О лабораторной работе №5 по курсу "Вычислительные алгоритмы" 2021-2022 уч. года

12 мая 2022 г.

Содержание

[Благодарности 1](#_Toc19227)

[1 Условие задачи 1](#_Toc19228)

[1.1 Математическая модель 2](#_Toc19229)

[1.2 Физическое содержание 2](#_Toc19230)

[1.3 Искомая величина 2](#_Toc19231)

[2 Подробное решение 3](#_Toc19232)

[2.1 Разбиение отрезка и сеточная функция 3](#_Toc19233)

[2.2 Переход к почти СЛАУ 3](#_Toc19234)

[2.3 Учет краевых условий 4](#_Toc19235)

[2.4 Линеаризация 5](#_Toc19236)

[2.5 Метод прогонки 6](#_Toc19237)

[2.6 Основной итерационный процесс 6](#_Toc19238)

[3 Краткий алгоритм 6](#_Toc19239)

[4 Результаты работы алгоритма 7](#_Toc19240)

[4.1 Тест 1 7](#_Toc19241)

[4.2 Тест 2 7](#_Toc19242)

[4.3 Тест 3 8](#_Toc19243)

[4.4 Тест 4 8](#_Toc19244)

[5 Источники 8](#_Toc19245)

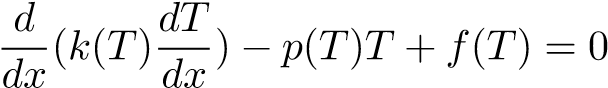
# Условие задачи

Условие задачи приводится по оргинальному файлу с заданием от преподавателя по курсу — Градова В. М.

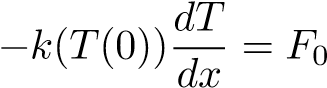
Единицы измерения по условию считаются согласованными и здесь не приводятся.

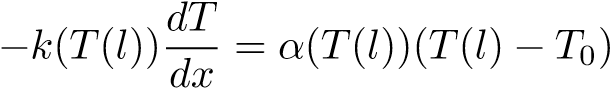
## Математическая модель

Задано уравнение

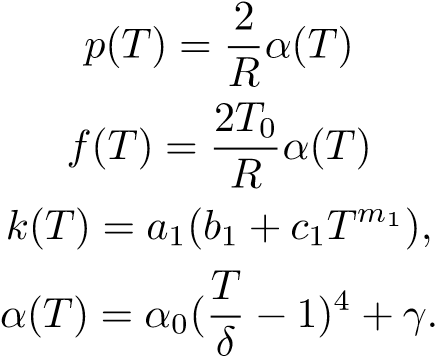
 (1)

с краевыми условиями

; (2)

*.* (3)

Заданные функции:

(4)

(5)

(6)

(7) Значения параметров:

*a*1 = 0*.*0134*, b*1 = 1*,*

*c*1 = 4*.*35 · 10−4*, m*1 = 1*,*

*α*0 = 1*.*94 · 10−2∗*, δ* = 1*.*5 · 103*,*

*γ* = 0*.*2 · 10−2*, l* = 10*,*

*T*0 = 300*,*

*R* = 0*.*5*,*

*F*0 = 50*.*

\* — в оригинальной задаче это значение было с противоположным знаком, однако требуемые свойства графика не выполнялись.

## Физическое содержание

Сформулированная математическая модель описывает температурное поле *T*(*x*) вдоль цилиндрического стержня радиуса *R* и длиной *l*, причем *R >> l* и температуру можно принять постоянной по радиусу цилиндра. Ось *x* направлена вдоль оси цилиндра и начало координат совпадает с левым торцом стержня. Слева при *x* = 0 цилиндр нагружается тепловым потоком *F*0. Стержень обдувается воздухом, температура которого равна *T*0. В результате происходит съем тепла с цилиндрической поверхности и поверхности правого торца при *x* = *l*. Функции *k*(*T*), *α*(*T*) являются, соответственно, коэффициентами теплопроводности материала стержня и теплоотдачи при обдуве. Эти коэффициенты привязаны к температуре, т.е. *k*(*T*) зависит от *T*.

## Искомая величина

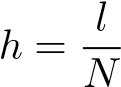
Необходимо найти приближенно зависимость *T*(*x*), а также проверить корректность определения искомой зависимости в следующих случаях:

1. *F*0 = 50 (нагрев);
2. *α*0 = 3 · *α*0*st* (*α*0*st* — значение *α*0 из условия, в этом случае теплоотдача выше, чем в предыдущем);
3. *F*0 = −10 (съем тепла);
4. *F*0 = 0 (отсутствие нагрева/съема тепла).

# Подробное решение

## Разбиение отрезка и сеточная функция

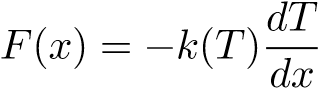
Для поиска зависимости *T*(*x*) интересующий нас отрезок стержня [0*,l*] необходимо разбить на участки длиной *h* каждый, причем

*,* (8)

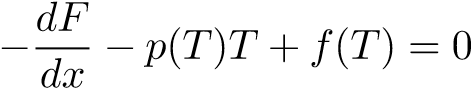
Граничные точки участков пронумеруем от 0 до *N* так, что 0 = *x*0*,h* = *x*1*,...l* = *xN*. Зависимость *T*(*x*) ищется как сеточная функция *y*(*x*), при этом под *сеточной функцией* понимаются значения *T*(*x*) в узлах введенной только что сетки. Далее под *yi* будем понимать *y*(*xi*), а под *ki*, *pi*, *fi* — соответственно *k*(*yi*), *p*(*yi*), *f*(*yi*).

## Переход к почти СЛАУ

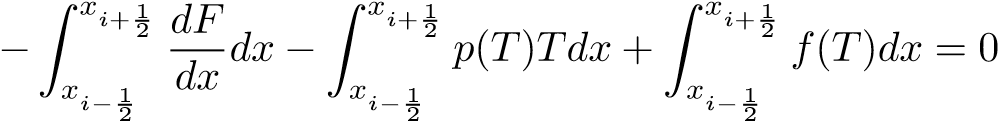
Введем в рассмотрение функцию

*.* (9)

Тогда уравнение (1) запишется в виде:

 (10)

Пусть *xi* — некоторая граничная точка участка, . Проинтегрируем уравнение (10) от до:

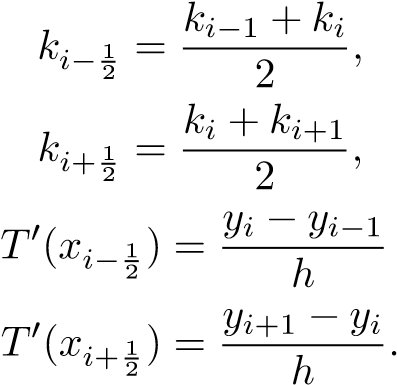
 (11)

При довольно малых *h* можно считать (11) равносильным следующему:

 (12) Снова пользуясь малостью *h*, отметим следующее:

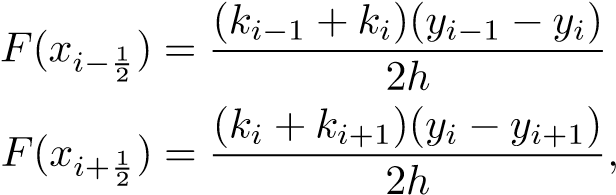
(13)

(14)

 *,* (15)

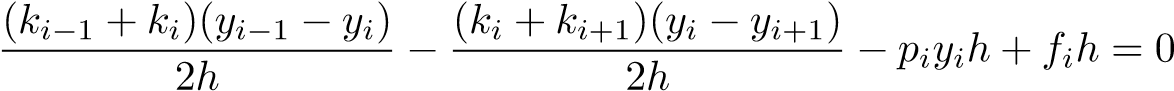
(16)

Тогда

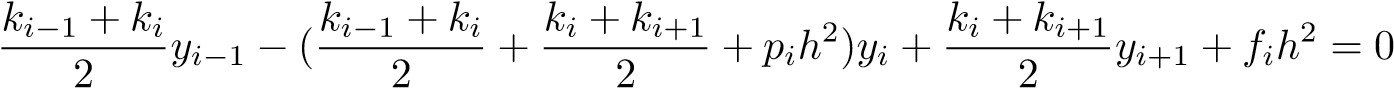
 *,* (17)

(18)

а уравнение (12) запишется как

*.* (19)

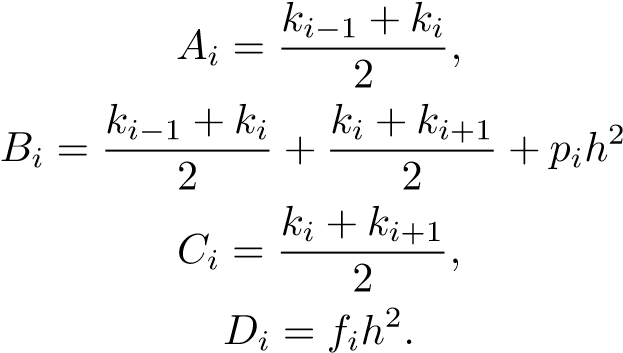
Умножим (19) на *h* и выделим коэффициенты при *yi*−1, *yi*, *yi*+1:

*.* (20)

Если сравнить написанное выше с уравнением СЛАУ с трехдиагональной матрицей

*Aiyi*−1 − *Biyi* + *Ciyi*+1 + *Di* = 0*,*

нетрудно выяснить, что в случае уравнения (20):

(21)

*,* (22)

(23)

(24)

## Учет краевых условий

На данный момент не решены две проблемы: не используются никак заданные краевые условия и для *N* + 1 переменной есть всего *N* − 1 уравнение. Ключом к решению обеих проблем является следующее. Давайте присмотримся к краевым условиям (2) и (3) и сопоставим их с выражением (9). Можно заметить, что краевые условия — это, по сути, значения *F*(*x*0) и *F*(*xN*)!

То есть:

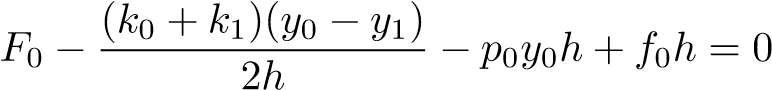
*F*(*x*0) = *F*0; (25)

*F*(*xN*) = *α*(*yN*)(*yN* − *T*0)*.* (26)

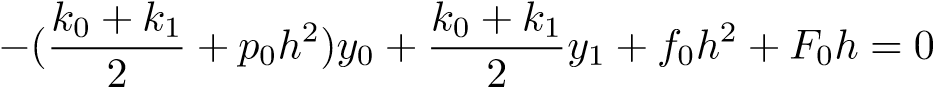
Если для отрезка  написать выражение, подобное (11), и поступить с этим выражением так же, как и с (11), получим аналог выражения (12):

*.* (27)

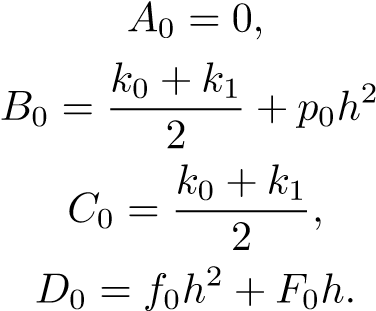
В выражении выше было допущено, опять же в силу малости *h*, что. Пользуясь достигнутыми ранее в этом разделе результатами, получим:

*.* (28)

Умножим на *h* и выделим коэффициенты при *y*0 и *y*1:

*.* (29)

Получили еще одно уравнение для нашей системы, а коэффициенты в нем будут такие:

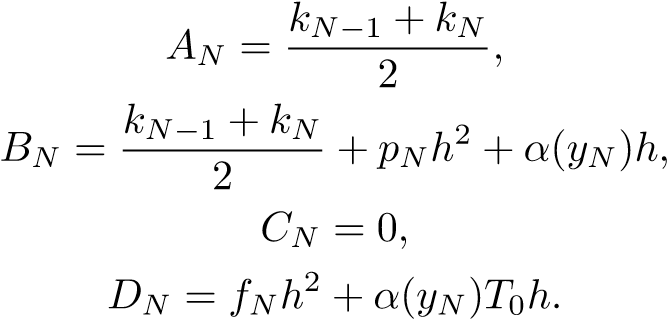
(30)

*,* (31)

(32)

(33)

(*N* + 1)—е уравнение для системы получается аналогичным образом, приведем лишь коэффициенты в нем:

(34)

(35)

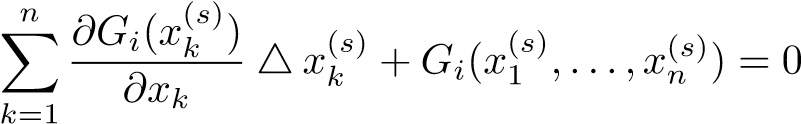
(36)

(37)

## Линеаризация

Раздел "Переход к *почти* СЛАУ" называется так неслучайно. Полученное уравнение (20) могло бы быть уравнением СЛАУ, если бы не нелинейность входящих в него выражений. Для того, чтобы получить действительную СЛАУ и решать её известными из предыдущих работ курса методами, проведем *линеаризацию* уравнения (20) со вспомогательными обозначениями (21) — (24) по методу Ньютона.

Метод Ньютона является итерационным, согласно ему, линеаризация производится так. Если задана система некоторых дифференцируемых функций *Gi*(*x*1*,x*2*,...,xn*)*,i* = 1*,*2*,...,n*, то на очередной итерации с номером *s*:

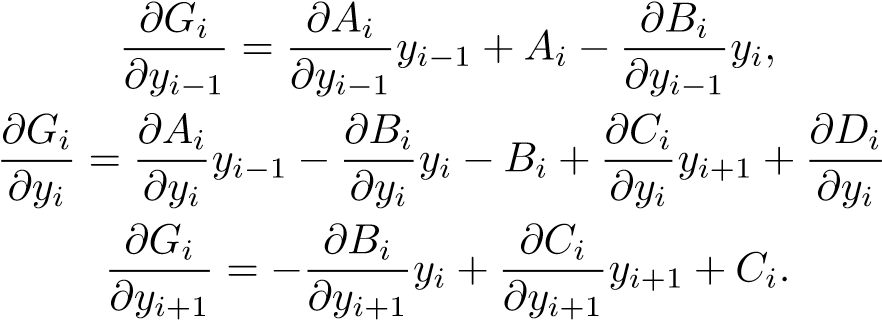
*,* (38)

а на следующей, (*s* + 1)—й итерации:

 (39)

Если принять на очередном шаге процесса поиска решения нашей задачи (с поправкой на равенство нулю *A*0 и *CN*)

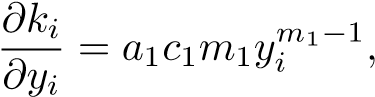
|  |  |
| --- | --- |
| *Gi*(*yi*−1*,yi,yi*+1) = *Aiyi*−1 − *Biyi* + *Ciyi*+1 + *Di,* тогда в силу равенств (21) — (24): | (40) |

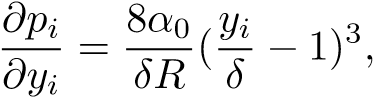
(41)

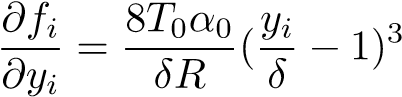
*,* (42)

(43)

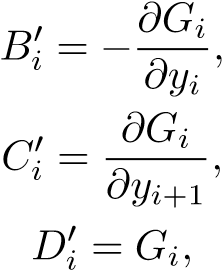
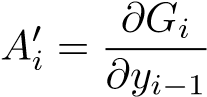
Предполагается, что читатель уже знаком с частными производными, и вычислить необходимые для выражений выше сможет самостоятельно. Приведем лишь некоторые полезные равенства:

 (44)

 (45)

*.* (46)

Вводя новые обозначения:

 *,* (47)

(48)

(49)

(50)

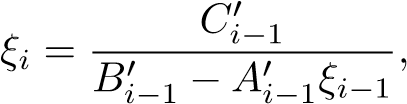
получим все уравнения для СЛАУ с трехдиагональной матрицей относительно переменных 4*y*0*,*4*y*1*,...,*4*yN*:

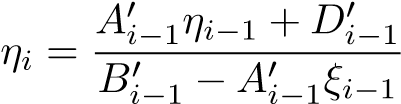
|  |  |
| --- | --- |
| *,* | (51) |
| *A*0*i* 4 *yi*−1 − *Bi*0 4 *yi* + *Ci*0 4 *yi*+1 + *Di*0 = 0*,i* = 1*,*2*,...,N* − 1*,* | (52) |
| *A*0*N* 4 *yN*−1 − *BN*0 4 *yN* + *DN*0 = 0*.* | (53) |

Заметим, что слагаемые, содержащие в качестве множителя, пропущены в силу равенства последних нулю, в чем нетрудно убедиться, вычислив соответствующие частные производные.

## Метод прогонки

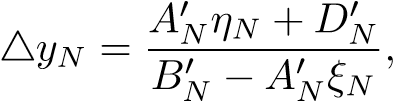
Согласно методу прогонки, необходимо найти прогоночные коэффициенты *ξi* и *ηi*, после чего станет возможным вычисление 4*yi*. Стандартные выражения для *ξi* и *ηi* (*i* = 1*,*2*,...,N*):

 (54)

*.* (55)

Здесь нужно учесть, что при *i* = 1, поэтому нет необходимости каким-то специальным образом задавать *η*0 и *ξ*0. Можно принять их равными нулю.

После вычисления прогоночных коэффициентов (прямой ход) выполяется уже вычисление 4*yi* (обратный ход) по формулам:

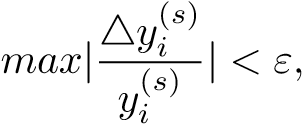
 (56)

4*yi* = *ξi*+1 4 *yi*+1 + *ηi*+1*,i* = 0*,*1*,...,N* − 1*.* (57)

Таким образом, в результате метода прогонки будут получены значения 4*y*0*,*4*y*1*,...,*4*yN* для очередной итерации, которые можно использовать для перехода к следующей итерации согласно (39).

## Основной итерационный процесс

Итерационный процесс состоит из вычисления 4*y*0*,*4*y*1*,...,*4*yN* согласно методу прогонки и корректировки значений сеточной функции в соответствии с (39). Условие прекращения процесса:

 (58)

|  |  |
| --- | --- |
| *i* = 0*,*1*,*2*,...,N,* | (59) |
| *ε* = 10−6 *...*10−4 | (60) |

Здесь *i* — номер граничной точки, *s* — номер итерации.

Стоит отметить, что в силу сложности краевых условий (нельзя явно найти значения *T*(0) и *T*(*l*)) в качестве начального приближения зависимости *T*(*x*) стоит брать *T*(*x*) = *T*0.

# Краткий алгоритм

1. Разбить отрезок на участки;
2. Установить начальное приближение *T*(*x*) = *yi* = *T*0*,i* = 0*,...,N*;
3. Для всех *i* = 0*,...,N* вычислить d*yi*:
   1. Вычислить *ki*, *pi*, *fi* (см. 1.1,2.1);
   2. Вычислить *Ai*, *Bi*, *Ci*, *Di* (см. (21) — (24), (30) — (33),(34) — (37));
   3. Вычислить *A*0*i*, *Bi*0, *Ci*0, *Di*0 (см. (47) — (50));
   4. Вычислить *ξi*, *ηi* (см. (54) — (55));
   5. Вычислить собственно 4*yi* (см. (56) — (57));
4. Если условие (58) выполняется, перейти к шагу 5, иначе — скорректировать *yi* на 4*yi* и перейти к шагу

3;

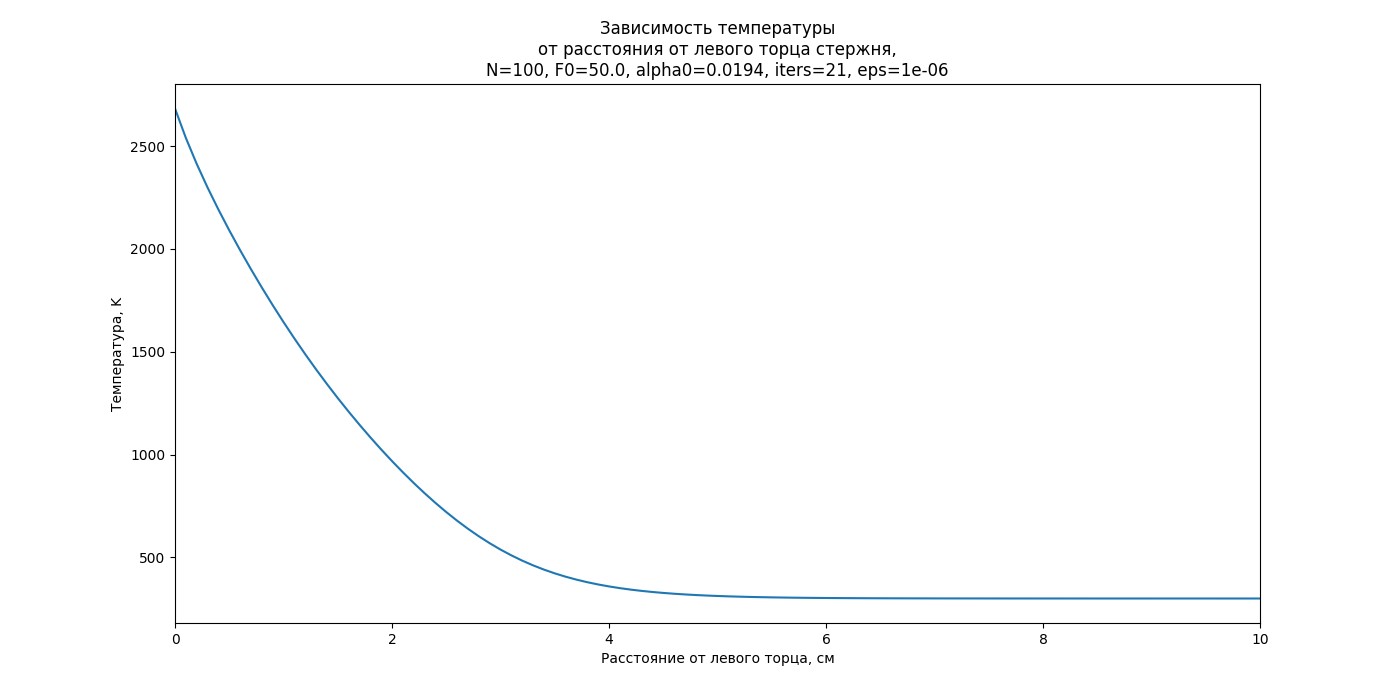
1. Представить полученный результат в виде графика.

# Результаты работы алгоритма

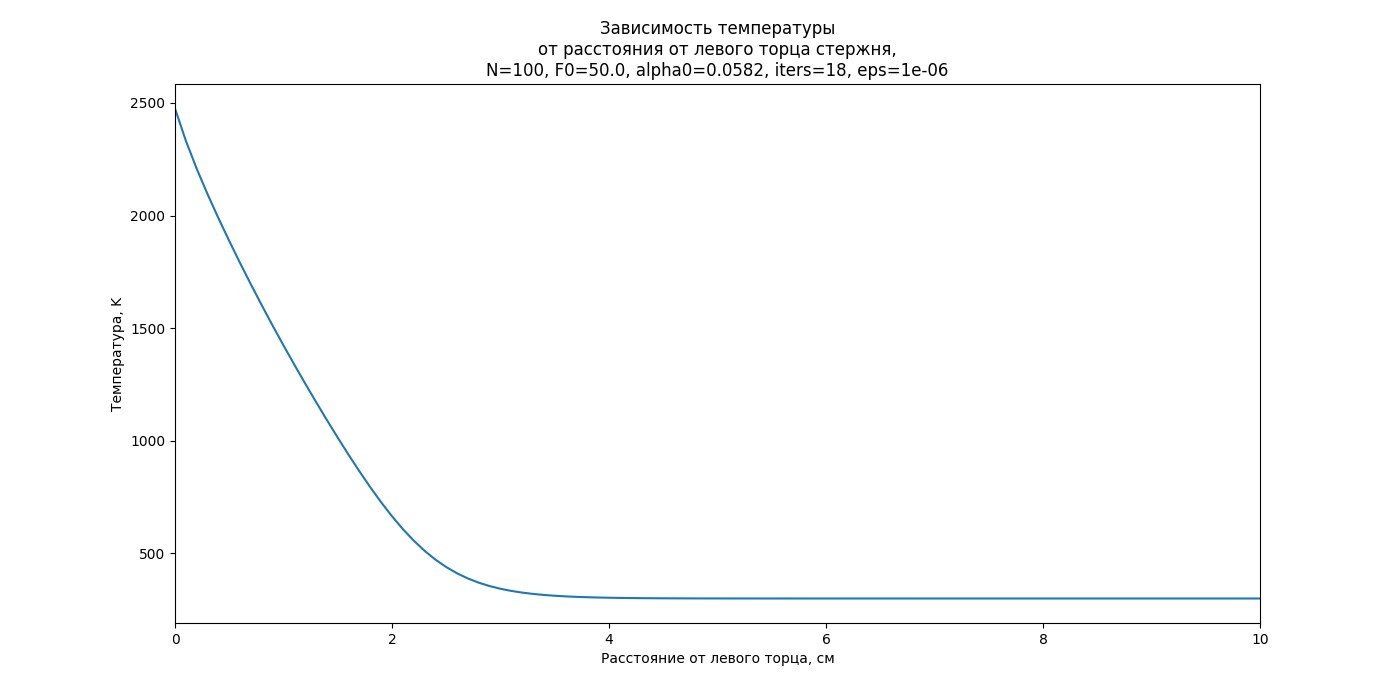
Приводятся графики, построенные с помощью matplotlib по результатам работы алгоритма. Обозначения на изображениях: *N* — число отрезков, *F*0 — значение потока у левого торца, *α*0 — значение соответствующего коэффициента в данном тесте, *iters* — число итераций, за которое был получен график, *ε* — значение точности.

Тесты приводятся в порядке, в котором они изложены в п. 1.3.

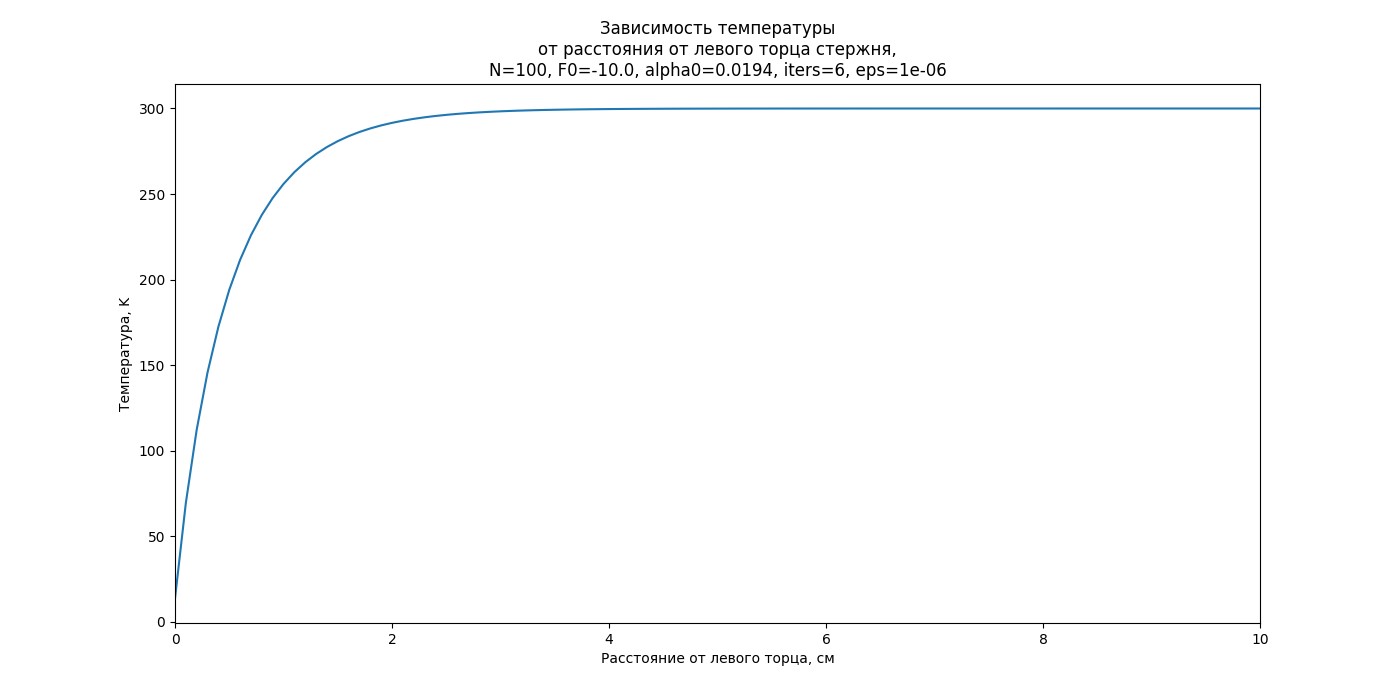
## Тест 1



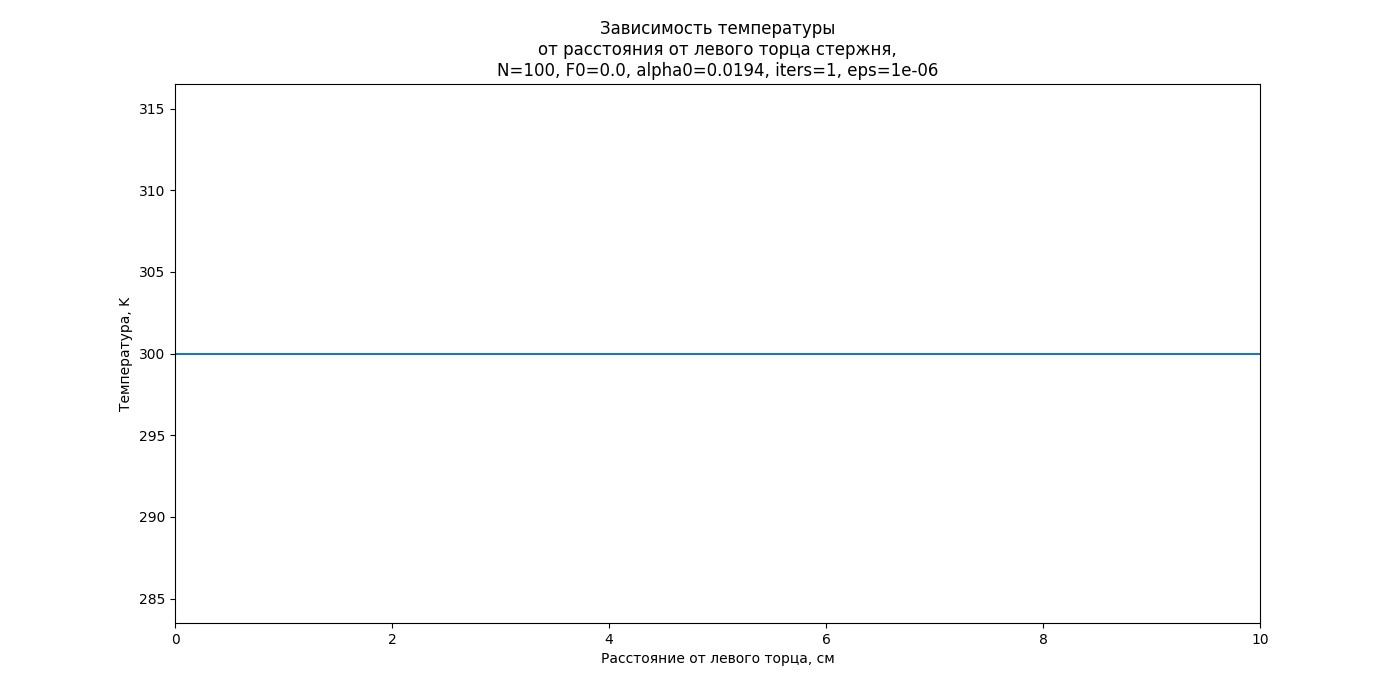
## Тест 2



## Тест 3



## Тест 4



# Источники

1. Курс лекций Градова В. М. по курсу "Вычислительные алгоритмы".
2. https://algowiki-project.org/ru/Метод\_Ньютона\_для\_систем\_нелинейных\_уравнений.
3. https://www.wikiwand.com/ru/Метод\_прогонки