Университет ИТМО

ФПИиКТ

Вычислительная математика

Лабораторная работа 4

Решение задачи Коши методом Адамса

Выполнил: Давтян Давид А.

Группа P3201

Преподаватель: Перл Ольга Вячеславовна

Санкт-Петербург, 2020 г.

Описание метода

Метод Адамса – многошаговый метод 4го порядка точности. Используемая в данном методе формула

прогноза получена интегрированием обратной интерполяционной формулы Ньютона и имеет вид:

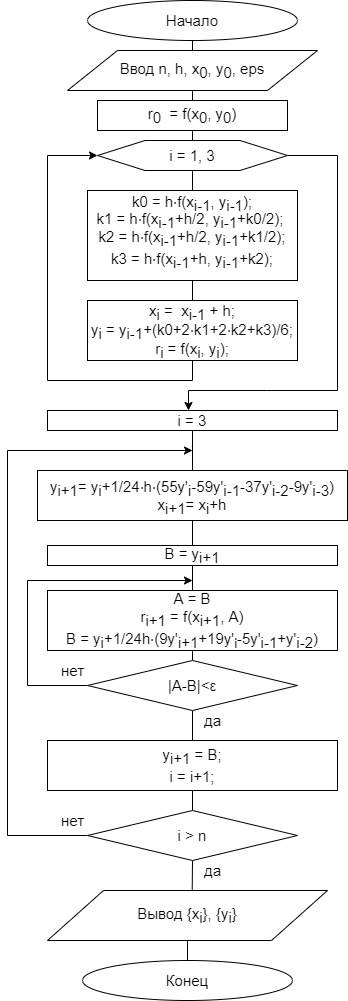
На этапе коррекции используется формула:

Данный метод имеет четвёртый порядок точности, использует в качестве интерполяционного

многочлена полином Лагранжа. А также ошибка, полученная на очередном шаге, не имеет тенденцию

к экспоненциальному росту.

Блок-схема



Листинг численного метода

package ru**.**david**.**compmath4**.**math**.**util**;**

**import** ru**.**david**.**compmath4**.**math**.**expression**.**MathFunctionAdapter**;**

**import** ru**.**david**.**compmath4**.**math**.**model**.**XY**;**

**import** java**.**util**.**LinkedList**;**

**import** java**.**util**.**List**;**

public class RungeKuttaMethodSolver **{**

public static List**<**XY**>** solve**(**

double startX**,**

double startY**,**

double iterations**,**

double step**,**

MathFunctionAdapter function

**)** **{**

List**<**XY**>** result **=** **new** LinkedList**<>();**

result**.**add**(new** XY**(**startX**,** startY**));**

double nextY **=** startY**;**

**for** **(**double currentX **=** startX**,** iteration **=** 0**;** iteration **<** iterations**;** currentX **+=** step**,** iteration**++)** **{**

nextY **=** generateNextY**(**currentX**,** nextY**,** step**,** function**);**

result**.**add**(new** XY**(**currentX **+** step**,** nextY**));**

**}**

**return** result**;**

**}**

private static double generateNextY**(**double x**,** double y**,** double step**,** MathFunctionAdapter function**)** **{**

double k0**,** k1**,** k2**,** k3**,** delta**;**

k0 **=** function**.**value**(**x**,** y**);**

k1 **=** function**.**value**(**x **+** step**/**2**,** y **+** k0 **\*** step**/**2**);**

k2 **=** function**.**value**(**x **+** step**/**2**,** y **+** k1 **\*** step**/**2**);**

k3 **=** function**.**value**(**x **+** step**,** y **+** k2 **\*** step**);**

delta **=** step **\*** **(**k0 **+** 2 **\*** k1 **+** 2 **\*** k2 **+** k3**)/**6**;**

**return** y **+** delta**;**

**}**

**}**

package ru**.**david**.**compmath4**.**math**.**util**;**

**import** ru**.**david**.**compmath4**.**math**.**expression**.**MathFunctionAdapter**;**

**import** ru**.**david**.**compmath4**.**math**.**model**.**XY**;**

**import** java**.**util**.**List**;**

public class AdamsMethodSolver **{**

public static List**<**XY**>** solve**(**

double startX**,**

double startY**,**

double step**,**

double stepsNum**,**

double accuracy**,**

MathFunctionAdapter function

**)** **{**

List**<**XY**>** result **=** RungeKuttaMethodSolver**.**solve**(**startX**,** startY**,** 3**,** step**,** function**);**

int i **=** 3**;**

**do** **{**

double nextY **=** result**.**get**(**i**).**y **+** 1.0**/**24 **\*** step **\*** **(**

55**\***function**.**value**(**result**.**get**(**i**).**x**,** result**.**get**(**i**).**y**)** **-**

59**\***function**.**value**(**result**.**get**(**i **-** 1**).**x**,** result**.**get**(**i **-** 1**).**y**)** **+**

37**\***function**.**value**(**result**.**get**(**i **-** 2**).**x**,** result**.**get**(**i **-** 2**).**y**)** **-**

9**\***function**.**value**(**result**.**get**(**i **-** 3**).**x**,** result**.**get**(**i **-** 3**).**y**)**

**);**

double nextX **=** result**.**get**(**i**).**x **+** step**;**

double b **=** nextY**;**

double a**;**

**do** **{**

a **=** b**;**

b **=** result**.**get**(**i**).**y **+** 1.0**/**24 **\*** step **\*** **(**

9**\***function**.**value**(**nextX**,** nextY**)** **+**

19**\***function**.**value**(**result**.**get**(**i**).**x**,** result**.**get**(**i**).**y**)** **-**

5**\***function**.**value**(**result**.**get**(**i **-** 1**).**x**,** result**.**get**(**i **-** 1**).**y**)** **+**

function**.**value**(**result**.**get**(**i **-** 2**).**x**,** result**.**get**(**i **-** 2**).**y**)**

**);**

**}** **while** **(**Math**.**abs**(**a **-** b**)** **>=** accuracy**);**

result**.**add**(new** XY**(**nextX**,** b**));**

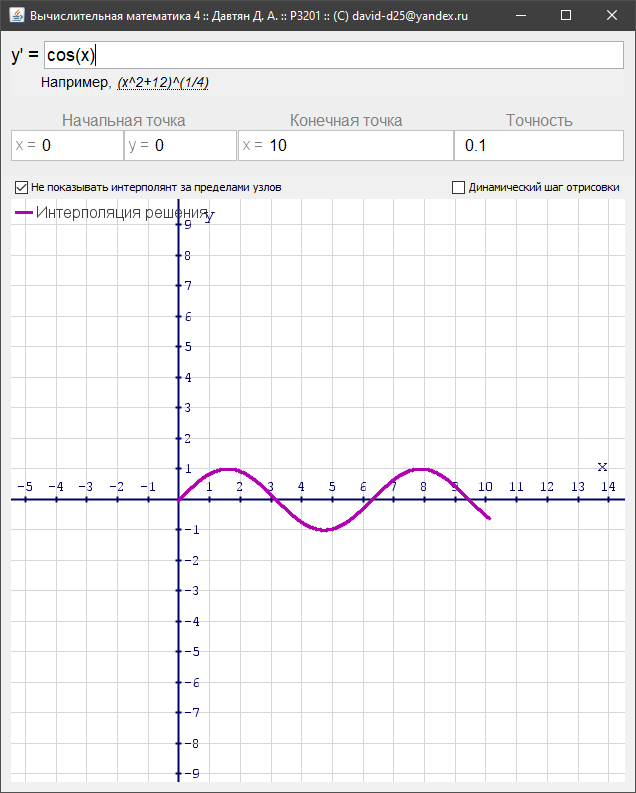
**}** **while** **(++**i **<=** stepsNum**);**

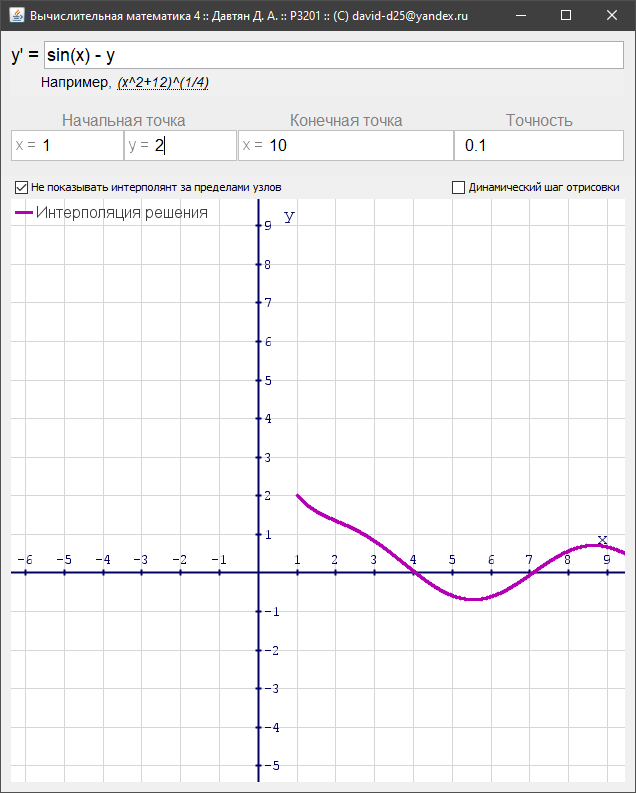
**return** result**;**

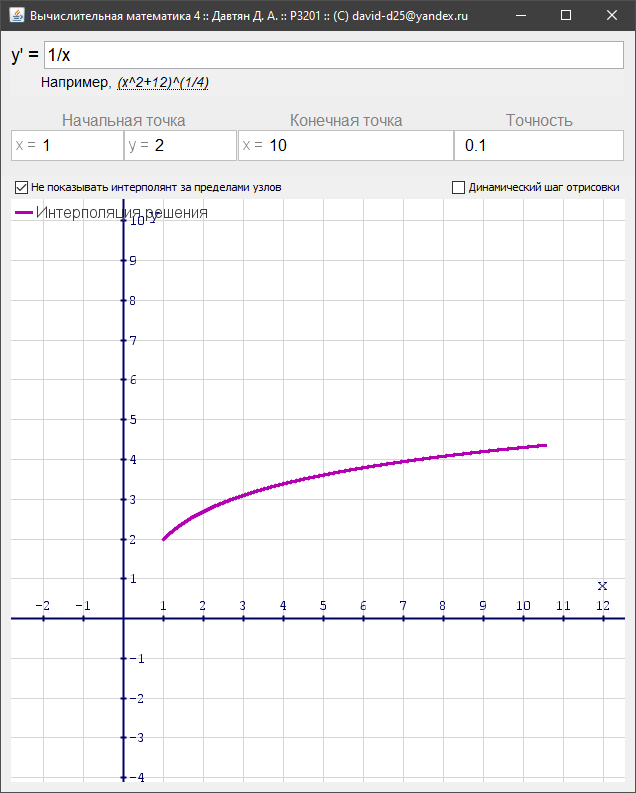
**}**

**}**

Примеры







Вывод

**Метод Адамса** относится к многошаговым методам и представляет один

из методов прогноза и коррекции, как и **метод Милна**. Разница между методами

Адамса и Милна заключается в использовании разных формул прогноза и коррекции: в методе Адамса в качестве интерполяционного полинома используется полином Лагранжа, в методе Милна - полином Ньютона. Оба этих метода имеют четвертый

порядок точности. Поскольку методы являются многошаговыми, для вычисления

значения нам необходимо знать результаты нескольких предыдущих шагов, поэтому невозможно, если так можно выразиться, запустить метод: для этого необходимо предварительно получить одношаговыми методами первые три точки. Кроме того, методы прогноза и коррекции требуют дополнительного расхода памяти – поскольку для них требуются данные о предыдущих точках.Сравнение одношаговых методов:

**Метод Эйлера** основан получении каждого следующего значение из предыдущего.

Данный метод имеет большую погрешность, которая, к тому же, накапливается на каждом шаге. Порядок точности данного метода - первый.**Усовершенствованный метод Эйлера** отличается от обычного тем, что значение правой части уравнения берется равным среднему арифметическому между и , т.е.

затем вычисляется первое приближение , затем подставляем значение в формулу выше и находим уточнённое значение

Данный метод точнее метода Эйлера и имеет второй порядок точности.

**Метод Рунге-Кутта** имеет несколько разновидностей, различающихся порядком точности. В этих методах допускается вычисление правых частей не только в точках сетки, но и в некоторых промежуточных точках.

Рассмотрим **метод Рунге-Кутта 4-го порядка точности**. В данном методе вводятся 4 вспомогательные величины и вычисление координат очередной точки сетки происходит исходя из известных координат предыдущей точки:

(формулы вычисления вспомогательных величин опущены). Таким образом, данный метод требует на каждом шаге четырехкратного вычисления правой части уравнения. Метод Рунге-Кутта требует большого объема вычислений, но имеет повышенную точность, что позволяет проводить вычисления с большим шагом.

При одинаковом шаге метод Эйлера и усовершенствованный метод Эйлера менее точные, в отличие от метода Рунге-Кутта четвёртого порядка.