Pascal 13. Решение нелинейных уравнений.

Нелинейные уравнения можно разделить на 2 класса - алгебраические и трансцендентные. Алгебраическими уравнениями называют уравнения, содержащие только алгебраические функции (целые, рациональные, иррациональные). В частности, многочлен является целой алгебраической функцией. Уравнения, содержащие другие функции (тригонометрические, показательные, логарифмические и др.) называются трансцендентными.

Методы решения нелинейных уравнений делятся на две группы:

- точные методы;
- итерационные методы (за счет последовательных приближений получить решение уравнения с необходимой точностью).

Точные методы решения уравнений основываются на поиске равносильных преобразований алгебраических выражений, например, перенос слагаемых из одной части уравнения в другую с противоположным знаком, деление обеих частей уравнения на одинаковое число не равное 0, а также точные способы решений позволяют записать корни уравнения в виде некоторого конечного соотношения (формулы). Точные решения существуют только для некоторых уравнений определенного вида (линейные, квадратные, тригонометрические и др.), поэтому для большинства уравнений приходится использовать методы приближенного решения с заданной точностью (графические или численные). В первую очередь это относится к большинству трансцендентных уравнений. Доказано также, что нельзя построить формулу, по которой можно было бы решить произвольное алгебраическое уравнение выше четвертой степени.

$$ax + b = 0$$
$$ax^2 + bx + c = 0$$

$\downarrow$

Точные методы решения

$$x = \frac{-b}{a}$$
$$D = b^2 - 4ac$$
$$x_{1,2} = \frac{-b \pm \sqrt{D}}{2a}$$

$$x^3 + \cos x = 0$$
$$x^3 - \sin x = 0$$

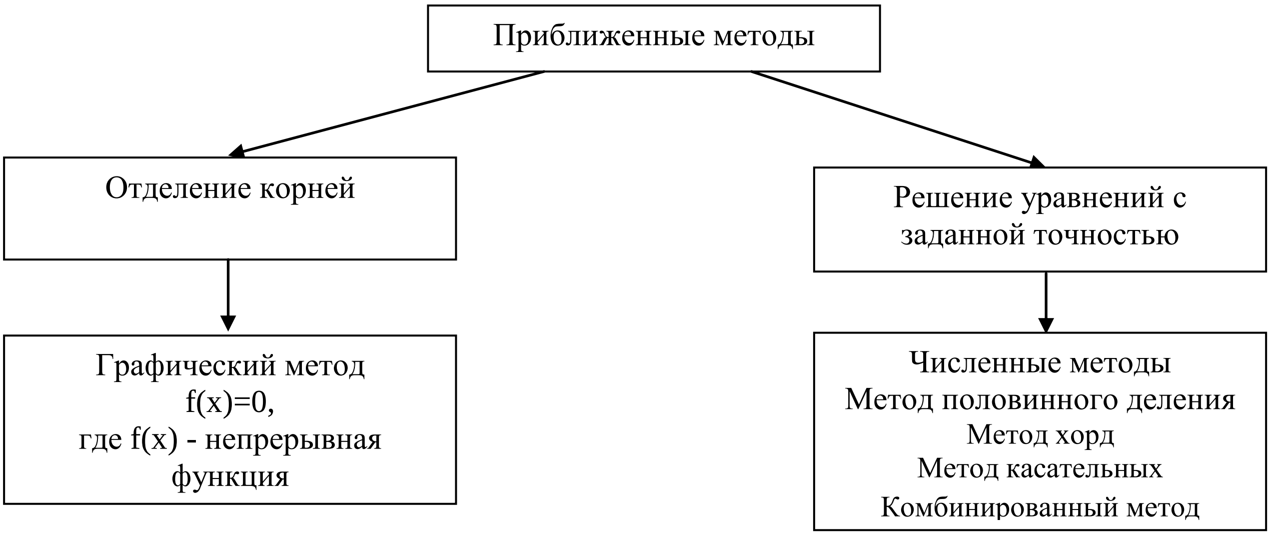
$\downarrow$

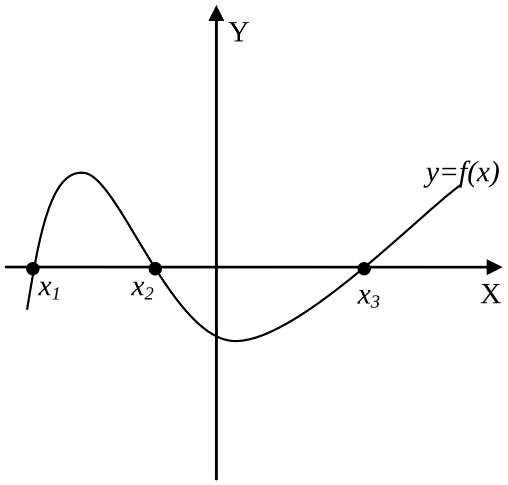
Приближенные методы решения

Например, уравнение $x^3 + cos\ x = 0$ нельзя решить путем равносильных алгебраических преобразований. Но это уравнение можно решать приближенно графическими и численными методами.

Решение уравнения проводят численно в два этапа. На первом этапе производится отделение корней - поиск интервалов, на которых содержится только по одному корню. Второй этап решения связан с уточнением корня на выбранном интервале (определением значения корня с заданной точностью). Далее будут рассмотрены несколько численных методов и приведены алгоритмы нахождения корней уравнений.

Отделение корней уравнения может проводиться графически, т.е. путем построения графика функции *y=f(x)*. Для уравнения вида *f (x) = 0*, где *f(x)* – некоторая непрерывная функция, корень (или корни) этого уравнения являются точкой (или точками) пересечения графика функции с осью абсцисс.





Отделение корней уравнения можно осуществить путем построения компьютерных моделей:

- построение графика функции с помощью одного из языков программирования (в данном случае Free Pascal);
- построение графика функции в электронных таблицах Microsoft Excel путем построения диаграммы типа *График*.

При построении графика функции корни уравнения можно получить лишь с небольшой степенью точности. Поэтому, чтобы эти значения получить с любой заданной степенью точности, необходимо применять методы, которые позволяют «уточнять» найденные значения.

Рассмотрим методы уточнения корней и их основные идеи. Отметим следующий момент: при прочих равных условиях, тот метод уточнения корней будет более эффективен, в котором результат с той же погрешностью найден за меньшее число раз вычисления функции *f(x).*

1.1. Метод половинного деления

Самый простой из них – метод половинного деления, или иначе метод дихотомии. Метод дихотомии получил свое название от древнегреческого слова διχοτομία, что в переводе означает деление надвое. Его мы используем довольно часто. Допустим, играя в игру "Угадай число", где один игрок загадывает число от 1 до 100, а другой пытается его отгадать, руководствуясь подсказками "больше" или "меньше". Логично предположить, что первым числом будет названо 50, а вторым, в случае если оно меньше - 25, если больше - 75. Таким образом, на каждом этапе неопределенность неизвестного уменьшается в 2 раза. Т.е. даже самый невезучий в мире человек отгадает загаданное число в данном диапазоне за 7 предположений вместо 100 случайных утверждений.

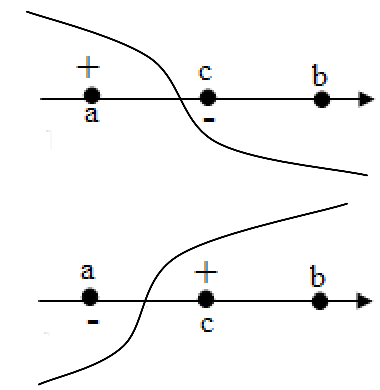
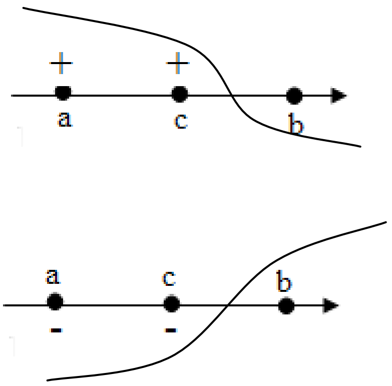
Алгоритм метода половинного деления основан на теореме Больцано - Коши о промежуточных значениях непрерывной функции и следствии из неё.

Теорема Больцано - Коши: если непрерывная функция принимает два значения, то она принимает любое значение между ними.

Следствие (теорема о нуле непрерывной функции): если непрерывная функция принимает на концах отрезка положительное и отрицательное значения, то существует точка, в которой она равна 0.

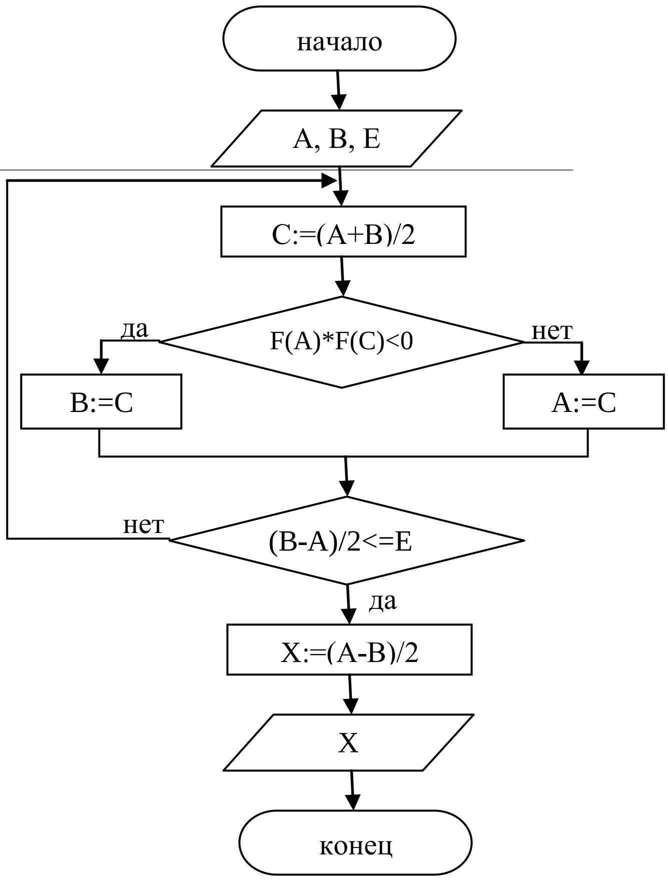
Алгоритм:

1. Задать отрезок [a,b] и погрешность ε.
2. Вычислить $c=(a+b)/2$
3. Определить интервал дальнейшего поиска: если $f(a)$ и $f(c)$ имеют разные знаки, т.е. $f(a)*f(c)<0$, то $b:=c$, в противном случае $a:=c$.

*f(a)*f(c)<0 (да)*	*f(a)*f(c)<0 (нет)*
	

4. Если длина нового отрезка $|b-a| \le \varepsilon$, то вычислить значение корня c=(a+b)/2 и остановиться, в противном случае перейти к шагу 2.

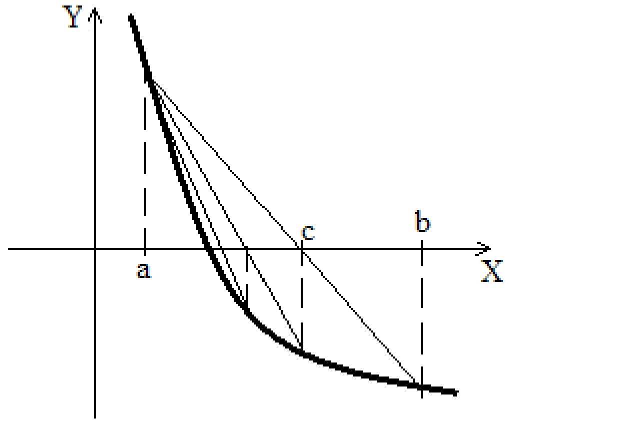
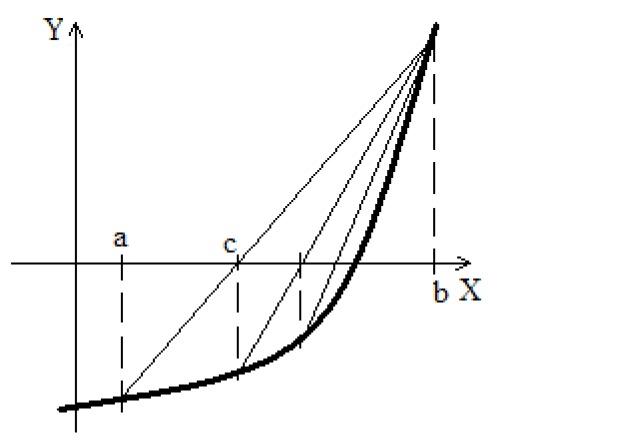
Блок-схема метода половинного деления



1.2. Метод хорд

При решении уравнения методом хорд нелинейная функция *f(x)* на отделенном интервале [a, b] заменяется линейной, в качестве которой берется хорда – прямая, стягивающая концы нелинейной функции. Вычисляются значения функции на концах отрезка, и строится "хорда", соединяющая точки *(a, f(a))* и *(b, f(b))*.

При решении нелинейного уравнения методом хорд задаются интервал [a,b] на котором существует только одно решение, и точность ε. Затем через две точки с координатами *(a, f(a))* и *(b, f(b))* проводим отрезок прямой линии (хорду) и определяем точку пересечения этой линии с осью абсцисс, точка *c*. Если при этом $f(a)\cdot f(c)<0$, то правую границу интервала переносим в точку *c* *(b=c)*. Если указанное условие не выполняется, то в точку *c* переносится левая граница интервала *(a=c)*. Поиск решения прекращается при достижении заданной точности $|f(c)| < \varepsilon$.

$f(a)\cdot f(c)<0$ (*да*)	$f(a)\cdot f(c)<0$ (*нет*)
	

Запишем уравнение прямой, проходящей через точки с координатами *(a, f(a))* и *(b, f(b))*:

$$\frac{y - f(a)}{f(b) - f(a)} = \frac{x-a}{b-a};$$

$$y = \frac{f(b) - f(a)}{b-a}(x-a) + f(a).$$

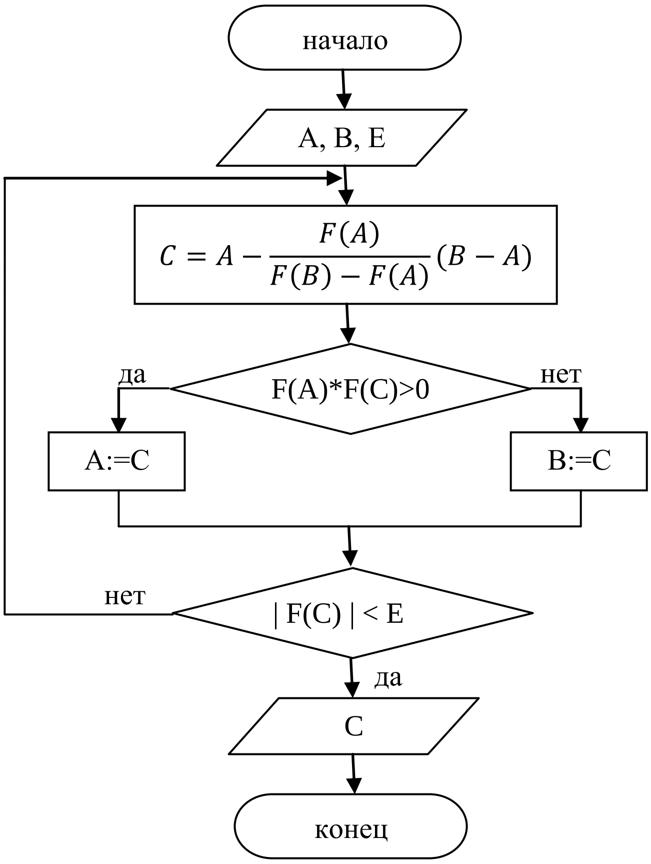
Прямая заданная полученным уравнением пересекает ось абсцисс при условии *y=0*. Найдем точку пересечения хорды с осью *X*.

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

Итак, $c = a - \dfrac{f(a)}{f(b) - f(a)}(b-a).$

Далее необходимо вычислить значение функции в точке *c*. Если | f(c) | < 0, то полученное число и есть корень уравнения с выбранной точностью, иначе необходимо построить следующую хорду и выполнить все рассмотренные ранее действия.

Блок-схема метода хорд



Пример 1. Решить уравнение методом дихотомии (найти один корень).

•Program6.pas*

```
function f(x: real): real;
begin
f:=x*x-4*x+4;
end;

var
 a, b, e, c, x: real;
begin
  a:=-1;   {границы заданы в программе}
  b:=2;
write ('e=');
read(e);
  c:=(a+b)/2;
while abs(b-a)>e do  begin
   if f(a)*f(c)<0 then
      b:=c
   else
      a:=c;
   c:=(a+b)/2;
end;
x:=(a+b)/2;
writeln ('x=',x:3:3,' f(x)=',f(x):4:4);
end.
```

Окно вывода

```
e=0.001
x=2.000 f(x)=0.0000
```