|  |
| --- |
|  |

Содержание

1. *Задание*
2. *График заданной функции (функций) в MS EXCEL*
3. *Описание нахождения корня уравнения численным методом половинного деления*
4. *Описание нахождения корня уравнения численным методом касательных*
5. *Блок-схемы алгоритмов*
6. *Листинг программы*
7. *Полученные результаты работы программы*
8. *Проверка вычислений*
9. *Выводы*
10. *Список литературы*
11. Задание

*Построить график исходной функции в Excel для определения интервала, в котором лежит значение корня или протабулировать ее в найденном интервале. Найти приближенное значение корня с точностью Т=10-5 заданными методами. Подставить найденные корни в исходное уравнение и определить достигнутую точность решения для каждого метода.*

**

1. График заданной функции в MS EXCEL

*Из графика видно, что уточнять корень уравнения можно на отрезке [5;15]*

1. Описание нахождения корня уравнения численным методом половинного деления

*Если функция f(x) непрерывна и меняет знак на отрезке [a;b], то можно использовать метод половинного деления для поиска корней уравнения f(x)=0. Для этого требуется разделить отрезок [a;b] пополам точкой с: . (1)*

*Вычислим значение функции f(x)в точке с. Если f(c)=0, то с - корень уравнения, если нет, то выберем ту половину отрезка [a;b], на концах которой значения функции разных знаков и примем для дальнейшего рассмотрения отрезок [a1;b1], где либо a1 =с и b1=b, либо b1 =с и a1=a. Повторим те же операции с отрезком [a1;b1] и так далее, до тех пор, пока не будет точно найден корень уравнения или будет достигнута требуемая точность в его определении. Если точность обозначить как e, то остановить половинное деление необходимо при выполнении неравенства . (2)*

1. Описание нахождения корня уравнения численным методом касательных

*При уточнении коня методом касательных на отрезке [a;b], требуется, чтобы значения функции f(x) на его концах имели разный знак, функция f(x) имела на этом отрезке непрерывные первую и вторую производные, отличные от нуля и сохраняющие свой знак на отрезке [a;b].*

*Существует следующая теорема: пусть функция f имеет на некотором промежутке конечные производные до n+1 порядка включительно. Еcли числа x и x0 принадлежат этому промежутку, то найдется число с между x и x0 такое, что*

*. (3)*

*Если t – точный корень уравнения, лежащий внутри отрезка [a;b] и xn – принадлежит этому отрезку и является приближенным значением корня, тогда в силу приведенной теоремы между t и xn найдется точка сn , такая что*

*(4)*

*Обычно третье слагаемое очень мало, поэтому можно ограничится первыми двумя. Выражая из (4) t получим:. (5)*

*Так можно последовательно вычислять приближенные значения корня, на каждой итерации использую формулу . (6)*

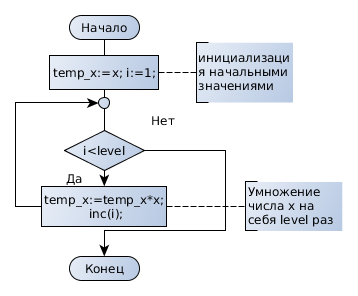
*Геометрический смысл метода заключается в том, что приближения корня равны абсциссам точек пересечения оси Ox и касательных к графику функции .*

*Рассмотрим вопрос выбора начального приближения корня. Существует правило, по которому если первая и вторая производные функции f(x) одного знака на отрезке [a;b], то следует принимать за начальное приближение точку b, если они разных знаков, то точку a.*

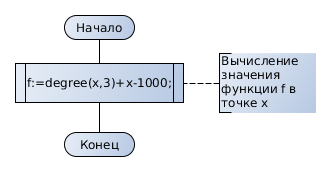
*Оценка погрешности приближения к корню x оценивается формулой:*

*, (7)*

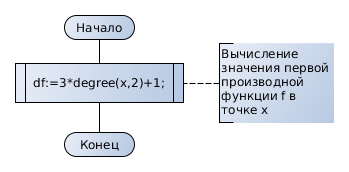
1. Блок-схемы алгоритмов
   1. *Подпрограмма возведения числа x в степень*



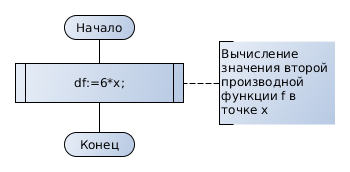
* 1. *Подпрограмма вычисления значения функции в точке x*



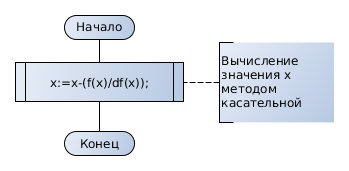
* 1. *Подпрограмма вычисления первой производной функции в точке x. Первая производная равна в точке x*



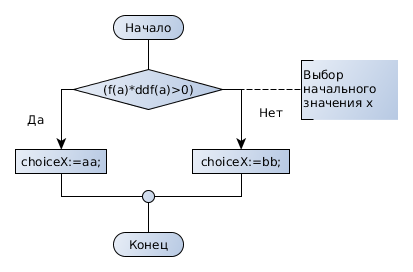
* 1. *Подпрограмма вычисления второй производной функции в точке x. Первая производная равна в точке x*



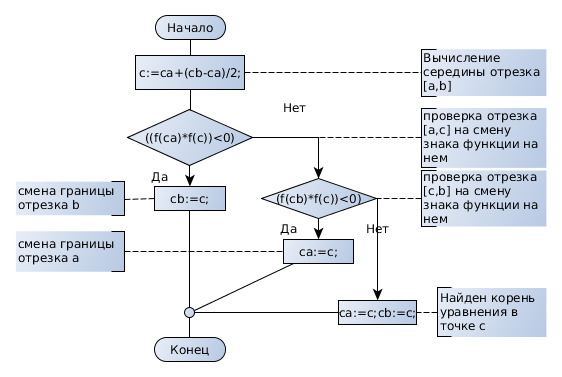
* 1. *Подпрограмма численного метода вычисления x методом касательных*



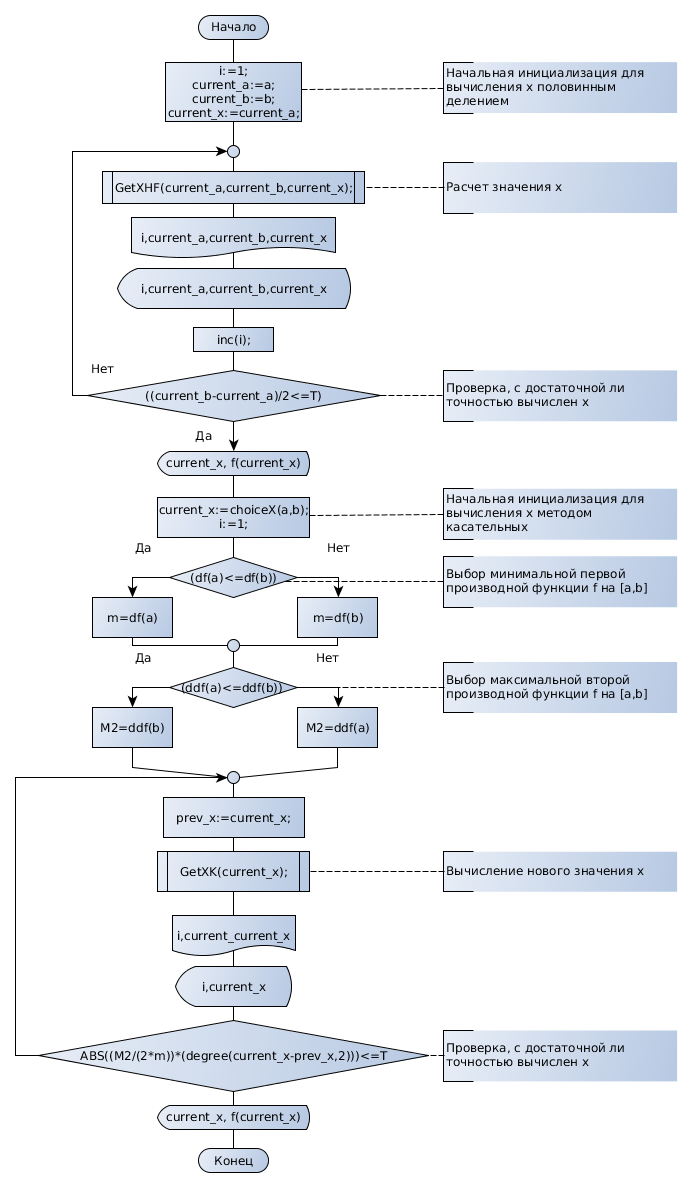
* 1. *Подпрограмма выбора начального значения х для метода касательной*



* 1. *Подпрограмма численного метода вычисления x методом половинного деления*



* 1. *Основная программа*



1. Листинг программы

*program evol;*

*Uses CRT;*

*const*

*a=5;*

*b=15;*

*T=0.00001;*

*var*

*i:integer;*

*current\_a,current\_b,current\_x,M2,m,prev\_x:real;*

*tf:text;*

*function degree(x:real;level:integer):real;*

*var i:integer;*

*temp\_x:real;*

*begin*

*temp\_x:=x;*

*i:=1;*

*while (i<level) do begin*

*temp\_x:=temp\_x\*x;*

*inc(i);*

*end;*

*degree:=temp\_x;*

*end;*

*function f(x: real ):real;*

*begin*

*f:=degree(x,3)+x-1000;*

*end;*

*function df(x:real):real;*

*begin*

*df:=3\*degree(x,2)+1;*

*end;*

*function ddf(x:real):real;*

*begin*

*ddf:=6\*x;*

*end;*

*procedure GetXHF(var ca:real; var cb:real;var x:real);*

*var c:real;*

*begin*

*c:=ca+(cb-ca)/2;*

*if ((f(ca)\*f(c))<0) then cb:=c else*

*if ((f(cb)\*f(c))<0) then ca:=c else*

*if ((f(cb)\*f(c))=0) then begin ca:=c; cb:=c; end;*

*x:=ca+(cb-ca)/2;*

*end;*

*procedure GetXK(var x:real);*

*begin*

*x:=x-(f(x)/df(x));*

*end;*

*function choiceX(aa:real;bb:real): real;*

*begin*

*if (f(a)\*ddf(a)>0) then choiceX:=aa*

*else choiceX:=bb;*

*end;*

*procedure OutData(i:integer;ca:real;cb:real;cx:real;var myf:text);*

*begin*

*writeln('|',i:8,'|',ca:9:6,'|',cb:9:6,'|',cx:9:6,'|');*

*writeln('-------------------------------------');*

*writeln(myf,'|',i:8,'|',ca:9:6,'|',cb:9:6,'|',cx:9:6,'|');*

*writeln(myf,'----------------------------------------');*

*end;*

*procedure OutData2(i:integer;cx:real;var myf:text);*

*begin*

*writeln('|',i:8,'|',cx:9:6,'|');*

*writeln('--------------------');*

*writeln(myf,'|',i:8,'|',cx:9:6,'|');*

*writeln(myf,'--------------------');*

*end;*

*begin*

*clrscr;*

*i:=1;*

*current\_a:=a;*

*current\_b:=b;*

*current\_x:=current\_a;*

*writeln('Calculation y(x)=x^3+x-1000 on [5,15] with half-part division method:');*

*writeln('----------------------------------------');*

*writeln('| N | a | b | x |');*

*writeln('----------------------------------------');*

*assign(tf,'Rez.txt');*

*rewrite(tf);*

*writeln(tf,'Calculation y(x)=x^3+x-1000 on [5,15] with half-part division method:');*

*writeln(tf,'----------------------------------------');*

*writeln(tf,'| N | a | b | x |');*

*writeln(tf,'----------------------------------------');*

*repeat*

*GetXHF(current\_a,current\_b,current\_x);*

*OutData(i,current\_a,current\_b,current\_x,tf);*

*inc(i);*

*until ((current\_b-current\_a)/2<=T);*

*writeln(' x=',current\_x:9:6,' y(',current\_x:9:6,')=',f(current\_x):9:6);*

*writeln(tf,'x=',current\_x:9:6,' y(',current\_x:9:6,')=',f(current\_x):9:6);*

*writeln('Calculation y(x)=x^3+x-1000 on [5,15] with tangential method:');*

*writeln('--------------------');*

*writeln('| N | x |');*

*writeln('--------------------');*

*writeln(tf,'Calculation y(x)=x^3+x-1000 on [5,15] with tangential method:');*

*writeln(tf,'--------------------');*

*writeln(tf,'| N | x |');*

*writeln(tf,'--------------------');*

*current\_x:=choiceX(a,b);*

*i:=1;*

*if (df(a)<=df(b)) then m:=df(a) else m:=df(b);*

*if (ddf(a)<=ddf(b)) then M2:=ddf(b) else M2:=ddf(a);*

*repeat*

*prev\_x:=current\_x;*

*GetXK(current\_x);*

*OutData2(i,current\_x,tf);*

*inc(i);*

*until (ABS((M2/(2\*m))\*(degree(current\_x-prev\_x,2)))<=T);*

*writeln(' x=',current\_x:9:6,' y(',current\_x:9:6,')=',f(current\_x):9:6);*

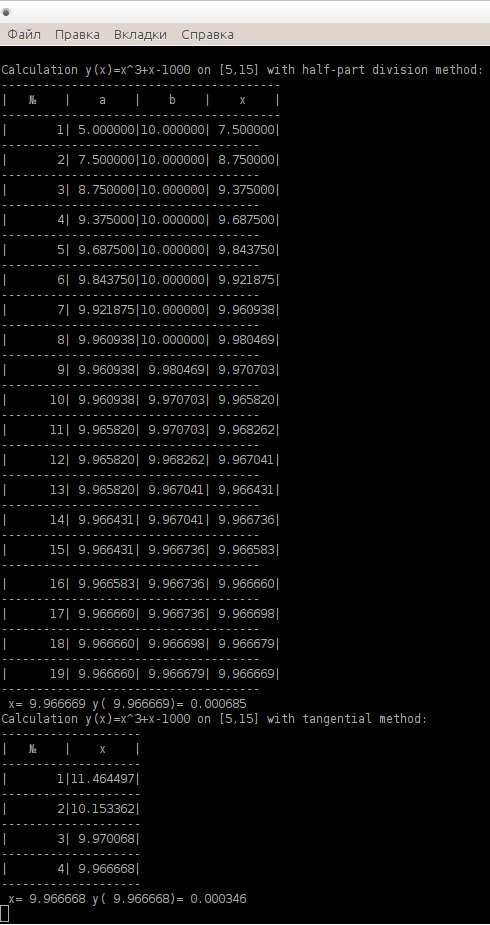
*writeln(tf,'x=',current\_x:9:6,' y(',current\_x:9:6,')=',f(current\_x):9:6);*

*close(tf);*

*readln;*

*end.*

1. Полученные результаты работы программы в виде скриншота

**

1. Выводы

*Для вычисления корня уравнения были использованы два численных метода: метод половинного деления и метод касательных.*

*Оба решения позволили получить приближенное значение корня уравнения с точностью e=10-5. Однако подстановка корня в исходное уравнение выявила различия в точности решения. Решение полученное методом касательных (0,00035) оказалось точнее решения, полученного половинным делением (0,00069). Кроме того, вычисление корня уравнения методом половинного деления потребовало 19 итераций до достижения требуемой точности корня, а решение, полученное методом касательных только – 4.*

1. Литература
2. *Исаков В.Б. Элементы численных методов: Учебное пособие для студентов, обучающихся по специальности Математика группы Педагогические специал. - М: Академия, 2003. – 192 с. : ил.*
3. *Фаронов В.В. Turbo Pascal 7.0. Учебный курс: Учебное пособие –М: КноРус, 2011. -727 с.: ил.*