Московский авиационный институт

(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Курсовая работа по курсу «Численные методы»

(6-й семестр)

Решение нелинейных уравнений методом продолжения по параметру

Студент: Сизонов А. А.

Преподаватель: Сластушенский Ю. В.

Группа: М8О-408Б

Дата:

Оценка:

Подпись:

**Москва, 2019**

1. Постановка задачи (математическое описание метода)

Введем в рассмотрение некоторую систему нелинейных уравнений:

H (x, t) =0

Такую, что:

1. При t=0 система H (x, 0) =0 имеет решение x0

2. При t=1 система H (x, 1) =0 имеет решение x\*

3. Вектор-функция H (x, t) непрерывна по t. Тогда меняя t от 0 до 1 и решая для каждого ti систему H (x, ti)=0, например, методом Ньютона, можно найти последовательно x0 , x1 , x2 , …, x\* .

Вектор-функция H(x, t) может быть выбрана различными способами. Рассмотрим три распространенных варианта:

1. H (x, t) =F (x) + (t-1) \*F (x0 ) =0

При t=0:

F (x0 ) - F (x0 ) =0**,**т.е. условие 1) выполнено.

При t=1:

F (x\*) - (1-1) \* F (x0 ) =F (x\*) =0.

Вектор-функция H (x, t) непрерывна по t.

2. H (x, t) =t\*F (x)

Условия 1) - 3) соблюдаются и для этой вектор-функции.

Идея метода:

Полагаем t1= ∆t и решаем систему H (x, t1) =0 при выбранном x0 . Получаем xt 1 .

Далее, берем его в качестве начального приближения и решаем при новом t2 =t1 +∆t систему H (x, t2 ) =0, получаем xt 2 и так далее до тех пор, пока не будет достигнута заданная точность. Нелинейные системы H (x, ti) =0 на каждом шаге по t решаются, например, методом Ньютона, который обычно сходится, так как xt i-1 и xt i лежат близко друг к другу. Если несмотря на это решение xt i не получается за 6-7 итераций, ∆t уменьшается и система H (x, ti ) =0 решается снова.

2. Последовательность шагов реализации алгоритма

**Шаг 1.**Формирование системы H(x, t) =0.

**Шаг 2.**Выбор начального приближения x0 , (например, x0 =0) и точности решения εgon .

**Шаг 3.**Полагаем i=1.

**Шаг 4.**Вычисляем ti =ti-1 +∆t (обычно вначале берут ∆t=0,1)

**Шаг 5.**Решаем систему H (x, ti ) =0. Получаем вектор xt i . При этом считаем число итераций m. Если m>10, значит метод Ньютона уже не сойдется, так как xt i-1 и xt i слишком далеки друг от друга. Тогда надо уменьшить ∆t в два раза и вернуться к шагу 4. Будем считать, что xt i найдено.

**Шаг 6.**Проверяем, достигли ли мы заданной точности. Например, используя первый способ,

|| xt i -xt i-1|| ≤ εgon .

Если последнее условие не соблюдается, то переходим к шагу 4. Иначе считаем, что x\*= xt i и расчеты закончены.

3. Программный код:

from numpy import linalg

import matplotlib.pyplot as plt

import numpy

from sympy import diff

import sys

import logging

def feval(expr, x):

return eval(expr)

def PC\_method(F, DF, x0, dt, eps):

t = dt

x = x0

tout = [t]

xout = [x0]

n = 1

dxout = [0 for i in range(0, n)]

m = 0

it = 0

while t <= 1:

ndx = 1

nh = 1

dx = 1

while max(ndx, nh) > eps:

h = -feval(F, x0) \* t

dx = h / feval(DF, x0)

x += dx

m += 1

ndx = linalg.norm(dx)

nh = linalg.norm(h)

if m > 10:

t = t - dt

dt = dt / 2

t = t + dt

x = x0

m = 0

x0 = x

tout.append(t)

xout.append(x)

dxout.append(dx)

if m < 4:

dt = dt \* 2

t = dt + t

it = it + 1

print("it:", it)

print("t:", t)

return xout, dxout, m, x, it, t

if \_\_name\_\_ == '\_\_main\_\_':

logging.basicConfig(filename="log.txt", level=logging.INFO)

full\_expr = input("Enter equation (F(x) = 0): ").split("=")

expr = full\_expr[0]

if len(full\_expr) > 1:

expr += "-" + full\_expr[1]

print("Transformed equation: " + expr + " = 0")

d\_expr = None

try:

d\_expr = diff(expr)

except:

print("Error expression")

sys.exit(1)

dt = 0.1

x0 = float(input("Enter the initial approximation (x0) (for example 0.5): "))

eps = float(input("Enter a valid error (epsilon) (for example 0.01): "))

xout, dxout, m, x, it, t = PC\_method(expr, str(d\_expr), x0, dt, eps)

logging.info(xout)

logging.info(dxout)

logging.info("lens: {0} {1}".format(len(xout), len(dxout)))

segment = numpy.linspace(0, 1, num=len(xout), endpoint=True, retstep=False, dtype=None)

#logging.info(segment)

logging.info(m)

plt.subplot(2, 1, 1)

plt.title("equation: "+ expr + " = 0\n"

+ "x = {0}\n".format(str(x))

+ "number of iterations: {0}".format(str(it)))

plt.text(0.5, 0.5, 'matplotlib', horizontalalignment='center', verticalalignment='center')

plt.plot(segment, xout, color='b', label='Solution')

plt.grid(True)

plt.legend()

plt.subplot(2, 1, 2)

plt.plot(segment, dxout, color='r', label='Errors')

plt.grid(True)

plt.legend()

plt.show()

print(str(d\_expr))