Міністерство освіти та Науки України

Національний технічний університет “Харківський політехнічний інститут”

Кафедра «Програмна інженерія і інформаційні технології управління»

Індивідуальне завдання №2

з дисципліни

«Чисельні методи»

Виконали:

студенти групи КН-34б

Рузняєв Д.А.

Костюк I.Ю.

Перевірив:

Проф. Гужва В.О.

Харків

2016

**МЕТОД НЬЮТОНА(МЕТОД КАСАТЕЛЬНЫХ)**

*Классический метод Ньютона* или *касательных* заключается в том, что если http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image022.png — некоторое приближение к корню http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image024.png уравнения http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image026.png, то следующее приближение определяется как корень касательной к функции http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image028.png, проведенной в точке http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image022_0000.png.

Уравнение касательной к функции http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image030.pngв точке http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image032.pngимеет вид:

http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image034.png

В уравнении касательной положим http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image036.png и http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image038.png.

Тогда алгоритм последовательных вычислений в методе Ньютона состоит в следующем:

http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image040.png

Сходимость метода касательных квадратичная, порядок сходимости равен 2.

*Таким образом, сходимость метода касательных Ньютона очень быстрая.*

Без всяких изменений метод обобщается на комплексный случай.

Если корень http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image024_0000.png является корнем второй кратности и выше, то порядок сходимости падает и становится линейным.

Основная идея метода заключается в следующем: на отрезке [*a;b*] выбирается такое число *x0,* при котором *f(x0)* имеет тот же знак, что и *f''(x0),* т. е. выполняется условие *f(x0)·f''(x) > 0*. Таким образом, выбирается точка с абсциссой *x0*, в которой касательная к кривой *y=f(x)* на отрезке [*a;b*] пересекает ось *Ox*. За точку *x0* сначала удобно выбирать один из концов отрезка.

Метод Ньютона (метод касательных) применяется в том случае, если уравнение *f(x) = 0* имеет корень http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/11.png, и выполняются условия:

1) функция *y= f(x)* определена и непрерывна при http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/12.png;

2) *f(a)·f(b) < 0* (функция принимает значения разных знаков на концах отрезка [*a;b*]);

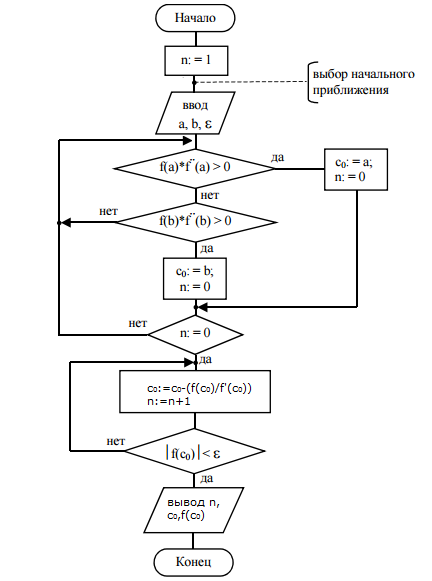
3) производные *f'(x)* и *f''(x)* сохраняют знак на отрезке [*a;b*] (т.е. функция *f(x)* либо возрастает, либо убывает на отрезке [*a;b*], сохраняя при этом направление выпуклости);

4) http://www.statistica.ru/upload/medialibrary/chisl-methods-resh-ur/13.png.

Основная идея метода заключается в следующем: на отрезке [*a;b*] выбирается такое число *x0,* при котором *f(x0)* имеет тот же знак, что и *f''(x0),* т. е. выполняется условие *f(x0)·f''(x) > 0*. Таким образом, выбирается точка с абсциссой *x0*, в которой касательная к кривой *y=f(x)* на отрезке [*a;b*] пересекает ось *Ox*. За точку *x0* сначала удобно выбирать один из концов отрезка.

-

БЛОК-СХЕМА МЕТОДА КАСАТЕЛЬНЫХ



КОД ПРОГРАММЫ МЕТОДА КАСАТЕЛЬНЫХ

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace niytona

{

class Program

{

static double f(double x)

{

return Math.Pow(x, 3) + (6 \* Math.Pow(x, 2)) + (9 \* x) + 1;

}

static double f1(double x)

{

return (3 \* Math.Pow(x, 2)) + (12 \* x) + 9;

}

static double f2(double x)

{

return (6 \* x) + 12;

}

static void Main(string[] args)

{

double aNext = -3;

double bNext = -1;

Console.WriteLine("--------------Метод касательных-----------------");

while (f2(aNext) \* f2(bNext) <= 0)

{

Console.WriteLine("F({0}) = {2}; F({1}) = {3}; F'({0}) = {4}; F'({1}) = {5}; F''({0}) = {6}; F''({1}) = {7};"

, aNext, bNext, f(aNext), f(bNext), f1(aNext), f1(bNext), f2(aNext), f2(bNext));

Console.WriteLine("Очевидно, что значения второй производной имеют разные знаки");

if (f(aNext) \* f((aNext + bNext) / 2) < 0)

{

bNext = (aNext + bNext) / 2;

Console.WriteLine("Выберем отрезок [a,b] на котором функция имеет разные знаки - [" + aNext + ";" + bNext + "]");

}

else

{

if (f((aNext + bNext) / 2) \* f(bNext) < 0)

{

aNext = (aNext + bNext) / 2;

Console.WriteLine("Выберем отрезок [a,b] на котором функция имеет разные знаки - [" + aNext + ";" + bNext + "]");

}

}

}

Console.WriteLine("F({0}) = {2}; F({1}) = {3}; F'({0}) = {4}; F'({1}) = {5}; F''({0}) = {6}; F''({1}) = {7};"

, aNext, bNext, f(aNext), f(bNext), f1(aNext), f1(bNext), f2(aNext), f2(bNext));

Console.WriteLine("Значения второй производной имеют одинаковые знаки");

//////Метод касательных------------------------------------

double a = 0;

if (f(aNext) \* f2(aNext) > 0)

{

Console.WriteLine("Значения функции и второй производной имеют одинаковый знак, найдем значение x");

int i = 1;

double xNext = aNext - (f(aNext) / f1(aNext));

Console.WriteLine("x1 = " + Math.Round(xNext, 3));

double xBefore = 0;

do

{

i++;

xBefore = xNext;

xNext = xBefore - (f(xBefore) / f1(xBefore));

Console.WriteLine("x" + i + " = " + Math.Round(xNext, 3));

}

while (Math.Round(Math.Abs(xNext - xBefore), 4) <= 0.001);

a = xNext;

}

else

{

if (f(bNext) \* f2(bNext) > 0)

{

Console.WriteLine("Значения функции и второй производной имеют одинаковый знак, найдем значение x");

int i = 1;

double xNext = bNext - (f(bNext) / f1(bNext));

Console.WriteLine("x1 = " + Math.Round(xNext, 3));

double xBefore = 0;

do

{

i++;

xBefore = xNext;

xNext = xBefore - (f(xBefore) / f1(xBefore));

Console.WriteLine("x" + i + " = " + Math.Round(xNext, 3));

}

while (Math.Round(Math.Abs(xNext - xBefore), 4) > 0.001);

a = xNext;

}

}

}

}

РУЧНОЙ ПРОСЧЕТ МЕТОДОМ КАСАТЕЛЬНЫХ

*Студента Рузняева Д.А.*

Пример №14

X^3+6X^2+9X+1=0

Нахождение приближенного корня методом касательных

x^3+6\*x^2+9\*x+1=0

f(x) = x^3+6\*x^2+9\*x+1

f ’(x) = 3\*x^2 + 12\*x+9

f ’’(x) = 6\*x +12

3\*x^2 + 12\*x+9 = 0

D=144-108=36

x1=-1 ; x2 = -3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | -∞ | -3 | -1 | ∞ |
| Sign f(x) | - | + | - | + |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | -4 | -3 | -1 | 0 |
| Sign f(x) | - | + | - | + |

[-4;-3] ; [-3;-1] ; [-1;0]

Уточним корень отрезка [-3;-1]

f(-3) = 1

f (-1) = -3

f’(-3) = 0

f’(-1) = 0

f ”(-3) = - 6

f ”(-1) = 6

Вторая производная на отрезке меняет знак, что не соответствует условиям применимости метода касательных.

Сузим интервал [-3;-1]

[-3;-2], [-2;-1]

|  |  |  |  |
| --- | --- | --- | --- |
| X | -3 | -2 | -1 |
| Sign f(x) | + | - | - |

Следовательно взять отрезок [-3;-2];

f (-3) = 1

f (-2) = -1

f’(-3) = 0

f’(-2) = -3

f ”(-3) = - 6

f ”(-2) = 0

Вторая производная на отрезке [-3;-2] меняет знак ,что не соответствует условиям применимости метода касательных.

Сузим интервал [-3;-2] до отрезка [-2.5;-2]

|  |  |  |  |
| --- | --- | --- | --- |
| X | -3 | -2,5 | -2 |
| Sign f(x) | + | + | - |

f (-2,5) = 0,375

f (-2) = - 1

f’(-2,5) = - 2,25

f’(-2) = -3

f ”(-2,5) = -3

f ”(-2) = 0

Вторая производная на отрезке [-2,5;-2] меняет знак.

Сузим интервал [-2,5;-2]

|  |  |  |  |
| --- | --- | --- | --- |
| X | -2,5 | -2,2 | -2 |
| Sign f(x) | + | - | - |

Отрезок [-2,5;-2,2]

f (-2,5) = 0,375

f (-2,2) = - 0,408

f’(-2,5) = - 2,25

f’(-2,2) = -2,88

f ”(-2,5) = -3

f ”(-2,2) =-1,2

На этом отрезке функция не меняет знак => [-2,5;-2,2] удовлетворяет условиям применимости метода касательных.

F(-2,2) \* F”(-2,2) > 0

Находим приближенный корень по формуле :

X1 = x – f(x)/f’(x)=-2,2-(-0,408/-2,88)=-2,2-0,142=-2,342

X2 = -2,342-(-0,014/-2,649)=-2,342-0,005=-2,337

X3 = -2,337-(-0,028/-2,659)=-2,337-0,012=-2,349

X4 = -2,349-(-0,005/-2,635)=-2,349+0,002=-2,347

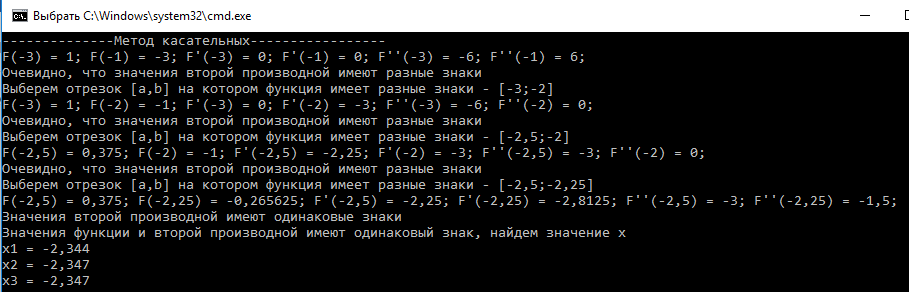
X5 = -2,347-(0/-2,639)=-2,347

|xn- xn+1 | 0,001

X\* = -2,347

ПРОСЧЕТ ПО ПРОГРАММЕ МЕТОДОМ КАСАТЕЛЬНЫХ

*Студента Рузняева Д.А*



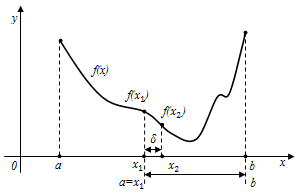
**МЕТОД ПОЛОВИННОГО ДЕЛЕНИЯ**

Нехай дано функцію Метод дихотомії, яка є унімодальною на проміжку metod_duhotomii2. На даному проміжку необхідно знайти точку мінімуму функції Метод дихотомії з заданою точністю Метод дихотомії. Для цього, використовуючи наступні формули, обчислюємо точки Метод дихотомії та Метод дихотомії:

Метод дихотомії

де Метод дихотомії.

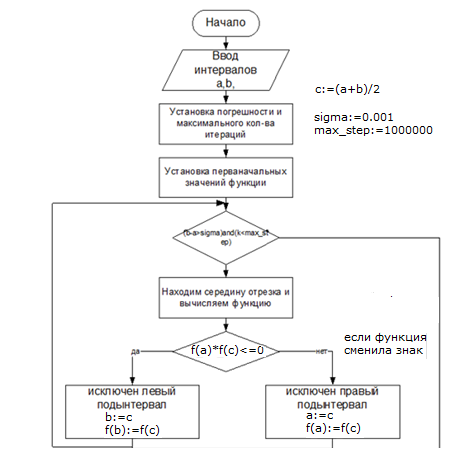
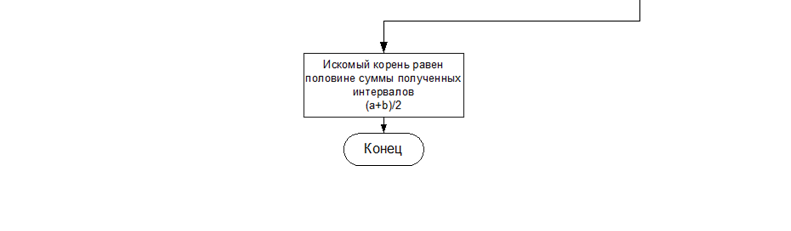
Після чого обчислюємо значення функції в знайдених точках Метод дихотомії, Метод дихотомії і порівнюємо їх між собою. Якщо Метод дихотомії то значення правої межі інтервалу невизначеності змістимо на значення точки Метод дихотомії, тобто Метод дихотомії. В протилежному випадку, якщо Метод дихотомії то змінюємо значення лівої межі інтервалу невизначеності на значення точки Метод дихотомії (Метод дихотомії).



Графічне представлення методу дихотомії

Далі, перейшовши до наступної ітерайції, знову обчислюємо точки Метод дихотомії та Метод дихотомії і таким чином визначаємо новий інтервал невизначеності. Даний проце продовжуємо до тих пір, поки довжина інтервалу невизначеності не стане меншою числа Метод дихотомії (Метод дихотомії).

БЛОК-СХЕМА МЕТОДА ПОЛОВИННОГО ДЕЛЕНИЯ

КОД ПРОГРАММЫ МЕТОДА ПОЛОВИННОГО ДЕЛЕНИЯ

using System;

using System.Collections.Generic;

namespace methodPolovDelenia

{

public abstract class AbstractMinimum

{

abstract public double F(double x);

public double Solve(double a, double b, double h)

{

double n = Math.Round(Math.Log(((b - a) / 0.001), Math.E) / Math.Log(2, Math.E));

Console.WriteLine("Необходимое количество итераций n > " + n);

int i = 2;

Console.WriteLine("----------------------Итерация 1---------------------");

double x = (a + b) / 2;

Console.WriteLine("xє[" + Math.Round(a, 5) + ";" + b + "]");

Console.WriteLine("x = " + Math.Round(x, 5));

while (Math.Abs(b - a) > h)

{

if (F(a) \* F(x) < 0)

{

Console.WriteLine("f(" + Math.Round(a, 5) + ") = " + Math.Round(F(a), 5) + " и f(" + Math.Round(x, 5) + ") = " + Math.Round(F(x), 5));

b = x;

Console.WriteLine("Выберем отрезок [a,x] = [" + Math.Round(a, 5) + ";" + Math.Round(x, 5) + "]");

}

if (F(b) \* F(x) < 0)

{

Console.WriteLine("f(" + Math.Round(b, 5) + ") = " + Math.Round(F(b), 5) + " и f(" + Math.Round(x, 5) + ") = " + Math.Round(F(x), 5));

a = x;

Console.WriteLine("Выберем отрезок [x,b] = [" + Math.Round(x, 5) + ";" + Math.Round(b, 5) + "]");

}

x = (a + b) / 2;

Console.WriteLine("----------------------Итерация " + i + "---------------------");

Console.WriteLine("x = " + Math.Round(x, 5));

i++;

}

return x;

}

}

class SpecificMinimum : AbstractMinimum

{

public override double F(double x)

{

return x \* Math.Pow(x + 1, 2) - 1; }

}

class Program

{

static void Main(string[] args)

{

SpecificMinimum sm = new SpecificMinimum();

sm.Solve(-3, -1, 0.001); } } }

РУЧНОЙ ПРОСЧЕТ МЕТОДОМ ПОЛОВИННОГО ДЕЛЕНИЯ

*Студента Рузняева Д.А.*

Пример №14

X^3+6X^2+9X+1=0

x^3+6\*x^2+9\*x+1=0

f(x) = x^3+6\*x^2+9\*x+1

f ’(x) = 3\*x^2 + 12\*x+9

f ’’(x) = 6\*x +12

3\*x^2 + 12\*x+9 = 0

D=144-108=36

x1=-1 ; x2 = -3

Вычислим количество итераций

**n>ln(|(b-a)|/eps)/ln2**

n > ln((|-1-3|)/0.001)/ln2 = 11.965

Необходимое количество итераций n > 11

----------------------Итерация 1---------------------

xє[-1;1]

x = 0

f(1) = 3 и f(0) = -1

Выберем отрезок [x,b] = [0;1]

----------------------Итерация 2---------------------

x = 0,5

f(0) = -1 и f(0,5) = 0,125

Выберем отрезок [a,x] = [0;0,5]

----------------------Итерация 3---------------------

x = 0,25

f(0,5) = 0,125 и f(0,25) = -0,60938

Выберем отрезок [x,b] = [0,25;0,5]

----------------------Итерация 4---------------------

x = 0,375

f(0,5) = 0,125 и f(0,375) = -0,29102

Выберем отрезок [x,b] = [0,375;0,5]

----------------------Итерация 5---------------------

x = 0,4375

f(0,5) = 0,125 и f(0,4375) = -0,09595

Выберем отрезок [x,b] = [0,4375;0,5]

----------------------Итерация 6---------------------

x = 0,46875

f(0,4375) = -0,09595 и f(0,46875) = 0,0112

Выберем отрезок [a,x] = [0,4375;0,46875]

----------------------Итерация 7---------------------

x = 0,45312

f(0,46875) = 0,0112 и f(0,45312) = -0,04319

Выберем отрезок [x,b] = [0,45312;0,46875]

----------------------Итерация 8---------------------

x = 0,46094

f(0,46875) = 0,0112 и f(0,46094) = -0,0162

Выберем отрезок [x,b] = [0,46094;0,46875]

----------------------Итерация 9---------------------

x = 0,46484

f(0,46875) = 0,0112 и f(0,46484) = -0,00255

Выберем отрезок [x,b] = [0,46484;0,46875]

----------------------Итерация 10---------------------

x = 0,4668

f(0,46484) = -0,00255 и f(0,4668) = 0,00431

Выберем отрезок [a,x] = [0,46484;0,4668]

----------------------Итерация 11---------------------

x = 0,46582

f(0,46484) = -0,00255 и f(0,46582) = 0,00088

Выберем отрезок [a,x] = [0,46484;0,46582]

----------------------Итерация 12---------------------

x = 0,46533

РАЗЧЕТ С ПОМОЩЬЮ ПРОГРАММЫ МЕТОДОМ ПОЛОВИННОГО ДЕЛЕНИЯ

*Cтудента Рузняева Д.А.*

