Численные методы решения обыкновенных

дифференциальных уравнений

Примеры реализации численных методов

// делегат функции вычисления правых частей ДУ

public delegate double[] FuncDelegate(double t, double[] x);

/// <summary>

/// Решение дифференциальных уравнений методом Эйлера

/// </summary>

public class Euler

{

double a; // начало отрезка

double b; // конец отрезка

double h; // шаг

double[] x; // вектор начальных состояний

public Euler(double a, double b, double[] xn, double h)

{

this.a = a;

this.b = b;

this.h = h;

x = xn;

}

/// <summary>

/// Решение дифференциальных уравнений методом Эйлера

/// </summary>

// результат – матрица решений в точках

public double[,] Calculate(FuncDelegate func)

{

int n; //количество шагов

double[] f1;

n = (int)((b - a) / h);//количество шагов

double[,] xr = new double[n + 1, x.Length];

double t = a;

double[] pr = new double[x.Length];

for (int j = 0; j < x.Length; j++)

xr[0, j] = x[j];

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

{

f1 = func(t, x); //1

t = t + h;

for (int k = 0; k < x.Length; k++)

x[k] = x[k] + h \* f1[k];

for (int j = 0; j < x.Length; j++)

{

xr[i, j] = x[j];

}

}

return xr;

}

public void View(double[,] arr) {

int n = (int)((b - a) / h);

Console.WriteLine();

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

{

for (int j = 0; j < x.Length; j++)

{

Console.WriteLine("x[" + i + "] = " + arr[i,j] );

}

}

}

}

Второй вариант

public class Diff

{

private readonly double \_rightBorder; // правая граница

private readonly double \_leftBorder; // левая граница

//делегат правых частей

private readonly Func<double, double[],double[]> \_function;

// массив значений

private double[] \_initialValue;

private readonly int \_n; // количество точек

private readonly double \_h; // шаг интегрирования

public Diff(double leftBorder, double rightBorder, Func<double, double[],double[]> function, double[] initialValue,int n)

{

\_rightBorder = rightBorder;

\_leftBorder = leftBorder;

\_function = function;

\_initialValue = initialValue;

\_n = n;

\_h = (\_rightBorder - \_leftBorder)/\_n;

}

// получение решения в виде матрицы

public Matrix GetValue()

{

var x = \_leftBorder;

Matrix result = new Matrix(\_n, \_initialValue.Length);

for (int i = 0; i < result.NumberOfColumns(); i++)

result[0, i] = \_initialValue[i];

for (int i = 1; i < \_n; i++)

{

// шаг интегрирования по методу Эйлера

\_initialValue = Euler (x);

for (int j = 0; j < \_initialValue.Length; j++)

result[i, j] = \_initialValue[j];

x += \_h;

}

return result;

}

private double[] Euler (double x)

{

double[] y = new double[\_initialValue.Length];

double[] k1 = \_function(x, \_initialValue);

for (int i = 0; i < \_initialValue.Length; i++)

y[i] = \_initialValue[i] + \_h \* k1[i] ;

return y;

}

}

// Пример головной программы

static void Main()

{

Diff D = new Diff(0,1,

(t,x) => new [] {x[1], -4\*x[0]+Math.Cos(3\*t)},

new double[] { 0.8, 2 },

21);

Console.WriteLine(D.GetValue());

Console.ReadKey();

}

Необходимо составить программы решения ДУ методами Рунге-Кутта 2 го и 4 –го порядков, любой из вариантов метода Адамса и сравнить эти методы на одном конкретном примере.