

## **Определение корня уравнения методами: табуляции, деления отрезка пополам.**

Метод деления отрезка пополам – один из наиболее распространенных простейших итерационных методов нахождения корней нелинейных уравнений. Во всех итерационных методах корни уравнения находятся приближенно при помощи многократных повторений вычислительной процедуры, пока не будет достигнута требуемая точность. Если после каждого повторения (итерации) погрешность в вычислении корня уменьшается – то итерационная процедура сходится (мы все более и более приближаемся к корню), иначе – процесс расходится.

Для использования метода деления отрезка пополам необходимо предварительно проанализировать поведение функции, например, построить таблицу значений функции (протабулировать), выделить окрестности корней и выбрать тот корень, который мы хотим найти первым – т.е. задать отрезок  $[a, b]$ , в котором он расположен. На концах отрезка окрестности корня функция будет иметь разные знаки. Делим начальный отрезок пополам – на отрезки  $[a, x_0]$  и  $[x_0, b]$ , где  $x_0 = (a+b)/2$ . Определяем, в каком из этих отрезков находится корень. Обозначаем через  $a$  и  $b$  границы этого отрезка. Для отрезка с корнем значения функции  $f(x)$  в граничных точках будут иметь разные знаки. В качестве критерия того, что функция имеет разные знаки, можно использовать знак выражения  $f(a)*f(b)$ . Если  $f(a)*f(b)$  будет меньше нуля, то функция в точках  $a$  и  $b$  имеет разные знаки. Второй отрезок, на котором знак  $f(x)$  не меняется, отбрасываем. В качестве первого приближения корня принимаем точку  $c_0$ . Проверяем значение функции  $f(x_0)$ , если оно по модулю меньше заданной погрешности –  $\epsilon$ , т.е.  $|f(x_0)| < \epsilon$ , то считаем, что корень найден. Если корень с допустимой погрешностью еще не найден, то переходим ко второй итерации – снова делим этот отрезок пополам, находим новую точку  $x_1 = (a+b)/2$  и повторяем итерационную процедуру.

В качестве критерия сходимости можно добавить анализ текущей ширины отрезка  $[a, b]$ , если она становится меньше допустимой погрешности  $e_x$ :  $[a,b] < e_x$  и модуль значения функции меньше заданной погрешности  $e$ :  $|f(x_0)| < e$ , то процесс прекращается.

Для того, чтобы в случае расходимости метода вычисления не длились бесконечно долго максимальное количество итераций ограничивают, как правило, для определения корня достаточно нескольких десятков итераций.

Алгоритм нахождения корня уравнения методом деления отрезка пополам на естественном языке:

1. Задаем начальные параметры алгоритма: точность нахождения корня –  $e$  и максимальное количество итераций –  $max_i$ .
2. Запрашиваем у пользователя границы отрезка  $a$  и  $b$ , табулируем функцию на заданном отрезке.
3. Задаем главный цикл от 1 до максимального количества итераций.
4. Проверяем, если на заданном отрезке функция меняет знак, то продолжаем вычисления, в обратном случае – отрезок выбран неудачно. Выход из программы.
5. Находим середину отрезка  $[a,b]$ , вычисляем значение функции в точке  $x_0$ . В случае удовлетворения точности нахождения корня и критерию сходимости завершаем итерации - работа программы закончена, в обратном случае – 6.
6. Переопределяем отрезок  $[a,b]$  таким образом, чтобы на нем выполнялось условие смены знака функции. Переходим к Шагу 5.

Алгоритм решения на языке VFP:

```
clear
e=0.001
maxi=1000
input "Введите границу отрезка a: " to a
input "Введите границу отрезка b: " to b
DO Tabulation WITH a,b,0.5

FOR i=1 TO maxi
fa=f(a)
```

```

fb=f(b)

IF fa*fb<0 then
x=(a+b)/2
fx=f(x)
?"  Итерация= "+ALLTRIM(STR(i))
??" : x= "+TRANSFORM(x,"99.99")
??" : (f(x))= "+transform(fx,"99.9999")
  IF ABS(fx)<e
  ?" Корень найден, x= "+ALLTRIM(str(x))
  ??" за "+ALLTRIM(STR(i))+ " итераций !!!"
  RETURN
  ELSE
  IF fa*fx<0 THEN
  b=x
  endif
  IF fx*fb<0 THEN
  a=x
  endif
  ENDIF
ELSE
?"  Итерация= "+ALLTRIM(STR(i))
??" : f(a)= "+TRANSFORM(fa,"99.9999")
??" : f(b)= "+TRANSFORM(fb,"99.9999")
?" Отрезок [a,b] выбран неудачно !!!"
return
ENDIF

ENDFOR
? "Решение не найдено за", maxі, "итераций"

FUNCTION f
PARAMETERS x
RETURN (x-4)*(x-2)
ENDFUNC

PROCEDURE Tabulation
lparameters a,b,abstep
?"---Табуляция функции---"
FOR x=a TO b STEP abstep
?"x= "+TRANSFORM(x,"99.99")+ " : "+"f(x)=
"+TRANSFORM(f(x),"99.9999")
ENDFOR
?"-----"
ENDPROC

```

### Алгоритм решения задачи на **Visual Basic**:

```
Sub Dichotomia()
```

```
e = 0.001
```

```

maxi = 1000

a = Val(TextBox("Введите границу отрезка a: "))
b = Val(TextBox("Введите границу отрезка b: "))

Call Tabulation(a, b, 0.5)

For i = 1 To maxi
fa = f(a)
fb = f(b)

If fa * fb < 0 Then
x = (a + b) / 2
fx = f(x)
Debug.Print " Итерация= " + CStr(i)
Debug.Print " x= " + Format(x, "00.00")
Debug.Print " (f(x))= " + Format(fx, "00.0000")
    If Abs(fx) < e Then
        Debug.Print " Корень найден, x= " + CStr(x)
        Debug.Print " за " + CStr(i) + " итераций !!!"
        solution = True
        Exit Sub
    Else
        If fa * fx < 0 Then
            b = x
        End If
        If fx * fb < 0 Then
            a = x
        End If
    End If

Else
Debug.Print " Итерация= " + CStr(i)
Debug.Print " : f(a)= " + Format(fa, "00.0000")
Debug.Print " : f(b)= " + Format(fb, "00.0000")
Debug.Print " Отрезок [a,b] выбран неудачно !!!"

```

```
Exit Sub
End If

Next i
If Not solution Then Debug.Print "Решение не найдено за",
maxi, "итераций"
End Sub

Function f(x)
f = (x - 4) * (x - 2)
End Function

Sub Tabulation(a, b, abstep)
Debug.Print "---Табуляция функции---"
For x = a To b Step abstep
Debug.Print "x= " + Format(x, "00.00") + " : " + "f(x)=
" + Format(f(x), "00.0000")
Next x
Debug.Print "-----"
End Sub
```