МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МУРМАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ИНФОРМАТИКА

Методические указания

к расчётно –графической работе №1 для студентов 1 курса очной и заочной формы обучения специальности 26.03.02 «Кораблестроение, океанотехника и системотехника объектов морской инфраструктуры»

Мурманск

2019

ОГЛАВЛЕНИЕ

Введение

Методические указания предназначены для расчётно- графического задания по Информатике.

Цель курса: выработать практические навыки работы с компьютером и его основным программным обеспечением, которое широко используется на практике.

Все рассматриваемые темы завершаются списком заданий для самостоятельной работы и контрольными вопросами.

I. Расчётно – графическое задание №1

“Численные методы решения алгебраических и трансцендентных уравнений”

Цель: научиться применять численные методы для нахождения корней трансцендентных уравнений.

Задание:

1) Написать программу отделения действительных корней уравнения на языке программирования Pascal;
2) Написать программы уточнения корней двумя способами: методом половинного деления, методом Ньютона (касательных) на языке Pascal.
3) Построить график заданной функции с помощью табличного процессора.
4) Проверить полученные результаты в программе Scilab.

1. Отделение корней уравнения

Методические указания.

Постановка задачи.

Пусть имеется уравнение вида $F(x)=0$, где $F(x)$- алгебраическая или трансцендентная функция[1].

Решить такое уравнение – значит установить, имеет ли оно корни, сколько корней, и найти значения корней с нужной точностью.

Если непрерывная функция $F(x)$ на концах отрезка [A, B] принимает значения разных знаков, т.е. $f(a)\cdot f(b)<0$, то внутри отрезка существует, по крайней мере, один корень уравнения $F(x)=0$. Требуется отделить корни уравнения, т.е. указать все отрезки [a, b]$\subset$[A, B], содержащие по одному корню.

---

[1] Пантина И. В., Синчуков А. В. Вычислительная математика: учебник / И. В. Пантина, А. В. Синчуков. – М.: Маркет ДС, 2010. – 176 с. (Университетская серия).

Будем вычислять значения $F(x)$, начиная с точки $x = A$, двигаясь вправо с некоторым шагом h (рис.1).

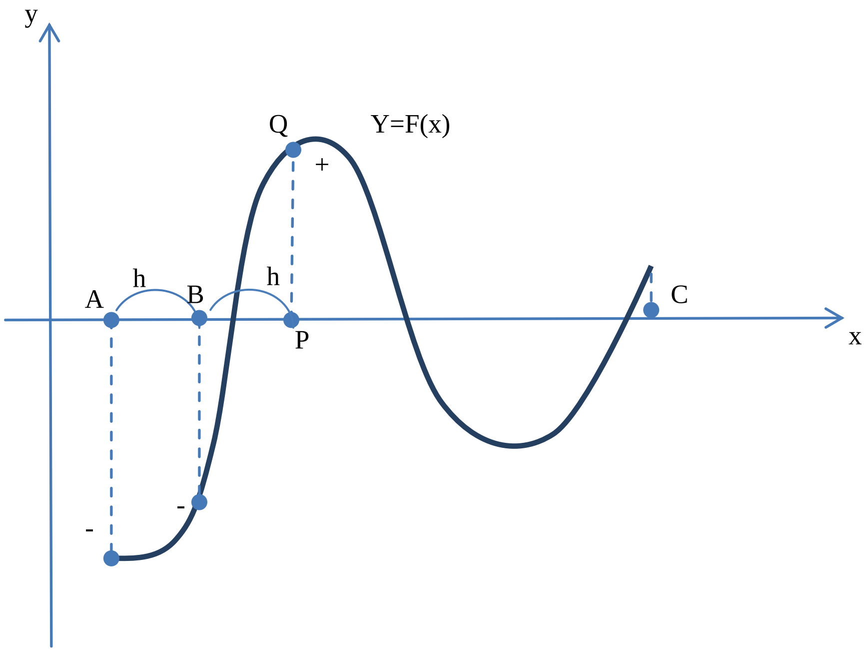


Рис. 1. Отделение корней уравнения

Как только обнаружится пара соседних значений $F(x)$, имеющих разные знаки, и функция $F(x)$ монотонная на этом отрезке, так соответствующие значения аргумента x можно считать концами отрезка, содержащего корень [5].

Графический способ локализации корней состоит в следующем: уравнение $F(x)$ представляют в виде $f_1(x) = f_2(x)$. Строятся графики этих функций. Тогда абсциссы точек пересечения этих графиков и будут точными корнями (точными решениями) исходного уравнения [5].

Пример 1. Отделить графически корни уравнения $x^3 - x^2 - 3x + 1 = 0$.

1. Перепишем исходное уравнение в виде $x^3 = x^2 + 3x - 1$.

   Тогда $f_1(x) = x^3, \quad f_2(x) = x^2 + 3x - 1$.

2. Построим графики этих функций в одной системе координат и локализуем абсциссы точек их пересечения. Графики функций

пересекаются в трёх точках, поэтому заданное уравнение имеет три действительных корня (больше корней оно иметь не может, поскольку левая часть уравнения - многочлен третьей степени):

$x_1^* \in [-2;-1], \quad x_2^* \in [0;1], \quad x_3^* \in [2;3]$ (рис. 2).

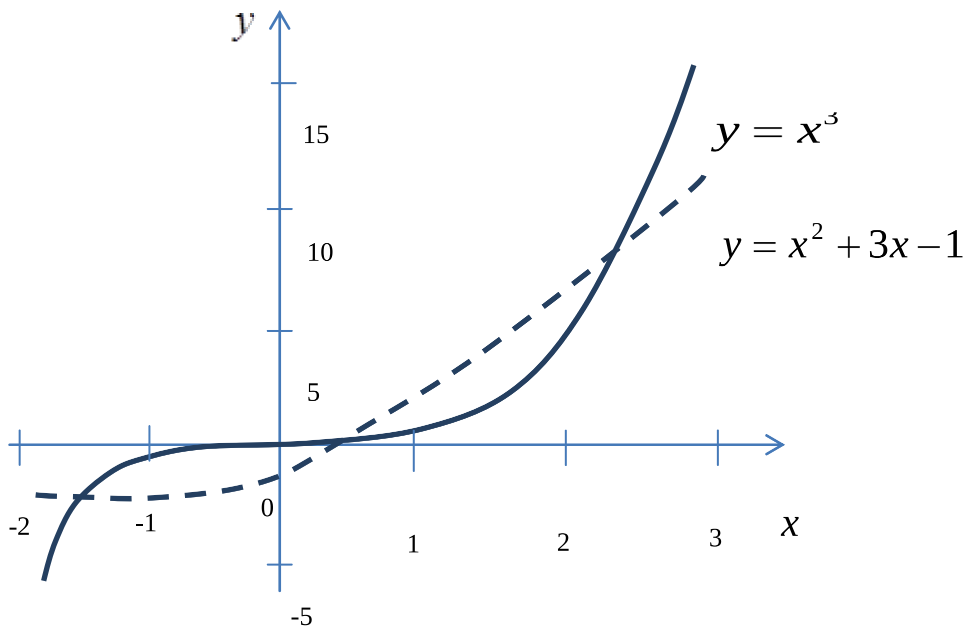


Рис.2. Графический способ отделения корней уравнения

Пример 2. Протабулируем функцию $\cos(x) = 0{,}2x$ на отрезке [-10, 10], $h = 0{,}1$. Решим эту задачу с помощью Excel (рис.3):

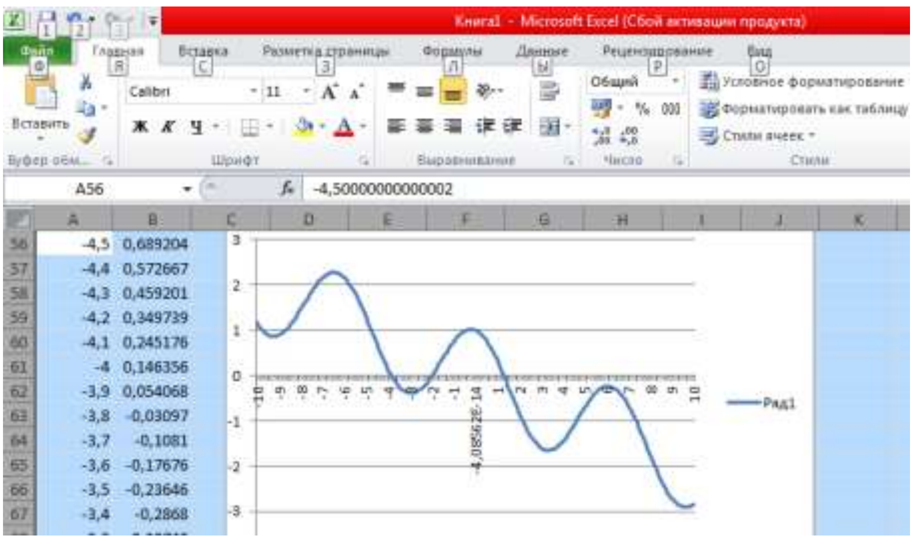


Рис. 3. Отделение корней функции с помощью MS Excel

Данная функция имеет три отрезка отделения корня с шагом 0,1:

$-3{,}9 < x_1 < 3{,}8$; $-2 < x_2 < 1{,}9$; $1{,}3 < x_3 < 1{,}4$

Пример 3. Решение задачи на языке программирования Pascal abc[4] (рис. 4).

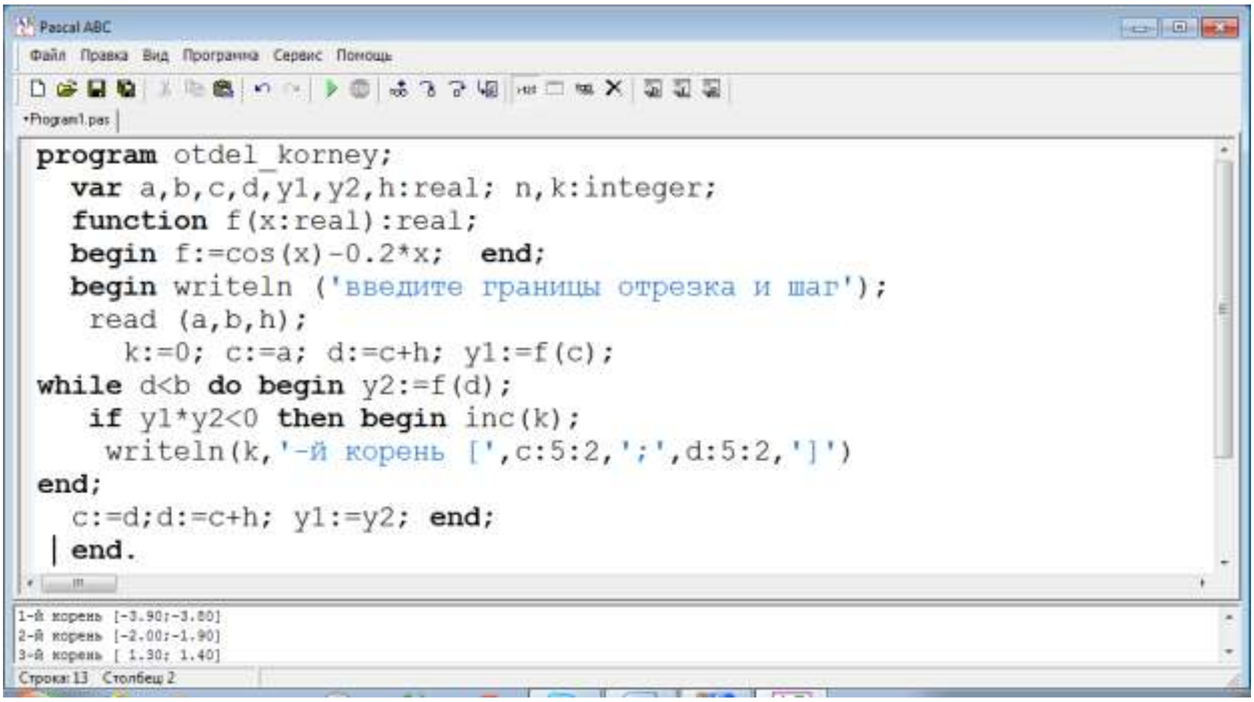
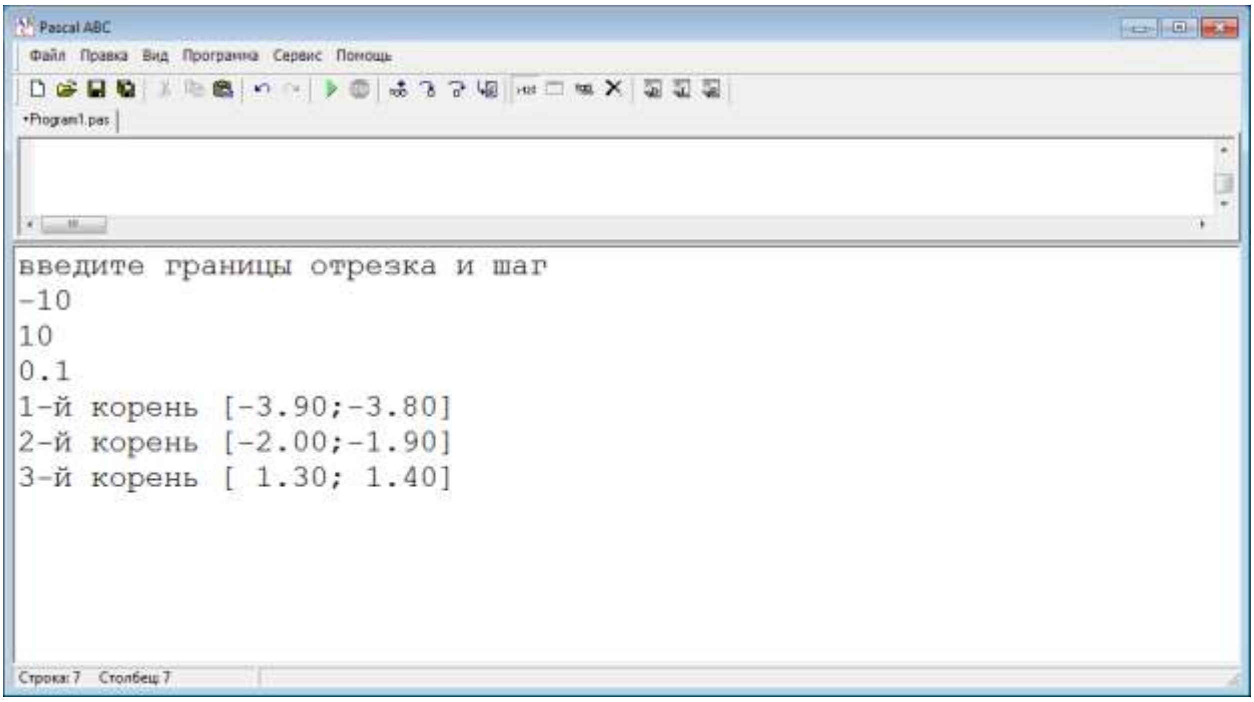


Рис. 4. Программа отделения корней уравнения

Результат работы программы:



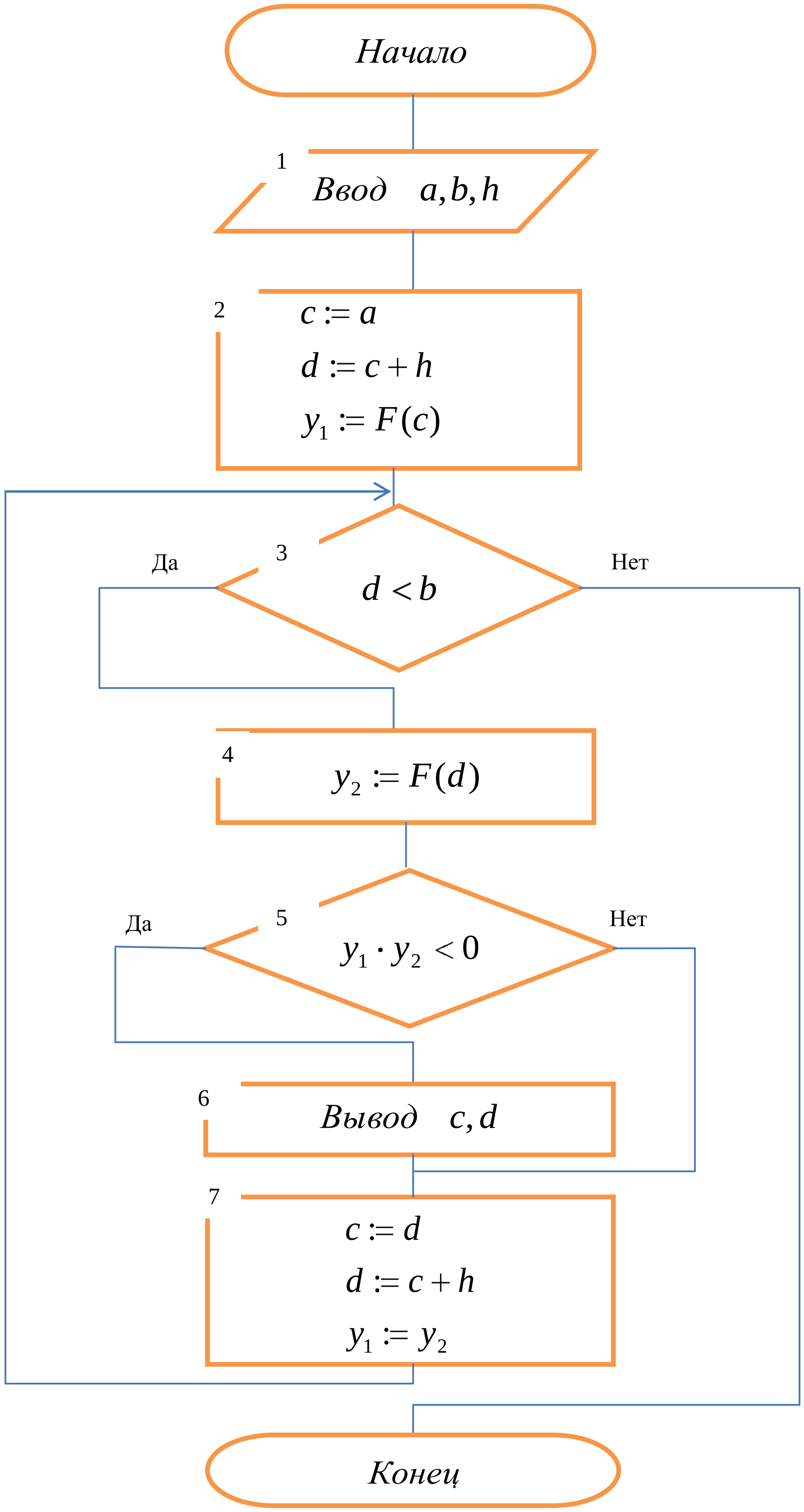


Рис. 5. Блок – схема алгоритма отделения корней уравнения

Надёжность данного алгоритма зависит от характера функции $F(x)$ и от величины шага h[2].

2. Уточнение корня уравнения методом половинного деления

Методические указания.

Постановка задачи.

Пусть функция $F(x)=0$ имеет на отрезке [a, b] единственный корень, причём функция $F(x)$ непрерывна на этом отрезке. Разделим отрезок [a, b] пополам точкой $c=\frac{(a+b)}{2}$. Если $F(c)\neq 0$ (что наиболее вероятно), то возможны два случая: $F(x)$ меняет знак либо на отрезке [a, c], либо на отрезке [c, b]. Выбирая в каждом случае тот из отрезков, на котором функция меняет знак, и, продолжая путь половинного деления дальше, можно дойти до сколь угодно малого отрезка, содержащего корень уравнения [5].

Если на каком-то этапе процесса получен отрезок [a, b], содержащий корень, то, приняв приближённо $x=\frac{(a+b)}{2}$, получим ошибку, не превышающую значения $\Delta x=\frac{(b-a)}{2}$ (погрешность метода) [5] (рис. 6).

[2] Лапчик М. П. Численные методы: Учеб. пособие для студ. вузов / М. П. Лапчик, М. И. Рагулина, Е. К. Хеннер;Под ред. М. П. Лапчика. – М.: Издательский центр «Академия», 2004. – 384 с.

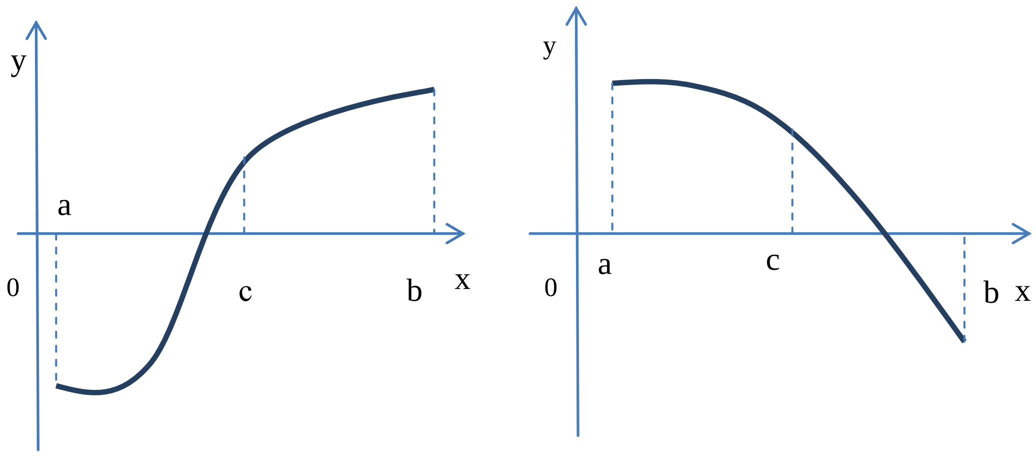
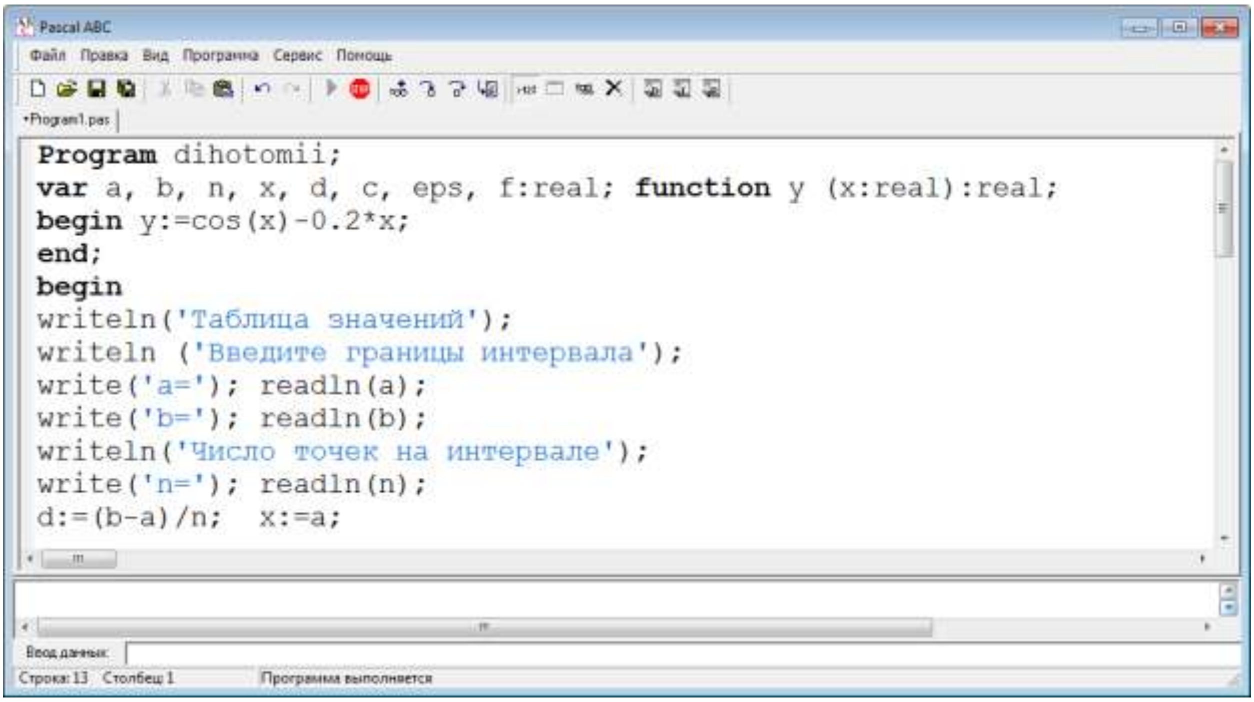


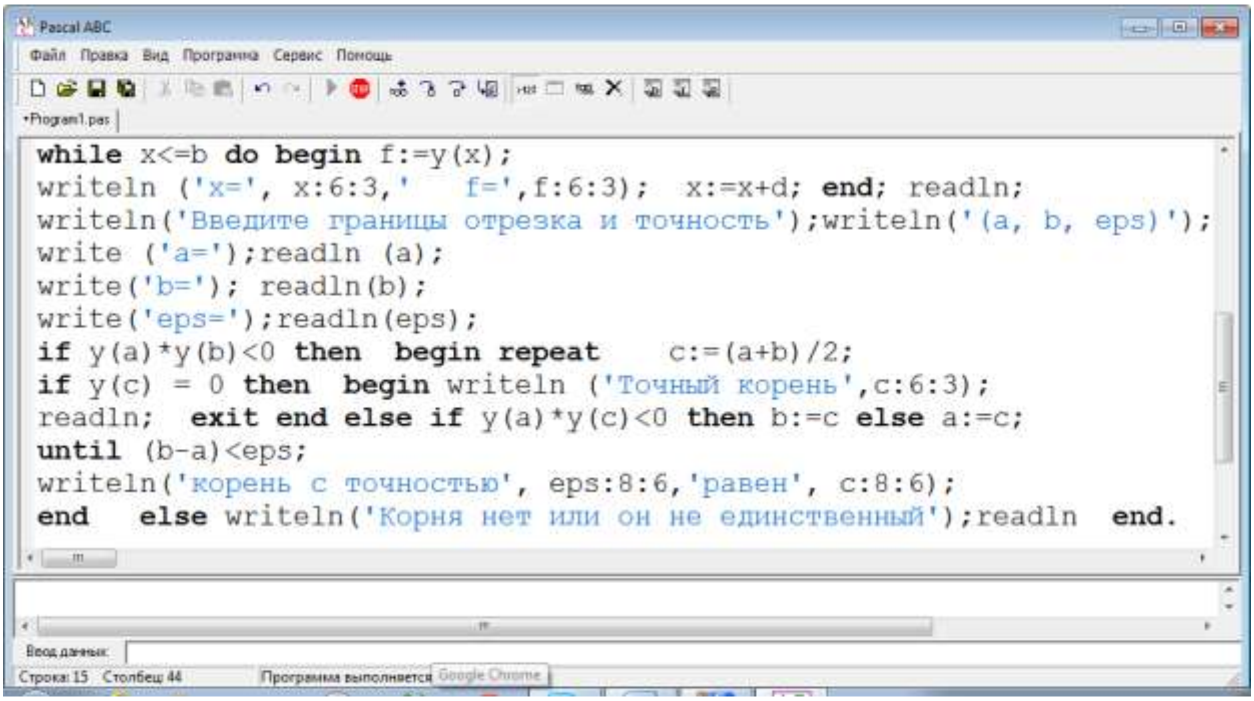
Рис. 6. Функция $F(x)$ меняет знак на отрезке [a; c]; функция $F(x)$ меняет знак на отрезке [c; b].

Пример 2. Решить это уравнение $\cos(x) - 0{,}2x = 0$ методом деления отрезка пополам[3] с точностью до $1 \cdot 10^{-4}$ (рис. 7).



```
Program dihotomii;
var a, b, n, x, d, c, eps, f:real; function y (x:real):real;
begin y:=cos(x)-0.2*x;
end;
begin
writeln('Таблица значений');
writeln ('Введите границы интервала');
write('a='); readln(a);
write('b='); readln(b);
writeln('Число точек на интервале');
write('n='); readln(n);
d:=(b-a)/n;  x:=a;
```

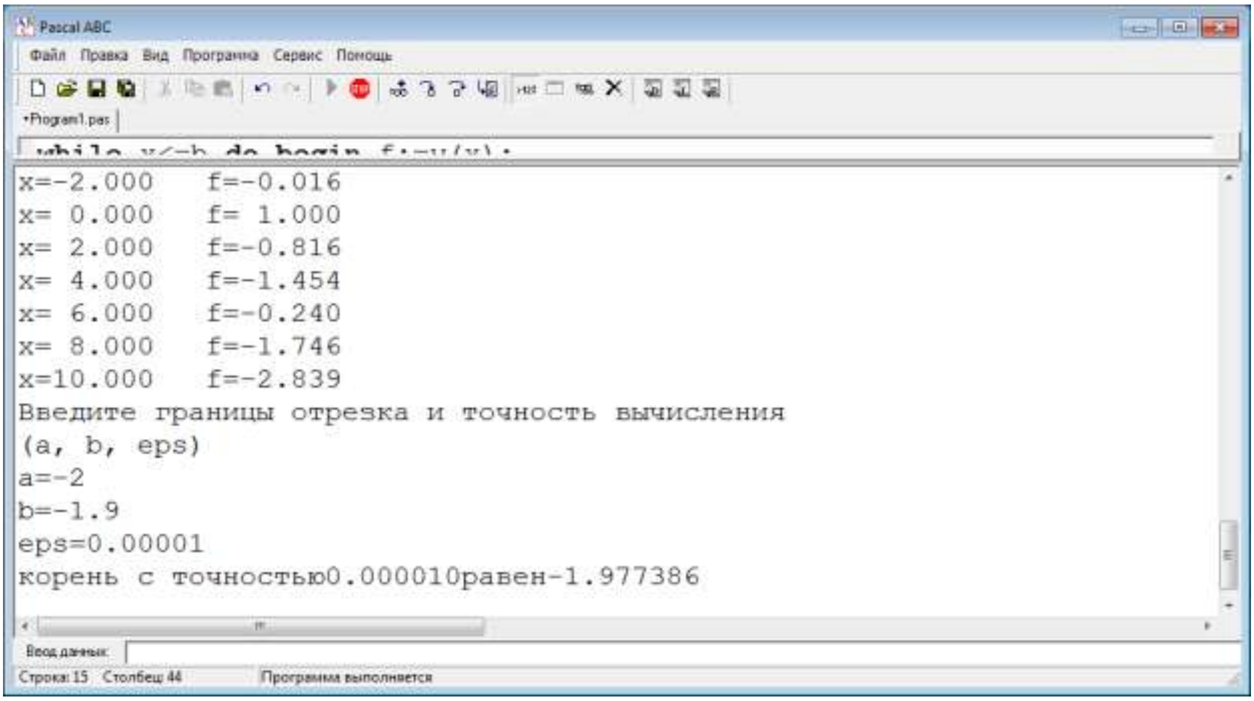

[3] Адаменко А.Н. Pascal на примерах из математики. - СПб.: БХВ-Петербург, 2005. – 416 с.: ил.



```
while x<=b do begin f:=y(x);
writeln ('x=', x:6:3,'    f=',f:6:3);  x:=x+d; end; readln;
writeln('Введите границы отрезка и точность');writeln('(a, b, eps)');
write ('a=');readln (a);
write('b='); readln(b);
write('eps=');readln(eps);
if y(a)*y(b)<0 then  begin repeat    c:=(a+b)/2;
if y(c) = 0 then  begin writeln ('Точный корень',c:6:3);
readln;  exit end else if y(a)*y(c)<0 then b:=c else a:=c;
until  (b-a)<eps;
writeln('корень с точностью', eps:8:6,'равен', c:8:6);
end   else writeln('Корня нет или он не единственный');readln  end.
```

Рис. 7. Программа, реализующая метод дихотомии

Результат работы программы:



```
x=-2.000   f=-0.016
x= 0.000   f= 1.000
x= 2.000   f=-0.816
x= 4.000   f=-1.454
x= 6.000   f=-0.240
x= 8.000   f=-1.746
x=10.000   f=-2.839
Введите границы отрезка и точность вычисления
(a, b, eps)
a=-2
b=-1.9
eps=0.00001
корень с точностью0.000010равен-1.977386
```

3. Уточнение корня уравнения методом Ньютона (касательных)

Методические указания.

Постановка задачи.

Пусть уравнение $F(x)=0$ имеет единственный корень на отрезке [a, b]. Преобразуем его к равносильному уравнению

$$x = x - \varphi(x)\cdot F(x), \qquad (1)$$

где $\varphi(x)$ - любая функция, определённая на отрезке [a, b] и не обращается на нём в ноль. Осуществляя различными способами выбор $\varphi(x)$, можно получить, в частности, данный метод [4].

Пусть $\varphi(x)=\dfrac{1}{F'(x)}$. Таким образом, итерационная последовательность строится с помощью рекуррентного соотношения

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)} \quad (n = 0, 1, 2, ...) \qquad (2)$$

Вопрос о выборе начального приближения x_0 и гарантированной сходимости итераций решается просто, если функция $F(x)$ удовлетворяет следующим условиям:

1) Является дважды дифференцируемой на отрезке [a, b];
2) Обе производные - первая и вторая не меняют знак на этом отрезке, т.е. функция $F(x)=0$ монотонная и не меняет характера выпуклости (рис. 3);

В такой ситуации за x_0 берётся тот конец отрезка [a, b], на котором функция $F(x)$ и её вторая производная имеют одинаковые знаки, т. е. выполняется условие $F(x_0)\cdot F''(x_0)>0$ (рис. 8).

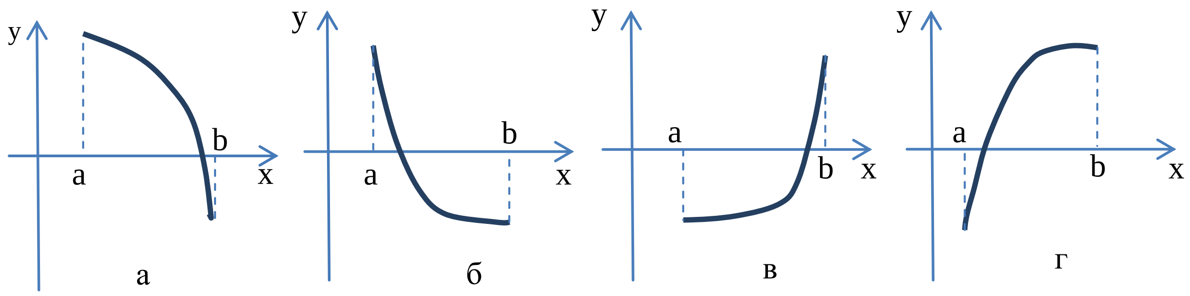


Рис. 8. Четыре возможности поведения функции $F(x)$ в окрестности корня:
а – функция убывает и выпукла; б – функция убывает и вогнута; в – функция возрастает и вогнута; г – функция возрастает и выпукла.

Данный метод называется методом касательных потому, что точка x_1, определяемая по формуле (2) при $n = 0$, есть точка пересечения касательной, проведённой к графику функции $y = F(x)$ в точке с абсциссой x_0, с осью абсцисс [4] (рис. 9).

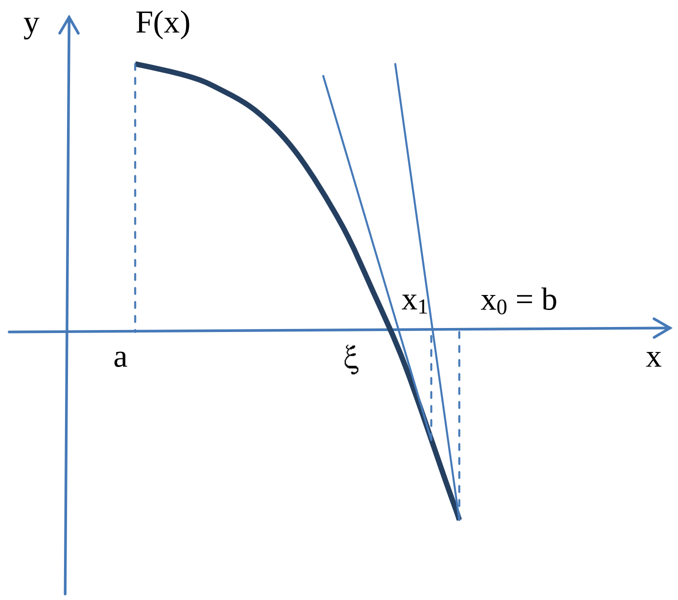


Рис. 9. Геометрический смысл метода касательных

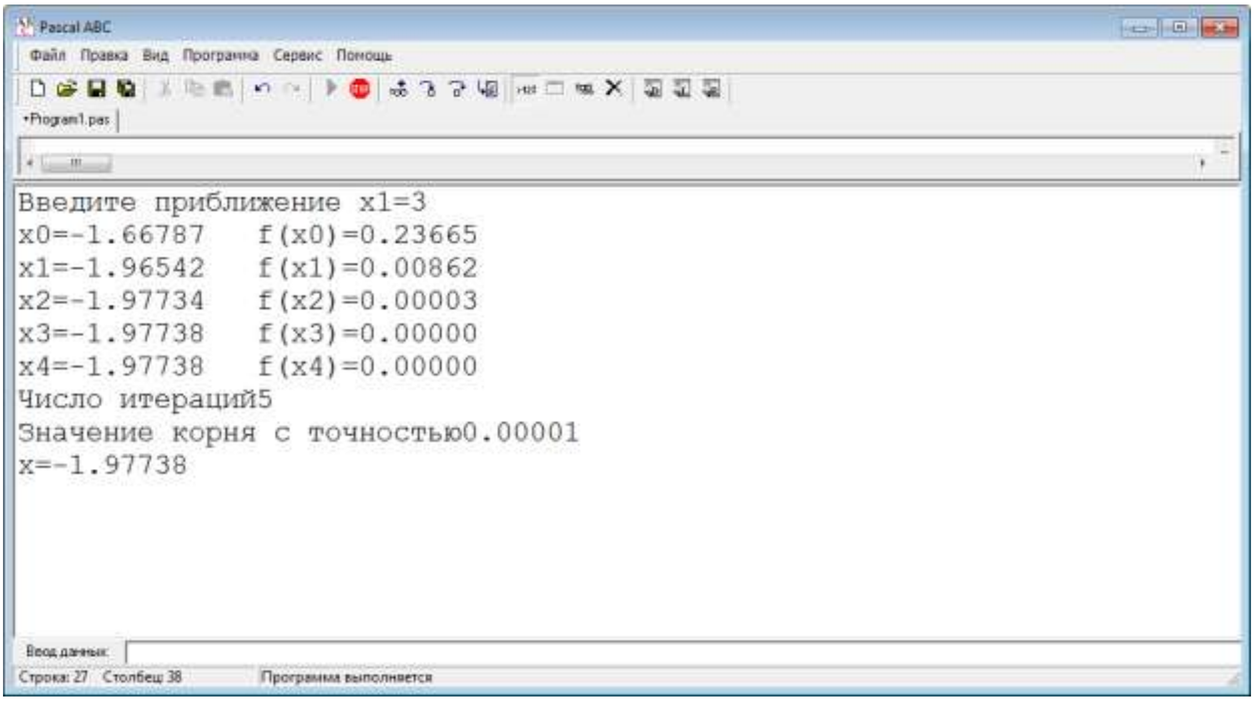
Пример .1. Решить это уравнение $\cos(x) - 0{,}2x = 0$ методом касательных с точностью до $1 \cdot 10^{-4}$ [1] (рис. 10).

```
program method_kasatelnyx;
  const delta=0.001; eps=0.00001;
  var a,b,x1,x0:real;
   function f(x:real):real;
    begin  f:=cos(x)-0.2*x;
      end;
   function df(x:real):real;
    begin
     df:=(f(x+delta)-f(x))/delta;
      end;
   function Newton (x1:real):real;
    var n:integer; x0:real;
    begin  n:=0;
```

```
    repeat  x0:=x1;
      x1:=x0-f(x0)/df(x0);
      writeln ('x',n,'=',x1:7:5,'   f(x',n,')=',f(x1):7:5);
      n:=n+1;
    until abs(x0-x1)<=eps;
     writeln ('Число итераций',n);
     Newton:=x1;
    end;
  begin
   write('Введите приближение x1='); readln(x1);
   x0:=Newton(x1);
   writeln ('Значение корня с точностью',eps:7:5);
   writeln('x=',x0:7:5);   readln   end.
```

Рис.10. Программа, реализующая метод касательных

Результат работы программы:



4. Построение графика функции

Методические указания.

Переменные *cx, cy* определяют центр системы координат.

Функции *getmaxx* и *getmaxy* определяют для данного видеорежима наибольшее экранное значение по оси X и по оси У. Поскольку коодинаты точки экрана должны быть целочисленными, то используется функция округления до целочисленного *trunc*.

Построение осей координат: *line (10, cy, getmaxx - 50, cy)*. Рисуется прямая линия, начиная от точки экрана (10, cy) до точки экрана *(getmaxx - 50, cy)*, т.е. посередине экрана с отступом 10 позиций слева и 50 позиций справа.

Переменные *mx, my* определяют масштаб вывода на экран.

Для вывода значений найденныхкорней функции в графическом режиме необходимо преобразовать значение вещественной переменной в строковую переменную: str (root (x - 0.1, x + 0.1, 0.0001) : 7 : 4, strx). Здесь root (x - 0.1, x + 0.1, 0.0001) : 7 : 4 – значение найденного корня на отрезке x - 0.1 до x + 0.1 с точностью до одной десятитысячной в формате вещественного числа 7:4 [1].

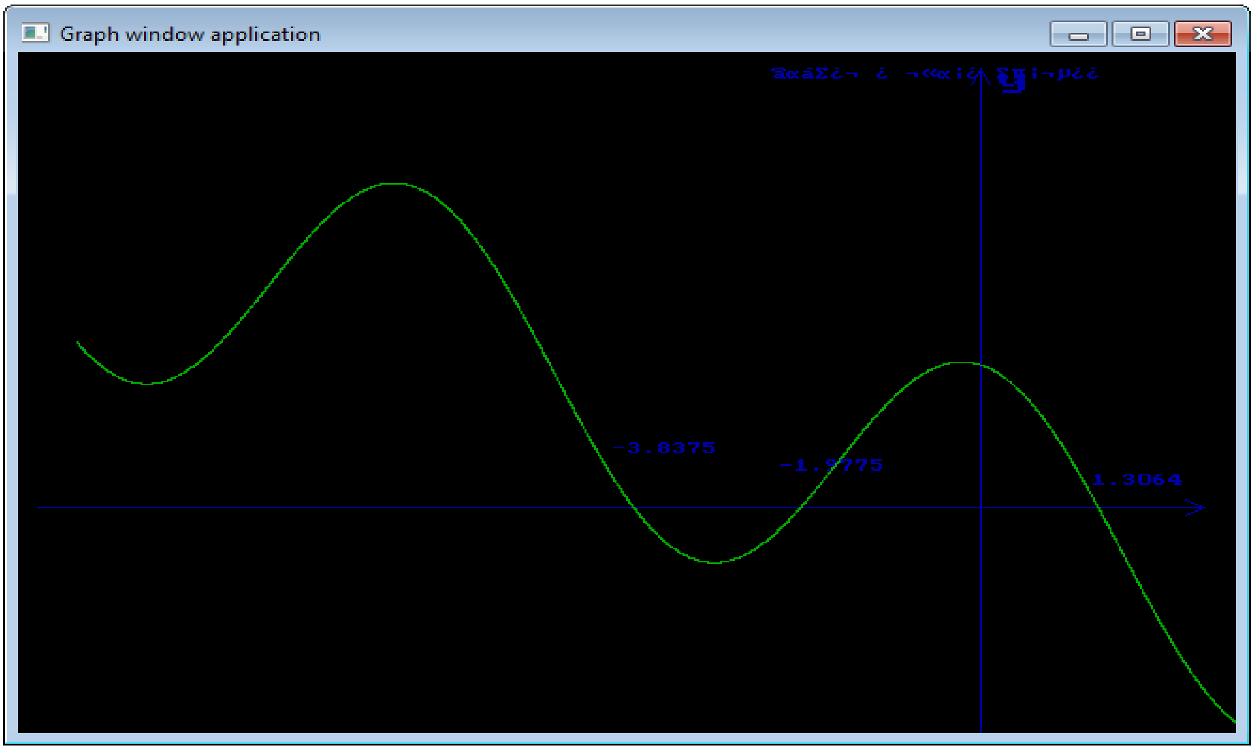
Пример. 1. Построить график функции $\cos(x) = 0{,}2x$ на отрезке [-10, 10], $h = 0{,}1$ и вывести корни [1].

```
Program graphic;
   uses crt, graph;
    const n = 10000;
   var x, y:real;
   driver, mode : integer;
cx, cy, mx, my, ex, ey : integer;
a, b, s : real;
strx, stry : string;
function f ( x : real) : real;
   begin
    f:=cos(x)-0.2*x;
  end;
function root ( a, b, e : real ) : real;
    var c : real;
begin
    repeat
      c  : = (a + b) / 2;
       if f ( a ) * f ( c ) < = 0 then b : = c else a : = c
    until abs ( f ( c ) ) < e;
      root : = c
 end;
function maxabsy ( a , b : real ) : real;
   var m,x:real;
   begin
      m : = abs ( f ( a ) ); x : = a + ( b – a ) / n;
repeat
```

```
if abs ( f ( x ) ) > m then m : = abs ( f ( x ) );
x : = x + ( b – a ) / n;
until x > b;
maxabsy : = m;
end;
procedure grafik (a, b : real );
   var p : integer;
      sw : real;
      strx : string;
begin
  cx : = trunk ( getmaxx / 2 );
  cy : = trunk (getmaxy / 2 );
  line ( 10, cy, 630, cy ); line ( 620, cy - 5, 630, cy ); line ( 620, cy + 5, 630, cy );
  line (cx - 5, 20, cx, 10 ); line (cx + 5, 20, cx, 10 ); line (cx, 10, cx, 470 );
 settextstyle ( 0, 0, 2 ); outtextxy ( cx + 10, 10, ' y ' );
 outtextxy ( getmaxx - 20, cy + 10, ' x ' ); settextstyle ( 0, 0, 0 );
outtextxy (400, 10, ' График и корни функции ' );
  s : = ( b – a ) / n;
 mx : = trunk ( ( getmaxx – 50 ) / ( b – a ) );
 my : = trunk ( ( getmaxy – 50 ) / ( 2 * maxabsy ( a, b ) ) );
 x : = a;  y : = f ( x );
 p : = - 40;
 sw : = a - 0.3;
 repeat
 ex : = trunk ( cx + x * mx ); ey : = trunk ( cy – y * my );
 if ( abs ( f ( x ) ) < = 0.01) and ( abs ( sw – x ) > 0.3) then
begin
settextstyle ( 0, 0, 0 );
str ( root ( x - 0.1, x + 0.1, 0.0001 ) : 7 : 4, strx );
```

```
outtextxy ( ex - 10, ey + p, strx );  p: = p + 10;
sw : = x;
end;
putpixel ( ex, ey, green );
x : = x + s;  y : = f ( x );
until x > b;
settextstyle ( 0, 0, 1 );
end;
begin
clrscr;
write ( ' левая граница a = ' );  readln ( a );
write ( ' правая граница b = ' ); readln ( b );
driver : =detect; initgraph (driver, mode, ' c : \ tp \ bgi ' );
setcolor ( blue ); setbkcolor ( yellow );
grafik ( a, b );
readln
end.
```

Результат работы программы:



5. Нахождение корней уравнения с помощью программы SCILAB:

Методические указания.

Для решения трансцендентных уравнений в Scilab применяют функцию fsolve(x0, f), где x0 - начальное приближение, f – функция, описывающая левую часть уравнения $y(x)=0$.

Пример 1. Построить график и найти решение уравнения $\cos(x)-0,2x=0$ (рис. 11) и (рис.12).

```
-->x=[-10:0.1:10];

-->y=cos(x)-0.2*x;

-->plot2d(x,y',style=[color("red")],axesflag=5);
```

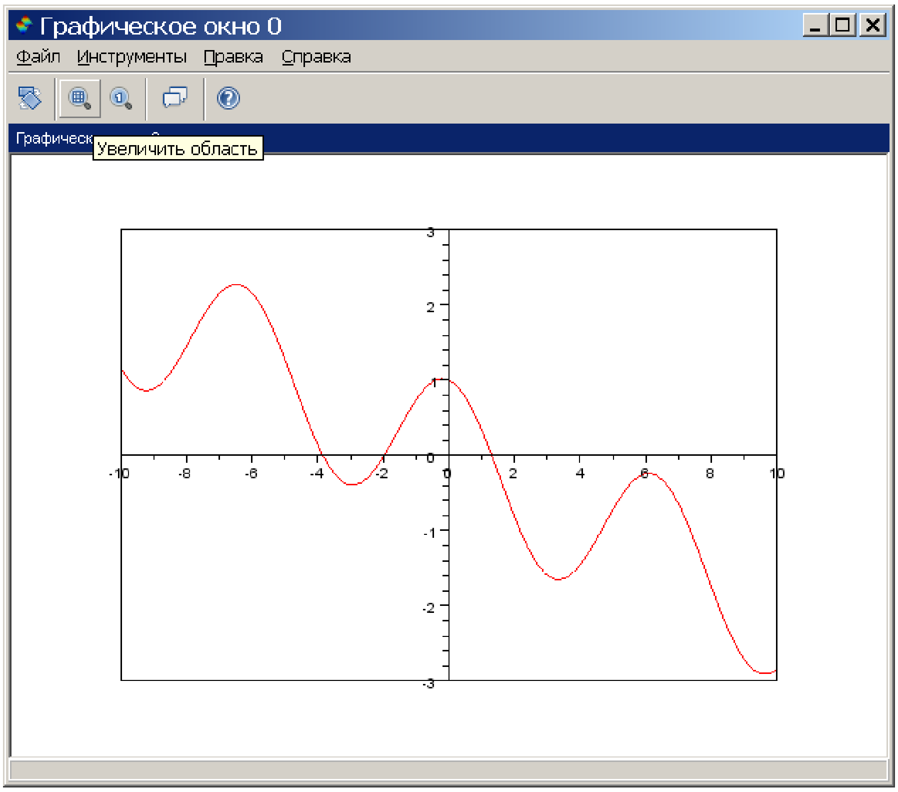


Рис. 11. Построение графика функции в системе Scilab

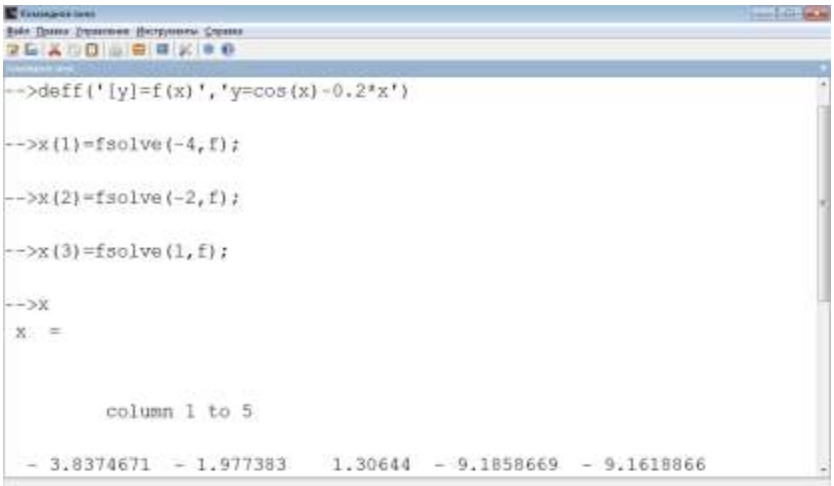
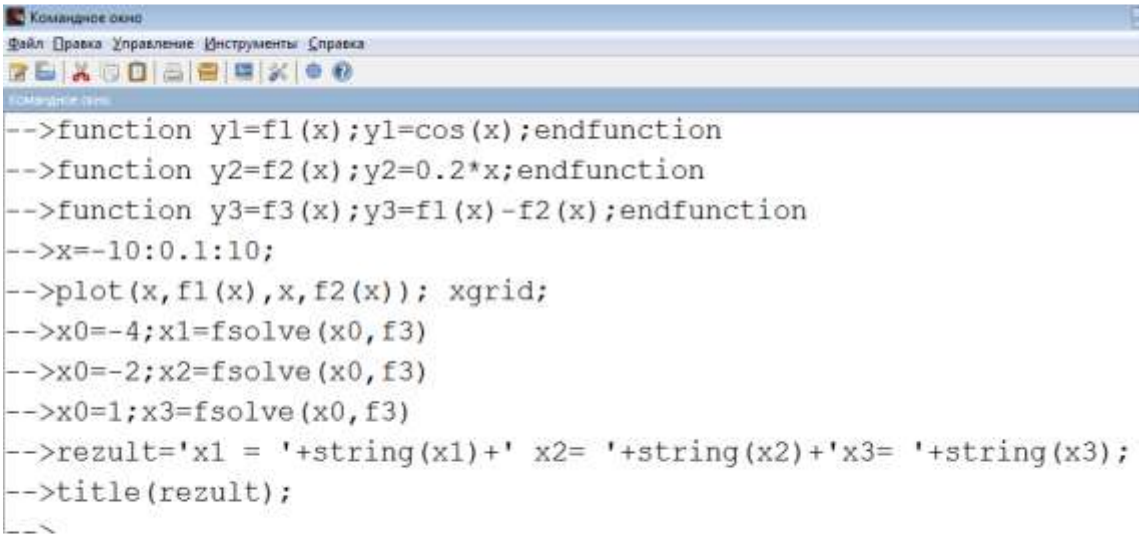


Рис. 12. Нахождение корней уравнения в системе scilab

Пример 2. Построить график функции и вывести на экран корни уравнения $\cos(x) - 0{,}2x = 0$ (рис. 13) и (рис.14) .



Командное окно
Файл Правка Управление Инструменты Справка

```
-->function y1=f1(x);y1=cos(x);endfunction
-->function y2=f2(x);y2=0.2*x;endfunction
-->function y3=f3(x);y3=f1(x)-f2(x);endfunction
-->x=-10:0.1:10;
-->plot(x,f1(x),x,f2(x)); xgrid;
-->x0=-4;x1=fsolve(x0,f3)
-->x0=-2;x2=fsolve(x0,f3)
-->x0=1;x3=fsolve(x0,f3)
-->rezult='x1 = '+string(x1)+' x2= '+string(x2)+'x3= '+string(x3);
-->title(rezult);
-->
```

Рис. 13. Нахождение корней уравнения в системе scilab

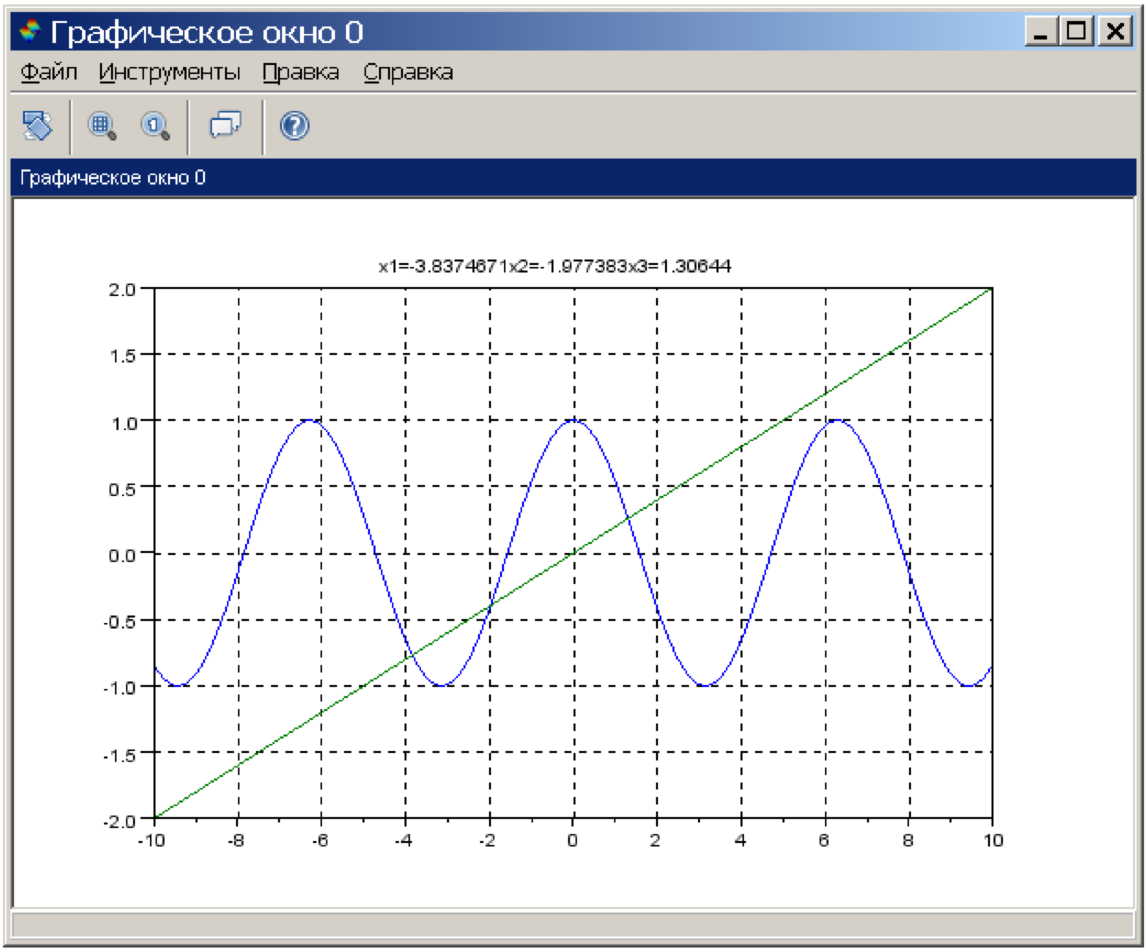


Рис. 14. Нахождение и вывод корней уравнения в системе scilab

6. Варианты заданий для самостоятельной работы:

Задание 1. Отделите корни уравнения $F(x)=0$ графически и уточните один корень методом половинного деления [5].

1. $e^x + x^2 - 3 = 0$
2. $\ln(x-1) + 2x - 1 = 0$
3. $x - \sin(x) - 4 = 0$
4. $\ln(x-3) - \frac{2}{x} = 0$
5. $\ln(x+2) - \sqrt{x} = 0$
6. $\sqrt{x+2} + x^3 - 2 = 0$
7. $x - \sin(x) - 4 = 0$
8. $\sqrt{x} - \cos(x) - 1 = 0$
9. $x^2 - \sin(x) - 1 = 0$
10. $\sqrt{x} - \frac{2}{x-3} + 1 = 0$

Задание 2. Отделите корни уравнения $F(x)=0$. Уточните два корня методом Ньютона (касательных) с точностью до $\varepsilon = 0{,}001$ [5].

1. $x^3 - 3x^2 + 9x - 8 = 0$
2. $x^3 + 2x^2 - 5x + 1 = 0$
3. $x^3 - 3x + 1 = 0$
4. $x^4 - 18x^2 + 6 = 0$
5. $x^3 - 4x^2 - 3x = 0$
6. $3x^4 + 4x^3 - 12x^2 - 5 = 0$
7. $x^3 - 0{,}2x^2 + 0{,}3x - 3 = 0$
8. $3x^4 + 8x^3 + 6x^2 - 5 = 0$
9. $x^3 + 4x + 2 = 0$

10. $x^3 - 3x^2 + 9x - 8 = 0$

7. Контрольные вопросы:

1. Что означает «решить уравнение аналитически» и «решить уравнение численно»?
2. В чём заключается задача отделения корней?
3. В чём суть графического метода отделения корней?
4. Какие свойства функции одной переменной используются для проверки правильности отделения корня и его единственности на отрезке?
5. В чём состоит основная идея метода половинного деления?
6. Может ли метод половинного деления дать точное значение корня уравнения?
7. По каким причинам метод касательных предпочтительней метода простой итерации?

II. Расчётно – графическое задание №2

“Численное интегрирование”

Цель: научиться применять численные методы для нахождения определённого интеграла.

Задание:

1) Написать программу расчёта определённого интеграла методом прямоугольника, трапеций и Симпсона (парабол) на языке программирования Pascal;
2) Выполнить расчёт определённого интеграла в среде Scilab.

1. Численное интегрирование

Методические указания

Постановка задачи.

Далеко не все интегралы можно вычислить по формуле Ньютона-Лейбница:

$$I = \int_a^b f(x)dx = F(x)\,|_a^b\; F(b) - F(a), \qquad (1)$$

где $F(x)$ - первообразная функции $f(x)$.

Например, в элементарных функциях не выражается интеграл $\int_0^b e^{-x^2} dx$.

Точное значение интеграла по формуле Ньютона-Лейбница нельзя получить, если функция $f(x)$ задаётся таблицей. В этих случаях применяются методы численного интегрирования, а в частности, метод прямоугольников, метод трапеций и метод Симпсона (парабол).

Суть численного интегрирования заключается в том, что подынтегральную функцию $f(x)$ заменяют другой, приближенной функцией так, чтобы, во-первых, она была близка к $f(x)$ и, во-вторых, интеграл от неё легко вычислялся. Например, можно заменить подынтегральную функцию интерполяционным многочленом. Также широко используются квадратурные формулы:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} A_i f(x_i), \qquad (2)$$

где x_i - некоторые точки на отрезке [a, b], называемые узлами квадратурной формулы;

A_i - числовые коэффициенты, называемые весами квадратурной формулы;

$n \geq 0$ - целое число [5].

1.1. Метод трапеций

Метод трапеций так же, как и метод прямоугольников, опирается геометрический смысл интеграла. Заменим график функции $y = f(x)$ ломаной линией (Рис. 3). Из точек $a = x_0, x_{1,} x_2, ..., x_n = b$ проведём ординаты до пересечения с кривой $y = f(x)$. Концы ординат соединим прямолинейными отрезками (рис.18)[5].

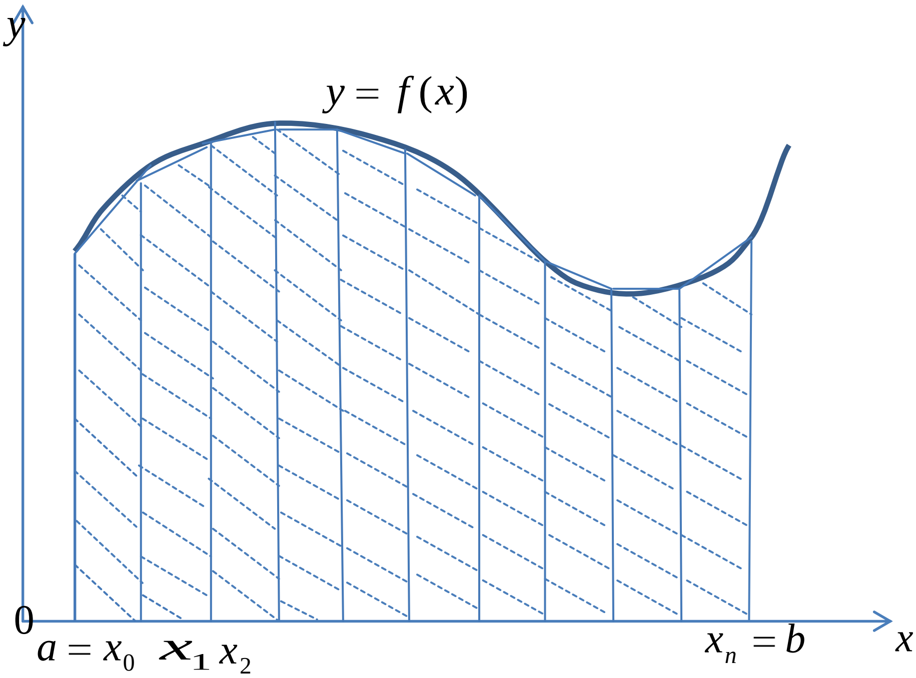


Рис. 18. Метод трапеций

Площадь криволинейной трапеции приближенно можно считать равной площади фигуры, составленной из трапеций. Так как площадь трапеции, построенной на отрезке x_i, x_{i+1} длины $h = \dfrac{b-a}{n}$, равна $\dfrac{f(x_i)+f(x_{i+1})}{2}$, то, пользуясь этой формулой для $i = 0, 2, \ldots, n-1$, получим квадратурную формулу трапеций:

$$I = \int_a^b f(x)dx \approx h\left[\frac{f(x_0)+f(x_1)}{2} + \frac{f(x_1)+f(x_2)}{2} + \ldots + \frac{f(x_{n-1})+f(x_n)}{2}\right] =$$

$$= \frac{b-a}{n}\left(\frac{f(x_0)+f(x_n)}{2} + \sum_{i=1}^{n-1} f(x_i)\right)$$

(7)

Теорема. Пусть функция f непрерывно дифференцируема на отрезке $[a, b]$. Тогда для формулы трапеций справедлива следующая оценка погрешности :

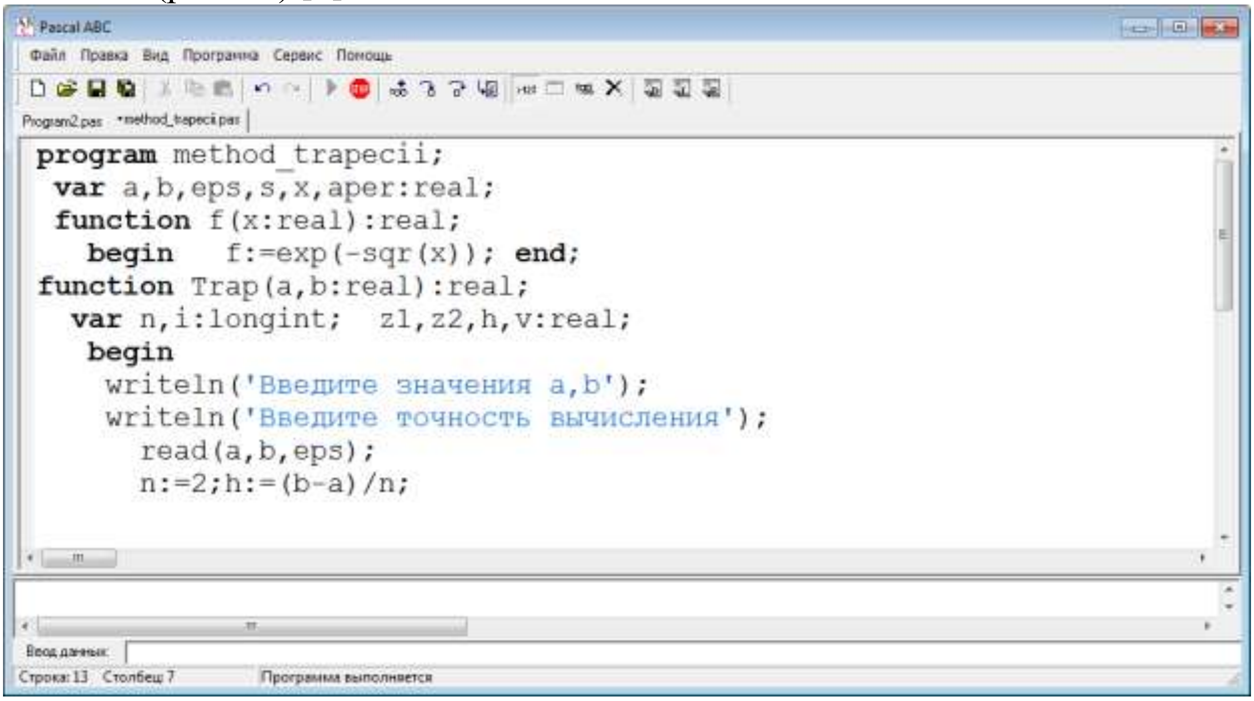
$$\left|I - I_{np}\right| \le \frac{M_2(b-a)}{12}h^2, \text{ где } x_n = b. \qquad (8)$$

Существует другой способ оценки погрешности на основе уже сделанных вычислений, поэтому этот способ называют апостериорной оценкой.

$$R_{2n} = (I_{2n} - I_n)/3 \qquad (9)$$

Формула (9) называется *правилом Рунге* практической (апостериорной) оценки погрешности методов трапеций и прямоугольников [5].

Пример.1. Написать программу вычисления определённого интеграла $\int_0^1 e^{-x^2} dx$. (рис. 19) [1]



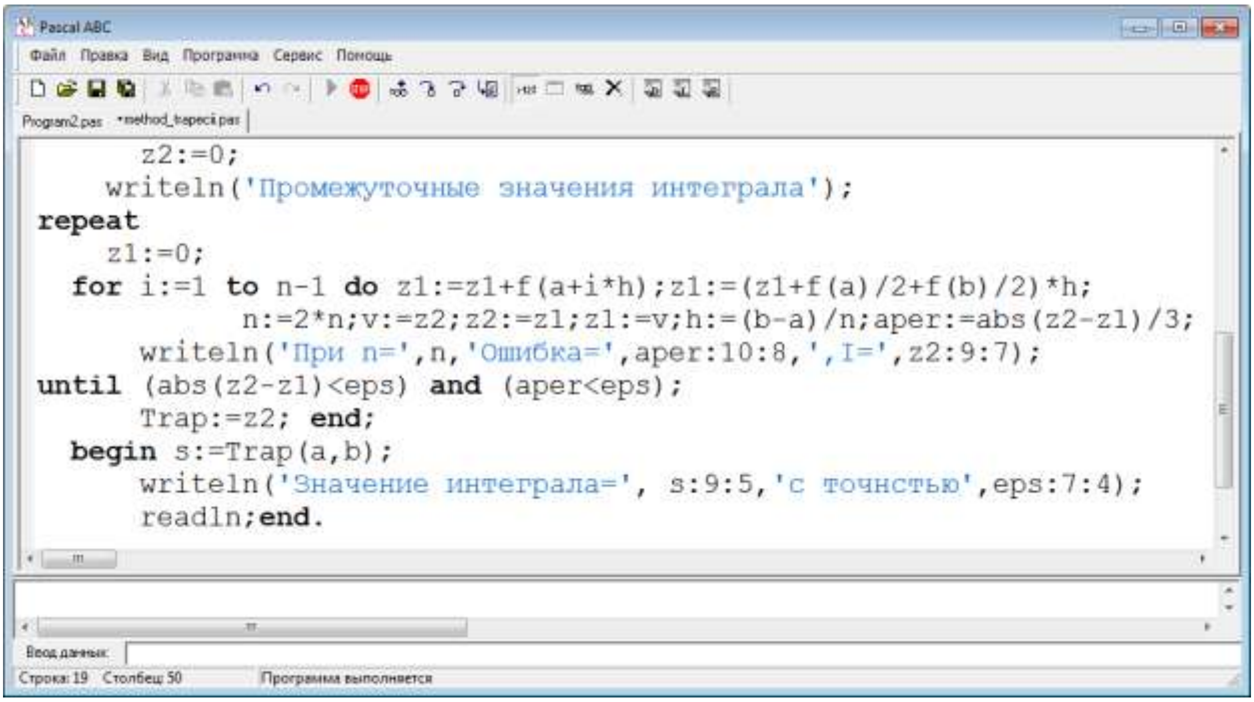
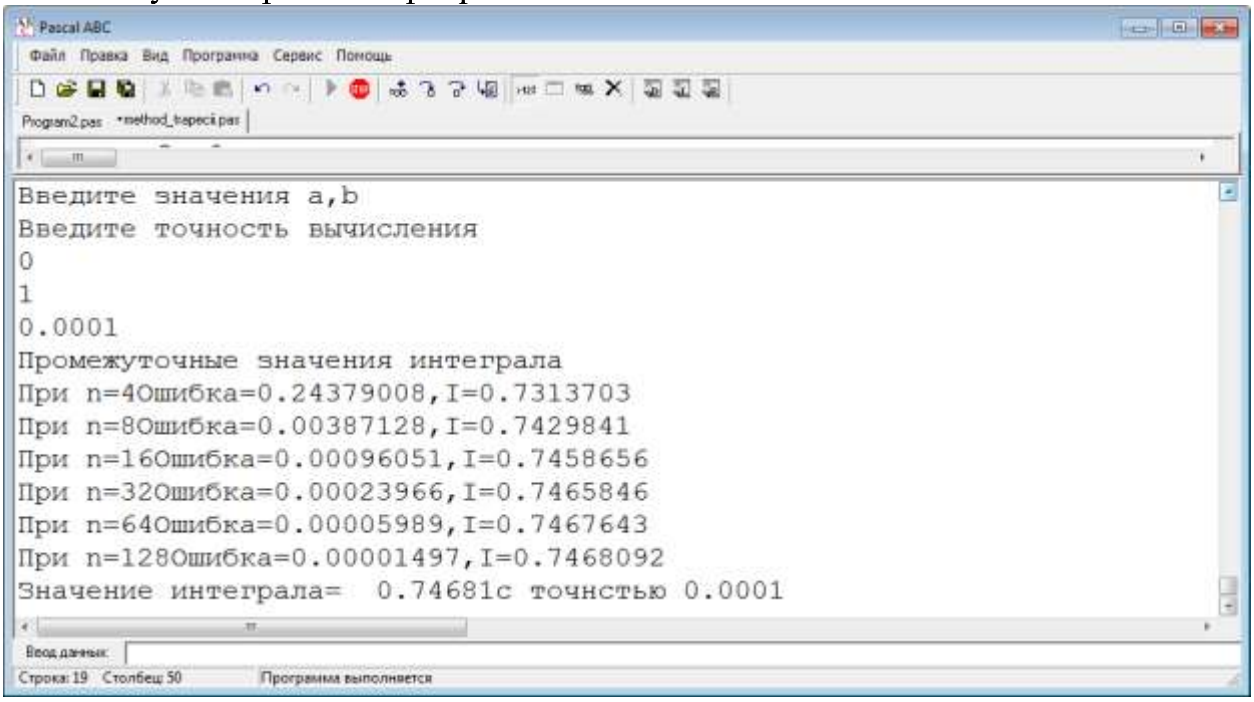


Рис. 19. Программа реализации метода трапеций

Результат работы программы:



1.2. Метод Симпсона (парабол)

При использовании метода Симпсона заменим график функции $y = f(x)$ на отрезке $[x_i, x_{i+1}]$, $i = 0, 2, ..., n-1$ параболой, проведённой через точки

$(x_i, f(x_i))$, $(x_i', f(x_i'))$, $(x_{i+1}, f(x_{i+1}))$, где x_i' - середина отрезка $[x_i, x_{i+1}]$. Эта парабола есть интерполяционный многочлен второй степени $L_2(x)$ с узлами x_i, x_i', x_{i+1}.

Квадратурная формула Симпсона имеет следующий вид:

$$I = \int_a^b f(x)dx \approx I_C \frac{h}{6}\left(f(x_0) + f(x_n) + 4\sum_{i=0}^{n-1} f(x_i') + 2\sum_{i=1}^{n-1} f(x_i) \right). \quad (9)$$

Теорема. Пусть функция f имеет на отрезке $[a, b]$ непрерывную производную четвёртого порядка $f^{(4)}(x)$. Тогда для формулы Симпсона (7) справедлива следующая оценка погрешности:

$$|I - I_C| \le \frac{M_4(b-a)}{2880} h^4 \text{, где } M_4 = \max_{[a,b]} \left| f^{4}(x) \right|. \quad (10)$$

Заметим, что если число элементарных отрезков, на которые делится отрезок $[a, b]$, чётно, т.е. $n = 2m$, то параболы можно проводить через узлы с целыми индексами и вместо элементарного отрезка x_i, x_{i+1} длины h рассматривать отрезок x_{2i}, x_{2i+2} длины $2h$. Тогда формула Симпсона примет вид:

$$I \approx \frac{h}{3}\left(f(x_0) + f(x_{2m}) + 4\sum_{i=1}^{m} f(x_{2i-1}) + 2\sum_{i=1}^{m-1} f(x_{2i}) \right). \qquad (11)$$

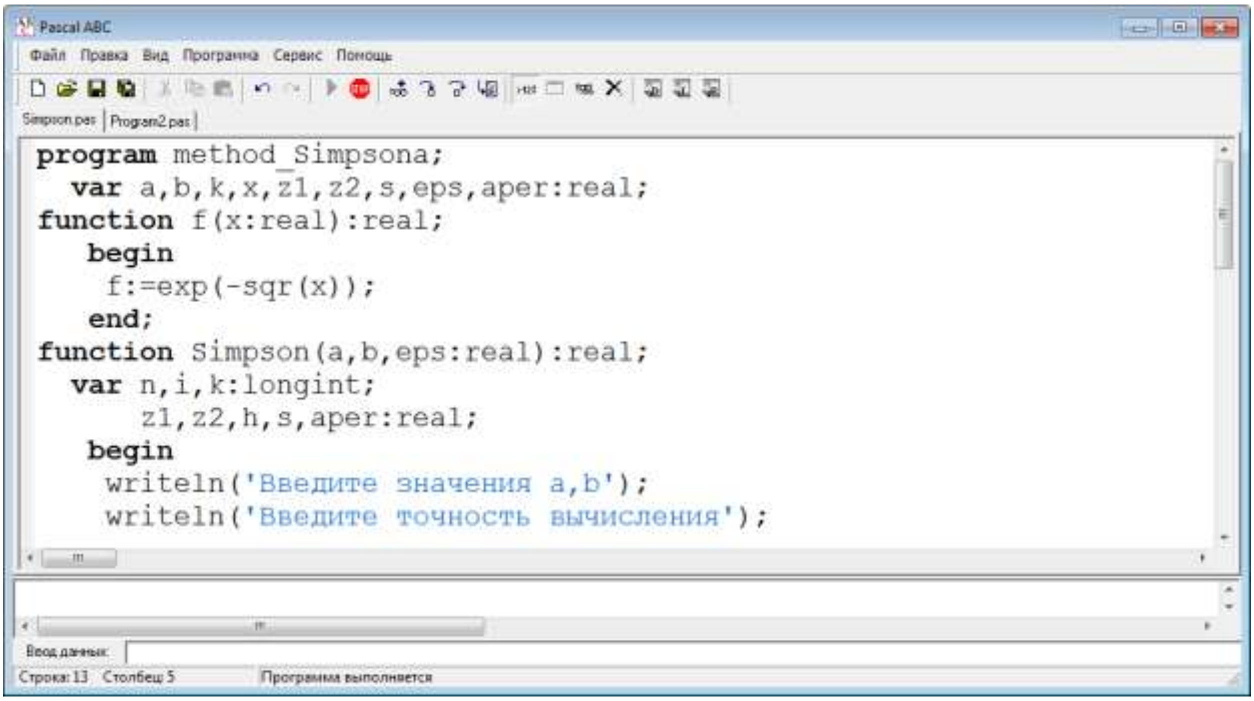
Погрешность вычисляется по формуле:

$$|I - I_C| \le \frac{M_4(b-a)}{180} h^4 .$$

$$R_{2n} = (I_{2n} - I_n)/15 \qquad (12)$$

Формула (12) называется *правилом Рунге* практической (апостериорной) оценки погрешности метода Симпсона. С помощью правила Рунге можно приближенно вычислить интеграл с заданной точностью ε. Нужно начать с некоторого значения шага h и последовательно уменьшать это значение в два раза, каждый раз вычисляя приближенное значение интеграла. Процедура прекращается, когда результаты двух последующий вычислений будут различаться меньше, чем на ε [5].

Пример 1. Написать программу вычисления определённого интеграла $\int_0^1 e^{-x^2} dx$ методом Симпсона [1] (рис. 20).



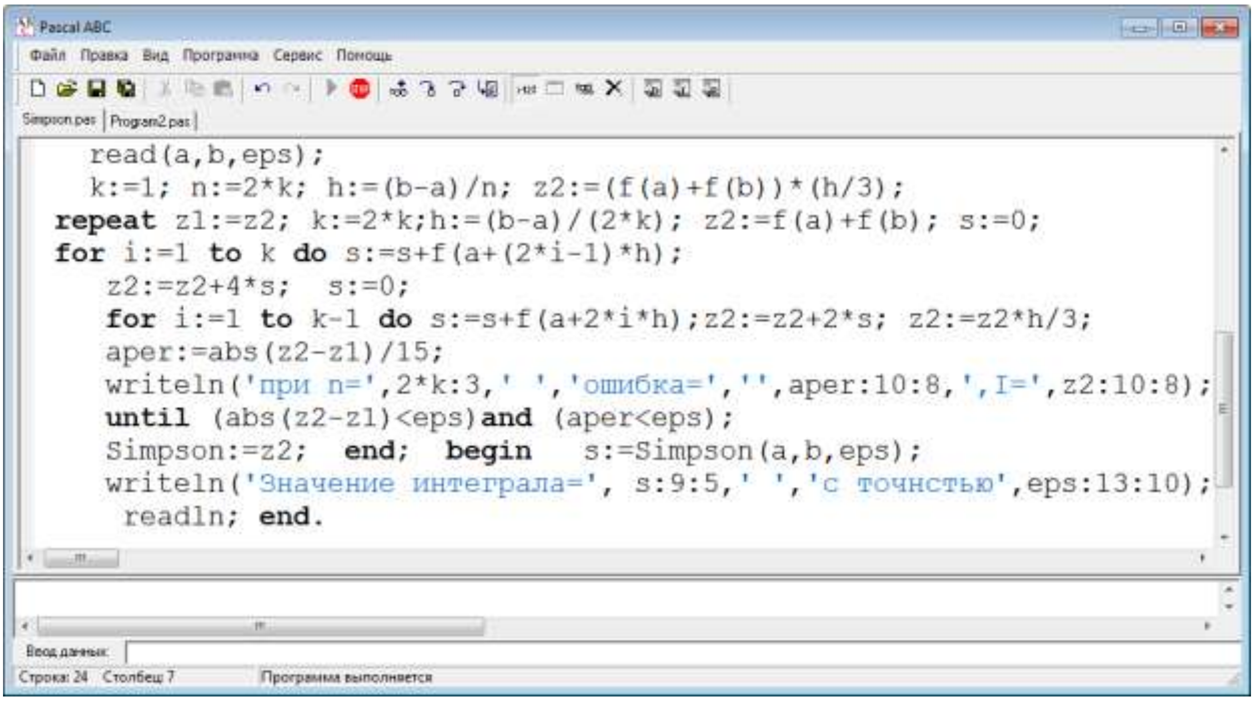
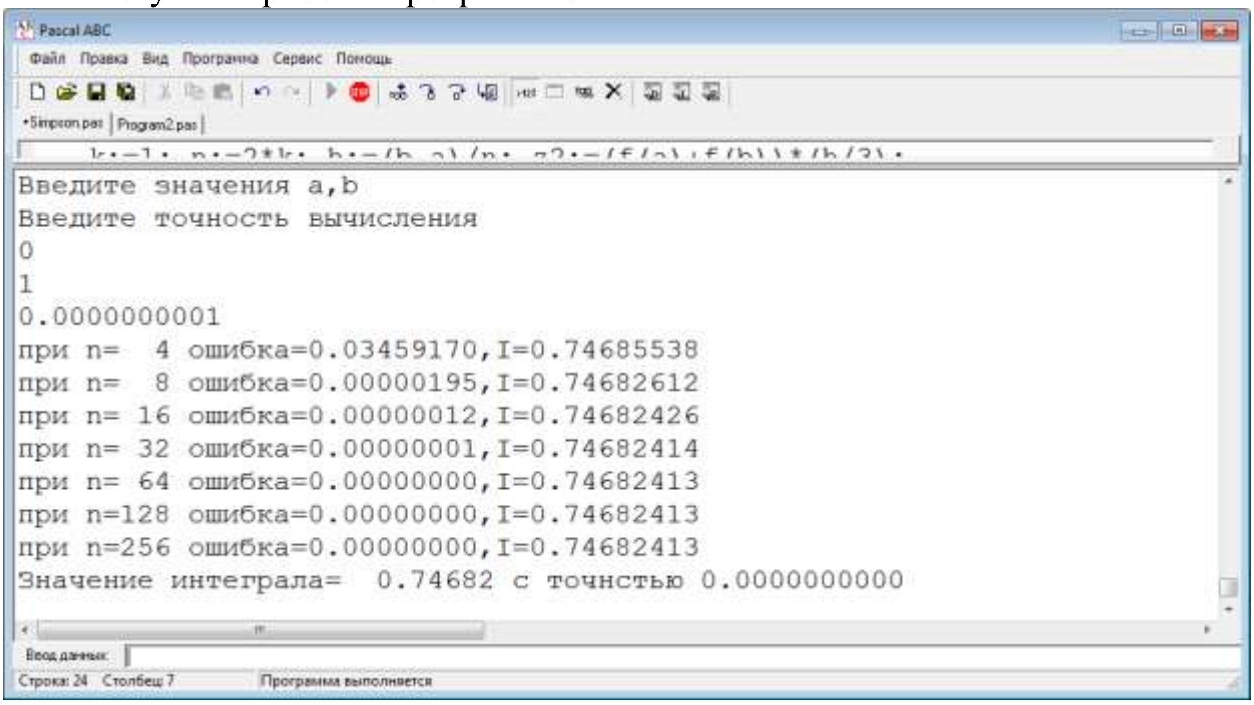


Рис. 20. Программа реализации метода Симпсона

Результат работы программы:



2. Вычисление определённого интеграла в программе SCILAB

Наиболее универсальная команда вычисления интеграла от функции $y = f(x)$ является: *[I, er]=intg(a, b, f).*

Пример 1. Вычислить определённый интеграл $\int_0^1 e^{-x^2}dx$ (рис. 21).

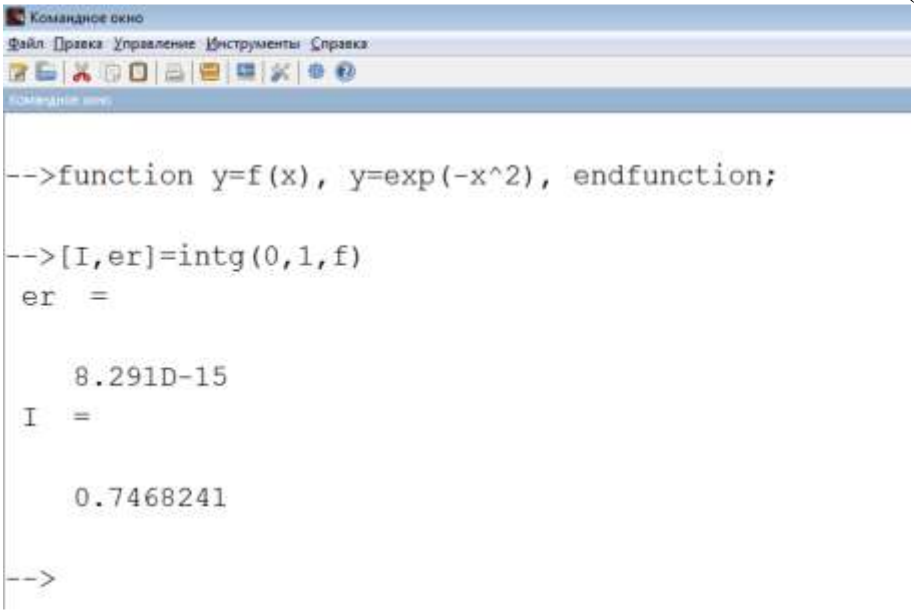


Рис. 21.Вычисление определённого интеграла в системе Scilab

Пример 2. Вычислить определённый интеграл $\int_0^1 e^{-x^2}dx$, разбив отрезок интегрирования на 5 и 20 частей (рис. 22)

.

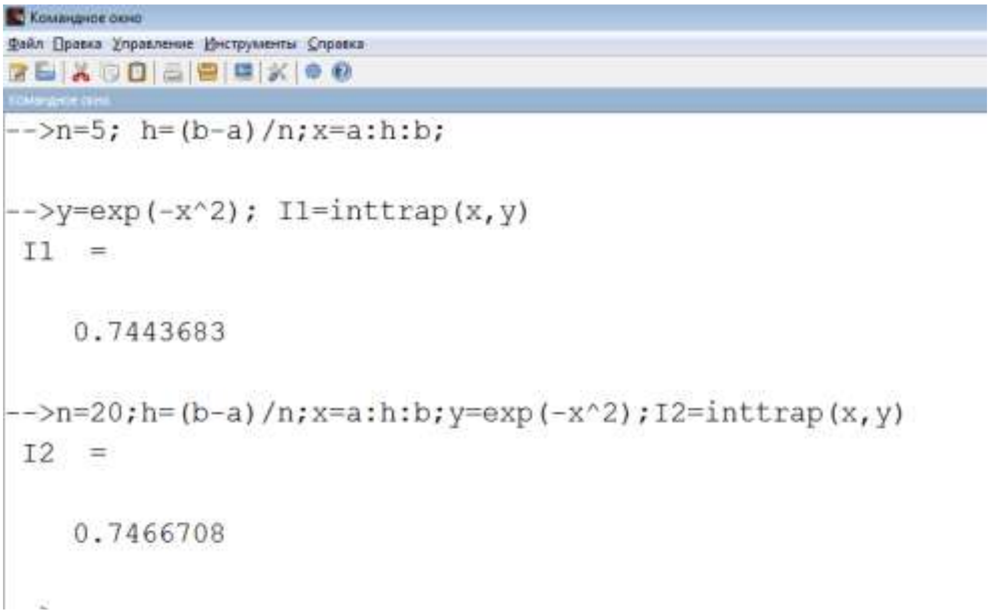


Рис. 22 .Вычисление определённого интеграла в системе Scilab

Пример 3. Вычислить определённый интеграл $\int_0^1 e^{-x^2}dx$ заданный в виде таблицы по формуле трапеций [5](рис.23).

x	e^{-x^2}	x	e^{-x^2}	x	e^{-x^2}
e^{-x^2}	1,000000	0,35	0,8847059	0,70	0,6126264
e^{-x^2}	0,9975031	0,40	0,8521438	0,75	0,5697828
e^{-x^2}	0,9900498	0,45	0,8166865	0,80	0,5272924
e^{-x^2}	0,9777512	0,50	0,7788008	0,85	0,4855369
e^{-x^2}	0,9607894	0,55	0,7389685	0,90	0,4448581
e^{-x^2}	0,9394131	0,60	0,6976763	0,95	0,4055545
e^{-x^2}	0,9139312	0,65	0,6554063	1,00	0,3678794

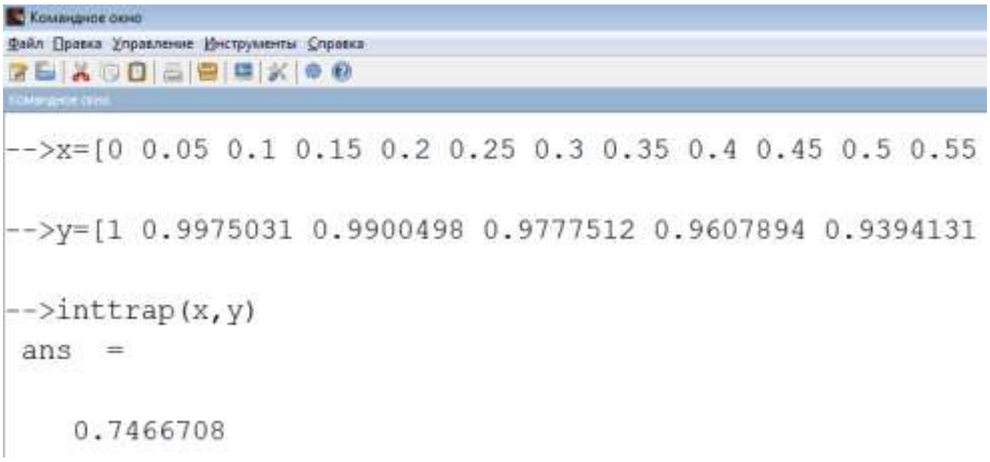


Рис. 23.Вычисление определённого интеграла, заданного в виде таблицы в системе Scilab

3. Варианты заданий для самостоятельной работы

Задание 1. Вычислить определённый интеграл [5]:

а) по формуле прямоугольников с точностью $\varepsilon = 10^{-2}$;

б) по формуле трапеций с точностью $\varepsilon = 10^{-3}$;

в) по формуле Симпсона с точностью $\varepsilon = 10^{-4}$.

1. $\int_0^1 \frac{x}{1+x}dx$	8. $\int_1^2 \ln x \cdot (1+x)dx$

2. $\int\limits_3^4 \frac{x}{\ln^2 x}dx$	9. $\int\limits_1^2 \frac{\sqrt{x}+1}{x^2}dx$
3. $\int\limits_3^4 \frac{e^x}{x^2}dx$	10. $\int\limits_0^{1,5} \frac{x}{(1+x)^3}dx$
4. $\int\limits_0^1 \frac{x}{1+x}dx$	11. $\int\limits_0^1 \frac{x}{1+x}dx$
5. $\int\limits_1^{2,5} \frac{e^x}{(1+x)^2}dx$	12. $\int\limits_1^{1,5} \frac{e^x}{1+x}dx$

4. Контрольные вопросы

1. Можно ли методами трапеций и прямоугольников получить точное значение интеграла?
2. Обязательно ли участок интегрирования разбивать при реализации метода на более мелкие участки?
3. Почему формула ньютона0лейбница может оказаться непригодной для реального вычисления определённого интеграла?
4. Как связаны задачи численного интегрирования и интерполирования?
5. В чём выражаются преимущества формулы Симпсона перед формулой трапеций?

Литература

Основная:

1. Лапчик, М. П**.**Численные методы: учеб. пособие для вузов / М. П. Лапчик, М. И. Рагулина, Е. К. Хеннер; под ред. М. П. Лапчика. - Москва: Academia, 2004. - 383, [1] с. - (Высшее профессиональное образование. Информатика и вычислительная техника). - Библиогр.: с. 381. - ISBN 5-7695-1339-X. ***Количество – 4.***

2. Волков, Е. А**.**Численные методы: учеб. пособие / Е. А. Волков. - Изд. 3-е, испр. - Санкт-Петербург; Москва; Краснодар: Лань, 2004. - 248 с. - ISBN 5-8114-0538-3. ***Количество -100***.
3. Васильков Ю. В., Василькова Н. Н. Компьютерные технологии вычислений в математическом моделировании: Учеб. пособие. – М.: Финансы и статистика, 2002. – 256 с.: ил. ***Количество -9***.
4. Информатика [Электронный ресурс]: метод. указания к самостоят. работам для студентов техн. специальностей / Федер. агентство по рыболовству, Мурман. гос. техн. ун-т, Каф. автоматики и вычисл.

техники ; сост. З. А. Масягина. - Электрон. текстовые дан. (1 файл : 748 Кб). - Мурманск: Изд-во МГТУ, 2015. - Доступ из локальной сети Мурман. гос. техн. ун-та. - Загл. с экрана. http://elib.mstu.edu.ru/2015/M_15_15.pdf.

5. Численные методы. Курс лекций: учеб. пособие для студентов высш. учеб. заведений, обучающихся по специальности 010200 "Прикладная математика и информатика" и по направлению 510200 "Прикладная математика и информатика" / В. А. Срочко. - Санкт-Петербург; Москва ; Краснодар : Лань, 2010. - 202 с. - (Учебники для вузов. Специальная литература). - Библиогр.: с. 200. - ISBN 978-5-8114-1014-9. ***Количество -5***.
6. Основы программирования в среде Free Pascal [Электронный ресурс]: метод. указания для студентов и курсантов техн. специальностей / Федер. агентство по рыболовству, Мурман. гос. техн. ун-т, Каф. автоматики и вычисл. техники; сост. Н. И. Долюк, О. В. Нефедова. - Электрон. текстовые дан. (1 файл : 440 Кб). - Мурманск: Изд-во МГТУ, 2015. - Доступ из локальной сети Мурман. гос. техн. ун-та. - Загл. с экрана. http://elib.mstu.edu.ru/2015/M_15_37.pdf.
7. Немнюгин, С. А.Turbo Pascal: учеб. для вузов / С. А. Немнюгин. - Санкт-Петербург [и др.]: Питер, 2002, 2001, 2003, 2000. - 496 с. : ил. - ISBN 5-8046-0137-7. ***Количество – 27***.
8. Паскаль. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. - Санкт-Петербург [и др.] : Питер, 2008. - 392 с. : ил. - (Серия "Учебник для вузов"). - Библиогр.: с. 382. - ISBN 978-5-94723-511-1. ***Количество – 20.***
9. Фаронов, В. В.Турбо Паскаль: начал. курс: учеб. пособие / В. В. Фаронов. - Москва: ОМД Групп, 2003. - 616 с.: ил. - На обл. загл.: Turbo Pascal 7.0. - ISBN 5-89251-054-9: 162-00; 162-00. *32.97 - Ф 24.* ***Количество – 49***.

Дополнительная:

1. Левкина, А.О. Компьютерные технологии в научно-исследовательской деятельности: учебное пособие для студентов и аспирантов социально-гуманитарного профиля / А.О. Левкина. - Москва; Берлин: Директ-Медиа, 2018. - 119 с.: ил. - Библиогр. в кн. - ISBN 978-5-4475-2826-3; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=496112 (14.12.2018).