МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Физико-механический институт

Высшая школа прикладной математики и вычислительной физики

Отчет по курсовому проекту  
по курсу:

ЧИСЛЕННОЕ РЕШЕНИЕ КРАЕВЫХ ЗАДАЧ

на тему:

Исследование нестационарного поля температур в плоской неограниченной пластине с использованием неявного метода конечных разностей на неравномерной сетке

Работу выполнил:

студент группы 5030301/10002

Тугай В.В.

Преподаватель:

к.т.н., доц. Плетнев А.А.

Санкт-Петербург

2023

# Содержание

1. Постановка задачи
2. Метод решения
3. Рисунок с расположением узлов сетки (*xi*) и границ сеточных ячеек (*xxi*) вблизи стенки (примерно 10 узлов)
4. Стационарный профиль температуры
5. Распределение коэффициента *λ* в сечении пластины
6. Вывод
7. Приложение

# Постановка задачи

Рассчитать стационарное поле температуры в плоской неограниченной пластине толщиной  1.5 см с равномерно распределённым источником внутреннего объемного тепловыделения Qv=50000 Вт/м3, и коэффициентом теплопроводности, линейно зависящим от температуры с коэффициентом линейной зависимости kt=-0.004 1/K. На боковых поверхностях пластины ГУ 3 рода с коэффициентами теплоотдачи α1=60 Вт/(м²⋅К) и α2=20 Вт/(м²⋅К). Задачу решить численным методом конечных разностей на неравномерной сетке с 35 узлами и коэффициентом сгущения узлов сетки k∆=1.2 с использованием неявной апроксимации.

*T*– температура пластины, K

*Te1*– температура на левой границе пластины, K

*Te2*– температура на правой границе пластины, K

*T0*– начальная температура пластины, K

*τ* – итерационный параметр

*x*– координата, м

*L* – толщина пластины, м

*q*– плотность теплового потока, Вт/м2

α1 – коэффициент конвективной теплоотдачи на левой границе пластины, Вт/(м2×К)

α2 – коэффициент конвективной теплоотдачи на правой границе пластины, Вт/(м2×К)

*λ* – коэффициент теплопроводности, Вт/(м×К)

Im – количество узлов сетки

k∆ – коэффициент сгущения узлов сетки

kt – коэффициент линейной зависимости λ(T), 1/K

Qv – источник внутреннего тепловыделения, Вт/м3

Физические свойства материала и значения физический величин (Вариант-26):

*λ0*=6.5 Вт/(м⋅К)

*L*=0.015 м

*Te1*=253.15 K

*Te2*=283.15 K

*T0*=273.15 K

A diagram of a function

Description automatically generated with low confidence

Рисунок 1

Расчетная область и стационарный профиль температуры

Одномерное стационарное уравнение теплопроводности с источником внутреннего тепловыделения:

– коэффициент линейной зависимости

Граничные условия:

Нестационарный аналог уравнения для решения методом установления

где – итерационный параметр.

# Метод конечных разностей с использованием неравномерной стеки

Построение неравномерной конечно-разностной сетки

Введем обозначение - шаг сетки. Для неравномерной сетки шаг не постоянен. Величина сгущения узлов определяется отношением двух соседних шагов сетки: коэффициент сгущения. Из теории разностных схем известно, что допустимый диапазон изменения коэффициента сгущения *,* , т.е. уменьшение (при ) или увеличение (при ) не должно быть слишком большим.

Узлы сетки целесообразно сгущать там, где ожидается появление больших градиентов искомой функции, в нашем случае – около границ пластины.

Сгущение вблизи левой границы пластины предполагает, что с возрастанием индекса *i* шаг сетки увеличивается. В этом случае Для практических целей вполне достаточно рассмотреть случай *Const.* Тогда последовательность шагов сетки образует геометрическую прогрессию, сумма всех членов которой равна (*k*, *Im* и *L* заданы)

(1)

Из (1) можно найти величину наименьшего шага сетки . Все последующие шаги и координаты узлов могут быть найдены по рекуррентным соотношениям

, *i* = 3 … *Im*;  *i* = 2 … *Im*.

Поскольку в нашей задаче сгущение узлов должно быть выполнено с двух сторон (вблизи левой и правой границ пластины), следует сначала построить сетку для одной половины расчетной области (при этом , где *Im* обязательно нечетное), а на вторую половину перенести координаты узлов отражением от центра пластины *х* = *L*/2 (например, по формуле (2)).

(1a)

, *i* = 3 … *Im\**;  *i* = 2 … *Im\**.

, *i* = (*Im\*+1) … Im*; (2)

Далее (см. рис.2) будет введено понятие границ сеточной ячейки, координаты которых обозначены и могут быть вычислены по формуле

(3)

Чтобы узлы располагались точно по центру сеточных ячеек, необходимо после вычисления по (3) выполнить коррекцию

(4)

Крайние узлы корректировать не нужно, так как их координаты по определению совпадают с границами области:

Вывод сеточных уравнений

Для построения разностной схемы введем в рассмотрение понятие элементарной сеточной ячейки, границы которой обозначены (рис. 2). При этом узлы сетки (за исключением граничных) должны быть расположены в центре сеточных ячеек.

A picture containing text, line, font, number

Description automatically generated

Рисунок 2

Шаблон аппроксимации уравнения теплопроводности

Считаем, что в пределах одной сеточной ячейки коэффициент постоянен: . Тогда тепловой поток на границе сеточной ячейки (см. рис. 2) может быть вычислен по формуле

(5)

С учетом (5) неявную разностную аппроксимацию уравнения для внутренних узлов (*i* = 2 ... *Im*-1) запишем в виде

(6)

Для крайних (граничных) ячеек получим

(7)

(12)

В выражениях (10) - (12) предполагается, что значения вычисляются на текущем временном слое, т.е. с «запаздыванием»:

Решение системы линейных алгебраических уравнений (СЛАУ)

Разностное уравнение (10) представим в виде

(13)

где искомая сеточная функция, – прогоночные коэффициенты.

Как видно из (13), коэффициенты представляют собой выражения, на которые умножаются соответствующие неизвестные значения *T* в разностном уравнении (10). Коэффициент объединяет слагаемые разностного уравнения, которые не вошли ни в один из коэффициентов .

Перенесем все слагаемые (10) в правую часть уравнения. Тогда получим:

Для крайних узловых точек некоторые из прогоночных коэффициентов не определены. Поэтому постулируем .

Когда все четыре массива ***A***, ***B***, ***C***, ***D*** сформированы, для нахождения массива следует выполнить обращение к процедуре векторной прогонки.

Вычисление теплового баланса

При достижении стационарного состояния количество теплоты, выделяемое внутри пластины, должно быть равно количеству теплоты, отводимой в окружающую среду от ее поверхности. Поскольку задача решается методом установления, то указанное равенство достигается только в пределе при . Поэтому условием достижения теплового баланса (и одновременно выхода из итеративного цикла) является:

(14)

где – заданная точность.

Этапы программной реализации

1) Задание исходных данных

2) Вычисление координат узлов сетки (массив) и координат границ сеточных

ячеек (массив ). Коррекция массива .

3) Вычисление итерационного параметра .

4) Присвоение начальных значений массиву .

5) Начало итеративного цикла.

6) Вычисление значений в узлах сетки.

7) Вычисление прогоночных коэффициентов (заполнение массивов *A*, *B*, *C*, *D* для всех *i* = 1 ... *Im*).

8) Обращение к процедуре прогонки (вычисление элементов массива ).

9) Проверка достижения теплового баланса (условие выхода из цикла).

10) Переприсвоение массивов ( ).

11) Возврат к п.6

# Результаты решения

# Программа main.f95, написанная на Fortran 95, состоит из основного тела программы и модуля methods, содержащий в себе 3 подпрограммы : подпрограмма sweep отвечает за Решение СЛАУ для нахождения массива T с помощью прогоночных коэффициентов; подпрограмма grid для построения неравномерной сетки, сгущающейся у стенок пластины; подпрограмма coeffs, использующие предыдущие две подпрограммы при работе для определения значений профиля температуры и отвечающая за расчет данных, представленных в этом отчете (см. Приложение (1)).

# Рисунок с расположением узлов сетки (xi) и границ сеточных ячеек (xxi) вблизи стенки (примерно 10 узлов)

# На рисунке 3 изображено расположение узлов сетки и границ сеточных ячеек вблизи правой стенки. Как видно из рисунка, сетка действительно является неравномерной и сгущается в направлении к стенкам.

A picture containing text, screenshot, line, rectangle

Description automatically generated

Рисунок 3

Расположение узлов сетки (xi) и границ сеточных

ячеек (xxi) вблизи правой стенки (10 узлов)

# Стационарный профиль температуры

# На рисунке 4 изображен стационарный профиль температуры в Кельвинах. Как видно из рисунка, профиль температуры имеет горизонтальную асимптоту – значения температуры от левой стенки до центра пластины линейно возрастает, и далее от центра пластины до правой стенки производная графика уменьшается и значения температуры увеличиваются, стремясь к максимальному значению. Значения на правой и левой стенках Tleft=269.626 K; Tright=271.224 K (см. Приложение (2)), соответственно, и максимальное значение температуры достигается на правой стенке Tmax=271.224 K (см. Приложение (2)).

A picture containing line, plot, diagram, text

Description automatically generated

Рисунок 4

Стационарный профиль температуры

# Распределение коэффициента λ в сечении пластины

# На рисунке 5 изображено распределение коэффициента теплопроводности в сечении пластины. Сравнивая рисунки 4, 5 можно подметить, что рисунок 5 является отражением по вертикали рисунка 4.

A picture containing text, line, plot, diagram

Description automatically generated

Рисунок 5

Распределение коэффициента λ в сечении пластины

# Тепловые потоки на границах и тепловой баланс

# Тепловой поток на левой границе qleft=991.45 Вт/м2, на правой qright=- 241.38 Вт/м2, тепловой баланс Ba= =1.0001 .

# Вывод

# В ходе решения поставленной задачи был изучен метод конечных разностей на неравномерной сетке. Решение задачи данным методом было реализовано на языке программирования Fortran (см. Приложение 1), результаты программы представлены в виде графиков с использованием пакета Matlab.

# По итогам работы программы были посчитаны суммарное тепловыделение в сечении пластины Qsum=750 Вт/м2, суммарный тепловой поток на границах qsum=750.08 Вт/м2, и точность решения ∆balance= Ba-1=0.0001 (см. Приложение (2)).

# Приложение 1

# Основная программа

module methods

contains

subroutine sweep(*Im*,*A*,*B*,*C*,*D*,*F*)

implicit none

integer, intent(in):: Im

real\*16, dimension(1:Im), intent(in):: A, B, C, D

real\*16, dimension(1:Im), intent(out):: F

real\*16, dimension(1:Im):: alpha, beta

real\*16 :: k0

integer::i

!Прямой ход

alpha(1) = -A(1) / B(1)

beta(1) = -D(1) / B(1)

do i = 2, (Im-1)

k0 = (B(i) + C(i)\*alpha(i-1))

alpha(i) = -A(i) / k0

beta(i) = -(D(i) + C(i)\*beta(i-1)) / k0

end do

!Обратный ход

F(Im) = -(D(Im) + C(Im)\*beta(Im-1)) / (B(Im) + C(Im)\*alpha(Im-1))

do i = (Im-1), 1, -1

F(i) = alpha(i)\*F(i+1) + beta(i)

end do

end subroutine

subroutine grid(*L*, *Im*, *koeff\_d*, *lambda\_0*, *x*, *xx*, *delt\_tau*)

implicit none

real\*16 :: L, koeff\_d, lambda\_0, alpha\_1, alpha\_2, T0, Te1, Te2, po, cp, delt\_tau

real\*16, dimension (Im) :: delta\_x, x, xx, lambda

integer :: Im, i, n, Im\_star

Im\_star=(Im+1.0)/2.0

x(1)=0.0

x(Im)=L

delta\_x(2)=(L\*(koeff\_d-1.0))/(2.0\*((koeff\_d\*\*(Im\_star-1.0))-1.0))

do i=2,Im\_star

if (i/=2) delta\_x(i)=delta\_x(i-1)\*koeff\_d

x(i)=x(i-1)+delta\_x(i)

end do

do i=Im\_star+1,Im

x(i)=x(i-1)+delta\_x(Im+2-i)

end do

do i=2,Im

xx(i)=(x(i)+x(i-1))/2.0

end do

do i=2,Im-1

x(i)=(xx(i+1)+xx(i))/2.0

end do

delt\_tau=(L\*\*2)/(2\*lambda\_0\*((Im-1)\*\*2))

end subroutine

subroutine coeffs(*L*, *Im*, *koeff\_d*, *Qv*, *lambda\_0*, *koeff\_t*, *alpha\_1*, *alpha\_2*, *T0*, *Te1*, *Te2*, *po*, *cp*, *eps*)

implicit none

real\*16 :: L, koeff\_d, Qv, lambda\_0, koeff\_t, alpha\_1, alpha\_2, T0, Te1, Te2, po, cp

real\*16 :: eps, delt\_tau, balance

real\*16, dimension (Im) :: x, xx, lambda, T, q, AA, BB, CC, DD

integer :: Im, i, n, Im\_star, check=0

call grid(L, Im, koeff\_d, lambda\_0, x, xx, delt\_tau)

T=T0; AA=0; CC=0

do n=1,200000

lambda=lambda\_0\*(1.0+koeff\_t\*(T-T0))

do i=2,Im-1

AA(i)=(lambda(i+1)\*lambda(i))/((xx(i+1)-xx(i))\*(lambda(i)\*(x(i+1)-xx(i+1))+lambda(i+1)\*(xx(i+1)-x(i))))

CC(i)=(lambda(i)\*lambda(i-1))/((xx(i+1)-xx(i))\*(lambda(i-1)\*(x(i)-xx(i))+lambda(i)\*(xx(i)-x(i-1))))

BB(i)=-AA(i)-CC(i)-(1.0/delt\_tau)

DD(i)=(T(i)/delt\_tau)+Qv

end do

AA(1)=(lambda(2)\*lambda(1))/((xx(2)-x(1))\*(lambda(1)\*(x(2)-xx(2))+lambda(2)\*(xx(2)-x(1))))

BB(1)=-AA(1)-(1.0/delt\_tau)-(alpha\_1/(xx(2)-x(1)))

DD(1)=(T(1)/delt\_tau)+((alpha\_1\*Te1)/(xx(2)-x(1)))+Qv

CC(Im)=(lambda(Im)\*lambda(Im-1))/((x(Im)-xx(Im))\*(lambda(Im-1)\*(x(Im)-xx(Im))+lambda(Im)\*(xx(Im)-x(Im-1))))

BB(Im)=-CC(Im)-(1.0/delt\_tau)-(alpha\_2/(x(Im)-xx(Im)))

DD(Im)=(T(Im)/delt\_tau)+((alpha\_2\*Te2)/(x(Im)-xx(Im)))+Qv

call sweep(Im,AA,BB,CC,DD,T)

balance=abs(1-(((alpha\_2\*(T(Im)-Te2))+alpha\_1\*(T(1)-Te1))/(Qv\*L)))

if (balance<=eps) exit

end do

write(16,\*) T+273.15; write(17,\*) lambda; write(14,\*) x; write(15,\*) xx

write(\*,'(a12,$)') ' 6) max T: ' ; write(\*,'(f7.3,$)') maxval(T)+273.15; write(\*,'(a19,$)') ' K , left wall T: ';

write(\*,'(f7.3,$)') T(1)+273.15; write(\*,'(a20,$)') ' K , right wall T: ' ; write(\*,'(f7.3,$)') T(Im)+273.15; write(\*,\*) 'K'

write(\*,'(a23,$)') ' 8) heat dissipation: ' ; write(\*,'(f8.3,$)') Qv\*L ; write(\*,\*) 'W/m^2'

write(\*,'(a16,$)') ' 9) heat flow: ' ; write(\*,'(f8.3,$)') (alpha\_2\*(T(Im)-Te2))+(alpha\_1\*(T(1)-Te1)); write(\*,\*) 'W/m^2'

write(\*,'(a20,$)') ' 10) balance error: '; write(\*,'(f14.12)') balance

write(\*,'(a17,$)') ' left heat flow: '; write(\*,'(f8.3,$)') alpha\_1\*(T(1)-Te1)

write(\*,'(a20,$)') ' , right heat flow: '; write(\*,'(f8.3,$)') alpha\_2\*(T(Im)-Te2)

write(\*,'(a19,$)') ' , term balance: '; write(\*,'(f14.12)') ((alpha\_2\*(T(Im)-Te2)+alpha\_1\*(T(1)-Te1))/(Qv\*L))

end subroutine

end module

program main

use methods

implicit none

real\*16 :: L, koeff\_d, Qv, lambda\_0, koeff\_t, alpha\_1, alpha\_2, T0, Te1, Te2, po, cp, eps

integer :: Im

open(201, file='bin/data.txt')

read(201,\*) L; read(201,\*) Im; read(201,\*) koeff\_d; read(201,\*) Qv; read(201,\*) lambda\_0; read(201,\*) koeff\_t; read(201,\*) alpha\_1;

read(201,\*) alpha\_2; read(201,\*) T0; read(201,\*) Te1; read(201,\*) Te2; read(201,\*) po; read(201,\*) cp; read(201,\*) eps

close(201)

open(14, file='bin/results/x\_i.txt'); open(15, file='bin/results/xx\_i.txt')

open(16, file='bin/results/T.txt'); open(17, file='bin/results/lambda\_i.txt')

write(\*,\*)'start program'; write(\*,\*); write(\*,'(a4,$)') ' Im='; write(\*,'(i3)') Im; write(\*,\*)

call coeffs(L, Im, koeff\_d, Qv, lambda\_0, koeff\_t, alpha\_1, alpha\_2, T0, Te1, Te2, po, cp, eps)

close(14); close(15); close(16); close(17)

write(\*,\*); write(\*,\*) 'end program'

end program

# Приложение 2

# Вывод основной программы

start program

Im= 35

6) max T: 271.081 K , left wall T: 269.674 K , right wall T: 271.081 K

8) heat dissipation: 750.000 W/m^2

9) heat flow: 750.075 W/m^2

10) balance error: 0.000099995654

left heat flow: 991.453 , right heat flow:-241.378 , term balance: 1.000099995654

end program

# Приложение 3

# Текстовый файл с исходными данными

0.015 - L

35 - Im

1.2 - k∆

50000 - Qv

6.5 - lamda\_0

-0.004 - kt

60 - alpha\_1

20 - alpha\_2

0 - T\_0

-20 - T\_e1

10 - T\_e2

8600 - po

380 - c

* 1. – eps