|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

Студент Журавлев Николай Вадимович

Группа РК6-62б

Тип задания Лабораторная работа

Тема лабораторной работы Метод конечных разностей

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Журавлев Н.В.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Трудоношин В.А\_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2022 г.*

Оглавление

[Цель выполнения лабораторной работы 3](#_Toc104724198)

[Задание 3](#_Toc104724199)

[Теоретическая часть 3](#_Toc104724200)

[Текст программы 4](#_Toc104724201)

[Результаты работы программы 10](#_Toc104724202)

[Результат работы ANSYS 11](#_Toc104724203)

[Вывод 12](#_Toc104724204)

# Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – ознакомиться с задачей теплопроводности и рассмотреть ее частный случай при программировании решения конкретной задачи.

# Задание

С помощью явной разностной схемы решить нестационарное уравнение теплопроводности для трубы, изображенной на рис.1, там же указаны размеры сторон.

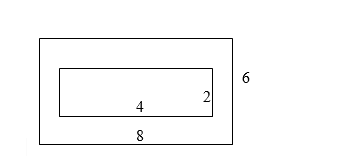


Рисунок 1

Граничные условия : внутри трубы протекает жидкость с температурой 100 градусов, на внешней границе задано условие:

dT/dn = T .

Начальное значение температуры трубы - 10 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 .

# Теоретическая часть

В условии указано: с помощью явной разностной схемы решить нестационарное уравнение теплопроводности для прямоугольной пластины. Уравнение имеет вид:

Или в разностной схеме это будет выглядеть:

Боковые граничные условия:

* Для нижней стороны;
* Для левой стороны;
* Для правой стороны;
* Для верхней стороны;

Так же имеется начальный нагрев трубы – 10 градусов. Время моделирования 25 секунд.

# Текст программы

Реализация представлена в листинге 1.

Листинг 1

|  |
| --- |
| #include <stdlib.h>  #include <stdio.h>  #include <math.h>  #include <string.h>  #define H 6.0  #define W 8.0  #define h 2.0  #define w 4.0  #define T\_LIQUID 100.0  #define T\_PIPE 10  #define TIME\_RUNNING 25  #define A 1  #define DX 0.1  #define DY 0.1  #define PATH\_PLT "t.plt"  #define PATH\_RES "r.txt"  void get\_index(int index, int ny, int\* ix, int\* iy) {  \*iy = index % ny;  \*ix = index / ny;  }  void write\_to\_file(double\*\* T, FILE\* fds1, int ny, int nx, int lx, int ly) {  FILE \*fd;  if ((fd = fopen(PATH\_PLT,"w")) == NULL) {  printf("Cannot open file!\n");  exit(0);  }  fprintf(fd, "set terminal png size 2000, 1000\n");  fprintf(fd, "set output 'screen.png'\n");  fprintf(fd, "set palette defined (0 \"black\", 1\"blue\", 2 \"green\", 3 \"yellow\", 4 \"orange\", 5 \"red\")\n");  fprintf(fd, "set xrange[0:%d]\n", lx);  fprintf(fd, "set yrange[0:%d]\n", ly);  fprintf(fd, "set cbrange[0:100]\n");  fprintf(fd, "plot '-' using 1:2:3 with image notitle\n");  for (int k = 0; k < ny \* nx; k++) {  int i, j;  get\_index(k, ny,&i, &j);  fprintf(fd, "%d %d %lf\n",i, j, T[i][j]);  fprintf(fds1,"%d %d %3lf\n", i, j, T[i][j]);  }  fprintf(fd, "e\n");  fprintf(fds1, "\n\n");  fclose(fd);  }  void initial\_pipe(double \*\*T, double \*\*T\_next, int N, int li, int bj, int tj, int ri, int ny) {  for (int k = 0; k < N; k++) {  int i, j;  get\_index(k, ny, &i, &j);  if (i == li && j >= bj && j <= tj) {  T\_next[i][j] = T\_LIQUID;  T[i][j] = T\_LIQUID;  continue;  }  // Правая  if (i == ri && j >= bj && j <= tj) {  T\_next[i][j] = T\_LIQUID;  T[i][j] = T\_LIQUID;  continue;  }  // Верхняя  if (j == tj && i >= li && i <= ri) {  T\_next[i][j] = T\_LIQUID;  T[i][j] = T\_LIQUID;  continue;  }  // Нижняя  if (j == bj && i >= li && i <= ri) {  T\_next[i][j] = T\_LIQUID;  T[i][j] = T\_LIQUID;  continue;  }  if (i > li && i < ri && j > bj && j < tj) {  T\_next[i][j] = T\_LIQUID;  T[i][j] = T\_LIQUID;  continue;  } else {  T[i][j] = T\_PIPE;  }  }  }  int main(int argc, char\* argv[]) {  if (argc != 4) {  printf("Use %s nx ny file\_name\n", argv[0]);  exit(1);  }  FILE \*file;  char \*file\_name;  int nx = atoi(argv[1]);  int ny = atoi(argv[2]);  file\_name = argv[3];  if ((file = fopen(file\_name,"w")) == NULL) {  printf("Cannot open file!\n");  exit(0);  }  int N = nx \* ny;  double l = (W - w) / 2.0;  double b = (H - h) / 2.0;  double hx = W / (nx - 1);  double hy = H / (ny - 1);  int li = (int)(l / hx);  int ri = (int)((l + w) / hx);  int bj = (int)(b / hy);  int tj = (int)((b + h) / hy);  double \*\*T = (double \*\*)malloc(sizeof(double \*) \* nx);  double \*\*T\_next = (double \*\*)malloc(sizeof(double \*) \* nx);  for (int k = 0; k < nx; k++) {  T[k] = (double \*) malloc(sizeof(double) \* ny);  T\_next[k] = (double \*) malloc(sizeof(double) \* ny);  }  initial\_pipe(T, T\_next, N, li, bj, tj, ri, ny);  hx = DX;  hy = DY;  int i, j;  for (int g = 0; g < TIME\_RUNNING; g++) {  double ht = 0;  for (int k = 0; k < N; k++) {  get\_index(k, ny, &i, &j);  // Внутренние границы  // Левая  if (i == li && j >= bj && j <= tj) {  T\_next[i][j] = T\_LIQUID;  T[i][j] = T\_LIQUID;  continue;  }  // Правая  if (i == ri && j >= bj && j <= tj) {  T\_next[i][j] = T\_LIQUID;  T[i][j] = T\_LIQUID;  continue;  }  // Верхняя  if (j == tj && i >= li && i <= ri) {  T\_next[i][j] = T\_LIQUID;  T[i][j] = T\_LIQUID;  continue;  }  // Нижняя  if (j == bj && i >= li && i <= ri) {  T\_next[i][j] = T\_LIQUID;  T[i][j] = T\_LIQUID;  continue;  }  if (i > li && i < ri && j > bj && j < tj) {  T\_next[i][j] = T\_LIQUID;  continue;  }  // Внешние границы  // Левая и правая  if (i == 0) {  continue;  }  if (i == nx - 1) {  continue;  }  //нижняя и верхняя  if (j == 0) {  continue;  }  if (j == ny - 1) {  continue;  }  T\_next[i][j] = ((T[i + 1][j] - 2 \* T[i][j] + T[i - 1][j]) / (hx \* hx) +  (T[i][j + 1] - 2 \* T[i][j] + T[i][j - 1]) / (hy \* hy)) \* A;  if (ht < fabs(T\_next[i][j])) {  ht = fabs(T\_next[i][j]);  }  }  for (int col = 0; col < nx; col++) {  for (int row = 0; row < ny; row++) {  if ((col == li && row >= bj && row <= tj) || (col == ri && row >= bj && row <= tj)  || (row == tj && col >= li && col <= ri) || (row == bj && col >= li && col <= ri)  || (col == 0) || (col == nx - 1) || (row == 0) || (row == ny - 1)  || (col > li && col < ri && row > bj && row < tj)) {  T[col][row] = T\_next[col][row];  } else {  T[col][row] = T\_next[col][row] \* (2 / ht) + T[col][row];  }  }  }  for (int k = 0; k < N; k++) {  get\_index(k, ny, &i, &j);  if (i == 0) {  T[0][j] = (T[1][j]) / (1 + hx);  continue;  }  if (i == nx - 1) {  T[i][j] = (T[i - 1][j]) / (1 + hx);  continue;  }  //нижняя и верхняя  if (j == 0) {  T[i][0] = (T[i][1]) / (1 + hy);  continue;  }  if (j == ny - 1) {  T[i][j] = (T[i][j - 1]) / (1 + hy);  continue;  }  }  T[0][0] = (T[0][1]) / (1 + hx);  T[0][ny - 1] = (T[0][ny - 2]) / (1 + hy);  }  for (j = ny - 1; j >= 0; j--) {  for (i = 0; i < nx; i++) {  printf("%8.3f", T[i][j]);  }  printf("\n");  }  FILE \*fd;  if ((fd = fopen(PATH\_RES,"w")) == NULL) {  printf("Cannot open file!\n");  exit(0);  }  for (j = ny - 1; j >= 0; j--) {  for (i = 0; i < nx; i++) {  fprintf(fd, "%8.3f", T[i][j]);  }  fprintf(fd, "\n");  }  printf("\nhx = %g\nhy = %g\n", hx, hy);  write\_to\_file(T,file, ny, nx, atoi(argv[1]) - 1, atoi(argv[2]) - 1);  fclose(file);  } |

# Результаты работы программы

В результате работы программы получился следующий график распределения температуры в трубе для сетки 41 на 31 (в силу симметрии представлена ¼ часть таблицы):

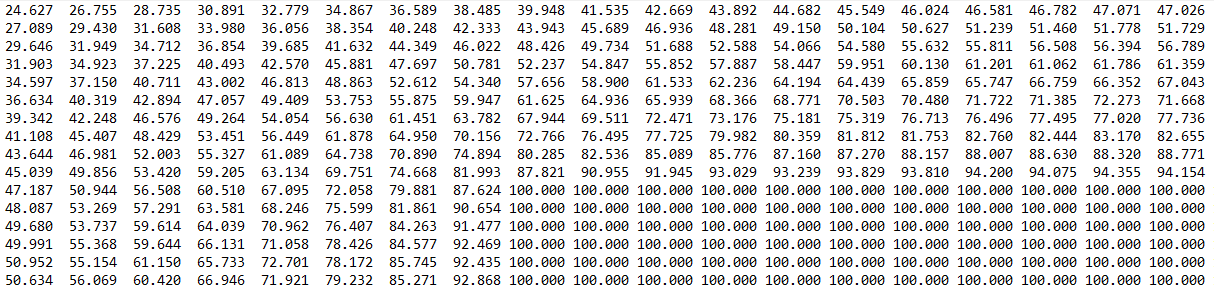


Рисунок 2. Таблица результатов работы программы

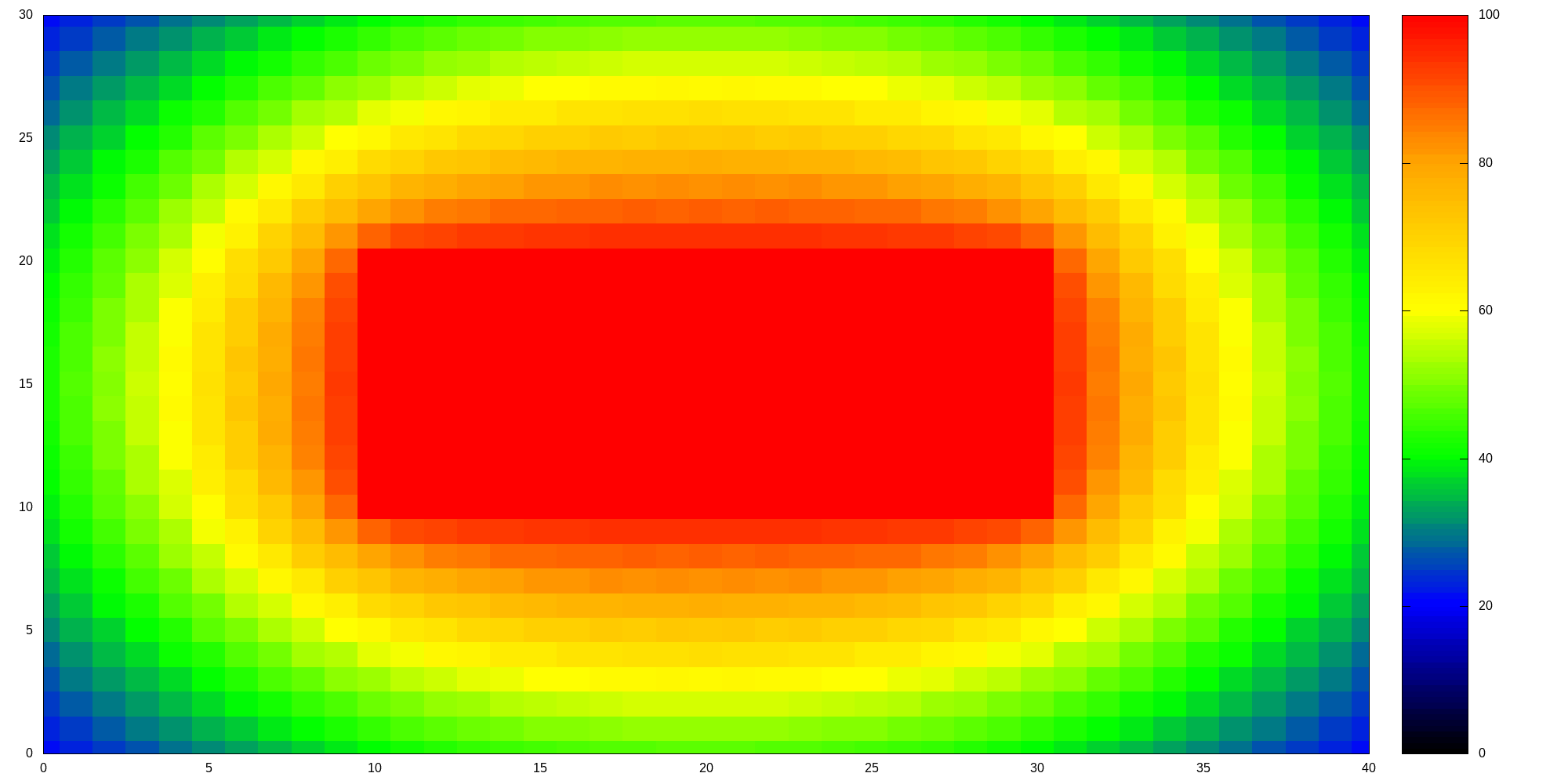


Рисунок 3. График результата работы программы

# Результат работы ANSYS

Строим схему. Выбираем тип конечного элемента:

**Main Menu: Preprocessor → Element Type → Add/Edit/Delete**

Выбираем **Thermal Solid** (вид анализа) и **Triangl 6node**.

Теперь разбиваем область на конечные элементы:

**Main Menu: Preprocessor Meshing → Mesh Tool**

Прикладываем граничное условие первого и третьего рода к линиям:

**Main Menu: Preprocessor → Loads → Define Loads → Apply → Thermal Temperature → On Lines**

**Main Menu: Preprocessor →Loads → Loads- Apply →-> Thermal- Convection → On Areas**

Задаем тип анализа:

**Main Menu: Solution → Analysis Type→New Analysis**

Выполняем команду расчета:

**Main Menu: Solution → Solve → Current LS**

Для просмотра поля температур выполняем команду:

**Main Menu: General PostProc → Plot Results → Contour Plot → Nodal Solution**

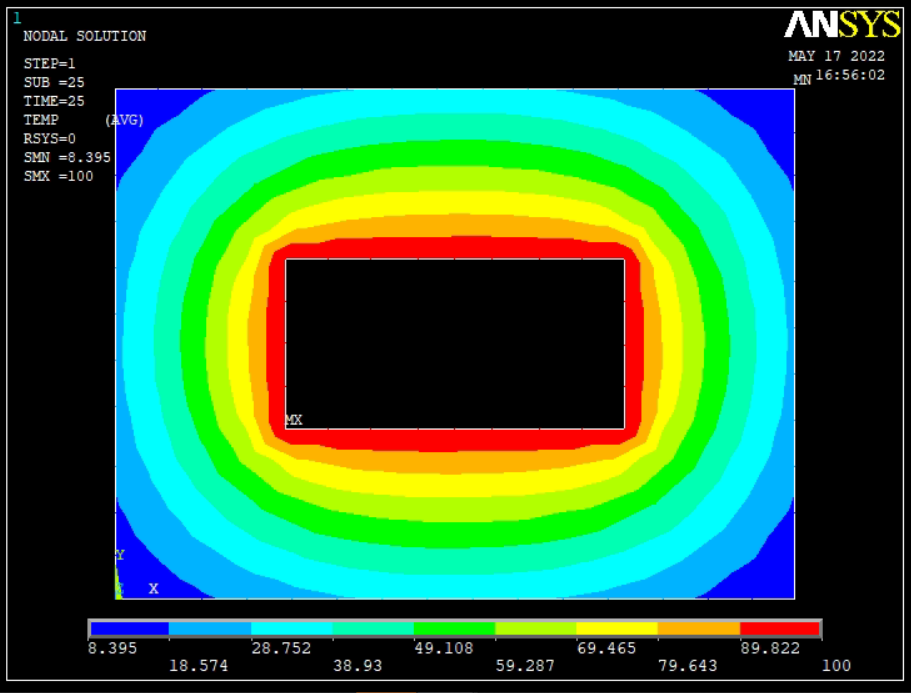


Рисунок 4. Результат в программе Ansys

# Вывод

При выполнении домашнего задание работ была решена двумерная стационарная задача теплопроводности методом конечных разностей и проведена проверка результатов в Ansys.