Министерство образования РФ

Новосибирский Государственный Технический Университет

Кафедра экономической информатики

**Курсовая работа**

по дисциплине "Численные методы"

Тема: **Исследование метода продолжения решения по параметру для нелинейных САУ**

Группа:

Выполнил:

Проверила: Сарычева О.М.

Новосибирск 2011 г.

Содержание

Введение

[1. Постановка задачи (математическое описание метода)](#_Toc291658249)

2. Описание программного обеспечения

[2.1 Общие сведения и требования к ПО и описание логической структуры](#_Toc291658251)

3. Описание тестовых задач

[4. Анализ результатов](#_Toc291658253)

Заключение

[Используемая литература](#_Toc291658255)

# Введение

В данной курсовой работе будет рассмотрен метод продолжения решения по параметру, с помощью которого можно эффективно находить корни нелинейных САУ. В работе исследуется влияние вектора начальных приближений **x**0 и заданной точности решения εgon на число итераций, время счета и сходимость метода. Так же дается описание программного обеспечения и тексты программ, использованные в данной работе для построения графиков сходимости метода для различных начальных значений вектора **x0**, графики ошибки.

# 1. Постановка задачи (математическое описание метода)

**Метод продолжения решения по параметру** является наиболее универсальным при решении нелинейных САУ. Пусть t - параметр, меняющийся от 0 до1. Введем в рассмотрение некоторую САУ

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\*.

Так как x0 при t=0 известно, то всегда можно найти t1, достаточно близкое к t0, при котором будут выполняться условия сходимости, например, метода Ньютона. Аналогично можно обеспечить условия сходимости метода Ньютона и для t2, t3,…, t=1.

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

При t=1 F (x\*) - (1-1) \* F (x0) =F (x\*) =0. И, наконец, вектор-функция H (x, t) непрерывна по t.

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

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

Идея метода состоит в следующем. Полагаем t1=∆t и решаем систему H (x, t1) =0 при выбранном x0. Получаем xt1. Далее, берем его в качестве начального приближения и решаем при новом t2=t1+∆t систему H (x, t2) =0, получаем xt2 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 xti-1 и xti лежат близко друг к другу. Если несмотря на это решение xti не получается за 6-7 итераций, ∆t уменьшается и система H (x, ti) =0 решается снова.

##### **Последовательность шагов реализации алгоритма состоит в следующем:**

**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 xti. При этом считаем число итераций m. Если m>10, значит метод Ньютона уже не сойдется, так как xti-1 и xti слишком далеки друг от друга. Тогда надо уменьшить ∆t в два раза и вернуться к шагу 4. Будем считать, что xti найдено.

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

|| xti-xti-1 || ≤ εgon.

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

# 2. Описание программного обеспечения

# 2.1 Общие сведения и требования к ПО и описание логической структуры

ПО состоит из следующих файлов: mpr. m, prog. m, funf. m, funj. m. Программы, реализующие метод, разработаны в среде МаtLab, предназначенной для выполнения математических операций. Программа состоит из программы-функции mpr. m, которая описывает метод, программы с данными - основная программа prog. m и двух подпрограмм-функций funf. m - для нахождения корней системы уравнений; funj. m - для нахождения матрицы Якоби. Рассмотрим их подробнее.

**Функциональное назначение**

Программа предназначена для решения систем нелинейных алгебраических уравнений в среде МаtLab методом продолжения решения по параметру.

**Используемые переменные:**

**t** - время выполнения итерационного процесса;

**x** - вектор начального приближения к решению;

**n** - размерность вектора;

**m** - номер итерационного процесса;

**it** - счетчик итераций.

**Входные параметры:**

**funf** - формальное имя программы, которое дает возможность вычислить корни нелинейных САУ.

#### **funj** - формальное имя программы, которое дает возможность вычислить матрицу Якоби.

**x0** - начальное приближение собственного вектора;

**dt -** приращение времени;

**edop** - заданная допустимая ошибка;

**trace** - установка режима вывода на экран;

**Выходные параметры:**

**tout** - выходное значение времени;

**xout** - конечное значение x;

**dxout** - конечное значение вектора ошибки.

**Тексты программ:**

Mpr. m

function [xout,dxout,tout] =mpr (funf,funj,x0,dt,edop,trace)

t=dt; x=x0; tout=t; xout=x0'; n=size (x0);

dxout=zeros (1,n); m=0; it=0;

f0=feval (funf,x0);

while (t<=1)

ndx=1;

nh=1;

nv= [ndx; nh];

while (max (nv) >edop)

J=feval (funj,x0);

F=feval (funf,x0);

h= (-F) \*t;

dx=J\h;

x=x+dx;

m=m+1;

ndx=norm (dx);

nh=norm (h);

nv= [ndx; nh];

if (m > 10)

t=t-dt;

dt=dt/2;

t=t+dt;

x=x0;

m=0;

end;

end;

x0=x;

tout= [tout; t];

xout= [xout; x'];

dxout= [dxout; dx'];

if (m < 4)

dt=dt\*2;

end;

t=dt+t;

it=it+1;

end;

disp ('it ='); %количество итераций

disp (it);

disp ('t ='); %время выполнения итерационного процесса

disp (t);

pause;

**xout** - конечное значение x;

dxout **- конечное значение вектора ошибки.**

m **- номер итерации.**

Prog. m

trace=1;

dt=0.1;

x0=0;

edop=0.1;

[xout,dxout,m] = mpr ('funf','funj',x0,dt,edop,trace);

plot (m,xout); %График значений x

pause;

plot (m,dxout); %График ошибки

pause;

Funf. m

function [f] =funf (x)

f= [0.0001\*exp (30\*x) +x-6];

end

Funj. m

function [j] =funj (x)

j= [30\*0.0001\*exp (30\*x) +1];

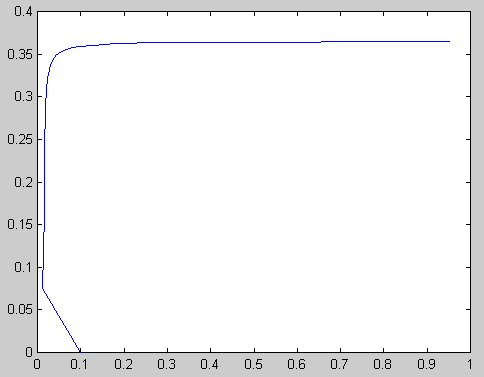
end

# 3. Описание тестовых задач

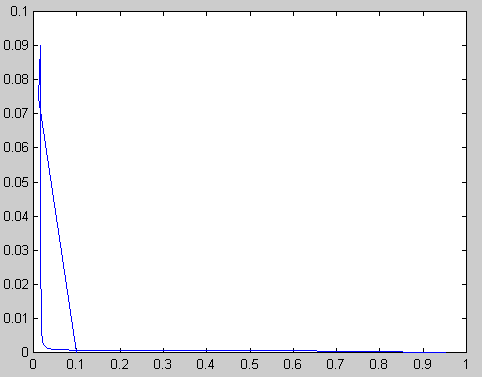
**Исследуем влияние вектора начальных приближений на время счета, число итераций и сходимость метода.**

Начальное приближение **x0=0**, заданная точность edop=0.1, dt=0.1.

**График значений x на каждом шаге итерации**



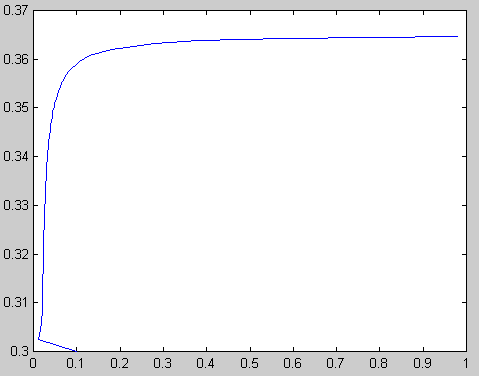
**График ошибки**



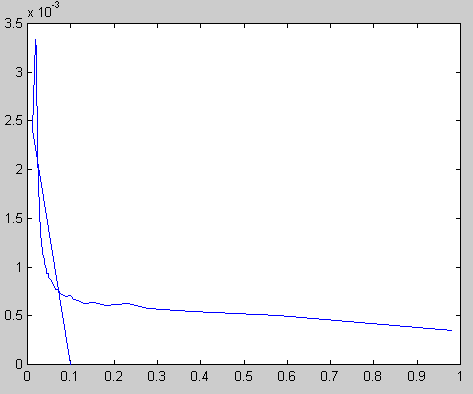
Результат выполнения программы: количество итераций=60, время счета=4с.

Начальное приближение **x0=0.3**, заданная точность edop=0.1, dt=0.1.

**График значений x на каждом шаге итерации**



**График ошибки**

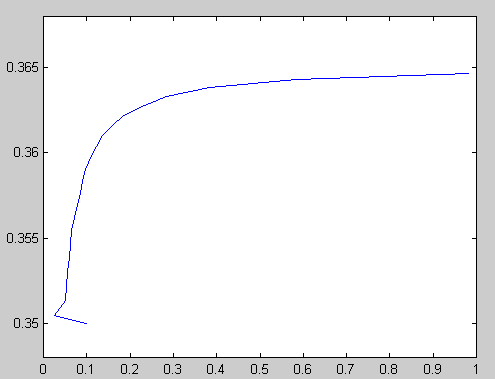


Результат выполнения программы: количество итераций=50, время счета=3,5с.

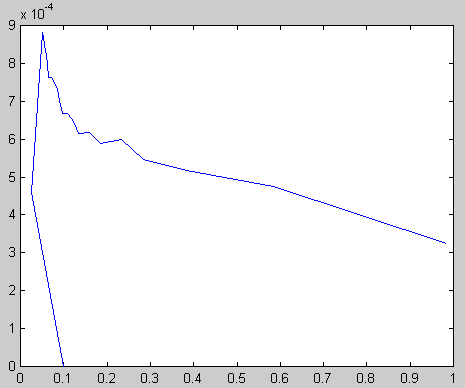
Начальное приближение **x0=0.35**, заданная точность edop=0.1, dt=0.1

нелинейный корень продолжение решение

**График значений x на каждом шаге итерации**



**График ошибки**



Результат выполнения программы: количество итераций=22, время счета=2с.

**Исследуем влияние заданной точности решения на время счета, число итераций и сходимость метода.**

Начальное приближение x0=0, заданная точность **edop=0.05**, dt=0.1.

**График значений x на каждом шаге итерации**

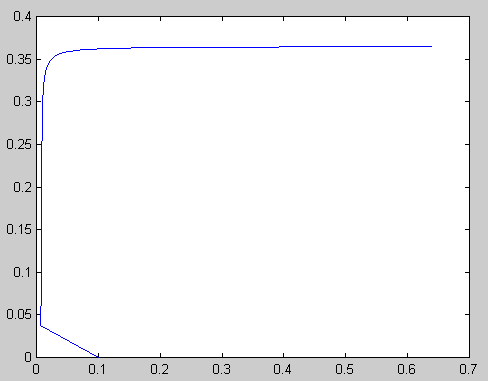
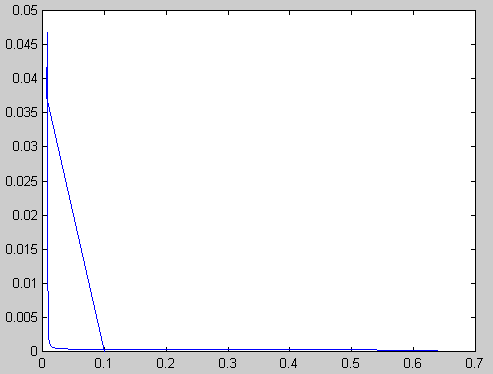


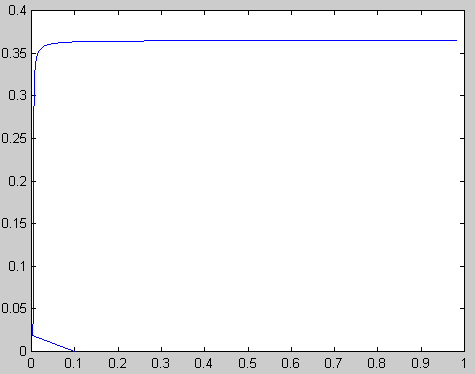
График ошибки



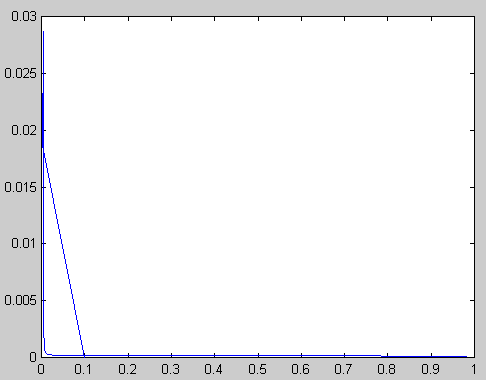
Результат выполнения программы: количество итераций=119, время счета=1,5с.

Начальное приближение x0=0, заданная точность **edop=0.03**, dt=0.1.

**График значений x на каждом шаге итерации**



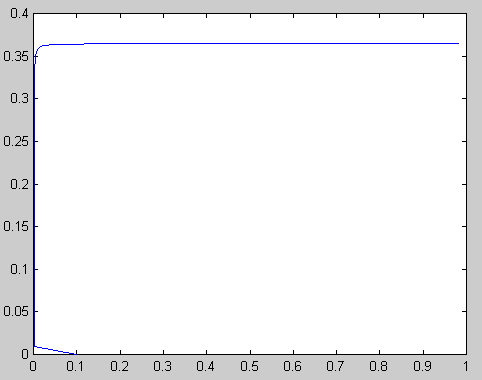
**График ошибки**



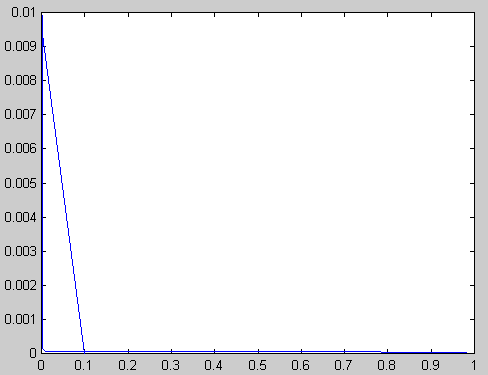
Результат выполнения программы: количество итераций=200, время счета=3с.

Начальное приближение x0=0, заданная точность **edop=0.01**, dt=0.1.

**График значений x на каждом шаге итерации**



**График ошибки**



Результат выполнения программы: количество итераций=600, время счета=5с.

# 4. Анализ результатов

Проанализировав приведенный выше графический и тестовый материал, описывающий решение систем нелинейных алгебраических уравнений методом продолжения решения по параметру можно сделать соответствующие выводы:

**1.** Метод используется для расширения области сходимости метода Ньютона, отличается простотой, не требуют слишком сложных вычислений, что является существенным преимуществом.

**2.** При задании начального приближения, находящегося далеко от точного решения, метод расходится. Если значение начального приближения выбрано близко к точному решению, то метод сходится, и чем ближе вектор начального приближения к точному решению, тем за меньшее число итераций сходится рассматриваемый метод и тем меньше время счета.

**3.** Выбор ошибки итерации также влияет на число итераций и время счета. При уменьшении значения допустимой ошибки итерации число итераций увеличивается, что необходимо для получения более точного значения решения системы нелинейных уравнений, время счета также увеличивается.

# Заключение

В работе были рассмотрены теоретические и практические характеристики метода продолжения решения по параметру. В ходе проведения тестирования и реализации метода была проведена справедливость теоретических выкладок. Наглядность результатов не оставляет сомнений в верности проведенного анализа.

Получены сведения о зависимости числа итераций, времени счета и сходимости метода от вектора начальных приближений и заданной точности решения.

Метод продолжения решения по параметру является эффективным и надежным для решения различных систем нелинейных алгебраических уравнений. Недостаток метода состоит в необходимости вычислять матрицу Якоби и решать систему нелинейных алгебраических уравнений на каждой итерации, что приводит к ограничению предельной сложности решаемых систем нелинейных алгебраических уравнений.

# Используемая литература

1. Кузьмик П.К., Маничев В.Б. Автоматизация функционального проектирования: - М.: Высшая школа, 1986. - Кн.5. Системы автоматизированного проектирования / Под ред. Норенкова И.П.
2. Сарычева О.М. Численные методы: Конспект лекций / Новосиб. гос. техн. ун.-т. - Новосибирск, 1995.