

# Вычисление производных (численное дифференцирование)

При вычислении производной функции, будем иметь в виду, что один из способов найти производную

$$f'(x_0)$$

- это взять достаточно малые значения справа и слева на равном расстоянии от  $x_0$
- точке, в которой мы хотим найти производную.

$$f'(x_0) \approx \frac{f(x_0 + dx/2) - f(x_0 - dx/2)}{dx}$$

Таким образом, вычисляется производная в середине промежутка.

По значениям  $f'$  можно таким же способом найти производную от  $f'$ , т.е.  $f''$ . Можно выразить  $f''$  непосредственно через  $f(x)$ :

$$f''(x) \approx \frac{f'(x_0 + dx/2) - f'(x_0 - dx/2)}{dx} \approx \frac{\frac{f(x_0 + dx) - f(x_0)}{dx} - \frac{f(x_0) - f(x_0 - dx)}{dx}}{dx} \approx \frac{f(x_0 + dx) - 2f(x_0) + f(x_0 - dx)}{dx^2}.$$

Для производной третьего порядка можно использовать следующую формулу:

$$f'''(x) \approx \frac{f(x_0 + 2dx) - 2f(x_0 + dx) + 2f(x_0 - dx) - f(x_0 - 2dx)}{2dx^3}.$$

***Возникают естественные вопросы, откуда происходят эти формулы и как оценивать точность вычисления производных по этим формулам?***

**Формулы являются результатом дифференцирования интерполяционных многочленов Ньютона и других. Сущность которых состоит в том, что заданная функция  $f(x)$  представляется в виде многочлена, который значительно проще дифференцировать, чем какие-либо другие функции, особенно трансцендентные или представляющие собой сложные выражения.**

**Оценка погрешности и точности вычисления не менее серьезный и сложный процесс, чем само приближенное вычисление.**

**Так для оценки погрешности дифференцирования могут быть применены следующие формулы:**

$$r_n(x) = (-1)^n \frac{dx^n}{n+1} f^{(n+1)}(\xi), \quad (2)$$

**где предполагается, что функция  $f(x)$  дифференцируемая**

**- некоторое промежуточное значение  $\xi$  между  $x_0$  - точкой, в которой находится производная и точками  $(x_0 - 2dx)$ ,  $(x_0 - dx)$ ,  $(x_0 + dx)$ ,  $(x_0 + 2dx)$ , ... из заданного промежутка  $[a, b]$ .**

**На практике  $f^{(n+1)}(c)$  оценивать непросто, поэтому при малых  $dx$  приближенно полагают:**

$$f^{(n+1)}(\xi) \approx \frac{\Delta^{(n+1)} y_0}{dx^{n+1}}$$

**и тогда получается следующая формула**

$$r_n(x_0) \approx \frac{(-1)^n \Delta^{n+1} y_0}{dx^{n+1}} \quad (3)$$

**Мы будем пользоваться формулой (2), а впоследствии и формулой (3), в зависимости от конкретной задачи и тех сложностей, которые могут возникнуть при составлении программ.**

**Используя эти формулы, составим функцию для вычисления первой производной. Точность вычисления  $eps$  задается пользователем, а первоначальная величина промежутка  $dx$  устанавливается 1, а затем, для уточнения вычисления - делится на 2. Впрочем, читатель может предложить другие способы изменения промежутка  $dx$ , когда значительно быстрее достигается вычисление производной с**

**заданной степенью точности.**

*{ Вычисление 1-й производной и опред. точности ее вычислен. }*

*{ derivative - производная }*

**Function** derivat1(x0, eps : real) : real;

**var**

dx, dy, dy2 : real;

**begin**

dx := 1;

**repeat**

dx := dx/2;

dy := fx(x0 + dx/2) - fx(x0 - dx/2);

dy2 := fx(5\*x0/4 + dx) - 2\*fx(5\*x0/4);

dy2 := dy2 + fx(5\*x0/4 - dx)

**until** abs(dy2/(2\*dx)) < eps;

derivat1 := dy/dx

**end;**

**Здесь, для определения точности вычисления, используется вторая производная в точке**

$$c = 5 \cdot \frac{x_0}{4};$$

$dy2 := fx(5 * x0 / 4 + dx) - 2 * fx(5 * x0 / 4) + fx(5 * x0 / 4 - dx);$

Запись ее вычисления выполнена в две строки только из-за лучшей наглядности написания программы.

Возможен и другой вариант написания функции с использованием формулы (3) для оценки точности вычисления.

**Тогда функция запишется так:**

***{Вычисление 1-й производной и опред. точности ее вычислен.}***

***{ derivative - производная }***

```
Function derivat1(x0, eps : real) : real;  
  var  
    dx, dy, dy2 : real;  
begin  
  dx := 1;  
  repeat  
    dx := dx/2;  
    dy := fx(x0 + dx/2) - fx(x0 - dx/2);  
    dy2 := fx(5*x0/4 + dx) - 2*fx(5*x0/4);  
    dy2 := dy2 + fx(5*x0/4 - dx)  
  until abs((dy2*dy2*fx(x0))/(2*dx)) < eps;  
  derivat1 := dy/dx  
  
end;
```