



Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»

С.И. Кадченко
О.А. Торшина

ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ НЕЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

*Утверждено Редакционно-издательским советом университета
в качестве учебного пособия*

Магнитогорск
2016

УДК 517.9
ББК 22

Рецензенты:

Заведующий кафедрой «Информационные системы» ЮУрГУ (НИУ), доктор технических наук, профессор

В.В. Мокеев

Заведующий кафедрой «Дифференциальные и стохастические уравнения» ЮУрГУ (НИУ), доктор физ.-мат. наук, доцент

С.А. Загребина

Кадченко С.И.

Численные методы решения нелинейных алгебраических уравнений [Электронный ресурс] : учебное пособие / Сергей Иванович Кадченко, Ольга Анатольевна Торшина ; ФГБОУ ВО «Магнитогорский государственный технический университет им. Г.И. Носова». – Изд. 2-е – Электрон. текстовые дан. (1,42 Мб). – Магнитогорск : ФГБОУ ВО «МГТУ им. Г.И. Носова», 2016. – 1 электрон. опт. диск (CD-R). – Систем. требования : IBM PC, любой, более 1 GHz ; 512 Мб RAM ; 10 Мб HDD ; MS Windows XP и выше ; Adobe Reader 8.0 и выше ; CD/DVD-ROM дисковод ; мышь. – Загл. с титул. экрана.

Включает теоретические сведения по методам решения нелинейных уравнений. Рассматриваются метод деления отрезка пополам, метод касательных, метод хорд, метод простой итерации и метод парабол позволяющий находить комплексные корни уравнений. Для решения систем нелинейных уравнений рассмотрен итерационный метод Ньютона. Обсуждается его сходимость. Каждый раздел включает примеры решения задач, задания для лабораторных работ и листинги программ. Пособие написано в соответствии с ФГОС ВПО по направлениям 01.03.02 «Прикладная математика и информатика», 44.03.01 «Педагогическое образование (математика)», 03.03.02 «Физика».

Может быть использовано студентами естественнонаучных специальностей для выполнения домашних заданий и подготовки к контрольным работам. Оно так же будет полезным преподавателям при подготовке к лабораторным занятиям по «Численным методам».

УДК 517.9
ББК 22

© Кадченко С.И., Торшина О.А., 2016
© ФГБОУ ВО «Магнитогорский
государственный технический
университет им. Г.И. Носова», 2016

Оглавление

Введение	5
1.Метод деления отрезка пополам	7
1.1. Теоретические сведения	7
1.2. Решение типового примера	9
1.3. Лабораторная работа. Графический и аналитический способы отделения корней нелинейного уравнения. Вычисление корня методом половинного деления.	10
1.4. Листинг программы	11
2. Метод касательных	14
2.1. Теоретические сведения	14
2.2. Решение типового примера	16
2.3. Лабораторная работа. Вычисление изолированного корня уравнения с помощью метода касательных	16
2.4. Листинг программы	17
3. Метод хорд	20
3.1. Теоретические сведения	20
3.2. Решение типового примера	23
3.3. Лабораторная работа. Вычисление изолированного корня уравнения с помощью метода хорд	24
3.4. Листинг программы	25
4. Метод простой итерации	28
4.1. Теоретические сведения	28
4.2. Решение типового примера	30
4.3. Лабораторная работа. Вычисление изолированного корня уравнения с помощью метода простой итерации	31
4.4. Листинг программы	32
5. Метод парабол	36
5.1. Теоретические сведения	36
5.2. Решение типового примера	38

5.3. Лабораторная работа. Вычисление изолированного корня уравнения с помощью метода парабол	40
5.4. Листинг программы	41
6. Решение систем нелинейных уравнений	45
6.1. Метод Ньютона	45
6.2. Локальная сходимость метода Ньютона	48
6.3. Методы с конечно-разностными производными для решения систем нелинейных уравнений	55
6.4. Решение типового примера	61
6.5. Лабораторная работа. Решение систем нелинейных уравнений методом Ньютона	64
6.6. Листинг программы	65
Заключение	68
Список литературы	69

Введение

Сфера исследований прикладной математики в современное время чрезвычайно расширилась: от исследований вопросов происхождения и эволюции Вселенной и анализов процессов микромира до расчета и проектирования машин и изучения процессов в области медицины. Многолетние исследования в области прикладной математики привели к формированию нового научного подхода – вычислительного эксперимента. Основными этапами, которого являются: формулировка математической модели, применение численных методов, составление программы проведения расчетов на ЭВМ, анализ полученных результатов. Когда разработана математическая модель необходимо решить, входящие в нее уравнения, при различных значениях параметров. Для этого используется основной аппарат вычислительно математики – численные методы. Современные численные методы являются развитленным разделом современной математики. Ее совершенствование снимает многие ограничения на сложность изучаемых математических моделей.

При разработке алгоритма исследования математической модели возникает необходимость в решении алгебраических нелинейных уравнений. Например, в экономике, развивая бизнес, необходимо узнать, когда прибыль достигнет определенного значения, в медицине при исследовании действия лекарственных препаратов, важно знать, когда концентрация вещества достигнет заданного уровня и т. д. В задачах оптимизации часто необходимо определять точки, в которых производная функции обращается в 0, что является необходимым условием локального экстремума. В статистике при построении оценок методом наименьших квадратов или методом максимального правдоподобия также приходится решать нелинейные уравнения и системы уравнений. В простейшем случае имеется функция f , заданная на отрезке $[a, b]$ и принимающая определенные значения. Каждому значению x из этого отрезка можно сопоставить число $f(x)$. Необходимо найти такое значение x , при котором $f(x) = 0$, такие x называются корнями функции f . В пособии рассматриваются методы нахождения корней уравнения, а именно метод деления отрезка пополам, метод касательных, метод хорд, метод простой итерации и метод парабол для нахождения комплексных корней. Также приводится метод Ньютона для решения систем нелинейных уравнений.

Метод Ньютона типичен для методов решения нелинейных задач. Он является итерационным процессом, генерирующим последовательность точек, которые становятся все более близкими к решению. Для большинства задач метод Ньютона сходится q -квадратично к корню одного нелинейного уравнения с одним неизвестным при условии, что имеется достаточно хорошее начальное приближение.

В пособии рассмотрены локальные алгоритмы, предназначенные для решения систем нелинейных уравнений и задач безусловной минимизации, также рассмотрены вопросы машинной реализации метода Ньютона. Доказана локальная Q -квадратичная сходимость метода Ньютона для систем нелинейных уравнений. Описан конечно-разностный вариант метода Ньютона решения нелинейных уравнений, включая вопрос о практическом выборе длины шага в конечных разностях. Также реализован алгоритм решения систем нелинейных уравнений с помощью модифицированного метода Ньютона в среде математического пакета Maple. Выявлены достоинства и недостатки метода. Достоинства:

- q -квадратичная сходимость из хорошего начального приближения при условии, что $J(x_*)$ не вырождена. Точное решение за одну итерацию для аффинной F (точное на каждой итерации для всех аффинных компонент F);
- точное решение за одну итерацию для аффинной F (точное на каждой итерации для всех аффинных компонент F).

Недостатки:

- отсутствие глобальной сходимости для многих задач;
- необходимость в $J(x_k)$ на каждой итерации;
- необходимость в решении на каждой итерации системы линейных уравнений, которая может быть вырожденной или плохо обусловленной.

В рассмотренных задачах на практике видно, что метод Ньютона является очень эффективным для решения систем нелинейных уравнений, так как имеет высокую скорость сходимости.

Наряду с серьезным теоретическим материалом в пособии большое внимание уделяется вопросам программной реализации численных методов.

1.Метод деления отрезка пополам

1.1. Теоретические сведения

Рассмотрим множество X , элементы которого обозначим через x , и множество Y с элементами y . Природа элементов может быть любой: это могут быть числа, функции и т. д. т.

Df 1. Если каждому элементу x из X ставится в соответствие некоторый элемент $y = A(x)$ множества Y , то говорят, что на множестве X задан оператор A .

Пусть взят какой-либо элемент y_0 из Y и нужно найти такие элементы x из X , для которых y_0 является изображением. Такую задачу можно записать в форме операторного уравнения

$$A(x) = y_0. \quad (1.1.1)$$

Нас будет интересовать уравнения (1.1.1) с численными неизвестными и системы таких уравнений, когда множества X и Y являются числовыми пространствами конечной размерности. В этом случае уравнение (1.1.1) можно привести к виду

$$f(x) = 0. \quad (1.1.2.)$$

Нелинейные уравнения можно разделить на 2 класса - алгебраические и трансцендентные. Функция f называется алгебраической, если для получения её значения нужно выполнить арифметические операции и возведение в степень с рациональным показателем. Примеры трансцендентных функций - показательная, логарифмическая, тригонометрические, обратные тригонометрические и др.

Решить уравнение - значит найти все его корни, то есть те значения x , которые обращают уравнение (1.1.2) в тождество, или доказать, что корней нет. Если алгебраическое или трансцендентное уравнение достаточно сложно, то довольно редко удастся точно найти его корни. Кроме того, в некоторых случаях уравнение может содержать коэффициенты, известные лишь приблизительно, поэтому сама задача о точном нахождении корней теряет смысл. В таких случаях применяют численные (приближенные) методы решения.

Методы решения нелинейных уравнений делятся на две группы:

1. *точные методы;*
2. *итерационные методы.*

Точные методы позволяют записать корни в виде некоторого конечного соотношения (формулы). Из школьного курса алгебры известны такие методы для решения тригонометрических, логарифмических, показательных, а также простейших алгебраических уравнений.

Как известно, многие уравнения и системы уравнений не имеют аналитических решений. В первую очередь это относится к большинству трансцендентных уравнений. Доказано также, что нельзя построить формулу, по которой можно было бы решить произвольное алгебраическое уравнение степени выше четвертой. Кроме того, в некоторых случаях уравнение содержит коэффициенты, известные лишь приблизительно, и, следовательно, сама задача о точном определении корней уравнения теряет смысл. Для их решения используются *итерационные методы* с заданной степенью точности.

Наиболее часто используется следующий критерий остановки итерационного процесса: $|x_{n+1} - x_n| < \varepsilon$, т.е. разница между соседними итерациями становится малой. Также для окончания итерационного процесса используется условие $|f(x_n)| < \varepsilon$, где $f(x_n)$ – невязка метода.

Прежде чем использовать приближенный метод, уравнение надо исследовать его на наличие корней и уточнить, где эти корни находятся, т.е. найти интервалы изоляции корней. *Интервалом изоляции корня* называется отрезок, на котором корень уравнения существует и единственен.

Необходимое условие существования корня уравнения на отрезке $[a, b]$: Пусть $f(x)$ непрерывна и $f(a)f(b) < 0$ (т.е., на концах интервала функция имеет разные знаки). Тогда внутри отрезка $[a, b]$ существует хотя бы один корень уравнения $f(x) = 0$.

Достаточное условие единственности корня на отрезке $[a, b]$:

Корень будет единственным, если $f(a)f(b) < 0$ и $f(x)$ не меняет знак на отрезке $[a, b]$, т.е. $f(x)$ – монотонная функция, в этом случае отрезок $[a, b]$ будет интервалом изоляции.

Если корней несколько, то для каждого нужно найти интервал изоляции.

Метод деления отрезка пополам (дихотомии).

Суть метода половинного деления заключается в делении интервала $[a, b]$ пополам точкой $c = \frac{a+b}{2}$ и отбрасывании той части интервала, в которой отсутствует корень, т.е. условие $f(a)f(b) < 0$ не выполняется. Оставшаяся часть является новым отрезком, и деление пополам продолжается до тех пор, пока оставшийся отрезок не будет сравним с заданным параметром точности ε , т.е. $b_n - a_n < \varepsilon$.

Находить интервалы изоляции можно *методом перебора*. В начале задаются начальное значение аргумента $x = a$ и шаг h , который при этом определяет и точность нахождения интервала. Пока выполняется условие $f(a)f(b) < 0$ аргумент x увеличиваем на шаг h ($x = x + h$). Если произведение $f(a)f(b) < 0$ становится отрицательным, то на интервале $[x, x + h]$ существует решение уравнения.

1.2. Решение типового примера

Найти первый корень уравнения

$$f(x) = x^3 - 6x^2 + 3x + 11 = 0$$

с точностью $\varepsilon = 0,05$.

Вычисления выполним при помощи электронной таблицы Excel, задавая начальные значения концов интервала изоляции и формул для выполнения итераций. Результаты оформляются в виде таблицы 1, где во второй строке во втором и третьем столбцах заданы a_0 , b_0 – начальные границы интервала изоляции корня;

Таблица 1 Результаты расчетов

	a	b	c	f(a)	f(c)	b-a
1	-2	-1	-1,5	-27	-10,4	1
2	-1,5	-1	-1,25	-10,4	-4,08	0,5
3	-1,25	-1	-1,13	-4,08	-1,39	0,25
4	-1,13	-1	-1,06	-1,39	-0,16	0,125
5	-1,06	-1	-1,03	-0,16	0,429	0,063
6	-1,06	-1,03	-1,05	-0,16	0,136	0,031

С вычисляется по формуле

$$c_i = \frac{a_i + b_i}{2}, i = 1, 2, 3, \dots,$$

а a_i, b_i в свою очередь определяются как

$$a_i = \begin{cases} c_{i-1}, & \text{если } f(a_{i-1}) \cdot f(c_{i-1}) > 0; \\ a_{i-1}, & \text{в противном случае;} \end{cases}$$

$$b_i = \begin{cases} b_{i-1}, & \text{если } f(a_{i-1}) \cdot f(c_{i-1}) > 0; \\ c_{i-1}, & \text{в противном случае.} \end{cases}$$

В результате расчета приближенное значение первого корня $x_1 \approx -1,047$ при заданной величине точности $\varepsilon = 0,05$.

1.3. Лабораторная работа. Графический и аналитический способы отделения корней нелинейного уравнения. Вычисление корня методом половинного деления

Задание:

1) Исследовать уравнение $f(x)=0$ на отрезке $[a; b]$ на существование и единственность корня с помощью аналитического и графического методов.

2) Вычислить корень методом половинного деления с погрешностью $\varepsilon = 0,01$.

Варианты заданий:

1. $f(x) = x^2 - 2x - 5, a = 1, b = 3$;
2. $f(x) = x^3 - 3x^2 - 10, a = 1, b = 2$;
3. $f(x) = x^3 - 3x^2 - 10, a = 3, b = 4$;
4. $f(x) = x^4 - 4x^3 + 3x^2 - 2x - 2, a = -1, b = 0$;
5. $f(x) = x^4 - 5x^3 - 4x^2 - 3x + 12, a = -4, b = -3$;
6. $f(x) = x^4 - 3x^3 + 3x^2 - 12, a = -1, b = 0$;
7. $f(x) = x^4 - 8x^3 - 2x^2 + 16x - 3, a = 0, b = 1$;
8. $f(x) = x^4 - 5x^3 - 4x^2 - 3x + 12, a = 1, b = 2$;
9. $f(x) = x^4 - 3x^3 + 3x^2 - 12, a = 2, b = 3$;
10. $f(x) = x^4 - 8x^3 - 2x^2 + 16x - 3, a = -2, b = -1$;
11. $f(x) = x^3 - 2x^2 - 4x + 7, a = -2, b = -1$;
12. $f(x) = x^3 - 0,2x^2 + 0,5x + 1,5, a = -1, b = 0$;
13. $f(x) = x^3 - 2x^2 - 4x + 7, a = 1, b = 2$;
14. $f(x) = x^5 + 2x + 5, a = -2, b = 0$;
15. $f(x) = x^3 - 2x^2 - 4x + 7, a = 2, b = 3$..

1.4. Листинг программы

```
> Digits:=25;
> f:=(x)->x^3-6*x^2+3*x+11: процедура функция определяющая
исходное уравнение
plot(f(x),x=-2..7); построение графика функции  $y = f(x)$ 
> a:=-2: начальная точка поиска
> B:=12: конечная точка поиска
> h:=0.1: шаг поиска
> EPS:=1e-21: точность вычисления нахождения интервала:
> K:=501: максимальное количество итераций
> процедура поиска интервалов изоляции
> POISK_INTERVALA:=proc(a,h,B) local fa,fb:
>                               global an,bn,prodolgenie:
> fa:=f(a):
> bn:=a+h:
> fb:=f(bn):
```

```

> while evalf(fa*fb) > 0 do
>   an:=bn:
>   fa:=f(an):
>   bn:=an+h:
>   fb:=f(bn):
>   if bn >= B then prodolgenie:=1: printf( ` ДОСТИГНУТА
КОНЕЧНАЯ ТОЧКА ПОИСКА \n` ): RETURN(): fi:
>   od:
> lprint( ` an=`,an,` bn=`,bn):
> RETURN():
> end:
> процедура вычисления корней уравнения  $f(x) = 0$  методом
деления отрезка пополам
> POPOLAM:=proc(an,bn,EPS) local ak,bk,fa,fb,fc,k:
>   global c:
>   ak:=an:
>   bk:=bn:
>   fa:=f(an):
>   fb:=f(bn):
>   c:=evalf((bn+an)/2): fc:=f(c):
>   k:=1:
>   while abs(fa-fb) > EPS and k <=K do
>     if fa*fc > 0 then ak:=c:
>       else bk:=c:
>     fi:
>     fa:=f(ak): fb:=f(bk):
>     c:=evalf((bk+ak)/2):
>     fc:=f(c):
>     k:=k+1:
>   od:
> printf( ` количество итераций = %4d | c = %24.18f | НЕВЯЗКА
f(c) = %24.18e \n`,k-1,c,abs(f(c))):
> RETURN():
> end:
> вызов процедуры POISK_INTERVALA
> prodolgenie:=0: индекс окончания поиска

```

```

> n:=1:
> while prodolgenie = 0 do
>   POISK_INTERVALA(a,h,B): уточнение
>   if prodolgenie = 0 then printf('  номер корня уравнения = %4d |
` ,n): POPOLAM(an,bn,EPS): fi: метод деления отрезка попола
>   a:=bn+h:
>   n:=n+1:
> od:
>
25
` an=`, -1.1,
` bn=`, -1.0
номер корня уравнения = 1 |
количество итераций = 71 |
c = -1.054084215386052227 |
НЕВЯЗКА f(c) = 6.000000000000000000e-23
` an=`, 2.1,
` bn=`, 2.2
номер корня уравнения = 2 |
количество итераций = 70 |
c = 2.111264157590213800 |
НЕВЯЗКА f(c) = 1.500000000000000000e-22
` an=`, 4.9,
` bn=`, 5.0
номер корня уравнения = 3 |
количество итераций = 71 |
c = 4.942820057795838427 |
НЕВЯЗКА f(c) = 4.200000000000000000e-22
ДОСТИГНУТА КОНЕЧНАЯ ТОЧКА ПОИСКА

```

2. Метод касательных

2.1. Теоретические сведения

Пусть для уравнения $f(x)=0$ отрезок $[a, b]$ является интервалом изоляции, причем $f'(x)$ и $f''(x) \forall x \in [a, b]$ непрерывны и сохраняют постоянные знаки. Рассмотрим случай, когда $f'(x)f''(x) > 0$, т. е. $f'(x)$ и $f''(x)$ имеют одинаковые знаки на отрезке $[a, b]$. Тогда возможны два случая построения кривой $y = f(x)$ в точке $B_0(b, f(b))$. Уравнение касательной к графику функции $y = f(x)$ в точке B_0 имеет вид $y - f(b) = f'(b)(x - b)$. В качестве очередного приближения к корню уравнения возьмем точку пересечения касательной с осью Ox .

Полагая $y = 0$, найдем $x_1 = b - \frac{f(b)}{f'(b)}$. Далее $x^* \in [a, x_1]$.

Применяя метод еще раз для отрезка $[a, x_1]$, получим

$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$. В результате получается рекуррентная формула

приближения к корню x^* уравнения $f(x) = 0$:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \in N.$$

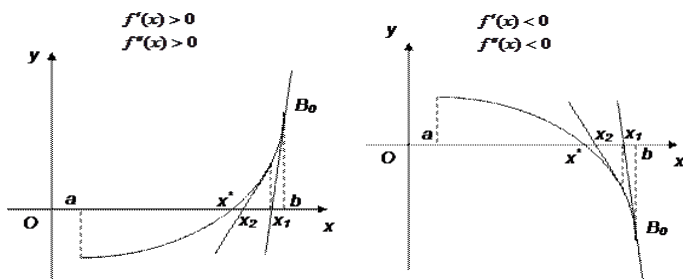


Рисунок 1 Геометрическая интерпретация метода касательных для случая

$$f'(x)f''(x) > 0.$$

В этом случае в качестве начального приближения выбрана точка $x_0 = b$. При этом приближение к корню происходит с правой стороны, поэтому мы находим приближенное значение корня с избытком.

Пусть теперь $f'(x)f''(x) < 0$. Тогда возможны на отрезке $[a, b]$ два случая построения графика функции $y = f(x)$ см. рис. 2.

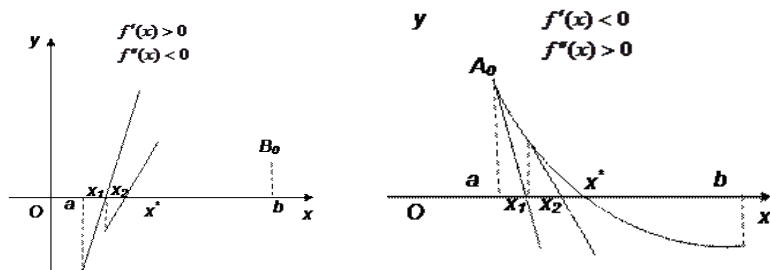


Рисунок 2 Геометрическая интерпретация метода касательных для случая $f'(x)f''(x) < 0$.

Рассуждая аналогично, найдем рекуррентную формулу вычисления приближений к корню, аналогичную первому случаю:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \in N.$$

В данном случае в качестве начального приближения к корню выбираем точку $x_0 = b$. Приближение к корню происходит с левой стороны, поэтому находим приближенное значение корня с недостатком. Заметим, что вычислительные формулы метода отличаются друг от друга только выбором начального приближения: в первом случае за x_0 принимаем конец b отрезка, во втором – конец a .

Убедитесь сами, что при выборе начального приближения корня можно руководствоваться правилом: за исходную точку следует выбрать тот конец отрезка $[a, b]$, в котором знак функции совпадает со знаком второй производной (см. рисунки 1, 2). Условие окончания вычислительного процесса: $|x_{n+1} - x_n| < \varepsilon$, где ε – заданная точность.

2.2. Решение типового примера

Найти с помощью метода касательных корень уравнения $x^3 - 6x^2 + 3x + 11 = 0$, лежащий на интервале $[4, 5]$, с точностью $\varepsilon = 0,001$.

Убедимся, что $f''(x)$ не меняет знака на этом отрезке.
 $f''(x) = 6x - 12$ $f''(x) > 0$ при $x > 2$, т.е. $f''(x) > 0$ на интервале $[4, 5]$. Так как $f(5) = 1 > 0$, то на этом конце знаки $f(x)$ и $f''(x)$ совпадают и $x_0 = 5$.

Вычисления оформим в виде таблицы:

Таблица 2 Результаты расчетов

Номер итерации	x_k	$f(x_k)$	$f'(x_k)$	$ x_{k+1} - x_k $
0	5	1	18	
1	4.944444	0.027606	17.00926	0.055556
2	4.942821	2.33E-05	16.98059	0.001623
3	4.94282	1.66E-11	16.98057	1.37E-06

Здесь $f(x_k) = x_k^3 - 6x_k^2 + 3x_k + 11 = 0$, $f'(x_k) = 3x_k^2 - 12x_k + 3$, $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$.

В качестве корня можно взять приближенное значение: $x = 4,943$. Из таблицы видно, что процесс сошелся уже на второй итерации.

2.3. Лабораторная работа. Вычисление изолированного корня уравнения с помощью метода касательных

Задание: определить с точностью 0,001 корень уравнения $f(x) = 0$, принадлежащий заданному отрезку $[a; b]$ с помощью метода касательных. Сделать выводы.

Варианты заданий:

1. $f(x) = x^4 - 5x^3 - 4x^2 - 3x + 12$, $a = 1$, $b = 2$;
2. $f(x) = x^4 - 3x^3 - 10$, $a = 2$, $b = 4$;

3. $f(x) = x^4 - 4x^2 + 5x - 7$, $a = 1$, $b = 3$;
4. $f(x) = x^4 + 8x^2 = 3$, $a = 0$, $b = 2$;
5. $f(x) = x^4 - 3x^3 + 2x^2 - x + 0,5$, $a = 0$, $b = 3$;
6. $f(x) = -x^4 + 5x^3 + 4x^2 + x - 3$, $a = 0$, $b = 2$;
7. $f(x) = x^3 + 5x^2 - 20x + 3$, $a = 1$, $b = 3$;
8. $f(x) = x^4 - x^3 + 9x^2 + 8x - 12$, $a = -2$, $b = 0$;
9. $f(x) = 3x^4 - x^3 + 12x^2 - x - 1$, $a = -1$, $b = 1$;

10. $f(x) = x^4 + 3x^3 - 10x^2 - 7x - 9$, $a = 2$, $b = 4$;
11. $f(x) = x^4 - 6x^2 - 8x - 1$, $a = -2$, $b = 0$;
12. $f(x) = -x^4 + x^3 + 6x^2 + 8x - 10$, $a = 2$, $b = 4$;
13. $f(x) = -x^3 + 5x^2 - 7x + 1$, $a = -1$, $b = 1$;
14. $f(x) = x^4 + 4x^3 + 6x^2 + 12x - 10$, $a = 0$, $b = 1$;
15. $f(x) = x^4 - 2x^3 + 8x^2 + 6x - 3$, $a = 3$, $b = 5$.

2.4. Листинг программы

```

> Digits:=55:
> f:=(x)->x^4-6*x^2+3*x+11: процедура функция определяющая
исходное уравнение
plot(f(x),x=-2..7); построение графика функции y = f ( x )
f:=(x)->evalf(sin(x)):
> pf :=(x)->diff(f(x),x): вычисление производной от функции y = f ( x )
> ppf:=(x)->diff(f(x),x$2): вычисление производной от функции y = f ( x )
)
> a:=-2: начальная точка поиска
> B:=12: конечная точка поиска
> h:=0.11111111: шаг поиска
> K:=501: максимальный номер итераций
> EPS:=1e-45: точность вычисления нахождения интервала:
> процедура поиска интервалов изоляции
> POISK_INTERVALA:=proc(a,h,B) local fa,fb:
> global an,bn,prodolgenie:
> fa:=f(a):
> bn:=a+h: fb:=f(bn):
> while fa*fb > 0 do
> an:=bn: fa:=f(an):

```

```

> bn:=an+h: fb:=f(bn):
> if evalf(bn) >= B then prodolgenie:=1: printf( ` ДОСТИГНУТА
КОНЕЧНАЯ ТОЧКА ПОИСКА \n`): RETURN(): fi:
> od:
> lprint( ` an=`,an, ` bn=`,bn):
> RETURN():
> end:
> процедура вычисления корней уравнения  $f(x) = 0$  методом
касательных
> KASATELNIE:=proc(a,b,EPS) local fa,fb,ppfa,ppfb,xn,xn1,k:
>   global c:
>   fa:=f(a): ppfa:=evalf(subs(x=a,pf(x))):
>   fb:=f(b): ppfb:=evalf(subs(x=b,pf(x))):
>   if fa*ppfa > 0 then xn:=a:
>     elif evalf(fb*ppfb) > 0 then xn:=b:
>     else printf( ` НАЧАЛЬНАЯ ТОЧКА РАСЧЕТОВ НЕ
ВЫБРАНА`):
>     fi:
>     xn1:=1e25:
>     k:=0:
>     while abs(xn1-xn) > EPS and k <= K do
>       k:=k+1:
>       xn:=xn1:
>       xn1:=evalf(xn-f(xn)/subs(x=xn,pf(x))):
>     od:
>     c:=xn:
>     printf( ` k = %4d | c = %24.18e | НЕВЯЗКА f(c) = %24.18e |
abs(xn1-xn) = %24.18e \n`,k-1,c,abs(f(c)),abs(xn1-xn)):
>     RETURN():
>   end:
>

```

```

> prodolgenie:=0: индекс окончания поиска
> n:=1:
> while prodolgenie = 0 do

```

```

> POISK_INTERVALA(a,h,B): уточнение
> if prodolgenie = 0 then printf(` n = %4d `,n):
KASATELNIE(an,bn,EPS): fi: метод деления отрезка попола
> a:=bn+h:
> n:=n+1:
> od:
>
` an=`, -1.11111112, ` bn=`, -1.00000001
n = 1 k = 146 | c = 4.942820057795838427e+00 | НЕВЯЗКА f(c) =
2.2500000000000000000e-51 | abs(xn1-xn) = 1.330000000000000000e-52
` an=`, 2.111111107, ` bn=`, 2.22222218
n = 2 k = 146 | c = 4.942820057795838427e+00 | НЕВЯЗКА f(c) =
2.2500000000000000000e-51 | abs(xn1-xn) = 1.330000000000000000e-52
` an=`, 4.88888882, ` bn=`, 4.99999993
n = 3 k = 146 | c = 4.942820057795838427e+00 | НЕВЯЗКА f(c) =
2.2500000000000000000e-51 | abs(xn1-xn) = 1.330000000000000000e-52
ДОСТИГНУТА КОНЕЧНАЯ ТОЧКА ПОИСКА

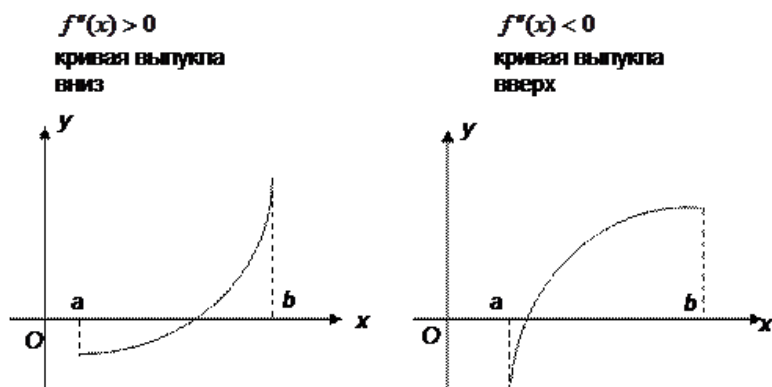
```

3. Метод хорд

3.1. Теоретические сведения

Пусть на отрезке $[a, b]$ функция непрерывна, принимает на концах отрезка значения разных знаков, а производная $f'(x)$ сохраняет знак. В зависимости от знака второй производной возможны следующие случаи расположения кривых (рис. 3).

$f(a) < 0, f(b) > 0, f'(x) > 0$ - функция возрастает



$f(a) > 0, f(b) < 0, f'(x) < 0$ - функция убывает

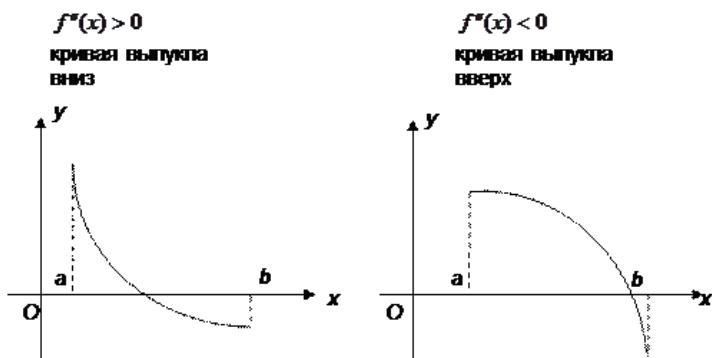


Рисунок 3 Возможные случаи расположения кривых

График функции проходит через точки $A_0(a, f(a))$ и $B_0(b, f(b))$. Искомый корень уравнения (точка X^*) нам неизвестен, вместо него возьмем точку X_1 пересечения хорды A_0B_0 с осью абсцисс (см. рис. 4). Это и будет приближенное значение корня.

$$\begin{aligned} f(a) < 0, f(b) > 0, \\ f'(x) > 0, f''(x) > 0 \end{aligned}$$

$$\begin{aligned} f(a) > 0, f(b) < 0, \\ f'(x) < 0, f''(x) < 0 \end{aligned}$$

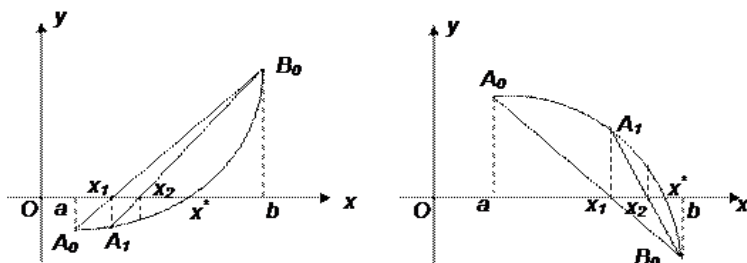


Рисунок 4 Геометрическая интерпретация метода хорд для случая

$$f'(x)f''(x) > 0.$$

Рассмотрим случай, когда $f'(x)$ и $f''(x)$ имеют одинаковые знаки (рис. 4). Из аналитической геометрии известно, что уравнение прямой, проходящей через две точки с координатами (x_1, y_1) и (x_2, y_2) имеет

вид $\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}$. Тогда уравнение хорды A_0B_0 запишется в

виде: $\frac{x - a}{b - a} = \frac{y - f(a)}{f(b) - f(a)}$. Найдем значение $x = x_1$, для которого

$$y = 0: x_1 = a - \frac{f(a)(b - a)}{f(b) - f(a)}.$$

Теперь корень находится на отрезке $[x_1, b]$. Применим метод хорд к этому отрезку. Продолжая этот процесс,

получим рекуррентную формулу вычисления приближений к корню

$$x_{n+1} = x_n - \frac{f(x_n)(b - x_n)}{f(b) - f(x_n)}.$$

В этом случае конец b отрезка $[a, b]$ остается неподвижным, а конец a перемещается. Таким образом, получаем расчетные формулы метода хорд:

$$x_{n+1} = x_n - \frac{f(x_n)(b - x_n)}{f(b) - f(x_n)}.$$

Вычисления очередных приближений к точному корню уравнения продолжается до тех пор, пока не достигнем заданной точности, т.е. должно выполняться условие: $|x_{n+1} - x_n| < \varepsilon$, где ε - заданная точность.

Теперь рассмотрим случай, когда первая и вторая производные имеют разные знаки, т.е. $f'(x)f''(x) < 0$ (рис. 5).

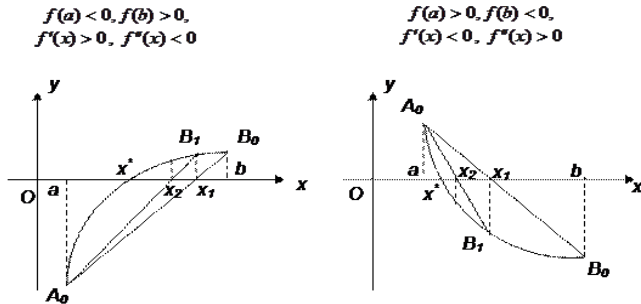


Рисунок 5 Геометрическая интерпретация метода хорд для случая

$$f'(x)f''(x) > 0.$$

Соединим точки $A_0(a, f(a))$ и $B_0(b, f(b))$. Хордой A_0B_0 . Точку пересечения хорды с осью Ox будем считать первым приближением корня. В этом случае неподвижным концом отрезка будет являться конец a . Уравнение хорды A_0B_0 : $\frac{x-b}{a-b} = \frac{y-f(b)}{f(a)-f(b)}.$

Отсюда найдем, полагая $y = 0$: $x_1 = b - \frac{f(b)(b-a)}{f(b) - f(a)}$ Теперь

корень уравнения $x^* \in [a, x_1]$. Продолжая и т.д., получим

$x_{n+1} = x_n - \frac{f(x_n)(x_n - a)}{f(x_n) - f(a)}$. Расчетные формулы метода:

$x_{n+1} = x_n - \frac{f(x_n)(x_n - a)}{f(x_n) - f(a)}$, $x_0 = b$. Условие окончания

вычислений: $|x_{n+1} - x_n| < \varepsilon$, где ε - заданная точность. Практический выбор той или иной формулы осуществляется, пользуясь следующим правилом: неподвижным концом отрезка является тот, для которого знак функции совпадает со знаком второй производной.

3.2. Решение типового примера

Найти два корня уравнения $x^3 - 6x^2 + 3x + 11 = 0$ с помощью метода хорд.

Концы интервала изоляции для первого корня $a = -2$ и $b = -1$, соответственно. Проверим знак выражения $f(a)f''(a)$ для данного уравнения:

$$((-2)^3 - 6(-2)^2 + 3(-2) + 11)(6(-2) - 12) = (-27)(-24) > 0.$$

Таким образом, расчет ведется по формулам: $x_0 = b$ и

$$x_{k+1} = a - \frac{f(a)(x_k - a)}{f(x_k) - f(a)}.$$

В результате получим таблицу:

Таблица 3 Результаты расчетов

Номер итерации	x_k	$f(x_k)$	$ x_{k+1} - x_k $
0	-1	1	-
1	-1.03571	0.345618	0.035714
2	-1.0479	0.117007	0.012187
3	-1.05201	0.039334	0.004108
4	-1.05339	0.013192	0.001379
5	-1.05385	0.004421	0.000462

Заданная точность достигнута на пятой итерации.

Для второго корня $a = 4$, $b = 5$, и
 $f''(4)f(4) = (4^3 - 6 \cdot 4^2 + 3 \cdot 4 + 11)(6 \cdot 4 - 12) = -9 \cdot 12 < 0$,

следовательно, расчет ведется по вторым формулам: $x_0 = a$ и

$$x_{k+1} = x_k - \frac{f(x_k)(b - x_k)}{f(b) - f(x_k)}.$$

Результаты вычислений показаны ниже:

Таблица 4 Результаты вычислений

Номер итерации	x_k	$f(x_k)$	$ x_{k+1} - x_k $
0	4	-9	-
1	4.9	-0.711	0.9
2	4.941555	-0.02147	0.041555
3	4.942783	-0.00062	0.001229
4	4.942819	-1.8E-05	3.57E-05

Заданная точность достигнута на четвертой итерации.

3.3. Лабораторная работа. Вычисление изолированного корня уравнения с помощью метода хорд

Задание: определить с помощью метода хорд корень уравнения $f(x) = 0$, принадлежащий заданному отрезку $[a; b]$ с точностью 0,001. Отрезок выбрать самостоятельно. Сделать выводы. Варианты заданий:

- $x e^x + x^2 - 1 = 0$;
- $4^x - 5x - 2 = 0$;
- $\ln(x+1) - x^3 + 1 = 0$;
- $x^4 - 2x - 1 = 0$;
- $\operatorname{tg} x - 5x^2 + 1 = 0, \quad x \in [-1, 1]$;

6. $3\sqrt{x+1} - e^x - 0,5 = 0;$
7. $10^x - 5x - 2 = 0;$
8. $\ln(x+2) - x^4 + 0,5 = 0;$
9. $x^6 - 5x - 2 = 0;$
10. $\sqrt{x+2} - 2\cos x = 0;$
11. $\lg(x+1) - x + 0,5 = 0;$
12. $x^6 - 5x^3 - 2 = 0;$
13. $\lg(2x+1) - x^3 + 1 = 0;$
14. $x^5 - 7x^2 + 3 = 0;$
15. $x \lg(x+2) + x^2 - 1 = 0.$

3.4. Листинг программы

```

> Digits:=155:
> f:=(x)->x^3-6*x^2+3*x+11: процедура функция определяющая
исходное уравнение
plot(f(x),x=-2..7); построение графика функции  $y = f(x)$ 
f:=(x)->evalf(sin(x)):
> ppf:=(x)->diff(f(x),x$2): вычисление производной от функции  $y = f(x)$ 
)
> a:=-2: начальная точка поиска
> B:=12: конечная точка поиска
> h:=0.11111111: шаг поиска
> EPS:=1e-6: точность вычисления нахождения интервала:
> процедура поиска интервалов изоляции
> POISK_INTERVALA:=proc(a,h,B) local fa,fb:
> global an,bn,prodolgenie:
> fa:=f(a):
> bn:=a+h: fb:=f(bn):
> while fa*fb > 0 do
> an:=bn: fa:=f(an):

```

```

> bn:=an+h: fb:=f(bn):
> if evalf(bn) >= B then prodolgenie:=1: printf( ' ДОСТИГНУТА
КОНЕЧНАЯ ТОЧКА ПОИСКА \n'): RETURN(): fi:
> od:
> printf( ' an = %24.18f bn = %24.18f \n',an,bn):
> RETURN():
> end:
> процедура вычисления корней уравнения  $f(x) = 0$  методом
хорд
> XORDI:=proc(a,b,EPS) local fa,fb,ppfa,ppfb,xn,xn1,k,q:
>   global c:
>   fa:=f(a): ppfa:=evalf(subs(x=a,ppf(x))):
>   fb:=f(b): ppfb:=evalf(subs(x=b,ppf(x))):
>   if abs(fa) < EPS then c:=a: RETURN(): fi:
>   if abs(fb) < EPS then c:=b: RETURN(): fi:
>
>   if evalf(fa*ppfa) > 0 then xn:=b: q:=1:
>   elif evalf(fb*ppfb) > 0 then xn:=a: q:=2:
>   else q:=3: printf( ' НАЧАЛЬНАЯ ТОЧКА РАСЧЕТОВ НЕ
ВЫБРАНА q = %1d \n',q): c:=a: RETURN():
>   fi:
>   xn1:=0:
>   k:=0:
>   while abs(xn1-xn) > EPS do
>     k:=k+1:
>     xn:=xn1:
>     if q = 1 then xn1:=evalf(xn-f(xn)*(xn-a)/(f(xn)-fa)):
>     elif q = 2 then xn1:=evalf(xn-f(xn)*(b-xn)/(fb-f(xn))):
>     fi:
>   od:
>   printf( ' k = %4d ',k):
>   c:=xn:
>   RETURN():
> end:
>

```

```

> prodolgenie:=0: индекс окончания поиска
> n:=1:
> while prodolgenie = 0 do
>   POISK_INTERVALA(a,h,B): уточнение
>   if prodolgenie = 0 then printf('      n = %4d `n):
>       XORDI(an,bn,EPS):
>       printf('    c = %24.18f НЕВЯЗКА f(c) = %24.18f
> \n`,c,abs(f(c))):
>   fi:
>   a:=bn+h:
>   n:=n+1:
> od:
>
  an = -1.11111112000000000000 bn = -1.00000001000000000000
    n = 1 k = 5 c = -1.054083318981025992 НЕВЯЗКА f(c) =
0.000017015813864554
  an = 2.11111110700000000000 bn = 2.22222218000000000000
    n = 2 k = 3 c = 2.111264158218863494 НЕВЯЗКА f(c) =
0.000000005634499737
  an = 4.88888882000000000000 bn = 4.99999993000000000000
    n = 3 k = 6 c = 4.942820380311766889 НЕВЯЗКА f(c) =
0.000005476505113723
ДОСТИГНУТА КОНЕЧНАЯ ТОЧКА ПОИСКА

```

4. Метод простой итерации

4.1. Теоретические сведения

С помощью эквивалентных преобразований приведем исходное уравнение $f(x)=0$ к виду, удобному для применения метода простой итерации: $x = \varphi(x)$. Выберем начальное приближение $x_0 \in [a, b]$. Следующие итерации находим по формуле: $x_{k+1} = \varphi(x_k)$. Итерационный процесс заканчивается, если $|x_{k+1} - x_k| < \varepsilon$. Представить исходное уравнение в эквивалентном виде $x = \varphi(x)$ можно разными способами. Из всевозможных таких представлений выбирают тот, который дает сходящуюся к корню последовательность вычислений. Очевидно, что $x^* = \varphi(x^*)$.

Достаточное условие сходимости: пусть $\varphi(x)$ имеет производную на отрезке $[a, b]$ и $|\varphi'(x)| \leq q < 1$ для всех x из отрезка $[a, b]$, тогда итерационный процесс сходится к корню x^* уравнения $x = \varphi(x)$ т.е. $\lim_{k \rightarrow \infty} x_k = x^*$.

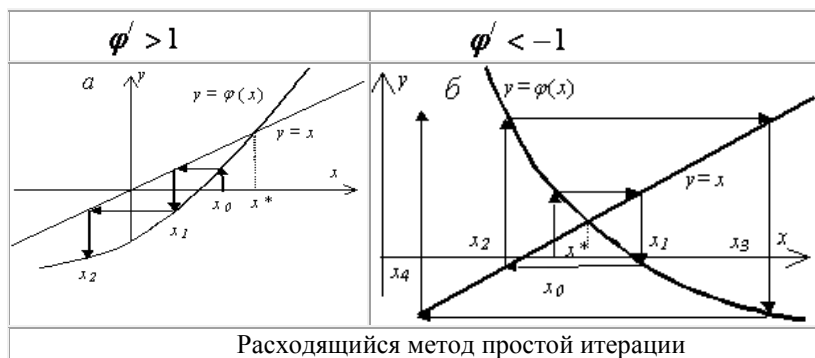
Доказательство следует из следующих оценок:

$$\begin{aligned} |x_{k+1} - x^*| &= |\varphi(x_k) - x^*| = |\varphi(x_k) - \varphi(x^*)| \leq |\varphi'(\xi)| |x_k - x^*| \leq q |x_k - x^*| \leq \\ &\leq q^2 |x_{k-1} - x^*| \leq q^3 |x_{k-2} - x^*| \leq \dots \leq q^k |b - a| \end{aligned}$$

Так как $q < 1$, то $\lim_{k \rightarrow \infty} q^k = 0$ следовательно $\lim_{k \rightarrow \infty} x_k = x^*$.

Геометрический смысл метода простой итерации.





В качестве начального приближения обычно берут середину отрезка $[a, b]$: $x_0 = \frac{a+b}{2}$.

На практике часто в качестве $\varphi(x)$ берут функцию $\varphi(x) = x - c f(x)$, где c – некоторая постоянная. Постоянную c выбирают таким образом, чтобы $|\varphi'(x)| \leq q < 1$ для всех $x \in [a, b]$. При таком выборе функции $\varphi(x)$ метод простой итерации называют методом релаксации.

Получим условия на выбор c :

$$|\varphi'(x)| = |1 - c f'(x)| < 1 \Rightarrow -1 < 1 - c f'(x) < 1 \Rightarrow -2 < -c f'(x) < 0 \Rightarrow 2 > c f'(x) > 0.$$

Таким образом, если $f'(x) < 0$, то

$$2 > -c |f'(x)| > 0 \Rightarrow -\frac{2}{|f'(x)|} < c < 0. \quad \text{Если } f'(x) > 0, \text{ то}$$

$$\frac{2}{f'(x)} > c > 0.$$

Очевидно, что знак c совпадает со знаком $f'(x)$. Часто c берут в виде: $c = \frac{2}{M+m}$, где $M = \max_{x \in [a,b]} f'(x)$, $m = \min_{x \in [a,b]} f'(x)$.

Убедимся, что такое c удовлетворяет условию сходимости.

Пусть $f'(x) > 0$ для всех $x \in [a, b]$. Тогда $M > 0$ и $m > 0 \Rightarrow c > 0$. Из условия для выбора C $2 > c f'(x) > 0$ получим

$$\frac{2}{f'(x)} - \frac{2}{M+m} = \frac{2(M+m-f'(x))}{f'(x)(M+m)} > 0,$$

т. к. $M \geq f'(x)$. Следовательно,

$$\frac{2}{f'(x)} > c > 0.$$

Пусть $f'(x) < 0$. Тогда $M < 0$ и $m < 0 \Rightarrow c < 0$ и

$$\begin{aligned} \frac{2}{M+m} - \frac{2}{|f'(x)|} &= \frac{2(|f'(x)| - M - m)}{|f'(x)|(M+m)} > 0 \\ \Rightarrow \frac{2}{f'(x)} &< c < 0. \end{aligned}$$

4.2. Решение типового примера

Найти второй корень уравнения $f(x) = x^3 - 6x^2 + 3x + 11 = 0$, лежащий на отрезке $[1, 3]$ с точностью $\varepsilon = 0,001$. Для определения значения параметра C необходимо найти максимальное и минимальное значения производной функции $f'(x)$ на заданном отрезке $[1, 3]$. Для этого необходимо найти значения $f'(x)$ на концах интервала и в точках экстремума, где $f''(x) = 0$ (если эти точки лежат на исследуемом отрезке). Далее среди этих значений выбираются максимальное и минимальное. В нашем случае

$$f'(x) = 3x^2 - 12x + 3, \quad f'(1) = 3 - 12 + 3 = -6$$

$$f'(3) = 27 - 36 + 3 = -6, \quad f''(x) = 6x - 12 = 0 \text{ при } x = 2$$

Экстремум производной находится на заданном отрезке $[1, 3]$, находим значение производной в этой точке:
 $f'(2) = 12 - 24 + 3 = -9$. Следовательно

$$M = \max(-6, -6, -9) = -6, \quad m = \min(-6, -6, -9) = -9,$$

$$c = \frac{2}{M+m} = \frac{2}{-6-9} = -\frac{2}{15}.$$

Таким образом, $\varphi(x) = x + \frac{2}{15}(x^3 - 6x^2 + 3x + 11)$.

Выберем начальное приближение $x_0 = \frac{1+3}{2} = 2$, следующие приближения вычисляются по формуле

$$x_{k+1} = x_k + \frac{2}{15}(x_k^3 - 6x_k^2 + 3x_k + 11), k=0, 1, 2, \dots$$

Условием окончания итерационного процесса является условие:

$$|x_{k+1} - x_k| < \varepsilon = 0,001 \text{ или } |f(x_k)| < 0,001.$$

Результаты вычислений оформим на рабочем листе Excel в виде таблицы:

Таблица 5 Результаты расчетов

Номер итерации	x_k	$ x_{k+1} - x_k $
0	2	
1	2,133333	0,133333
2	2,106983	0,026351
3	2,1121	0,005117
4	2,111101	0,000999
5	2,111296	0,000195

Здесь формулы для вычисления x_1 имеют вид $= B2 + \frac{2}{15} * (B2^3 - 6 * B2^2 + 3 * B2 + 11)$, а для $|x_1 - x_0| = \text{ABS}(B3 - B2)$. Остальные вычисления получаются копированием формул.

4.3. Лабораторная работа. Вычисление изолированного корня уравнения с помощью метода простой итерации

Задание: определить с точностью 0,001 корень уравнения $f(x) = 0$, принадлежащий отрезку $[a; b]$ с помощью метода простой итерации. Отрезок выбрать самостоятельно. Сделать выводы. Варианты заданий:

1. $2^x - x^2 - 0,5 = 0;$

2. $\ln(x+2) - x^2 = 0;$
3. $\sqrt{1-x^2} - e^x + 0,1 = 0;$
4. $x^3 + x^2 - x - 0,5 = 0;$
5. $\cos x + 0,25x - 0,5 = 0;$
6. $e^x - 2x - 2 = 0;$
7. $2^x + x^2 - 2 = 0;$
8. $\ln(x+1) - 2x^2 + 1 = 0;$
9. $x^3 + x^2 - 2x - 1 = 0;$
10. $\sin x - 2x^2 + 0,5 = 0;$
11. $e^x - x^3 + 3x^2 - 2x - 3 = 0;$
12. $3^x - 5x^2 + 1 = 0;$
13. $\ln(x+1) - 2x + 0,5 = 0;$
14. $x^3 - 2x^2 - 10x + 15 = 0;$
15. $\sin x - x^2 + 1 = 0.$

4.4. Листинг программы

> **Digits:=55:**

> **f:=(x)->x^3-6*x^2+3*x+11:** процедура функция определяющая
исходное уравнение

plot(f(x),x=-2..7); построение графика функции $y = f(x)$

f:=(x)->evalf(sin(x)):

> **pf:=(x)->diff(f(x),x):**

> **ppf:=(x)->diff(f(x),x\$2):** вычисление производной от функции $y = f(x)$
)

> **FI:=(x)->x-KONST_c*f(x):**

> **a:=-2:** начальная точка поиска

> **B:=7:** конечная точка поиска

```

> h:=0.51111111: шаг поиска
> EPS:=1e-25: точность вычисления нахождения интервала:
> K:=501: максимальный номер итерации
> процедура поиска интервалов изоляции
> POISK_INTERVALA:=proc(a,h,B) local fa,fb:
>     global an,bn,prodolgenie:
>     fa:=f(a):
>     bn:=a+h: fb:=f(bn):
>     while evalf(fa*fb) > 0 do
>         an:=bn: fa:=f(an):
>         bn:=an+h: fb:=f(bn):
>         if bn >= B then prodolgenie:=1: printf( ` ДОСТИГНУТА
КОНЕЧНАЯ ТОЧКА ПОИСКА \n`): RETURN(): fi:
>     od:
>     printf( ` an = %24.18f bn = %24.18f \n`,an,bn):
>     RETURN():
> end:
> процедура нахождения константы C
> KONSTANTA_c:=proc(a,b) local pfa,pfb,xk,pfxk,m,M:
>     global KONST_c:
>     pfa :=subs(x=a,pf(x)):
>     pfb :=subs(x=b,pf(x)):
>     xk :=solve(ppf(x)=0,x):
>     if xk > a and xk < b then pfxk:=subs(x=xk,pf(x)):
>         M:=max(pfa,pfb,pfxk):
>         m:=min(pfa,pfb,pfxk):
>         else M:=max(pfa,pfb):
>         m:=min(pfa,pfb):
>     fi:
>     KONST_c:=2/(M+m):
lprint( ` pfa=`,pfa,` pfb=`,pfb,` M=`,M,` m=`,m,`
KONST_c=`,KONST_c):
>     RETURN():

```

```

> end:
> процедура вычисления корней уравнения  $f(x) = 0$  методом
простой итерации
> PROSTAJ_ITERACHIJ:=proc(a,b,EPS) local k,xn,xn1:
>             global K:
>   xn:=(a+b)/2:
>   k:=0:
>   while abs(f(xn)) > EPS and k <= K do
>     k:=k+1:
>     xn1:=evalf(subs(x=xn,FI(xn))):
>     xn:=xn1:
>   od:
>   c:=xn:
>   printf('    k = %4d | c = %24.18e | НЕВЯЗКА f(c) = %24.18e \n`,k-
1,c,abs(f(c)))':
>   RETURN():
> end:
>

```

```

> prodolgenie:=0: индекс окончания поиска
> n:=1:
> while prodolgenie = 0 do
>   printf('  n = %4d`,n):
>   POISK_INTERVALA(a,h,B): уточнение
>   if prodolgenie = 0 then KONSTANTA_c(an,bn):
>     PROSTAJ_ITERACHIJ(an,bn,EPS):
>   fi:
>   a:=bn+h:
>   n:=n+1:
> od:
>
>

```

>

Warning, `c` is implicitly declared local to procedure
`PROSTAJ\_ITERACHIJ`

n = 1 an = -1.488888890000000000 bn = -
0.977777780000000000
k = 31 | c = -1.054084215386052227e+00 | НЕВЯЗКА f(c) =
4.400222008818711851e-26
n = 2 an = 2.088888880000000000 bn =
2.599999990000000000
k = 20 | c = 2.111264157590213800e+00 | НЕВЯЗКА f(c) =
4.729113791486465654e-26
n = 3 an = 4.644444430000000000 bn =
5.155555540000000000
k = 16 | c = 4.942820057795838427e+00 | НЕВЯЗКА f(c) =
2.272047028910675277e-26
n = 4 ДОСТИГНУТА КОНЕЧНАЯ ТОЧКА ПОИСКА

5. Метод парабол

5.1. Теоретические сведения

Метод парабол предназначен для нахождения корней алгебраического многочлена

$$P_n(z) = a_n z^n + a_{n-1} z^{n-1} + a_{n-2} z^{n-2} + \dots + a_1 z + a_0, \quad a_n \neq 0,$$

с комплексными коэффициентами. Сходимость метода в общем случае не доказана, но в практике вычислений не известно ни одного примера, когда этот метод не сходил.

Пусть заданы три произвольных различных комплексных числа z_0, z_1, z_2 . Используя эти числа как узлы интерполяции строится интерполяционный многочлен Лагранжа для $P_n(z)$. Найдя оба его корня, и за z_3 берется тот из корней, который ближе к z_2 . Следующая тройка чисел будет z_1, z_2, z_3 . Используя эти числа, строится новый интерполяционный многочлен Лагранжа, и за z_4 берется его корень, ближайший к z_3 . Получается новая тройка чисел z_2, z_3, z_4 и т. д.

Последовательность точек, построенная таким образом, всегда сходится к некоторому корню многочлена $P_n(z)$ (этот факт установлен эмпирическим путем). После того, как вычислен один из корней, он выделяется, и далее метод применяется к многочлену меньшей степени.

Получим расчетные формулы итерационного метода парабол. Допустим, что известно значение многочлена $P_n(z)$ в трех попарно различных точках z_{i-2}, z_{i-1}, z_i . По этим точкам строится интерполяционный многочлен Лагранжа для многочлена $P_n(z)$

$$L^{(i-2)}(z) = P_n(z_{i-2}) \frac{(z - z_i)(z - z_{i-1})}{(z_{i-2} - z_i)(z_{i-2} - z_{i-1})} + P_n(z_{i-1}) \frac{(z - z_i)(z - z_{i-2})}{(z_{i-1} - z_i)(z_{i-1} - z_{i-2})} + P_n(z_i) \frac{(z - z_{i-1})(z - z_{i-2})}{(z_i - z_{i-1})(z_i - z_{i-2})}. \quad (5.1.1)$$

Обозначим:

$$h = z - z_i, \quad h_i = z_i - z_{i-1}, \quad h_{i-1} = z_{i-1} - z_{i-2}, \quad (5.1.2)$$

$$\lambda = \frac{h}{h_i}, \quad \lambda_i = \frac{h_i}{h_{i-1}}, \quad \delta_i = 1 + \lambda_i.$$

Подставляя (5.1.2) в (5.1.1), получим

$$L^{(i-2)}(\lambda) = \delta_i^{-1} [P_n(z_{i-2})\lambda_i^2 - P_n(z_{i-1})\lambda_i \delta_i + P_n(z_i)\lambda_i] \lambda^2 + \\ + \delta_i^{-1} [P_n(z_{i-2})\lambda_i^2 - P_n(z_{i-1})\delta_i^2 + P_n(z_i)(\lambda_i + \delta_i)] \lambda + P_n(z_i).$$

Решаем уравнение $L^{(i-2)}(\lambda) = 0$ относительно λ , получим

$$\lambda_{1,2} = \frac{-2P_n(z_i)\delta_i}{g_i \pm \sqrt{g_i^2 - 4P_n(z_i)\lambda_i\delta_i[P_n(z_{i-2})\lambda_i - P_n(z_{i-1})\delta_i + P_n(z_i)]}} \quad (5.1.3)$$

где $g_i = P_n(z_{i-2})\lambda_i^2 - P_n(z_{i-1})\delta_i^2 + P_n(z_i)(\lambda_i + \delta_i)$.

Из двух возможных значений $\lambda_{1,2}$ берется λ_* наименьшее по модулю и далее вычисляются

$$\lambda_{i+1} = \lambda_*, \quad h_{i+1} = \lambda_{i+1}h_i, \quad z_{i+1} = z_i + h_{i+1}. \quad (5.1.4)$$

Для большей гарантии сходимости метода и для преодоления трудностей, связанных с появлением больших чисел, рекомендуется дополнительно проверять выполнение неравенства

$$|h_{i+1}| \geq \sqrt[n]{|P_n(z_i)| \frac{1}{|a_0|}}. \quad (5.1.5)$$

Если (5.1.5) выполнено, то λ берется равным единице и пересчитываются величины (5.1.4). Если неравенство не выполнено, то λ не изменяется.

После этого проверяется выполнение другого неравенства

$$|P_n(z_{i+1})| \leq 10|P_n(z_i)|. \quad (5.1.6)$$

Если оно не выполнено, то λ заменяется на $\frac{\lambda}{2}$ и пересчитываются

величины (5.1.4) и снова проверяется (5.1.6). Замена λ на $\frac{\lambda}{2}$ и пересчет (5.1.4) делается до тех пор, пока неравенство (5.1.6) впервые не будет выполнено. Окончательно полученное z_{i+1} берется за

следующее приближение к корню. После этого вместо точек Z_{i-2}, Z_{i-1}, Z_i берутся точки Z_{i-1}, Z_i, Z_{i+1} и процесс продолжается.

Таким образом, получается последовательность точек Z_k , которая всегда сходится к некоторому корню многочлена $P_n(z)$.

Метод позволяет определять лишь один, вообще говоря, случайный корень, поэтому для нахождения остальных корней необходимо найденный корень выделять и далее применять метод к многочлену меньшей на единицу степени.

Для деления многочлена на $z - z^*$, где z^* - корень многочлена $P_n(z)$ можно использовать схему Горнера. Используя метод неопределенных коэффициентов, положим

$$\begin{aligned} & a_n z^n + a_{n-1} z^{n-1} + a_{n-2} z^{n-2} + \dots + a_1 z + a_0 = \\ & = (z - z^*) (A_{n-1} z^{n-1} + A_{n-2} z^{n-2} + \dots + A_1 z + A_0) + r, \end{aligned}$$

откуда

$$\begin{aligned} A_{n-1} &= a_n, \\ A_{n-2} &= a_{n-1} + z_0 A_{n-1}, \\ A_{n-3} &= a_{n-2} + z_0 A_{n-2}, \\ &\dots\dots\dots \\ A_0 &= a_1 + z^* A_1, \\ r &= a_0 + z^* A_0. \end{aligned}$$

Аналогично находятся остальные корни.

5.2. Решение типового примера

Решить уравнение

$$f(x) = x^3 + x^2 + x - 5$$

методом парабол на отрезке $[-2; 2]$ с точностью $\varepsilon = 0,1$.

$$f'(x) = 3x^2 + 2x + 1 > 0 \text{ на всем } [-2; 2]$$

Вычислим значения на концах отрезка и в середине интервала:

$$f(a) = f(-2) = -11$$

$$f(b) = f(2) = 9$$

$$f(c) = f(0) = -5$$

Для нахождения корней необходимо найти коэффициенты А, В и С по формулам:

$$A = \frac{\frac{f(b)-f(c)}{b-c} - \frac{f(c)-f(a)}{c-a}}{b-a};$$

$$B = \frac{f(c) - f(a)}{c - a} + A \cdot (a - c);$$

$$C = f(a)$$

Подставив известные значения в формулы, получим:
 $A=1$, $B=1$ и $C=-11$.

Найдем два корня уравнения, а затем выберем тот, который удовлетворяет отрезку:

$$x_* = a - \frac{2 \cdot C}{B \pm \sqrt{B^2 - 4 \cdot A \cdot C}}$$

$$x_{1_1} = 0,857 \in [-2; 2]$$

$$x_{1_2} = -5,86 \notin [-2; 2]$$

Проверим приближенное значение корня на предмет заданной точности ε :

$$|-2 - 0,857| = 2,857 > \varepsilon$$

Рассмотрим отрезок $[-0,857; 2]$ и найдем корни, принадлежащие этому отрезку:

$$f(a) = f(x_{1_1}) = f(-0,857) = -2,78$$

$$f(b) = f(2) = 9$$

$$f(c_1) = f(1,4285) = 1,384$$

$A=4,75$, $B=5,185$, $C=-2,78$

$$x_{2_1} = -0,63 \notin [-0,857; 2]$$

$$x_{2_2} = 1,251 \in [-0,857; 2]$$

$$|0,857 - 1,251| = 0,394 > \varepsilon$$

Рассмотрим отрезок $[1,251; 2]$ и найдем корни, принадлежащие отрезку:

$$f(a) = f(x_{2_2}) = f(1,251) = -0,2262$$

$$f(b) = f(2) = 9$$

$$f(c_2) = f(1,6255) = 3,5655$$

$$A=5,85, B=7,93, C=-0,2262$$

$$x_{3_1} = 1,279 \in [1,251; 2]$$

$$x_{3_2} = -0,12 \notin [1,251; 2]$$

$$|1,251 - 1,279| = 0,028 < \varepsilon$$

Заданная точность достигнута. В итоге получим $x = 1,279$.

5.3. Лабораторная работа. Вычисление изолированного корня уравнения с помощью метода парабол

Задание: определить с точностью 0,001 корень уравнения

$f(x) = 0$, принадлежащий отрезку $[a; b]$ с помощью метода парабол.

Отрезок выбрать самостоятельно. Сделать выводы.

Варианты заданий:

1. $3x^4 + 4x^3 - 12x^2 - 5 = 0;$
2. $0,5^x + 1 = (x - 2)^2;$
3. $(x - 4)^2 \log_{0,5}(x - 3) = -1;$
4. $x^2 \cos(2x) = -1;$
5. $(x - 2)^2 2^x = -1;$
6. $((x - 2)^2 - 1)2^x = 1;$
7. $(x - 2)\cos x = 1;$
8. $(x - 2)^3 \lg(x + 11) = 1;$
9. $5\sin x = x - 1;$
10. $x^4 3^x = 2;$
11. $2\lg x - \frac{x}{3} + 1 = 0;$
12. $x - \sin x = 0,35;$
13. $\sqrt{x} - \cos(0,37 + x) = 0;$
14. $\sin(0,5 + x) = 2x - 0,5;$
15. $\ln x + (x + 1)^3 = 0.$

5.4. Листинг программы

```
> restart:
> printlevel:=1:
> katalog:='e: /МЕТОД ПАРАБОЛ/БИБЛИОТЕКА/':
> read(cat(katalog,'PARABOLA.m')):
> with(PARABOLA):
> Digits:=149:
> a_digits:=Digits:
> p_digits:=12:
> K:=31:
> EPS:=1e-44:
> st:=15:
> Pn:=array(1..st):
> zo:=array(1..st):
> for n from 1 to st do zo[n]:=evalf((2+(-1)^n*I)*n^2): od: задание
корней многочлена
> Pn[st]:=product(z-zo[k],k=1..st): составление многочлена
> ao:=coeff(Pn[st],z,st): вычисление коэффициентов при st степени
z
> for n from st by -1 to 1 do
>   if n >= 2 then METOD_PARABOL(EPS,K,Pn[n]):
>     else zi:=-Pn[n]+z:
>       Pni:=subs(z=zi,Pn[n]):
>       koliter:=0:
>   fi:
>   printf(' СТЕПЕНЬ многочлена = %2d | k_iter = %4d | Re(zi) =
%12.6f + Im(zi) = %12.6f*I | НЕВЯЗКА = %12.6e
\n',n,k_iter,Re(zi),Im(zi),abs(Pni)):
>   if st >= 2 then
>     Pn[n-1]:=quo(Pn[n],z-zi,z): деление многочлена на одночлен z - zi
>     ao:=coeff(Pn[n-1],z,n-1): вычисление коэффициентов при n-
1степени z
>     st:=st-1:
>   fi:
> od:
>
```

СТЕПЕНЬ многочлена = 15 | k_iter = 9 | Re(zi) = 2.000000 + Im(zi)
 = -1.000000*I | НЕВЯЗКА = 8.724082e-100
 СТЕПЕНЬ многочлена = 14 | k_iter = 11 | Re(zi) = 8.000000 + Im(zi)
 = 4.000000*I | НЕВЯЗКА = 1.177469e-82
 СТЕПЕНЬ многочлена = 13 | k_iter = 13 | Re(zi) = 18.000000 +
 Im(zi) = -9.000000*I | НЕВЯЗКА = 1.826868e-99
 СТЕПЕНЬ многочлена = 12 | k_iter = 14 | Re(zi) = 32.000000 +
 Im(zi) = 16.000000*I | НЕВЯЗКА = 5.460760e-80
 СТЕПЕНЬ многочлена = 11 | k_iter = 15 | Re(zi) = 50.000000 +
 Im(zi) = -25.000000*I | НЕВЯЗКА = 8.579726e-68
 СТЕПЕНЬ многочлена = 10 | k_iter = 15 | Re(zi) = 72.000000 +
 Im(zi) = 36.000000*I | НЕВЯЗКА = 1.316171e-56
 СТЕПЕНЬ многочлена = 9 | k_iter = 19 | Re(zi) = 162.000000 +
 Im(zi) = -81.000000*I | НЕВЯЗКА = 4.595868e-76
 СТЕПЕНЬ многочлена = 8 | k_iter = 14 | Re(zi) = 98.000000 + Im(zi)
 = -49.000000*I | НЕВЯЗКА = 2.411545e-96
 СТЕПЕНЬ многочлена = 7 | k_iter = 17 | Re(zi) = 200.000000 +
 Im(zi) = 100.000000*I | НЕВЯЗКА = 7.237949e-59
 СТЕПЕНЬ многочлена = 6 | k_iter = 12 | Re(zi) = 128.000000 +
 Im(zi) = 64.000000*I | НЕВЯЗКА = 8.816836e-48
 СТЕПЕНЬ многочлена = 5 | k_iter = 16 | Re(zi) = 338.000000 +
 Im(zi) = -169.000000*I | НЕВЯЗКА = 1.929889e-46
 СТЕПЕНЬ многочлена = 4 | k_iter = 12 | Re(zi) = 242.000000 +
 Im(zi) = -121.000000*I | НЕВЯЗКА = 1.221036e-86
 СТЕПЕНЬ многочлена = 3 | k_iter = 11 | Re(zi) = 288.000000 +
 Im(zi) = 144.000000*I | НЕВЯЗКА = 7.731362e-50
 СТЕПЕНЬ многочлена = 2 | k_iter = 2 | Re(zi) = 392.000000 + Im(zi)
 = 196.000000*I | НЕВЯЗКА = 9.569096e-139
 СТЕПЕНЬ многочлена = 1 | k_iter = 2 | Re(zi) = 450.000000 + Im(zi)
 = -225.000000*I | НЕВЯЗКА = 0.000000e+00

```

> PARABOLA[METOD_PARABOL]:=proc(EPS,K,POLINOM) local
k,zi2,zi1,qi,gi,Pni2,Pni1,hi,hi1,hin,lami,deli,lam1,lam2,lamn,zin,Pnin:
>
> global ao,zi,Pni,k_iter,a_digits,p_digits:
>

```

```

> k_iter:=1:
> zi2:=0: zi1:=1: zi:=2: задание начальных значений узлов
интерполяции
> Pni2:=subs(z=zi2,POLINOM): вычисление значений многочлена
в узлах интерполяции
> Pni1:=subs(z=zi1,POLINOM):
> Pni :=subs(z=zi ,POLINOM):
> if evalf(abs(Pni2)) <= EPS then zi:=zi2: Pni:=subs(z=zi,POLINOM):
RETURN(): проверка близости значений zi1 и zi2 к корню z*
>   elif evalf(abs(Pni1)) <= EPS then zi:=zi1:
Pni:=subs(z=zi,POLINOM): RETURN():
>   elif evalf(abs(Pni)) <= EPS then Pni:=subs(z=zi,POLINOM):
RETURN():
> fi:
> while evalf(abs(Pni)) >= EPS and k_iter <= K do проверка
близости значений zi к корню z*
>   hi :=zi-zi1: hi1:=zi1-zi2: вычисление начальных значений
характеристик итерации
>   lami:=hi/hi1: deli:=1+lami:
>   решение уравнения  $L^{(i-2)}(lam) = 0$ 
>   qi:=evalf(Pni2*lami^2-Pni1*deli^2+Pni*(lami+deli)):
>   gi:=evalf(sqrt(qi^2-4*Pni*lami*deli*(Pni2*lami-Pni1*deli+Pni))):
>   lam1:=-evalf(2*Pni*deli/(qi+gi)): lam2:=-evalf(2*Pni*deli/(qi-gi)):
>   if evalf(abs(lam1)) > evalf(abs(lam2)) then lami:=lam2:
>                                     else lami:=lam1:
> fi:
> вычисление характеристик итерации
> hin:=lami*hi:
> zin:=zi+hin:
> проверка неравенств для устранения появления больших чисел
> if evalf(abs(hin)) >= evalf(abs(Pni/ao)^(1/st)) then lami:=1:
>                                     hin:=evalf(lami*hi):
>                                     zin:=evalf(zi+hin):
> fi:
> Pnin:=subs(z=zin,POLINOM):
> while evalf(abs(Pnin)) > evalf(abs(Pni)) do lami:=evalf(lami/2):

```

```

> hin:=evalf(lami*hi):
> zin:=evalf(zi+hin):
> Pnin:=subs(z=zin,POLINOM):
> od:
> zi2:=zi1: zi1:=zi: zi:=zin:
> Pni2:=subs(z=zi2,POLINOM): вычисление значений полинома в
новых узлах интерполяции
> Pni1:=subs(z=zi1,POLINOM):
> Pni :=subs(z=zi ,POLINOM):
> k_iter:=k_iter+1:
> od:
> RETURN():
> end:
>
> save PARABOLA,(cat(katalog,'PARABOLA.m')):

```

6. Решение систем нелинейных уравнений

6.1. Метод Ньютона

Базовой задачей является задача решения систем нелинейных уравнений:

$$F: R^n \rightarrow R^n, \text{ найти } x_* \in R^n, \text{ такое, что } F(x_*) = 0, \quad (6.1.1)$$

где F непрерывно дифференцируема.

Метод Ньютона для задачи (6.1.1) получается с помощью нахождения корня аффинной аппроксимации для F , соответствующей текущему приближению x_c . Эта аппроксимация строится с использованием тех же приемов, что и в случае задачи с одной переменной. Итак,

$$F(x_c + p) = F(x_c) + \int_{x_c}^{x_c+p} J(z) dz. \quad (6.1.2)$$

Поскольку в целях получения аффинной аппроксимации для F относительно приращений p к x_c мы аппроксимируем интеграл в (6.1.2) линейным членом $J(x_c) \cdot p$, то

$$M_c(x_c + p) = F(x_c) + J(x_c) \cdot p \quad (6.1.3)$$

(M_c обозначает текущую модель). Затем s^N находится из условия $M_c(x_c + s^N) = 0$, что дает итерацию метода Ньютона решения (6.1.1). Таким образом, находим решение системы

$$J(x_c) s^N = -F(x_c), \quad x_+ = x_c + s^N. \quad (6.1.4)$$

В результате получаем точку, которая одновременно обращает в нуль аффинные модели n функций, являющихся компонентами F . Эти модели имеют вид

$$(M_c)_i(x_c + s^N) = f_i(x_c) + \nabla f_i(x_c)^T s^N, \quad i=1, \dots, n,$$

Так как мы не предполагаем, что X_+ окажется равным X_* , а будет лишь приближением к X_* , более удачным, чем X_c , то дадим на базе ньютоновского шага (6.1.4.) алгоритм, его применяя из начального приближения X_0 .

Алгоритм. Метод Ньютона для систем нелинейных уравнений.

Заданы: непрерывно дифференцируемая функция $F : R^n \rightarrow R^n$ и $x_0 \in R^n$; на очередной, k -й итерации решить

$$J(x_k)s_k = -F(x_k),$$

$$x_{k+1} = x_k + s_k.$$

Здесь проявляется основное преимущество метода Ньютона: если X_0 находится достаточно близко к решению X_* и якобиан $J(x_*)$ не вырожден, то, последовательность X_k сходится к x_* q -квадратично. Метод Ньютона будет также успешно работать на почти линейных задачах, поскольку он находит нуль (невырожденной) аффинной функции за одну итерацию. Отметим также, что если какие-либо компоненты-функции в F аффинны, то каждая итерация, вырабатываемая методом Ньютона, будет решением этих уравнений, так как используемые им аффинные модели будут для этих функций всегда точными.

С другой стороны, что касается сходимости из плохих начальных приближений, то, естественно, метод Ньютона будет вести себя не лучше, чем в случае задач с одной переменной.

Следовательно, особенности сходимости метода Ньютона указывают на то, как его использовать в многомерных алгоритмах. А именно, желательно применять его, по крайней мере на заключительных итерациях любого нелинейного алгоритма, с тем чтобы использовать его быструю локальную сходимость, но, для того чтобы метод сходился глобально, его необходимо модифицировать.

Имеются также две основные проблемы, касающиеся реализации итерации алгоритма. Во-первых, якобиан функции F может не быть задан аналитически. Это обычно встречается в

реальных приложениях, когда, например, F не задана аналитически. Таким образом, аппроксимация $J(x_k)$ конечными разностями или каким-либо другим способом является отдельной задачей. Во-вторых, $J(x_k)$ может быть вырожденной или плохо обусловленной, и тогда не представляется возможным получить достоверное решение S_k линейной системы $J(x_k)s_k = -F(x_k)$. Возможная модификация алгоритма заключается в том, чтобы внести некоторое возмущение в $J(x_k)$, достаточное, чтобы сделать ее хорошо обусловленной, и перейти к дальнейшим действиям на итерации. Однако такая модификация $J(x_k)$ не имеет убедительного обоснования в терминах исходной задачи (6.1.1), и поэтому не рекомендуется к использованию. Вместо этого при плохой обусловленности $J(x_k)$ предпочтительно переходить непосредственно к такому глобальному методу решения нелинейных уравнений, который соответствует возмущению линейной модели, вполне осмысленному с точки зрения исходной нелинейной задачи.

Рассмотренные выше достоинства и недостатки метода Ньютона служат ключом к развитию многомерных алгоритмов, поскольку они указывают на свойства, которые желательно сохранить, и на области, где могут потребоваться улучшение и модификация. Рассмотрим достоинства и недостатки метода Ньютона.

Достоинства:

- Q -квадратичная сходимость из хорошего начального приближения при условии, что $J(x_*)$ не вырождена;

- точное решение за одну итерацию для аффинной F (точное на каждой итерации для всех аффинных компонент F).

Недостатки:

- отсутствие глобальной сходимости для многих задач;
- необходимость в $J(x_k)$ на каждой итерации;

— необходимость в решении на каждой итерации системы линейных уравнений, которая может быть вырожденной или плохо обусловленной.

6.2. Локальная сходимость метода Ньютона

Докажем локальную Q -квадратичную сходимость метода Ньютона для систем нелинейных уравнений. Техника доказательства аналогична технике используемой в случае одной переменной. Она также служит прототипом большинства доказательств сходимости.

Для заданной векторной нормы $\|\cdot\|$ через $N(x, r)$ обозначим открытую окрестность радиуса r с центром в x

$$N(x, r) = \{x \in R^n : \|x - x\| < r\}.$$

Теорема 6.2.1. Пусть функция $F : R^n \rightarrow R^n$ непрерывно дифференцируема в открытом выпуклом множестве $D \subset R^n$. Предположим, что существуют $x_* \in R^n$ и $r, \beta > 0$, такие, что

$N(x_*, r) \subset D$, $F(x_*) = 0$, существует $J(x_*)^{-1}$, причем $\|J(x_*)^{-1}\| \leq \beta$ и $J \in Lip_y(N(x_*, r))$. Тогда существует $\varepsilon > 0$,

такое, что для всех $x_0 \in N(x_*, \varepsilon)$ последовательность $x_1, x_2, \dots$, порождаемая соотношением

$$x_{k+1} = x_k - J(x_k)^{-1} F(x_k), \quad k = 0, 1, \dots,$$

корректно определена, сходится к x_* и удовлетворяет неравенству

$$\|x_{k+1} - x_*\| \leq \beta \gamma \|x_k - x_*\|^2, \quad k = 0, 1, \dots \quad (6.2.1)$$

Доказательство. Выберем ε так, чтобы якобиан $J(x)$ был невырожденным для любых $x \in N(x_*, \varepsilon)$, покажем, что, поскольку локальная погрешность аффинной модели, используемой для

проведения всех итераций метода Ньютона, имеет порядок $O(\|x_k - x_*\|^2)$, сходимость q -квадратична. Пусть

$$\varepsilon = \min\left\{r, \frac{1}{2\beta\gamma}\right\}. \quad (6.2.2)$$

Докажем с помощью метода индукции по k , что на каждом шаге выполняется (6.2.1), а также, что

$$\|x_{k+1} - x_*\| \leq \frac{1}{2} \|x_k - x_*\|$$

и

$$x_{k+1} \in N(x_*, \varepsilon). \quad (6.2.3)$$

Сначала покажем, что якобиан $J(x_0)$ не вырожден. Из

$\|x_0 - x_*\| \leq \varepsilon$, непрерывности J по Липшицу в x_* и (6.2.2) следует, что

$$\|J(x_*)^{-1}[J(x_0) - J(x_*)]\| \leq \|J(x_*)^{-1}\| \|J(x_0) - J(x_*)\| \leq \beta \cdot \gamma \|x_0 - x_*\| \leq \beta \cdot \gamma \cdot \varepsilon \leq \frac{1}{2}$$

Таким образом, из соотношении для возмущенных матриц вытекает не вырожденность $J(x_0)$ и

$$\|J(x_0)^{-1}\| \leq \frac{\|J(x_*)^{-1}\|}{1 - \|J(x_*)^{-1}[J(x_0) - J(x_*)]\|} \leq 2\|J(x_*)^{-1}\| \leq 2 \bullet \beta. \quad (6.2.4)$$

Следовательно, x_1 корректно определено и

$$\begin{aligned} x_1 - x_* &= x_0 - x_* - J(x_0)^{-1} F(x_0) = x_0 - x_* - J(x_0)^{-1} [F(x_0) - F(x_*)] = \\ &= J(x_0)^{-1} [F(x_*) - F(x_0) - J(x_0)(x_* - x_0)] \end{aligned}$$

Заметим, что выражение в квадратных скобках представляет собой разность между $F(x_*)$ и значением аффинной модели $M_c(x)$, вычисленным в x_* . Следовательно

$$\|x_1 - x_*\| \leq \|J(x_0)^{-1}\| \|F(x_*) - F(x_0) - J(x_0)(x_* - x_0)\| \leq 2\beta \bullet \frac{\gamma}{2} \|x_0 - x_*\|^2 = \beta\gamma \|x_0 - x_*\|^2$$

Это доказывает (6.2.1). Поскольку $\|x_0 - x_*\| < \varepsilon \leq 1/(2\beta\gamma)$, получаем

$$\|x_1 - x_*\| \leq \frac{1}{2} \|x_0 - x_*\|,$$

что указывает на справедливость (6.2.3) и тем самым завершает доказательство для случая $k=0$. Идентично проводится доказательство индуктивного предположения.

Константы γ и β можно объединить в одну константу $\gamma_{\text{Rel}} = \gamma\beta$, которая по сути есть константа Липшица для измерения относительной нелинейности F в X_* , так как

$$\|J(x_*)^{-1}[J(x) - J(x_*)]\| \leq \|J(x_*)^{-1}\| \|J(x) - J(x_*)\| \leq \beta\gamma \|x - x_*\| = \gamma_{\text{Rel}} \|x - x_*\|$$

для $x \in N(X_*, r)$. Следовательно, теорему 6.2.1 можно также трактовать как утверждение о том, что радиус сходимости метода Ньютона обратно пропорционален относительной нелинейности F в X_* . Относительная нелинейность функции является основным фактором, определяющим поведение этой функции в алгоритме, и все теоремы сходимости могли бы быть легко переформулированы и доказаны в терминах этого понятия.

Сформулируем второй результат, касающийся сходимости метода Ньютона. Этот сильный результат был получен Канторовичем (1948). Также покажем, что относящиеся к нему предположения и методика его доказательства связаны с классическим результатом, касающимся сходимости итерационных алгоритмов и известным как теорема о сжимающем отображении.

Теорема Канторовича отличается от теоремы 6.2.1 главным образом тем, что она не предполагает существование решения $F(x_*) = 0$. Вместо этого она показывает, что, если якобиан $J(x_0)$ не вырожден, J непрерывен по Липшицу в некоторой области, содержащей x_0 , и первый шаг метода Ньютона достаточно мал относительно нелинейности F , то в этой области должен существовать корень и, более того, он должен быть единственным. В результате более слабых предположений доказывается лишь Γ -квадратичная скорость сходимости. Для большинства методов получение наилучших оценок скорости сходимости требует в каждом случае своего отдельного доказательства.

Теорема 6.2.2 (Канторовича). Пусть $r > 0$, $x_0 \in R^n$, $F: R^n \rightarrow R^n$, и предположим, что функция F непрерывно дифференцируема в $N(x_0, r)$. Пусть для заданных векторной и индуцированной операторной норм $J \in Lip_\gamma(N(x_0, r))$, причем $J(x_0)$ не вырожден, и существуют константы $\beta, \eta \geq 0$, такие, что

$$\|J(x_0)^{-1}\| \leq \beta, \quad \|J(x_0)^{-1}F(x_0)\| \leq \eta.$$

Доказательство.

Введем

обозначения

$$\gamma_{Rel} = \beta\gamma, \quad \alpha = \gamma_{Rel}\eta. \quad \text{Если} \quad \alpha \leq \frac{1}{2} \quad \text{и}$$

$r \geq r_0 \equiv (1 - \sqrt{1 - 2\alpha}/(\beta\gamma))$, то последовательность $\{x_k\}$, задаваемая соотношением

$$x_{k+1} = x_k - J(x_k)^{-1}F(x_k), \quad k = 0, 1, \dots,$$

корректно определена и сходится к x_* , который является единственным нулем F в замыкании множества $N(x_0, r_0)$. Если же в

этом случае $\alpha < \frac{1}{2}$, то x_* будет единственным нулем F в $N(x_0, r_1)$, где $r_1 \equiv \min[r, (1 + \sqrt{1 - 2\alpha}/(\beta\gamma))]$, и

$$\|x_k - x_*\| \leq (2\alpha)^{2k} \frac{\eta}{\alpha}, \quad k = 0, 1, \dots \quad (6.2.5)$$

Зафиксируем в n -мерном пространстве итерации $x_0, x_1, \dots$ и протянем нить, соединяющую их в указанном порядке. Пометим на нити те места, где находятся x_k , а затем вытянем ее в одну прямую, возможно, при этом немного растянув ее в длину. То, что в результате получится, представляет собой линейный сегмент, состоящий из последовательности верхних оценок для расстояний $\|x_{k+1} - x_k\|$, $k = 0, 1, \dots$. Доказательство теоремы Канторовича по существу показывает, что для любой последовательности $\{x_k\}$ эти расстояния меньше или равны соответственно $|t_{k+1} - t_k|$. Последние

получаются, если применить метод Ньютона, начиная с $t_0 = 0$, к функции

$$f : R \rightarrow R, \quad f(t) = \frac{\gamma}{2}t^2 - \frac{t}{\beta} + \frac{\eta}{\beta}$$

(видно, что она удовлетворяет условиям теоремы 6.2.2). Поскольку последовательность $\{t_k\}$ сходится монотонно к меньшему корню $t_* = (1 - \sqrt{1 - 2\alpha})/(\beta\gamma)$ функции $f(t)$, общая длина части X - нити, лежащей за X_k , не должна превосходить $|t_k - t_*|$. Таким образом, $\{X_k\}$ является последовательностью Коши и поэтому она сходится к некоторому X_* , причем погрешности $\|X_k - X_*\|$ ограничены Q -квадратично сходящейся последовательности $|t_k - t_*|$. Эта методика доказательства называется *мажоризацией*. Она нигде не использует конечность размерности пространства R^n . Говорят, что последовательность $\{t_k\}$ мажорирует последовательность $\{X_k\}$.

Благоприятным обстоятельством в теореме Канторовича является то, что она не предполагает существования X_* и невырожденности $J(X_*)$; в действительности существуют функции F и точки X_0 , которые удовлетворяют условиям теоремы 2.3.1, хотя $J(X_*)$ вырожден. Неудачным моментом здесь является то, что r - порядок сходимости ничего не говорит о продвижениях на каждой итерации, а даваемые теоремой оценки погрешности на практике часто слишком неточны. Но это не так важно, так как Дэннисом в (1971) показано, что

$$\frac{1}{2}\|x_{k+1} - x_k\| \leq \|x_k - x_*\| \leq 2\|x_{k+1} - x_k\|.$$

Кроме того, проверку предположений теоремы 6.2.2 традиционно представляют себе как выяснение вопроса о том, будет ли метод Ньютона сходиться для некоторого X_0 и конкретной функции F . Однако на практике фактически всегда для тщательной оценки константы Липшица γ требуется больше труда, чем для того, чтобы

просто попробовать метод Ньютона и посмотреть, сходится он или нет.

Теорема Канторовича по своему виду очень сходна с теоремой о сжимающем отображении. В ней рассматривается произвольный итерационный метод вида $x_{k+1} = G(x_k)$ и устанавливается условие относительно G , при которых последовательность $\{x_k\}$ будет сходиться q -линейно к точке x_* из любой точки x_0 в области D . Более того, доказывается, что x_* является единственной точкой в D , такой, что $G(x_*) = x_*$. Таким образом, теорема о сжимающем отображении представляет собой более общий, но вместе с тем и более слабый результат, чем теорема Канторовича.

Теорема 6.2.3. (теорема о сжимающем отображении).

Пусть $G : D \rightarrow D$, где D есть замкнутое подмножество R^n . Если для некоторой нормы $\|\cdot\|$ существует $\alpha \in [0,1)$, такое, что

$$\|G(x) - G(y)\| \leq \alpha \|x - y\|, \quad \forall x, y \in D, \quad (6.2.6)$$

то:

- существует единственное $x_* \in D$, такое, что $G(x_*) = x_*$;
- для любых $x_0 \in D$ последовательность $\{x_k\}$, удовлетворяющая соотношению $x_{k+1} = G(x_k)$, $k = 0, 1, \dots$, сходится q -линейно к x_* с константой α ;
- для любого $\eta \geq \|G(x_0) - x_0\|$,

$$\|x_k - x_*\| \leq \frac{\eta \alpha^k}{1 - \alpha}, \quad k = 0, 1, \dots$$

Итерационная функция G , удовлетворяющая (6.2.6) называется *сжимающей* в области D . Из этого свойства следует, что, начиная из любой начальной точки $x_0 \in D$, длина шага $\|x_{k+1} - x_k\|$ уменьшается на каждой итерации по крайней мере на сомножитель α , так как

$$\|x_{k+1} - x_k\| = \|G(x_k) - G(x_{k-1})\| \leq \alpha \|x_k - x_{k-1}\|. \quad (6.2.7)$$

На основе (6.2.7) легко показать, что $\{x_k\}$ мажорируется q - линейно сходящейся последовательностью. Отсюда как следствие вытекает теорема 6.2.3. Таким образом, доказательства теоремы Канторовича и теоремы о сжимающем отображении тесно связаны.

Теорему о сжимающем отображении можно применить к задаче (6.1.1), задав итерационную функцию $G(x)$, так чтобы $G(x) = x$ в том и только в том случае, когда $F(x) = 0$. Примером служит

$$G(x) = x - AF(x), \text{ где } A \in R^{n \times n} \text{ не вырождена.} \quad (6.2.8)$$

Тогда эту теорему можно использовать для выяснения существования какой-либо области D , такой, что точки, генерируемые итерационной функцией (6.2.8) из $x_0 \in D$, будут сходиться к корню функции F . Однако теорема о сжимающем отображении может быть использована для установления только линейной сходимости.

Итак, для системы:

$$\begin{cases} F(x, y) = 0; \\ G(x, y) = 0. \end{cases}$$

согласно методу Ньютона последовательные приближения вычисляются по формулам:

$$x_{n+1} = x_n - \frac{\Delta_x^{(n)}}{J(x_n, y_n)}, y_{n+1} = y_n - \frac{\Delta_y^{(n)}}{J(x_n, y_n)},$$

$$\Delta_x^{(n)} = \begin{vmatrix} F(x_n, y_n) & F'_y(x_n, y_n) \\ G(x_n, y_n) & G'_y(x_n, y_n) \end{vmatrix}, \Delta_y^{(n)} = \begin{vmatrix} F'_x(x_n, y_n) & F(x_n, y_n) \\ G'_x(x_n, y_n) & G(x_n, y_n) \end{vmatrix}.$$

где

$$J(x_n, y_n) = \begin{vmatrix} F'_x(x_n, y_n) & F'_y(x_n, y_n) \\ G'_x(x_n, y_n) & G'_y(x_n, y_n) \end{vmatrix} \neq 0.$$

Начальные приближения x_0, y_0 определяются приближенно (например, графически). Метод Ньютона эффективен при достаточной близости начального приближения к решению системы.

6.3. Методы с конечно-разностными производными для решения систем нелинейных уравнений

Рассмотрим влияние на метод Ньютона замены аналитически заданного якобиана $J(x)$ на конечно-разностную аппроксимацию. Легко показать, что, если длина конечно-разностного шага выбрана надлежащим образом, то сохраняется q - квадратичная сходимость метода Ньютона. Для большинства задач метод Ньютона с использованием аналитических производных и метод Ньютона с надлежащим выбором конечных разностей фактически неразличимы [4].

Теорема 6.3.1. Пусть F и x_* удовлетворяют условиям теоремы 6.2.3, где $\|\cdot\|$ обозначает векторную t_1 -норму и соответствующую индуцированную матричную норму. Тогда существуют ε , $h > 0$, такие, что если $\{h_k\}$ есть последовательность действительных чисел, причем $0 < |h_k| \leq h$ и $x_0 \in N(x_*, \varepsilon)$, то последовательность $x_1, x_2, \dots$ задаваемая формулами

$$(A_k)_j = \frac{F(x_k + h_k e_j) - F(x_k)}{h_k}, \quad j = 1, \dots, n, \quad (6.3.1)$$

$$x_{k+1} = x_k - A_k^{-1} F(x_k), \quad k = 0, 1, \dots,$$

корректно определена и сходится q -линейно к x_* . Если

$$\lim_{k \rightarrow 0} h_k = 0,$$

то сходимость q -сверхлинейна. Если существует некоторая константа C_1 , такая, что

$$|h_k| \leq C_1 \|x_k - x_*\|, \quad (6.3.2)$$

или, что эквивалентно, константа C_2 , такая, что

$$|h_k| \leq C_2 \|F(x_k)\|, \quad (6.3.3)$$

то сходимость q -квадратична.

Доказательство. Выберем ε и h , так чтобы матрица A_k была невырожденной для любых $x_k \in N(x_*, \varepsilon)$ и $|h_k| < h$. Пусть $\varepsilon \leq r$ и

$$\varepsilon + h \leq \frac{1}{2\beta\gamma}, \quad (6.3.4)$$

С помощью метода математической индукции по k , покажем что на каждом шаге

$$\|x_{k+1} - x_*\| \leq \frac{1}{2}\|x_k - x_*\|, \quad x_{k+1} \in N(x_*, r). \quad (6.3.5)$$

Для $k=0$ покажем прежде всего, что A_0 не вырождена. Из неравенства треугольника, непрерывности J по Липшицу в x_* ,

$$\begin{aligned} \|x_0 - x_*\| &\leq \varepsilon \text{ и (6.3.4) вытекает} \\ \|J(x_*)^{-1}[A_0 - J(x_*)]\| &\leq \|J(x_*)^{-1}\| \| [A_0 - J(x_0)] + [J(x_0) - J(x_*)] \| \leq \\ &\leq \beta\left(\frac{\gamma h}{2} + \gamma\varepsilon\right) \leq \frac{1}{2}. \end{aligned} \quad (6.3.6)$$

Из соотношения для возмущенных матриц и неравенства (6.3.6) следует, что A_0 не вырождена и

$$\|A_0^{-1}\| \leq 2\beta. \quad (6.3.7)$$

Поэтому x_1 корректно определено и

$$x_1 - x_* = A_0^{-1} \left([F(x_*) - F(x_0) - J(x_0)(x_* - x_0)] + [(J(x_0) - A_0)(x_* - x_0)] \right).$$

Заметим, что эта формула отличается от соответствующей ей в методе Ньютона только тем, что $J(x_0)^{-1}$ здесь заменено на A_0^{-1} и добавлено слагаемое $(J(x_0) - A_0)(x_* - x_0)$. Используя неравенства (6.3.4), (6.3.7) и $\|x_0 - x_*\| \leq \varepsilon$ получим

$$\begin{aligned} \|x_1 - x_*\| &\leq \|A_0^{-1}\| \|F(x_*) - F(x_0) - J(x_0)(x_* - x_0)\| + \|A_0 - J(x_0)\| \|x_* - x_0\| \leq \\ &\leq 2\beta \left[\frac{\gamma}{2} \|x_* - x_0\|^2 + \frac{\eta}{2} \|x_0 - x_*\| \right] \leq \beta\gamma(\varepsilon + h) \|x_0 - x_*\| \leq \frac{1}{2} \|x_0 - x_*\|. \end{aligned}$$

Это доказывает неравенство (6.3.5). Идентично проводится доказательство индуктивного предположения.

Для доказательства q - сверхлинейной и q - квадратичной сходимости необходимо получить улучшенную оценку величины $\|A_k - J(x_k)\|$.

Как и в случае одной переменной рассмотренная теорема не указывает как в точности следует выбирать длину конечно-разностного шага на практике. Это вызвано тем, что здесь имеются два взаимно противоречивых источника ошибок, возникающих при вычислении конечно-разностных производных. Рассмотрим конечно-разностное соотношение. При этом опустим ради удобства нижний индекс k , отвечающий номеру итерации. По лемме ошибка аппроксимации $(J(x))_j$ с помощью A_j равна $O(\eta)$ в точной арифметике, что подсказывает выбрать h настолько малым, насколько это возможно. Однако при малых значениях h существенно сказывается другие ошибки, связанные с вычислением в арифметике конечной точности числителя этой формулы. Поскольку числитель делится на h , ошибка δ в числителе приведет в (6.3.1) к ошибке δ/h . Относительно этой ошибки в числителе, которая проистекает из неточных значений функции и из потери точности при их вычитании, можно с достаточным основанием сказать, что она составляет лишь малую часть от $F(x)$. В действительности если h слишком мало, то $x + he_i$ может быть настолько близким к x , что $F(x + he_i) \cong F(x)$, и поэтому A_j может вовсе не иметь достоверных разрядов. Таким образом, h должно выбираться так, чтобы сбалансировать ошибки $O(\eta)$ и $O(F(x)/h)$.

Если $F(x)$ имеет t достоверных разрядов, то целесообразно стремиться к тому, чтобы $F(x + he_i)$ отличалось от $F(x)$ правой половиной этих разрядов. Точнее говоря, если относительная ошибка при вычислении $F(x)$ равна η (η представляет собой относительный уровень шума), то необходимо получить

$$\frac{\|F(x + he_i) - F(x)\|}{\|F(x)\|} \leq \sqrt{\eta}, \quad j = 1, \dots, n. \quad (6.3.8)$$

При отсутствии какой-либо дополнительной информации получить неравенство (6.3.8) возможно при использовании для возмущения каждой компоненты x_j своей *собственной* длины шага

$$h_j = \sqrt{\eta} x_j,$$

которая отвечает относительному изменению $\sqrt{\eta}$ на величину x_j .

Тогда A_j вычисляется по формуле

$$A_j = \frac{F(x + h_j e_j) - F(x)}{h_j}. \quad (6.3.9)$$

Для удобства мы предполагали, что используется одинаковая длина шага h для всей конечно-разностной аппроксимации, однако нет оснований так поступать на практике. Более того, единая длина шага может привести к пагубным последствиям, если компоненты вектора x сильно различаются по величине. Теорему легко переработать для случая выбора для каждой компоненты своей длины шага h_j .

В случае когда $F(x)$ задана простой формулой, разумно полагать $\eta \cong macheps$, и тогда длина шага h_j есть $\sqrt{macheps} \cdot x_j$.

Основной причиной ошибок при использовании программного обеспечения по решению каких-либо нелинейных задач является неверное программирование производных. Таким образом, если предоставлены аналитические производные, то рекомендуется, чтобы

они были проверены на соответствие с конечно-разностной аппроксимацией в начальной точке X_0 . Если $F(x)$ вычисляется очень длинной программой или итерационной процедурой, то η может оказаться гораздо больше, чем *macheps*. Когда η настолько велико, что можно ожидать неточность в конечно-разностных производных, тогда следует воспользоваться аппроксимацией $J(x)$ по секущей. Это относится и к случаю, когда недопустимы издержки, связанные с n дополнительными вычислениями $F(x)$ на каждой итерации для получения A .

Так как формула (6.3.9) может дать неприемлемо малое значение h_j , когда X_j подходит близко к нулю, в алгоритме она несколько видоизменена, так что

$$h_j = \sqrt{\eta} \max\{|x_j|, typx_j\} sign(x_j), \quad (6.3.10)$$

где $typx_j$ представляет собой характерный размер X_j , задаваемый пользователем.

В заключении покажем, как длина шага может быть отчасти обоснована анализом. Естественно предположить, что X_j есть некоторое точное машинное число. Для упрощения анализа предположим, что $x_j + h_j$ также есть некоторое точное машинное число, причем $x_j + h_j \neq x_j$. Это имеет место, когда h_j вычисляется по какому-либо правилу выбора длины шага, которое гарантирует, что $|h_j| < macheps |x_j|$, тогда, прежде чем использовать h_j , вычислим

$$temp = x_j + h_j,$$

$$h_j = temp - x_j.$$

Этот прием улучшает точность, поэтому используется на практике. Теперь легко убедиться, что погрешность, возникающая при

вычислении A_j из-за ошибок округления и погрешностей вычисления функции, ограничена величиной

$$\frac{2(\eta + \text{macheps})\hat{F}}{|h_j|}, \quad (6.3.11)$$

где $\hat{F}$ есть верхняя граница для $\|F(x)\|$ при $x \in [x, x + h_j e_j]$. Кроме того разницa в точной арифметике между A_j и $J(x)_j$ ограничена величиной

$$\frac{\gamma_j |h_j|}{2}, \quad (6.3.12)$$

где γ_j есть константа Липшица, такая, что

$$\|J(x + te_j) - J(x)\| \leq \gamma_j |t|$$

для всех $t \in [0, h_j]$. Значение h_j , минимизирующее сумму величин (6.3.11) и (6.3.12), равно

$$h_j^* = \left(\frac{4(\eta + \text{macheps})\hat{F}}{\gamma_j} \right)^{1/2}. \quad (6.3.13)$$

При достаточно разумных предположениях о том, что

$$\gamma_j = O \left(\frac{\hat{F}}{(\max\{|x_j|, \text{typ} x_j\})^2} \right)^{1/2},$$

(6.3.13) сводится к (6.3.10) с точностью до некоторого постоянного множителя. Заметим, что проведенный анализ можно модифицировать с учетом абсолютной погрешности в вычисленном значении $F(x)$.

6.4. Решение типового примера

Используя метод Ньютона, решить систему нелинейных уравнений с точностью до 0,002.

$$\begin{cases} \sin(2x - y) - 1.2x = 0.4, \\ 0.8x^2 + 1.5y^2 = 1; \end{cases}$$

Отделение корней производим графически. Для построения графиков функций (рис. 6), составим таблицу 1 значений функций, входящих в первое и второе уравнение.

Таблица 6 Значение функции

x	-1,1	-1	-0,8	-0,6	-0,4	-0,2	0,2	0,4	0,5
x^2	1,21	1	0,64	0,36	0,16	0,04	0,04	0,16	0,25
$0.8x^2$	0,97	0,80	0,51	0,29	0,13	0,03	0,03	0,13	0,20
$1-0.8x^2$	0,03	0,20	0,49	0,71	0,87	0,97	0,97	0,87	0,80
$\frac{1-0.8x^2}{1.5}$									
	0,02	0,13	0,33	0,47	0,58	0,65	0,65	0,58	0,53
y_2	0,15	0,37	0,57	0,69	0,76	0,80	0,80	0,76	0,73
$1.2x$	-1,32	-1,2	-0,96	-0,72	-0,48	-0,24	0,24	0,48	0,6
$0.4+1.2x$	-0,92	-0,8	-0,56	-0,32	-0,08	0,16	0,64	0,88	1
$2x-y$	-1,17	-0,93	-0,59	-0,33	-0,08	0,16	0,69	1,08	1,57
y_1	-1,03	-1,07	-1,01	-0,87	-0,72	-0,56	-0,29	-0,28	-0,57

Значения x можно брать исходя из следующих условий:

- из первого уравнения $-1 < 1.2x + 0.4 < 1$, $-1,16 < x < 0,5$;
- из второго уравнения $-\sqrt{1.25} < x < \sqrt{1.25}$, $-1,12 < x < 1,12$.

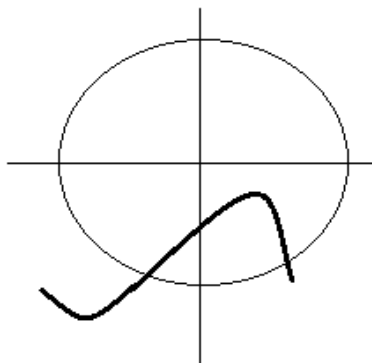


Рисунок 6 График функции

Таким образом, $-1,12 < x < 0,5$.

Система имеет два решения. Уточним одно из них, принадлежащее области D: $0,4 < x < 0,5$; $-0,76 < y < -0,73$.

За начальное приближение примем $x_0 = 0,4$; $y_0 = -0,75$.

Получим системы:

$$\begin{cases} F(x, y) = \sin(2x - y) - 1,2x - 0,4, \\ G(x, y) = 0,8x^2 + 1,5y^2 - 1; \end{cases}$$

$$\begin{cases} F'_x = 2\cos(2x - y) - 1,2, \\ G'_x = 1,6x; \end{cases}$$

$$\begin{cases} F'_y = -\cos(2x - y), \\ G'_y = 3y; \end{cases}$$

Найдем элементы матрицы Якоби $W_{i,j} = \frac{\partial f_i}{\partial x_j}$, где $i, j = \overline{1, n}$,

и значения функций в $x_0 = 0,4$; $y_0 = -0,75$:

$$W_{1,1} = \frac{\partial f}{\partial x} = 2\cos(2x - y) - 1,2 \approx -1,158$$

$$W_{1,2} = \frac{\partial f}{\partial y} = -\cos(2x - y) \approx -0,021$$

$$W_{2,1} = \frac{\partial g}{\partial x} = 1,6x = 0,64$$

$$W_{2,2} = \frac{\partial g}{\partial y} = 3y = -2,25$$

$$F(0,4;-0,75) = 0,11978$$

$$G(0,4;-0,75) = -0,02825$$

$$\begin{cases} -1,158 \cdot \Delta x - 0,021 \cdot \Delta y = 0,11978, \\ 0,64 \cdot \Delta x - 2,25 \cdot \Delta y = -0,02825; \end{cases}$$

$$\text{где } \Delta x = \frac{\det W_1}{\det W}; \quad \Delta y = \frac{\det W_2}{\det W};$$

$$\det W = \begin{vmatrix} F'_x(x_n, y_n) & F'_y(x_n, y_n) \\ G'_x(x_n, y_n) & G'_y(x_n, y_n) \end{vmatrix}$$

$$\det W_1 = \begin{vmatrix} F'_y(x_n, y_n) & F(x_n, y_n) \\ G'_y(x_n, y_n) & G(x_n, y_n) \end{vmatrix}$$

$$\det W_2 = \begin{vmatrix} F(x_n, y_n) & F'_x(x_n, y_n) \\ G(x_n, y_n) & G'_x(x_n, y_n) \end{vmatrix}$$

Итерационные формулы:

$$\begin{cases} x_{n+1} = x_n + \Delta x, \\ y_{n+1} = y_n + \Delta y; \end{cases}$$

Все вычисления производим в таблице 7

Таблица 7 Результаты вычислений

n	x_n	$0.8x_n^2$	$2x_n \cdot y_n$	$\sin(2x_n \cdot y_n)$	$F(x_n, y_n)$	$F'_x(x_n, y_n)$	$F'_y(x_n, y_n)$	detW	$\det W_1$	Δx
	y_n	$1.5y_n^2$		$\cos(2x_n \cdot y_n)$	$G(x_n, y_n)$	$G'_x(x_n, y_n)$	$G'_y(x_n, y_n)$		$\det W_2$	Δy
0	0,40	0,128	1,55	0,99	0,12	-1,16	-0,02	2,62	0,27	0,10
	-0,75	0,843		0,02	-0,02	0,64	-2,25		0,04	0,02
1	0,50	0,202	1,74	0,99	-0,02	-1,54	0,168	3,24	-0,040	-0,01
	-0,73	0,806 44		-0,17	0,01	0,81	-2,19		-0,00	-0,00
2	0,49	0,19	1,716 28	0,99	-0,00	-1,49	0,14	3,164 40	-0,00	-0,00
	-0,73	0,81		-0,14	0,00	0,79	-2,20		-0,00	-0,00
3	0,49									
	-0,73									

Так как $\delta < \varepsilon$, $\delta = \max(\Delta x, \Delta y)$ то можно прекратить
 производить вычисления. Таким образом, ответ:
$$\begin{cases} x \approx 0,49124, \\ y \approx -0,7335; \end{cases}$$

6.5. Лабораторная работа. Решение систем нелинейных уравнений методом Ньютона

Задание: решить систему уравнений с помощью метода Ньютона. Результаты получить с точностью $\varepsilon = 0,001$. Начальные приближения найти графически.

Варианты заданий:

1. $tg(xy + 0,1) = x^2$, $0,5x^2 + 2y^2 = 1$;
2. $tg(xy + 0,2) = x^2$, $0,6x^2 + y^2 = 1$;
3. $tg(xy + 0,4) = x^2$, $0,8x^2 + 2y^2 = 1$;

4. $tg(xy) = x^2, 0,5x^2 + 2y^2 = 1;$
5. $tg(xy + 0,2) = x^2, 0,7x^2 + 2y^2 = 1;$
6. $tg(xy + 0,1) = x^2, 0,6x^2 + 2y^2 = 1;$
7. $tg(xy + 0,3) = x^2, 0,8x^2 + 2y^2 = 1;$
8. $tg(xy + 0,4) = x^2, 0,6x^2 + 2y^2 = 1;$
9. $tg(xy + 0,1) = x^2, 0,8x^2 + 2y^2 = 1;$
10. $tg(xy) = x^2, 0,6x^2 + 2y^2 = 1;$
11. $tg(xy + 0,1) = x^2, 0,9x^2 + 2y^2 = 1;$
12. $tg(xy + 0,3) = x^2, 0,5x^2 + 2y^2 = 1;$
13. $tg(xy + 0,4) = x^2, 0,5x^2 + 2y^2 = 1;$
14. $tg(xy + 0,2) = x^2, 0,9x^2 + 2y^2 = 1;$
15. $tg(xy + 0,5) = x^2, 0,1x^2 + 2y^2 = 1;$

6.6. Листинг программы

Строим матрицу Якоби:

```
> restart;
> with(LinearAlgebra):
> f1:=0.1-x0^2+2*y0*z0-x0;
f1 := 0.1 - x0^2 + 2 y0 z0 - x0
> f2:=-0.2+y0^2-3*x0*z0-y0;
f2 := -0.2 + y0^2 - 3 x0 z0 - y0
> f3:=0.3-z0^2-2*x0*y0-z0;
f3 := 0.3 - z0^2 - 2 x0 y0 - z0
> f1x:=diff(f1,x0);
> f1y:=diff(f1,y0);
> f1z:=diff(f1,z0);
> f2x:=diff(f2,x0);
> f2y:=diff(f2,y0);
```

```

> f2z:=diff(f2,z0);
> f3x:=diff(f3,x0);
> f3y:=diff(f3,y0);
> f3z:=diff(f3,z0);
> A:=<<f1x|f1y|f1z>,<f2x|f2y|f2z>,<f3x|f3y|f3z>>;

```

$$A := \begin{bmatrix} -2 x0 - 1 & 2 z0 & 2 y0 \\ -3 z0 & 2 y0 - 1 & -3 x0 \\ -2 y0 & -2 x0 & -2 z0 - 1 \end{bmatrix}$$

Выбираем начальное приближение, близкое к уже известному нам корню и строим последовательность:

```

> x0:=0;y0:=0;z0:=0;

```

```

x0 := 0

```

```

y0 := 0

```

```

z0 := 0

```

```

> A:=A;

```

$$A := \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

```

> A1:=A^(-1);

```

$$A1 := \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

```

> f:=<f1,f2,f3>;

```

$$f := \begin{bmatrix} 0.1 \\ -0.2 \\ 0.3 \end{bmatrix}$$

```

> X0:=<x0,y0,z0>;

```

```

> X:=Add(X0,(Multiply(A1,f),1,-1);

```

$$X := \begin{bmatrix} 0.10000000000000000004 \\ -0.20000000000000000010 \\ 0.29999999999999999988 \end{bmatrix}$$

```

> X0:=X;

```

```

> x0:=X[1];y0:=X[2];z0:=X[3];
> A:=<<f1x|f1y|f1z>,<f2x|f2y|f2z>,<f3x|f3y|f3z>>;
A := 
$$\begin{bmatrix} -1.2000000000 & 0.6000000000 & -0.4000000000 \\ -0.9000000000 & -1.4000000000 & -0.3000000000 \\ 0.4000000000 & -0.2000000000 & -1.6000000000 \end{bmatrix}$$

> A1:=A^(-1);
> f:=<f1,f2,f3>;
f := 
$$\begin{bmatrix} -0.1300000000 \\ -0.0500000000 \\ -0.0500000000 \end{bmatrix}$$

> X:=Add(X0,(Multiply(A1,f)),1,-1);
X := 
$$\begin{bmatrix} 0.0224532224532224546 \\ -0.174324324324324320 \\ 0.246153846153846140 \end{bmatrix}$$

> i:=2;
> while (Norm(f))>0.0001 do
X0:=X;
x0:=X[1];y0:=X[2];z0:=X[3];
A:=<<f1x|f1y|f1z>,<f2x|f2y|f2z>,<f3x|f3y|f3z>>;
A1:=A^(-1);
f:=<f1,f2,f3>;
X:=Add(X0,(Multiply(A1,f)),1,-1);
i:=i+1;
end do;
> X:=X;

```

Получаем ответ:

$$X := \begin{bmatrix} 0.0128241509376391898 \\ -0.177800663726073254 \\ 0.244688047122264718 \end{bmatrix}.$$

Заключение

Существует множество методов *решения нелинейных алгебраических уравнений* и систем нелинейных уравнений, которые принято разделять на 3 группы: аналитические, графические и численные. Аналитические методы позволяют определить точные значения решения уравнений, хотя область их применения очень ограничена. Графические методы менее точны, но позволяют в сложных уравнениях определить наиболее приближенные значения, с которых в дальнейшем можно начинать находить более точные решения уравнений. Численное решение нелинейных уравнений предполагает прохождения двух этапов: отделение корня и его уточнение до определенно заданной точности. Отделение корней осуществляется различными способами: графически или при помощи различных специализированных компьютерных программ.

В методическом пособии рассмотрены различные численные методы решения нелинейных алгебраических уравнений. Классические методы (метод деления отрезка пополам, метод касательных, метод хорд, метод простой итерации) позволяют находить изолированные корни уравнений вида $f(x) = 0$, когда его корни действительные. Надо отметить, что при современном развитии компьютерной техники наиболее вычислительно эффективным методом нахождения корней нелинейных алгебраических уравнений становится метод деления отрезка пополам. Он очень просто программируется и не требует предварительных исследований. Метод парабол позволяет находить комплексные корни. Метод Ньютона является итерационным и позволяет находить корни систем уравнений.

Каждый раздел пособия включает примеры решения задач, задания для лабораторных работ и листинг программы, что поможет лучшему усвоению теоретического материала и написанию программ для численного решения уравнений.

Список литературы

1. Бережная Е. В. Математические методы моделирования экономических систем: учеб. пособие для вузов / Е. В. Бережная. - М. : Финансы и статистика, 2006. - 431 с.
2. Вержбицкий В. М. Численные методы: Математический анализ и обыкновенные дифференциальные уравнения :учеб. пособие для вузов -/ В. М. Вержбицкий. - М.: ОНИКС 21 век, 2005. - 399 с.
3. Волков Е.А. Численные методы [Электронный ресурс]: учеб. пособие / Е.А. Волков.– СПб.: Издательство «Лань», 2008. - 256 с. – Режим доступа: <http://e.lanbook.com/view/book/54/> - Численные методы
4. Демидович Б.П. Численные методы анализа. Приближение функций, дифференциальные и интегральные уравнения [Электронный ресурс]: учеб. пособие / Б.П. Демидович, И.А. Марон., Э.З. Шувалова.– СПб.: Издательство Лань, 2010. – 400 с. – Режим доступа: <http://e.lanbook.com/view/book/537/> - Численные методы анализа. Приближение функций, дифференциальные и интегральные уравнения.
5. Карпухин Э.В. К вопросу применения эффективных численных методов для моделирования магнитострикционных приборов уровня [Электронный ресурс] / Э.В. Карпухин, Е.С. Демин, С.Б. Демин // Вестник ТГТУ. 2012. Т. 18. № 1. С. 58-64. – Режим доступа: <http://e.lanbook.com/view/journal/93516/> - К вопросу применения эффективных численных методов для моделирования магнитострикционных приборов уровня.
6. Охорзин В.А. Прикладная математика в системе MATHCAD [Электронный ресурс]: учеб. пособие / В. А. Охорзин - СПб. [и др.]: Издательство Лань, 2009. - 348 с. – Режим доступа: <http://e.lanbook.com/view/book/294/> - Прикладная математика в системе MATHCAD.

7. Торшина О.А. Практикум по курсу «Уравнения математической физики» / О.А. Торшина. - Магнитогорск: МаГУ, 2012. - 50 с.
8. Торшина О.А. Регуляризованные следы дифференциальных операторов./ О.А. Торшина.- Магнитогорск, 2015. 122 с.
9. Торшина О.А. Численный метод вычисления поправок теории возмущений / О.А. Торшина // Альманах современной науки и образования. Научно-теоретический/ тематический журнал. Тамбов: Грамота, № 12, 2013. С. 168-170.

Учебное текстовое электронное издание

**Кадченко Сергей Иванович
Торшина Ольга Анатольевна**

**ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ НЕЛИНЕЙНЫХ
АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ**

Учебное пособие

1,42 Мб

1 электрон. опт. диск

г. Магнитогорск, 2016 год
ФГБОУ ВО «МГТУ им. Г.И. Носова»
Адрес: 455000, Россия, Челябинская область, г. Магнитогорск,
пр. Ленина 38

ФГБОУ ВО «Магнитогорский государственный
технический университет им. Г.И. Носова»
Кафедра прикладной математики и информатики
Центр электронных образовательных ресурсов и
дистанционных образовательных технологий
e-mail: ceor_dot@mail.ru