

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
Московский государственный технический университет им. Н.Э. Баумана
Факультет «Фундаментальные науки»

Лабораторная работа №8
по курсу «Вычислительная физика»
Тема: «Одношаговые численные методы решения задачи Коши»
Вариант 6

Выполнили: студенты группы ФН4-72Б
Хижик А.И., Мистрюкова Л.А.
Проверил: доцент, к.физ.-мат.н.
Хасаншин Р.Х.

Москва, 2019

Оглавление

1.	Теоретическая часть	3
1.1.	Введение	3
1.2.	Метод Эйлера	4
1.3.	Модифицированный метод Эйлера	6
1.4.	Усовершенствованный метод Эйлера	7
1.5.	Метод Эйлера-Кромера	8
1.6.	Метод Рунге-Кутты четвёртого порядка	8
2.	Постановка задачи	10
3.	Программа	11
4.	Результаты вычислений	15
4.1.	Задание А	15
4.2.	Задание Б	16
4.3.	Задание С	19
5.	Вывод	23

1. Теоретическая часть

1.1. Введение

Одношаговым методом численного решения дифференциального уравнения (ДУ) называют метод, при использовании которого для получения решения в каждой новой узловой точке достаточно иметь значение сеточной функции лишь в предыдущем узле.

Суть разностного метода решения обыкновенных дифференциальных уравнений (ОДУ) заключается в следующем. Область непрерывного изменения аргумента заменяют дискретным множеством точек, называемых узлами. Совокупность этих узлов составляет разностную сетку. Искомую функцию непрерывного аргумента заменяют функцией дискретного аргумента на заданной сетке, которую называют сеточной функцией. Путём аппроксимации производных конечно-разностными соотношениями исходное дифференциальное уравнение заменяют разностным уравнением относительно сеточной функции. Такую замену ДУ разностным называют его аппроксимацией на сетке (или разностной аппроксимацией).

Любая дифференциальная задача может быть представлена в следующем виде:

$$L[y] = F(x), \quad x \in G, \quad (1)$$

$$l[y] = \Phi(x), \quad x \in \Gamma. \quad (2)$$

где L, l – линейные дифференциальные операторы, $F(x)$ – заданная функция, G – область решения задачи, Γ – область, в которой заданы дополнительные условия.

Введем сетку g_h – конечное множество точек, принадлежащих области G , плотность распределения которых характеризуется параметром h – шагом сетки; с помощью формул численного дифференцирования заменим линейные дифференциальные операторы L, l на разностные аналоги L_h, l_h ; функцию непрерывного аргумента $y(x)$ заменим сеточной функцией u_h , а правую часть уравнений (1-2) на сеточные функции $f_h(x)$ и $\varphi_h(x)$ соответственно. В результате получим систему разностных уравнений, называемую разностной схемой (РС):

$$L_h[u_h] = f_h(x), \quad x \in g_h, \quad (3)$$

$$l_h[u_h] = \varphi_h(x), \quad x \in \gamma_h. \quad (4)$$

Основные понятия теории разностных схем

Погрешностью РС (3-4) называется величина

$$\delta_h(x) = u_h(x) - P_h[y(x)], \quad x \in g_h, \quad (5)$$

где P_h – оператор проектирования.

Выразив из (5) функцию $u_h(x)$, подставим ее в (3-4):

$$L_h[y_h] + L_h[P_h[y(x)]] = f_h(x),$$

$$l_h[y_h] + l_h[P_h[y(x)]] = \varphi_h(x).$$

Функция $R_h(x) = L_h[\delta_h(x)] = f_h(x) - L_h[P_h[y(x)]]$ называется невязкой ДУ, а $r_h(x) = \varphi_h(x) - l_h[P_h[y(x)]]$ – невязкой для дополнительных условий.

Характерные значения для невязок и погрешности на сетке: $R = \max_{g_h} R_h$, $r = \max_{\gamma_h} r_h$, $\delta = \max_{g_h} \delta_h$.

Если характерные величины для невязок имеют вид $R = O(h^k)$, $r = O(h^k)$, то говорят, что РС (3-4) аппроксимирует исходное ДУ порядком k .

РС (3-4) аппроксимирует исходную дифференциальную задачу (1-2), если $\|R_h(x)\|_h \rightarrow 0$ и $\|r_h(x)\|_h \rightarrow 0$ при $h \rightarrow 0$.

РС (3-4) сходится к решению дифференциальной задачи (1-2), если при $h \rightarrow 0$ и $\|u_h - P_h[y]\|_h = \|\delta_h\|_h \rightarrow 0$.

Для того, чтобы РС была сходящейся, необходимо и достаточно, чтобы она была аппроксимирующей и устойчивой.

Задача Коши

Рассмотрим задачу Коши для ДУ

$$u' = f(x, u), \quad x > x_0, \quad (6)$$

с начальным условием $u(x_0) = u_0$.

Теорема Коши: если правая часть ДУ (6) и ее частная производная $f'_u(x, u)$ определены и непрерывны в некоторой области G' изменения переменных x и u , то для всякой внутренней точки (x_0, u_0) этой области данное уравнение имеет единственное решение, принимающее заданное значение u_0 при $x = x_0$.

Методы решения дифференциальной задачи (6) распространяются и на случай систем ДУ, а к ним, в свою очередь, можно привести также уравнения высших порядков.

1.2. Метод Эйлера

Метод Эйлера является простейшим численным методом решения задачи Коши для ДУ. Рассмотрим дифференциальную задачу

$$y' = f(x, y), \quad x \in [a, b] \quad (7)$$

$$y(a) = y_0 \quad (8)$$

Введем на отрезке $[a, b]$ сетку $g_h = a = x_0 < x_1 < \dots < x_N = b$ с шагом сетки $h_n = x_{n+1} - x_n$.

Разложим решение $y(x)$ в окрестности точки x_n в ряд Тейлора:

$$y(x_{n+1}) = y(x_n) + h_n y'(x_n) + \frac{1}{2!} h_n^2 y''(x_n) + \dots \quad (9)$$

Если функция $f(x, u)$ имеет непрерывные частные производные до порядка s , то в выражении (9) можно оставить члены вплоть до $O(h_n^{s+1})$, тогда, например, первую и вторую производные соответственно можно представить в виде

$$y'(x_n) = f(x_n, y(x_n))$$

$$y''(x_n) = \frac{dy'}{dx}(x_n) = f'_y(x_n, y(x_n))y'(x_n) + f'_x(x_n) = f \frac{\partial f}{\partial y}(x_n) + \frac{\partial f}{\partial x}(x_n)$$

Использование выражения (9) с большим числом членов имеет следующие основные недостатки:

1. С увеличением порядка производных выражения для них усложняются;
2. Если функция f известна лишь приближенно или задана в виде таблицы, ее производные вычисляются с большой погрешностью.

Поэтому в ряде оставляют только два первых члена. При такой замене вместо точного решения $y(x_{n+1})$ в узловых точках получают его приближённое значение u_{n+1} :

$$u_{n+1} = u_n + h_n f(x_n, u_n), \quad n = \overline{0, N-1}. \quad (10)$$

Поскольку значение $u_0 = y_0$ известно из начальных условий (8), то, используя формулу (10), последовательно находим u_1, \dots, u_N – приближённое значение дифференциальной задачи (7-8) в узловых точках. Формулу (10) для равномерной сетки ($h_n = h = \text{const}$),

$$u_{n+1} = u_n + h f(x_n, u_n), \quad (11)$$

называют методом Эйлера (методом ломанных).

Погрешность метода

Погрешность при использовании метода обусловлена тем, что приращение значения функции при переходе от точки $x_n, n = \overline{0, N-1}$ к точке x_{n+1} заменяется приращением ординаты касательной к соответствующей интегральной кривой.

Погрешность δ_n в точке x_n равна разности точного решения дифференциальной задачи (7-8) $y(x_n)$ и значения сеточной функции u_n :

$$\delta_n = y(x_n) - u_n \rightarrow u_n = y(x_n) - \delta_n, \quad (12)$$

$$\delta_{n+1} = y(x_{n+1}) - u_{n+1} \rightarrow u_{n+1} = y(x_{n+1}) - \delta_{n+1}. \quad (13)$$

Подставим выражения (12-13) в формулу (11):

$$y(x_{n+1}) - \delta_{n+1} = y(x_n) - \delta_n + hf(x_n, y(x_n) - \delta_n). \quad (14)$$

Разложим функцию f в ряд в окрестности точки $(x_n, y(x_n))$:

$$f(x_n, y(x_n) - \delta_n) = f(x_n, y(x_n)) - \delta_n \frac{\partial f}{\partial y} + O(\delta_n^2) = f(x_n, y(x_n)) + O(\delta_n).$$

Следовательно,

$$\delta_{n+1} - \delta_n = O(h^2) + hO(\delta_n),$$

где первое слагаемое определяет погрешность аппроксимации производной, а второе – неточность значения u_h .

Для нахождения значения u_1 используем начальное значение $u_0 = y_0$, которое задаётся, как правило, точно, т.е. $\delta_0 = 0$. Отсюда $\delta_1 = O(h^2)$, $\delta_2 = \delta_1 + O(h^2) + hO(h^2) = \delta_1 + O(h^2)$, ..., $\delta_{n+1} = \delta_n + O(h^2)$, $n = \overline{0, N-1}$. δ_n называется локальной погрешностью. Поскольку $h = \frac{b-a}{N}$, то для погрешности на сетке δ_N (глобальной погрешности) получим следующее соотношение

$$\delta_N = NO(h^2) = \frac{b-a}{h}O(h^2) = O(h).$$

Таким образом, метод Эйлера имеет первый порядок точности на сетке и второй порядок точности на шаге.

1.3. Модифицированный метод Эйлера

Рассмотрим дифференциальное уравнение (7) в окрестности точки $x = x_n + \frac{h}{2}$, $n = \overline{0, N-1}$, являющейся серединой отрезка $[x_n, x_{n+1}]$. В левой части дифференциального уравнения (7) производную заменим центральной разностью

$$y' \left(x_n + \frac{h}{2} \right) \approx \frac{u_{n+1} - u_n}{h},$$

а в правой части уравнения значением функции $f \left(x_n + \frac{h}{2}, y(x_n + \frac{h}{2}) \right)$ – средним арифметическим значением функции в точках (x_n, u_n) и (x_{n+1}, u_{n+1}) . Тогда

$$u_{n+1} = u_n + \frac{h}{2} [f(x_n, u_n) + f(x_{n+1}, u_{n+1})]. \quad (15)$$

Если искомое решение u_{n+1} входит в правую часть уравнения (15) и оно не может быть разрешено относительно u_{n+1} , то формулу (15) называют неявной схемой. Для вычисления значения u_{n+1} можно применить один из итерационных методов. Если имеется значение u_n , то решение можно построить с использованием двух итераций следующим образом. Считая значение u_n начальным приближением, вычислим первое приближение \tilde{u}_{n+1} к значению u_{n+1} по формуле (11) метода Эйлера:

$$\tilde{u}_{n+1} = u_n + hf(x_n, u_n). \quad (16)$$

Вычисленное значение \tilde{u}_{n+1} подставим вместо u_{n+1} в правую часть уравнения (15) и найдем

окончательное значение u_{n+1} :

$$u_{n+1} = u_n + \frac{h}{2}[f(x_n, u_n) + f(x_{n+1}, \tilde{u}_{n+1})]. \quad (17)$$

Таким образом, получили следующие соотношения:

$$u_{n+1} = u_n + \frac{h}{2}[f(x_n, u_n) + f(x_{n+1}, u_n + hf(x_n, u_n))], \quad n = \overline{0, N-1}. \quad (18)$$

Рекуррентные соотношения (17) описывают модифицированный метод Эйлера.

При замене производной в левой части уравнения (7) центральной разностью допускается погрешность порядка $O(h^2)$. Покажем, что погрешность такого же порядка допускается при замене правой части уравнения (7) в точке $x = x_n + \frac{h}{2}$ средним арифметическим значением функции $f(x, y(x))$ в точках (x_n, u_n) и (x_{n+1}, u_{n+1}) . Действительно, разложим $f(x, y(x))$ в окрестности точки $(x_n + \frac{h}{2}, y(x_n + \frac{h}{2}))$:

$$\begin{aligned} \frac{1}{2}[f(x_n, u_n) + f(x_{n+1}, u_{n+1})] &= \frac{1}{2}\left[f\left(x_n + \frac{h}{2}, y\left(x_n + \frac{h}{2}\right)\right] - \frac{h}{2}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}y'\right) + O(h^2) + \right. \\ &\quad \left. + f\left[x_n + \frac{h}{2}, y\left(x_n + \frac{h}{2}\right)\right] + \frac{h}{2}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}y'\right) + O(h^2)\right] = \\ &= f\left[x_n + \frac{h}{2}, y\left(x_n + \frac{h}{2}\right)\right] + O(h^2) \end{aligned}$$

Таким образом, погрешность вычислений составляет $hO(h^2) = O(h^3)$. Этот порядок погрешности сохраняется и при использовании двух итераций, поскольку

$$f(x_{n+1}, \tilde{u}_{n+1}) = f(x_{n+1}, u_{n+1}) + \frac{\partial f}{\partial y}(\tilde{u}_{n+1} - u_{n+1}) + O(h^2) = f(x_{n+1}, u_{n+1}) + O(h^2).$$

Следовательно, локальная погрешность имеет порядок $O(h^3)$, а глобальная – $O(h^2)$, т.е. модифицированный метод Эйлера имеет второй порядок точности.

1.4. Усовершенствованный метод Эйлера

Также как и в случае модифицированного метода Эйлера, рассмотрим ДУ (7) в окрестностях точек $x = x_n + \frac{h}{2}$, $n = \overline{0, N-1}$. Левую часть ДУ заменим центральной разностью, а правую оставим без изменений:

$$\frac{u_{n+1} - u_n}{h} = f\left(x_n + \frac{h}{2}, y\left(x_n + \frac{h}{2}\right)\right). \quad (19)$$

Приближённое значение функции $y(x)$ в точке $(x_n + \frac{h}{2})$ вычислим с помощью метода Эйлера:

$$\tilde{u}_{n+1} = u_n + \frac{h}{2}f(x_n, u_n). \quad (20)$$

Выразим u_{n+1} из (19), заменив $y(x_n + \frac{h}{2})$ его приближением \tilde{u}_{n+1} по формуле (11):

$$u_{n+1} = u_n + hf\left(x_n + \frac{h}{2}, \tilde{u}_{n+1}\right). \quad (21)$$

Алгоритм решения дифференциальной задачи (7-8) по формулам (20-21) называют усовершенствованным методом Эйлера.

1.5. Метод Эйлера-Кромера

Метод Эйлера-Кромера является улучшенной модификацией метода Эйлера.

Рассмотрим классическое уравнение движение материальной точки

$$\ddot{x} = f(x, \dot{x}, t), \quad t > t_0, \quad (22)$$

с начальными условиями $x(t_0) = x_0, \dot{x}(t_0) = v_0$.

Уравнение (22) можно представить в виде системы
$$\begin{cases} \dot{v}(t) = f(x, v, t), \\ \dot{x}(t) = v. \end{cases}$$

Введем сетку $g_\tau = \{t_n\}, t_n = n\tau, n = 0, 1, 2, \dots$, где τ - шаг сетки.

Алгоритм решения методом Эйлера-Кромера:

$$v_{n+1} = v_n + \tau f(x_n, v_n, t_n),$$

$$x_{n+1} = x_n + \tau v_{n+1}.$$

Алгоритм решения методом Эйлера:

$$v_{n+1} = v_n + \tau f(x_n, v_n, t_n),$$

$$x_{n+1} = x_n + \tau v_n.$$

Метод Эйлера-Кромера, несмотря на первый порядок точности, в случае решения уравнения колебаний вида (22) значительно повышает реальную точность по сравнению с явным методом Эйлера, особенно при больших значениях t . В методе Эйлера-Кромера повышенная точность обеспечивается тем, что при вычислении нового значения координаты x_{n+1} используется значение скорости материальной точки v_{n+1} в новой узловой точке.

1.6. Метод Рунге-Кутты четвёртого порядка

Основная идея методов Рунге-Кутты повышенной точности заключается в построении специального алгоритма решения задач Коши для ОДУ, такого, чтобы максимально приблизить приращение сеточной функции $\Delta u = u_{n+1} - u_n$ на шаге $n+1$ к приращению точного решения $\Delta y = y(x_{n+1}) - y(x_n)$, которое определяется из ряда Тейлора (9) в окрестности точки x_n с учётом возможно большего числа членов ряда. При этом, чтобы избежать громоздких выражений, приводящих к увеличению погрешности метода Рунге-Кутты, вторые и последующие производные определяют не дифференцированием, а путём многократного вычисления правой части дифференциального уравнения – функции $f(x, y)$ в некоторых промежуточных точках.

Широкое применение для численного решения задач Коши для дифференциальных уравнений получил метод Рунге-Кутты четвёртого порядка. Его алгоритм для решения дифференциальной

задачи (7-8) выглядит следующим образом:

$$u_{n+1} = u_n + \frac{h}{6}(k_0 + 2k_1 + 2k_2 + k_3), \quad n = 0, 1, \dots, \quad (23)$$

$$k_0 = f(x_n, u_n), \quad k_1 = f\left(x_n + \frac{h}{2}, u_n + \frac{hk_0}{2}\right), \quad k_2 = f\left(x_n + \frac{h}{2}, u_n + \frac{hk_1}{2}\right), \quad k_3 = f(x_n + h, u_n + hk_2).$$

Метод Рунге-Кутты требует на каждом шаге четырёхкратного вычисления правой части $f(x, y(x))$ ДУ (7). Глобальная погрешность этого метода имеет порядок $O(h^4)$, а локальная погрешность – $O(h^5)$.

Таким образом, метод Рунге-Кутты в отличие от метода Эйлера и его модификаций связан с большим объёмом вычислений, однако это окупается повышенной точностью, что даёт возможность проводить счёт с большим шагом.

2. Постановка задачи

- Решить задачу Коши для дифференциального уравнения $u' = (2 - u) \tan(x)$, $x \in [0, 1]$ с начальным условием $u(0) = -1$, используя метод Эйлера, модифицированный метод Эйлера, усовершенствованный метод Эйлера и метод Рунге-Кутты четвёртого порядка.
- Решить методом Эйлера и методом Эйлера-Кромера задачу об апериодическом колебательном движении:

$$\ddot{x} + 2\lambda\dot{x} + \omega_0^2 x = 0, \quad 0 < t < T, \quad \lambda > \omega_0, \quad T = \frac{3}{\lambda},$$
$$x(0) = 0, \quad \dot{x}(0) = v_0.$$

Сравнить численное решение с аналитическим

$$x(t) = \frac{v_0}{\theta} \exp(-\lambda t) \sinh(t\theta), \quad \theta = \sqrt{\lambda^2 - \omega_0^2},$$

где $\omega_0 = \frac{1}{2}$, $\lambda = 1$, $v_0 = 1$.

Построить графики точного и численного решений, а также графики погрешности $\delta_n = x(t_n) - x_n$.

- Решить методом Эйлера и методом Эйлера-Кромера задачу о свободных затухающих колебаниях:

$$\ddot{x} + 2\lambda\dot{x} + \omega_0^2 x = 0, \quad 0 < t < 2T, \quad \lambda < \omega_0, \quad T = \frac{2\pi}{\omega},$$
$$\omega = \sqrt{\omega_0^2 - \lambda^2}, \quad \omega_0 > \lambda, \quad x(0) = 0, \quad \dot{x}(0) = v_0.$$

Сравнить численное решение с аналитическим

$$x(t) = \frac{v_0}{\omega} \exp(-\lambda t) \sin(\omega t),$$

где $\omega_0 = 1$, $\lambda = 0.1$, $v_0 = 1$.

Построить графики точного и численного решений, а также графики погрешности $\delta_n = x(t_n) - x_n$.

3. Программа

Задание А

```
#include <iostream>
#include <fstream>

using std::cout;
using std::cin;
using std::endl;
using std::ofstream;

double a = 0, b = 1, y_zero = -1;
int N = 1000;
double h = (b-a) / N;
double fun(double x, double y);
double fun_ex(double x);
void Euler_Method(double* x, double* y, double* delta, int i);
void Modified_Euler_Method(double* x, double* y, double* delta, int i);
void Improved_Euler_Method(double* x, double* y, double* delta, int i);
void RungeKutta(double* x, double* y, double* delta, int i);

int main() {
    double *y = new double[N+1];
    double *x = new double[N+1];
    double *delta = new double[N+1];
    y[0] = y_zero;
    x[N] = b;
    int i;
    for(i=0; i<=N; i++){
        x[i] = a + i*h;
    }

    cout << "EM" << endl;
    Euler_Method(x, y, delta, i);

    cout << endl << "MEM" << endl;
    Modified_Euler_Method(x, y, delta, i);

    cout << endl << "IEM" << endl;
    Improved_Euler_Method(x, y, delta, i);

    cout << endl << "R-K" << endl;
    RungeKutta(x, y, delta, i);
    return 0;
}

void Euler_Method(double* x, double* y, double* delta, int i){
    ofstream fout("Euler_Method");
    for(i=0; i<=N; i++){
        y[i+1] = y[i] + h*fun(x[i],y[i]);
        delta[i] = fun_ex(x[i]) - y[i];
    }
}
```

```

        cout << i << "\t" << x[i] << "\t" << y[i] << "\t" << delta[i] << endl;
        fout << i << "\t" << x[i] << "\t" << y[i] << "\t" << delta[i] << endl;
    }
    fout.close();
}

void Modified_Euler_Method(double* x, double* y, double* delta, int i){
    ofstream fout("Modified_Euler_Method");
    for(i=0; i<=N; i++){
        y[i+1] = y[i] + h*(fun(x[i],y[i]) + fun(x[i+1],y[i] + h*fun(x[i],y[i]))) / 2;
        delta[i] = fun_ex(x[i]) - y[i];
        cout << i << "\t" << x[i] << "\t" << y[i] << "\t" << delta[i] << endl;
        fout << i << "\t" << x[i] << "\t" << y[i] << "\t" << delta[i] << endl;
    }
    fout.close();
}

void Improved_Euler_Method(double* x, double* y, double* delta, int i){
    ofstream fout("Improved_Euler_Method");
    for(i=0; i<=N; i++){
        y[i+1] = y[i] + h*fun(x[i] + h / 2, y[i] + h * fun(x[i], y[i]) / 2);
        delta[i] = fun_ex(x[i]) - y[i];
        cout << i << "\t" << x[i] << "\t" << y[i] << "\t" << delta[i] << endl;
        fout << i << "\t" << x[i] << "\t" << y[i] << "\t" << delta[i] << endl;
    }
    fout.close();
}

void RungeKutta(double* x, double* y, double* delta, int i){
    ofstream fout("RungeKutta");
    double *k0 = new double[N+1];
    double *k1 = new double[N+1];
    double *k2 = new double[N+1];
    double *k3 = new double[N+1];
    for(i=0; i<=N; i++){
        k0[i] = fun(x[i], y[i]);
        k1[i] = fun(x[i] + h / 2, y[i] + h * k0[i] / 2);
        k2[i] = fun(x[i] + h / 2, y[i] + h * k1[i] / 2);
        k3[i] = fun(x[i] + h, y[i] + h * k2[i]);
        y[i+1] = y[i] + h * (k0[i] + 2 * k1[i] + 2 * k2[i] + k3[i]) / 6;
        delta[i] = fun_ex(x[i]) - y[i];
        cout << i << "\t" << x[i] << "\t" << y[i] << "\t" << delta[i] << endl;
        fout << i << "\t" << x[i] << "\t" << y[i] << "\t" << delta[i] << endl;
    }
    fout.close();
}

double fun(double x, double y){
    return (2-y) * tan(x);
}

double fun_ex(double x) {
    return -3 * cos(x) + 2;
}

```

Задание Б

```
#include <iostream>
#include <cmath>
#include <fstream>

using std::cout;
using std::cin;
using std::endl;
using std::ofstream;

double N = 1000;
double lambda = 1, omega_zero = 1. / 2, upsilon_zero = 1, x_zero = 0;
double T = 3 / lambda;
double tau = T / N;
double theta = sqrt(pow(lambda, 2) - pow(omega_zero, 2));

void Euler_Method(double* x, double* v, double* t, double* delta, int i);
void Euler_Kromer(double* x, double* v, double* t, double* delta, int i);
double fun_ex(double t);
double fun(double x, double v);
int main() {
    double *x = new double[N+1];
    double *v = new double[N+1];
    double *t = new double[N+1];
    double *delta = new double[N+1];
    x[0] = x_zero;
    v[0] = upsilon_zero;
    int i;
    for(i=0; i<=N; i++){
        t[i] = i * tau;
    }
    cout << "EM" << endl;
    Euler_Method(x, v, t, delta, i);

    cout << endl << "E-K" << endl;
    Euler_Kromer(x, v, t, delta, i);
    return 0;
}

void Euler_Method(double* x, double* v, double* t, double* delta, int i){
    ofstream fout("Euler_Method");
    for(i=0; i<=N; i++){
        v[i + 1] = v[i] + tau * fun(x[i], v[i]);
        x[i + 1] = x[i] + tau * v[i];
        delta[i] = fun_ex(t[i]) - x[i];
        cout << i << "\t" << t[i] << "\t" << v[i] << "\t" << x[i] << "\t" << delta[i] << endl;
        fout << i << "\t" << t[i] << "\t" << v[i] << "\t" << x[i] << "\t" << delta[i] << endl;
    }
    fout.close();
}

void Euler_Kromer(double* x, double* v, double* t, double* delta, int i){
```

```

ofstream fout("Euler_Kromer");
for(i=0; i<=N; i++){
    v[i + 1] = v[i] + tau * fun(x[i], v[i]);
    x[i + 1] = x[i] + tau * v[i + 1];
    delta[i] = fun_ex(t[i]) - x[i];
    cout << i << "\t" << t[i] << "\t" << v[i] << "\t" << x[i] << "\t" << delta[i] << endl;
    fout << i << "\t" << t[i] << "\t" << v[i] << "\t" << x[i] << "\t" << delta[i] << endl;
}
fout.close();
}

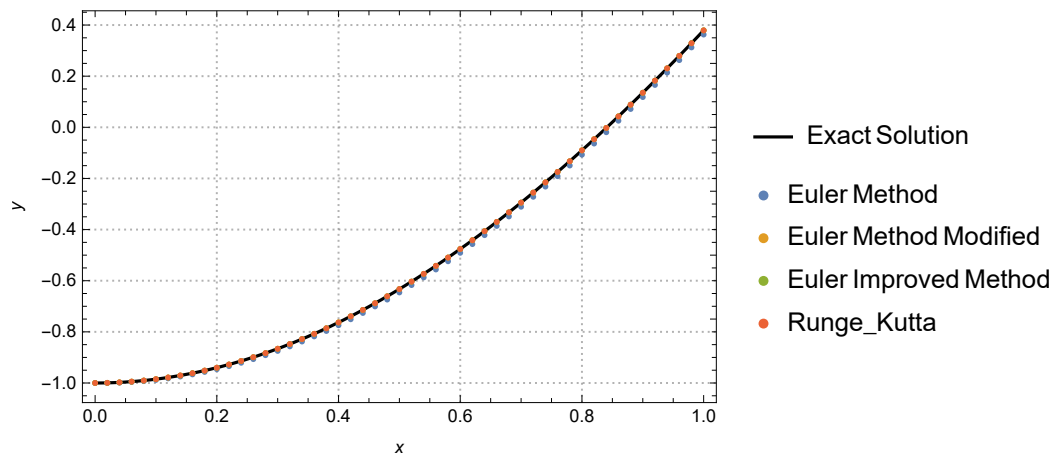
double fun(double x, double v){
    return -2 * lambda * v - pow(omega_zero, 2) * x ;
}

double fun_ex(double t){
    return upsilon_zero * exp(- t * lambda ) * sinh(t * theta) / theta;
}

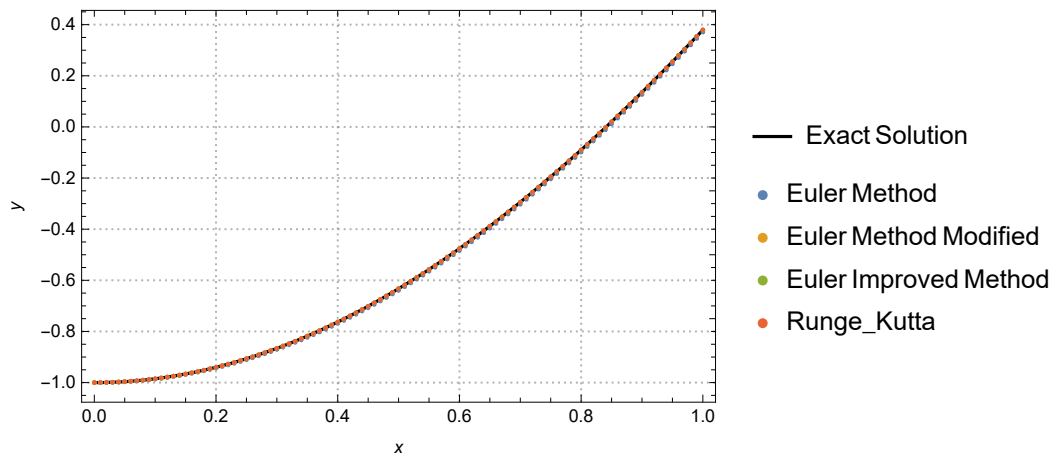
```

4. Результаты вычислений

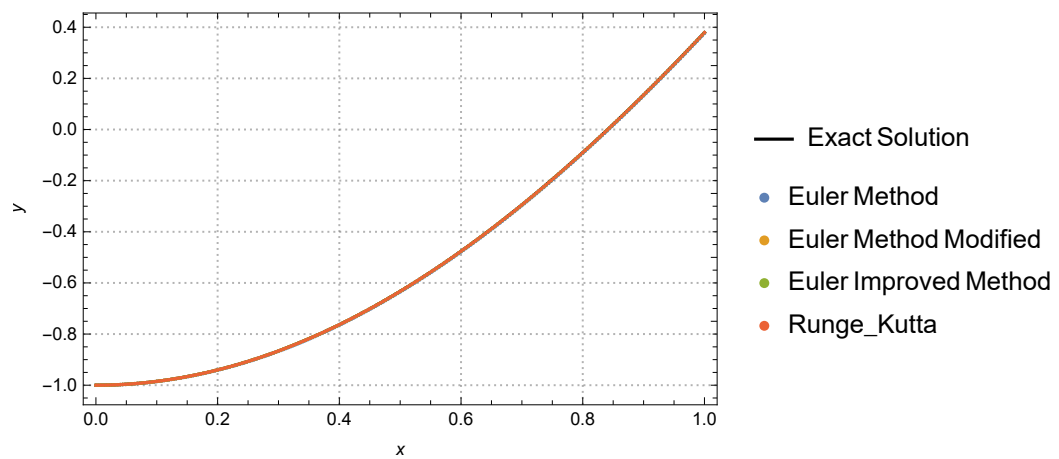
4.1. Задание А



a)



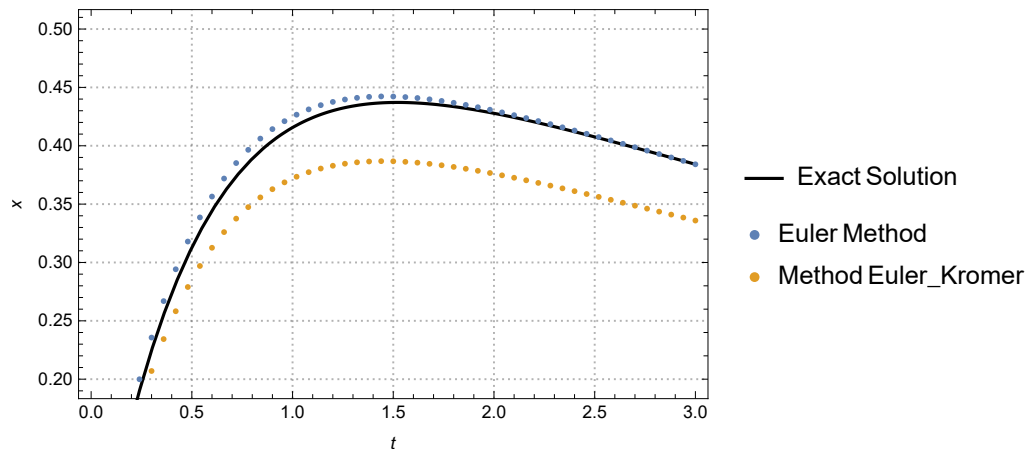
b)



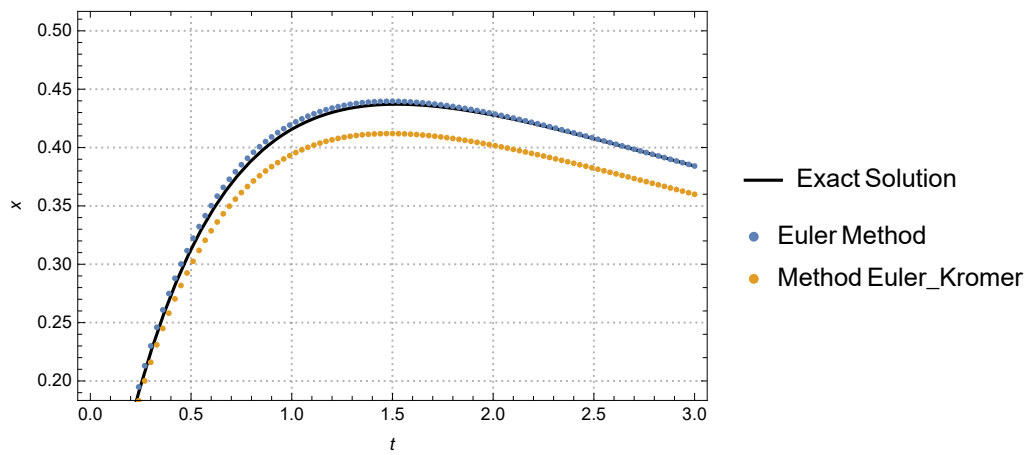
c)

Рис. 1. Результаты точного и численного решений задачи Коши для ДУ $y' = (2-y)\tan(x)$, $x \in [0, 1]$, $y(0) = -1$ стандартным, модифицированным, усовершенствованным методами Эйлера и методом Рунге-Кутты четвертого порядка для а) $N = 50$, б) $N = 100$, в) $N = 250$.

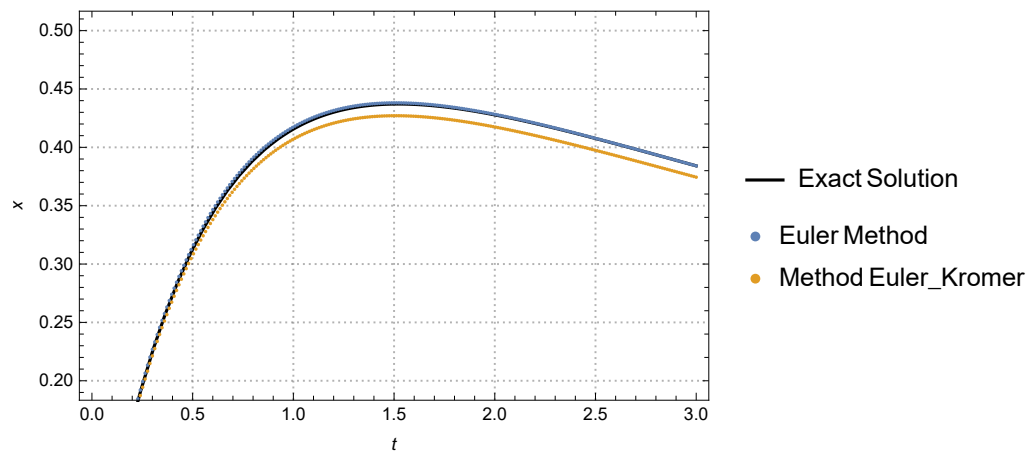
4.2. Задание Б



a)

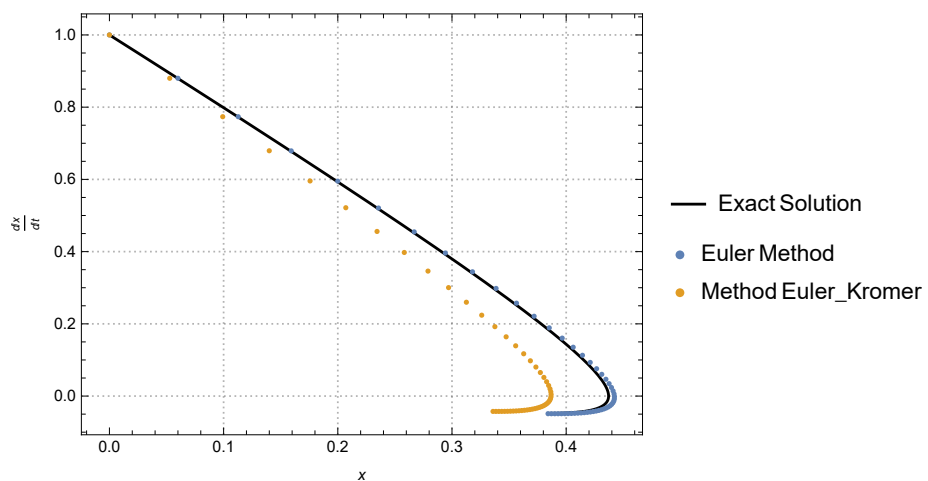


b)

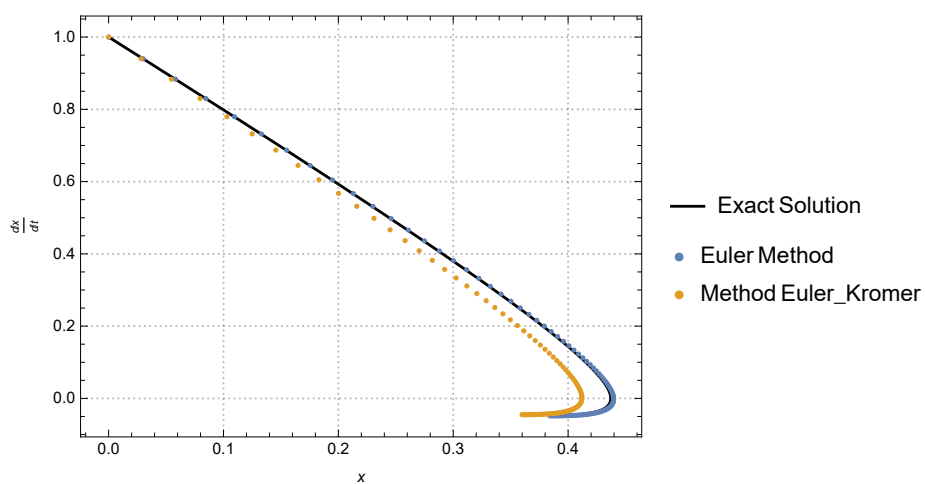


c)

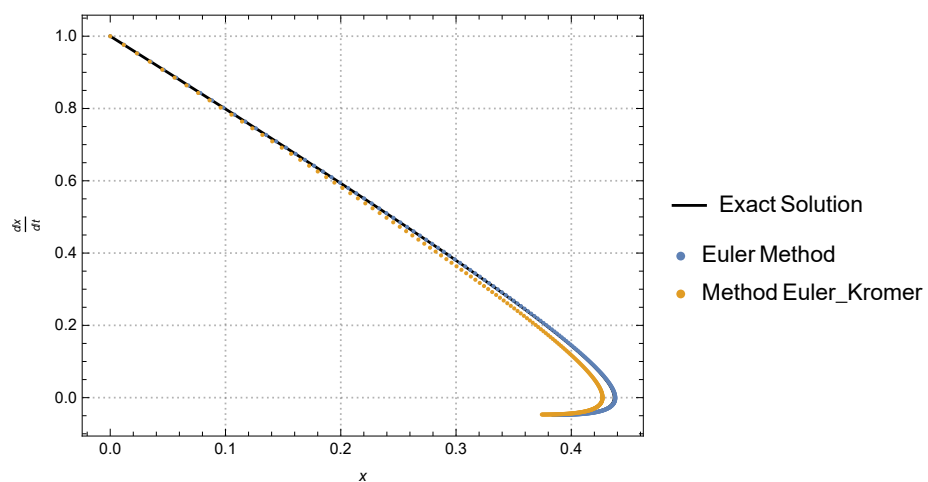
Рис. 2. Результаты точного и численного решений задачи об аperiodическом колебательном движении: $\ddot{x} + 2\dot{x} + \frac{x}{4} = 0$, $0 < t < 3$, $x(0) = 0$, $\dot{x}(0) = 1$ методами Эйлера и Эйлера-Кромера для а) $N = 50$, б) $N = 100$, в) $N = 250$.



a)



b)



c)

Рис. 3. Фазовые портреты для точного и численного решений при а) $N = 50$, б) $N = 100$, в) $N = 250$.

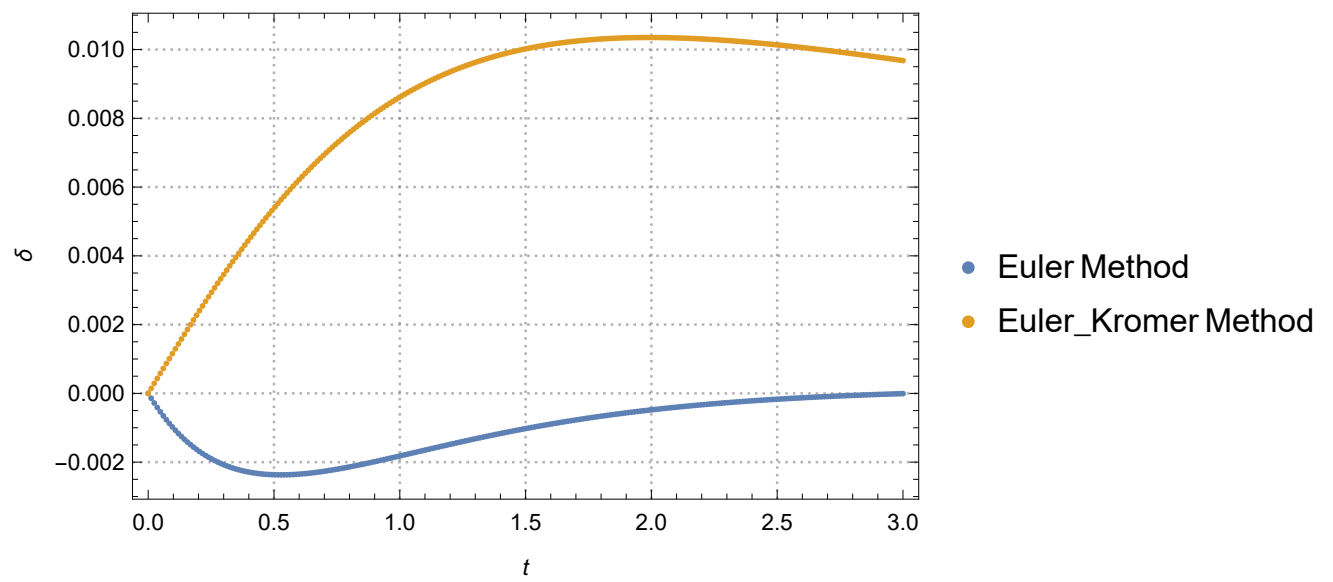
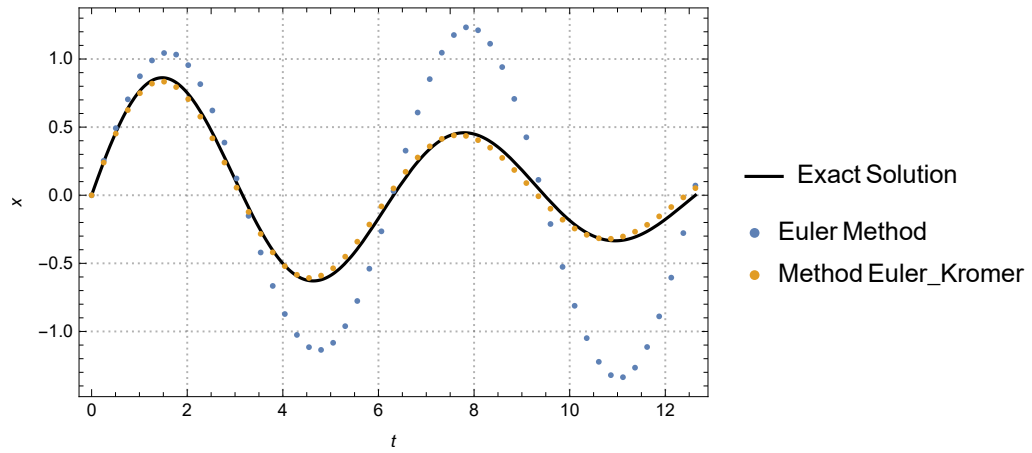
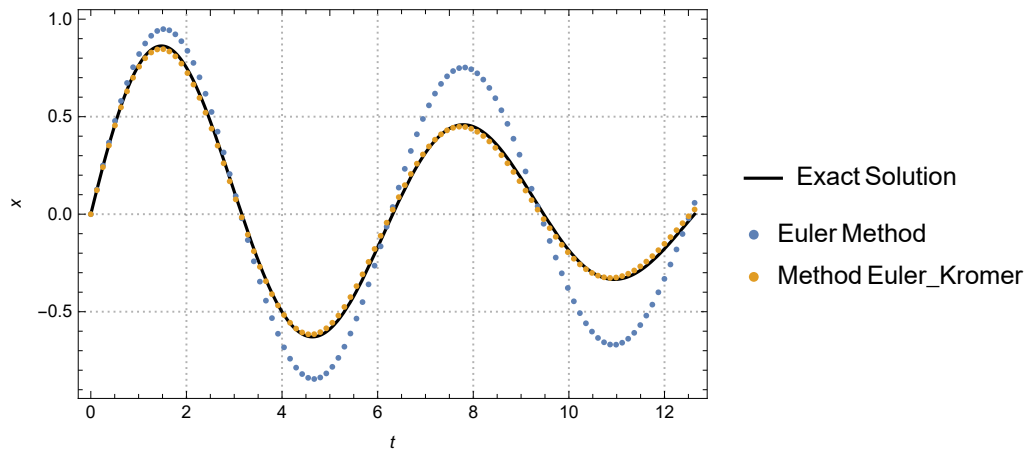


Рис. 4. Погрешность вычисления при $N = 250$.

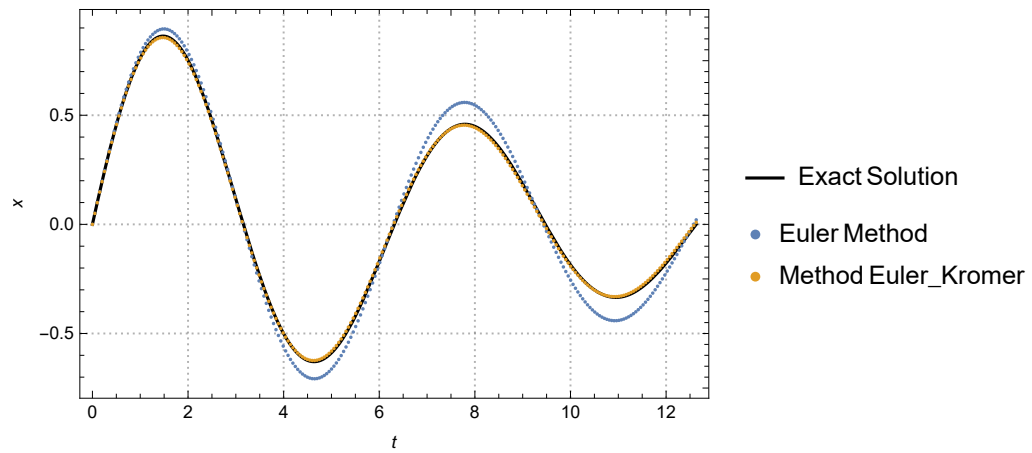
4.3. Задание С



a)

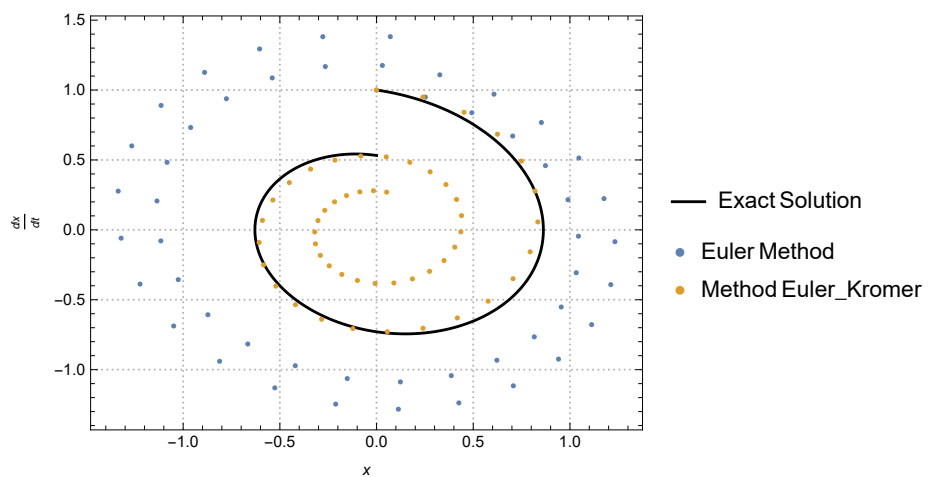


b)

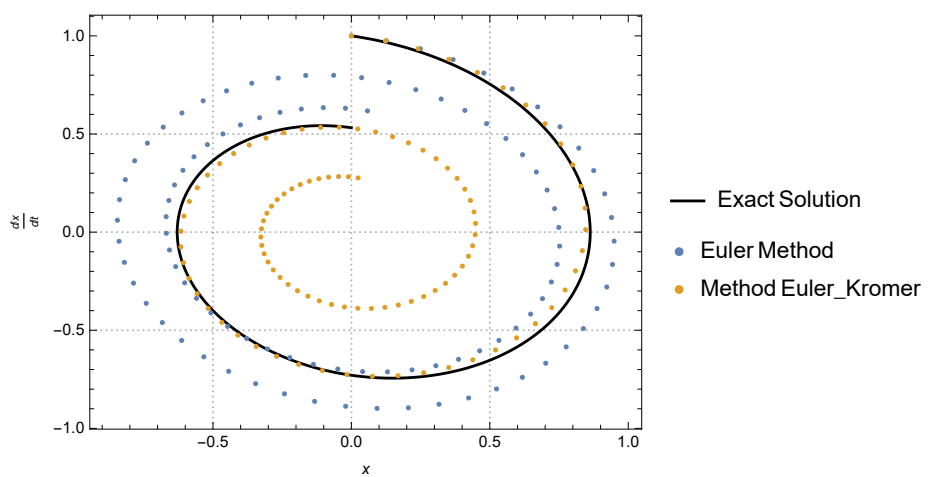


c)

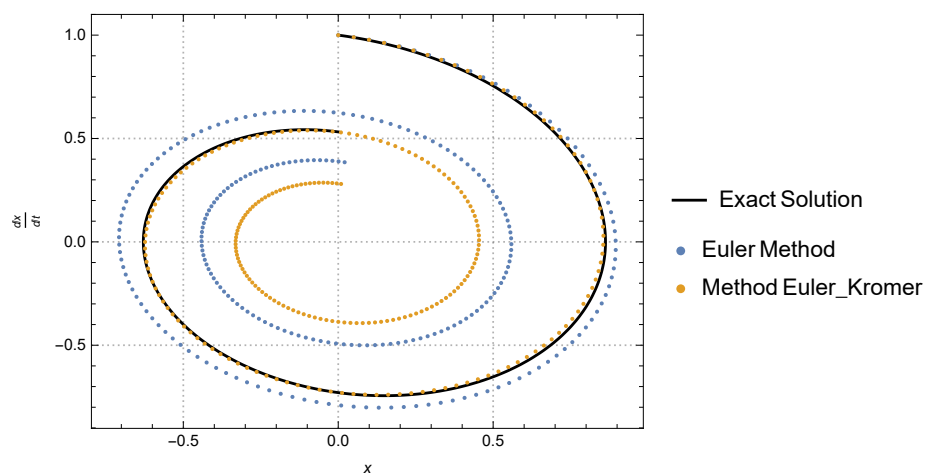
Рис. 5. Результаты точного и численного решений задачи о свободных затухающих колебаниях: $\ddot{x} + 0.2\dot{x} + x = 0$, $0 < t < \frac{2\pi}{3\sqrt{11}}$, $x(0) = 0$, $\dot{x}(0) = 1$ методами Эйлера и Эйлера-Кромера для а) $N = 50$, б) $N = 100$, в) $N = 250$.



a)



b)



c)

Рис. 6. Фазовые портреты для точного и численного решений при а) $N = 50$, б) $N = 100$, в) $N = 250$.

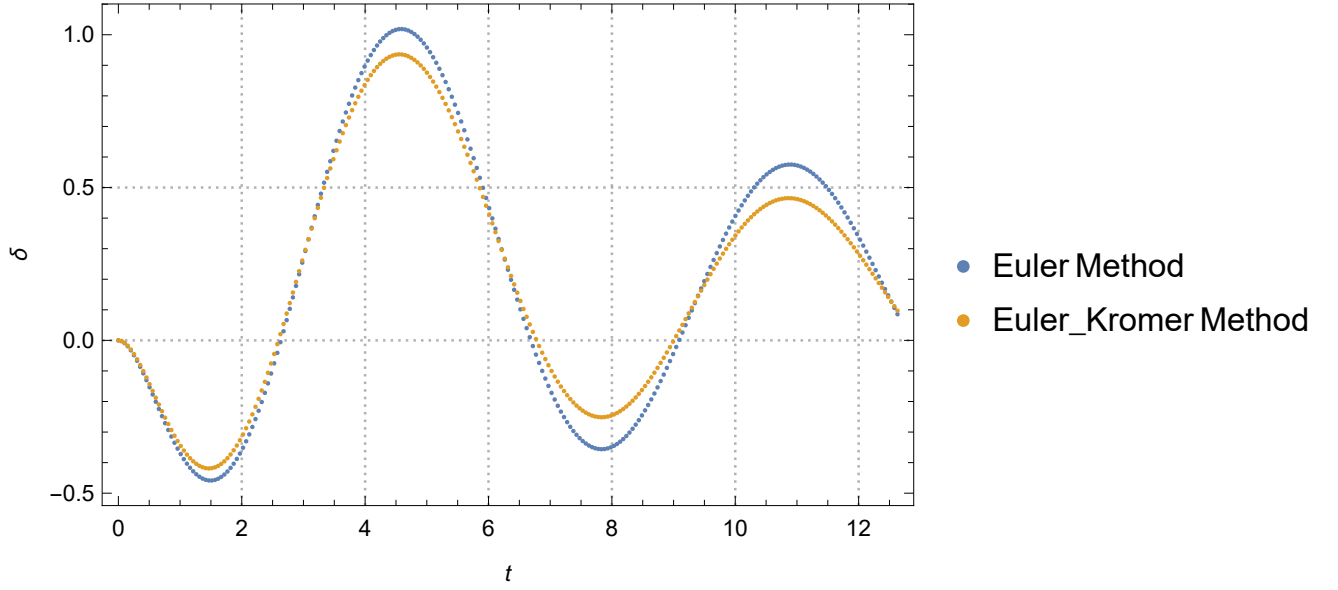


Рис. 7. Погрешность вычисления при $N = 250$.

Оптимальный выбор метода решения задач на колебательные процессы

Рассмотрим математический маятник, совершающий малые колебания в поле силы тяжести. Энергия математического маятника: $E = \frac{1}{2}ml^2\omega^2 + mgl(1 - \cos(\theta)) \underset{\theta \ll 1}{\approx} \frac{1}{2}ml^2 \left(\omega^2 + \frac{g}{l}\theta^2 \right)$.

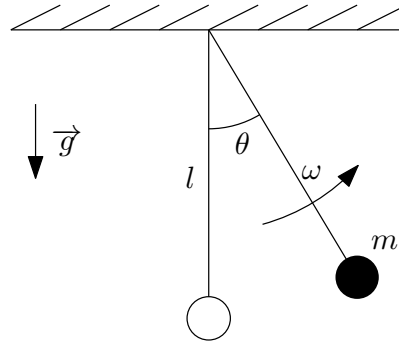


Рис. 8. Математический маятник.

Энергия в момент времени t_{i+1} :

$$E_{i+1} = \frac{1}{2}ml^2 \left(\omega_{i+1}^2 + \frac{g}{l}\theta_{i+1}^2 \right).$$

В случае использования метода Эйлера:

$$\begin{cases} \omega_{i+1} = \omega_i - \frac{g}{l}\theta_i\tau, \\ \theta_{i+1} = \theta_i + \omega_i\tau \end{cases} \Rightarrow E_{i+1} = E_i + \frac{mgl}{2} \left(\frac{g}{l}\theta_i^2 + \omega_i^2 \right) \tau^2. \quad (24)$$

Второе слагаемое выражения (24) всегда положительно, следовательно, энергия колебаний будет увеличиваться с каждым шагом независимо от его величины. В случае использования метода Эй-

лера-Кромера:

$$\begin{cases} \omega_{i+1} = \omega_i - \frac{g}{l}\theta_i\tau, \\ \theta_{i+1} = \theta_i + \omega_{i+1}\tau \end{cases} \Rightarrow E_{i+1} = \frac{ml^2}{2} \left[\theta_i^2 + 2\theta_i \left(\omega_i - \frac{g}{l}\theta_i\tau \right) \tau + \left(\omega_i - \frac{g}{l}\theta_i\tau \right)^2 \tau^2 \right].$$

Пренебрегая величинами более второго порядка малости по τ , получим:

$$E_{i+1} = E_i + \frac{g}{l} \left(\omega_i^2 - \frac{g}{l}\theta_i^2 \right) \tau^2. \quad (25)$$

В уравнение (25) также как и в (24) на каждом шаге прибавляется величина порядка τ^2 , однако, при ближайшем рассмотрении нетрудно заметить, что первый член в скобках во втором слагаемом в выражении (25) определяет кинетическую энергию, а второй - потенциальную энергию на данном шаге. При суммировании по всем шагам эти слагаемые сокращаются, следовательно, отсутствует вклад величин второго порядка по τ и амплитуда колебаний остаётся неизменной на протяжении всего моделирования. Таким образом, выбор метода Эйлера-Кромера более целесообразный при решении ДУ для консервативных колебательных систем.

5. Вывод

Получено численное решение задачи Коши для ДУ $y' = (2 - y) \tan(x)$, $x \in [0, 1]$, $y(0) = -1$ стандартным, модифицированным, усовершенствованным методами Эйлера и методом Рунге-Кутты четвёртого порядка. Методами Эйлера и Эйлера-Кромера численно решена задача об апериодическом колебательном движении: $\ddot{x} + 2\dot{x} + \frac{x}{4} = 0$, $0 < t < 3$, $x(0) = 0$, $\dot{x}(0) = 1$, также построены фазовые портреты для точного и численного решений и график погрешности $\delta_n = x(t_n) - x_n$. Результаты получены для $N = 50, 100, 250$.

Метод Рунге-Кутты четвёртого порядка показал наибольшую точность при решении ДУ для любого N по сравнению с остальными одношаговыми методами, использованными в данной работе.

Было показано, что метод Эйлера-Кромера имеет большую эффективность, чем метод Эйлера, при рассмотрении консервативных колебательных систем. Также он показал большую точность при решении задачи о свободных затухающих колебаниях.