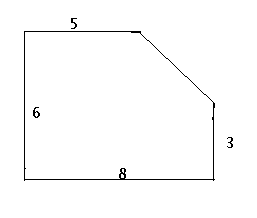
***Шевченко Артем, РК6-61***

***Задание 18***

Методом конечных разностей решить задачу распространения тепла в пластине, изображенной на рис.



Размеры сторон показаны на рисунке. Граничные условия : слева 100 градусов, внизу 0, справа и вверху 50 , на скошенной границе задано условие:

dT/dn = T .

Отчет должен содержать: текст программы, таблицы результатов, рисунок объекта с распределением фазовой переменной, сравнение результатов расчета с результатами, полученными с помощью пакета ANSYS .

**Текст программы.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define UP 50

#define DOWN 0

#define LEFT 100

#define RIGHT 50

#define X 8

#define Y 6

#define EDGE 3

#define dX 0.25

#define dY 0.25

int n;

double \*a;

double \*b;

double \*x;

int plus(int i)

{

double d;

if((EDGE/dY -1-i)>1E-5)

{

d=(X-EDGE + (i+1)\*dY)/dX;

return (int)round(d);

}

else

{

return (int)round(X/dX-1);

}

}

int pl()

{

return (int)round(X/dX-1);

}

int num(int i, int j)

{

return i\*round(X/dX-1)+j;

}

void fill()

{

int i, j;

for(i=0; i<round(Y/dY-1); i++)

{

for(j=0; j<round(X/dX-1); j++)

{

if(j>=plus(i))

{

a[num(i, j)\*n+num(i, j)]=1;

b[num(i, j)]=8;

continue;

}

if((j==(plus(i)-1)) && (EDGE/dY -1-i)>1E-5)

{

/\*if(i%2)

{

a[num(i, j)\*n+num(i, j)]=-2;

a[num(i, j)\*n+num(i, j)-1]=1;

}

else

{

a[num(i, j)\*n+num(i, j)]=-2;

a[num(i, j)\*n+num(i, j)+pl()]=1;

}\*/

double dn=sqrt(dX\*dX+dY\*dY);

a[num(i, j)\*n+num(i, j)]=-(1+dn);

a[num(i, j)\*n+num(i, j)+pl()-1]=1;

continue;

}

a[num(i, j)\*n+num(i, j)]=-4;

if(j==0)

b[num(i, j)]-=LEFT;

else

a[num(i, j)\*n+num(i, j)-1]=1;

if(i==0)

b[num(i, j)]-=UP;

else

a[num(i, j)\*n+num(i, j)-pl()]=1;

if((j==(plus(i)-1)) && (EDGE/dY -1-i)<1E-5)

b[num(i, j)]-=RIGHT;

else

a[num(i, j)\*n+num(i, j)+1]=1;

if((Y/dY-i)-2<1E-5)

b[num(i, j)]-=DOWN;

else

a[num(i, j)\*n+num(i, j)+pl()]=1;

}

}

}

void out()

{

int i, j;

for(i=0; i<round(Y/dY-1); i++)

{

for(j=0; j<round(X/dX-1); j++)

{

printf("%5.2f ", x[num(i, j)]);

//printf("%d\n", num(i, j));

}

printf("\n");

}

}

void lae\_solver\_gauss(int \*ier)

{

int i, j, k;

double sum, bv, mik, rab[2];

for(i=0; i<n-1; i++)

{

printf("%d\n", i);

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

{

if(fabs(a[j\*n+i]) > fabs(a[i\*n+i]))

{

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

{

rab[k]=a[i\*n+k];

a[i\*n+k]=a[j\*n+k];

}

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

{

a[j\*n+k]=rab[k];

}

bv=b[i];

b[i]=b[j];

b[j]=bv;

}

if(a[i\*n+i]==0)

{

\*ier=1;

return;

}

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

{

mik=a[j\*n+i]/a[i\*n+i];

a[j\*n+i]=0.0;

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

{

a[j\*n+k]=a[j\*n+k]-mik\*a[i\*n+k];

}

b[j]=b[j]-mik\*b[i];

}

}

}

if(a[(n-1)\*n+(n-1)]==0)

{

\*ier=1;

return;

}

x[n-1]=b[n-1]/a[(n-1)\*n+(n-1)];

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

{

for(j=i+1, sum=0; j<n; j++)

sum=sum+a[i\*n+j]\*x[j];

if(a[i\*n+i]==0)

{

\*ier=1;

return;

}

x[i]=(b[i]-sum)/a[i\*n+i];

}

}

int main()

{

n=round((X/dX-1)\*(Y/dY-1));

a=malloc(n\*n\*sizeof(double));

b=malloc(n\*sizeof(double));

x=malloc(n\*sizeof(double));

int ier=0, i, j;

fill();

/\*for(i=0; i<n; i++)

{

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

printf("%2.0f ", a[i\*n+j]);

printf("\t%2.0f\n", b[i]);

}\*/

lae\_solver\_gauss(&ier);

if(!ier)

out();

else

printf("ERR!\n");

printf("\n");

return 0;

}

**Таблица результатов.**

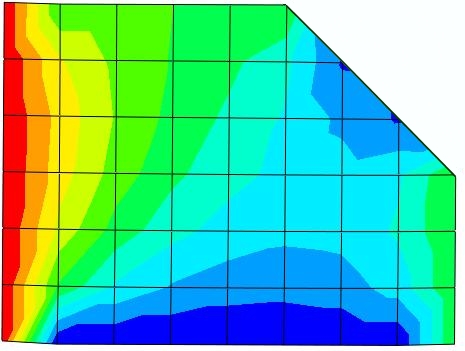
70.92 57.42 49.48 42.97 33.84 12.41

76.28 59.26 47.51 38.58 29.96 19.48 10.41

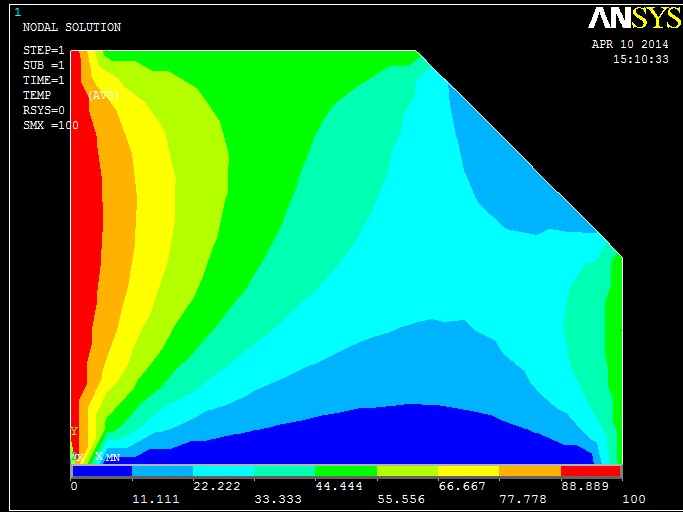
74.94 55.85 42.73 33.88 27.94 25.14 29.35

67.62 46.46 33.68 26.26 22.79 23.77 31.85

49.08 28.70 19.26 14.68 13.20 15.32 24.29



**Расчет в ANSYS.**



Исследование показало, что результаты при решении двумя разными методами получились схожими с небольшой погрешностью.