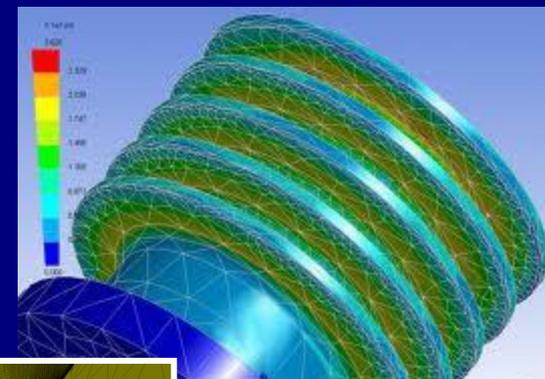


I.I. Vorovich
Institute
Mathematics
Mechanics
Computer Science

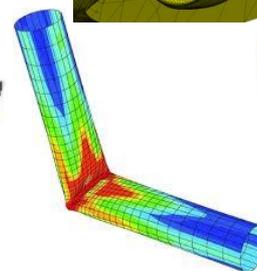
Работа в программе FlexPDE



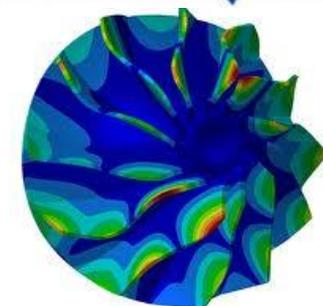
ANSYS



Dr. Anna A. Nasedkina
Southern Federal University



ABAQUS



Содержание

- Описание программы FlexPDE
- Синтаксис FlexPDE, входной файл (скрипт)
- Граничные условия
- Основные этапы решения задачи в FlexPDE
- Переменные и уравнения
- Построение геометрии: области (regions), операторы построения границы: line, spline, arc
- Генерация конечно-элементной сетки
- Задание материальных свойств
- Графический вывод результатов решения

Краткое описание FlexPDE

- FlexPDE – программа для конечно-элементного моделирования и численного решения задач математической физики. На основе входного файла (скрипта), написанного пользователем, FlexPDE выполняет необходимые операции по созданию конечно-элементной модели начально-краевой задачи, решению системы и графическому представлению результатов решения.
- FlexPDE – полноценная среда для решения систем уравнений в частных производных, содержащая графический редактор для создания скрипта, генератор конечно-элементного разбиения области, конечно-элементный решатель для нахождения решения и графическую среду для вывода результатов
- В FlexPDE нет изначально заданных геометрических областей или уравнений. Пользователь сам записывает уравнение и задает начальные и граничные условия.

Какие задачи может решать FlexPDE?

- FlexPDE может решать системы уравнений в частных производных первого и второго порядка в декартовой системе координат или осесимметричной постановке для двумерного моделирования или в декартовой системе координат для 3D
- Задачи могут быть стационарными или нестационарными, также FlexPDE умеет решать задачи на собственные значения.
- Можно задать любое число уравнений в системе, которые будут решаться одновременно.
- Можно решать линейные или нелинейные уравнения .
Нелинейные системы FlexPDE решает с помощью модифицированной схемы Ньютона-Рафсона.
- Можно задать любое число областей с различными материальными свойствами. Предполагается, что решение непрерывно на границе раздела материальных свойств.

Кто может использовать FlexPDE?

- Исследователи могут применять FlexPDE для моделирования своих экспериментов, прогнозирования или верификации результатов.
- Инженеры могут использовать FlexPDE для оптимизации конструкций, технического обоснования и концептуального анализа. Одна и та же программа может применяться для моделирования всех аспектов дизайна.
- Преподавателям FlexPDE может быть полезна для обучения инженерному делу, решению задач математической физике. В программе студенты видят математическую постановку задачи и уравнения, и могут экспериментировать с изменением уравнений и расчетных областей.

Структура скрипта FlexPDE, основные разделы

- TITLE – название решаемой задачи
- SELECT – описание опций для решения (например, задаваемой точности)
- VARIABLES – описание переменных (неизвестных уравнения).
- DEFINITIONS – задание параметров, вспомогательных соотношений и функций.
- EQUATIONS – описание уравнений в частных производных (каждая переменная должна быть ассоциирована с уравнением)
- INITIALVALUES – начальные условия для нелинейных и нестационарных (динамических) задач.
- BOUNDARIES – геометрия задачи описывается в порядке обхода по периметру области, состоит из отдельных сегментов, составляющих границу области. Одновременно с построением границы задаются граничные условия
- PLOTS – содержит команды для графического вывода решения: CONTOUR, SURFACE, ELEVATION, VECTOR.

Обозначения

□ Комментарии помещаются в фигурные скобки
{ Anything inside curly brackets is a comment. }

Любые символы после ! игнорируются

! Комментарий

□ Производные первого порядка, например, du/dx , записываются как $dx(u)$, вторые производные – как $dxx(u)$. Распознаются названия координат и операторы дивергенции Div, градиента Grad и ротора Curl.

□ FlexPDE не является регистрозависимой системой. "F" и "f" – одно и то же.

Пример входного файла (скрипта)

```
TITLE 'Simple diffusion equation'  
{ this problem lacks sources and boundary conditions }  
VARIABLES  
phi  
DEFINITIONS  
k=3 { conductivity }  
EQUATIONS  
div(-k*grad(phi)) =0  
BOUNDARIES  
REGION 1  
START(0,0)  
LINE TO (1,0)  
TO (1,1)  
TO (0,1)  
TO CLOSE  
PLOTS  
CONTOUR(phi)  
VECTOR(-k*grad(phi))  
END
```

Основные этапы решения задачи в FlexPDE

- Определение неизвестных и описание уравнений
- Построение области (геометрии задачи)
- Задание материальных свойств
- Задание граничных условий
- Графический вывод результатов

Уравнения и неизвестные

- Каковы неизвестные переменные для анализа?
- Какие дифференциальные уравнения их описывают?

Разделы **VARIABLES** и **EQUATIONS** содержат информацию о неизвестных и уравнениях. Эти разделы связаны друг с другом, так как для каждой неизвестной должно быть описано уравнение, определяющее ее.

VARIABLES

Phi

EQUATIONS

$$\text{Div}(\text{grad}(\text{Phi})) = 0$$

VARIABLES

A,B

EQUATIONS

$$A: \text{Div}(\text{grad}(A)) = 0$$

$$B: \text{Div}(\text{grad}(B)) = 0$$

Построение геометрии: описание 2D области

- Расчетная область для двумерного случая описывается в разделе **BOUNDARIES**.
- В свою очередь, область может содержать подобласти (регионы) **REGION**. Описание каждого региона начинается с ключевого слова **REGION** <номер> (или **REGION** "название"). При этом все последующие замкнутые области входят в описанный регион.
- При этом первый **REGION** должен содержать всю расчетную область. Это требование не является обязательным, но значительно облегчает постановку граничных условий.
- Формы областей описываются с помощью обхода областей по периметру и заданием сегментов **LINE**, **SPLINE** или **ARC**, следующих друг за другом. При этом предполагается, что каждый сегмент начинается в точке, где закончился предыдущий сегмент. Каждый сегмент начинается с ключевого слова **START** и задания координат начальной точки, а заканчивается ключевым словом **CLOSE** (или **TO CLOSE**).

Построение пути вдоль границы области (Boundary paths)

- Каждое описание замкнутой границы начинается со слова **START** и имеет синтаксис вида:
START(a,b) segment TO (c,d) ...
где **(a,b)** и **(c,d)** – координаты концов сегмента, где **segment** – один из трех возможных вариантов: **LINE**, **SPLINE** или **ARC**.
- Путь вдоль границы состоит из сегментов, которые связывают между собой ключевые точки границы. Конечная точка каждого сегмента становится начальной точкой следующего за ним сегмента. Построение пути заканчивается, если очередной сегмент нельзя построить или когда новый сегмент ведет в начальную точку пути
- **... segment TO CLOSE.**
или
... segment CLOSE.

Пути вдоль границы: сегменты line и spline

- Сегменты прямых линий **LINE** имеют вид

LINE TO (x,y)

Когда описывается несколько следующих друг за другом сегментов **LINE**, зарезервированное слово **LINE** может быть опущено для краткости, например:

LINE TO (x1,y1) TO (x2,y2) TO (x3,y3) TO ...

- Сегменты сплайнов **SPLINE** синтаксически аналогичны сегментам линий:

SPLINE TO (x,y) TO (x2,y2) TO (x3,y3) TO ...

Пути вдоль границы: сегменты arc

- Сегмент дуги **ARC** может задавать дугу окружности или эллипса. Для сегмента **ARC** возможны четыре варианта синтаксиса (дуга начинается из конечной точки предыдущего сегмента):
 - ARC TO (x1,y1) to (x2,y2)** {задается конечная точки дуги}
 - ARC (RADIUS = R) to (x,y)** {задается радиус и конечная точки дуги}
 - ARC (CENTER = x1,y1) to (x2,y2)** {задаются координаты центра окружности (эллипса) и конечная точки дуги}
 - ARC (CENTER = x1,y1) ANGLE=angle** {задаются координаты центра окружности (эллипса) и угол}
- Здесь значение угла **angle** задается в градусах (положительное значение – вращение против часовой стрелки, отрицательное значение – вращение по часовой стрелке)
- С помощью синтаксиса **ARC (CENTER=x1,y1) to (x2,y2)** в случае, когда расстояния от центра **(x1,y1)** до начальной и конечной точки дуги различны, создается дуга эллипса с полуосями, параллельными координатным осям.

Некоторые примеры путей вдоль границы

- Прямоугольник
START(x1,y1)
LINE TO(x2,y1)
TO (x2,y2)
TO (x1,y2)
TO CLOSE
- Окружность
START(r,0)
ARC(CENTER=0,0) ANGLE=360
- Дуга окружности 90 градусов
START(r,0)
ARC(CENTER=0,0) TO (0,r)
- Дуга эллипса 90 градусов
START(x0+a,y0)
ARC(CENTER=x0,y0) TO (x0,y0+b)

Использование регионов для создания областей

- Разные регионы REGION : можно использовать для задания областей из разных материалов, моделирования включений, например: область из материала 2 внутри области из материала 1
BOUNDARIES
REGION 1 {Material 1 with its set of material properties}
START (X1,Y1) <closed loops around the domain> TO CLOSE
REGION 2 {Material 2 with its set of material properties, can be located inside material 1}
START (X2,Y2) <closed loops around overlays of the second material>
TO CLOSE
- Один регион REGION : если регион содержит несколько стартовых точек, т.е. содержит несколько замкнутых границ, то результирующая область будет состоять из объединенных областей за вычетом их пересечений, можно использовать для моделирования отверстий
- BOUNDARIES
REGION 1 {Material 1 with its set of material properties}
START (X1,Y1) <closed loops around the domain> TO CLOSE
START (X2,Y2) <closed loops around overlays of the second material>
TO CLOSE

Пример построения области: включение

DEFINITIONS

$R=1/2$ {ring radius}

BOUNDARIES

REGION 1 'box' { the bounding box }

START(-1,-1)

LINE TO(1,-1)

TO (1,1)

TO (-1,1)

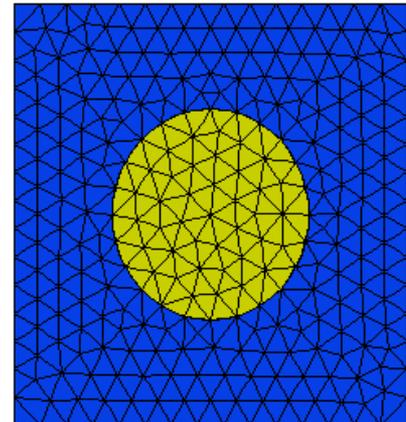
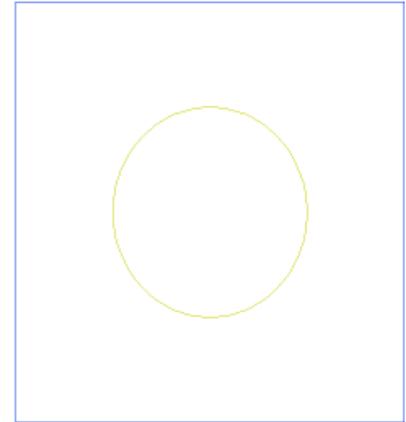
TO CLOSE

REGION 2 'blob' { the embedded circular 'blob' }

START 'ring' (R,0)

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

❑ Разные регионы показаны разными цветами



Пример построения области: отверстие

DEFINITIONS

R=1/2 {ring radius}

BOUNDARIES

REGION 1 'box' { the bounding box }

START(-1,-1)

LINE TO(1,-1)

TO (1,1)

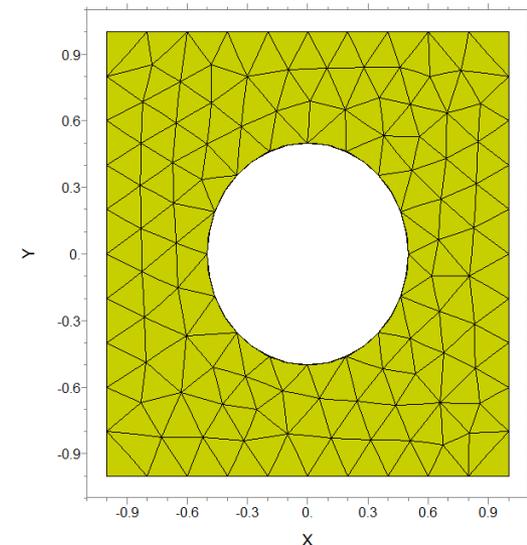
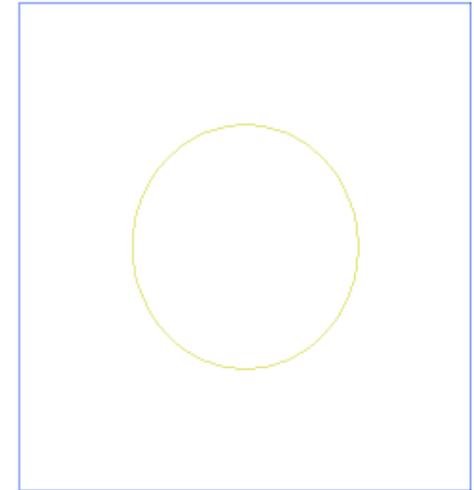
TO (-1,1)

TO CLOSE

{ circular hole' }

START 'ring' (R,0)

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



Создание конечно-элементной сетки

- При нажатии кнопки "Run Script" из верхнего меню или выбора Controls->Run Script, FlexPDE автоматически создает конечно-элементное разбиение созданной области. При этом размеры конечных элементов будут выбираться на основе длин сегментов, кривизны дуг, а также заданных пользовательских настроек разбиения.
- Для задания пользовательских настроек автоматического разбиения используются параметры MESH_DENSITY и MESH_SPACING
- Однако, в большинстве случаев, изменять автоматическое разбиение не требуется, так как FlexPDE адаптивно сгущает конечно-элементную сетку в местах с большой кривизной и при обнаружении особенных точек.

Пользовательские настройки для конечно-элементной сетки: параметр MESH_DENSITY

- MESH_DENSITY: задает минимальное число ячеек на единицу длины

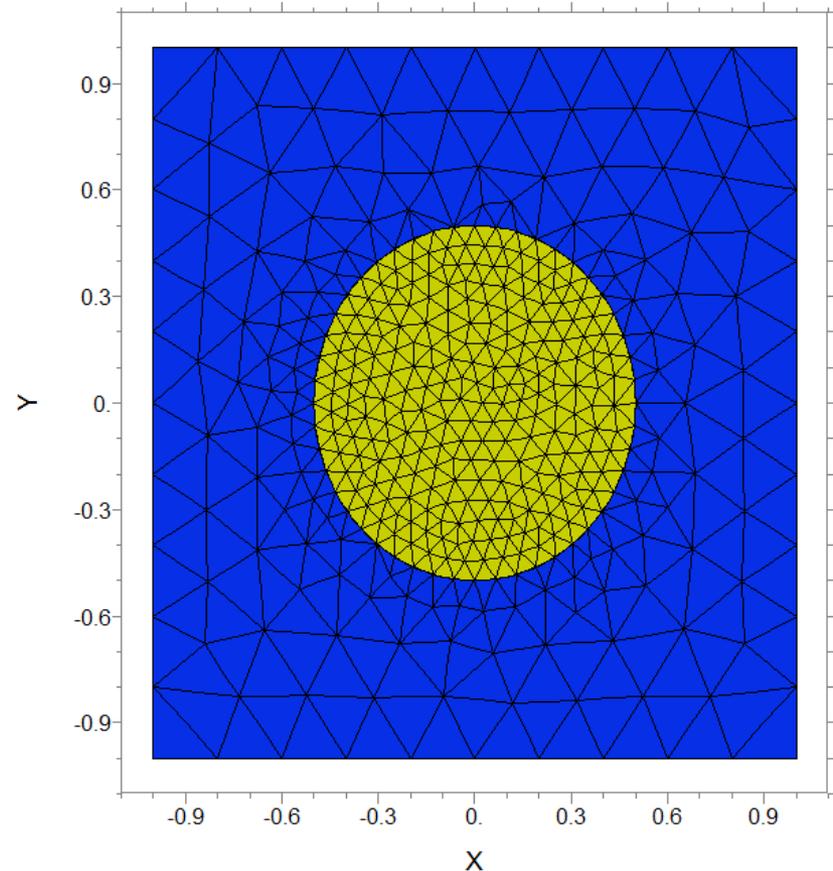
REGION 2 'blob'

MESH_DENSITY = 15

START(1/2,0)

ARC(CENTER=0,0) ANGLE=360

TO CLOSE



Пользовательские настройки для конечно-элементной сетки: параметр MESH_SPACING

- MESH_SPACING: задает максимальный размер ячейки

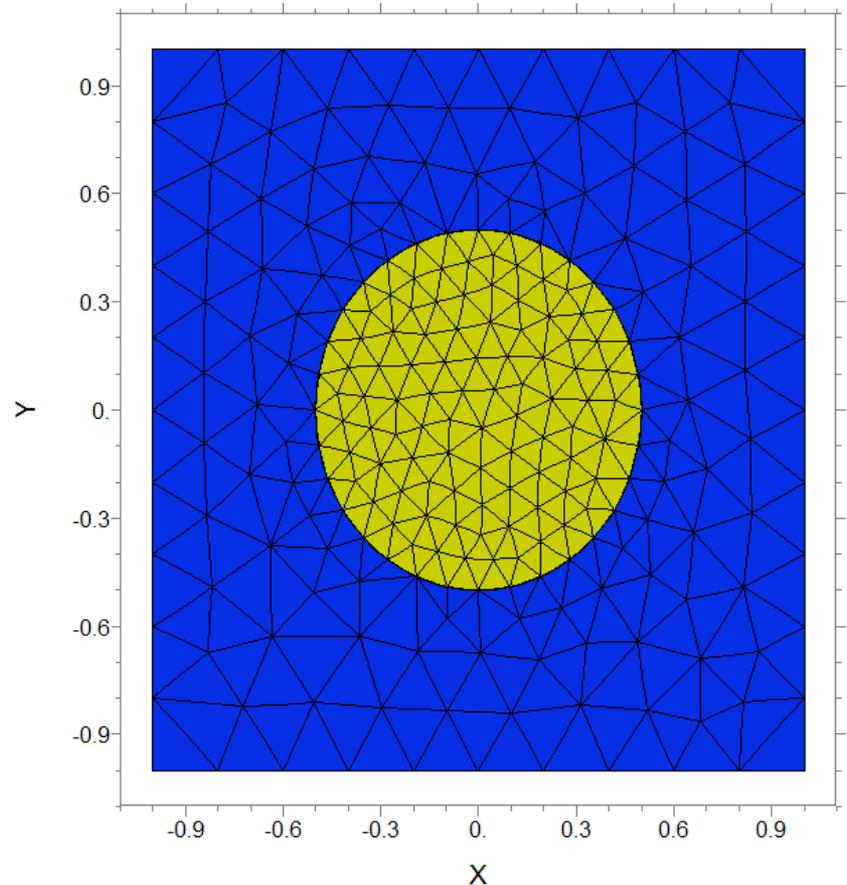
```
REGION 2 'blob'
```

```
MESH_SPACING = 0.1
```

```
START(1/2,0 )
```

```
ARC(CENTER=0,0) ANGLE=360
```

```
TO CLOSE
```



Задание материальных свойств

- Задание материальных свойств в FlexPDE происходит в два этапа. Сначала материальные свойства описываются в разделе **DEFINITIONS**, где им можно задать значения по умолчанию. Затем конкретные значения параметров материальных свойств задаются для отдельных регионов области **REGIONS**.

DEFINITIONS

k = 3

- Значение по умолчанию для параметра "k", заданное в разделе DEFINITIONS, будет использовано для каждого региона REGION области, пока не будет задано другое значение, например:

REGION 2 'blob' { the embedded blob }

k = 0.001

START(1/2,0)

ARC(CENTER=0,0) ANGLE=360

Задание граничных условий

- Каждое граничное условие задается оператором, который в качестве аргумента принимает имя неизвестной из решаемой системы уравнений. Имя неизвестной связывает граничное условие с одним из уравнений, так как в реальности каждое граничное условие модифицирует одно из уравнений.
- **Основные типы граничных условий:**
- Главные граничные условия (типа Дирихле) описываются оператором VALUE, который задает значение для переменной на границе. Значение можно задать в виде арифметического выражения или функций от других переменных.
- Естественные граничные условия (типа Неймана или нормальной производной для уравнения Лапласа) описываются оператором NATURAL (или LOAD) , который задает поток на границе области. Для этого можно использовать любые арифметические выражения или функций от других переменных. Конкретный смысл граничного условия NATURAL зависит от дифференциального уравнения, для которого задается это условие.

Задание граничных условий: примеры

- Каждое граничное условие действует на определенном сегменте границы и всех последующих сегментах, пока не будет задано другое граничное условие
- Замечание: после каждого изменения граничного условия оператор сегмента надо задавать заново

```
REGION 1 'box' { the bounding box }  
START(-1,-1)  
{ Phi=0 on base line: }  
VALUE(PHI)=0 LINE TO(1,-1)  
{ normal derivative =0 on right side: }  
NATURAL(PHI)=0 LINE TO (1,1)  
{ Phi = 1 on top: }  
VALUE(PHI)=1 LINE TO (-1,1)  
{ normal derivative =0 on left side: }  
NATURAL(PHI)=0 LINE TO CLOSE
```

Задание граничных условий в задаче теории упругости

variables

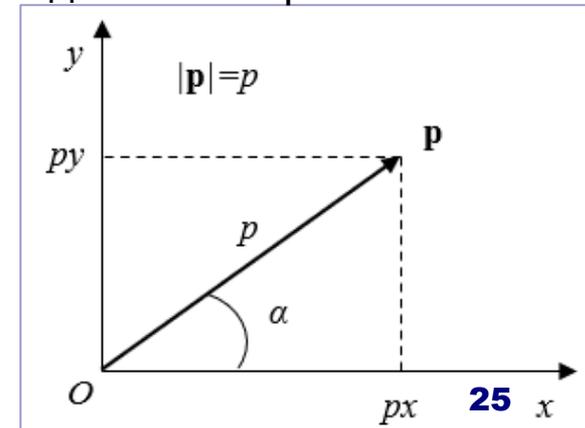
U, V {displacements $U=UX$ and $V=UY$ are the variables}

equations {differential equilibrium equations for a plane elastic problem}

$$U: dx(T_{xx}) + dy(T_{xy}) = 0$$

$$V: dx(T_{xy}) + dy(T_{yy}) = 0$$

- Жестко закрепленная граница:
value(U)=0
value(V)=0
- Симметрия относительно оси OX :
load(U)=0
value(V)=0
- Симметрия относительно оси OY :
value(U)=0
load(V)=0
- Давление на сегменте границы (задать компоненты вектора нагрузки $p = \{p_x, p_y\} = \{p \cdot \cos \alpha, p \cdot \sin \alpha\}$ – проекции на координатные оси)
load(U)= p_x
load(V)= p_y
- Граница, свободная от напряжений
load(U)=0
load(V)=0



Графический вывод результатов

- Секции **MONITORS** и **PLOTS** содержат команды графического вывода. Результаты работы всех графических команд из **PLOTS** записываются в файл **.PG6** (с номером версии FlexPDE)
 - Секция **MONITORS** содержит команды для вывода промежуточных результатов в процессе решения.
 - Секция **PLOTS** содержит команды для итогового вывода графических результатов, которые можно экспортировать в графические файлы.
 - Секция **SUMMARY** создает страницу вывода результатов в текстовом виде, полученных с помощью операторов **REPORT**
- **Команды вывода результатов:**
 - CONTOUR** – картина распределения заданной скалярной переменной; может быть с градиентной заливкой (опция **PAINTED**) или изображением изолиний
 - SURFACE** – трехмерная поверхность для распределения заданной скалярной переменной
 - VECTOR** – векторное поле (распределение векторной переменной)
 - ELEVATION** – график вдоль пути
 - REPORT** – текстовый отчет

Примеры команд вывода результатов

■ Секции PLOTS и SUMMARY

PLOTS

CONTOUR(Phi) {isolines only}

CONTOUR(Phi) painted {color filling}

VECTOR(-k*grad(Phi))

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

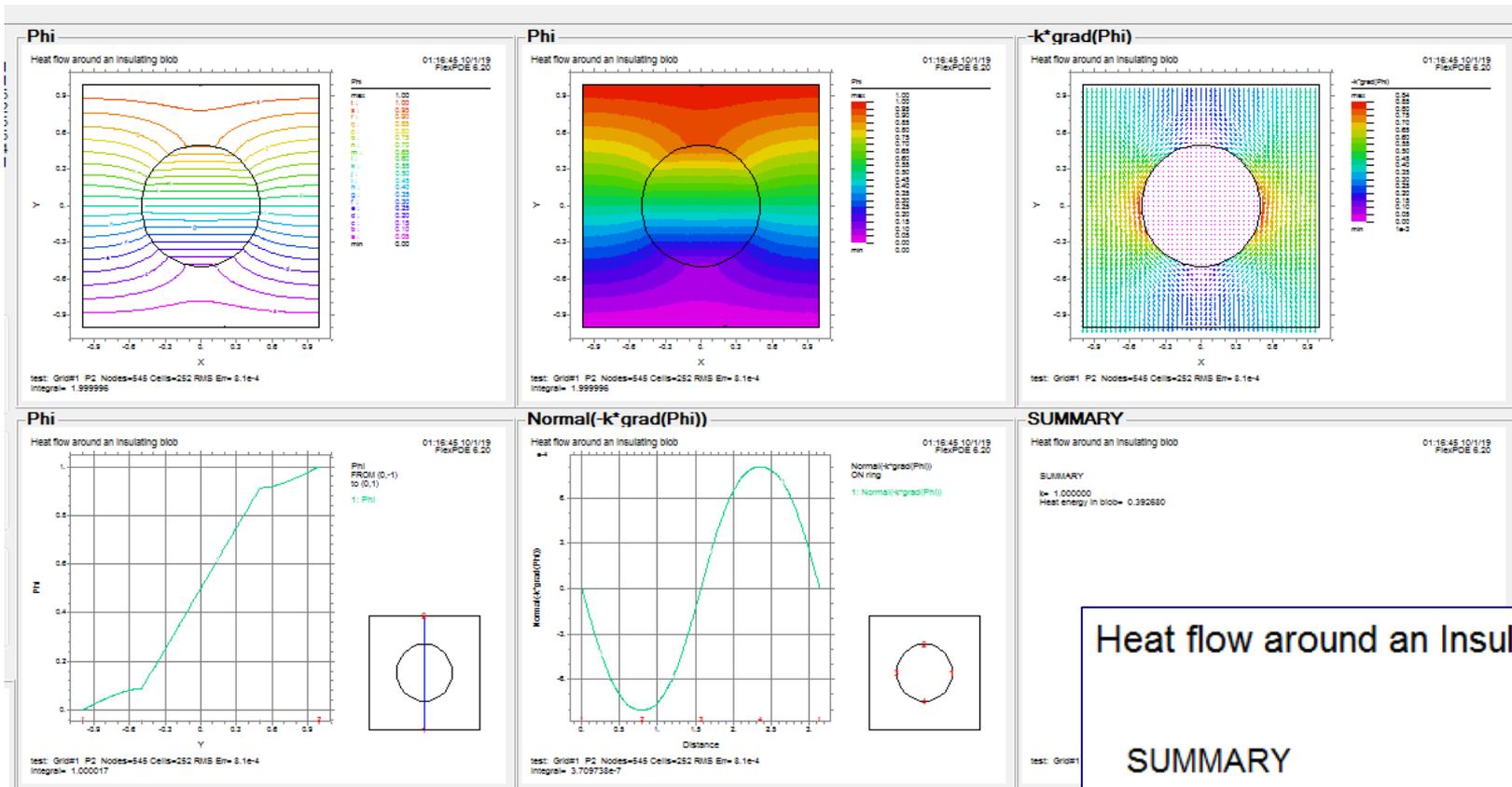
ELEVATION(Normal(-k*grad(Phi))) ON 'ring'

SUMMARY

REPORT(k)

REPORT(INTEGRAL(Phi,'blob')) as "Heat energy in blob"

Графические результаты команд разделов PLOTS и SUMMARY



Heat flow around an Insulating blob

SUMMARY

$k = 1.000000$
Heat energy in blob = 0.392680