Министерство общего и профессионального образования
Российской Федерации

МАТИ – Российский Государственный технологический
Университет им. К. Э. Циолковского

Кафедра “ Высшая математика”

Методы решения нелинейного уравнения

Методические указания к лабораторной работе
по курсу “Информатика”

Составители **Сидоров Б. Н.**
Никулин А. М.

Москва 1999

Данное руководство предназначено для студентов, изучающих предмет «Численные методы» и выполняющих лабораторные работы по курсу «Информатика». В методических указаниях рассмотрены ряд методов нахождения корней нелинейного уравнения и приведены примеры решения задач на языке программирования и в среде Mathcad.

ПОСТАНОВКА ЗАДАЧИ. ОТДЕЛЕНИЕ КОРНЕЙ.

Множество значений переменной x, при которых уравнение F(x)=0 является тождеством, называется решением уравнения. При этом каждое значение x из этого множества называется корнем этого уравнения. Нахождение точных значений корней возможно, как правило, только в исключительных случаях. Поэтому большое значение имеют методы приближенного решение уравнения с заданной точностью. При этом решение задачи можно разбить на два этапа:
!) отделение корней т.е выделение промежутков внутри которых содержится только один корень уравнения;
2) вычисление корня, принадлежащего выделенному промежутку с заданной точностью.
Решение задачи отделения корней для непрерывной функции основано на том, что, если функция на концах отрезка [a,b] имеет значения разных знаков, то внутри этого отрезка функция проходит через нуль, т. е содержится корень уравнения. Таким образом, чтобы произвести отделение корней необходимо разбить область предполагаемого нахождения корней на равные отрезки длиной h и вычислить значение функции на концах отрезка. Если будет выполнятся условие F(x)*F(x+h)<=0, то корень внутри отрезка [x,x+h]. Величина шага разбиения подбирается интуитивно; при большом шаге разбиения возможно пропустить корень; при маленьком – увеличивается время вычислений.
Ниже приведена программа на языке Паскаль, решающая задачу отделения корней для произвольной функции. (В программе рассматривается уравнение: $x^3 +2*x^2 – 6*x+1=0$).

```
uses crt;
const m=100; {отрезок [-m,m]}
      h=0.1;  {шаг разбиения }
var x:real;
function f(x:real):real;
{функция, задающая решаемое уравнение }
begin
  f:=x*x*x+2*x*x-6*x+1;
end;
{-----------------------------------------------------------------}
begin { main }
  clrscr;
  x:=-m;
  while x<m do
    begin
      if f(x)*f(x+h)<=0 then
         begin
           { здесь следует разместить вызов функции уточняющий корень
              на отрезке [x,x+h]   }
            writeln('корень находится внутри отрезка [',x:5:1,x+h:5:1,']');
         end;
       x:=x+h;
    end;
   repeat until keypressed;
 end.
```

МЕТОД ПОЛОВИННОГО ДЕЛЕНИЯ.

Рассмотрим методы нахождения корня уравнения на заданном отрезке. Все методы предполагают, что предварительно произведено отделение корней и на отрезке находится только один корень.

Метод половинного деления состоит в том, что мы уменьшаем длину отрезка так, что корень остается внутри отрезка; процесс продолжается до тех пор, пока длина отрезка не станет меньше заданной точности. Уменьшение длины отрезка производится самым естественным образом: делением отрезка пополам и выбором той половины, внутри которой находится корень(т.е на концах которой функция имеет разный знак.)

Алгоритм:

```
WHILE | a-b| > epsi
     Вычисляем середину отрезка x=(a+b)/2;
     Выбираем половину, где корень:
          Если F(a)*F(x)<=0  то корень в [a,x], поэтому b переносим в x b=x
          Если F(b)*F(x)<=0  то корень в [b,x], поэтому a переносим в x a=x
END
```

МЕТОД НЬЮТОНА (КАСАТЕЛЬНЫХ).

Суть метода:

За начальное приближение принимается какая-либо точка заданного отрезка, для которой
$F(a)\cdot F''(a)>0$;

Из этой точки проводится касательная к графику функции F(x). Уравнение касательной
$y=F'(a)\cdot(x-a)+F(a)$

Точка пересечения касательной с осью 0X (y=0) задает следующее приближение

$$x_1 : y=0 \Rightarrow F'(a)\cdot(x_1-a)+F(a)=0 \Rightarrow x_1=a-{F(a)}/{F'(a)}$$

Процесс продолжается до тех пор пока расстояние между двумя последовательными приближениями не станет меньше заданной точности
$|x_{i-1}-x_i|<epsi$

Условия применимости метода:

В интервале есть корень

В интервале существуют $F'(x)\, и\ F''(x)$

За начальное приближение принимается точка, в которой $F(a)\cdot F''(a)>0$

Алгоритм

```
X0 – начальное приближение
X1 – последующее приближение
F(x) – функция, задающая уравнение
Fp(x) – производная
X1=a x2=0 { необходимо для первоначальной проверки условия цикла}
WHILE |X1-X0|>epsi
   X0=X1
   X1=X0-F(X0)/Fp(X0)
END.
```

РЕШЕНИЕ ЗАДАЧИ В MATHCAD.

Для решения уравнения необходимо прежде всего построить график функции, задающий уравнение и по графику визуально определить начальное приближение корня.
Уточнение корня производится стандартной функцией
root(F(x),x),
где F(x) – функция, задающий уравнение.
x – имя переменной, варьируя которую ищется корень; перед использованием переменной обязательно должно быть присвоено начальное значение; для уравнения с несколькими корнями, ищется корень, наиболее близкий к начальному значению.
Пример решения уравнения $x^3 + 2*x^2 - 6*x+1=0$

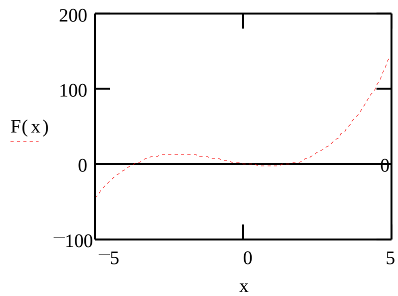
Решение кубического уравнения

$a := 1$ $b := 2$ $c := -6$ $d := 1$

$F(x) := a \cdot x^3 + b \cdot x^2 + c \cdot x + d$

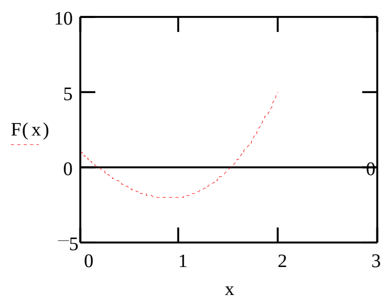
График

$x := -5, -4.9 .. 5$



Уточнение графика на интервале 0 ..2

$x := 0, 0.01 .. 2$



С помощью функции root (требуется начальное приближение)

$x := -3$ $x2 := root(F(x), x)$ $x2 = -3.696$

$x := 4$ $x1 := root(F(x), x)$ $x1 = 1.518$

$x := 0$ $x3 := root(F(x), x)$ $x3 = 0.178$

Примеры написание программ

Метод половинного деления

```
Kor1(F,a,b,tol) :=  | break  if  F(a)·F(b)>0
                    | c ← a
                    | d ← b
                    | while |c − d| >tol
                    |     | x ← (c + d)/2
                    |     | c ← x  if  F(c)·F(x)>0
                    |     | d ← x  otherwise
                    | x
```

Kor1(F, - 7, 0, 0.001) = −3.696

Kor1(F, 7, 1, 0.001) = 1.518

Kor1(F, 0, 1, 0.001) = 0.179

Метод Ньютона

```
Kor2(F,a,b,tol) :=  | x1 ← a  if  F(a)·(d²/da²) F(a)
                    | x1 ← b  otherwise
                    | x0 ← 0
                    | while |x1 − x0| >tol
                    |     | x0 ← x1
                    |     | x1 ← x0 − F(x0) / ((d/dx0) F(x0))
                    | x1
```

Kor2(F, - 7, 0, 0.001) = −3.696

Kor2(F, 1, 0, 0.001) = 1.518

Kor2(F, - 1, 0, 0.001) = 0.178

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ.

Требуется решить уравнение в Mathcade, написать и отладить программу нахождения всех корней уравнения, предложенным преподавателем методом.

1. $2*x^3+12*x^2+13*x+15=0$
2. $2*x^3-3*x^2+4*x+9=0$
3. $x^3-4*x^2-4*x-5=0$
4. $2*x^4+3*x^3+8*x^2+6*x+5=0$
5. $2*x^4-3*x^3+2*x^2-15*x+14=0$
6. $15*x^4-4*x^3-6*x^2-4*x-1=0$
7. $2*x^4-x^3+3*x^2-x+1=0$
8. $x^4+3*x^3-44*x^2+15*x+25=0$
9. $6*x^4+25*x^3+12*x^2-25*x+6=0$
10. $x^4-2*x^3-11*x^2+12*x+36=0$
11. $100*x^3+45*x^2-12*x+2.5=0$
12. $10*x^3+20*x^2-0.1*x-0.2=0$
13. $x^3+4.05*x^2-0.03*x+0.02=0$
14. $x^3+79.9*x^2-1988*x-200=0$
15. $x^3-4.6*x^2-52*x-20=0$
16. $x^3-0.5*x^2-0.5*x=0$
17. $200*x^3+78*x^2-41.2*x+0.42=0$
18. $2*x^3-0.6*x^2+0.06*x-0.002=0$
19. $0.5*x^3-2.3*x^2-26*x-10=0$
20. $-0.1*x^3+0.405*x^2+0.003*x-0.002=0$

ЛИТЕРАТУРА.

1. Mathcad 6.0 plus. Руководство пользователя. М., Филинъ.1998.
2. Ракитин В. И. , Первушин В.Е. Практическое руководство по методам вычислений. М., Высшая школа 1998.
3. Попов В. Б. Turbo Pascal 7.0 для школьников
М.; Финансы и статистика, 1996. -464