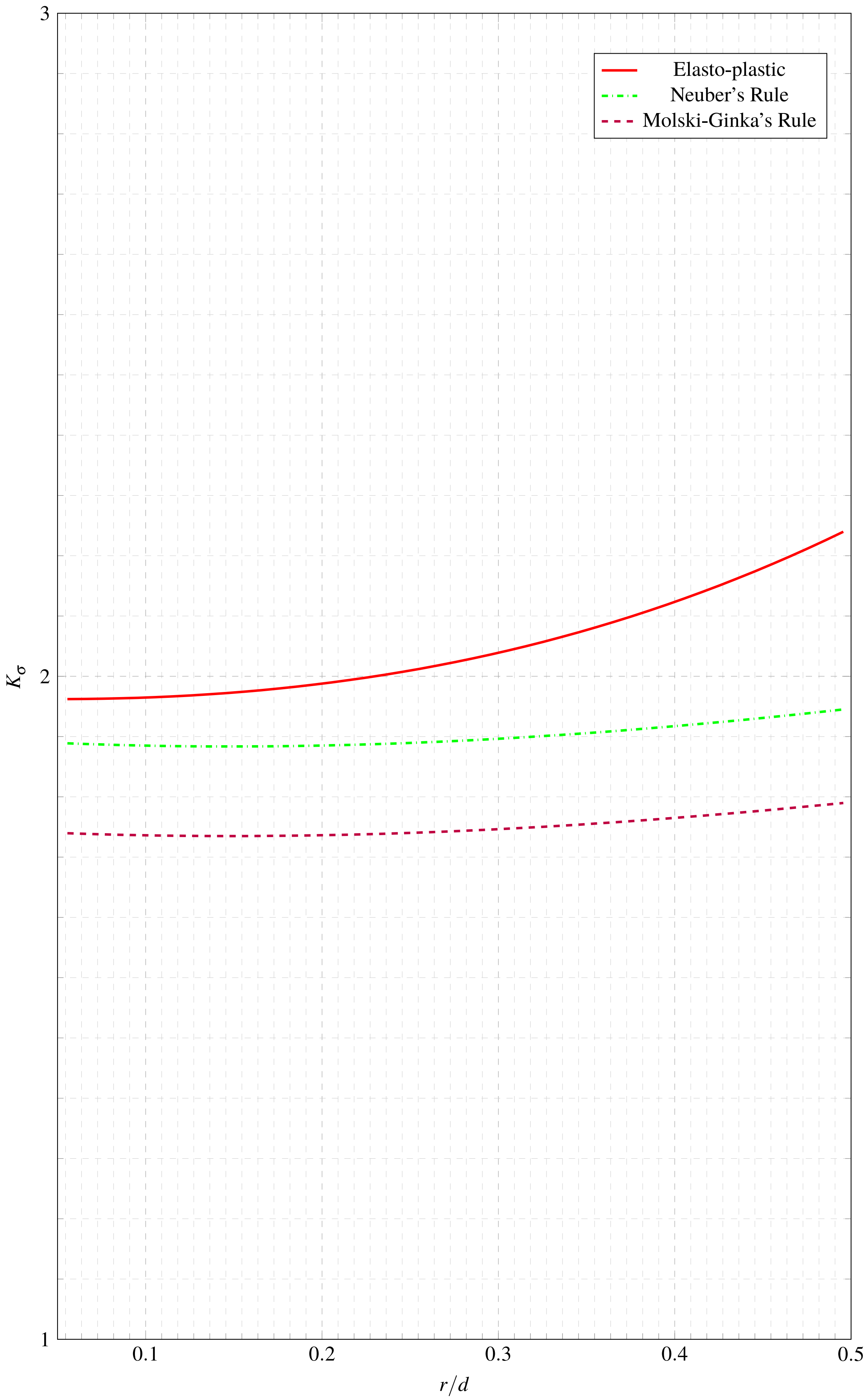

Figure 4.9 Comparison between K_σ from elastic and elasto-plastic models and values from Neuber's and Molski-Glinka's methods in opposite semicircular notched plates.

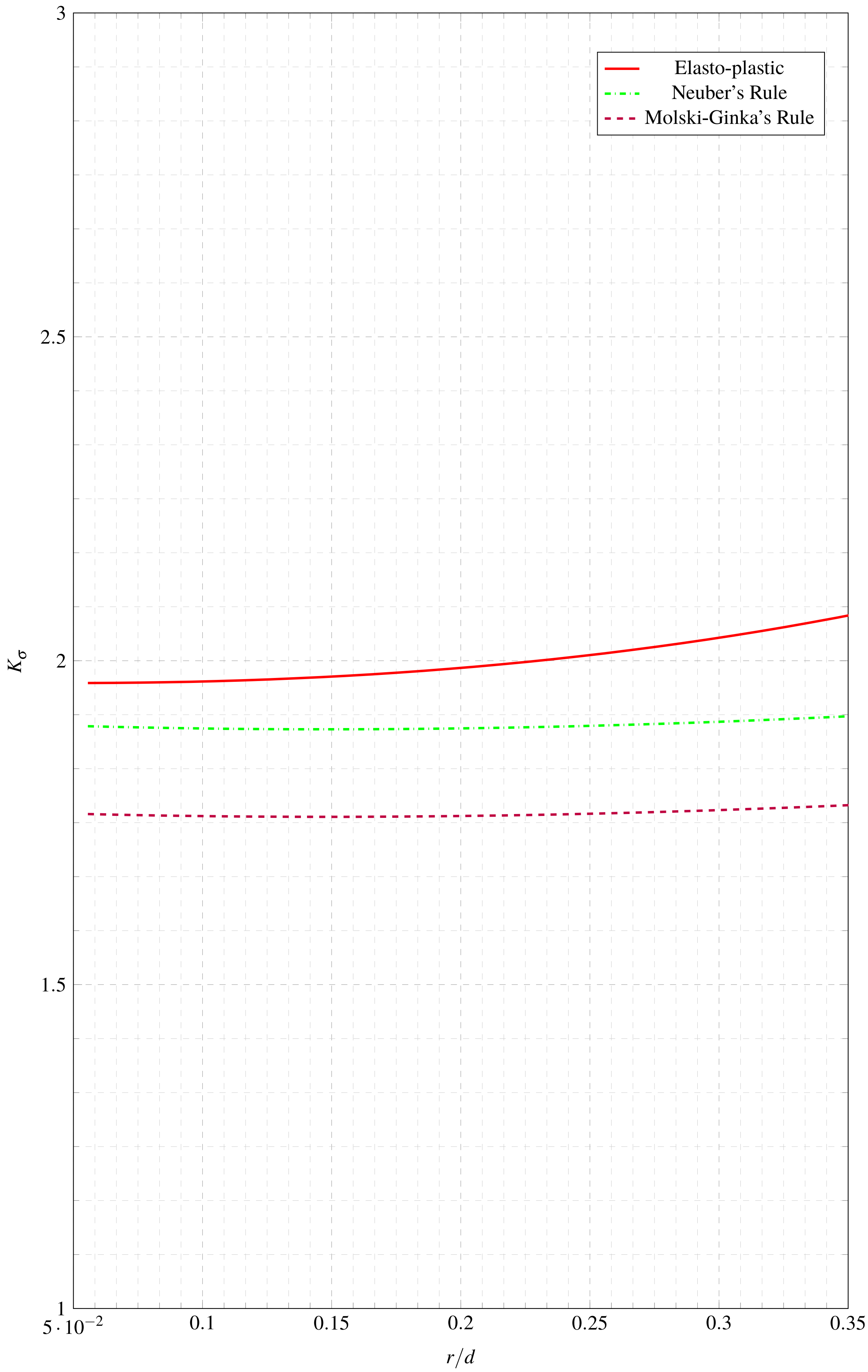

Figure 4.10 Comparison between K_σ from elastic and elasto-plastic models and values from Neuber's and Molski-Glinka's methods in single semicircular notched plates.

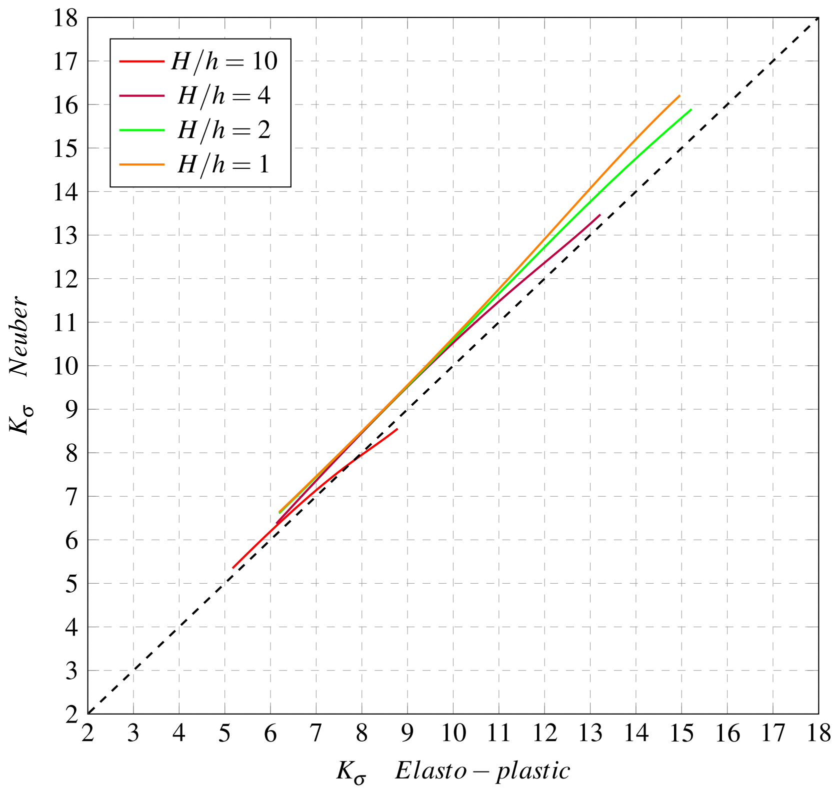

Figure 4.11 Comparison between K_σ obtained from elasto-plastic models and Neuber's method from plates with different thickness.

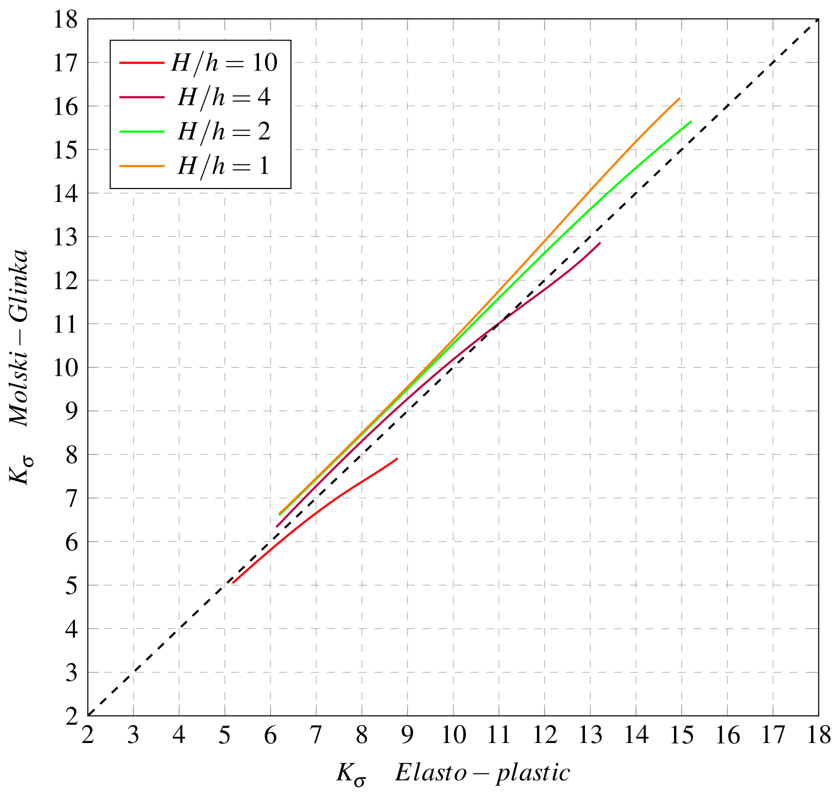

Figure 4.12 Comparison between K_σ obtained from elasto-plastic models and Molski-Glinka's method from plates with different thickness.

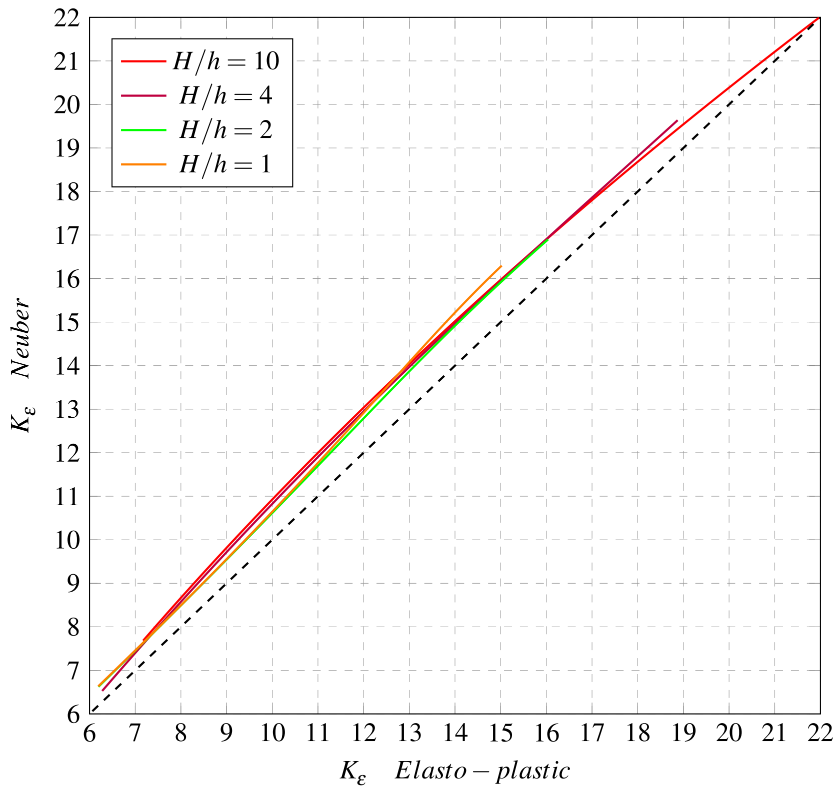

Figure 4.13 Comparison between K_ε obtained from elasto-plastic models and Neuber's method from plates with different thickness.

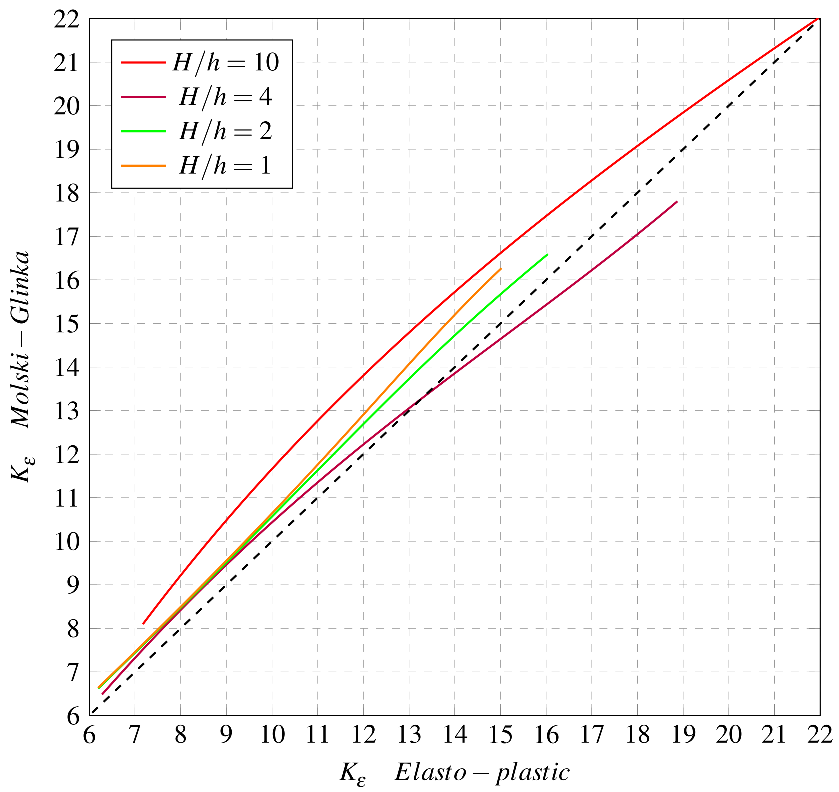

Figure 4.14 Comparison between K_ε obtained from elasto-plastic models and Molski-Glinka's method from plates with different thickness.

4.7 Codes

Code 4.1 MATLAB implementation for Neuber's method algorithm.

```
function [ S, e ] = NeuberSe( E, n, K, Se )
%[ S, e ] = NeuberSe( E, n, K, Se )
%   Given E, n, K and a Pseudostress (Se = Kt*Sn), NeuberSe calculates real
    Stress
%   and Strain in a notched piece intersecting Neuber's Hyperbola and Ramberg-
    Osgood function.

tol=1.0;

Lc=((2*n)/(n+1))*((3*n+1)/(n+1))^((1-n)/(2*n));
Bc=sqrt((n+1)/(3*n+1));

L=(E/Se)*(Se/K)^(1/n);

if Lc==L
    B=Bc;
elseif L<Lc
    Bo=Bc;
    Bi=sqrt(1-L*Bo^(1/n+1));
    while abs((Bo-Bi)/Bo)*100>tol
        Bo=Bi;
        Bi=sqrt(1-L*Bo^(1/n+1));
    end
    B=Bi;
else
    Bo=Bc;
    Bi=((1-Bo^2)/L)^(n/(n+1));
    while abs((Bo-Bi)/Bo)*100>tol
        Bo=Bi;
        Bi=((1-Bo^2)/L)^(n/(n+1));

    end
    B=Bi;
end

S=B*Se;
e=S/E+(S/K)^(1/n);

end
```

Code 4.2 MATLAB implementation for Molski-Glinka's method algorithm.

```
function [ S, e ] = MolskiGlinkaSe( E, n, K, Se )
%[ S, e ] = MolskiGlinkaSe( E, n, K, Se )
%   Given E, n, K and a Pseudostress (Se = Kt*Sn), MolskiGlinkaSe calculates
    real Stress
%   and Strain in a notched piece according to Molski-Glinka's rule.

tol=1.0;
```

```
Lc=((2*n)/(n+1))*((3*n+1)/(n+1))^((1-n)/(2*n));
Bc=sqrt((n+1)/(3*n+1));

L=2/(n+1)*E/Se*(Se/K)^(1/n);

if Lc==L
    B=Bc;
elseif L<Lc
    Bo=Bc;
    Bi=sqrt(1-L*Bo^(1/n+1));
    while abs((Bo-Bi)/Bo)*100>tol
        Bo=Bi;
        Bi=sqrt(1-L*Bo^(1/n+1));
    end
    B=Bi;
else
    Bo=Bc;
    Bi=((1-Bo^2)/L)^(n/(n+1));
    while abs((Bo-Bi)/Bo)*100>tol
        Bo=Bi;
        Bi=((1-Bo^2)/L)^(n/(n+1));
    end
    B=Bi;
end

S=B*Se;
e=S/E+(S/K)^(1/n);

end
```

Code 4.3 MATLAB implementation for Neuber's and Molski-Glinka's methods algorithms for opposite U-shape notched plates.

```
clear
close all
clc

E=202375e6;
n=0.211;
K=1283e6;

H=1;   % ANCHO
h=0.05; % PROFUNDIDAD
P=4e6; % Carga

B=0.3048;
T30=0.3*H;
T10=0.1*H;
T05=0.05*H;

hr=[1.1 2.1 3.1 4.1 5.1 6.1 7.1 8.1 9.1];
R30=T30./hr;
R10=T10./hr;
R05=T05./hr;
```

```
d=zeros([1 length(hr)]);
ratio30=zeros([1 length(hr)]);
ratio10=zeros([1 length(hr)]);
ratio05=zeros([1 length(hr)]);
Pn=zeros([1 length(hr)]);

for i=1:length(hr)
   ratio30(i)=T30/R30(i);
   ratio10(i)=T10/R10(i);
   ratio05(i)=T05/R05(i);
   Pn(i)=P/(h*H);
end


%% PS30

DATOS_PLANE_STRESS_30 = load('OPPOSITE_PLANE_STRESS/PS_30/DATOS.dat');
Sps30=DATOS_PLANE_STRESS_30';

Ktps30=zeros([1 length(hr)]);
for i=1:length(hr)
   Ktps30(i)=Sps30(i)/Pn(i);
end

ratiofinoPS30=ratio30(1):0.01:ratio30(length(ratio30));
KTPS30=spline(ratio30,Ktps30,ratiofinoPS30);
KTPS30_FIT=fit(ratio30',Ktps30','smoothingspline');

%% PS10
DATOS_PLANE_STRESS_10 = load('OPPOSITE_PLANE_STRESS/PS_10/DATOS.dat');
Sps10=DATOS_PLANE_STRESS_10';

Ktps10=zeros([1 length(hr)]);
for i=1:length(hr)
   Ktps10(i)=Sps10(i)/Pn(i);
end

ratiofinoPS10=ratio10(1):0.01:ratio10(length(ratio10));
KTPS10=spline(ratio10,Ktps10,ratiofinoPS10);
KTPS10_FIT=fit(ratio10',Ktps10','smoothingspline');


%% PLASTICITY 30

DATOS_PLAS_30 = load('OPPOSITE_PLASTICITY/PLAS_30/DATOS.dat');
S_PLAS_30=DATOS_PLAS_30';

Kt_PLAS_30=zeros([1 length(hr)]);
for i=1:length(hr)
   Kt_PLAS_30(i)=S_PLAS_30(i)/Pn(i);
end

KT_PLAS_30=spline(ratio30,Kt_PLAS_30,ratiofinoPS30);
KT_PLAS_30_FIT=fit(ratio30',Kt_PLAS_30','smoothingspline');
```

```
%% PLASTICITY 10

DATOS_PLAS_10 = load('OPPOSITE_PLASTICITY/PLAS_10/DATOS.dat');
S_PLAS_10=DATOS_PLAS_10';

Kt_PLAS_10=zeros([1 length(hr)]);
for i=1:length(hr)
   Kt_PLAS_10(i)=S_PLAS_10(i)/Pn(i);
end

KT_PLAS_10=spline(ratio10,Kt_PLAS_10,ratiofinoPS10);
KT_PLAS_10_FIT=fit(ratio10',Ktps10','smoothingspline');


%% RESULTADOS NEUBER 30

Kt_neu_30=zeros([1 length(hr)]);
for i=1:length(hr)
    [ Sn, en ] = NeuberSe( E, n, K, S_PLAS_30(i) );
    Kt_neu_30(i)=Sn/Pn(i);
end

KT_NEU_30=spline(ratio30,Kt_neu_30,ratiofinoPS30);
KT_NEU_30_FIT=fit(ratio30',Kt_neu_30','smoothingspline');


%% RESULTADOS NEUBER 10

Kt_neu_10=zeros([1 length(hr)]);
for i=1:length(hr)
    [ Sn, en ] = NeuberSe( E, n, K, S_PLAS_10(i) );
    Kt_neu_10(i)=Sn/Pn(i);
end

KT_NEU_10=spline(ratio10,Kt_neu_10,ratiofinoPS10);
KT_NEU_10_FIT=fit(ratio10',Kt_neu_10','smoothingspline');

%% MOLSKI-GLINKA 30

Kt_mg_30=zeros([1 length(hr)]);
for i=1:length(hr)
    [ Smg, emg ] = MolskiGlinkaSe( E, n, K, S_PLAS_30(i) );
    Kt_mg_30(i)=Smg/Pn(i);
end

KT_MG_30=spline(ratio30,Kt_mg_30,ratiofinoPS30);
KT_MG_30_FIT=fit(ratio30',Kt_mg_30','smoothingspline');


%% PLOT

figure(1)
plot(KTPS30_FIT,'b')
hold on
plot(KT_PLAS_30_FIT,'g--')
hold on
plot(KT_NEU_30_FIT,'r-.')
```

```
hold on
plot(KT_MG_30_FIT,'m..')
legend('ANSYS Plane Stress 30%','ANSYS Plasticity 30%','NEUBER 30%','Molski-
    Glinka 30%')
axis([1 10 2 13])
title('Comparison between plasticity and plane stress')
xlabel('t/r')
ylabel('Kt')


figure(2)
plot(ratiofinoPS10,KTPS10,ratiofinoPS10,KT_PLAS_10,ratiofinoPS10,KT_NEU_10)
legend('ANSYS Plane Stress 10%','ANSYS Plasticity 10%','NEUBER 10%')
title('Comparison between plasticity and plane stress')
xlabel('t/r')
ylabel('Kt')

figure(3)
plot(ratiofinoPS30,KTPS30,'b',ratiofinoPS30,KT_PLAS_30,'b--',ratiofinoPS30,
    KT_NEU_30,'b-.',ratiofinoPS10,KTPS10,'r',ratiofinoPS10,KT_PLAS_10,'r--',
    ratiofinoPS10,KT_NEU_10,'r-.')
legend('ANSYS Plane Stress 30%','ANSYS Plasticity 30%','NEUBER 30%','ANSYS
    Plane Stress 10%','ANSYS Plasticity 10%','NEUBER 10%')
title('Comparison between plasticity, solid and plane stress')
xlabel('t/r')
ylabel('Kt')
```

Code 4.4 MATLAB implementation for Neuber's and Molski-Glinka's methods algorithms for single U-shape notched plates.

```
clear
close all
clc

E=202375e6;
n=0.211;
K=1283e6;

H=1;   % ANCHO
h=0.05; % PROFUNDIDAD
P=10000000; % Carga

B=3.5;
T30=0.3*H;
T10=0.1*H;
T05=0.05*H;

hr=[1.1 2.1 3.1 4.1 5.1 6.1 7.1 8.1 9.1];
R30=T30./hr;
R10=T10./hr;
R05=T05./hr;

d=zeros([1 length(hr)]);
ratio30=zeros([1 length(hr)]);
ratio10=zeros([1 length(hr)]);
```

```
ratio05=zeros([1 length(hr)]);
Pn=zeros([1 length(hr)]);

for i=1:length(hr)
   ratio30(i)=T30/R30(i);
   ratio10(i)=T10/R10(i);
   ratio05(i)=T05/R05(i);
   Pn(i)=P/(h*H);
end


%% PS30

DATOS_PLANE_STRESS_30 = load('U_SHAPE_PLANE_STRESS/PS_30/DATOS_BARATTA.dat');
Sps30=DATOS_PLANE_STRESS_30';

Ktps30=zeros([1 length(hr)]);
for i=1:length(hr)
   Ktps30(i)=Sps30(i)/Pn(i);
end

ratiofinoPS30=ratio30(1):0.01:ratio30(length(ratio30));
KTPS30_FIT=polyval(polyfit(ratio30,Ktps30,3),ratiofinoPS30);


%% PS10
DATOS_PLANE_STRESS_10 = load('U_SHAPE_PLANE_STRESS/PS_10/DATOS_BARATTA.dat');
Sps10=DATOS_PLANE_STRESS_10';

Ktps10=zeros([1 length(hr)]);
for i=1:length(hr)
   Ktps10(i)=Sps10(i)/Pn(i);
end

ratiofinoPS10=ratio10(1):0.01:ratio10(length(ratio10));
KTPS10_FIT=polyval(polyfit(ratio10,Ktps10,3),ratiofinoPS10);


%% PS05
DATOS_PLANE_STRESS_05 = load('U_SHAPE_PLANE_STRESS/PS_05/DATOS_BARATTA.dat');
Sps05=DATOS_PLANE_STRESS_05';

Ktps05=zeros([1 length(hr)]);
for i=1:length(hr)
   Ktps05(i)=Sps05(i)/Pn(i);
end

ratiofinoPS05=ratio05(1):0.01:ratio05(length(ratio05));
KTPS05_FIT=polyval(polyfit(ratio05,Ktps05,3),ratiofinoPS05);



%% PLASTICITY 30

DATOS_PLAS_30 = load('U_SHAPE_PLASTICITY/PLAS_30/DATOS_BARATTA.dat');
S_PLAS_30=DATOS_PLAS_30';
```

```
Kt_PLAS_30=zeros([1 length(hr)]);
for i=1:length(hr)
   Kt_PLAS_30(i)=S_PLAS_30(i)/Pn(i);
end

KT_PLAS_30_FIT=polyval(polyfit(ratio30,Kt_PLAS_30,3),ratiofinoPS30);


%% PLASTICITY 10

DATOS_PLAS_10 = load('U_SHAPE_PLASTICITY/PLAS_10/DATOS_BARATTA.dat');
S_PLAS_10=DATOS_PLAS_10';

Kt_PLAS_10=zeros([1 length(hr)]);
for i=1:length(hr)
   Kt_PLAS_10(i)=S_PLAS_10(i)/Pn(i);
end

KT_PLAS_10_FIT=polyval(polyfit(ratio10,Kt_PLAS_10,3),ratiofinoPS10);



%% PLASTICITY 05

DATOS_PLAS_05 =load('U_SHAPE_PLASTICITY/PLAS_05/DATOS_BARATTA.dat');
S_PLAS_05=DATOS_PLAS_05';

Kt_PLAS_05=zeros([1 length(hr)]);
for i=1:length(hr)
   Kt_PLAS_05(i)=S_PLAS_05(i)/Pn(i);
end

KT_PLAS_05_FIT=polyval(polyfit(ratio05,Kt_PLAS_05,3),ratiofinoPS05);

%% RESULTADOS NEUBER 30

Kt_neu_30=zeros([1 length(hr)]);
for i=1:length(hr)
    [ Sn, en ] = NeuberSe( E, n, K, S_PLAS_30(i) );
    Kt_neu_30(i)=Sn/Pn(i);
end

KT_NEU_30_FIT=polyval(polyfit(ratio30,Kt_neu_30,3),ratiofinoPS30);


%% RESULTADOS NEUBER 10

Kt_neu_10=zeros([1 length(hr)]);
for i=1:length(hr)
    [ Sn, en ] = NeuberSe( E, n, K, S_PLAS_10(i) );
    Kt_neu_10(i)=Sn/Pn(i);
end

KT_NEU_10_FIT=polyval(polyfit(ratio10,Kt_neu_10,3),ratiofinoPS10);


%% RESULTADOS NEUBER 05
```

```
Kt_neu_05=zeros([1 length(hr)]);
for i=1:length(hr)
    [ Sn, en ] = NeuberSe( E, n, K, S_PLAS_05(i) );
    Kt_neu_05(i)=Sn/Pn(i);
end

KT_NEU_05_FIT=polyval(polyfit(ratio05,Kt_neu_05,3),ratiofinoPS05);


%% MOLSKI-GLINKA 30

Kt_mg_30=zeros([1 length(hr)]);
for i=1:length(hr)
    [ Smg, emg ] = MolskiGlinkaSe( E, n, K, S_PLAS_30(i) );
    Kt_mg_30(i)=Smg/Pn(i);
end

KT_MG_30_FIT=polyval(polyfit(ratio30,Kt_mg_30,3),ratiofinoPS30);


%% MOLSKI-GLINKA 10

Kt_mg_10=zeros([1 length(hr)]);
for i=1:length(hr)
    [ Smg, emg ] = MolskiGlinkaSe( E, n, K, S_PLAS_10(i) );
    Kt_mg_10(i)=Smg/Pn(i);
end

KT_MG_10_FIT=polyval(polyfit(ratio10,Kt_mg_10,3),ratiofinoPS10);


%% MOLSKI-GLINKA 05

Kt_mg_05=zeros([1 length(hr)]);
for i=1:length(hr)
    [ Smg, emg ] = MolskiGlinkaSe( E, n, K, S_PLAS_05(i) );
    Kt_mg_05(i)=Smg/Pn(i);
end

KT_MG_05_FIT=polyval(polyfit(ratio05,Kt_mg_05,3),ratiofinoPS05);


%% PLOT

figure(1)
plot(ratiofinoPS30,KTPS30_FIT,'b')
hold on
plot(ratiofinoPS30,KT_PLAS_30_FIT,'g--')
hold on
plot(ratiofinoPS30,KT_NEU_30_FIT,'r-.')
hold on
plot(ratiofinoPS30,KT_MG_30_FIT,'m:')
legend('ANSYS Plane Stress 30%','ANSYS Plasticity 30%','NEUBER 30%','Molski-
    Glinka 30%')
axis([1 10 0 13])
title('Comparison between plasticity and plane stress')
```

```
xlabel('t/r')
ylabel('Kt')
grid on
grid minor

figure(2)
plot(ratiofinoPS10,KTPS10_FIT,'b')
hold on
plot(ratiofinoPS10,KT_PLAS_10_FIT,'g--')
hold on
plot(ratiofinoPS10,KT_NEU_10_FIT,'r-.')
hold on
plot(ratiofinoPS10,KT_MG_10_FIT,'m:')
legend('ANSYS Plane Stress 10 %','ANSYS Plasticity 10 %','NEUBER 10 %','Molski-
    Glinka 10 %')
axis([1 10 0 10])
title('Comparison between plasticity and plane stress')
xlabel('t/r')
ylabel('Kt')
grid on
grid minor

figure(3)
plot(ratiofinoPS05,KTPS05_FIT,'b')
hold on
plot(ratiofinoPS05,KT_PLAS_05_FIT,'g--')
hold on
plot(ratiofinoPS05,KT_NEU_05_FIT,'r-.')
hold on
plot(ratiofinoPS05,KT_MG_05_FIT,'m:')
legend('ANSYS Plane Stress 5 %','ANSYS Plasticity 5 %','NEUBER 5 %','Molski-
    Glinka 5 %')
axis([1 10 0 9])
title('Comparison between plasticity and plane stress')
xlabel('t/r')
ylabel('Kt')
grid on
grid minor

figure(4)
plot(ratiofinoPS30,KTPS30_FIT,'b')
hold on
plot(ratiofinoPS30,KT_PLAS_30_FIT,'b--')
hold on
plot(ratiofinoPS30,KT_NEU_30_FIT,'b-.')
hold on
plot(ratiofinoPS30,KT_MG_30_FIT,'b:')
hold on
plot(ratiofinoPS10,KTPS10_FIT,'r')
hold on
plot(ratiofinoPS10,KT_PLAS_10_FIT,'r--')
hold on
plot(ratiofinoPS10,KT_NEU_10_FIT,'r-.')
hold on
plot(ratiofinoPS10,KT_MG_10_FIT,'r:')
hold on
plot(ratiofinoPS05,KTPS05_FIT,'g')
```

```
hold on
plot(ratiofinoPS05,KT_PLAS_05_FIT,'g--')
hold on
plot(ratiofinoPS05,KT_NEU_05_FIT,'g-.')
hold on
plot(ratiofinoPS05,KT_MG_05_FIT,'g:')
legend('ANSYS Plane Stress 30 %', 'ANSYS Plasticity 30 %', 'NEUBER 30 %', '
    Molski-Glinka 30 %','ANSYS Plane Stress 10 %', 'ANSYS Plasticity 10 %', '
    NEUBER 10 %', 'Molski-Glinka 10 %','ANSYS Plane Stress 5 %', 'ANSYS
    Plasticity 5 %', 'NEUBER 5 %', 'Molski-Glinka 5 %')
axis([1 9 0 13])
xlabel('t/r')
ylabel('Kt')
grid on
grid minor
```

Code 4.5 MATLAB implementation for Neuber's and Molski-Glinka's methods algorithms for opposite semicircular notched plates.

```
clear
close all
clc

E=202375000000;
n=0.211;
K=1283e6;

H=2;   % ANCHO
h=0.05; % PROFUNDIDAD
P=2*10000000; % Carga
Pn=P/(H*h);
B=5;

DIV=[2,2.5,3,3.5,4,4.5,5,5.5,6,6.5,7,7.5,8,8.5,9,9.5,10];
R=(H/2)./DIV;

d=H-2*R;

rd=R./d;
ratio=rd;
ratiofino=rd(length(rd)):0.01:rd(1);


%% ANSYS OPPOSITE DIST
DATOS_PLANE_STRESS_DIST = load('SEMICIRCULAR_OPPOSITE_DIST/DATOS.dat');
Sps_dist=DATOS_PLANE_STRESS_DIST';

Ktps_dist=zeros([1 length(R)]);
for i=1:length(R)
   Ktps_dist(i)=Sps_dist(i)/Pn;
end

KTPS_FIT_DIST=polyval(polyfit(rd,Ktps_dist,3),ratiofino);

%% ANSYS OPPOSITE PLASTICITY
```

```
DATOS_PLANE_STRESS_PLAS = load('SEMICIRCULAR_OPPOSITE_PLASTICITY/DATOS.dat');
S_PLAS=DATOS_PLANE_STRESS_PLAS';

Ktps_plas=zeros([1 length(R)]);
for i=1:length(R)
   Ktps_plas(i)=S_PLAS(i)/Pn;
end

KT_PLAS=polyval(polyfit(rd,Ktps_plas,3),ratiofino);

%% NEUBER

Kt_neu=zeros([1 length(rd)]);
for i=1:length(rd)
    [ Sn, en ] = NeuberSe( E, n, K, S_PLAS(i) );
    Kt_neu(i)=Sn/Pn;
end

KT_NEU=polyval(polyfit(ratio,Kt_neu,3),ratiofino);

%% MOLSKI-GLINKA

Kt_mg=zeros([1 length(rd)]);
for i=1:length(rd)
    [ Smg, emg ] = MolskiGlinkaSe( E, n, K, S_PLAS(i) );
    Kt_mg(i)=Smg/Pn;
end

KT_MG=polyval(polyfit(rd,Kt_mg,3),ratiofino);

%% PLOT

figure(1)
plot(ratiofino,KTPS_FIT_DIST,'b')
hold on
plot(ratiofino,KT_PLAS,'g-.')
hold on
plot(ratiofino,KT_NEU,'r--')
hold on
plot(ratiofino,KT_MG,'m--')
xlabel('r/d')
ylabel('Kt')
axis([0 0.5 1 4])
legend('Plane Stress (Pure Elastic)','Elasto-plastic','Neuber','Molski-Glinka')
grid on
grid minor
```

Code 4.6 MATLAB implementation for Neuber's and Molski-Glinka's methods algorithms for single semicircular notched plates.

```
clear
close all
clc

E=202375000000;
```

```
n=0.211;
K=1283e6;

H=2;   % ANCHO
h=0.05; % PROFUNDIDAD
P=10000000; % Carga
Pn=P/(H*h);
B=5;

DIV=[2,2.5,3,3.5,4,4.5,5,5.5,6,6.5,7,7.5,8,8.5,9,9.5,10];
R=(H/2)./DIV;

d=H-R;
Pnd=P./(d*h);

rd=R./d;
ratio=rd;
ratiofino=rd(length(rd)):0.01:rd(1);


%% ANSYS SINGLE DIST
DATOS_PLANE_STRESS_DIST = load('SEMICIRCULAR_SINGLE_DIST/DATOS.dat');
Sps_dist=DATOS_PLANE_STRESS_DIST';

Ktps_dist=zeros([1 length(R)]);
for i=1:length(R)
   Ktps_dist(i)=Sps_dist(i)/Pn;
end

KTPS_FIT_DIST=polyval(polyfit(rd,Ktps_dist,3),ratiofino);

%% ANSYS SINGLE PLASTICITY
DATOS_PLANE_STRESS_PLAS = load('SEMICIRCULAR_SINGLE_PLASTICITY/DATOS.dat');
S_PLAS=DATOS_PLANE_STRESS_PLAS';

Ktps_plas=zeros([1 length(R)]);
for i=1:length(R)
   Ktps_plas(i)=S_PLAS(i)/Pn;
end

KT_PLAS=polyval(polyfit(rd,Ktps_plas,3),ratiofino);

%% NEUBER

Kt_neu=zeros([1 length(rd)]);
for i=1:length(rd)
    [ Sn, en ] = NeuberSe( E, n, K, S_PLAS(i) );
    Kt_neu(i)=Sn/Pn;
end

KT_NEU=polyval(polyfit(ratio,Kt_neu,3),ratiofino);

%% MOLSKI-GLINKA

Kt_mg=zeros([1 length(rd)]);
for i=1:length(rd)
    [ Smg, emg ] = MolskiGlinkaSe( E, n, K, S_PLAS(i) );
```

```
    Kt_mg(i)=Smg/Pn;
end

KT_MG=polyval(polyfit(rd,Kt_mg,3),ratiofino);

%% PLOT

figure(1)
plot(ratiofino,KTPS_FIT_DIST,'b')
hold on
plot(ratiofino,KT_PLAS,'g-.')
hold on
plot(ratiofino,KT_NEU,'r--')
hold on
plot(ratiofino,KT_MG,'m--')
xlabel('r/d')
ylabel('Kt')
axis([0.05 0.35 2 5])
legend('Plane Stress (Pure Elastic)','Elasto-plastic','Neuber','Molski-Glinka')
grid on
grid minor
```

5 Conclusions

The Finite Elements models created in this project have been compared with experimental and theoretical data from various authors in order to provide reliable results. This reliability allow us to easily modify the parameters that define these models and to obtain useful data from an infinity of different geometries and materials, avoiding the need of expensive laboratory studies. This gives this project a lot of potential, being a more ambitious goal of creating a big database including several different geometries of diverse materials. This could be the starting point for future end-of-degree or research projects that can use these texts and codes as a guide to expand the database.

The goal of applying Neuber's and Molski-Glinka's rules to the data obtained from the Finite Elements models has been fulfilled having set a particular file structure to create an output file with the data obtained from ANSYS, where several loop codes have been implemented to ease the job, to input in MATLAB where K_σ have been calculated and convergence algorithms have been implemented to apply the mentioned rules.

The goals that have not been achieved are to create a reliable solid model for very thin plates and to create a convergent elasto-plastic model with a load applied in a line at the center of the minimum section.

Regarding the data obtained from the analysis that have been carried out in this project, we can get the following conclusions:

1. The effect of the notch's depth is increased when it is bigger than recommended, a distance at least 3 times it's depth to the end of the plate.

2. Plates with a ratio H/h lower or close to 2 can be considered as to be in Plane Strain.

3. Neuber's and Molski-Glinka's methods adapt well for stresses and strains in all range of thicknesses, giving very similar values to the ones obtained from elasto-plastic models.

4. Considering K_σ based on the gross section, Neuber's and Molski-Glinka's methods are generally less conservative than the results obtained from elasto-plastic models.

List of Figures

List of Tables

Listings

Bibliography

[1] Rudolph E. Peterson, revised by Walter D. Pilkey and Deborah F. Pilkey *Peterson's Stress Concentration Factors, Third Edition.*. 2008 John Wiley & Sons, Inc.

[2] ANSYS help guide, `https://www.sharcnet.ca/Software/Ansys/16.2.3/en-us/help/ans_cmd/Hlp_C_CmdTOC.html`

[3] F. I. Baratta, D. M. Neal *Stress-concentration factors in U-shaped and semi-elliptical edge notches*. Journal of Strain Analysis, Vol. 5, No. 2, 1970.

[4] R. B. Heywood *Designing by Photoelasticity*. Chapman & Hall, London, p. 163, 1952.

[5] C. E. Inglis *Stresses in a plate due to the presence of cracks and sharp corners*. Trans. Instn nav. Archit. 1913, 55, 219.

[6] Chih-Bing Ling *Stresses in a Notched Strip Under Tension*. Journal of Applied Mechanics, A-275 to A-280, December 1947.

[7] F. G. Maunsell *LXII. Stresses in a Notched Plate under Tension*. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 21:142, 765-773, DOI: 10.1080/14786443608561624.

[8] Rudolph E. Peterson *Stress Concentration Factors*. Wiley, 1974.

[9] Norman E. Dowling, contributions by Katakam S. Prasad and R. Narayanasamy *Mechanical Behavior of Materials. Engineering Methods for Deformation, Fracture and Fatigue. Fourth Ed.*. Wiley, 1974.

[10] eFatigue - Constant Amplitude Strain-Life Technical Background `https://www.efatigue.com/constantamplitude/background/strainlife.html`

[11] H. Neuber *Theory of stress concentration for shear-strained prismatical bodies with arbitrary nonlinear stress–strain law*. J. Appl. Mech. 28 (4) (1961) 544–550.

[12] A. Navarro *An unconditionally convergent iterative algorithm for the intersection of Neuber's and Molski–Glinka's rules with the Ramberg–Osgood stress–strain*. Theoretical and Applied Fracture Mechanics 69 (2014) 53–62.

[13] Krzysztof Molski, Grzegorz Glinka *A method of elastic-plastic stress and strain calculation at a notch root*. Institute of Cranes and Heavy Machinery Engineering, Warsaw Technical University, 02-524 Warsaw, ul. Narbutta 85 (Poland).