Министерство образования РФ

Государственное образовательное учреждение

высшего профессионального образования

“Тверской государственный университет”

Факультет прикладной математики и кибернетики

ВЫПУСКНАЯ РАБОТА БАКАЛАВРА

Направление: «*Прикладная математика и информатика»*

Специализация: *Бакалавриат*

Тема: *«Численный анализ тепловых процессов с учетом фазового перехода с использованием метода сеток»*

Автор:

студент 44 группы

Доронин Виталий Евгеньевич

Тверь – 2018

Оглавление

Введение

Первой работой в данной области считают статью Г. Ламе и Б. П. Клапейрона «Об отвердевании охлаждающегося жидкого шара» 1831 года, в которой было установлено, что толщина твердой фазы, образующейся при затвердевании однородной жидкости, пропорциональна $\sqrt{t}$. Значительно позже в 1889 году австрийский физик и математик Йозеф Стефан опубликовал четыре статьи, посвященные задачам с фазовыми переходами. Впоследствии задачи данного класса с подвижными межфазными границами стали называть задачами Стефана. В своих работах он сформулировал и решил задачи, определяющие процессы теплопроводности и диффузии для однофазной или двухфазной областей. Кроме того Й. Стефан сформулировал уравнение теплового баланса на границе раздела фаз с учетом скрытой теплоты, и теперь подобные условия сопряжения фаз принято называть условиями Стефана.

В настоящее время приближение Стефана используется при решении узкого круга задач: затвердевание чистых металлов, рост монокристаллов, при исследовании прочности льда, при моделировании 3D-печати.

Постановка задачи

При изменении температуры тела может происходить изменение его физического состояния, в частности при переходе температуры через точку плавления – переход из твёрдой фазы в жидкую или наоборот. На поверхности фазового перехода всё время сохраняется постоянная температура. При движении поверхности фазового перехода происходит выделение скрытой теплоты плавления (затвердевания).

Математической моделью, описывающей процесс фазового перехода, является задача с подвижной границей (задача Стефана).

Рассмотрим процесс замерзания воды, при котором температура фазового переходы равна нулю. Будем рассматривать массу воды $x \geq 0$, ограниченную с одной стороны плоскостью $x = 0$. В начальный момент $t = 0$ вода обладает постоянной температурой $c > 0$. Если на поверхности $x = 0$ всё время поддерживается постоянная температура $c_1 < 0$, то граница замерзания $x = \xi$ будет со временем проникать вглубь жидкости.

Задача о распределении температуры при наличии фазового перехода и о скорости движения границы раздела фаз (например, внутри замерзающей воды) сводится к решению уравнений

$$\begin{cases} \frac{\partial U_1}{\partial t} = {a^2}_1 \frac{\partial^2 U_1}{\partial x^2}, \ 0 < x < \xi \\ \frac{\partial U_2}{\partial t} = {a^2}_2 \frac{\partial^2 U_2}{\partial x^2}, \ \xi < x < \infty \end{cases} \qquad (1)$$

с дополнительными условиями

$$\begin{cases} u_1 = c_1, \ x = 0 \\ u_2 = c_2, \ t = 0 \end{cases} \qquad (2)$$

и условиями на границе замерзания

$$u_1 = u_2 = 0\,, \ x = \xi \qquad (3)$$

$$k_1 \frac{\partial U_1}{\partial x}\Big|_{x=\xi} - k_2 \frac{\partial U_2}{\partial x}\Big|_{x=\xi} = \lambda\rho \frac{\partial \xi}{\partial t}, \quad (4)$$

где $k_1, {a^2}_1, k_2, {a^2}_2$ – коэффициенты теплопроводности и температуропроводности твердой и, соответственно, жидкой фаз.

Задачу (1) – (4) часто называют задачей Стефана, задачей о фазовом переходе или задачей о промерзании.

Решение

Предположим, что имеется две фазы 1, 2 с коэффициентами теплоёмкости $c_1(U)$, $c_2(U)$, и теплопроводности $k_1(U)$, $k_2(U)$.

Имеем слой конечной толщины, $0 < x < L$. Требуется найти поле температур $u(x,t)$ из условий:

$$\mathrm{c}(U)\rho(U)\frac{\partial U}{\partial t} = \frac{\partial}{\partial x}\left(k(U)\frac{\partial U}{\partial x}\right) \quad (1)$$

$$\frac{\partial U}{\partial x}|_{x=0} = \alpha(t) \quad (2)$$

$$\frac{\partial U}{\partial x}|_{x=L} = 0 \quad (3)$$

Аппроксимируем производные с помощью их разностных аналогов и будем использовать чисто неявную схему:

$$\frac{\partial U}{\partial t} \approx \frac{U_{ij} - U_{i,j-1}}{\tau}$$ – левая разностная производная

$$\frac{\partial^2 U}{\partial x^2} \approx \frac{U_{i+1,j} - 2U_{ij} + U_{i-1,j}}{h^2}$$ – вторая разностная производная

Получим :

$$\mathrm{c}(U)\rho(U)\frac{U_{ij} - U_{i,j-1}}{\tau} \approx \frac{1}{h}\left(k(U_{i+1,j})\frac{U_{i+1,j} - U_{ij}}{h} - k(U_{i-1,j})\frac{U_{ij} - U_{i-1,j}}{h}\right)$$

$$\frac{U_{1,j} - U_{0,j}}{h} \approx \alpha(\tau_j)$$

$$\frac{U_{N,j} - U_{N-1,j}}{h} \approx 0$$

Решение будем искать с помощью метода прогонки.

Заменим U_{ij} на $y_i \Rightarrow U_{i+1,j} = y_{i+1}$; $U_{i-1,j} = y_{i-1}$.

$$\mathrm{c}(U)\rho(U)\frac{y_i - U_{i,j-1}}{\tau} = \frac{1}{h}\left(k(U_{i+1,j})\frac{y_{i+1} - y_i}{h} - k(U_{i-1,j})\frac{y_i - y_{i-1}}{h}\right)$$

Перенесём $c(U)\rho(U)\frac{U_{i,j-1}}{\tau}$ в правую часть уравнения, всё остальное – в левую.

$$c(U)\rho(U)\frac{y_i}{\tau} - \frac{k(U_{i+1,j})y_{i+1} - k(U_{i+1,j})y_i}{h^2} + \frac{k(U_{i-1,j})y_i - k(U_{i-1,j})y_{i-1}}{h^2} = \\ = c(U)\rho(U)\frac{U_{i,j-1}}{\tau}$$

$$y_i\left(\frac{c(U)\rho(U)}{\tau} + \frac{k(U_{i+1,j}) + k(U_{i-1,j})}{h^2}\right) + y_{i-1}\left(\frac{-k(U_{i-1,j})}{h^2}\right) \\ + y_{i+1}\left(\frac{-k(U_{i+1,j})}{h^2}\right) = c(U)\rho(U)\frac{U_{i,j-1}}{\tau}$$

Выражения в скобках заменим на коэффициенты:

$$A_i = \frac{-k(U_{i-1,j})}{h^2};\ B_i = \frac{-k(U_{i+1,j})}{h^2};$$

$$C_i = \frac{c(U)\rho(U)}{\tau} + \frac{k(U_{i+1,j}) + k(U_{i-1,j})}{h^2};$$

$$F_i = c(U)\rho(U)\frac{U_{i,j-1}}{\tau}$$

Имеем:

$$A_i y_{i-1} + B_i y_{i+1} + C_i y_i = F_i$$

Будем находить слои итерационным методом с помощью линейной функции:

$$y_{i-1} = \alpha_i y_i + \beta_i \Rightarrow y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}$$

Подставляем y_{i-1} в уравнение:

$$A_i(\alpha_i y_i + \beta_i) + B_i y_{i+1} + C_i y_i - F_i = 0$$

$$y_i(A_i\alpha_i + C_i) + B_i y_{i+1} + (A_i\beta_i - F_i) = 0$$

Подставим $y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}$:

$$\alpha_{i+1} y_{i+1} A_i \alpha_i + A_i\alpha_i\beta_{i+1} + C_i\alpha_{i+1}y_{i+1} + C_i\beta_{i+1} + B_i y_{i+1} + (A_i\beta_i - F_i) = 0$$

$$y_{i+1}[\alpha_{i+1}(A_i\alpha_i + C_i) + B_i] + [\beta_{i+1}(C_i + A_i\alpha_i) + (A_i\beta_i - F_i)] = 0$$

$y_{i+1} \neq 0 \Rightarrow$ выражения в квадратных скобках должны быть равны 0.

$$\alpha_{i+1}(A_i\alpha_i + C_i) + B_i = 0; \ \beta_{i+1}(C_i + A_i\alpha_i) + (A_i\beta_i - F_i) = 0$$

$$\alpha_{i+1} = -\frac{B_i}{A_i\alpha_i + C_i}; \ \beta_{i+1} = -\frac{A_i\beta_i - F_i}{A_i\alpha_i + C_i};$$

Для граничных условий:

$$U_{1,j} - U_{0,j} = h\alpha(\tau_j) \Rightarrow U_{0,j} = -h\alpha(\tau_j) + U_{1,j}$$

$$U_{N,j} - U_{N-1,j} = 0 \Rightarrow U_{N,j} = U_{N-1,j}$$

Расчеты на ЭВМ

Нахождение численного решения

Для численного значения интеграла будем использовать R – язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом в рамках проекта GNU. R широко используется как статистическое программное обеспечение для анализа данных и фактически стал стандартом для статистических программ.

Для расчётов будем использовать начальные значения:

$k_1(U) = 2.3$ #Теплопроводность льда
$k_2(U) = 0.58$ #Теплопроводность воды
$c_1(U) = 2000$ #Теплоёмкость льда
$c_2(U) = 4195$ #Теплоёмкость воды
h = 0.00035 #шаг по x

$$a = \sqrt{\frac{k(U)}{c(U)}} = 0.0006910293$$

$\tau = \frac{h^2}{a^2} = 0.2565329$ #шаг по времени

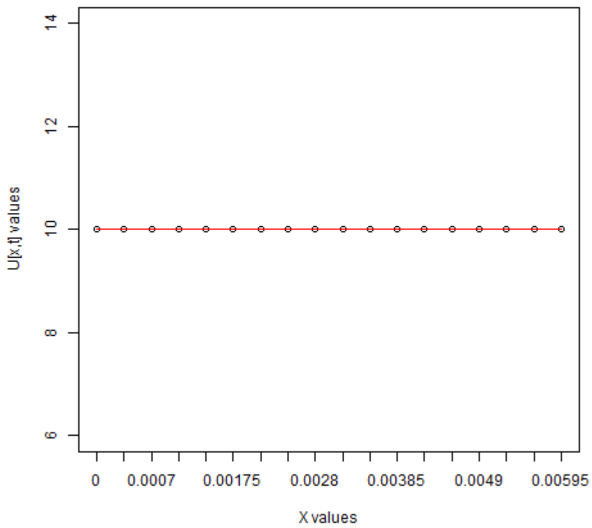
L = 0.006 #длина отрезка

$$U_{0,j} = -10$$

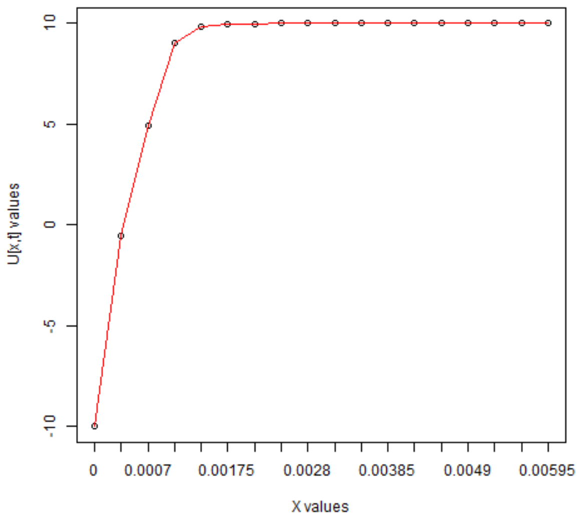
$$U_{N,j} = 10$$

Анализ результатов

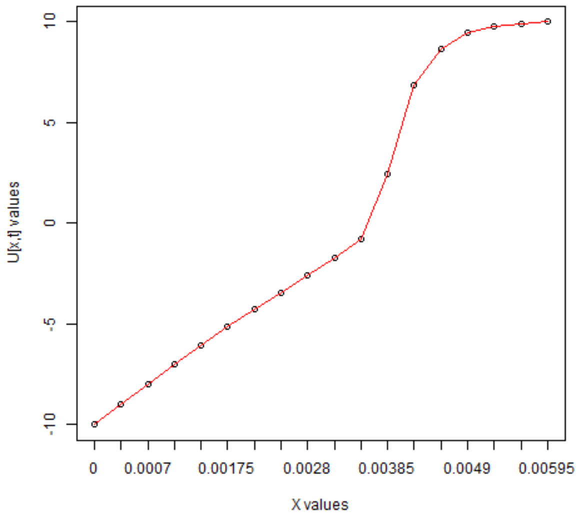
При t_j= 1 :



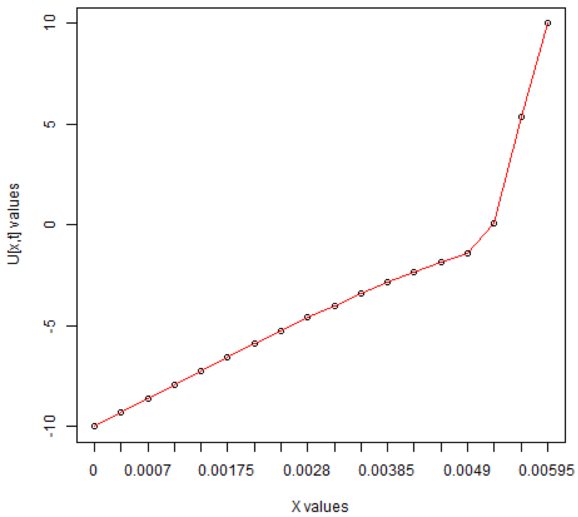
При t_j = 2 :



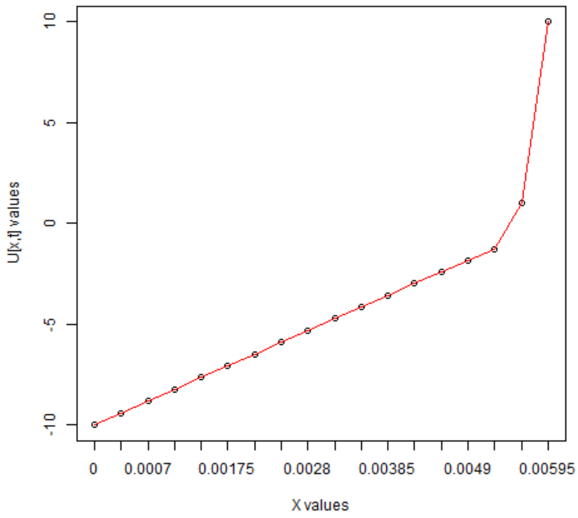
При t_j=25 :



При t_j= 50 :

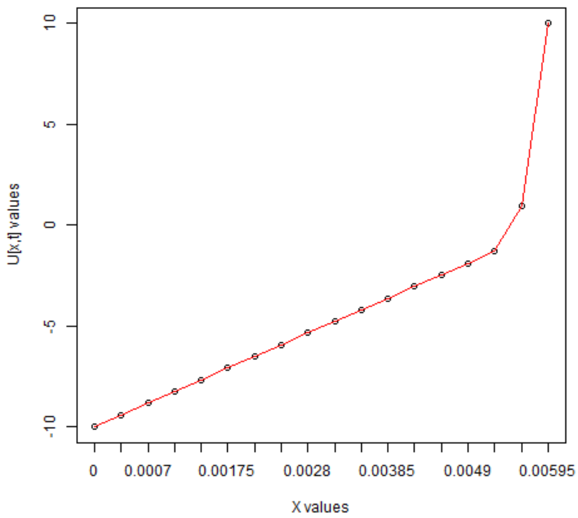


При t_j= 75:



U[x,t] values
10
5
0
-5
-10
0
0.0007
0.00175
0.0028
0.00385
0.0049
0.00595
X values

При t_j= 100:



U[x,t] values
10
5
0
-5
-10
0
0.0007
0.00175
0.0028
0.00385
0.0049
0.00595
X values

Код программы

```
rm(list = ls())

#k1=k2=1
k1=k2=0

k1_U = 2.3
#k2_U = k1_U
k2_U = 0.58
density_1  = 918.7
density_2 = 999.7
#mean_density= mean(c(density1,density2))


c1_U = 2000 * density_1
c2_U = 4195 * density_2
mean_K = mean(c(k1_U,k2_U))
mean_C = mean(c(c1_U,c2_U))
#c2_U = c1_U
a=(mean_K/mean_C)^(1/2)
h = 0.00035 #шаг по x
tau = h^2/a^2

L = 0.006 #длина
N = round(L/h) #Число шагов
n = 150 #Число шагов по времени
tj = numeric(n) #tau j
nj = numeric(n) #ветор моментов времени

kU = function(U_i)
{
  if( U_i < 0)
    return(k1_U)
  else return(k2_U)
}

CU = function(U_i)
{
  if( U_i < 0)
    return(c1_U)
  else return(c2_U)
}


A = function(t)
{
  #0
  #1/(1+t)
  -10
}
B = function(t)
{
  #0
  #1/(1+t)
  10
}

x = numeric(N+1)
fij = numeric(N+1) #сила

Ai = numeric(N) #коэффициенты
Bi = numeric(N)
Ci = numeric(N)
```

```
Fi = numeric(N-1)


alpha = numeric(N)
beta = numeric(N)

for(j in 1:n)
{
  tj[j]=(j-1)*tau
  nj[j]=j-1
}

Ux0=function(x)
{
  #A(tj[1]) + (B(tj[1])-A(tj[1]))*(x/L)^2;
  #1
  10
}

for(i in 1:N+1)
{
  x[i]= (i-1)*h
  fij[i]=0
}

U = matrix(data=NA,nrow=n,ncol=N+1)
colnames(U)=x
row.names(U)= c(0:(n-1))


for(i in 0:N+1) #Считаем 0-й слой
{
  U[1,i] = Ux0(x[i])
}

CoeffF = function(j) #Поправка на коэфф
{

  alpha[1] = k1
  beta[1] = A(tj[j])
  Ai[1] = - kU(U[j,1]) / h^2
  Bi[1] = - kU(U[j,1]) / h^2
  Ci[1] = CU(U[j,1]) / tau + 2 * kU(U[j,1]) / h^2

  for(i in 2:(N-1)) # Считаем очередные Ai, Bi , Ci
  {
    Ai[i] = - kU(U[j,i-1]) / h^2
    Bi[i] = - kU(U[j,i+1]) / h^2
    Ci[i] = CU(U[j,i]) / tau + (kU(U[j,i+1]) + kU(U[j,i-1]))  / h^2
  }

  # for(i in 1:(N-1)) # Считаем очередные Ai, Bi , Ci
  # {
  #   Ai[i] = - kU(U[j,i]) / h^2
  #   Bi[i] = - kU(U[j,i]) / h^2
  #   Ci[i] = CU(U[j,i]) / tau + 2 * kU(U[j,i]) / h^2
  # }

  for(i in 1:(N-1)) #Считаем Fi
    Fi[i] = CU(U[j,i]) * U[j-1,i+1] / tau

  for(i in 2:N) #Считаем альфа и бета коэффициенты
  {
```

```
    alpha[i]=-Bi[i-1]/(Ci[i-1]+Ai[i-1]*alpha[i-1])
    beta[i]= (Fi[i-1]-Ai[i-1]*beta[i-1])/(Ci[i-1]+Ai[i-1]*alpha[i-1])
  }

  U[j,N+1]=(B(tj[j])+k2*beta[N])/(1-k2*alpha[N])
  for(i in N:1) #Считаем Uj
    U[j,i]= alpha[i] * U[j,i+1] + beta[i]

  return(U[j,])
}

IterF = function(j) #Считаем остальные слои
{

  alpha[1] = k1
  beta[1] = A(tj[j])

  for(i in 1:(N-1)) # Считаем очередные Ai, Bi , Ci
  {
    Ai[i] = - kU(U[j-1,i]) / h^2
    Bi[i] = - kU(U[j-1,i]) / h^2
    Ci[i] = CU(U[j-1,i]) / tau + 2 * kU(U[j-1,i]) / h^2
  }

  for(i in 1:(N-1)) #Считаем Fi
    Fi[i] = CU(U[j-1,i])*U[j-1,i+1]/tau


  for(i in 2:N) #Считаем альфа и бета коэффициенты
  {
    alpha[i]=-Bi[i-1]/(Ci[i-1]+Ai[i-1]*alpha[i-1])
    beta[i]= (Fi[i-1]-Ai[i-1]*beta[i-1])/(Ci[i-1]+Ai[i-1]*alpha[i-1])
  }

  U[j,N+1]=(B(tj[j])+k2*beta[N])/(1-k2*alpha[N])

  for(i in N:1) #Считаем Uj
    U[j,i]= alpha[i] * U[j,i+1] + beta[i]
  #U[j,1]=k1*U[j,2]-h*A(tj[j])

  return(U[j,])
}

for(j in 2:n)
{
  U[j,] = IterF(j)
  U[j,] = CoeffF(j)
}

options(scipen = 999) # Disable exponential notation (e.g. 1.81e+09)
#print("Численное решение")
#print(U)
#print("Точное решение")
#print(Uacc)

library(animation)
oopt = ani.options(interval = 0.3)

ani.record(reset = TRUE)
for(j in 1:n)
{
  plot(U[j,], xaxt="n", xlab = 'X values', ylab = 'U[x,t] values')
  lines(U[j,],col="red")
```

```
  #lines(Uacc[j,],col="blue")
  axis(1, at = c(1:(N+1)), labels = x)
  ani.pause()
  ani.record()

}


# while(TRUE)
# {
# ani.replay()
# ani.options(oopt, loop = 100)
# }
```

Список использованной литературы

1) Тихонов А.Н., Самарский А.А. - Уравнения математической физики , Москва, 1977
2) https://ru.wikipedia.org/wiki/Задача_Стефана