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 метода конечных разностей

Работу выполнил:

студент группы 5030301/10002

Тугай В.В.

Преподаватель:

к.т.н., доц. Плетнев А.А.

Санкт-Петербург

2023

# Содержание

1. Физическая постановка задачи
2. Математическая постановка задачи
3. Метод решения
4. Тестовый расчет
5. Результаты решения задачи
6. Выводы
7. Приложение

# Физическая постановка задачи

Плоская неограниченная пластина из бронзы толщиной  60 см испытывает конвективный теплообмен с окружающей средой (с обеих сторон пластины интенсивность конвективного теплообмена одинакова). В начальный момент времени температура пластины постоянна во всем сечении и равна  500 °С. Температура окружающей среды  130 °С. Найти распределение температуры пластины в зависимости от координаты и времени для трех значений коэффициента конвективной теплоотдачи:

α1 = 35 Вт/(м²⋅К);   α2= 400 Вт/(м²⋅К);   α3= 25000 Вт/(м²⋅К).

*T*– температура пластины, K

*Tw*– температура на границе пластины, K

*Te*– температура окружающей среды, K

*T0*– начальная температура пластины, K

*τ* – время, с

*x*– координата, м

*δ* – толщина пластины, м

*q*– плотность теплового потока, Вт/м2

a – коэффициент конвективной теплоотдачи, Вт/(м2×К)

Материал пластины: бронза. Физические свойства материала и значения физический величин (Вариант-7):

*p*(1)=8600 кг/м2

*c*(2)=380 Дж/(К⋅кг)

*λ*=110 Вт/(м⋅К)

*δ*=0.6 м

*Te*=130 С˚

*T0*=500 С˚

(1) http://ru.solverbook.com/spravochnik/ximiya/plotnost/plotnost-bronzy/

(2) http://thermalinfo.ru/eto-interesno/tablitsy-udelnoj-teploemkosti-veshhestv

# Математическая постановка задачи

Diagram, engineering drawing

Description automatically generatedНачало координат – в центре пластины.  
Искомая функция – температура, которая зависит от двух переменных: координаты и времени

Плотность теплового потока (закон Фурье) : , где — коэффициент теплопроводности, Вт/(м⋅К)

Начальное условие (НУ):  
 – однородный профиль температуры

Граничные условия (ГУ):  
 – ГУ симметрии

– ГУ 3-го рода

Переход к безразмерным величинам:

— безразмерная координата ( X∈[0,1] )

— безразмерная избыточная температура( θ∈[0,1] )

— критерий Фурье (безразмерное время)

— число Био (безразмерный коэффициент теплоотдачи)

Уравнение теплопроводности в безразмерных величинах:

НУ и ГУ в безразмерных величинах:

# Метод решения

# Область непрерывного изменения аргумента (координата *X*) разбивается на конечное число интервалов, в пределах каждого интервала размещается узел, в которой задается значение искомой функции (температуры) для этого интервала. Совокупность узлов с упорядоченной нумерацией называется конечно-разностной сеткой. В нашем случае для безразмерных координат:

0 = 𝑋1 < 𝑋2 < 𝑋3 < ⋯ < 𝑋𝐼𝑚−1 < 𝑋𝐼𝑚 = 1, где *Im* – номер последнего узла.

Дискретизация выполняется как по оси *X*, так и по оси времени. В результате вместо непрерывной получаем дискретную функцию: 𝜃(𝑋, 𝐹𝑜) → (𝑋𝑖, 𝐹𝑜𝑛), где *i* – номер узла по координате *X*, n – номер узла по оси времени.

*Chart, box and whisker chart

Description automatically generated*

Проще всего строится равномерная сетка, когда шаг по координате постоянен:

В результате: 𝑋𝑖+1 = 𝑋𝑖 + ∆𝑋;

𝐹𝑜𝑛+1 = 𝐹𝑜𝑛 + ∆𝐹𝑜*.*

Далее выполняем аппроксимацию – замену производных, входящих в уравнение, конечно-разностными аналогами.

Формальная замена производных на равномерной сетке дает:

После замены производных уравнение теплопроводности, которое в безразмерны координатах записывается как , будет иметь вид:

Следовательно,

Это выражение справедливо для внутренних узлов сетки, т.е. для 𝑖 = 2 … (𝐼𝑚 − 1).

Из него видно, что значение не должно быть отрицательным (иначе увеличение приведет к уменьшению , что нефизично), откуда следует ограничение для шага по времени – условие устойчивости для явной схемы:

Выражения для крайних узлов, расположенных на границах расчетной области, можно получить из граничных условий (ГУ).

ГУ на левой границе:

, откуда и

ГУ на правой границе:

, откуда и

Блок-схема вычислительной программы



Начальные условия

𝜃1 = 1, 𝑖 = 1 … 𝐼𝑚

𝑖

Начало цикла по времени n = 1

Расчет шагов по времени и координате



**Начало**



Ввод исходных данных Bi, Fomax, Im и др.

Объявление массивов и переменных



Расчет температуры на слое n+1 во внутренних

узлах, формула (1)

Расчет температуры на слое n+1 в граничных узлах, формулы (3), (4)

Новое значение времени

*Fo = Fo +* *Fo*

Сохранение значений температуры, требуемых

для построения графиков

Fo  Fomax

Да

Нет

𝜃𝑛 = 𝜃𝑛+1, 𝑖 = 1 … 𝐼𝑚 ;

𝑖 𝑖

*n = n +* 1

**Конец**

# Тестовый расчет

# Для значений X=0, Fo=6, Bi=0.45 и Im=81, урезанная программа для тестирования (в ней осуществляется лишь расчет θ при заданных значениях X, Fo, Bi, Im) выдает значение θ=0.099; для X=0, Fo=3, Bi=0.5, Im=81: θ=0.297; для X=0, Fo=4, Bi=1.6, Im=81: θ=0.019; данные значения θ соответствуют значениям, полученным с помощью урезанной программы для нахождения θ методом Фурье (см. Приложение 2 и 6). Отсюда делаем вывод, что программа работает исправно.

# Результат решения задачи

# На рисунках 1 и 2 изображены зависимости температуры от времени при Im = 101 для трех разных сечений пластины (х=0; х=δ/4; х=δ/2) при Bi=0.0955 и Bi=68.2. На рисунках видно, что с увеличением значения параметра Bi уменьшается время, необходимое для достижения теплового равновесия. График зависимости температуры от времени для x=δ/2 испытывает резкий спад, а для х=δ/4 и х=0 графики асимптотически приближаемся к температуре, соответствующей состоянию теплового равновесия.

# На рисунках 3, 4 и 5 изображены зависимости температуры от координаты при Im = 101 для трех разных временных точек (t=0.1⋅Fomax; t=0.5⋅Fomax; t=0.9⋅Fomax) при Bi=0.0955, Bi=1.091 и Bi=68.2. На рисунках видно, что с увеличением значения параметра Bi разница температуры между центром пластины и ее краем возрастает, и при приближении к точке времени Fomax температура в точках пластины приближается к значениям температуры в момент теплового равновесия.

# Изображение выглядит как диаграмма Автоматически созданное описание

Рисунок 1 Рисунок 2

Зависимость температуры от Зависимость температуры от

времени при Bi=0.095 времени при Bi=68.2

Im=101 Im=101

**Изображение выглядит как диаграмма

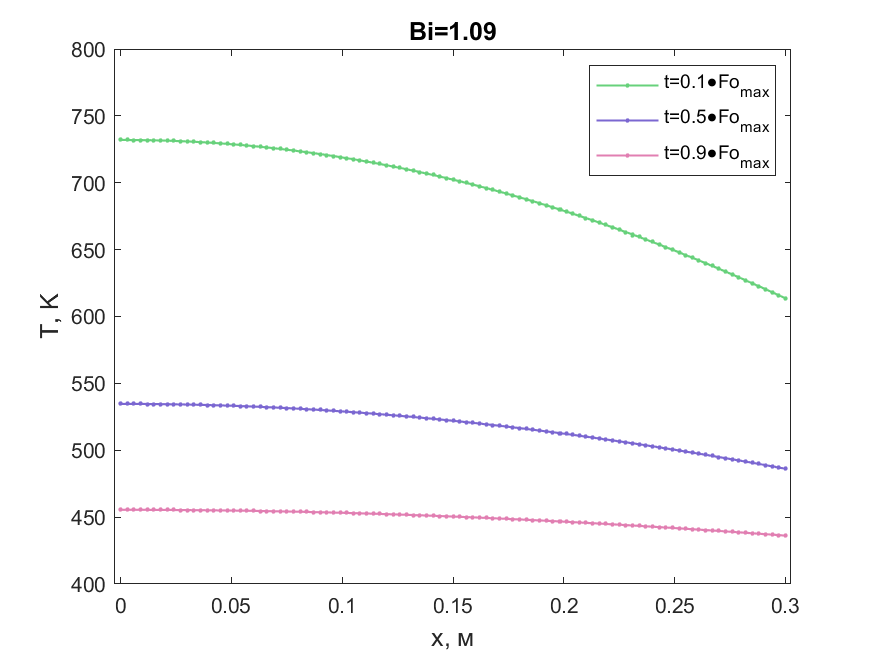
Автоматически созданное описание**

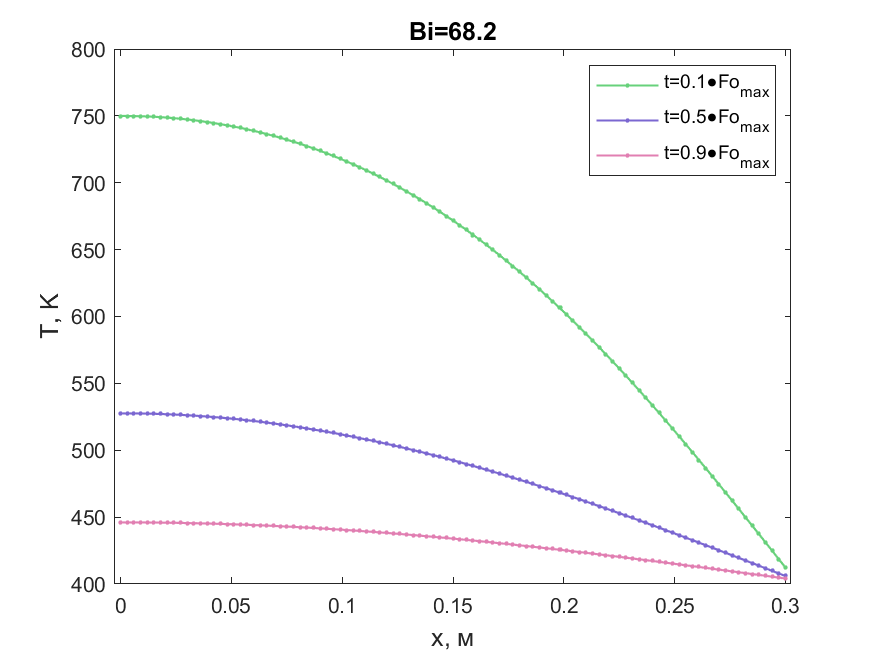
Рисунок 3 Рисунок 4

Зависимость температуры от Зависимость температуры от

координаты при Bi=0.095 координаты при Bi=1.09

Fomax = 55752 с Fomax = 7750 с

Im=101 Im=101

****

# Рисунок 5

Зависимость температуры от

координаты при Bi=68.2

Fomax = 2941 с

Im=101

# Выводы

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

# Для выявления влияния величины шагов по координате и времени на точность результатов была использована измененная тестовая программа для метода Фурье (см. Приложение 3). Были выбраны значения координаты X=0 и времени Fo=1(см. Приложение 4). Основная программа (Приложение 5) для различных Im выдала следующие результаты (таблица 1):

# Таблица 1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Im | Fo | TBi=0.095, K | TBi=1.09, K | TBi=68.2, K | ΔTBi=0.095, K | ΔTBi=1.09, K | ΔTBi=68.2, K | δBi=0.095, % | δBi=1.09, % | δBi=68.2, % |
| 11 | 1.00000000000000000000000000000000058 | 742.144 | 578.172 | 437.059 | 3.631 | 14.557 | 8.962 | 0.5 | 2.5 | 2 |
| 21 | 0.999999999999999999999999999999986326 | 743.863 | 585.128 | 441.245 | 1.912 | 7.601 | 4.776 | 0.3 | 1.3 | 1.1 |
| 41 | 1.00000000000000000000000000000005970 | 744.749 | 588.817 | 443.554 | 1.026 | 3.912 | 2.467 | 0.2 | 0.7 | 0.6 |
| 81 | 1.00000000000000000000000000000019356 | 745.199 | 590.716 | 444.767 | 0.576 | 2.013 | 1.254 | 0.1 | 0.3 | 0.3 |
| Фурье | 1.0 | 745.775 | 592.729 | 446.021 | - | - | - | - | - | - |

# Погрешность менее 1% для всех трех Bi достигается при Im=41, что является приемлемой точностью. Если посмотреть на рисунки 1 и 2, то можно понять, с чем связана то, что погрешности для Bi=0.095 получились самыми маленькими – на этих рисунках фиолетовыми чертами отмечены целые числа Fo. Для Bi=0.095 изменения значений температур для всех трех сечений пластины менее 50 К, а для Bi=68.2 от 300 до 350. Т. е. с увеличением параметра Bi растет и скорость изменения температуры, с чем и может быть связана самая маленькая погрешность для Bi=0.095.

# На рисунке 6 изображена зависимость температуры от времени на интервале от 0 до 300 секунд для Im=21 и Bi=68.2 при невыполнении условия устойчивости, т. к. был задан шаг на 10% больше допустимого (). Из рисунка видно, что значения температуры имеют колебательный характер с увеличивающейся амплитудой. Отсюда вывод: при нарушении условия устойчивости значения температуры не будут соответствовать действительности (явная конечно-разностная схема теряет устойчивость).

# 

# Рисунок 6

Зависимость температуры от

времени при Bi=0.095 и невыполнении

условия устойчивости

Im=21

# В программной реализации конечно-разностный метод решения оказался проще, нежели аналитический: в программе с аналитическим решением используется тройной цикл, код программы состоит из 291 строки, содержит в себе 3 функции с формулами разных физических величин, а также использует метод бисекции. Программа с методом конечных разностей содержит лишь двойной цикл, состоит из 233 строк кода и не имеет какие-либо дополнительные функций и не содержит другие методы.

# Приложение

# Основная программа

module methods

contains

character(len=20) function str(k) !function for translating integer to string

integer, intent(in) :: k

write (str, \*) k

str = adjustl(str)

end function

subroutine finit\_differences(*p*, *Te*, *T0*, *delta*, *alpha*, *lambda*, *cp*, *po*, *check*)

implicit none

real\*16 Te, T0, delta, lambda, cp, po, Bi, Fo, Fo\_max, st, delt\_X, delt\_Fo, one

real\*4 Bi\_wr

integer :: i, j, n, alpha, p, p\_Fo, Fo\_points, popul, check, t\_1

real\*16, allocatable, dimension (:,:) :: O\_n,T,tau\_x

real\*16, allocatable, dimension (:) :: points, tau\_points

integer, allocatable, dimension (:) :: points\_dot

Bi=(alpha\*delta)/(2\*lambda)

if (Bi<=0.5) then !Fo\_max based on Bi

write(13,'(F5.3,$)') Bi

write(\*,'(F5.3)') Bi

elseif (Bi>=10) then

write(13,'(F4.1,$)') Bi

write(\*,'(F4.1)') Bi

else

write(13,'(F4.2,$)') Bi

write(\*,'(F4.2)') Bi

end if

if (Bi<1.25) then

Fo\_max=3.11\*Bi\*\*(-0.81)

elseif (Bi>20) then

Fo\_max=1.1

else

Fo\_max=2.76\*Bi\*\*(-0.31)

end if

write(\*,'(A9,$)')'Fo\_max= '

write(\*,'(F12.6)')(Fo\_max\*cp\*po\*(delta\*\*2))/(4\*lambda)

allocate(points(6))

allocate(points\_dot(6))

points(1)=0

points(2)=0.5 !delta/4

points(3)=1 !delta/2

points(4)=Fo\_max\*0.1 !(Fo\_max\*0.1\*cp\*po\*(delta\*\*2))/(4\*lambda)

points(5)=Fo\_max\*0.5 !(Fo\_max\*0.5\*cp\*po\*(delta\*\*2))/(4\*lambda)

points(6)=Fo\_max\*0.9 !(Fo\_max\*0.9\*cp\*po\*(delta\*\*2))/(4\*lambda)

one=1

delt\_X=one/(p-1)

delt\_Fo=(delt\_X\*\*2)/2

if (check==1) delt\_Fo=delt\_Fo\*1.1

p\_Fo=int(Fo\_max/delt\_Fo)

write(13,\*) p, p\_Fo, alpha

allocate(O\_n(p+1,2))

allocate(T(p+1,2))

allocate(tau\_x(p\_Fo+1,2))

st=0

popul=0

do i=1,p

popul=popul+1

O\_n(i,2)=1

T(i,2)=(T0-Te)+Te

tau\_x(i,2)=(st\*delta)/2.0

st=st+delt\_X

do j=1,3

if ((points(j)>st-delt\_x) .and. (points(j)<st+delt\_x)) then

points\_dot(j)=popul

end if

end do

end do

tau\_x(1,1)=0

if (check==0) then

open(10, file='bin/res/Im='//trim(str(p))//'/alpha'//trim(str(alpha))//'\_temp\_t.txt')

open(101, file='bin/res/Im='//trim(str(p))//'/alpha'//trim(str(alpha))//'\_temp\_x.txt')

open(11, file='bin/res/Im='//trim(str(p))//'/alpha'//trim(str(alpha))//'\_time.txt')

open(21, file='bin/res/Im='//trim(str(p))//'/alpha'//trim(str(alpha))//'\_coord.txt')

else

open(10, file='bin/res/Im='//trim(str(p))//'\_10\_perc/alpha'//trim(str(alpha))//'\_temp\_t.txt')

open(101, file='bin/res/Im='//trim(str(p))//'\_10\_perc/alpha'//trim(str(alpha))//'\_temp\_x.txt')

open(11, file='bin/res/Im='//trim(str(p))//'\_10\_perc/alpha'//trim(str(alpha))//'\_time.txt')

open(21, file='bin/res/Im='//trim(str(p))//'\_10\_perc/alpha'//trim(str(alpha))//'\_coord.txt')

end if

Fo=delt\_Fo

n=0

tau\_x(1,1)=0

do n=1,p\_Fo+1 !time

do i=4,6

if ((points(i)>Fo-delt\_fo) .and. (points(i)<Fo+delt\_fo)) then

points\_dot(i)=n

end if

end do

O\_n(:,1)=O\_n(:,2)

T(:,1)=T(:,2)

do i=2,p-1

O\_n(i,2)=((delt\_Fo)/(delt\_X\*\*2))\*(O\_n(i+1,1)+O\_n(i-1,1))+O\_n(i,1)\*(1-(2\*(delt\_Fo)/(delt\_X\*\*2)))

T(i,2)=O\_n(i,2)\*(T0-Te)+Te

end do

O\_n(1,2)=O\_n(2,2)

O\_n(p,2)=(O\_n(p-1,2))/(1+(Bi\*delt\_X))

T(1,2)=O\_n(1,2)\*(T0-Te)+Te

T(p,2)=O\_n(p,2)\*(T0-Te)+Te

tau\_x(n+1,1)=(Fo\*cp\*po\*(delta\*\*2))/(4\*lambda) !real time

write(10,\*) T(1,1), T(points\_dot(2),1), T(points\_dot(3),1)

write(10,\*)

do i=4,6

if ((points(i)>Fo-delt\_fo/2) .and. (points(i)<Fo+delt\_fo/2)) then

do j=1,p

write(101,'(f18.10,$)') T(j,1)

end do

write(101,\*)

end if

end do

if ((abs(Fo-1)<0.000000001) .and. (check==0)) then

t\_1=n

write(\*,'(A10,$)')'For iter '

write(\*,'(i5,$)')t\_1

write(\*,'(A11,$)')' and time '

write(\*,\*) Fo, ': T(0,Fo=1)=', T(1,1)

end if

Fo=Fo+delt\_Fo

if (Fo>=Fo\_max) exit

end do

do i=1,p\_Fo

write(11,\*) tau\_x(i,1)

end do

do i=1,p

write(21,\*) tau\_x(i,2)

end do

allocate(tau\_points(100))

if (alpha/=400) then

if (check==0) then

open(19, file='bin/res/Im='//trim(str(p))//'/fo\_points\_alpha'//trim(str(alpha))//'.txt')

else

open(19, file='bin/res/Im='//trim(str(p))//'\_10\_perc/fo\_points\_alpha'//trim(str(alpha))//'.txt')

end if

do Fo\_points=1,100

tau\_points(Fo\_points)=(Fo\_points\*cp\*po\*(delta\*\*2))/(4\*lambda)

write(19,\*) Fo\_points, tau\_points(Fo\_points)

end do

end if

deallocate(tau\_points)

close(19)

write(20,\*) 1, points\_dot(2), points\_dot(3), points\_dot(4), points\_dot(5), points\_dot(6)

deallocate(points)

deallocate(points\_dot)

close(10)

close(11)

close(101)

deallocate(O\_n)

deallocate(T)

deallocate(tau\_x)

end subroutine

end module

program main

use methods

implicit none

real\*16 Te, T0, delta, lambda, cp, po, E

integer, allocatable, dimension (:) :: alpha\_n, p\_n

real\*16, allocatable, dimension (:,:) :: points

integer :: n, i, alpha, p, j, check=0

Te=130.0+273.15

T0=500.0+273.15

delta=0.6

lambda=110.0

cp=380.0

po=8600.0

allocate(p\_n(5))! grid size

allocate(alpha\_n(3))

alpha\_n(1)=35

alpha\_n(2)=400

alpha\_n(3)=25000

p\_n(1)=11

p\_n(2)=21

p\_n(3)=41

p\_n(4)=81

p\_n(5)=101

write(\*,\*)'start program'

do i=1,5

p=p\_n(i)

write(\*,\*) '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_'

write(\*,'(a3,$)') 'Im='

write(\*,'(i3)') p

write(\*,\*)

open(13, file='bin/res/Im='//trim(str(p))//'/bi\_iFORcoord\_iFORtime\_alpha.txt')

open(20, file='bin/res/Im='//trim(str(p))//'/points.txt')

do j=1,3

alpha=alpha\_n(j)

write(\*,'(A7,$)') 'Bi='

call finit\_differences(p, Te, T0, delta, alpha, lambda, cp, po,check)

write(\*,\*)

end do

close(20)

close(13)

write(\*,\*) 'this Im done'

end do

p=21

write(\*,\*) '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_'

write(\*,\*) 'Fluct'

write(\*,'(a3,$)') 'Im='

write(\*,'(i3)') p

write(\*,\*)

open(13, file='bin/res/Im='//trim(str(p))//'\_10\_perc/bi\_iFORcoord\_iFORtime\_alpha.txt')

open(20, file='bin/res/Im='//trim(str(p))//'\_10\_perc/points.txt')

check=1

j=3

alpha=alpha\_n(j)

write(\*,'(A7,$)') 'Bi='

call finit\_differences(p, Te, T0, delta, alpha, lambda, cp, po,check)

write(\*,\*)

close(20)

close(13)

write(\*,\*) 'this Im done'

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

end program

# Урезанная программа для тестирования

module methods\_test

contains

character(len=20) function str(k) !function for translating integer to string

integer, intent(in) :: k

write (str, \*) k

str = adjustl(str)

end function

subroutine finit\_differences(*X*, *Fo\_final*, *Bi*, *p*)

implicit none

real\*16 Bi, Fo, Fo\_final, delt\_X, delt\_Fo, one, X

integer :: i, j, n,p, x\_i, Fo\_i

real\*16, allocatable, dimension (:,:) :: O\_n

one=1

delt\_X=one/(p-1)

delt\_Fo=(delt\_X\*\*2)/2

allocate(O\_n(p+1,2))

do i=1,p

O\_n(i,2)=1

end do

Fo=delt\_Fo

Fo\_i=0

do n=1,100000 !time

O\_n(:,1)=O\_n(i,2)

Fo\_i=Fo\_i+1

do i=2,p-1

O\_n(i,2)=((delt\_Fo)/(delt\_X\*\*2))\*(O\_n(i+1,1)+O\_n(i-1,1))+O\_n(i,1)\*(1-(2\*(delt\_Fo)/(delt\_X\*\*2)))

end do

O\_n(1,2)=O\_n(2,2)

O\_n(p,2)=(O\_n(p-1,2))/(1+(Bi\*delt\_X))

Fo=Fo+delt\_Fo

if (Fo>Fo\_final) exit

end do

if (X==0) then

x\_i=1

elseif (X==1) then

x\_i=p

end if

write(\*,'(A13,$)')'Your theta = '

write(\*,\*) O\_n(x\_i,2)

end subroutine

end module

program main

use methods\_test

implicit none

real\*16 X, Fo, Bi

integer p

X=0

p=81

Fo=8.0

Bi=0.6

call finit\_differences(X, Fo, Bi, p)

end program

# Программа для нахождения значений аналитического решения

module functions\_temp\_find

contains

real\*16 Function f(Mu, Bi) result(res)

real\*16 :: Mu, Bi

res=(cos(Mu))/(sin(Mu))-(Mu/Bi)

end function

real\*16 Function A\_f(Mu) result(res)

real\*16 :: Mu

res=(2\*sin(Mu))/(Mu+sin(Mu)\*cos(Mu))

end function

real\*16 Function O\_f(Mu, A, Fo, X) result(res)

real\*16 :: Mu, A, Fo, X

res=A\*exp( (-1)\*Fo\*(Mu\*\*2))\*cos(Mu\*X)

end function

end module

module methods\_temp\_find

contains

real\*16 Function bis(n, E, Bi) result(res)

use functions\_temp\_find

real\*16 :: E, a\_Mu, b\_Mu, c\_Mu, f1, f2, check, pi=3.1415926535897932, Bi, delt, n\_r

integer :: n, i

n\_r=real(n, 16)

a\_Mu=pi\*(n\_r-1)

b\_Mu=pi\*(n\_r-0.5)

f1=f(b\_Mu, Bi)

do i=1,1000

c\_Mu=(a\_Mu+b\_Mu)/2.0

f2=f(c\_Mu, Bi)

check=f1\*f2

if (check>0) then

b\_Mu=c\_Mu

f1=f2

elseif (check<0) then

a\_Mu=c\_Mu

elseif (check==0) then

a\_Mu=c\_Mu

b\_Mu=c\_Mu

end if

delt=abs(a\_Mu-b\_Mu)

if (delt<E) exit

end do

res=c\_Mu

end function

subroutine Fourier(*X*, *Fo*, *Bi*, *E*)

use functions\_temp\_find

implicit none

real\*16 O, Bi, Mu, A, Fo, X, E, O\_n, check,te,t0

integer :: i, n

n=0

O=0

O\_n=0

check=1

do i=1,1000

n=n+1

Mu=bis(n, E, Bi)

A=A\_f(Mu)

O\_n=O\_f(Mu, A, Fo, X)

O=O+O\_n

check=abs(O)

if (check<E) exit

end do

Te=130.0+273.15

T0=500.0+273.15

write(\*,\*)'T=', O\*(T0-Te)+Te

end subroutine

end module

program main

use functions\_temp\_find

use methods\_temp\_find

implicit none

real\*16 X, Fo, Bi, E

X=0

Fo=1.0

Bi=0.095

call Fourier(X, Fo, Bi, E)

Bi=1.09

call Fourier(X, Fo, Bi, E)

Bi=68.2

call Fourier(X, Fo, Bi, E)

end program

# Вывод программы для нахождения значений аналитического решения

T= 745.774738704271646185167089264396912

T= 592.729044179203509135672440553685478

T= 446.021134057810731810182512579699915

# Вывод основной программы

start program

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Im= 11

Bi=0.095

Fo\_max= 55751.558487

For iter 200 and time 1.00000000000000000000000000000000058 : T(0,Fo=1)= 741.970453571908812728933843563418171

Bi=1.09

Fo\_max= 7749.675765

For iter 200 and time 1.00000000000000000000000000000000058 : T(0,Fo=1)= 577.414358238616446467216745342145394

Bi=68.2

Fo\_max= 2941.200297

For iter 200 and time 1.00000000000000000000000000000000058 : T(0,Fo=1)= 436.610264760338336307250259144593128

this Im done

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Im= 21

Bi=0.095

Fo\_max= 55751.558487

For iter 800 and time 0.999999999999999999999999999999986326 : T(0,Fo=1)= 743.821342985503067789188407483434645

Bi=1.09

Fo\_max= 7749.675765

For iter 800 and time 0.999999999999999999999999999999986326 : T(0,Fo=1)= 584.939860338920886561923165662689248

Bi=68.2

Fo\_max= 2941.200297

For iter 800 and time 0.999999999999999999999999999999986326 : T(0,Fo=1)= 441.124910647887847465577838724250744

this Im done

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Im= 41

Bi=0.095

Fo\_max= 55751.558487

For iter 3200 and time 1.00000000000000000000000000000005970 : T(0,Fo=1)= 744.739081968787766492698339820658862

Bi=1.09

Fo\_max= 7749.675765

For iter 3200 and time 1.00000000000000000000000000000005970 : T(0,Fo=1)= 588.770649623202096500663626800418669

Bi=68.2

Fo\_max= 2941.200297

For iter 3200 and time 1.00000000000000000000000000000005970 : T(0,Fo=1)= 443.522911070247045613878193143700471

this Im done

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Im= 81

Bi=0.095

Fo\_max= 55751.558487

For iter 12800 and time 1.00000000000000000000000000000019356 : T(0,Fo=1)= 745.196277961595048104354459430171309

Bi=1.09

Fo\_max= 7749.675765

For iter 12800 and time 1.00000000000000000000000000000019356 : T(0,Fo=1)= 590.704853087183261577471682213037037

Bi=68.2

Fo\_max= 2941.200297

For iter 12800 and time 1.00000000000000000000000000000019356 : T(0,Fo=1)= 444.759435791052694558652358643961707

this Im done

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Im=101

Bi=0.095

Fo\_max= 55751.558487

For iter 20000 and time 0.999999999999999999999999999999981896 : T(0,Fo=1)= 745.287591632043998151999120511690536

Bi=1.09

Fo\_max= 7749.675765

For iter 20000 and time 0.999999999999999999999999999999981896 : T(0,Fo=1)= 591.093262444409058255567998212695837

Bi=68.2

Fo\_max= 2941.200297

For iter 20000 and time 0.999999999999999999999999999999981896 : T(0,Fo=1)= 445.009832737797274926862333961463031

this Im done

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fluct

Im= 21

Bi=68.2

Fo\_max= 2941.200297

this Im done

end program

# Вывод урезанной программы для метода Фурье

Input X: 0

Input Fo: 6

Input Bi: 0.45

O= 0.102553513049214762797759192916365133

Input X: 0

Input Fo: 3

Input Bi: 0.5

O= 0.297449345355111229166080552273732902

Input X: 0

Input Fo: 4

Input Bi: 1.6

O= 1.98441311715528375238824378099299295E-0002