

Министерство образования и науки Российской Федерации Государственное

образовательное учреждение высшего профессионального образования

«Московский государственный технический университет

имени Н. Э. Баумана»

**Лабораторная работа по дисциплине**

**«Модели и методы анализа проектных решений»**

**на тему:**

**«Метод конечных разностей»**

Вариант №42

Выполнил: Буличев О.В.

Группа: РК6-63

Преподаватель: Трудоношин В.А.

Проверил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2015

Условие задачи.

С помощью неявной разностной схемы решить нестационарное уравнение теплопроводности для прямоугольной пластины размером 8\*3см. Начальное значение температуры пластины - 10 градусов.

Граничные условия следующие: верхняя и верхняя половина правой границы теплоизолированы, на остальной части границы температура 400 градусов.

При выводе результатов показать динамику изменения температуры (например, с помощью цветовой гаммы).

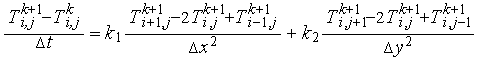
Отчет должен содержать: текст программы, рисунок объекта с распределением температуры в момент времени 25 сек сравнение результатов расчета с результатами, полученными с помощью пакета ANSYS.

Решение

Неявная разностная схема

1. Метод решения

Неявная разностная схема



Концепция программы

Визуализацию было решено делать в гнуплоте. Программа разделяется на 3 блока : Гаусс(прямой и обратный ходы), генерация матрицы и свободных членов, и печать в файл неизвестных членов.

Было решено задать матрицу не двумерным массивом, а одномерным, поэтому получить нужные индексы было нетривиальной задачей:

Пример : **mA[Nm\*(sv) + i - 1 + (NoX\*j)]**

Nm\*(sv) - перемещаемся по уравнениям(каждый раз перемещаемся на одно уравнение ниже)

когда i +/- 1 то просто прибавляем их

когда j+/-1 то +/- NoX так как у нас уравнение записывется в одну линию и чтобы переместиться на слой выше/ниже мы проходим как раз 1 раз по х

NoX\*j - чтобы записывать уравнения в зависимости от уровня(смещение получается как бы)

Генерация матрицы была сделана так, чтобы можно было просто добавить или убавить граничное условие( нужно добавить просто новое условие)

Текст программы (с комментариями)

#define nodePerL 1 // число узлов на ед длины

#define LENX 9 // число узлов по x

#define LENY 5 // число узлов по y

#define T0 10.0 // начальная температура

#define T1 400.0 // граничное условие право-низ

#define T2 0.0 // термоизоляция

#define Leng LENX\*LENY //Длина

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <unistd.h>

#include <fcntl.h>

int Nm, modTime; // nm - размер матрицы Modtime - время решения (то которое задано по заданию)

double mA[Leng\*Leng], vX[Leng], vB[Leng]; // вектора матрицы

double L, H, deltaX, deltaT, startT, Trl; // длина, высота, deltax = deltay

double startT, Trl; // startT - начальная температура, остальное - граничные условия

double a = 1.0; // a из формул

int NoX, NoY, edge; //число узлов по x и по y из дефайна

// Гаусс [5]

void frw\_one\_th () //Прямой ход

{

int i, j, k;

long double dgE;

for (k = 0; k < Nm; k++)

{

dgE = mA[Nm\*k + k];

for (j = k; j < Nm; j++)

mA[Nm\*k + j]/= dgE;

vB[k]/= dgE;

for (i = k + 1; i < Nm; i++)

{

dgE = mA[Nm\*i + k];

for (j = k; j < Nm; j++)

mA[Nm\*i + j]-= mA[Nm\*k + j]\*dgE;

vB[i]-= vB[k]\*dgE;

}

}

}

void bck\_one\_th () //Обратный ход

{

int i, j;

vX[Nm - 1] = vB[Nm - 1];

for (i = Nm - 2; i >= 0; i--)

{

vX[i] = vB[i];

for (j = i + 1; j < Nm; j++)

vX[i]-= mA[Nm\*i + j]\*vX[j];

}

}

//[5]

void all\_gen() //генерация начальной температуры

{

int i;

for (i = 0; i < Nm; i++)

{

vX[i] = startT;

}

}

void generan\_matrx()

{

/\*почему такое обращение к координатам :

Nm\*(sv) - перемещаемся по уравнениям(каждый раз перемещаемся на одно уравнение ниже)

когда i +/- 1 то просто прибавляем их

когда j+/-1 то +/- NoX так как у нас уравнение записывется в одну линию и чтобы переместиться на слой выше/ниже мы проходим как раз 1 раз по х

NoX\*j - чтобы записывать уравнения в зависимости от уровня(смещение получается как бы)

\*/

int i=0, j=0, sv=0;

//заполнение 0ями массивы

for (i = 0; i < Nm\*Nm; i++)

mA[i] = 0.0;

for (i = 0; i < Nm; i++)

vB[i] = 0.0;

for (j = 0; j < NoY; j++)

{

for (i = 0; i < NoX; i++)

{

if((i==NoX-1 && j==NoY-1)) //Углы, не понятно как их правильно (i==0 && j==NoY-1) ||

{

mA[Nm\*(sv) + i + (NoX\*j)] = 1.0/deltaX;

mA[Nm\*(sv) + i - 1 + (NoX\*j)] = -1.0/deltaX;

vB[sv] = 0.0;

sv++;

continue;

}

if(j==NoY-1)//условие второго рода сверху

{

mA[Nm\*(sv) + i + (NoX\*j)] = 1.0/deltaX;

mA[Nm\*(sv) + (i-NoX) + (NoX\*j)] = -1.0/deltaX;

vB[sv] = 0.0;

sv++;

continue;

}

if(i==NoX-1 && j<NoY/2) //условие первого рода правая половина низ

{

vB[sv] = Trl;

mA[Nm\*(sv) + i + (NoX\*j)] = 1.0/deltaX;

sv++;

continue;

}

if(i==NoX-1 && j>=NoY/2) //условие второго рода правая половина вверх

{

mA[Nm\*(sv) + i + (NoX\*j)] = 1.0/deltaX;

mA[Nm\*(sv) + i - 1 + (NoX\*j)] = -1.0/deltaX;

vB[sv] = 0.0;

sv++;

continue;

}

// Стандартный случай заполнения

if((i-1)>=0)

mA[Nm\*(sv) + i - 1 + (NoX\*j)] = -a/deltaX\*deltaX;

mA[Nm\*(sv) + i+(NoX\*j)] = 1.0/deltaT + 4.0\*a/deltaX\*deltaX;

if((i+1)<=Nm)

mA[Nm\*(sv) + i + 1 + (NoX\*j)] = -a/deltaX\*deltaX;

if((j\*NoX+i+NoX)<=Nm)

mA[Nm\*(sv) + (i + NoX) + (NoX\*j)] = -a/deltaX\*deltaX;

if((j\*NoX+i-NoX)>=0)

mA[Nm\*(sv) + (i - NoX) + (NoX\*j)] = -a/deltaX\*deltaX;

vB[sv] = vX[sv]/deltaT;

sv++;

}

}

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////

int main(int argc, char\* argv[])

{

int i=0, j, k, ret;

char string[80];

FILE \*fp;

fp = fopen( "result", "w" ); //открытие файла на запись данных

// проверка на наличие аргумента [1]

if (argc < 2)

{

printf("Нету аргумента - времени\n");

return(1);

}

//[1]

modTime = atoi(argv[1]); //задача времени

//задаем данные из дефайна и глобальных переменных [2]

L = LENX;

H = LENY;

Nm = Leng; // количество узлов всего

NoX = LENX\*nodePerL;

NoY = LENY\*nodePerL;

deltaX = H\*L/Nm; // дельта X

deltaT = 1; // дельта t

startT = T0;

Trl = T1;

//[2]

printf("Количество узлов: %d\n", Nm);

all\_gen(); // генерация начальных данных

for (i = 0; i < modTime/deltaT; i++) //решение уравнений

{

generan\_matrx();

frw\_one\_th();

bck\_one\_th ();

// Печать

//ret = Nm+NoX;

ret=0;

for (j = 0; j < NoY; j++)

{

// ret=ret-2\*NoX;

for (k = 0; k < NoX; k++)

{

sprintf(string, "%.3f ",vX[ret++]);

fputs(string,fp);

}

sprintf(string, "\n");

fputs(string,fp);

}

sprintf(string, "\n\n");

fputs(string,fp);

}

return(0);

}

Расчет температурного поля в Ansys.

1. Построение сечения.

Выполним команду Preprocessor->Modeling->Create->Keypoints->On working plane

В появившемся окне введем координаты точек (0,0) (8,0) (8,2) (0,4) (8,4) и нажимаем ok.

Затем необходимо соединить эти точки линиями:

Preprocessor ->Modeling->Create->Lines->Straight Line

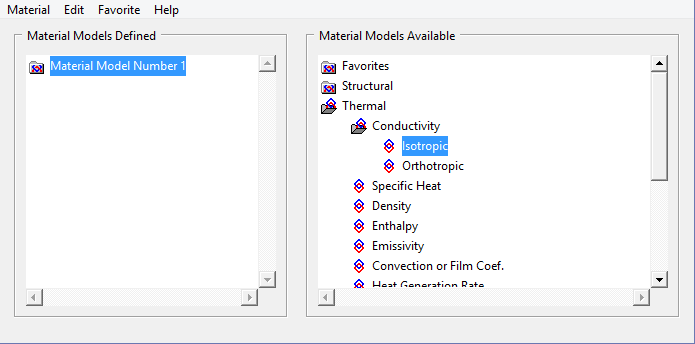
Соединяем построенные нами точки и нажимаем ok.

Затем строим «тело» трубы с помощью областей(Areas):

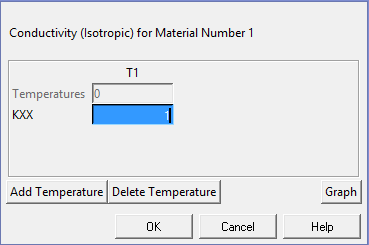
Preprocessor ->Modeling->Create->Areas->Arbitrary->By Lines

После это надо задать свойство материала.

Preprocessor->Material Props->Material Models.



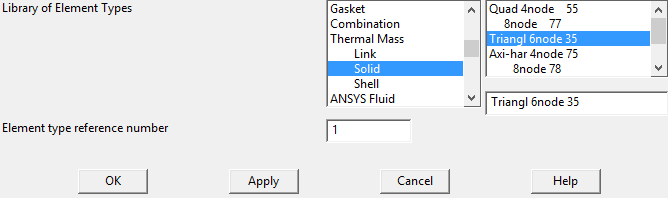
В появившемся окне КХХ- коэффициент тепловодности.



Далее задаем тип конечного элемента.

Preprocessor->Element Type->Add/Edit/Delete.

В появившемся окне нажимаем Add и выбираем тип элемента



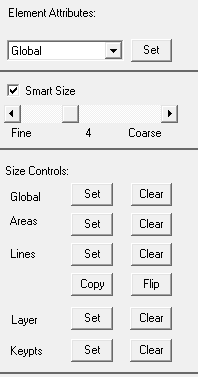
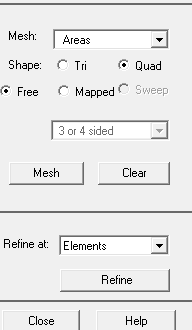
Нажимаем Ok.

В списке типов конечных элементов появится PLANE35.

Для того, чтобы задать сетку, выполним команду:

Preprocessor->Meshing->MeshTool

В меню Mesh Tool активизируем опцию Smart Size и устанавливаем значение 4.

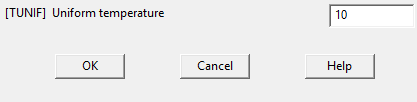
 

После задания размеров, нажимаем кнопку Mesh и выбираем сечение трубы

Для задания начального условия выполняем команду:

Main Menu: Preprocessor->Loads->Define Loads->Apply->Thermal->Temperature->Uniform Temp …

Появится меню, в котором можно задать начальную температуру 10 ºС:



Для задания граничных условий выполним:

Условие 1 рода: Preprocessor->Loads->Define Loads->Apply->Thermal->Temperature->On Lines

Задаем на половине правой стенки температуру 400 градусов

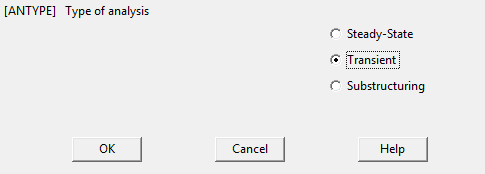
Условие 2 рода: Preprocessor->Loads->Define Loads->Apply->Thermal->Heat flux

Задаем на половине правой и верхней границ 0

Задаем тип анализа:

Main Menu: Solution –> -Analysis Type – New Analysis …

Появится меню, в котором можно указать тип анализа:

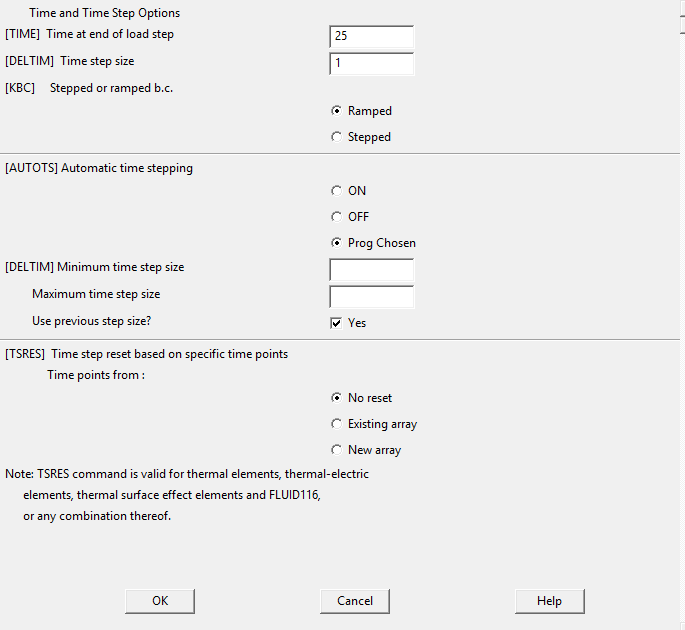


С помощью мыши устанавливаем опцию Transient – нестационарный анализ, нажимаем OK. Появится меню Transient Analysis, в котором также нажимаем OK.

Задаем временные параметры анализа:

Main Menu: Solution –> -Load Step Opts – Time/Frequenc -> Time – Time Step ..

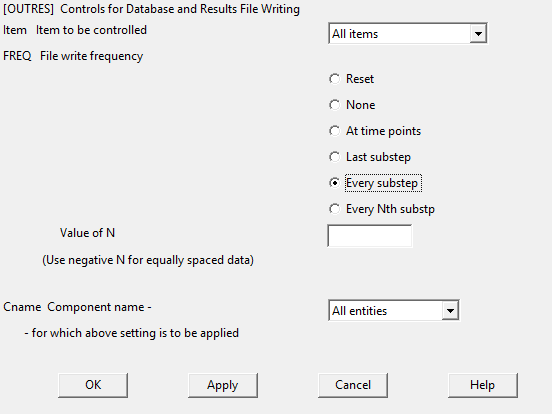
Появится меню, в котором можно указать временные параметры анализа:



В поле [TIME] Time at end of load step задаем значение 25, то есть время расчета равно 25с. В поле [DELTIM] Time step size задаем значение 1, то есть временной шаг равен 1 с.

Устанавливаем режим записи результатов расчета в файл (на каждом шаге)

Main Menu:  Solution ->-Load Step Opts->**Output Ctrls->DB/Results File**



Ввыпавшем меню помечаем «Every substep»

Выполняем команду расчета:

Main Menu: Solution **->** Solve **->**Current LS

Просмотр результатов с анимацией.

Main Menu:  **General Postproc**->Read Results->**First Set**

Utility Menu: **Plot**->**Elements**

Utility Menu:  **PlotCtrls**->**Style**->**Contours**->**Uniform Contours**

**В выпавшем окне выбираем количество контуров (Number of countours) – 10, и интервал между контурами, рассчитываемым автоматически (Auto calculated)**

Utility Menu**:  PlotCtrls->Animate->Over Time**

Получаем изменение температурного поля во времени.

Сравнение результатов.

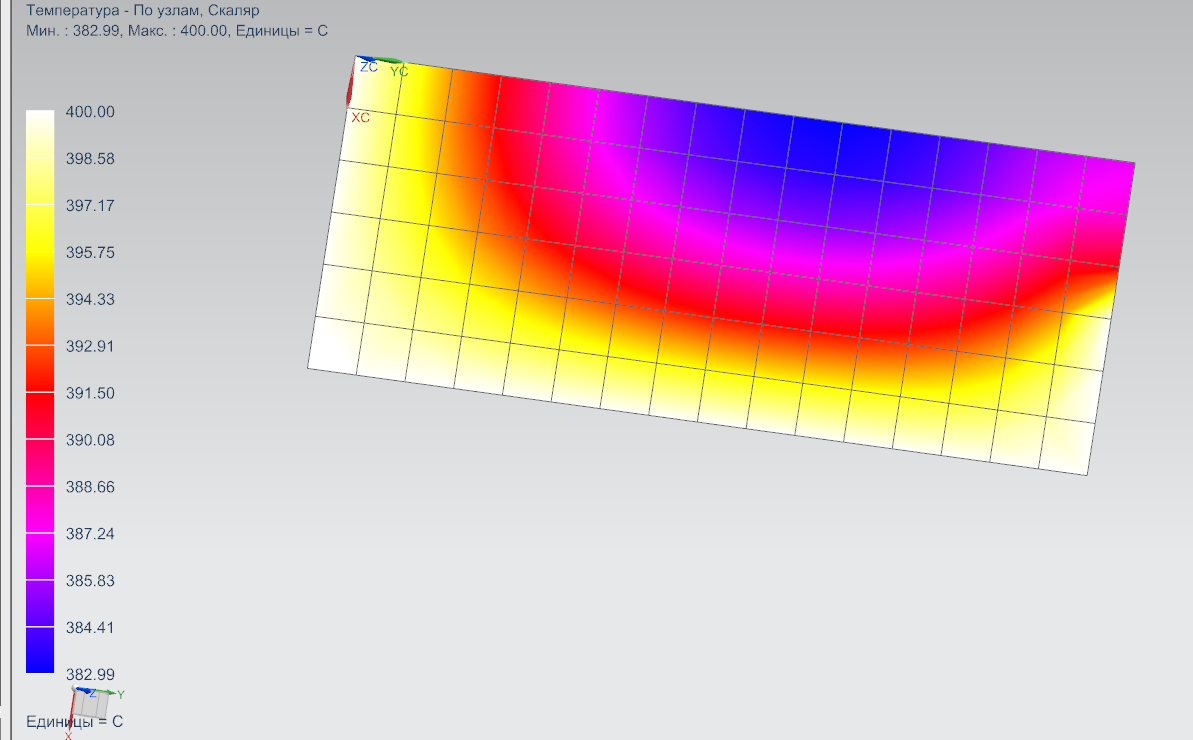
Возникла проблема, в Ансисе не получался качественно нужный график (вся пластина закрашивалась одним цветом). Хотя, проведя исследования и сделав другие варианты, как в ансисе так и в программе были получены сравнимые результаты. Было выявлено, что в Ансисе не был достигнут правильный результат, если была задана только одна температура (к примеру, только 1 первый род), иначе все строилось как должно.

Было решено изучить NX CAE и построить задание там и сравнить.

Использовался решатель Nastran 159

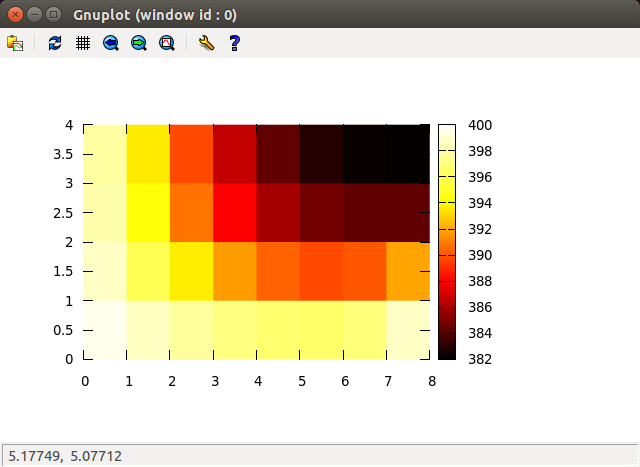
Мой вариант :

NX:



Отличие во времени и немного в градусах объяснимы тем, что NX требует данные которые нам не были заданы в задании(в реальных условиях влияет : Теплопроводность, толщина пластины, материал), но концептуально форма распространения была выдержана.

Программа:



Так же был решено взять другой вариант задания(Где Ансис строился), чтобы доказать состоятельность NX и программы.

**Был выбран соседний вариант – 43**

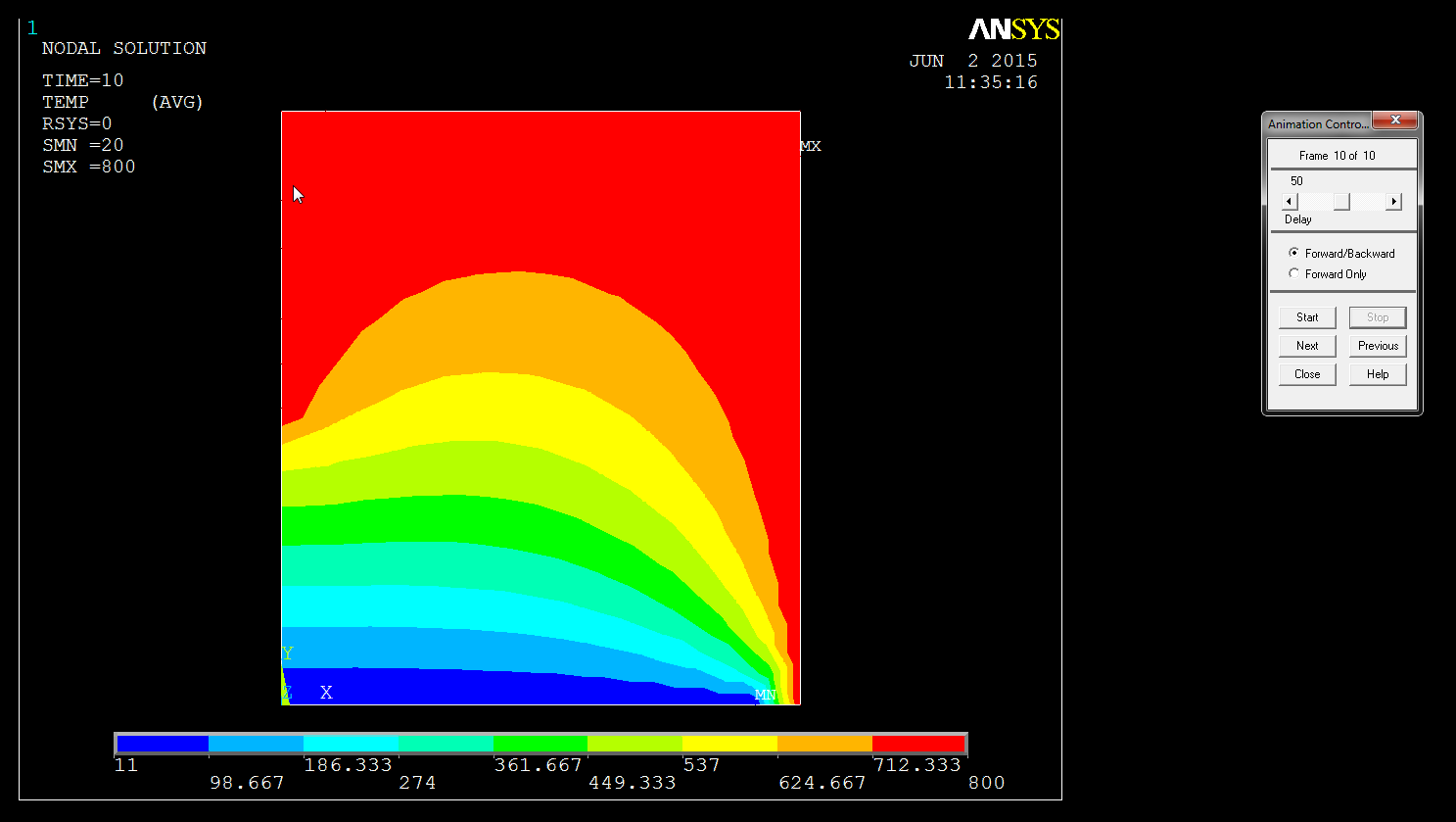
С помощью неявной разностной схемы решить нестационарное уравнение теплопроводности для прямоугольной пластины размером 7\*6см. Начальное значение температуры пластины - 10 градусов.

Граничные условия следующие: нижняя половина левой границы теплоизолирована, на нижней границе поддерживается 20, на остальной части границы температура 800 градусов.

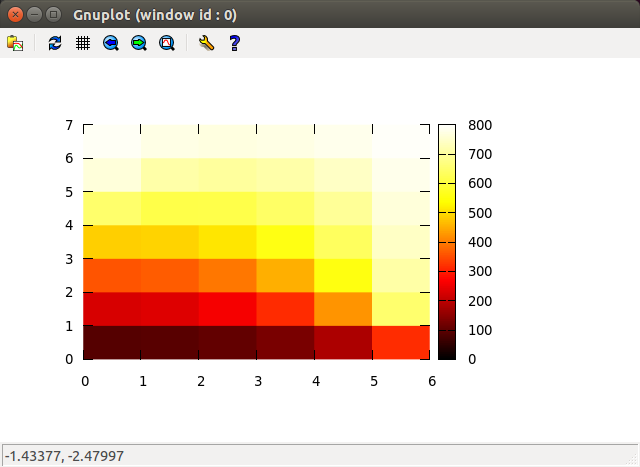
При выводе результатов показать динамику изменения температуры (например, с помощью цветовой гаммы).

Отчет должен содержать: текст программы, рисунок объекта с распределением температуры в момент времени 10 сек сравнение результатов расчета с результатами, полученными с помощью пакета ANSYS.

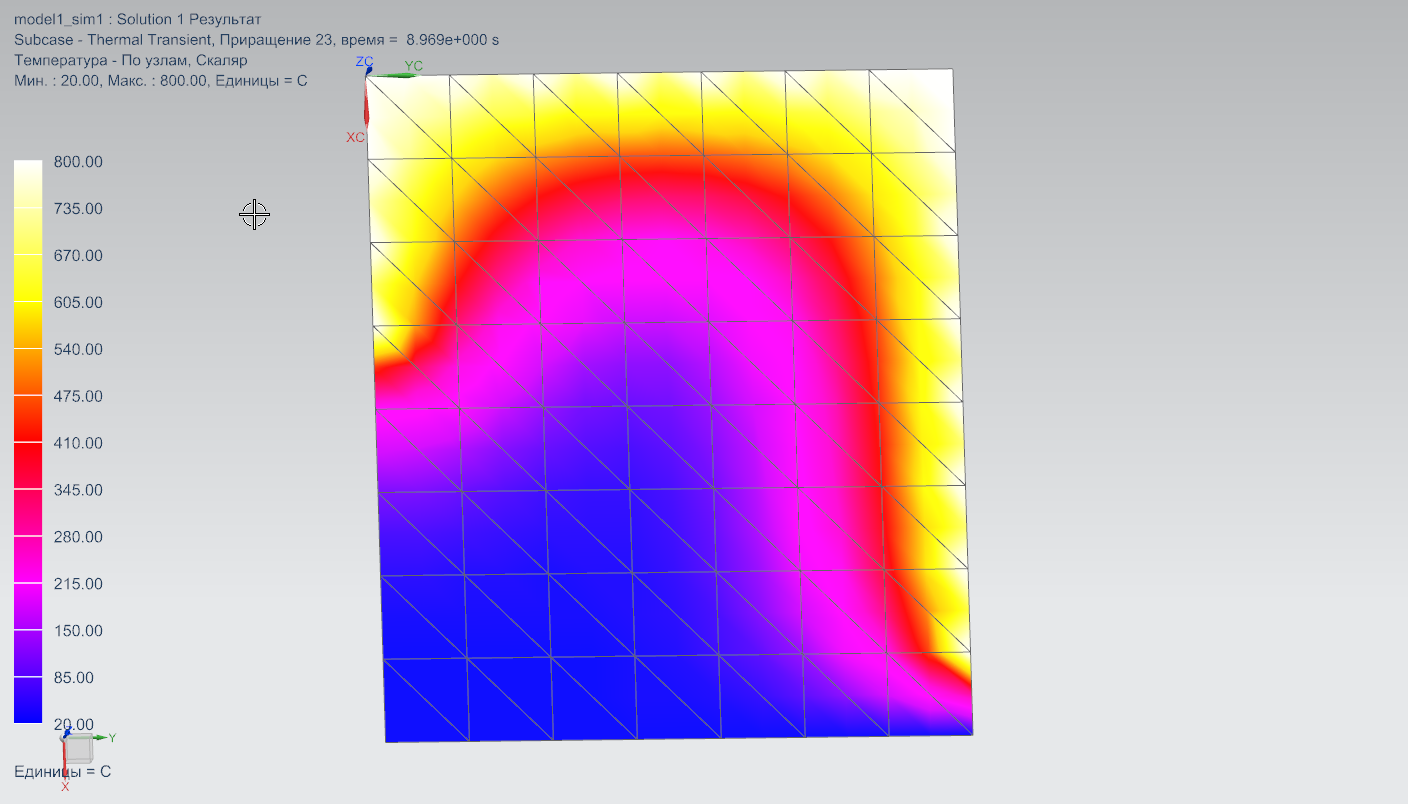
Ансис:



Программа:



NX:



Данным я доказываю, что программа работает правильно.

В ансисе можно было запилить, добавив доп условия, надо искать в материалах.