

## **Поиск максимального и минимального значения функции одной переменной методом последовательного перебора.**

Большинство итерационных алгоритмов определения максимума (минимума) функции, как одной, так и нескольких переменных, применяются для определения локального максимума (минимума). Чтобы определить примерное расположение интересующего нас максимума (минимума) можно использовать табулирование функции или основанные на табулировании методы прямого перебора значений функции на некоторой сетке аргумента (или аргументов, в случае многомерной функции).

Рассмотрим функцию одной переменной  $f(x)$ . Пусть эта функция имеет максимум  $f_m=f(x_m)$  в точке  $x_m$  на отрезке  $a < x < b$ .

Описание алгоритма определения максимального (минимального) значения функции одной переменной методом последовательного перебора:

1. Задаем начальные значения рабочего шага  $h_0$ , точность определения максимума (минимума)  $d$ , начального значения аргумента функции  $x_0$  и величину.
2. Вычисляем значение функции в начальной точке  $f_0=f(x_0)$ .
3. В цикле вычисляем:
  - 3.1. новое приближение  $x$ :  $x_1=x_0+d$ ;
  - 3.2. новое значение функции  $f_1$  в точке  $x_1$ ;
  - 3.3. производную функции  $df=(f_1-f_0)/(x_1-x_0)$ ;
  - 3.4. в соответствии со знаком производной определяем рабочий шаг  $h=\text{знак}(df)*h_0$ ;
  - 3.5. если производная меньше нуля, то рабочий шаг  $h$  будет равен  $-h_0$ ;
  - 3.6. если значение функции в начальной точке меньше значения функции в следующей точке и если  $h_0$  меньше шага  $d$ , то максимум найден
  - 3.7. иначе дробим  $h_0$ .
4. Если цикл завершен, то печатать сообщение, что максимум (минимум) не найден.

Алгоритм определения максимального (минимального) значения функции одной переменной методом последовательного перебора на естественном языке

```
d=0.01
```

```
x0=0.5
```

```
h0=1
```

```
f0=функция(x0)
```

```
цикл по k от 1 до 10000
```

```
x1=x0+d
```

```
f1=функция(x1)
```

```
df=(f1-f0)/(x1-x0)
```

```
h=знак(df)*h0
```

```
x0=x0+h
```

```
f0=функция(x0)
```

```
если f0<f1 то
```

```
    если h0<d то
```

```
        печать "Максимум x0=", x0
```

```

        печать " f0= ", f0, " x1=", x1, " f1=", f1
        выход из программы
    конец если
h0=h0/5
конец если
конец цикла

```

печать "Максимум не найден"

```

ОПРЕДЕЛЕНИЕ ФУНКЦИИ функция
параметр x
возврат -10*(x-20)*(x-400)

```

Знак(x) в данном алгоритме – стандартная функция, возвращающая знак выражения x

**Пример решения на языке VFP:**

```

_screen.FontSize = 10
clear

d=0.01
x0=0.5

h0=1
f0=функция(x0)

FOR k=1 TO 10000
x1=x0+d
f1=функция(x1)
df=(f1-f0)/(x1-x0)
h=SIGN(df)*h0
x0=x0+h
f0=функция(x0)

IF f0<f1

IF h0<d
? "Максимум x0=", x0, " f0= ", f0, " x1=", x1, " f1=", f1
RETURN
endif
h0=h0/5
ENDIF
endifor

? "Максимум не найден"

FUNCTION функция
LPARAMETERS x
RETURN -10*(x-20)*(x-400)

```

**Пример решения на языке VBA:**

```

Sub maxf()

```

```

d = 0.01
x0 = 0.5
h0 = 1
f0 = функция(x0)

For k = 1 To 10000
x1 = x0 + d
f1 = функция(x1)
df = (f1 - f0) / (x1 - x0)
h = Sgn(df) * h0
x0 = x0 + h
f0 = функция(x0)

If f0 < f1 Then
If h0 < d Then
Debug.Print "Максимум x0=", x0, " f0= ", f0, " x1=", x1, "
f1=", f1
Return
End If

h0 = h0 / 5
End If
Next

Debug.Print "Максимум не найден"
End Sub

Function функция(x)
функция = -10 * (x - 20) * (x - 400)
End Function

```