

Решение 3D задач

Задача о растекании ТОКОВ

Раздел BOUNDARIES

BOUNDARIES

Region 1

START(-1,-1)

LINE TO(1,-1) TO (1,1) TO (-1,1)

TO CLOSE

Region 2

START(r,0)

ARC(CENTER=0,0) ANGLE=360

.....

2D задача

TITLE 'Heat flow around an Insulating blob'

VARIABLES

Phi { the temperature }

DEFINITIONS

K = 1 { default conductivity }

R = 0.5 { blob radius }

BOUNDARIES

REGION 1 'box'

START (-1,-1)

LINE TO (1,-1) **TO** (1,1)

TO (-1,1) **TO CLOSE**

REGION 2 'blob' { the embedded blob }

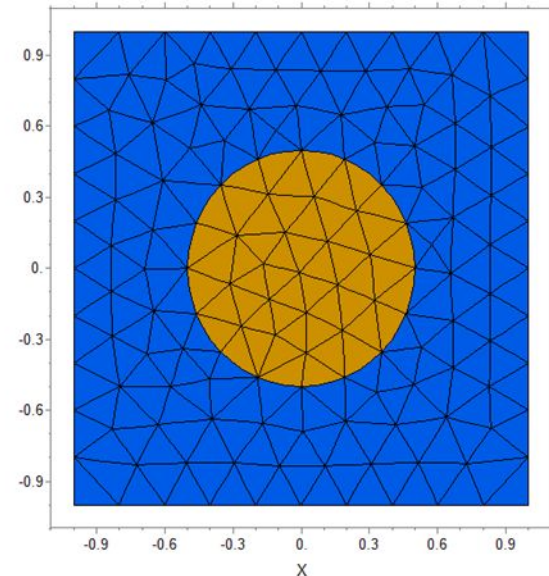
START 'ring' (R,0)

ARC(CENTER=0,0) **ANGLE**=360 **TO CLOSE**

PLOTS

grid(x,y) **paintregions** as 'final mesh'

END

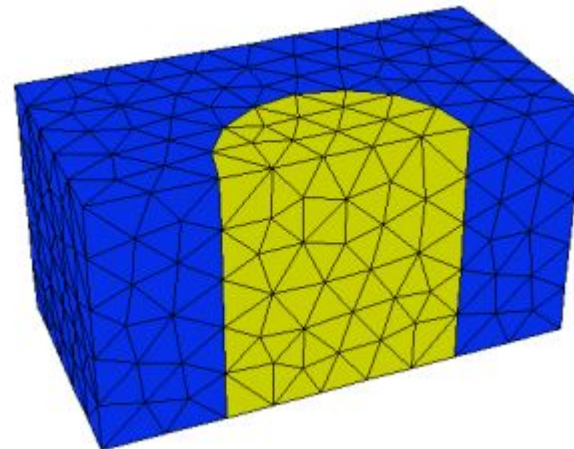
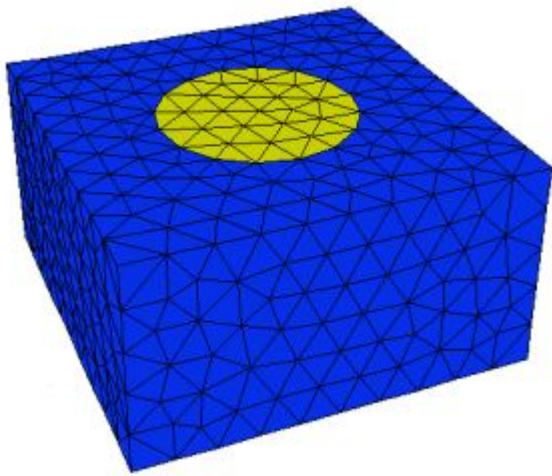


Понятие экструзии

Экструзия (от англ. *extrusion* — выталкивание, выдавливание)

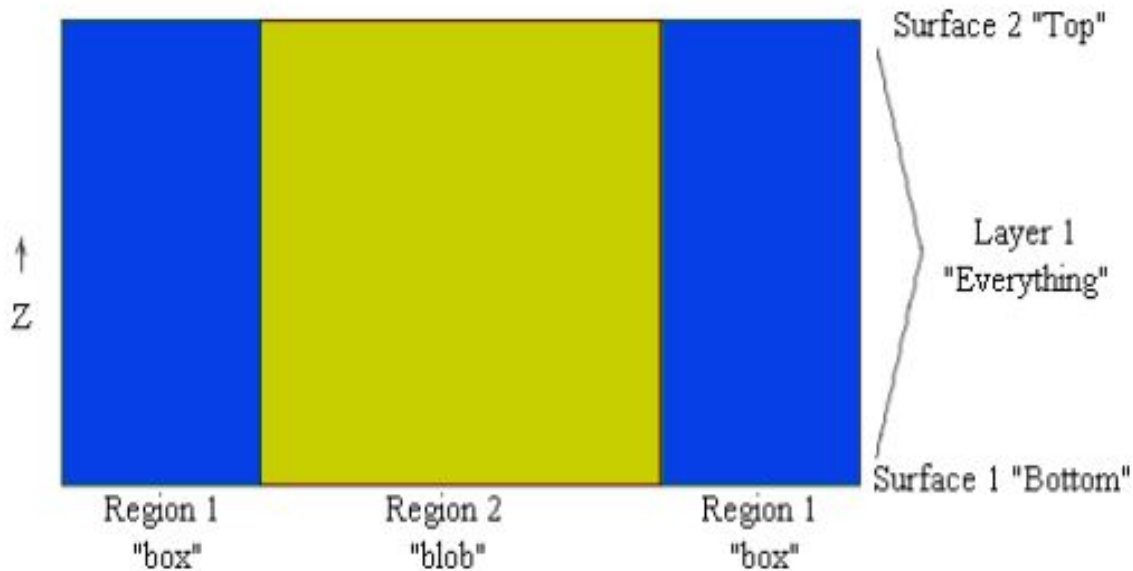
В программу (2D-скрипт) необходимо внести следующие изменения:

- В разделе координаты необходимо указать CARTESIAN3.
- Необходимо добавить новый раздел, чтобы задать расслоение экструзии - EXTRUSION.
- В разделе PLOTS графики и мониторы должны быть изменены, чтобы указать на каких поверхностях ведется изображение



Формы записи экструзии

Сложная	Сокращенная
EXTRUSION SURFACE 'Bottom' z=0 LAYER 'Everything' SURFACE 'Top' z=1	EXTRUSION z = 0, 1



TITLE 'Heat flow around an Insulating blob'

COORDINATES cartesian3

VARIABLES Phi { the temperature }

DEFINITIONS

K = 1 { default conductivity }

R = 0.5 { blob radius }

EXTRUSION

SURFACE 'Bottom' z=0

LAYER 'Everything'

SURFACE 'Top' z=1

BOUNDARIES

REGION 1 'box'

START(-1,-1)

LINE TO (1,-1) TO (1,1)

TO (-1,1) TO CLOSE

REGION 2 'blob' { the embedded blob }

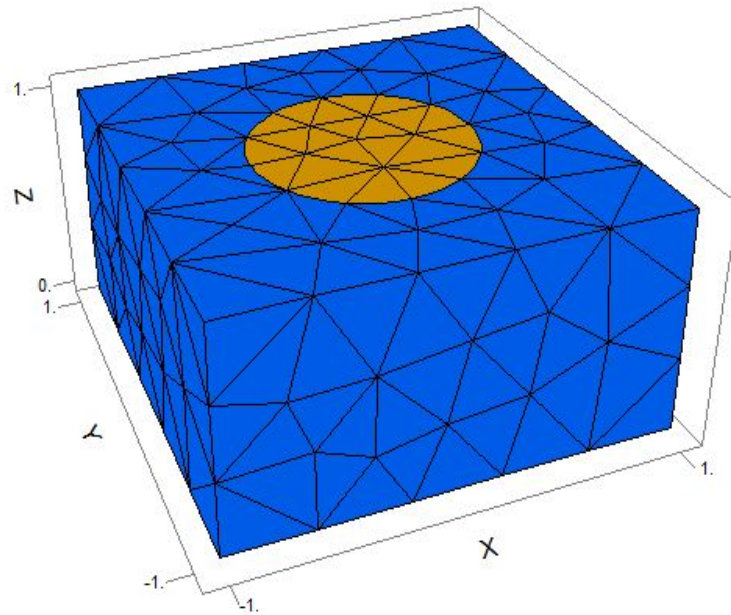
START 'ring' (R,0)

ARC(CENTER=0,0) ANGLE=360 TO CLOSE

PLOTS

grid(x,y,z) paintregions as "final mesh"

END



СЛОИ

EXTRUSION

SURFACE "Bottom" $z=-1/2$

LAYER "Underneath"

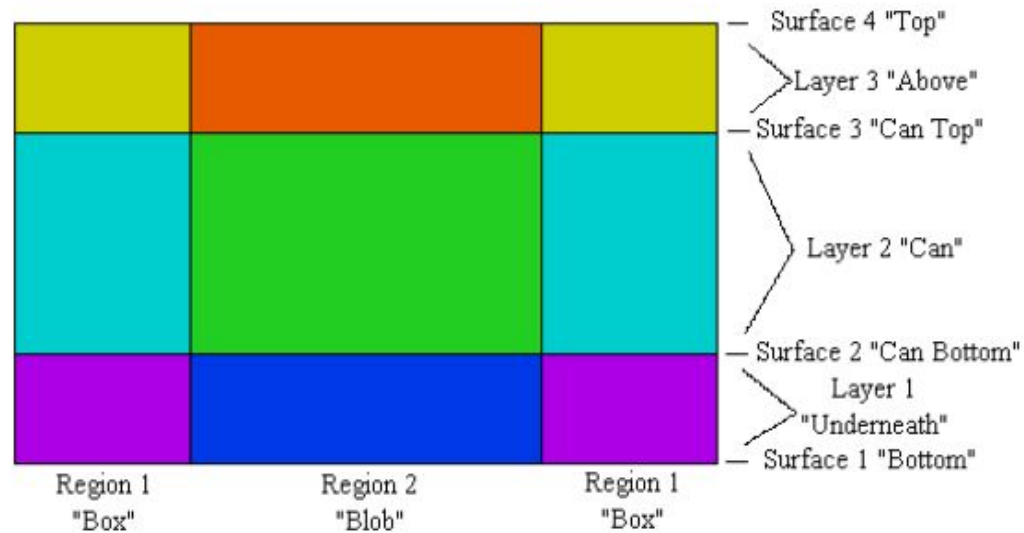
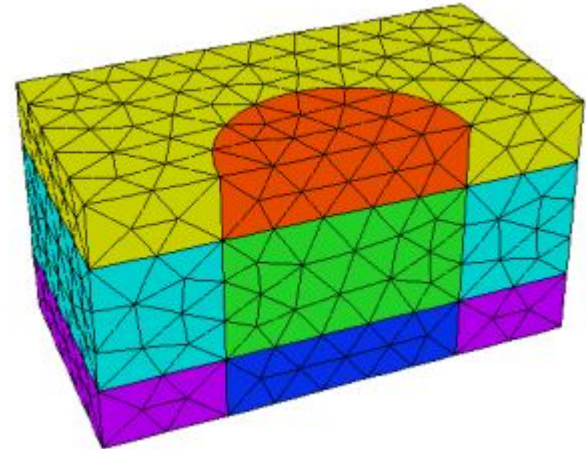
SURFACE "Can Bottom" $z=-1/4$

LAYER "Can"

SURFACE "Can Top" $z=1/4$

LAYER "Above"

SURFACE "Top" $z=1/2$

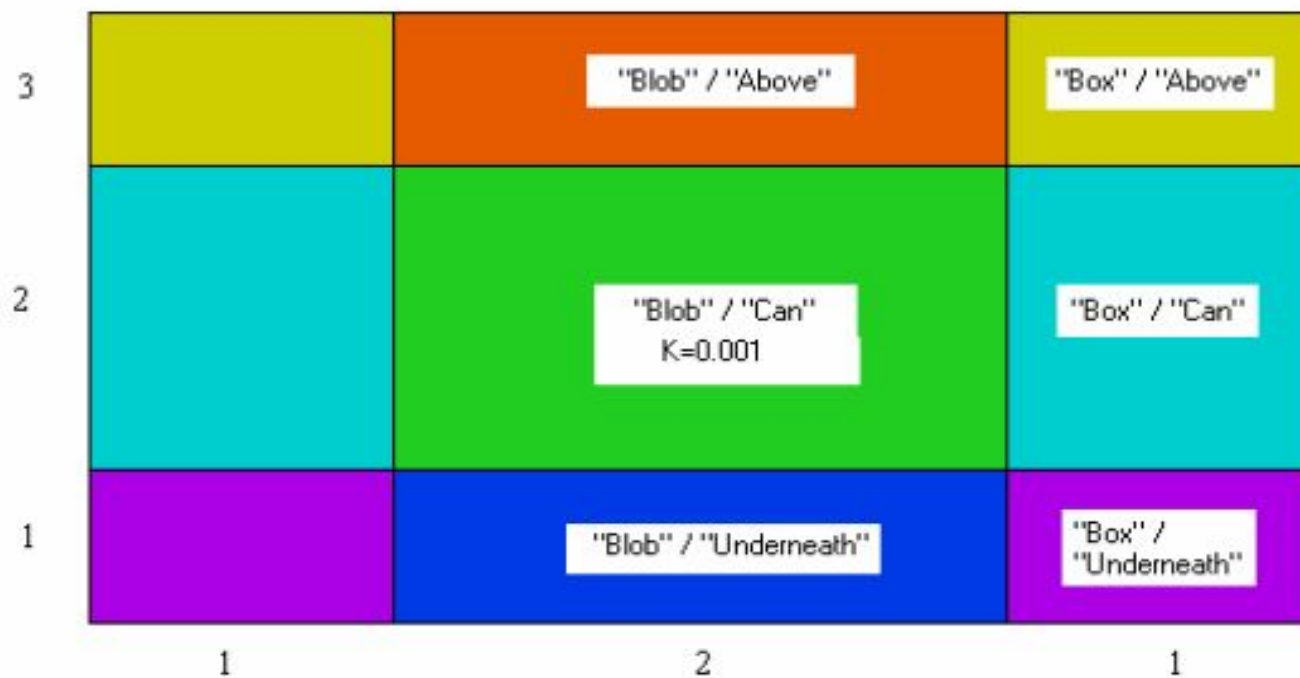


Настройка свойств материала по регионам И СЛОЯМ

```
BOUNDARIES
REGION 1
params(1,all) { parameter redefinitions for all layers of region 1 }
LAYER 1
params(1,1){ parameter redefinitions restricted to layer 1 of region 1 }
LAYER 2
params(1,2){ parameter redefinitions restricted to layer 2 of region 1 }
LAYER 3
params(1,3){ parameter redefinitions restricted to layer 3 of region 1 }
START(,) ....TO CLOSE { trace the perimeter }
REGION 2
params(2,all) { parameter redefinitions for all layers of region 2 }
LAYER 1
params(2,1) { parameter redefinitions restricted to layer 1 of region 2 }
LAYER 2
params(2,2) { parameter redefinitions restricted to layer 2 of region 2 }
LAYER 3
params(2,3) { parameter redefinitions restricted to layer 3 of region 2 }
START(,) ....TO CLOSE { trace the perimeter }
{ ... and so forth for all regions }
```

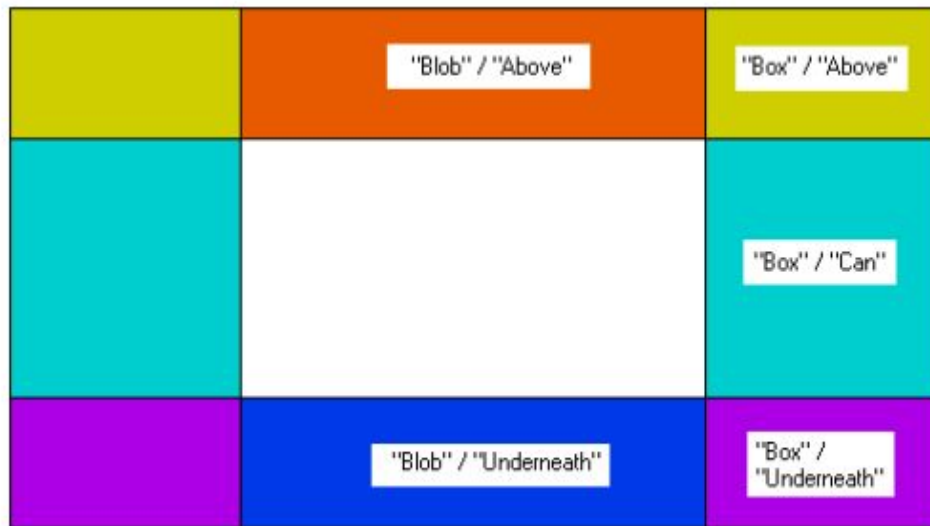
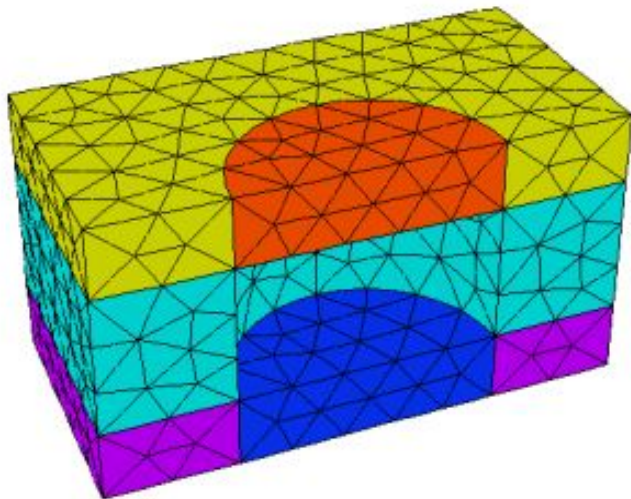

Настройка свойств материала по регионам И СЛОЯМ

```
REGION 2 'blob' {the embedded blob }  
LAYER 'Can' K = 0.001  
START 'ring' (R,0)  
ARC(CENTER=0,0) ANGLE=360
```

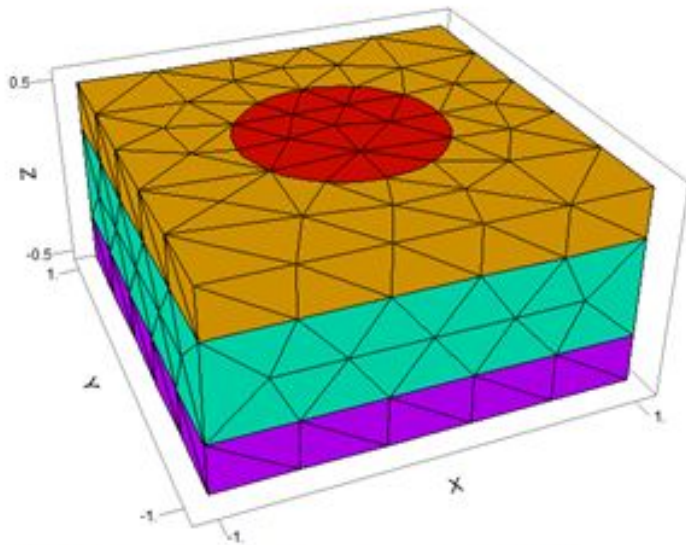


Пустые области. Зарезервированное слово VOID

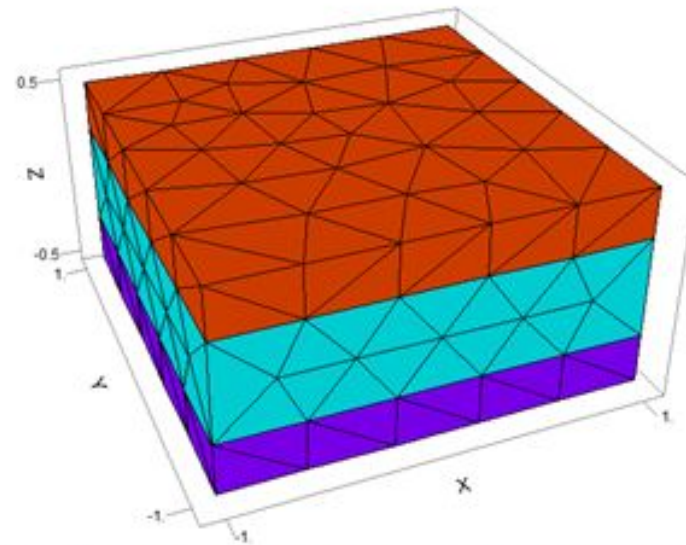
```
REGION 2 'blob' {the embedded blob }  
LAYER 'Can' VOID  
START 'ring' (R,0)  
ARC(CENTER=0,0) ANGLE=360
```



Ограниченные регионы



Экструзия цилиндра на всю область



Цилиндрическая полость внутри среднего слоя

```
LIMITED REGION 2 'blob' {the embedded blob }  
LAYER 'Can' K = 0.001  
START 'ring' (R,0)  
ARC(CENTER=0,0) ANGLE=360 TO CLOSE
```

Задание плоскости среза

Осуществляется добавлением спецификатора ON:

PLOTS

CONTOUR(Phi) ON $x=0$

Можно также запросить графики вычислительной сетки (и, следовательно, структуры домена) командой:

GRID(x,z) ON $y=0$

Задание плоскости среза

TITLE 'Heat flow around an Insulating blob'

COORDINATES cartesian3

VARIABLES

Phi { the temperature }

DEFINITIONS

R = 0.5 { blob radius }

EXTRUSION

SURFACE 'Bottom' z=0

layer "niz"

surface "gran1" z=1/4

layer "seredina"

surface "gran2" z=1/2

layer "verh"

SURFACE 'Top' z=1

BOUNDARIES

REGION 1 'box'

START{-1,-1}

LINE TO (1,-1) TO (1,1)

TO {-1,1} TO CLOSE

limited REGION 2 'blob' { the embedded blob }

layer "verh" void

START 'ring' (R,0)

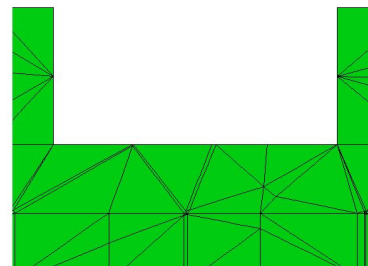
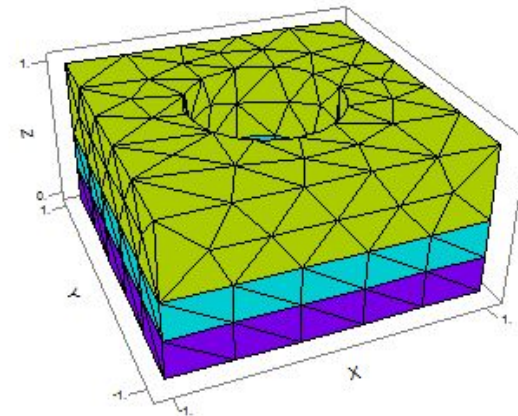
ARC(CENTER=0,0) ANGLE=360 TO CLOSE

PLOTS

grid [y,z] on x=0

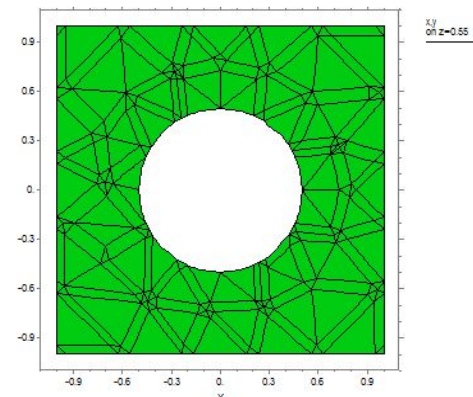
grid [x,y,z] paintregions as 'final mesh'

END



example1: Grid#1 P3 Nodes=251 Cells=877 RMS Err= 0.

grid (x,y) on z=0.55



Граничные условия

BOUNDARIES

REGION 1 'box'

START(-1,-1)

VALUE(Phi)=0 LINE TO (1,-1)

NATURAL(Phi)=0 LINE TO (1,1)

VALUE(Phi)=1 LINE TO (-1,1)

NATURAL(Phi)=0 LINE TO CLOSE

REGION 2 'blob' { the embedded blob }

SURFACE 'Can Bottom' VALUE(Phi)=Tcan

SURFACE 'Can Top' VALUE(Phi)=Tcan

{ parameter redefinition in the 'Can' layer only: }

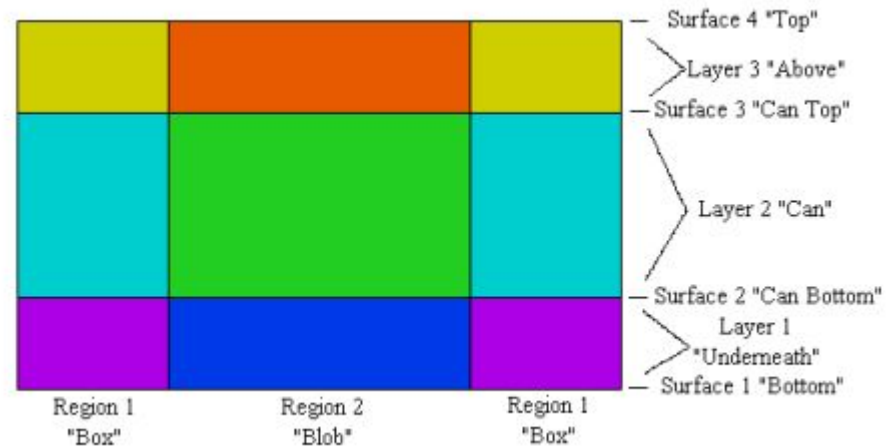
LAYER 2 k = 0.001

START 'ring' (R,0)

{ boundary condition in the 'Can' layer only: }

LAYER 'Can' VALUE(Phi)=Tcan

ARC(CENTER=0,0) ANGLE=360 TO CLOSE



Объемные поверхности и слои

- Фигуры должны поддерживать вытянутую форму, с боковыми стенками и слоями (боковые стенки не могут расти или сжиматься)
- Поверхности слоев должны быть непрерывными по границам областей. Если поверхность имеет вертикальный скачок, она должна быть разделена на слои.
- Поверхности слоев могут сливаться, но не инвертироваться. Используйте функцию MAX или MIN в определении поверхности, чтобы заблокировать инверсию.

TITLE 'Heat flow around an Insulating Sphere'

COORDINATES

Cartesian3

VARIABLES

Phi { the temperature }

DEFINITIONS

K = 1 { default conductivity }

R = 0.5 { sphere radius }

{ shape of hemispherical cap: }

Zsphere = sqrt(max(R²-x²-y²,0))

EQUATIONS

Div{-k*grad(phi)} = 0

EXTRUSION

SURFACE 'Bottom' z=-1

LAYER 'underneath'

SURFACE 'Sphere Bottom' z = -max(Zsphere,0)

LAYER 'Can'

SURFACE 'Sphere Top' z = max(Zsphere,0)

LAYER 'above'

SURFACE 'Top' z=1

BOUNDARIES

REGION 1 'box'

START{-1,-1}

VALUE(Phi)=0 **LINE TO** (1,-1)

NATURAL(Phi)=0 **LINE TO** (1,1)

VALUE(Phi)=1 **LINE TO** (-1,1)

NATURAL(Phi)=0 **LINE TO CLOSE**

LIMITED REGION 2 'blob' { the embedded blob }

LAYER 2 K = 0.001

START 'ring' (R,0)

ARC{**CENTER**=0,0} **ANGLE**=360

TO CLOSE

PLOTS

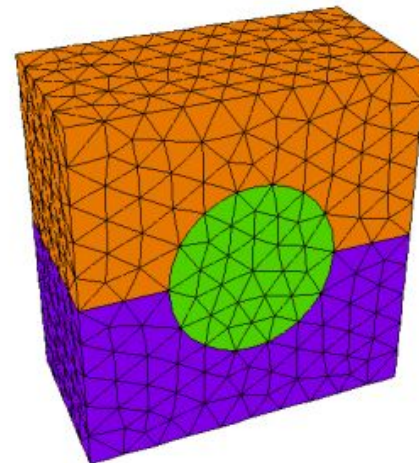
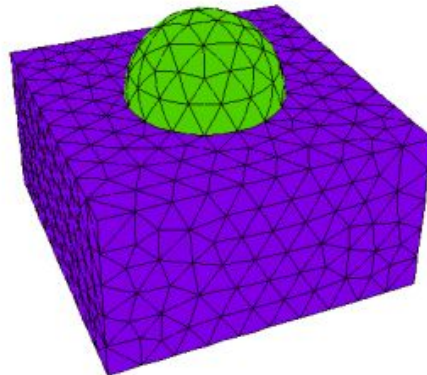
GRID{y,z} **on** x=0

CONTOUR(Phi) **on** x=0

VECTOR{-k*grad(Phi)} **on** x=0

ELEVATION(Phi) **FROM** (0,-1,0) **to** (0,1,0)

END



Функции поверхности

FlexPDE включает в себя три функции формирования поверхности (плоскость, цилиндр и сфера) для упрощения построения 3D-доменов:

PLANE (point1 , point2 , point3) - Определяет плоскую поверхность, содержащую три указанные точки.

CYLINDER (point1 , point2 , radius) - Определяет верхнюю поверхность цилиндра с осью вдоль линии от точки 1 до точки 2 и с заданным радиусом

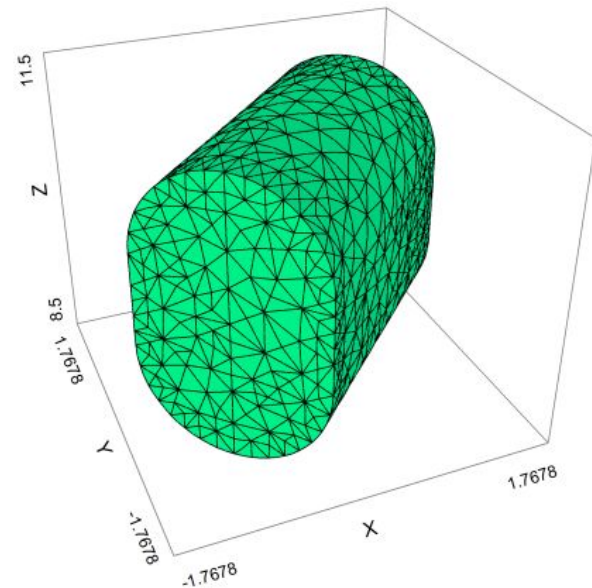
SPHERE (point , radius) - Определяет верхнюю поверхность сферы заданного радиуса с центром в указанной центральной точке

extrusion

surface $z = -\text{SPHERE} ((0,0,0),R0)$ { the bottom hemisphere }
surface $z = \text{SPHERE} ((0,0,0),R0)$ { the top hemisphere }

Функции поверхности: цилиндр

```
zs = CYLINDER((x0,y0,0.5), (x0+Len*c,y0+Len*s, 0.5), R0)
EXTRUSION
SURFACE z = zoff-zs { the bottom half-surface }
SURFACE z = zoff+zs { the top half-surface }
BOUNDARIES
REGION 1
START (x0,y0)
LINE TO (x0+R0*c,y0-R0*s)
TO (x0+Len*c+R0*c,y0+Len*s-R0*s)
TO (x0+Len*c-R0*c,y0+Len*s+R0*s)
TO (x0-R0*c,y0+R0*s)
TO CLOSE
```



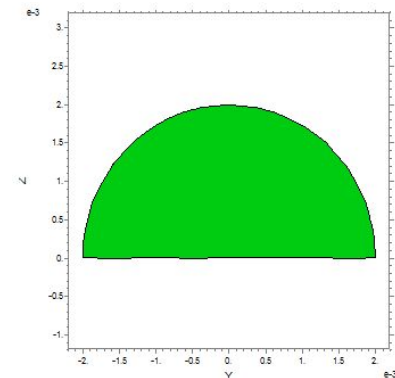
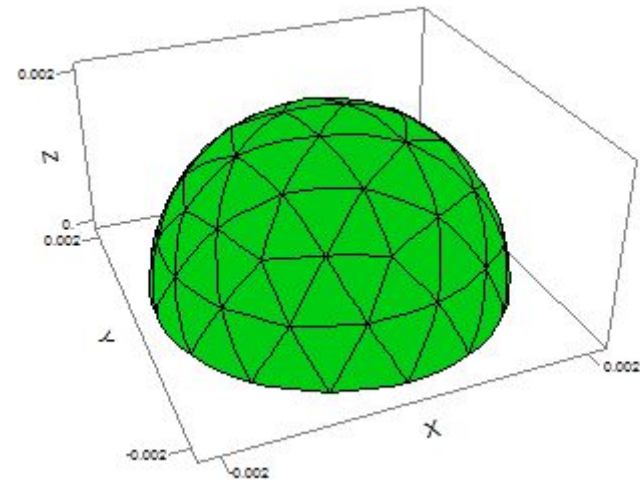
Полусфера

```
TITLE 'hemisphere'  
SELECT errlim=1e-3  
COORDINATES cartesian3  
VARIABLES U  
DEFINITIONS  
r0=2e-3  
rad=max(0,sqrt(x^2+y^2))  
EXTRUSION  
surface z=0  
surface z=sphere((0,0,0),r0)
```

```
BOUNDARIES  
region 1  
surface 2  
start (r0, 0) arc( center=0,0) angle=360
```

```
PLOTS  
grid(x,y,z) paintregions as "final mesh"  
grid(y,z) on x=0 nolines paintregions as "Region Map"
```

```
END
```



```
TITLE 'hemisphere'  
SELECT erlim=1e-3  
COORDINATES cartesian3  
VARIABLES U  
DEFINITIONS  
r0=2e-3  
rad=max(0,sqrt(x^2+y^2))  
EXTRUSION  
surface z=sphere((0,0,0),r0)  
surface z=h
```

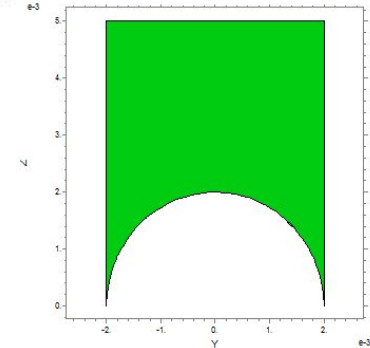
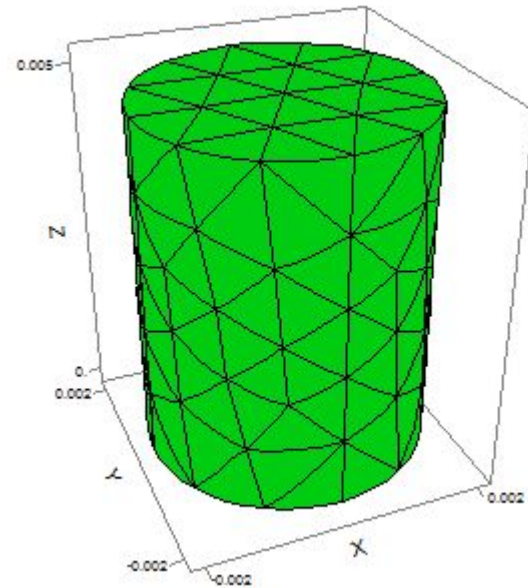
BOUNDARIES

```
region 1  
surface 2  
start (r0, 0) arc[ center=0,0] angle=360
```

PLOTS

```
grid(x,y,z) paintregions as 'final mesh'  
grid(y,z) on x=0 nolines paintregions as 'Region Map'
```

```
END
```

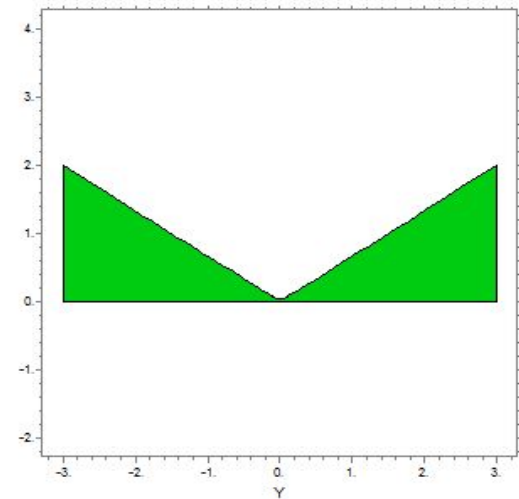
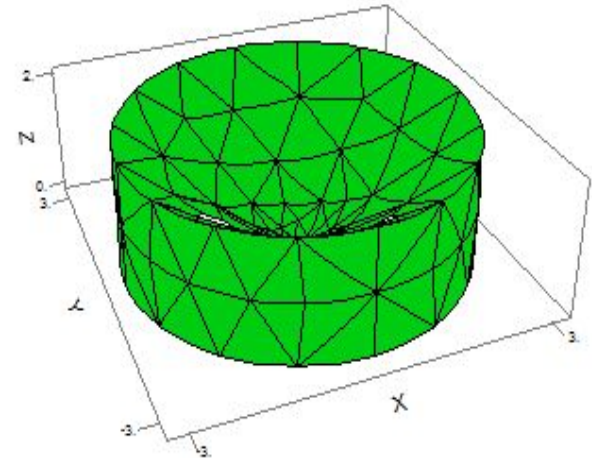


Конус

```
TITLE 'Cone'  
COORDINATES cartesian3  
VARIABLES U  
DEFINITIONS  
r0=2 r=3 h=2  
rad=max(0,sqrt(x^2+y^2))  
EXTRUSION  
surface z=0  
surface z=h*(rad)/(r)
```

```
BOUNDARIES  
region 1  
start (r, 0) arc( center=0,0) angle=360
```

```
PLOTS  
grid(x,y,z) paintregions as 'final mesh'  
grid(y,z) on x=0 nolines paintregions as 'Region Map'  
END
```

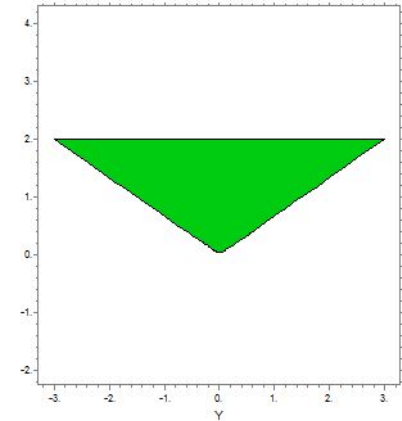
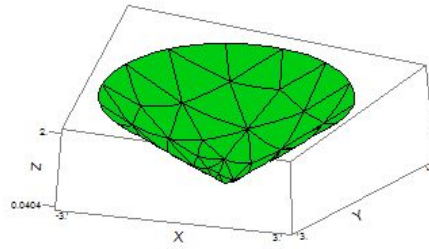


Конус

EXTRUSION

surface $z=h*(rad)/(r)$

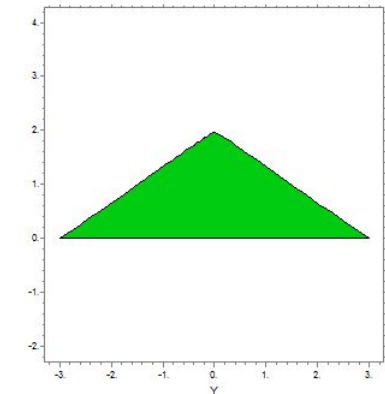
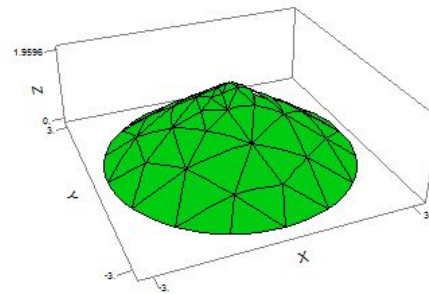
surface $z=h$



EXTRUSION

surface $z=0$

surface $z=h-h*(rad)/(r)$



COORDINATES cartesian3

DEFINITIONS

```
rad=sqrt(x^2+y^2)
router = 0.3    { outer radius of glass }
zglass = 0.5    { glass height }
rbase = 0.2    { radius of the base }
zbase = 0.02   { thickness of the base and cone }
rstem = 0.02   { radius of the stem }
zstem = 0.3    { height of the stem }
zslope = (zglass-zstem)/(router-rstem){ slope of conic surface }
glassangle = arctan(zslope)           { slope of conic surface }
zcone = max(0,(rad-rstem)*zslope)     { conic surface of the glass }
```

EXTRUSION

```
surface 'bottom' z=0
layer 'base layer'
surface 'stem1' z=zbase
layer 'stem layer'
surface 'lower' z = zstem + zcone
layer 'cone layer'
surface 'upper' z = zbase*cos(glassangle) + min(zglass, zstem + zcone)
```

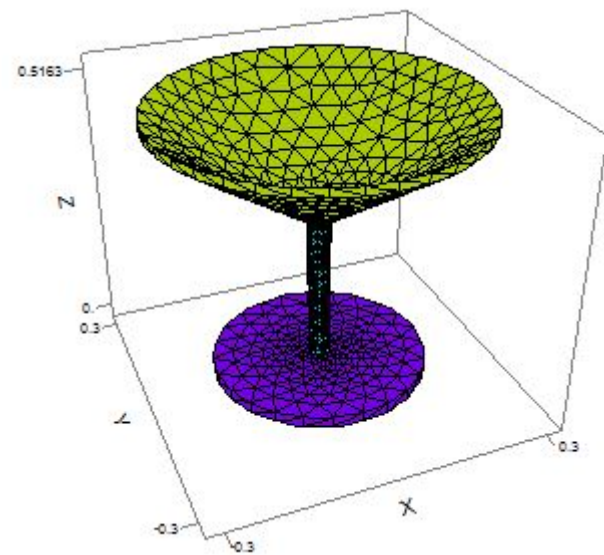
BOUNDARIES

```
limited region 'outer'
layer 'cone layer' { outer region exists only in cone }
start (router,0) arc(center=0,0) angle=360
limited region 'base'
layer 'base layer' { base region exists only in base }
start(rbase,0) arc(center=0,0) angle=360
limited region 'stem'
layer 'stem layer' { stem region exists in the stem and the bottom of the cone }
layer 'cone layer'
start(rstem,0) arc(center=0,0) angle=360
```

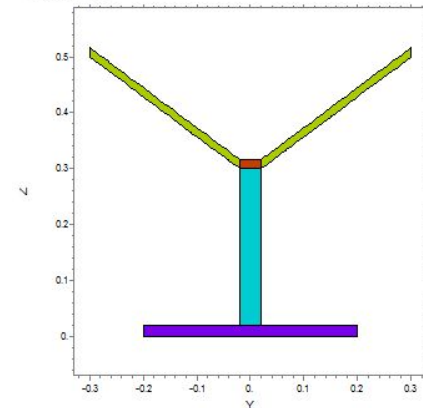
PLOTS

```
grid(x,y,z) paintregions as "final mesh"
grid(y,z) on x=0 nolines paintregions as "Region Map"
```

END



Cocktail Glass



Интегральные операторы

Операторы INTEGRAL и VOL_INTEGRAL являются синонимами и производят интегрирование по **объему**.

Формат:

INTEGRAL(<подынтегральное выражение:», <имя_области>)

или

VOL_INTEGRAL(подынтегральное выражением
<имя_области>)

Пример: power=**vol_integral**(Jm*Em)

Поверхностные интегралы

Операторы `SINTEGRAL` и `SURF_INTEGRAL` являются синонимами и выполняют интегрирование по поверхности. Формат:

SINTEGRAL (<integrand>, <named_boundary>)

или

SURF_INTEGRAL(<integrand>,<named_boundary>)

Пример

```
I1= abs(sintegral(normal(D), 'electrode1'))
```

Линейные интегралы

Операторы `VINTEGRAL` и `LINE_INTEGRAL` являются синонимами и выполняют интегрирование по линии.

Форма записи:

VINTEGRAL(<integrand>, <named_boundary>)

или

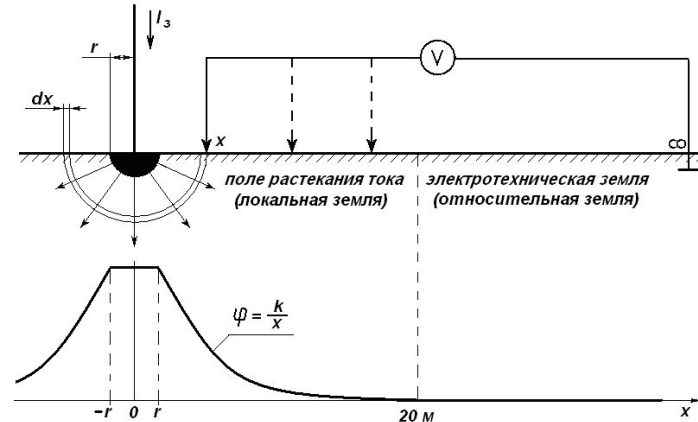
LINE_INTEGRAL(<integrand>, <named_boundary>)

Задача растекания токов

Постановка задачи растекания токов: Задача формулируется в виде уравнения Пуассона относительно скалярного электрического потенциала (предполагается, что вектор плотности тока лежит в плоскости модели). Для плоскопараллельных задач уравнение имеет вид

для осесимметричного случая:

Вектор плотности тока



Вычисляемые физические величины в задачах растекания токов

Локальные величины:

· Скалярный электрический потенциал U ;

· Вектор напряженности электрического поля $\vec{E} = -\text{grad}(U)$, где

$$E_x = -\frac{\partial U}{\partial x}, \quad E_y = -\frac{\partial U}{\partial y}$$

– в плоском случае,

$$E_z = -\frac{\partial U}{\partial z}, \quad E_r = -\frac{\partial U}{\partial r}$$

– в осесимметричном случае;

· Вектор плотности тока:

$$j = \sigma E.$$

Интегральные величины:

· Электрический ток через заданную поверхность:

$$I = \oint_s (\vec{j} \cdot \vec{n}) ds,$$

где n - единичный вектор нормали к поверхности. Поверхность s может быть замкнутой или разомкнутой;

· Мощность тепловыделения в заданном объеме:

$$W = \int (\vec{E} \cdot \vec{j}) dV.$$

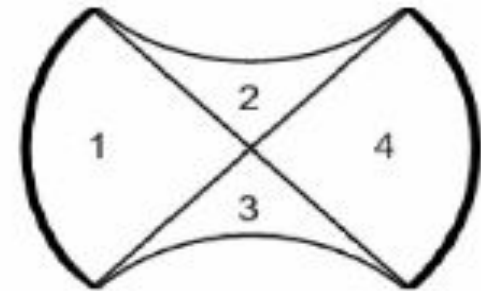
Величина	Формула	Запись в тексте программы
Системная переменная	U	<code>variables U</code>
Напряженность электрического поля E	$\vec{E} = -grad(U)$	<code>E = -grad(U)</code>
Плотность тока (изотропный материал)	$\vec{J} = \sigma \cdot \vec{E}$	<code>J = sigma*E</code>
Плотность тока (анизотропный материал)	$\vec{J} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix} \cdot \vec{E}$	<code>Jx = dot(vector(sigma_xx, sigma_xy), E)</code> <code>Jy = dot(vector(sigma_xy, sigma_yy), E)</code> <code>J = vector(Jx, Jy)</code>
Поверхностная плотность тока на электроде	$j = \vec{J} \cdot \vec{n}$	<code>-normal(J)</code> (выражение используется только в контексте границы области)
Линейная плотность тока на электроде (плоская конструкция)	$j = h \cdot \vec{J} \cdot \vec{n}$	<code>-normal(J)*h</code> (выражение используется только в контексте границы области)
Ток электрода (контур описания области электрода имеет имя «electrode»)	$I = \int_S j dS = \int_S (\vec{J} \cdot \vec{n}) dS$	<code>I = sintegral(-normal(J), 'electrode')</code> <code>I = sintegral(-normal(J)*h, 'electrode')</code>

Величина	Формула	Запись в тексте программы
Мощность тепло-выделения	$Pe = \int_V (\vec{J} \cdot \vec{E}) dV$	<code>Pe = integral(dot(J, E)),</code> <code>Pe = integral(h*dot(J, E))</code>
Проводимость конструкции, выраженная через ток электрода при единичной разности потенциалов	$Y = I = \left \int_S (\vec{J} \cdot \vec{n}) dS \right $	<code>Y = abs(sintegral(normal(J),</code> <code>'electrode')),</code> <code>Y = abs(sintegral(h*normal(J), 'electrode'))</code>
Проводимость конструкции, выраженная через мощность тепловыделения при единичной разности потенциалов	$Y = Pe _{\Delta V=1}$	<code>Y = integral(dot(J, E)),</code> <code>Y = integral(h*dot(J, E))</code>
Максимальная величина напряженности электрического поля или плотности тока		<code>globalmax(magnitude(E))</code> <code>globalmax(magnitude(J))</code>

Граничные условия

<i>Тип границы</i>	<i>Выражение для граничного условия</i>
Проводник	$\text{value}(U) = \text{const}$
Плоскость симметрии	$\text{natural}(U) = 0$
Плоскость антисимметрии	$\text{value}(U) = 0$
Ось симметрии	$\text{natural}(U) = 0$
Точка, удаленная на большое расстояние от источников электрического поля	$\text{value}(U) = 0$
Граница резистивного материала	$\text{natural}(U) = 0$

Пример расчета сопротивления пленочного резистора



TITLE 'Resistor'

VARIABLES U

definitions

$R = 5e-3$! радиус дуг

$\rho_1 = 200$ $\rho_2 = 800$! значения пленочных сопротивлений

$\rho_3 = 50$ $\rho_4 = 500$

ρ ! параметр расчета, задаваемый в каждой расчетной области

$a = R/\sqrt{2}$ $b = R*(\sqrt{2}-1)$! координаты опорных точек

$E = -\text{grad}(U)$

$J = E/\rho$

$P = \text{integral}(\text{dot}(E, J))$! мощность тепловыделения

$I_1 = \text{abs}(\text{sintegral}(\text{normal}(J), \text{'electrode1'}))$! токи электродов

$I_2 = \text{abs}(\text{sintegral}(\text{normal}(J), \text{'electrode2'}))$

EQUATIONS

$\text{div}(J) = 0$

Пример расчета сопротивления пленочного резистора

EQUATIONS

$\text{div}(J) = 0$

BOUNDARIES

REGION 1 rho = rho1

START(a, -a) value(U) = 1 arc to(R, 0) to(a, a)

natural(U) = 0 arc to(0, b) to(-a, a)

value(U) = 0 arc to(-R, 0) to(-a, -a)

natural(U) = 0 arc to(0, -b) to close

region 2 rho = rho2

start(0, 0) line to(a, a) arc to(0, b) to(-a, a)

line to close

region 3 rho = rho3

start(-a, -a) arc to(0, -b) to(a, -a)

line to(0, 0) to close

region 4 rho = rho4

start(0, 0) line to(a, -a) arc to(R, 0) to(a, a)

line to close

feature 5

start 'electrode1' (a, -a) arc to(R, 0) to(a, a)

feature 6

start 'electrode2' (-a, a) arc to(-R, 0) to (-a, -a)

PLOTS

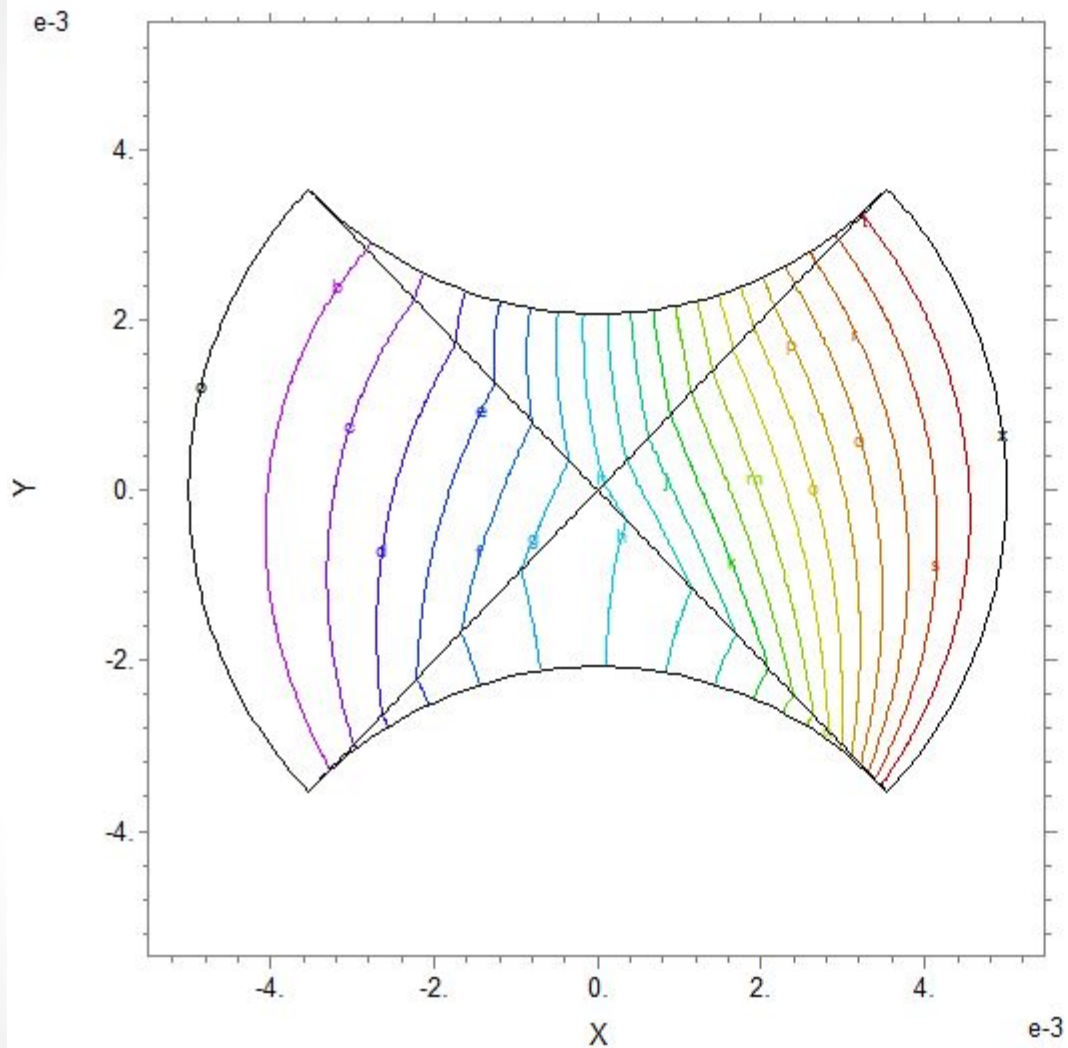
CONTOUR(u) report(1/P) report(1/I1) report(1/I2)

vector(J) norm

END

Resistor

04:31:45 10/18/19
FlexPDE Lite 7.15/W64



u	
max	1.00
u :	1.00
t :	0.95
s :	0.90
r :	0.85
q :	0.80
p :	0.75
o :	0.70
n :	0.65
m :	0.60
l :	0.55
k :	0.50
j :	0.45
i :	0.40
h :	0.35
g :	0.30
f :	0.25
e :	0.20
d :	0.15
c :	0.10
b :	0.05
a :	0.00
min	0.00

primer: Grid#1 P3 Nodes=86 Cells=138 RMS Err= 2.3e-4
1/P= 443.8259 1/I1= 439.6402 1/I2= 443.7484 Integral= 2.103560e-5