Предиктор-корректорный метод Адамса 2-го порядка

Задача

Решить задачу Коши для ОДУ методом Адамса с заданной точностью. Использовать правило Рунге для оценки погрешности.

Исследовать метод Адамса графиком фактической точности от размера шага и на устойчивость.

Постановка

Дано ур-ие y’ = f(x,y) на отрезке [a,b], (y(x0) = y0) и шаг h. Необходимо решить задачу Коши в каждой точке, отстоящей на шаг от предыдушей.

Алгоритм метода и его требования

Методы прогноза и коррекции базируются на парах явных и неявных методах Адамса одинакового порядка.   
Пусть – приближенное значение решения y(xi+1), подсчитываемое по явной экстраполяционной ф-ле Адамса-Башфорта, и составим пару из частных формул Адамса-Башфорта и Адамса-Моултона.  
Приведем такую систему:

Реализация алгоритма:

1. *Задаем промежуток [a; b] и кол-во узлов n (Либо сразу размер шага h. Тогда переходим к п. 3).*
2. *Считаем шаг сетки по формуле h = (b-a)/n.*
3. *Задаем начальные данные y(a). y(a+h) считаем методом Рунге-Кутты.*
4. *Примем за текущий x значение (a+h). yi-1 = y(a); yi = y(a+h), xi-1 = a; xi = a+h; xi+1 = xi+h;*
5. *По формуле 1 и 2 вычисляем и .*
6. *Если xi+1 < b, то наращиваем i на 1, принимаем xi+1 = xi + h и идем к шагу 5  
   иначе – завершаем цикл.*

Тестовый пример

Начальные данные:

y' = f(x,y) = 2x/y

Точное решение: y = x^2.

Промежуток: [1,3], кол-во узлов n = 4,   
h = (b-a)/n = 0.5

y(a) = 1 (yi-1)

y(a+h) = 2.25

Начало вычислений:

X = a+h = 1.5

yb = y + h / 2 \* (3 \* f(x, y) - f(x-h, yi-1));

y = y + h / 2 \* (f(x + h, yb) + f(x, y));

1. X = 1.5 (вычисляем для xi+1, то есть, x+h)

X+h = 2  
yb(2) = 2.25 + 0.5/2 \* (3 \* 2\*1.5/2.25 – 2\*1/1 ) = 4  
y(2) = 2.25 + 0.5/2 \*( 2\* 2/4 + 2\*1.5/2.25 ) = 4  
y\*(2) = 4  
  
x+=h

1. x = 2  
   X+h = 2.5  
   yb(2.5) = 2.25 + 0.5/2 \* (3 \* 2\*1.5/2.25 – 2\*1/1 ) = 4  
   y(2.5) = 2.25 + 0.5/2 \*( 2\* 2/4 + 2\*1.5/2.25 ) = 4

y\*(2.5) = 4  
x+=h

1. x == 2.500000  
   X+h = 3  
   yb(3) = 6.25 + 0.5/2 \* (3 \* 5 – 4) = 9  
   y(3) = 9 + 0.5/2 \* (6. + 7.2) = 9  
   y\*(3) = 9  
   x+=h  
   x+h>b => заканчиваем алгоритм.

Контрольные тесты

y’ = f(x, y) =

y\* = – точное значение

[a, b] = [1, 3]

Каждый тест шаг будет уменьшаться в 2 раза. От 20 до 2-14.

Для графика зависимости точности от ошибок в начальных данных, будем вносить ошибку в размере 10^-i, где i = 1…15.

Кол-во узлов будет константным и равняться 2^13.

Фактическая точность вычислялась как бесконечная норма вектора погрешности по каждому узлу.  
Погрешность = Точное значение – Вычисленное значение.

Модульная структура программы

Int Main(void) – считывает из файла кол-во узлов, отрезок и ошибку в начальных данных. Вызывает Method(a,b,n) и записывает в файл его значения.

Double Method(double a, double b, int n) – принимает границы отрезка и кол-во узлов (начальные данные сделал глобальными), выполняет алгоритм и возвращает максимальную погрешность.

Численный анализ

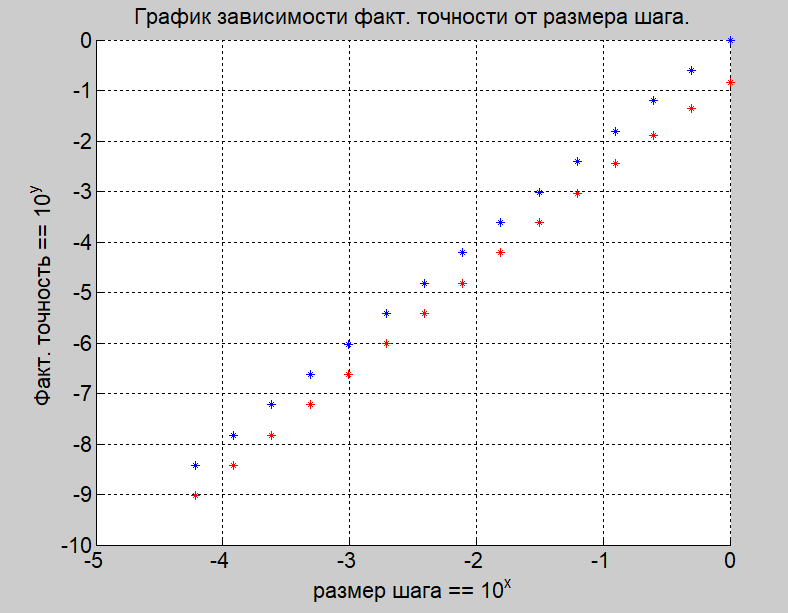


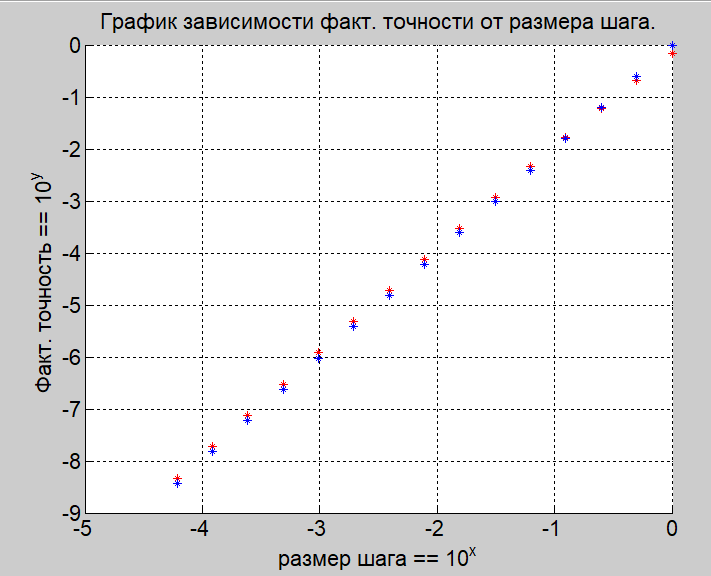
График предиктор-корректорного метода.  
Синим отмечены точки шага в степени порядка точности метода.

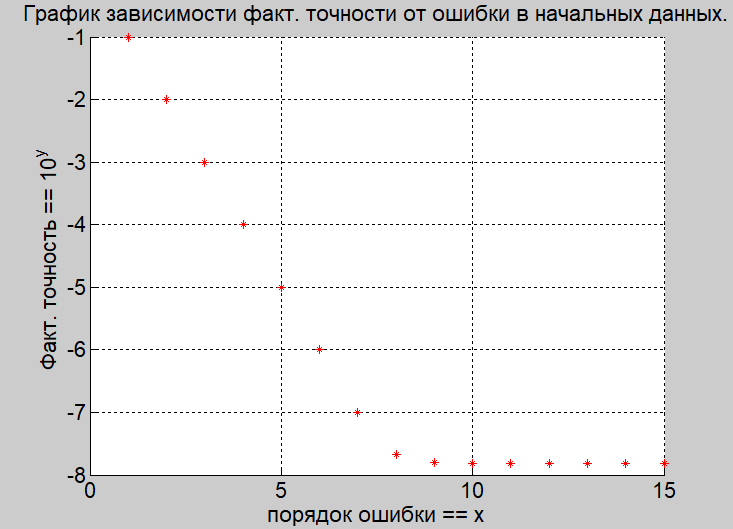
Красным – действительные результаты тестов.

Линейная зависимость log10 факт. точности от log10 размера шага.

Даже всего при 2-х узлах алгоритм считает с точностью, близкой к 0.1.

Ниже представлен аналогичный график только для **предиктора**.





Шаг в этом случае был константным и равнялся 2^-12.

Наблюдается красивая зависимость порядка ошибки и порядка факт. точности:

какая ошибка – такая и точность.

Однако после 9 порядка ошибки факт. точность почти не меняется. Это происходит из-за того, что узлов слишком мало, чтобы считать более точно.

Вывод

Сравнив предиктор-корректорный метод с явным, мы убедились, что он действительно точнее. А также, считает почти на порядок лучше, чем ожидалось.  
Так как в реальности начальные условия не всегда заданы точно, важно знать, как алгоритм реагирует на ошибки в них. В данном случае, метод устойчив к ошибкам в начальных данных. Какая ошибка – такая и факт. точность.