go\_forward(matrix, size);

show(matrix, "Треугольная матрица");

Result res;

res.operator=(go\_backward(matrix, size));

return res;

}

Результати виконання програми

n=3

Матриця =

0.82 0.43 -0.57 0.48

-0.35 1.12 -0.48 0.52

0.48 0.23 0.37 1.44

x1 = 0.978105

x2 = 1.49563

x3 = 1.69328

Висновок:

Результати програми співпадають з ручним рішенням

***Список використаних джерел***

1. Дубовик В.П., Юрик І.І. Вища математика. - К.: А.С.К., 2006. - 648с.
2. Зеленський К.Х. Вища математика. - К.: Університет "Україна", 2006. - Ч.2 - 212 с.
3. Коваленко І.П. Вища математика. - К.: Вища школа, 2006. - 343 с.
4. Лавренчук В.П., Готинчан Т.І., Дронь В.С., Кондур О.С. Вища математика. - Вид. 3-тє, випр. - Чернівці: Рута, 2007. - 175с.
5. Макаренко В.О. Вища математика для економістів. - К.: Знання, 2008. - 517с.
6. Овчинников П.П., Яремчук Ф.П., Михайленко В.М. Вища математика. - К.: Техніка, 2007. - 600c.

res.operator=(X);

return res;

}

Результати виконання програми

n=3

Матриця =

0.82 0.43 -0.57 0.48

-0.35 1.12 -0.48 0.52

0.48 0.23 0.37 1.44

x1 = 0.978105

x2 = 1.49563

x3 = 1.69328

Висновок:

Результати програми співпадають з ручним рішенням

***Список використаних джерел***

1. Щуп Т. Решение инженерных задач на ЕВМ. – М.: Мир, 1982. – 235с.

2. Демидович Б. П., Марон И. А. Основы вычислительной математики. – М.: Наука, 1970. – 664 с.

3. Демидович Б. П., Марон И. А., Шувалова Е. З. Численные методы анализа. – М.: Мир, 1967Волков Е. А. Численные методы. – М.: Наука, 1988.

Определитель равен

2.77118

Код програми:

#include"Gauss\_det\_alg.h"

Result Gauss\_det\_alg::do\_algorithm(vector<vector<double>>matrix, int size, vector<vector<double>>\*arg\_m)

{

vector<vector<double>> temp = matrix;

double det = 1;

for (int i = 0; i < size; i++)

{

show(matrix);

cout << "Делим строку " << i + 1 << " на " << matrix[i][i] << endl;

for (int j = i; j < size; j++)

{

temp[i][j] = matrix[i][j] / matrix[i][i];

}

cout << "det = " << det << " \* " << matrix[i][i] << " = " << det\*matrix[i][i] << endl;

det \*= matrix[i][i];

line\_sum(temp, size, i);

matrix = temp;

}

Result res;

res.operator=(det);

return res;

}

void Gauss\_det\_alg::line\_sum(vector<vector<double>>&a, int size, int index)

{

vector<double>first = a[index];

vector<double>temp = a[index];

double multi;

for (int i = index + 1; i < size; i++)

{

multi = -a[i][index];

cout << "Домножим строку " << index + 1 << " на " << multi << " и прибавим эту строку к строке " << i + 1 << endl;

for (int j = 0; j <size; j++)

{

first[j] \*= multi;

a[i][j] += first[j];

}

first = temp;

}

}

Результати виконання програми

n=4

Матриця =

1 2.14 0.42 -1.13

0.23 0.42 -1.5 0.16

0.34 -0.12 0.18 0.57

0.83 -0.17 0.62 -0.83

Det = 2.77118

Висновок:

Результати програми співпадають з ручним рішенням

***Список використаних джерел***

1. Гельфанд И. М. Лекции по линейной алгебре. — 4-е изд. — Москва : Наука, 1971. — 271 с. — ISBN 5791300158.(рос.)
2. Мальцев А. И. Основы линейной алгебры. — 3-е изд. — Новосибирск : Наука, 1970. — 400 с.(рос.)

for (int i = index + 1; i < size; i++)

{

multi = -a[i][index];

for (int j = 0; j <size\*2; j++)

{

first[j] \*= multi;

a[i][j] += first[j];

}

first = temp;

}

}

void Reverse\_alg::reverse\_line\_sum(vector<vector<double>>&a, int size, int index)

{

vector<double>first = a[index];

vector<double>temp = a[index];

double multi;

for (int i = index; i > 0; i--)

{

multi = -a[i-1][index];

for (int j = i; j <size \* 2; j++)

{

first[j] \*= multi;

a[i-1][j] += first[j];

}

first = temp;

}

}

Результати виконання програми

n=4

Матриця =

1 0.47 -0.11 0.55

0.42 1 0.35 0.17

-0.25 0.67 1 0.36

0.54 -0.32 -0.74 1

Обернена матриця =

1.97602 -1.20179 -0.0118969 -0.878222

-1.28847 2.10048 -0.486995 0.526897

1.4923 -1.72407 1.08741 -0.919142

-0.375057 0.0453026 0.655272 0.962682

Висновок:

Результати програми співпадають з ручним рішенням

***Список використаних джерел***

1. Дубовик В.П., Юрик І.І. Вища математика. - К.: А.С.К., 2006. - 648с.
2. Зеленський К.Х. Вища математика. - К.: Університет "Україна", 2006. - Ч.2 - 212 с.
3. Коваленко І.П. Вища математика. - К.: Вища школа, 2006. - 343 с.
4. Лавренчук В.П., Готинчан Т.І., Дронь В.С., Кондур О.С. Вища математика. - Вид. 3-тє, випр. - Чернівці: Рута, 2007. - 175с.

Результати виконання програми

n=4

Матриця =

0.14 -0.66 -0.18 0.24 0.89

-4.32 0.13 5.97 1.36 9.37

0.12 -0.05 0 -0.85 0.57

x1 = 2.98658

x2 = -1.80388

x3 = 3.80226

x4 = -0.142897

Висновок:

Результати програми співпадають з ручним рішенням

***Список використаних джерел***

1. Математический энциклопедический словарь. — М.: «Сов. энциклопедия », 1988. — С. 847.

Результати виконання програми

n=4

Матриця =

0.14 -0.66 -0.18 0.24 0.89

-4.32 0.13 5.97 1.36 9.37

0.12 -0.05 0 -0.85 0.57

x1 = 2.98658

x2 = -1.80388

x3 = 3.80226

x4 = -0.142897

Висновок:

Результати програми співпадають з ручним рішенням

***Список використаних джерел***

1. Щуп Т. Решение инженерных задач на ЕВМ. – М.: Мир, 1982. – 235с.

2. Демидович Б. П., Марон И. А. Основы вычислительной математики. – М.: Наука, 1970. – 664 с.

3. Демидович Б. П., Марон И. А., Шувалова Е. З. Численные методы анализа. – М.: Мир, 1967Волков Е. А. Численные методы. – М.: Наука, 1988.

Console.WriteLine("Уточним {0} корень", i + 1);

begin = roots.ElementAt(i)[0];

end = roots.ElementAt(i)[1];

do

{

Console.WriteLine("a = {0}, b = {1}", begin.ToString(), end.ToString());

x = ((begin + end) / 2);

Console.WriteLine("x = ({0} + {1})/2 = {2}", begin.ToString(), end.ToString(),x.ToString());

if (func(x)\*func(begin) >= 0)

{

Console.WriteLine("f(x)\*f(a)>=0\na = {0}", x.ToString());

begin = x;

}

else

{

Console.WriteLine("f(x)\*f(a)<0\nb = {0}", x.ToString());

end = x;

}

} while (Math.Abs(begin - end) > eps);

Console.WriteLine("a - b < eps");

answ[i] = (begin + end) / 2;

Console.WriteLine("Корень {0} = ({1} + {2}) / 2 = {3}", i + 1, begin, end, answ[i]);

}

return answ;

}

}

}

Результати виконання програми

a=-10

b=10

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

x1 = -3,532421875

x2 = -2,347265625

x3 = -0,120703125

Висновок

Результати програми співпадають з результатами ручного рішення

***Список використаних джерел***

1. Ананий В. Левитин Глава 11. Преодоление ограничений: Метод деления пополам — М.:[«Вильямс»](https://dic.academic.ru/dic.nsf/ruwiki/131224), 2006. — С. 476-480.
2. Амосов А.А., Дубинский Ю. А., Копченова Н.П. Вычислительные методы для инженеров. — М.: Мир, 1998.
3. Бахвалов Н.С., Жидков Н.П., Кобельков Г.Г. Численные методы. — 8-е изд.. — М.: Лаборатория БазовыхЗнаний, 2000.
4. Волков Е.А. Численные методы. — М.: Физматлит, 2003.

q[i].resize(n);

}

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

if (j == 0) { q[j][i] = 1; }

else

{

q[j][i] = (lambdas[i] \* q[j - 1][i]) - p[j-1];

}

}

}

return q;

}

Результати виконання програми

n=3

Матриця =

2 3 2

4 -6 -4

-1 4 7

Початковий вектор =

1

0

0

Власні значення =

x1 = -5.8503

x2 = 3.08162

x3 = 5.76868

Власні вектори =

14.0763 -19.5852 1.50896

-47.4012 -11.6735 -0.925292

15.8503 6.91838 4.23132

Висновок

Результати програми співпадають з результатами ручного рішення

***Список використаних джерел***

**1. Островский А.М.** [Решение уравнений и систем уравнений](http://pmpu.ru/vf4/references#островский). М. ИЛ, 1963, c. 137-142

2. **Уилкинсон Дж.Х.** Алгебраическая проблема собственных значений. М.Наука. 1970, с.93-94

3. **Фаддеев Д.К., Фаддеева В.Н.** Вычислительные методы линейной алгебры. М.ГИФМЛ. 1960

4. **Хорн Р.**, **Джонсон Ч.** Матричный анализ. М.Мир.1989

Ручне рішення

2 3 2

4 -6 -4

-1 4 7

Начальный вектор

1

0

0

A^1 =

2 3 2

4 -6 -4

-1 4 7

SP1 = 3

A^2 =

14 -4 6

-12 32 4

7 1 31

SP2 = 77

A^3 =

6 90 86

100 -212 -124

-13 139 227

SP3 = 21

Характерестический многочлен

(104\*x^0)+(-34\*x^1)+(-3\*x^2)+(1\*x^3)

Собственные значения

x1 = -5.8503

x2 = 3.08162

x3 = 5.76868

Коэфициенты Горнера

1 1 1

-8.8503 0.0816206 2.76868

17.7769 -33.7485 -18.0284

y0

2

4

-1

y1

14

-12

7

y2

6

100

-13

Собственные вектора

14.0763 -19.5852 1.50896

-47.4012 -11.6735 -0.925292

15.8503 6.91838 4.23132

Фрагмент коду програми

#include"Leverye\_alg.h"

Result Leverye\_alg::do\_algorithm(vector<vector<double>>matrix, int size, vector<vector<double>>\*arg)

{

vector<double>S(size);

S[0] = Algorithms::SP(matrix);

cout << "A^1 = ";

show(matrix);

cout << "SP" << 1 << " = " << S[0] << endl << endl;

vector <vector<double>>temp = matrix;

for (int i = 1; i < size; i++)

{

temp = Algorithms::matrix\_multi(temp, matrix);

cout << "A^" << i+1<< " = ";

show(temp);

S[i] = Algorithms::SP(temp);

cout << "SP" << i+1 << " = " << S[i] << endl << endl;

}

vector<double>p(size);

p[0] = -S[0];

double sum = 0;

for (double i = 1; i < size; i++)

{

sum = S[i];

for (int j = 0, k = i - 1; j < i; j++, k--)

{

sum += p[j] \* S[k];

}

p[i] = -sum / (i + 1);

}

vector<double>rev = p;

for (int i = 0; i < size; i++)

{

rev[i] = p[size - 1 - i];

p[size - 1 - i]\*=-1;

}

rev.push\_back(1);

Polynomal P(rev);

cout << "Характерестический многочлен" << endl;

cout << P << endl;

vector<double>lambdas = \*Half\_div\_alg::solve(P.get\_function());

cout << "Собственные значения" << endl;

show\_row(lambdas);

Result res;

auto q = gorner\_coefs(lambdas, p);

show(q, "Коэфициенты Горнера");

vector<vector<double>>X(size);

for (int i = 0; i < size; i++)

{

X[i].resize(size);

}

vector<vector<double>>Y = \*arg;

vector<vector<double>>\*ys = new vector<vector<double>>[size];

ys[0] = matrix\_multi(matrix, Y);

Algorithms::show(ys[0], "y0");

for (int i = 1; i < size; i++)

{

cout << "y" << i;

ys[i] = matrix\_multi(matrix, ys[i - 1]);

Algorithms::show(ys[i]);

}

for (int k = 0; k < size; k++)

{

for (int i = size - 1, b = 0; i >= 0; i--, b++)

{

for (int j = 0, l = size - 1; j < size - 1; j++, l--)

{

X[k][b] += ys[l - 1][b][0] \* q[j][k];

}

}

}

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

X[i][j] += q[size - 1][i] \* Y[j][0];

}

}

transpose(X);

res = X;

return res;

}

vector<vector<double>>Leverye\_alg::gorner\_coefs(vector<double>lambdas, vector <double>p)

{

int n = lambdas.size();

vector<vector<double>>q(n);

for (int i = 0; i < n; i++)

{

q[i].resize(n);

}

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

if (j == 0) { q[j][i] = 1; }

else

{

q[j][i] = (lambdas[i] \* q[j - 1][i]) - p[j - 1];

}

}

}

return q;

}

Результати виконання програми

n=3

Матриця =

2 3 2

4 -6 -4

-1 4 7

Початковий вектор =

1

0

0

Власні значення =

x1 = -5.8503

x2 = 3.08162

x3 = 5.76868

Власні вектори =

14.0763 -19.5852 1.50896

-47.4012 -11.6735 -0.925292

15.8503 6.91838 4.23132

Висновок

Результати програми співпадають з результатами ручного рішення

***Список використаних джерел***

**1. Островский А.М.** [Решение уравнений и систем уравнений](http://pmpu.ru/vf4/references#островский). М. ИЛ, 1963, c. 137-142

2. **Уилкинсон Дж.Х.** Алгебраическая проблема собственных значений. М.Наука. 1970, с.93-94

3. **Фаддеев Д.К., Фаддеева В.Н.** Вычислительные методы линейной алгебры. М.ГИФМЛ. 1960

4. **Хорн Р.**, **Джонсон Ч.** Матричный анализ. М.Мир.1989

Ручне рішення

2 3 2

4 -6 -4

-1 4 7

Начальный вектор

1

0

0

B1

-1 3 2

4 -9 -4

-1 4 4

A2

8 -13 0

-24 50 16

10 -11 10

q2 = SP(A2)/2 = 34

B2

-26 -13 0

-24 16 16

10 -11 -24

A3

-104 0 0

0 -104 0

0 0 -104

q3 = SP(A3)/3 = -104

Характерестический многочлен

(104\*x^0)+(-34\*x^1)+(-3\*x^2)+(1\*x^3)

Собственные значения

x1 = -5.8503

x2 = 3.08162

x3 = 5.76868

Коэфициенты Горнера

1 1 1

-8.8503 0.0816206 2.76868

17.7769 -33.7485 -18.0284

y0

2

4

-1

y1

14

-12

7

y2

6

100

-13

Собственные вектора

14.0763 -19.5852 1.50896

-47.4012 -11.6735 -0.925292

15.8503 6.91838 4.23132

Фрагмент коду програми

#include"Leverye\_Faddeev\_alg.h"

Result Leverye\_Faddeev\_alg::do\_algorithm(vector<vector<double>>matrix, int size, vector<vector<double>>\*arg)

{

vector<vector<double>>E(size);

vector<vector<double>>y(size);

vector<vector<double>>x(size);

vector<vector<double>>b(size - 1);

for (int i = 0; i < size - 1; i++)

{

b[i].resize(size);

}

for (int i = 0; i < size; i++)

{

y[i].resize(size);

x[i].resize(size);

E[i].resize(size);

for (int j = 0; j < size; j++)

{

if (i == j)E[i][j] = 1;

else E[i][j] = 0;

}

}

vector<vector<double>>temp = matrix;

vector<vector<double>>B = matrix;

vector<double>q(size);

q[0] = SP(matrix);

string bs = "B";

string as = "A";

for (int i = 1; i < size; i++)

{

B = matrix\_minus\_matrix(temp, matrix\_multi\_number(E, q[i - 1]));

show(B, bs+to\_string(i));

b[i-1] = B[0];

temp = matrix\_multi(matrix, B);

show(temp,as+to\_string(i+1));

q[i] = SP(temp) / (i + 1);

cout << "q" << to\_string(i + 1) << " = SP(A"+to\_string(i+1)+")/"<<(i+1)<<" = " <<q[i] << endl;

}

vector<double>rev = q;

for(int i =0,k=size-1;i<size;i++,k--)

{

rev[i] = -q[k];

}

rev.push\_back(1);

Polynomal P(rev);

cout << "Характерестический многочлен" << endl;

cout << P << endl;

vector<double>lambdas = \*Half\_div\_alg::solve(P.get\_function());

cout << "Собственные значения" << endl;

show\_row(lambdas);

Result res;

auto qq = gorner\_coefs(lambdas, q);

show(qq, "Коэфициенты Горнера");

vector<vector<double>>X(size);

for (int i = 0; i < size; i++)

{

X[i].resize(size);

}

vector<vector<double>>Y = \*arg;

vector<vector<double>>\*ys = new vector<vector<double>>[size];

ys[0] = matrix\_multi(matrix, Y);

Algorithms::show(ys[0], "y0");

for (int i = 1; i < size; i++)

{

cout << "y" << i;

ys[i] = matrix\_multi(matrix, ys[i - 1]);

Algorithms::show(ys[i]);

}

for (int k = 0; k < size; k++)

{

for (int i = size - 1, b = 0; i >= 0; i--, b++)

{

for (int j = 0, l = size - 1; j < size - 1; j++, l--)

{

X[k][b] += ys[l - 1][b][0] \* qq[j][k];

}

}

}

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

X[i][j] += qq[size - 1][i] \* Y[j][0];

}

}

transpose(X);

res = X;

return res;

}

vector<vector<double>>Leverye\_Faddeev\_alg::gorner\_coefs(vector<double>lambdas, vector <double>p)

{

int n = lambdas.size();

vector<vector<double>>q(n);

for (int i = 0; i < n; i++)

{

q[i].resize(n);

}

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

if (j == 0) { q[j][i] = 1; }

else

{

q[j][i] = (lambdas[i] \* q[j - 1][i]) - p[j - 1];

}

}

}

return q;

}

Результати виконання програми

n=3

Матриця =

2 3 2

4 -6 -4

-1 4 7

Початковий вектор =

1

0

0

Власні значення =

x1 = -5.8503

x2 = 3.08162

x3 = 5.76868

Власні вектори =

14.0763 -19.5852 1.50896

-47.4012 -11.6735 -0.925292

15.8503 6.91838 4.23132

Висновок

Результати програми співпадають з результатами ручного рішення

***Список використаних джерел***

**1. Островский А.М.** [Решение уравнений и систем уравнений](http://pmpu.ru/vf4/references#островский). М. ИЛ, 1963, c. 137-142

2. **Уилкинсон Дж.Х.** Алгебраическая проблема собственных значений. М.Наука. 1970, с.93-94

3. **Фаддеев Д.К., Фаддеева В.Н.** Вычислительные методы линейной алгебры. М.ГИФМЛ. 1960

4. **Хорн Р.**, **Джонсон Ч.** Матричный анализ. М.Мир.1989

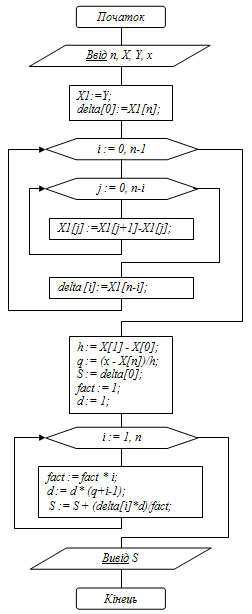


Рисунок 1 – Блок схема другої інтерполяційної формули методом Ньютона

Ручне рішення

|  |  |
| --- | --- |
| 7 | 2 |
| 10 | 4 |
| 13 | 14 |
| 16 | 20 |
| 19 | 10 |

2 10 6 -10

8 -4 -16

-12 -12

0

h = x(i+1) - x(i) = 3

q = (x - xn)/h = (x - 19)/3

f(x) = 10 + ((q + 0) \* -10/1!) + ((q + 0)(q + 1) \* -16/2!) + ((q + 0)(q + 1)(q + 2) \* -12/3!) + ((q + 0)(q + 1)(q + 2)(q + 3) \* 0/4!)

x=11

f(11) = 7.03704

Фрагмент коду програми:

#include"Newton\_interpolation\_polynomal.h"

Newton\_interpolation\_polynomal::Newton\_interpolation\_polynomal(vector<vector<double>>matrix, int size)

{

deltaY.resize(size-1);

int delta0 = size-1;

for (int i = delta0,k=0; k<size-1; i--,k++)

{

deltaY[k].resize(i);

}

for (int i = 0; i < delta0; i++)

{

for (int j = 0; j < deltaY.size()-i; j++)

{

if (i == 0)

deltaY[i][j] = matrix[j + 1][1] - matrix[j][1];

else

deltaY[i][j] = deltaY[i - 1][j + 1] - deltaY[i - 1][j];

}

}

ma = matrix;

Algorithms::show(deltaY);

}

double Newton\_interpolation\_polynomal::do\_algorithm(double x)

{

double y = ma[ma.size() - 1][1];

double qi = 1;

double h = ma[1][0] - ma[0][0];

int size = ma.size();

for (int i = 0, k = size - 1; i < size-1; i++, k--)

{

for (int j = 0; j <= i; j++)

{

qi \*= ((x - ma[size - 1][0]) / h) + j;

}

y += (qi / Algorithms::fact(i+1))\*deltaY[i][k-1];

qi = 1;

}

return y;

}

string Newton\_interpolation\_polynomal::get\_polynomal()

{

if(polynomal!="")

return polynomal;

else

{

polynomal += "h = x(i+1) - x(i) = " + toString(ma[1][0] - ma[0][0])+"\n";

polynomal += "q = (x - xn)/h = (x - " + toString(ma[ma.size() - 1][0]) + ")/" + toString(ma[1][0] - ma[0][0])+"\n";

polynomal += "f(x) = ";

polynomal += toString(ma[ma.size() - 1][1]);

polynomal += " + ";

double y = ma[ma.size() - 1][1];

double h = ma[1][0] - ma[0][0];

int size = ma.size();

for (int i = 0, k = size - 1; i < size - 1; i++, k--)

{

polynomal += "(";

for (int j = 0; j <= i; j++)

{

polynomal += "(q + " + toString(j) + ")";

}

polynomal += " \* " + toString(deltaY[i][k - 1]);

polynomal += "/" + toString(i+1) + "!) + ";

}

polynomal.erase(polynomal.end() - 2);

return polynomal;

}

}

function<double(double)> der(function<double(double)>& func, double x, double eps)

{

return [&func, &eps](double x) {double a = func(x + eps);

double b = func(x);return ((a - b) / eps); };

}

double Newton\_interpolation\_polynomal::R(double x,function<double(double)>&func)

{

int n = deltaY.size();

int size = ma.size();

double avg=0;

for (int i = 0; i < size; i++)

{

avg += ma[i][0];

}

avg = avg / size;

double h = ma[1][0] - ma[0][0];

double qi = 1;

double f = 1;

for (int j = 0; j < n; j++)

{

qi \*= (((x - ma[size - 1][0]) / h) + j);

f \*= j + 1;

}

qi = qi\*pow(h, n)/f;

function<double(double)>\*derevate = new function<double(double)>[n];

derevate[0] = der(func, avg, 0.000001);

for (int i = 1; i < n; i++)

{

derevate[i] = der(derevate[i-1], avg, 0.000001);

}

return qi\*derevate[n-1](avg);

}

Результат виконання програми

n=5

x,y =

7 2

10 4

13 14

16 20

19 10

xi = 11

f(xi) = 7.03704

Висновок:

Результати програми і ручного рішення збігаються

***Список використаних джерел***

1) Гончаров В. Л., Теория интерполирования и приближения функций, 2 изд., М., 1954;

2) Самарский А.А. Гулин А.В. Численные методы 1989г.

3) Костомаров Д.П., Фаворский А.П. Вводные лекции по численным методам

Ручне рішення

|  |  |
| --- | --- |
| 7 | 2 |
| 10 | 4 |
| 13 | 14 |
| 16 | 20 |
| 19 | 10 |

2 10 6 -10

8 -4 -16

-12 -12

0

h = x(i+1) - x(i) = 3

q = (x - x0)/h = (x - 13)/3

f(x) = 14 + ((q + 0) \* 6/1!) + ((q + 1)(q + 0) \* -4/2!) + ((q + 1)(q + 0)(q - 1) \* -12/3!) + ((q + 2)(q + 1)(q + 0)(q - 1) \* 0/4!)

х = 11

f(11) = 7.03704

Фрагмент коду програми

#include"Gauss\_interpolation\_polynomal.h"

Gauss\_interpolation\_polynomal::Gauss\_interpolation\_polynomal(vector<vector<double>>matrix, int size)

{

deltaY.resize(size - 1);

int delta0 = size - 1;

for (int i = delta0, k = 0; k<size - 1; i--, k++)

{

deltaY[k].resize(i);

}

for (int i = 0; i < delta0; i++)

{

for (int j = 0; j < deltaY.size() - i; j++)

{

if (i == 0)

deltaY[i][j] = matrix[j + 1][1] - matrix[j][1];

else

deltaY[i][j] = deltaY[i - 1][j + 1] - deltaY[i - 1][j];

}

}

ma = matrix;

Algorithms::show(deltaY);

}

double Gauss\_interpolation\_polynomal::do\_algorithm(double x)

{

int begin\_index;

if (ma.size() % 2 == 0)

begin\_index = (int)floor(ma.size() / 2) - 1;

else

begin\_index = (int)floor(ma.size() / 2);

double y = ma[begin\_index][1];

int f = 1;

double h = ma[1][0] - ma[0][0];

double qi = (x - ma[begin\_index][0])/h;

int size = ma.size();

double temp = 0;

for (int i = 0; i < size-1; i++)

{

y += build\_qi(i, qi)\*deltaY[i][begin\_index - ceil((float)i / 2.)] /Algorithms::fact(i+1);

}

return y;

}

double Gauss\_interpolation\_polynomal::build\_qi(int index, double q)

{

double res = 1;

for (int i = 0; i <= index; i++)

{

res \*= q + (int)floor((float)index / 2.) - i;

}

return res;

}

string Gauss\_interpolation\_polynomal::get\_polynomal()

{

if (polynomal != "")

return polynomal;

else

{

int begin\_index;

if (ma.size() % 2 == 0)

begin\_index = (int)floor(ma.size() / 2) - 1;

else

begin\_index = (int)floor(ma.size() / 2);

polynomal += "h = x(i+1) - x(i) = " + toString(ma[1][0] - ma[0][0]) + "\n";

polynomal += "q = (x - x0)/h = (x - " + toString(ma[begin\_index][0]) + ")/" + toString(ma[1][0] - ma[0][0]) + "\n";

polynomal += "f(x) = ";

polynomal += toString(ma[begin\_index][1]);

polynomal += " + ";

double y = ma[ma.size() - 1][1];

double h = ma[1][0] - ma[0][0];

int size = ma.size();

for (int i = 0, k = size - 1; i < size - 1; i++, k--)

{

polynomal += "(";

for (int j = 0; j <= i; j++)

{

if(((int)ceil((float)i / 2.) - j)>=0)

polynomal += "(q + " + toString((int)ceil((float)i / 2.) - j)+")";

else

polynomal += "(q - " + toString(-((int)ceil((float)i / 2.) - j)) + ")";

}

polynomal += " \* " + toString(deltaY[i][begin\_index - ceil((float)i / 2.)]);

polynomal += "/" + toString(i + 1) + "!) + ";

}

polynomal.erase(polynomal.end() - 2);

return polynomal;

}

}

Результат виконання програми

n=5

x,y =

7 2

10 4

13 14

16 20

19 10

xi = 11

f(xi) = 7.03704

Висновок:

Результати програми і ручного рішення збігаються

***Список використаних джерел***

1) Гончаров В. Л., Теория интерполирования и приближения функций, 2 изд., М., 1954;

2) Самарский А.А. Гулин А.В. Численные методы 1989г.

3) Костомаров Д.П., Фаворский А.П. Вводные лекции по численным методам

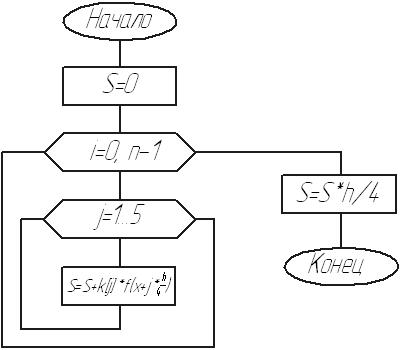


Рисунок 1 – Блок схема алгоритму Ньютона – Котеса для n=4

Ручне рішення

f(x) = (3.5\*x^2+x)/(x^4+1)

Вычисление интеграла на интервале [0; 1,2] методом Ньютона-Котеса с коэфициентами 10 порядка

Разобьем интервал [0; 1,2] на 1 частей

H = (1,2 - 0)/(1) = 1,2

h = (1,2 - 0)/(10\*1) = 0,12

H[0] = 0,0268341483624996

H[1] = 0,177535941422722

H[2] = -0,0810435706215784

H[3] = 0,45494628826882

H[4] = -0,435155122638504

H[5] = 0,713764630412152

H[6] = -0,435155122638499

H[7] = 0,454946288268821

H[8] = -0,0810435706215813

H[9] = 0,177535941422721

H[10] = 0,0268341483625003

Рассмотрим промежуток 0 интервал которого = [0; 1,2]

Резделим его на 10 частей

xi = 0 + (0 + 0 \* 10) \* 0,12 = 0

I = 0 + H[0] \* f(0) \* 0,12 = 0

xi = 0 + (1 + 0 \* 10) \* 0,12 = 0,12

I = 0 + H[1] \* f(0,12) \* 0,12 = 0,00362950231661145

xi = 0 + (2 + 0 \* 10) \* 0,12 = 0,24

I = 0,00362950231661145 + H[2] \* f(0,24) \* 0,12 = -0,000650957040929158

xi = 0 + (3 + 0 \* 10) \* 0,12 = 0,36

I = -0,000650957040929158 + H[3] \* f(0,36) \* 0,12 = 0,0430326422522285

xi = 0 + (4 + 0 \* 10) \* 0,12 = 0,48

I = 0,0430326422522285 + H[4] \* f(0,48) \* 0,12 = -0,0207552566859349

xi = 0 + (5 + 0 \* 10) \* 0,12 = 0,6

I = -0,0207552566859349 + H[5] \* f(0,6) \* 0,12 = 0,12027897269437

xi = 0 + (6 + 0 \* 10) \* 0,12 = 0,72

I = 0,12027897269437 + H[6] \* f(0,72) \* 0,12 = 0,0159683910582278

xi = 0 + (7 + 0 \* 10) \* 0,12 = 0,84

I = 0,0159683910582278 + H[7] \* f(0,84) \* 0,12 = 0,136594790029191

xi = 0 + (8 + 0 \* 10) \* 0,12 = 0,96

I = 0,136594790029191 + H[8] \* f(0,96) \* 0,12 = 0,114583817513989

xi = 0 + (9 + 0 \* 10) \* 0,12 = 1,08

I = 0,114583817513989 + H[9] \* f(1,08) \* 0,12 = 0,161176446051459

xi = 0 + (10 + 0 \* 10) \* 0,12 = 1,2

I = 0,161176446051459 + H[10] \* f(1,2) \* 0,12 = 0,167713864809215

I = 0,167713864809215 \* 10 = 1,67713864809215

Фрагмент коду програми

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using org.mariuszgromada.math.mxparser;

namespace seven\_task

{

public static class NewtonCotesAlg

{

private static double[] Hi = {

0.026834148362499605,

0.17753594142272203,

-0.081043570621578381,

0.45494628826882,

-0.43515512263850387,

0.71376463041215155,

-0.43515512263849887,

0.45494628826882105,

-0.081043570621581323,

0.17753594142272075,

0.026834148362500271

};

public static double solve(Expression e, double a, double b, int N)

{

const int n = 10;

double sum = 0;

double k = a;

double h = Math.Abs(((b - a) / (N\*n)));

double H = Math.Abs((b - a) / N);

double temp;

Console.WriteLine("f(x) = {0}", e.getExpressionString());

Console.WriteLine("Вычисление интеграла на интервале [{0}; {1}] методом Ньютона-Котеса с коэфициентами {2} порядка", a, b, n);

Console.WriteLine("Разобьем интервал [{0}; {1}] на {2} частей", a, b, N);

Console.WriteLine("H = ({0} - {1})/({2}) = {3}", b, a, N, H);

Console.WriteLine("h = ({0} - {1})/({2}\*{3}) = {4}", b, a, n, N, h);

for (int i = 0; i <= n; i++)

{

Console.WriteLine("H[{0}] = {1}", i, Hi[i]);

}

for (int j = 0; j < N; j++)

{

Console.WriteLine("Рассмотрим промежуток {2} интервал которого = [{0}; {1}]", a + j \* H, a + (j + 1) \* H, j);

Console.WriteLine("Резделим его на {0} частей", n);

for (int i = 0; i <= n; i++)

{

k = a + (i + j \* n) \* h;

Console.WriteLine("xi = {0} + ({1} + {2} \* {3}) \* {4} = {5}", a, i, j, n, h, k);

e.setArgumentValue("x", k);

temp = sum;

sum += Hi[i] \* e.calculate() \* h;

Console.WriteLine("I = {0} + H[{1}] \* f({2}) \* {3} = {4}", temp, i, k, h, sum);

}

}

Console.WriteLine("I = {0} \* {1} = {2}", sum, n, sum \* n);

return sum\*n;

}

}

}

Результати програми

a = 0

b = 1.2

f(x) = (3.5\*x^2+x)/(x^4+1)

n = 10000

Висновок

Результат програми різниться з ручним рішенням в п’ятому знакі після коми

***Список використаних джерел***

1. Лященко М.Я., Головань М.С. Чисельні методи:Підручник.-К.:Либідь.
2. Волков Е.А. Численые методы: Учеб. пособие для вузов. – 2-е издание.
3. Краскевич В.Е., Зеленский К.Х. Численые методы в инжинерных расчетах. – К.: Вища школа.

Ручне рішення

y’ + y = 0.5\*x\*y2, y(0) = 2, a=0, b=2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k | xk | yk | dy | y`k | qk | dqk | d2qk | d3qk |
| 0 | 0 | 2 | -0,17714 | -2 | -0,4 | 0,10188 | -0,02862 | 0,01357 |
| 1 | 0,2 | 1,82286 | -0,11582 | -1,49058 | -0,29812 | 0,07327 | -0,01505 | 0,00317 |
| 2 | 0,4 | 1,70704 | -0,0654 | -1,12424 | -0,22485 | 0,05822 | -0,01188 | 0,00099 |
| 3 | 0,6 | 1,64164 | -0,14888 | -0,83315 | -0,16663 | 0,04634 | -0,01089 | 0,00505 |
| 4 | 0,8 | 1,49277 | -0,10325 | -0,60143 | -0,12029 | 0,03546 | -0,00583 | 0,00365 |
| 5 | 1 | 1,38952 | -0,07201 | -0,42414 | -0,08483 | 0,02963 | -0,00218 | 0,0038 |
| 6 | 1,2 | 1,31751 | -0,04471 | -0,27601 | -0,0552 | 0,02744 | 0,00162 | 0,00474 |
| 7 | 1,4 | 1,2728 | -0,01631 | -0,13879 | -0,02776 | 0,02906 | 0,00636 | 0,00824 |
| 8 | 1,6 | 1,25648 | 0,01508 | 0,00652 | 0,0013 | 0,03542 | 0,0146 |  |
| 9 | 1,8 | 1,27156 | 0,05531 | 0,18362 | 0,03672 | 0,05002 |  |  |
| 10 | 2 | 1,32687 |  | 0,43372 | 0,08674 |  |  |  |

Фрагмент коду програми

using System;

using org.mariuszgromada.math.mxparser;

namespace seven\_task

{

public static class AdamsDifEqAlg

{

public static double[] solve(Expression e, double begin, double end, int n, double y0)

{

double h = Math.Abs((begin - end) / n);

AdamsTable at = new AdamsTable(n, begin, end,h);

at.x[0] = begin;

at.x[1] = begin+h;

at.x[2] = begin+2\*h;

at.x[3] = begin+3\*h;

at.y[0] = RungeKuttaDifEqAlg.solve(e, at.x[0], at.x[0] + h, 1, y0)[0];

at.y[1] = RungeKuttaDifEqAlg.solve(e, at.x[1], at.x[1]+h, 1, at.y[0])[1];

at.y[2] = RungeKuttaDifEqAlg.solve(e, at.x[2], at.x[2]+h, 1, at.y[1])[1];

at.y[3] = RungeKuttaDifEqAlg.solve(e, at.x[3], at.x[3]+h, 1, at.y[2])[1];

for (int i = 0; i <= 3; i++)

{

e.setArgumentValue("x", at.x[i]);

e.setArgumentValue("y", at.y[i]);

at.yd[i] = e.calculate();

at.q[i] = h \* at.yd[i];

}

for (int i = 0; i < 3; i++)

{

at.dy[i] = at.y[i + 1] - at.y[i];

at.dq[i] = at.q[i + 1] - at.q[i];

}

for (int i = 0; i < 2; i++)

{

at.d2q[i] = at.dq[i + 1] - at.dq[i];

}

for (int i = 0; i < 1; i++)

{

at.d3q[i] = at.d2q[i + 1] - at.d2q[i];

}

for (int i = 3; i < n; i++)

{

at.dy[i] = at.q[i] + at.dq[i - 1] / 2 + 5.0/ 12 \* at.d2q[i - 2] - (3.0 / 8) \* at.d3q[i - 3];

at.x[i + 1] = at.x[i] + h;

at.y[i + 1] = at.dy[i] + at.y[i];

e.setArgumentValue("x",at.x[i + 1]);

e.setArgumentValue("y", at.y[i + 1]);

at.yd[i + 1] = e.calculate();

at.q[i + 1] = at.yd[i + 1] \* h;

at.dq[i] = at.q[i + 1] - at.q[i];

at.d2q[i - 1] = at.dq[i] - at.dq[i - 1];

at.d3q[i - 2] = at.d2q[i - 1] - at.d2q[i - 2];

}

at.show();

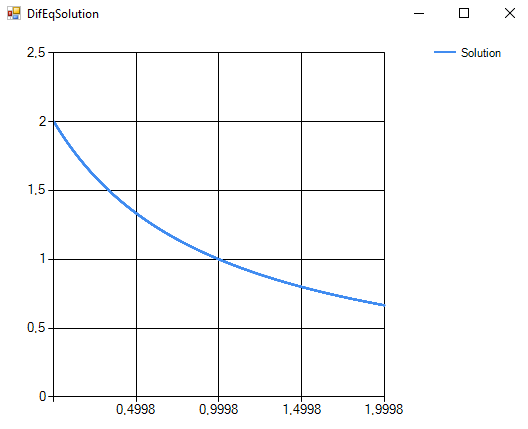
return at.y;

}

}

}

Результат програми



Висновок

Програма малює графік рішення диференційного рівняння і якщо порівнювати значення пораховані ручним методом, то видно, шо вони збігаються

***Список використаних джерел***

1. Г. Корн, Т.Корн. Справочник по математике. Для научных работников и инженеров. 1984г.
2. В.М.Вержбицкий. Численные методы. Математический анализ и обыкновенные дифференциальные уравнения. 2001г.