Метод Эйлера-Коши

Задача

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

Исследовать метод Эйлера-Коши с помощью графиков факт. точности от требуемой точности, числа итераций от треб. точности и факт. точности от величины ошибки.

Постановка

Дана задача Коши для ур-ия вида y’ = f(x,y) на отрезке [a,b] и шаг h. Необходимо решить задачу Коши в каждой точке, отстоящей на шаг от предыдушей.

Алгоритм и условия применимости

Задача Коши в простейшем случае ставится для дифференциального уравнения первого порядка с начальным условием y′ = f (x, y) x ∈ [a,b] y(a) = y. Строится сетка (x0=a, xn=b) (в общем случае она может быть неравномерной). n+1 – кол-во узлов.

Идея метода заключена в том, что на малом промежутке изменения независимой переменной х интегральная кривая ДУ заменяется отрезком прямой (касательной). Иными словами, строятся ломаные Эйлера. Метод Эйлера-Коши более точно определяет направление перехода их (xi, yi) в (xi+1, yi+1), чем метод Эйлера.

1. Прогноз  
    - следует из ур-ия касательной.

y – y0 = f(x0, y0)(x-x0) => y1 = y0 + f(x0, y0)h – ф-ла для метода Эйлера.

1. Коррекция  
   yi+1 = yi + h \* – ф-ла интегрирования   
   аппроксимация имеет второй порядок.

Геометрически это означает, что сначала определяется направление интегральной кривой в исходной точке ), а в качестве окончательного направления выбирается среднее значение этих направлений.

Выберем заведомо малый шаг h, построим, начиная с точки х0 = а, систему равностоящих точек xi = x0 + i\*h,(i = 1…n, xn = b). Для каждо1 точки xi требуется решить задачу Коши.  
  
Формула для оценки погрешности по правилу Рунге: , где - решение задачи Коши с шагом h, а 3 – это 2^k – 1, где k – порядок метода.  
  
Этапы решения:   
1) Разбить промежуток на отрезки с шагом h или задать кол-во узлов и посчитать шаг с помощью формулы h = (b-a)/(n+1).  
2) Для каждой точки последовательности найти решение задачи Коши .

Алгоритм метода:   
1) Задать начальное условие y(x0), где x0 = a, промежуток [a,b], и eps.

2) Задать начальное кол-во узлов (m = 2)

3) Вычислить решение задачи коши на промежутке [xi, xi+1]

4) Увеличить m в 2 раза и снова найти решение в каждой точке

5) Вычислить модуль разности результатов п.3 и 4. Если он больше eps, то уменьшаем шаг в 2 раза и переходим к п. 3.

6) Увеличиваем i на единицу и переходим к п.2. Заканчиваем, когда i = n <=> xi+1 = b;

Тестовый пример

y’ = 3(x+1)2  - y

x € [1,1.5]

y0 = y(1) = 6, n = 3

h = (1.5-1) /(3-1) = 0.25

x0 = 1

Вычисляем для т. Х = 1.25

Вычисляем для т. Х = 1.5

Контрольные тесты

Требуется решить задачу Коши для удовлетворения вида y’ = f(x,y).

Дано ур-ие y’ =-(1+xy)/x2

Промежуток – [1,3];

Y(1) = 1;

Точное решение: у = (1-ln(x))/x

Проведем 3 опыта:   
1. Зависимость факт. точности от требуемой.

Сделаем 7 тестов с eps = 10^-1, …, 10^-7.

Найдем факт. точность и отметим результат на графике.

Сохраним в отдельный массив получившееся кол-во узлов (понадобится для след. пункта)

2. Кол-во узлов от треб. точности.

Отметим на новом графике с такой же осью x значения кол-ва узлов из прошлого пункта.

3. Зависимость факт. точности от начальной ошибки.

Возьмем eps = 10^-6 для всех тестов.

Проведем 7 тестов.

В каждом тесте внесем ошибку в начальные данные в размере 10^-i, где i – номер теста.

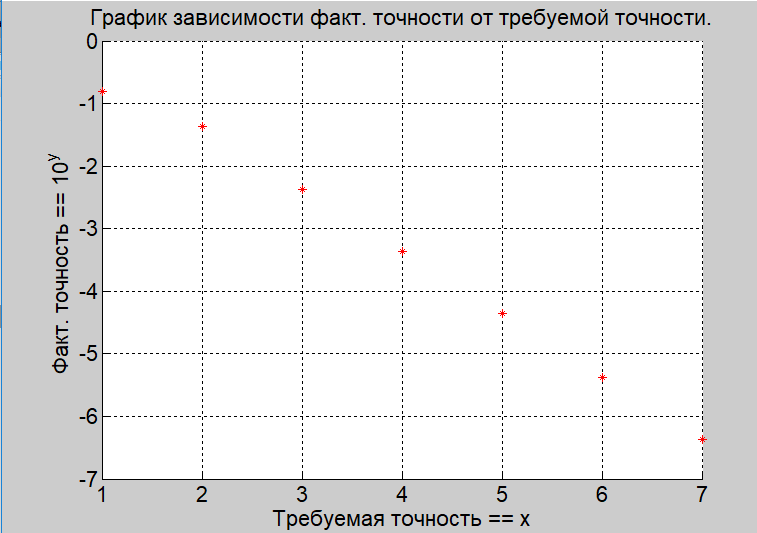
Найдем факт. точность и отметим результат на графике.

Модульная структура программы

int main(void) – считывает необходимую точность, отрезок, величину ошибки начальных значений, вызывает Method и выводит в файл кол-во разбиений и фактическую точность, которые возвращает Method. (число разбиений – глобальная переменная).

double Method(double a, double b, int F\_Type) – принимает на вход границы отрезка, ошибку начального значения и необходимую точность функции, выполняет алгоритм и возвращает бесконечную норму разности значений полученной функции и точного решения.

Численный анализ

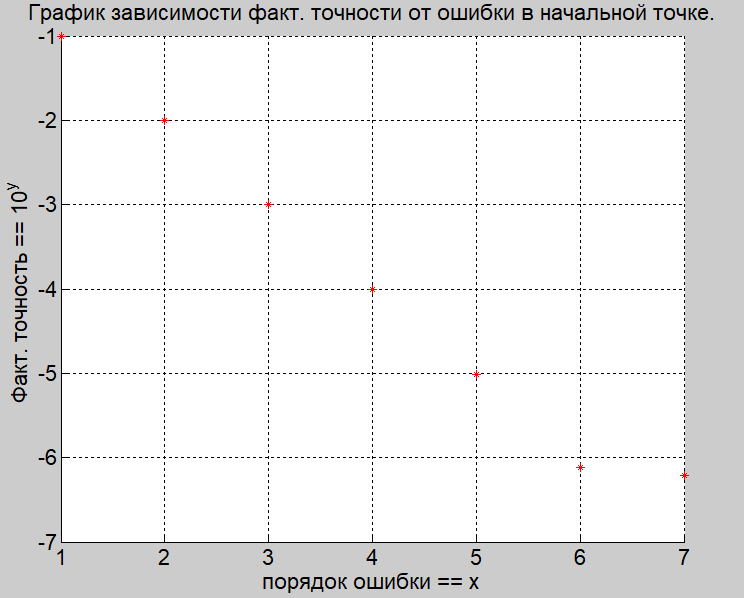
  
\*По Ох – порядок требуемой точности

В большинстве случаев факт. точность хуже примерно на полпорядка.



\*По Ох – порядок требуемой точности

Стабильный линейный рост в логарифмических координатах (логарифм по основанию 2).

­­­­

Факт. точность линейно зависит от ошибки вплоть до 6 порядка.  
При величине ошибки < 10^-6 факт. точность сильно меняться не будет.

Вывод

Метод достаточно точный и устойчивый к ошибкам. Ошибка в начальных данных к катастрофическим последствиям не приведет. Кол-во узлов растет экспоненциально, следовательно, и время будет расти так же с увеличением требуемой точности.