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

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:

$$\alpha_1 = 35 \text{ Вт/(м}^2\cdot\text{К)}; \quad \alpha_2 = 400 \text{ Вт/(м}^2\cdot\text{К)}; \quad \alpha_3 = 25000 \text{ Вт/(м}^2\cdot\text{К)}.$$

T– температура пластины, К
T_w– температура на границе пластины, К
T_e– температура окружающей среды, К
T_0– начальная температура пластины, К
τ – время, с
x– координата, м
δ – толщина пластины, м
q– плотность теплового потока, Вт/м2
а – коэффициент конвективной теплоотдачи, Вт/(м2×К)

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

$p^{(1)}$=8600 кг/м^2

$c^{(2)}$=380 Дж/(К·кг)

λ=110 Вт/(м·К)

δ=0.6 м

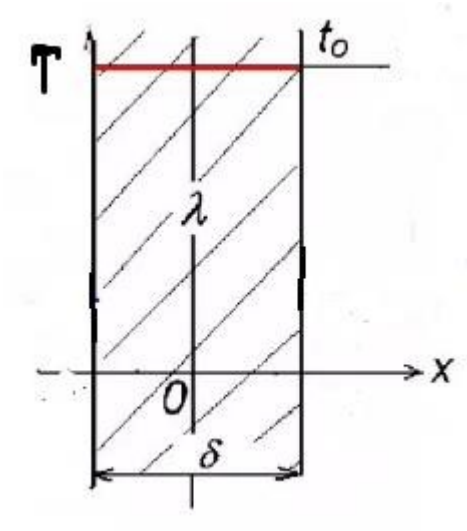
T_e=130 С˚

T_0=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



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

$$T = T(\tau, x)$$

Плотность теплового потока (закон Фурье) : $q = -\lambda \frac{\partial T}{\partial n}$, или $q = -\lambda\, \mathrm{grad T} = -\lambda \nabla T$, где λ — коэффициент теплопроводности, Вт/(м·К)

Начальное условие (НУ): $T(0, x) = T_0$ – однородный профиль температуры

Граничные условия (ГУ): $\frac{\partial T}{\partial x}(\tau, 0) = 0$ – ГУ симметрии

$$q|_{x=\frac{\delta}{2}} = -\lambda \frac{\partial T}{\partial x}\Big|_{x=\frac{\delta}{2}} = \alpha(T_w - T_e)$$ – ГУ 3-го рода

Переход к безразмерным величинам:

$X = \frac{2x}{\delta}$ — безразмерная координата ($X \in [0,1]$)

$\theta = \frac{T - T_e}{T_0 - T_e}$ — безразмерная избыточная температура($\theta \in [0,1]$)

$Fo = \frac{\tau}{\left(\frac{\delta}{2}\right)^2} \frac{\lambda}{cp}$ — критерий Фурье (безразмерное время)

$Bi = \frac{\alpha \delta}{2\lambda}$ — число Био (безразмерный коэффициент теплоотдачи)

Уравнение теплопроводности в безразмерных величинах:

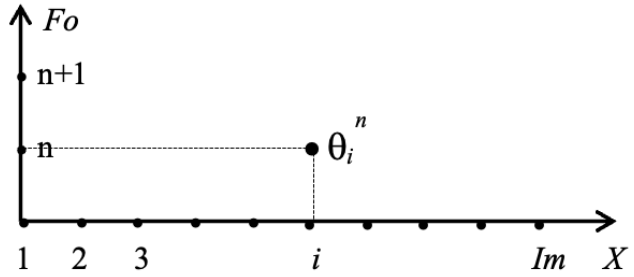
$$\frac{\partial \theta}{\partial Fo} = \frac{\partial^2 \theta}{\partial X^2}$$

НУ и ГУ в безразмерных величинах: $\begin{cases} \theta(0, X) = 1 \\ \frac{\partial \theta}{\partial X}(Fo; 0) = 0 \\ -\frac{\partial \theta}{\partial X}(Fo; 1) = Bi\theta \end{cases}$

Метод решения

Область непрерывного изменения аргумента (координата X) разбивается наконечное 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: $0 = X_1 < X_2 < X_3 < \cdots < X_{Im-1} < X_{Im} = 1$, где Im – номер последнего узла.

Дискретизация выполняется как по оси X, так и по оси времени. В результате вместо непрерывной получаем дискретную функцию: $\theta(X, Fo) \rightarrow \theta_i^n(X_i, Fo^n)$, где i – номер узла по координате X, n – номер узла по оси времени.



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

$$\Delta X = \frac{1}{\mathrm{Im} - 1}$$

В результате:

$$X_{i+1} = X_i + \Delta X;$$
$$Fo^{n+1} = Fo^n + \Delta Fo.$$

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

Формальная замена производных на равномерной сетке дает:

$$\left.\frac{\partial \theta}{\partial Fo}\right|_i^n \approx \frac{\theta_i^{n+1} - \theta_i^n}{\Delta Fo}; \quad \left.\frac{\partial^2 \theta}{\partial X^2}\right|_i^n \approx \frac{\theta_{i+1}^n - 2\,\theta_i^n + \theta_{i-1}^n}{\Delta X^2}$$

После замены производных уравнение теплопроводности, которое в безразмерны координатах записывается как $\frac{\partial \theta}{\partial Fo} = \frac{\partial^2 \theta}{\partial X^2}$, будет иметь вид:

$$\frac{\theta_i^{n+1} - \theta_i^n}{\Delta Fo} = \frac{\theta_{i+1}^n - 2\,\theta_i^n + \theta_{i-1}^n}{\Delta X^2}$$

Следовательно,

$$\theta_i^{n+1} = \frac{\Delta Fo}{\Delta X^2}(\theta_{i+1}^n + \theta_{i-1}^n) + \theta_i^n\left(1 - \frac{2\Delta Fo}{\Delta X^2}\right)$$

Это выражение справедливо для внутренних узлов сетки, т.е. для $i = 2 \ldots (Im - 1)$.

Из него видно, что значение $\left(1 - \frac{2\Delta Fo}{\Delta X^2}\right)$ не должно быть отрицательным (иначе увеличение θ_i^n приведет к уменьшению θ_i^{n+1}, что нефизично), откуда

следует ограничение для шага по времени – условие устойчивости для явной схемы:

$$\Delta Fo \leq \frac{\Delta X^2}{2}.$$

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

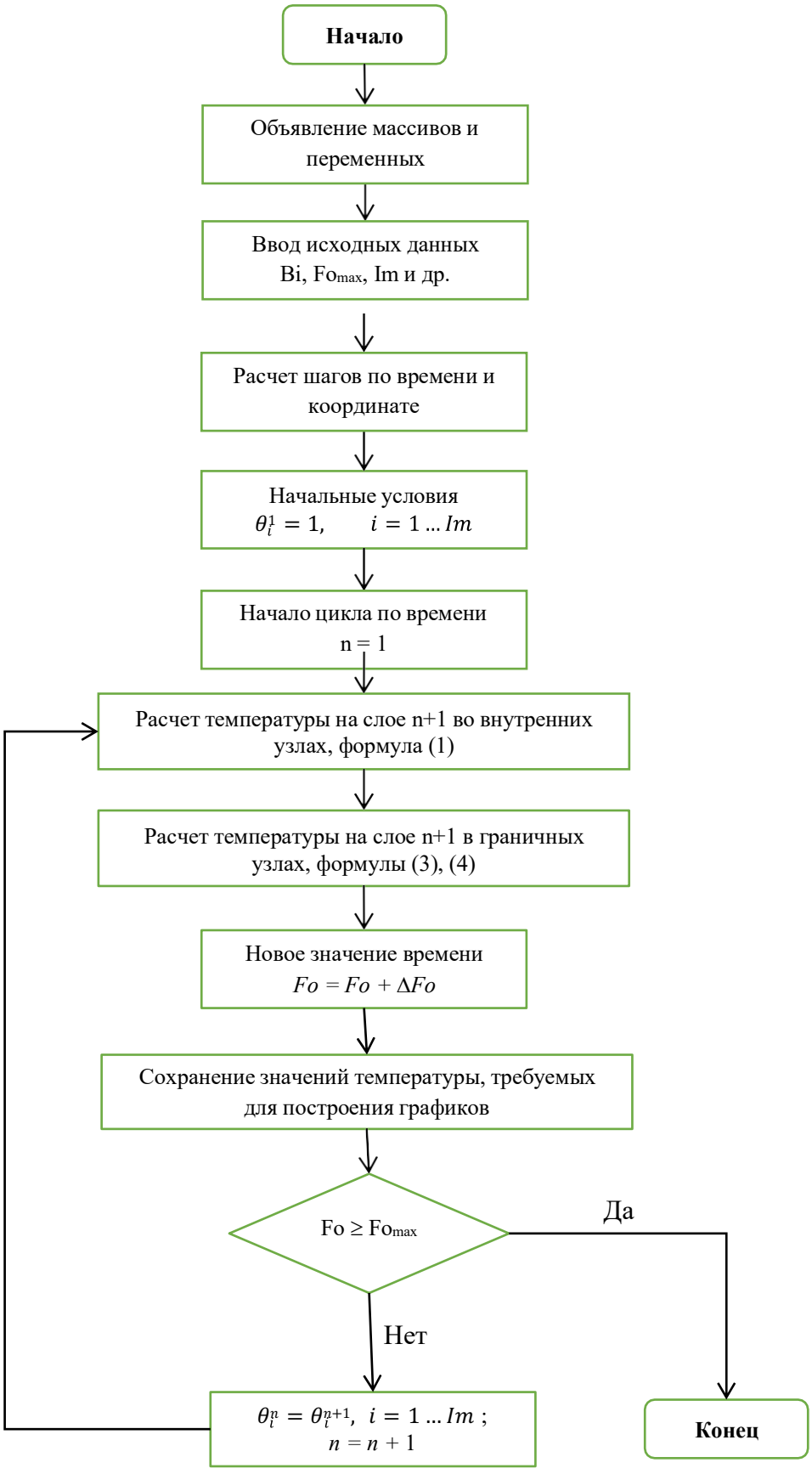
ГУ на левой границе:

$\frac{\partial \theta}{\partial X}(Fo;0) = 0$, откуда $\frac{\theta_2^{n+1} - \theta_1^{n+1}}{\Delta X^2} = 0$ и $\theta_1^{n+1} = \theta_2^{n+1}$

ГУ на правой границе:

$-\frac{\partial \theta}{\partial X}(Fo;1) = Bi\theta$, откуда $-\frac{\theta_{Im}^{n+1} - \theta_{Im-1}^{n+1}}{\Delta X} = Bi\theta_{Im}^{n+1}$ и $\theta_{Im}^{n+1} = \frac{\theta_{Im-1}^{n+1}}{1+Bi\Delta X}$

Блок-схема вычислительной программы



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 (x=0; x=δ/4; x=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 x=δ/4 и x=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 ($t=0.1\cdot Fo_{max}$; $t=0.5\cdot Fo_{max}$; $t=0.9\cdot Fo_{max}$) 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 Fo_{max} температура в точках пластины приближается к значениям температуры в момент теплового равновесия.

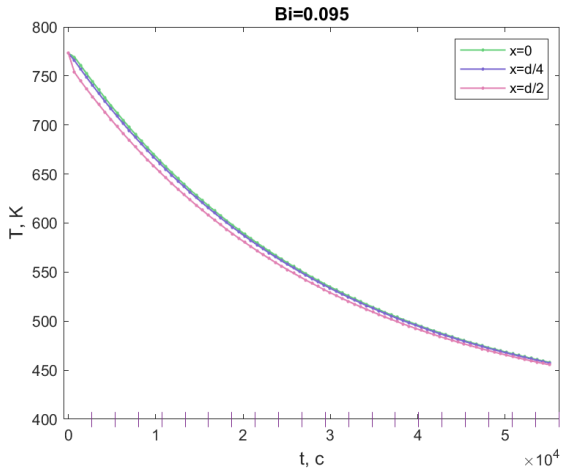


Рисунок 1
Зависимость температуры от времени при Bi=0.095 Im=101

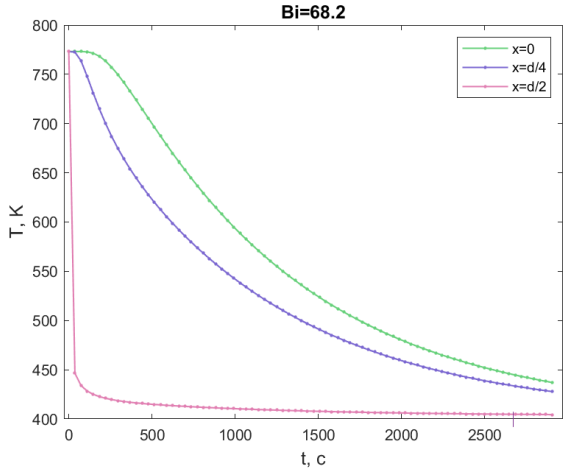


Рисунок 2
Зависимость температуры от времени при Bi=68.2 Im=101

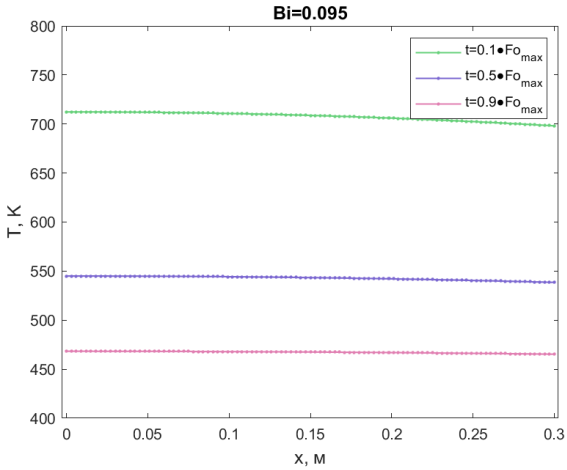


Рисунок 3
Зависимость температуры от координаты при Bi=0.095
Fo_{max} = 55752 с
Im=101

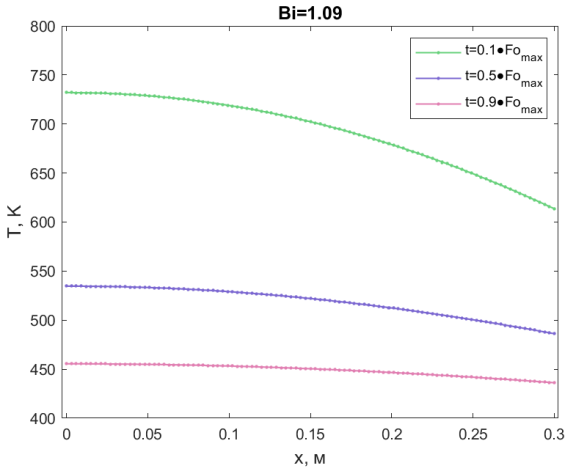


Рисунок 4
Зависимость температуры от координаты при Bi=1.09
Fo_{max} = 7750 с
Im=101

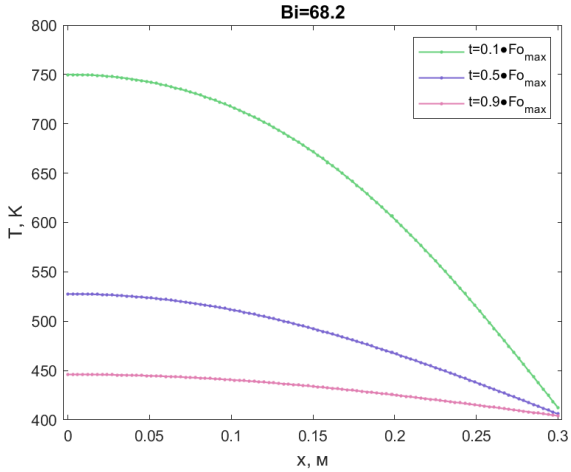


Рисунок 5
Зависимость температуры от координаты при Bi=68.2
Fo_{max} = 2941 с
Im=101

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	$T_{Bi=0.095}$, K	$T_{Bi=1.09}$, K	$T_{Bi=68.2}$, K	$\Delta T_{Bi=0.095}$, K	$\Delta T_{Bi=1.09}$, K	$\Delta T_{Bi=68.2}$, K	$\delta_{Bi=0.095}$, %	$\delta_{Bi=1.09}$, %	$\delta_{Bi=68.2}$, %
11	1.0000000000000000000000000000000058	742.144	578.172	437.059	3.631	14.557	8.962	0.5	2.5	2
21	0.9999999999999999999999999999999986326	743.863	585.128	441.245	1.912	7.601	4.776	0.3	1.3	1.1
41	1.0000000000000000000000000000005970	744.749	588.817	443.554	1.026	3.912	2.467	0.2	0.7	0.6
81	1.0000000000000000000000000000019356	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 ($\Delta Fo = \frac{\Delta X^2}{2} * 1.1$). Из рисунка видно, что значения температуры имеют колебательный характер с увеличивающейся амплитудой. Отсюда вывод: при нарушении условия устойчивости $\left(\Delta Fo \leq \frac{\Delta X^2}{2}\right)$ значения температуры не будут соответствовать действительности (явная конечно-разностная схема теряет устойчивость).

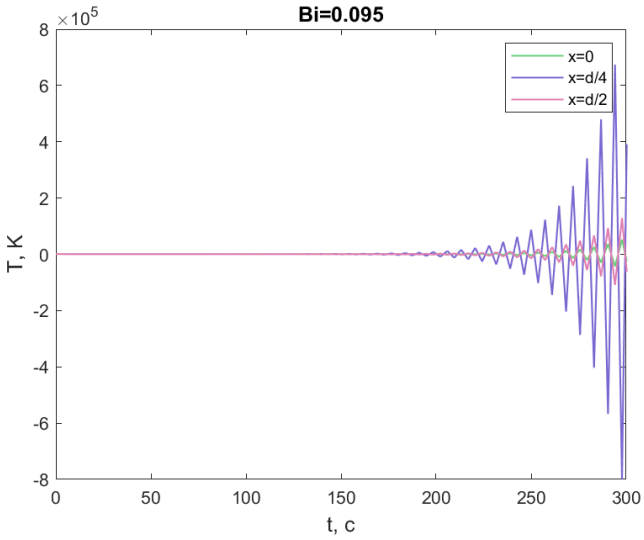


Рисунок 6
Зависимость температуры от
времени при Bi=0.095 и невыполнении
условия устойчивости
Im=21

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

1) 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
```

2) 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
```

3) 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
```

4) Вывод программы для нахождения значений аналитического решения

```
 T=   745.774738704271646185167089264396912
 T=   592.729044179203509135672440553685478
 T=   446.021134057810731810182512579699915
```

5) Вывод основной программы

```
start program
 ________________________
Im= 11

    Bi=0.095
 Fo_max= 55751.558487
 For iter   200  and time    1.000000000000000000000000000000000058       :    T(0,Fo=1)=
741.970453571908812728933843563418171

    Bi=1.09
 Fo_max=  7749.675765
 For iter   200  and time    1.000000000000000000000000000000000058       :    T(0,Fo=1)=
577.414358238616446467216745342145394

    Bi=68.2
```

```
 Fo_max=  2941.200297
 For iter   200  and time    1.000000000000000000000000000000058       :    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    1.000000000000000000000000000005970       :    T(0,Fo=1)=
744.739081968787766492698339820658862

    Bi=1.09
 Fo_max=  7749.675765
 For iter  3200  and time    1.000000000000000000000000000005970       :    T(0,Fo=1)=
588.770649623202096500663626800418669

    Bi=68.2
 Fo_max=  2941.200297
 For iter  3200  and time    1.000000000000000000000000000005970       :    T(0,Fo=1)=
443.522911070247045613878193143700471

 this Im done
 ________________________
Im= 81

    Bi=0.095
 Fo_max= 55751.558487
 For iter 12800  and time    1.000000000000000000000000000019356       :    T(0,Fo=1)=
745.196277961595048104354459430171309

    Bi=1.09
 Fo_max=  7749.675765
 For iter 12800  and time    1.000000000000000000000000000019356       :    T(0,Fo=1)=
590.704853087183261577471682213037037
```

```
    Bi=68.2
 Fo_max=  2941.200297
 For iter 12800  and time    1.000000000000000000000000000019356       :    T(0,Fo=1)=
444.75943579105269455865235864396170 7

 this Im done
 _____________________
Im=101

    Bi=0.095
 Fo_max= 55751.558487
 For iter 20000  and time   0.999999999999999999999999999999981896       :    T(0,Fo=1)=
745.28759163204399815199912051169053 6

    Bi=1.09
 Fo_max=  7749.675765
 For iter 20000  and time   0.999999999999999999999999999999981896       :    T(0,Fo=1)=
591.09326244440905825556799821269583 7

    Bi=68.2
 Fo_max=  2941.200297
 For iter 20000  and time   0.999999999999999999999999999999981896       :    T(0,Fo=1)=
445.00983273779727492686233396146303 1

 this Im done
 _____________________
 Fluct
Im= 21

    Bi=68.2
 Fo_max=  2941.200297

 this Im done
 end program
```

6) Вывод урезанной программы для метода Фурье

```
Input X: 0
Input Fo: 6
Input Bi: 0.45
 O=  0.10255351304921476279775919291636513 3

Input X: 0
Input Fo: 3
Input Bi: 0.5
 O=  0.29744934535511122916608055227373290 2

Input X: 0
Input Fo: 4
Input Bi: 1.6
 O=   1.98441311715528375238824378099299295E-0002
```