

Контакт на ANSYS

1 Построение геометрии в SpaceClaim

```
# isFirst = True при первом запуске (когда модель пустая)
# isFirst = False при последующих запусках
# Если всё пересоздавать, то обновляются ID тел
# И для сферы на следующем этапе сбрасывается свойство "Rigid"

shareTopology_tol = 1.e-10
#isFirst = True
isFirst = False

# первом запуске
if isFirst:
    ClearAll()
    ComponentHelper.CreateNewComponent()
    GetRootPart().Components[0].SetName("halfSpace")
    ComponentHelper.SetRootActive()
    ComponentHelper.CreateNewComponent()
    GetRootPart().Components[1].SetName("sphere")
    ComponentHelper.SetRootActive()

if not isFirst:
    GetRootPart().Components[0].Content.ClearAllPartData()

# входные параметры
R = Parameters.R
c1 = Parameters.c1
c2 = Parameters.c2
c3 = c2*2
eps = R/10000

from System import Array
v = Array.CreateInstance(Point, 19)

# z = -c3
v[0] = Point.Create(-c2, -c2, -c3)
v[1] = Point.Create(0, -c2, -c3)
v[2] = Point.Create(-c2, 0, -c3)
v[3] = Point.Create(0, 0, -c3)
# z = -c2
v[4] = Point.Create(-c2, -c2, -c2)
v[5] = Point.Create(0, -c2, -c2)
v[6] = Point.Create(-c2, 0, -c2)
v[7] = Point.Create(0, 0, -c2)
# z = -c1
v[8] = Point.Create(-c1, -c1, -c1)
v[9] = Point.Create(0, -c1, -c1)
v[10] = Point.Create(-c1, 0, -c1)
v[11] = Point.Create(0, 0, -c1)
# z = 0
v[12] = Point.Create(-c2, -c2, 0)
v[13] = Point.Create(0, -c2, 0)
v[14] = Point.Create(-c1, -c1, 0)
v[15] = Point.Create(0, -c1, 0)
v[16] = Point.Create(-c2, 0, 0)
v[17] = Point.Create(-c1, 0, 0)
v[18] = Point.Create(0, 0, 0)

# Шестигранные подобласти
SketchRectangle.Create(v[0],v[1],v[2])
SketchRectangle.Create(v[4],v[5],v[6])
ViewHelper.SetViewMode(InteractionMode.Solid)
selection = Selection.Create(GetRootPart().Bodies[0].Faces[0], GetRootPart().Bodies[1].Faces[0])
Loft.Create(selection)
selection = Selection.Create(GetRootPart().Bodies[0])
component = Selection.Create(GetRootPart().Components[0])
ComponentHelper.MoveBodiesToComponent(selection, component, False)
```

```

SketchRectangle.Create(v[4],v[5],v[6])
SketchRectangle.Create(v[8],v[9],v[10])
ViewHelper.SetViewMode(InteractionMode.Solid)
selection = Selection.Create(GetRootPart().Bodies[0].Faces[0], GetRootPart().Bodies[1].Faces[0])
Loft.Create(selection)
selection = Selection.Create(GetRootPart().Bodies[0])
component = Selection.Create(GetRootPart().Components[0])
ComponentHelper.MoveBodiesToComponent(selection, component, False)

```

```

SketchRectangle.Create(v[4],v[5],v[12])
SketchRectangle.Create(v[8],v[9],v[14])
ViewHelper.SetViewMode(InteractionMode.Solid)
selection = Selection.Create(GetRootPart().Bodies[0].Faces[0], GetRootPart().Bodies[1].Faces[0])
Loft.Create(selection)
selection = Selection.Create(GetRootPart().Bodies[0])
component = Selection.Create(GetRootPart().Components[0])
ComponentHelper.MoveBodiesToComponent(selection, component, False)

```

```

SketchRectangle.Create(v[4],v[6],v[12])
SketchRectangle.Create(v[8],v[10],v[14])
ViewHelper.SetViewMode(InteractionMode.Solid)
selection = Selection.Create(GetRootPart().Bodies[0].Faces[0], GetRootPart().Bodies[1].Faces[0])
Loft.Create(selection)
selection = Selection.Create(GetRootPart().Bodies[0])
component = Selection.Create(GetRootPart().Components[0])
ComponentHelper.MoveBodiesToComponent(selection, component, False)

```

```

SketchRectangle.Create(v[8],v[9],v[10])
SketchRectangle.Create(v[14],v[15],v[17])
ViewHelper.SetViewMode(InteractionMode.Solid)
selection = Selection.Create(GetRootPart().Bodies[0].Faces[0], GetRootPart().Bodies[1].Faces[0])
Loft.Create(selection)
selection = Selection.Create(GetRootPart().Bodies[0])
component = Selection.Create(GetRootPart().Components[0])
ComponentHelper.MoveBodiesToComponent(selection, component, False)

```

if isFirst:

```

# 1/8 шара
# Добавление шара
SphereBody.Create(Point.Create(0, 0, R), Point.Create(R, 0, R), ExtrudeType.None)
# Разрезание шара 3-мя плоскостями
# Плоскость X = 0
bodySelection = Selection.Create(GetRootPart().Bodies[0])
frame = Frame.Create(Point.Create(eps, 0, 0), Direction.Create(1, 0, 0))
plane = Plane.Create(frame)
SplitBody.Execute(bodySelection, plane)
selection = Selection.Create(GetRootPart().Bodies[1])
Delete.Execute(selection)
# Плоскость Y = 0
bodySelection = Selection.Create(GetRootPart().Bodies[0])
frame = Frame.Create(Point.Create(0, eps, 0), Direction.Create(0, 1, 0))
plane = Plane.Create(frame)
SplitBody.Execute(bodySelection, plane)
selection = Selection.Create(GetRootPart().Bodies[1])
Delete.Execute(selection)
# Плоскость Z = R
bodySelection = Selection.Create(GetRootPart().Bodies[0])
frame = Frame.Create(Point.Create(0, 0, R), Direction.Create(0, 0, 1))
plane = Plane.Create(frame)
SplitBody.Execute(bodySelection, plane)
selection = Selection.Create(GetRootPart().Bodies[1])
Delete.Execute(selection)

selection = Selection.Create(GetRootPart().Bodies[0])
component = Selection.Create(GetRootPart().Components[1])
ComponentHelper.MoveBodiesToComponent(selection, component, False)

```

именованные тела

полупространство

```

primarySelection = Selection.Create([
    GetRootPart().Components[0].Content.Bodies[0],
    GetRootPart().Components[0].Content.Bodies[1],
    GetRootPart().Components[0].Content.Bodies[2],

```

```

    GetRootPart().Components[0].Content.Bodies[3],
    GetRootPart().Components[0].Content.Bodies[4]))
secondarySelection = Selection()
NamedSelection.Create(primarySelection, secondarySelection)
NamedSelection.Rename("    1", "halfSpace")
# шаг
if isFirst:
    primarySelection = Selection.Create(GetRootPart().Components[1].Content.Bodies[0])
    secondarySelection = Selection()
    NamedSelection.Create(primarySelection, secondarySelection)
    NamedSelection.Rename("    1", "sphere")

# именованные поверхности
# X = 0
primarySelection = Selection.Create([
    GetRootPart().Components[0].Content.Bodies[2].Faces[0],
    GetRootPart().Components[0].Content.Bodies[4].Faces[0],
    GetRootPart().Components[0].Content.Bodies[1].Faces[0],
    GetRootPart().Components[0].Content.Bodies[0].Faces[0]])
secondarySelection = Selection()
NamedSelection.Create(primarySelection, secondarySelection)
NamedSelection.Rename("    1", "bc1_X")
# Y = 0
primarySelection = Selection.Create([
    GetRootPart().Components[0].Content.Bodies[4].Faces[3],
    GetRootPart().Components[0].Content.Bodies[3].Faces[0],
    GetRootPart().Components[0].Content.Bodies[1].Faces[3],
    GetRootPart().Components[0].Content.Bodies[0].Faces[3]])
secondarySelection = Selection()
NamedSelection.Create(primarySelection, secondarySelection)
NamedSelection.Rename("    1", "bc1_Y")
# Z = 0
primarySelection = Selection.Create(GetRootPart().Components[0].Content.Bodies[0].Faces[5])
secondarySelection = Selection()
NamedSelection.Create(primarySelection, secondarySelection)
NamedSelection.Rename("    1", "bc1_Z")
# контакт contact
primarySelection = Selection.Create([
    GetRootPart().Components[0].Content.Bodies[4].Faces[4],
    GetRootPart().Components[0].Content.Bodies[3].Faces[3],
    GetRootPart().Components[0].Content.Bodies[2].Faces[3]])
secondarySelection = Selection()
NamedSelection.Create(primarySelection, secondarySelection)
NamedSelection.Rename("    1", "contact")
# контакт target
if isFirst:
    primarySelection = Selection.Create(GetRootPart().Components[1].Content.Bodies[0].Faces[3])
    secondarySelection = Selection()
    NamedSelection.Create(primarySelection, secondarySelection)
    NamedSelection.Rename("    1", "target")

# именованные рёбра
# рёбра без сгущения с разбиением N
primarySelection = Selection.Create2([
    GetRootPart().Components[0].Content.Bodies[4].Edges[4],
    GetRootPart().Components[0].Content.Bodies[4].Edges[3],
    GetRootPart().Components[0].Content.Bodies[4].Edges[6],
    GetRootPart().Components[0].Content.Bodies[4].Edges[0],
    GetRootPart().Components[0].Content.Bodies[4].Edges[7],
    GetRootPart().Components[0].Content.Bodies[4].Edges[10],
    GetRootPart().Components[0].Content.Bodies[4].Edges[9],
    GetRootPart().Components[0].Content.Bodies[4].Edges[11],
    GetRootPart().Components[0].Content.Bodies[4].Edges[1],
    GetRootPart().Components[0].Content.Bodies[4].Edges[8],
    GetRootPart().Components[0].Content.Bodies[4].Edges[2],
    GetRootPart().Components[0].Content.Bodies[4].Edges[5],
    GetRootPart().Components[0].Content.Bodies[3].Edges[0],
    GetRootPart().Components[0].Content.Bodies[3].Edges[10],
    GetRootPart().Components[0].Content.Bodies[3].Edges[7],
    GetRootPart().Components[0].Content.Bodies[3].Edges[4],
    GetRootPart().Components[0].Content.Bodies[2].Edges[10],
    GetRootPart().Components[0].Content.Bodies[2].Edges[4],
    GetRootPart().Components[0].Content.Bodies[2].Edges[0],

```

```

GetRootPart().Components[0].Content.Bodies[1].Edges[10],
GetRootPart().Components[0].Content.Bodies[1].Edges[0],
GetRootPart().Components[0].Content.Bodies[0].Edges[7],
GetRootPart().Components[0].Content.Bodies[0].Edges[10],
GetRootPart().Components[0].Content.Bodies[0].Edges[0],
GetRootPart().Components[0].Content.Bodies[0].Edges[4]])
secondarySelection = Selection()
NamedSelection.Create(primarySelection, secondarySelection)
NamedSelection.Rename(" 1", "edges_in")
# рёбра со сгущением
primarySelection = Selection.Create([
    GetRootPart().Components[0].Content.Bodies[1].Edges[6],
    GetRootPart().Components[0].Content.Bodies[1].Edges[9],
    GetRootPart().Components[0].Content.Bodies[1].Edges[3],
    GetRootPart().Components[0].Content.Bodies[1].Edges[1],
    GetRootPart().Components[0].Content.Bodies[2].Edges[8],
    GetRootPart().Components[0].Content.Bodies[2].Edges[3],
    GetRootPart().Components[0].Content.Bodies[3].Edges[1]])
secondarySelection = Selection()
NamedSelection.Create(primarySelection, secondarySelection)
NamedSelection.Rename(" 1", "edges_thickening")
# рёбра без сгущения с разбиением N/2
primarySelection = Selection.Create([
    GetRootPart().Components[0].Content.Bodies[0].Edges[9],
    GetRootPart().Components[0].Content.Bodies[0].Edges[6],
    GetRootPart().Components[0].Content.Bodies[0].Edges[3],
    GetRootPart().Components[0].Content.Bodies[0].Edges[1]])
secondarySelection = Selection()
NamedSelection.Create(primarySelection, secondarySelection)
NamedSelection.Rename(" 1", "edges_half_N")

# Share Topology
options = ShareTopologyOptions()
options.Tolerance = shareTopology_tol
result = ShareTopology.FindAndFix(options)

```

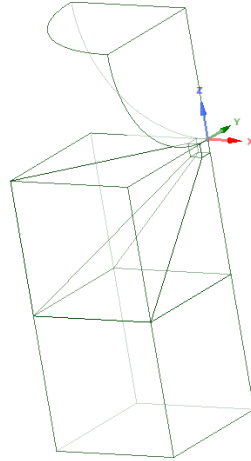


Рис. 1: Геометрия при $n = 0$, $\bar{\delta}_{max} = 110$, $\frac{E^*}{Y} = 550$

2 Сетка

Для полупространства заданы фиксированные количества промежутков на рёбрах и сгущение к центру. Для поверхности шара заданы размеры элементов вблизи точки начального контакта.

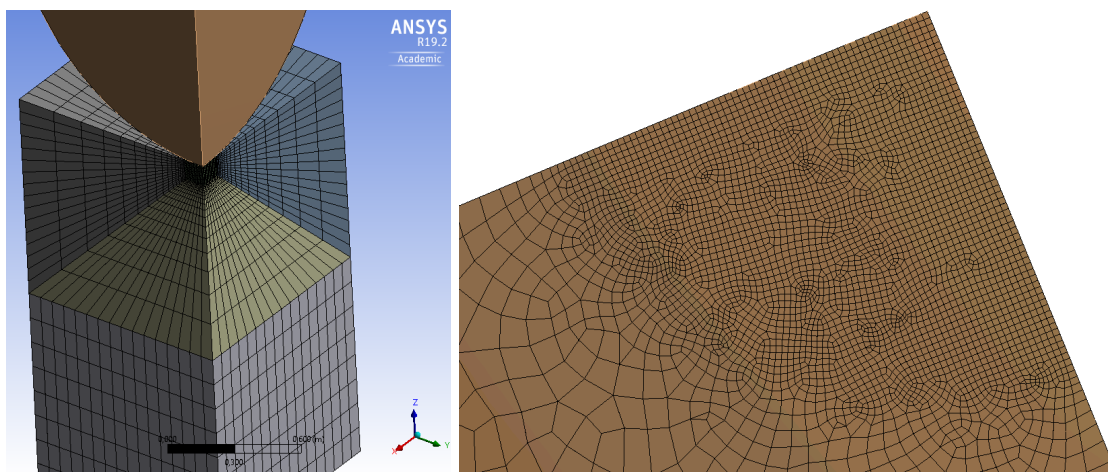


Рис. 2: Сетка при $n = 0$, $\bar{\delta}_{max} = 110$, $\frac{E^*}{Y} = 550$, $N = 16$

3 Контакт

Задаётся метод множителей Лагранжа (метод штрафа требует регулировки параметра контактной жёсткости; расширенный метод Лагранжа, который заключается в многократном применении метода штрафа, тоже требует настройки параметров и плохо сходится). Направление силы, действующей на полупространство — по нормали к поверхности шара ("Nodal-Normal To Target" = "Nodal-Normal To Target"). Для поверхности шара задана преимущественно 4-угольная сетка 2-го порядка со сглаживанием сферой (сглаживаются узлы в центрах 4-угольников).

Поверхность полупространства ("контактная") задаётся элементами CONTAC174, поверхность шара ("целевая") задаётся элементами TARGE170.

Details of "Frictionless - contact To target"	
[-] Scope	
Scoping Method	Named Selection
Contact	contact
Target	target
Contact Bodies	Multiple
Target Bodies	sphere\Твердое тело
Protected	Yes
[-] Definition	
Type	Frictionless
Scope Mode	Manual
Behavior	Asymmetric
Trim Contact	Off
Suppressed	No
[-] Advanced	
Formulation	Normal Lagrange
Small Sliding	Off
Detection Method	Nodal-Normal To Target
Stabilization Damping Factor	Q,
Pinball Region	Radius
Pinball Radius	Q,1 m
Time Step Controls	None
[-] Geometric Modification	
Interface Treatment	Add Offset, No Ramping
<input type="checkbox"/> Offset	Q, m
Contact Geometry Correction	None
Target Geometry Correction	Smoothing
Target Orientation	Sphere Center Point
Target Center Point	sphereCenter

Рис. 3: Условия контакта

4 Материал полупространства

Упруго-пластичный материал с изотропным упрочнением задаётся кусочно-линейной диаграммой одноосного растяжения (TB,MISO,,). Диаграмма для степенного упрочнения:

$$\begin{cases} \sigma = E\varepsilon, & \text{если } \sigma < Y, \\ \sigma = Y \left(\frac{\varepsilon}{\varepsilon_Y} \right)^n, & \text{если } \sigma \geq Y, \end{cases} \quad (1)$$

где $\varepsilon_Y = Y/E$.

При $n = 0$ задаётся билинейная диаграмма (TB,BISO,,).

Базисные функции первого порядка, элементы SOLID185.

Details of "Mesh"	
+ Display	
- Defaults	
Physics Preference	Nonlinear Mechanical
Element Order	Linear
<input type="checkbox"/> Element Size	Default (0,66247 m)
+ Sizing	
+ Quality	
+ Inflation	
+ Advanced	
- Statistics	
<input type="checkbox"/> Nodes	20479
<input type="checkbox"/> Elements	18755

Рис. 4: Условия контакта

Количество элементов: $16^3 * 4.5$ шестигранных, 16^2 контактных, плюс, видимо, соседние $16 * 4 + 3$ контактных (+3 т.к. в углу по диагонали нет соседнего), итого 18755. То есть количество элементов правильное.

(Встроенное степенное упрочнение (TB,NLISO,,,POWER) задаёт другую кривую при $n \neq 0$.)

5 Краевые условия

Зафиксированы перемещения по x , y , z на поверхностях $x = 0$, $y = 0$, $z = h$, где h — высота образца. Шар как твёрдое тело движется по траектории к полупространству, затем возвращается.

6 Параметры решателя

Задан метод Ньютона-Рафсона, фиксированное количество подшагов.

[-] Step Controls	
Number Of Steps	2,
Current Step Number	1,
Step End Time	110, s
Auto Time Stepping	Off
Define By	Substeps
Number Of Substeps	110,
[-] Solver Controls	
Solver Type	Program Controlled
Weak Springs	Off
Solver Pivot Checking	Program Controlled
Large Deflection	Off
Inertia Relief	Off
[-] Rotordynamics Controls	
Coriolis Effect	Off
[-] Restart Controls	
Generate Restart Poi...	Off
Retain Files After Fu...	No
Combine Restart Files	No
[-] Nonlinear Controls	
Newton-Raphson O...	Full
Force Convergence	On
--Value	Calculated by solver
--Tolerance	1,e-006%
--Minimum Reference	1,e-002 N
Moment Convergence	Remove
Displacement Conve...	On
--Value	Calculated by solver
--Tolerance	1,e-006%
--Minimum Reference	0, m
Rotation Convergen...	Remove
Line Search	Off
Stabilization	Off

Рис. 5: Параметры решателя

7 PRE ADPL

Команды, запускающиеся перед первым шагом:

```
! Main parameters
n_powerlaw = ARG1      !power law coefficient
bd_max = ARG2          !nondimensional indentation depth
bYldCoef = ARG3        !E*/Y
n_substeps = ARG4      !fixed load/unload substeps number

! Calculated parameters
sigma_y = ARG5         !yield sigma
d_max = ARG6           !max indentation depth
! Calculated parameters for calculating nondimensional values
d_Y = ARG7             !begin flow indentation depth
P_Y = ARG8             !begin flow pressure
A_Y = ARG9             !begin flow area

! Fixed parameters
mat_index = 1          !index of elasto-plastic matherial (change if index != 1)
mat_E = 10000000000    !Young's modulus
mat_NU = 0.3           !Poisson's ratio
mat_curve_N = 100      !the number of multilinear curve points
mat_curve_max_eps = 0.1 !curve for eps = eps_y...(eps_y + mat_curve_max_eps)

! Rigid sphere displacement table d(time)
! Time values
_loadvari180uz(1,0,1) = 0.
_loadvari180uz(2,0,1) = 110.
_loadvari180uz(3,0,1) = 220.
```



```

! Load values
_loadvari180uz(1,1,1) = 0.
_loadvari180uz(2,1,1) = -d_max
_loadvari180uz(3,1,1) = 0

! Plastic curve
TBDELE,PLAS,mat_index      !delete wb plastic curve
*IF, n_powerlaw, EQ, 0, THEN
! 1) n_powerlaw = 0: bilinear isotropic hardening model
TB,BISO,mat_index,1      !"1" - one fixed temperature point
TBDAT,1, sigma_y      !yield sigma
TBDAT,2, 0.      !horizontal (tan = 0)
*ELSE
! 2) n_powerlaw != 0: multilinear isotropic hardening model
!note: TB,MISO,, - 1D sigma_eqv(eps_eqv) curve (isotropic multilinear hardening)
!note: TB,PLAS,,MISO - 1D sigma_eqv(q) curve (isotropic multilinear hardening)
TB,MISO,mat_index,1,mat_curve_N      !"1" - one fixed temperature point
!TB,PLAS,mat_index,1,mat_curve_N,MISO      !"1" - one fixed temperature point, MISO - isotropic hardening
TBTEMP,22      !our fixed default temperature value

mat_curve_fn = 'curve_%n_powerlaw%'      !curve output file name
! Delete output file
!fn_out1
/OUTPUT, mat_curve_fn, txt,
/OUTPUT
/DELETE, mat_curve_fn, txt,

!curve function: F(eps) = sigma_y*(eps/eps_y)^n
eps_y = sigma_y/mat_E
*do,i,0,mat_curve_N-1,      !multilinear curve points cycle
!curve_eps = eps_y + (i/(mat_curve_N-1.))*mat_curve_max_eps
!need more points at begin of curve, so
curve_eps = eps_y + (i**3)/((mat_curve_N-1.）**3)*mat_curve_max_eps
curve_sigma = sigma_y*((curve_eps/eps_y)**n_powerlaw)
TBPT,,curve_eps,curve_sigma
! Output curve points in file
/OUTPUT, mat_curve_fn, txt,, APPEND
*VWRITE,curve_eps,curve_sigma, "" "" "" ""
%17.10G %17.10G
/OUTPUT
*enddo !end of i cycle
*ENDIF

! Elastic modulus of plastic matherial
MP,EX,mat_index,mat_E,
MP,NUXY,mat_index,mat_NU,

! Fictitious elastic matherial for rigid sphere
MP,EX,6,1e+20,      ! Pa
MP,NUXY,6,0.3,

```

Команды, запускающиеся перед каждым шагом:

```

nsub,n_substeps,n_substeps      !fixed substeps number
NEQIT, 100      !max iterations number

```

8 POST1 ADPL

Площадь рассчитывается 2-мя способами:

1. Суммирование площадей контакта на всех контактных элементах (etable, carea, nmisc, 58). Такая площадь совпадает с площадью, которая выводится по-

средством GUI. Площадь внутри ANSYS рассчитывается по формуле

$$S = \sum_i \frac{N_i}{4} S_i \quad (2)$$

где S_i — площадь 4-угольного элемента контактной поверхности, N_i — количество узлов этого 4-угольника, находящихся в состоянии контакта. Выяснилось благодаря совпадению с таким же расчётом площади своим решателем. Если "Detection Method" = "Nodal-Projected Normal From Contact" то площадь рассчитывается каким-то другим способом и получается более гладкая (наверно учитывается сетка целевой поверхности контакта).

2. Суммирование площадей контактных элементах, которые находятся в статусе контактирующих (т.е. нижняя граница). Равносильно суммированию площадей 4-угольных элементов контактной поверхности, у которых все 4 узла находятся в состоянии контакта:

$$S = \sum_i \left[\frac{N_i}{4} \right] S_i. \quad (3)$$

Сила реакции рассчитывается суммированием сил реакции узлов, принадлежащих контактным элементам, вдоль оси Z . Совпадает с результатом в GUI.

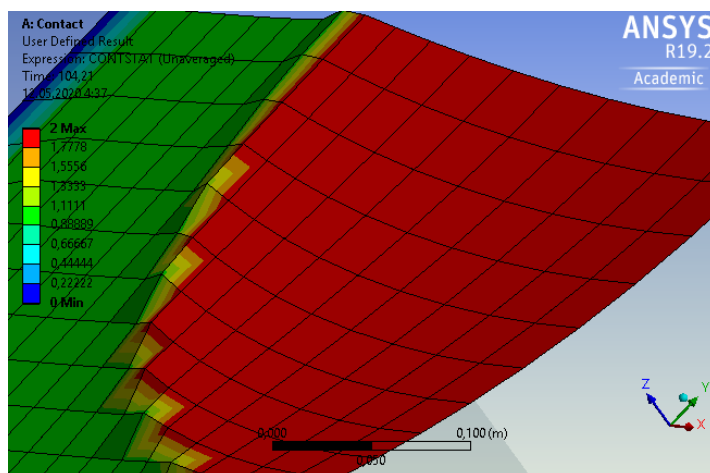


Рис. 6: Статус контакта

RESUME

```

fn_out1 = 'rt_%n_powerlaw%_%bd_max%_%bYeldCoef%' !output1 file name
fn_out2 = 'rv_%n_powerlaw%_%bd_max%_%bYeldCoef%' !output2 file name

!delete output files
!fn_out1
/OUTPUT, fn_out1, txt,
/OUTPUT
/DELETE, fn_out1, txt,
!fn_out2
/OUTPUT, fn_out2, txt,
/OUTPUT
/DELETE, fn_out2, txt,

sep_d = 0 !variable for finding separation depth

*do,loadstep,1,2, !loadsteps cycle
*do,substep,1,n_substeps,!substeps cycle
SET, loadstep, substep !use our step and substep numbers

! 1) contact area by NMISC
CMSEL, S, contact !select contact (contact - named selection)
ESLN,S,1 !select elements attached to these nodes
ESEL,R,ENAME,,174 !reselect the contact elements (174 - type of contact elements)
ESEL, STAT !write debug information
ETABLE,carea,NMISC,58 !create table of element contact areas (carea - name of our table)
ESEL, STAT !write debug information
SSUM
*GET, nmisc_area, SSUM, 0, ITEM, carea !solve sum of contact element areas from our table carea

! 2) contact area by STAT+VOLU
CMSEL, S, contact !select contact (contact - named selection)
ESLN,S,1 !select elements attached to these nodes
ESEL,R,ENAME,,174 !reselect the contact elements (174 - type of contact elements)
ESEL, STAT !write debug information
ETABLE, etstat, CONT, STAT !create table of contact stats (etstat - name of our table)
ESEL, S, ETAB, etstat, 2,3 !select from our table etstat if value between 2 and 3
ESEL, STAT !write debug information
*GET,num_el,ELEM,0,COUNT !count number of selected contact elements
ETABLE, c_area, VOLU !create table of contact elements areas
ETABLE, STAT, CONT, STAT !write debug information
! PRETAB, c_area !write debug information
SSUM
*GET, t_area, SSUM, 0, ITEM, c_area !solve sum of contact element areas from our table c_area

ETABLE, etstat, ERAS !erase etstat
ETABLE, c_area, ERAS !erase c_area
ETABLE, carea, ERAS !erase carea
ETABLE, STAT, CONT, STAT !write debug information

! 3) contact reaction
CMSEL, S, contact
ESLN,S,1 !select elements attached to these nodes
ESEL,R,ENAME,,174 !reselect the contact elements (174 - type of contact elements)
NSLE, S !select nodes attached to these elements
NSEL, STAT !write debug information
FSUM, ,CONT !solve sum of contact nodal forces

*GET, t_P, FSUM, 0, ITEM, FZ !FZ sum

!out our results
*IF, loadstep, EQ, 1, THEN
d = d_max*substep/n_substeps
*ELSE
d = d_max*(n_substeps-substep)/n_substeps
*ENDIF
P = -t_P*4
A = t_area*4
A_nmisc = nmisc_area*4

bd = d/d_Y
bP = P/P_Y

```

```

bA = A/A_Y
bA_nmisc = A_nmisc/A_Y

!seporation depth
*IF, sep_d, EQ, 0, THEN      !sep_depth is not found yet
*IF, loadstep, EQ, 2, THEN  !unloading
*IF, P, EQ, 0, THEN         !separated
sep_d = d
*ENDIF
*ENDIF
*ENDIF

! Output results in file (P, A and A_nmisc divided by 4 to compare easy with gui wb result)
/OUTPUT, fn_out1, txt,, APPEND
*VWRITE, loadstep, substep, num_el, d, P/4, A/4, A_nmisc/4, bd, bP, bA, bA_nmisc,, ,,,
%7I %7I %7I      %17.10E %17.10E %17.10E %17.10E      %17.10E %17.10G %17.10G %17.10G
!(f8.0, f8.0, f8.0, e18.10, e18.10, e18.10, e18.10, e18.10, e18.10)
/OUTPUT

*enddo !end of sybsteps cycle
*enddo !end of loadsteps cycle

/OUTPUT, fn_out2, txt,, APPEND
*VWRITE, sep_d/d_max,, ,,, ,,, ,,, ,,,
%17.10G

/OUTPUT

```

9 Входные параметры, Exel VBS

```

Attribute VB_Name = "fill"
Function SQR(x As Double) As Double
    SQR = x * x
End Function
Function sqrt(x As Double) As Double
    sqrt = x ^ 0.5
End Function
Function pow(x As Double, y As Double) As Double
    pow = x ^ y
End Function

Sub fill()
    ' Пу
    Dim PI As Double
    PI = Application.WorksheetFunction.PI
    'магическая константа
    Dim c As Double
    c = 1.08
    ' координаты ячейки
    Dim i, j As Integer
    ' основные параметры
    Dim n_powerlaw, bd_max, bYeldCoef As Double
    Dim n_substeps As Integer 'количество шагов нагружения/разгрузки
    Dim NN As Integer 'разбиение сетки
    ' зафиксированные параметры
    ' для NN = 16 коэффициент сгущения 1/q = 1/7.970957e-01 = 1,2545545033
    ' для NN = 8 коэффициент сгущения принят на глаз 1.6 (Должно быть 1/q = 1/6.216904e-01 = 1.60851768018)
    Dim R As Double 'радиус шара
    Dim EE As Double 'модуль Юнга
    Dim Nu As Double 'коэффициент Пуассона
    Dim grid_mnojitel1 As Double 'c1=grid_mnojitel1*L_contact
    Dim grid_mnojitel2 As Double 'c2=grid_mnojitel2*L_contact
    R = 1
    EE = 10000000000#
    Nu = 0.3
    grid_mnojitel1 = 1.5
    grid_mnojitel2 = 15#

```

```

' проход по строкам
i = 3 ' основные параметры начинаются с 3-й строки
Do While 1
    If IsEmpty(Cells(i, 1)) Then
        Exit Do
    End If
    ' Чтение основных параметров
    j = 1
    n_powerlaw = Cells(i, j): j = j + 1 ' коэффициент степенного упрочнения
    bd_max = Cells(i, j): j = j + 1 ' безразмерная глубина индентации
    bYeldCoef = Cells(i, j): j = j + 1 ' E*/Y
    n_substeps = Cells(i, j): j = j + 1 ' количество шагов нагружения/разгрузки
    NN = Cells(i, j): j = j + 1 ' разбиения полупространства
    NN_sphere = Cells(i, j): j = j + 1 ' разбиение сферы (размер элемента = c1/NN_sphere)
    ' Расчёт параметров
    Dim sigma_y As Double ' предел текучести
    Dim E As Double ' приведённый модуль упругости
    Dim d_Y As Double ' глубина индентации в начале текучести
    Dim P_Y As Double ' сила реакции опоры в начале текучести
    Dim A_Y As Double ' глубина индентации в начале текучести
    Dim d_max As Double ' глубина индентации
    Dim c1 As Double ' размер внутреннего куба
    Dim c2 As Double ' размер внешнего куба
    Dim c3 As Double ' 2*c2 - общий размер образца

    E = EE / (1# - SQR(Nu))
    sigma_y = E / bYeldCoef
    P_Y = 9# / 16# * pow(PI, 3) * pow(c, 3) * SQR(R) * sigma_y / SQR(E / sigma_y)
    A_Y = 9# / 16# * pow(PI, 3) * pow(c, 2) * SQR(R) / SQR(E / sigma_y)
    d_Y = 9# / 16# * pow(PI, 2) * pow(c, 2) * R / SQR(E / sigma_y)
    d_max = bd_max * d_Y
    Dim L_contact As Double
    L_contact = sqrt(2 * R * d_max - SQR(d_max))
    c1 = grid_mnojitel1 * L_contact
    c2 = grid_mnojitel2 * L_contact
    c3 = c2 * 2#

    ' 2 пустые строки
    j = j + 1: j = j + 1
    ' Заполнение ячеек входных параметров для ANSYS
    Cells(i, j) = R: j = j + 1
    Cells(i, j) = c1: j = j + 1
    Cells(i, j) = c2: j = j + 1
    Cells(i, j) = NN: j = j + 1
    Cells(i, j) = NN: j = j + 1
    Cells(i, j) = NN / 2: j = j + 1
    Cells(i, j) = R * 2#: j = j + 1
    Cells(i, j) = R / 2#: j = j + 1
    Cells(i, j) = c1: j = j + 1
    Cells(i, j) = c1 / NN_sphere: j = j + 1 ' Размер элементов поверхности сферы в зоне возможного контакта
    Cells(i, j) = n_powerlaw: j = j + 1
    Cells(i, j) = bd_max: j = j + 1
    Cells(i, j) = bYeldCoef: j = j + 1
    Cells(i, j) = n_substeps: j = j + 1
    Cells(i, j) = sigma_y: j = j + 1
    Cells(i, j) = d_max: j = j + 1
    Cells(i, j) = d_Y: j = j + 1
    Cells(i, j) = P_Y: j = j + 1
    Cells(i, j) = A_Y: j = j + 1

    i = i + 1
Loop
End Sub

```

10 Учёт больших деформаций

Details of "Analysis Settings"	
[-] Step Controls	
Number Of Steps	2,
Current Step Number	1,
Step End Time	110, s
Auto Time Stepping	Off
Define By	Substeps
Number Of Substeps	110,
[-] Solver Controls	
Solver Type	Program Controlled
Weak Springs	Off
Solver Pivot Checking	Program Controlled
Large Deflection	On
Inertia Relief	Off
+ Rotordynamics Controls	
+ Restart Controls	
[-] Nonlinear Controls	
Newton-Raphson O...	Full
Force Convergence	On
--Value	Calculated by solver
--Tolerance	1,e-002%
--Minimum Reference	1,e-002 N
Moment Convergence	Remove
Displacement Conve...	On
--Value	Calculated by solver
--Tolerance	1,e-002%
--Minimum Reference	Q, m
Rotation Convergen...	Remove
Line Search	On
Stabilization	Off

Details of "Frictionless - contact To target"	
Contact	contact
Target	target
Contact Bodies	Multiple
Target Bodies	sphere\Твердое тело
Protected	Yes
[-] Definition	
Type	Frictionless
Scope Mode	Manual
Behavior	Asymmetric
Trim Contact	Off
Suppressed	No
[-] Advanced	
Formulation	Augmented Lagrange
Small Sliding	Off
Detection Method	Nodal-Normal To Target
Penetration Tolerance	Value
Penetration Tolerance Value	1,e-005 m
Normal Stiffness	Factor
Normal Stiffness Factor	Q,1
Update Stiffness	Never
Stabilization Damping Factor	Q,
Pinball Region	Radius
Pinball Radius	Q,1 m
Time Step Controls	None
[-] Geometric Modification	
Interface Treatment	Add Offset, No Ramping
<input type="checkbox"/> Offset	Q, m
Contact Geometry Correction	None
Target Geometry Correction	Smoothing
Target Orientation	Sphere Center Point
Target Center Point	sphereCenter

Рис. 7: Параметры решателя и контакта

Для сходимости пришлось установить 440 шагов нагружения/разгрузки вместо 110, увеличить требуемые невязки по силам и перемещениям до 10^{-4} , изменить решатель контакта на расширенный метод Лагранжа с высоким допуском зазора (сходится при зазоре примерно $\frac{1}{200}d_{max} - \frac{1}{100}d_{max}$). Особенно плохо сходится при $n = 0.9$.