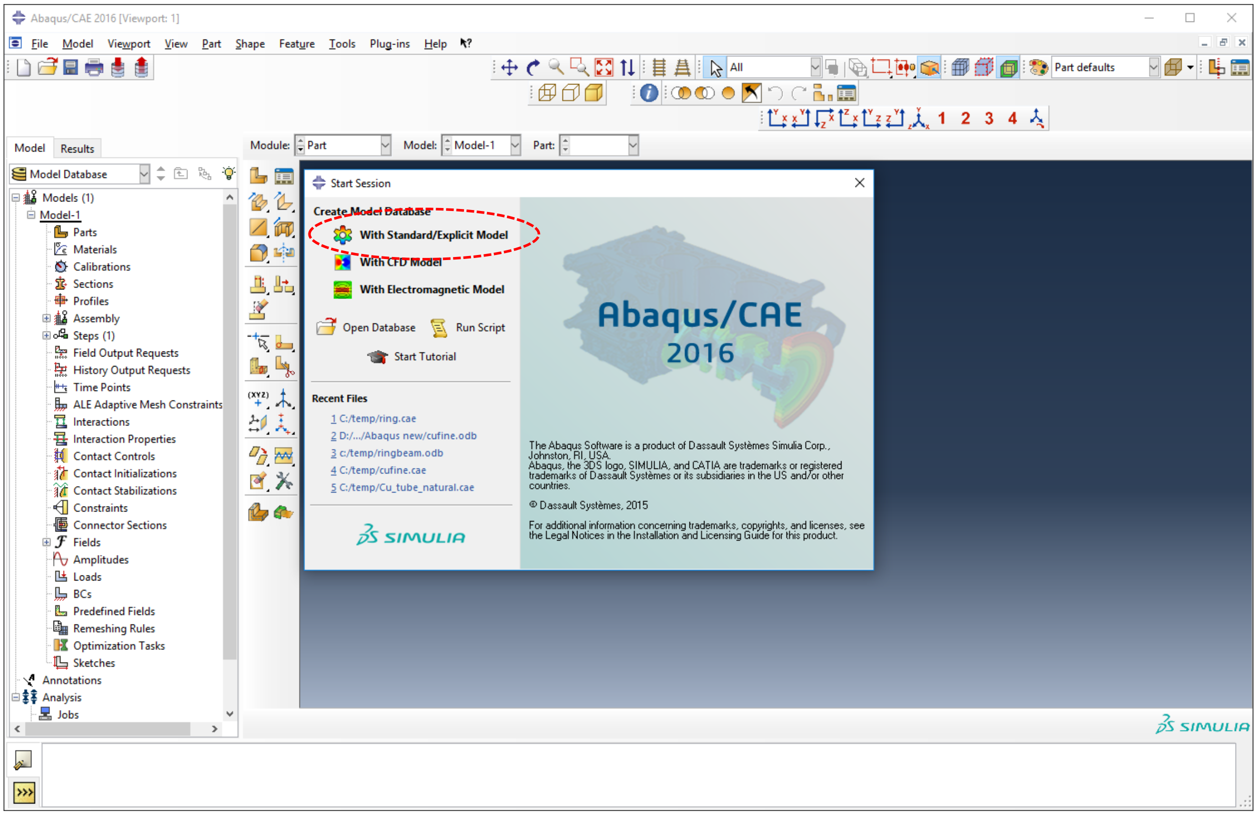
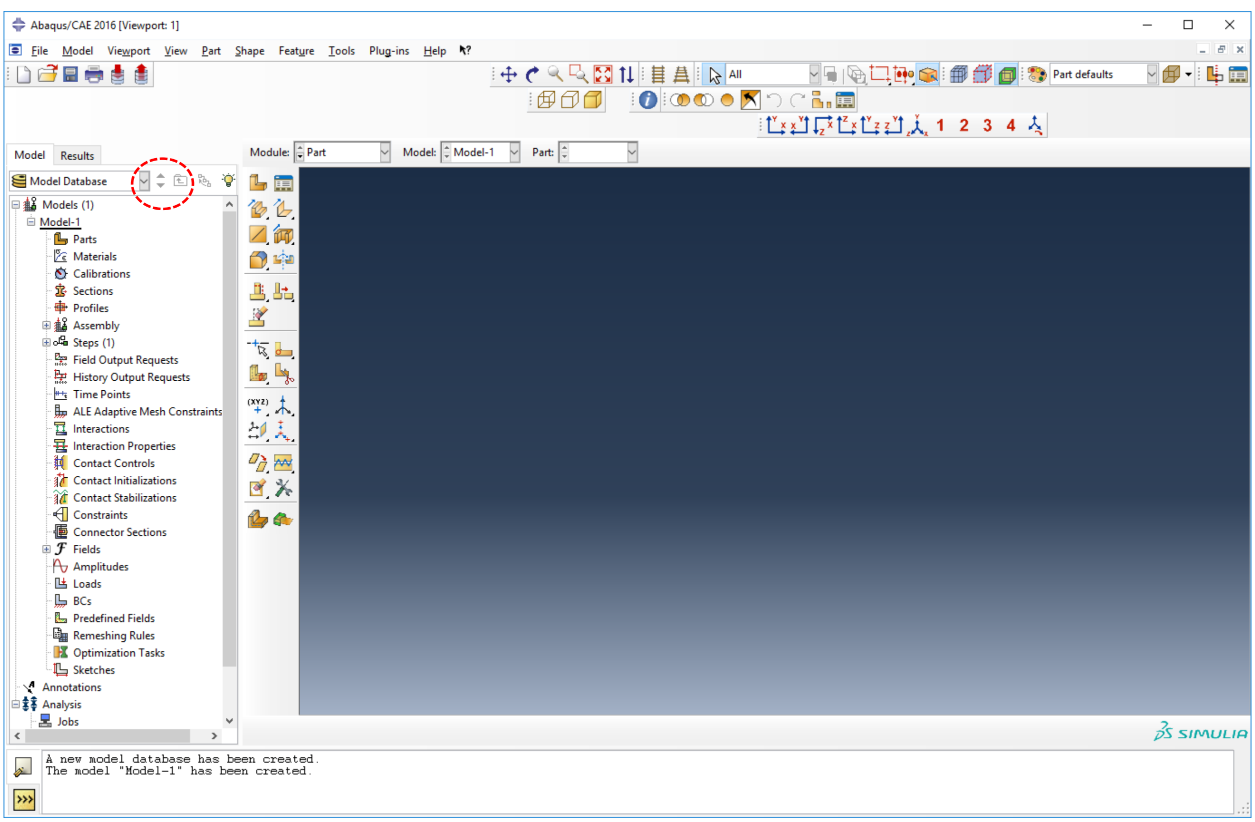
Indítsuk el az **Abaqus CAE**-t. A felugró **DOS** ablakkal ne törődjünk, de ne zárjuk be!



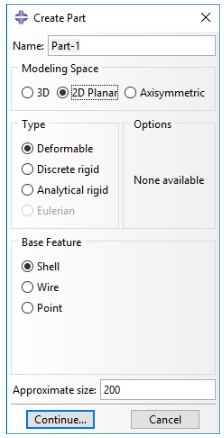
Kattintsunk a **With Standard/Explicit Model**-ra.

ALKATRÉSZEK RAJZOLÁSA

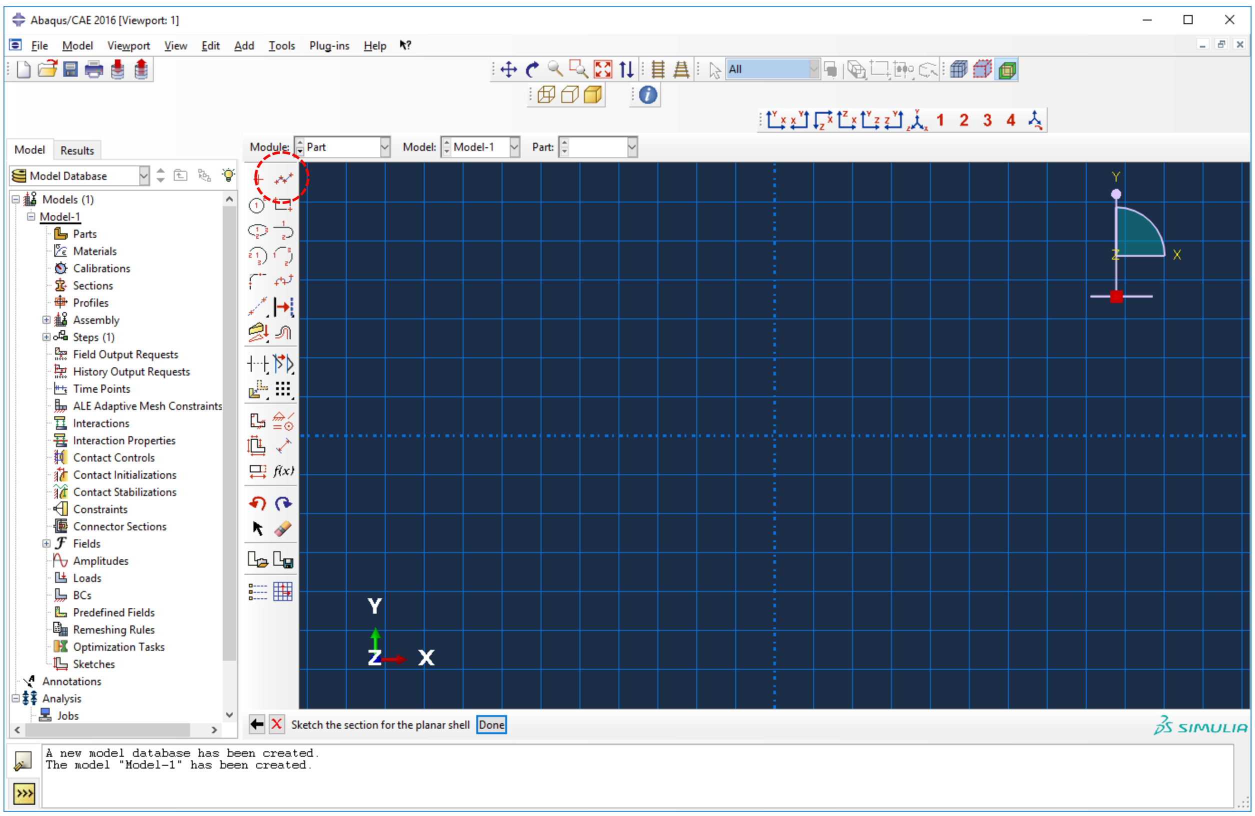
Készítsük el az első **Part**-ot. Klikk a **Create Part** ikonra:

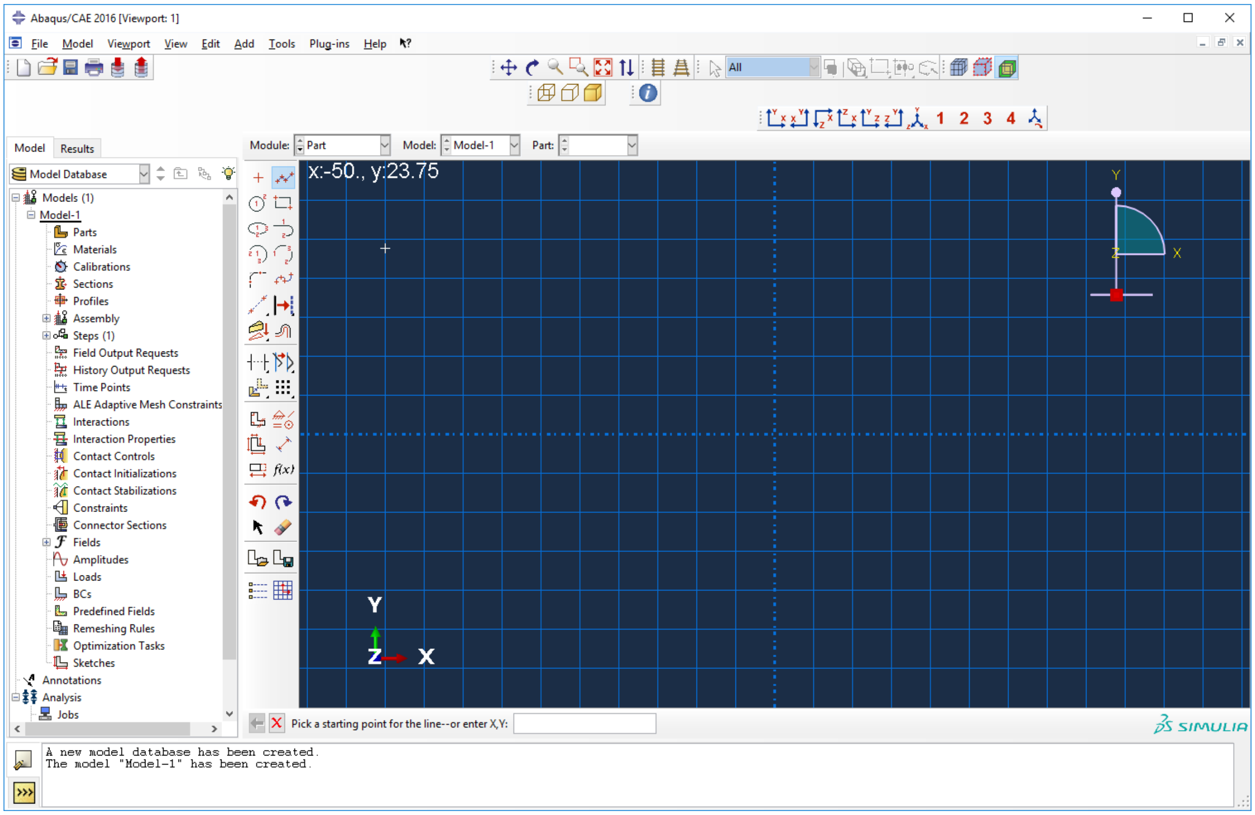


A felugró ablakban nevet adhatunk neki. Állítsuk át úgy, ahogy a lenti képen van (2D Planar) majd **Continue**.



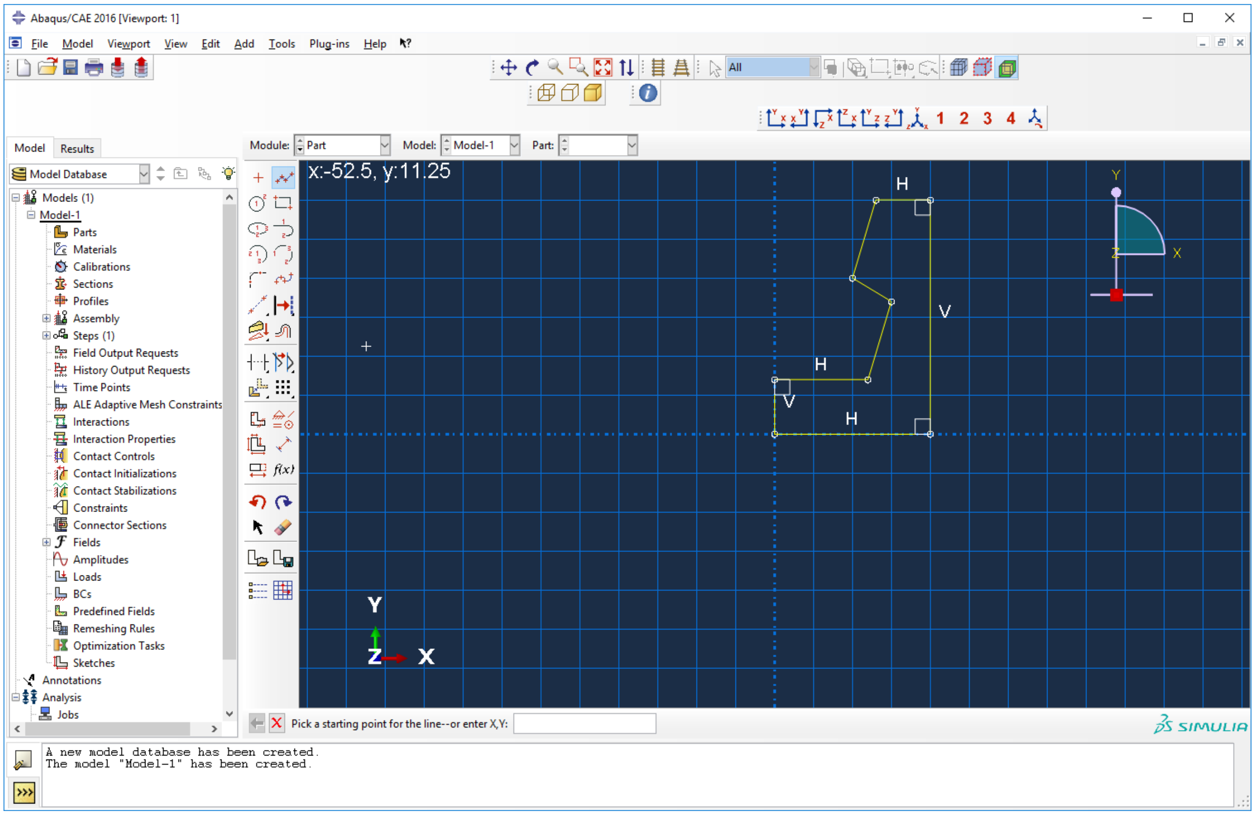
Előjön a szerkesztő ablak. Rajzoljuk meg a kontúrt alkotó egyeneseket. Klikk a **Create Lines** ikonra:



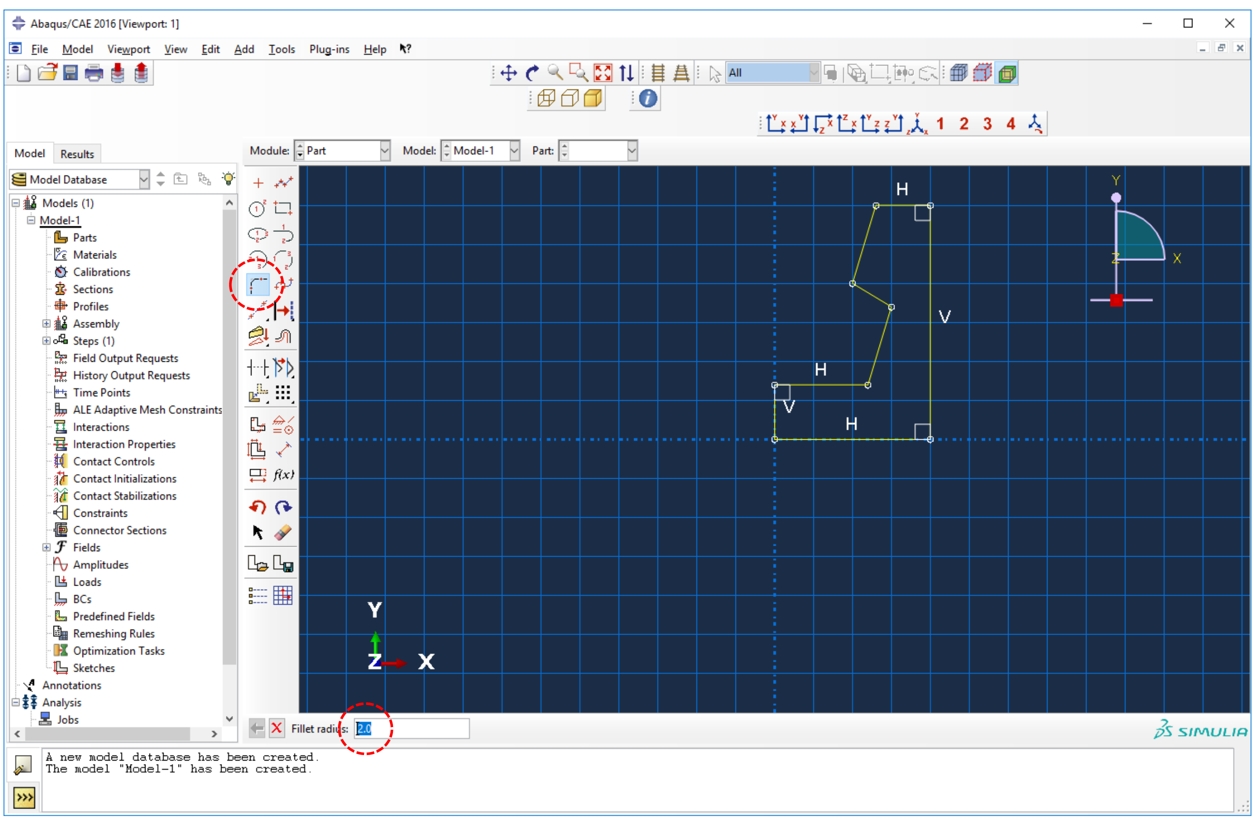


Ezt követően be lehet írni a koordinátáit az egyes pontoknak. Adjuk meg az alábbi értékeket úgy, hogy mindegyik után **Enter**-t nyomunk.

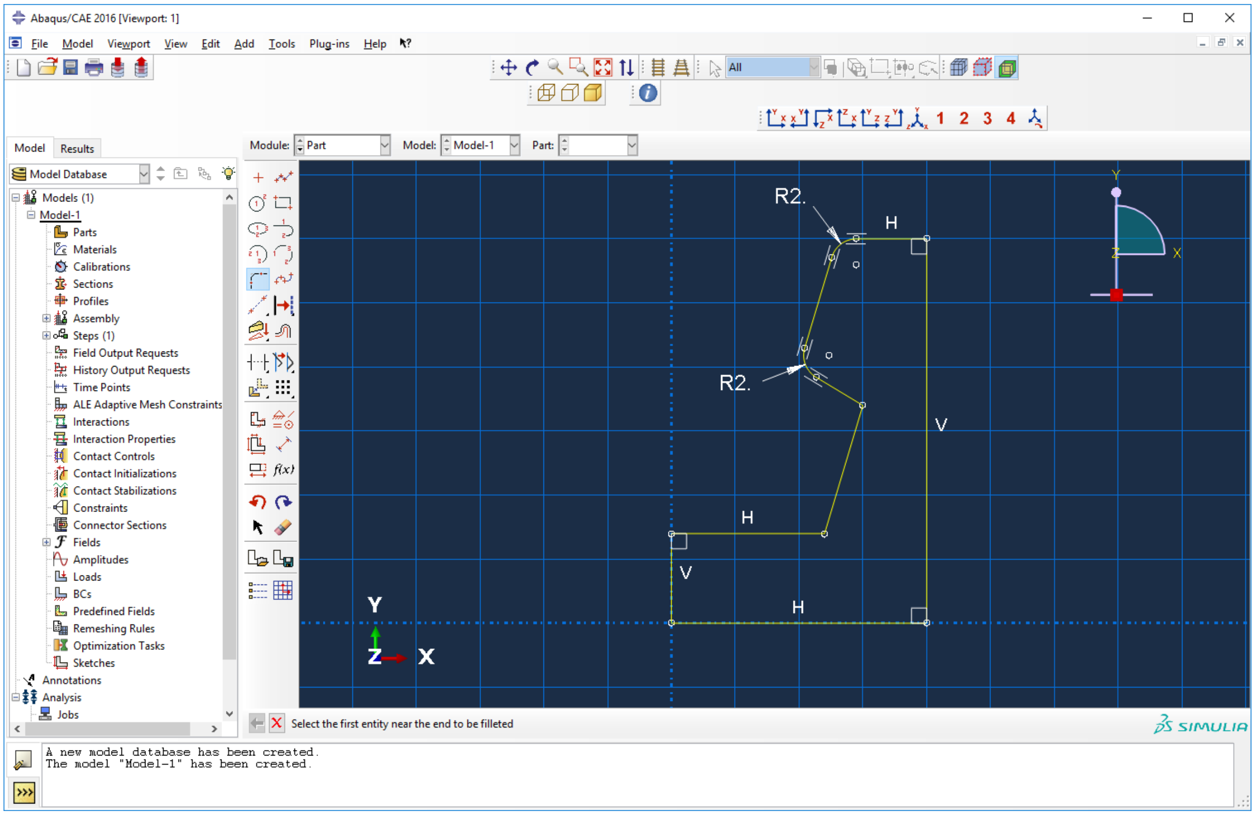
0,0
20,0
20,30
13,30
10,20
15,17
12,7
0,7
0,0



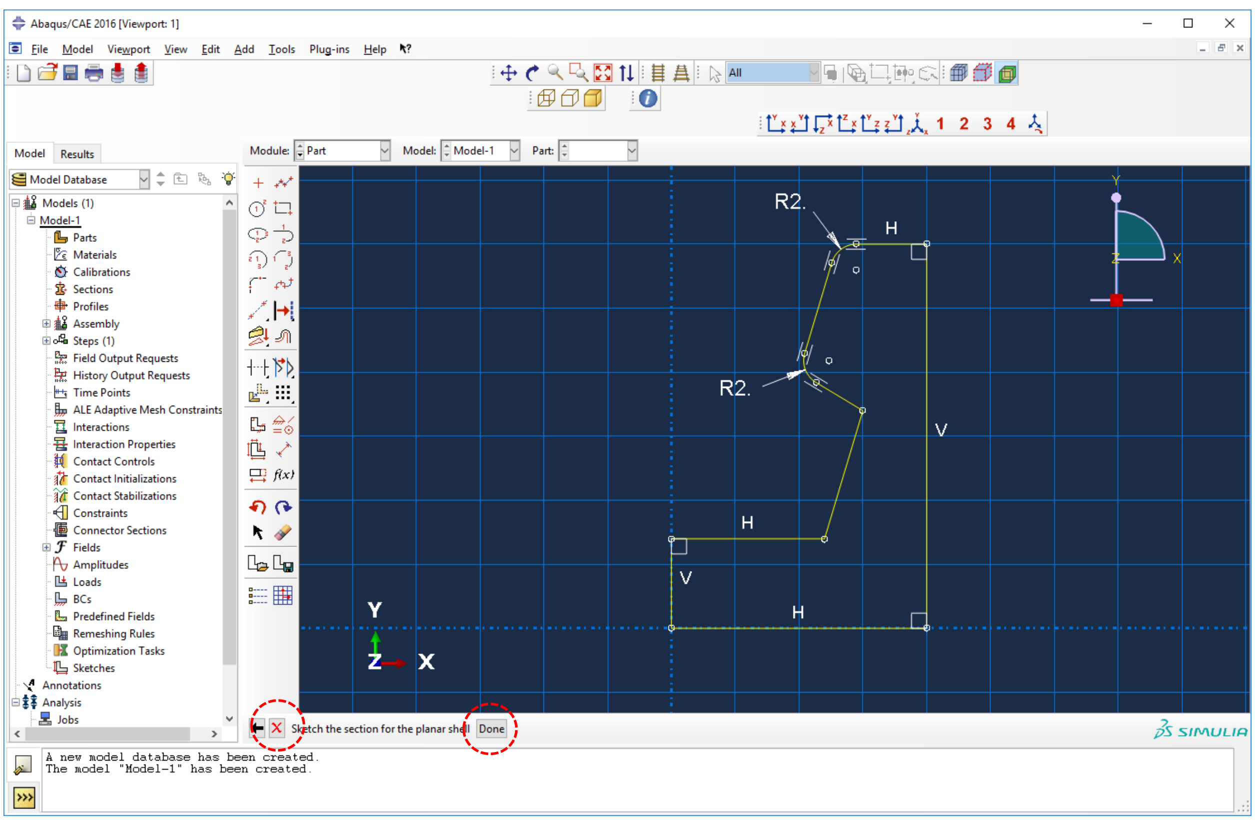
Az utolsó után készítsük el a lekerekítéseket. Katt a **Create Fillet** ikonra majd adjunk meg a 2.0 értéket.



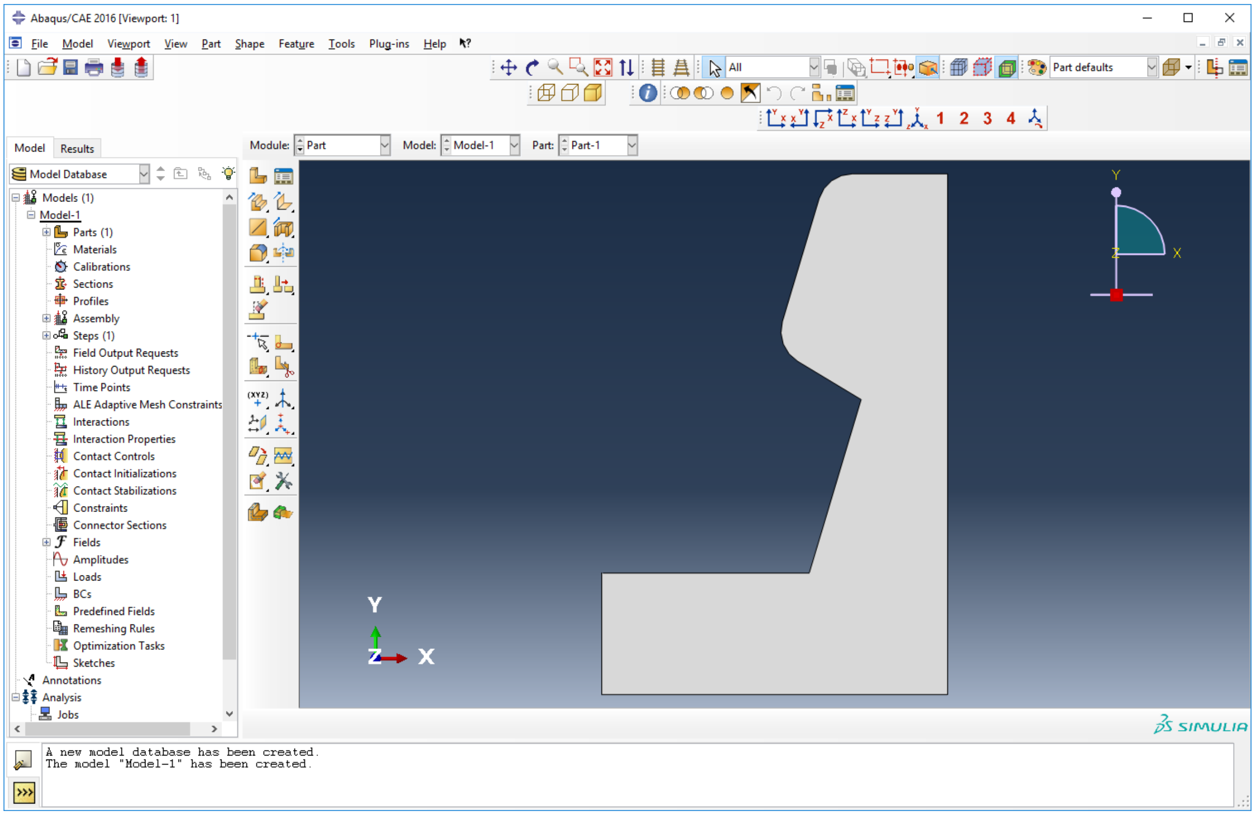
Válasszuk ki a két egyenest ami közé a lekerekítést kívánjuk szerkeszteni. Két lekerekítést készítsünk:



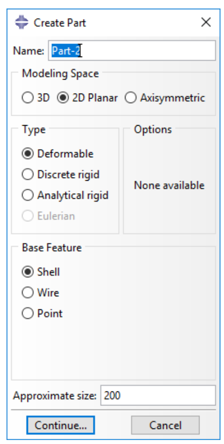
Ezt követően zárjuk be a szerkesztőt. Katt az **X**-re alul majd a **Done**-ra:



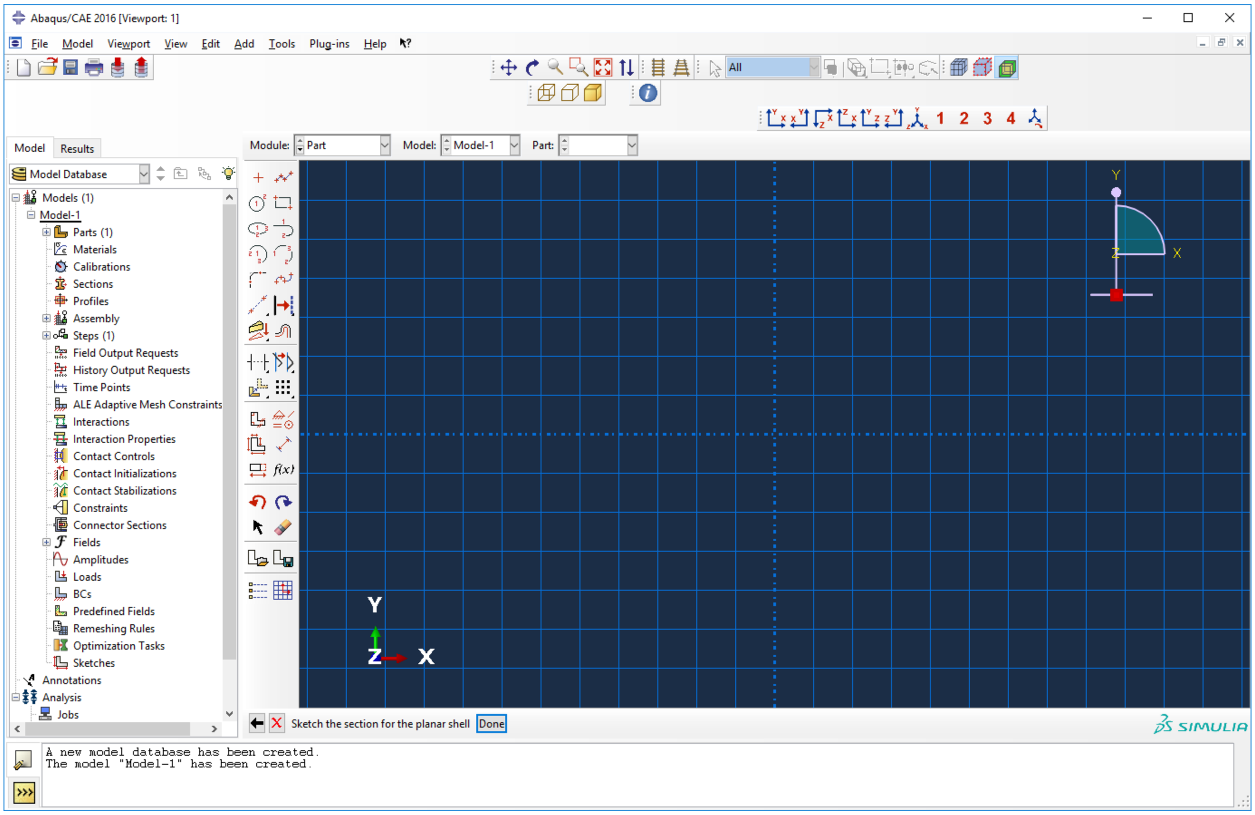
Ezzel kész az első 2D test:



Készítsük el a másikat is. Katt a **Create Part** ikonra. Ugyanolyan beállítások:

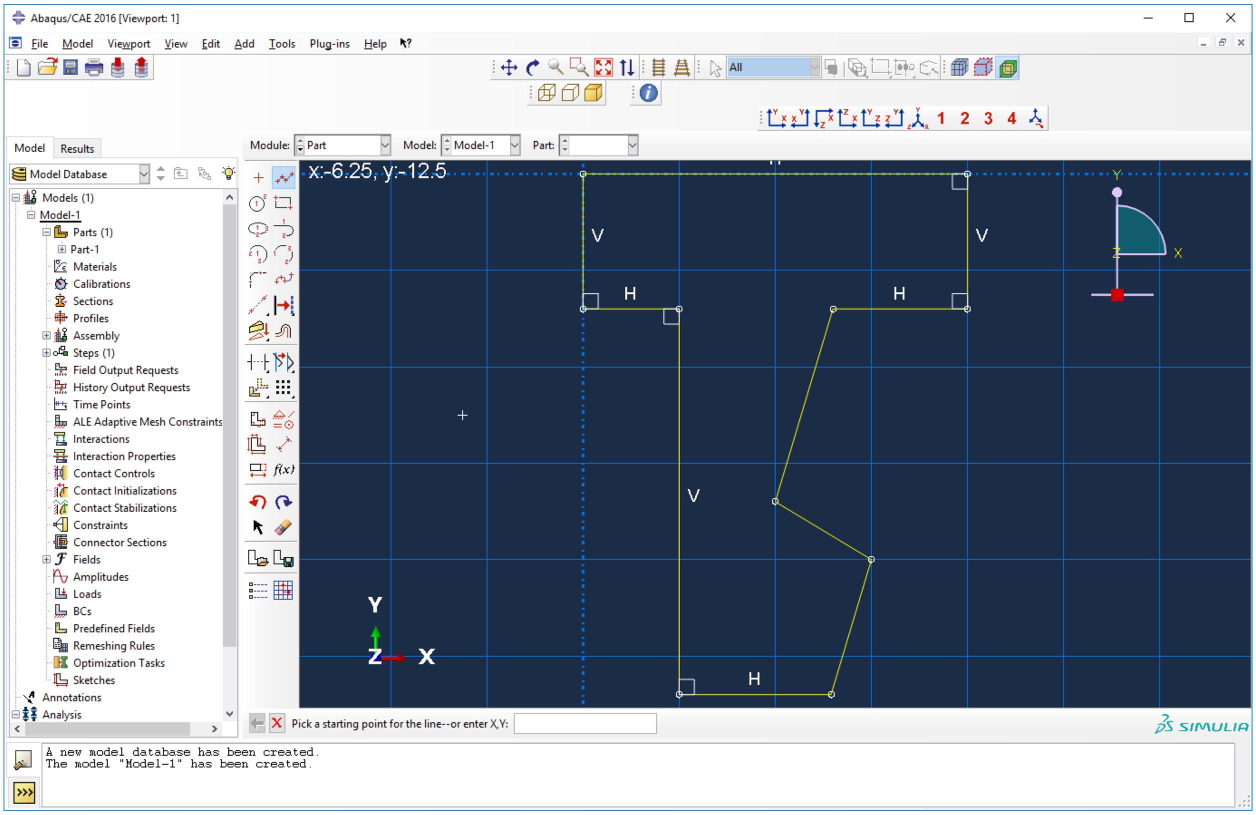


Vonalsereg készítése:

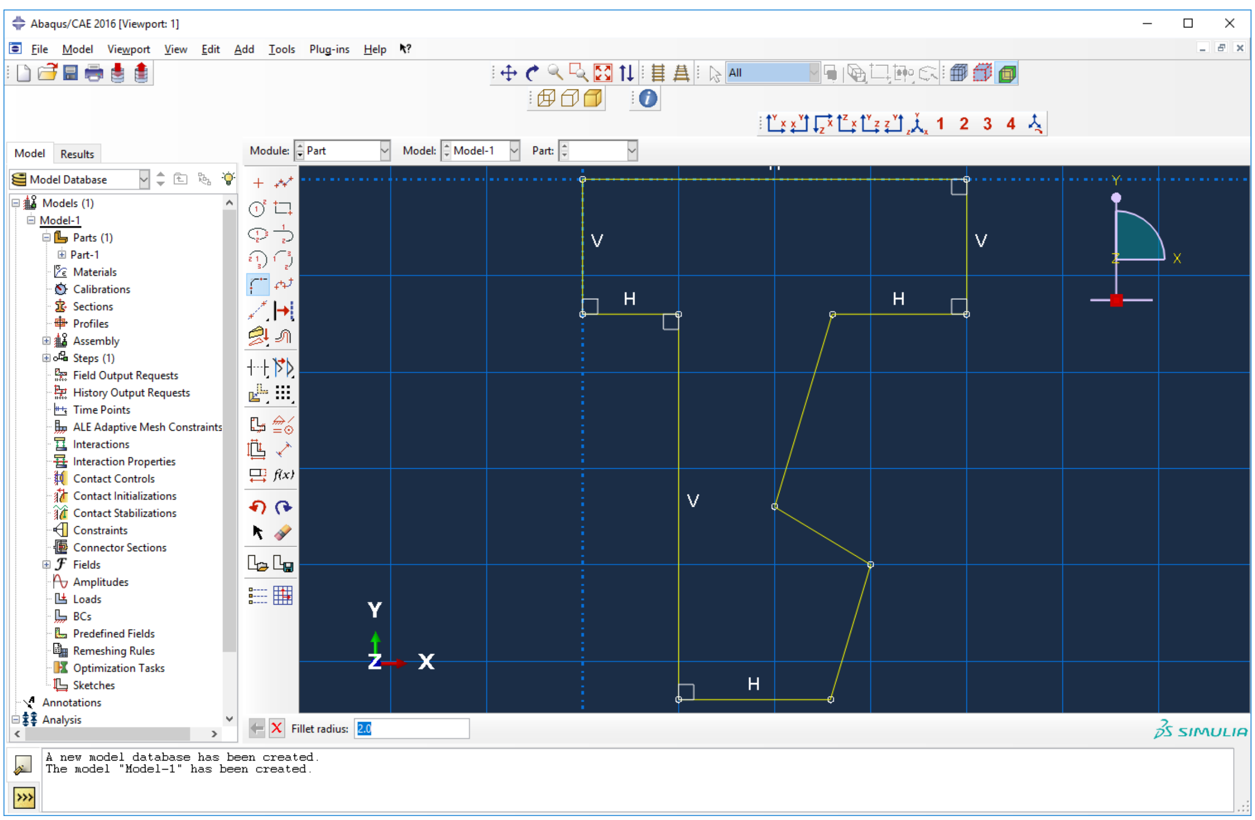


Majd az értékek megadása:

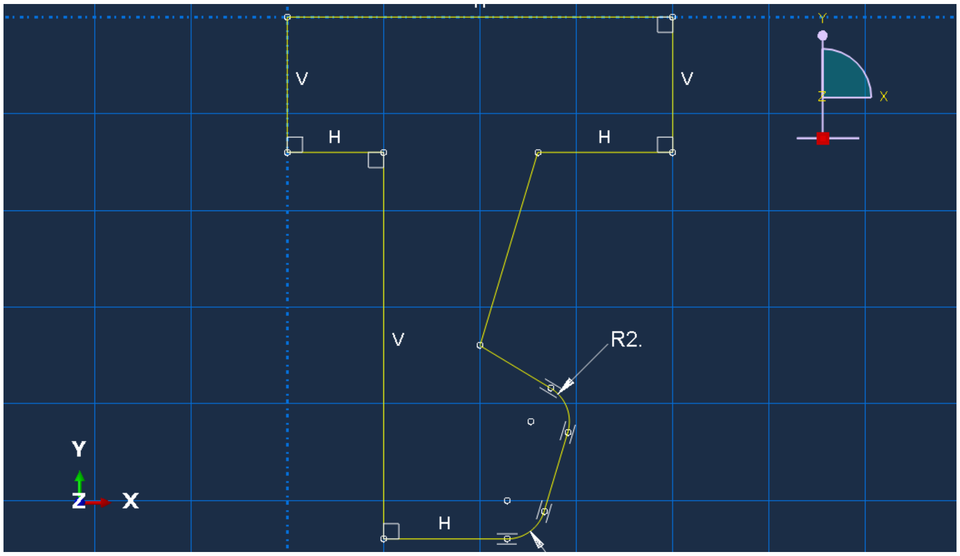
0,0
20,0
20,-7
13,-7
10,-17
15,-20
12.9,-27
5,-27
5,-7
0,-7
0,0



Ha megvan akkor lekerekítések szintén 2-es sugárral. 2-t beírni majd **Enter**.



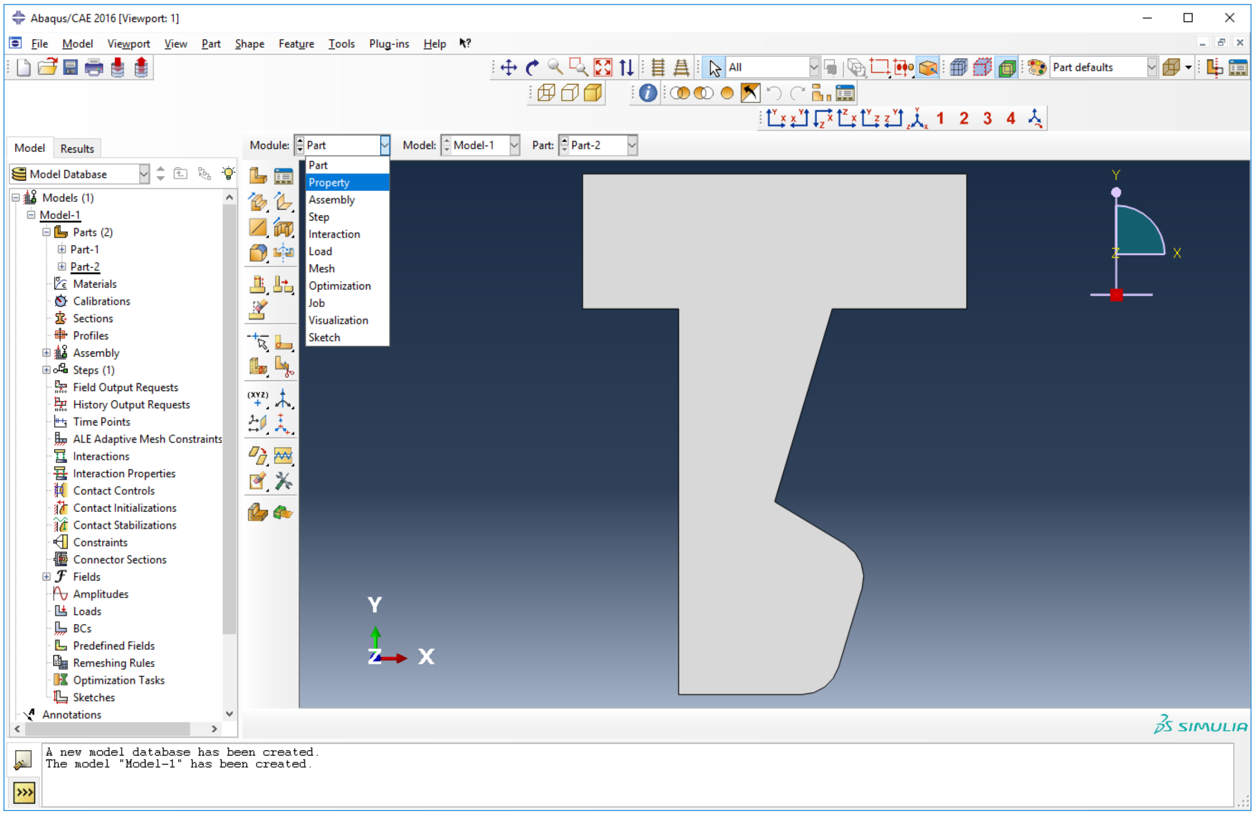
Lekerekítés helyeinek megadása:



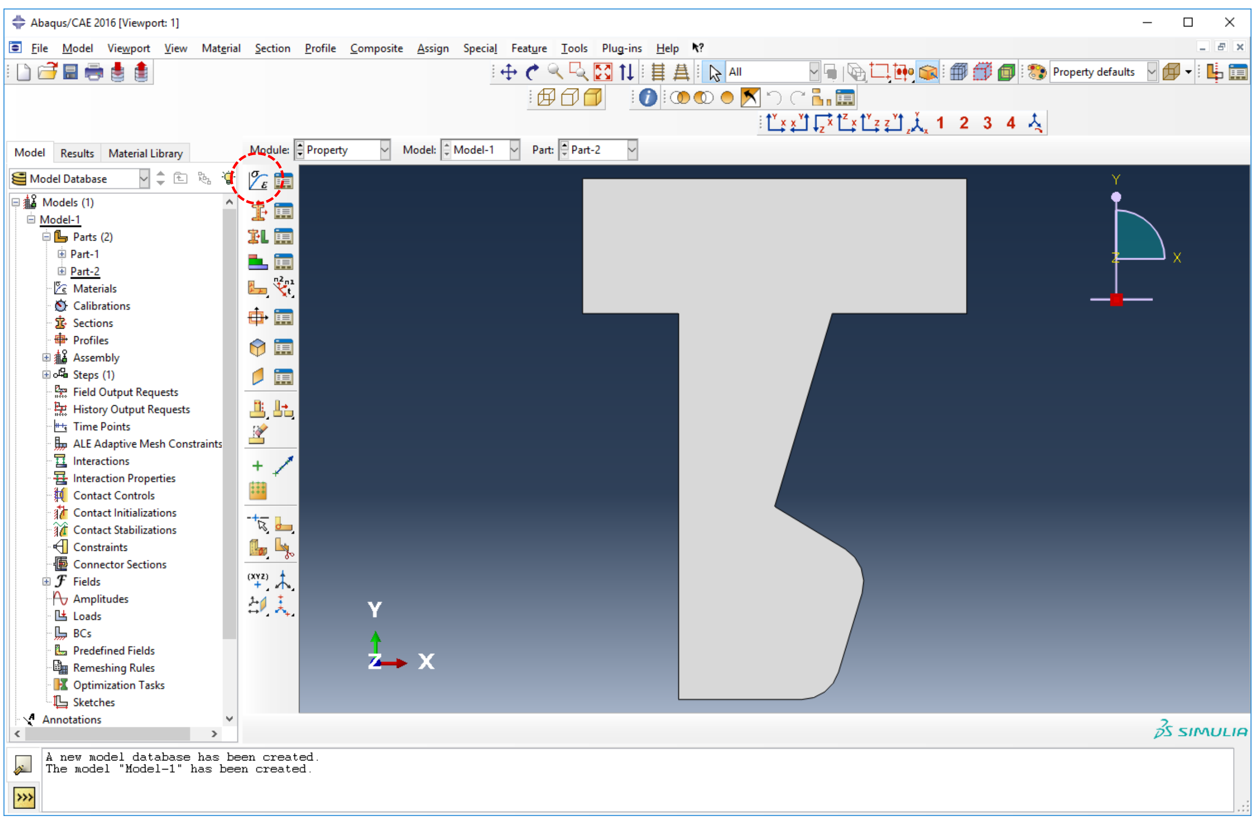
Ha megvan akkor X aztán Done. Kész a második test:

ANYAGMODELL

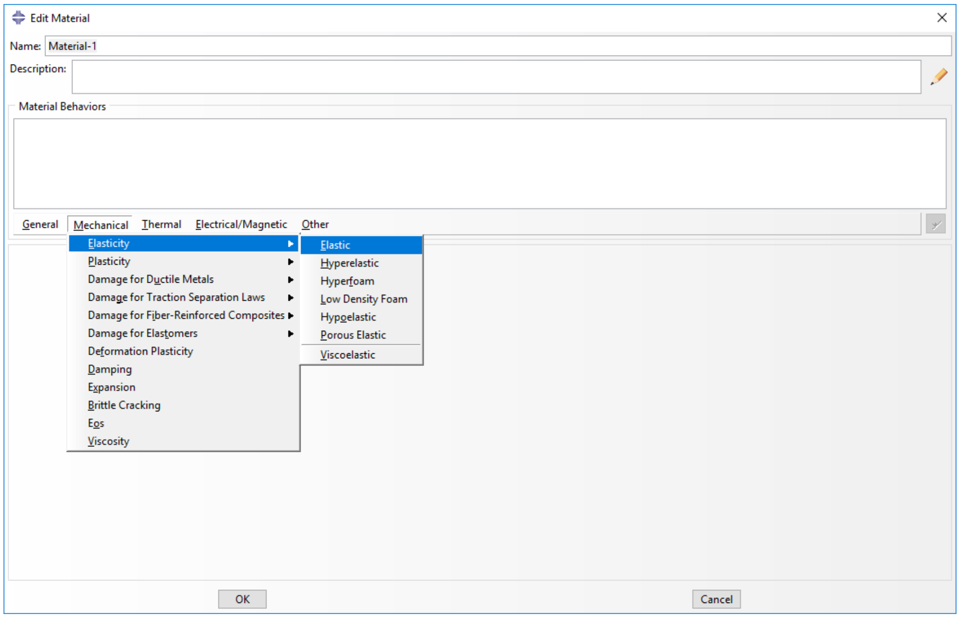
A legördülő menüből válasszuk ki a **Property**-t:



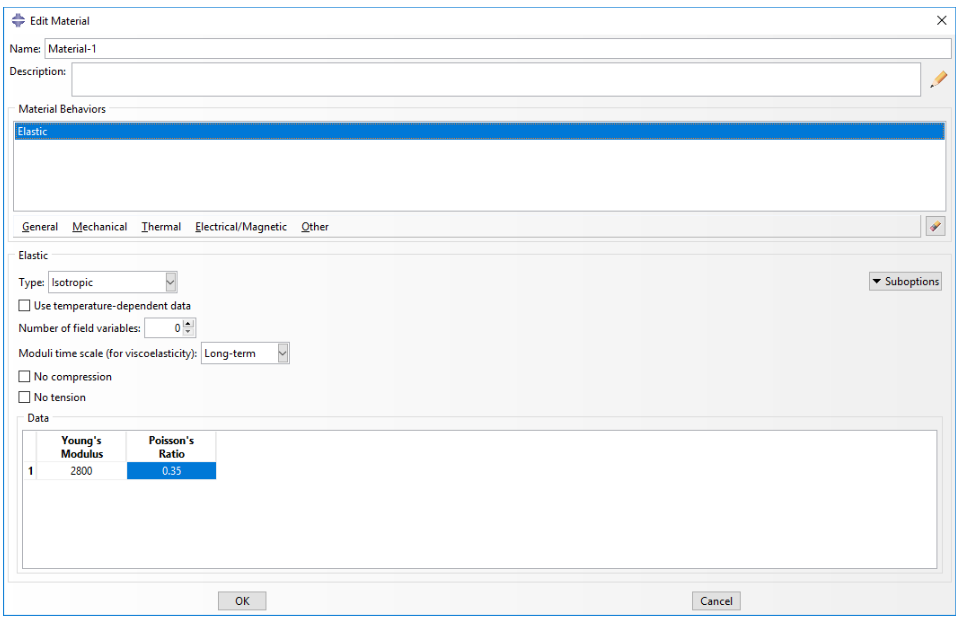
Ezt követően más ikonsor lesz:



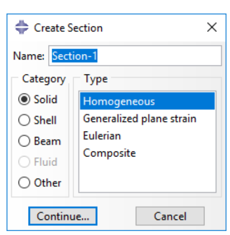
Kattintsunk a **Create Material** gombra: . Felugrik egy új ablak. Menüből az **Elastic** kiválasztása:



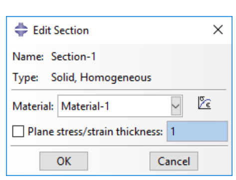
Felugró ablakban írjunk be **2800**-t és **0.35**-t a rugalmassági moduluszhoz és a Poisson-tényezőhöz:



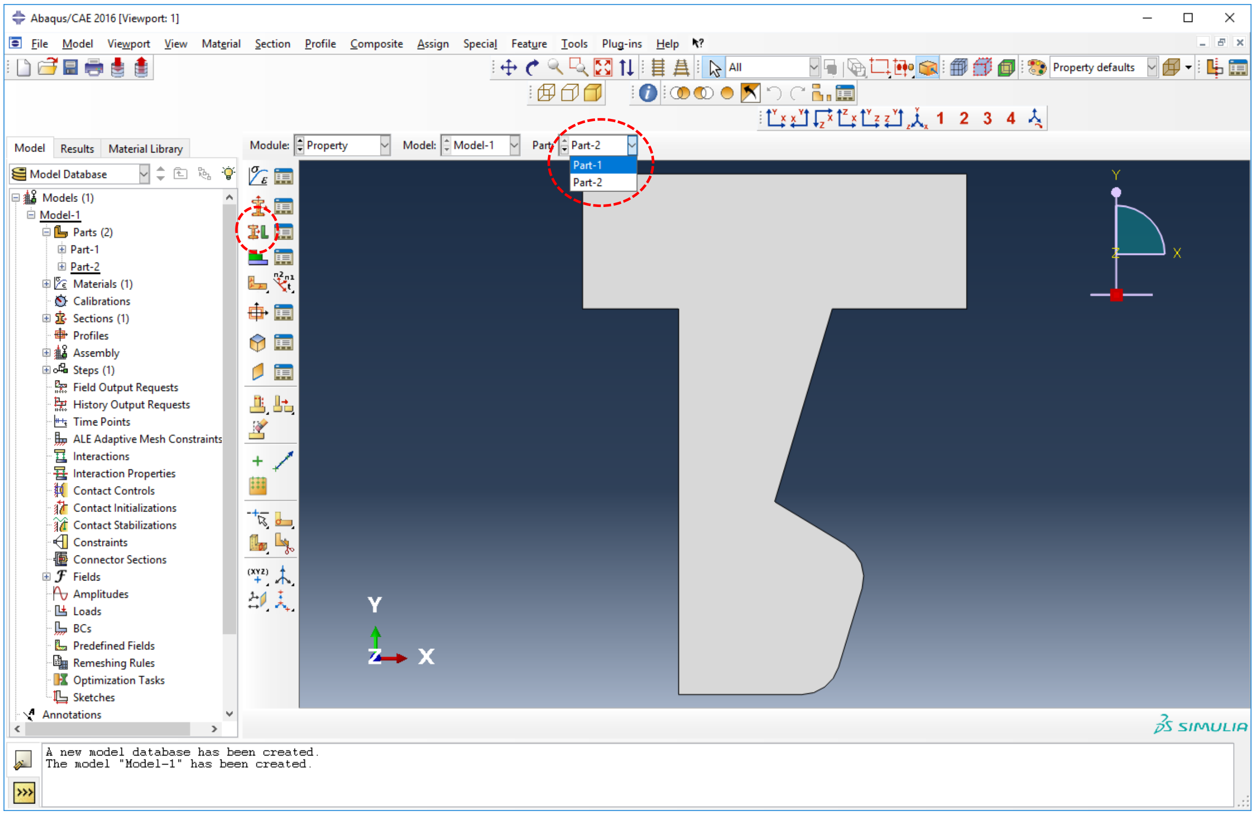
Majd **OK**. Készítsünk egy **Section**-t. Katt az  ikonra. Felugró ablak:



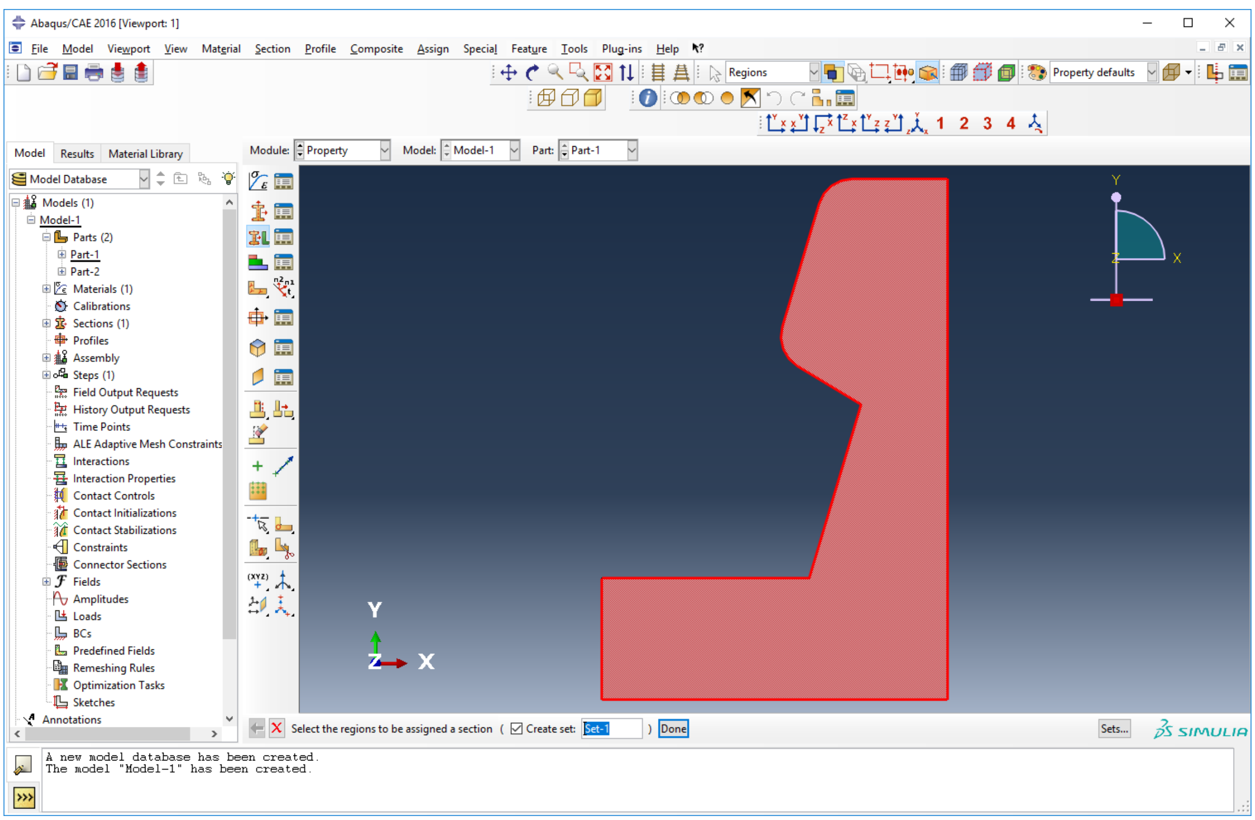
Maradhat így, **Continue**. Új felugró ablak, fogadjuk el, **OK**.



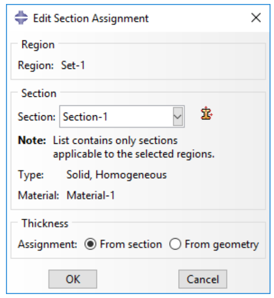
Legördülő menüből válasszuk ki a **Part 1**-t:



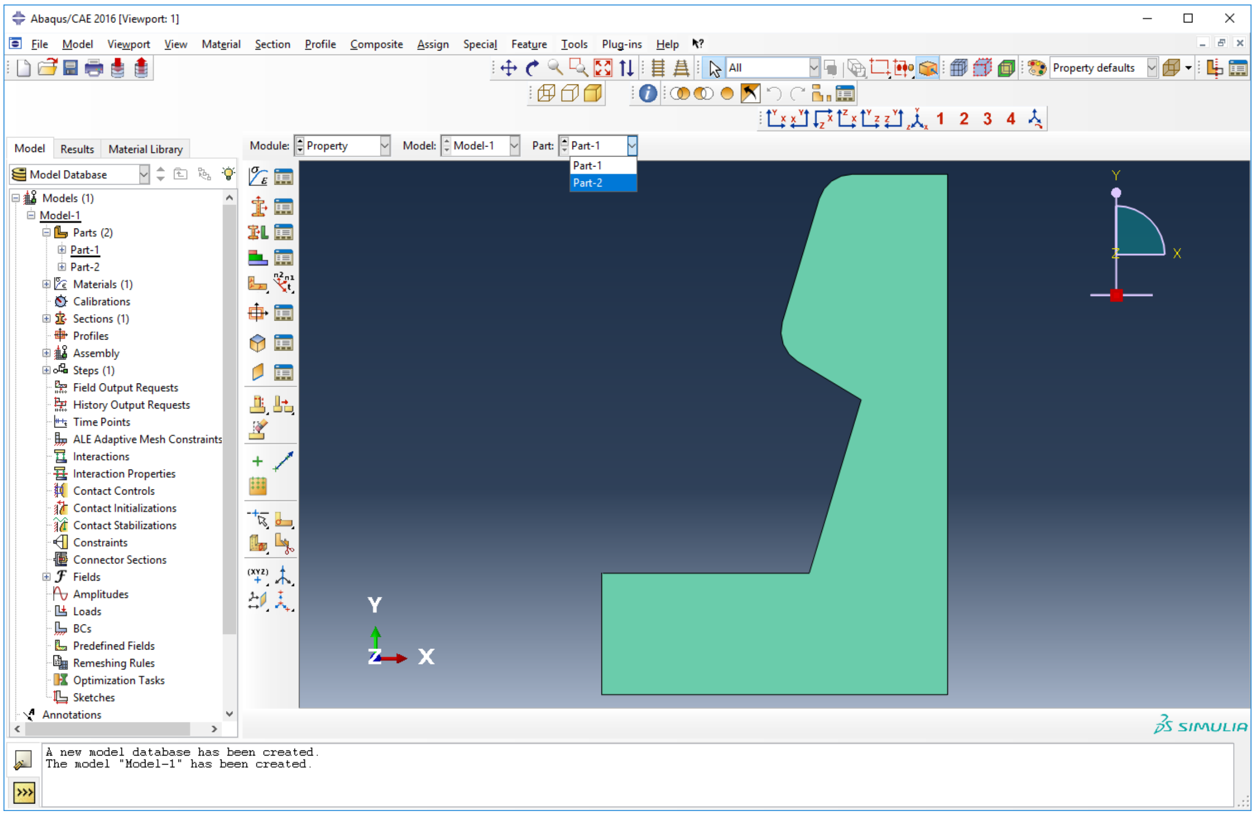
Ezt követően az **Assign Section** ikon: . Kéri, hogy jelöljük ki a tartományt. Katt egyszer a testen belülre (kijelöli pirossal):



Ezt követően **Done** gomb alul: Done. Felugró ablakot fogadjuk el és **OK**.



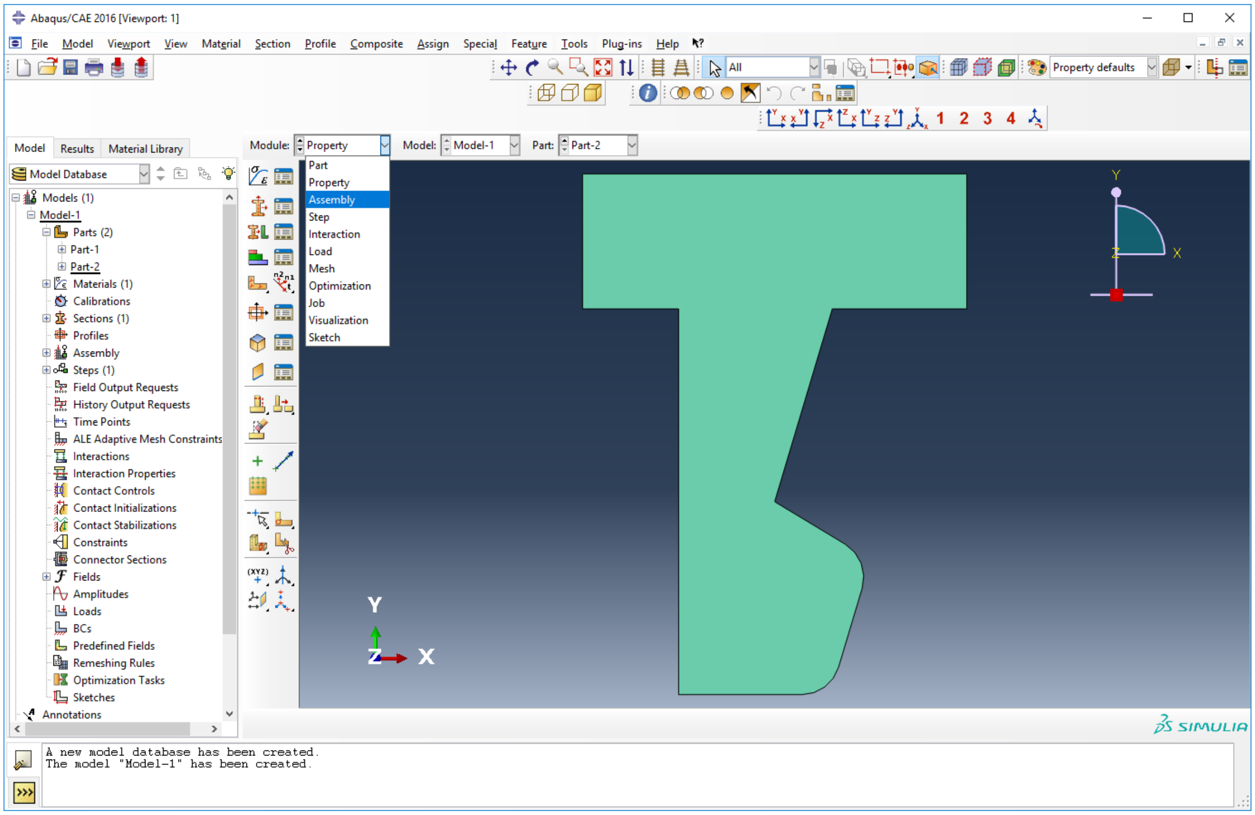
A Part 2 esetén is végezzük el a fenti lépéseket:



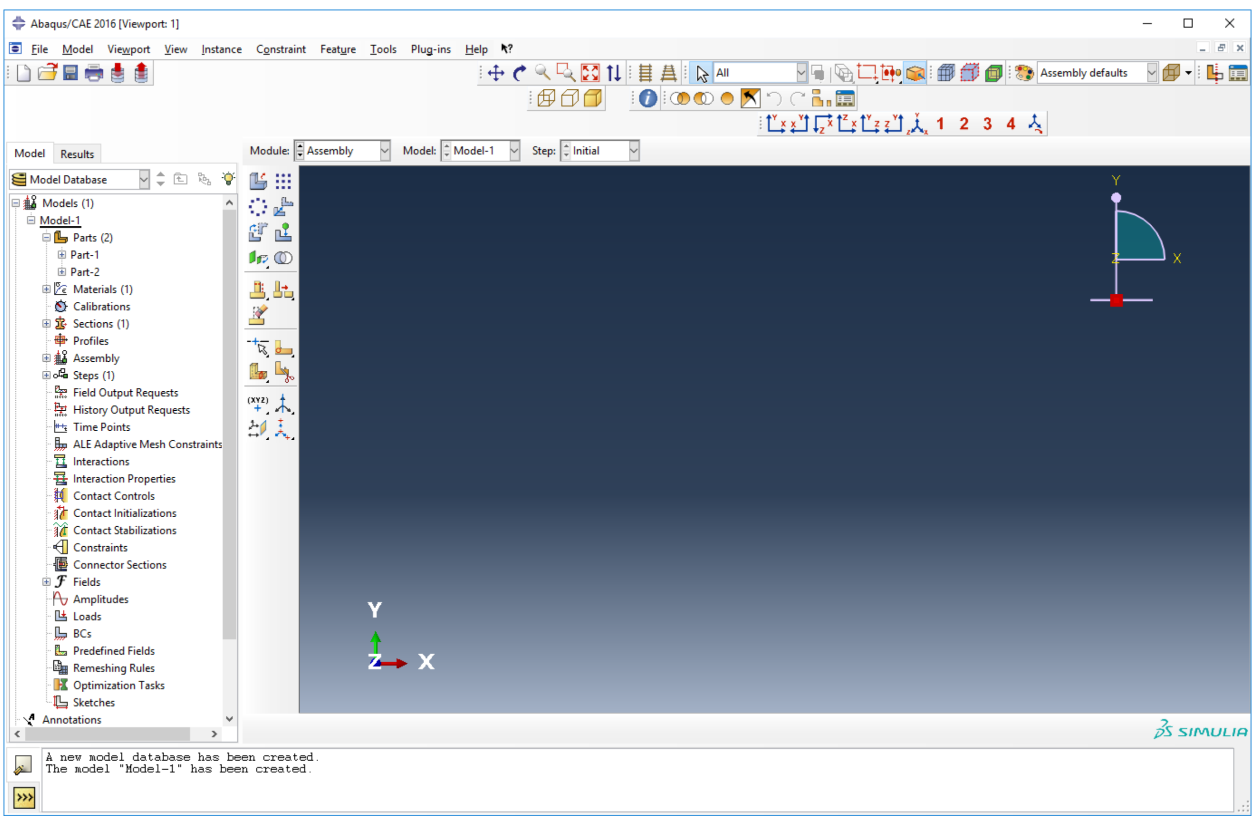
Majd ikon. Belekattintani valahova és **Done**. Felugró ablakban **OK**. Ezzel kész az anyag hozzárendelése.

<u>ÖSSZEÁLLTÁS</u>

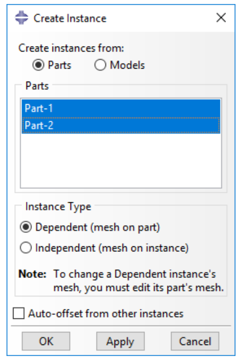
Assembly modul kiválasztása:



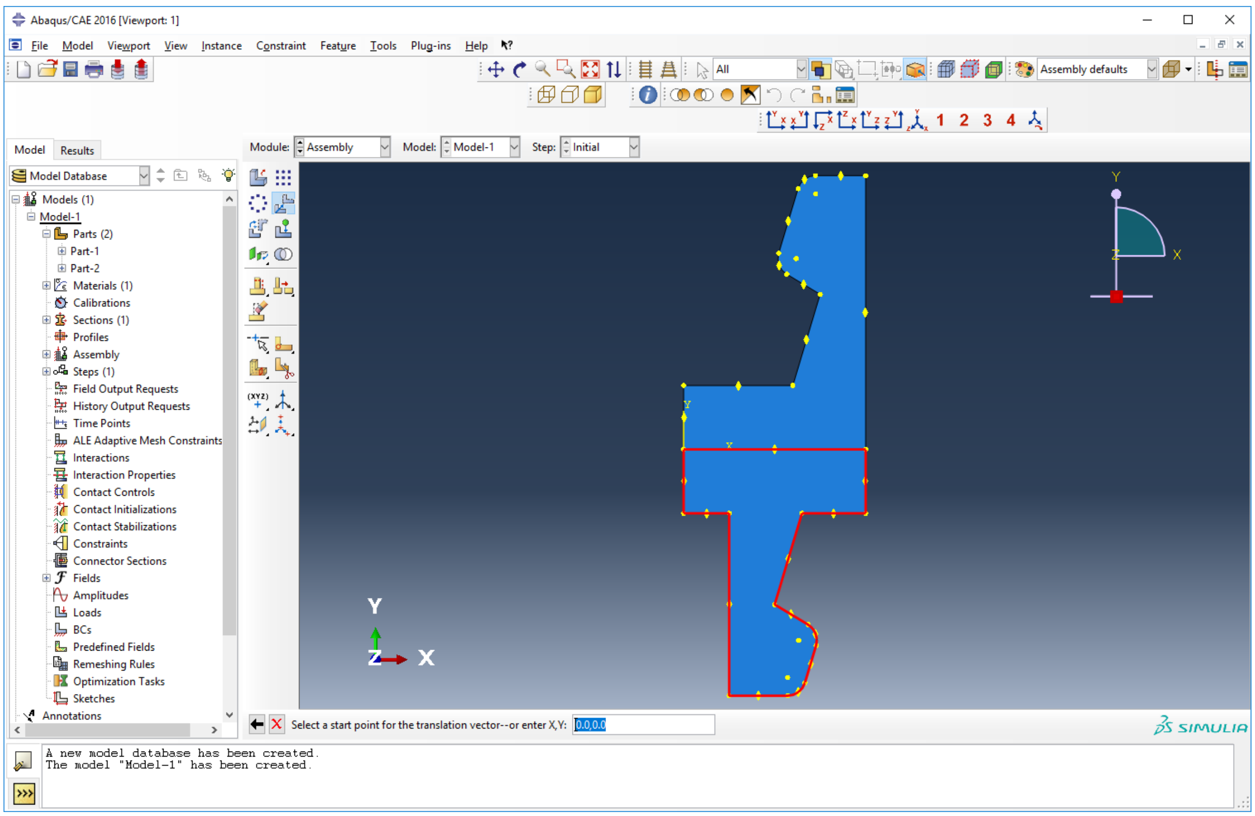
Megváltozik az ikonsor:



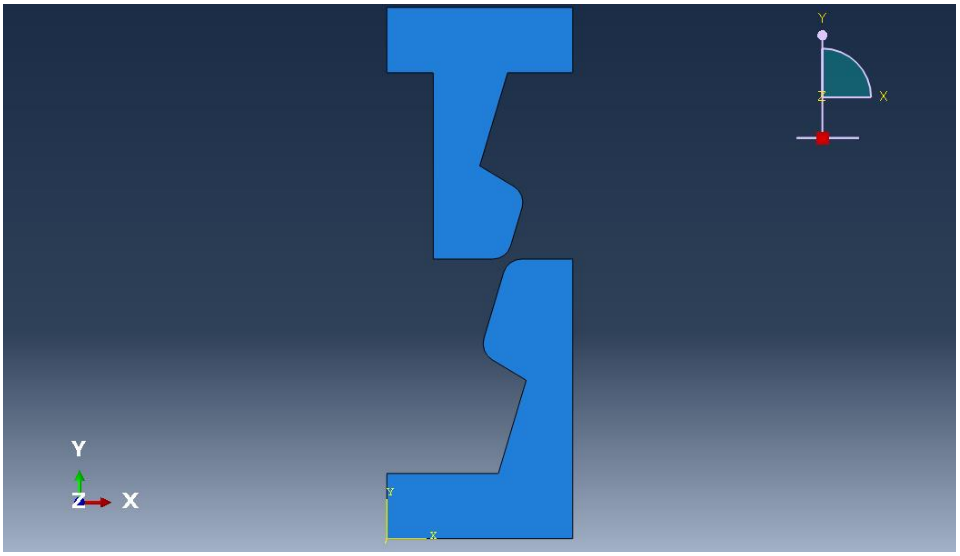
Illesszük be a két alkatrészt. Katt az ikonra. Felugró ablakban nyomjuk le a shift-et, majd egérrel válasszuk ki mindkét alkatrészt, majd **OK**.



Mozgassuk feljebb a fentre kerülő alkatrészt. Katt a **Translate Instance** ikonra: . Kéri, hogy válasszuk ki amit mozgatni akarunk. Egyszer katt az alsó alkatrészre majd alul **Done**. Ekkor kéri az elmozdulás vektor kezdő és végpontját. Kezdő pont legyen **0,0**:

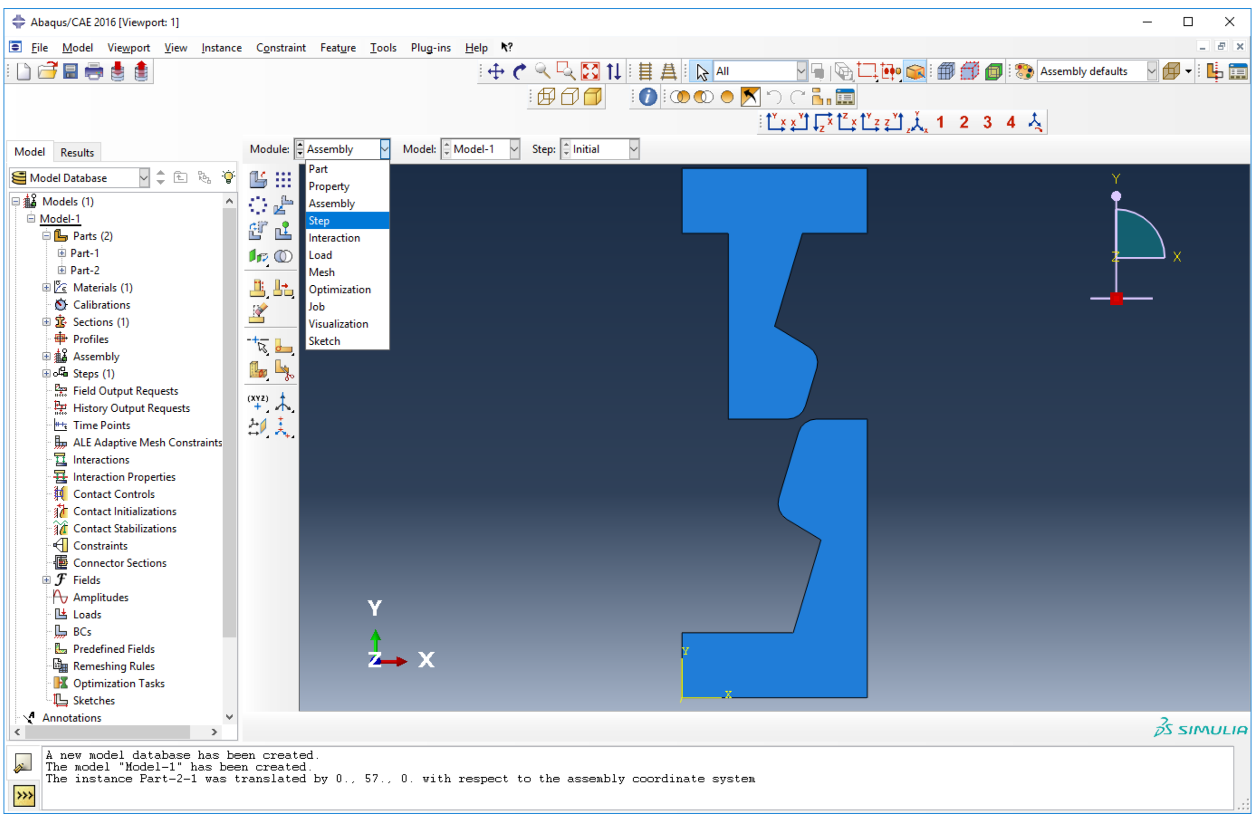


Majd **Enter**, aztán a végpont: **0,57**. **Enter**. Majd **OK** gomb. Katt az **Auto-Fit View** ikonra , hogy lássunk mindent.

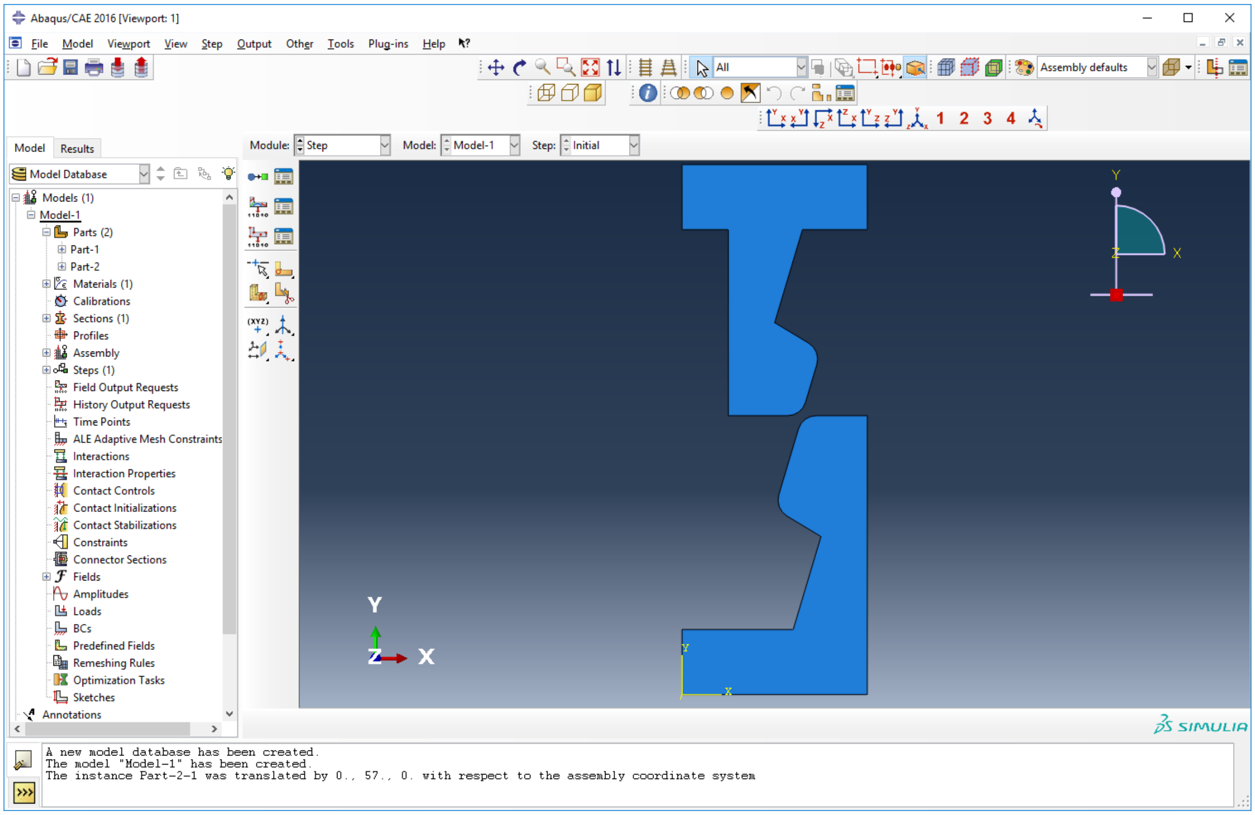


STEP

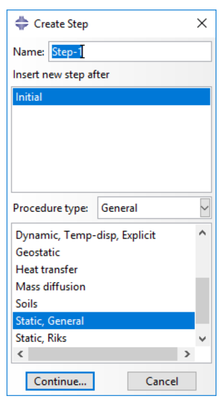
Step modul kiválasztása:



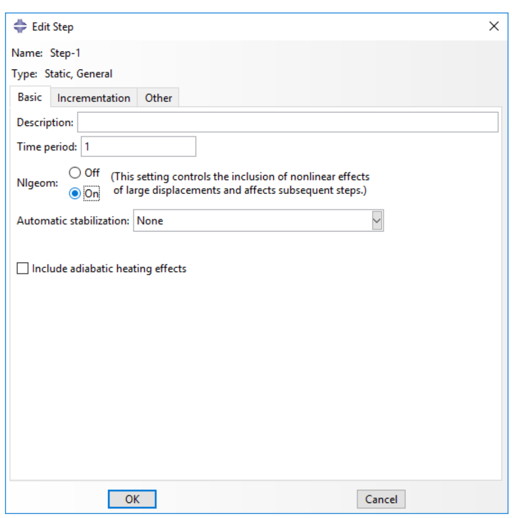
Megváltozik az ikonsor:



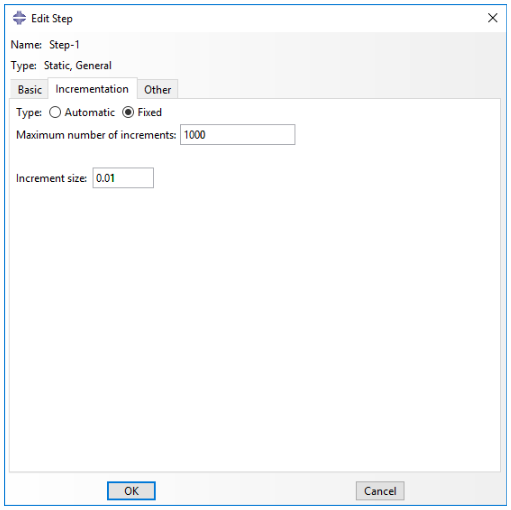
Katt a **Create Step** ikonra: . Felugró új ablak:



Fogadjuk el. **Continue** gomb. Felugró új ablakban kattintsunk át **On**-ra az **Nlgeom** opciót:

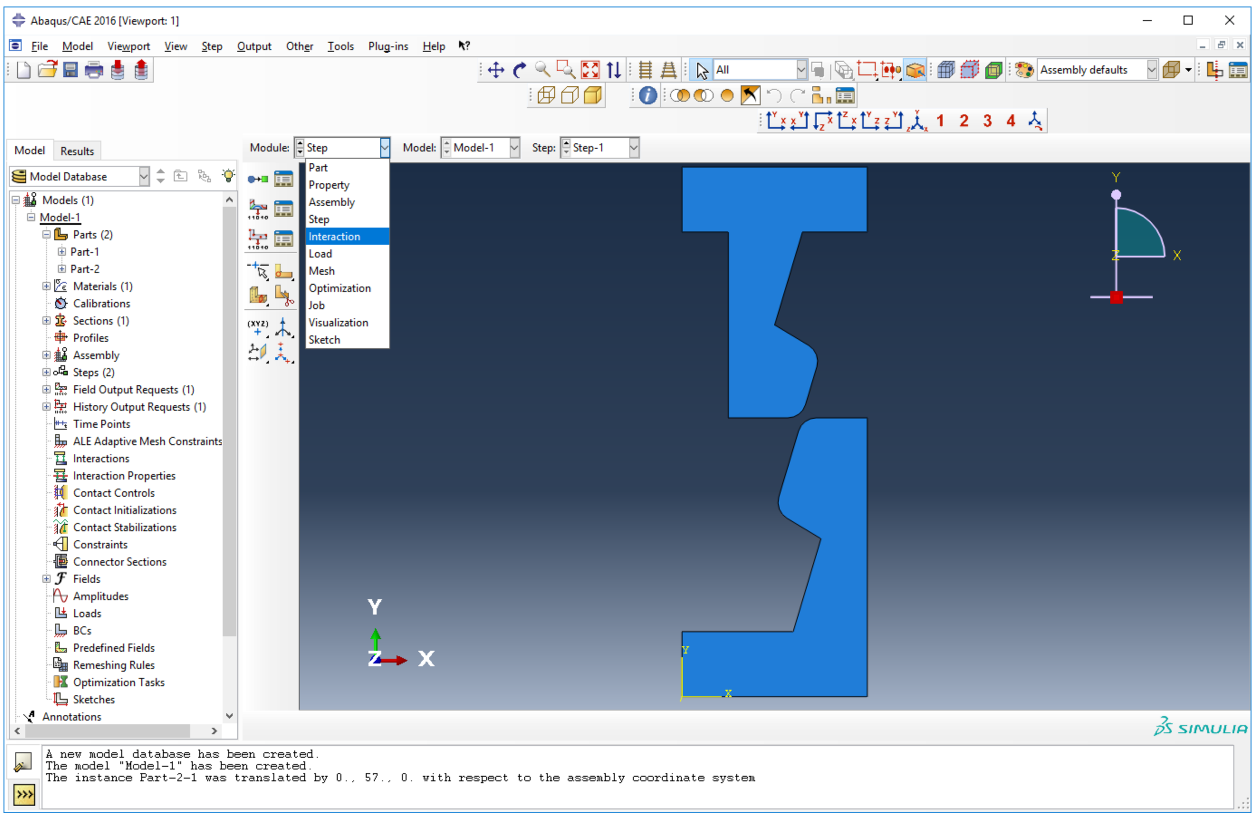


Aztán katt az **Incrementation** fülre. Váltsunk át **Fixed** típusra. A maximum increment számot írjuk át 1000-re. Az **increment size** pedig legyen 0.01. Majd **OK**.

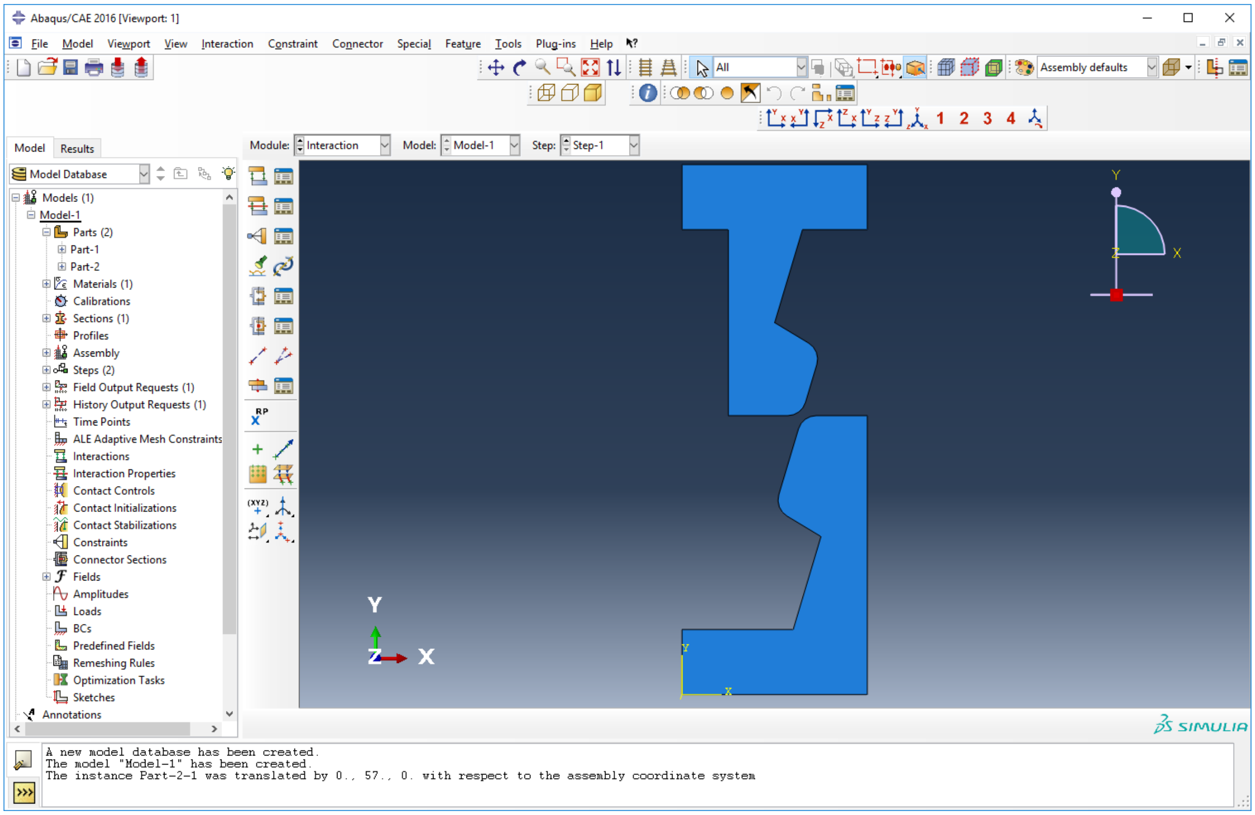


KONTAKT

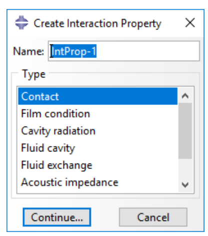
Váltsunk át az **Interaction** modulra:



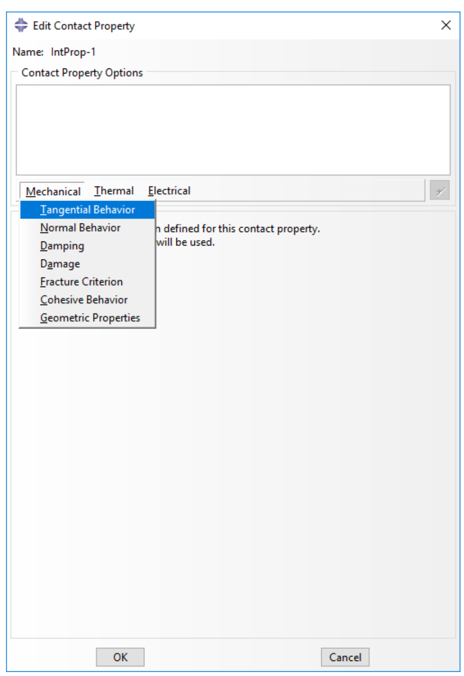
Megváltozik az ikonsor:



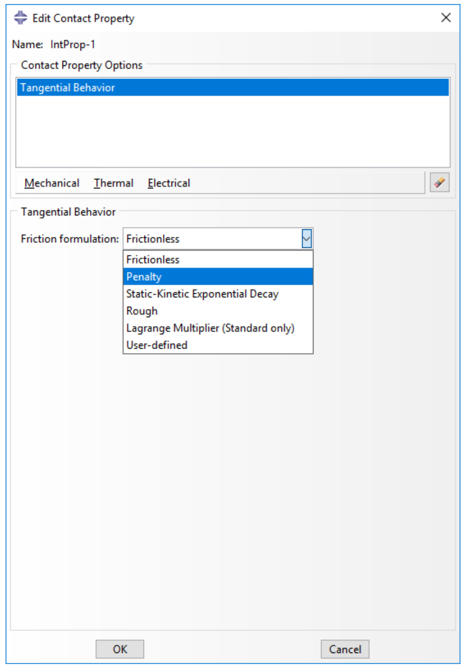
Katt a **Create Interaction Property**-re: . Felugró ablak:



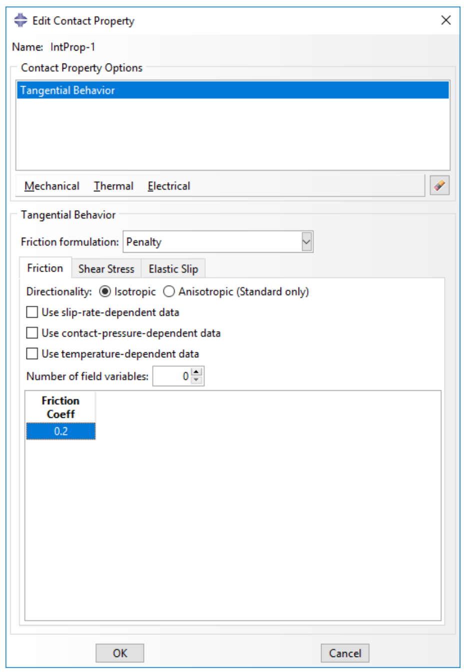
Continue. Felugró ablakban **Mechanical / Tangential Behavior** menü:



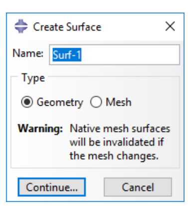
Az új ablakban a legördülőből válasszuk ki a **Penalty**-t:



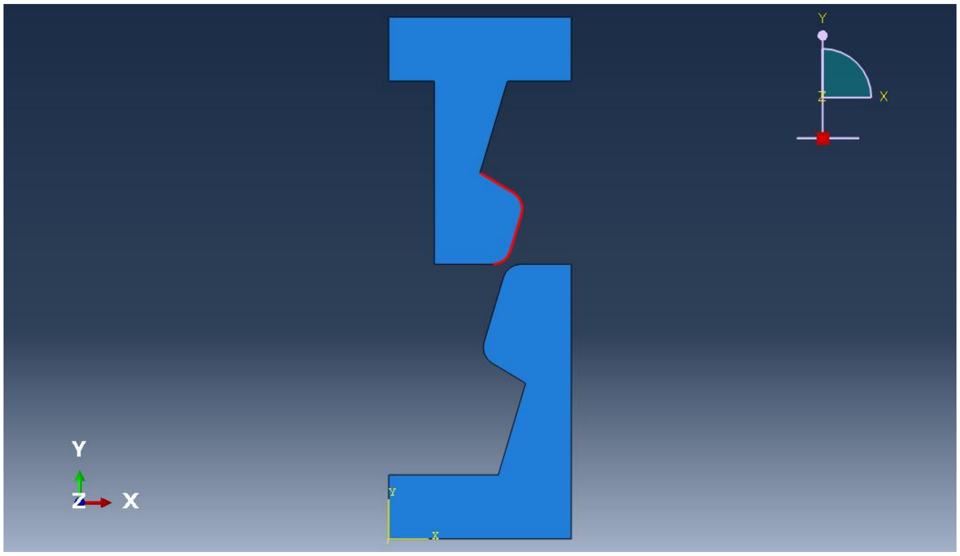
Adjuk meg a 0.2 értéket. Majd **OK**.



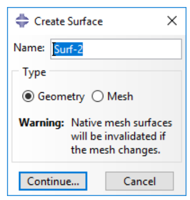
A program fenti menüsorából **Tools / Surface / Create**. Új ablak jelenik meg:



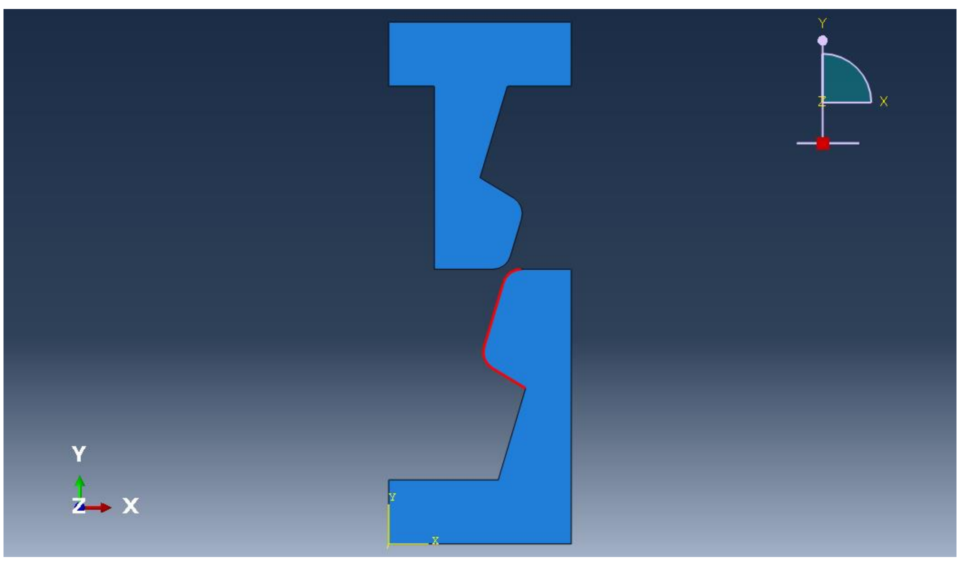
Continue. Válasszuk ki a felső alkatrészen lévő kontaktba kerülő éleket. Majd alul **Done**.



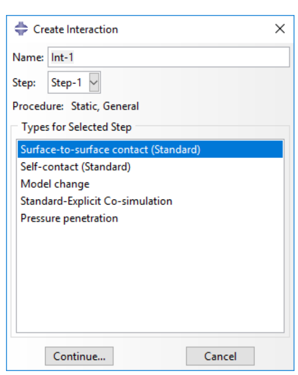
A fenti menüsorból **Tools / Surface / Create**. Új ablak jelenik meg:



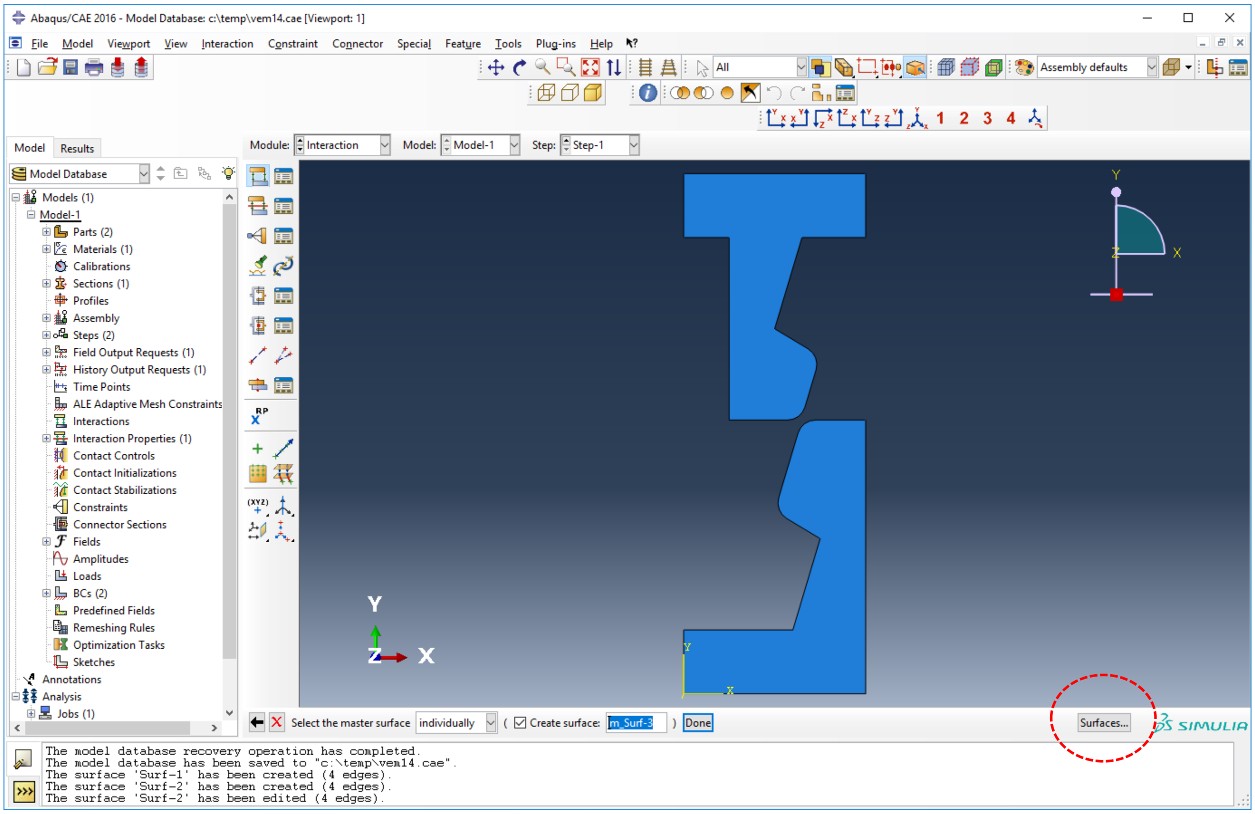
Continue. Válasszuk ki az alsó alkatrészen lévő kontaktba kerülő éleket. Majd alul **Done**.

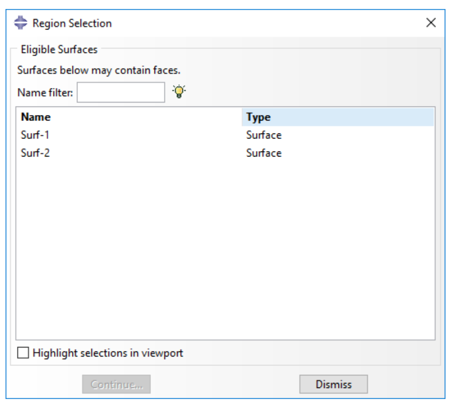


Ezt követően **Create Interaction** . Felugró ablakban a Step-nél Step-1, majd **Surface-to-Surface contact** opció és **Continue.**



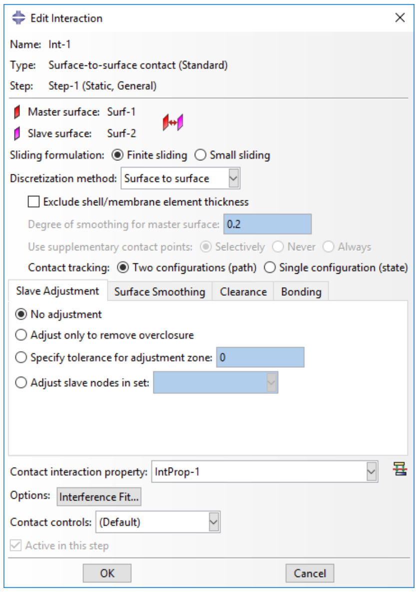
Kéri, hogy válasszuk kis az első **Surface**-t. Jobb alul felnyithatjuk a **Surfaces…** ablakot:





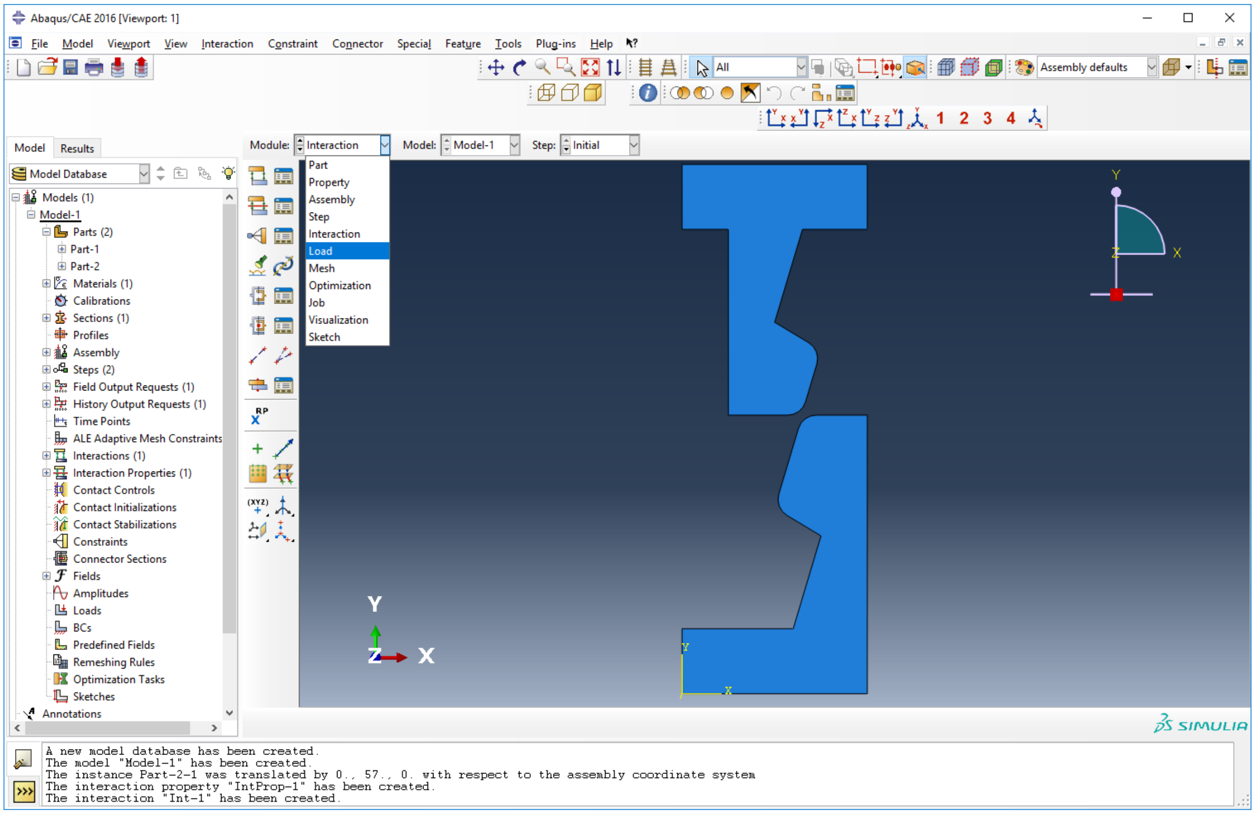
Válasszuk ki a **Surf-1**-t majd **Continue**. Kéri a második felületet. A lenti opcióból válasszuk a **Surface**-t:

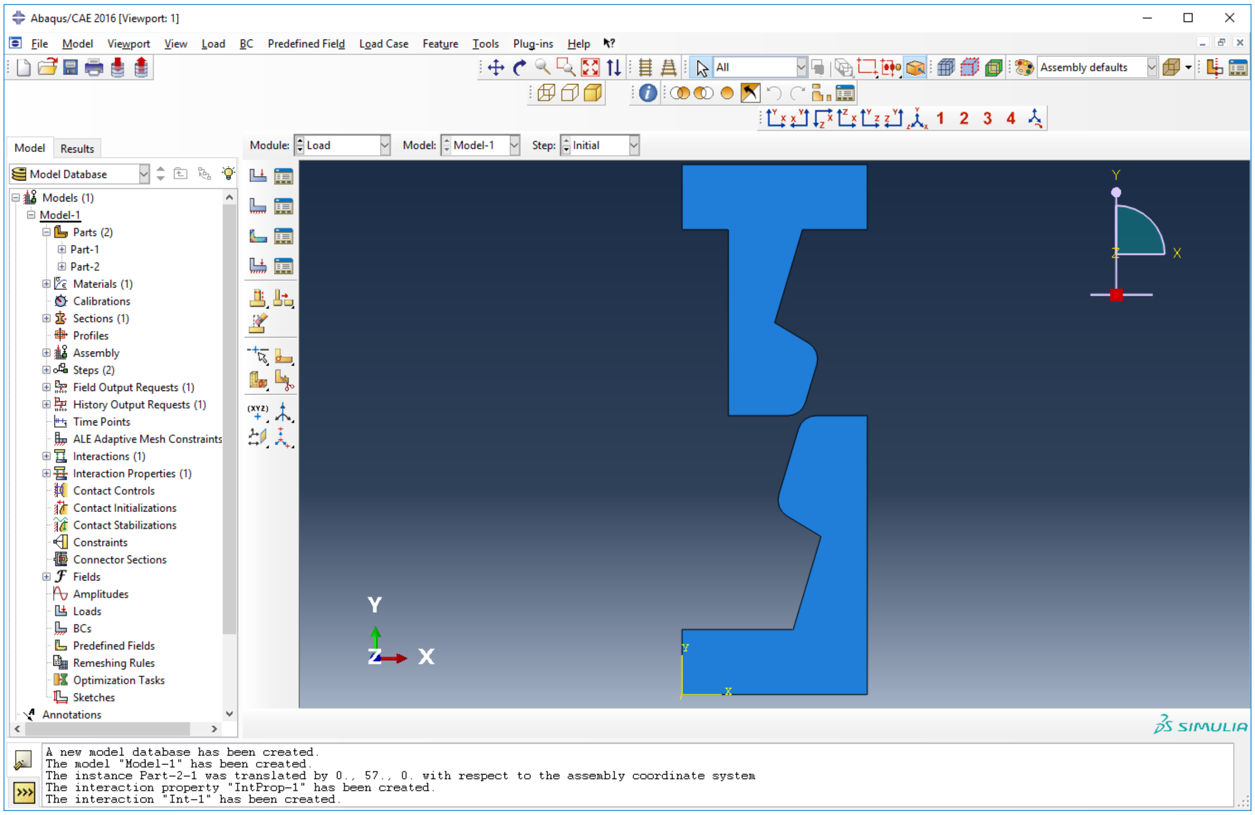
Válasszuk ki a második felületet majd **Continue**. Új ablakra **OK**.



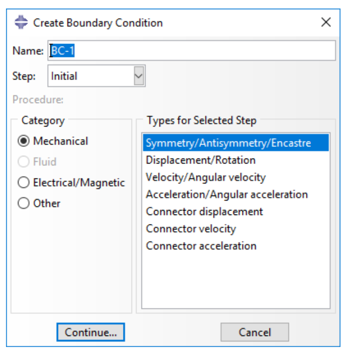
LOADS

Térjünk át a **Load** modulra:

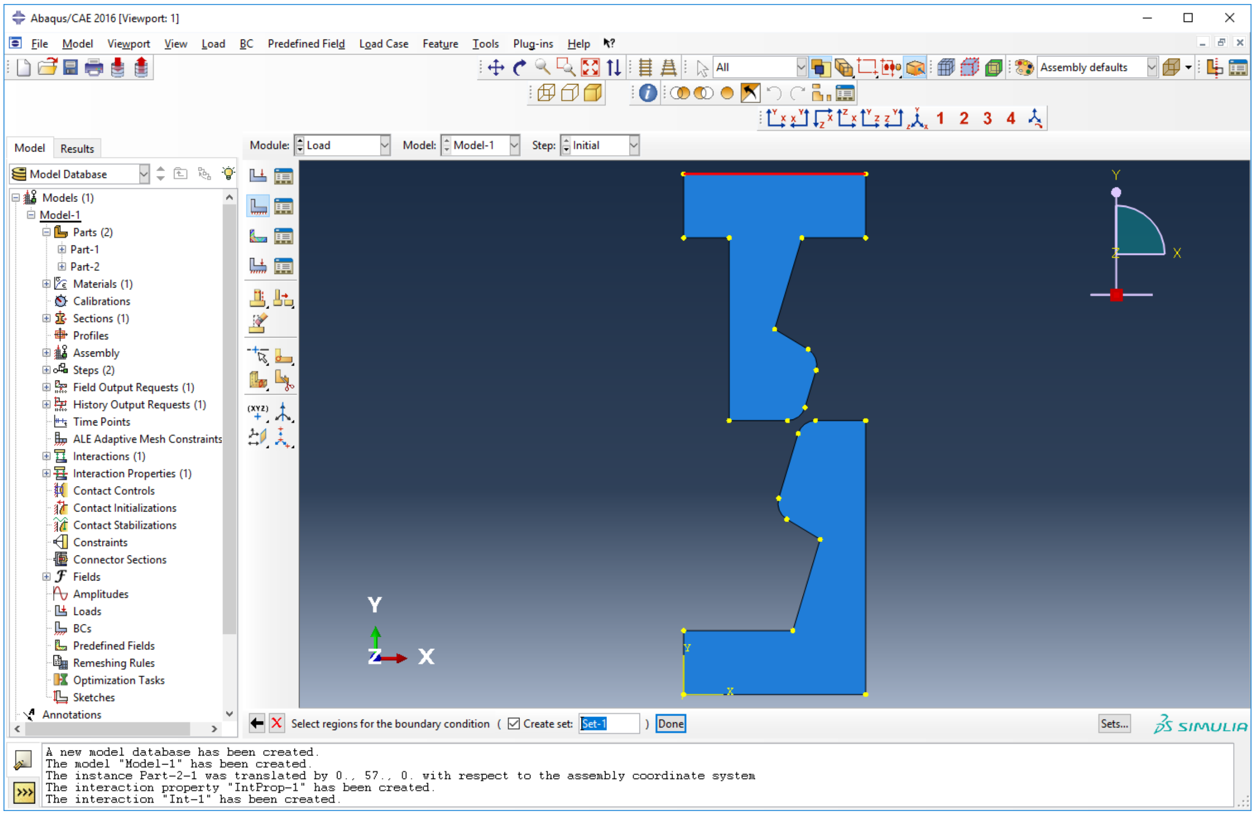




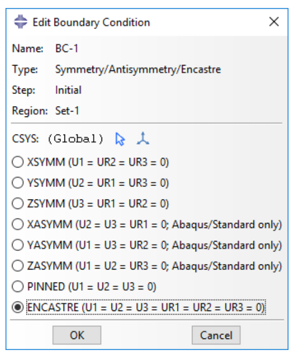
Katt a **Create Boundary Condition** ikonra: . Felugró ablak: A Step-nél az Initial legyen!



Continue. (*Ha a Surface generálásból adódó ablak még aktív akkor azt a Dismiss gombbal eltűntethetjük*). Válasszuk ki a felső élt majd **Done**.

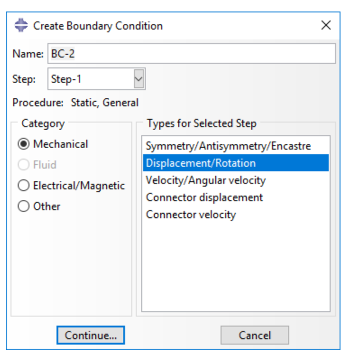


Felugró ablakban **ENCASTRE**, majd **OK**:

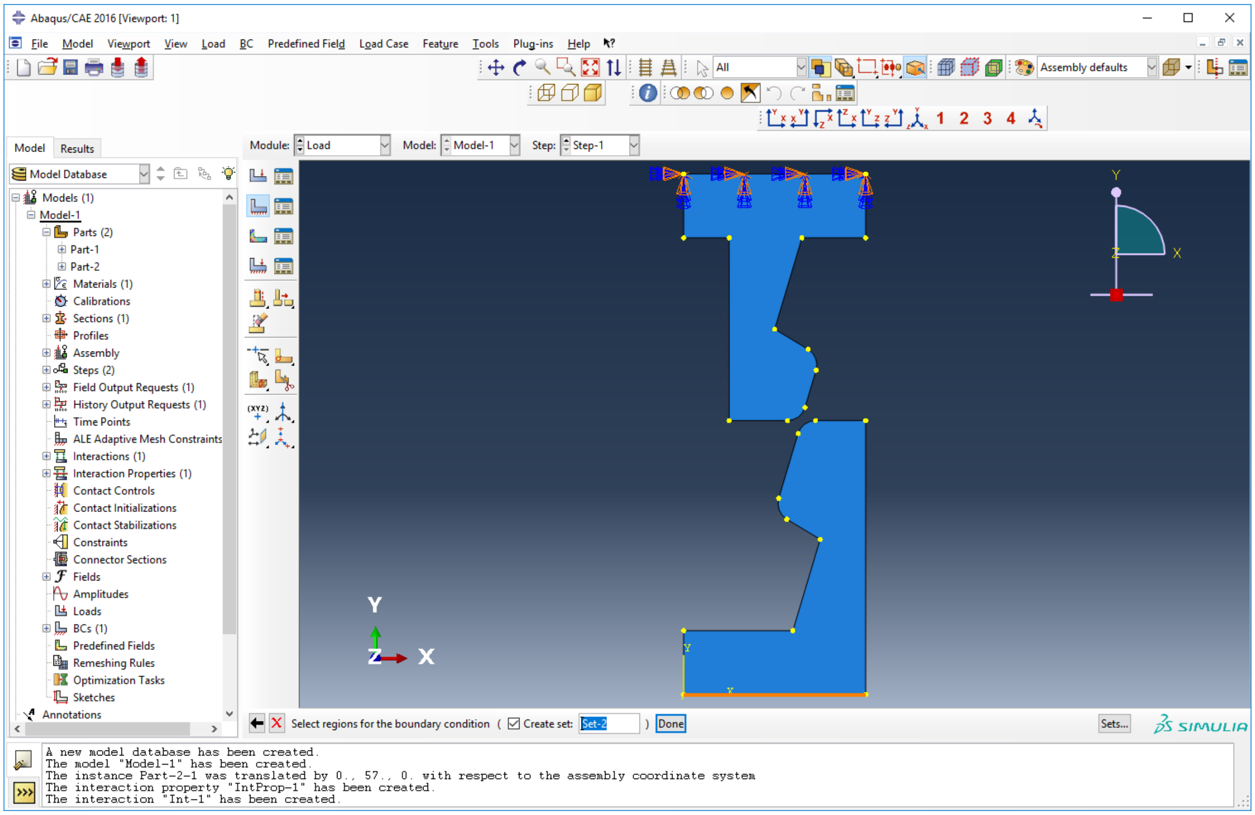


Katt a **Create Boundary Condition** ikonra: . Felugró ablak: A Step-nél az Step-1 legyen!

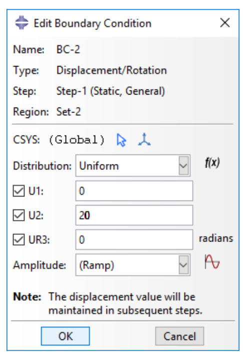
A listából válasszuk ki a **Displacement/Rotation**-t. **Continue**.



Válasszuk ki az alsó élt és **Done**.

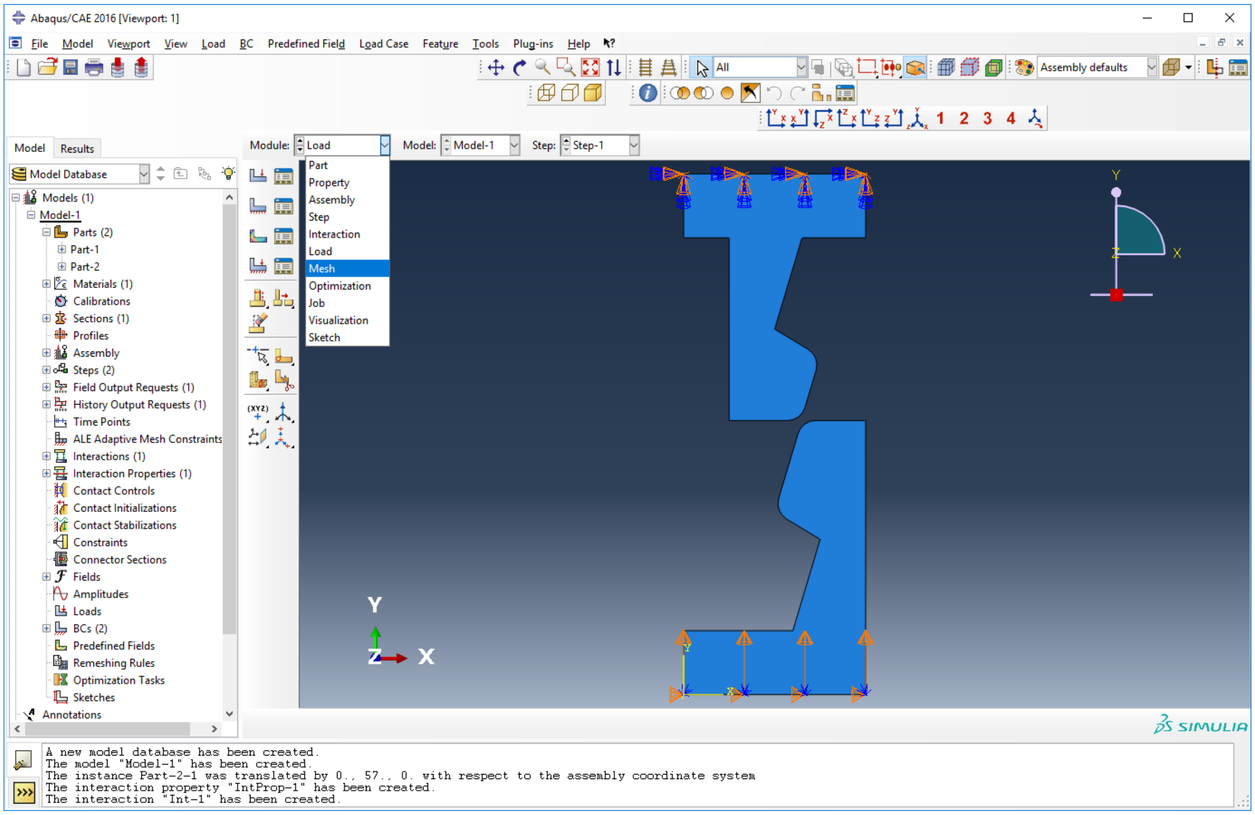


Felugró ablakban az **U2**-höz 20 máshova 0 és **OK**.

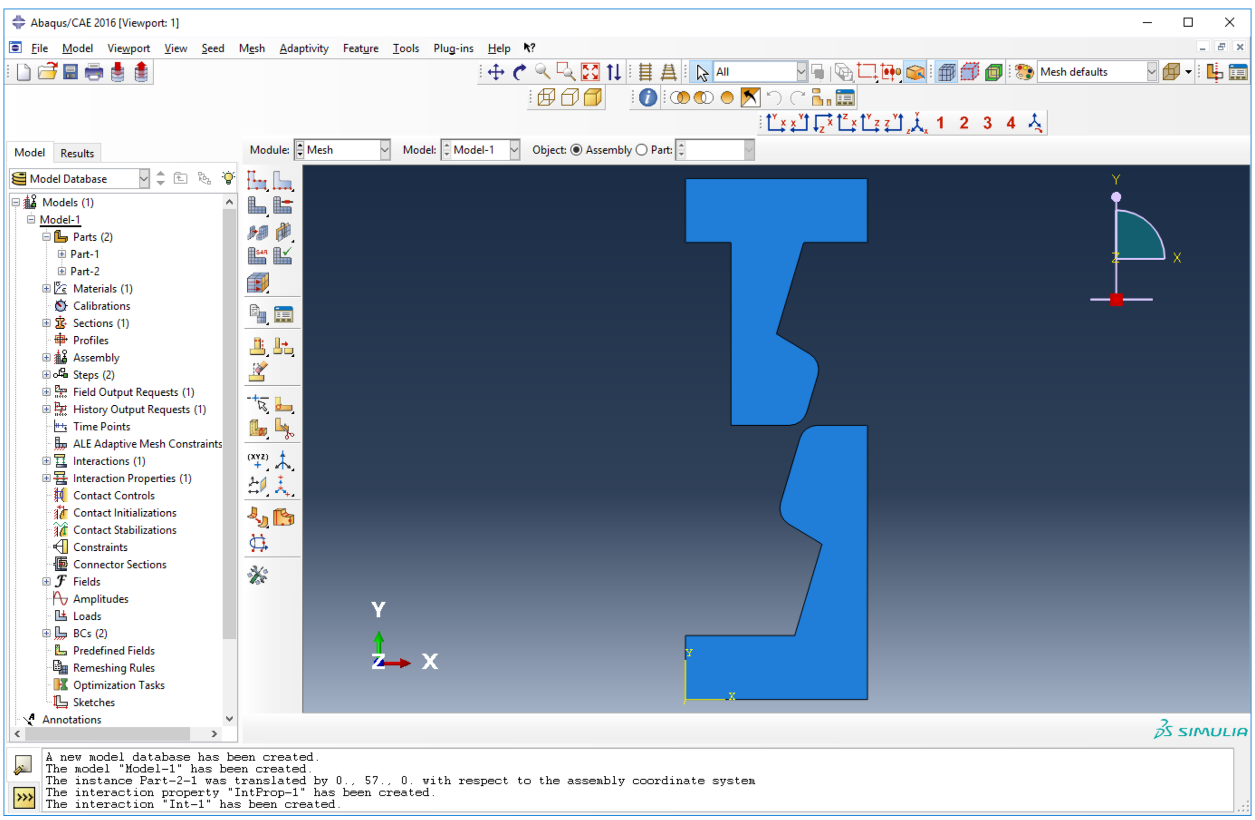


MESH

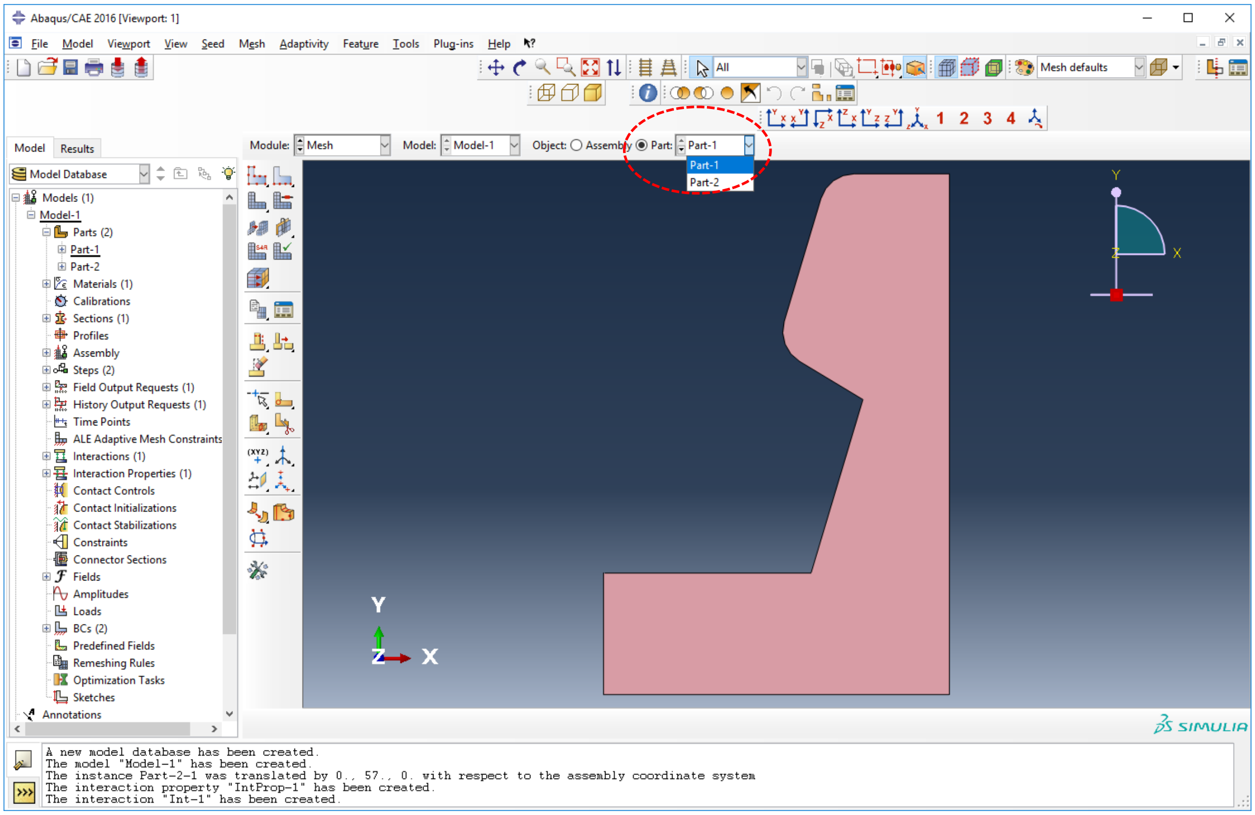
Mesh modul:



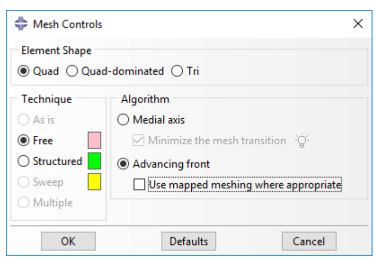
Változik az ikonsor:



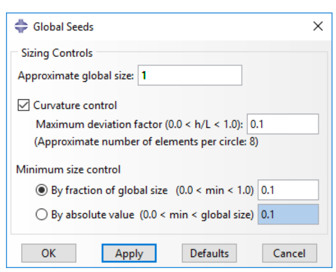
Válasszuk ki a Part-1-t: Katt a **Part** gombra és legördülő menüből válasszuk ki.



Assign Mesh Control-nál adjuk meg, hogy csak Quad elemeket használjon. Felugró ablakban Quad és alábbi opciók (szedjük most ki az *Use mapped meshing…* opciót):

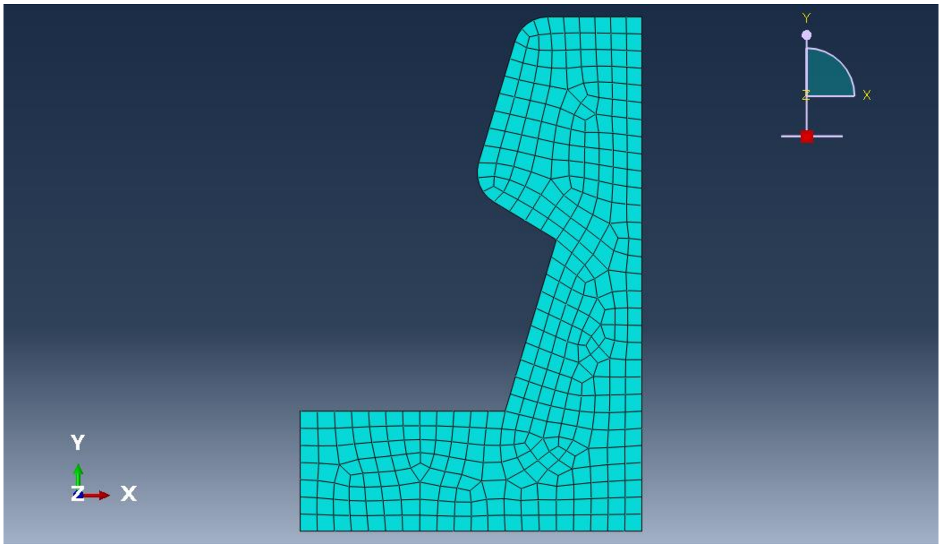


Ezt követően **Seed Part** . Felugróban legyen 1 az elemméret:



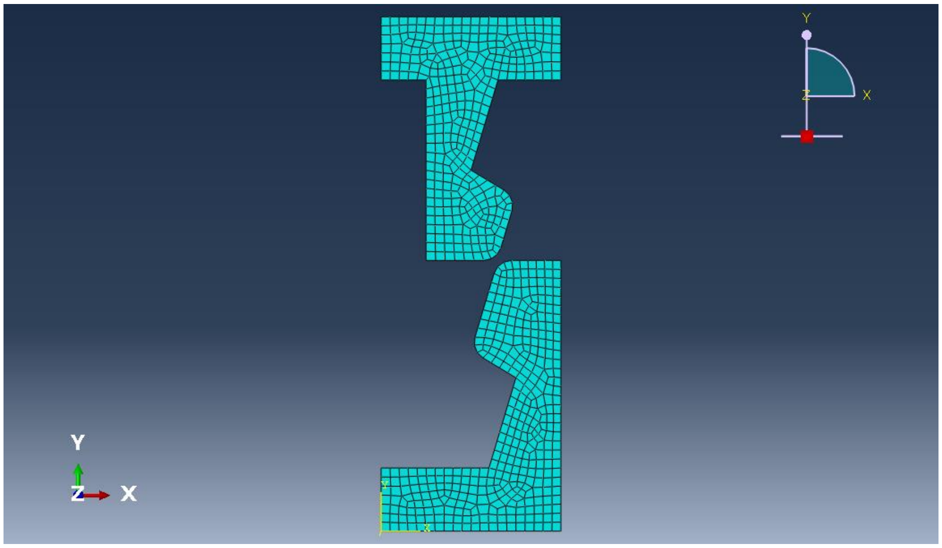
Majd **OK**. Aztán **Done** alul. Ezt követően **Mesh Part** és alul **Yes**:

Eredmény:



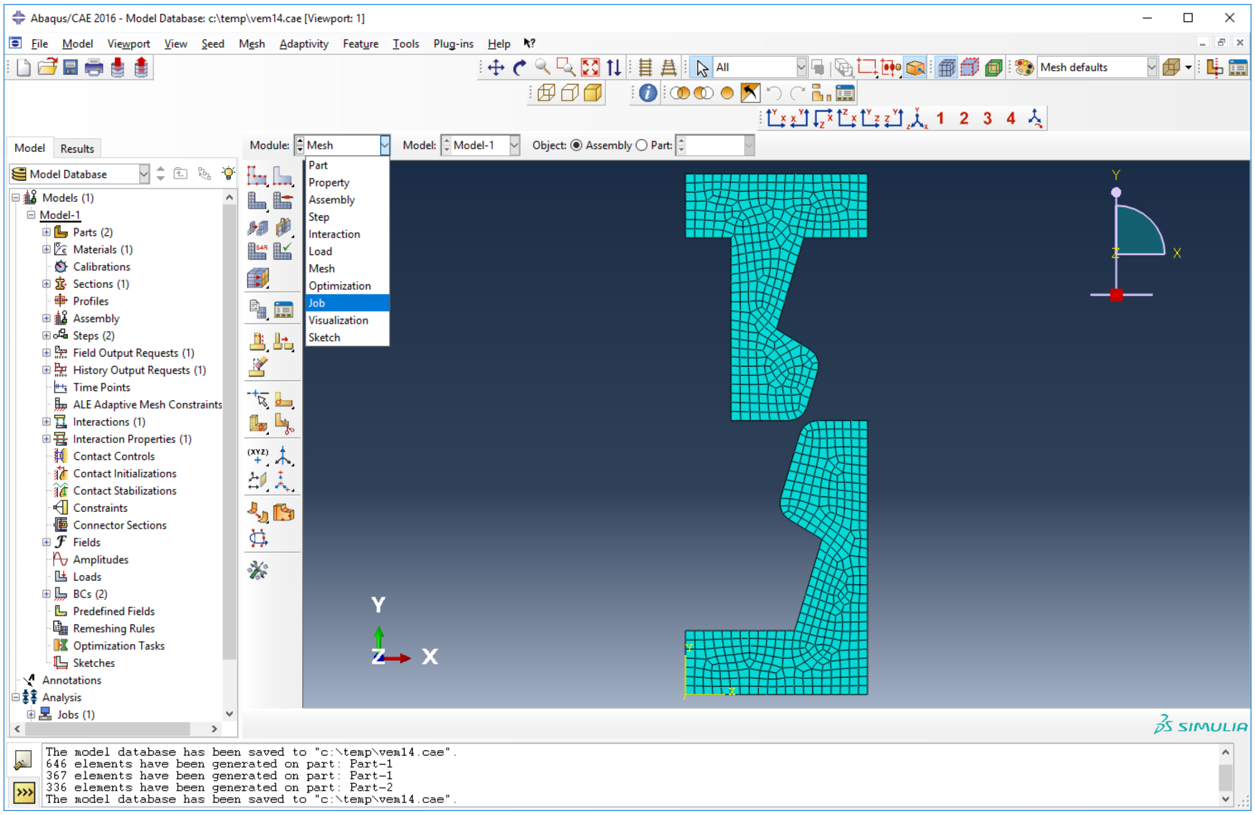
Másik Part-nál is végezzük el a fenti műveleteket.

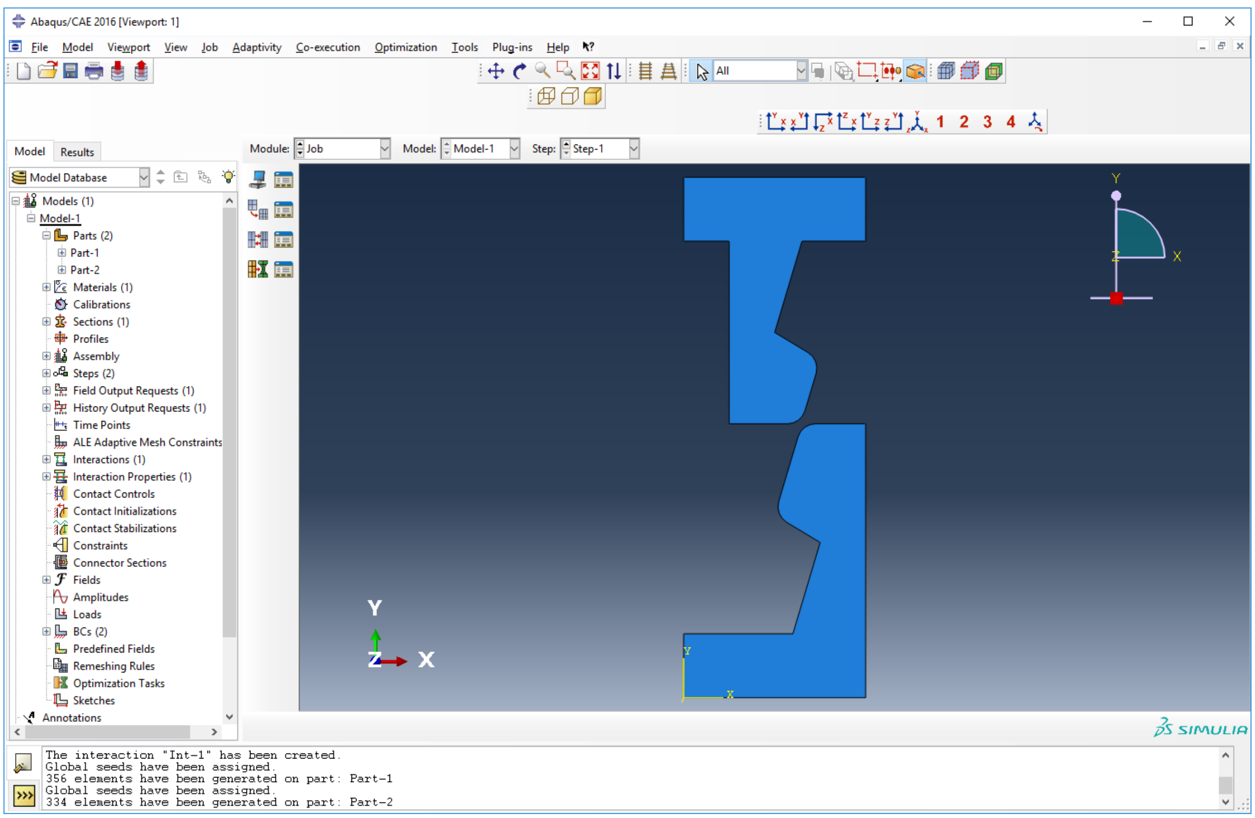
Ha átkattintunk az **Assembly** gombra akkor együtt látjuk:



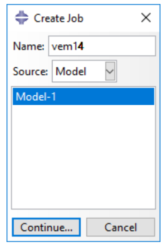
JOB

Job modul kiválasztása:

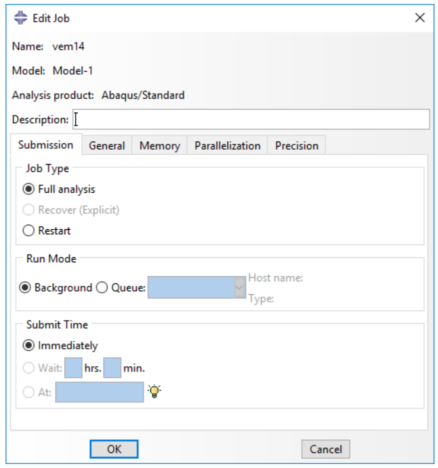




Create Job ikon: . Felugró ablakban adhatunk nevet neki: vem14. **Continue**.

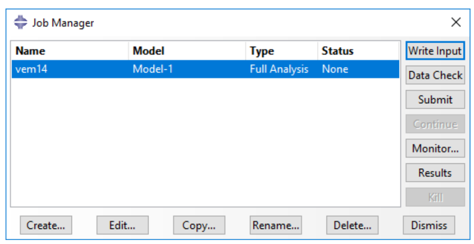


Felugró ablak:

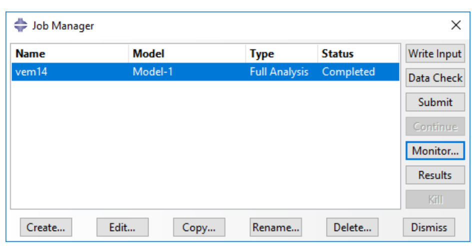


OK. Mentsük el a modellt. **File / Save as**.

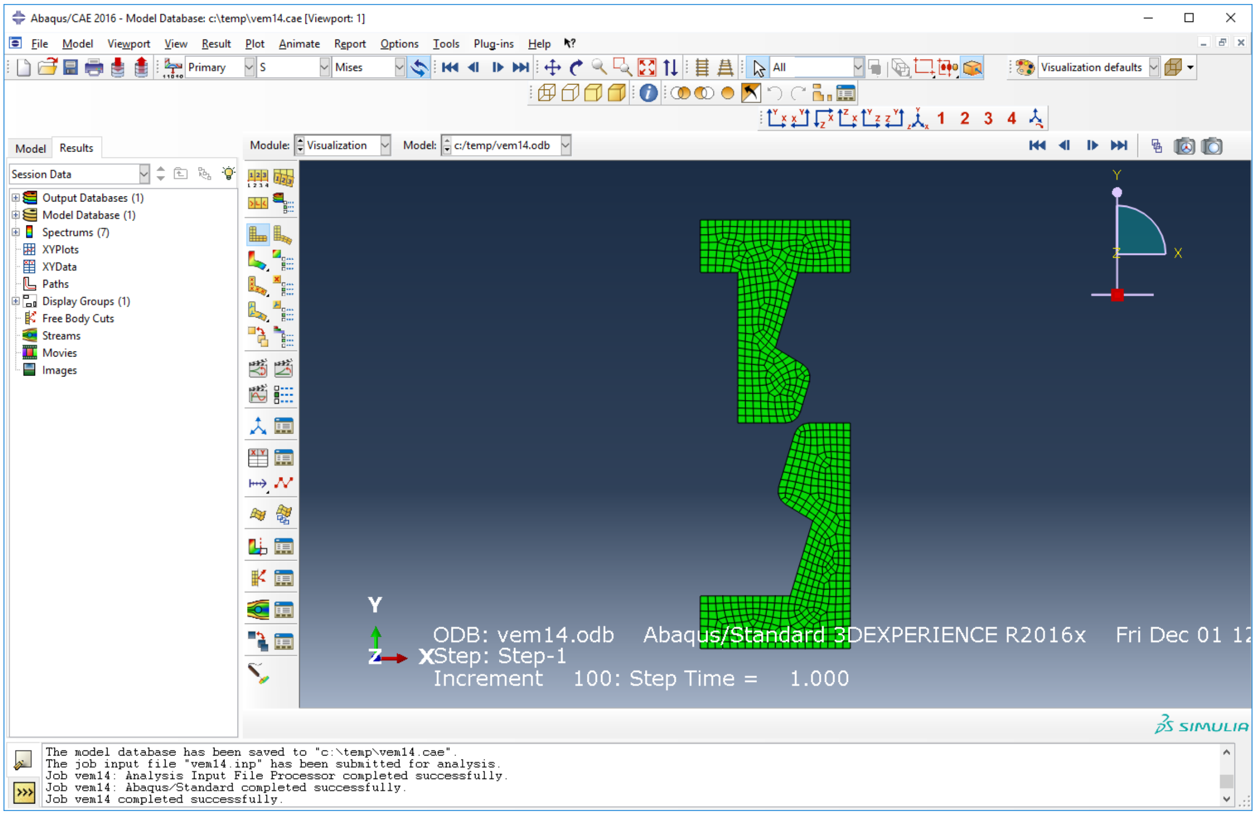
Job manager ikon: . Látjuk a kész Job-okat:



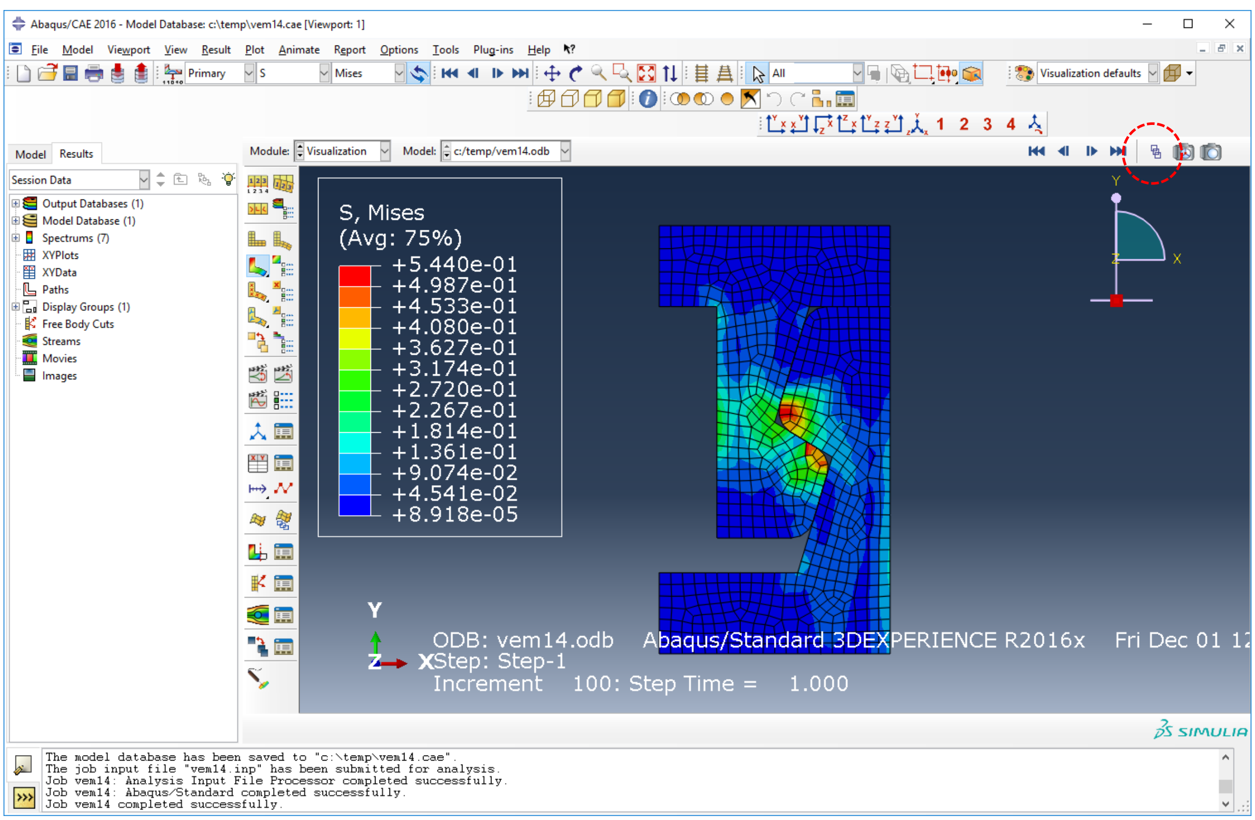
Katt a **Submit** ikonra jobb oldalon. Ha végzett akkor átvált **Completed**-re.



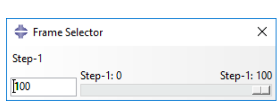
Katt a **Results** gombra. Alapból a kezdeti alakot mutatja:

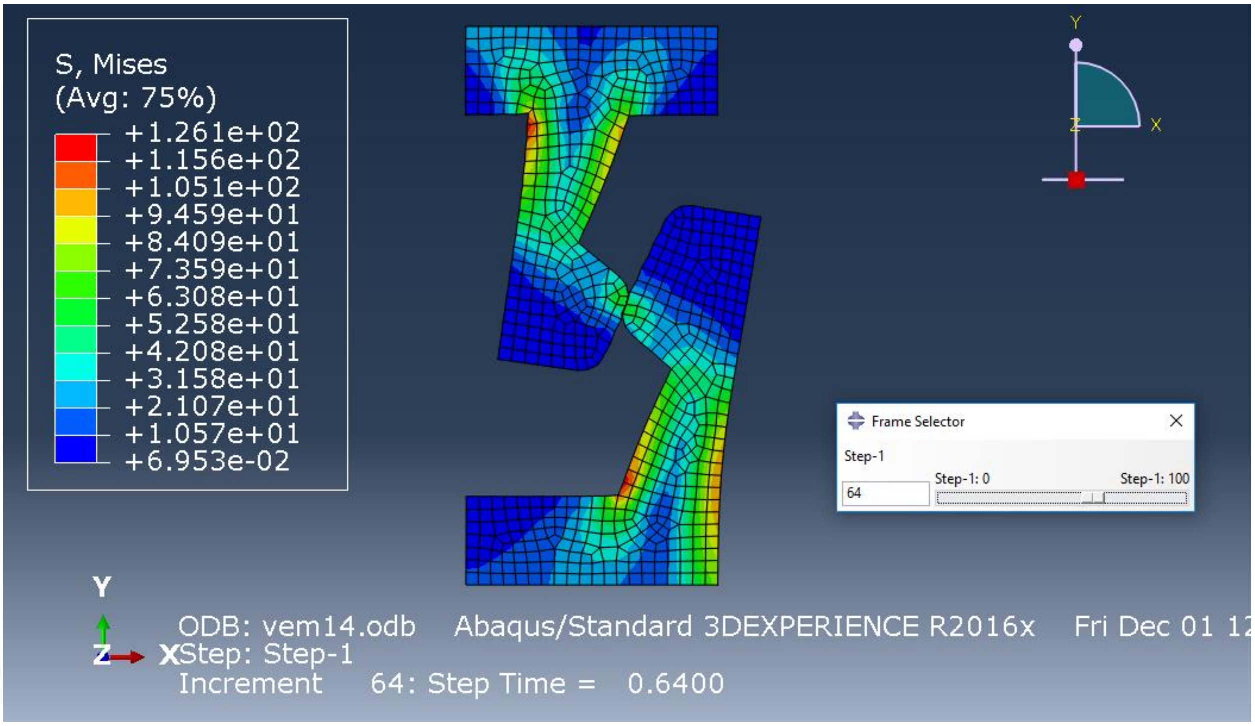


Katt a **Plot Contours on Deformed Shape** ikonra: . Standard beállítás, hogy ekkor a Mises-féle egyenértékű feszültséget mutatja a Step végén:



Ha a megoldás során lévő értékek is érdekelnek, akkor használhatjuk a **Frame Selector** csúszkát jobb oldalon felül: . Felugró ablak:

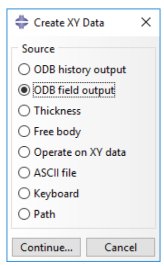




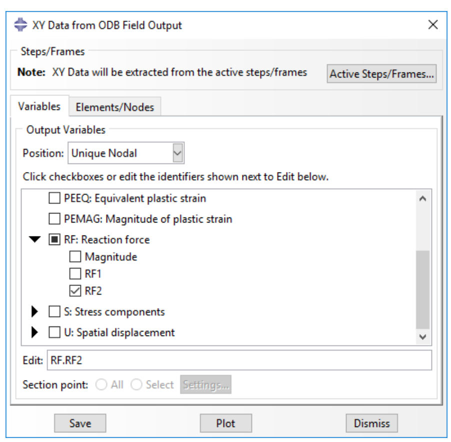
Animáció: Ki/Be kapcsolható.

Kérdezzük le a reakcióerőket. Reakció erők ébrednek a node-okban. Deformálatlan alak: . Auto Fit .

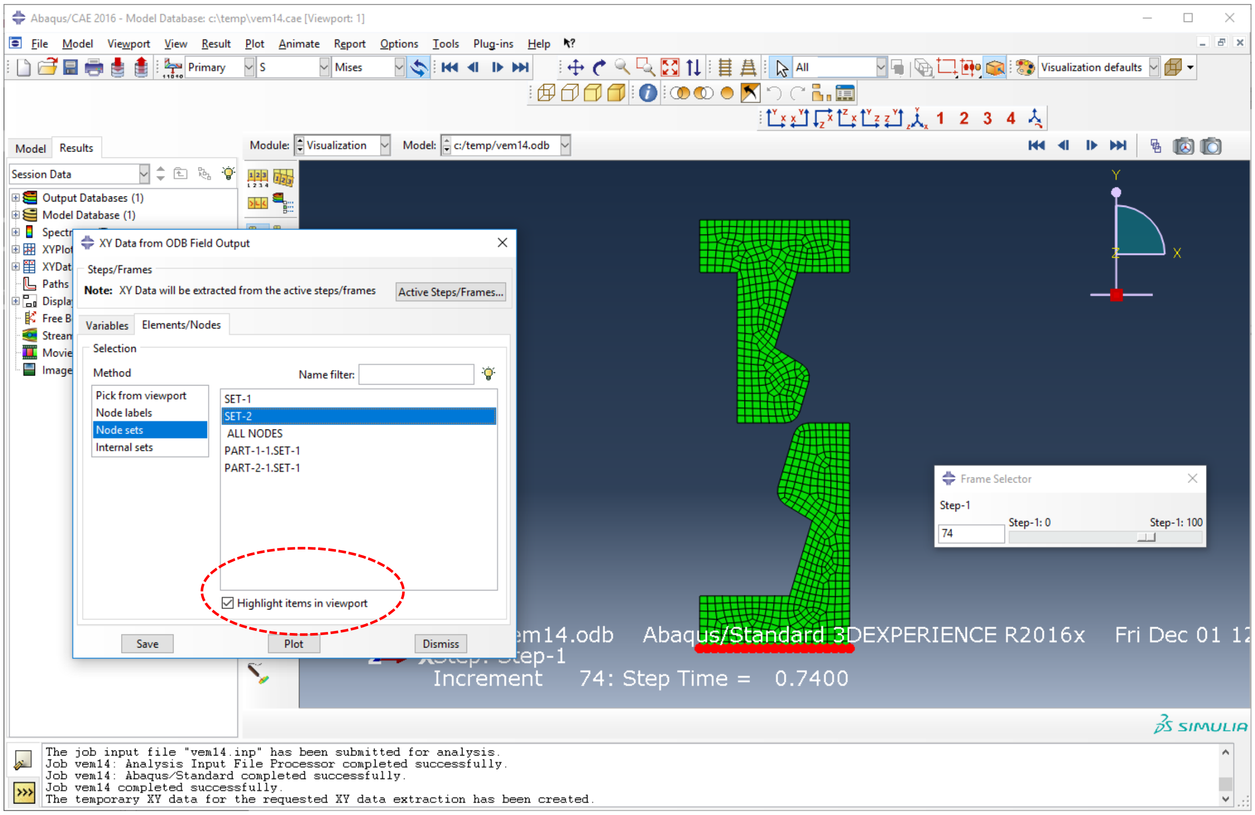
Create XY Data . Felugróban ablakban **ODB field output**. Majd **Continue**.



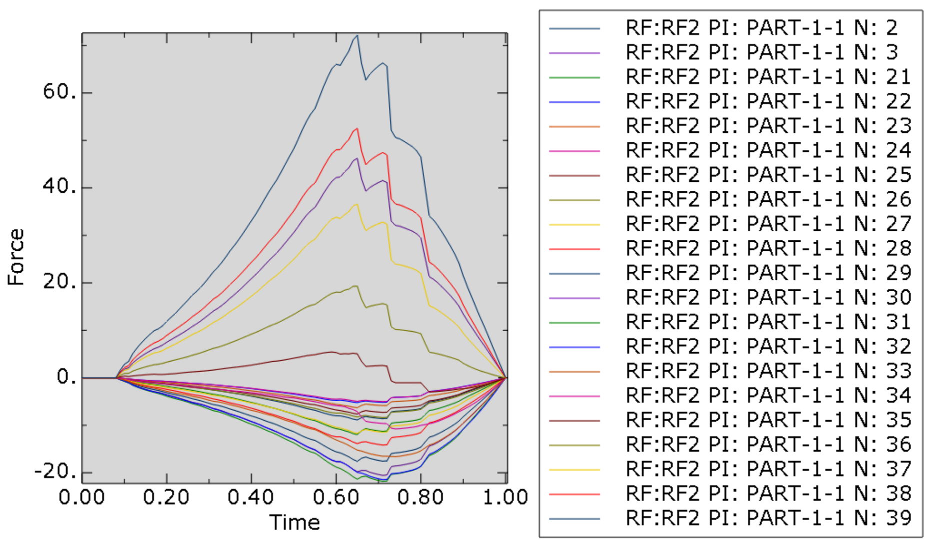
Felugró ablakban a **Position**-nál **Unique Nodal**, majd **RF2** a lenti választható értékekből



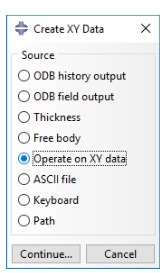
Váltsunk át az **Element/Nodes** fülre. Ott válasszuk ki a **Node sets**-t majd a **Highligth items in viewport** opciót kapcsoljuk be. Válasszuk ki az a **Set**-t ami az alsó élhez tartozik (jelen segédletben SET-2). Mutatja az ábrán pirossal.



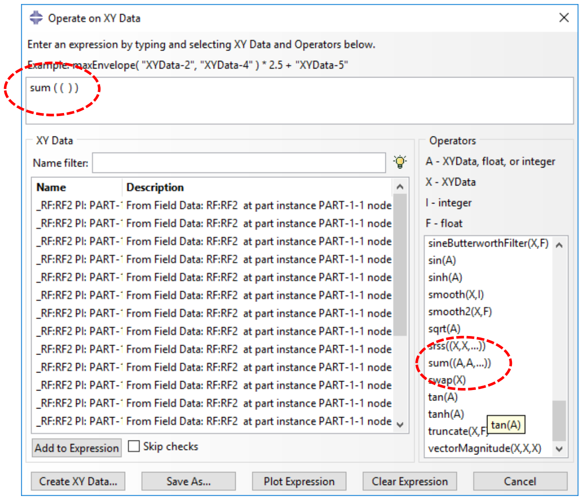
Aztán **Plot**. **Dissmiss**-sel eltűntethetjük az ablakot. Ez most a reakcióerőket (N/mm dimenzióban tekintettel a síkfeladatra) mutatja minden **node**-hoz.



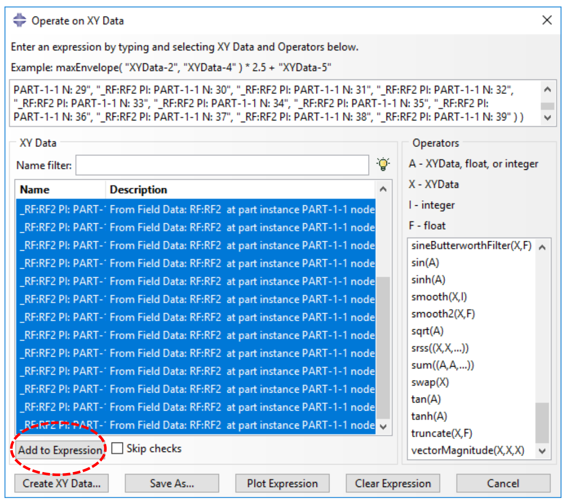
Ha minket az összegük érdekel (vagyis az eredő erő a felső élen) akkor össze lehet adni. **Create XY Data**, majd **Operate on XY data**. **Continue**.



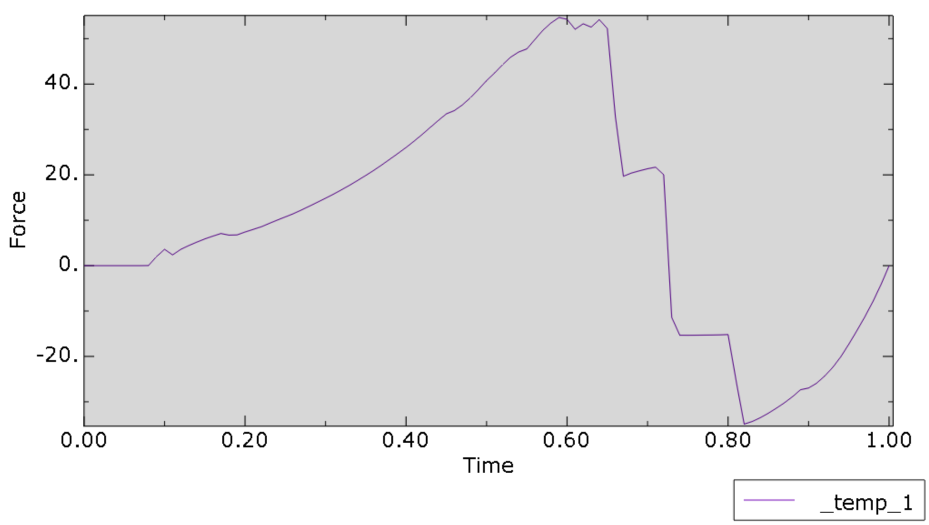
Jobb oldali függvénylistából válasszuk ki a **sum((A, A, …))**-t, ezzel bekerül az egyenlet mezőbe.



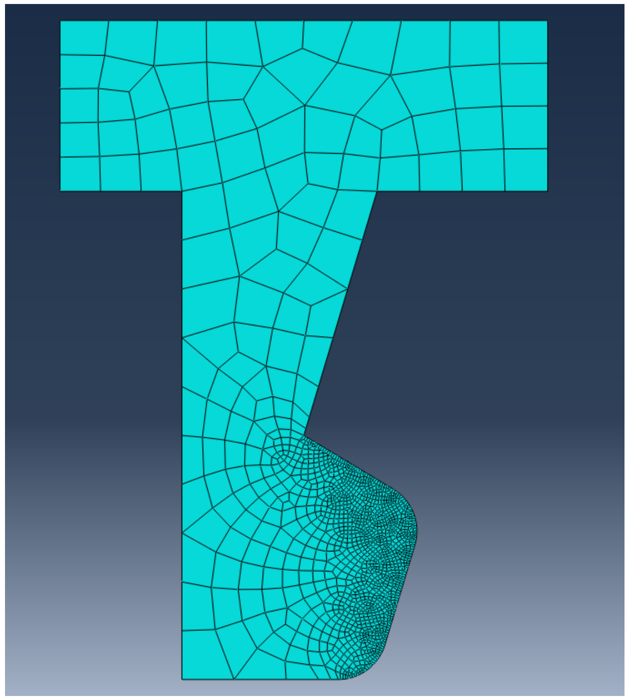
Válasszuk ki az első elemet majd shift lenyomása után az utolsó elemet. Ezzel az összes ki van jelölve. **Add to Expression**.



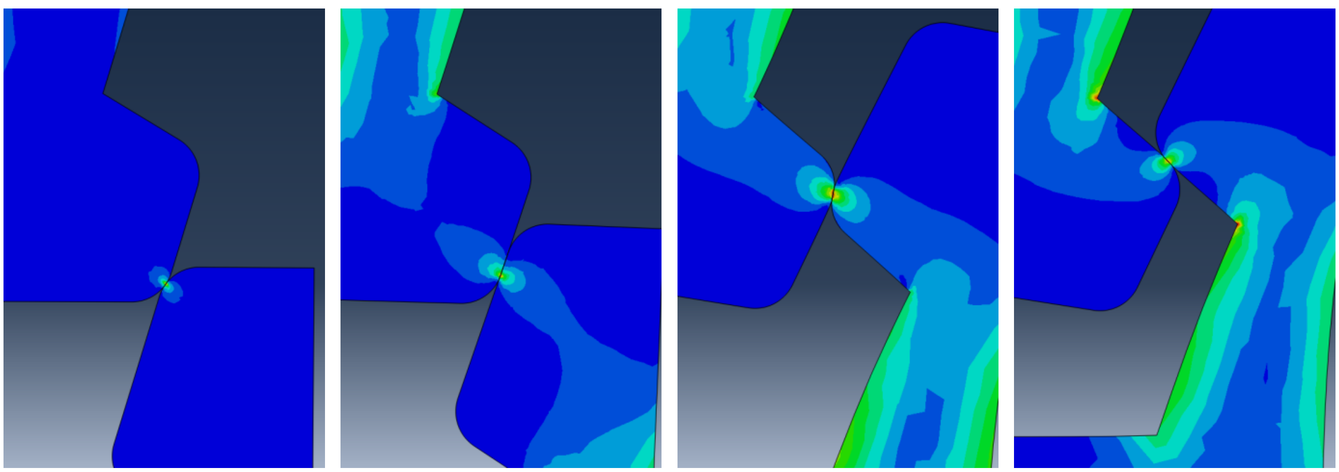
Plot Expression. A Time változó (ami jelen esetben 0…1 volt) mentén a reakcióerő (Y irányú) változása:



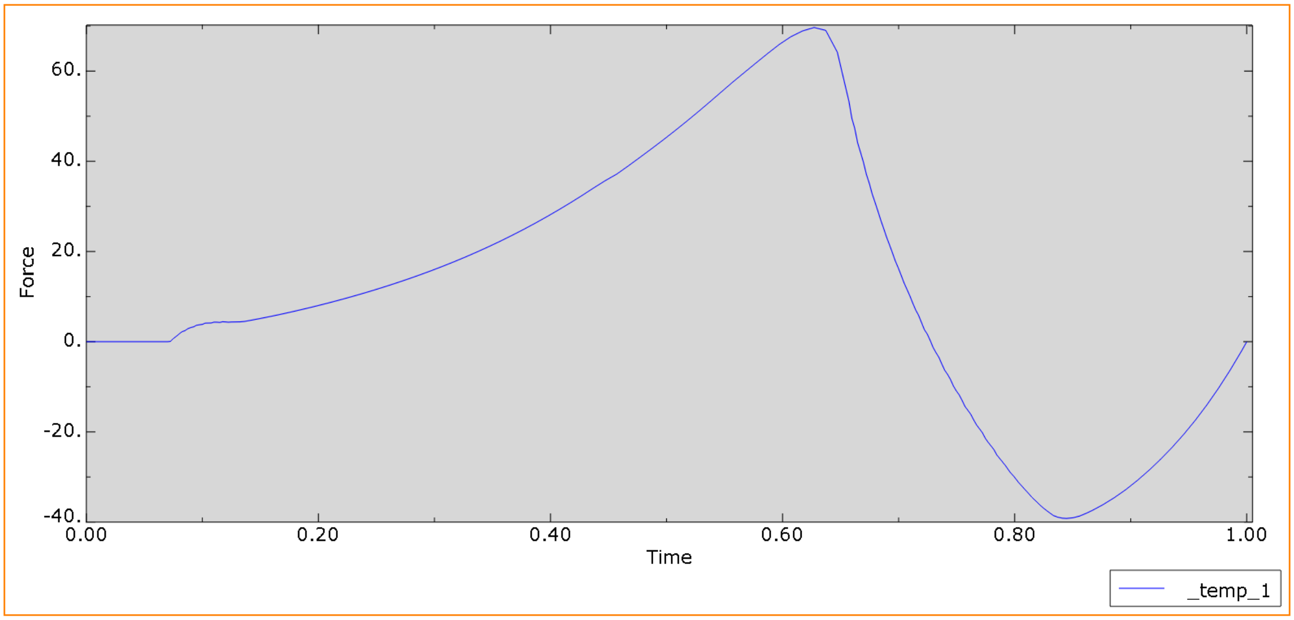
A megoldás elég „darabos" a durva háló miatt. Az eredmények sűrűbb hálóval:



A feszültség eloszláson a terhelés során most szebben látszik a pontszerű érintkezés az elemek között:



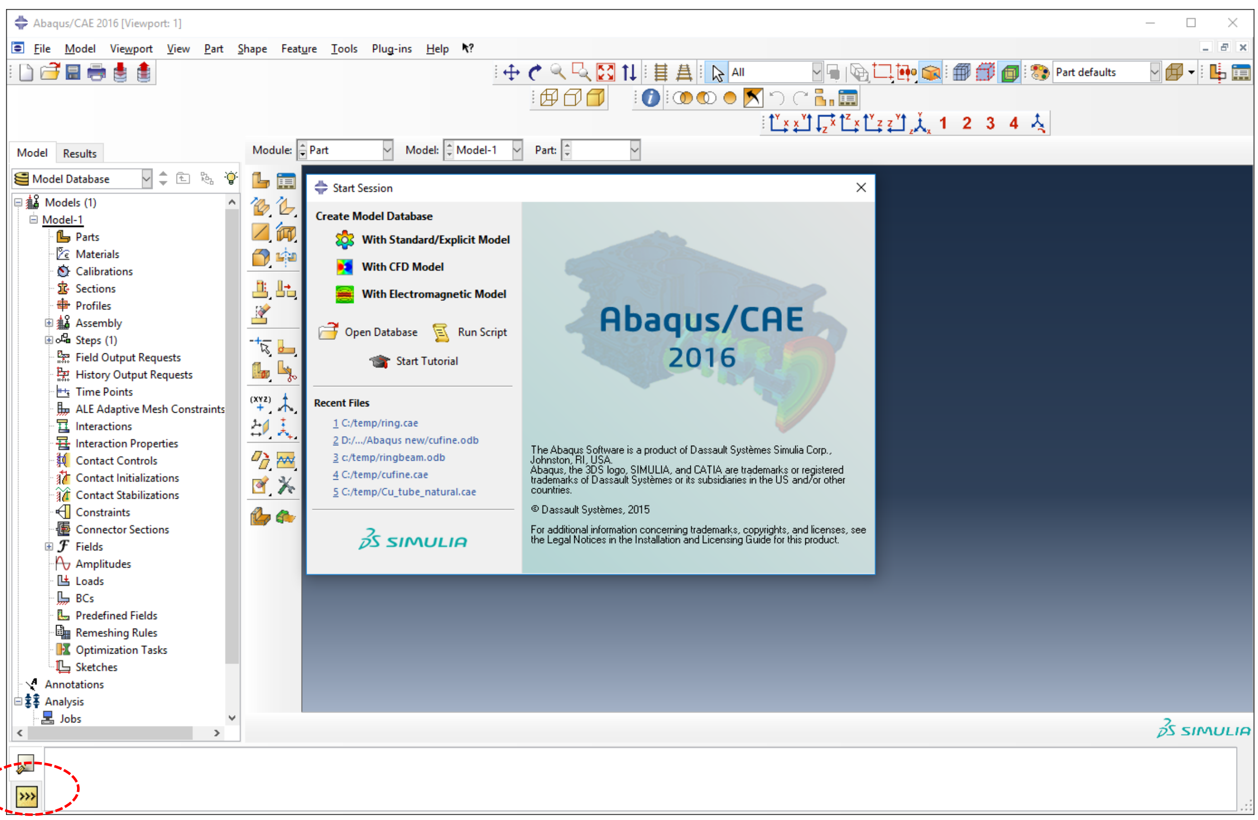
Az eredő Y reakcióerő változása a terhelés során:



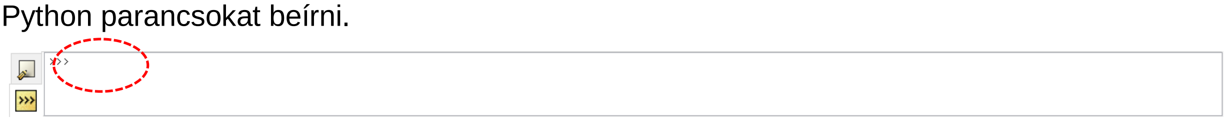
SZKRIPT

A modell felépítése Python szkript segítségével.

Indítsuk el a szoftvert.



Zárjuk be a kezdő kisablakot. Majd alul kattintsunk a ikonra. Ezt követően lehetőségünk van a prompt helyére Python parancsokat beírni.



Jelöljük ki a lenti kódrészt és illesszük be CTRL+C és CTRL+V kombinációval. Láthatjuk, ahogy végigfut és elkészít mindent. Most már futtathatjuk is. Váltsunk át a **Job** modulra és futtassuk le az elkészített **Job**-ot.

```
# -*- coding: mbcs -*-
#
# Abaqus/CAE Release 2016 replay file
# Internal Version: 2015_09_24-22.31.09 126547
# Run by Attila Kossa on Fri Dec 01 14:30:53 2017
#

# from driverUtils import executeOnCaeGraphicsStartup
# executeOnCaeGraphicsStartup()
#: Executing "onCaeGraphicsStartup()" in the site directory ...
from abaqus import *
from abaqusConstants import *
session.Viewport(name='Viewport: 1', origin=(0.0, 0.0), width=184.004180908203,
    height=117.765747070313)
session.viewports['Viewport: 1'].makeCurrent()
session.viewports['Viewport: 1'].maximize()
from caeModules import *
from driverUtils import executeOnCaeStartup
executeOnCaeStartup()
session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(
    referenceRepresentation=ON)
Mdb()
#: A new model database has been created.
#: The model "Model-1" has been created.
session.viewports['Viewport: 1'].setValues(displayedObject=None)
cliCommand("""session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)""")
s = mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
    sheetSize=200.0)
g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.setPrimaryObject(option=STANDALONE)
s.Line(point1=(0.0, 0.0), point2=(20.0, 0.0))
s.HorizontalConstraint(entity=g.findAt((10.0, 0.0)), addUndoState=False)
s.Line(point1=(20.0, 0.0), point2=(20.0, 30.0))
s.VerticalConstraint(entity=g.findAt((20.0, 15.0)), addUndoState=False)
s.PerpendicularConstraint(entity1=g.findAt((10.0, 0.0)), entity2=g.findAt((
    20.0, 15.0)), addUndoState=False)
s.Line(point1=(20.0, 30.0), point2=(13.0, 30.0))
s.HorizontalConstraint(entity=g.findAt((16.5, 30.0)), addUndoState=False)
s.PerpendicularConstraint(entity1=g.findAt((20.0, 15.0)), entity2=g.findAt((
    16.5, 30.0)), addUndoState=False)
s.Line(point1=(13.0, 30.0), point2=(10.0, 20.0))
s.Line(point1=(10.0, 20.0), point2=(15.0, 17.0))
s.Line(point1=(15.0, 17.0), point2=(12.0, 7.0))
s.Line(point1=(12.0, 7.0), point2=(0.0, 7.0))
s.HorizontalConstraint(entity=g.findAt((6.0, 7.0)), addUndoState=False)
s.Line(point1=(0.0, 7.0), point2=(0.0, 0.0))
s.VerticalConstraint(entity=g.findAt((0.0, 3.5)), addUndoState=False)
s.PerpendicularConstraint(entity1=g.findAt((6.0, 7.0)), entity2=g.findAt((0.0,
    3.5)), addUndoState=False)
s.FilletByRadius(radius=2.0, curve1=g.findAt((16.5, 30.0)), nearPoint1=(
    15.6112060546875, 29.3963203430176), curve2=g.findAt((11.5, 25.0)),
    nearPoint2=(12.6639556884766, 26.9160079956055))
s.FilletByRadius(radius=2.0, curve1=g.findAt((11.286204, 24.287348)),
    nearPoint1=(11.4666366577148, 23.4251937866211), curve2=g.findAt((12.5,
    18.5)), nearPoint2=(12.1113433837891, 19.107608795166))
p = mdb.models['Model-1'].Part(name='Part-1', dimensionality=TWO_D_PLANAR,
    type=DEFORMABLE_BODY)
p = mdb.models['Model-1'].parts['Part-1']
p.BaseShell(sketch=s)
s.unsetPrimaryObject()
p = mdb.models['Model-1'].parts['Part-1']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
del mdb.models['Model-1'].sketches['__profile__']
s1 = mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
    sheetSize=200.0)
g, v, d, c = s1.geometry, s1.vertices, s1.dimensions, s1.constraints
s1.setPrimaryObject(option=STANDALONE)
s1.Line(point1=(0.0, 0.0), point2=(20.0, 0.0))
s1.HorizontalConstraint(entity=g.findAt((10.0, 0.0)), addUndoState=False)
s1.Line(point1=(20.0, 0.0), point2=(20.0, -7.0))
s1.VerticalConstraint(entity=g.findAt((20.0, -3.5)), addUndoState=False)
s1.PerpendicularConstraint(entity1=g.findAt((10.0, 0.0)), entity2=g.findAt((
    20.0, -3.5)), addUndoState=False)
```

```
s1.Line(point1=(20.0, -7.0), point2=(13.0, -7.0))
s1.HorizontalConstraint(entity=g.findAt((16.5, -7.0)), addUndoState=False)
s1.PerpendicularConstraint(entity1=g.findAt((20.0, -3.5)), entity2=g.findAt((
    16.5, -7.0)), addUndoState=False)
s1.Line(point1=(13.0, -7.0), point2=(10.0, -17.0))
s1.Line(point1=(10.0, -17.0), point2=(15.0, -20.0))
s1.Line(point1=(15.0, -20.0), point2=(12.9, -27.0))
s1.Line(point1=(12.9, -27.0), point2=(5.0, -27.0))
s1.HorizontalConstraint(entity=g.findAt((8.95, -27.0)), addUndoState=False)
s1.Line(point1=(5.0, -27.0), point2=(5.0, -7.0))
s1.VerticalConstraint(entity=g.findAt((5.0, -17.0)), addUndoState=False)
s1.PerpendicularConstraint(entity1=g.findAt((8.95, -27.0)), entity2=g.findAt((
    5.0, -17.0)), addUndoState=False)
s1.Line(point1=(5.0, -7.0), point2=(0.0, -7.0))
s1.HorizontalConstraint(entity=g.findAt((2.5, -7.0)), addUndoState=False)
s1.PerpendicularConstraint(entity1=g.findAt((5.0, -17.0)), entity2=g.findAt((
    2.5, -7.0)), addUndoState=False)
s1.Line(point1=(0.0, -7.0), point2=(0.0, 0.0))
s1.VerticalConstraint(entity=g.findAt((0.0, -3.5)), addUndoState=False)
s1.PerpendicularConstraint(entity1=g.findAt((2.5, -7.0)), entity2=g.findAt((
    0.0, -3.5)), addUndoState=False)
s1.FilletByRadius(radius=2.0, curve1=g.findAt((12.5, -18.5)), nearPoint1=(
    14.3217849731445, -19.1994743347168), curve2=g.findAt((13.95, -23.5)),
    nearPoint2=(14.690185546875, -22.2309684753418))
s1.FilletByRadius(radius=2.0, curve1=g.findAt((8.95, -27.0)), nearPoint1=(
    10.7298202514648, -27.0997352600098), curve2=g.findAt((13.726565,
    -24.244782)), nearPoint2=(13.4928665161133, -25.1706008911133))
p = mdb.models['Model-1'].Part(name='Part-2', dimensionality=TWO_D_PLANAR,
    type=DEFORMABLE_BODY)
p = mdb.models['Model-1'].parts['Part-2']
p.BaseShell(sketch=s1)
s1.unsetPrimaryObject()
p = mdb.models['Model-1'].parts['Part-2']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
del mdb.models['Model-1'].sketches['__profile__']
session.viewports['Viewport: 1'].partDisplay.setValues(sectionAssignments=ON,
    engineeringFeatures=ON)
session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(
    referenceRepresentation=OFF)
mdb.models['Model-1'].Material(name='Material-1')
mdb.models['Model-1'].materials['Material-1'].Elastic(table=((2800.0, 0.35), ))
mdb.models['Model-1'].HomogeneousSolidSection(name='Section-1',
    material='Material-1', thickness=None)
p = mdb.models['Model-1'].parts['Part-1']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
p = mdb.models['Model-1'].parts['Part-1']
f = p.faces
faces = f.findAt(((4.0, 4.666667, 0.0), ))
region = p.Set(faces=faces, name='Set-1')
p = mdb.models['Model-1'].parts['Part-1']
p.SectionAssignment(region=region, sectionName='Section-1', offset=0.0,
    offsetType=MIDDLE_SURFACE, offsetField='',
    thicknessAssignment=FROM_SECTION)
p = mdb.models['Model-1'].parts['Part-2']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
p = mdb.models['Model-1'].parts['Part-2']
f = p.faces
faces = f.findAt(((1.666667, -4.666667, 0.0), ))
region = p.Set(faces=faces, name='Set-1')
p = mdb.models['Model-1'].parts['Part-2']
p.SectionAssignment(region=region, sectionName='Section-1', offset=0.0,
    offsetType=MIDDLE_SURFACE, offsetField='',
    thicknessAssignment=FROM_SECTION)
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
    optimizationTasks=OFF, geometricRestrictions=OFF, stopConditions=OFF)
a = mdb.models['Model-1'].rootAssembly
a.DatumCsysByDefault(CARTESIAN)
p = mdb.models['Model-1'].parts['Part-1']
a.Instance(name='Part-1-1', part=p, dependent=ON)
p = mdb.models['Model-1'].parts['Part-2']
a.Instance(name='Part-2-1', part=p, dependent=ON)
a = mdb.models['Model-1'].rootAssembly
a.translate(instanceList=('Part-2-1', ), vector=(0.0, 57.0, 0.0))
#: The instance Part-2-1 was translated by 0., 57., 0. with respect to the assembly coordinate system
session.viewports['Viewport: 1'].view.fitView()
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
    adaptiveMeshConstraints=ON)
```

```
mdb.models['Model-1'].StaticStep(name='Step-1', previous='Initial',
    maxNumInc=1000, timeIncrementationMethod=FIXED, initialInc=0.01,
    noStop=OFF, nlgeom=ON)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-1')
session.viewports['Viewport: 1'].assemblyDisplay.setValues(interactions=ON,
    constraints=ON, connectors=ON, engineeringFeatures=ON,
    adaptiveMeshConstraints=OFF)
mdb.models['Model-1'].ContactProperty('IntProp-1')
mdb.models['Model-1'].interactionProperties['IntProp-1'].TangentialBehavior(
    formulation=PENALTY, directionality=ISOTROPIC, slipRateDependency=OFF,
    pressureDependency=OFF, temperatureDependency=OFF, dependencies=0, table=((
    0.2, ), ), shearStressLimit=None, maximumElasticSlip=FRACTION,
    fraction=0.005, elasticSlipStiffness=None)
#: The interaction property "IntProp-1" has been created.
a = mdb.models['Model-1'].rootAssembly
s1 = a.instances['Part-2-1'].edges
side1Edges1 = s1.findAt(((12.040758, 30.101425, 0.0), ), ((14.246746,
    34.489153, 0.0), ), ((14.167268, 37.373443, 0.0), ), ((10.916617, 39.45003,
    0.0), ))
a.Surface(side1Edges=side1Edges1, name='Surf-1')
#: The surface 'Surf-1' has been created (4 edges).
a = mdb.models['Model-1'].rootAssembly
s1 = a.instances['Part-1-1'].edges
side1Edges1 = s1.findAt(((13.859242, 29.898575, 0.0), ), ((12.041024,
    26.803413, 0.0), ), ((10.364043, 20.836884, 0.0), ), ((12.250148,
    18.649911, 0.0), ))
a.Surface(side1Edges=side1Edges1, name='Surf-2')
#: The surface 'Surf-2' has been created (4 edges).
a = mdb.models['Model-1'].rootAssembly
region1=a.surfaces['Surf-1']
a = mdb.models['Model-1'].rootAssembly
region2=a.surfaces['Surf-2']
mdb.models['Model-1'].SurfaceToSurfaceContactStd(name='Int-1',
    createStepName='Step-1', master=region1, slave=region2, sliding=FINITE,
    thickness=ON, interactionProperty='IntProp-1', adjustMethod=NONE,
    initialClearance=OMIT, datumAxis=None, clearanceRegion=None)
#: The interaction "Int-1" has been created.
session.viewports['Viewport: 1'].assemblyDisplay.setValues(loads=ON, bcs=ON,
    predefinedFields=ON, interactions=OFF, constraints=OFF,
    engineeringFeatures=OFF)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Initial')
a = mdb.models['Model-1'].rootAssembly
e1 = a.instances['Part-2-1'].edges
edges1 = e1.findAt(((5.0, 57.0, 0.0), ))
region = a.Set(edges=edges1, name='Set-1')
mdb.models['Model-1'].EncastreBC(name='BC-1', createStepName='Initial',
    region=region, localCsys=None)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-1')
a = mdb.models['Model-1'].rootAssembly
e1 = a.instances['Part-1-1'].edges
edges1 = e1.findAt(((5.0, 0.0, 0.0), ))
region = a.Set(edges=edges1, name='Set-2')
mdb.models['Model-1'].DisplacementBC(name='BC-2', createStepName='Step-1',
    region=region, u1=0.0, u2=20.0, ur3=0.0, amplitude=UNSET, fixed=OFF,
    distributionType=UNIFORM, fieldName='', localCsys=None)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(mesh=ON, loads=OFF,
    bcs=OFF, predefinedFields=OFF, connectors=OFF)
session.viewports['Viewport: 1'].assemblyDisplay.meshOptions.setValues(
    meshTechnique=ON)
p = mdb.models['Model-1'].parts['Part-2']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
session.viewports['Viewport: 1'].partDisplay.setValues(sectionAssignments=OFF,
    engineeringFeatures=OFF, mesh=ON)
session.viewports['Viewport: 1'].partDisplay.meshOptions.setValues(
    meshTechnique=ON)
p = mdb.models['Model-1'].parts['Part-1']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
p = mdb.models['Model-1'].parts['Part-1']
f = p.faces
pickedRegions = f.findAt(((4.0, 4.666667, 0.0), ))
p.setMeshControls(regions=pickedRegions, elemShape=QUAD, allowMapped=False)
p = mdb.models['Model-1'].parts['Part-1']
p.seedPart(size=1.0, deviationFactor=0.1, minSizeFactor=0.1)
p = mdb.models['Model-1'].parts['Part-1']
p.generateMesh()
p = mdb.models['Model-1'].parts['Part-2']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
p = mdb.models['Model-1'].parts['Part-2']
f = p.faces
```

```
pickedRegions = f.findAt(((1.666667, -4.666667, 0.0), ))
p.setMeshControls(regions=pickedRegions, elemShape=QUAD, allowMapped=False)
p = mdb.models['Model-1'].parts['Part-2']
p.seedPart(size=1.0, deviationFactor=0.1, minSizeFactor=0.1)
p = mdb.models['Model-1'].parts['Part-2']
p.generateMesh()
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
a1 = mdb.models['Model-1'].rootAssembly
a1.regenerate()
session.viewports['Viewport: 1'].assemblyDisplay.setValues(mesh=OFF)
session.viewports['Viewport: 1'].assemblyDisplay.meshOptions.setValues(
    meshTechnique=OFF)
mdb.Job(name='vem14', model='Model-1', description='', type=ANALYSIS,
    atTime=None, waitMinutes=0, waitHours=0, queue=None, memory=90,
    memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
    explicitPrecision=SINGLE, nodalOutputPrecision=SINGLE, echoPrint=OFF,
    modelPrint=OFF, contactPrint=OFF, historyPrint=OFF, userSubroutine='',
    scratch='', resultsFormat=ODB, multiprocessingMode=DEFAULT, numCpus=1,
    numGPUs=0)
```