МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО

Физико-механический институт

Высшая школа прикладной математики и вычислительной физики

Научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)

Работу выполнил:

студент группы 5030301/20001

Медведев К.М.

Преподаватель:

к.т.н., доц. Плетнев А.А.

Санкт-Петербург

2024

Оглавление

[1 Постановка задачи 3](#_Toc170490745)

[1.1Физическая постановка задачи 3](#_Toc170490746)

[1.2 Математическая постановка задачи 3](#_Toc170490747)

[2 Метод решения 6](#_Toc170490748)

[2.1 Вывод сеточных уравнений для уравнения электропроводности 6](#_Toc170490749)

[2.2 Вычисление источника объёмного тепловыделения 8](#_Toc170490750)

[2.3 Вывод сеточных уравнений для уравнения теплопроводности 9](#_Toc170490751)

[3 Этапы программной реализации 10](#_Toc170490752)

[3.1 Алгоритм вычисления T(x) по заданному ϕm 10](#_Toc170490753)

[3.2 Алгоритм вычисления ϕm по Tm 10](#_Toc170490754)

[4 Результаты расчёта 11](#_Toc170490755)

[4.1 Расчёт показателей тепловыделения 11](#_Toc170490756)

[4.2 Построение зависимостей 11](#_Toc170490757)

[Приложение 13](#_Toc170490758)

# 1 Постановка задачи

## 1.1Физическая постановка задачи

Тонкая металлическая проволока длиной и диаметром , нагревается за счет протекания по ней постоянного электрического тока. Температура на концах зафиксирована и равна температуре окружающей среды . Отвод тепла от поверхности проволоки осуществляется конвекцией с коэффициентом теплоотдачи , постоянным по всей её длине.

Задача: определить, при какой разнице потенциалов на концах проволоки ее максимальная температура в стационарном тепловом режиме равна заданному с абсолютной точностью не менее 0,01 °C.

## 1.2 Математическая постановка задачи

Изображение выглядит как линия, диаграмма, График, Шрифт

Автоматически созданное описание

Рис 1. Расчетная область и граничные условия. Красной линией показан примерный профиль температуры

Двумерное стационарное уравнение теплопроводности с источником внутреннего объемного тепловыделения в цилиндрических координатах имеет вид:

Граничные условия:

Выполним ряд преобразований, которые сведут двумерное уравнение к одномерному. В дальнейшем будем решать именно его

Оценим продольный и поперечный перепады температур:

Так как диаметр мал, то Усредним значение температуры по поперечному сечению

Проинтегрируем уравнение (1) указанным образом. Тогда для последнего слагаемого

Используя граничные условия (4) и (5), окончательно преобразуем уравнение (1) к виду:

Знак осреднения в дальнейшем опустим. Новые граничные условия имеют теперь вид

Электрический потенциал вдоль проволоки зависит только от координаты х и описывается уравнением:

Граничные условия:

# 2 Метод решения

Используем неявный метод конечных разностей на равномерной сетке. Нелинейность, связанную с зависимостью 𝜎(𝑇), линеаризуем методом запаздывающих коэффициентов. Итерации по нелинейности ведём до достижения заданной точности – относительная невязка по Т не более

## 2.1 Вывод сеточных уравнений для уравнения электропроводности

Для построения разностной схемы для уравнения (8) введем в рассмотрение понятие элементарной сеточной ячейки, границы которой обозначены 𝑥𝑥𝑖 (рис. 2). При этом узлы сетки (за исключением граничных) должны быть расположены в центре сеточных ячеек.

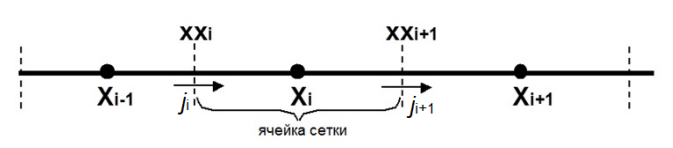


Рис 2. Шаблон аппроксимации уравнения электропроводности

Считаем, что в пределах одной сеточной ячейки коэффициент σ постоянен:

𝜎𝑖 = 𝑓(𝑇𝑖) Тогда плотность электрического тока на границе ячейки:

С учетом того, что сетка выбрана равномерной: 𝑑𝑥 = (𝑥𝑖 − 𝑥𝑖−1) = 𝑑𝑥𝑥 = 𝐶𝑜𝑛𝑠𝑡, неявная разностная аппроксимация уравнения (8) во внутренних узлах:

На граничных узлах с учётом условий (9) разностная аппроксимация будет иметь вид:

Представим разностное уравнение в виде:

Запишем выражения для коэффициентов. Отметим, что постоянную можно опустить.

Система уравнений получается трёхдиагональной и может быть решена с помощью метода прогонки

## 2.2 Вычисление источника объёмного тепловыделения

Источник объёмного тепловыделения вычисляется с помощью формулы (2)

Аппроксимация производной потенциала во внутренних узлах

Выедем формулу аппроксимации в граничных узлах

Умножим (11) на 4, вычтем из него (12) и разделим на h. Получим для левой границы

Аналогично для правой границы:

## 2.3 Вывод сеточных уравнений для уравнения теплопроводности

Совершенно аналогично получаются коэффициенты для уравнения теплопроводности

Вспоминая, граничные условия (7)

Система также является трёхдиагональной и может быть решена методом прогонки. Отметим также, что (15), (16), (17) могут быть вычислены заранее – до начала итеративного цикла по n.

# 3 Этапы программной реализации

## 3.1 Алгоритм вычисления T(x) по заданному ϕm

1. Считывание из файла всех исходных данных, в том числе ϕm
2. Вычисление шага равномерной сетки (**Im = 41**)
3. Присвоение начальных значений массиву
4. Вычисление прогоночных коэффициентов A, B, C для уравнения теплопроводности
5. Начало итеративного цикла по n
6. Вычисление значений 𝜎𝑖 = 𝑓() в узлах сетки (где n – номер итерации).
7. Вычисление прогоночных коэффициентов для уравнения электропроводности
8. Обращение к процедуре прогонки (вычисление элементов массива ).
9. Вычисление значений
10. Вычисление прогоночного коэффициента D для уравнения теплопроводности
11. Обращение к процедуре прогонки (вычисление элементов массива
12. Вычисление максимальной температуры и максимального значения относительной невязки -
13. Проверка условия выхода из цикла – невязка меньше 10-5
14. Переприсвоение массивов
15. Возврат к п.6

## 3.2 Алгоритм вычисления ϕm по Tm

1. Методом подбора определить ϕ1 и ϕ2 такие что Tm1 <Tm<Tm2
2. Методом половинного деления найти корень уравнения Tmax(ϕm) – Tm = 0, где функция Tmax(ϕm) вычисляется по предыдущему алгоритму

Программу, реализующую указанные алгоритмы можно найти в приложении 1

# 4 Результаты расчёта

С помощью написанной программы (приложение 1) определим: ϕm соответствующий заданному Tm, построим зависимости температуры, объёмного тепловыделения и проводимости от координаты. Также выполним расчёт основных показателей тепловыделения. Число узлов Im=41

## 4.1 Расчёт показателей тепловыделения

|  |  |
| --- | --- |
| Физическая величина | Расчётное значение |
| Объемное тепловыделение в проволоке, Qtot, | 7.99945754 Вт |
| Тепловой поток на боковой поверхности, | 7.45047622 Вт |
| Тепловой поток на торцах, | 0.54898132 Вт |
| Потенциал на правом конце, ϕm | 0.702745 В |
| Численное значение Tm | 80.0001 °C |
| Температурная невязка | -1.33\*10-15 |

## 4.2 Построение зависимостей

Чёрными точками обозначены узлы сетки

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Рис 3. Зависимость температуры от координаты

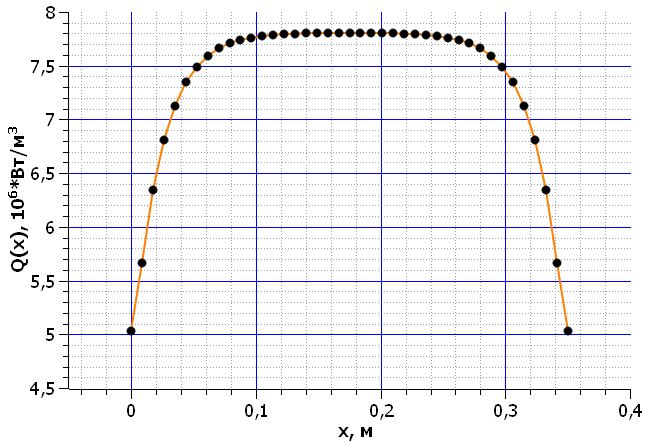


Рис 4. Зависимость объёмного тепловыделения от координаты

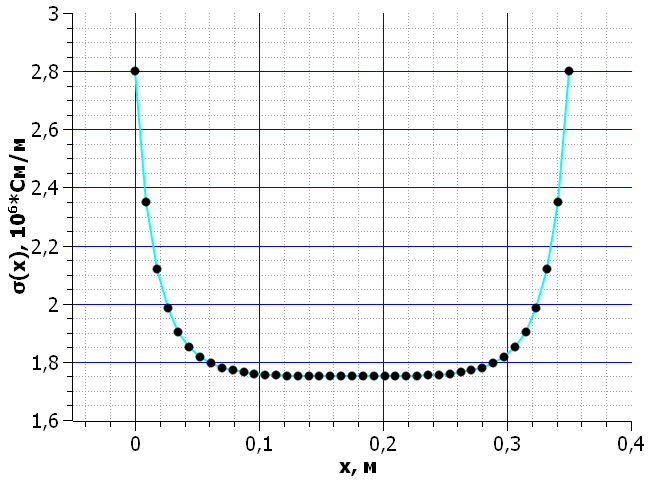


Рис 5. Зависимость удельной проводимости от координаты

# Приложение

Приложение 1

Код программы

1 Program PMF\_Practic

program PMF\_PRACTIC

use methods

implicit none

integer :: i, logic

integer, parameter :: Im = 41

real(8), parameter :: eps = 1e-5, pi = 4d0\*datan(1d0)

real(8) :: Q\_tot, potok\_s, potok\_t, c1, Tb, Tc, dX

real(8) :: L, d, T0, Tm, lambda, a, sig0, kt, a1, b1, phim !Входные данные

real(8), dimension(Im) :: At, Bt, Ct

real(8), dimension(Im) :: T, sig, phi, Q

real(8), dimension(12) :: input\_data

!Считываем входные данные

open(1, File="Input\_Data.txt")

do i = 1,12

read(1,\*) input\_data(i)

end do

close(1)

call get\_data(input\_data, L, d, T0, Tm, lambda, a, sig0, kt, logic, a1, b1, phim)

dX = L/(Im - 1d0) !Шаг сетки

!Вычисление At, Bt, Ct

At(2:Im-1) = lambda/dX\*\*2

Bt(2:Im-1) = -2d0\*(lambda/(dX\*\*2) + a\*2d0/d)

Ct(2:Im-1) = At(2:Im-1)

if (logic == 1) then

!Нахождение phim МПД

do while(abs(b1-a1) >= eps)

c1 = (a1+b1)/2d0

call get\_T(Im, At, Bt, Ct, T, phi, sig, T0, dX, d, a, kt, b1, sig0, eps)

Tb = maxval(T) - Tm

call get\_T(Im, At, Bt, Ct, T,phi, sig, T0, dX, d, a, kt, c1, sig0, eps)

Tc = maxval(T) - Tm

if (Tb\*Tc < 0d0) then

a1 = c1

else

b1 = c1

end if

end do

phim = (a1+b1)/2d0

else

!phim было введено. Ничего не делаем

end if

call get\_T(Im, At, Bt, Ct, T, phi, sig, T0, dX, d, a, kt, phim, sig0, eps)

!Формируем и интегрируем Q

Q=0d0

Q(1) = sig(1)\*((-3d0\*phi(1)+4d0\*phi(2)-phi(3))/(2d0\*dX))\*\*2

do i = 2, Im-1

Q(i) = sig(i)\*((phi(i+1)-phi(i-1))/(2d0\*dX))\*\*2

end do

Q(Im) = sig(Im)\*((3d0\*phi(Im)-4d0\*phi(Im-1)+phi(Im-2))/(2d0\*dX))\*\*2

Q\_tot = (prim(Im, Q, dX))\*pi\*(d\*\*2)/4d0

potok\_s = pi\*a\*D\*prim(Im, T-T0,dX)

potok\_t = 0.25d0\*pi\*(d\*\*2)\*lambda\*(T(2)-T(1) + T(Im-1) - T(Im))/dX !First order

!Вывод данных

10 format(f16.6, a, f16.6)

open(1, File="T(x).csv")

open(2, File="sig(x).csv")

open(3, File="Q(x).csv")

write(1,\*) "x", ";", "T"

write(2,\*) "x", ";", "sig"

write(3,\*) "x", ";", "Q"

do i = 1, Im

write(1,10) (0d0 + dX\*(i-1d0)), ';', T(i)

write(2,10) (0d0 + dX\*(i-1d0)), ';', sig(i)

write(3,10) (0d0 + dX\*(i-1d0)), ';', Q(i)

end do

write(\*,\*) "phim", phim

write(\*,\*) "numeric\_Tm", maxval(T)

write(\*,\*) "potok\_t", potok\_t

write(\*,\*) "potok\_s", potok\_s

write(\*,\*) "Q\_tot", Q\_tot

write(\*,\*) "nev", (potok\_s+potok\_t)/Q\_tot - 1d0

close(1)

close(2)

close(3)

pause

end program PMF\_PRACTIC

2 module methods

module methods

implicit none

contains

!Метод прогонки; F - вектор решения

!A, B, C, D – векторы прогоночных коэффициентов

!Im – число уравнений (нумерация начинается с Ib, стнандарт 1)

subroutine Progonka(Ib, Im, A, B, C, D, F)

integer, intent(in):: Im, Ib

real(8), dimension(1:Im), intent(in):: A, B, C, D

real(8), dimension(1:Im), intent(out) :: F

real(8), dimension(1:Im) :: alpha, beta

real(8) :: k0

integer :: i

F = 0d0

!Прямой ход

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

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

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

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

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

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

end do

end subroutine Progonka

subroutine get\_phi(Im, sig, phi) !Получаем значение потенциала на сетке

integer :: i, Im

real(8), dimension(Im) :: Ap, Bp, Cp, Dp, Fp, sig, phi

!Вектора Слау

Ap(2) = sig(2)\*sig(3)/(sig(2)+sig(3))

Cp(2) = 0d0

Dp(2) = 0d0

Bp(2) = -(sig(3)\*sig(2)/(sig(3) + sig(2)) + sig(2)\*sig(1)/(sig(2) + sig(1)))

do i = 3, Im-2

Ap(i) = sig(i)\*sig(i+1)/(sig(i)+sig(i+1))

Cp(i) = sig(i-1)\*sig(i)/(sig(i-1) + sig(i))

Bp(i) = -(Ap(i) + Cp(i))

Dp(i) = 0d0

end do

Ap(Im-1) = 0d0

Cp(Im-1) = (sig(Im-2)\*sig(Im-1)/(sig(Im-2) + sig(Im-1)))

Bp(Im-1) = -(Cp(Im-1) + sig(Im-1)\*sig(Im)/(sig(Im-1) + sig(Im)))

Dp(Im-1) = phi(Im)\*sig(Im-1)\*sig(Im)/(sig(Im-1) + sig(Im))

!Вызов метода прогонки

call Progonka(2,Im,Ap,Bp,Cp,Dp,Fp)

phi(2:Im-1) = Fp(2:Im-1)

end subroutine

subroutine get\_T(Im, At, Bt, Ct, T, phi, sig, T0, dX, d, a, kt, phim, sig0, eps)

integer :: i, Im

real(8) :: T0, dX, d, a, kt, phim, eps, sig0

real(8), dimension(Im) :: At, Bt, Ct, Dt, Ft, T, phi, sig, Tprev

T = T0

Tprev = T0

phi(Im) = phim

do while(1)

sig = sig0\*(1d0 + kt\*(Tprev-T0))\*\*(-1) !Проводимость

call get\_phi(Im, sig, phi) !Потенциал

!Вычисляем темппературу

Dt(2) = sig(2)\*((phi(3)-phi(1))/(2d0\*dX))\*\*2 + T0\*Ct(2) + 4d0\*a\*T0/d

do i = 3,Im-2

Dt(i) = sig(i)\*((phi(i+1)-phi(i-1))/(2d0\*dX))\*\*2 + 4d0\*a\*T0/d

end do

Dt(Im-1) = sig(Im-1)\*((phi(Im)-phi(Im-2))/(2d0\*dX))\*\*2 + T0\*At(Im-1) + 4d0\*a\*T0/d

call Progonka(2, Im, At, Bt, Ct, Dt, Ft)

T(2:Im-1) = Ft(2:Im-1)

T(1) = T0

T(Im) = T0

if(maxval(abs(T - Tprev))/maxval(abs(T)) < eps) exit

Tprev = T

end do

end subroutine

function prim(Im, Q, dX) result(Q\_tot)

integer :: Im

real(8) :: dX, Q\_tot

real(8), dimension(Im) :: Q

Q\_tot = sum(Q(2:Im-1))\*dX

end function

subroutine get\_data(input\_data, L, d, T0, Tm, lambda, a, sig0, kt, logic, a1, b1, phim)

integer :: logic

real(8) :: L, d, T0, Tm, lambda, a, sig0, kt, a1, b1, phim

real(8), dimension(12) :: input\_data

L = input\_data(1)

d = input\_data(2)

T0 = input\_data(3)

Tm = input\_data(4)

lambda = input\_data(5)

a = input\_data(6)

sig0 = input\_data(7)

kt = input\_data(8)

logic = int(input\_data(9))

a1 = input\_data(10)

b1 = input\_data(11)

phim = input\_data(12)

end subroutine

end module